Analysis of Polynomial Functions with the TI-59 Calculator

Part 1

Pierre Chancé
20 Rue de Longchamp
75116 Paris FRANCE

Between the elementary functions accessible by direct calculation and higher-order equations reserved for the computer, there exists an intermediate domain where programmable calculators are useful.

Certain concrete problems sometimes lead to such equations without this creating an exceptional volume of calculation. Some have been encountered, for example, in the handling of small 6 by 6 matrices concerning medical data. Other technicians are also familiar with this type of obstacle in their fields. Consequently, the feeling is that it is worthwhile proposing a convenient program worked out on the Texas Instruments TI-59 and designed for sixth-order and lower-degree polynomial functions.

Independent of the advantages in mathematical terms, I hope that this article will give the user a meaningful introduction to this highly advanced calculator. For practical reasons, the original goals were as follows:

- calculate all the characteristic elements of the function (roots, maximums, minimums, and points of inflection where applicable)
- automatically plot the function curve
- control the program with a single key

Obtaining these conditions virtually eliminates any chance of operating error, and frees the user for other tasks once the calculation has begun. This is especially the case since the main program can be stored on a single magnetic card as can the printout program. Altogether, this provides a simplified procedure which nonetheless permits execution of the successive steps in the following sequence:

- obtaining the appropriate boundaries of the interval to be studied
- choice of the increment
- recall of the maximum error
- calculation of roots in increasing order
- printout of correctly sampled tables of values

All of the above is applicable both for the initial polynomial and for derived polynomials. Because of the geometric significance of the derivative, these provide the maximums and minimums of the function as well as possible points of inflection.

Given that excessive automation can be inconvenient in certain cases, a manual procedure has been provided to permit using the keys to enter the lower and upper boundaries of the interval to be studied along with the value of the increment desired.

After a brief discussion of the calculation principles, the main program and then the automatic printout program for the function curve will be examined. A commentary on numerical applications will conclude the examination.

Calculation Principles

Here is the type of polynomial that will be dealt with:

$$P(x) = a_0x^n + a_1x^{n-1} + \ldots + a_n \quad (a_0 \neq 0,\ n \leq 6)$$

where x is a real number and coefficients a_0, a_1, ... a_n, are known real numbers.

The method used to determine the roots of $P(x) = 0$ is bipartition. This consists of successive dichotomies of the interval (a, b) chosen with the function being continuous over this interval. The calculation is performed sequentially, and the step increment is designated by $\triangle x$.

To determine the root of the equation which belongs to the segment $\triangle x$, the latter is divided in two, and the calculator retains that half at whose extremes the function has opposite signs. The new shortened segment is further divided in two, and the process is repeated iteratively until the upper value of the residual interval is limited by the error limit. The middle of this final interval represents a root of the function to within the error.

This method provides only a single value in an interval $\triangle x$ and requires more calculating time if boundaries a and b are taken too far apart. If they are taken too close together, the risk is obviously one of losing a root; the same applies if $\triangle x$ is too large. Therefore we attempt to eliminate these drawbacks due to too much and too little by programming Lagrange's theorem. This replaces a subjective estimation of the boundaries by a calculation guaranteeing a reliable interval (a, b).

Let $a_0 > 0$ and a_k ($k \geq 1$) be the first of the negative coefficients of the polynomial $P(x)$. The following number as the upper limit of the positive roots of equation $P(x) = 0$ can then be used:

$$R = 1 + \sqrt[k]{\frac{B}{a_0}}$$

where B is the largest of the absolute values of the negative coefficients of the polynomial $P(x)$. Now the user no longer has to distinguish the two values of x between which the roots are supposed to fall. The calculator finds and prints them. To determine the possible limit of the negative roots of the equation, use $x = -z$. This involves changing the sign of the coefficients of the odd registers. However, if the latter equation has no positive roots, the initial equation has no negative roots and the calculator will not provide any.

