JETSET NOTE 92 - 13
2 April 1992

ORGANISATION EUROPÉENNE POUR LA RECHERCHE NUCLÉAIRE
CERN EUROPEAN ORGANISATION FOR NUCLEAR RESEARCH

CERN JETSET GROUP

How and why the Forward Calorimeter will benefit from TDCs

D. Drijard, M. Ferro-Luzzi, N. Hamann,
R. Jones, B. Mouëllic, J.-M. Perreau and S. Ohlsson.

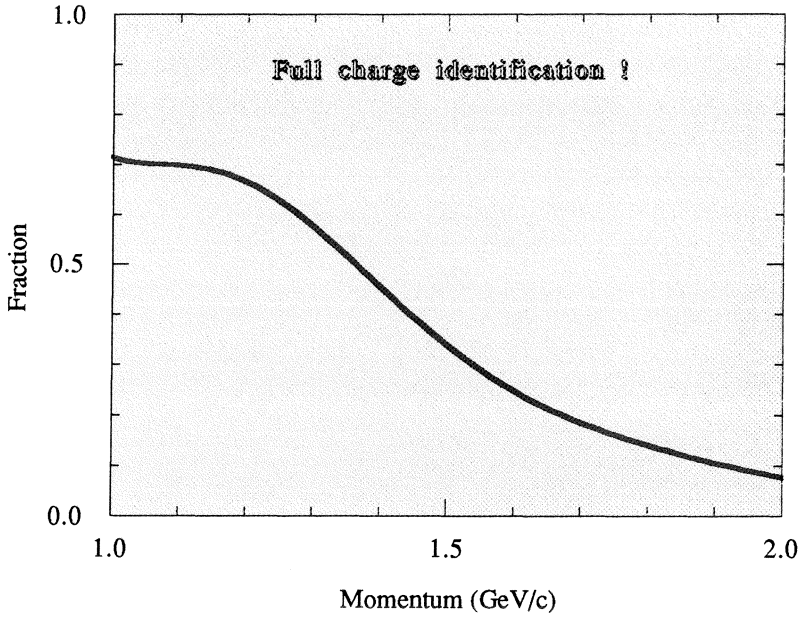

Table of contents

1. Introduction

In our previous report [1] we made the point that the introduction of a time-measuring device on the Forward Calorimeter elements would greatly enhance the usefulness of the latter (which is close to zero at present) for our study of the four-kaon reaction. A belated and unexpected rebuttal of our arguments has recently arrived from Illinois [2]. Apparently our message has not been understood.

The estimates of ref. [1] were based on order-of-magnitude calculations and were mainly aimed at proving that the small amount of already installed TDCs on the elements of the Barrel Calorimeter (BC) seemed capable of detecting with very good efficiency the decays (hence the identity) of those K^+ which entered the BC. It was suggested that the introduction of TDCs on each of the elements of the Forward Calorimeter (FC) would provide the means for identifying a good fraction of the positive kaons in four-kaon events.

Lacking time for further calculations, because of more pressing tasks and in the hope that either our common-sense justifications would be sufficiently convincing or the burden of proof could be shifted onto more responsible hands, no detailed proof of our claim had been produced. It should be added that no need for additional calculations was deemed necessary in view of the apparent enthusiastic acceptance of the idea plus the claim that additional TDCs were anyway available. We felt that the latter would provide the necessary demonstration of their usefulness directly at the data-taking stage.

It now appears that, first, our arguments are disbelieved and, second, the mythical hoard of TDCs has vanished. Therefore — at the risk of boring our devoted readership with our endless diatribes — we are obliged to come back to our earlier study and spend more of our precious time calculating in tedious detail the expected rates.

We can confirm that what we said earlier is true, viz. that TDCs will indeed help. In this new note we try again — with admirable patience and commendable altruism — to convince our disbelieving collaborators that they should not underestimate the capabilities of their own detector and rather make an effort to equip it properly.

2. Monte Carlo calculation

For this more quantitative study we have resurrected and modified an old program which had been developed in the context of our in-house Macintosh analysis chain. This program — *MIDJET*, i.e. **M**ontecarlo **I**nteractions and **D**ecays in **JET**set — although not operating in the sledgehammer manner of the well known ready-made alternative it is capable of generating events, track the particles through the detector, account for energy loss, multiple scattering and interactions and now also for decays while remaining infinitely more easy to handle, modify, operate and run than the alternative approach.

We have considered four-kaon events coming from two ϕ's where kaons which happen to enter either the FC or the BC are allowed to lose energy, stop and decay[1]. The complex geometry of the FC and BC elements has been painfully incorporated, each individual module being defined both in shape and space position. The charged decay product (only the muon has been considered at this stage but similar arguments can be applied to other decay modes) was in turn followed in its meanderings to find out if it had emerged from the block where it was created and had triggered other blocks before expiring or escaping. Finally, the time development of the production-to-decay chain was taken into account to assess the fraction of the spectrum detectable with a standard TDC unit. Range-energy relations have been introduced to calculate the stopping points using an average calorimeter density $\rho = 4.58$ g/cm^3.

The following figures (1 to 8) illustrate some of the distributions relevant to the positions inside the FC when the kaons or the muons stop. The coordinate system has the z-axis along the antiproton direction, y in the vertical direction and x as required for our preferite left-handed system. All these plots have the expected appearance and are shown only to reassure the reader about the performance of our simulation program. Notice that for these plots all tracks have been used because the distributions in question are not supposed to be sensitive to the specific type of event.

---

[1] The case of four independent kaons can also be treated but after noticing very little change with respect to the two ϕ's we have not pursued it.

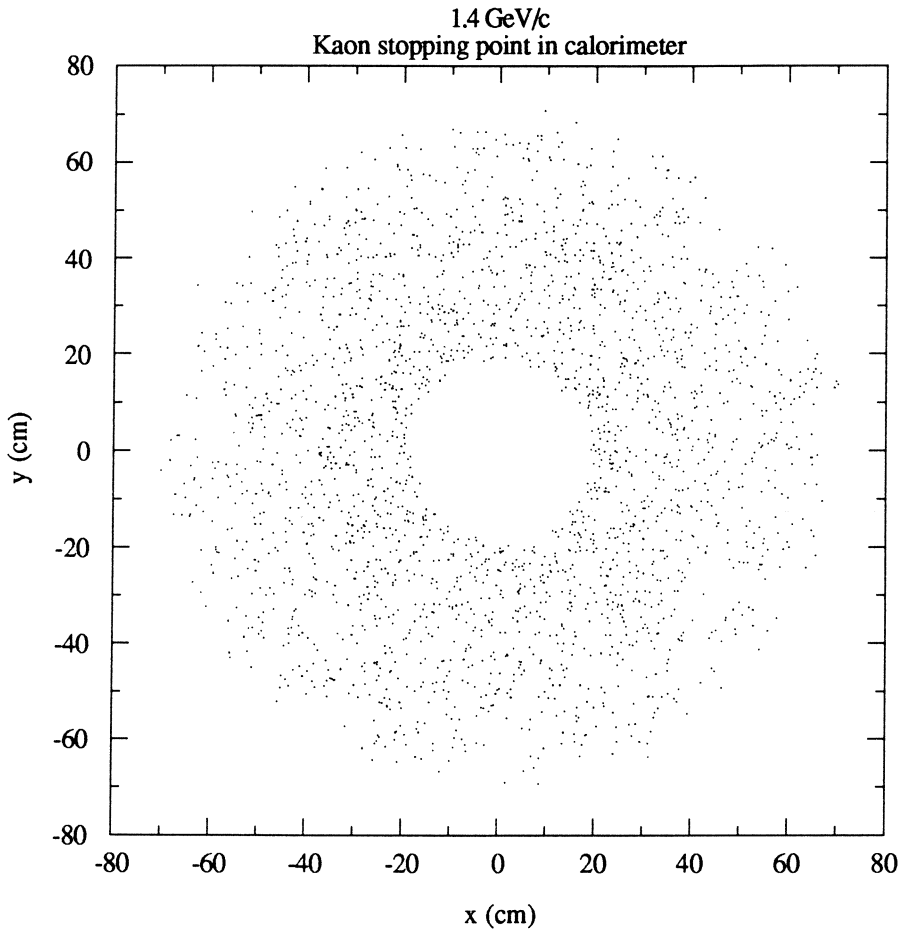

Fig. 1 Stopping point of the kaon: y vs x distribution.

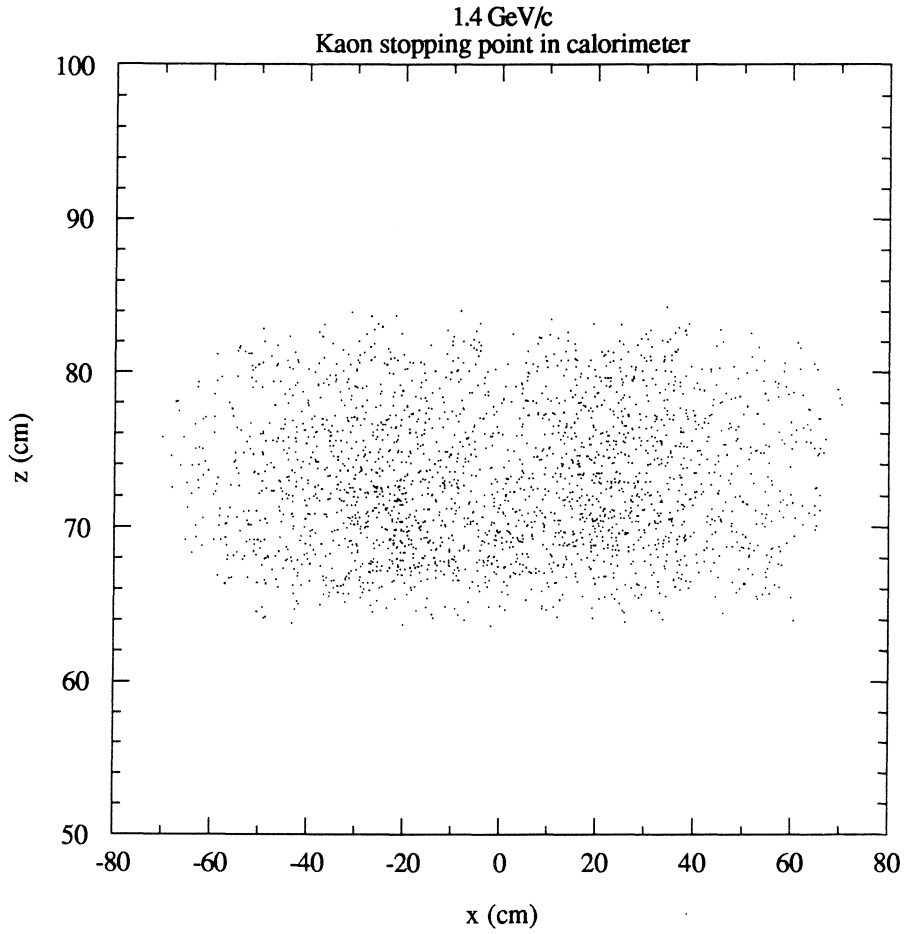

Fig. 2 Stopping point of the kaon: z vs x distribution.

