LBL--13987

DE82 012362

LBL-13987
UC-32

SOME CALCULATOR PROGRAMS FOR PARTICLE PHYSICS

Charles G. Wohl
Lawrence Berkeley Laboratory
University of California
Berkeley, CA 94720

January 1982

This work was supported by the Director, Office of Energy Research, Office of High Energy and Nuclear Physics, Division of High Energy Physics of the U.S. Department of Energy under Contract Number W-7405-ENG-48.

The ELLIPSE, DALITZ RECTANGULAR, and CM programs work as given on the HP-19C and HP-29C, while the other programs need changes from the following list. In storage-register addresses and in RCL and STO statements:

• for A read .1		
• " B " .2		
• " C " .3		[HP-19C, -29C
• " D " .4		
• " I " 0		

Also:

• for RCL(i) read RCL i		
• " STO(i) " STO i		[HP-19C, -29C
• " DSZ I " DSZ		
• " ISZ I " ISZ		

In the LEGENDRE program, change step 32 to RCL .0, and make $R_{.0}$, not R_0, the storage register for a_0.

The write-up for each program tells what the program does and how to run it, discusses any limitations or special cases or pitfalls, and gives an example or two to test that the program is stored properly and that its operation is understood. The examples have all been rechecked from the final typescript on an HP-97 and, with the necessary changes, on an HP-29C.

LEGENDRE

The LEGENDRE program calculates the values of the Legendre polynomial series

$$A(x) = C \sum_{n=0}^{N} a_n P_n(x)$$

at $x = x_0$, $x_0 + \Delta x$, $x_0 + 2\Delta x$, $\cdots$. The input data are the values of x_0, Δx, N (< 10), the overall normalization constant C, and the expansion coefficients a_n.

The method used is of interest. The straightforward way to calculate A(x) is to start at the bottom of the ladder of Legendre polynomials with P_0 and P_1, use the standard recursion relation to climb to higher rungs, and accumulate the products $a_n P_n$ along the way. Suppose, however, we define a new set of (x-dependent) coefficients c_n recursively by

$$c_n = a_n + \left(\frac{2n+1}{n+1}\right) x\, c_{n+1} - \left(\frac{n+1}{n+2}\right) c_{n+2}$$

and work <u>downward</u> from c_N, with $c_{N+1} = c_{N+2} = 0$. This leads to the remarkable result that

$$\sum_{n=0}^{N} a_n P_n = c_0 \ ,$$

and so the need to accumulate $a_n P_n$ terms and keep track of the sum along the way is eliminated. This method, which was discovered by C.W. Clenshaw, is widely applicable to the summation of series of orthogonal polynomials; see, for example, p. 11 of F.S. Acton, *Numerical Methods that Work* (Harper and Row, New York, 1970). The recursion relation for the new coefficients is of course related to that for the orthogonal polynomials, and has to be worked out separately for each case. In general, the <u>two</u> lowest coefficients (here c_0 and c_1) are involved in the final sum, but for Legendre polynomials the result is particularly simple.

To run the program, store the data in the indicated registers and start with a GSB 0 command. After some seconds, the value of x

is displayed briefly, and then A(x) is displayed and the program stops. For easy access, x is at this point in the *Y* register. Thereafter, the stored value of x is incremented by Δx and A(x) is calculated for the new value each time R/S is pressed.

To test the program, zero the registers and then set $a_9 = 1$, $N = 9$, $x_0 = 1$, $\Delta x = -0.25$, and $C = 1$. This input generates $P_9(x)$ at $x = 1.0, 0.75, 0.5, \cdots$. The first few results are:

x = 1.0	P_9 = 1.000000001
0.75	0.310331851
0.5	-0.267898560
0.25	0.176824421
0.0	0.000000000

These results agree perfectly with 8-place tables of P_9 in chap. 8 of M. Abramowitz and I. Stegun, *Handbook of Mathematical Functions* (Dover, New York, 1972). Some error is to be expected in the ninth place, as in $P_9(1)$ above, but this far exceeds normal requirements of accuracy.

Program: LEGENDRE

Step	Keys
*01	LBL 0
02	0
03	ENTER
04	ENTER
05	RCL B
06	1
07	+
08	STO I
*09	LBL 1
10	1/x
11	1
12	+
13	÷
14	CHS
15	x⇄y
16	2
17	RCL I
18	1/x
19	-
20	RCL C
21	x
22	x⇄y
23	ENTER
24	R↓
25	x
26	+
27	DSZ I
28	GTO 2
29	RCL C
30	PAUSE

Step	Keys
31	x⇄y
†32	RCL 0
33	+
34	RCL A
35	x
36	R/S
37	RCL D
38	RCL C
39	+
40	STO C
41	GTO 0
*42	LBL 2
43	RCL (i)
44	+
45	x⇄y
46	RCL I
47	GTO 1

Registers	
†R_0	a_0
R_1	a_1
R_2	a_2
R_3	a_3
R_4	a_4
R_5	a_5
R_6	a_6
R_7	a_7
R_8	a_8
R_9	a_9
R_A	C
R_B	N
R_C	x_0 (→x)
R_D	Δx
R_I	used

† HP-19C and -29C users see the introduction.

