A MONTE CARLO METHOD FOR SIMULATION OF π^0 PRODUCTION NEAR THRESHOLD

by

MICHAEL WILLIAM MADDEN

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

Physics

APPROVED:

D.A. Jenkins, Chairman

M.J. Reiner

D.D. Long

May, 1982
Blacksburg, Virginia

ACKNOWLEDGMENTS

This thesis represents a portion of the research I have conducted under the guidance of Dr. D. A. Jenkins. From the time I began working with him until the present, he has continuously provided the support as well as the criticism necessary for my success.

I would also like to thank the other members of my advisory committee, Dr. D. D. Long and Dr. M J. Reiner, for their guidance in the preparation of the thesis.

TABLE OF CONTENTS

I INTRODUCTION 1
A. The Monte Carlo Method 1

II THE METHOD 4
A. Restrictions on the Angular Distribution Parameters 4
B. Generation of the Pion Angle in the CM 7
C. Reaction Kinematics 14
D. Transformation of the π^o Angles from the CM to the Lab 16
E. The Second Monte Carlo - Generation of the Gamma Angles . . . 18
F. A Final Rotation 20

III. PUTTING IT ALL TOGETHER - THE PROGRAM CARLOS . . . 24
A. Input Requirements 24
B. Output Format 25
C. Sample Input and Output Files 26

IV. A TEST OF THE METHOD 29

V. CONCLUSIONS 31

REFERENCES 32

TABLES 33

FIGURES 34

APPENDICES
A. The Subroutine KINMAT 42
B. The Subroutine LTRANS 50
C. The Subroutine ROTATE 58
D. The Subroutine INVTAN 67
E. Listing of CARLOS 73

VITA 78

I. INTRODUCTION

Experiments are currently being conducted in π^o production with light nuclei near threshold.[1] In particular, the angular distribution of the π^o in the center of mass (CM) frame is desired. The short lifetime of the π^o (10^{-16} sec) precludes observing it directly. Fortunately, it decays into 2 gamma rays which may be detected with relative ease. The angular distribution for the coincidence gammas in the lab frame may then be related to the angular distribution for the π^o in the CM system.

With the advent of this work, questions have arisen as to the effect of the measuring scheme on the angular distribution for the gammas:

1. Does the apparatus sample a large enough solid angle to specify the required features of the pion angular distribution?
2. What is the effect of a finite angular resolution?

A method is required which simulates pion production so that these effects may be investigated. The purpose of this project is to develop and test a computer implementation of such a method.

A. The Monte Carlo Method

Suppose that z is a random variable given by the probabilty density g(z) with corresponding distribution function

G(z), namely

$$g(z) = \frac{dG(z)}{dz} . \tag{1}$$

The probability density g(z) may be related to a uniform distribution f of the random variable x,

$$f(x) = \begin{matrix} 1 & 0 \leq x < 1 \\ 0 & \text{for all other } x \end{matrix} , \tag{2}$$

by the relation

$$f(x)dx = g(z)dz . \tag{3}$$

Integrating both sides of Eq. (3) yields

$$x = G(z) = \int_{-\infty}^{z} g(t)dt . \tag{4}$$

Now, if x is generated from the uniform distribution f and Eq. (4) is inverted to obtain z, then z will be distributed with probability density g(z). In this manner, any physical

process for which the probability density g(z) is known may be modelled using random numbers drawn from a uniform distribution.[2]

Since the probability density for the pions in the CM system is known, this method may be used to simulate π^0 production given computer generated quasi-uniform random numbers. Sections II and III detail how the Monte Carlo method is applied to pion production. To test the method, a fitting algorithm is applied in Section IV. Each portion begins with a development of the necessary formulae and computational techniques followed by computer analysis of the same.

Many of the techniques required are provided in part by the International Mathematical and Statistical Library (IMSL). These routines utilize some of the most current algorithms and numerical techniques which have been optimized with respect to the IBM 360/370 operating system. Documentation for the routines described is available from the Users Services Department of the Virginia Tech Computing Center. All other programs are given in the Appendices, with sample calculations and tests of the method where appropriate. The conclusion, at Section V, summarizes the results as well as suggesting some further applications of the method.

II. THE METHOD

A. Restrictions on the Angular Distribution Parmeters

The reaction may be illustrated schematically as

$$\begin{array}{l} p + N \rightarrow N' + \pi^{o} \\ \qquad\qquad\quad \downarrow \\ \qquad\qquad\quad 2\gamma \ . \end{array} \tag{5}$$

Let θ^{*}_{π} represent the polar angle of the π^{o} in the center of mass frame with respect to the incident proton direction. Then, for the case of unpolarized protons, the differential cross section for the π^{o} in the CM frame may be expanded as a second degree polynomial in $\cos\theta^{*}_{\pi}$,

$$\frac{d\sigma}{d\Omega} = a + b\cos\theta^{*}_{\pi} + c\cos^{2}\theta^{*}_{\pi} \ , \tag{6}$$

if S and P-waves are retained and all other (higher) partial waves are neglected. The angular distribution parameter a consists of both S and P-waves, b is composed of the interference between the S-wave and P-waves and c is entirely P-wave.[3] The distribution for the pion is thus specified by the angular distribution parameters a, b, and c. By definition, the pion distribution must be positive or zero over the entire range of the variable θ^{*}_{π}. Clearly, this restriction implies that certain choices for a, b, and c may not be allowed.

Let $z=\cos\theta^*_\pi$. Then Eq. (6) becomes

$$\frac{d\sigma}{d\Omega} = M(z) = a + bz + cz^2 \ , \tag{7}$$

where $z \in [-1,1]$. The requirement that the cross section be physically meaningful is simply

$$M(z) \geq 0 \text{ for all } z. \tag{8}$$

Since M(z) must be positive or zero for all z, it must certainly be positive or zero for some fixed value of z, namely z=0. Thus the parameter a must be non-negative.

M(z) is a continuous function of z on the interval from -1 to 1. Theorems from the calculus insure that M(z) must behave in one of three (mutually exclusive) manners:

1. M(z) has a critical point (extremum),
2. M(z) is monotone increasing or decreasing, or
3. M(z) is constant,

on the closed interval [-1,1] for fixed a,b and c.

Consider case 1. The condition for critical points in M(z) is determined by the second derivative of M(z) with respect to z:

if $\frac{d^2M}{dz^2} = 2c > 0$ -> a critical point exists and it is a minimum,

< 0 -> a critical point exists and it is a maximum,

= 0 -> M(z) is monotone increasing, decreasing or is constant.

Suppose that c is nonzero. Then a critical point z_c exists. Setting the first derivative of M(z) equal to zero yields the value of the critical point

$$\left.\frac{dM}{dz}\right|_{z_c} = b + 2cz_c = 0 \ , \tag{9}$$

and

$$z_c = \frac{-b}{2c} \ . \tag{10}$$

If c is strictly greater than zero, z_c is a minimum, and it is sufficient to require that

$$\begin{aligned} M(z_c) &= a + bz_c + cz_c^2 \\ &= a - \frac{b^2}{4c} \geq 0 \end{aligned} \tag{11}$$

to insure that M(z) is positive or zero. Recall that c represents the P-wave contribution to the cross section. For this reason, c must be non-negative. The condition at Eq. (11) may now be written

$$|b| \leq 2\sqrt{ac} \ . \tag{12}$$

Further, if c is chosen to be zero, b must in turn be zero for it represents the interference between the S and P-waves. M(z) will be monotone increasing or decreasing

only in the event that c is zero. However, since b is also zero, M(z) is a constant function equal to the parameter a. Earlier arguments show that the angular distribution parameter a is non-negative, hence M(z) is also.

The conditions on a,b and c to insure a physically meaningful distribution for the pions may be summarized as:

1. $a \geq 0$ (13)

2. $c \geq 0$ (14)

3. $|b| \leq 2\sqrt{ac}$. (15)

B. Generation of the Pion Angle in the CM

The probability of an event with $-1 \leq z < z_\pi$ is given by

$$P(z<z_\pi) = \frac{1}{M_o} \int_{-1}^{z_\pi} M(z)dz, \tag{16}$$

where

$$M_o = \int_{-1}^{1} M(z)dz$$

$$= 2(a + \frac{c}{3}). \tag{17}$$

Performing the integration at Eq. (16) yields a cubic equation in z_π,

$$(\frac{c}{3M_o})z_\pi^3 + (\frac{b}{2M_o})z_\pi^2 + (\frac{a}{M_o})z_\pi +$$

$$\{ \frac{1}{M_o}(a - \frac{b}{2} + \frac{c}{3}) - P \} = 0 \quad . \tag{18}$$

If a random number P is cast from a uniform distribution on the interval [0,1), and Eq. (18) is inverted to obtain z_π,

then the random variable z_π is determined which is distributed with probability density given at Eq. (7). Since z_π is defined on the interval [-1,1], the inverse cosine function is single valued and

$$\theta_\pi^* = \cos^{-1}(z_\pi) \tag{19}$$

is also a random variable with (unnormalized) probability density specified by Eq. (6).

This process may be readily adapted for computer simulation. P is obtained from a basic uniform pseudo-random number generator. What remains is inversion of the cubic equation in z_π.

When the parameter c is zero, b must also be zero and the cubic equation in z_π reduces to

$$az_\pi + (a - M_o P) = 0 . \tag{20}$$

But,

$$M_o = 2(a + \frac{c}{3}) = 2a .$$

Eq. (20) becomes

$$a\{z_\pi + (1 - 2P)\} = 0$$

$$\rightarrow z_\pi = 2P - 1 , \tag{21}$$

and thus z_π is uniformly distributed on the interval [-1,1) as expected.

In the event that c is nonzero, Eq. (18) may be written as

$$z_\pi^3 + qz_\pi^2 + rz_\pi + s = 0 \ , \tag{22}$$

where $q = \left(\frac{3b}{2c} \right)$

$$r = \left(\frac{3a}{c} \right)$$

$$s = \frac{3}{c} \left\{ \left(a - \frac{b}{2} + \frac{c}{3} \right) - M_o P \right\} . \tag{23}$$

Making the substitution

$$z_\pi = y - \frac{q}{3} \ ,$$

Eq. (22) may be reduced to the form

$$y^3 + ty + u = 0 \ , \tag{24}$$

where $t = \frac{1}{3} (3r - q^2)$

$$u = \frac{1}{27} (2q^3 - 9qr + 27s) \ . \tag{25}$$

If

$$A = \left\{ \frac{-u}{2} + \left(\frac{u^2}{4} + \frac{t^3}{27} \right)^{\frac{1}{2}} \right\}^{\frac{1}{3}}$$

$$\& \quad B = \left\{ \frac{-u}{2} - \left(\frac{u^2}{4} + \frac{t^3}{27} \right)^{\frac{1}{2}} \right\}^{\frac{1}{3}} \ , \tag{26}$$

then the roots to Eq. (24) are

$$y_1 = A + B$$
$$y_2 = \frac{-(A + B)}{2} + \frac{\sqrt{3}\ (A - B)i}{2}$$
$$y_3 = \frac{-(A + B)}{2} - \frac{\sqrt{3}\ (A - B)i}{2} \ . \qquad (27)$$

The type of roots are determined by the variable

$$\alpha = \frac{u^2}{4} + \frac{t^3}{27} \ . \qquad (28)$$

If

1. $\alpha > 0$ -> one real and two conjugate imaginary roots;
2. $\alpha = 0$ -> three real roots of which at least two are equal;
3. $\alpha < 0$ -> three real and unequal roots.

Up to this point, the problem of interpreting more than one value of z_π for a given value of P has been ignored. Consider α in terms of the angular distribution parameters to determine if some light may be shed on this dilemma. For one real root, α must be strictly positive. But,

$$t = \frac{1}{3}(3r - q^2)$$
$$= \frac{3}{c^2}(4ac - b^2) \ . \qquad (29)$$

The conditions needed to insure that the cross section be non-negative imply that t is strictly positive. Since u^2 is positive or zero for all values of u, α must be strictly positive. Thus a necessary and sufficient condition for one

real root to the cubic in z_π is a non-negative cross section.

One method of obtaining the required root z_π, is the simple yet tedious algebraic calculation which has already been outlined. Another method may be employed which is extremely fast, consistently provides whatever precision is desired (up to the limit of the machine) and is computationally simple - the method of "Regula Falsi" (false position). The description and figure which follow are adapted from Reference 4.

The method of Regula Falsi is illustrated geometrically in Figure 1. Suppose that there exists an interval I = [l,r] on which a function f at some point s is zero. Further suppose that f(l)f(r) < 0. Construct a line from f(l) to f(r) and note that at some point, say x_1, this line intersects the x-axis, namely at

$$x_1 = \frac{af(b) - bf(a)}{f(b) - f(a)} \quad . \tag{30}$$

x_1 is then the first approximation to the root s. Now, the interval I has been subdivided into 2 intervals, one of which contains the root s. The desired interval I_1 is the one for which f(x)f(e) < 0, where e is one of the endpoints l or r. Continuing in this manner, a sequence of intervals I_n are generated each of which contains the desired root and

has associated with it an estimate x_n of the root. The procedure is continued until x_n is sufficiently close to s. Note that the lengths of the intervals I_n in general do not converge to zero. However, it can be shown that Regula Falsi converges for all continuous functions for which $f(l)f(r) < 0$.

This method may be applied to the cubic equation in z_π, provided points l and r may be found such that $f(l)f(r) < 0$. Consider then the function

$$f(z_\pi) = \left(\frac{c}{3M_o}\right)z_\pi^3 + \left(\frac{b}{2M_o}\right)z_\pi^2 + \left(\frac{a}{M_o}\right)z_\pi + \left\{\frac{1}{M_o}\left(a - \frac{b}{2} + \frac{c}{3}\right) - P\right\} . \tag{31}$$

It can be shown that l=-1, r=1 satisfy the condition above.

Suppose first that the parameter c is zero. As stated earlier, the parameter b must also vanish. Under these conditions Eq. (31) reduces to

$$f(z_\pi) = \frac{z_\pi}{2} + \tfrac{1}{2} - P . \tag{32}$$

f(-1) is then

$$f(-1) = -P \leq 0 , \tag{33}$$

and f(1) is

$$f(1) = 1 - P > 0 \tag{34}$$

which satisfies the required condition.

