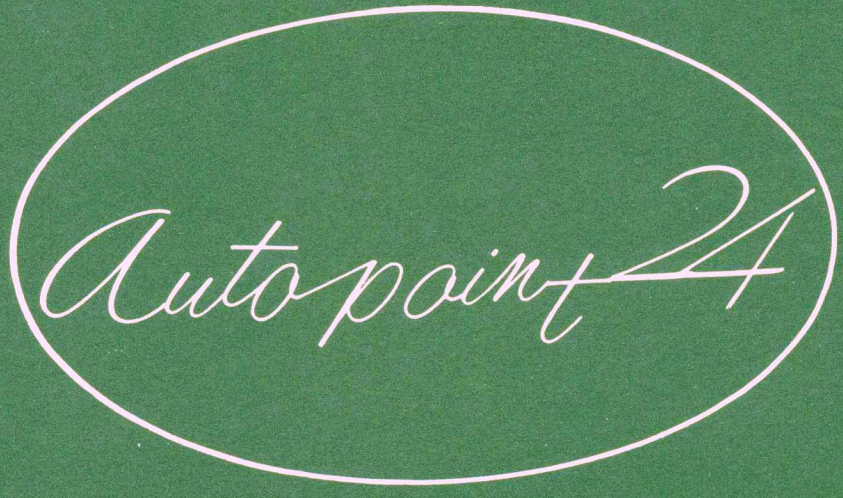

Autopoint 24

AUTOPOINT 24

PROGRAMMING SYSTEM

FOR THE BENDIX G-15

GENERAL PURPOSE

DIGITAL COMPUTER

ACKNOWLEDGEMENT

The Autopoint 24 programming system for the Bendix G-15 Computer was prepared and developed by Dr. Arthur L. Squyres and his staff of the Eastern Laboratories, E. I. du Pont de Nemours and Co., Gibbstown, New Jersey.

Dr. Squyres first presented the paper on Autopoint 24 at the Conference of the Bendix Users' Exchange Group in Palo Alto, California, in September 1959.

The manual of instructions presented here is the result of interest generated in Autopoint 24 at the Users' Exchange Group. We wish to thank Dr. Squyres for his assistance in the preparation of this manual.

TABLE OF CONTENTS

AUTOPOINT 24
PROGRAMMING SYSTEM

Autopoint 24 is a fixed-point, double precision, machine language, programming system. The system simplifies the preparation of a program and eliminates many of the troublesome details of scaling and machine language coding. However, speed of computation is that of a fixed-point, machine language program.

As Autopoint is a double precision system, output is in the form of 14 decimal digits and sign. Input data consists of the sign and 14 decimal digits or less. Leading and trailing zeros need not be typed during input. A number may have a total of 14 decimal digits of which as many as 7 may be to the left of the decimal point or as many as 10 may be to the right of the decimal point. For instance, numerical values might be:

6984312.5703684
5036.9758640125
12.5
.8642357918
683.52

Data may range from less than 10^{7} to greater than 10^{-7}.

The G-15 computer has 58 binary digits in a double precision word. The binary point precedes the binary digit at the extreme left of the binary number. Autopoint moves the binary point 24 bit positions to the right of the G-15 internal binary point, that is, Autopoint scales numeric data to 2^{24}. As Autopoint automatically shifts the binary point during input-output and computation, the system relieves the programmer of the tedious task of scaling.

Command Structure

Autopoint commands are machine language commands and have the form of the commands described in the "Coding Manual for the Bendix G-15 General Purpose Computer."

Autopoint commands govern input-output, arithmetic and transfer operations. As an added facility, the commands of the Coding Manual may be used in a program written for Autopoint.

Memory

Line 01 is the active command line for Autopoint 24. Programs are stored in a long line, 06 through 14, and must be copied into line 01 for execution. Lines 00, 02, 03, 04, and 05 are not available to the programmer. Lines 15 16, and 17 are not available during program preparation and entrance. Line 18 is used during input and output.

The short lines, two-word registers and the one-word register AR may be used for program storage. However, these may be also used by the Autopoint subroutines. The use of the short lines and registers will be discussed in the individual routines. The one-word register when addressed as 29 may be used as a source of zero.

Autopoint Routine

The Autopoint system consists of two sections, a service routine and a set of subroutines. The service routine is used during program preparation and the subroutines during computation.

The service routine converts commands from the form used in coding to the binary form used internally in the computer and occupies lines 05, 15, 16, and 17 in the computer memory. If the service routine is not needed, lines 15, 16, and 17 may be used by the programmer.

The subroutines occupy lines 00, 02, 03, and 04 and are for the following operations:

Multiplication, division, and square root
Fixed-to-floating and floating-to-fixed point conversion
Data input and output, via typewriter and paper tape.

Autopoint 24 Intercom 1000 Compatibility

An additional set of subroutines is available for evaluating functions not included in the basic Autopoint routine. Each appendix subroutine to be used must be stored in one of the available lines from 06 through 14. The appendix subroutines are:

Exponential Routine for calculation of 2^x, e^x, and 10^x;

Logarithmic Routine for calculation of $\log_2 x$, $\log_e x$, $\log_{10} x$;

Sine and Cosine Routine to find x where the angle is expressed in either degrees or radians;

Arctangent Routine where x is found in either degrees or radians.

Appendix Subroutines

The fixed-to-floating and floating-to-fixed point conversion subroutines provide data compatibility between Autopoint and other G-15 double precision, programming systems. The fixed-to-floating point conversion routine permits the use of data found by an Autopoint program with a program written in another G-15 programming system. Similarly the floating-to-fixed point conversion routine permits the use of data found by means of a program written in a G-15 double precision routine with an Autopoint program. For instance, the results of a program written in Intercom 1000, double precision may be used with Autopoint and the results of computation using an Autopoint program may be used with an Intercom 1000, double precision program.

Autopoint 24 Reference Manual

The programmer can write a program in Autopoint from the information contained in this manual and the G-15 Coding Manual. The programmer needs only this manual to operate his program. Another manual is available which contains detailed technical information and coding sheets for each section of the Autopoint system.