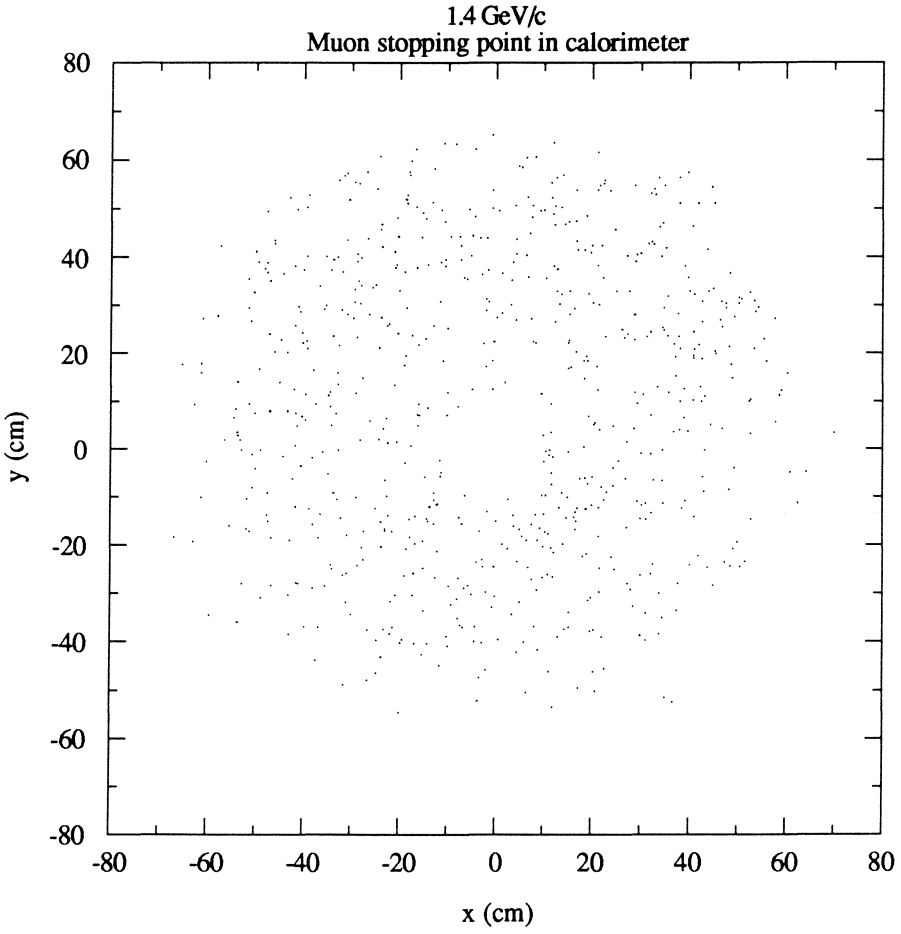

Fig. 3 Stopping point of the muon: y vs x distribution.

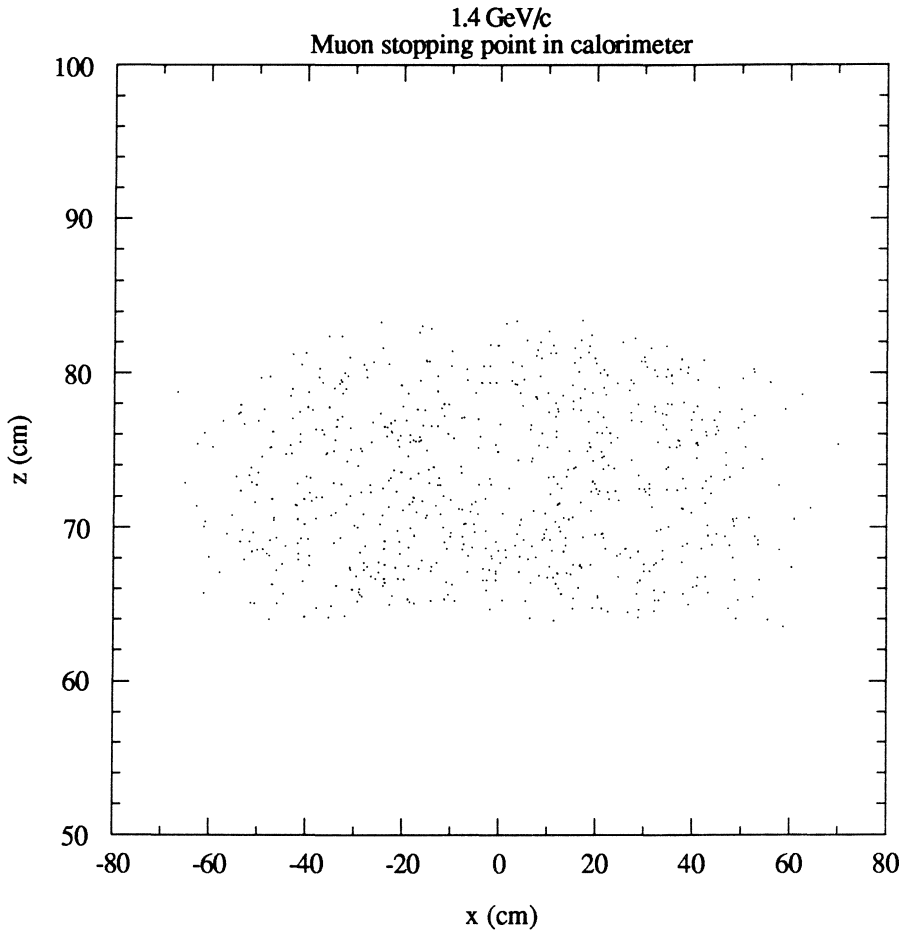

Fig. 4 Stopping point of the muon: z vs x distribution.

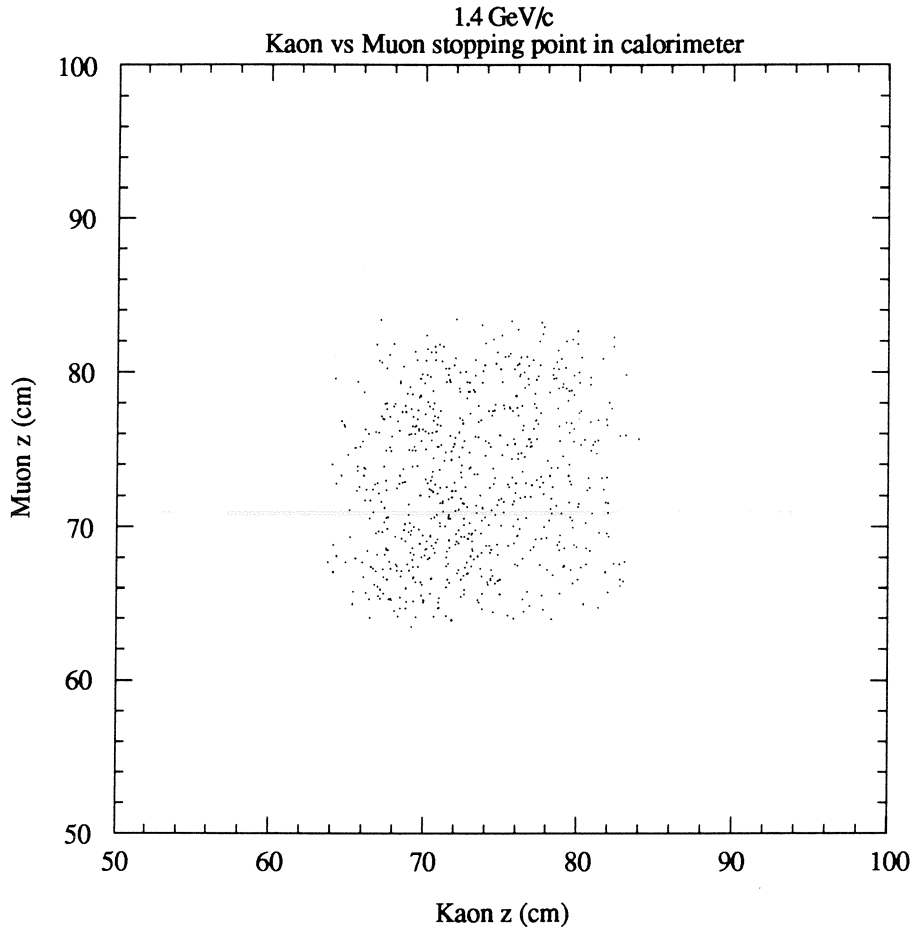

Fig. 5 Stopping point of the muon versus stopping point of the kaon

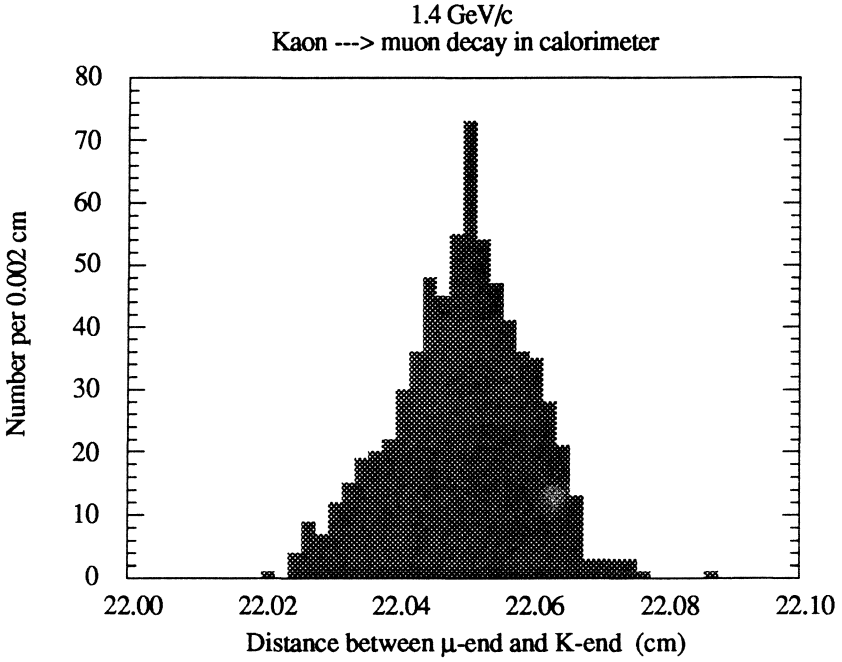

Fig. 6 Range of the muon stopping inside the FC. This spectrum should really be a line; the method used for track-following is responsible for the half-mm widening seen above.

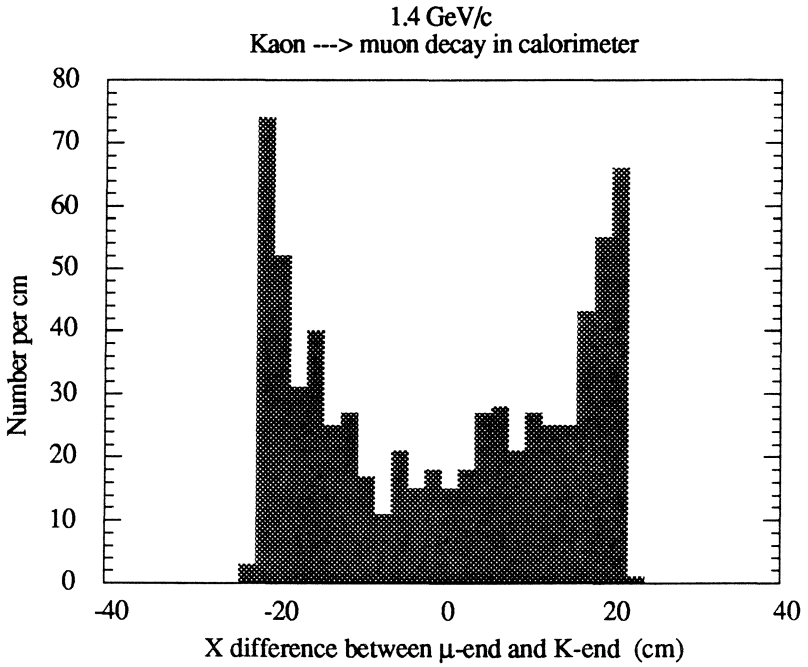

Fig. 7 Difference in x between the kaon and muon stopping points

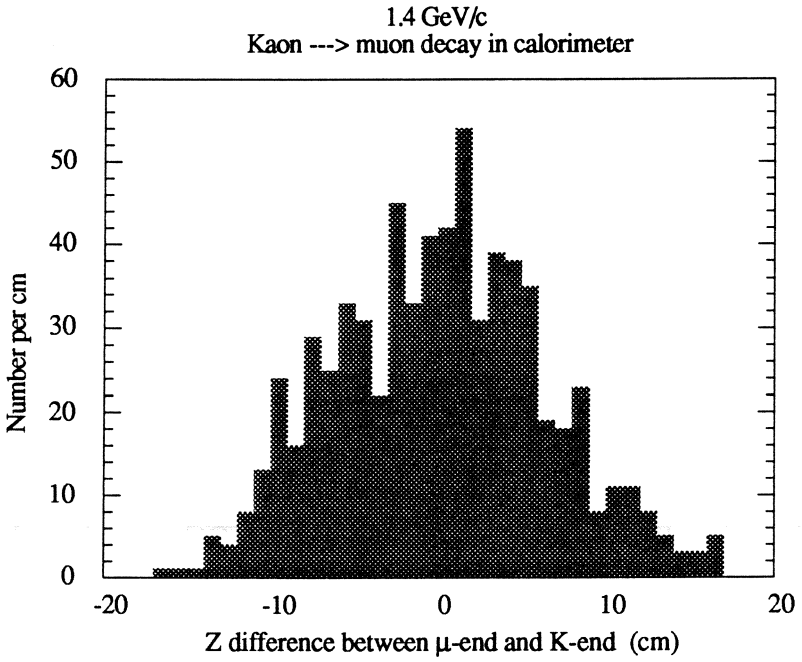

Fig. 8 Difference in z between the kaon and muon stopping points

The events we are interested in are those where the two positive kaons stop in the FC (we assume that all negative kaons are lost before decay i.e. they interact before stopping or escaping). The stopping kaons are followed in turn either by a muon traversing one or more calorimeter blocks in addition to the block where it was generated (*outer-block* detection) or by a muon escaping the calorimeter directly from the primary block (*inner-block* detection).