Suppose on the other hand that c is strictly greater than zero. f(-1) is determined from Eq. (31) to be

$$f(-1) = -P \leq 0 \tag{35}$$

and f(1) is

$$f(1) = 1 - P > 0 \, . \tag{36}$$

Again, $f(-1)f(1) < 0$, as required.

Investigation of the conditions under which Regula Falsi may be employed has yielded a further bonus. It was first established that only one root to the cubic in z_π was required, but no consideration was given to the magnitude of that root! Since $f(-1)f(1) < 0$ for all allowed choices of the parameters, the root has now been established to fall in the required interval.

The method of Regula Falsi is implemented through the IMSL routine ZXMIN. To illustrate the desirability of this method, 5000 events were generated for the case of a proton incident upon a carbon target at 147 MeV with

$$a = 0.57$$

$$b = 0.41$$

$$c = 0.24.$$

ZXMIN was requested to either provide the root to 16 signi-

ficant digits (the difference between the final root selected and the root from the last iteration agree to 16 significant digits) or terminate execution when

$$|f(z_\pi)| \leq 10^{-16} .$$

The maximum number of iterations for the 5000 events was 11, with an average of approximately 7.6 iterations per event. The maximum in absolute value of $f(z_\pi)$ for the 5000 events was

$$\text{MAX} \{ |f(z_\pi)| \} \approx 0.67 \times 10^{-15} .$$

C. Reaction Kinematics

The angular distribution for the pion is independent of the azimuthal angle ϕ_π^*, so ϕ_π^* may be generated from a uniform distribution on the angular interval from 0 to 2π radians. Having already determined the polar angle of the π^o in the CM system, its direction is entirely specified. At this point, the energy and momenta of the pion in the CM frame, velocity of the CM in the lab frame, and the Lorentz factor are required to make the transformation of the pion angles from the CM to the lab frame. Given the masses of the constituents and the kinetic energy of the incident proton for the reaction (5), relativistic kinematics may be employed to define the required quantities.[5]

Let M_s denote the mass of particle s and T_p denote the kinetic energy of the proton. Then the total CM energy is given by

$$S = \sqrt{(M_p + M_N)^2 + 2T_p M_N} \quad . \tag{37}$$

The velocity of the CM in the lab frame is

$$\beta_c = \frac{\sqrt{T_p(T_p + 2M_p)}}{(M_p + M_N + T_p)} \quad , \tag{38}$$

and the corresponding Lorentz factor may in turn be written in the form

$$\gamma_c = \frac{M_p + M_N + T_p}{S} \quad . \tag{39}$$

If quantities referring to an observer in the CM frame are labeled with the superscript "*", then the pion CM momentum and energy are defined by

$$p_\pi^* = \frac{1}{2S}\sqrt{(S + M_\pi + M_N')(S + M_\pi - M_N')(S - M_\pi + M_N')} \; x \tag{40}$$

$$\sqrt{(S - M_\pi - M_N')}$$

and

$$E_\pi^* = \sqrt{p_\pi^{*2} + M_\pi^2} \tag{41}$$

respectively.

Equations (37) through (41) are calculated via the subroutine "KINMAT". A summary of the routine, a listing of the FORTRAN source code and an example illustrating its use

may be found at Appendix A.

D. Transformation of the π^o Angles from the CM to the Lab

The π^o is moving in the CM system at a polar angle θ_π^* with momentum p_π^* and total energy E_π^*. Given the velocity of the CM in the lab frame β_c and the Lorentz factor γ_c determined earlier, the corresponding quantities observed in the lab frame are specified by [6]

$$p_\pi \sin\theta_\pi = p_\pi^* \sin\theta_\pi^* \tag{42}$$

$$p_\pi \cos\theta_\pi = \gamma_c (p_\pi^* \cos\theta_\pi^* + \beta_c E_\pi^*) \tag{43}$$

$$E_\pi = \gamma_c (\beta_c p_\pi^* \cos\theta_\pi^* + E_\pi^*) \tag{44}$$

$$\phi_\pi = \phi_\pi^* . \tag{45}$$

To implement this transformation on a computer, the case for $\theta_\pi^* = 0$ or π must be considered separately. When the CM angle of the pion is 0 or π, the angle in the lab frame is also 0 or π, respectively. The momentum in the lab frame may then be determined from Eq. (43)

$$p_\pi = \gamma_c (p_\pi^* \cos\theta_\pi^* + \beta_c E_\pi^*)/\cos\theta_\pi \tag{46}$$

and the total energy from Eq. (44).

For all other θ_π^* the angle of the pion in the lab frame is obtained by dividing Eq. (42) by Eq. (43), i.e.

$$\tan\theta_\pi = \frac{p_\pi^* \sin\theta_\pi^*}{\gamma_c (p_\pi^* \cos\theta_\pi^* + \beta_c E_\pi^*)} \quad , \tag{47}$$

or

$$\theta_\pi = \tan^{-1} \left\{ \frac{p_\pi^* \sin\theta_\pi^*}{\gamma_c (p_\pi^* \cos\theta_\pi^* + \beta_c E_\pi^*)} \right\} \quad . \tag{48}$$

The inverse tangent function is single valued on the interval from $-\pi/2$ to $\pi/2$, however θ_π is expected to be in the interval $[0,\pi]$. Since the tan(angle) = tan(angle $+\pi$), the negative angles may be mapped onto the branch above the principal branch $[-\pi/2, \pi/2]$ (see Figure 2). The denominator of Eq. (48) may vanish (for rather uncharacteristic cases), so a divide check is required. If the denominator does vanish, the lab angle of the pion is simply $\pi/2$. The remainder of the calculation is unchanged. Eq. (42) then provides the momentum in the lab frame and, as before, the total energy is subsequently determined from Eq. (44).

The azimuthal angle ϕ_π^* is always perpendicular to the boost, hence it is unaltered by the transformation (recall Eq. 45).

LTRANS, a subroutine detailed at Appendix B, provides the Lorentz transformation of the pion angles from the CM to

the lab frame.

E. The Second Monte Carlo - Generation of the Gamma Angles

Let $\theta^*_{\gamma 1}$ and $\theta^*_{\gamma 2}$ be the polar angles of the gammas with respect to the pion rest frame. Note that in this frame the gammas come out back-to-back. Thus if one of the polar angles, say $\theta^*_{\gamma 1}$, is randomly generated on $[0, \pi]$, the other is determined by

$$\theta^*_{\gamma 2} = \pi - \theta^*_{\gamma 1} . \tag{49}$$

Further, the azimuthal angles $\phi^*_{\gamma 1}$ and $\phi^*_{\gamma 2}$ must be chosen so that

$$\Phi^*_{\gamma} = \phi^*_{\gamma 2} - \phi^*_{\gamma 1} = \pi \tag{50}$$

(see Figure 3). The corresponding process for generating the azimuthal angles is then to generate $\phi^*_{\gamma 1}$ in the closed interval $[0, \pi]$, whence

$$\phi^*_{\gamma 2} = \phi^*_{\gamma 1} + \pi . \tag{51}$$

The gamma angles previously obtained may now be transformed into the lab frame of the pion. The required Lorentz transformation is identical to the one utilized earlier if the following identifications are made:

1. The momentum and energy for the π^0 are now the cor-

responding quantities for the gammas, namely

$$p^*_\pi \to p^*_{\gamma 1} = p^*_{\gamma 2}$$
$$E^*_\pi \to E^*_{\gamma 1} = E^*_{\gamma 2} = M_\pi/2 \ ; \tag{52}$$

2. The velocity of the CM in the lab frame β_c and the Lorentz factor γ_c are now with respect to the π^o . In other words,

$$\beta_c = \beta_\pi = \frac{p_\pi}{E_\pi}$$
$$\gamma_c = \gamma_\pi = (1 - \beta_\pi^2)^{-\frac{1}{2}} \ ; \tag{53}$$

3. Again, since the boost is in the z direction, the azimuthal angles are unaltered

$$\phi'_{\gamma 1} = \phi^*_{\gamma 1}$$
$$\phi'_{\gamma 2} = \phi^*_{\gamma 2} \ . \tag{54}$$

The opening angle for the gammas, ψ , is defined by

$$\cos\psi = \cos\theta_{\gamma 1}\cos\theta_{\gamma 2} + \sin\theta_{\gamma 1}\sin\theta_{\gamma 2}\cos(\phi_{\gamma 2} - \phi_{\gamma 1}) \ . \tag{55}$$

In the CM frame, the opening angle is then

$$\cos\psi^* = \cos\theta^*_{\gamma 1}\cos(\pi - \theta^*_{\gamma 1}) + \sin\theta^*_{\gamma 1}\sin(\pi - \theta^*_{\gamma 1})\cos(\pi) \tag{56}$$

and utilizing the required trigonometric identities obtains

$$\cos\psi^* = -1 \ . \tag{57}$$

Taking the inverse cosine reveals that the opening angle in

the CM frame is π . This is precisely what is meant by the statement that the gammas are produced back-to-back. The effect of the Lorentz transformation is to squeeze the gammas in the forward direction (see Figure 4).

F. A Final Rotation

The gamma angles just found were determined with respect to the pion lab frame, not (!) the lab frame of the reaction. A final rotation must be performed to obtain the desired gamma angles in the lab.

Denote the lab frame of the π^o by the primed coordinate system (x',y',z'), where the π^o direction in the lab is coincident with the z'-axis. Denote the lab frame of the reaction by the unprimed coordinate system (x,y,z). Figure 5 details the relevant angles. Note that the x-axis is chosen perpendicular to the plane formed by the z and the z' axes such that a right hand screw will advance z into z'. The angle between the x and x'-axis is then $\pi/2 + \phi_\pi$.

Let the direction of the gamma in the pion lab frame be given by

$$\hat{k}'_\gamma = \sin\theta'_\gamma\cos\phi'_\gamma\hat{x}' + \sin\theta'_\gamma\sin\phi'_\gamma\hat{y}' + \cos\theta'_\gamma\hat{z}' \ . \qquad (58)$$

The transformation matrix A is required such that

$$\hat{k}_\gamma = A\hat{k}'_\gamma \ , \qquad (59)$$

where $\hat{k}_\gamma$ is the direction of the gamma in the lab frame of the reaction. To find A, consider a series of two rotations which send the primed coordinates into the unprimed coordinates:

1. The matrix F which provides a rotation about the x'-axis by $-\theta_\pi$ resulting in the alignment of the z and z' axes.

$$F = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(-\theta_\pi) & \sin(-\theta_\pi) \\ 0 & -\sin(-\theta_\pi) & \cos(-\theta_\pi) \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_\pi & -\sin\theta_\pi \\ 0 & \sin\theta_\pi & \cos\theta_\pi \end{pmatrix}; \qquad (60)$$

2. The matrix E which performs a rotation about the z,z'-axis (now coincident) by $-(\phi_\pi + \pi/2)$ to align the x and x' axes.

$$E = \begin{pmatrix} \cos\{-(\phi_\pi + \frac{\pi}{2})\} & \sin\{-(\phi_\pi + \frac{\pi}{2})\} & 0 \\ -\sin\{-(\phi_\pi + \frac{\pi}{2})\} & \cos\{-(\phi_\pi + \frac{\pi}{2})\} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} -\sin\phi_\pi & -\cos\phi_\pi & 0 \\ \cos\phi_\pi & -\sin\phi_\pi & 0 \\ 0 & 0 & 1 \end{pmatrix}. \qquad (61)$$

The matrix A is then

$$A = EF$$

$$= \begin{pmatrix} -\sin\phi_\pi & -\cos\phi_\pi & 0 \\ \cos\phi_\pi & -\sin\phi_\pi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_\pi & -\sin\theta_\pi \\ 0 & \sin\theta_\pi & \cos\theta_\pi \end{pmatrix}$$

$$= \begin{pmatrix} -\sin\phi_\pi & -\cos\phi_\pi\cos\theta_\pi & \cos\phi_\pi\sin\theta_\pi \\ \cos\phi_\pi & -\sin\phi_\pi\cos\theta_\pi & \sin\phi_\pi\sin\theta_\pi \\ 0 & \sin\theta_\pi & \cos\theta_\pi \end{pmatrix} \qquad (62)$$

and Eq. (59) implies

$$\begin{pmatrix} \sin\theta_\gamma\cos\phi_\gamma \\ \sin\theta_\gamma\sin\phi_\gamma \\ \cos\theta_\gamma \end{pmatrix} = A \begin{pmatrix} \sin\theta'_\gamma\cos\phi'_\gamma \\ \sin\theta'_\gamma\sin\phi'_\gamma \\ \cos\theta'_\gamma \end{pmatrix}$$

$$= \begin{pmatrix} -\sin\phi_\pi\sin\theta'_\gamma\cos\phi'_\gamma - \cos\phi_\pi\cos\theta_\pi\sin\theta'_\gamma\sin\phi'_\gamma \\ \qquad + \cos\phi_\pi\sin\theta_\pi\cos\theta'_\gamma \\ \cos\phi_\pi\sin\theta'_\gamma\cos\phi'_\gamma - \sin\phi_\pi\cos\theta_\pi\sin\theta'_\gamma\sin\phi'_\gamma \\ \qquad + \sin\phi_\pi\sin\theta_\pi\cos\theta'_\gamma \\ \sin\theta_\pi\sin\theta'_\gamma\sin\phi'_\gamma + \cos\theta_\pi\cos\theta'_\gamma \end{pmatrix} . \qquad (63)$$

The gamma angles are determined from the coordinates of $\hat{k}_\gamma$, namely

$$\cos\theta_\gamma = (\hat{k}_\gamma)_z = \sin\theta_\pi\sin\theta'_\gamma\sin\phi'_\gamma + \cos\theta_\pi\cos\theta'_\gamma$$

$$\tan\phi_\gamma = \frac{(\hat{k}_\gamma)_y}{(\hat{k}_\gamma)_x} . \qquad (64)$$

Subroutine ROTATE finds the coordinates of both gammas in the lab frame of the reaction by calculating the matrix A

and multiplying A by the vector which specifies the respective gamma in the π^o lab frame. ROTATE then finds the polar angle of the gamma utilizing the inverse cosine function. Since the inverse tangent returns angles in the range $[-\pi/2, \pi/2]$, the azimuthal angle may not be determined directly from Eq. (64). A simple algorithm to circumvent this difficulty is based upon which quadrant the x and y components fall within. Figure 6 indicates the quadrants in the xy plane and the 4 cases which must be dealt with. Define

$$\phi^{\dagger} = \tan^{-1}\left(\frac{y}{x}\right) , \tag{65}$$

and recall that

$$\tan^{-1}\left(\frac{-y}{x}\right) = -\tan^{-1}\left(\frac{y}{x}\right) . \tag{66}$$

Thus the azimuthal angle is given by

$$\begin{aligned} \phi_{\gamma} &= \phi^{\dagger} && : \text{ in the } 1^{st} \text{ quadrant} \\ &= \pi + \phi^{\dagger} && : \text{ in the } 2^{nd} \text{ and } 3^{rd} \text{ quadrants} \\ \& \quad &= 2\pi + \phi^{\dagger} && : \text{ in the } 4^{th} \text{ quadrant.} \end{aligned}$$

This algorithm is provided by the subroutine INVTAN described at Appendix D.