The reference manual is of interest to programmers who may wish to modify certain sections of the Autopoint routine.

The manual contains a complete description and coding sheets for a general purpose loading routine which is not a part of the basic Autopoint tape.

The reference manual, Users' Project No. 196, may be obtained upon request.

COMMAND LIST

	T or L_k	N	C	S	D
Transfer Double Precision Number between Two Addresses	T_{even}	N	4	S	D
Transfer Double Precision Number to or from Two-word Register	T_{even}	N	5	S	D
Clear and Add to PN	T_{even}	N	5	S	26
Add to PN	T_{even}	N	5	S	30
Subtract from PN	T_{even}	N	7	S	30
Clear and Add Absolute Value to PN	T_{even}	N	4	S	26
Add Absolute Value to PN	T_{even}	N	6	S	30
Store Sum or Difference from PN	T_{even}	N	5	26	D
Prepare Product in PN for Addition or Subtraction	T_{even}	N	4	26	26
Load Multiplicand or Denominator	T_{even}	N	4	S	25
Load Multiplier	T_{even}	N	4	S	24
Load Numerator	T_{even}	N	4	S	26
Multiply	wT	44	0	21	31
Store PN	T_{even}	N	4	26	D
Divide	wT	48	0	21	31
Store MQ	T_{even}	N	4	24	D
Take Square Root	wT	97	0	21	31
Store ID	T_{even}	N	4	25	D
Test if Sign of AR Negative	L_2	N	0	22	31
Test for Overflow	L_2	N	0	29	31
Transfer Line S to Line 01	uL_1	N	0	S	01
Halt and Permit Manual Control	00	00	0	21	31
Permit Type-in	wT	99	3	21	31
Convert Fixed-point Number Read from Paper Tape	wT	24	3	21	31
Type-out and Tab	wT	06	2	21	31
Type-out and Carriage Return	wT	05	2	21	31
Punch Line 18 on Tape	wT	76	0	21	31
Convert Floating-point Number to Fixed-point Number	wT	68	0	21	31
Convert Fixed-point Number to Floating-point Number	wT	69	0	21	31

DESCRIPTION OF AUTOPOINT COMMANDS

Commands having a C value of 4 or greater operate on double precision numbers. Therefore, the T of the command code must be even and the command operates for two-word times.

Information Transfer

Transfer Double Precision Number between Two Addresses T_{even} N 4 S D

Transfer the information in words T and T+1 of line S to words T and T+1 in line D.

The command does not destroy the contents of S. T. and S. T+1. The destination must not be a two-word register.

If the transfer is between a short line and a long line, the word-number of the short line may be determined by dividing the word-number in the long line by 4; the remainder is the word-number in the short line.

Transfer Double Precision Number to or from Two-word Register T_{even} N 5 S D

Transfer contents of words T and T+1 of line S to words T and T+1 of line D. Enter the sign into the least significant bit position of the even-word. If the number is negative, subtract the number from zero and transfer the result to D. T and D. T+1.

The command does not destroy the contents of S. T and S. T+1.

Arithmetic Operations

Clear and Add to PN T_{even} N 5 S 26

Clear two-word register PN to zero and add to PN the contents of S. T. and S. T+1. Copy the sign of S. T and S. T+1 into the least significant bit position of the even-word of PN. If the number is negative, subtract the number from zero and copy the results to PN.

The command is used in preparation for double precision addition or subtraction and does not destroy the contents of S. T. and S. T+1.

Add to PN T_{even} N 5 S 30

Add the contents of S. T. and S. T+1 to the contents of the PN register. If the number is negative, subtract the number from zero and copy the results to PN.

The programmer should test for overflow if the condition is likely to occur. The command does not destroy the contents of S. T. and S. T+1.

Subtract from PN T_{even} N 7 S 30

Subtract the double precision number in S. T and S. T+1 from the contents of the PN register.

The programmer should test for overflow if the condition is likely to occur. The command does not destroy the contents of S. T. and S. T+1.

Clear and Add Absolute Value to PN T_{even} N 4 S 26

Clear the PN register and add the absolute value of the number in S. T. and S. T.+1 into the PN register. Enter the sign bit into the special sign register associated with the two-word registers.

Add Absolute Value to PN T_{even} N 6 S 30

Add the absolute value of the contents of S. T. and S. T.+1 to the contents of the PN register.

The programmer should test for overflow if the condition is likely to occur. The command does not destroy the contents of S. T. and S. T.+1.

Store Sum or Difference from PN T_{even} N 5 26 D

Transfer the results of addition or subtraction in PN to address D. T. and D. T.+1.

Load Multiplicand or Denominator T_{even} N 4 S 25

Copy the contents of words S. T. and S. T.+1 into the ID register. Enter the sign into the special sign register associated with the two-word registers.

The number transferred to the ID register serves as the multiplicand in multiplication, the numerator in division, and the operand for the square root and the appendix subroutines. The ID register must be loaded before the MQ or PN registers and should not be loaded from the MQ or PN registers.

Example 1

Subtract B from A where B is in location 22.02, 03 and A is in location 10.02, 03. Store the difference in 22.02, 03.

L	P	T	N	C	S	D	BP	NOTES
00		02	04	5	10	26		Clear and add to PN. As the command operates for two-word times, 02 and 03, the next available word time is 04.
04		06	08	7	22	30		Subtract from PN. To obtain the word number of the short line, divide T by 4 and the remainder is the word-time of the short line. As the command operates for two-word times, 08 is the next available word-time.
08		10	12	5	26	22		Store sum or difference from PN.

Load Multiplier T_{even} N 4 S 24

Transfer the contents of S. T. and S. T.+1 into the two-word register MQ. Combine the sign with the sign of the multiplicand.

The MQ register should not be loaded from the ID or PN registers.

Load Numerator T_{even} N 4 S 26

Copy the contents of S. T. and S. T.+1 into register PN to serve as a numerator in division. Combine the sign with the sign of the divisor to form the correct sign of the quotient.

The PN register should not be loaded from the MQ and ID registers.