We have apportioned all the events as follows:

class (i) : events where only one or none of the kaons decays;

class (ii) : events where both kaons are followed by *outer-block* muons;

class (iii) : events where both kaons are followed by an *inner-block* muon;

class (iv) : events where one kaon is followed by an *inner-block* muon and the other by an *outer-block* muon .

Events in class (i) are of no interest to us (although it may be possible to find something about them by using the ADC read-out of the block, but this is another story[2]). Class (ii) events are those we have been talking about in our earlier note; these can be identified by single-hit TDCs by looking at time differences between adjacent moduli. Class (iii) events require TDCs of the multi-hit variety in order to signal the occurrence of the decay following the entry and stop of the kaon. The last class (iv) is a mixture of the above two also requiring the use of multi-hit TDCs.

3. Results and conclusions

Table 1 lists the results of our analysis in steps of 100 MeV/c over the 1 - 2 GeV/c range of incident antiproton laboratory momenta. The "*multi-hit*" column refers to the fraction of events which need the use of a multi-hit TDC as explained above. Similarly, "*single-hit*" refers to the use of a single-hit TDC. The second column represents the (undetectable) fraction of K-decays which take place in the first 2 ns after the kaon's stop. The "*total*" column is the sum of "*single-*" and "*multihit*" fractions reduced by the 2-ns fraction of events decaying in the first 2 ns and not detectable by the TDCs[3]. Fig. 9 shows the same numbers in graphic form.

If one feels like remarking that in the above considerations we have not taken into account the decays which occur before the kaon reaches the FC one should remember that none of these can be present in our sample, having already been rejected by either the track-fitting operation or the Threshold Cherenkovs constraints.

---

2 Wait for our forthcoming analysis of the ADC information on the calorimeter hits !

3 This 2 ns figure is admittedly optimistic; we have used it mainly to facilitate the comparison with the estimates of ref. [2].

Table 1

Fraction of two-kaon decays over the momentum range from 1 to 2 GeV/c.

		multi-hit	single-hit	mult + sin	total >2ns	multi-hit	single-hit	mult + sin	total >2ns
Mom.	frac > 2 ns	<---------- 4F triggers ---------->				<---------- 3F triggers ---------->			
1.000	0.757	0.335	0.608	0.943	0.714	0.356	0.475	0.831	0.629
1.100	0.762	0.309	0.607	0.915	0.697	0.350	0.486	0.836	0.637
1.200	0.789	0.276	0.565	0.841	0.663	0.307	0.428	0.735	0.580
1.300	0.808	0.243	0.472	0.715	0.578	0.251	0.347	0.598	0.483
1.400	0.826	0.210	0.338	0.548	0.453	0.199	0.279	0.477	0.394
1.500	0.852	0.142	0.258	0.400	0.341	0.154	0.212	0.366	0.312
1.600	0.870	0.099	0.185	0.284	0.247	0.108	0.155	0.263	0.229
1.700	0.891	0.074	0.133	0.207	0.184	0.077	0.105	0.182	0.162
1.800	0.904	0.051	0.103	0.155	0.140	0.052	0.066	0.118	0.107
1.900	0.920	0.035	0.078	0.114	0.105	0.027	0.032	0.058	0.053
2.000	0.935	0.027	0.054	0.081	0.076	0.010	0.011	0.021	0.020

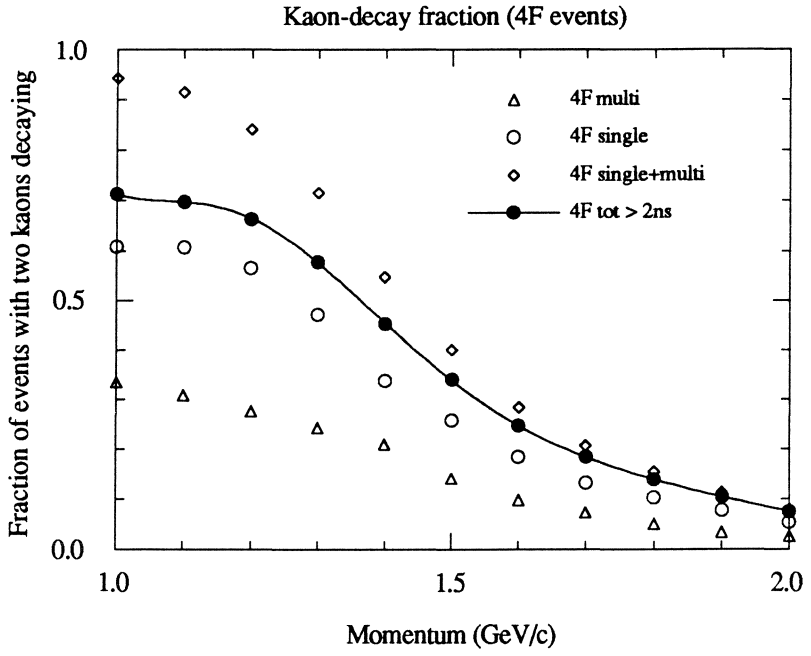

Fig. 9 Fraction of $\phi\phi$ events with two identifiable positive kaons among the **4F** triggers. Triangles refer to events needing a multi-hit TDC, empty circles to events which are identified with a single-hit TDC, lozanges to the sum of the above and full circles to the sum reduced by the fraction of short-lived decays. The curve is a polynomial fit to the full circles.

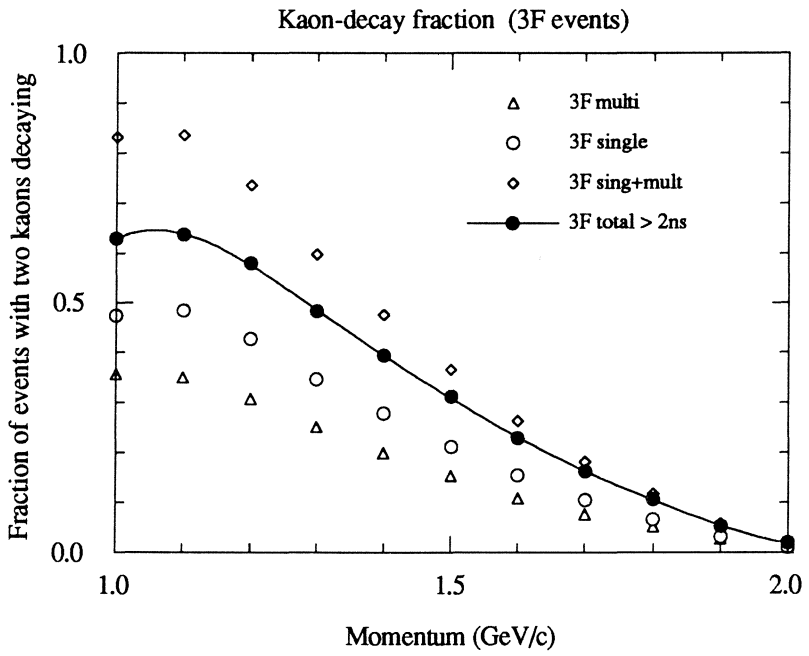

Fig. 10 Fraction of $\phi\phi$ events with two identifiable positive kaons among the **3F** triggers. The decaying kaons may occur either in the BC or the FC. The symbols are as in fig. 9.

We see that the "*single-hit*" data anticipated in our note of ref. [1] range from a hefty $\approx 61\%$ at the lowest momentum to a meager $\approx 5\%$ at the highest momentum. If one adds up the "*multi-hit*" fraction then the above limits climb to $\approx 94\%$ and to $\approx 8\%$ respectively. Clearly the facts of real-life will reduce these numbers. Still it doesn't seem to us that these fractions deserve the contempt expressed in ref. [2].

One last thing. Of course nobody is going to mix up mass plots of fully identified four-kaon events with the others. What one will do instead — if endowed with a reasonable sample of fully identified events — is looking at the shape of the clean sample so as to be able to make intelligent guesses on the interpretation of the other.

The multiple-hit TDCs that could do the job exist. They are the LECROY *model 1877*: 96 channels per module, 4,096 bins per channel, ranging from 0.5 ns to ≈ 2 μs (also capable of identifying up to 40% of the μ --> e decays thereby providing a reassuring confirmatory evidence of the kaon decay). They do not come cheap, costing as

they do about 10 KSFR per module. The overall price would be in the 30 to 40 KSFR range. It seems to us that this is not such an outrageous sum to pay in view of the advantages reaped by a better $\phi\phi$ study.

4. References

[1] Unexploited features of the calorimeter. D. Drijard, M. Ferro-Luzzi,, N. Hamann, R. Jones, B. Mouëllic, J.-M. Perreau and S. Ohlsson. CERN, JETSET Note 92-02 (14 January 1992)

[2] Unexploitable features of the calorimeter. P. Harris. Illinois. JETSET note 92- 12 (March 1992)