Optimization of the process is completed by another method. As soon as a root is found, it serves as the lower boundary of the cycle of the following calculation. This sets the increment interval in the new segment to be explored. The correction is aimed at improving the reliability of root detection: this is an essential point.

Observe that details of the program code depend on some of the special capabilities of the TI-59 and PC-100A printing cradle. Naturally, the reader is referred to the instruction manual for full details. In passing, it is merely my intention to mention the decisive factors in my work.

The user has 960 program statements or 100 storage locations available with the possibility of adjusting their respective size as a function of the problem. Each memory block occupies eight program statements. The standard distribution adopted uses 480 program statements and sixty data-storage locations which will be filled entirely. The micromemory connects to the calculator and provides a library of twenty-five programs totalling 5000 steps.

Of the forty functions to which the OP key provides access, special mention can be made of the following:

- printout of alphanumeric characters
- sign indicator
- error indicator
- incrementing and decrementing of memories
- listing of memory content
- listing of labels

Lastly, the T register is very important. Here, it is possible to store and recall a number and test it with respect to the contents of the display register.

In the final analysis, the TI-59 has the quantitative and qualitative features which prove useful in writing a program of the type that is being presented.

Main Program

Data entry:

For reasons of efficiency, the initialization sequence and data entry is not placed at the beginning of the program but at statement 066 with the LBL A instruction and statement 073 with the LBL B instruction (see listing 1). The coefficients of the polynomial are stored by conventional indirect addressing from x^6 at R_{16} to x^0 at R_{10} with a zero introduced when a corresponding term of a power of x is missing.

Evaluation of the polynomial:

This is the role of the LBL A′ instruction placed at location 000 to save calculating time, since this sequence is called frequently.

Determination of boundaries and step increment:

The calculation is monitored by LBL C which, in particular, uses subroutines RCL and STO and PGM 08 of the Solid-State Software. After execution of the sequences the following results are given:

- the lower boundary a is printed out at location 091
- the upper boundary b is printed out at location 099
- the absolute value of the interval (b-a) is printed out at location 117
- the step increment $\triangle x$ is printed out at step 124 immediately after steps 120 thru 122 which contain the variable number of partitions of interval (a, b) or 020 in our listing

The appearance of a zero as a boundary value means the absence of roots for the interval considered, the coefficients of the polynomial being positive or zero. And by three successive calls (PGM 08 A, PGM 08 B, PGM 08 C) program C finally aligns the assignments with those of the library by storing a at R_{01}, b at R_{02} and $\triangle x$ at R_{03}.

Program execution:

This discussion of the mathematical method used will save the trouble of describing the principles again. As for execution:

- LBL RCL (statement 133) changes the sign of the coefficients of the odd registers (R_{15}, R_{13}, R_{11})
- LBL STO (statement 155) plays a complex role. At statement 176, it stores the first coefficient which is

Listing 1: *Main program listing of the polynomial-function analysis on the TI-59.*