Multiply wT 44 0 21 31

Multiply the contents of the two-word registers, ID and MQ, and store the product in PN. Return to the program at word-time T.

The routine resets the overflow indicator and destroys the contents of the AR register. If the product is greater than 2^{24} (a decimal value of 16,777,216), the computer rings a bell and halts and a zero characteristic appears on the control panel.

Store PN T_{even} N 4 26 D

Transfer the contents of PN and the sign from the special sign register to address D. T and D. T+1.

The location in which the contents are stored must not be a two-word register.

Prepare Product in PN for Addition or Subtraction T_{even} N 4 26 26

Copy the sign from the special sign register to the least significant bit position of the PN register. If the number is negative, subtract the number from zero and copy the results to PN.

Divide wT 48 0 21 31

Divide the contents of the PN register by the contents of the ID register and store the result in the MQ register. Return to word-time T in the program.

The routine destroys the contents of the AR register and resets the overflow indicator. If the quotient is greater than 2^{24} (a decimal value of 16,777,216), the computer rings a bell and halts and a zero characteristic appears on the control panel.

Store MQ T_{even} N 4 24 D

Transfer the contents of the MQ register and the sign from the special sign register into address D. T and D. T+1.

The location in which the contents are stored must not be a two-word register.

Take Square Root wT 97 0 21 31

Take the square root of the contents of the ID register. Put the results in the ID register. Return to word-time T in the program.

The programmer must clear the PN register prior to executing the command. The contents of the two-word registers, the AR register, and addresses 21.01, 02, and 03 are destroyed by the square root command.

If the operand is negative, the computer rings a bell and halts. A one characteristic appears on the control panel. If computation is continued, the computer takes the square root of the absolute value of the operand. To continue operation, put the Compute switch off and then to GO.

Store ID T_{even} N 4 25 D

Transfer the contents of the ID register and the sign from the special sign register into address D. T and D. T+1.

The location in which the value is stored must not be a two-word register.

Example 2

Divide A by B. A is in location 22.02, 03. B is in location 10.02, 03. Store quotient in 22.00, 01

L	P	T	N	C	S	D	BP	NOTES
00		02	04	4	10	25		Load multiplicand or denominator
04		06	08	4	22	26		Load numerator.
08	w	11	48	0	21	31		Divide, Return to program at word-time 11.
11		12	14	4	24	22		Transfer MQ.

Special Commands

Transfer Line S to Line 01 uL_1 N 0 S 01

Copy Line S to Line 01. Take the next command at word-time N.

Halt and Permit Manual Control 00 00 0 21 31

The computer halts, rings a bell, and permits manual operation. The instructions to the Autopoint subroutines, listed on page 16, may now be typed.

Conditional Transfer of Control

Test if the Sign of AR is Negative L_2 N 0 22 31

Take the next command from N if AR is positive or equal to zero. Take the next command from N+1 if AR is negative.

A double precision number occupies two consecutive addresses in memory and the sign is in the least significant bit position of the even-word time. Therefore, the even-word must be transferred to AR to be tested for negative.

Test for Overflow L_2 N 0 29 31

If at anytime, since the last overflow test was made, the absolute value of the contents of the PN register exceeded 2^{24}, take the next command from N + 1. If the absolute value of the contents of PN has always remained less than 2^{24}, take the next command from N.

Overflow occurs as a result of addition or subtraction. The overflow light on the control panel of the computer goes on when overflow exists. The light remains lighted until an overflow test is made. The decimal value of 2^{24} is 16,777,216.

Example 3

Test A for negative. If negative, halt. If positive or zero, continue computation. A is in location 22.02, 03.

L	P	T	N	C	S	D	BP	NOTES
00		02	03	0	22	28		Copy 22.02 into AR. Location 02 is copied into register AR as the even-word of a double precision number contains the sign of the number. As the command is a single precision command and operates for one word-time, the next available word-time is 03.
03		05	05	0	22	31		Test if sign of AR is negative.
05		06	08	5	22	26		No — Continue computation. Clear and add to PN.
06		00	00	0	21	31		Yes — Halt.

Input-Output Operations

Permit Type-in wT 99 3 21 31

Permit type-in of fixed-point number, convert the number to binary and store in address 23.00, 01. Reconvert the number to decimal form and type-out for verification. Return to word-time T in the program.

The T part of the command must not be .04. The number entered into 23.00, 01 must be used by the next command or transferred to another memory location by the next command.

The operator types numbers in fixed-point form with the slash key ("/") representing the decimal point. The slash must be present as the subroutine distinguishes between numbers and instructions by the presence or absence of a slash. For example, to enter 12.5, type: *12/5 (tab)* Ⓢ. To enter 6.6%, type: */066 (tab)* Ⓢ. Numbers must not have more than 7 digits to the left of the decimal point; 10 digits to the right of the decimal point; nor more than 14 digits in the whole number.

If more than one slash is typed, the last one determines the location of the decimal point. Do not use lines 21 and 23, location 20.02, the two-word registers, or register AR for storage of information.

The subroutine locates the decimal point by means of dummy information stored in 23.00. If an error is made during type-in, first restore the dummy information by typing: *xxx (tab)* Ⓢ. Type the correct value.

Convert Fixed-point Number Read from Paper Tape wT 24 3 21 31

Convert the fixed-point number in the ID register to binary and locate the decimal point by means of the constant in AR. Store in 23.00, 01. Reconvert the number to decimal form and type-out for verification. Return to word-time T in the program.

The constants for locating the decimal point are:

.0000000	for numbers less than 10,000 (4 digits to the left of the decimal point)
.0200000	for numbers 10,000 or more but less than 100,000
.0400000	for numbers 100,000 or more but less than 1,000,000
.0600000	for numbers 1,000,000 or more but less than 10,000,000.

The T part of the command must not be .04. The contents of the two-word registers, the AR register, lines 21 and 23, and location 20.02 are destroyed by the command.

Type-out and Tab wT 06 2 21 31

Type-out and Carriage Return wT 05 2 21 31

Type out contents of register PN and the sign from the special sign register as a decimal number. If 06 is in the N position, move carriage to the next tab stop. If 05 is in the N position, return carriage. Return to word-time T in the program.