5. Appendix

Here we list — as we customarily do in our reports — the main routines employed for this study. They are embedded in the *DISPLAY* program where the more general geometry of the detectors can also be found.

```
      options mix=off ! on
      Public UserStart,User,UserSpit
      Include '*F:Source:CDE1.For'
      COMMON /GENIN / NP,TECM,AMASS(18),KGENEV
      COMMON /GENOUT/ PCMG(5,18),WT
       common / mychecks / locev,nevtried,n4for,n3for1bar
        common /decayhistos/nbseen(10),nanosec(50),nevkseen(5)
     1    ,nevkplusseen(5),nevkplusseenbar(5)
     1    ,nevkplusunseen(5),nevkplusunseenbar(5)
      1     ,nevkplus11,nevkplus11bar,nevtrack3F(3,3),nevtrack4F(3,3)
      common /mytv/idrawdecay
       Common/Menus/NbMenus,MenuID(3),MenuHandle(3),MenuFlag
        common/histos/iWantHisto,his_Kx(100),his_Ky(100),his_Kz(100)
     *            ,his_mux(100),his_muy(100),his_muz(100)
     *           ,his_Krho(100),his_murho(100)
      Integer*2 MenuID
        Integer*4 his_Kx,his_Ky,his_Kz,his_mux,his_muy,his_muz
     *     ,his_Krho,his_murho

          Subroutine UserStart
        common /mycalo/roatentry(9),jcaloring(2),jcalophi(2),calparc(2)
       do i=1,8
         roatentry(i)=rocal(3,i)+(65.91-zcal(3,i))*
     1       (rocal(2,i)-rocal(3,i))/ (zcal(2,i)-zcal(3,i))
      enddo
         roatentry(9)=rocal(1,8)+(65.91-zcal(1,8))*
     1       (rocal(4,8)-rocal(1,8))/ (zcal(4,8)-zcal(1,8))
        type '(1x,9f8.2)',roatentry
       call vzero(nbseen,87)
       call vzero(nevtrack3F,9)
       call vzero(nevtrack4F,9)
        call vzero(his_Kx,800) ! a la Jean-Marie .....
      print 1000
1000 format(//,' ---------- K+ decay in the calorimeter ----------',/,
     *,'      ')
      iWantHisto=0
      type *,' histograms ? (y/n) (CR=n) ---> $' ; read *,yesno
      if (yesno.eq.'y')then
         iWantHisto=1
           print '(//,1x,'Saving coordinates of stopping points for each track',//)'
          open(30,file='',access='Write')
     endif
     end
```

```
      Subroutine User
        common / caloxyz / xcalo(8,48,8),ycalo(8,48,8),zcalo(8,48,8),nptplan(3,6)
       1       ,xcalbar(8,24),ycalbar(8,24),zcalbar(8,24),rintbar,rextbar
       common/ transfert / centerofmass(4),cotetaphicm
       common/reclocal/pact(4),erec(4),phifound(4),thetafound(4)
       common /decaykplus/cosdir(3),xv(3),pk,itrack
        common /decay1/cdir(3),xstart(3),pstart,s1,s2,s1init(2),s2init(2)
       common /decay2/nri(10),ncou(10),ran(10),nseen,tdecay
       common / derivees/p1,den,deriv,deriv2,dden,coplanarity(4),cp(4),cpp(4)
       common/study/ncall0,ncall1,ncall2,den1,den2
      common /random/myrnd(2)
       common /decayxyz/xyzkstop(3),xyzmustop(3),kstopped,mustopped
       dimension pgener(4,4),SumP(3),resmass(6),ploc(4),pkincom(4),lastrnd(2)
      character text*15
      real *4 mymassk
c ------------------------  Initialisation process  ------------------
       call pgeninit
       call ucopy(centerofmass,pcm,4)
        type '(' pcm given to Recons =',4e15.8)',(pcm(i),i=1,4)
       type *,np,'  particles system'
       type *,'  masses=',(amass(i),i=1,np)
      call MyUserMenus
      nevtried=0 ; menuflag=0
      mymassk=0.49365
c      myrnd(1)=-1047737140 ; myrnd(2)=-1047999284
      NbMenus=3 ; mymaxev=1000 ; nbsolrec=0 ; ncall0=0
      call ReadVal(' give max number of events --->',mymaxev,'i10')
       call ucopy(centerofmass,pcm,4)
      n4for=0 ; n3for1bar=0 ; locev=0
         itop=40 ; ileft= 5 ; ibot=460 ; irght=635
       LunitGraphics = 20
       lu2 = LunitGraphics ; text=' PS 202 '
       if(NewColor) lu2 =  lu2 +  3000
       if(NewPic  ) lu2 =  lu2 + 10000
       if(OldPic  ) lu2 = -lu2
      idrawdecay=0
       type *,' draw decay projection ? (y/n) (CR=n) ---> $' ; read *,yesno
       if (yesno.eq.'y')  idrawdecay=1
      if (idrawdecay.eq.1) then
       ie=TvBgn(lu2,text,ITOP,ILEFT,IBOT,IRGHT)
         call TvRng(4HDISP,20.,20.,1003.,1003.)
        xmin=-1. ; xmax=100. ; ymin=xmin ; ymax=70.
        call TvRng(4huser,xmin,ymin,xmax,ymax)
       call TvMode (8hscaleon )
      endif
       ktrack1=1 ; ktrack2=3
 150    type *,' K plus tracks are 1 and 3   OK ?   (y/n) (CR=y) --->$' ; read *,yesno
        if (yesno.ne.''.and.yesno.ne.'y') then
        type *,' give new track numbers for K plus ---> $' ; read *,ktrack1,ktrack2
        goto 150
      endif
       myrnd(1)=-1569627880 ; myrnd(2)=-1764992502
     call calocoord
c ------------------------  done only once down to here ------------------
  33  lastrnd(1)=myrnd(1) ;lastrnd(2)=myrnd(2)
c      if (locev.ge.1309) type '(' evnt',i8,2i20)',locev,myrnd
      call phigener(pgener) ; nevtried=nevtried+1
     SumE=0. ; SumE2=0. ; SumP(1)=0.; SumP(2)=0.; SumP(3)=0. ; nbarrel=0
     do j=1,np
      SumE=SumE+pgener(4,j)
      do i=1,3 ; SumP(i)=SumP(i)+pgener(i,j) ; enddo
          ploc(j)=sqrt(pgener(1,j)**2+pgener(2,j)**2+pgener(3,j)**2)
         SumE2=SumE2+sqrt(pgener(1,j)**2+pgener(2,j)**2+pgener(3,j)**2 + mymassk**2)
        phiev1=atan(pgener(2,j)/pgener(1,j))
          if (phiev1.lt.0.) phiev1=phiev1+pi ; if(pgener(2,j).lt.0) phiev1=phiev1+pi
        thetaev1=acos( pgener(3,j)/ploc(j) )
        phifound(j)=phiev1 ; thetafound(j)=thetaev1
          if (thetaev1.lt.pi/4. .and. thetaev1.gt.pi/12.) goto 35
```

```
          if (thetaev1.lt.(65.*pi/180.) .and. thetaev1.gt.pi/4.) then nbarrel=nbarrel+1
           if (nbarrel.gt.1) goto 33
         else goto 33
        endif
      35 continue
       enddo
       locev=locev+1
       if (ipriuser.gt.0.and.idrawdecay.eq.1) then
        if (locev.gt.1) call tvpicstop
       read *,text
        if (locev.gt.1) call tvnext
        xx(1)=0. ; xx(2)=xmax ; yy(1)=0. ; yy(2)=0. ; call tvdraw(xx,yy,2)
       do i=1,8
         do j=1,4 ; xx(j)=zcal(j,i) ; yy(j)=rocal(j,i) ; enddo
          xx(5)=xx(1) ; yy(5)=yy(1) ; call tvdraw(xx,yy,5)
       enddo
         xx(1)=0.      ; yy(1)=rextbar ; xx(2)=rintbar ; yy(2)=rextbar ;
         xx(3)=rintbar ; yy(3)=rintbar ; xx(4)=0.      ; yy(4)=rintbar ;
          xx(5)=xx(1) ; yy(5)=yy(1) ; call tvdraw(xx,yy,5)
      endif
       if (nbarrel.eq.0) n4for=n4for+1
       if (nbarrel.eq.1) n3for1bar=n3for1bar+1
c      type '(' Events=',i7,'  tried=',i7,
c     1          '  n4for=',i6,'  n3for=',i6)',locev
c     1             ,nevtried,n4for,n3for1bar
       if (ipriuser.ge.1) type '(' Sums of P and E=',5e15.8)',SumP,SumE,SumE2
       xv(1)=0.2*rndm(myrnd) ; xv(2)=0.2*rndm(myrnd) ; xv(3)=0.2*rndm(myrnd)
      nkplusmarked=0 ; nkmarked=0 ; nkplusunmarked=0
      do j=1,np ; itrack=j
          cosdir(1)=sin(thetafound(j))*cos(phifound(j))
          cosdir(2)=sin(thetafound(j))*sin(phifound(j))
         cosdir(3)=cos(thetafound(j))
        pk=ploc(j)
        call decay
c        nseen=1  ---> mu not going into adjacent module.
c        nseen=0  ---> kaon does not stop
c        nseen>0  ---> kstopped>0
c ---      at this point the next common (self explanatory) is filled ----------
c          if(ipriUser.gt.0)print '(1x,' kaon stops [if flag=',i2,' is 1] in ',3f6.1,/
c    *      ,1x,' muon stops [if flag =',i2,' is 1] in',3f6.1,/,10x
c    *     ,' no. of blocks traversed by muon ---> nseen = '
c    *       ,i3)',kstopped,xyzkstop,mustopped,xyzmustop,nseen
          if(ipriUser.gt.0) type '(' kstopped=',i2,' mustopped=',i2,'  nseen=',i3)',kstopped,
     1     mustopped,nseen
        if(iWantHisto.eq.1.and.(kstopped+mustopped).ne.0)then
          write(30,'(1x,6f10.2)') xyzkstop,xyzmustop
c         K stopping point (zero when it doesnt.....)
       endif
            ibin=xyzkstop(1)+0.5 ; if(ibin.le.0)ibin=1; if(ibin.ge.100)ibin=100
         his_Kx(ibin)=his_Kx(ibin)+1
            ibin=xyzkstop(2)+0.5 ; if(ibin.le.0)ibin=1; if(ibin.ge.100)ibin=100
         his_Ky(ibin)=his_Ky(ibin)+1
            ibin=xyzkstop(3)+0.5 ; if(ibin.le.0)ibin=1; if(ibin.ge.100)ibin=100
         his_Kz(ibin)=his_Kz(ibin)+1
          rhoK=sqrt(xyzkstop(1)**2+xyzkstop(2)**2)
           ibin=rhoK+0.5 ; if(ibin.le.0)ibin=1; if(ibin.ge.100)ibin=100
         his_Krho(ibin)=his_Krho(ibin)+1
c         muon stopping point (zero when it doesnt.....)
           ibin=xyzmustop(1)+0.5 ; if(ibin.le.0)ibin=1; if(ibin.ge.100)ibin=100
         his_mux(ibin)=his_mux(ibin)+1
           ibin=xyzmustop(2)+0.5 ; if(ibin.le.0)ibin=1; if(ibin.ge.100)ibin=100
         his_muy(ibin)=his_muy(ibin)+1
           ibin=xyzmustop(3)+0.5 ; if(ibin.le.0)ibin=1; if(ibin.ge.100)ibin=100
         his_muz(ibin)=his_muz(ibin)+1
         rhomu=sqrt(xyzmustop(1)**2+xyzmustop(2)**2)
           ibin=rhomu+0.5 ; if(ibin.le.0)ibin=1; if(ibin.ge.100)ibin=100
         his_murho(ibin)=his_murho(ibin)+1

        if (nseen.gt.0) then
```

```
          ibin=nseen ; nbseen(ibin)=nbseen(ibin)+1
           ibin=tdecay/2+1 ; if (ibin.lt.1) ibin=1 ; if (ibin.gt.50) ibin=50
         nanosec(ibin)=nanosec(ibin)+1
       nkmarked=nkmarked+1
           if (nseen.gt.1.and.(itrack.eq.ktrack1.or.itrack.eq.ktrack2)) nkplusmarked=nkplusmarked+1
           if (nseen.eq.1.and.(itrack.eq.ktrack1.or.itrack.eq.ktrack2)) nkplusunmarked=nkplusunmarked+1
     endif
  if (j.eq.ktrack1) nseen1=nseen+1
  if (j.eq.ktrack2) nseen3=nseen+1
 enddo
  if (nseen1.gt.3) nseen1=3 ; if (nseen3.gt.3) nseen3=3
  nevkseen(nkmarked+1)=nevkseen(nkmarked+1)+1
 if (nbarrel.eq.0) then
              nevtrack4F(nseen1,nseen3)=nevtrack4F(nseen1,nseen3)+1
             nevkplusseen(nkplusmarked+1)=nevkplusseen(nkplusmarked+1)+1
             nevkplusunseen(nkplusunmarked+1)= nevkplusunseen(nkplusunmarked+1)+1
       if (nkplusmarked.eq.1.and.nkplusunmarked.eq.1)nevkplus11=nevkplus11+1
            else
             nevtrack3F(nseen1,nseen3)=nevtrack3F(nseen1,nseen3)+1
             nevkplusseenbar(nkplusmarked+1)=nevkplusseenbar(nkplusmarked+1)+1
             nevkplusunseenbar(nkplusunmarked+1)=nevkplusunseenbar(nkplusunmarked+1)+1
       if (nkplusmarked.eq.1.and.nkplusunmarked.eq.1)nevkplus11bar=nevkplus11bar+1
 endif
  call menuaction(mflag)
  if (menuflag.eq.1) type '(i7,' Events',i7,'  tried    :',i6,
1           '  n4for',i6,' n3for',i6)',locev
1             ,nevtried,n4for,n3for1bar
  if (menuflag.eq.1) type '(i10,' Events Requested')',mymaxev
 if (menuflag.eq.2) call userspit
 if (menuflag.eq.3 .or. locev.ge.mymaxev) call OutDump
 if (menuflag.eq.4) then
 if (idrawdecay.eq.1) call tvend(LunitGraphics,4hKEEP)
 stop ; endif
 menuflag=0
 goto 33
 end
```