III. PUTTING IT ALL TOGETHER - THE PROGRAM CARLOS

Now that the method has been described at length, it is time to consolidate the pieces. The program CARLOS provides the skeleton from which all the previously described subroutines are managed. See Appendix E for a listing of the program.

A. Input Requirements

CARLOS requires the input be resident on FORTRAN unit 5 in the following format:

```
(SEED) (A) (B) (C) (NPNTS) (NCDEV) (NDDEV)
(NNAME) (NPNAME)
(NZ) (NPZ) (NA) (NPA) (MASXN) (MASXNP) (TP) (NPRN) (NDEV)
```

where

SEED	Starting seed for the Monte Carlo (double precision INTEGER)
A, B, C	. Angular Distribution Parameters (REAL*8)
NPNTS	Number of data points to be generated (INTEGER)
NCDEV	FORTRAN unit number of output device for the console file (INTEGER)
NDDEV	FORTRAN unit number of output device for the data file (INTEGER)
NNAME	Name (2 alphanumeric characters) which identifies target (INTEGER)

NPNAME	Name (2 alphanumeric characters) which identifies residual nucleus (INTEGER)
NZ	Atomic number of target nucleus (INTEGER)
NPZ	Atomic number of residual nucleus (INTEGER)
NA	Mass number of target (INTEGER)
NPA	Mass number of residual nucleus (INTEGER)
MASXN	Mass excess of target in MeV (REAL*8)
MASXNP	Mass excess of residual nucleus in MeV (REAL*8)
TP	Incident proton's kinetic energy in MeV (REAL*8)
NPRN	Output option: NPRN=0 -> do not write out parameters NPRN=1 -> write parameters to device specified by NDEV (INTEGER)
NDEV	FORTRAN unit number of output device for description of kinematics calculation (INTEGER)

The first and third records above are scanned by an unformatted read statement. The second record is handled by a formatted read statement.

B. Output Format

Two output files are generated by CARLOS: a 'console' file and the data file. The console file records the infor-

mation which characterizes the data generation (the input parameters, kinematics, and the final seed for the Monte Carlo). The data file contains the angles which comprise the gamma distribution in the lab frame of the reaction. Structurally, the file consists of an initial record which indicates the number of events followed by the gamma angles. Events, comprised of a pair of gamma angles, occur in two sequential records. The first record corresponds to one pair of gamma angles and the second to the other pair. Angles stored in the data file are provided by an unformatted write statement, hence they should be read in the same manner to avoid possible confusion or errors.

C. Sample Input and Output Files

This example generates 10 events for a proton incident on a carbon target with kinetic energy 147 MeV.

Input: (at FORTRAN unit 5)

```
card
column
         1         2         3         4         5         6
1...5....0....5....0....5....0....5....0....5....0....5....0

421909 0.57 0.41 0.24 10 2 7
 C  N
6 7 12 13 0.0 5.345 147.0 1 2
```

Output:

Console File (at FORTRAN unit 2)

```
 MONTE CARLO SIMULATION OF PIO PRODUCTION NEAR THRESHOLD

 STARTING SEED FOR MONTE CARLO:    421909.0000
    NUMBER OF EVENTS REQUESTED:     10
      CROSS SECTION PARAMETERS:
                                A:    .5700000
                                B:    .4100000
                                C:    .2400000

            RELATIVISTIC KINEMATICS

    12 C   (    P     ,   PIO    )    13 N
  11174.9532( 938.2796, 134.9626) 12111.2888

 PROTON KINETIC ENERGY:    147.0000     MEV
        MASS EXCESS
                     C:      0.0        MEV
                     N:      5.3450     MEV

              THRESHOLD:    144.9789     MEV
       TOTAL CM ENERGY:  12248.0956     MEV
       PIO CM MOMENTUM:     22.2619     MEV/C
       PIO CM   ENERGY:    136.7863     MEV
 VELOCITY OF CM IN LAB:      0.0444854 C
        LORENTZ FACTOR:      1.0009909


   FINAL SEED FROM MONTE CARLO:    1445744356.
```

Date File (at FORTRAN unit 7)

```
        10
.494194070566698465       5.41334897863597253
2.774041527918351 68      1.583181382164571 72
1.127232373705361 33      .425089449125203148
1.972951070589670 90      3.922855929368648 69
1.229015033005654 40      2.216084659392016 71
2.006223933892129 05      5.377771267788612 47
.393304324696188923       2.477485886196768 35
2.755751354052305 09      6.101161451780848 12
1.150465841937396 54      2.193433552725001 07
1.581671925800569 59      5.379738017147855 04
.780539947597750458       3.561127864748812 04
2.629460616740581 40      .354889531820728715
.636670289260325400       .726323684672041406
2.566032607979981 56      4.343429859640931 53
1.028889642789466 49      5.988116040067632 98
1.735652290030731 63      2.997861424364192 34
1.456421297649696 63      3.155085140934218 88
1.649110357233211 31      5.974662700776164 79
.888169920480567754       6.270470891469573 48
1.862008602782154 74      3.032153870141833 80
```

IV. A TEST OF THE METHOD

Now that a gamma distribution may be generated, a method is required to verify that the generated distribution corresponds to the expected theoretical distribution. What then is the theoretical distribution and how may the distributions be compared?

The pion angular distribution may be transformed into a distribution for the coincidence gammas in the lab frame. The resulting gamma distribution is provided at Reference 1. Reference 1 also suggests a method of fitting the generated distribution using the method of maximum likelihood. The fitting process determines the ratios of the parameters b/a and c/a as well as their error. A means of normalizing the parameters to obtain a, b, and c is outlined, but for the purposes of this work the ratios of the parameters is sufficient.

Nine data files, consisting of 1000 events each, were generated using different starting seeds for a proton with kinetic energy 147 MeV incident on a carbon target. The angular distribution parameters were fixed at the following values:

a = 0.57

b = 0.41

c = 0.24.

The results of the fitting are given at Table I and displayed graphically at Figures 7 (b/a) and 8 (c/a).

The errors provided correspond to one standard deviation (denoted by σ). Statistical theory suggests that the fitted values should be normally distributed about the expected value. That is, the probability that a single trial lies within one standard deviation of the expected value is to be approximately 68.2%, within two standard deviations 95.4%, etc. For the nine trials performed, at least 6 should be within 1σ of the expected value and all but possibly 1 should be within 2σ. Figures 7 and 8 indicate that the fitted parameters are in agreement with the expected results subject to the confines of statistical theory.

V. CONCLUSIONS

A method has been developed for the simulation of π^0 production near threshold. The fitting method described at Reference 1 was applied and the resulting ratios of the parameters agrees extremely well with the expected results.

The questions raised in the INTRODUCTION concerning the measuring scheme for the gammas may now be dealt with. To test the effect of finite counter size, one need only select from the gamma angles generated by CARLOS those which fall into the region occupied by the counters. The resulting distribution may be fitted in the manner used in Section IV. Reference 1 also describes a procedure for simulating and correcting for effects of finite angular resolution. Again, the program CARLOS may be employed with minor modifications.

REFERENCES

1. See for example: D.A. Jenkins and M. Madden, The Analysis of (p, π^{o}) Reactions (to be published)

2. See for example: Jerome Spanier and Ely M. Gelbard, Monte Carlo Principles and Neutron Transport Problems (Addison-Wesley, Reading, Massachusetts, 1969)

3. Gunar Kallen, Elementary Particle Physics (Addison-Wesley, 1964), pp. 79-90

4. Lee W. Johnson and R. Dean Reese, Numerical Analysis (Addison-Wesley, 1977), pp. 115-117

5. See for example: W.S.C. Williams, Relativistic Kinematics

6. Ibid

Table I

Ratios of the Fitted Parameters and Their Errors

Trial	Seed	B/A	Error	C/A	Error
1	421909	0.70	(0.08)	0.29	(0.16)
2	1556907563	0.82	(0.09)	0.52	(0.17)
3	1416370126	0.71	(0.08)	0.31	(0.14)
4	906260504	0.78	(0.10)	0.87	(0.24)
5	839251678	0.62	(0.08)	0.26	(0.16)
6	1493298135	0.85	(0.10)	0.71	(0.21)
7	1860202464	0.64	(0.08)	0.25	(0.16)
8	600132954	0.72	(0.08)	0.33	(0.15)
9	1890156240	0.70	(0.08)	0.29	(0.14)

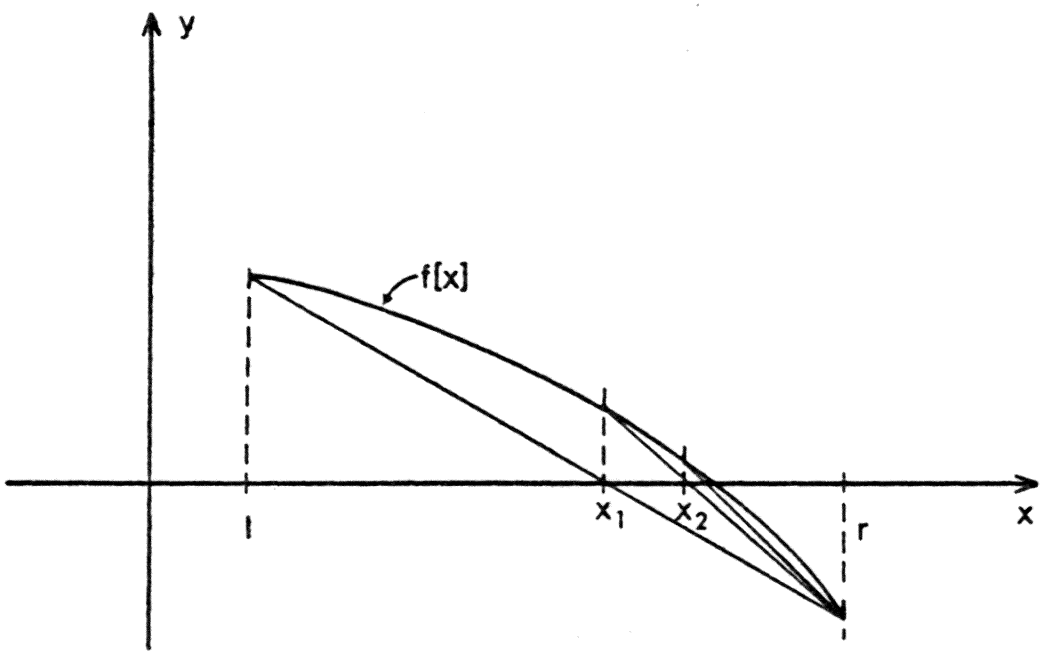

Figure 1. The method of Regula Falsi.

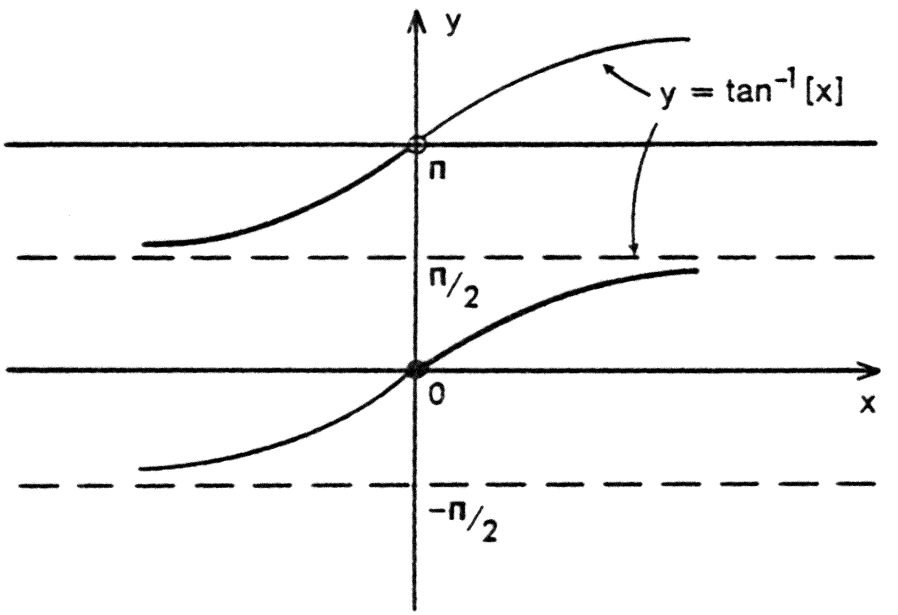

Figure 2. The inverse tangent function. Illustration depicts 2 branches with the bold line indicating the desired range of the function.