000	76	LBL	104	11	A	208	77	GE	312	99	PRT
001	16	A'	105	43	RCL	209	02	02	313	69	DP
002	53	(	106	18	18	210	20	20	314	18	18
003	42	STO	107	36	PGM	211	69	DP	315	87	IFF
004	17	17	108	08	08	212	31	31	316	07	07
005	69	DP	109	12	B	213	97	DSZ	317	02	02
006	10	10	110	98	ADV	214	00	00	318	96	96
007	42	STO	111	43	RCL	215	02	02	319	98	ADV
008	18	18	112	18	18	216	05	05	320	18	C'
009	43	RCL	113	75	-	217	00	0	321	92	RTN
010	01	01	114	43	RCL	218	92	RTN	322	76	LBL
011	42	STO	115	17	17	219	50	I×I	323	18	C'
012	19	19	116	95	=	220	42	STO	324	24	CE
013	06	6	117	99	PRT	221	07	07	325	43	RCL
014	42	STO	118	55	÷	222	43	RCL	326	02	02
015	00	00	119	68	NOP	223	00	00	327	75	-
016	01	1	120	00	0	224	42	STO	328	93	.
017	06	6	121	02	2	225	02	02	329	05	5
018	42	STO	122	00	0	226	94	+/-	330	49	PRD
019	01	01	123	95	=	227	85	+	331	03	03
020	43	RCL	124	99	PRT	228	43	RCL	332	43	RCL
021	17	17	125	36	PGM	229	20	20	333	03	03
022	45	YX	126	08	08	230	95	=	334	65	×
023	43	RCL	127	13	C	231	42	STO	335	03	3
024	00	00	128	98	ADV	232	20	20	336	09	9
025	65	×	129	19	D'	233	43	RCL	337	95	=
026	24	CE	130	15	E	234	02	02	338	42	STO
027	73	RC*	131	92	RTN	235	42	STO	339	01	01
028	01	01	132	76	LBL	236	00	00	340	02	2
029	69	DP	133	43	RCL	237	85	+	341	01	1
030	30	30	134	01	1	238	09	9	342	42	STO
031	69	DP	135	05	5	239	95	=	343	00	00
032	31	31	136	42	STO	240	42	STO	344	03	3
033	85	+	137	01	01	241	01	01	345	09	9
034	43	RCL	138	03	3	242	29	CP	346	42	STO
035	17	17	139	42	STO	243	73	RC*	347	09	09
036	45	YX	140	00	00	244	01	01	348	43	RCL
037	43	RCL	141	01	1	245	22	INV	349	01	01
038	00	00	142	94	+/-	246	77	GE	350	72	ST*
039	65	×	143	64	PD*	247	02	02	351	00	00
040	24	CE	144	01	01	248	67	67	352	69	DP
041	43	RCL	145	02	2	249	69	DP	353	20	20
042	18	18	146	22	INV	250	31	31	354	43	RCL
043	65	×	147	44	SUM	251	97	DSZ	355	03	03
044	73	RC*	148	01	01	252	00	00	356	44	SUM
045	01	01	149	97	DSZ	253	02	02	357	01	01
046	85	+	150	00	00	254	42	42	358	97	DSZ
047	69	DP	151	01	01	255	01	1	359	09	09
048	31	31	152	41	41	256	85	+	360	03	03
049	97	DSZ	153	92	RTN	257	43	RCL	361	48	48
050	00	00	154	76	LBL	258	07	07	362	24	CE
051	00	00	155	42	STO	259	22	INV	363	02	2
052	20	20	156	07	7	260	45	YX	364	01	1
053	43	RCL	157	42	STO	261	43	RCL	365	22	INV
054	10	10	158	00	00	262	20	20	366	90	LST
055	54	)	159	01	1	263	95	=	367	24	CE
056	42	STO	160	06	6	264	42	STO	368	98	ADV
057	17	17	161	42	STO	265	02	02	369	02	2
058	43	RCL	162	01	01	266	92	RTN	370	01	1
059	19	19	163	29	CP	267	50	I×I	371	42	STO
060	42	STO	164	73	RC*	268	32	X:T	372	09	09
061	01	01	165	01	01	269	43	RCL	373	22	INV
062	43	RCL	166	22	INV	270	07	07	374	86	STF
063	17	17	167	67	EQ	271	77	GE	375	07	07
064	92	RTN	168	01	01	272	02	02	376	73	RC*
065	76	LBL	169	76	76	273	49	49	377	09	09
066	11	A	170	69	DP	274	73	RC*	378	16	A'
067	01	1	171	31	31	275	01	01	379	72	ST*
068	06	6	172	97	DSZ	276	50	I×I	380	09	09
069	42	STO	173	00	00	277	42	STO	381	69	DP
070	00	00	174	01	01	278	07	07	382	29	29
071	92	RTN	175	64	64	279	61	GTO	383	69	DP
072	76	LBL	176	42	STO	280	02	02	384	18	18
073	12	B	177	07	07	281	49	49	385	87	IFF
074	72	ST*	178	43	RCL	282	76	LBL	386	07	07
075	00	00	179	00	00	283	14	D	387	03	03
076	99	PRT	180	42	STO	284	99	PRT	388	73	73
077	69	DP	181	20	20	285	36	PGM	389	24	CE
078	30	30	182	73	RC*	286	08	08	390	02	2
079	92	RTN	183	01	01	287	14	D	391	01	1
080	76	LBL	184	55	÷	288	98	ADV	392	22	INV
081	13	C	185	43	RCL	289	92	RTN	393	90	LST
082	98	ADV	186	07	07	290	76	LBL	394	98	ADV
083	71	SBR	187	95	=	291	15	E	395	98	ADV
084	43	RCL	188	72	ST*	292	36	PGM	396	98	ADV
085	71	SBR	189	01	01	293	08	08	397	92	RTN
086	42	STO	190	69	DP	294	15	E	398	76	LBL
087	68	NOP	191	31	31	295	99	PRT	399	19	D'
088	94	+/-	192	97	DSZ	296	22	INV	400	93	.
089	42	STO	193	00	00	297	86	STF	401	00	0
090	17	17	194	01	01	298	07	07	402	00	0
091	99	PRT	195	82	82	299	95	=	403	00	0
092	71	SBR	196	43	RCL	300	43	RCL	404	00	0
093	43	RCL	197	20	20	301	08	08	405	00	0
094	71	SBR	198	42	STO	302	65	×	406	01	1
095	42	STO	199	00	00	303	01	1	407	68	NOP
096	68	NOP	200	85	+	304	00	0	408	68	NOP
097	42	STO	201	09	9	305	95	=	409	68	NOP
098	18	18	202	95	=	306	36	PGM	410	68	NOP
099	99	PRT	203	42	STO	307	08	08	411	14	D
100	43	RCL	204	01	01	308	11	A	412	92	RTN
101	17	17	205	73	RC*	309	36	PGM	413	76	LBL
102	36	PGM	206	01	01	310	08	08	414	17	B'
103	08	08	207	22	INV	311	15	E	415	43	RCL

