ADDENDUM/ERRATUM

Ref. TH.3760-CERN

Ref. TH.3761-CERN

R. Odorico^(*)

COJETS: A MONTE CARLO PROGRAM SIMULATING QCD IN HADRONIC
PRODUCTION OF JETS AND HEAVY FLAVORS WITH INCLUSION OF
INITIAL QCD BREMSSTRAHLUNG

This Addendum/Erratum contains, besides corrections of minor bugs, modifications directed to include a renormalizing K factor and the calculations of total fusion cross-sections for heavy flavor production. The version of the program correspondingly corrected has been given number 2.4.

Associated (minor) corrections affecting the companion program WIZJET are also listed. Its corrected version is given number 2.4.

September 1984

(*) Present address: Dipartimento di Fisica, Università di Bologna,
via Irnerio 46, 40126 Bologna, Italy

i) K factor. The UA1 group has recently found (Phys.Lett. 136B (1984) 294) that a renormalizing K factor $\simeq 2$ is necessary to reproduce their high p_T data at the CERN proton-antiproton collider. The program has been modified to include a K factor which can be fixed by the user via the variable FACTK, appearing in the common block COMMON/KFACT/FACTK. The default value of FACTK is 2.. To change it, the user must give to FACTK its value through an execution statement before the call to PREGEN in the main program.

ii) Heavy flavor production. The original version of the program generates events, and calculates the corresponding QCD cross-sections, only for a $Q^2 \geq PTMIN^{**2}$, where Q^2 is the Feynman-Field-Fox variable. This applies also to heavy flavor production via fusion, although in this case the total cross-section is finite. That was in line with the basic motivation of the program, i.e. to provide a simulation of QCD for large p_T physics. On the other hand, with a minimum amount of modification, the program can also be conveniently used for calculating heavy flavor production (via fusion) at small p_T , with the inclusion of initial and final QCD bremsstrahlung effects. The only limitation is associated with the evolution scale, which must always be larger than the initial evolution scale for parton densities, i.e. QZEV, set at 4. GeV in the program. In practice only charm production is affected by this limitation. To eliminate it also for charmed quarks the user can reduce QZEV, consistently modifying at the same time the input parton densities.

For heavy flavor production the Field-Feynman-Fox evolution variable Q^2 is then modified as follows:

$$Q^2 = 2 s \bar{t} \bar{u} / (s^2 + \bar{t}^2 + \bar{u}^2)$$

with

$$\bar{t} = t - m_Q^2, \quad \bar{u} = u - m_Q^2$$

s , t and u being the conventional Mandelstam invariants for the parton subprocess, and m_Q the heavy quark mass. One has:

$$s + \bar{t} + \bar{u} = 0$$

The minimum Q^2 at a given s is determined by

$$(t)_{\min} = - \left(\sqrt{s}/2 - \sqrt{s/4 - m_Q^2} \right)^2$$

This gives

$$\lim_{s \rightarrow \infty} (Q^2)_{\min} = m_Q^2$$

At $s = 4m_Q^2$ one has $Q^2 = 4/3 m_Q^2$. Therefore, with the adopted definition, Q^2 is always larger than m_Q^2 .

When a call to FUSION is made (as indicated in the long-write-up) PTMIN is set equal to m_Q (as given in the array QMAS, after conversion to GeV), unless m_Q is less than QZEV, in which

case PTMIN is set equal to QZEV (actually to $QZEV \times 1.01$, to avoid computer precision problems). In the output tableau printed by the program the quantity PT CUTOFF is set equal to $\sqrt{PTMIN^2 - m_Q^2}$. This corresponds to the asymptotic value of the cutoff in the parton transverse momentum, generated in the (two-body) hard scattering. If PT CUTOFF is printed equal to zero, and PTMGE has been left by the user to its default value, no extra p_T cutoff is operating and the printed cross-section corresponds to the total fusion cross-section. If PTMGE is set to a value larger than PTMIN (fixed as stated above), thus forcing the event generation to large parton transverse momenta ($PT_{HARD} \geq PTMGE$), the printed cross-section refers only to the kinematic region over which the generation has been forced. (All cross-sections are now given in millibarns for uniformity, differently from the original version in which the (partial) fusion cross-section was given in nanobarns.)

When no call to FUSION is made, the quantity PT CUTOFF printed in the output tableau is set equal to PTMIN (as resulting from the default value or as set by the user). If NFLAV (the number of quark flavors activated in the final state) is set larger than 3, the calculated cross-sections and the event generation include the production (via fusion) of heavy quarks up to flavor NFLAV (with their QCD rates). The evolution variable Q^2 remains defined as above, with the understanding that $m_Q = 0$ when the final scattered partons are light. The lower cutoff on Q^2 is always $PTMIN^2$, where PTMIN now is set equal to its default value or is otherwise fixed by the user.

The threshold condition for fusion is given by $s \geq 4m_Q^2$, where m_Q is the heavy quark mass. That is, it is not taken into account the dressing of the quarks into hadrons, which gives limited effects for the higher flavors (bottom and top). In comparing with other calculations of the fusion cross-section, one should also be aware that in the program the QCD coupling constant runs with Q^2 ($\alpha_s = \alpha_s(Q^2)$), as for light partons, and that therefore the value of α_s changes with the heavy quark transverse momentum at a given parton subenergy.

Internally, T and U correspond to the above defined $\bar{t}$ and $\bar{u}$, respectively, except in the function subprograms DSHVQ and DSHVG, where they correspond to the conventional t and u Mandelstam variables.

In running the program no new action is required on the part of the user with respect to those indicated in the original long-write-up. As indicated above, the event generation can still be forced to large parton p_T by setting PTMGE accordingly, and taking the precaution of checking the PTHARD distribution for the finally chosen event sample so as to avoid undue bias of the sample, as it is indicated in the long-write-up.

iii) Bugs. Corrections of minor bugs have been included. Effects from the bugs were comparable to those associated with structure function uncertainties. On computers with a 32 bit word, the use of RNDM(D) in the generation of integer numbers caused on rare occasions the generation of a non-correct integer because of rounding errors. The corresponding instructions have now been

changed so that this problem is eliminated. I would like to thank Dr. S. Pashkov for having pointed out some of the bugs to me.

The corrections for WIZJET are limited to the problem, indicated above, of using RNDM(D) in the generation of integer numbers. The introduction of a K factor is trivial in the case of WIZJET. It simply amounts to multiply the cross sections and the rates by K.

Program COJETS and WIZJET are now published in Computer Physics Communications, 32 (1984), p. 139 and 173, respectively.

```

*****
*
* CORRECTIONS FOR COJETS VERSION 2.4 -- 1 SEP 84 *
*
*****

```

(Card numbers refer to listing in CERN-TH-3760 (1983))

***Replace card 12 by next card
C- VERSION 2.4 - 1 SEP 84

***Replace cards 20 to 21 by next 3 cards
C- REFERENCE - R.ODORICO NUCL.PHYS. B228 (1983) 381, B244 (1984) 313
C- LONG WRITE UP - COMP.PHYS.COMM. 32 (1984) 139 (CERN-TH-3760 (1983))
C- AND ADDENDUM/ERRATUM (SEP 1984)

***Insert next card after card 203
COMMON/KFACT/FACTK

***Insert next 2 cards after card 204
C-- K FACTOR (FROM UA1)
DATA FACTK/2./

***Insert next card after card 618
COMMON/THRFLA/THRFLA(6),AM2HEV(6)

***Insert next 6 cards after card 634
PTCUT=PTMIN
IF(KFRFLA.EQ.0) GO TO 60
DIFF=PTMIN**2-AM2HEV(KFRFLA)
IF(DIFF.LT.1.E-10) PTCUT=0.
IF(DIFF.GE.1.E-10) PTCUT=SQRT(DIFF)
60 CONTINUE

***Replace card 635 by next card
PRINT 21,ALAMB,PTCUT,PT2INT,SIGINL

***Remove card 642

***Replace cards 645 to 646 by next 2 cards
IF(KFRFLA.GT.0) SIGFS=FRHARD*SIGINL
IF(KFRFLA.GT.0.AND.PTMGE.LE.PTMIN) PRINT 29,CHFLA2(KFRFLA-3),SIGFS

***Insert next card after card 647
IF(PTMGE.GT.PTMIN) PRINT 30,PTMGE,SIGHRD

***Replace card 677 by next card
11 FORMAT(T2,47(1H*),26H VERSION 2.4 -- 1 SEP 84 ,47(1H*))

***Replace card 682 by next card
29 FORMAT(T2,1H*,3X,14HX-SECTION FOR ,A4,E10.3,5H (MB) ,T121,1H*/T2,

***Replace card 685 by next card
*6H .GT. ,E10.3,25H (GEV) WITH X-SECTION = ,E10.3,5H (MB) ,T121,

***Insert next card after card 803
COMMON/KFACT/FACTK

```

***Insert next card after card 824
      IF(KFRFLA.GT.0) PTMIN=AMAX1(QZEV*1.01,QMAS(KFRFLA)/1000.)

***Replace card 859 by next card
      AM2HEV(L)=(QMAS(L)/1000.)*2

***Remove card 864 ( XMIN=AMAX1(... )

***Replace card 920 by next card
      FRHARD=WGTHD1/FLOAT(NCASC)*(.19733)**2*10./SIGINL*FACTK

***Replace card 924 by next card
      PROPT=PROPT*(.19733)**2*10./SIGINL*FACTK

***Replace cards 927 to 928 by next 2 cards
      IF(PTMGE.LE.PTMIN) GO TO 105
C-- FORCED GENERATION -- HARD X-SECTION OVER FORCED PT. RANGE

***Replace card 934 by next card
      SIGHRD=SIGHRD/FLOAT(NCASC)*(.19733)**2*10.*FACTK

***Replace card 1110 by next card
      IX=INT(RNDM(0)*NBIN)+1

***Replace cards 1112 to 1113 by next 2 cards
      X=XV(IX)
      IF(X.GE.XVMIN) GO TO 110

***Replace card 1123 by next card
      IX=INT(RNDM(0)*NBIN)+1

***Replace cards 1125 to 1126 by next 2 cards
      X=XV(IX)
      IF(X.GE.XVMIN) GO TO 210

***Replace card 1128 by next card
      IF(RNDM(0).GT.FGLU(X)/F2XQV(7)) GO TO 205

***Replace cards 1297 to 1301 by next 11 cards
91  PARACT(IN,1,IB)=0.
      JC=0
      NB=NBOOK(IB)
      DO 92 J=1,NB
      JZ=PARACT(J,1,IB)
      IF(JZ.EQ.0) GO TO 92
      JC=JC+1
      DO 93 L=1,7
93  PARACT(JC,L,IB)=PARACT(J,L,IB)
92  CONTINUE
      NBOOK(IB)=NBOOK(IB)-ICN

***Replace card 1318 by next card
      XM2=PARACT(J2,3,2)

```


***Insert next 2 cards after card 1328

NENT1=1
NENT2(J2)=1

***Remove cards 1338 to 1339

***Insert next card after card 1361

NENTR=1

***Replace card 1364 by next card

IF(NENT1.GT.0) GO TO 110

***Remove cards 1367 to 1368

***Replace cards 1377 to 1390 by next 22 cards

C-- CANCEL QUANTA TOO SOFT TO PASS THRESHOLD

DO 250 IB=1,2
IF(ICANC(IB).EQ.0) GO TO 250
IF(ICANC(IB).EQ.NBOOK(IB)) RETURN

ICN=ICANC(IB)

DO 251 I=1,ICN

IN=NBKNC(I,IB)

251 PARACT(IN,1,IB)=0.

JC=0

NB=NBOOK(IB)

DO 252 J=1,NB

JZ=PARACT(J,1,IB)

IF(JZ.EQ.0) GO TO 252

JC=JC+1

DO 253 L=1,7

253 PARACT(JC,L,IB)=PARACT(J,L,IB)

252 CONTINUE

NBOOK(IB)=NBOOK(IB)-ICN

ICANC(IB)=0

250 CONTINUE

C

200 CONTINUE

***Replace card 1484 by next card

FUNCTION DSHVQ(A2,S,TQ,UQ)

***Insert next 6 cards after card 1486

DSHVQ=0.

T=TQ+A2

U=UQ+A2

TUMN=-(.5*SQRT(S)-SQRT(.25*S-A2))**2

IF(T.GT.TUMN) RETURN

IF(U.GT.TUMN) RETURN

***Replace card 1491 by next card

FUNCTION DSHVG(A2,S,TQ,UQ)

***Insert next 6 cards after card 1493

DSHVG=0.

T=TQ+A2

U=UQ+A2

TUMN=-(.5*SQRT(S)-SQRT(.25*S-A2))**2

IF(T.GT.TUMN) RETURN

IF(U.GT.TUMN) RETURN

***Insert next card after card 1629
IF(WGT,LT.1.E-30) GO TO 150

***Replace card 1633 by next card
NLOOP=WGT/WGALQ(IALQ,IPACH,2)+1

***Replace card 1661 by next card
IR=INT(2*RNDM(LF))+1

***Replace card 1679 by next card
IR=INT(3*RNDM(LF))+1

***Insert next 2 cards after card 1987
COMMON/NFLAV/NFLAV,KFRFLA
COMMON/THRFLA/THRFLA(6),AM2HEV(6)

***Replace card 1999 by next 15 cards
IF(KFRFLA.GE.4) GO TO 2
IF(MFLA1.EQ.LGLUT) GO TO 5
IF(MFLA2.EQ.LGLUT) GO TO 5
IF(MFLA1.GE.4.AND.MFLA2.GE.4) GO TO 1
GO TO 5
1 CONTINUE
MFLAI1=ABS(PARACT(1,1,1))
IF(MFLAI1.GE.4.AND.MFLAI1.LT.LGLUS) GO TO 5
MFLAI2=ABS(PARACT(1,1,2))
IF(MFLAI2.GE.4.AND.MFLAI2.LT.LGLUS) GO TO 5
2 COSCM=(2.*T+SHAT)/(SQRT(SHAT)*
1 SQRT(SHAT-4.*AM2HEV(MFLAI1)))
GO TO 6
5 COSCM=T/(.5*SHAT)+1.
6 CONTINUE

***Replace card 2433 by next card
IFLAG=INT(NFLAVS*RNDM(0))+1

***Replace card 3815 by next card
MX=INT(RNDM(D)*NB)+1

***Insert next 4 cards after card 4213
IF(IP,LT.15.OR.(IP.GE.51.AND.IP.LE.60).OR.IP.GE.91) GO TO 31
IF((-1.)*IP.GT.0..AND.JJ.GT.0) JJ=JCNPAR(JJ)
IF((-1.)*IP.GT.0..AND.JJ.LT.0) JJ=-JCNQQB(-JJ)
31 CONTINUE

***Replace card 4252 by next card
IF(IFLBEG.EQ.0) IFLBEG=(INT(3*RNDM(0))+1)*(-1)**(INT(2*RNDM(1))+1)

In a previous Erratum the following correction was also made:

***Insert next card after card 1691
IF(KEYBRE.GE.2) NQUA=0

(this is immediately before the only call to subroutine HARD). Lack of this instruction caused errors only when QCD bremsstrahlung was suppressed by the user (i.e. for options KEYBRE = 2 or 3) and did not affect the normal operating mode of the program.

```

*****
*
*  CORRECTIONS FOR WIZJET VERSION 2.4 -- 1 SEP 84  *
*
*****

```

(Card numbers refer to listings in CERN-TH-3761 (1983))

***Replace card 12 by next card

C- VERSION 2.4 -- 1 SEP 84

***Replace card 22 by next card

C- LONG WRITE UP - COMP.PHYS.COMM. 32 (1984) 173 (CERN-TH-3761 (1984))

***Insert next card after card 199

COMMON/KFACT/FACTK

***Insert next 2 cards after card 200

C-- K FACTOR (FROM UA1)

DATA FACTK/2./

***Replace card 670 by next card

22 FORMAT(T2;47(1H*),26H VERSION 2.4 -- 1 SEP 84 ,47(1H*))

***Replace card 1274 by next card

IX=INT(NBIN*RNDM(1))+1

***Replace card 2248 by next card

IFLAQ=INT(NFLAVS*RNDM(0))+1

***Replace card 3630 by next card

MX=INT(RNDM(0)*NB)+1

***Insert next 4 cards after card 4030

IF(IP.LT.15.OR.(IP.GE.51.AND.IP.LE.60).OR.IP.GE.91) GO TO 31

IF((-1.)*IP.GT.0..AND.JJ.GT.0) JJ=JCNPAR(JJ)

IF((-1.)*IP.GT.0..AND.JJ.LT.0) JJ=-JCNQQB(-JJ)

31 CONTINUE

***Replace card 4069 by next card

IF(IFLBEG.EQ.0) IFLBEG=(INT(3*RNDM(0))+1)*(-1)**(INT(2*RNDM(1))+1)

N.B. Corrections referring to cards 199, 200, 2248, 3630, 4030 and 4069 appear in routines also used by COJETS. They are already included in the package of corrections for COJETS.

ERRATUM

Ref.TH.3760-CERN

R. Odorico

COJETS: A MONTE CARLO PROGRAM SIMULATING QCD IN HADRONIC
PRODUCTION OF JETS AND HEAVY FLAVORS WITH INCLUSION OF
INITIAL QCD BREMSSTRAHLUNG

Immediately before the only call to subroutine HARD in the program (i.e. between lines 1691 and 1692 of the published listing) the following line should be inserted:

IF(KEYBRE.GE.2) NQUA=0

Lack of this instruction causes errors only when QCD bremsstrahlung is suppressed by the user (i.e. for the options KEYBRE = 2 or 3). It does not affect the normal operating mode of the program.

Ref.TH.3760-CERN

COJETS: A MONTE CARLO PROGRAM SIMULATING QCD IN HADRONIC
PRODUCTION OF JETS AND HEAVY FLAVORS WITH INCLUSION OF
INITIAL QCD BREMSSTRAHLUNG

R. Odorico*
CERN -- Geneva

Submitted to Computer Physics Communications

* Permanent address: Dipartimento di Fisica dell'Universita', via Irnerio 46,
40126 Bologna, Italy, and INFN, Sezione di Bologna.

PROGRAM SUMMARY

TITLE OF PROGRAM: COJETS

CATALOGUE NUMBER:

PROGRAM OBTAINABLE FROM: CPC Program Library, Queen's University of Belfast, N. Ireland.

COMPUTER: CDC 7600, IBM 3081 and VAX 11/780. INSTALLATION: CERN, Geneva.

OPERATING SYSTEM: NOS/BE (CDC), MVS (IBM), VMS/11 (VAX).

PROGRAMMING LANGUAGE USED: FORTRAN 77, also compatible with FTN 5 on CDC and FORTRAN H on IBM.

HIGH SPEED STORAGE REQUIRED: 113K without user's routines, 150K for the test-run set-up. Can be varied according to the choice of parameters.

NO. OF BITS IN A WORD: 64 (also compatible with 32).

OVERLAY STRUCTURE: none.

OTHER PERIPHERALS USED: line printer.

NO. OF CARDS IN COMBINED PROGRAM AND TEST DECK: 5137.

CARD PUNCHING CODE: EBCDIC.

KEYWORDS: Monte Carlo simulation, Quantum Chromodynamics, hadronic high energy collisions, physics at hadron colliders.

NATURE OF PHYSICAL PROBLEM

Simulation of Quantum Chromodynamics (QCD) in hadronic production of jets and heavy flavor particles, with inclusion of initial and final QCD bremsstrahlung.

METHOD OF SOLUTION

Monte Carlo generation of hard process and of initial and final QCD emission of quanta.

TYPICAL RUNNING TIME

In antiproton-proton collisions at a c.m. energy of 540 GeV, with final state calculated at the particle level, and no selection on hard events (present test-run set-up), ~ 12 events per c.p.u. second of CDC 7600. Forcing production of hard events, and with final state calculated at the particle level, ~ 9 events per second. Forcing production of hard events, and with final state calculated only at the level of quanta (jets), ~ 370 events per second.

UNUSUAL FEATURES OF THE PROGRAM: none.

REFERENCES

- [1] R. Odorico, Nucl. Phys. B228 (1983) 381, CERN preprint TH-3744 (1983).

1. INTRODUCTION

Hadronic physics at large p_t at the CERN and Fermilab antiproton-proton colliders, as well as at future multi-TeV facilities presently under discussion (see e.g. the Proceedings of the 1982 Snowmass meeting [1]), is characterized by a high degree of complexity when it comes to work out the expectations of the present underlying theory. Although its basic features can be simply understood in terms of the hard scattering of two Quantum Chromodynamics (QCD) quanta (or partons), and the recent clean evidence from the UA1 and UA2 experiments for narrow jets of particles [2,3] to be associated with the fragmentation of such quanta provides dramatic support for this picture, one expects from QCD also other effects which are quantitatively sizeable and offer valuable testing ground for the theory. These are associated with QCD radiation of quanta off the initial and final partons in the hard scattering, which is a phenomenon similar to QED bremsstrahlung but much more important in magnitude because of the larger effective coupling constant in QCD. The most obvious manifestations of QCD bremsstrahlung are the well known scaling violations of the parton densities inside the colliding hadrons, the appearance of further jets besides the two leading ones, and the presence of extra (diffuse) excitation in transverse energy observable in hard events. The latest data from the UA1 and UA2 experiments give ample evidence for these effects, namely the existence of three-jet events and the increased yield of transverse energy, which are found to be in good agreement with theoretical expectations [4].

As is well known, perturbative QCD theory is not directly comparable with the experimental data, since the QCD partons must eventually convert into hadrons, and there is at present no model-independent way to calculate this stage of the dynamics. One can in part go around this difficulty by using proper jet reconstruction algorithms and identifying then the momenta of the reconstructed jets with those of the originating partons (see e.g. [5]), or by using calorimetric "window" techniques. But a quantitative assessment of the effects of parton fragmentation, even if only for the sake of estimating needed corrections, demands the inclusion of some description of the parton fragmentation process in a final comparison with the data. Besides hard parton scattering, QCD bremsstrahlung and parton fragmentation, one should not forget the part of the final state contributed by the so-called spectator, or beam, jets, i.e. the soft jets originated by the partons inside the colliding hadrons which do not participate in the hard interaction and which produce a substantial yield of small p_t particles.

Neglect of a sufficiently realistic description of such soft jets was at the root [6,7] of the original misinterpretation of the first calorimetric data with large angular acceptance, the so-called NA5 puzzle [8]. As discussed at length in [7,9], in fact, the diffuse transverse energy generated by the spectator jets can easily swamp the angularly concentrated transverse energy signal associated with the hard jets, if some elementary precautions are not taken.