ASSOCIATED LEGENDRE

The ASSOCIATED LEGENDRE program calculates the values of the first-associated Legendre polynomial series

$$B(x) = C \sum_{n=1}^{N} b_n P_n^1(x) = -C\sqrt{1-x^2} \sum_{n=1}^{N} b_n dP_n/dx$$

at $x = x_0$, $x_0 + \Delta x$, $x_0 + 2\Delta x$, $\cdots$ ($|x| \leq 1$). The input data are the values of x_0, Δx, N (< 10), the overall normalization constant C, and the expansion coefficients b_n. Note the minus sign in the definition of P_n^1. Angular distributions are sometimes expanded with a sign convention opposite to the above, the remedy for which is to make C negative.

The method used here (see the LEGENDRE write-up) is to define a new set of coefficients d_n recursively by

$$d_n = b_n + \left(\frac{2n+1}{n}\right) x\, d_{n+1} - \left(\frac{n+2}{n+1}\right) d_{n+2}$$

and work downward from d_N, with $d_{N+1} = d_{N+2} = 0$. This leads to

$$\sum_{n=1}^{N} b_n P_n^1 = -d_1\sqrt{1-x^2} \quad ,$$

and so again the need to keep track of a partial sum along the way is eliminated.

To run the program, store the data in the indicated registers and start with a GSB 0 command. After some seconds, the value of x is displayed briefly, and then B(x) is displayed and the program stops. For easy access, x is at this point in the *Y* register. Thereafter, the stored value of x is incremented by Δx and B(x) is calculated for the new value each time R/S is pressed. If x is outside the range -1 to +1, *Error* is displayed.

To test the program, zero the registers and then set $b_9 = 1$, N = 9, $x_0 = 1$, $\Delta x = -0.25$, and C = 1. This input generates $P_9^1(x)$ at x = 1.0, 0.75, 0.5, $\cdots$. The first few results are:

x = 1.0	P_9^1 = 0.000000000
0.75	0.478134503
0.5	-0.626763685
0.25	1.827987321
0.0	-2.460937500

These results agree perfectly with those obtained using 7-place tables of dP_9/dx in chap. 8 of M. Abramowitz and I. Stegun, *Handbook of Mathematical Functions* (Dover, New York, 1972). In fact, they are almost certainly accurate to eight places, but some error is to be expected in the ninth place.

Program: ASSOCIATED LEGENDRE

Step	Keys
*01	LBL 0
02	0
03	ENTER
04	ENTER
05	RCL B
06	STO I
*07	LBL 1
08	1
09	+
10	$1/x$
11	1
12	+
13	x
14	CHS
15	$x \rightleftarrows y$
16	RCL I
17	$1/x$
18	2
19	+
20	RCL C
21	x
22	$x \rightleftarrows y$
23	ENTER
24	R↓
25	x
26	+
27	RCL (i)
28	-
29	DSZ I
30	GTO 2

Step	Keys
31	RCL C
32	$\cos^{-1}$
33	sin
34	x
35	RCL C
36	PAUSE
37	$x \rightleftarrows y$
38	RCL A
39	x
40	R/S
41	RCL D
42	RCL C
43	+
44	STO C
45	GTO 0
*46	LBL 2
47	$x \rightleftarrows y$
48	RCL I
49	GTO i

Registers	
R_1	b_1
R_2	b_2
R_3	b_3
R_4	b_4
R_5	b_5
R_6	b_6
R_7	b_7
R_8	b_8
R_9	b_9
R_A	C
R_B	N
R_C	x_0 (→ x)
R_D	Δx
R_I	used

CONFIDENCE and EVEN N and POISSON

The CONFIDENCE program calculates confidence levels for the χ^2 probability distribution with N degrees of freedom. The calculation is much shorter when N is even than when it is odd, so there is a short program, EVEN N, for that case (CONFIDENCE handles any N). CONFIDENCE may also be used to get confidence levels for the Gaussian (or normal) and the Poisson probability distributions, but for the Poisson case there is a much shorter program, POISSON.

Figure 1 shows how the confidence levels will be defined here.

(a) The confidence level $CL(\chi_0^2, N)$ for the χ^2 probability distribution $P_N(\chi^2)$ with N degrees of freedom is the probability that a χ^2 greater than χ_0^2 would be obtained.

(b) The confidence level $CL_g(\chi_0)$ for the Gaussian probability distribution $P(\chi)$ is the probability that a result more than χ_0 standard deviations from the mean would be obtained; this is related to $CL(\chi_0^2, N)$ by

$$CL_g(\chi_0) = CL(\chi_0^2, 1) \quad .$$

(c) The confidence level $CL_p(n_0, \bar{n})$ for the Poisson probability distribution $P_{\bar{n}}(n)$ with mean $\bar{n}$ is the probability that a value of n greater than n_0 would be obtained; this is related to $CL(\chi_0^2, N)$ by

$$CL_p(n_0, \bar{n}) = 1.0 - CL(\chi_0^2, N) \quad ,$$

with $\chi_0^2 = 2\bar{n}$ and $N = 2n_0 + 2$.