Subroutine calocoord

```
      common / caloxyz / xcalo(8,48,8),ycalo(8,48,8),zcalo(8,48,8),nptplan(3,6)
     1       ,xcalbar(8,24),ycalbar(8,24),zcalbar(8,24),rintbar,rextbar
      data nptplan/1,4,5,  2,6,3,  1,2,3,  3,5,4,  5,6,7,  1,8,7/
c     l'ordre de ces sommets est choisi pour chaque plan en sorte que le vecteur vperp
c     utilisé dans caloinout, qui sera le produit vectoriel de
c     (sommet2-sommet1) x (sommet3-sommet1) soit dirigé vers l'intérieur du bloc de calorimètre.
      n=3*8*48*8 ; call vzero(xcalo,n) ; n=3*8*24 ; call vzero(xcalbar,n)
      rintbar=39. ; rextbar=48.4 ; zmaxcalbar=rintbar
      do ir=1,8 ; ncounter=48
       if (ir.lt.4) ncounter=24 ; if (ir.eq.1) ncounter=12
       angle=2.*pi/ncounter
       do ic=1,ncounter  ; ang1=pi-(ic-1)*angle ; ang2=pi-ic*angle
        c1=cos(ang1) ; s1=sin(ang1) ; c2=cos(ang2) ; s2=sin(ang2)
        xcalo(1,ic,ir)=rocal(1,ir)*c1 ; ycalo(1,ic,ir)=rocal(1,ir)*s1 ; zcalo(1,ic,ir)=zcal(1,ir)
        xcalo(2,ic,ir)=rocal(4,ir)*c1 ; ycalo(2,ic,ir)=rocal(4,ir)*s1 ; zcalo(2,ic,ir)=zcal(4,ir)
        xcalo(3,ic,ir)=rocal(4,ir)*c2 ; ycalo(3,ic,ir)=rocal(4,ir)*s2 ; zcalo(3,ic,ir)=zcal(4,ir)
        xcalo(4,ic,ir)=rocal(1,ir)*c2 ; ycalo(4,ic,ir)=rocal(1,ir)*s2 ; zcalo(4,ic,ir)=zcal(1,ir)
        xcalo(5,ic,ir)=rocal(2,ir)*c2 ; ycalo(5,ic,ir)=rocal(2,ir)*s2 ; zcalo(5,ic,ir)=zcal(2,ir)
        xcalo(6,ic,ir)=rocal(3,ir)*c2 ; ycalo(6,ic,ir)=rocal(3,ir)*s2 ; zcalo(6,ic,ir)=zcal(3,ir)
        xcalo(7,ic,ir)=rocal(3,ir)*c1 ; ycalo(7,ic,ir)=rocal(3,ir)*s1 ; zcalo(7,ic,ir)=zcal(3,ir)
        xcalo(8,ic,ir)=rocal(2,ir)*c1 ; ycalo(8,ic,ir)=rocal(2,ir)*s1 ; zcalo(8,ic,ir)=zcal(2,ir)
       if (ir.eq.3) then  ! use this ring for which ncounter=24 to fill barrel blocks
c                  which are also 24
        xcalbar(1,ic)=rextbar*c1 ; ycalbar(1,ic)=rextbar*s1 ; zcalbar(1,ic)=0.
        xcalbar(2,ic)=rextbar*c1 ; ycalbar(2,ic)=rextbar*s1 ; zcalbar(2,ic)=zmaxcalbar
        xcalbar(3,ic)=rextbar*c2 ; ycalbar(3,ic)=rextbar*s2 ; zcalbar(3,ic)=zmaxcalbar
        xcalbar(4,ic)=rextbar*c2 ; ycalbar(4,ic)=rextbar*s2 ; zcalbar(4,ic)=0.
        xcalbar(5,ic)=rintbar*c2 ; ycalbar(5,ic)=rintbar*s2 ; zcalbar(5,ic)=0.
        xcalbar(6,ic)=rintbar*c2 ; ycalbar(6,ic)=rintbar*s2 ; zcalbar(6,ic)=zmaxcalbar
        xcalbar(7,ic)=rintbar*c1 ; ycalbar(7,ic)=rintbar*s1 ; zcalbar(7,ic)=zmaxcalbar
        xcalbar(8,ic)=rintbar*c1 ; ycalbar(8,ic)=rintbar*s1 ; zcalbar(8,ic)=0.
       endif
      enddo ! ncounters
     enddo ! nrings
      do ip=1,6
c       type '('  plan',i2,'  sommets',3i2)',ip,(nptplan(i,ip),i=1,3)
     enddo
     end
```