Listing 1 continued on page 126

Listing 1 continued:

```
416  11   11     436  04   4     456  16   16     476  10   10
417  42  STO     437  95   =     457  43  RCL     477  99  PRT
418  10   10     438  42  STO    458  16   16     478  98  ADV
419  43  RCL     439  13   13    459  99  PRT     479  92  RTN
420  12   12     440  43  RCL    460  43  RCL
421  65   X      441  15   15    461  15   15
422  02   2      442  65   X     462  99  PRT
423  95   =      443  05   5     463  43  RCL
424  42  STO     444  95   =     464  14   14     001  16  A'
425  11   11     445  42  STO    465  99  PRT     066  11   A
426  43  RCL     446  14   14    466  43  RCL     073  12   B
427  13   13     447  43  RCL    467  13   13     081  13   C
428  65   X      448  16   16    468  99  PRT     133  43  RCL
429  03   3      449  65   X     469  43  RCL     155  42  STO
430  95   =      450  06   6     470  12   12     283  14   D
431  42  STO     451  95   =     471  99  PRT     291  15   E
432  12   12     452  42  STO    472  43  RCL     323  18  C'
433  43  RCL     453  15   15    473  11   11     399  19  D'
434  14   14     454  00   0     474  99  PRT     414  17  B'
435  65   X      455  42  STO    475  43  RCL
```

not zero in register R_{07} and recalls its rank in R_{00} to store it at STO 20.

At statements 182 thru 192, all the terms of the polynomial, starting with the first, are divided by the first coefficient which is not zero. This make a_0 positive and equal to 1. This operation must be kept in mind to correctly interpret the change from one polynomial to the next when reading the results.