The equations used to calculate $CL(\chi_0^2, N)$ are as follows. When N is even,

$$CL(\chi_0^2, N) = \sqrt{2\pi}\, Z(\chi_0) \left(1 + \sum_{n=1}^{N'} \frac{\chi_0^{2n}}{2 \cdot 4 \cdot 6 \cdots (2n)} \right) \quad ,$$

where $Z(\chi_0) = (2\pi \exp \chi_0^2)^{-\frac{1}{2}}$ and $N' = (N-2)/2$. When N is odd,

$$CL(\chi_0^2, N) = CL_g(\chi_0) + 2Z(\chi_0) \sum_{n=1}^{N''} \frac{\chi_0^{2n-1}}{1 \cdot 3 \cdot 5 \cdots (2n-1)} \quad ,$$

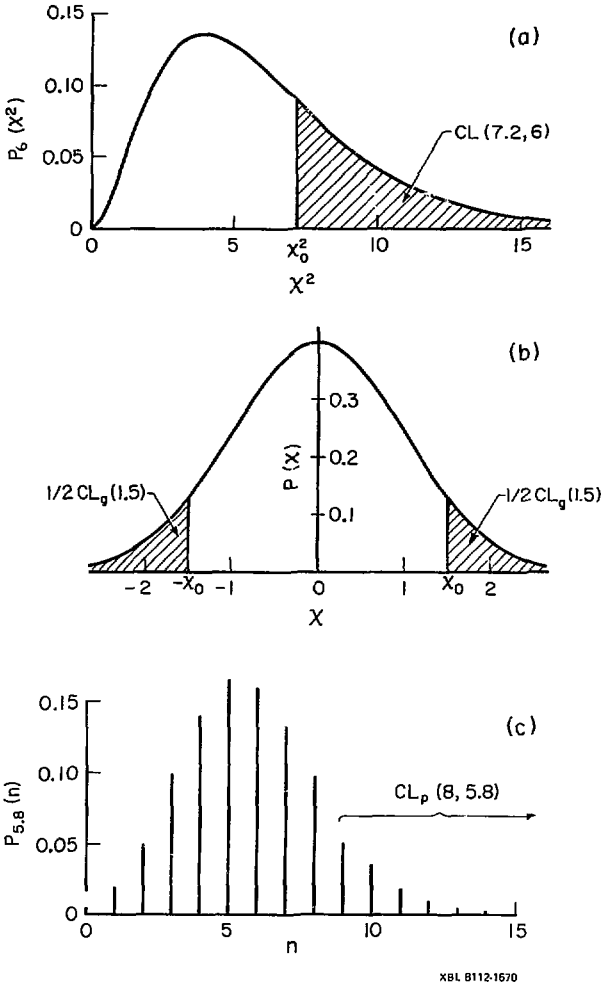

(a)
0.15
0.10
0.05
0
$P_6(\chi^2)$
CL (7.2, 6)
0
5
χ_0^2
10
15
χ^2
(b)
0.3
0.2
0.1
P (X)
1/2 CL_g(1.5)
1/2 CL_g(1.5)
-2
$-X_0$
-1
0
1
X_0
2
X
(c)
0.15
0.10
0.05
0
$P_{5.8}$ (n)
CL_p (8, 5.8)
0
5
10
15
n
XBL 8112-1670

Figure 1

where N" = (N - 1)/2. There is no closed expression for $CL_g(\chi_0)$. For $\chi_0 \geq 2$, a truncated continued fraction has been used:

$$CL_g(\chi_0) = 2Z(\chi_0)\left[\frac{1}{\chi_0+}\Big/\frac{1}{\chi_0+}\Big/\frac{2}{\chi_0+}\Big/\frac{3}{\chi_0+}\Big/\cdots\Big/\frac{19}{\chi_0+20/6}\right] .$$

For $\chi_0 < 2$, the equation used is

$$CL_g(\chi_0) = 1.0 - 2Z(\chi_0)\sum_{n=1}^{\infty}\frac{\chi_0^{2n-1}}{1\cdot3\cdot5\cdots(2n-1)} ,$$

where "infinity" is reached when the last term added to the series is smaller than 10^{-9}. For N > 1, the early terms of the series here are cancelling terms of the series in the equation at the bottom of p. 10. Account is taken of the fact that for very large N or very small χ_0^2 the "infinite" series can be shorter than the finite one.

To run CONFIDENCE or EVEN N, store χ_0^2 and N in the indicated registers and start the program with a GSB 0 command. When it stops, the confidence level is displayed. The range of χ_0^2 covered is $0.0 < \chi_0^2 \leq 460.5$; CL(0.0, N) is of course 1.0, and when χ_0^2 exceeds 460.5 there is an overflow.

To run POISSON, store n_0 and $\bar{n}$ as indicated, press GSB 0, and see the confidence level when the program stops. It is easy with a few trials to home in on a value of $\bar{n}$ that for a given n_0 produces a confidence level of, say, 0.95.

Following are some examples, with approximate running times. The starred examples test all the parts of CONFIDENCE, and they at least should be tried.

	CL(0.04, 1)	= 0.841480581	= $CL_g(0.2)$	(7 sec)
	CL(3.61, 1)	= 0.057433120	= $CL_g(1.9)$	(19 sec)
	CL(4.00, 1)	= 0.045500263	= $CL_g(2.0)$	(17 sec)
	CL(25.0, 1)	= 0.000000573	= $CL_g(5.0)$	(17 sec)
*	CL(3.61, 9)	= 0.935159132		(24 sec)
*	CL(25.0, 9)	= 0.002971180		(21 sec)
	CL(2.00, 2)	= 0.367879441	= 1.0 - $CL_p(0, 1.0)$	(2 sec)
*	CL(50., 50)	= 0.473398469	= 1.0 - $CL_p(24, 25.0)$	(26 sec)

For N = 1, there are 15-place tables of (1 - CL/2) in chap. 26 of M. Abramowitz and I. Stegun, *Handbook of Mathematical Functions* (Dover, New York, 1972). For all N, there are 9-place tables of (1 - CL) in H.L. Harter, *New Tables of the Incomplete Gamma-Function Ratio and of Percentage Points of the Chi-Square and Beta Distributions* (Aerospace Research Laboratories, U.S. Air Force, 1964). Comparisons show that for N = 1 the absolute error from the program never exceeds 1×10^{-9}. For N < 100, the absolute error rarely exceeds 1×10^{-9}, but for N > 100 absolute errors as large as 3×10^{-9} have been found.