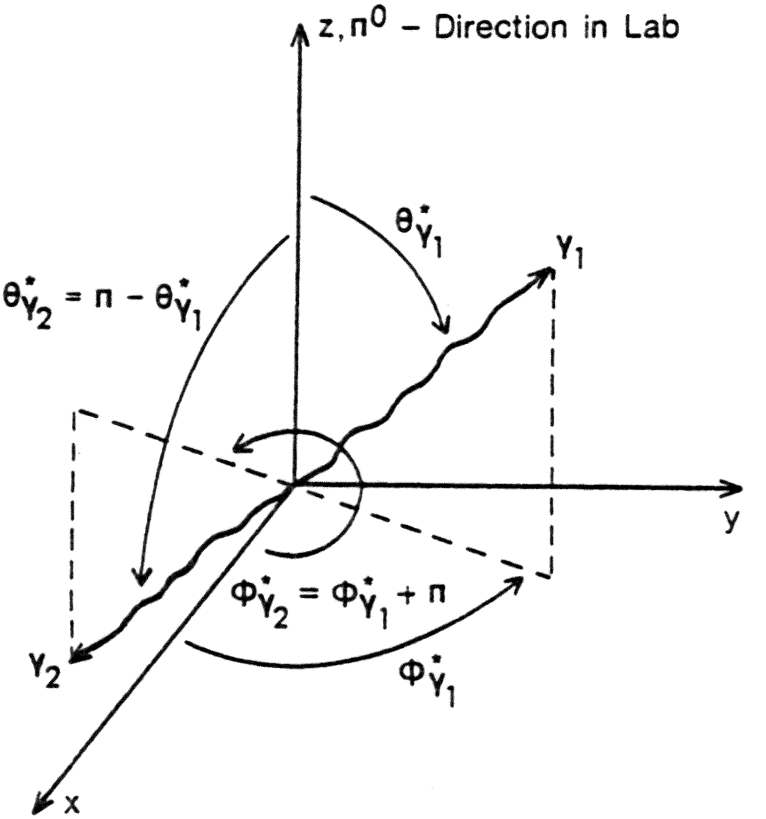

Figure 3. The gammas are back-to-back in the CM frame of the pion.

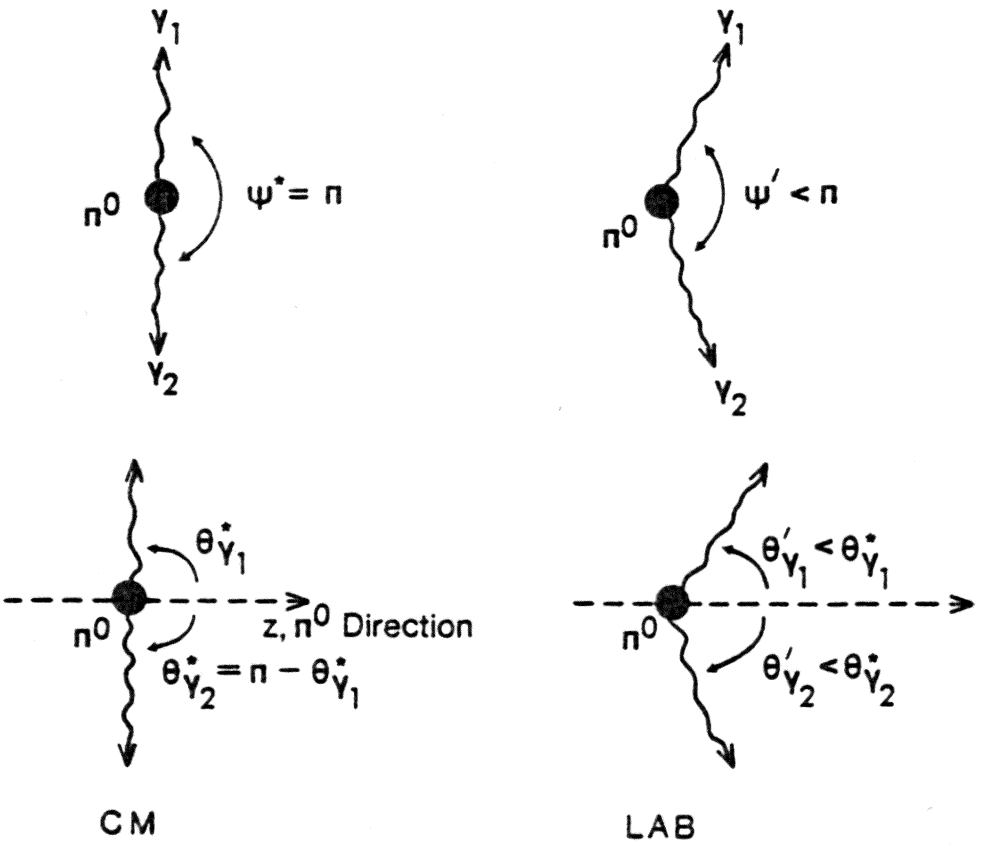

Figure 4. Relation between the gamma angles in the CM and lab frame of the pion.

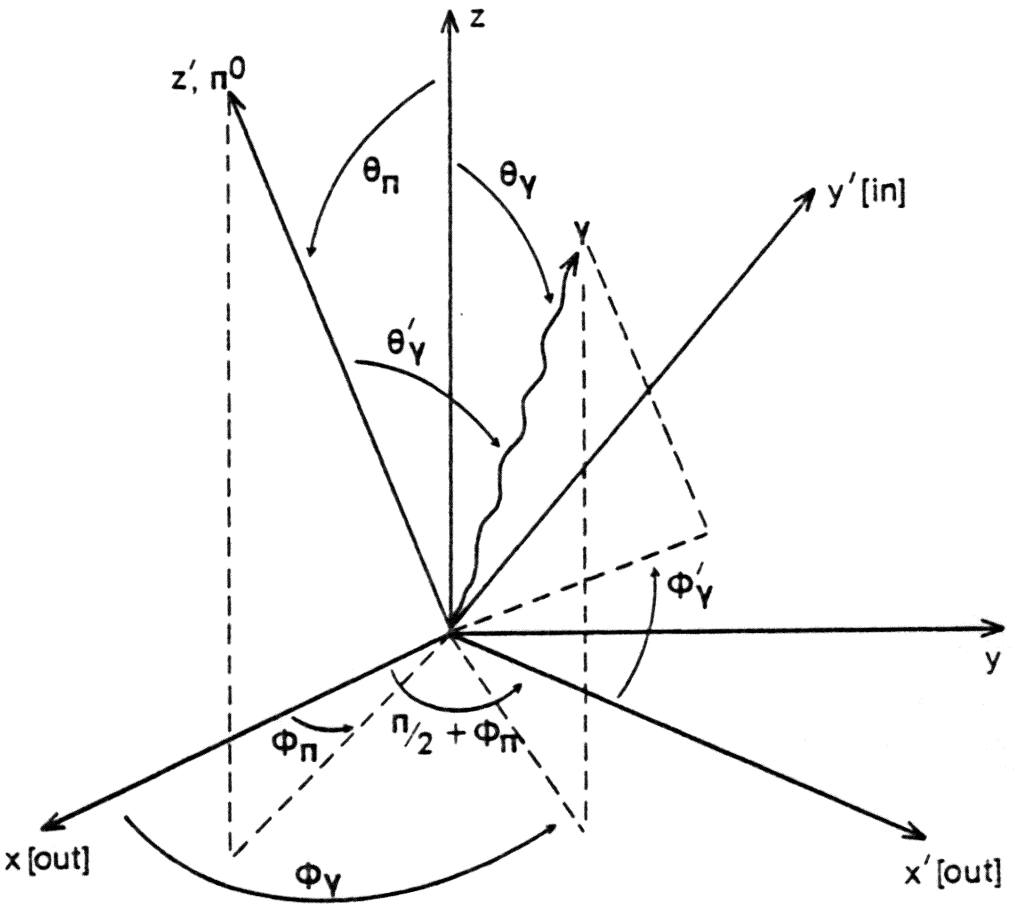

Figure 5. One of the two gammas depicting the angles with respect to the lab frame of the pion and the lab frame of the reaction.

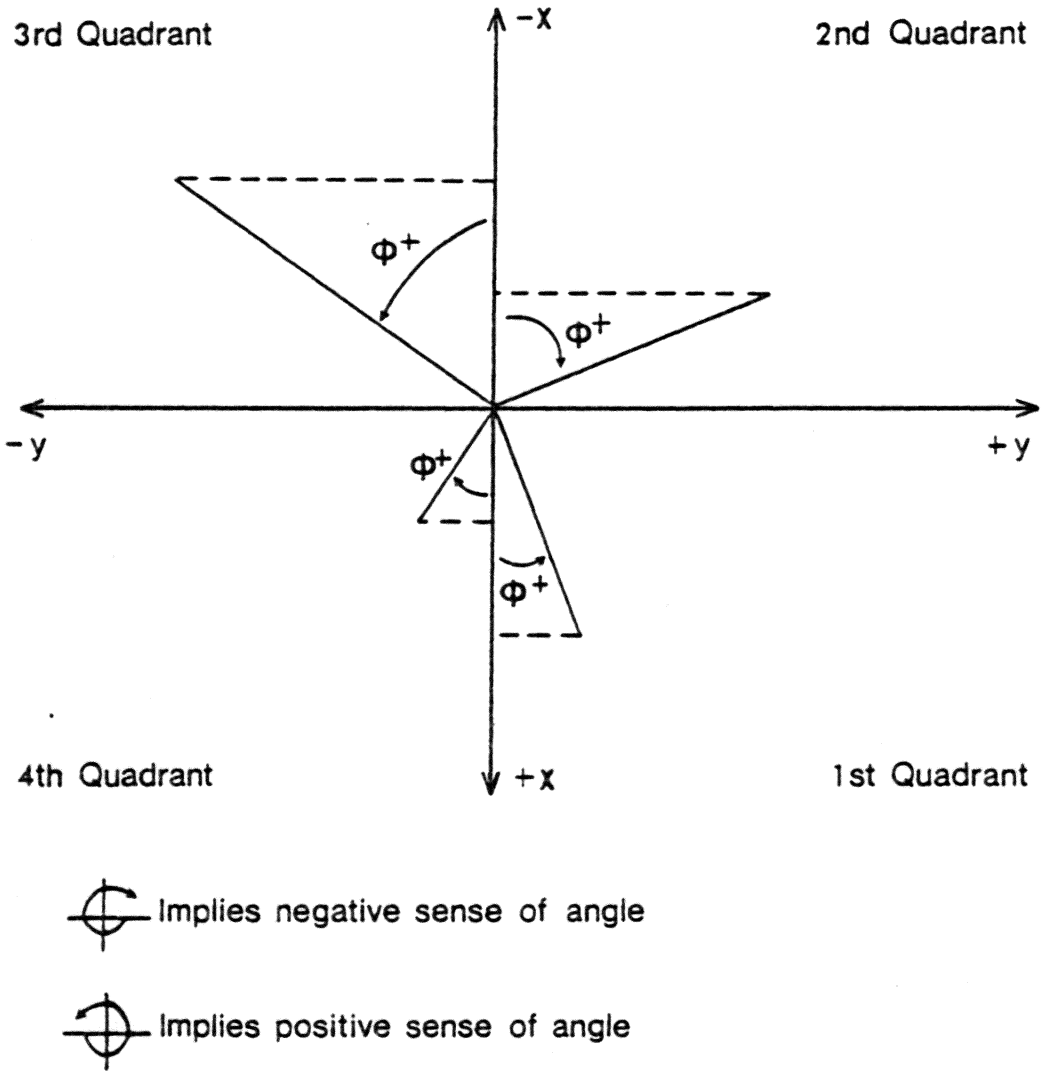

Figure 6. The angle $\phi^{\dagger}$ and the four quadrants in the xy plane.

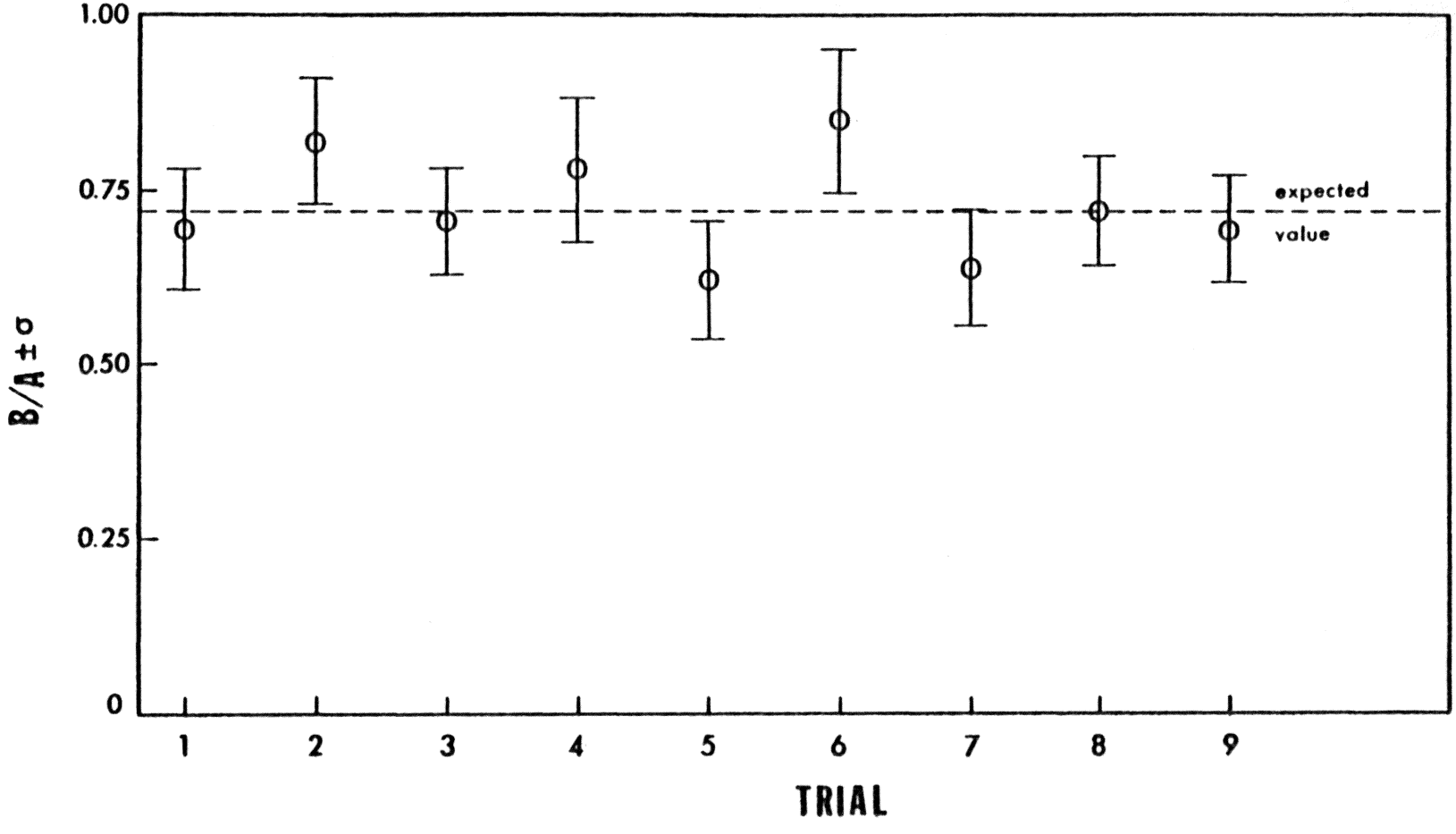

Figure 7. The parameter b/a for a number of trials (different starting seeds).