Location of the first negative coefficient to determine its value and rank begins at statement 196 and uses two loops, statements 203 thru 205 and 207 thru 219. Finally, if the negative coefficient exists, its absolute value is stored in register R_{07} and its rank in register R_{02}, and then its relative position with respect to the first coefficient which is not zero is stored in register R_{20}. Incidentally, the register number of a coefficient (R_{01}) can be determined easily by adding 9 to its ordinal number (R_{00}).

The calculation of the negative coefficient which has the highest absolute value starts at statement 233 and uses the T register with a relatively sophisticated process. This employs four loops, 251 thru 242, 245 thru 267, 271 thru 249 and 279 thru 249. The evaluation of R in Lagrange's formula takes place at statements 255 thru 265.

On the whole, the STO program can be considered to end with the RTN instruction of statement 218 with a long conditional branch with multiple options which operates as a subroutine and ends at the RTN of statement 266.

Maximum error:

This factor is introduced by LBL D (statement 283) which is none other than the assignment of the error ϵ in R_{03} in accordance with the assignment of PGM 08 D in the library. From experience it can be seen that repetition of the error coefficient for each calculation sequence constitutes a constraint, and that setting it at 0.01 in the absence of error entry, as provided by PGM 08, does not really spare the user from this preoccupation.

The fact is that although the precision required varies from one operator to the next, everyone generally uses a rather constant factor for a series of calculations.

It is thus practical to keep ϵ in the program, even if this means modifying it to the programming mode as soon as the need arises. This is the role of LBL D′ (statement 399) where statements 400 thru 410 can contain ϵ up to 1×10^{-10} unless less precision is preferred. It is then sufficient to fill the empty spaces with NOP instructions or simply with zeros after the first significant figure. Since LBL D′ calls D at statement 411 but is itself called by C at statement 129, it is clear that key C finally controls recall, printout and then entry of the maximum error ϵ programmed by the operator.

Calculation of roots:

The heart of this calculation is PGM 08 E from the library which we call at statements 292 and 309. Determination of the successive roots is implemented by our LBL E (statement 291). From the second root, the lower boundary a takes the value of the preceding root augmented by a minimum quantity equal to $\epsilon \times 10$. This augmentation is an artifice designed to move the calculator off the solution it has just found.

The process continues up to unsuccessful exploration of the last interval. At printout this initiates the characteristic series of 9.999...? provided by the manufacturer's PGM 08. LBL E itself is controlled by LBL C at statement 130. This is why key C in fact initiates determination of the roots at the right time.

Tables of values of x and P(x):

These two tables are successively printed out by LBL C' (statement 323) which samples thirty-nine suitable stored values of x from registers R_{21} thru R_{59} and replaces them immediately in the same registers with the thirty-nine corresponding values of P(x). The median of x may be very close to zero. This means that the median of P(x) corresponds to the value of the polynomial for $x = 0$ when $P(x) = P(-x)$.

The sequence C' starts with restoration of the lower boundary a in R_{01} and stores a new increment in R_{03} taken from forty statements between a and b. An automatic listing of the memories with loop and error-indicator control provides indexing of the values.

Sequence C' is itself controlled by LBL E at statement 320 after FLAG 07 has used the error signal from the end of root determination. Given that LBL E is subordinate to LBL C, as was stated earlier, sequence C' is finally implemented by key C also. Given the partition used, the thirty-nine sample values of x and then of P(x) occupy statements 480 thru 959. Those of P(x) can be recorded on a magnetic card in groups 3 and 4 for automatic printout by points of the function curve. The polynomials derived from P(x) could obviously be recorded in the same manner.