Therefore, a proper comparison of perturbative QCD with the collider jet data requires a multi-component calculation, at least at the present stage of the art, i.e. with the present approximation techniques. The complexity of such calculations may offer a weak side to a certain type of criticism, according to which the complexity itself and some extra model-dependent assumptions, which are needed, blur the significance of such calculations for the sake of QCD testing. Of course, we are the first to acknowledge that the more direct is a test, the more conclusive is its outcome. It should be appreciated, though, that with a small amount of ingenuity the various components of these type of calculations can be, to a considerable extent, tested separately, with the other components playing just a corrective role. The parton fragmentation model, for instance, can be tested by itself (apart from jet broadening due to final bremsstrahlung) by looking at jet profiles, with the initial bremsstrahlung and spectator jets contributions essentially affecting only the plateau underneath the jets. Initial and final bremsstrahlung effects can be largely isolated by looking at the transverse energy components orthogonal to the hard scattering plane, and to the residual total transverse energy after subtraction of the leading jets contribution to it or, to be more precise, to the excess of these quantities with respect to the values they attain in a calculation neglecting QCD radiation [4]. The spectator jet model can be fixed, independently of the rest, by constraining it to reproduce the characteristics of the so-called minimum bias events, i.e. events with extremely loose trigger conditions, in which the relative frequency of jetty events is very low.

QCD testing, anyway, does not provide the only motivation for this type of calculations. Proper simulation of hadronic events is needed to estimate backgrounds to other signals of interest. A typical example is the production of heavy flavored jets (i.e. carrying charm, bottom and, possibly, top quantum numbers) which through semileptonic decays of the heavy flavor particles they originate may contaminate other large p_t lepton signals (see e.g. [10]).

Keeping in mind the potentially wide range of applications of this type of calculations, in QCD testing and event simulation, we have taken care in the program that is presented here to organize the various components in a way which can be easily disassembled, with the possibility of optional modifications or even replacements by the user. The hard jet fragmentation routines and the spectator jet routines can have their working modified by acting on them externally, through parameters in common blocks. For hard jet fragmentation routines, the longitudinal fragmentation functions, determining the longitudinal spectrum and the hadron multiplicity, and the transverse momentum distribution can be altered by simply changing two parameters. Similarly, in the spectator jets routines the longitudinal and transverse momentum distributions of leading nucleons, and the mean multiplicity and p_t distribution of the other soft hadrons can be modified by correspondingly changing four parameters. Also the shape of the (scaling) multiplicity distribution can be easily modified by the user. If wishing so, the user can also supply his hard jet fragmentation and/or spectator jet routines using the program only as a generator of the final state at the level of QCD quanta. QCD bremsstrahlung can be turned off to various degrees, so as to study its effects. The user has also easy access to critical inputs, like parton density functions and particle data (in particular decay branching fractions) and can easily modify them.

An aspect which cannot be forgotten in this type of physics is the extreme rarity, in absolute terms, of some specific events which one may be interested to study separately. For instance, production of jets with very high transverse energies, production of heavy flavor particles, and especially heavy flavor particles undergoing semileptonic decay. Having just a simulation program generating events with the frequencies with which nature likes them to occur would, of course, make one quite uncomfortable in this type of applications. To cope with that, a set of forcing options is provided, with the user in full control of the possible biases which may arise from using them.

The program COJETS for hadronic production of jets and heavy flavors presented here goes in pair with the program WIZJET for the production of W and Z weak bosons and the Drell-Yan continuum [11]. Most of the routines of COJETS are also used by WIZJET, which in addition has other routines of its own. Apart from the main programs and the user's routines, one can arrange all the other routines of the two programs in a single library, compiled once and for all. It has been preferred to present

them distinctly, so as to provide the user with set-ups ready for computer running in typical conditions.

Computer programs simulating QCD in hadronic interactions have already been presented, in various forms of availability. In particular, we mention the ISAJET program [12], documented in BNL reports, and the program of [13], for which a number of results have been published, with documentation limited to a description of the physical assumptions going into the program. It is worthwhile to assess the differences between the program presented here, COJETS, and those of [12,13]. ISAJET is quite similar to COJETS. In particular, partons also there are fragmented independently according to the Field-Feynman model [14], and QCD bremsstrahlung off the final hard scattered partons is taken care of. Initial QCD bremsstrahlung is not included, however, apart from longitudinal scaling violation effects in the initial parton densities. The importance of initial QCD bremsstrahlung in accounting for a number of experimental effects and for the examination of potential problems one may find in event triggering has been underlined in the work of [13] and by this author [7,9]. Its inclusion in the calculation must overcome non-trivial technical difficulties, as it will be discussed in Sect. 4. The program of [13], as well as the present program, have been largely motivated by the need of including it. As far as it has been possible to make a comparison, the results of these two programs concerning initial QCD bremsstrahlung agree with each other at the parton level [4]. In both cases one finds an extra contribution from it to the total parton transverse energy of the order of 20%. The program of [13], however, uses a scheme for parton fragmentation completely different from the one used here and in ISAJET. It is based, in fact, on a rearrangement of the final partons in color neutral clusters before fragmentation. As discussed above, the parton fragmentation scheme represents a model-dependent part of the calculation, and choices about it can be exclusively assessed by a quantitative comparison with the experimental data. Of course, especially relevant in this respect are data at very high energy where subasymptotic effects, which the perturbative QCD approach is intrinsically unsuited to cope with, go away. An extensive comparison of the results of this program with the existing CERN $\bar{p}p$ collider data has been made [9,4], and we think the agreement is satisfactory. So far, results from the program of [13] have been compared only with fixed target data at much lower energies. It will certainly be interesting to have a comparison of its results with the collider data, as well.

2. PHYSICAL ASSUMPTIONS

A detailed discussion of the underlying physics and the way it is treated in the calculation can be found in [9]. We report here only the main features, essentially for the sake of quantitatively defining the calculation so as to make its results reproducible.

HARD PARTON PROCESS. The treatment of the parton subprocess, hard scattering or fusion into heavy quark pairs, is the standard one. The corresponding matrix elements are those of [15] (also reported in the Appendix of [9]) and can be read, with obvious notations, from subroutine DSDT, for light parton scattering, and from the function subprograms DSHVQ and DSHVG for quark-antiquark and gluon-gluon fusion, respectively, into heavy quark pairs. These matrix elements are multiplied by the appropriate factor $\alpha_s^2(Q^2) \cdot \pi / \text{SHARD}^2$, where SHARD is the c.m. energy squared of the parton subprocess, in the calling routines. We take $\alpha_s(Q^2) = 1/(\text{BALPH} \cdot \ln(Q^2/\Lambda^2))$, where $\text{BALPH} = (33 - 2 \cdot \text{NFLAVS}) / (12 \cdot \pi)$, and NFLAVS, number of flavors considered in the initial state, is equal to 3. We have taken $Q^2 = 2stu/(s^2 + t^2 + u^2)$ for the hard scale, as proposed in [16], where s , t and u are here the Mandelstam invariants of the parton subprocess.

Heavy flavor production via flavor excitation (i.e. $q+Q \rightarrow q+Q$, $G+Q \rightarrow G+Q$) is not included in the calculation. Certainly not because it represents a negligible contribution (in the case of charm it dominates over fusion, especially at small transverse momenta [17]). Rather because its calculation is potentially unstable and extra experimental input is required in order to make it sufficiently stable. Such experimental input is available in the case of charm, but not in the case of bottom and top. Because of such instability and of the major technical complications that its inclusion would require (one should include heavy flavors in the initial parton evolution, with the associated kinematic constraints appropriate to massive quarks [17]) we have preferred to leave it out of the present program. Its contribution can obviously be added by the user by making a separate calculation.

PARTON DENSITIES. The initial quark parton densities (before evolution) are those determined from the massive muon pair production data of the NA3 experiment [18]. For the gluon distribution a form $xG(x) \propto (1-x)^5$ is assumed. Justification for this form comes, besides other considerations,

from the good quality of the fit it gives to the relevant UA1 and UA2 data [9]. Quark parton densities are defined, with obvious notations, in the function subprogram F2IN, and the gluon density in FGLU. The expressions given there are for the densities times x , the longitudinal momentum fraction of the parton. These expressions can be changed by the user. The initial evolution scale is set at $Q_{ZEV} = 4$ GeV, and can also be changed by the user (see Sect. 5). The initial partons (before evolution) are given a primordial transverse momentum according to a gaussian distribution with $\langle p_t^2 \rangle = PT2INT = 0.4 \text{ (GeV/c)}^2$, which is found to reproduce, together with the parton transverse momentum generated in the evolution, the existing transverse momentum data for Drell-Yan pairs [19].

INITIAL EVOLUTION CASCADES. The calculation of initial parton cascades proceeds identically to the way in which the evolution of parton densities is calculated in [20]. The program for the parton density evolution has been published [21]. The calculation, of course, reproduces the evolution of structure function moments with any wanted accuracy (the results of a numerical test are reported in [21]). The infrared cutoff needed in the computation does not have the quality of a physical cutoff, but is just a parameter which can be reduced at will according to the precision required. Expressions giving the value of this parameter as a function of the requested precision, and the corresponding increase/decrease of the execution time are given in [20,21]*.

In this program the calculation of the evolution of parton densities is supplemented by the calculation of the transverse momentum developed at each parton branching, and by the recording of the kinematic characteristics of the timelike parton generated at each radiation branching (which, differently from the partner spacelike parton, does not participate in the interaction leading to the hard scattering or fusion process). As is well known, in a spacelike QCD cascade the enforcement of kinematic bounds is somewhat ambiguous because of the lack of strong ordering. Because of that, the transverse momentum generated in a branching cannot be determined from exact kinematics, as in the case of timelike cascades. We take for its square $K_T^2 = z(1-z) |Q^2|$, where Q^2 is the value of the evolution scale at the branching, and z is the parent momentum fraction carried by one of the secondaries. The azimuth of the transverse momentum is uniformly

* The infrared cutoff on the splitting variable z is given in the program by ZMCUT/ECM, and can be changed by the user (see Table 3).

generated at random. The timelike secondary is further evolved with an initial virtuality $z|Q^2|$.

The longitudinal scaling variable x is identified with the fraction of the light-cone variable $E+P_L$ of the hadron of origin, which makes the variable longitudinally Lorentz invariant. Since also the transverse momentum K_T of the parton is known, one can determine the corresponding fraction of the hadron $E-P_L$ carried by the parton, and hence its energy and longitudinal momentum, separately. For a spacelike parton, neglecting its virtual mass, one obtains that when $K_T^2/(xs)$ (where s is here the hadron c.m. energy squared) is larger than x , the parton moves backwards with respect to the hadron of origin. Although it is rather rare for that to occur, and even more rare (although not kinematically impossible) that with such a parton one can pass the energy threshold condition for an interaction, we have explicitly excluded, on the grounds of physical sensibility, such partons from interacting.

The evolution scale is set by the variable $Q^2 = 2stu/(s^2+t^2+u^2)$, where the Mandelstam invariants are here referred to the parton hard process. This variable is close to P_T^2 , where P_T (PTHARD in the program) is the transverse momentum generated in the hard process [16].

TIMELIKE EVOLUTION CASCADES. The hard scattered partons and the timelike partons emitted in the initial cascades are made to QCD evolve according to the same calculational scheme developed for partons in electron-positron annihilation [22]. Partons originated from the initial cascades are assigned an initial virtuality as stated above. The two partons arising from the hard process are given an initial virtuality (i.e. invariant mass squared) $Q^2/4 \sim P_T^2/4$, where Q^2 is again the variable of [16]. As is well known, the evolution scale is uncertain in the leading pole approximation. We have taken the above value in order to make sure that the invariant masses of the two partons stay within kinematic bounds. (The user can change them by simply replacing in the two calls to PSQGEN made in subroutine HARD the quantity $0.75*QSQ$ with the wanted values, QSQ corresponding in the program to Q^2). Timelike cascades are stopped when partons reach a minimum virtual mass. The cutoff mass (QZ in the program) cannot be fixed theoretically, and must be induced from the experimental data. A value of $QZ = 3$ GeV gives an adequate fit to electron-positron annihilation data [22], and we assume this value in the program. For heavy flavored partons we add to the cutoff the values of the heavy quark

masses (as stored in QZMASS). Initial bremsstrahlung quanta not surviving the cutoff are not fragmented into Field-Feynman jets, and also are not included in the array PARQUA, storing information on final quanta. They are freely emitted, however, in the initial cascades (as indicated above) not to alter the evolution of spacelike partons. Transverse effects associated with initial bremsstrahlung quanta not surviving the cutoff are very small. Anyway, if the user wants to have them included in PARQUA, it is sufficient to remove the cutoff condition for them in subroutine JTBOOK (IF(E.LT.QZ) RETURN).

HARD JET FRAGMENTATION. The Field-Feynman model [14] is assumed for the conversion of partons into jets of hadrons. We use for this the FORTRAN program of [23], with some modifications. Stable particles moving backwards with respect to the jet direction are eliminated. Exception is made for decay secondaries of heavy flavor particles, which are always kept. Gluons are fragmented as light quarks of random flavor. If one likes to modify the characteristics of gluon jets, this can be easily done, as explained in Sect. 7. The longitudinal $E+P_L$ fraction carried out by the quark at each decay step is given by $D(x)dx \propto (1-CX^2+3CX^2(1-x)^2)dx$, with $CX^2 = 0.77$. The fragmentation functions of heavy quarks into heavy hadrons are taken from [24]: $D_h(x)dx = (1/(x(1-1/x-EPSPET*(M_c/M_h)^2/(1-x)^2)))dx$, where $EPSPET = 0.25$, M_c is the charm quark mass and M_h is the mass of the heavy quark (as stored in QMAS). Quarks are generated with a transverse momentum distribution $\exp(-p_t^2/(2*SIGMA^2))dp_t^2$, with $SIGMA = 600$ MeV. Particle masses and decay properties can be printed together with the values of the above parameters by calling PARTAB. All parameters can be easily altered by the user as indicated in Sect. 5. The model for the decay of heavy flavor hadrons coded in the package of [23] is that of [25].

SPECTATOR JETS. The model for the spectator jets is a simple longitudinal phase-space model including leading baryon effects. The model is constrained to (reasonably) reproduce the existing UA1 and UA5 data at small p_t . Specifically: rapidity, transverse momentum and multiplicity distributions, and particle production ratios. Leading baryons are generated with an $\exp(-CLPRO*X_F)dX_F$ Feynman X_F distribution with $CLPRO = 0.7$, and an $\exp(-p_t^2/PT2PRO)dp_t^2$ transverse momentum distribution with $PT2PRO = 0.15$ (GeV/c)², as measured at the ISR [26]. Transverse momenta of the other particles are generated with an $\exp(-b*(p_t^2+m^2)^{1/2})dp_t^2$ distribution, where m is the particle mass. The

parameter b is made to decrease with multiplicity, so as to reproduce the growth of $\langle p_t \rangle$ with charged multiplicity, as observed in the UA1 experiment [9]. For the multiplicity distribution we assume the KNO scaling parametrization of de Groot [27], which gives an adequate fit to the UA1 data [9]. Particle ratios are those measured by the UA5 experiment [28] (they are listed by PARTAB). The mean total multiplicity (charged+neutrals) is arranged so as to reproduce the mean charged multiplicity measured by UA1 [9]. It varies with the invariant mass W of the residual system, which is left out after hard jet hadrons and leading baryons of the spectator jets have been taken out. Calling N_R the mean total multiplicity of the residual system, we use $N_R = 2.2*[2.50+0.28*\ln(WN)+0.53*\ln^2(WN)]$, where $WN = 1.6 W$ (GeV). The expression in square bracket is obtained in [29] from a fit to experimental data for the mean charged multiplicity vs. c.m. energy.

3. GENERAL FEATURES OF THE SOFTWARE

The organization of the software approximately follows the breakdown in physical components described in Sect. 2.

There are no input cards. The main program acts as a steering program. The basic calls in it are to PREGEN, which handles preparatory work before the event generation, and the calls, within a loop, to GEVENT which produce the generation of events. At each event generations a call is made to the user's routine FILLH, where the necessary event recording can be done via a set of common blocks carrying event information. After event generation PRHEAD is called to print basic data about the event generation run. Optional features of the main program include a call to user's routines BOOKH, at the very beginning of execution, and OUTH, at the end of execution, where the user can perform desired operations before (typically, histogram booking) and after (typically, histogram printing) event generation, respectively. The user may also call PARTAB, to have a listing of the particle table and of other hadronization parameters. In the main program, before the call to PREGEN, the user may also override default values of input parameters (via common blocks), and set forcing and freezing conditions (parameters PTMGE and IPTWGT for forcing of parton transverse momentum generation, call to FUSION to force heavy flavor production via fusion, call to FREEZE to freeze jet fragmentation or heavy flavor particle decays with the option of letting them develop under user's control in FILLH).

The rest of the code is broadly organized in sections, for convenience of reference. We have not enforced strict modularity. Some sections, though, (especially those handling hadronization) can easily be replaced by the user, following the directions of Sect. 7. Except for special modifications, all the routines, apart from the main program and the user's routines BOOKH, FILLH and OUTH, can be compiled once and for all and organized in a library.

Section A contains the BLOCK DATA, a routine SBLOCK which executes manipulations of the input data, and the parton density functions. The user is supposed to have easy access to this part of the program and can make desired modifications, if needed.

Section B handles the production dynamics: initial QCD evolution and the hard parton process. It also contains the subroutine PRHEAD, described above, and the subroutine OVERDM, which takes control if arrays containing quanta and particle information are going to be overflowed.

Section C handles the timelike evolution of quanta emitted by initial bremsstrahlung or resulting from the hard parton process.

Section D deals with hadronization. It also contains subroutine PARTAB, described above, and subroutine DUMPEV, which can be called from FILLH to have the dump of events (each call producing the dump of the current event). Subsection D1 contains Sjostrand's package for hard jet fragmentation. (N.B. the Sjostrand package uses MeV and not GeV units).

Section E contains forcing and freezing routines to be optionally called by the user.

Section F contains routine TIMELF, giving the remaining time (in seconds) left for execution, and routine RNDM, which is the uniform random number generator. The user should modify these routines according to local implementations.

Section G contains the user's routines BOOKH, FILLH and OUTH, whose functions have been described above. The user can change these routines at will, respecting only their external characteristics. At each event generation information about the final state is stored in a series of

common blocks (/EVENT/ for particles, /PARQUA/ for quanta, and others specifying the internal structure of the event) which are supposed to be accessed from subroutine FILLH.

The program is entirely written in FORTRAN code. It is compatible with FTN4 and FTN5 compilers on CDC machines, with FORTRAN H and FORTRAN 77 compilers on IBM, and with FORTRAN 77 on VAX computers*.

4. SPECIFIC FEATURES OF THE SOFTWARE

A novel feature of the program concerns the treatment of the hard parton process in the presence of initial QCD bremsstrahlung. Technically, the treatment of the associated dynamics presents non-trivial difficulties, especially if one wishes to maintain good standards of Monte Carlo efficiency and storage utilization. The generation of the initial cascades demands knowledge of the evolution scale, which is set by the transverse momentum P_T generated in the parton process (more precisely, the Field-Feynman-Fox variable Q^2 defined in Sect. 2). On the other hand, the generation of the hard process must await generation of the initial cascades, which define the kinematic characteristics of the (spacelike) partons entering the hard interaction.

In order to break this vicious circle, before event generation the dynamical probability for having a given Q^2 is determined. This is achieved by generating a large number of pairs of initial cascades (one per each colliding hadron) driven by Q^2 , and appending to each pair of interacting spacelike partons belonging to the two cascades a weight given by $d\sigma(s,t)/dt$, where s and t are the Mandelstam invariants of the parton subprocess, which are determined by Q^2 and the kinematic characteristics (longitudinal and transverse momenta) of the two interacting partons (two values of t correspond to a given s and Q^2 , and both contribute). This is done for a set of sufficiently dense values of Q^2 . More precisely, pairs of initial partons are sampled from the parton densities, and an evolution

* N.B. With the VAX FORTRAN 77 compiler hollerith arguments in subroutine calls should be given via real variables defined in data statements. Such types of calls are (optionally) made only in the main program and in the user's routines.

sweep in Q^2 is made each time. By forced hardening of the parton densities (with associated corrective weights) the procedure turns out to be quite fast and statistically efficient. Also storage requirements are limited (as can be seen from the statistics giving the filling of PARACT printed at the end of execution), since partons which at a given Q^2 are too soft to undergo any interaction can be eliminated from consideration at higher values of Q^2 .

In this way one obtains the probability distribution of Q^2 , $P(Q^2)$. A possible approach to (unweighted) event generation can then be the following: i) sampling Q^2 on $P(Q^2)$, ii) sampling of two initial partons on the parton densities, iii) development of the two initial cascades up to Q^2 , iv) recording of all pairs of (spacelike) partons having subenergy s high enough to pass threshold at the given Q^2 , and calculation for each of them of $d\sigma(s,t)/dt$ (t is determined by s and Q^2 with a two-fold ambiguity, symmetric with respect to u - t interchange -- the two possibilities, both recorded in the pretabulation stage, are to be considered as distinct and possible interactions), v) random choosing of one interaction according to the probabilities given by the $d\sigma(s,t)/dt$ values. If no pair has enough subenergy to give an interaction, one must repeat the procedure at the same Q^2 , until one event is generated.

This approach is certainly simple and safe, but has two drawbacks. Before making the final choice of the interacting pair one must keep in storage the whole information about the two (treelike) cascades, including this time also the emitted (timelike) partons, and without the possibility of early removal of soft spacelike partons which was open in the pretabulation stage. Especially at intermediate values of Q^2 , the multiplicity of possibly interacting pairs may become sizeable. By picking up only one of them, and throwing away the rest, all the execution time gone in the calculation of the cascades and of the $d\sigma(s,t)/dt$ values is used inefficiently.