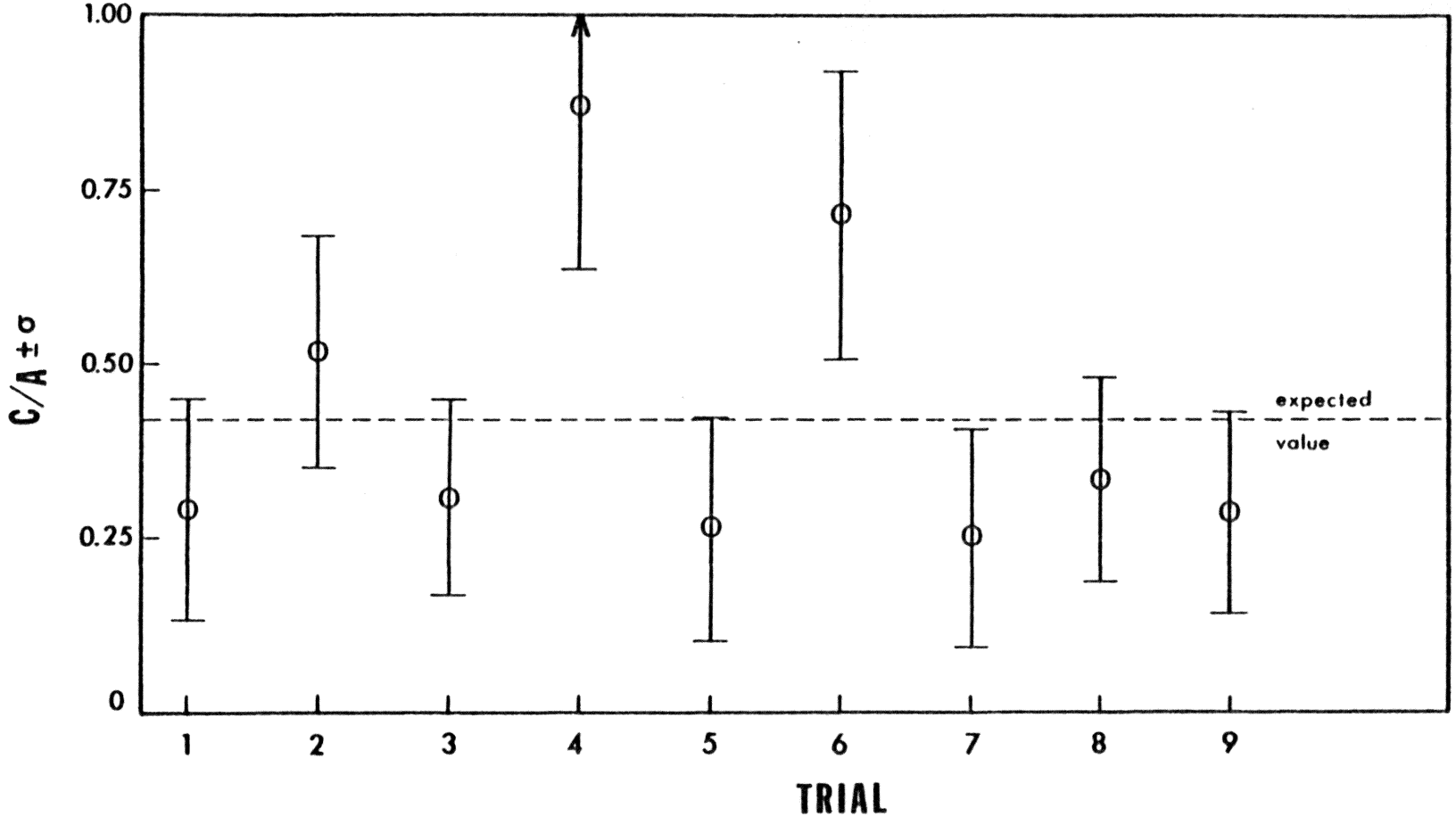

Figure 8. The parameter c/a for a number of trials.

APPENDIX A - THE SUBROUTINE KINMAT

1. Program Summary

* Name: KINMAT

* Description/Purpose: Calculates the following kinematic parameters for the reaction $p + N \rightarrow N' + \pi^0$:

  - mass of target
  - mass of residual nucleus
  - threshold for π^0 production
  - total CM energy
  - CM energy of π^0
  - CM momentum of π^0
  - velocity of CM in the lab
  - Lorentz factor,

  given the atomic number (Z), the mass number (A), the mass excess in MeV for both the target (N) and the residual nucleus (N'), and the kinetic energy of the incident proton in MeV.

* Precision: Double

* Calling Format: CALL KINMAT

* Common Areas and Data Initialization:

/INKIN/ Parameters which define the reaction and any i/o requirements (input)

```
COMMON/INKIN/NNAME,NPNAME,NZ,NPZ,NA,NPA,MASXN,MASXNP,
             TP,NPRN,NDEV
```

/KINOUT/ Desired kinematic quantities (output)

```
COMMON/KINOUT/MPIO,CMEPIO,CMPPIO,CGAMMA,CBETA
```

* Description of Variables: (in order of appearance)

AMU	energy equivalent to atomic mass unit in MeV/c/c (REAL*8)
ME	Electron mass in MeV/c/c (REAL*8)
MP	Proton mass in MeV/c/c (REAL*8)
MPI	π^0 mass in MeV/c/c (REAL*8)
NNAME	Name (2 alphanumeric characters) which identifies target (input/INTEGER)
NPNAME	Name (2 alphanumeric characters) which identifies residual nucleus (input/INTEGER)
NZ	atomic number of target nucleus (input/INTEGER)
NPZ	atomic number of residual nucleus (input/INTEGER)
NA	mass number of target (input/INTEGER)
NPA	mass number of residual nucleus (input/INTEGER)
MASXN	Mass excess of target in MeV (input/REAL*8)

MASXNP — Mass excess of residual nucleus in MeV (input/REAL*8)

TP — Incident proton's kinetic energy in MeV (input/REAL*8)

NPRN — Input/output option:
NPRN=0 -> do not write out parameters
NPRN=1 -> write parameters to device specified by NDEV
(input/INTEGER)

NDEV — FORTRAN unit number for output device (input/INTEGER)

MPIO — π^0 mass in MeV/c/c (output/REAL*8)

CMEPIO — CM energy of π^0 in MeV (output/REAL*8)

CMPPIO — CM momentum of π^0 in MeV/c (output/REAL*8)

CGAMMA — Lorentz factor (output/REAL*8)

CBETA — Velocity of CM in lab in units of c (speed of light) (output/REAL*8)

MN — Mass of target nucleus in MeV/c/c (REAL*8)

MNP — Mass of residual nucleus in MeV/c/c (REAL*8)

All other variables are either defined internally or their function may be determined from context.

* Programming Notes and Suggestions:

1. The values of the following physical constants were taken from the Particle Data Group Table of Particle Properties (Rev. Mod. Phys., Vol. 52, No 2, Part II, April 1980):

AMU	931.5016	MeV/c/c
ME	0.5110034	MeV/c/c
MP	938.2796	MeV/c/c
MPI	134.9626	Mev/c/c

2. The mass of the respective nucleus is determined by multiplying the mass number by the atomic mass unit, subtracting the mass of the electrons in the atom (subtracting Z multiplied by the electron mass) and adding the mass excess. These quantities are readily available in standard isotope tables.

2. Program Listing

```
      SUBROUTINE KINMAT
      IMPLICIT REAL*8(A-H,M,O-Z)
C*
C* SUBROUTINE CALCULATES KINEMATIC PARAMETERS FOR THE
C* REACTION
C*           P + N -> N' + PIO:
C*   - MASS OF TARGET
C*   - MASS OF RESIDUAL NUCLEUS
C*   - THRESHOLD FOR PIO PRODUCTION
C*   - TOTAL CM ENERGY
C*   - CM ENERGY OF PIO
C*   - CM MOMENTUM OF PIO
C*   - VELOCITY OF CM IN THE LAB
C*   - LORENTZ FACTOR,
C* GIVEN THE ATOMIC NUMBER (Z), THE MASS NUMBER (A),
C* THE MASS EXCESS IN MEV FOR BOTH THE TARGET (N) AND THE
C* RESIDUAL NUCLEUS (N'), AND THE KINETIC ENERGY OF THE
C* INCIDENT PROTON IN MEV.  FOR A DESCRIPTION OF THE
C* VARIABLES NOT DEFINED INTERNALLY, PLEASE SEE THE
C* ACCOMPANYING PROGRAM SUMMARY.
      REAL*8 TWO/2.0D0/,FOUR/4.0D0/
      REAL*8 AMU/931.5016D0/,ME/0.5110034D0/
      REAL*8 MP/938.2796D0/,MPI/134.9626D0/
      COMMON/INKIN/NNAME,NPNAME,NZ,NPZ,NA,NPA,MASXN,MASXNP,
     &           TP,NPRN,NDEV
      COMMON/KINOUT/MPIO,CMEPIO,CMPPIO,CGAMMA,CBETA
      MPIO=MPI
C*
C* CALCULATE THE MASS OF THE TARGET ('MN') FROM THE MASS
C* NUMBER ('NA'), THE ATOMIC NUMBER ('NZ') AND THE
C* MASS EXCESS ('MASXN') PROVIDED
      MN=NA*AMU-NZ*ME+MASXN
C*
C* CALCULATE THE MASS OF THE RESIDUAL NUCLEUS ('MNP')
C* FROM THE MASS NUMBER ('NPA'), THE ATOMIC NUMBER
C* ('NZP') AND THE MASS EXCESS ('MASXNP') PROVIDED
      MNP=NPA*AMU-NPZ*ME+MASXNP
C*
C* THRESHOLD FOR PIO PRODUCTION ('TO')
      TO=((MPIO+MNP)*(MPIO+MNP)-(MP+MN)*(MP+MN))/TWO/MN
C*
C* TOTAL CM ENERGY ('S')
      S2=(MP+MN)*(MP+MN)+TWO*TP*MN
      S=DSQRT(S2)
C*
C* PIO CM MOMENTUM ('CMPPIO')
      PPIO2=(S+MPIO+MNP)*(S+MPIO-MNP)*(S-MPIO+MNP)*
```

```
     &      (S-MPIO-MNP)/FOUR/S2
       CMPPIO=DSQRT(PPIO2)
C*
C* PIO CM ENERGY ('CMEPIO')
       EPIO2=PPIO2+MPIO*MPIO
       CMEPIO=DSQRT(EPIO2)
C*
C* VELOCITY OF THE CM IN THE LAB ('CBETA')
       E=MP+MN+TP
       CBETA=DSQRT(TP*(TP+TWO*MP))/E
C*
C* LORENTZ FACTOR ('CGAMMA')
       CGAMMA=E/S
C*
C* WRITE OUT THE QUANTITIES CALCULATED, IF REQUESTED
       IF(NPRN.NE.1)RETURN
C*
       WRITE(NDEV,1000)NA,NNAME,NPA,NPNAME,MN,MP,MPIO,MNP,TP,
     &NNAME,MASXN,NPNAME,MASXNP,TO,S,CMPPIO,CMEPIO,CBETA,
     &CGAMMA
 1000 FORMAT( /11X,'RELATIVISTIC KINEMATICS'/ /
     &4X,I3,A2,3X,'(    P    ,   PIO   )',3X,I3,A2/
     &1X,F11.4,'(',F9.4,',',F9.4,')',F11.4/ /
     &' PROTON KINETIC ENERGY: ',F10.4,4X,'MEV'/
     &'         MASS EXCESS     '/
     &               20X,A2,': ',F10.4,4X,'MEV'/
     &               20X,A2,': ',F10.4,4X,'MEV'/ /
     &'              THRESHOLD: ',F10.4,4X,'MEV'/
     &'        TOTAL CM ENERGY: ',F10.4,4X,'MEV'/
     &'        PIO CM MOMENTUM: ',F10.4,4X,'MEV/C'/
     &'        PIO CM   ENERGY: ',F10.4,4X,'MEV'/
     &' VELOCITY OF CM IN LAB: ',4X,F9.7,' C'/
     &'         LORENTZ FACTOR: ',4X,F9.7/ )
C*
       RETURN
       END
```

3. Test Run with Sample Output

The example which follows illustrates the calculation of the kinematic parameters for a proton with kinetic energy 147 MeV incident upon a carbon target. Since NPRN=1, the quantities calculated are written to the device specified by NDEV (NDEV=6 -> write to line printer). Note that if the parameters were required by the test routine, the common KINOUT need only be inserted.