The contents of the two-word registers, the AR register, and line 21 are destroyed by these commands.

For numbers less than 10,000, 10 places are typed to the right of the decimal point; 9 places to the right of the decimal point are typed for numbers from 10,000 to 100,000; 8 places for numbers from 100,000 to 1,000,000; 7 places for numbers from 1,000,000 to 10,000,000.

The output routine is also available as a single precision type-out. The check sum is –.00xx002. The single precision routine types-out 7 digits and sign with 3 decimal places to the right of the decimal point for numbers less than 10,000. There are two places to the right of the decimal point for numbers from 10,000 to 100,000; one place to the right for numbers from 100,000 to 1,000,000; no places for numbers from 1,000,000 to 10,000,000.

Punch Line 18 on Tape wT 76 0 21 31

Punch Line 18 on tape and type-out the check sum. Resume computation when punching is completed. Return to word-time T in the program.

The contents of address 21.02 are destroyed by the command. The service routine in Line 05 is required by the subroutine.

Convert Floating-point Number to Fixed-point Number wT 68 0 21 31

Convert floating-point number in the ID register to a fixed-point number. Put the fixed-point number in the ID register. Return to word-time T in the program.

The two-word registers, the AR, and addresses 21.00, 01, and 20.00 are not available for storage of information.

If the binary equivalent of the fixed-point number is greater than 2^{24}, the computer rings a bell and halts and a zero characteristic appears on the control panel.

Convert Fixed-point Number to Floating-point Number wT 69 0 21 31

Convert the fixed-point number in the ID register to a floating-point number. Put the converted number in the ID register. Return to word-time T in the program.

The two-word registers, the AR, and addresses 21.00, 01 and 20.00 are not available for storage of information.

Appendix Subroutines

The appendix subroutines have certain features in common. Each subroutine must be stored in a line from 06 through 14 and copied into line 01 for execution. The operand must be in the ID register and a return command must be in the AR register. Exit from each routine is at word-time .12 and the results are in the ID register.

Exponential Routine **Check Sum: .01xx012**

Enter the routine at word-time .97 for 2^x; .98 for e^x; and .99 for 10^x. Do not use the two-word registers or line 21 for storage of information.

Twenty-four consecutive locations, .40 through .63, are available for storage of numbers or for program.

If the binary equivalent of the decimal number is greater than 2^{24}, the computer halts and rings a bell; a two characteristic appears on the control panel. To locate the address of the command being executed, follow the instructions given in the section, Error Halts.

Logarithm Routine **Check Sum: .00xx013**

Enter at word-time .97 for $\log_2 x$; .98 for $\log_e x$; and .99 for $\log_{10} x$.

Do not use the two-word registers or line 21 for storage of information.

If the operand is negative, a bell rings and the computer halts with a 3 characteristic on the control panel. If the operand is zero, a large negative number (-9999999.9999998) is put in the ID register. The logarithm routine must be transferred to line 01 each time it is used.

Sine and Cosine Routine — Check Sum: .00xx014

Enter the routine at word-time .94 for sine x where x is in radians; .95 for sine x where x is in degrees; .96 for cosine x where x is in radians; and .97 for cosine x where x is in degrees. Do not use the two-word registers or line 21 for storage.

Twenty consecutive locations, .74 through .93, are available for use. The sine-cosine routine must be transferred to line 01 each time it is used.

Arctangent Routine — Check Sum: .00xx015

Enter the program at word-time .97 to find the arctangent x in radians; and at word-time .98 to find the arctangent x in degrees. Do not use the two-word registers and line 21.

Example 4

Find $\log_2 A$. The logarithm routine is in line 12. The main program is in line 06. A is in location 22.02, 03. Store the results in 22.00, 01.

L	P	T	N	C	S	D	BP	NOTES
00		02	04	4	22	25		Load ID.
04		05	06	0	06	28		Copy the contents of word-time 05, line 06 into AR. As line 06 was copied into line 01, the active command line, the command could have been written: 05.06.0.01.28.
05	u	13	07	0	06	01		Return Command. Copy line 06 into line 01. As a block command operates for one-word time more than the number of locations to be copied and the exit from the logarithm routine is at word-time 12, the T of the return command is 13. See page 15 of the G-15 coding manual for a discussion of block commands.
06	u	07	97	0	12	01		Copy line 12 into line 01. Take next command at word 97 to evaluate $\log_2 A$.
07		08	10	4	25	22		Transfer ID. Copy $\log_2 A$ into 22.00, 01.

COMPUTER OPERATION WITH AUTOPOINT 24

The computer is assumed to be on. If the computer is off, see page 21 of the Operating Manual for the turn-on procedure.

Program Entrance and Execution

1. Place Autopoint 24 magazine on photo-reader. Be certain tape is rewound.
2. Put the Compute, Punch, and Enable switches on typewriter base to the off positions (centered).
3. Put the Enable switch ON and type: *p.** Put the Enable switch off and wait for the photo-reader light to go off.
4. Put the Compute switch to GO (or BP). Wait for photo-reader light to go off.
5. Type: *x00 (tab)* Ⓢ,** which clears line 18 to zero.
6. Type: *yCD— (tab)* Ⓢ, where CD is the word position of the first command.
7. Type program commands; after each command type: *(tab)* Ⓢ. The commands enter line 18. After a command is entered, the computer automatically types-out the number of the N position of the command. The N position is the word location where the next typed command enters line 18.

   When a code for an Autopoint subroutine is typed during program entrance, the Autopoint service routine types out the T instead of the N as the location of the next command. The reason is that the subroutine returns to the main program at word-time T.

8. To enter a sexadecimal constant into line 18, type: *zCD(tab)±FGHIJKL (tab)* Ⓢ. The number, ±FGHIJKL, in sexadecimal notation enters word-position CD of line 18. Repeat command as often as necessary.
9. Type: *x06(tab)* Ⓢ.
   The computer prepares a program tape by punching the contents of line 18 on paper tape and automatically computes and types-out a check sum. The process does not clear line 18.