In order to profit at each Q^2 generation of all the potentially interactive pairs, one can follow a different course of action. In the pretabulation stage, one can determine, besides $P(Q^2)$, also the mean multiplicity of interactive pairs, $N(Q^2)$, (excluding from the mean entries with no interacting pairs) and the mean $d\sigma(s,t)/dt$, $S(Q^2)$. One can then sample Q^2 not on $P(Q^2)$, but on $P(Q^2)/N(Q^2)$. In the event generation, one can now use all the pairs, taking due care of the variance for each of them of $d\sigma(s,t)/dt$ from $S(Q^2)$. This variance depends not only on s , but also on the specific parton channel in which the interaction occurs. Without weight-

ing the events, one can correct for such a variance by well-known procedures. That is, appropriate random rejection of pairs having $d\sigma(s,t)/dt$ lower than $S(Q^2)$, appropriate repetition of pairs having $d\sigma(s,t)/dt$ higher than $S(Q^2)$. Of course, if the variance is too high such a method becomes statistically unsafe. The point is that in hard parton scattering the important channels are dominated by gluon exchange. This implies, for not extreme values of t (with respect to s) and for not too small s , that $d\sigma(s,t)/dt$ is approximately independent of s in such channels. Besides allowing for a fuller exploitation of the execution time, one can in this way also considerably reduce the storage requirements for the two initial cascades. In fact, since each parton pair that can undergo interaction does actually give rise to an interaction, one does not have to store simultaneously all such pairs and the associated information. One can develop the initial cascade trees one segment at the time, deal with a segment for possible interactions, and then get rid of it. To do that, one must, for each segment of the first tree, fully develop by segments and then discard the second tree (starting again from the same initial parton). This of course requires some extra computing work, but in this way storage requirements at this stage can be kept at a very reasonable level even at very high energy. By freezing the final state at the level of quanta, one can in fact run the program with the present array dimensions up to a c.m. energy of 50 TeV and beyond. Eventually, it has been the considerable gain in storage that has convinced us to follow this second approach in the design of the program. With a minimum amount of effort, though, the program can be converted to the simpler, but more resource demanding, type of working of the first approach.

The algorithm for the evolution of timelike partons has been discussed in detail in [22].

The hard jet fragmentation routines have been slightly modified with respect to those presented in [23]. The heavy flavor fragmentation functions have been changed, using those of [24]. Decay products of heavy flavor particles are now always kept, even if backward moving with respect to the jet direction. The coding of the hadron origin ($K(I,1)$ in /JET/) has been changed, to allow reconstruction of heavy flavor decay cascades. A check is made whether overflows of the particle arrays (K and P in /JET/) are going to occur. (N.B. the Sjostrand package uses MeV instead of GeV units).

The spectator jets routines correspond to a more or less direct coding of the model presented in Sect. 2. Since it is an independent emission model, at the end a corrective algorithm is implemented to account for exact 4-momentum conservation (subroutine P4CORR). The correction is made by acting only on the particle system generated in the subroutine KNO, that is on the system left over after the hard fragmentation particles and the leading nucleons are taken out.

5. USAGE

There are no input cards. The program acts as an event generator. The user is supposed to steer the event generation from the main program, and to make his event recording in subroutine FILLH, which is called every time an event is generated.

MAIN PROGRAM. (See Table 1). The basic structure of the main program, not to be altered by the user, consists of a call to PREGEN, for the pretabulation, a loop making calls to GEVENT, for the event generation, and a call to PRHEAD, for the printing of the main physical input parameters, user's options, basic event generation data, and statistics about dimension utilization of arrays carrying particle and quanta information. One should be aware that a call to GEVENT may produce the generation of several events, and therefore the user should not do his event recording using the output common blocks in the main program, but exclusively in FILLH. The loop on GEVENT can be controlled as the user prefers.

In the test set-up, in which the user will find the program arranged, the main program makes also calls to BOOKH, before event generation, and to OUTH, after event generation. These are user's routines, and are called only in the main program. They can arbitrarily modified, and also eliminated.

Optional calls in the main program can be made to FUSION(H) and FREEZE(H), before the call to PREGEN.

A call to FUSION(H) drives the exclusive generation of events in which heavy flavors are produced via the fusion mechanism. H is a hollerith variable which should be set equal to 1HC, 1HB or 1HT (or 'C', 'B' and

'T', depending on the FORTRAN compiler) according to the heavy flavor desired, i.e. charm, bottom and top, respectively*.

A call to FREEZE(H) allows for the freezing of the final state at various levels. With H = 1HQ, the final state is frozen at the level of quanta, and no hadronization is carried out. With H = 1HC, 1HB or 1HT, the decay of hadrons with the corresponding heavy flavor is not carried out, and one finds such hadrons as stable particles in the final state. The fragmentation of quanta and the decay of heavy flavored hadrons can be subsequently driven from FILLH, under user's control and with the optional imposition of forcing conditions (typically, forced semileptonic decay), as indicated below.

In the main program the user can also override, before the calls to FREEZE and PREGEN, default values of input data set in BLOCK DATA or in subroutine SBLOCK. Of course, for this purpose the appropriate common blocks must be present in the main program. Setting IDECAY(N) = 0, where N is a particle code number (listed in the particle table printed by calling PARTAB) sets the particle stable even if in the particle table it is listed as unstable.

MAIN OUTPUT COMMON BLOCKS. (Table 2). Particle information in the final state is stored in /EVENT/. Information about the final quanta (or jets) is stored in /PARQUA/. /PTHARD/ gives the value of the parton transverse momentum generated in the hard scattering. The distribution in PTHARD should be watched when applying forcing conditions on the transverse momentum generation via the parameter PTMGE (see below). Details are reported in Table 2.

COMMON BLOCK /HADQN/. The user may find useful to have handy information about some particle quantum numbers stored in this common block. The arrays LHADR(N), LCHARG(N), LCHARM(N), LBOTT(N) AND LTOP(N) give, for the particle of code number N, the hadron-lepton quality (1 or 0, respectively), the electric charge, the charm q.n., bottom q.n. and top q.n., respectively (± 1 or 0 in all cases).

* For VAX, see note in Sect. 3.

OTHER OUTPUT COMMON BLOCKS. (Table 2). Information about the partons originating the initial QCD cascades (i.e. those sampled on parton densities) is stored in /ENTREV/, together with the value of the F.F.F. variable Q^2 in the event. Information about the two (spacelike) partons undergoing the hard interaction and the timelike partons emitted in the initial cascades is stored in /PARQ/, together with the value of the sub-energy of the two interacting partons. The value of the 4-momentum and invariant mass of the spectator jets system is stored in /PMBIAS/.

CHANGING DIMENSIONS OF PARTICLE AND QUANTA ARRAYS. The utilization of array dimensions containing information about quanta and particles varies with energy and with other input parameters. The maximum utilization of such dimensions in a run is reported in the standard printout made by PRHEAD. The user can change the current dimensions, simultaneously for all the arrays listed on the same line of the PRHEAD printout and together with the control parameter also indicated on the same line. The dimension to change, and for which the maximum utilization is shown, is always the first dimension of the array.

MONITORING COMMON BLOCKS. (Table 2). /SPYFOR/ contains parameters reporting current freezing and forcing conditions (see below). /OVER/ monitors occurrences in which particle or quanta to be generated in events would overflow the dimensions of the corresponding arrays (see below). /MAXN/ reports statistics about maximum utilization of such arrays.

MAIN INPUT PARAMETERS. (Table 3). Current default values of input parameters correspond to the generation of unweighted (i.e. WEIGHT = 1) events for antiproton-proton collisions (IPBAR = 1) at c.m. energy ECM = 540. GeV, QCD Λ (ALAMB) = 0.1 GeV, mean primordial transverse momentum squared of partons PT2INT = 0.4 GeV/c, and with an assumed total inelastic cross-section SIGINL = 40. mb (which determines relative frequencies of QCD events with jets, and events consisting only of spectator jets). The parton transverse momentum cutoff (actually the square root of the F.F.F. Q^2 variable) is set at PTMIN = 4. GeV/c. This value is chosen because giving a QCD cross-section sufficiently smaller than the total inelastic cross-section, as required by the applicability of perturbative QCD. At higher energies (or higher Λ) it may turn out that PTMIN should be increased for this reason. If the QCD cross-section (printed by PRHEAD) is higher than 0.3*SIGINL the program gives a warning message and stops. Since the present initial evolution scale is set at QZEV = 4.

GeV, PTMIN cannot be lowered below this value. The user can decrease QZEV, but then, for physical consistency, he should also modify the input parton densities (in F2IN and FGLU). The number of flavors (NFLAV) for partons arising from the hard process and final bremsstrahlung can go up to 6 (i.e. including top). The default value is 3. If the user sets it higher, it activates also fusion (with its QCD probability, among other events). If a call to FUSION is made, NFLAV is automatically increased so as to include the flavor for which FUSION is called. Input parameters, together with the common block to which they belong, are listed in Table 3.

OTHER INPUT PARAMETERS OF INTEREST. (Table 3). The exit from event generation is controlled by the parameter TLEAVE, the value (in seconds) of the time left for execution at which event generation is supposed to stop (default 5. sec). FRATAB fixes the fraction of execution time devoted to the pretabulation (default 0.1 when FUSION is not called, 0.3 when FUSION is called). The user may wish to increase FRATAB to have a more accurate determination of the QCD cross-section. NCASC determines the maximum number of cascades per parton channel computed in the pretabulation (default 5000). The program automatically calls DUMPEV for the first NDUMP events, to have event dumping (default 5). To assess the importance of QCD bremsstrahlung effects, the user may wish to inhibit QCD bremsstrahlung to various degrees by using the control parameter KEYBRE. KEYBRE = 0 corresponds to the normal situation. If 1, transverse momenta generated in the bremsstrahlung are neglected (i.e. they are set to zero). If 2, only longitudinal scaling violation effects on parton densities are considered. If 3, QCD bremsstrahlung is completely turned off.

Timelike evolution of partons is controlled by the virtual mass cutoff QZ (default 3. GeV, from fits to electron-positron annihilation data). For heavy flavor partons, the cutoff is increased by adding heavy quark masses, as stored in QZMASS.

For hard jet fragmentation, important parameters are CX2, controlling the shape of the longitudinal Field-Feynman splitting function for light quarks, and EPSPET, doing the same for the heavy quark fragmentation function (ϵ parameter in Peterson et al. [24]). SIGMA (N.B. in MeV) defines the mean p_t of quarks emitted in the Field-Feynman cascade. Current values of these parameters are listed in PARTAB.

The parameters for the spectator jets have been determined essentially from fits to the CERN collider data. The user may wish to change them, especially when running the program at different energies. Mean multiplicity and mean p_t can be increased/decreased by correspondingly increasing/decreasing the parameters FMULMB and FPTMB, respectively (default 1.). The particle ratios (from UA5 [28]) are stored in RPAIR. Here π^0 and η are put together (in RPAIR(1)), with the $\eta/(\eta+\pi^0)$ ratio stored in RETPIZ. The parameters controlling leading nucleon generation are CLPRO, in $\exp(-CLPRO \cdot X_F) dX_F$ for the longitudinal distribution, and PT2PRO, for the mean transverse momentum squared (default 0.7 and 0.15 $(\text{GeV}/c)^2$, respectively, as from ISR data [26]). All current values of these parameters are listed by PARTAB.

Particle masses are stored in PMAS (N.B. in MeV), and decay branching fraction information in CBR. CBR does not directly give the branching fraction for each decay channel (as numbered by PARTAB). Data are stored there in generation table format. The branching fraction for a channel is given by the difference of the CBR for the channel and the CBR of the previous decay channel of the particle. The CBR of the first decay channel coincides with the corresponding branching ratio. The last decay channel of the particle has CBR = 1.

PARTICLE DATA TABLE AND HADRONIZATION PARAMETERS. As already indicated elsewhere, a listing of the current particle data input and of the values of the main hadronization parameters is obtained by calling PARTAB.

EVENT DUMPING. The user can call, any time he needs, subroutine DUMPEV from FILLH. He will obtain in this way a dump of parton and particle parameters in the current event.

FORCING. (Table 4). The program, as it stands, generates events according to theoretical probabilities (i.e. mimicking nature) with no type of forcing. On the other hand, simulation studies very often deal with specific types of events, relevant for a given problem. Letting the program to generate all types of events according to their dynamical probabilities and recording afterwards only those of interest represents, of course, a very inefficient way of spending computer time. Naturally, one would like to generate only, or at least mostly, the events of interest. I.e., one would like to "force" the generation of such events. In doing that, when it can

be done, one is of course at the same time "biasing" the generated event sample. In order to be acceptable, the bias must not affect relative probabilities within the total event sample of interest for the problem at hand.

The program allows for two types of forcing: generation of large p_t jets, and electron semileptonic decays of heavy flavors.

Generation of large p_t jets can be forced in two ways: with event weight, and without event weight. WEIGHTED EVENTS are obtained by distorting the Q^2 generation table by a factor $Q^{4+1.}$, and associating a corrective weight to the events. To use this option IPTWGT = 1 must be set in the main program before the call to PREGEN. The relative frequency of hard events with respect to purely spectator jets events can be altered via FRAGEH, which approximately defines the fraction of hard events (default 0.5). FRAGEH is ignored if events are not weighted (IPTWGT = 0, normal situation). The corrective weights take into account that for each Q^2 generation several events can be generated, so that the sum of all event weights in a run is within a certain approximation equal to the number of events.

Forced p_t generation of UNWEIGHTED EVENTS is obtained by generating events only above some minimum value of Q^2 . To use this option, one must set, before the call to PREGEN, PTMGE equal to the cutoff value of the square root of Q^2 (PTMGE must be greater than PTMIN to make the option active). In using this option there is the danger of unduly biasing the generated event sample. For instance, suppose we wish to generate an unbiased sample of events with transverse energy $E_T > E_T(\text{cut})$. Only part of the transverse energy is contributed by the hard parton scattering. Some of it comes from hard jet fragmentation, some from QCD bremsstrahlung, and some from the spectator jets. Therefore, there are events which survive the E_T cut even if $\sqrt{Q^2}$ is much lower than $E_T(\text{cut})/2$. Setting the cut at $\text{PTMGE} \approx E_T(\text{cut})/2$ would then deprive the generated event sample of all these contributions, i.e. the event sample would be unduly biased. PTMGE, therefore, must be set lower. How much lower must be found by looking at the distribution in $\text{PTHARD} = \sqrt{Q^2}$ (in $/\text{PTHARD}/$), for non-forced (or, more conveniently, for weighted) events satisfying the $E_T > E_T(\text{cut})$ condition. PTMGE should then be set (immediately) below the lowest PTHARD value which is populated.

FORCING OF SEMILEPTONIC DECAY of heavy flavors is particularly liable to undue bias, especially when considering multilepton events. For this reason, the user is enabled to do his own forcing by employing a set of routines which steer the fragmentation process. First, the user must freeze in the final state the decay of the heavy flavor particles under study, by calling FREEZE (one, or more times, if needed) with the appropriate arguments (1HC, 1HB or 1HT) in the main program (before the call to PREGEN)*. In FILLH, then, for each event the user must pick up the order number in the event ($I=1, \text{NPART}$) of the heavy flavor particle of interest (he may find useful to employ the particle q.n. arrays in /HADQN/ to do that). By calling LOOSEH(I,N), he will have the heavy flavor particles decayed according to the branching fractions listed in the particle table. N is the total number of particles developed in the decay cascade, including both stable and unstable particles. Information on the cascade particles is stored in /JET/ (see Table 2). The cascade is listed by calling SHOCAS(N), with indication of parent-daughter relationships. The absolute value of $K(IC,1)$, for the particle of order no. IC in the cascade, gives the order no. in the cascade of the parent. $K(IC,1)$ is positive if the particle is stable, is negative or zero (for the heavy flavor particle initiating the cascade) if unstable. By calling STABKP(N) unstable particles are eliminated from /JET/, and N is correspondingly updated. In order to force electron semileptonic decay of flavor H (1HC, 1HB or 1HT) in the cascade, before the call to LOOSEH(I,N) one must call FORSLE(H,1), which correspondingly alters the decay branching fractions. By calling FORSLE(H,-1) afterwards, the original decay branching fractions for hadrons of heavy flavor H are reestablished. To know about the current semileptonic forcing conditions, the user can interrogate the spy parameters SPYCEL, SPYBEL and SPYTEL in /SPYFOR/, for charm, bottom and top, respectively, which are 1. if the forcing conditions are active, -1. if inactive. From /SPYFOR/ one can also know about freezing conditions, via the parameters SPYC, SPYB, SPYT and SPYQ for charm, bottom, top hadrons and quanta, respectively. As in the previous case, these are equal to 1. if the corresponding freezing condition is active, to -1. if it is inactive.

If one is not interested in the full hadronization of the event, forcing of (electron) semileptonic decay of heavy flavors can be done more conveniently by acting on the quanta before hadronization. For this, one should

* For VAX, see note in Sect. 3.

freeze the final state at the level of quanta, with a call to FREEZE(1HQ) in the main program, and then hadronize in FILLH the quanta of interest, identified by their order number IQ ($=1, NQUA$), by making calls to LOOSEQ(IQ,N), where N returns the number of stable particles in the jet cascade. By calling FORSLE(H,1) before the LOOSEQ call, branching fractions are changed so that cascade particle of flavor H are forced to undergo (electron) semileptonic decay. A subsequent call to FORSLE(H,-1) reestablishes the non-forcing conditions, as already indicate above. Stable particles only are kept in parton fragmentation. Information on the cascade particles is available from /JET/ (Table 2). Also in this case a call to SHOCAS(N) produces a printed list of the cascade particles.

Test set-ups 1-4, listed at the end, give illustrative examples of the use of freezing and forcing routines.

DO-IT-YOURSELF HADRONIZATION. By freezing the final state at the level of quanta and using LOOSEQ, as indicated when discussing forcing, one can hadronize one parton at the time in the final state. A final call to SPECTJ allows for the generation of the spectator jets hadrons. In doing that, SPECTJ reads the content of array PARHAD up to particle NPART, and completes the final state adding hadrons to PARHAD, and correspondingly updating NPART, so as to balance the total 4-momentum. Therefore, once acted on LOOSEQ, the user should transfer particle information from arrays K and P to PARHAD, for each quantum, and eventually call SPECTJ. However, if the user wants to avoid the transfer and use jet particle information directly from K and P, he can simply sum all jet particle 4-momenta in PARHAD(1,J), $J=1,4$, and set NPART = 1. SPECTJ will then complete the final state storing particle parameters in PARHAD(I,J), $J=1,7$, with $I=2, NPART$, NPART being now the number of particles generated by SPECTJ plus one.

By handling hadronization himself, the user may achieve considerable savings in execution time and storage, if he does not need to compute the whole final state but only certain parts of it. Also, when running at very high energies (tens of TeV), in order to keep storage requirements within reasonable limits, it may be convenient not to generate and store the whole of the hadron final state, postponing utilization of the information to a subsequent stage, but rather to generate the hadron final state one piece at the time, utilize it, and then get rid of it.

6. DIAGNOSTICS

The program prints diagnostic messages when input parameters are given inconsistent values, and gives directions to the user about the action to take.

If in the computation of the final state, the generation of particles and quanta is found to overflow declared dimensions in the relevant arrays, the program prints a warning message giving directions about which array dimensions to increase, and control is then taken over by subroutine OVERDM. According to the option chosen, two courses of actions can be taken by OVERDM. If (in /OVER/) KOVER = 1 (default), for the first 100 overflows the (uncompleted) event is ignored and execution goes on. After which, execution is terminated by deliberately producing an arithmetic error in order to activate a memory dump, according to local debug implementations. If KOVER = 0, the execution is immediately terminated with the same procedure. SPYOV in /OVER/ is normally negative. If overflowing requests have occurred during execution, it counts their number.

7. USER'S POSSIBLE MODIFICATIONS

The user may especially wish to modify or change parts of the hadronization scheme, which represents a model-dependent component in this type of calculation.

For hard jet fragmentation, if besides changing the parameters governing it (and externally accessible, see Sect. 5 on Usage, and Table 3) the user wants to make more substantial modifications, he should act on subroutine JETGEN. Input to subroutine JETGEN is: i) the parton flavor IFLBEG (in /PAR/) which can take values as indicated for PARQUA(IQ,6), ii) the jet energy EBEG (in /PAR/, N.B. in MeV), iii) the polar and azimuthal angles THETA and PHI (in /EDPAR/, in radians). As output from JETGEN, one has the number of stable particles in the jet, N, which is the only argument of JETGEN, and stable particle information in K and P (/JET/). For P the same conventions hold as for the corresponding array locations of PARHAD. K(1,1) is not used in this application, and K(1,2) gives the particle code number with the same conventions as PARHAD(1,6). One should note that, like the input parameter EBEG, also the output array P is given in MeV units. (The array P is given in GeV when output from

LOOSEH and LOOSEQ). If the user wants to change the characteristics of gluon fragmentation with respect to those of quark fragmentation, this can be simply done by altering at the very beginning of JETGEN, before any other execution statement, the longitudinal and transverse parameters CX2 and SIGMA, respectively, when IFLBEG = 0, and then by resetting them at the original quark values immediately before the normal RETURN. Otherwise, JETGEN can be replaced by another subroutine having the same input-output characteristics.

For spectator jets, if one wants to replace the current model, one has to simply substitute subroutine MNBIAS (and eliminate subroutine KNO called by it). Input to MNBIAS is the array PMBIAS, in /PMBIAS/, which contains the 4-momentum and, as fifth component, the invariant mass (GeV) of the spectator jets system. Output particles are stored in PARHAD, with NPART correspondingly incremented with the respect to the value it has before entering MNBIAS. If the user wishes just to change the multiplicity distribution, he can do so by altering the KNO scaling distribution function in the function subprogram FMULDG.

8. TEST RUN

In the present set-up the program runs with default values for the input parameters and no optional routine calls. The CERN HBOOK package for histogramming is employed in the user's routines. Some typical distributions of interest are calculated. The test-run output excerpts include the beginning of an event dump, and the tableau printed by PRHEAD (corresponding to an execution time on CDC 7600 of 300B sec). If the HBOOK package is not locally implemented, a first run of the program can be obtained by replacing the user's routines BOOKH, FILLH and OUTH with void routines (i.e. containing only a RETURN statement).

ACKNOWLEDGEMENTS

This work has been originally stimulated by C. Rubbia. The convenience of a number of user-oriented adaptations of the program has emerged from several conversations with G. Goggi and R. Kinnunen. The debugging of the final modifications of the program and its adaptation to IBM and VAX FORTRAN compilers has been largely due to the work of V. Vercesi, which I gratefully acknowledge.