Program: CONFIDENCE

Step	Keys	Step	Keys	Step	Keys
*01	LBL 0	31	2	61	$x \rightleftarrows y$
02	RCL 2	32	0	62	$x > y?$
03	STO I	33	STO I	63	GTO 5
04	1	34	6	64	1
*05	LBL 1	*35	LBL 3	65	STO I
06	RCL I	36	RCL I	66	RCL 4
07	2	37	$x \rightleftarrows y$	67	ABS
08	-	38	÷	68	CHS
09	$x < 0?$	39	RCL 3	*69	LBL 6
10	GTO 2	40	+	70	2
11	STO I	41	DSZ I	71	π
12	$x = 0?$	42	GTO 3	72	÷
13	GTO 7	43	$1/x$	73	$\sqrt{x}$
14	RCL 1	44	RCL 4	74	x
15	÷	45	+	75	ENTER
16	÷	46	GTO 6	*76	LBL 7
17	1	*47	LBL 4	77	R↓
18	+	48	RCL 3	78	RCL 1
19	GTO 1	49	STO - 4	79	2
*20	LBL 2	*50	LBL 5	80	÷
21	+	51	ISZ I	81	e^x
22	RCL	52	ISZ I	82	÷
23	$\sqrt{x}$	53	RCL 1	83	RCL I
24	STO 3	54	x	84	+
25	÷	55	RCL I	85	RTN
26	STO 4	56	÷		
27	RCL 1	57	STO - 4		
28	4	58	EEX		
29	$x > y?$	59	CHS		
30	GTO 4	60	9		

Registers	
R_1	X_0^2
R_2	N
R_3	used
R_4	used
R_I	used

Program: EVEN N

Step	Keys
*01	LBL 0
02	RCL 2
03	STO I
04	1
*05	LBL 1
06	DSZ I
07	DSZ I
08	GTO 2
09	RCL 1
10	2
11	÷
12	e^x
13	÷
14	RTN
*15	LBL 2
16	RCL 1
17	RCL I
18	÷
19	x
20	1
21	+
22	GTO 1

Registers	
R_1	χ_0^2
R_2	N
R_I	used

Program: POISSON

Step	Keys
*01	LBL 0
02	RCL 1
03	STO I
04	1
05	ENTER
06	ISZ I
*07	LBL 1
08	DSZ I
09	GTO 2
10	RCL 2
11	e^x
12	÷
13	-
14	RTN
*15	LBL 2
16	RCL 2
17	RCL I
18	÷
19	x
20	+
21	GTO 1

Registers	
R_1	n_0
R_2	$\bar{n}$
R_I	used

TWO BODY and CM

In the 2-body reaction $a + b \rightarrow 1 + 2$, let m_i be the mass of particle i and let P_a be the momentum of particle a in the inertial frame in which particle b is at rest (the lab frame). The four masses and P_a are the input data for the TWO BODY program, which calculates the following quantities:

s	=	c.m. energy squared
E	=	c.m. energy
p	=	initial-state c.m. momentum
p'	=	final-state c.m. momentum
$\Delta t = \Delta u$	=	range of 4-momentum transfer squared
t_0 (t_π)	=	4-momentum-transfer squared between particles a and 1 when 1 is produced at 0° (180°) in the c.m.
u_0 (u_π)	=	4-momentum-transfer squared between particles a and 2 when 2 is produced at 0° (180°) in the c.m.

CM is a shorter program that uses P_a, m_a, and m_b as input to calculate s, E, and p.

To run TWO BODY, store the four masses and P_a in the indicated registers, and start the program with a GSB 0 command. When it stops, the calculated quantities are in the storage registers (u_π is displayed). If *Error* appears when the program is started, then P_a may be below the threshold for the reaction. In some cases, however, P_a can be too low without such notice being given.

To run CM, store P_a, m_a, and m_b in the indicated registers, and start the program with a GSB 0 command. When it stops, s, E, and p are in the storage registers (p is displayed).

As a test example, consider $\pi^+ p \rightarrow K^+ \Sigma^+$ scattering at 4 GeV/c. Then, in units of GeV or GeV/c or GeV^2, the input numbers are P_a = 4.0, m_a = 0.1396, m_b = 0.9383, m_1 = 0.4937, and m_2 = 1.1894, and the output numbers are s = 8.411, E = 2.900, p = 1.294, p' = 1.146, $\Delta t = \Delta u$ = 5.934, t_0 = -0.019, t_π = -5.953, u_0 = +0.101, and u_π = -5.834.