*Throughout the manual, lower case letters indicate alphabetic characters which are to be typed. An upper case letter represents a numerical value supplied by the programmer.

**The Ⓢ key is located in the upper left hand corner of the alphanumeric typewriter keyboard and is used for all control operations initiated from the keyboard with the Enable switch ON.
The earlier model typewriter has no Ⓢ key. For operations calling for the Ⓢ key, use the s key of the alphabetic keyboard.

10. Type: *ABx03 (tab)* (S), where AB is the number of the line in which the information in line 18 is to be stored.
    The computer copies the contents of line 18 into line AB which should be 06 through 14 in Autopoint 24. The instruction may be given after Step 7 if desired. Repeat Steps 6 through 10 for each additional line of the program entered into the computer.
11. If single precision output is desired, put Enable switch ON, type: *p*. Return the Compute switch to GO. The single precision routine replaces the double precision routine.
    If the appendix subroutines are not needed, proceed to Step 18. If the appendix subroutines are needed, proceed to Step 13.
12. If single precision output is not wanted, put the Enable switch ON, type: *p*. Wait for photo-reader light to go out. Do NOT put Compute switch to GO.
13. To enter the appendix subroutines, with the Compute switch off and the Enable switch ON, type: *p*. Return the Compute switch to GO. The computer reads paper tape until a Stop Code appears, types-out a check sum, and returns control to the service routine with the subroutine in line 18.
14. Compare the check sum against the check sum of the routine desired. If the routine is the desired one, proceed to Step 15. If the routine is not the desired one after the comparison of the check sums, return to Step 13.
15. Type: *x06 (tab)* (S), which punches the routine on tape.
16. Type: *ABx03 (tab)* (S), which transfers the contents of line 18 to line AB.
17. Put the Enable switch ON, type: (S) *cf*. The instruction transfers control to Autopoint 24.
18. Put Compute to GO and Enable off.
19. To type-in fixed-point number, type: *ABCDy (tab)* (S). Type decimal number followed by: *(tab)* (S). The number enters line AB, words CD and CD+1. Word CD must be even and not greater than u6.
    If a minus sign is added to the instruction, the instruction will automatically be incremented by 2.
20. Type: *ABCDz (tab)* (S), which transfers line AB to line 01. Computation begins at word-time CD.

Program Error Detection and Correction

1. If a typing error is made in Step 7 of program Entrance and Execution, type: *yCD (tab)* (S), where CD is the location of the command in error. Type the correct instruction.
2. To correct an error detected while checking out the program, load Autopoint 24 as described in Steps 1 through 4 of Program Entrance and Execution.
3. Replace the Autopoint 24 package with the program tape.
4. Put the Compute switch to BP.
5. Type: *w00 (tab)* (S). The computer reads a block of tape into line 18 and types out a check sum.

6. Compare the check sum against a check sum previously obtained. If the error is in the block of tape read, proceed to Step 7. If the error is not in the block of tape, move the Compute switch to off to BP. Another block of tape will be read and the check sum will be typed-out.
7. Put the Compute switch off and the Enable switch ON, type: *f*.
8. Put the Compute switch to BP and type: *yCD (tab)* (S), where CD is the word location of the command in error. The computer automatically returns the typewriter carriage and types-out location CD and the contents of CD.
9. Type-in the correct instruction for CD.
10. Repeat Steps 8 and 9 for each command to be corrected in a block of tape.
11. Put Compute switch off, Enable switch ON, and type: (S) *c5f*.
12. Put the Compute switch to GO.
13. Type: *x06 (tab)* (S), which punches line 18 on tape.
14. If there are errors present in other blocks of tape, repeat Steps 5 through 13.
15. If no error is present in the following blocks of tape, put the Compute switch to GO and type: *w02 (tab)* (S). The computer reads tape, types-out the check sum of each block, and punches line 18 on tape.
16. When the computer finishes reading and punching, put the Compute switch off, the Enable switch ON, and type: (S). Proceed to Step 7 of Execution from Program Tape.
17. If no error is present in the following block of tape, but errors are present in other blocks, put the Compute switch to BP and type: *w02 (tab)* (S). The computer reads tape, types-out the check sum of the block of tape, and halts. Put the compute switch to the off position and then back to BP. The computer now punches the block in line 18 on tape, reads another block of tape, and types out its check sum. If there is an error present, return to Step 7.

    When each block of tape is read, corrected, and punched on tape, proceed to Step 7 of Execution from Program Tape.

Error Halts

During execution of an Autopoint subroutine, conditions may occur which will cause an error in computation. A method exists by which the programmer may determine the nature and location of the error.

When an error occurs, the computer rings a bell and halts. The characteristic, which appears on the control panel neons marked "Characteristic," determines the nature of the halt. Table I lists the characteristic, the control panel configuration, and meaning of the halt.

To locate the address of the command where the halt occurs, execute one more command from the keyboard. With the Enable switch ON, type: *i,* which executes the return command, and then type: *ta,* which types-out the word-time of the next command.

Characteristic	*Control Panel Neons*	*Meaning*
0	○ ○	Overflow in multiplication, division, or floating-to-fixed point conversion routines.
1	○ ●	Square root of a negative number. The control panel "P sign" neon goes on.
2	● ○	Overflow in the exponential routine.
3	● ●	The operand for the logarithm routine is negative. The control panel "P sign" neon goes on.