REFERENCES

- [1] Proceedings of the 1982 DPF Summer Study on Elementary Particle Physics and Future Facilities, Snowmass, Colorado (1982).
- [2] M. Banner et al. (UA2 Collaboration), Phys. Lett. 118B (1982) 203;
P. Bagnaia et al. (UA2 Collaboration), CERN preprint EP-/83-94 (1983).
- [3] G. Arnison et al. (UA1 Collaboration), Phys. Lett. 123B (1983) 115;
CERN preprints EP/83-118,119 (1983).
- [4] R. Odorico, CERN preprint TH-3744 (1983).
- [5] A. Donati, R. Odorico and V. Roberto, Z. Physik C20 (1983) 9.
- [6] R. Singer, T. Fields and W. Selove, Phys. Rev. D25 (1982) 2451.
- [7] R. Odorico, Phys. Lett. 118B (1982) 151.
- [8] C. DeMarzo et al. (NA5 Collaboration), Phys. Lett. 112B (1982) 173.
- [9] R. Odorico, Nucl. Phys. B228 (1983) 381.
- [10] R. Odorico, CERN preprint TH-3678 (1983).
- [11] R. Odorico, CERN preprint TH-3761 (1983).
- [12] F.E. Paige and S. Protopopescu, Brookhaven preprints BNL-29777 (1981), and BNL-31987 (1982).
- [13] G.C. Fox and R.L. Kelly, Proc. of the Forward Collider Workshop, Madison, Wisconsin (1982), p. 585;
R.D. Field, G.C. Fox and R.L. Kelly, Phys. Lett. 119B (1982) 439.
- [14] R.D. Field and R.P. Feynman, Nucl. Phys. B136 (1978) 1.
- [15] B.L. Combridge, J. Kripfganz and J. Ranft, Phys. Lett. 70B (1977) 234;
B.L. Combridge, Nucl. Phys. B151 (1979) 429.

- [16] R.P. Feynman, R.D. Field and G.C. Fox, Phys. Rev. D18 (1978) 3320.
- [17] R. Odorico, Nucl. Phys. B209 (1982) 27.
- [18] J. Badier et al. (NA3 Collaboration), Phys. Lett. 89B (1979) 145, 96B (1980) 422 and Orsay preprint (1981).
- [19] R. Odorico, Nucl. Phys. B199 (1982) 189.
- [20] R. Odorico, Phys. Lett. 102B (1981) 341.
- [21] R. Odorico, Comp. Phys. Comm. 25 (1982) 253.
- [22] R. Odorico, Nucl. Phys. B172 (1980) 157;
P. Mazzanti and R. Odorico, Phys. Lett. 95B (1980) 133; Z. Physik C7 (1980) 81;
R. Odorico, Comp. Phys. Comm. 24 (1981) 73.
- [23] T. Sjostrand, Lund preprint LU TP 80-3 (1980).
- [24] C. Peterson, D.F. Schlatter, I. Schmitt and P.M. Zerwas, Phys. Rev. D27 (1983) 105.
- [25] A. Ali, G. Korner, G. Kramer and J. Willrodt, Z. Physik C1 (1979) 203; Nucl. Phys. B168 (1980) 409;
A. Ali and E. Pietarinen, Nucl. Phys. B154 (1979) 519.
- [26] G. Alberi and G. Goggi, Phys. Rep. 74 (1981) 1.
- [27] E.H. de Groot, Phys. Lett. 57B (1975) 159.
- [28] K. Alpgard et al. (UA5 Collaboration), Phys. Lett. 121B (1983) 209.
- [29] E. Albini et al., Nuovo Cimento 32A (1976) 101.

TABLE 1 - SCHEMATIC OUTLAY OF THE MAIN PROGRAM

(Features within brackets are optional)

(Resetting of input param.)

(CALL BOOKH) call to user's histogram booking (etc.) routine

(CALL FREEZE(H)) (possibly multiple calls) to freeze the final state at the level of quanta (H=1HQ) or to freeze decays of heavy flavor hadrons (H=1HC, 1HB and 1HT for charm, bottom and top hadrons, respectively)

(CALL FUSION(H)) to generate only heavy flavor production (H=1HC, 1HB and 1HT for charm, bottom and top hadrons, respectively).*

CALL PREGEN to compute pretabulation for parton transverse momentum generation

Loop on:

CALL GEVENT at each call several events may be generated, at each event generation a call to FILLH is made.

CALL PRHEAD print basic input parameters and event generation results

(CALL OUTH) call to user's output routine

(CALL PARTAB) to print particle table and basic hadronization parameters

* With the VAX FORTRAN 77 compiler, H should be defined via a data statement (as a real variable).

TABLE 2 - OUTPUT COMMON BLOCKS

BASIC EVENT INFORMATION

COMMON/PARQUA/ PARQUA(79,7),NQUA,MAXQUA

--Parameters of final quanta PARQUA(I,J), I=1,NQUA

J=1,5 4-momenta and mass (given by cutoff) of quanta, in order PX, PY, PZ, E and MASS (GeV) (positive z axis coincides with initial antiproton direction)

J=6 flavor index, for quarks 1-6 corresponding to u, d, s, c, b and t, in order, negative for corresponding antiquarks, 0 for gluons

J=7 origin, -1 initial bremsstrahlung, 1 otherwise

NQUA = no. of final quanta

MAXQUA (=79) control parameter on 1st dimension of PARQUA

COMMON/NEVENT/NEVENT,ECM,PARHAD(401,7),NPART,MXPART,WEIGHT

--Parameters of particles in the final state

NEVENT = event number (at the end of event generation gives the number of generated particles)

ECM = c.m. energy (GeV)

PARHAD(I,J), I=1,NPART

J=1,5 4-momentum and mass of the particle, in order PX, PY, PZ, E and MASS (GeV) (positive z axis coincides with initial antiproton direction)

J=6 particle code number -- see particle table (printed calling PARTAB)

J=7 no. IQ of originating quantum times corresponding sign of PARQUA(IQ,7) (i.e. negative if quantum is generated by initial bremsstrahlung), if 0 particle belongs to spectator jets

NPART = number of particles in final state

MXPART (=401) control parameter on 1st dimension of PARHAD

WEIGHT = 1, unless weighted event forcing is chosen (IPTWGT=1), in which case gives event weight

COMMON/PTHARD/PTHARD

PTHARD = SQRT(QSQ), where QSQ is the value of the Field-Feynman-Fox variable for the event (approxim., parton transverse momentum generated in the hard process)

MORE INFORMATION ABOUT THE EVENT

COMMON/ENTREV/S,QSQ,YQSQ,IPACH,XING(2),XMING(2)

,PXING(2),PYING(2),IFLING(2),ICHING(2)

--Initial parton parameters (before evolution) and evolution scale

S = ECM**2

QSQ = F.F.F. variable (=PTHARD**2)

YQSQ = ALOG(ALOG(QSQ/ALAMB**2)/ALOG(QZEV**2/ALAMB**2))

IPACH = initial parton channel, 1 for q-q and q-qb of different flavors, 2 for q-q of same flavors, 3 for q-qb of same flavors, 4 for q-G, 5 for G-G

--Initial parton parameters before evolution, L=1,2 for the two partons (L=1 for the antiproton parton)

XING(L) = E+PL fraction

XMING(L) = E-PL fraction

PXING(L) = PT x-component

PYING(L) = PT y-component

IFLING(L) = flavor code (1-3 for u, d and s, respectively, 7 (LGLUS) for gluons)

ICHING(L) = internal (positive) parton flavor code

COMMON/PARQ/ PARACT(47,7,2),MXACT,IBEAM,NBOOK(2),JBOOK,MBOOK(2)

,PARSPE(39,6,2),MXSPE,SHARD

--Parameters of interacting (spacelike) partons and associated bremsstrahlung (timelike) partons before (timelike) evolution

--Interacting partons

PARACT(1,J,L), J=1,7, L=1,2 (the rest of the 1st dimension is used only in the pretabulation)

L=1,2 for the 1st (antiproton) and 2nd parton

J=1 flavor code (same as IFLING)

J=2 internal (parent) code

J=3 E-PL fraction

J=4 evolution variable (like YQSQ above) defining parton virtuality

J=5 E+PL fraction

J=6 PT x-component

J=7 PT y-component

MXACT (=47) control parameter on 1st dimension of PARACT

TABLE 2
(Continuation)

--Initial bremsstrahlung partons before timelike evolution
 PARSPE(I,J,L), I according to internal ordering (see below), J=1,6, L=1,2
 L=1,2 according to whether it is associated to 1st or 2nd interacting parton
 J=1 PT x-component
 J=2 PT y-component
 J=3 E+PL (not fraction of)
 J=4 internal (parent) code
 J=5 virtual mass square (times a - sign for internal use)
 J=6 flavor code, as for IFLING
 --Bremsstrahlung quanta associated with a given interacting parton pair are to
 be found within PARSPE following parent-daughter relationships (given by J=4).
 If really interested in this information, the best way to collect it is to
 transfer the content of PARSPE into a similarly dimensioned user's array
 within subroutine TEVOL, in the loop transferring PARSPE into SETIN (which is
 the input of JETQCD).
 MXSPE (=39) control parameter on 1st dimension of PARSPE
 --The remaining parameters in the common block are for internal use

COMMON/PMBIAS/PMBIAS(5)

--Spectator jets
 PMBIAS(J), J=1,5, 4-momentum and invariant mass of the spectator jets system
 (same conventions as for corresponding locations of PARHAD)

OUTPUT FROM LOOSEQ AND LOOSEH

COMMON/JET/K(399,2),P(399,5)

--Parameters of particles in the cascades generated by LOOSEQ and LOOSEH
 K(I,M), I=1,N, N = number of cascade particles, given by 2nd argument of
 LOOSEQ and LOOSEH (stable particles only from LOOSEQ, stable and intermediate
 unstable particles from LOOSEH)
 K(I,1) = order no. in the cascade of the parent of particle having order no.
 I, with a - sign (or zero if primary particle) when particle I is unstable
 (only for LOOSEH output)
 K(I,2) = particle code no., with same conventions as for PARHAD(,6)
 P(I,J), I=1,N, J=1,5, 4-momentum and mass (GeV), with same conventions as for
 corresponding locations of PARHAD

MONITORING COMMON BLOCKS

COMMON/SPYFOR/SPYC,SPYB,SPYT,SPYQ,SPYCEL,SPYBEL,SPYTEL

--Spy parameters for freezing and forcing conditions
 SPYC, SPYB, SPYT and SPYQ = 1. if freezing of decays of hadrons with charm,
 bottom, top q.n. and of (all) quanta, respectively, is active, -1. otherwise
 SPYCEL, SPYBEL, SPYTEL = 1. if forcing of electron semileptonic decay of
 hadrons with charm, bottom and top q.n., respectively, is active, -1.
 otherwise

COMMON/OVER/KOVER,SPYOVR

--Reports occurrence of event generations overflowing particle and quanta
 array dimensions
 KOVER = see Table 3
 SPYOVR = number (in floating point) of overflow occurrences

COMMON/MAXN/MNBOOK,MNBOOK,MNPART,MNQUA,MNCUE,MNCUEJ,MNSJO

--Maximum utilization of 1st dimension of particle and quanta arrays
 MNBOOK for PARACT, MNBOOK for PARSPE, MNPART for PARHAD, MNQUA for PARQUA,
 MNCUE for CUE (sub. EVOL), MNCUEJ for CUE (sub. JETQCD), MNSJO for K and P

COMMON/HADQN/LHADR(110),LCHARG(110),LCHARM(110),LBOTT(110),LTOP(110)

LHADR(M), LCHARG(M), LCHARM(M), LBOTT(M) AND LTOP(M) give the hadron/lepton
 quality (1/0 respectively), electric charge, charm q.n., bottom q.n. and top
 q.n. (1,-1 or 0), respectively, of particle with code number M

TABLE 3 - INPUT PARAMETERS

(Default values in parenthesis)

BASIC INPUT PARAMETERS

IPBAR	/ALQGEN/	(1)	1 for antiproton-proton, 0 for proton-proton collisions
ECM	/EVENT/	(540.)	c.m. energy (GeV)
ALAMB	/ALQGEN/	(0.1)	QCD lambda parameter (GeV)
PT2INT	/ALQGEN/	(0.4)	mean primordial transverse momentum square of initial partons ((GeV/c)**2)
SIGINL	/ALQGEN/	(40.)	total inelastic cross section (mb)
PTMIN	/ALQGEN/	(4.)	minimum parton transverse momentum cutoff (GeV/c)
(QZEV)	/ALQGEN/	(4.)	initial evolution scale (GeV) (see text for changes)
NFLAV	/NFLAV/	(3)	number of flavors for final partons
TOPMAS	/TOP/	(35.)	top quark mass (GeV)

GENERAL PARAMETERS

TLEAVE	/TLEAVE/	(5.)	event generation stops when time left for execution is less than TLEAVE (sec)
FRATAB	/FRATAB/	(0.1/0.3)	fraction of execution time devoted to pretabulation (default 0.1 normally, 0.3 if FUSION is called)
NDUMP	/NDUMP/	(5)	number of events to be dumped
KOVER	/OVER/	(1)	controls action to be taken if overflow occurs (see text)

QCD BREMSSTRAHLUNG CONTROL

KEYBRE	/KEYBRE/	(0)	if 0 full QCD bremsstrahlung, if 1 longitudinal effects only, if 2 scal. viol. of parton densities only, if 3 no QCD bremsstrahlung at all
--------	----------	-----	--

SPACELIKE QCD EVOLUTION

ZMCUT	/ALQGEN/	(1.)	(numerical) infrared cutoff in the calculation of parton densities evolution (GeV). The cutoff on the splitting variable z is ZMCUT/ECM. See Ref. 20 on a 21 for analytic expressions giving accuracy and execution time as functions of the z cutoff
-------	----------	------	---

TIMELIKE QCD EVOLUTION

QZ	/CUTOFF/	(3.)	non-perturbative cutoff for gluons and light quarks
QZMASS(6)	/CUTOFF/	----	quark masses used to increase non-perturbative cutoffs for heavy quarks (GeV) (default: 0.,0.,0.,1.5,5.,35. GeV, for u,d,s,c,b,t, respectively) (N.B. QZMASS(6) is defined via TOPMAS)

HARD JET FRAGMENTATION

CX2	/PAR/	(0.77)	controls shape of long. fragm. funct. for light quarks $[(1-CX2+3CX2*(1-x)**2)dx]$
EPSPET	/PETER/	(0.25)	controls shape of long. fragm. funct. for heavy quarks $[(1/(x*(1-1/x-EPSPET*(MC/MH)**2/(1-x)**2))dx]$
QMAS(6)	/DATA1/	----	quark masses used in the hard process and in quark fragmentation (N.B. in MeV) (default as for QZMASS)
SIGMA	/PAR/	(600.)	defines the transverse momentum distribution of quarks (N.B. in MeV) $[exp(-PT**2/(2*SIGMA**2))dPT**2]$

SPECTATOR JETS

FMULMB	/MB/	(1.)	factor multiplying mean multiplicity
FPTMB	/MB/	(1.)	factor multiplying mean transverse momentum (leading nucleons not included)
CLPRO	/MB/	(0.7)	controls shape of long. distr. of leading nucleons $[exp(-CLPRO*XF)dx]$
PT2PRO	/MB/	(0.15)	mean transverse momentum square of leading nucleons
RPAIR	/MB/	----	particle ratios (default values printed by PARTAB)
RETPIZ	/MB/	(0.257)	ETA/(ETA+PI0) ratio

PARTICLE DATA

(Current values printed by PARTAB)

PMAS(110)	/DATA1/		particle masses (N.B. in MeV)
CBR(110)	/DATA2/		decay b.r.'s arranged in generation table format (= b.r. for 1st decay channel of particle, =1. for last decay channel)
IDECAY(110)	/STABLE/		by setting IDECAY(M)=0, particle of code no. M is made stable

FORCED PT GENERATION PARAMETERS

IPTWGT	/ALQGEN/	(0)	if 1 forced parton transverse momentum with weighted events (generation distribution multiplied by $Q**2+1.$), if 0 no forcing
FRAGEN	/ALQGEN/	(0.5)	(approx.) fraction of hard events (active only if IPTWGT=1)
PTMGE	/FORGEN/	(0.)	if PTMGE.GT.PTMIN parton transverse momentum is generated only above PTMGE, with the original distribution

TABLE 4 - FREEZING AND FORCING ROUTINES

CALLLED FROM MAIN PROGRAM

CALL FUSION(H) forces generation via fusion of heavy flavors; H=IHC, IHB or IHT, for charm, bottom and top, respectively

CALL FREEZE(H) freezes decay of heavy flavor particles with charm, bottom and top or of (all) quanta when H=IHC, IHB, IHT or IHQ, respectively (possibly, multiple calls)

CALLLED FROM FILLH

CALL LOOSEQ(IQ,N) develops fragmentation of (previously frozen) quantum with order no. IQ in the final state (IQ=1,NQUA). N returns the number of particles in the cascade (only stable particles are kept, respecting freezing conditions for heavy flavors). Cascade output available from /JET/ (see Table 2).

CALL LOOSEH(I,N) develops decay of (previously frozen) particle with order no. I in the final state (I=1,NPART). N returns the total number of stable and unstable (intermediate) particles in the cascade, whose parameters are stored in /JET/

CALL STABKP(N) eliminates unstable particles from /JET/, N being given by a previous call to LOOSEH

CALL SHOCAS(N) lists content of /JET/, N being given by a previous call to LOOSEH or LOOSEQ

CALL SPECTJ carries on hadronization of spectator jets, by adding particles to PARHAD so as to satisfy 4-momentum conservation

CALL FORSLE(H,KEY) if KEY=1 modifies current decay branching fractions so as to have forcing of electron semileptonic decays on particles of heavy flavor H (=IHC, IHB or IHT for charm, bottom or top, respectively). If KEY=-1 reestablishes original branching fractions.

N.B. On VAX, hollerith arguments in subroutine calls must be given via a real variable defined in a data statement

TABLE 5 - LIST OF COMMON BLOCKS
(Including also those in WIZJET)

```

COMMON/ALQGEN/WGALQT(127),WGALQ(127,5,3),Q2VECT(128),NALQ,DALQ
1 ,ALAMB,ALLAM2,PTMIN,PTMAX,PT2INT,QZEV,ALQZM,ALQI,ALQF,YF
2 ,IPBAR,NCASC,SIGINL,PROPT,IPTWGT,PROPTW,PTWGTM,FRHARD,FRAGEH,FRAT
3 ,ZMCUT,ZLW,ZUP,XMIN
COMMON/BOSON/PBOS(6),ICHDB,IQRKFW
COMMON/BKMG/BKMG,FBKMG(7)
COMMON/BXGEN/TBXGEN(127),TBXAL(127),LBXAL(127),F2X
COMMON/CANC/ICANC(2),NBKNC(47,2)
COMMON/CBROR/CBROR(180),IDBOR(110)
COMMON/CFUN1/IFUN
COMMON/CUTOFF/QZ,QZSQ,QZMASS(6),QZFL(7),QZFLSQ(7),QTHRSQ(7)
1 ,NFLAVT,LGLU
COMMON /DATA1/ MESO(36), CMIX(6,2), PMAS(110), QMAS(6)
COMMON /DATA2/ CBR(180), KDP(180,3), WTCOR(10)
COMMON /DATA3/ CHA1(36), CHA2(110), CHA3(2)
COMMON /DECPAR/ PUDD, PSID, CND1, CND2, IDB(110)
COMMON /EDPAR/ ITHROW, PZMIN, PMIN, THETA, PHI, BETA(3)
COMMON/ENTREV/S,QSQ,YQSQ,IPACH,XING(2),XING(2)
1 ,PXING(2),PYING(2),IFLING(2),ICHING(2)
COMMON/EVENT/NEVENT,ECH,PARHAD(401,7),NPART,MXPART,WEIGHT
COMMON/FFMUL/AVNCH
COMMON/FINT/ALLAM2,LFLA
COMMON/FORGEN/PTMGE,WGGE,SIGHRD
COMMON/FLAPRE/FLAPRE
COMMON/FRATAB/FRATAB
COMMON/FXWDY/FXWDY
COMMON/F2QG/F2QG(7)
COMMON/HADQN/LHADRI(110),LCHARG(110),LCHARM(110),LBOTT(110)
*,LTOP(110)
COMMON/INMAT/IFLAIN(2),XIN(2),PXIN(2),PYIN(2),IMATCH(4,4)
1 ,ICHA(2,3),IWGV(3)
COMMON/KDUMP/NDUMP
COMMON/KEYBRE/KEYBRE
COMMON/KEYJET/IZFAST
COMMON/KHADRO/KHADRO
COMMON /JET/ K(399,2), P(399,5)
COMMON/JETN/NP,NP1
COMMON/MAXN/MNBOOK,MNBOOK,MNPART,MNQUA,MNCUE,MNCUEJ,MNSJO
COMMON/MAXSJO/MAXSJO
COMMON/NB/CLPRO,PT2PRO,FPTMB,FMULMB,RETPIZ,RPAIR(9)
COMMON/NEVOL/NEVOL,ISTOP(2),IBACK(2),MBOOKS(2)
COMMON/NFLAV/NFLAV,KFRFLA
COMMON/OVER/KOVER,SPYOV
COMMON /PAR/ PUD, PS1, SIGMA, CX2, EBEG, WFIN, IFLBEG
COMMON/PARQ/PARACT(47,7,2),MXACT,IBEAM,NBOOK(2),JBOOK,MBOOK(2)
1 ,PARSPE(39,6,2),MXSPE,SHARD
COMMON/PARQUA/PARQUA(79,7),NQUA,MAXQUA
COMMON/PETER/EPSPET
COMMON/PMASS/PMASHD(110)
COMMON/PMBIAS/PMBIAS(5)
COMMON/PTHARD/PTHARD
COMMON/QUAOR/IQUAOR
COMMON/QCD/ALAMB,CF,CA,BALPH,PI
COMMON/QCDS/CF,CA,BALPH,PI,PI2,PROGL,PROFLA(2),NFLAVS,LGLUS,IVAL
COMMON/ROTQ/IROT,CT,ST,CPH,SPH
COMMON/SHEAVY/IHEV(399)
COMMON/SPYFOR/SPYC,SPYB,SPYT,SPYQ,SPYCEL,SPYBEL,SPYTEL
COMMON/STABLE/IDECAY(110)
COMMON/TABPSQ/TABPSQ(5,129),QSQMAX,ALMAX,ALMIN(7),DSTEP(7)
1 ,NPREC,NTAB,KODTAB(7),MAXTAB
COMMON/TABGRK/TABGRK(127,7),NBIN,OBIN,F2XQV(7),WGQV(7),AGLU,HDBIN
1 ,XV(127),AL1,AL2,IFCODE(5)
2 ,TGLIAS(127),LGLIAS(127),TGLIAS(127),LGLIAS(127)
COMMON/THRFLA/THRFLA(6),AM2HEV(6)
COMMON/TLEAVE/TLEAVE
COMMON/TOP/TOPMAS
COMMON/TRANSL/ALAMBT,NFLATR,LGLUT,QSQMXT
COMMON/TWEIGH/TWEIGH,TENTRS,TWEIMN
COMMON/VV/L1(180),L2(180),L3(180)
COMMON/WEAKBO/WEAKBO,BOMASS,IDECWK,IWIDTH
COMMON/WEAKON/WEAKON,IFDC,BIM2,BIML2,BIMX2,TG2CAB,FZUDS,FXSCT,NGEX
*,WIDTH
COMMON/ZPAR/S2THW,COSCAB