Test Program:

```
      IMPLICIT REAL*8(A-H,M,O-Z)
C*
C* TEST PROGRAM ILLUSTRATING THE USE OF THE SUBROUTINE
C* KINMAT
      INTEGER TARGET/'C'/,RESID/'N'/
C*
C* COMMON PROVIDES DEFINITION OF INPUT PARAMETERS
      COMMON/INKIN/NNAME,NPNAME,NZ,NPZ,NA,NPA,MASXN,MASXNP,
     &             TP,NPRN,NDEV
C*
C* SET THE INPUT PARAMETERS TO KINMAT:
C*   - NAME OF THE TARGET ('NNAME')
C*   - NAME OF THE RESIDUAL NUCLEUS ('NPNAME')
C*   - ATOMIC NUMBER OF TARGET ('NZ')
C*   - ATOMIC NUMBER OF RESIDUAL NUCLEUS ('NPZ')
C*   - MASS NUMBER FOR TARGET ('NA')
C*   - MASS NUMBER OF RESIDUAL NUCLEUS ('NPA')
C*   - MASS EXCESS FOR TARGET ('MASXN') IN MEV
C*   - MASS EXCESS OF RESIDUAL NUCLEUS ('MASXNP') IN MEV
C*   - KINETIC ENERGY OF THE PROTON ('TP') IN MEV
C*   - PRINT OPTION ('NPRN' -> PRINT PARAMETERS)
C*   - OUTPUT DEVICE ('NDEV')
      NNAME=TARGET
      NPNAME=RESID
      NZ=6
      NPZ=7
      NA=12
      NPA=13
      MASXN=0.0D0
      MASXNP=5.345D0
      TP=147.0D0
```

```
      NPRN=1
      NDEV=6
C*
      CALL KINMAT
C*
      STOP
      END
```

Output:

```
            RELATIVISTIC KINEMATICS

    12C     (     P     ,    PIO    )     13N
 11174.9532( 938.2796, 134.9626) 12111.2888

 PROTON KINETIC ENERGY:    147.0000     MEV
        MASS EXCESS
                     C :      0.0        MEV
                     N :      5.3450     MEV

              THRESHOLD:    144.9789     MEV
       TOTAL CM ENERGY:  12248.0956     MEV
       PIO CM MOMENTUM:     22.2619     MEV/C
       PIO CM   ENERGY:    136.7863     MEV
 VELOCITY OF CM IN LAB:      0.0444854 C
        LORENTZ FACTOR:      1.0009909
```

Verification:

Computation of the above parameters with a hand calculator yields agreement to all (displayed) digits.

APPENDIX B - THE SUBROUTINE LTRANS

1. Program Summary

* Name: LTRANS

* Description/Purpose: Provides the Lorentz transformation of:

  - θ^* polar angle of the particle in radians
  - p^* momentum of particle in MeV/c
  - E^* total energy of particle in MeV

  in the CM frame to the quantities that are observed in the lab frame (θ,p,E), given the velocity of the CM in the lab in units of c β_c, and the Lorentz factor γ_c.

* Precision: Double

* Calling Format: CALL LTRANS

* Common Areas and Data Initialization:

  /INLTR/ quantities in the CM and the required velocity and Lorentz factor (input)

```
COMMON/INLTR/TCM,PCM,ECM,GAMMA,BETA
```

  /TRIG/ Trigonometric constants: multiples of π (input)

```
COMMON/TRIG/PI1OR2,PI,PI3OR2,TWOPI
```

  /LTROUT/ Desired quantities in the lab (output)

```
COMMON/LTROUT/TLAB,PLAB,ELAB
```

* Description of Variables: (in order of appearance)

  TCM θ^* (input/REAL*8)

PCM	p^* (input/REAL*8)
ECM	E^* (input/REAL*8)
GAMMA	γ_c (input/REAL*8)
BETA	β_c (input/REAL*8)
PI1OR2	$\pi/2$ in radians (input/REAL*8)
PI	π in radians (input/REAL*8)
PI3OR2	$3\pi/2$ in radians (not in use!)
TWOPI	2π in radians (not in use!)
TLAB	θ (output/REAL*8)
PLAB	p (output/REAL*8)
ELAB	E (output/REAL*8)

All other variables are either defined internally or their function may be determined from context.

* Programming Notes and Suggestions:

1. The polar angle of the particle in the CM system must (!) be provided in radians. The value of the polar angle in the lab frame which is returned is also in radians.

2. Program Listing

```
      SUBROUTINE LTRANS
      IMPLICIT REAL*8(A-H,O-Z)
C*
C* ROUTINE PROVIDES THE LORENTZ TRANSFORMATION OF:
C*     - POLAR ANGLE OF THE PARTICLE IN RADIANS ('TCM')
C*     - MOMENTUM OF PARTICLE IN MEV/C ('PCM')
C*     - TOTAL ENERGY OF PARTICLE IN MEV ('ECM')
C* IN THE CM TO THE QUATITIES THAT ARE OBSERVED IN THE
C* LAB FRAME ('TLAB','PLAB','ELAB'), GIVEN THE VELOCITY
C* OF THE CM IN THE LAB IN UNITS OF C ('BETA'), AND THE
C* LORENTZ FACTOR ('GAMMA').
      REAL*8 ZERO/0.0D0/
      COMMON/INLTR/TCM,PCM,ECM,GAMMA,BETA
      COMMON/TRIG/PI1OR2,PI,PI3OR2,TWOPI
      COMMON/LTROUT/TLAB,PLAB,ELAB
C*
      ZTCM=DCOS(TCM)
      STCM=DSIN(TCM)
C*
C* IS THE POLAR ANGLE IN THE CM EQUAL TO ZERO OR PI?
C* IF IT IS, IT MUST BE HANDLED SEPARATELY
      IF(STCM.NE.ZERO)GO TO 10
C*
      TLAB=TCM
      ZTLAB=ZTCM
      PLAB=GAMMA*(PCM*ZTCM+BETA*ECM)/ZTLAB
      ELAB=GAMMA*(BETA*PCM*ZTCM+ECM)
      RETURN
C*
   10 CONTINUE
C*
C* THIS IS THE DENOMINATOR IN THE EXPRESSION FOR THE
C* THE POLAR ANGLE IN THE LAB FRAME
      DENOM=GAMMA*(PCM*ZTCM+BETA*ECM)
C*
C* IF THE DENOMINATOR IS ZERO, THEN THE TAN(ANGLE) = + OR
C* - INFINITY, AND THE ANGLE IS PI/2.  THE TWO STATEMENTS
C* WHICH FOLLOW PROVIDE A DIVIDE CHECK AS WELL AS ASSIGNING
C* THE APPROPRIATE VALUE FOR THE ANGLE.
      TLAB=PI1OR2
      IF(DENOM.EQ.ZERO)GO TO 20
C*
      TLAB=DATAN(PCM*STCM/DENOM)
```

```
C*
C* INVERSE TANGENT FUNCTION RETURNS ANGLES IN THE CLOSED
C* INTERVAL FROM -PI/2 UP TO PI/2.  THE ACTUAL RANGE OF THE
C* POLAR ANGLE IS THE CLOSED INTERVAL FROM 0 TO PI.
C* WHEN THE ANGLE IS NEGATIVE, IT IS COMPLEMENTED.
C* (I.E. TAN(ANGLE+PI) = TAN(ANGLE)).
      IF(TLAB.LT.ZERO)TLAB=TLAB+PI
C*
   20 PLAB=PCM*STCM/DSIN(TLAB)
      ELAB=GAMMA*(BETA*PCM*ZTCM+ECM)
      RETURN
      END
```

3. Test Run with Sample Output

As an example of the use of this program, consider the same reaction given at the test case for the subroutine KINMAT, namely a proton incident upon a carbon target with kinetic energy 147 MeV. Suppose that the angle, momentum and energy of the pion were desired for a number of different values of the angle of the pion in the CM frame. Just such an application follows.

To verify the operation of the subroutine, the rest mass of the pion is calculated from the momentum and energy of the pion in the lab. Since the rest mass is an invariant, it is expected to be constant for any choice of the polar angle in the CM frame.

Test Program:

```
      IMPLICIT REAL*8(A-H,M,O-Z)
C*
C* TEST PROGRAM ILLUSTRATING THE USE OF THE SUBROUTINE
C* LTRANS
      REAL*8 ZERO/0.0D0/,ONE/1.0D0/,TWO/2.0D0/
      REAL*8 DUMMY/0.0D0/,THIRTY/30.0D0/
C*
C* COMMON PROVIDES DEFINITION OF INPUT PARAMETERS TO LTRANS
      COMMON/INLTR/TCM,PCM,ECM,GAMMA,BETA
C*
C* COMMON PROVIDES THE TRIGONOMETRIC CONSTANTS REQUIRED
C* BY LTRANS.  THE UNUSED CONSTANTS ARE 'DUMMIED'
      COMMON/TRIG/PI1OR2,PI,PI3OR2,TWOPI
C*
C* COMMON CONTAINS QUANTITIES IN THE LAB CORRESPONDING TO
C* THE QUANTITIES IN THE CM PROVIDED
      COMMON/LTROUT/TLAB,PLAB,ELAB
```

```
          .
          .
* generate the kinematic parameters here using KINMAT *
          .
          .
C*
C* CALCULATE TRIGONOMETRIC CONSTANTS REQUIRED
C* TO ACCURACY OF MACHINE
      PI1OR2=TWO*DATAN(ONE)
      PI=TWO*PI1OR2
      PI3OR2=DUMMY
      TWOPI=DUMMY
C*
C* SET THE STEP SIZE FOR THE ANGLE IN THE CM
      STEP=PI/THIRTY
C*
C* START WITH THE ANGLE IN THE CM EQUAL TO ZERO
      TCM=ZERO
C*
C* SET THE REMAINING STATIC PARAMETERS TO LTRANS
      PCM=CMPPI0
      ECM=CMEPI0
      GAMMA=CGAMMA
      BETA=CBETA
C*
C* WRITE OUT A HEADER TO INDICATE WHAT HAS BEEN CALCULATED
      WRITE(6,1000)
 1000 FORMAT(4X,'TCM',7X,'TLAB',7X,'PLAB',7X,'ELAB',7X,
     &'MPI0'/
     &' (RADIANS)',1X,'(RADIANS)',3X,'(MEV/C)',5X,'(MEV)',
     &4X,'(MEV/C/C)'/ )
C*
C* DO THE LORENTZ TRANSFORMATION FOR ANGLES RANGING FROM
C* ZERO TO PI, IN STEPS INDICATED BY THE STEP SIZE ABOVE
      DO 10 I=1,30
      CALL LTRANS
C*
C* CALCULATE THE (INVARIANT) REST MASS OF THE PI0 FROM
C* THE ENERGY AND MOMENTUM OF THE PI0 IN THE LAB
C* JUST DETERMINED
      M=DSQRT(ELAB*ELAB-PLAB*PLAB)
C*
      WRITE(6,2000)TCM,TLAB,PLAB,ELAB,M
 2000 FORMAT(1X,2(F8.6,2X),3(F10.4,1X))
      TCM=TCM+STEP
   10 CONTINUE
C*
      STOP
      END
```

Output:

.
.

PIO CM MOMENTUM: 22.2619 MEV/C
PIO CM ENERGY: 136.7863 MEV
VELOCITY OF CM IN LAB: 0.0444854 C
LORENTZ FACTOR: 1.0009909

.
.

TCM (RADIANS)	TLAB (RADIANS)	PLAB (MEV/C)	ELAB (MEV)	MPIO (MEV/C/C)
0.0	0.0	28.3750	137.9132	134.9626
0.104720	0.082178	28.3486	137.9077	134.9626
0.209440	0.164468	28.2695	137.8915	134.9626
0.314159	0.246986	28.1382	137.8647	134.9626
0.418879	0.329847	27.9555	137.8275	134.9626
0.523599	0.413172	27.7223	137.7804	134.9626
0.628319	0.497086	27.4400	137.7239	134.9626
0.733038	0.581720	27.1104	137.6586	134.9626
0.837758	0.667214	26.7353	137.5852	134.9626
0.942478	0.753716	26.3172	137.5045	134.9626
1.047198	0.841385	25.8587	137.4175	134.9626
1.151917	0.930393	25.3628	137.3251	134.9626
1.256637	1.020924	24.8329	137.2282	134.9626
1.361357	1.113177	24.2730	137.1280	134.9626
1.466077	1.207367	23.6871	137.0255	134.9626
1.570796	1.303725	23.0801	136.9219	134.9626
1.675516	1.402497	22.4572	136.8182	134.9626
1.780236	1.503943	21.8242	136.7158	134.9626
1.884956	1.608332	21.1873	136.6155	134.9626
1.989675	1.715937	20.5534	136.5187	134.9626
2.094395	1.827025	19.9300	136.4262	134.9626
2.199115	1.941840	19.3254	136.3392	134.9626
2.303835	2.060588	18.7480	136.2585	134.9626
2.408554	2.183408	18.2071	136.1852	134.9626
2.513274	2.310342	17.7120	136.1199	134.9626
2.617994	2.441306	17.2724	136.0634	134.9626
2.722714	2.576056	16.8973	136.0163	134.9626
2.827433	2.714162	16.5953	135.9791	134.9626
2.932153	2.855008	16.3738	135.9522	134.9626
3.036873	2.997796	16.2385	135.9360	134.9626

APPENDIX C - THE SUBROUTINE ROTATE

1. Program Summary

* Name: ROTATE

* Description/Purpose: Program transforms (by rotation) the angles of the gammas in the lab frame of the pion ($\theta'_{\gamma 1}$, $\phi'_{\gamma 1}$, $\theta'_{\gamma 2}$, $\phi'_{\gamma 2}$) to those observed in the lab frame of the reaction ($\theta_{\gamma 1}$, $\phi_{\gamma 1}$, $\theta_{\gamma 2}$, $\phi_{\gamma 2}$) given the angles of the pion in the lab frame of the reaction (θ_{π}, ϕ_{π}). All angles must (!) be provided in radians.