TABLE I

Execution from Program Tape

1. Place Autopoint 24 magazine on the photo-reader. Be certain that the tape is rewound.
2. Put the Compute, Enable and Punch switches on the base of the typewriter to the off positions (centered).
3. Put the Enable switch ON and type: *p*. Wait for the photo-reader light to go off and put the Enable switch off.
4. Put Compute switch to GO (or BP) and wait for photo-reader light to go off.
5. To enter single precision routine, put the Enable switch ON and type: *p*. Wait for the photo-reader light to go out.
6. Type: Ⓢ *c5f*.
7. Replace the Autopoint magazine with the pre-punched program tape.
8. Put the Compute switch to BP.
9. Type: *w00 (tab)* Ⓢ. The computer reads paper tape until a Stop Code appears on the tape; copies the information into line 18; and types-out a check sum of the information.
10. Put the Enable switch ON, Compute off and type: *f*.
11. Put the Compute switch to BP.
12. Type: *ABx03 (tab)* Ⓢ which transfers the contents of line 18 to line AB. In Autopoint 24 line AB should be one of the long lines from 06 through 14.
13. Repeat Steps 6 through 12 for each additional block of tape to be read.
14. Put the Compute switch off, the Enable switch ON, and type: Ⓢ *cf*, which transfers control to Autopoint 24.
15. Put Compute switch to GO; type: *ABCDz (tab)* Ⓢ, which transfers line AB to line 01, the active command line. Computation begins at word-time CD.

INSTRUCTION CODES TO SERVICE ROUTINES

w 00	Read tape, type-out check sum.
w 01	Read tape, type-out check sum, type-out information read.
w 02	Read tape, type-out check sum, punch information read.
w 03	Read tape, type-out check sum, type-out and punch information read. (Punch switch must be ON after check sum type-out.)
x 00	Clear Line 18.
x 01	Type-out check sum, punch number track.
ABCD x 02–	Line AB to Line 18, type-out word CD as a standard decimal command.
ABCD x 02	Line AB to Line 18, type-out word CD in sexadecimal form.
AB x 03	Line 18 to Line AB.
x 05	Type-out check sum, type-out contents of Line 18.
x 06	Type-out check sum, punch contents of Line 18.
x 07	Type-out check sum, type-out and punch contents of Line 18. (Punch switch must be ON after check sum type-out.)
y CD–	Decimal command input into word CD of Line 18, type-out previous contents of CD as a standard decimal command.
y CD	Decimal command input into word CD of Line 18, type-out previous contents of CD in sexadecimal form.
z CD(tab)± FGHIJKL	Sexadecimal number input into word CD of Line 18.

INSTRUCTION CODES TO SUBROUTINES

ABCD x	Type-out number in word CD and CD+1 of line AB.
ABCD x–	Type-out number in word CD and CD+1 of line AB. Continue operation until ABu6 is reached.
ABCD y	Decimal number input into word CD and CD+1 of line AB.
ABCD y–	Decimal number input into word CD and CD+1 of line AB. Continue operation until ABu6 is reached.
ABCD z	Transfer line AB to line 01 and begin execution of program at word-time CD.

All instructions followed by: *(tab)* Ⓢ.

COMPUTER INSTRUCTION DURING OPERATION

The instruction codes discussed in the accompanying sections are applicable when the contents of the Autopoint package have been loaded into the computer.
The first section covers instructions which are performed by the service routine.

The second section deals with instructions which are performed by the Autopoint subroutines.

Instructions to Service Routines

Before typing the instruction codes the control panel neons must be in the following configuration:

Command Line	● 0 ●

Source	● ● ● 0 0
Destination	● ● ● ● ●
Input-Output	0 ● ● 0 0

To obtain the neon pattern:

1. Put the Compute and Punch switches off, the Enable switch ON, and type: Ⓢ *c5f*.
2. Put Enable switch off and Compute switch to GO (or BP).

If the Compute switch is at BP, the computer halts after a check sum is typed-out. To resume operation, move the Compute switch to the center (off) position and back to BP. If the Compute switch is at GO, the computer does not halt after a check sum is typed.

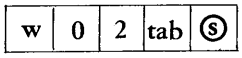
Instruction codes which begin with the letter "w" continue to operate on additional blocks of tape. If the Compute switch is at GO, the operation continues on successive blocks of tape. If the Compute switch is at BP, the computer halts after one block of tape is read. The operation proceeds to the next block of tape by putting the Compute switch off and back to BP.

w	0	0	tab	Ⓢ

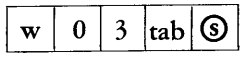
Read paper tape until a Stop code appears on the tape and copy information read into line 18. Type-out a check sum of the information on tape.

w	0	1	tab	Ⓢ

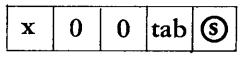
Read paper tape until a Stop code appears on the tape and copy information into line 18. Type-out check sum. Type-out information read.



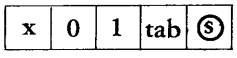
Read paper tape until a Stop code appears on tape and copy information read into line 18. Type-out check sum. Punch information read on new tape.



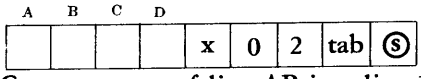
Read paper tape until a Stop code appears on tape and copy information read into line 18. Type-out check sum. Type-out information read and punch out new tape. (Put punch switch ON after type-out of check sum.)



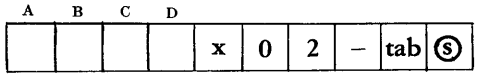
Clear line 18 to zero.



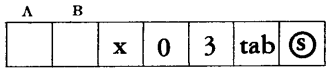
Copy number track into line 18. Type-out check sum. Punch number track on tape.



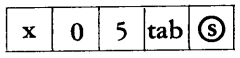
Copy contents of line AB into line 18. Type-out contents of word CD in sexadecimal notation.



Copy contents of line AB into line 18. Type-out contents of word CD in the form of a standard command.

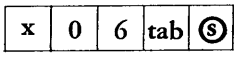


Copy contents of line 18 into line AB. In Autopoint 24, AB must be 06 through 14.

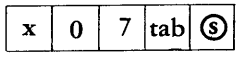


Type-out check sum of line 18. Type-out contents of line 18 starting at the highest numbered location in the line.

To obtain check sum only, put the Compute switch to BP; type: *x05 (tab)* Ⓢ. After the check sum is typed-out, put the Compute switch off, the Enable switch ON, and type: *f*. Move the Compute switch to GO (or BP).

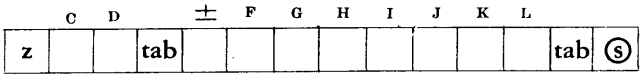


Type-out check sum of line 18. Punch contents of line 18 on tape starting at the highest numbered location in the line.