```

TABLE 6 - LIST OF SUBROUTINES AND FUNCTIONS

(No. on the right gives position in the listing)

BLOCK DATA	000151
SUBROUTINE ALIAS(L,S,NPTS)	001200
FUNCTION ASIMP(A1,B1,EP,M,N,FUN)	002532
FUNCTION BETAEU(A,B)	000554
FUNCTION BKWEIG(X)	001145
SUBROUTINE BOOK(IFLA1,IBOOK,XM1,Y1,X1,PX1S,PY1S)	002438
SUBROUTINE BOOKH	004995
SUBROUTINE CHDEC(IP,IDC,CH)	004174
SUBROUTINE CHEAVY(IL)	004666
SUBROUTINE DECAY(IPD,I)	004379
SUBROUTINE DSDT(IM,S,T,U,WTOSDT,WFDSDT)	001394
FUNCTION DSHVG(A2,S,T,U)	001491
FUNCTION DSHVQ(A2,S,T,U)	001484
SUBROUTINE DUMPEV	004011
SUBROUTINE EDIT(N)	004608
SUBROUTINE EVOL(NBK,YQSQ)	002139
FUNCTION FGLU(X)	000560
FUNCTION FIGLU(ALPSQ)	002774
SUBROUTINE FILLH	005024
SUBROUTINE FINDX(ICH,IXST,X,IX)	001943
FUNCTION FIQRK(ALPSQ)	002757
FUNCTION FMUL(AN)	003463
FUNCTION FMULDG(Z)	003472
SUBROUTINE FORSLE(HFLA,KEY)	004879
FUNCTION FPMUL(D)	003794
SUBROUTINE FREEZE(HFLAQ)	004688
FUNCTION FUN1(ALX)	001153
FUNCTION FUN2(X)	001163
FUNCTION FUN3(ALX)	001172
SUBROUTINE FUSION(HFLA)	000744
FUNCTION F2IN(X,IT)	000504
SUBROUTINE GECODE(ICHING,IFLING)	001895
SUBROUTINE GEFLA(IX,IFLA,JFLA)	001180
SUBROUTINE GEPAIR	001773
SUBROUTINE GEQSQ(ALQ,IALQ)	001739
SUBROUTINE GEVENT	001503
SUBROUTINE GEXIN(ICH,XST,IXST,ALST,ALST1,X,XM,PX,PY)	001913
SUBROUTINE HADRON(NQUA1,NQUA2,IMBIAS)	003222
SUBROUTINE HARD(J1,J2,IFLAF1,IFLAF2,SHAT,T)	001973
SUBROUTINE INIQCD	002642
SUBROUTINE INPART(ICH,IFLA,X,PX,PY,WG)	001008
SUBROUTINE INQCD5	002109
SUBROUTINE JETGEN(N)	004238
SUBROUTINE JETQCD(SETIN)	002794
SUBROUTINE JTBOOK(PX,PY,EPP,PSQ,IFLA)	003000
SUBROUTINE KNO(W,PMB)	003479
SUBROUTINE LIST(N)	004587
SUBROUTINE LOOSEH(IH,N)	004797
SUBROUTINE LOOSEQ(IQ,N)	004748
SUBROUTINE LOREN4(PMB,V1,V2)	003449
SUBROUTINE MATCH(IWG,WGTCAS)	001242
SUBROUTINE MNBIA5	003345
SUBROUTINE OUTH	005113
SUBROUTINE OVERDM	002503
FUNCTION PAPINT(IFUN,Z)	003142
SUBROUTINE PARTAB	003907
SUBROUTINE PREGEN	000775
SUBROUTINE PRHEAD	000597
SUBROUTINE PSQGEN(IFLA,XPSQ,PSQ,IGO)	003039
FUNCTION PTWGT(QSQ)	000585
SUBROUTINE P4CORR(W,PMB,NP,IFLAG)	003850
SUBROUTINE Q2GEN(IFLA,YIN,Y,QSQ)	002361
FUNCTION RNDM(A)	004979
SUBROUTINE ROTQUA	002091
SUBROUTINE SBLOCK	000383
SUBROUTINE SHOCAS(N)	004860
SUBROUTINE SPECTJ	004789
SUBROUTINE STABKP(N)	004840
SUBROUTINE TABLES	002683
SUBROUTINE TBOOK(IBOOK,IFLA1,TPSQ1,X1T,PX1T,PY1T,IBOOK1)	002473
SUBROUTINE TEVOL(IFS)	002047
SUBROUTINE TIMELF(TLE)	004972
FUNCTION WPARDN(X)	000578
FUNCTION X1EXP(A)	003821
SUBROUTINE ZGEN(IFUN,ZLW,ZUP,FLAG,Z)	003091
SUBROUTINE ZGENS(IFUN,Z,IFLAQ)	002379
SUBROUTINE ZLIM(PSQ,QZ1SQ,QZ2SQ,ZLW,ZUP,FLAG)	003177
SUBROUTINE ZLIMSY(PSQ,QZSQ,ZLW,ZUP,FLAG)	003196

K(I,2) PARHAD(I,6)	PARTICLE	MASS	DECAY PRODUCTS			B.R.	DEC. CHANN. IDB(K(I,2))
1	GAMM	0.0000					
2	Z0	95.0000	E- MU- TAU- NUE NUM NUT U-UB D-DB S-SB C-CB B-BB T-TB	E+ MU+ TAU+ NUEB NUMB NUTB		.03 .03 .03 .06 .06 .06 .11 .14 .14 .11 .14 .06	150 151 152 153 154 155 156 157 158 159 160 161
3	W+	81.0000	NUE NUM NUT U-DB C-SB T-BB	E+ MU+ TAU+		.09 .09 .09 .27 .27 .19	170 171 172 173 174 175
4	W-	81.0000	E- MU- TAU- D-UB S-CB B-TB	NUEB NUMB NUTB		.09 .09 .09 .27 .27 .19	170 171 172 173 174 175
5							
6							
7	E-	.0005					
8	E+	.0005					
9	NUE	0.0000					
10	NUEB	0.0000					
11	MU-	.1057					
12	MU+	.1057					
13	NUM	0.0000					
14	NUMB	0.0000					
15	TAU-	1.7820	NUEB NUMB NUT	E- MU- D-UB	NUT NUT	.17 .17 .66	1 2 3
16	TAU+	1.7820	NUEB NUMB NUT	E- MU- D-UB	NUT NUT	.17 .17 .66	1 2 3
17	NUT	0.0000					
18	NUTB	0.0000					
19							
20							
21	PI+	.1396					
22	PI-	.1396					
23	K+	.4937					
24	K-	.4937					

25	K0	.4977	K0S K0L			.50 .50	4 5
26	KB	.4977	K0S K0L			.50 .50	4 5
27	D0	1.8633	E+ MU+ U-DB	NUE NUM S-UB	S-UB S-UB	.05 .05 .90	6 7 8
28	DB	1.8633	E+ MU+ U-DB	NUE NUM S-UB	S-UB S-UB	.05 .05 .90	6 7 8
29	D+	1.8683	E+ MU+ U-DB	NUE NUM S-DB	S-DB S-DB	.16 .16 .68	9 10 11
30	D-	1.8683	E+ MU+ U-DB	NUE NUM S-DB	S-DB S-DB	.16 .16 .68	9 10 11
31	F+	2.0400	E+ MU+ U-DB U-DB	NUE NUM S-SB	S-SB S-SB	.10 .10 .50 .30	12 13 14 15
32	F-	2.0400	E+ MU+ U-DB U-DB	NUE NUM S-SB	S-SB S-SB	.10 .10 .50 .30	12 13 14 15
33	BU-	5.0700	NUEB NUMB NUTB D-UB S-CB	E- MU- TAU- C-UB C-UB	C-UB C-UB C-UB	.17 .17 .05 .51 .10	16 17 18 19 20
34	BU+	5.0700	NUEB NUMB NUTB D-UB S-CB	E- MU- TAU- C-UB C-UB	C-UB C-UB C-UB	.17 .17 .05 .51 .10	16 17 18 19 20
35	B00	5.0700	NUEB NUMB NUTB D-UB S-CB	E- MU- TAU- C-DB C-DB	C-DB C-DB C-DB	.17 .17 .05 .51 .10	21 22 23 24 25
36	BDB	5.0700	NUEB NUMB NUTB D-UB S-CB	E- MU- TAU- C-DB C-DB	C-DB C-DB C-DB	.17 .17 .05 .51 .10	21 22 23 24 25
37	B00	5.2700	NUEB NUMB NUTB D-UB S-CB C-CB	E- MU- TAU- C-SB C-SB	C-SB C-SB C-SB	.16 .16 .05 .48 .10 .05	26 27 28 29 30 31
38	BSB	5.2700	NUEB NUMB NUTB D-UB S-CB C-CB	E- MU- TAU- C-SB C-SB	C-SB C-SB C-SB	.16 .16 .05 .48 .10 .05	26 27 28 29 30 31
39	BC-	6.3700	TAU- NUEB NUMB	NUTB E- MU-	C-CB C-CB	.07 .11 .11	32 33 34

			NUTB	TAU-	C-CB	.03	35
			D-UB	C-CB		.34	36
			S-CB	C-CB		.07	37
			S-CB			.27	38
40	BC+	6.3700	TAU-	NUTB		.07	32
			NUEB	E-	C-CB	.11	33
			NUMB	MU-	C-CB	.11	34
			NUTB	TAU-	C-CB	.03	35
			D-UB	C-CB		.34	36
			S-CB	C-CB		.07	37
			S-CB			.27	38
41	TU0	35.2700	E+	NUE	B-UB	.09	39
			MU+	NUM	B-UB	.09	40
			TAU+	NUT	B-UB	.08	41
			U-DB	B-UB		.29	42
			C-SB	B-UB		.22	43
			B-DB			.23	44
42	TUB	35.2700	E+	NUE	B-UB	.09	39
			MU+	NUM	B-UB	.09	40
			TAU+	NUT	B-UB	.08	41
			U-DB	B-UB		.29	42
			C-SB	B-UB		.22	43
			B-DB			.23	44
43	TD+	35.2700	E+	NUE	B-DB	.12	45
			MU+	NUM	B-DB	.12	46
			TAU+	NUT	B-DB	.10	47
			U-DB	B-DB		.37	48
			C-SB	B-DB		.29	49
44	TD-	35.2700	E+	NUE	B-DB	.12	45
			MU+	NUM	B-DB	.12	46
			TAU+	NUT	B-DB	.10	47
			U-DB	B-DB		.37	48
			C-SB	B-DB		.29	49
45	TS+	35.4700	E+	NUE	B-SB	.12	50
			MU+	NUM	B-SB	.12	51
			TAU+	NUT	B-SB	.10	52
			U-DB	B-SB		.37	53
			C-SB	B-SB		.29	54
46	TS-	35.4700	E+	NUE	B-SB	.12	50
			MU+	NUM	B-SB	.12	51
			TAU+	NUT	B-SB	.10	52
			U-DB	B-SB		.37	53
			C-SB	B-SB		.29	54
47	TC0	36.5700	E+	NUE	B-CB	.09	55
			MU+	NUM	B-CB	.09	56
			TAU+	NUT	B-CB	.08	57
			U-DB	B-CB		.29	58
			C-SB	B-CB		.22	59
			B-SB			.23	60
48	TCB	36.5700	E+	NUE	B-CB	.09	55
			MU+	NUM	B-CB	.09	56
			TAU+	NUT	B-CB	.08	57
			U-DB	B-CB		.29	58
			C-SB	B-CB		.22	59
			B-SB			.23	60
49	TB+	39.7700	E+	NUE	B-BB	.09	61
			MU+	NUM	B-BB	.09	62
			TAU+	NUT	B-BB	.08	63
			U-DB	B-BB		.29	64
			C-SB	B-BB		.22	65
			C-SB			.23	66
50	TB-	39.7700	E+	NUE	B-BB	.09	61
			MU+	NUM	B-BB	.09	62
			TAU+	NUT	B-BB	.08	63
			U-DB	B-BB		.29	64
			C-SB	B-BB		.22	65

			C-SB			.23	66
51	PIO	.1350	GAMM GAMM	GAMM E+	E-	.99 .01	67 68
52	ETA	.5488	GAMM PIO PI- GAMM GAMM	GAMM PIO PI+ PI- GAMM	PIO PIO PI+ PIO	.38 .30 .24 .05 .03	69 70 71 72 73
53	ETAP	.9576	PI+ PIO GAMM GAMM GAMM	PI- PIO RHO0 OMEG GAMM	ETA ETA	.43 .24 .30 .02 .02	74 75 76 77 78
54	ETAC	2.8500	U-UB D-DB S-SB			.66 .17 .17	79 80 81
55	ETAB	9.4000	U-UB D-DB S-SB C-CB			.40 .10 .10 .40	82 83 84 85
56	ETAT	69.9700	U-UB D-DB S-SB C-CB B-BB			.37 .09 .09 .36 .09	86 87 88 89 90
57	K0S	.4977	PI+ PIO	PI- PIO		.69 .31	91 92
58	K0L	.4977					
59							
60							
61	RHO+	.7659	PI+	PIO		1.00	93
62	RHO-	.7659	PI+	PIO		1.00	93
63	K*+	.8922	K0 K+	PI+ PIO		.67 .33	94 95
64	K*-	.8922	K0 K+	PI+ PIO		.67 .33	94 95
65	K*0	.8963	K+ K0	PI- PIO		.67 .33	96 97
66	K*B	.8963	K+ K0	PI- PIO		.67 .33	96 97
67	D*0	2.0060	D0 D0	PIO GAMM		.55 .45	98 99
68	D*B	2.0060	D0 D0	PIO GAMM		.55 .45	98 99
69	D*+	2.0086	D0 D+ D+	PI+ PIO GAMM		.60 .20 .20	100 101 102
70	D*-	2.0086	D0 D+ D+	PI+ PIO GAMM		.60 .20 .20	100 101 102

71	F**	2.1400	F+	GAMM		1.00	103
72	F*-	2.1400	F+	GAMM		1.00	103
73	BU*-	5.1100	BU-	GAMM		1.00	104
74	BU**	5.1100	BU-	GAMM		1.00	104
75	BD*0	5.1100	BD0	GAMM		1.00	105
76	BD*B	5.1100	BD0	GAMM		1.00	105
77	BS*0	5.3100	BS0	GAMM		1.00	106
78	BS*B	5.3100	BS0	GAMM		1.00	106
79	BC*-	6.4100	BC-	GAMM		1.00	107
80	BC**	6.4100	BC-	GAMM		1.00	107
81	TU*0	35.3100	TU0	GAMM		1.00	108
82	TU*B	35.3100	TU0	GAMM		1.00	108
83	TD**	35.3100	TD+	GAMM		1.00	109
84	TD*-	35.3100	TD+	GAMM		1.00	109
85	TS**	35.5100	TS+	GAMM		1.00	110
86	TS*-	35.5100	TS+	GAMM		1.00	110
87	TC*0	36.6100	TC0	GAMM		1.00	111
88	TC*B	36.6100	TC0	GAMM		1.00	111
89	TB**	39.8100	TB+	GAMM		1.00	112
90	TB*-	39.8100	TB+	GAMM		1.00	112
91	RH00	.7702	PI+	PI-		1.00	113
92	OMEG	.7826	PI- GAMM PI-	PI+ PI0 PI+	PI0	.90 .09 .01	114 115 116
93	PHI	1.0196	K+ K0 PI+ GAMM	K- KB PI- ETA	PI0	.49 .35 .15 .02	117 118 119 120
94	PSI	3.0970	E+ MU+ U-UB D-DB S-SB	E- MU-		.07 .07 .56 .15 .15	121 122 123 124 125
95	UPSI	9.4600	E+ MU+ TAU+ ETAB	E- MU- TAU- GAMM		.04 .04 .04 .88	126 127 128 129

96	PHIT	70.0100	E+	E-	.08	130
			MU+	MU-	.08	131
			TAU+	TAU-	.08	132
			ETAT	GAMM	.76	133
97						
98						
99						
100						
101	P	.9383				
102	PB	.9383				
103	N	.9396				
104	NB	.9396				
105	LAM	1.1156				
106	LAMB	1.1156				
107	SIGP	1.1894				
108	SGPB	1.1894				
109	SIGM	1.1973				
110	SGMB	1.1973				

```

C
C-- TEST 1
C-- INCLUSIVE ELECTRON PT SPECTRUM FROM TOP DECAY
C-- SET IN MAIN PROGRAM (BEFORE CALL PREGEN)
C--      CALL FREEZE(1HQ)
C--      CALL FUSION(1HT)
C--
C
      SUBROUTINE BOOKH
      COMMON//B(2000)
      CALL HLIMIT(2000)
      CALL HBOOK1(5001,10HEL PT      ,50,0.,50.)
C-- FORCE ELECTRON S.L. DECAY OF TOP
      CALL FORSLE(1HT,1)
      RETURN
      END
C
      SUBROUTINE FILLH
      COMMON/PARQUA/PARQUA(79,7),NQUA,MAXQUA
      COMMON /JET/ K(399,2), P(399,5)
      DO 100 IQ=1,NQUA
      IF LAQ=PARQUA(IQ,6)
C-- NEGLECT NON-TOP QUANTA
      IF(IABS(IFLAQ).NE.6) GO TO 100
C-- LET TOP QUARK DECAY
      IQQ=IQ
      CALL LOOSEQ(IQQ,N)
C-- PICK UP ELECTRONS AND POSITRONS
      DO 101 I=1,N
      IF(K(I,2).NE.7.AND.K(I,2).NE.8) GO TO 101
      PTEL=SQRT(P(I,1)**2+P(I,2)**2)
      CALL HFILL(5001,PTEL,0.,1.)
101  CONTINUE
100  CONTINUE
      RETURN
      END
C
      SUBROUTINE OUTH
      CALL HISTDO
      RETURN
      END

```

```

C
C-- TEST 2
C-- SAME SIGN DI-ELECTRONS FROM TOP-TOPB DECAYS (2ND VIA BOTTOM)
C-- BY HANDLING FRAGMENTATION OF QUANTA
C-- GIVES SCATTER PLOT IN THE TRANSVERSE MOMENTA OF THE TWO ELECTRONS
C-- SET IN MAIN PROGRAM (BEFORE CALL FREGEN)
C--      CALL FREEZE(1HQ)
C--      CALL FUSION(1HT)
C--
C
      SUBROUTINE BOOKH
      COMMON//B(2000)
      CALL HLIMIT(2000)
      CALL HBOOK2(5002,10HPT1 VS PT2,50,0.,50.,50,0.,50.)
      RETURN
      END

C
      SUBROUTINE FILLH
      COMMON/PARQUA/PARQUA(79,7),NQUA,MAXQUA
      COMMON /JET/ K(399,2), P(399,5)
      REAL PTELV(2)
      PTELV(1)=0.
      PTELV(2)=0.
C-- RANDOMIZE CASCADE FOR FORCING
      ICAS=2.*RNDM(0.)+1.
      DO 200 IQ=1,NQUA
        IFLAQ=PARQUA(IQ,6)
C-- NEGLECT NON-TOP QUANTA
        IF(IABS(IFLAQ).NE.6) GO TO 200
C-- FORCE ELECTRON S.L. DECAY OF TOP IN 1ST CASCADE
        IF(ICAS.EQ.1) CALL FORSLE(1HT,1)
C-- FORCE ELECTRON S.L. DECAY OF BOTTOM IN 2ND CASCADE
        IF(ICAS.EQ.2) CALL FORSLE(1HB,1)
C-- LET TOP QUARK TO DECAY
        IQQ=IQ
        CALL LOOSEQ(IQQ,N)
C-- DEACTIVATE ELECTRON S.L. DECAY FORCING
        IF(ICAS.EQ.1) CALL FORSLE(1HT,-1)
        IF(ICAS.EQ.2) CALL FORSLE(1HB,-1)
C-- CHOOSE ELECTRON CHARGES
        IF(ICAS.EQ.1.AND.IFLAQ.EQ. 6) IPARS=8
        IF(ICAS.EQ.1.AND.IFLAQ.EQ.-6) IPARS=7
        IF(ICAS.EQ.2.AND.IFLAQ.EQ.-6) IPARS=8
        IF(ICAS.EQ.2.AND.IFLAQ.EQ. 6) IPARS=7
C-- PICK UP ELECTRONS IN THE CASCADE
        DO 201 I=1,N
          IF(K(I,2).NE.IPARS) GO TO 201
          PTEL=SQRT(P(I,1)**2+P(I,2)**2)
          PTELV(ICAS)=AMAX1(PTELV(ICAS),PTEL)
201      CONTINUE
          ICAS=3-ICAS
200      CONTINUE
          CALL HFILL(5002,PTELV(1),PTELV(2),1.)
          RETURN
          END

C
      SUBROUTINE OUTH
      CALL HIST00
      RETURN
      END