Program: TWO BODY

Step	Keys	Step	Keys	Step	Keys
*01	LBL 0	31	GSB 2	61	-
02	RCL 0	32	-	62	GSB 2
03	RCL 2	33	$\sqrt{x}$	63	-
04	RCL 1	34	STO 8	64	CHS
05	RCL 0	35	x	65	RTN
06	GSB 2	36	4	*66	LBL 2
07	+	37	x	67	x^2
08	$\sqrt{x}$	38	STO 9	68	$x \rightleftarrows y$
09	+	39	CHS	69	x^2
10	GSB 2	40	RCL 1	70	RTN
11	-	41	GSB 1		
12	STO 5	42	STO A		
13	$\sqrt{x}$	43	+		
14	STO 6	44	STO B		
15	÷	45	R↓		
16	RCL 2	46	RCL 2		
17	x	47	GSB 1		
18	STO 7	48	STO C		
19	RCL 3	49	+		
20	RCL 4	50	STO D		
21	RCL 3	51	RTN		
22	GSB 2	*52	LBL 1		
23	-	53	RCL 7		
24	RCL 5	54	GSB 2		
25	+	55	+		
26	RCL 6	56	$\sqrt{x}$		
27	÷	57	RCL D		
28	2	58	-		
29	÷	59	RCL 7		
30	STO D	60	RCL B		

Registers

R_0	p_a
R_1	m_a
R_2	m_b
R_3	m_1
R_4	m_2
R_5	s
R_6	E
R_7	p
R_8	p'
R_9	$\Delta t = \Delta u$
R_A	t_0
R_B	t_π
R_C	u_0
R_D	u_π

Program: CM

Step	Keys
*01	LBL 0
02	RCL 0
03	RCL 1
04	→P
05	RCL 2
06	+
07	x^2
08	RCL 0
09	x^2
10	-
11	STO 3
12	$\sqrt{x}$
13	STO 4
14	RCL 0
15	$x \rightleftarrows y$
16	÷
17	RCL 2
18	x
19	STO 5
20	RTN

Registers

R_0	p_a
R_1	m_a
R_2	m_b
R_3	s
R_4	E
R_5	p

ESTIMATES OF RADIATION DOSES IN TISSUE AND ORGANS AND RISK OF EXCESS CANCER IN THE SINGLE-COURSE RADIOTHERAPY PATIENTS TREATED FOR ANKYLOSING SPONDYLITIS IN ENGLAND AND WALES[1,2]

LBL--13999

DE82 008423

Jacob I. Fabrikant, M.D., Ph.D.[3,4]
Lawrence Berkeley Laboratory, Donner Laboratory
University of California, Berkeley
Berkeley, California

and

John T. Lyman, Ph.D.
Division of Biology and Medicine
Lawrence Berkeley Laboratory
University of California
Berkeley, CA 94720

951 3024

[1]Presented at Scientific Session, Third International Symposium of Radiation Protection-Advances in Theory and Practice, Inverness, Scotland, June 6-11, 1982.

[2]Supported by the Office of Health and Environmental Research of the U.S. Department of Energy under contract No. W-7405-ENG-48.

[3]Professor of Radiology, University of California School of Medicine, San Francisco, California.

[4]Mailing address: Donner Laboratory, University of California, Berkeley, California 94720.

MGW

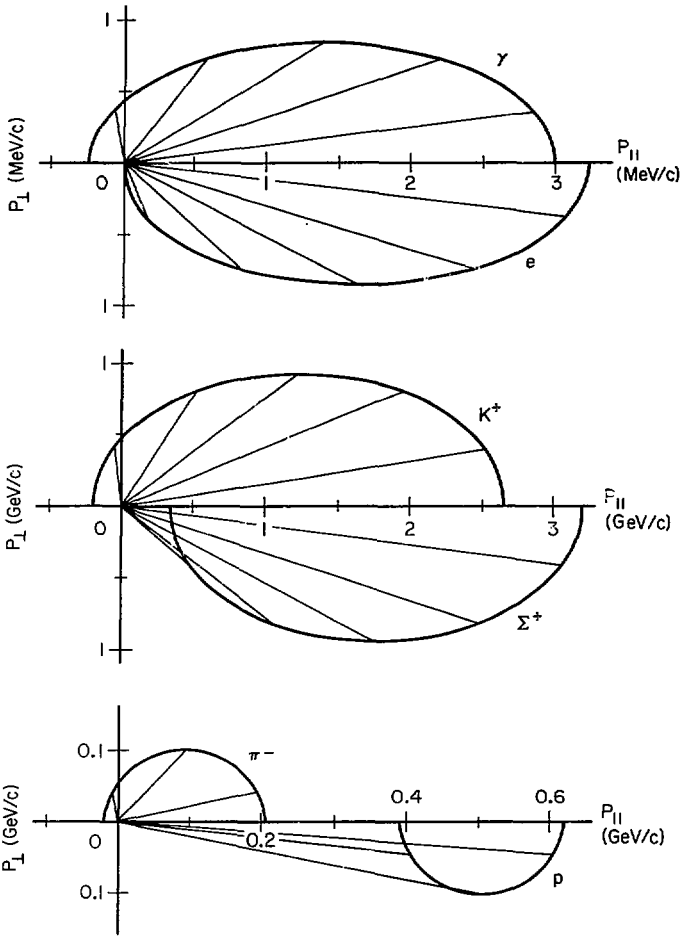

1
γ
P⊥ (MeV/c)
0
1
2
3
P∥ (MeV/c)
e
1
1
K+
P⊥ (GeV/c)
0
1
2
3
P∥ (GeV/c)
Σ+
1
0.1
π−
0.4
0.6
P⊥ (GeV/c)
0
0.2
P∥ (GeV/c)
p
0.1
XBL 8112-1669