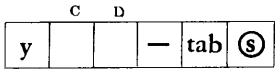
Type-out check sum of line 18. Type-out and punch contents of line 18 on tape, starting at the highest numbered location in the line.

A leader is punched after which the computer halts. The Punch switch must then be put to the ON position. The contents of line 18 are punched and typed-out. When punching is complete, the Punch switch must be put to the off position.



Enter the number "± FGHIJKL" in sexadecimal notation into word CD of line 18.

After the instruction is typed, there is an automatic carriage return and location CD is typed-out. If the location had information, the former contents are typed-out.



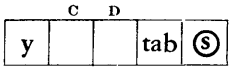
Permit commands in standard decimal form to be typed into line 18. Put the first command typed into word-position CD of line 18. Put each following command into the word-position corresponding to the number in the N position of the immediately preceding command.

After this instruction is typed, there is an automatic carriage return and location CD is typed-out. The first command may then be typed-in. Automatically, the computer returns the carriage and types the number in the N position of the command. The next command may then be typed followed by: *(tab)* Ⓢ. The next command enters word position N.

If a location in which a command is to be stored is not empty, its contents are typed-out as well as the location itself. Inadvertent replacement of a stored word may thus be avoided. The type-out is in standard decimal command form.

If a command of the form "T N C 28 31" is typed, the number N+1, instead of N will be automatically typed-out; the next command typed then is entered into word position N+1.

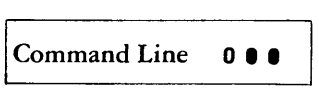
If a command of the form "wT N C 21 31" is typed, the number T is automatically typed-out instead of the N as the location of the next command. If the w prefix is omitted, the number typed is T-1.

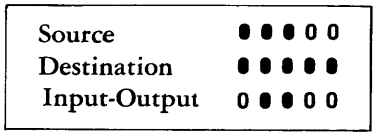


The instruction may be typed instead of: *yCD-(tab)* Ⓢ. Operation is the same but the original contents of a location are typed-out in sexadecimal form rather than standard decimal command form.

Instructions to Subroutines

The control panel neons must be in configuration listed below before typing the instruction codes.

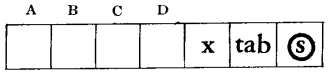




To obtain the neon pattern:

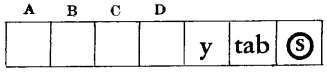
1. Put Compute and Punch switches off, Enable switch ON, and type: Ⓢ *cf.*

2. Put the Enable switch off and the Compute switch to GO (or BP).



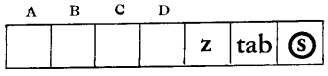
Type-out the number in CD and CD+1 of line AB. Word CD must be even and less than u6. Line AB must not be less than 06.

By adding a minus sign to the instruction, that is: *ABCDx— (tab)* Ⓢ, the instruction is automatically incremented by 2 and repeated until ABu6 is reached. To terminate the listing before word u6 is reached, put the Compute switch to BP and type: *f*, with the Enable switch ON.



Permit type-in of fixed-point number to words CD and CD+1 of line AB. Word CD must be even and less than u6. Line AB must not be less than 06. Type-in fixed-point number with the slash for the decimal point. The number must not have more than 7 places to the left of the decimal point nor have more than 10 places to the right of the decimal point.

By adding a minus sign to the instruction, that is: *ABCDy— (tab)* Ⓢ, the instruction is automatically incremented by 2 and repeated until ABu6 is reached.



Transfer line AB to line 01 and begin execution of program at word-time CD. Word CD must not be greater than u7. Line AB must not be less than 06. The command does not destroy the contents of line AB.

To re-enter service routine after execution of the above commands, type: Ⓢ *c5f.*

Index Register

Autopoint has no index register subroutine. However, a method exists for modifying addresses of commands and for controlling the exit from loops. As the programmer may adapt the method to the requirements of any particular program, the method is faster than any general purpose index register subroutine.

Intercom programmers are familiar with base, difference, and limit registers, respectively designated B, D, and L, to modify either a line or a word in a line. The B register modifies an address; the D register increments the B register; and the L register limits the value of the B register. In Autopoint the B, D, and L registers are any memory locations assigned by the programmer. A different set of registers must be used to modify a line or a word.

By repetitively changing the word base during computation, the same command may operate on the contents of different addresses. The programmer may therefore store in memory many values of a given variable. Initially the B register is set to zero. To modify a command, the program copies the command into the AR register and adds the B register to the command. The modified command is executed from the AR register and computation continued. The contents of the B register are then incremented by the contents of the D register and the results stored as a new value in the B register. The B register is then tested against the L register. The results of the comparison determine whether the program returns through the loop or continues computation. The L register must be equal to n x D where n is the number of passes through the loop.

As the commands to be modified usually will operate on double precision numbers stored in memory, the commands should be copied into the AR register with a zero (0) C value. The command to copy another command into AR has the form: T N 0 S 28. The contents of the D register used for incrementing the B register should be some multiple of 2 to insure that a double precision command being modified will proceed to the proper address in memory.

As the B register is used twice, once to modify the command in AR and second to be incremented, the B register should be established in a short line for minimum access time. Use short line 22 as it is not used by the Autopoint routine. The D and L registers are used only once for each pass through the loop and may be stored in any convenient location in the program.

Minimum access coding is the coding of a program such that the commands are placed in optimum locations which reduce the time for one command to proceed to the next command and eliminate non-productive computer time. Example 5 is written to illustrate minimum access coding of an Autopoint program which involves an address modification.