* Precision: Double

* Calling Format: CALL INVTAN

* Common Areas and Data Initialization:

/INROT/ Angles of the gammas in the lab frame of the pion and the angles of the pion in the lab frame of the reaction (input)

```
COMMON/INROT/THPI,PHIPI,THGMP1,PHGMP1,THGMP2
```

/ININV/ Rectangular coordinates x and y as well as FORTRAN unit number of output device if error occurs (input)

/INVOUT/ Azimuthal angle (output)

```
COMMON/INVOUT/PHI

COMMON/ININV/X,Y,NERDEV
```

/ROTOUT/ Angles of the gammas in the lab frame of the reaction (output)

```
COMMON/ROTOUT/THGAM1,PHGAM1,THGAM2,PHGAM2
```

* Description of Variables: (in order of appearance)

THPI	θ_{π} (input/REAL*8)
PHIPI	ϕ_{π} (input/REAL*8)
THGMP1	θ'_{γ_1} (input/REAL*8)
PHGMP1	ϕ'_{γ_1} (input/REAL*8)
THGMP2	θ'_{γ_2} (input/REAL*8)
X	Rectangular Coorinate x (input/REAL*8)
Y	Rectangular Coordinate y (input/REAL*8)
NERDEV	FORTRAN unit number of output device in the event that an error occurs (input/REAL*8)
PHI	Azimuthal angle ϕ in radians (output/REAL*8)
THGAM1	θ_{γ_1} (output/REAL*8)
PHGAM1	ϕ_{γ_1} (output/REAL*8)

THGAM2 θ_{γ_2}
(output/REAL*8)

PHGAM2 ϕ_{γ_2}
(ouput/REAL*8)

All other variables are either defined internally or their function may be determined from context.

* Programming Notes and Suggestions:

1. Since the routine INVTAN is called to provide the azimuthal angle from the components of the respective gamma in the lab frame of the reaction, the trigonometric constants (multiples of π) must be generated in the main routine. The constants generated must be supplied to INVTAN through the common TRIG.

2. Program Listing

```
      SUBROUTINE ROTATE
      IMPLICIT REAL*8 (A-H,K,O-Z)
C*
C* PROGRAM TRANSFORMS (BY ROTATION) THE ANGLES OF THE GAMMAS
C* IN THE LAB FRAME OF THE PION ('THGMP1','PHGMP1';'THGMP2',
C* 'THGMP1'+PI) TO THE LAB FRAME OF THE REACTION ('THGAM1',
C* 'PHGAM1';'THGAM2','PHGAM2') GIVEN THE DIRECTION OF THE
C* PION IN THE LAB FRAME ('THPI','PHIPI').  ALL ANGLES MUST
C* BE PROVIDED IN RADIANS.
      REAL*8 ZERO/0.0D0/
      COMMON/INROT/THPI,PHIPI,THGMP1,PHGMP1,THGMP2
      COMMON/ININV/X,Y,NERDEV
      COMMON/INVOUT/PHI
      COMMON/ROTOUT/THGAM1,PHGAM1,THGAM2,PHGAM2
C*
C* SET THE OUTPUT DEVICE IN CASE OF ERROR
      NERDEV=6
C*
C* SOME TRIG CONSTANTS...
      ZTHPI=DCOS(THPI)
      STHPI=DSIN(THPI)
      ZPHIPI=DCOS(PHIPI)
      SPHIPI=DSIN(PHIPI)
      ZTHGP1=DCOS(THGMP1)
      STHGP1=DSIN(THGMP1)
      ZPHGP1=DCOS(PHGMP1)
      SPHGP1=DSIN(PHGMP1)
      ZTHGP2=DCOS(THGMP2)
      STHGP2=DSIN(THGMP2)
C*
C* THESE ARE THE ELEMENTS OF THE ROTATION MATRIX A
      A11=-SPHIPI
      A12=-ZPHIPI*ZTHPI
      A13=ZPHIPI*STHPI
      A21=ZPHIPI
      A22=-SPHIPI*ZTHPI
      A23=SPHIPI*STHPI
      A31=ZERO
      A32=STHPI
      A33=ZTHPI
C*
C* COMPONENTS OF GAMMA 1 IN THE LAB FRAME OF THE PION
      KXP1=STHGP1*ZPHGP1
      KYP1=STHGP1*SPHGP1
      KZP1=ZTHGP1
```

```
C*
C* COMPONENTS OF GAMMA 2 IN THE LAB FRAME OF THE PION
      KXP2=-STHGP2*ZPHGP1
      KYP2=-STHGP2*SPHGP1
      KZP2=ZTHGP2
C*
C* COMPONENTS OF GAMMA 1 IN THE LAB FRAME OF THE REACTION
      KX1=A11*KXP1 + A12*KYP1 + A13*KZP1
      KY1=A21*KXP1 + A22*KYP1 + A23*KZP1
      KZ1=A31*KXP1 + A32*KYP1 + A33*KZP1
C*
C* COMPONENTS OF GAMMA 2 IN THE LAB FRAME OF THE REACTION
      KX2=A11*KXP2 + A12*KYP2 + A13*KZP2
      KY2=A21*KXP2 + A22*KYP2 + A23*KZP2
      KZ2=A31*KXP2 + A32*KYP2 + A33*KZP2
C*
C* FIND THE POLAR ANGLE OF THE GAMMAS ('THGAM*')
      THGAM1=DARCOS(KZ1)
      THGAM2=DARCOS(KZ2)
C*
C* FIND THE AZIMUTHAL ANGLE PHI ('PHGAM*') FROM THE
C* RECTANGULAR COORDINATES X ('KX*') AND Y ('KY*') FOR
C* BOTH GAMMAS
      X=KX1
      Y=KY1
      CALL INVTAN
      PHGAM1=PHI
      X=KX2
      Y=KY2
      CALL INVTAN
      PHGAM2=PHI
C*
      RETURN
      END
```

3. Test Run with Sample Output

The example rotates 5 pairs of gamma angles from the pion lab frame to the lab frame of the reaction. Since the opening angle between the gammas should be invariant under this transformation, it is calculated in both frames. The angles and the opening angle in both frames are output for verification of the code.

Test Program:

```
      IMPLICIT REAL*8(A-H,O-Z)
C*
C* TEST PROGRAM ILLUSTRATING THE USE OF THE SUBROUTINE
C* ROTATE
      REAL*8 ONE/1.0D0/,TWO/2.0D0/,THREE/3.0D0/
C*
C* COMMON PROVIDES INPUT TO THE SUBROUTINE ROTATE
      COMMON/INROT/THPI,PHIPI,THGMP1,PHGMP1,THGMP2
C*
C* COMMON SUPPLIES REQUIRED TRIGONOMETRIC CONSTANTS
C* TO INVTAN
      COMMON/TRIG/PI1OR2,PI,PI3OR2,TWOPI
C*
C* THE GAMMA ANGLES IN THE LAB FRAME OF THE REACTION ARE
C* RETURNED HERE
      COMMON/ROTOUT/THGAM1,PHGAM1,THGAM2,PHGAM2
C*
C* CALCULATE TRIGONOMETRIC CONSTANTS REQUIRED
C* TO ACCURACY OF MACHINE
      PI1OR2=TWO*DATAN(ONE)
      PI=TWO*PI1OR2
      PI3OR2=THREE*PI1OR2
      TWOPI=TWO*PI
C*
C* HERE WE GO...
      DO 10 I=1,5
C*
C* READ THE ANGLES WHICH SPECIFY THE DIRECTION OF THE PION
C* IN THE LAB FRAME OF THE REACTION
      READ(5,*)THPI,PHIPI
```

```
C*
C* READ THE ANGLES OF THE GAMMAS IN THE LAB FRAME OF THE
C* PION
      READ(5,*)THGMP1,PHGMP1,THGMP2
C*
C* FIND THE PHI ANGLE FOR GAMMA 2 EVEN THOUGH IT IS NOT USED
      PHGMP2=PHGMP1+PI
C*
C* CALCULATE PSI FOR THE ANGLES PROVIDED
      ZTHGP1=DCOS(THGMP1)
      STHGP1=DSIN(THGMP1)
      ZTHGP2=DCOS(THGMP2)
      STHGP2=DSIN(THGMP2)
      ZPSIP=ZTHGP1*ZTHGP2-STHGP1*STHGP2
      PSIP=DARCOS(ZPSIP)
C*
C* ROTATE THE GAMMA ANGLES IN THE LAB FRAME OF THE REACTION
      CALL ROTATE
C*
C* CALCULATE PSI FOR THE ROTATED GAMMAS
      ZTHGM1=DCOS(THGAM1)
      STHGM1=DSIN(THGAM1)
      ZTHGM2=DCOS(THGAM2)
      STHGM2=DSIN(THGAM2)
      PHI=PHGAM2-PHGAM1
      ZPHI=DCOS(PHI)
      ZPSI=ZTHGM1*ZTHGM2+STHGM1*STHGM2*ZPHI
      PSI=DARCOS(ZPSI)
C*
C* WRITE OUT THE RESULT
      WRITE(6,1000)THPI,PHIPI,
     &             THGMP1,PHGMP1,THGMP2,PHGMP2,PSIP,
     &             THGAM1,PHGAM1,THGAM2,PHGAM2,PSI
 1000 FORMAT( /'           THPI,PHIPI: ',2(F6.4,1X)/
     &         ' TP1,PP1,TP2,PP2,PSI: ',5(F6.4,1X)/
     &         '     T1,P1,T2,P2,PSI: ',5(F6.4,1X))
   10 CONTINUE
C*
      STOP
      END
```

Data:

```
1.394836 6.136521
1.040187 1.943105 1.792477
1.417651 5.349155
1.313537 0.420436 1.499199
1.479773 2.167909
0.253398 1.388969 2.792193
0.616991 1.788072
0.390319 0.876095 2.568711
0.215491 5.868785
1.337017 1.073134 1.391025
```

Output:

```
      THPI,PHIPI: 1.3948 6.1365
TP1,PP1,TP2,PP2,PSI: 1.0402 1.9431 1.7925 5.0847 2.8327
    T1,P1,T2,P2,PSI: 0.4958 5.4164 2.7739 1.5845 2.8327

      THPI,PHIPI: 1.4177 5.3492
TP1,PP1,TP2,PP2,PSI: 1.3135 0.4204 1.4992 3.5620 2.8127
    T1,P1,T2,P2,PSI: 1.1275 0.4235 1.9730 3.9232 2.8127

      THPI,PHIPI: 1.4798 2.1679
TP1,PP1,TP2,PP2,PSI: 0.2534 1.3890 2.7922 4.5306 3.0456
    T1,P1,T2,P2,PSI: 1.2307 2.2160 2.0050 5.3778 3.0456

      THPI,PHIPI: 0.6170 1.7881
TP1,PP1,TP2,PP2,PSI: 0.3903 0.8761 2.5687 4.0177 2.9590
    T1,P1,T2,P2,PSI: 0.3939 2.4755 2.7554 6.1008 2.9590

      THPI,PHIPI: 0.2155 5.8688
TP1,PP1,TP2,PP2,PSI: 1.3370 1.0731 1.3910 4.2147 2.7280
    T1,P1,T2,P2,PSI: 1.1494 2.1933 1.5810 5.3798 2.7280
```

APPENDIX D - THE SUBROUTINE INVTAN

1. Program Summary

* Name: INVTAN

* Description/Purpose: This routine is to be utilized when one requires the azimuthal angle ϕ (in radians) from the rectangular coordinates x and y. Since the azimuthal angle is undefined for the origin, if the origin is specified, an error message is written to the FORTRAN unit designated by the user and program execution terminates.

* Precision: Double

* Calling Format: CALL INVTAN

* Common Areas and Data Initialization:

/ININV/	Rectangular coordinates x and y as well as FORTRAN unit number of output device if error occurs (input)

```
COMMON/ININV/X,Y,NERDEV
```

/TRIG/	Trigonometric constants: multiples of π (input)

```
COMMON/TRIG/PI1OR2,PI,PI3OR2,TWOPI
```

/INVOUT/	Azimuthal angle (output)

```
COMMON/INVOUT/PHI
```

* Description of Variables: (in order of appearance)

X	Rectangular Coorinate x (input/REAL*8)

Y	Rectangular Coordinate y (input/REAL*8)
NERDEV	FORTRAN unit number of output device in the event that an error occurs (input/REAL*8)
PI1OR2	$\pi/2$ in radians (input/REAL*8)
PI	π in radians (input/REAL*8)
PI3OR2	$3\pi/2$ in radians (input/REAL*8)
TWOPI	2π in radians (input/REAL*8)
PHI	Azimuthal angle ϕ in radians (output/REAL*8)

All other variables are either defined internally or their function may be determined from context.

* Programming Notes and Suggestions:

1. BEWARE!!

   Since the azimuthal angle is undefined for the origin, i.e. x and y equal to zero, this routine will terminate execution of whatever program is calling it if both the x and y coordinate vanish.

2. Program Listing

```
      SUBROUTINE INVTAN
      IMPLICIT REAL*8(A-H,O-Z)
C*
C* THIS ROUTINE IS TO BE UTILIZED WHEN ONE REQUIRES THE
C* AZIMUTHAL ANGLE PHI (IN RADIANS) FROM THE RECTANGULAR
C* COORDINATES X AND Y.
      REAL*8 ZERO/0.0D0/
      COMMON/ININV/X,Y,NERDEV
      COMMON/TRIG/PI1OR2,PI,PI3OR2,TWOPI
      COMMON/INVOUT/PHI
C*
C* PHI NOT (!) DEFINED FOR THE ORIGIN...
C* IF THE ORIGIN IS PROVIDED AS INPUT TO THIS ROUTINE,
C* AN ERROR MESSAGE IS PROVIDES AND EXECUTION IS TERMINATED
      IF(X.NE.ZERO)GO TO 20
      IF(Y.NE.ZERO)GO TO 10
      WRITE(NERDEV,1000)
 1000 FORMAT(' ** ERROR IN ROUTINE INVTAN **')
      STOP
C*
C* POINT GIVEN IS ON THE Y-AXIS
   10 IF(Y.GT.ZERO)PHI=PI1OR2
      IF(Y.LT.ZERO)PHI=PI3OR2
      RETURN
   20 PHIDAG=DATAN(Y/X)
      IF(X.GT.ZERO)GO TO 30
C*
C* PHI IS IN THE 2ND OR 3RD QUADRANT
      PHI=PI+PHIDAG
      RETURN
C*
C* PHI IS IN THE 1ST QUADRANT
   30 IF(Y.GE.ZERO)PHI=PHIDAG
C*
C* PHI IS IN THE 4TH QUADRANT
      IF(Y.LT.ZERO)PHI=PHIDAG+TWOPI
      RETURN
      END
```

3. Test Run with Sample Output

The example which follows illustrates the calculation of the azimuthal angle from rectangular coordinates provided in each of the four quadrants. Also, the origin is specified to illustrate the danger to the potential user.