Subroutine Decay

```
      common /mycalo/roatentry(9),jcaloring(2),jcalophi(2),calparc(2)
      common /decayxyz/xyzkstop(3),xyzmustop(3),kstopped,mustopped
      common /decaykplus/cosdir(3),xv(3),pk,itrack
      common /decay1/cdir(3),xstart(3),pstart,s1,s2,s1init(2),s2init(2)
      common /decay2/nri(10),ncou(10),ran(10),nseen,tdecay
      common /random/myrnd(2)
      dimension xout(3)
c     type '(' pk=',f10.3,'  xv=',3f8.3,'  cosdir=',3f8.3)',pk,xv,cosdir
      nseen=0 ; kstopped=0 ; mustopped=0 ; incalbar=0
      call vzero(xyzkstop,6)
      rovertex=sqrt(xv(1)**2+xv(2)**2)
      call calofind2
      parctot=calparc(1)+calparc(2)
c     type '(' r,n,p  ',i2,i3,f7.2
c    1       ,' r,n,p (2)  ',i2,i3,f7.2,' ptot',f7.2)',(jcaloring(i)
c    1       ,jcalophi(i),calparc(i),i=1,2),parctot
      nring=jcaloring(1)
      ekplus=sqrt(pk**2+theKchmass**2) ; rkstop=RangeEnergy(Ekplus)
      if (calparc(1).le.0.) then ! try the barrel calorimeter ------
       call calbarfind ; incalbar=1 ; parctot=calparc(1)
       if (calparc(1).le.0.) goto 80
      endif
      if (ipriuser.gt.0) type '(' K+  E and P=',2f8.3,'  Range',f8.3,'  parctot',f8.3
     1           ,'  calparc',2f8.3)',ekplus,pk,rkstop,parctot,calparc
      j=1 ; rkstop1=rkstop
        if (calparc(1).lt.rkstop) then j=2 ; rkstop=rkstop-calparc(1)
          if (rkstop.gt.calparc(2) ) then s=s1init(1)+25.
           do i=1,3 ; xstart(i)=xv(i)+s*cosdir(i)  ; enddo
           rodecay=sqrt(xstart(1)**2+xstart(2)**2)
            if (ipriuser.gt.0.and.nring.eq.10) type '(' xstart=',
     1       3f8.3,' rodecay=',f8.3)',xstart,rodecay
           xx(1)=xv(3) ; xx(2)=xstart(3) ; yy(1)=rovertex
           yy(2)=rodecay ; if (ipriuser.gt.0.and.idrawdecay.eq.1) call tvdraw(xx,yy,2)
          goto 200
        endif
      endif
      nring=jcaloring(j) ; nc=jcalophi(j) ; s=s1init(j)+rkstop
      if (nring.eq.9.or.nring.eq.0) goto 80
      call vzero(nri,30)
      kstopped=1
c------     generate decay ----------
      decayrange=22. ! dixit max for the mu+ from K+ ---> mu + neutrino
      decaylength=0.
       coteta=2.*rndm(myrnd)-1. ;  siteta=sqrt(1.-coteta**2)
      phi=2*pi*rndm(myrnd) ; cophi=cos(phi) ; siphi=sin(phi)
      cdir(1)=siteta*cophi ; cdir(2)=siteta*siphi ; cdir(3)=coteta
       do i=1,3 ; xstart(i)=xv(i)+s*cosdir(i) ; xyzkstop(i)=xstart(i) ; enddo
       rodecay=sqrt(xstart(1)**2+xstart(2)**2)
        if (ipriuser.gt.0.and.nring.eq.10) type '(' xstart=',3f8.3,' rodecay=',f8.3)',xstart,rodecay
      xx(1)=xv(3) ; xx(2)=xstart(3) ; yy(1)=rovertex
      yy(2)=rodecay ; if (ipriuser.gt.0.and.idrawdecay.eq.1) call tvdraw(xx,yy,2)
      nseen=0 ; nri(1)=nring ; ncou(1)=nc ; ran(1)=rkstop
      tdecay=-12.5*Alog(Rndm(myrnd))
c     ncon=0
   10 call caloblock(nring,nc,ninplan,noutplan)
      if (noutplan.le.0) goto 300
      decaylength=decaylength+s2 ; if (decaylength.gt.decayrange) s2=decayrange-decaylength+s2
      nseen=nseen+1 ; if (nseen.gt.10) nseen=10
      nri(nseen)=nring ; ncou(nseen)=nc
      ran(nseen)=s2
c      type '(' decay in',2i3,' plans',2i3,' s1  s2=',2f10.3)',nring,nc,ninplan,noutplan,s1,s2
      do i=1,3 ; xstart(i)=xstart(i)+(s2+0.01)*cdir(i) ; enddo
      rodecay=sqrt(xstart(1)**2+xstart(2)**2)
      xx(1)=xx(2) ; xx(2)=xstart(3) ; yy(1)=yy(2)
      yy(2)=rodecay
       if (ipriuser.gt.0.and.idrawdecay.eq.1) call tvdraw(xx,yy,2)
      if (noutplan.eq.1.or.noutplan.eq.2) goto 300
```

```
      if (decaylength.gt.decayrange) goto 500
c      type '(' xnew ',3f10.3,' rodecay',f10.3,' cdir',3f10.3)',xstart,rodecay,cdir
       if (incalbar.eq.1.and.(noutplan.eq.3.or.noutplan.eq.5))goto 300
      if (noutplan.eq.3) nring=nring+1
      if (noutplan.eq.5) nring=nring-1
       if (nring.eq.9.or.nring.eq.0) goto 300
       phinew=pi-atan2(xstart(2),xstart(1))
        nc=phinew*24/pi +1 ; if (nring.le.3.or.nring.eq.10) nc=phinew*12/pi+1
      if (nring.eq.1) nc=phinew*6/pi+1
       if (nring.eq.nri(nseen).and.nc.eq.ncou(nseen)) goto 600
     goto 10
   80 continue ; if (ipriuser.le.0) return
     type '(' track',i2,' out of calorimeter ring ='i2)',itrack,nring ; return
  100 return
  200 continue ; if (ipriuser.le.0) return
     type '(' track',i2,' k+ escapes : range=',f10.3,)',itrack,rkstop1 ; return
  300 continue ; if (ipriuser.le.0) return
      type '(' track',i2,' in block',2i3,' t=',f8.2,'  out  after blocks',9(/,10x,2i3,3x,f10.3))',itrack
    1     ,nri(1),ncou(1),tdecay,(nri(i),ncou(i),ran(i),i=1,nseen)
     type '('     incalbar=',i2,' noutplan=',i3)',incalbar,noutplan
   return
  500 do i=1,3 ; xyzmustop(i)=xstart(i) ; enddo ; mustopped=1
    if (ipriuser.le.0) return
      type '(' track',i2,' in block',2i3,' t=',f8.2,'  stop after blocks',9(/,10x,2i3,3x,f10.3))',itrack
    1     ,nri(1),ncou(1),tdecay,(nri(i),ncou(i),ran(i),i=1,nseen) ; return
  600 continue ; if (ipriuser.le.0)  return
      type '(' track',i2,' in block',2i3,' t=',f8.2,'  loop after blocks',9(/,10x,2i3,3x,f10.3))',itrack
    1     ,nri(1),ncou(1),tdecay,(nri(i),ncou(i),ran(i),i=1,nseen)
   return
   end

        Function RangeEnergy(E)
c     from my approx. reading of the PDG table and fitting to 4th degree
       RangeEnergy=-296.16+1746.7*E-3906.2*E**2+3859.7*E**3-1331.6*E**4
    end

        subroutine calbarfind
       common / caloxyz / xcalo(8,48,8),ycalo(8,48,8),zcalo(8,48,8),nptplan(3,6)
     1      ,xcalbar(8,24),ycalbar(8,24),zcalbar(8,24),rintbar,rextbar
      common /mycalo/roatentry(9),jcaloring(2),jcalophi(2),calparc(2)
     common /decaykplus/cosdir(3),xv(3),pk,itrack
      common /decay1/cdir(3),xstart(3),pstart,s1,s2,s1init(2),s2init(2)
    call vzero(jcaloring,6)
   nc=0 ; parcours=0.
   parcours2=0. ; parcours1=0.
    s1nc=0. ; s2nc=0. ; s1up=0. ; s2up=0. ; s1down=0. ; s2down=0.
     sit=sqrt(1.-cosdir(3)**2) ; s=0.5*(rintbar+rextbar)/sit
     x=xv(1)+s*cosdir(1) ; y=xv(2)+s*cosdir(2) ; z=xv(3)+s*cosdir(3) ; ro=sqrt(x**2+y**2)
      if ( z.lt.0. .or. z.gt.rintbar .or. ro.lt.rintbar.or.ro.gt.rextbar) return
    call ucopy(cosdir,cdir,7)
  1  phinew=pi-atan2(y,x)
     nc=phinew*12/pi +1 ; nring=10 !  by convention ---> means barrel calorimeter
      call caloblock(nring,nc,ninplan,noutplan) ; nhors=noutplan
    if (noutplan.lt.0) goto 10
     parcours=s2-s1 ; s1nc=s1 ; s2nc=s2
     jcaloring(1)=nring     ; jcalophi(1)=nc
     calparc(1)=parcours    ; s1init(1)=s1nc   ; s2init(1)=s2nc
    if (ipriuser.gt.0) type '(' calbarfind says nc= ',i3,' parcours=',f10.3)',nc,parcours
     if (ipriuser.gt.0) type '(' s1 s2=',2f8.3)',s1init(1),s2init(1)
   return
  10 continue
     type '(' calbarfind --> track is not in block, nring=',i2,' ncounter=',i3,
     1     '  ninplan',i2,' noutplan',i2,' s1 s2=',2e12.5)',nring,nc,ninplan,noutplan,s1,s2
   end
```

```
      subroutine calofind2
      common /mycalo/roatentry(9),jcaloring(2),jcalophi(2),calparc(2)
      common /decaykplus/cosdir(3),xv(3),pk,itrack
      common /decay1/cdir(3),xstart(3),pstart,s1,s2,s1init(2),s2init(2)
      call vzero(jcaloring,6)
      nc=0 ; nring=0 ; parcours=0.
      parcours2=0. ; parcours1=0.
      s1nc=0. ; s2nc=0. ; s1up=0. ; s2up=0. ; s1down=0. ; s2down=0.
      cot=cosdir(3) ; s=(65.91-xv(3))/cot
      x=xv(1)+s*cosdir(1) ; y=xv(2)+s*cosdir(2) ; ro=sqrt(x**2+y**2)
      call ucopy(cosdir,cdir,7)
      if (ro.lt.roatentry(1)) return ; nring=1
      do iedge=2,9 ; if (ro.lt.roatentry(iedge)) goto 1 ; nring=iedge ; enddo
    1 phinew=pi-atan2(y,x)
      nc=phinew*24/pi +1 ; if (nring.le.3) nc=phinew*12/pi+1 ; if (nring.eq.1) nc=phinew*6/pi+1
      nup=phinew*24/pi +1 ; if (nring+1.le.3) nup=phinew*12/pi+1
      ndown=phinew*24/pi +1 ; if (nring-1.le.3) ndown=phinew*12/pi+1
      nringm1=nring-1 ; nringpl1=nring+1
      if (nring-1.eq.1) ndown=phinew*6/pi+1
      if (nring.eq.9) then
        call caloblock(nringm1,ndown,ninplan,noutplan)
        if (noutplan.gt.0) then parcours1=s2-s1 ; s1down=s1 ; s2down=s2 ; endif
        goto 20
      endif
      call caloblock(nring,nc,ninplan,noutplan) ; nhors=noutplan
      if (noutplan.lt.0) goto 10
      parcours=s2-s1 ; s1nc=s1 ; s2nc=s2
      if (ninplan.eq.5.and.nring.gt.1) then
        call caloblock(nringm1,ndown,ninplan,noutplan)
        if (noutplan.gt.0) then parcours1=s2-s1 ; s1down=s1 ; s2down=s2 ; endif
      endif
      if (nhors.eq.3.and.nring.lt.8) then
        call caloblock(nringpl1,nup,ninplan,noutplan)
        if (noutplan.gt.0) then parcours2=s2-s1 ; s1up=s1 ; s2up=s2 ; endif
      endif
   20 if (parcours1.gt.0.) then
        jcaloring(1)=nring-1  ; jcalophi(1)=ndown
        calparc(1)=parcours1  ; s1init(1)=s1down ; s2init(1)=s2down
        jcaloring(2)=nring    ; jcalophi(2)=nc
        calparc(2)=parcours   ; s1init(2)=s1nc   ; s2init(2)=s2nc
      else
        jcaloring(1)=nring    ; jcalophi(1)=nc
        calparc(1)=parcours   ; s1init(1)=s1nc   ; s2init(1)=s2nc
        jcaloring(2)=nring+1  ; jcalophi(2)=nup
        calparc(2)=parcours2  ; s1init(2)=s1up   ; s2init(2)=s2up
      endif
      return
   10 type '(' calofind2 --> track is not in block, nring=',i2,' ncounter=',i3,
     1     ' ninplan',i2,' noutplan',i2,' s1 s2=',2e12.5)',nring,nc,ninplan,noutplan,s1,s2
      end

            Subroutine caloblock(nring,nc,ninplan,noutplan)
      common / caloxyz / xcalo(8,48,8),ycalo(8,48,8),zcalo(8,48,8),nptplan(3,6)
     1     ,xcalbar(8,24),ycalbar(8,24),zcalbar(8,24),rintbar,rextbar
      common /decay1/cdir(3),xstart(3),pstart,s1,s2,s1init(2),s2init(2)
      dimension coord(3,8),v1(3),v2(3),vperp(3),vx0(3),splan(6),isigneplan(6)
      dimension xout(3),nporder(6)
c --- Convention : nring=10 ---> means barrel calorimeter