Calculation of derived polynomials:

The derivation of each polynomial term of the general expression ax^n gives a term of the expression anx^{n-1}. The calculation is performed by LBL B' (statement 414) which, by depressing key B' once, prints out all the coefficients from x^6 to x^0. The sequence has been designed to provide P'(x) from P(x), P''(x) from P'(x) and so forth as long as the polynomial remains differentiable. Since the program then divides the polynomial by its first nonzero coefficient, it will come as no surprise to find a derivative divided by this term. This in no way changes the final results.

As soon as key B' has played its role, it is sufficient to depress key C for the derived polynomial to be handled in accordance with the same complete cycle as described for the initial polynomial. No other intervention is necessary, unless it is desired to return to the initial polynomial to evaluate it as a function of the roots found for the derived polynomials. This determination is only made after all the derived polynomials that are deemed useful have been used in sequence by the automatic procedure just indicated.

When the coefficients of the initial polynomial have been reentered from R_{16} to R_{10} as at the beginning, enter each root on the keyboard, and each time depress A'. This evaluates the corresponding P(x). The function curve is then completed by virtue of the geometric significance of the derivative by the following coordinates:

- to the root of $P'(x) = 0$ taken as the abscissa corresponds an ordinate by P(x) which defines a maximum or minimum of $P(x) = 0$
- to the root of $P''(x) = 0$ taken as the abscissa corresponds an ordinate by P(x) which defines a point of inflection of $P(x) = 0$ if there is one

Program of Function Curve

Principle:

The curve of the polynomial is automatically plotted as shown in the program in listing 2. It was necessary to conceive an algorithm that compensates for the relative weakness of the TI-59 in this area, since it accepts only twenty whole positive values on a 2.5 inch tape.

With the exception of special cases, the spacing of the plotted points is manifestly insufficient. It can be seen that to cover an 8.5 by 14 inch sheet of paper (a standard European A4 sheet, 21 by 29.7 cm), six strips of machine