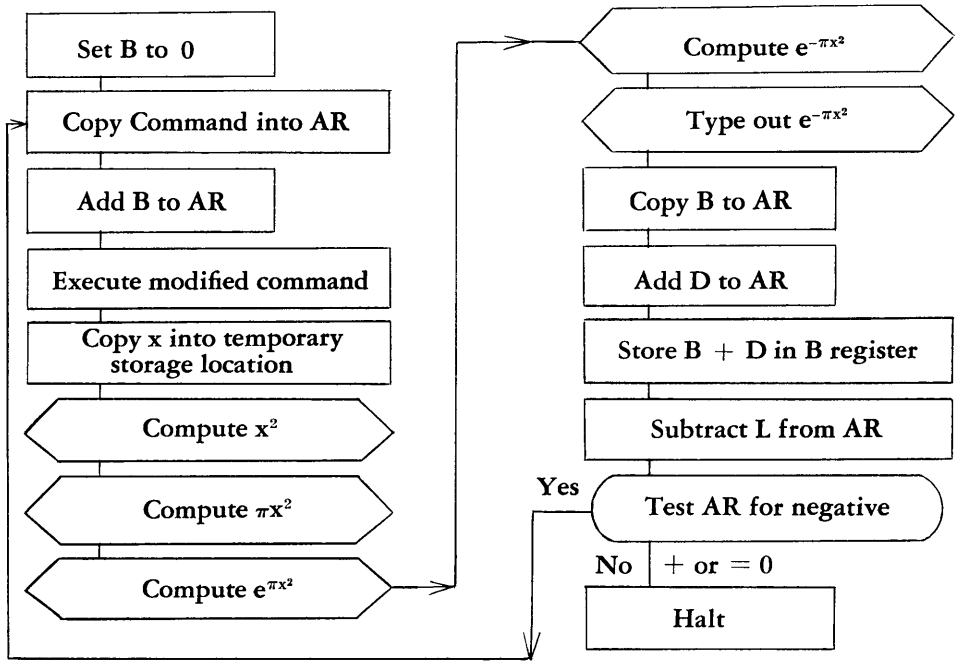
Example 5

Problem: Compute a table of values for $e^{-\pi x^2}$ where x varies between 0 and 1.

Formula: $y = e^{-\pi x^2}$

Procedure: Ten values of x are stored as double precision numbers in line 08, words 14 through 32. The program calculates x^2, πx^2, $e^{\pi x^2}$, and $e^{-\pi x^2}$. The exponential routine is stored in line 07. The main program is stored in line 06. Two double precision constants, π and 1, are stored in line 06.

Flow Diagram:



L	P	T	N	C	S	D	BP	NOTES
00		01	02	0	29	22		Set B to zero
02		04	05	0	06	28		Copy command in 06.04 into AR
04		14	34	5	08	26		Copy x into PN register
05		09	10	0	22	29		Add B to command
10		12	12	0	31	31		Execute modified command in AR
34		36	39	5	26	23		Copy x into temporary storage location
39		72	75	4	23	25		Copy x into ID register
75		80	83	4	23	24		Copy x into MQ register
83	w	11	44	0	21	31		Multiply x by x
11		12	15	4	26	21		Copy x^2 from PN into temporary storage location
15		16	19	4	21	25		Copy x^2 into ID register
19		20	23	4	06	24		Copy π into MQ register
23	w	25	44	0	21	31		Multiply x^2 by π
25		26	29	4	26	21		Copy πx^2 from PN register to temporary storage location
29		70	73	4	21	25		Copy πx^2 into ID register
73		88	89	0	06	28		Copy command in 06.88 into AR register
88	u	13	14	0	06	01		Return command for exponential routine; copy line 06 into line 01
89	u	90	98	0	07	01		Exponential routine; copy line 07 into line 01; enter routine at word 98 to find $e^{\pi x^2}$
14		16	18	4	06	26		Copy 1.0 into PN register
18	w	43	48	0	21	31		Divide 1 by $e^{\pi x^2}$
43		44	47	4	24	21		Store $e^{-\pi x^2}$ from MQ register to temporary storage location
47		80	85	4	21	26		Copy $e^{-\pi x^2}$ to PN register
85	w	06	05	2	21	31		Type out $e^{-\pi x^2}$ and return carriage
06		49	50	0	22	28		Copy B to AR register
50		51	52	0	06	29		Add D to B
52		53	54	0	28	22		Replace contents of register B with B + D
54		55	56	3	06	29		Subtract L from B + D
56		58	01	0	22	31		Test AR register for negative. If positive or equal to zero, halt. If negative, return to 02.
01		00	00	0	21	31		Halt

Operating Instructions:

Follow instructions in Program Entrance and Execution.

Enter sexadecimal constants below in indicated locations

Location	*Constant*	*Use and Decimal Meaning*
51	0200000	Register D, 020000000
55	1400000	Register L, 200000000

Enter 10 values of x in locations 08.14 through 32 by means of:

ABCDy— tab (S).

Use ABCDy tab (S) to enter 1.0 into address 06.16 and 3.1415926536 (π) into address 06.20.

FIELD OFFICES

LOS ANGELES
291 S. La Cienega Boulevard
Beverly Hills, California
Telephone OLeander 5-9610

CLEVELAND
55 Public Square
Cleveland 13, Ohio
Telephone CHerry 1-7789

CHICAGO
919 N. Michigan Avenue
Chicago 11, Illinois
Telephone MIchigan 2-6692

DETROIT
12950 West Eight Mile Road
Detroit 37, Michigan
Telephone JOrdan 6-8789

DALLAS
1511 Bryan Street
Dallas 1, Texas
Telephone RIverside 7-8805

KANSAS CITY
3430 Broadway
Kansas City 11, Missouri
Telephone VAlentine 1-8681

WASHINGTON
1000 Connecticut Avenue, N.W.
Washington 6, D.C.
Telephone STerling 3-0311

SAN FRANCISCO
1330 Broadway
Suite 1121
Oakland 12, California
Telephone GLencourt 2-3664

CANADA
Computing Devices of Canada
P.O. Box 508
Ottawa 4, Ontario
Telephone TAlbot 8-2711

NEW YORK
205 East 42nd Street
Room 1205
New York 17, New York
Telephone OREgon 9-6990

ALL OTHER COUNTRIES
Bendix International
205 East 42nd Street
New York 17, New York
Telephone MUrrayhill 3-1100

THE Bendix CORPORATION

Bendix Computer Division
LOS ANGELES 45, CALIFORNIA