      if (nring.lt.1.or.nring.eq.9.or.nring.gt.10) goto 200
      if (nring.eq.10)
      then
          do i=1,8 ; coord(1,i)=xcalbar(i,nc)
                coord(2,i)=ycalbar(i,nc)
                coord(3,i)=zcalbar(i,nc)
          enddo
      else
          do i=1,8 ; coord(1,i)=xcalo(i,nc,nring)
                coord(2,i)=ycalo(i,nc,nring)
```

```
                coord(3,i)=zcalo(i,nc,nring)
            enddo
          endif
           do iplan=1,6 ; i1=nptplan(1,iplan) ; i2=nptplan(2,iplan) ; i3=nptplan(3,iplan)
            do i=1,3 ; v1(i)=coord(i,i2)-coord(i,i1) ; v2(i)=coord(i,i3)-coord(i,i1)
            vx0(i)=xstart(i)-coord(i,i1)
           enddo
            call cross(v1,v2,vperp) ; d2=vdot(vperp,cdir,3)
            splan(iplan)=-vdot(vperp,vx0,3)/d2
            isigneplan(iplan)=-1 ; if (d2.lt.0.) isigneplan(iplan)=1
         enddo
           call sortzv(splan,nporder,6,1,0,0)
          noutplan=0 ; ninplan=0 ; s1=0. ; s2=0.
         do i=1,6 ; j=nporder(i)
            if (isigneplan(j).lt.0.and.noutplan.eq.0) then noutplan=j ; s2=splan(j)
             if (i.gt.1) then ninplan=nporder(i-1) ; s1=splan(ninplan) ; endif
          endif
            if (isigneplan(j).gt.0.and.noutplan.gt.0) noutplan=-1
c       type '(' plan',i2,' signe',i3,' splan',f10.3,' s1 s2',2f10.3)',j
c    1          ,isigneplan(j),splan(j),s1,s2
         enddo

          if(ipriUser.gt.0) then
            if (noutplan.lt.0.and.nring.eq.10) then type '(' error in caloblock, nring='
          1       ,i2,' ncounter=',i3,' ninplan',i2,' noutplan',i2,' s1 s2=',2e12.5)',nring
          1      ,nc,ninplan,noutplan,s1,s2
              do i=1,6 ; j=nporder(i)
                   type '(' plan',i2,' signe',i3,' splan',f10.3,' s1 s2',2f10.3)',j
          1             ,isigneplan(j),splan(j),s1,s2
            enddo
           endif
         endif
         return
       200 type '(' Error : Caloblock called with nring=',i3)',nring
         end
```

```
      Subroutine UserSpit
     type *,' End of run, now spit'
      type '(i7,' Events',i7,'  tried     :',i6,'  n4for',i6,' n3for',i6)',locev
     1             ,nevtried,n4for,n3for1bar
     type '(' nbseen       ',10i7)',nbseen
     type '(' nevkseen     ',5i7)',nevkseen
     print '(/,1x,'4F events with muons traversing at least one block beyond the first:'
      * ,/,1x,'decays per event =',t30,'0',t40,'1',t50,'2',t60,'3',t70,'4'
      * ,/,t21,5i10,/,t21,5f10.3)',nevkplusseen,(float(nevkplusseen(i))/float(n4for),i=1,5)
     print '(/,1x,'3F events with muons traversing at least one block beyond the first:'
      * ,/,1x,'decays per event =',t30,'0',t40,'1',t50,'2',t60,'3',t70,'4'
      * ,/,t21,5i10,/,t21,5f10.3)',nevkplusseenbar,(float(nevkplusseenbar(i))/float(n3for1bar),i=1,5)

     print '(/,1x,'4F events with muons traversing only the first block:'
      * ,/,1x,'decays per event =',t30,'0',t40,'1',t50,'2',t60,'3',t70,'4'
      * ,/,t21,5i10,/,t21,5f10.3,/,1x,'useful for mult-hit TDCs =',i10)',nevkplusunseen
      * ,(float(nevkplusunseen(i))/float(n4for),i=1,5),nevkplus11
     print '(/,1x,'3F events with muons traversing only the first block:'
      * ,/,1x,'decays per event =',t30,'0',t40,'1',t50,'2',t60,'3',t70,'4'
      * ,/,t21,5i10,/,t21,5f10.3,/,1x,'useful for mult-hit TDCs =',i10)',nevkplusunseenbar
      * ,(float(nevkplusunseenbar(i))/float(n3for1bar),i=1,5),nevkplus11bar

     type '(//,1x,' 4F events ----------',/,'  nev nseen track1',3i7,/,
     1      ' versus   track3',3i7,/,
     2     '           ',3i7)',nevtrack4F
      N_multihit=nevtrack4F(2,2)+nevtrack4F(2,3)+nevtrack4F(3,2)
      N_total   =nevtrack4F(2,2)+nevtrack4F(2,3)+nevtrack4F(3,2)+nevtrack4F(3,3)
      print '(1x,'multi-hit needed for ',t30,i10,'[',f5.3,']',/
      *,1x,'single-hit sufficient for',t30,i10,'[',f5.3,']',/
      *,1x,'total identifiable = ',t30,i10,'[',f5.3,']')',N_multihit,float(N_multihit)/float(n4for)
     *,nevtrack4F(3,3),float(nevtrack4F(3,3))/float(n4for)
     *,N_total,float(N_total)/float(n4for)

     type '(//,'  3F events ----------',/,'  nev nseen track1',3i7,/,
     1      ' versus   track3',3i7,/,
     2     '           ',3i7)',nevtrack3F
      N_multihit=nevtrack3F(2,2)+nevtrack3F(2,3)+nevtrack3F(3,2)
      N_total   =nevtrack3F(2,2)+nevtrack3F(2,3)+nevtrack3F(3,2)+nevtrack3F(3,3)
      print '(1x,'multi-hit needed for ',t30,i10,'[',f5.3,']',/
      *,1x,'single-hit sufficient for',t30,i10,'[',f5.3,']',/
      *,1x,'total identifiable = ',t30,i10,'[',f5.3,']')',N_multihit,float(N_multihit)/float(n3for1bar)
     *,nevtrack3F(3,3),float(nevtrack3F(3,3))/float(n3for1bar)
     *,N_total,float(N_total)/float(n3for1bar)
      print '(/,1x,'Decay distribution in 2 ns bins:',5(/,13x,10i7),/
      * ,5(/,13x,10f7.3))',nanosec,(0.25*float(nanosec(i))/float(locev),i=1,50)
         print '(/,1x,'col. a = bin number',/
     *       ,1x,'col. b = x of Kaon stopping point (1 cm bins)',/
     *       ,1x,'col. c = y of Kaon stopping point (1 cm bins)',/
     *       ,1x,'col. d = z of Kaon stopping point (1 cm bins)',/
     *       ,1x,'col. e = rho of Kaon stopping point (1 cm bins)',/
     *       ,1x,'col. f = x of muon stopping point (1 cm bins)',/
     *       ,1x,'col. g = y of muon stopping point (1 cm bins)',/
     *       ,1x,'col. h = z of muon stopping point (1 cm bins)',/
     *       ,1x,'col. i = rho of muon stopping point (1 cm bins)',/
      *          ,t6,'a',t16,'b',t26,'c',t36,'d',t46,'e',t56,'f',t66,'g'
      *         ,t76,'h',t87,'i',t96,'j',t106,'k',/)'
        do i=1,100
             print '(1x,i5,10i10)',i,his_Kx(i),his_Ky(i),his_Kz(i),his_Krho(i)
     *          ,his_mux(i),his_muy(i),his_muz(i),his_murho(i)
       enddo
       if(iWantHisto.eq.1)then
        call FileStatus(30,OK,i,FileOutNameMax)
       close(30)
      endif
     end
```

```
      Subroutine MyUserMenus
     integer*2 Id
     MenuID(3) = 81 ! Must be different from other menus. Standard is 80
     Mh = NewMenu(MenuID(3),'User Control')
     MenuHandle(3) = Mh
     call AppendMenu(Mh,'Status')    ! menuflag will be 1
     call AppendMenu(Mh,'Spit')   ! menuflag will be 2
     call AppendMenu(Mh,'End')   ! menuflag will be 3
     call AppendMenu(Mh,'Stop')   ! menuflag will be 4
     call InsertMenu(Mh,0)
    call DrawMenuBar
    Nbmenus=3
   END

c ---------------------------------------------------------------------------------------------------------------------

     Public phigener,pgeninit
    COMMON /GENIN / NP,TECM,AMASS(18),KGENEV
    COMMON /GENOUT/ PCMG(5,18),WT
     common /random/srnd(2)
     integer *4 srnd
c  Différent de Phigener par la possibilité d'engendrer les decays de phi's en K's de
c  manière non isotrope.
         Subroutine phigener(pgener)
      common/ transfert / centerofmass(4),cotetaphicm
       Common/ myphi / mp,mk,mpi,mphi1,mphi2,ephi1,ephi2,phimom,me,P
     1          ,pin(4),pcm(4),ppbar(4),pphi(4,2),pkcm(4,4),pkphi(4,4),pout(4)
     2          ,pk(2),ek(2),apcm(4),pp(4),pklab(4,4),pi,phiwidth1,phiwidth2
    3       ,nphi,massdecay(4)
     dimension pgener(2),pwork(3)
      real *4 mp,mk,mphi1,mphi2,mpi,me,massdecay
     relmom(A,B,C)=sqrt( (a**2-(b+c)**2) * (a**2-(b-c)**2) )/(2.*a)
     if (nphi.eq.0) goto 10
     coteta=2.*rndm(srnd)-1.
c     coteta= rndm(srnd)+rndm(srnd)  ; if (coteta.gt.1.) coteta=2.-coteta
     siteta=sqrt(1.-coteta**2)
     cotetaphicm=coteta
      phi=2.*pi*rndm(srnd) ; cophi=cos(phi) ; siphi=sin(phi)
   1  call normal(srnd,xmass1,xmass2) ; actualmass1=mphi1+phiwidth1*xmass1
     actualmass2=mphi2+phiwidth2*xmass2
     if (actualmass1+actualmass2+.0001.gt.pin(4)) goto 1
     if (actualmass1.lt.massdecay(1)+massdecay(2)) goto 1
     if (actualmass2.lt.massdecay(3)+massdecay(4)) goto 1
     phimom=relmom(pin(4),actualmass1,actualmass2)
      ephi1=sqrt(phimom**2+actualmass1**2) ; ephi2=sqrt(phimom**2+actualmass2**2)
    a=ephi1+ephi2
      pphi(3,1)=coteta*phimom ; pphi(2,1)=siteta*siphi*phimom
     pphi(1,1)=siteta*cophi*phimom
     pphi(4,1)=ephi1 ; pphi(4,2)=ephi2
      do i=1,3 ; pphi(i,2)=-pphi(i,1) ; enddo
    pk(1)=Relmom(actualmass1,massdecay(1),massdecay(2))
    pk(2)=Relmom(actualmass2,massdecay(3),massdecay(4))
   do 2 j=1,2
c   pwork is the phi vector, used for its direction, will be normalized in scatt
c   and returned as unit vector rotated by teta and phi according to some law.
     do i=1,3 ; pwork(i)=pphi(1,j) ; enddo
     call scatt(pwork)
c       swork2=pwork(1)**2+pwork(2)**2+pwork(3)**2
c       type *,' back from scatt, pwork=',pwork,swork2
     k=2*j-1
      pkcm(1,k)=pwork(1)*pk(j) ; pkcm(2,k)=pwork(2)*pk(j)
     pkcm(3,k)=pwork(3)*pk(j)
      pkcm(4,k)=sqrt(pk(j)**2+massdecay(k)**2) ; pkcm(4,k+1)=sqrt(pk(j)**2+massdecay(k+1)**2)
      do i=1,3 ; pkcm(i,k+1)=-pkcm(i,k) ; enddo
  2 continue
c     ---------------------------------- Retour au système du phi-phi au repos
      call loren4(pphi(1,2),pkcm(1,1),pkphi(1,1))
      call loren4(pphi(1,2),pkcm(1,2),pkphi(1,2))
      call loren4(pphi(1,1),pkcm(1,3),pkphi(1,3))
```

```
       call loren4(pphi(1,1),pkcm(1,4),pkphi(1,4))
      do i=1,4 ; pout(i)=pkphi(i,1)+pkphi(i,2) ; enddo
      do i=1,4 ; pout(i)=pkphi(i,3)+pkphi(i,4) ; enddo
c     ------------------------------------------------- Retour dans le Laboratoire
     do j=1,4
       call loren4(apcm,pkphi(1,j),pklab(1,j)) ; enddo
c     ------------------------------------------------- Fin de la Génération
      call ucopy(pklab,pgener,16)
     return
    10 continue ; np=4 ; kgenev=1
     do i=1,4 ; amass(i)=massdecay(i) ; enddo
     call genbod
     do j=1,4
       call loren4(apcm,pcmg(1,j),pklab(1,j)) ; enddo