```

```

C
C-- TEST 3
C-- HANDLING TOP DECAYS PRESERVING CASCADE INFORMATION
C-- USEFUL FOR VERTEX DETECTOR STUDIES
C-- ONE TOP DECAY WITH ELECTRON S.L. FORCING
C-- GIVES CASCADE LISTING THROUGH CALL TO SHOCAS
C-- SET IN MAIN PROGRAM (BEFORE CALL PREGEN)
C--      CALL FREEZE(1HT)
C--      CALL FUSION(1HT)
C--      NDUMP=0
C--
C
      SUBROUTINE BOOKH
      RETURN
      END

C
      SUBROUTINE FILLH
      COMMON/EVENT/NEVENT,ECM,PARHAD(401,7),NPART,MXPART,WEIGHT
      COMMON /JET/ K(399,2), P(399,5)
      COMMON/HADQN/LHADR(110),LCHARG(110),LCHARM(110),LBOTT(110)
      *,LTOP(110)
      PRINT 301,NEVENT
C-- RANDOMIZE TOP CASCADE FOR FORCING
      ICAS=2.*RNDM(0.)+1.
      DO 300 IH=1,NPART
      IPAR=PARHAD(IH,6)
C-- NEGLECT NON-TOP PARTICLES
      IF(LTOP(IPAR).EQ.0) GO TO 300
C-- FORCE ELECTRON S.L. DECAY OF 1ST TOP PARTICLE
      IF(ICAS.EQ.1) CALL FORSLE(1HT,1)
C-- LET TOP PARTICLE DECAY
      IHH=IH
      CALL LOOSEN(IHH,N)
C-- DEACTIVATE ELECTRON S.L. DECAY FORCING
      IF(ICAS.EQ.1) CALL FORSLE(1HT,-1)
C-- LIST TOP DECAY CASCADE
      IF(ICAS.EQ.1) PRINT 302
      IF(ICAS.EQ.2) PRINT 303
      CALL SHOCAS(N)
      ICAS=3-ICAS
300  CONTINUE
      RETURN
301  FORMAT(5(/),1X,9HEVENT NO.,I10/1X,9(1H*))
302  FORMAT(3(/),1X,41HTOP CASCADE -- ELECTRON S.L. DECAY FORCED)
303  FORMAT(3(/),1X,31HTOP CASCADE -- NO DECAY FORCING)
      END

C
      SUBROUTINE OUTH
      RETURN
      END

```



```

C
C-- TEST 4
C-- SAME SIGN DI-ELECTRONS FROM TOP-TOPB DECAYS (2ND VIA BOTTOM)
C-- BY HANDLING TOP PARTICLE DECAYS
C-- GIVES SCATTER PLOT IN THE TRANSVERSE MOMENTA OF THE TWO ELECTRONS
C-- SET IN MAIN PROGRAM (BEFORE CALL PREGEN)
C--      CALL FREEZE(IHT)
C--      CALL FUSION(IHT)
C--
C
C      SUBROUTINE BOOKH
C      COMMON//B(2000)
C      CALL HLIMIT(2000)
C      CALL HBOOK2(5004,10HPT1 VS PT2,50,0.,50.,50,0.,50.)
C      RETURN
C      END

C
C      SUBROUTINE FILLH
C      COMMON/EVENT/NEVENT,ECH,PARHAD(401,7),NPART,MXPART,WEIGHT
C      COMMON /JET/ K(399,2), P(399,5)
C      COMMON/HADQN/LHADR(110),LCHARG(110),LCHARM(110),LBOTT(110)
C      *,LTOP(110)
C      REAL PTELV(2)
C      PTELV(1)=0.
C      PTELV(2)=0.
C-- RANDOMIZE TOP CASCADE FOR FORCING
C      ICAS=2.*RNDM(0.)+1.
C      DO 400 IH=1,NPART
C      IPAR=PARHAD(IH,6)
C-- NEGLECT NON-TOP PARTICLES
C      IF(LTOP(IPAR).EQ.0) GO TO 400
C-- FORCE ELECTRON S.L. DECAY OF TOP IN 1ST CASCADE
C      IF(ICAS.EQ.1) CALL FORSLE(IHT,1)
C-- FORCE ELECTRON S.L. DECAY OF BOTTOM IN 2ND TOP CASCADE
C      IF(ICAS.EQ.2) CALL FORSLE(IHB,1)
C-- LET TOP PARTICLE DECAY
C      IHH=IH
C      CALL LOOSEH(IHH,N)
C-- DEACTIVATE ELECTRON S.L. DECAY FORCING
C      IF(ICAS.EQ.1) CALL FORSLE(IHT,-1)
C      IF(ICAS.EQ.2) CALL FORSLE(IHB,-1)
C-- CHOOSE ELECTRON CHARGES
C      IF(ICAS.EQ.1.AND.LTOP(IPAR).GT.0) IPARS=8
C      IF(ICAS.EQ.1.AND.LTOP(IPAR).LT.0) IPARS=7
C      IF(ICAS.EQ.2.AND.LTOP(IPAR).LT.0) IPARS=8
C      IF(ICAS.EQ.2.AND.LTOP(IPAR).GT.0) IPARS=7
C-- PICK UP ELECTRONS WITHIN CASCADE
C      DO 401 I=1,N
C      IF(K(I,2).NE.IPARS) GO TO 401
C      PTEL=SQRT(P(I,1)**2+P(I,2)**2)
C      PTELV(ICAS)=AMAX1(PTELV(ICAS),PTEL)
401  CONTINUE
C      ICAS=3-ICAS
400  CONTINUE
C      CALL HFILL(5004,PTELV(1),PTELV(2),1.)
C      RETURN
C      END

C
C      SUBROUTINE OUTH
C      CALL HISTDO
C      RETURN
C      END

```

EXCERPT OF TEST RUN OUTPUT NO. 1

(JET PARAMETERS ARE BEFORE FRAGMENTATION)

EVENT NO. 1

JET NO. 1 FLAVOR = GL E = .581E+01
 PJX = -.102E+01 PJY = .146E+00 PJZ = -.562E+01
 ET = .103E+01 Y = -.240E+01 THETA = .170E+03 PHI = .172E+03
 (INITIAL BREMSSTRAHLUNG)

I	ORI	PART	STAB	PX	PY	PZ	E	M
1	0	PI-	STAB	-.658E+00	.400E+00	-.499E+01	.505E+01	.140E+00

JET NO. 2 FLAVOR = GL E = .409E+01
 PJX = -.639E+00 PJY = -.207E+01 PJZ = -.269E+01
 ET = .216E+01 Y = -.104E+01 THETA = .141E+03 PHI = .253E+03
 (INITIAL BREMSSTRAHLUNG)

I	ORI	PART	STAB	PX	PY	PZ	E	M
1	0	PI-	STAB	-.887E+00	-.951E+00	-.205E+01	.243E+01	.140E+00
2	0	PI0	STAB	-.991E-01	-.886E+00	-.109E+01	.142E+01	.135E+00

JET NO. 3 FLAVOR = GL E = .577E+01
 PJX = .315E+01 PJY = -.595E+00 PJZ = -.374E+01
 ET = .320E+01 Y = -.995E+00 THETA = .139E+03 PHI = .349E+03
 (INITIAL BREMSSTRAHLUNG)

I	ORI	PART	STAB	PX	PY	PZ	E	M
1	0	PI+	STAB	.848E+00	-.929E-01	-.106E+01	.136E+01	.140E+00
2	0	PI-	STAB	.409E+00	.272E-01	-.131E+00	.452E+00	.140E+00
3	0	PI0	STAB	.652E+00	-.205E+00	-.793E+00	.106E+01	.135E+00
4	0	PI+	STAB	.142E+01	-.920E+00	-.685E+00	.183E+01	.140E+00

JET NO. 4 FLAVOR = UB E = .389E+02
 PJX = -.219E+02 PJY = .886E+00 PJZ = .320E+02
 ET = .219E+02 Y = .117E+01 THETA = .345E+02 PHI = .178E+03
 (PARTON FINAL STATE)

EXCERPT OF TEST RUN OUTPUT NO. 2

```

*****
*****
*****
***** COJETS *****
*****
*****
***** VERSION 2.2 -- 7 NOV 83 *****
*****
*****
ANTIPROTON-PROTON COLLISIONS          ECM = .540E+03 GEV
*
*
LAMBDA = .100E+00 GEV  PT CUTOFF = .400E+01 GEV  MEAN KT**2 = .400E+00 GEV**2  INEL. X-SECT. = .400E+02 MB
*
NO. OF PRE-CASCADES =      333  (PER PARTON CHANNEL -- .LT. 10. PER CENT OF EXEC TIME)
*
QCD X-SECT. =      .4030E-01 OF INEL. X-SECT.
*
UNWEIGHTED EVENTS
*
NO. OF (FINAL) FLAVORS  3
*
TOTAL NO. OF EVENTS    2131    NO. OF HARD EVENTS    26
*
*
**UTILIZATION OF ARRAY DIMENSIONS**
MAX UTILIZATION IN THIS JOB OF THE 1ST DIMENSION OF THE INDICATED ARRAYS IS REPORTED
( THESE DIMENSIONS CAN BE CHANGED TOGETHER WITH PARAMETER IN LAST BRACKET ACCORDING TO NEEDS)
PARHAD( ,7) (/EVENT/), TMV( ), YV( ) (SUB. KNO)          UP TO 232 (CURR. DIM. MXPART= 401)
PARQUA( ,7) (/PARQUA/)          UP TO 7 (CURR. DIM. MAXQUA= 79)
K( ,2), P( ,5) (/JET/), IHEV( ) (/SHEAVY/), TM2V( ) (SUB P4CORR) UP TO 259 (CURR. DIM. MAXSJO= 399)
PARACT( ,7,2) (/PARQ/), NBKCNC( ,2) (/CANC/), NENT2( ) (SUB. MATCH) UP TO 13 (CURR. DIM. MXACT = 47)
PARSPE( ,6,2) (/PARQ/)          UP TO 7 (CURR. DIM. MXSPE = 39)
CUE( ,8,2) (SUB. EVOL)          UP TO 2 (CURR. DIM. -- SEE MXNCUE IN EVOL)
CUE( ,5) (SUB. JETQCD)          UP TO 0 (CURR. DIM. -- SEE MXNCUE IN JETQCD)
*****

```

```

COJETS.T37.
ACCOUNT CARD
FIND,LIB1,PROPACKLIB,IO=PROGLIB.
FIND,NEWPROD,IO=OPSYSTEM.
LIBRARY,LIB1.
BEGIN,NEWPROD.
FTNS,DB=0/PHD,PL=20000,LO=0,GO.

PROGRAM COJETS(OUTPUT)
*****
C- VERSION 2.2 - 7 NOV 83
C- PROGRAM SIMULATING JET PRODUCTION (INCLUDING HEAVY FLAVORS) ACCORDING TO QCD IN ANTI-PROTON-PROTON AND PROTON-PROTON COLLISIONS, WITH
C- INCLUSION OF QCD RADIATION OFF INITIAL AND FINAL PARTONS
C- OUTPUT COMMON BLOCKS AND INPUT PARAMETERS ARE SPECIFIED BELOW
C-
C- REFERENCE - R. ODORICO NUCL. PHYS. B228 (1983) 381, CERN-TH-3744 (1983)
C- , CERN-TH-3760 (1983) (LONG WRITE UP)
C- FOR QUESTIONS ABOUT THIS PROGRAM CONTACT:
C- R. ODORICO, IST. FISICA, 40126 BOLOGNA, VIA IRNERIO 46, TEL. (51)
C- 26.09.91 (SWITCHBOARD), 26.62.39 (ROOM), TELEX 21.16.64 INFNBO I
C-
C- OUTPUT COMMON BLOCKS -- QUANTA IN /PARQUA/, HADRONS IN /EVENT/ :
C-
C- /PARQUA/
C- NQUA = NO. OF FINAL QUANTA -- PARQUA(I,J), I=1,NQUA, J=1 (PX),
C- 2 (PY), 3 (PZ), 4 (ENERGY), 5 (MASS, GIVEN BY CUTOFF), 6 (FLAVOR
C- INDEX, FOR QUARKS 1-6 CORRESPONDING TO U, D, S, C, B, T, IN ORDER --
C- NEGATIVE FOR ANTIQUARKS -- 0 FOR GLUONS), 7 (ORIGIN: -1 INITIAL
C- BREMSSTRAHLUNG, 1 OTHERWISE)
C- MAXQUA = DIAGNOSTICS ISSUED IF NQUA.GT.MAXQUA
C-
C- /EVENT/
C- NEVENT = EVENT NUMBER -- NPART = NO. OF PARTICLES -- PARHAD(I,J),
C- I=1,NPART, J=1,2,3,4,5 (AS FOR QUANTA), 6 (PARTICLE TYPE, INTEGER,
C- ACCORDING TO TABLE AT P. 33 OF T. SJOSTRAND, UNIV. LUND PREPRINT
C- LU TP 79-8 (1979), PLUS MODIFICATIONS -- TO HAVE THE FULL TABLE
C- PRINTED JUST CALL PARTAB), 7 (NO. OF ORIGINATING QUANTUM TIMES
C- PARQUA(I,7) -- 0 FOR SPECTATOR JET HADRONS) ---- WEIGHT = EVENT
C- WEIGHT IF WEIGHTED EVENTS ARE GENERATED, =1 IF EVENTS ARE UNWEIGHTED
C- POSITIVE Z AXIS DEFINED BY INITIAL ANTI-PROTON DIRECTION
C- HXPART = DIAGNOSTICS ISSUED IF NPART.GT.HXPART
C-
C- /PTHARD/
C- PTHARD = PARTON PT GENERATED IN THE HARD SCATTERING -- SET =1.E-10
C- FOR PURE SPECTATOR JET EVENTS
COMMON/PARQUA/PARQUA(79,7),NQUA,MAXQUA
COMMON/NEVENT/NEVENT,ECH,PARHAD(401,7),NPART,HXPART,WEIGHT
COMMON/PTHARD/PTHARD
C-
C- COMMON BLOCKS TRANSFERRING INPUT PARAMETERS (IF DEFINED IN
C- MAIN PROGRAM TO OVERRIDE DEFAULT VALUES)
COMMON/ALQGEN/ALQGEN(127),HGBALQ(127,5,3),Q2VECT(128),NALQ,DALQ
1 ,ALAMB,ALLAM2,PTHIN,PTHAX,PTZINT,QZEV,ALQZM,ALQI,ALQF,YF
2 ,IPBAR,NCASC,SIGINL,PROPT,IPTMGT,PROPTH,PTMGTH,FRHARD,FRAGEN,FRAT
3 ,ZHCUT,ZLN,ZUP,XMIN
COMMON/FORGEN/PTMGE,MOGE,SIGHRD
COMMON/NFLAV/NFLAV,KFRFLA
COMMON/TOP/TOPMAS
COMMON/TLEAVE/TLEAVE
COMMON/FRATAB/FRATAB
COMMON/KDUMP/KDUMP
COMMON/STABLE/IDECAY(110)
C-
C- INPUT PARAMETERS -- TO BE SET BEFORE ALL ROUTINE CALLS
C-
C- ECH = C.M. ENERGY (GEV) (DEFAULT 540.)
C- ALAMB = QCD LAMBDA (GEV) (DEFAULT 0.1)
C- PTHIN = MIN PT CUTOFF FOR HARD PARTONS (GEV/C) (DEFAULT 4.)
C- PTMGE = IF PTMGE.GT.PTHIN GENERATE ONLY HARD EVENTS WITH PT.GT.PTHMGE
C- (DEFAULT 0.)
C- PTZINT = MEAN KTY*2 OF INITIAL PARTONS, BEFORE EVOLUTION ((GEV/C)**2)
C- (DEFAULT 0.4)
C- SIGINL = TOTAL INELASTIC X-SECTION (MB) (DEFAULT 40.)
C- IPBAR = 1 ANTI-PROTON-PROTON, OTHERWISE PROTON-PROTON (DEFAULT 1)
C- NFLAV = NO. OF QUARK FLAVORS (FINAL STATE -- FUSION ONLY) (DEFAULT 3)
C- TOPMAS = MASS OF TOP QUARK (GEV) (DEFAULT 35.)
C-
C- TLEAVE = EVENT GENERATION STOPS WHEN TIME LEFT .LT. TLEAVE (SEC)
C- (DEFAULT 5.)
C- KDUMP = NO. OF EVENTS TO DUMP (DEFAULT 5)
C- FRATAB = FRACTION OF EXECUTION TIME RESERVED TO PRETABULATION
C- (DEFAULT AND MINIMUM VALUES 0.1 NORMALLY, 0.3 FOR FORCED
C- GENERATION OF HEAVY FLAVORS)

```



```

F2QG(2)=F2IN      000529      COMMON/OVER/KOVER,SPYOV      000617
RETURN            000530      COMMON/TOP/TOPTAS      000618
C-- D             000531      REAL CMFLA(2,3),CMFLA2(3)      000619
30 F2IN=AD*SQRT(X)*(1.-X)**4.3+AS*(1.-X)**7.7      000532      C      000620
F2QG(3)=F2IN      000533      DATA CMFLA/4MCHAR,1MM,4MBOTT,2MOM,3HTOP,1N /      000621
RETURN            000534      DATA CMFLA2/4MC-CB,4MB-BB,4MT-TB/      000622
C-- DB            000535      C      000623
40 F2IN=AS*(1.-X)**7.7      000536      PRINT 1      000624
F2QG(4)=F2IN      000537      PRINT 2      000625
RETURN            000538      PRINT 26      000626
C-- S             000539      PRINT 5      000627
50 F2IN=AS*(1.-X)**7.7      000540      PRINT 11      000628
F2QG(5)=F2IN      000541      PRINT 2      000629
RETURN            000542      PRINT 3      000630
C-- SB            000543      PRINT 3      000631
60 XU=(1.-X)**3.3      000544      IF(IPBAR.EQ.1) PRINT 18,ECM      000632
F2IN=SQRT(X)*(AU*XU+AD*XU*(1.-X))+5.*AS*(1.-X)**7.7      000545      IF(IPBAR.NE.1) PRINT 19,ECM      000633
F2QG(6)=F2IN      000546      PRINT 3      000634
RETURN            000547      PRINT 21,ALAMB,PTMIN,PT2INT,SIGINL      000635
C-- GLUON         000548      FRATX=FRAT*100.      000636
70 F2IN=FGLU(X)      000549      PRINT 25,NCASC,FRATX      000637
F2QG(7)=F2IN      000550      IF(PTHGE,LE,PTMIN,AND,KFRFLA.EQ.0) PRINT 31,FRHARD      000638
RETURN            000551      IF(IPTMGT.EQ.0) PRINT 22      000639
END              000552      FRAGEX=FRAGEN*100.      000640
C                000553      IF(IPTMGT.EQ.1) PRINT 23,FRAGEX      000641
C                000554      IF(PTHGE,GT,PTMIN) PRINT 30,PTHGE,SIGHRO      000642
C                000555      PRINT 27,NFLAV      000643
C                000556      IF(KFRFLA.GT.0) PRINT 28,(CMFLA(1,KFRFLA-3),L=1,2)      000644
C                000557      IF(KFRFLA.GT.0) SIGHR6=SIGHRO*1.E6      000645
C                000558      IF(KFRFLA.GT.0) PRINT 29,CMFLA2(KFRFLA-3),SIGHR6      000646
C                000559      IF(KFRFLA.EQ.6) PRINT 32,TOPTAS      000647
C                000560      IF(SPYQ.GT.0.) PRINT 40      000648
C                000561      IF(SPYB.GT.0.) PRINT 44      000649
C-- GLUON MOMENTUM DISTRIBUTION      000562      IF(SPYT.GT.0.) PRINT 46      000650
COMMON/TABGRK/TABGRK(127,7),NBSIN,DBIN,F2QGV(7),MDQV(7),AGLU,HOBIN      000563      ITENT=INT(TEINTS)      000651
1 ,XVI(127),AL1,AL2,IFCODE(5)      000564      PRINT 24,NEVENT,ITENT      000652
2 ,TGLIAS(127),LGLIAS(127),TGLIAS(127),LGLIAS(127)      000565      1      000653
DATA EGLU/S./      000566      2      000654
FGLU=AGLU*(1.+EGLU)*(1.-X)**EGLU      000567      3      000655
RETURN            000568      4      000656
END              000569      5      000657
C                000570      18      000658
C                000571      19      000659
C-- SECTION (B)      000572      21      000660
C-- PRODUCTION DYNAMICS      000573      24      000661
C                000574      25      000662
C                000575      26      000663
C                000576      27      000664
C                000577      28      000665
C                000578      29      000666
C                000579      30      000667
C-- CALLED BY BKWEIG -- INITIAL WEIGHT ON PARTON DENSITIES      000580      31      000668
WPARON=X/(1.-X)**3      000581      32      000669
RETURN            000582      33      000670
END              000583      34      000671
C                000584      35      000672
C                000585      36      000673
C                000586      37      000674
C-- ORIGINAL QSQ (APPROX. EQ. TO PT**2 OF HARD PARTONS) PRETABULATION      000587      38      000675
C-- IS MULTIPLIED BY PTMGT(QSQ) IF IPTMGT=1      000588      39      000676
COMMON/ALQGEN/MGALQT(127),MGALQ(127,5,3),QZVECT(126),NALQ,DALQ      000589      40      000677
1 ,ALAMB,ALLAM2,PTMIN,PTMAX,PT2INT,QZEV,ALQZM,ALQI,ALQF,YF      000590      41      000678
2 ,IPBAR,NCASC,SIGINL,PROPT,IPTMGT,PROPTM,PTMGTH,FRHARD,FRAGEN,FRAT*000591      000679
3 ,ZHCUT,ZLN,ZUP,XMIN      000592      42      000680
PTMGT=QSQ**2*1.      000593      43      000681
RETURN            000594      44      000682
END              000595      45      000683
C                000596      46      000684
C-- SUBROUTINE PRHEAD      000597      47      000685
C                000598      48      000686
C-- PRINTS HEADING      000599      49      000687
COMMON/THETA/THETA,TEINTS,TEINTM      000600      50      000688
COMMON/THAXN/MHBOOK,MHBOOK,MHPART,MHQUA,MHQUE,MHQUEJ,MHNSJO      000601      51      000689
COMMON/CUTOFF/QZ,QZSR,QZMASS(6),QZFL(7),QZFLSQ(7),QTHRSQ(7)      000602      52      000690
1 ,NFLAVT,LGLU      000603      53      000691
COMMON/KEYBRE/KEYBRE      000604      54      000692
COMMON/PARA/PARACT(47,2),MXACT,IBEAH,NBOOK(2),JBOOK,MBOOK(2)      000605      55      000693
1 ,PARSPE(39,6,2),MXSPE,SHARD      000606      56      000694
COMMON/QUAQUA/QUAQUA(79,7),NQUA,MAXQUA      000607      57      000695
COMMON/NEVENT,ECM,PARHAD(401,7),NPART,MXPART,WEIGHT      000608      58      000696
COMMON/MAJSJO/MAJSJO      000609      59      000697
COMMON/NFLAV/NFLAV,KFRFLA      000610      60      000698
COMMON/ALQGEN/MGALQT(127),MGALQ(127,5,3),QZVECT(126),NALQ,DALQ      000611      61      000699
1 ,ALAMB,ALLAM2,PTMIN,PTMAX,PT2INT,QZEV,ALQZM,ALQI,ALQF,YF      000612      62      000700
2 ,IPBAR,NCASC,SIGINL,PROPT,IPTMGT,PROPTM,PTMGTH,FRHARD,FRAGEN,FRAT*000613      000701
3 ,ZHCUT,ZLN,ZUP,XMIN      000614      63      000702
COMMON/FORGEN/PTHGE,MGGE,SIGHRO      000615      64      000703
COMMON/SPYFOR/SPYC,SPYB,SPYT,SPYQ,SPYCEL,SPYBEL,SPYTEL      000616      65      000704