Figure 2

registers; for a decay $a \rightarrow 1 + 2$, set $m_b = 0$. Start the program with a GSB 0 command. When it stops, the values of $\gamma\beta e_1$, $\gamma p'$, p', E (the total c.m. energy), and $\cos\theta$ will be in the storage registers, with $\cos\theta = +1$. The components $P_{1\parallel}$ and $P_{1\perp}$ for this value of $\cos\theta$ will be in the *X* and *Y* registers. Each time R/S is pressed, the value of $\cos\theta$ is decreased by 0.1 and the program stops with the corresponding components of $\vec{P}_1$ in the *X* and *Y* registers:

$$(X, Y) = (P_{1\parallel}, P_{1\perp}) \quad .$$

When $\cos\theta$ eventually goes below -1, *Error* is displayed. Then, to obtain the half ellipse for $\vec{P}_2$, interchange m_1 and m_2 in the registers (i.e., now consider particle 2 to be particle 1), and start over with a GSB 0 command. If *Error* appears when the program is started, then P_a may be below the threshold for the reaction. *Error* also appears if $P_a = 0$.

The $\rightarrow$P key changes the rectangular coordinates $P_{1\parallel}$ and $P_{1\perp}$ in the *X* and *Y* registers to the polar coordinates P_1 and Θ_1, where Θ_1 is the lab angle at which particle 1 is produced. After the first pass through the program has been made, the lab angle ϑ_1 corresponding to any c.m. angle θ may be found by storing the value of $(\cos\theta + 0.1)$ in the R_9 register, then pressing R/S, then $\rightarrow$P: Θ_1 is in the *Y* register.

As a test example, consider $\pi^+p \rightarrow K^+\Sigma^+$ scattering at 3 GeV/c. The input numbers, in GeV/c or GeV, are $P_a = 3.0$, $m_a = 0.1396$, $m_b = 0.9383$, $m_1 = 0.4937$, and $m_2 = 1.1894$. Start the program with a GSB 0 command, and it stops with $P_{1\parallel} = 2.659$ GeV/c, $P_{1\perp} = 0.0$ GeV/c, $\gamma\beta e_1 = 1.231$ GeV, $\gamma p' = 1.427$ GeV/c, $p' = 0.926$ GeV/c, E = 2.557 GeV, and $\cos\theta = 1.0$. Press R/S and the program stops with $P_{1\parallel} = 2.516$ GeV/c, $P_{1\perp} = 0.404$ GeV/c ($P_1 = 2.548$ GeV/c, $\Theta_1 = 9.113^\circ$), and $\cos\theta = 0.9$.

Program: ELLIPSE

Step	Keys	Step	Keys
*01	LBL 0	31	RCL 3
02	RCL 0	32	GSB 2
03	RCL 2	33	-
04	RCL 1	34	CHS
05	RCL 0	35	$\sqrt{x}$
06	GSB 2	36	STO x 6
07	+	37	STO 7
08	$\sqrt{x}$	38	1
09	+	39	STO 9
10	STO 6	*40	LBL 1
11	GSB 2	41	RCL 7
12	-	42	RCL 9
13	STO 8	43	$\cos^{-1}$
14	÷	44	sin
15	RCL 4	45	x
16	RCL 3	46	RCL 6
17	GSB 2	47	RCL 9
18	-	48	x
19	x	49	RCL 5
20	+	50	+
21	2	51	R/S
22	÷	52	·
23	STO 5	53	1
24	$x \rightleftarrows y$	54	STO - 9
25	÷	55	GTO 1
26	RCL 8	*56	LBL 2
27	$\sqrt{x}$	57	x^2
28	STO 8	58	$x \rightleftarrows y$
29	STO ÷ 6	59	x^2
30	x	60	RTN

Registers	
R_0	p_a
R_1	m_a
R_2	m_b
R_3	m_1
R_4	m_2
R_5	$\gamma\beta e_1$
R_6	$\gamma p'$
R_7	p'
R_8	E
R_9	$\cos\theta$

DALITZ RECTANGULAR

The DALITZ RECTANGULAR program calculates coordinates of points on the boundary of the rectangular Dalitz plot for the decay of a particle or system of mass M into three particles having masses m_1, m_2, and m_3. The x and y coordinates of the plot are m_{12}^2 and m_{13}^2, where m_{ij} is the invariant mass of the system of particles i and j.

Let $m_{12}^2\downarrow$ and $m_{12}^2\uparrow$ be the smallest and largest values that m_{12}^2 attains anywhere on the boundary. The values of m_{12}^2 for which boundary coordinates (m_{12}^2, m_{13}^2) are going to be calculated are

$$m_{12}^2\downarrow, \quad m_{12}^2\downarrow + \Delta, \quad m_{12}^2\downarrow + 2\Delta, \quad \cdots, \quad m_{12}^2\uparrow \quad ,$$

where Δ is a step size in m_{12}^2. (If, however, $m_1 = m_2 = 0$, see below.) This sequence repeats over and over, first for points along the upper boundary (the boundary on which m_{13}^2 is larger), then for points along the lower boundary, then back to the upper boundary, and so on.

Store the four masses and the step size Δ in the indicated registers (Δ in fact need not be chosen until after the first pass is made through the program, when $m_{12}^2\downarrow$ and $m_{12}^2\uparrow$ are available). Start the program with a GSB 0 command. When it stops, the values of $m_{12}^2\downarrow$ and $m_{12}^2\uparrow$ will be in the indicated registers, and the coordinates (m_{12}^2, m_{13}^2) of the leftmost point of the boundary will be in the *X* and *Y* registers. From then on a new boundary point is obtained each time R/S is pressed:

$$(X, Y) = (m_{12}^2, m_{13}^2) \quad .$$

The step size Δ, or the *current* value of m_{12}^2 (in R_5), or the boundary one is on (± 1 in R_8) may be changed at any time (Δ should not be made negative). No error message is given if M is less than $(m_1 + m_2 + m_3)$.