      call ucopy(pklab,pgener,16)
     end


        Subroutine Pgeninit
     common/ transfert / centerofmass(4)
       Common/ myphi / mp,mk,mpi,mphi1,mphi2,ephi1,ephi2,phimom,me,P
     1          ,pin(4),pcm(4),ppbar(4),pphi(4,2),pkcm(4,4),pkphi(4,4),pout(4)
     2          ,pk(2),ek(2),apcm(4),pp(4),pklab(4,4),pi,phiwidth1,phiwidth2
    3       ,nphi,massdecay(4)
     character *4 part
      real *4 mp,mk,mphi1,mphi2,mpi,me,massdecay
     srnd(1)=65432 ; srnd(2)=94756
      mp=.93827 ; mk= 0.49365 ; mpi=.13957 ; mphi1=1.0195 ; mphi2=1.0195 ; me=.0005
     pi=4.*atan(1.)
     phiwidth1=0. ; phiwidth2=0. ;P=2. ; nphi=2
     call ReadVal(' How many phis ? (0,1, or 2)--->',nphi,'i3')
     call ReadVal(' give momentum of pbar ( for phi generation )--->',P,'f10.6')
     if (nphi.ge.1) then
      call ReadVal(' give Mass  of phi1 --->',mphi1,'f10.6')
      call ReadVal(' give width of phi1 --->',Phiwidth1,'f10.6')
    endif
     if (nphi.eq.2) then
      call ReadVal(' give Mass  of phi2--->',mphi2,'f10.6')
      call ReadVal(' give width of phi2 --->',Phiwidth2,'f10.6')
    endif
   10  type *,' give th 4 particles decays (P,K,p) --->$' ; read *,part
     do i=1,4 ; if (part(i:i).eq.'P') massdecay(i)=mp ;if (part(i:i).eq.'K') massdecay(i)=mk
       if (part(i:i).eq.'p') massdecay(i)=mpi
        if (part(i:i).ne.'K'.and.part(i:i).ne.'P'.and.part(i:i).ne.'p') goto 10 ; enddo
     type *,' Masses are ',massdecay
     do i=1,4 ; amass(i)=massdecay(i) ; enddo
     call vzero(pp,4) ; call vzero(ppbar,4)
     ppbar(3)=P ; ppbar(4)=sqrt(ppbar(3)**2+mp**2)
c    assume the jet's proton has 0.0 ev/c
     pp(4)=sqrt(pp(2)**2+mp**2)
     pcm(1)=ppbar(1)+pp(1) ; apcm(1)=-pcm(1)
     pcm(2)=ppbar(2)+pp(2) ; apcm(2)=-pcm(2)
     pcm(3)=ppbar(3)+pp(3) ; apcm(3)=-pcm(3)
     pcm(4)=ppbar(4)+pp(4) ; apcm(4)= pcm(4)
     print 2000,' pcm = ',pcm
     call loren4(pcm,pcm,pin)
     tecm=pin(4) ; np=4
     ephi=pin(4)/2.
     call ucopy( pcm,centerofmass,4)
 2000  format(1x,a,5e12.5)
    end
c    Ancienne routine de rotation dans l'espace d'un vecteur unitaire utilisée pour fabriquer les k decays de phis
     Subroutine scatt(v)
       Common/ myphi / mp,mk,mpi,mphi1,mphi2,ephi1,ephi2,phimom,me,P
     1          ,pin(4),pcm(4),ppbar(4),pphi(4,2),pkcm(4,4),pkphi(4,4),pout(4)
     2          ,pk(2),ek(2),apcm(4),pp(4),pklab(4,4),pi,phiwidth1,phiwidth2,nphi
      dimension n(3),normp(3),v(3)
     real n,normp
```

```
      pi=4.*atan(1.)
       a=sqrt(v(1)**2+v(2)**2+v(3)**2)
      do i=1,3 ; v(i)=v(i)/a ; enddo
       a=sqrt(v(1)**2+v(2)**2)
      n(1)=v(2)/a
      n(2)=-v(1)/a
      n(3)=0.
      normp(3)=-a
       normp(1)=v(1)*v(3)/a
       normp(2)=v(2)*v(3)/a
c     distribution de costeta = plate
      a=2.*rndm(srnd)-1. ; siteta=sqrt(1.-a*a) ; coteta=a
       phi  =rndm(srnd)*2.*pi ; si=sin(phi)*siteta ; co=cos(phi)*siteta
      v(1)=  v(1)*coteta + n(1)*co + normp(1)*si
      v(2)=  v(2)*coteta + n(2)*co + normp(2)*si
      v(3)=  v(3)*coteta + n(3)*co + normp(3)*si
     return
     end


        FUNCTION PDK(A,B,C)
C-- CALLED FROM - GENEV
C     PDK = SQRT(A*A+(B*B-C*C)**2/(A*A) - 2.0*(B*B+C*C))/2.0
     A2 = A*A
     B2 = B*B
     C2 = C*C
      PDK = 0.5*SQRT(A2 + (B2-C2)**2/A2 - 2.0*(B2+C2))
     RETURN
     END

          SUBROUTINE ROTES2(C,S1,C2,S2,PR,I)
C-- CALLED FROM - GENEV
C       THIS SUBROUTINE NOW DOES TWO ROTATIONS (XY AND XZ)
c      modified 28/6/88 to replace variable S (used in common for rndm generator) by S1
     DIMENSION PR(50)
     K1 = 5*I - 4
     K2 = K1 + 1
     SA = PR(K1)
     SB = PR(K2)
     A     = SA*C - SB*S1
     PR(K2) = SA*S1 + SB*C
     K2 = K2 + 1
     B = PR(K2)
     PR(K1) = A*C2 - B*S2
     PR(K2) = A*S2 + B*C2
     RETURN
     END

        SUBROUTINE FLPSOR(A,N)
C
C CERN PROGLIB# M103    FLPSOR        .VERSION KERNFOR  3.15  820113
C ORIG. 29/04/78
C
C  SORT THE ONE-DIMENSIONAL FLOATING POINT ARRAY A(1),...,A(N) BY
C  INCREASING VALUES
C
C-   PROGRAM  M103  TAKEN FROM CERN PROGRAM LIBRARY,  29-APR-78
C
     DIMENSION A(N)
     COMMON /SLATE/ LT(20),RT(20)
     INTEGER R,RT
C
     LEVEL=1
     LT(1)=1
     RT(1)=N
  10 L=LT(LEVEL)
     R=RT(LEVEL)
     LEVEL=LEVEL-1
  20 IF(R.GT.L) GO TO 200
```

```
      IF(LEVEL) 50,50,10
C
C  SUBDIVIDE THE INTERVAL L,R
C    L : LOWER LIMIT OF THE INTERVAL (INPUT)
C    R : UPPER LIMIT OF THE INTERVAL (INPUT)
C    J : UPPER LI T OF LOWER SUB-INTERVAL (OUTPUT)
C    I : LOWER LIMIT OF UPPER SUB-INTERVAL (OUTPUT)
C
  200 I=L
      J=R
      M=(L+R)/2
      X=A(M)
  220 IF(A(I).GE.X) GO TO 230
      I=I+1
      GO TO 220
  230 IF(A(J).LE.X) GO TO 231
      J=J-1
      GO TO 230
C
  231 IF(I.GT.J) GO TO 232
      W=A(I)
      A(I)=A(J)
      A(J)=W
      I=I+1
      J=J-1
      IF(I.LE.J) GO TO 220
C
  232 LEVEL=LEVEL+1
      IF((R-I).GE.(J-L)) GO TO 30
      LT(LEVEL)=L
      RT(LEVEL)=J
      L=I
      GO TO 20
   30 LT(LEVEL)=I
      RT(LEVEL)=R
      R=J
      GO TO 20
   50 RETURN
      END

          SUBROUTINE GENBOD
c  Modified 3/7/1988 to rename PCM as pcmg to avoid clash with other common.
C  SUBROUTINE TO GENERATE N-BODY EVENT
C  ACCORDING TO FERMI LORENTZ-INVARIANT PHASE SPACE
C  ADAPTED FROM FOWL (CERN W505) SEPT. 1974 BY F. JAMES
C  EVENTS ARE GENERATED IN THEIR OWN CENTER-OF-MASS,
C  BUT MAY BE TRANSFORMED TO ANY FRAME USING LOREN4
C
C  INPUT TO SUBROUTINE IS THRU COMMON BLOCK GENIN
C         NP=NUMBER OF OUTGOING PARTICLES (.LT. 19)
C         TECM=TOTAL ENERGY IN CENTER-OF-MASS
C         AMASS(I)=MASS OF ITH OUTGOING PARTICLE
C         KGENEV=1 FOR CONSTANT CROSS SECTION
C                2 FOR FERMI ENERGY-DEPENDANCE
C
C  OUTPUT FROM SUBROUTINE IS THRU COMMON BLOCK GENOUT
C         PCM(1,I)=X-MOMENTUM IF ITH PARTICLE
C         PCM(2,I)=Y-MOMENTUM IF ITH PARTICLE
C         PCM(3,I)=Z-MOMENTUM IF ITH PARTICLE
C         PCM(4,I)=ENERGY OF ITH PARTICLE
C         PCM(5,I)=MOMENTUM OF ITH PARTICLE
C         WT=WEIGHT OF EVENT
C    COMMON/GENIN / NP, TECM, AMASS(18), KGENEV
C    COMMON/GENOUT/ PCM(5,18) , WT
     DIMENSION EMM(18)
     DIMENSION RNO(50)
C--      PCM1 IS LINEAR EQUIV. OF PCM TO AVOID DOUBLE INDICES
      DIMENSION EM(18),PD(18),EMS(18),SM(18),FFQ(18),PCM1(90)
```

```
      EQUIVALENCE (NT,NP),(EM,AMASS),(PCM1,pcmg)
       save ffq,knt,twopi
 C FFQ(N)=PI * (TWOPI)**(N-2) / (N-2)FACTORIAL
        DATA FFQ/0.,3.141592, 19.73921, 62.01255, 129.8788, 204.0131,
     2                  256.3704, 268.4705, 240.9780, 189.2637,
     3                  132.1308,  83.0202,  47.4210,  24.8295,
     4                  12.0006,   5.3858,   2.2560,   0.8859/
       DATA KNT/0/,TWOPI/6.2831853073/
 C     INITIALIZATION
      KNT=KNT + 1
      IF(KNT.GT.1) GOTO 100
      PRINT 1160
      PRINT 1200, NP,TECM,(AMASS(JK),JK=1,NP)
  100 CONTINUE
      IF(NT.LT.2) GOTO 1001
      IF(NT.GT.18) GOTO 1002
      NTM1=NT-1
      NTM2=NT-2
      NTP1=NT+1
      NTNM4=3*NT - 4
      EMM(1)=EM(1)
      TM=0.0
      DO 200 I=1,NT
      EMS(I)=EM(I)**2
      TM=TM+EM(I)
  200  SM(I)=TM
 C     CONSTANTS DEPENDING ON TECM
      TECMTM=TECM-TM
      IF(TECMTM.LE.0.0) GOTO 1000
      EMM(NT)=TECM
      IF(KGENEV.GT.1) GOTO 400
 C     CONSTANT CROSS SECTION AS FUNCTION OF TECM
      EMMAX=TECMTM+EM(1)
      EMMIN=0.0
      WTMAX=1.0
      DO 350 I=2,NT
      EMMIN=EMMIN+EM(I-1)
      EMMAX=EMMAX+EM(I)
       if (emmax.le.emmin+em(i)) goto 100
  350 WTMAX=WTMAX*PDK(EMMAX,EMMIN,EM(I))
      WTMAXQ=1.0/WTMAX
      GOTO 455
 C--    FERMI ENERGY DEPENDENCE FOR CROSS SECTION
  400 WTMAXQ=TECMTM**NTM2*FFQ(NT) / TECM
 C     CALCULATION OF WT BASED ON EFFECTIVE MASSES EMM
  455 CONTINUE
 C--          FILL RNO WITH 3*NT-4 RANDOM NUMBERS,
 C--        OF WHICH THE FIRST NT-2 ARE ORDERED.
      DO 457 I= 1, NTNM4
  457 RNO(I)=RNDM(Smd)
       IF(NTM2) 900,509,460
  460 CONTINUE
     CALL FLPSOR(RNO,NTM2)
     DO 508 J=2,NTM1
  508 EMM(J)=RNO(J-1)*(TECMTM)+SM(J)
  509 WT=WTMAXQ
     IR=NTM2
     DO 530 I=1,NTM1
      if (emm(i+1).le.emm(i)+em(i+1)) goto 100
     PD(I)=PDK(EMM(I+1),EMM(I),EM(I+1))
  530 WT=WT*PD(I)
 C--     COMPLETE SPECIFICATION OF EVENT (RAUBOLD-LYNCH METHOD)
       pcmg(1,1)=0.0
      pcmg(2,1)=PD(1)
       pcmg(3,1)=0.0
     DO 570 I=2,NT
      pcmg(1,I)=0.0
     pcmg(2,I)=-PD(I-1)
      pcmg(3,I)=0.0
```

```
      IR=IR+1
      BANG=TWOPI*RNO(IR)
      CB=COS(BANG)
      SB=SIN(BANG)
      IR=IR+1
       C=2.0*RNO(IR)-1.0
       S=SQRT(1.0-C*C)
      IF(I.EQ.NT) GOTO 1567
      ESYS=SQRT(PD(I)**2+EMM(I)**2)
      aBETA=PD(I)/ESYS
      GAMA=ESYS/EMM(I)
       DO 568 J=1,I
      NDX=5*J - 5
       AA= PCM1(NDX+1)**2 + PCM1(NDX+2)**2 + PCM1(NDX+3)**2
      PCM1(NDX+5)=SQRT(AA)
      PCM1(NDX+4)=SQRT(AA+EMS(J))
       CALL ROTES2(C,S,CB,SB,PCMg,J)
       PSAVE=GAMA*(pcmg(2,J)+aBETA*pcmg(4,J))
  568 pcmg(2,J)=PSAVE
      GOTO 570
 1567 DO 1568 J=1,I
        AA=pcmg(1,J)**2 + pcmg(2,J)**2 + pcmg(3,J)**2
      pcmg(5,J)=SQRT(AA)
      pcmg(4,J)=SQRT(AA+EMS(J))
       CALL ROTES2(C,S,CB,SB,PCMg,J)
 1568 CONTINUE
  570 CONTINUE
  900 CONTINUE
      RETURN
C       ERROR RETURNS
 1000 PRINT 1100
      GOTO 1050
 1001 PRINT 1101
      GOTO 1050
 1002 PRINT 1102
 1050 PRINT 1150, KNT
      PRINT 1200, NP,TECM,(AMASS(JK),JK=1,NP)
      STOP
 1100 FORMAT (28H0 AVAILABLE ENERGY NEGATIVE )
 1101 FORMAT (33H0 LESS THAN 2 OUTGOING PARTICLES )
 1102 FORMAT (34H0 MORE THAN 18 OUTGOING PARTICLES )
 1150 FORMAT (47H0 ABOVE ERROR DETECTED IN GENBOD AT CALL NUMBER,I7)
 1160 FORMAT (34H0 FIRST CALL TO SUBROUTINE GENBOD )
 1200 FORMAT (36H  INPUT DATA TO GENBOD.       NP=  ,I6/
     + ,8H   TECM=,E16.7,18H  PARTICLE MASSES=,5E15.5/(42X,5E15.5)
     +)
      END
```