```

```

49  FORMAT(T2,1H*,48H****DIMENSION OVERFLOW HAS OCCURRED (NOT COUNTED,000705
162H IN STATISTICS BELOW) -- LOOK FOR DIAGNOSTICS BEFORE THIS PRIN,000706
28HTING****      ,T121,1H*/T2,1H*,T121,1H*) 000707
PRINT 50 000708
50  FORMAT(T2,1H*,1X,35H**UTILIZATION OF ARRAY DIMENSIONS** ,T121, 000709
11H*/T2,1H*,1X,48H**MAX UTILIZATION IN THIS JOB OF THE 1ST DIMENSION,000710
236H OF THE INDICATED ARRAYS IS REPORTED ,T121,1H*, 000711
1 /T2,1H*,1X,45H**THESE DIMENSIONS CAN BE CHANGED TOGETHER WIT,000712
447H PARAMETER IN LAST BRACKET ACCORDING TO NEEDS) ,T121,1H*)000713
PRINT 52,MNPART,MXPART,MNQQA,MAXQUA,MNSJD,MAXSJD,MNBOOK,MXACT 000714
1,MNBOOK,MXSPE 000715
52  FORMAT( 000716
1 T2,1H*,1X,50HPARHADI(.7) (/EVENT/), TMV( ) ,YV( ) (SUB. KNO) 000717
2 ,T72,5HUP TO,I6,1X,19H(CURR. DIM. MXPART=,I6,1H),T121,1H* 000718
3 /T2,1H*,1X,22HPARQUA(.7) (/PARQUA/) 000719
4 ,T72,5HUP TO,I6,1X,19H(CURR. DIM. MAXQUA=,I6,1H),T121,1H* 000720
5 /T2,1H*,1X,42HKK(.2), P(.5) (/JET/), IMEV( ) (/SHEAVY/),000721
622H, TM2V( ) (SUB P4CRR) 000722
7 ,T72,5HUP TO,I6,1X,19H(CURR. DIM. MAXSJD=,I6,1H),T121,000723
81H* /T2,1H*,1X,37HPARACT(.7,2) (/PARQ/), N8KNC(.2) (.000724
930H/CANC/), NENT2( ) (SUB. MATCH) 000725
1 ,T72,5HUP TO,I6,1X,19H(CURR. DIM. MXACT =,I6,1H),000726
2T121,1H* /T2,1H*,1X,22HPARSPE(.6,2) (/PARQ/) 000727
3 ,T72,5HUP TO,I6,1X,19H(CURR. DIM. MXSPE =,I6,000728
41H),T121,1H* ) 000729
PRINT 53,MNCUE,MNCUEJ 000730
53  FORMAT( 000731
1 T2,1H*,1X,22HCUE(.8,2) (SUB. EVOL) 000732
2 ,T72,5HUP TO,I6,1X,34H(CURR. DIM. -- SEE MNCUE IN EVOL),T121, 000733
31H* /T2,1H*,1X,22HCUE(.5) (SUB. JETQCO) 000734
4 ,T72,5HUP TO,I6,1X,36H(CURR. DIM. -- SEE MNCUE IN JETQCO),000735
5T121,1H* ) 000736
C 000737
PRINT 3 000738
PRINT 4 000739
C 000740
RETURN 000741
END 000742
C 000743
SUBROUTINE FUSION(HFLA) 000744
***** 000745
C-- FORCES FUSION PRODUCTION OF FLAVOR HFLA (=IHC,IHB,IHT) 000746
COMMON/NFLAV/NFLAV,KFRFLA 000747
COMMON/FLAPRE/FLAPRE 000748
C 000749
DATA HC,HB,HT,IHC,IHB,IHT/ 000750
C 000751
IF(FLAPRE.GT.0) GO TO 600 000752
C 000753
KFRFLA=0 000754
IF(HFLA.EQ.HC) KFRFLA=4 000755
IF(HFLA.EQ.HB) KFRFLA=5 000756
IF(HFLA.EQ.HT) KFRFLA=6 000757
IF(KFRFLA.EQ.0) GO TO 500 000758
NFLAV=KFRFLA 000759
RETURN 000760
C 000761
C-- ABNORMAL EXIT 000762
500 PRINT 501,HFLA 000763
501 FORMAT(5(/),1X,A4,40H IS INCORRECT ARGUMENT IN CALL TO FUSION 000764
1 //1X,47HACCEPTED ARGUMENTS ARE C,B AND T (IN HOLLERITH) 000765
2 //1X,22HCALL TO FUSION IGNORED ) 000766
RETURN 000767
600 PRINT 601 000768
601 FORMAT(5(/),1X,47HCALL TO FUSION INCORRECTLY PLACED AFTER CALL TO,000769
17H PREGEN , 000770
2 //1X,20HEXECUTION TERMINATED ) 000771
STOP 000772
END 000773
C 000774
SUBROUTINE PREGEN 000775
***** 000776
C-- HANDLES PREPARATION OF TABLES FOR HARD PT GENERATION 000777
COMMON/QCDS/CF,CA,BALPH,PI,P12,PROGL,PROFLA(2),NFLAVS,LGLUS,IVAL 000778
COMMON/PARG/PARG(47,2),MXACT,IBEAM,NBOOK(2),JBOOK,MBOOK(2) 000779
1 ,PARSPE(39,6,2),MXSPE,SHARD 000780
COMMON/ALQGEN/MGALQT(127),MGALQ(127,5,3),Q2VECT(128),NALQ,DALQ 000781
1 ,ALAMB,ALLAM2,PTHIN,PTMAX,PT2INT,QZEV,ALQZH,ALQI,ALQF,YF 000782
2 ,IPBAR,NCASC,SIGINL,PROPT,IPTHGT,PROPTM,PTMGTH,FRHARD,FRAGEN,FRAT000783
3 ,ZHCUT,ZLM,ZUP,XMIN 000784
COMMON/FORGEN/PTMGE,MGE,SIGHRD 000785
COMMON/LEAVE/TLEAVE 000786
COMMON/INMAT/IFLAIN(2),XIN(2),PXIN(2),PYIN(2),IMATCH(4,4) 000787
1 ,ICHA(2,3),IMGV(3) 000788
COMMON/BKMG/BKMG,FBKMG(7) 000789
COMMON/ENTREV/S,QSQ,YQSQ,IPACH,XING(2),XMING(2) 000790
1 ,PXING(2),PYING(2),IFLING(2),ICING(2) 000791
COMMON/EVENT/NEVENT,ECM,PARHADI(401,7),NPART,MXPART,WEIGHT 000792
COMMON/NFLAV/NFLAV,KFRFLA 000793
COMMON/STABLE/IDECAY(110) 000794
COMMON/DECPAR/PUDD,PSID,CND1,CND2,CND3,IDB(110) 000795
COMMON/THRFLA/THRFLA(6),AM2HEV(6) 000796
COMMON/DATA1/MESQ(36),CMIX(6,2),PHAS(110),QHAS(6) 000797
COMMON/TWIGH/TWIGH,TENTRS,TWEIMN 000798
COMMON/TRANS/ALAMBT,NFLATR,LGLUT,QSQMKT 000799
COMMON/NEVOL/NEVOL,ISTOP(2),IBACK(2),MBOOKS(2) 000800
COMMON/KEYBRE/KEYBRE 000801
COMMON/FRATAB/FRATAB 000802
COMMON/FLAPRE/FLAPRE 000803
REAL WGIN(2) 000804
C-- INITIALIZATION 000805
FLAPRE=1 000806
CALL SBLOCK 000807
S=ECH**2 000808
LGLUS=7 000809
IVAL=100 000810
ICHA(1,1)=IVAL 000811
ICHA(2,1)=IVAL 000812
ICHA(1,2)=LGLUS 000813
ICHA(2,2)=IVAL 000814
ICHA(1,3)=LGLUS 000815
ICHA(2,3)=LGLUS 000816
IMGV(1)=1 000817
IMGV(2)=4 000818
IMGV(3)=5 000819
NEVENT=0 000820
JBOOK=1 000821
PTMAX=ECM*.45 000822
ALLAM2=ALOG(ALAMB**2) 000823
NALQ=127 000824
ALQZH=ALOG(QZEV/ALAMB**2) 000825
ALQI=(ALOG(PTHIN**2)-ALLAM2)/ALQZH 000826
IF(ALQI.LT.1.) GO TO 500 000827
ALQF=(ALOG(PTMAX**2)-ALLAM2)/ALQZH 000828
YF=ALOG(ALQF) 000829
DALQ=(ALQF-ALQI)/FLOAT(NALQ) 000830
Q2VECT(1)=PTHIN**2 000831
DO 5 I=1,NALQ 000832
I1=I+1 000833
Q2VECT(I1)=EXP((ALQI+DALQ*FLOAT(I))*ALQZH+ALLAM2) 000834
C-- FORCED PT GENERATION -- INITIALIZATION 000835
IF(PTMGE.LE.PTHIN) GO TO 8 000836
IF(PTMGE.GT.PTHAX) GO TO 700 000837
ALQGE=(ALOG(PTMGE**2)-ALLAM2)/ALQZH 000838
IGE1=(ALQGE-ALQI)/DALQ*1. 000839
IGE2=IGE1+1 000840
ALQGE=ALQI+DALQ*FLOAT(IGE1) 000841
ALQGE=ALQI+DALQ*FLOAT(IGE2) 000842
IPTNGT=0 000843
GO TO 9 000844
8 CONTINUE 000845
C-- FORCED FLAVOR GENERATION -- INITIALIZATION 000846
IF(KFRFLA.LT.4) GO TO 9 000847
IGE1=1 000848
IGE2=2 000849
ALQGE=ALQI+DALQ*FLOAT(IGE1) 000850
ALQGE=ALQI+DALQ*FLOAT(IGE2) 000851
CONTINUE 000852
ZHCUT=1. 000853
ZLM=ZHCUT/ECM 000854
ZUP=1.-ZLM 000855
DO 3 L=4,6 000856
AM2HEV(L)=(QHAS(L)/1000.*.5)**2 000857
THRFLA(L)=4.*AM2HEV(L) 000858
XMIN=(3.*PTHIN**2-5.*PT2INT)/S 000859
IF(KFRFLA.GT.0) XMIN=(THRFLA(KFRFLA)-5.*PT2INT)/S 000860
IF(PTMGE.GT.PTHIN) XMIN=(3.*PTMGE**2-5.*PT2INT)/S 000861
XMIN=AMAX(XMIN,ZLM) 000862
CALL INQCD 000863
C 000864
ALAMBT=ALAMB 000865
NFLATR=NFLAV 000866
QSQMKT=ECH**2 000867
CALL INQCD 000868
C 000869
C-- MAIN LOOP 000870
C 000871
DO 6 J1=1,3 000872
DO 6 J2=1,5 000873
DO 6 J3=1,NALQ 000874
MGALQ(J3,J2,J1)=0. 000875
DO 7 N=1,NALQ 000876
MGALQT(N)=0. 000877
000878
000879
000880

```



```

10  DO 10 IX=1,NBIN
    MGG=MGG+TABGRK(IX,6)
    AGLIAS=0.
    DO 3 IX=1,NBIN
      TGLIAS(IX)=TABGRK(IX,6)*BKWEIG(XV(IX))
    3  AGLIAS=AGLIAS+TGLIAS(IX)
    DO 6 IX=1,NBIN
      TGLIAS(IX)=TGLIAS(IX)/AGLIAS
    6  CALL ALIAS(LGLIAS,TGLIAS,NBIN)
C-- GLUONS
    IFUN=6
    AGLU=1.-ASIMP(0.,1.,.001,M,2,FUN2)
    TABGRK(1,7)=0.
    DO 11 IX=2,NBIN
      X1=DBIN*FLOAT(IX-2)
      X2=X1+DBIN
      IF(X2.LE.XMIN) GO TO 12
      X1=AMAX1(X1,XMIN)
      TABGRK(IX,7)=ASIMP(ALOG(X1),ALOG(X2),.001,M,2,FUN3)
    12 GO TO 11
    11 CONTINUE
    MGG=0.
    DO 13 IX=1,NBIN
      MGG=MGG+TABGRK(IX,7)
    13 AGLIAS=0.
    DO 5 IX=1,NBIN
      TGLIAS(IX)=TABGRK(IX,7)*BKWEIG(XV(IX))
    5  AGLIAS=AGLIAS+TGLIAS(IX)
    DO 7 IX=1,NBIN
      TGLIAS(IX)=TGLIAS(IX)/AGLIAS
    7  CALL ALIAS(LGLIAS,TGLIAS,NBIN)
C
    DO 21 J=1,7
      TBKMG(1,J)=0.
      TBKMG(2,J)=0.
    DO 22 IX=1,NBIN
      TBKMG(1,J)=TBKMG(1,J)+TABGRK(IX,J)*BKWEIG(XV(IX))
      TBKMG(2,J)=TBKMG(2,J)+TABGRK(IX,J)
    22 CONTINUE
    FBKMG(J)=TBKMG(1,J)/TBKMG(2,J)
    21 CONTINUE
C
    AL1=ALOG(XMIN)
    AL2=ALOG(XVMIN)
    DO 14 J=1,7
      J1=J
      F2XQV(J)=F2ZIN(XMIN,J1)
    14 CONTINUE
    50 C-- GENERATION OF PARTON INITIATING INITIAL CASCADE - X AND FLAVOR FIRST
    IF(ICH.EQ.LGLUS) GO TO 200
C-- QUARK
    IX=RNDRM(0)*FLOAT(NBIN)+1.
    IF(RNDRM(0).GT.TGLIAS(IX)) IX=LGLIAS(IX)
    X=DBIN*FLOAT(IX-1)+HOBIN
    IF(X.GE.DBIN) GO TO 110
    X=EXP(RNDRM(0)*(AL2-AL1)+AL1)
    IF(RNDRM(0).GT.F2ZIN(X,6)/F2XQV(6)) GO TO 105
    105 CONTINUE
    CALL GEFLA(IX,IPLA,JFLA)
    BKMG=BKWEIG(X)
    MG=MGG/BKMG+FBKMG(JFLA)
    GO TO 300
C-- GLUON
    200 CONTINUE
    IX=RNDRM(0)*FLOAT(NBIN)+1.
    IF(RNDRM(0).GT.TGLIAS(IX)) IX=LGLIAS(IX)
    X=DBIN*FLOAT(IX-1)+HOBIN
    IF(X.GE.DBIN) GO TO 210
    X=EXP(RNDRM(0)*(AL2-AL1)+AL1)
    IF(RNDRM(0).GT.FGLU(X)/AGLU) GO TO 205
    210 CONTINUE
    IFLA=LGLUS
    BKMG=BKWEIG(X)
    MG=MGG/BKMG+FBKMG(7)
    JFLA=7
    300 CONTINUE
C-- TRANSVERSE MOMENTUM
    310 PT2=-PT2INT*ALOG(RNDRM(0))
    IF(PT2/(X*9).GT.X) GO TO 310
    PT=SQRT(PT2)
    PHI=PI2*RNDRM(0)
    PX=PT*COS(PHI)
    PY=PT*SIN(PHI)
    RETURN
    END
C
001057 FUNCTION BKWEIG(X)
001058 *****
001059 C-- ORGANIZATION OF BLOCK HEIGHT ON PARTON DENSITIES (IN PRETABULATION)
001060 XN=AMT(X/.1)*.1+.05
001061 BKWEIG=MPARDN(XIN)
001062 RETURN
001063 END
001064 C
001065 FUNCTION FUN1(ALX)
001066 *****
001067 C-- USED BY INPART
001068 COMMON/F2QG/F2QG(7)
001069 COMMON/CFUN1/IFUN
001070 X=EXP(ALX)
001071 FUN1=F2ZIN(X,IFUN)
001072 RETURN
001073 END
001074 C
001075 FUNCTION FUN2(X)
001076 *****
001077 C-- USED BY INPART
001078 COMMON/F2QG/F2QG(7)
001079 COMMON/CFUN1/IFUN
001080 FUN2=F2ZIN(X,IFUN)
001081 RETURN
001082 END
001083 C
001084 FUNCTION FUN3(ALX)
001085 *****
001086 C-- USED BY INPART
001087 X=EXP(ALX)
001088 FUN3=FGLU(X)
001089 RETURN
001090 END
001091 C
001092 SUBROUTINE GEFLA(IX,IPLA,JFLA)
001093 *****
001094 C-- GENERATION OF INITIAL PARTON FLAVOR - CALLED BY INPART
001095 COMMON/TABGRK/TABGRK(127,7),NBIN,DBIN,F2XQV(7),MGGV(7),AGLU,HOBIN
001096 1 ,XV(127),AL1,AL2,IFCODE(5)
001097 2 ,TGLIAS(127),LGLIAS(127),TGLIAS(127),LGLIAS(127)
001098 COMMON/BKMG/BKMG,FBKMG(7)
001099 TF=0.
001100 RTF=RNDRM(0)*TABGRK(IX,6)
001101 DO 115 J=1,5
001102 TF=TF+TABGRK(IX,J)
001103 JOUT=J
001104 IF(RTF.LT.TF) GO TO 116
001105 115 CONTINUE
001106 IFLA=IFCODE(JOUT)
001107 JFLA=JOUT
001108 IF(IFLA.EQ.3.AND.RNDRM(0).GT..5) IFLA=IFLA
001109 RETURN
001110 END
001111 C
001112 SUBROUTINE ALIAS(L,NPTS)
001113 *****
001114 C-- ORGANIZES ARRAYS FOR ALIAS METHOD GENERATION
001115 REAL S(250)
001116 INTEGER L(250),ST(250)
001117 C INPUT TO SUBROUTINE: ARRAY OF S(I) I=1,...,NPTS
001118 C OUTPUT: SAME ARRAY S(I) AND L(I)
001119 C ST ARRAY USED TO STACK INITIAL S(I).LT.1 (FORWARD)
001120 C AND S(I).GT.1 (BACKWARD)
001121 C ONE PLUS AND MINUS EPSILON
001122 ONEP=1.-1.E-5
001123 ONEM=1.+1.E-5
001124 C INITIALIZE
001125 IG=NPTS+1
001126 IS=0
001127 DO 1 I=1,NPTS
001128 S(I)=FLOAT(NPTS)*S(I)
001129 IF(S(I)-1.) 2,1,5
001130 IG=IG-1
001131 ST(IG)=I
001132 GO TO 1
001133 IS=IS+1
001134 ST(IS)=I
001135 1 CONTINUE
001136 C MAIN LOOP
001137 3 IF(IS.EQ.0) RETURN
001138 IX=ST(IS)
001139 IY=ST(IG)
001140 L(IY)=IY
001141 S(IY)=S(IY)-1.+S(IX)
001142 IF(IG.EQ.NPTS.AND.IS.EQ.1) RETURN
001143 IS=IS-1
001144 C CONSTANT REALLY ONE, CHANGED TO ACCOUNT FOR ROUND OFF