Listing 2: *Listing of the program that will plot the function curve.*

```
000  76 LBL      105  06  6       210  00  0
001  98 ADV      106  54  )       211  00  0
002  69 OP       107  55  ÷       212  00  0
003  00  00      108  53  (       213  00  0
004  04  4       109  43 RCL      214  69 OP
005  00  0       110  07  07      215  01  01
006  00  0       111  85  +       216  69 OP
007  00  0       112  43 RCL      217  05  05
008  00  0       113  08  08      218  92 RTN
009  00  0       114  54  )       219  76 LBL
010  00  0       115  54  )       220  95  =
011  00  0       116  95  =       221  00  0
012  00  0       117  72 ST*      222  32 X:T
013  00  0       118  00  00      223  43 RCL
014  69 OP       119  69 OP       224  00  00
015  01  01      120  20  20      225  75  -
016  69 OP       121  97 DSZ      226  04  4
017  05  05      122  09  09      227  00  0
018  92 RTN      123  00  00      228  95  =
019  76 LBL      124  81  81      229  67  EQ
020  28 LOG      125  92 RTN      230  02  02
021  73 RC*      126  76 LBL      231  02  02
022  00  00      127  15  E       232  71 SBR
023  75  -       128  53  (       233  36 PGM
024  43 RCL      129  43 RCL      234  43 RCL
025  20  20      130  07  07      235  06  06
026  95  =       131  85  +       236  65  ×
027  69 OP       132  43 RCL      237  43 RCL
028  07  07      133  08  08      238  01  01
029  43 RCL      134  54  )       239  95  =
030  06  06      135  55  ÷       240  32 X:T
031  65  ×       136  06  6       241  92 RTN
032  07  7       137  55  ÷       242  76 LBL
033  95  =       138  43 RCL      243  36 PGM
034  72 ST*      139  03  03      244  22 INV
035  00  00      140  95  =       245  67  EQ
036  92 RTN      141  42 STO      246  00  00
037  76 LBL      142  06  06      247  02  02
038  75  -       143  06  6       248  92 RTN
039  22 INV      144  42 STO      249  76 LBL
040  77  GE      145  05  05      250  10 E'
041  00  00      146  00  0       251  69 OP
042  21  21      147  42 STO      252  00  00
043  71 SBR      148  01  01      253  19 D'
044  95  =       149  29 CP       254  69 OP
045  92 RTN      150  02  2       255  01  01
046  76 LBL      151  01  1       256  19 D'
047  11  A       152  42 STO      257  69 OP
048  07  7       153  00  00      258  02  02
049  42 STO      154  03  3       259  19 D'
050  00  00      155  09  9       260  69 OP
051  92 RTN      156  42 STO      261  03  03
052  76 LBL      157  09  09      262  19 D'
053  12  B       158  43 RCL      263  69 OP
054  50 I×I      159  06  06      264  04  04
055  72 ST*      160  65  ×       265  69 OP
056  00  00      161  43 RCL      266  05  05
057  55  ÷       162  01  01      267  92 RTN
058  01  1       163  95  =       268  76 LBL
059  00  0       164  42 STO      269  19 D'
060  95  =       165  20  20      270  04  4
061  74 SM*      166  69 OP       271  00  0
062  00  00      167  21  21      272  04  4
063  69 OP       168  43 RCL      273  00  0
064  20  20      169  06  06      274  04  4
065  92 RTN      170  65  ×       275  00  0
066  76 LBL      171  43 RCL      276  04  4
067  13  C       172  01  01      277  00  0
068  14  D       173  99 PRT      278  04  4
069  15  E       174  98 ADV      279  00  0
070  92 RTN      175  98 ADV      280  92 RTN
071  76 LBL      176  98 ADV
072  14  D       177  95  =
073  03  3       178  32 X:T
074  09  9       179  73 RC*
075  42 STO      180  00  00      001  98 ADV
076  09  09      181  71 SBR      020  28 LOG
077  02  2       182  75  -       038  75  -
078  01  1       183  69 OP       047  11  A
079  42 STO      184  20  20      053  12  B
080  00  00      185  97 DSZ      067  13  C
081  53  (       186  09  09      072  14  D
082  73 RC*      187  01  01      127  15  E
083  00  00      188  79  79      201  99 PRT
084  55  ÷       189  98 ADV      220  95  =
085  43 RCL      190  98 ADV      243  36 PGM
086  03  03      191  10 E'       250  10 E'
087  85  +       192  98 ADV      269  19 D'
088  43 RCL      193  98 ADV
089  07  07      194  98 ADV
090  55  ÷       195  97 DSZ
091  43 RCL      196  05  05
092  03  03      197  01  01
093  54  )       198  49  49
094  65  ×       199  92 RTN
095  53  (       200  76 LBL
096  53  (       201  99 PRT
097  43 RCL      202  69 OP
098  03  03      203  00  00
099  65  ×       204  07  7
100  01  1       205  05  5
101  09  9       206  00  0
102  93  .       207  00  0
103  08  8       208  00  0
104  65  ×       209  00  0
```

paper must be juxtaposed. In practice, this means making the data positive, preparing a suitable format and then dividing it into six parts. Thus, the calculator can sequentially print the asterisks corresponding to the thirty-nine values of registers R_{21} thru R_{59}. This can be accomplished in six runs.

Since asterisks will be printed for only thirty-nine pieces of data on 39 by 6 runs, a printout arrangement by points on the base line is used to mark the nonoperation. The interval between points is equal to the increment of the table of the values of x.

Location in the plane is completed by two other arrangements:

- a sign in the shape of a triangle, in place of a point, marks the middle of the base line when there is no value on the zero abscissa
- the ordinates are marked laterally by a column of points with twenty per tape

Initialization and data entry:

These operations are performed by LBL A (statement 047) and LBL B (statement 053). The lower data item entered first is stored in register R_{07}, and the upper data item, entered second, is stored in register R_{08}. The choice of these values determines the amplitude of the graphic reproduction. If it is desired to cover a maximum field, it is necessary to determine the extremes of the values to be reproduced by concurrently consulting the table of the values of P(x) and the group of values of P(x) for x taken from the roots of $P'(x) = 0$.