There is a special case. If $m_1 = m_2 = 0$, then $m_{12}^2\downarrow$ is zero, and to avoid a division by zero the program skips the first in the sequence of m_{12}^2 values given above (i.e., the sequence becomes Δ, 2Δ, $\cdots$, $m_{12}^2\uparrow$). In this case, a nonzero value of Δ does need to be stored before the first pass is made. Now the upper and lower

boundaries end at the points $(0, M^2)$ and $(0, m_3^2)$, and the segment of the m_{13}^2 axis between these points is part of the boundary.

As a test example, consider the decay of a 3-GeV system into $\bar{K}^0\pi^-p$. The input masses, in GeV, are $M = 3.0$, $m_1 = 0.4977$, $m_2 = 0.1396$, and $m_3 = 0.9383$. After the GSB 0 command, the program stops with the coordinates (0.406, 7.152) — here and below the units are GeV^2. The values of $m_{12}^2\downarrow$ and $m_{12}^2\uparrow$ are 0.406 and 4.251, which suggests that a reasonable value of Δ, at least to start with, might be 0.2. Storing this and pressing R/S gives the coordinates (0.606, 8.179). Pressing R/S again gives the coordinates (0.806, 8.119). Another test is to set $M = 1.0$, $m_1 = m_2 = m_3 = 0$, and $\Delta = 0.2$. The boundary is the triangle with vertices at (0, 0), (1, 0), and (0, 1).

Program: DALITZ RECTANGULAR

Step	Keys
*01	LBL 0
02	RCL 1
03	RCL 2
04	+
05	x^2
06	STO 6
07	RCL 0
08	RCL 3
09	-
10	x^2
11	STO 7
12	1
13	STO 8
*14	LBL 1
15	RCL 6
*16	LBL 2
17	STO 5
18	$x = 0$?
19	GTO 3
20	0
21	STO 9
22	RCL 3
23	RCL 0
24	RCL 3
25	GSB 5
26	RCL 1
27	RCL 2
28	RCL 1
29	GSB 5
30	RCL 8
31	x
32	-
33	RCL 9
34	GSB 6
35	RCL 5
36	R/S
*37	LBL 3
38	RCL 7
39	RCL 5
40	$x = y$?
41	GTO 1
42	RCL 4
43	+
44	$x \rightleftarrows y$
45	$x > y$?
46	GTO 4
47	1
48	CHS
49	STO x 8
*50	LBL 4
51	R↓
52	GTO 2
*53	LBL 5
54	GSB 6
55	RCL 5
56	+
57	ABS
58	RCL 5
59	$\sqrt{x}$
60	÷
61	2
62	÷
63	STO + 9
64	GSB 6
65	ABS
66	$\sqrt{x}$
67	RTN
*68	LBL 6
69	x^2
70	$x \rightleftarrows y$
71	x^2
72	-
73	RTN

Registers	
R_0	M
R_1	m_1
R_2	m_2
R_3	m_3
R_4	Δ
R_5	m_{12}^2
R_6	$m_{12}^2\downarrow$
R_7	$m_{12}^2\uparrow$
R_8	±1
R_9	used

DALITZ TRIANGULAR

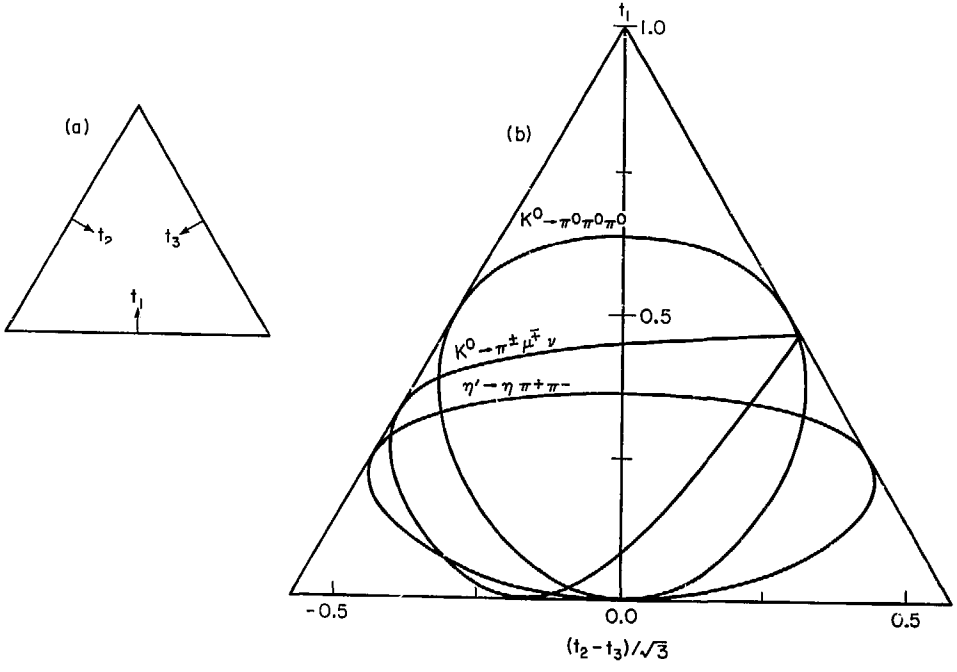
The DALITZ TRIANGULAR program calculates coordinates of points on the boundary of the normalized triangular Dalitz plot for the decay of a particle or system of mass M into three particles having masses m_1, m_2, and m_3. Let T_i be the kinetic energy of particle i in the rest frame of M, $Q = (T_1 + T_2 + T_3)$ be the total kinetic energy released by the decay, and $t_i = T_i/Q$ be the fraction of Q taken up by particle i. Then the fractions t_1, t_2, and t_3 are the distances measured inward from the base, left-hand side, and right-hand side of an equilateral triangle whose altitude is unity. Figure 3(a) shows a schematic of the plot, and fig. 3(b) shows some boundaries drawn using the program. Each boundary touches each side of the circumscribing triangle at one point. A boundary has a sharp corner where it touches a side if the particle whose kinetic energy is measured from that side has a mass of zero.