Test Program:

```
      IMPLICIT REAL*8(A-H,O-Z)
C*
C* TEST PROGRAM ILLUSTRATING THE USE OF THE SUBROUTINE
C* INVTAN
      REAL*8 ONE/1.0D0/,TWO/2.0D0/,THREE/3.0D0/
C*
C* COMMON PROVIDES INPUT TO THE SUBROUTINE INVTAN
      COMMON/ININV/X,Y,NERDEV
C*
C* COMMON SUPPLIES REQUIRED TRIGONOMETRIC CONSTANTS
C* TO INVTAN
      COMMON/TRIG/PI1OR2,PI,PI3OR2,TWOPI
C*
C* THE AZIMUTHAL ANGLE PHI IS RETURNED HERE
      COMMON/INVOUT/PHI
C*
C* CALCULATE TRIGONOMETRIC CONSTANTS REQUIRED
C* TO ACCURACY OF MACHINE
      PI1OR2=TWO*DATAN(ONE)
      PI=TWO*PI1OR2
      PI3OR2=THREE*PI1OR2
      TWOPI=TWO*PI
C*
C* SPECIFY THE FORTRAN UNIT 6 AS THE OUTPUT DEVICE
C* IF AN ERROR IS ENCOUNTERED IN INVTAN
      NERDEV=6
C*
C* HERE WE GO...
      DO 10 I=1,5
C*
C* READ IN THE COORDINATES X AND Y
      READ(5,*)X,Y
C*
C* CALCULATE THE AZIMUTHAL ANGLE PHI
      CALL INVTAN
C*
C* WRITE OUT THE RESULT
```

```
      WRITE(6,1000)X,Y,PHI
 1000 FORMAT(' X,Y,PHI: ',2(D14.7,1X),F8.5)
   10 CONTINUE
C*
      STOP
      END
```

Data:

```
 0.50  3.70
-0.26  0.50
-4.00 -0.85
 0.75 -2.33
 0.00  0.00
```

Output:

```
 X,Y,PHI:  0.5000000D+00  0.3700000D+01  1.43647
 X,Y,PHI: -0.2600000D+00  0.5000000D+00  2.05032
 X,Y,PHI: -0.4000000D+01 -0.8500000D+00  3.35098
 X,Y,PHI:  0.7500000D+00 -0.2330000D+01  5.02380
 ** ERROR IN ROUTINE INVTAN **
```

APPENDIX E - LISTING OF CARLOS

```
      IMPLICIT REAL*8(A-H,L,M,O-Z)
C*
C* PROGRAM CARLOS PROVIDES MONTE CARLO SIMULATION OF PIO
C* PRODUCTION NEAR THRESHOLD
      REAL*8 ZERO/0.0D0/,ONE/1.0D0/,TWO/2.0D0/,THREE/3.0D0/,
     &       SIX/6.0D0/
      REAL*8 EPS/1.0D-16/,XL/-1.0D0/,XR/1.0D0/
      REAL RANDOM(4)
      INTEGER NSIG/16/
      EXTERNAL CUBIC
      COMMON/TRIG/PI1OR2,PI,PI3OR2,TWOPI
      COMMON/INKIN/NNAME,NPNAME,NZ,NPZ,NA,NPA,MASXN,MASXNP,
     &             TP,NPRN,NDEV
      COMMON/KINOUT/MPIO,CMEPIO,CMPPIO,CGAMMA,CBETA
      COMMON/INCUB/A,B,C,D
      COMMON/INLTR/TCM,PCM,ECM,GAMMA,BETA
      COMMON/LTROUT/TLAB,PLAB,ELAB
      COMMON/INROT/THPI,PHIPI,THGMP1,PHGMP1,THGMP2
      COMMON/ROTOUT/THGAM1,PHGAM1,THGAM2,PHGAM2
C*
C* GET THE REQUIRED PARAMETERS FOR THE DATA GENERATION
      READ(5,*)DSEED,P1,P2,P3,NPNTS,NCDEV,NDDEV
C*
C* WRITE OUT THE INFORMATION PROVIDED
      WRITE(NCDEV,100)DSEED,NPNTS,P1,P2,P3
  100 FORMAT( /' MONTE CARLO SIMULATION OF PIO PRODUCTION',
     &' NEAR THRESHOLD'/ /
     &          ' STARTING SEED FOR MONTE CARLO: ',G17.10/
     &          '    NUMBER OF EVENTS REQUESTED: ',I5/
     &          '      CROSS SECTION PARAMETERS: '/
     &          '                             A: ',G14.7/
     &          '                             B: ',G14.7/
     &          '                             C: ',G14.7)
C*
C* CHECK TO SEE IF THE CHOICE OF ANGULAR DISTRIBUTION
C* PARAMETERS CORRESPONDS TO A PHYSICALLY MEANINGFUL
C* DISTRIBUTION FOR THE PIONS
      IF(P1.LT.ZERO)GO TO 10
      IF(P3.LT.ZERO)GO TO 10
      CONSTR=TWO*DSQRT(P1*P3)
      IF(DABS(P2).LE.CONSTR)GO TO 20
   10 CONTINUE
      WRITE(NCDEV,200)
  200 FORMAT( /' ** ERROR: INVALID CHOICE OF ANGULAR',
     &          ' DISTRIBUTION PARAMETERS **')
      STOP
```

```
   20 CONTINUE
C*
C* DEFINE THE TRIGONOMETRIC CONSTANTS: MULTIPLES OF PI
C* (TO ACCURACY OF MACHINE)
      PI1OR2=TWO*DATAN(ONE)
      PI=TWO*PI1OR2
      PI3OR2=THREE*PI1OR2
      TWOPI=TWO*PI
C*
C* GET THE PARAMETERS WHICH DEFINE THE REACTION
      READ(5,300)NNAME,NPNAME
  300 FORMAT(2(A2,1X))
      READ(5,*)NZ,NPZ,NA,NPA,MASXN,MASXNP,TP,NPRN,NDEV
C*
C* CALCULATE THE VELOCITY ('CBETA'), THE LORENTZ FACTOR
C* ('CGAMMA'), THE PIO ENERGY ('CMEPIO') AND THE PIO
C* MOMENTUM ('CMPPIO') IN THE CENTRE OF MASS (CM) FROM
C* REACTION KINEMATICS
      CALL KINMAT
C*
C* FIND THE REMAINING KINEMATIC PARAMETERS WHICH ARE
C* REQUIRED: THE ENERGY ('EGAMMA') AND THE MOMENTUM
C* ('PGAMMA') OF THE TWO GAMMAS IN THE REST FRAME OF
C* THE PIO
      EGAMMA=MPIO/TWO
      PGAMMA=EGAMMA
C*
C* DEFINE THE STATIC COEFFICIENTS FOR THE CUBIC EQUATION IN
C* ZPI AND THE TOTAL CROSS SECTION ('MO')
      MO=TWO*(P1+P3/THREE)
      A=TWO*P3
      B=THREE*P2
      C=SIX*P1
      DPRIME=SIX*P1-THREE*P2+TWO*P3
C*
C* THE FIRST RECORD IN THE DATA FILE WILL BE THE NUMBER
C* OF DATA POINTS TO FOLLOW
      WRITE(NDDEV,*)NPNTS
C*
C* HERE WE GO...
      DO 30 I=1,NPNTS
C*
C* GENERATE THE FOUR RANDOM NUMBERS REQUIRED PER DATA POINT
      CALL GGUBS(DSEED,4,RANDOM)
C*
C* FIND THE POLAR ANGLE OF THE PIO IN THE CM ('CTPIO').
C* BEGIN BY DETERMINING THE PROBABILITY ('PCTPIO') FOR SOME
C* VALUE OF THE COSINE OF THE CM ANGLE OF THE PIO ('ZPIO')
      PCTPIO=RANDOM(1)
```

```
C*
C* DEFINE THE DYNAMIC COEFFICIENT FOR THE CUBIC EQUATION
C* ('DYNAMIC' = DEPENDS UPON 'PCTPIO' GENERATED ABOVE)
      D=DPRIME-SIX*MO*PCTPIO
C*
C* FIND 'ZPIO' = COS('CTPIO') BY FINDING THE ROOTS OF THE
C* CUBIC EQUATION IN 'ZPIO'.
C* INITIALIZE DYNAMIC PARAMETERS FOR ROOT FINDING ROUTINE
C* (SEE DOCUMENTATION FOR THE IMSL ROUTINE 'ZFALSE')
      ITMAX=25
      IER=0
      CALL ZFALSE(CUBIC,EPS,NSIG,XL,XR,ZPIO,ITMAX,IER)
C*
C* THIS IS THE POLAR ANGLE OF THE PIO IN THE CM ('CTPIO')
      CTPIO=DARCOS(ZPIO)
C*
C* RANDOMLY GENERATE THE AZIMUTHAL ANGLE OF THE PIO IN THE
C* CM ('CPPIO')
      CPPIO=RANDOM(2)*TWOPI
C*
C* TRANSFORM THE CM ANGLE OF THE PIO ('C*PIO') INTO THE
C* LAB FRAME OF THE REACTION ('L*PIO').  NOTE THAT THE
C* AZIMUTHAL ANGLE IS UNALTERED (!)
      TCM=CTPIO
      PCM=CMPPIO
      ECM=CMEPIO
      GAMMA=CGAMMA
      BETA=CBETA
      CALL LTRANS
      LTPIO=TLAB
      LPPIO=CPPIO
      PLPIO=PLAB
      ELPIO=ELAB
C*
C* BEGIN THE SECOND MONTE CARLO... GENERATING THE ANGLES
C* OF THE GAMMAS WITH RESPECT TO (W.R.T.) THE PIO IN THE CM.
C* RANDOMLY GENERATE THE POLAR ANGLES OF THE GAMMAS W.R.T
C* THE DIRECTION OF THE PIO ('CTGP*').  NOTE THAT IN THIS
C* FRAME THEY ARE 'BACK TO BACK'
      CTGP1=RANDOM(3)*TWO-ONE
      CTGP1=DARCOS(CTGP1)
      CTGP2=PI-CTGP1
C*
C* RANDOMLY GENERATE THE AZIMUTHAL ANGLES OF THE GAMMAS
C* W.R.T. THE PIO DIRECTION ('CPGP*')
C* NOTE THAT 'CPGP2' = 'CPGP1' + 'PI'
      CPGP1=RANDOM(4)*PI
```

```
C*
C* THE VELOCITY ('BPI0') AND THE LORENTZ FACTOR ('GPI0') FOR
C* THE PI0 ARE REQUIRED IN THE TRANSFORMATION WHICH FOLLOWS
      BPI0=PLPI0/ELPI0
      GPI0=ONE/DSQRT(ONE-BPI0*BPI0)
C*
C* TRANSFORM THE GAMMA ANGLES W.R.T THE PI0 DIRECTION
C* ('CTGP*','CPGP*') INTO THE LAB FRAME OF THE PI0
C* ('LTGP*',LPGP*').  NOTE AGAIN THAT THE AZIMUTHAL
C* ANGLES ARE UNALTERED (!)
      PCM=PGAMMA
      ECM=EGAMMA
      GAMMA=GPI0
      BETA=BPI0
      TCM=CTGP1
      CALL LTRANS
      LTGP1=TLAB
      LPGP1=CPGP1
C*
      TCM=CTGP2
      CALL LTRANS
      LTGP2=TLAB
C*
C* THE GAMMA ANGLES WE JUST FOUND ('LTGP*','LPGP*') WERE
C* W.R.T. THE PI0, NOT (!) THE LAB FRAME OF THE REACTION.
C* THUS, THEY ARE TRANSFORMED (BY ROTATION) INTO THE
C* THE DESIRED FRAME
      THPI=LTPI0
      PHIPI=LPPI0
      THGMP1=LTGP1
      PHGMP1=LPGP1
      THGMP2=LTGP2
      CALL ROTATE
      THETA1=THGAM1
      PHI1=PHGAM1
      THETA2=THGAM2
      PHI2=PHGAM2
C*
C* STORE THE GAMMA ANGLES GENERATED IN THE DATA FILE.
C* NOTE THAT ONE EVENT IS DEFINED BY TWO DATA CARDS...
C* THE FIRST CARD FOR ONE PAIR OF GAMMA ANGLES AND THE
C* SECOND FOR THE OTHER PAIR OF GAMMA ANGLES
      WRITE(NDDEV,*)THETA1,PHI1
      WRITE(NDDEV,*)THETA2,PHI2
   30 CONTINUE
```

```
C*
C* THE DATA GENERATION IS COMPLETE!
C* WRITE OUT THE FINAL SEED FROM THE MONTE CARLO
      WRITE(NCDEV,400)DSEED
  400 FORMAT( /'   FINAL SEED FROM MONTE CARLO: ',G17.10)
C*
      STOP
      END
C*
C*
      REAL FUNCTION CUBIC*8 (Z)
      IMPLICIT REAL*8(A-H,O-Z)
C*
C* THIS FUNCTION SUBROUTINE REPRESENTS A GENERAL CUBIC
C* EQUATION OF THE FORM:
C*     F(Z) = A*Z**3 + B*Z**2 + C*Z + D = 0
C* COMMON BLOCK 'COEFF' CONTAINS THE COEFFICIENTS OF THE
C* THIRD DEGREE POLYNOMIAL IN Z.
      COMMON/INCUB/A,B,C,D
      CUBIC=D+Z*(C+Z*(B+Z*A))
C*
      RETURN
      END
```

The vita has been removed from the scanned document

A MONTE CARLO METHOD FOR SIMULATION OF π^0 PRODUCTION NEAR THRESHOLD

by

Michael William Madden

(ABSTRACT)

The Monte Carlo method has been utilized to determine a procedure for the simulation of π^0 production near threshold. A number of computer programs have been written to perform as well as test the simulation process.

The method consists of generating the pion angles in the center of mass frame and subsequently its decay gammas in the lab frame of the reaction. The gamma distribution generated is fitted via the method of maximum likelihood, and the angular distribution parameters are compared to those utilized in the generation process. Due to an ambiguity in the normalization of the parameters, the ratios of the parameters are considered instead of their absolute magnitude. The fitted parameters are found to agree with the parameters used in the generation program within the limits expected from statistical theory.