Note that LBL B continues (statement 057) with the addition of the tenth of each value entered. This automatically provides a margin for the sheet.

Service labels:

Since there is no point in spreading signs on a page without identification, a certain number of sequences permit projections along the abscissa and ordinates. LBL ADV (statement 001) prints one point on the base line of the strip when no data appears on the corresponding abscissa. You will recognize the alphanumeric code controlled by instructions OP 00, OP 01 and OP 05.

Instead of a point, LBL PRT (statement 201) prints a small triangle in the middle of the base line. This distinctive sign marks the zero abscissa when no data item corresponds to it. This median is recognized by monitoring register R_{40} in passing and, by subtracting its ordinal number, it checks for the zero condition using the T register ($=$ t or $\neq$ t). The conditional transfer is executed by means of the LBL = instruction at statement 220 and LBL PGM at statement 243 (the first being called as a subroutine at statement 043 by the LBL − instruction and the second at statement 232 by the LBL =instruction). Naturally, the T register is restored to its previous value immediately after statement 234 and before returning to the main program to serve in the test of the upper limit for the following data item.

Incidentally, it can be observed here that the user is dealing with a structure with four levels of subroutines (main program → SBR − → SBR = → SBR PGM → SBR PRT). The calculator can handle them with no difficulty, since it can accept up to six successive calls. The ordinate location is provided by LBL E' (statement 250), called at statement 181, which prints a column of points at the end of the tape. For reasons of economy, the alphanumeric characters are grouped in LBL D' at statement 269 and recalled as a subroutine whenever needed.

Data printout:

LBL LOG (statement 020) prints an asterisk when the value of R^*_{00} recalled by indirect addressing is between the lower and upper limits of the tape considered. Printout uses a special instruction OP 07. Conditional transfer is provided by LBL — which transfers execution to LBL LOG if the data item is acceptable after subtracting the value of the lower limit stored in register R_{20}. Finally, the data item processed is excluded from the printing field by addition of the group of seven instructions of the tape format contained in register R_{06} (statements 029 thru 035).

Data conversion:

This operation is executed by LBL D (statement 072). It assigns the thirty-nine data items collected by recording in groups 3 and 4 of registers R_{21} thru R_{59} on completion of calculation of the initial P(x) polynomial. However, this could just as well be a polynomial derived for another calculation purpose. The positive value and formatting of this data for printout are obtained with a better spread by dividing them by the increment of the table of values of x contained in register R_{03}. Each converted data item replaces the previous data item term for term in the same register R_{21} thru R_{59}.

Tape printout:

Printout of the six tapes is controlled by LBL E (statement 127). This sequence begins with calculation of the tape format stored in register R_{06}. Tape indexing depends on register R_{01}, initially loaded with zero at statement 146, then incremented at statement 166 and printed at statement 173. The lower tape limit is calculated at statement 165 (STO 20) and the upper limit at statement 177 for loading in the T register.

Transfer to the test of the upper tape limit is executed by instruction SBR — at statement 181. The mechanism of LBL E uses a double loop:

- 149 thru 198 for register R_{09} for data counting loaded at 39
- 179 thru 185 for register R_{05} for tape counting loaded at 6

The entire system is actuated by simply depressing key C, since LBL C at statement 067 monitors D and E. Part 2 of this article will discuss the numerical applications of this program. Samples will be provided to illustrate the initialization and plotting procedures to be followed to output the function curve.■

Glossary

Lagrange's method: Several theorems exist that can solve for the real root(s) of a polynomial equation by means of successive approximations. Lagrange's method obtains the real root using only integer calculations, thereby eliminating any roundoff error. This process is therefore very useful for separation of roots located in a small interval.