To avoid actually having to plot in triangular coordinates, the boundary points are calculated using a rectangular system having its origin at the center of the base: the vertical coordinate is t_1 and the horizontal coordinate is $(t_2-t_3)/\sqrt{3}$. Let $t_1\dagger$ be the largest value that t_1 attains anywhere on the boundary. The values of t_1 for which boundary coordinates are going to be calculated are

$$0,\ \Delta,\ 2\Delta,\ \cdots,\ t_1\dagger\ ,$$

where Δ is a step size in t_1. (If, however, $m_2 = m_3 = 0$, see below.) This sequence repeats over and over, first for points along the right side of the boundary, then for points along the left side, then back to the right side, and so on.

Store the four masses and the step size Δ in the indicated registers (a value of 0.05 or 0.1 for Δ is about right), and start the program with a GSB 0 command. When it stops, the values of $t_1\dagger$ and Q will be in the indicated registers, and the coordinates $\left(t_1,\ (t_2-t_3)/\sqrt{3}\right)$ of the lowermost point of the boundary will be in the *X* and *Y* registers. From then on a new boundary point is obtained



XBL 8112 1608

Figure 3

each time R/S is pressed:

$$(X,\ Y) \ = \ (t_1,\ (t_2-t_3)/\sqrt{3}) \quad .$$

The step size Δ, or the current value of t_1 (in R_5), or the boundary one is on (±1 in R_8) may be changed at any time (Δ should not be made negative). No error message is given if M is less than $(m_1+m_2+m_3)$.

There is a special case. If $m_2 = m_3 = 0$, then to avoid a division by zero the program skips the last in the sequence of t_1 values given above. In this case, the upper boundary of the plot is the horizontal line segment at $t_1 = t_1\uparrow$ that crosses from one side to the other of the circumscribing triangle.

As a test example, consider $K^0 \rightarrow \pi^{\pm}\mu^{\mp}\nu$ decay. The input masses, in GeV, are M = 0.4977, m_1 = 0.1396, m_2 = 0.1057, and m_3 = 0; and set Δ = 0.05. When, after the GSB 0 command, the program stops, the coordinates of the first point to be plotted are (0.0, -0.170); and $t_1\uparrow$ = 0.466 and Q = 0.2524 GeV. Press R/S and the next coordinates are (0.05, -0.040). Press it again, and the coordinates are (0.10, +0.020). Another test is to set M ≠ 0 and $m_1 = m_2 = m_3 = 0$. Then $t_1\uparrow$ = 0.5, Q = M, and the boundary is the triangle with vertices at the centers of the sides of the circumscribing triangle.

Program: DALITZ TRIANGULAR

Step	Keys	Step	Keys	Step	Keys	Step	Keys
*01	LBL 0	31	RCL 1	61	-	91	ABS
02	RCL 0	32	GSB 5	62	-	92	$\sqrt{x}$
03	RCL 1	33	STO A	63	RCL 7	93	RTN
04	-	34	GSB 5	64	3		
05	STO 7	35	$x=0$?	65	$\sqrt{x}$		
06	RCL 2	36	GTO 3	66	x		
07	RCL 3	37	STO B	67	÷		
08	+	38	RCL 2	68	RCL 5		
09	STO - 7	39	x^2	69	R/S		
10	+	40	RCL 3	*70	LBL 3		
11	RCL 0	41	x^2	71	RCL 6		
12	2	42	-	72	RCL 5		
13	x	43	RCL B	73	$x=y$?		
14	÷	44	÷	74	GTO 1		
15	STO 6	45	STO x 9	75	RCL 4		
16	1	46	+	76	+		
17	STO 8	47	RCL 2	77	$x \rightleftarrows y$		
*18	LBL 1	48	2	78	$x>y$?	Registers	
19	0	49	x	79	GTO 4	R_0	M
*20	LBL 2	50	GSB 5	80	1	R_1	m_1
21	STO 5	51	RCL A	81	CHS	R_2	m_2
22	RCL 7	52	x	82	STO x 8	R_3	m_3
23	x	53	RCL 8	*83	LBL 4	R_4	Δ
24	RCL 1	54	x	84	R↓	R_5	t_1
25	+	55	RCL 9	85	GTO 2	R_6	$t_1\uparrow$
26	RCL 0	56	+	*86	LBL 5	R_7	Q
27	$x \rightleftarrows y$	57	RCL B	87	x^2	R_8	± 1
28	-	58	÷	88	$x \rightleftarrows y$	R_9	used
29	STO 9	59	RCL 2	89	x^2	R_A	used
30	LAST x	60	RCL 3	90	-	R_B	used