```

```

IF(S(IY),LT,ONE) GO TO 7
IF(S(IY),LT,ONEP) GO TO 8
GO TO 3
7
IS=IS+1
ST(IS)=ST(IG)
8
IG=IG+1
GO TO 3
END
C
SUBROUTINE MATCHING,MGTCS)
*****
C-- MATCHES PARTONS FROM INITIAL CASCADES (CALLED BY PREGEN)
COMMON/DCS/CF,CA,BAUPH,PI,P12,PROGL,PROFLA(2),NFLAVS,UGLUS,IYAL
COMMON/PARG/PARACT(647,7,2),HFACT,IBEAM,NBOOK(2),JBOOK,NBOOK(2)
1
PARSPC(33,6,2),HXSPE,SHARD
COMMON/ALQEN/MGALQ(127),MGALQ(127,5,3),Q2VECT(128),NALQ,DALQ
1
ALAMB,ALLAH2,PTMIN,PTMAX,PTZINT,QZEY,ALQZN,ALQI,ALRF,YF
2
IPBAR,NCASC,SIGDNL,PROPT,IPDTGT,PROPTM,PTMGTH,FRHARD,FRAGEN,FRAT001250
3
ZHCUT,ZUH,ZUP,XMIN
COMMON/INM/IFLAIN(2),XIN(2),PXIN(2),PYIN(2),INATCH(4,4)
1
ICHA(2,3),IMGV(3)
COMMON/BKAG/BKAG,FBKMG(7)
COMMON/ENTREV/S,QSQ,YQSQ,IPACH,XING(2),XYING(2)
1
PXING(2),PYING(2),IFLING(2),ICHING(2)
COMMON/EVENT/NEVENT,ECM,PARHAD(401,7),NPART,HXPART,WEIGHT
COMMON/CANC/ICANC(2),NBKCN(47,2)
COMMON/NEVOL/NEVOL,ISTOP(2),IBACK(2),NBOOKS(2)
COMMON/HAXO/HBOOK,NBOOK,NPART,NKQA,NKCU,NKUEJ,NMSJO
COMMON/NFLA/NFLA,KFRFLA
COMMON/THRFLA/THRFLA(6),AM2HEV(6)
REAL WFSOT(6)
INTEGER NENT2(47)
C
C-- PREENTERING
DO 10 IB=1,2
IBEAM=IB
NBOOK(IBEAM)=0
ICANC(IBEAM)=0
CALL Q2GEN(IFLAIN(IB),0.,Y,QSQ)
CALL BOOK(IFLAIN(IB),0,0.,Y,XIN(IB),PXIN(IB),PYIN(IB))
10
CONTINUE
C
C-- MAIN LOOP
DO 200 M=1,NALQ
YQSQ=ALQI+DALQ*FLOAT(M)
NENTR=0
DO 100 IB=1,2
IBEAM=IB
NB=NBOOK(IB)
ICANC(IB)=0
C-- EVOLVE ONE M STEP FURTHER
DO 90 J=1,NB
Y=PARACT(J,4,IB)
IF(Y.GT.YQSQ) GO TO 90
JX=J
IBACK(IBEAM)=0
CALL EVOL(JX,YQSQ)
NBOOK=HAXO(NBOOK,NBOOK(IB))
90
CONTINUE
IF(ICANC(IB).EQ.0) GO TO 100
C-- CANCEL DECAYED QUANTA
ICN=ICANC(IB)
DO 91 I=1,ICN
IN=NBKCN(I,IB)
NBK=NBOOK(IB)
DO 92 L=1,7
PARACT(IN,L,IB)=PARACT(NBK,L,IB)
NBOOK(IB)=NBOOK(IB)-1
91
CONTINUE
ICANC(IB)=0
100
CONTINUE
NB1=NBOOK(1)
NB2=NBOOK(2)
IF(NB1.EQ.0.OR.NB2.EQ.0) RETURN
C-- CALCULATION OF HARD WEIGHT AND BOOKING
DO 110 J1=1,NB1
IFLAI=PARACT(J1,1,1)
XN1=PARACT(J1,3,1)
XP1=PARACT(J1,5,1)
PX1=PARACT(J1,6,1)
PY1=PARACT(J1,7,1)
MFLAU=MINO(4,IABS(IFLAI))
NENT1=0
DO 120 J2=1,NB2
IFLAI2=PARACT(J2,1,2)
XN2=PARACT(J2,3,2)
XP2=PARACT(J2,5,2)
PX2=PARACT(J2,6,2)
001233
PY2=PARACT(J2,7,2)
001234
MFLAI2=MINO(4,IABS(IFLAI2))
001235
NENT2(J2)=0
001236
SHAT=(XP1*XP2+XN1*XN2)*S-2.*(PX1*PX2+PY1*PY2)
001237
IF(SHAT.LT.3.*Q2VECT(M)) GO TO 120
001238
IF(KFRFLA.LT.4) GO TO 126
001239
IF(SHAT.LT.THRFLA(KFRFLA)) GO TO 120
001240
126
CONTINUE
001241
IM=INATCH(MFLAI,MFLAI2)
001242
IF(IM.NE.2) GO TO 122
001243
IF(IFLAI2*IFLAI2.LT.0) IM=3
001244
122
CONTINUE
001245
IF(KFRFLA.LT.4) GO TO 127
001246
IF(IM.EQ.1) GO TO 120
001247
IF(IM.EQ.2) GO TO 120
001248
IF(IM.EQ.4) GO TO 120
001249
127
CONTINUE
001250
NENT1=1
001251
NENT2(J2)=1
001252
DSQ1=SQRT(1.-4./(1.+SHAT/Q2VECT(M)))
001253
T1=-SHAT*.5*(1.+DSQ1)
001254
U1=-SHAT*.5*(1.-DSQ1)
001255
IF(SHAT.LT.3.*Q2VECT(M+1)) GO TO 124
001256
DSQ2=SQRT(1.-4./(1.+SHAT/Q2VECT(M+1)))
001257
T2=-SHAT*.5*(1.+DSQ2)
001258
U2=-SHAT*.5*(1.-DSQ2)
001259
GO TO 125
001260
124
T2=-SHAT*.5
001261
U2=T2
001262
125
CONTINUE
001263
T=(T1+T2)*.5
001264
U=(U1+U2)*.5
001265
CALL OSDT(IM,SHAT,T,U,MT1,WFSOT)
001266
CALL OSDT(IM,SHAT,U,T,MT2,WFSOT)
001267
Q2M=(Q2VECT(M)+Q2VECT(M+1))**.5
001268
WGT=(MT1+MT2)
001269
I=PT/(BALPH*ALOG(Q2M/ALAMB*2)*SHAT)*2
001270
Z=(T2-T1)
001271
WGT=WGT*WGTCS
001272
MGALQ(M,IMG,1)=MGALQ(M,IMG,1)+WGT
001273
MGALQ(M,IMG,2)=MGALQ(M,IMG,2)+WGTCS*(T2-T1)
001274
120
CONTINUE
C-- KEEP RECORD OF QUANTA TOO SOFT TO PASS THRESHOLD
IF(NENT1.GT.0) GO TO 109
ICANC(1)=ICANC(1)+1
NBKCN(ICANC(1),1)=J1
GO TO 110
109
NENTR=1
110
CONTINUE
C
DO 130 J2=1,NB2
IF(NENT2(J2).GT.0) GO TO 130
ICANC(2)=ICANC(2)+1
NBKCN(ICANC(2),2)=J2
130
CONTINUE
IF(NENTR.GT.0) MGALQ(M,IMG,3)=MGALQ(M,IMG,3)+WGTCS*(T2-T1)
200
CONTINUE
C-- CANCEL QUANTA TOO SOFT TO PASS THRESHOLD
DO 250 IB=1,2
IF(ICANC(IB).EQ.0) GO TO 250
ICN=ICANC(IB)
DO 251 I=1,ICN
IN=NBKCN(I,IB)
NBK=NBOOK(IB)
DO 252 L=1,7
PARACT(IN,L,IB)=PARACT(NBK,L,IB)
NBOOK(IB)=NBOOK(IB)-1
251
CONTINUE
ICANC(IB)=0
250
CONTINUE
RETURN
END
C
SUBROUTINE OSDT(IN,S,T,U,MTOSOT,WFSOT)
*****
C-- SIGMA/DT OF HARD PROCESSES
COMMON/NFLA/NFLA,KFRFLA
COMMON/THRFLA/THRFLA(6),AM2HEV(6)
REAL WFSOT(6)
DO 1 L=1,6
WFSOT(L)=0.
MTOSOT=0.
IF(KFRFLA.GE.4) GO TO 100
S2=S*2
T2=T*2
U2=U*2
IF(IM.EQ.1) GO TO 10
001321
001322
001323
001324
001325
001326
001327
001328
001329
001330
001331
001332
001333
001334
001335
001336
001337
001338
001339
001340
001341
001342
001343
001344
001345
001346
001347
001348
001349
001350
001351
001352
001353
001354
001355
001356
001357
001358
001359
001360
001361
001362
001363
001364
001365
001366
001367
001368
001369
001370
001371
001372
001373
001374
001375
001376
001377
001378
001379
001380
001381
001382
001383
001384
001385
001386
001387
001388
001389
001390
001391
001392
001393
001394
001395
001396
001397
001398
001399
001400
001401
001402
001403
001404
001405
001406
001407
001408

```

```

IF(IM.EQ.2) GO TO 20
IF(IM.EQ.3) GO TO 30
IF(IM.EQ.4) GO TO 40
IF(IM.EQ.5) GO TO 50
C
C-- Q-Q --+ Q-Q NON-ID.
10 WFDSDT(1)=(4./9.)*(S2+U2)/T2
WTDSDT=WFDSDT(1)
RETURN
C
C-- Q-Q --+ Q-Q ID.
20 WFDSDT(1)=(4./9.)*(S2+U2)/T2+(S2+T2)/U2-(8./27.)*S2/(U*T)
WTDSDT=WFDSDT(1)
RETURN
C
C-- QB-Q --+ QB-Q ANNIH. F=I
30 WFDSDT(1)=(4./9.)*(S2+U2)/T2+(T2+U2)/S2-(8./27.)*U2/(S*T)
C-- QB-Q --+ QB-Q ANNIH. F=NE.I
WFDSDT(2)=(4./9.)*(T2+U2)/S2*2.
C-- QB-Q --+ G-G
WFDSDT(3)=(32./27.)*(U2+T2)/(U*T)-(8./3.)*(U2+T2)/S2
C-- QB-Q --+ HEAVY QUARKS
LF=3
DO 31 L=4,6
IF(NFLAV.LT.L) GO TO 35
IF(S.LE.THRFLA(L)) GO TO 35
LF=L
WFDSDT(L)=DSHVQ(AM2HEV(L),S,T,U)
31 CONTINUE
35 CONTINUE
WTDSDT=0.
DO 36 L=1,LF
36 WTDSDT=WTDSDT+WFDSDT(L)
RETURN
C
C-- Q-G --+ Q-G
40 WFDSDT(1)=(4./9.)*(U2+S2)/(U*S)+(U2+S2)/T2
WTDSDT=WFDSDT(1)
RETURN
C
C-- G-G --+ G-G
50 WFDSDT(1)=(9./2.)*(3.-U*T/S2-U*S/T2-S*T/U2)
C-- G-G --+ Q-QB
WFDSDT(2)=(1./6.)*(U2+T2)/(U*T)-(3./8.)*(U2+T2)/S2
WFDSDT(3)=WFDSDT(2)*3.
C-- G-G --+ HEAVY QUARKS
LF=2
DO 51 L=4,6
IF(NFLAV.LT.L) GO TO 55
IF(S.LE.THRFLA(L)) GO TO 55
LF=L
WFDSDT(L)=DSHVQ(AM2HEV(L),S,T,U)
51 CONTINUE
55 CONTINUE
WTDSDT=0.
DO 56 L=1,LF
56 WTDSDT=WTDSDT+WFDSDT(L)
RETURN
C
C-- FORCING PRODUCTION OF HEAVY FLAVOR KFRFLA
100 IF(S.LE.THRFLA(KFRFLA)) GO TO 200
IF(IM.EQ.1) GO TO 200
IF(IM.EQ.2) GO TO 200
IF(IM.EQ.3) GO TO 200
IF(IM.EQ.4) GO TO 200
IF(IM.EQ.5) GO TO 150
130 WFDSDT(KFRFLA)=DSHVQ(AM2HEV(KFRFLA),S,T,U)
WTDSDT=WFDSDT(KFRFLA)
RETURN
150 WFDSDT(KFRFLA)=DSHVQ(AM2HEV(KFRFLA),S,T,U)
WTDSDT=WFDSDT(KFRFLA)
RETURN
200 RETURN
END
C
FUNCTION DSHVQ(A2,S,T,U)
*****
C-- QB-Q --+ QHB-QH, A2=MASS**2 OF HEAVY QUARK
DSHVQ=(4./9.)*(A2-T)*(A2-U)**2+(A2-U)**2*2.*A2*S)/S**2
END
C
FUNCTION DSHVG(A2,S,T,U)
*****
C-- G-G --+ GHB-GH, A2=MASS**2 OF HEAVY QUARK
DSHVQ=(3./4.)*(A2-T)*(A2-U)/S**2
1+(1./6.)*(A2-T)*(A2-U)-2.*A2*(A2+T)/(A2-T)**2
2+(1./6.)*(A2-T)*(A2-U)-2.*A2*(A2+U)/(A2-U)**2
001409 3-(1./24.)*A2*(S-4.*A2)/(A2-T)*(A2-U)
001410 4-(3./8.)*(A2-T)*(A2-U)+A2*(U-T)/(S*(A2-T))
001411 5-(3./8.)*(A2-T)*(A2-U)+A2*(T-U)/(S*(A2-U))
001412 RETURN
001413 END
001414 C
001415 SUBROUTINE GEVENT
001416 *****
001417 C-- HANDLES EVENT GENERATION
001418 COMMON/QCDS/CF,CA,BALPH,PI,PI2,PROGL,PROFLA(2),NFLAYS,LGLUS,IVAL
001419 COMMON/PARQUA/PARQUA(79,7),NQUA,MAXQUA
001420 COMMON/PARQ/PARQ(47,7,2),MXACT,IBEAM,NBOOK(2),JBOOK,MBOOK(2)
001421 1,PARSPE(39,6,2),MXSPE,SHARD
001422 COMMON/ALQGEN/WSALQT(127),WALQT(127,5,3),Q2VECT(128),NALQ,OALQ
001423 1,ALAMB,ALLAM2,PTMIN,PTMAX,PT2INT,QZEV,ALQZN,ALQZ,ALQF,YF
001424 2,IPBAR,NCASC,SIGINL,PROPT,IPTMGT,PROPTM,PTWGTM,FRHARD,FRAGEN,FRATO
001425 3,ZHCUT,ZLU,ZUP,XMIN
001426 COMMON/ENTREV/S,QSQ,YGSQ,IPACH,XING(2),XMING(2)
001427 1,FXING(2),PYING(2),IFLING(2),ICHING(2)
001428 COMMON/EVENT/NEVENT,ECH,PARHAD(401,7),NPART,MXPART,WEIGHT
001429 COMMON/PTHARO/PTHARO
001430 COMMON/FORGEN/PTHARO,WGSE,SIGHRO
001431 COMMON/NFLAV/NFLAV,KFRFLA
001432 COMMON/THRFLA/THRFLA(6),AM2HEV(6)
001433 COMMON/TLEIGH/TLEIGH,TENTRS,TWEIHM
001434 COMMON/THMAT/THMAT(2),XINI(2),PXINI(2),PYINI(2),IMATCH(4,4)
001435 1,ICHA(2,3),INGVI(3)
001436 COMMON/NEVOL/NEVOL,ISTOP(2),ZBACK(2),MBOOKS(2)
001437 COMMON/TLEAVE/TLEAVE
001438 COMMON/HASH/MBOOK,MNPART,MNQUA,MNCUE,MNCUEJ,MNSJO
001439 COMMON/KEYBRE/KEYBRE
001440 COMMON/KOUM/KOUM
001441 COMMON/KHADRO/KHADRO
001442 REAL TV(2),UVI(2),HGV(2,6),WFDSDT(6),WTDSD(2)
001443 C
001444 700 IFIREF=0
001445 WEIGHT=1.
001446 C
001447 C-- DECIDE WHETHER IT IS A MINIBIAS (SMALL PT) EVENT OR NOT
001448 IF(PTMGE.GT.PTMIN) GO TO 3
001449 IF(KFRFLA.GT.0) GO TO 3
001450 IF(RNDM(0).GT.FRAGEN) GO TO 300
001451 3 CONTINUE
001452 C-- GENERATE HARD QSQ (X2STU/(S**2+T**2+U**2), ABOUT =PT**2)
001453 CALL GEQSQ(ALQ,ZALQ)
001454 QSQ=EXP(ALQ*ALQZM+ALLAM2)
001455 PTHARO=SQRT(QSQ)
001456 YQSQ=ALOG(ALQ)
001457 FACTM=PI/BALPH*ALOG(QSQ/ALAMB**2)**2
001458 IF(IPTMGT.EQ.0) GO TO 205
001459 WEIGHT=PTWGTM/PTWGT(QSQ)*WEIGHT
001460 205 CONTINUE
001461 NEIGOR=WEIGHT
001462 C-- GENERATE COLLISION CHANNEL
001463 IFIRE=0
001464 NITGE=0
001465 1 CONTINUE
001466 NITGE=NITGE+1
001467 RR=RNDM(0)
001468 DO 10 I=1,5
001469 IPACH=IW
001470 IF(RR.LT.WALQT(IALQ,IW,1)) GO TO 11
001471 10 CONTINUE
001472 11 CONTINUE
001473 C-- GENERATE PAIR OF INITIAL PARTONS
001474 CALL GEPAR
001475 C-- GENERATE PARTON CASCADE (LLA QCD)
001476 JBOOK=2
001477 NQUA=0
001478 DO 12 IB=1,2
001479 MBOOK(IB)=1
001480 MBOOK(IB)=0
001481 MBOOKS(IB)=0
001482 ISTOP(IB)=0
001483 IBACK(IB)=0
001484 IBEAM=1
001485 CALL QZGEN(IFLING(1),0.,YINI,QSQINI)
001486 CALL BOOK(IFLING(1),0.,YINI,XING(1),PXING(1),PYING(1))
001487 IBACK(IBEAM)=0
001488 100 CONTINUE
001489 IBEAM=1
001490 CALL EVOL(1,YGSQ)
001491 MIBOOK=MAX(MBOOK,MBOOK(1))
001492 IF(ISTOP(1).EQ.1) GO TO 200
001493 IBS=1
001494 MBOOK(2)=0
001495 IBEAM=2
001496 CALL QZGEN(IFLING(2),0.,YIN2,QSQIN2)
001497 001498
001499 001500
001501 001502
001502 001503
001503 001504
001504 001505
001505 001506
001506 001507
001507 001508
001508 001509
001509 001510
001510 001511
001511 001512
001512 001513
001513 001514
001514 001515
001515 001516
001516 001517
001517 001518
001518 001519
001519 001520
001520 001521
001521 001522
001522 001523
001523 001524
001524 001525
001525 001526
001526 001527
001527 001528
001528 001529
001529 001530
001530 001531
001531 001532
001532 001533
001533 001534
001534 001535
001535 001536
001536 001537
001537 001538
001538 001539
001539 001540
001540 001541
001541 001542
001542 001543
001543 001544
001544 001545
001545 001546
001546 001547
001547 001548
001548 001549
001549 001550
001550 001551
001551 001552
001552 001553
001553 001554
001554 001555
001555 001556
001556 001557
001557 001558
001558 001559
001559 001560
001560 001561
001561 001562
001562 001563
001563 001564
001564 001565
001565 001566
001566 001567
001567 001568
001568 001569
001569 001570
001570 001571
001571 001572
001572 001573
001573 001574
001574 001575
001575 001576
001576 001577
001577 001578
001578 001579
001579 001580
001580 001581
001581 001582
001582 001583
001583 001584

```

```

CALL BOOK(IFLNG(2),0.0.,YIN2,XING(2),PXING(2),PYING(2))
IBACK(IBEAM)=0
ISTOP(IBEAM)=0
150 CONTINUE
CALL EVOL(1,YQSQ)
MMBOOK=MAX0(MMBOOK,MBOOK(2))
IF(ISTOP(2).EQ.1) GO TO 100
C-- PREPARE TO ENTER HARD PROCESS
IFLAF1=PARACT(1,1,1)
XM1=PARACT(1,3,1)
XP1=PARACT(1,5,1)
PX1=PARACT(1,6,1)
PY1=PARACT(1,7,1)
MFLA1=MIN0(4,IABS(IFLAF1))
IFLAF2=PARACT(1,1,2)
XM2=PARACT(1,3,2)
XP2=PARACT(1,5,2)
PX2=PARACT(1,6,2)
PY2=PARACT(1,7,2)
SHAT=(XP1*XP2+XM1*XM2)*5-2.*(PX1*PX2+PY1*PY2)
IF(SHAT.LT.3.*QSQ) GO TO 150
IFLAF2=MIN0(4,IABS(IFLAF2))
IM=IMATCH(MFLA1,MFLA2)
IF(IM.NE.2) GO TO 212
IF(IFLAF1*IFLAF2.LT.0) IM=3
212 CONTINUE
IF(KFRFLA.LT.4) GO TO 218
IF(IM.NE.3.AND.IM.NE.5) GO TO 150
IF(SHAT.LE.THRFLA(KFRFLA)) GO TO 150
218 CONTINUE
C-- THRESHOLD CONDITION PASSED - CALCULATE MATRIX ELEMENTS
IFIRE=1
FACTS=FACTW/SHAT**2
DSQ=SQRT(1.-4./(1.+SHAT/QSQ))
TV(1)=-SHAT*.5*(1.+DSQ)
TV(2)=-SHAT*.5*(1.-DSQ)
UV(1)=-SHAT-TV(1)
UV(2)=-SHAT-TV(2)
CALL DSOT(IM,SHAT,TV(1),UV(1),WTDS(1),WFDSOT)
DO 216 L=1,6
216 WGV(L)=WFDSOT(L)
CALL DSOT(IM,SHAT,TV(2),UV(2),WTDS(2),WFDSOT)
DO 217 L=1,6
217 WGV(L)=WFDSOT(L)
WGT=(WTDS(1)+WTDS(2))*FACTS
RTU=WTDS(1)/(WTDS(1)+WTDS(2))
NIT=0
C-- HANDLE FLUCTUATIONS FROM MEAN WEIGHT (TO HAVE WEIGHT=1 EVENTS)
NLOOP=WGT/MGALQ(IALQ,IPACH,2)
210 IF(WGT.GT.MGALQ(IALQ,IPACH,2)) GO TO 211
IF(WGT.LT.RNDM(0)*MGALQ(IALQ,IPACH,2)) GO TO 150
211 CONTINUE
NIT=NIT+1
IF(NIT.GT.2*NLOOP) GO TO 150
CALL TIMELF(TLEFT)
IF(TLEFT.LT.TLEAVE) RETURN
WGT=WGT-MGALQ(IALQ,IPACH,2)
C-- Q.N. FINAL PARTONS
ITU=1
IF(RNDM(0).GT.RTU) ITU=2
IFLAF1=IFLAF1
IFLAF2=IFLAF2
IF(IM.NE.3.AND.IM.NE.5) GO TO 260
LF=KFRFLA
IF(KFRFLA.GE.4) GO TO 230
RW=RNDM(ITU)*WTDS(ITU)
WCHT=0
DO 221 L=1,6
221 LF=L
WCHT=WCHT+WGV(ITU,L)
IF(RW.LT.WCHT) GO TO 230
230 CONTINUE
IF(IM.EQ.5) GO TO 250
C-- Q-Q ANNIHILATION CHANNEL
IF(LF.EQ.1) GO TO 260
IF(LF.GT.2) GO TO 233
IR=2.*RNDM(LF)+1
I3=ISIGN(MOD(IABS(IFLAF1)-1+IR,3)+1,IFLAF1)
IFLAF1=I3
IFLAF2=-IFLAF1
GO TO 260
233 IF(LF.GT.3) GO TO 234
IFLAF1=LGLUS
IFLAF2=LGLUS
GO TO 260
234 IFLAF1=LF
IFLAF2=-IFLAF1
GO TO 260
001585 C-- G-G ANNIHILATION CHANNEL
001586 250 IF(LF.GT.1) GO TO 251
001587 IFLAF1=LGLUS
001588 IFLAF2=LGLUS
001589 GO TO 260
001590 251 IF(LF.GT.3) GO TO 252
001591 IR=3.*RNDM(LF)+1.
001592 IFLAF1=IR
001593 IFLAF2=-IFLAF1
001594 GO TO 260
001595 252 IFLAF1=LF
001596 IFLAF2=-IFLAF1
001597 260 CONTINUE
001598 C
001599 IF(WEIGHT.LT.1.E-30) RETURN
C-- PARTONS HAVE INTERACTED
SHAD=SHAT
IF(KEYBRE.LT.2)
*CALL TEVOL(IRS)
CALL HARD(1,1,IFLAF1,IFLAF2,SHAT,TV(ITU))
MNQUA=MAX0(MNQUA,MQUA)
IS=2
NEVENT=NEVENT+1
TWEIGH=TWEIGH+WEIGHT
TENTRS=TENTRS+1.
NPART=0
NQUA=MQUA
IF(KHADRO.EQ.1)
*CALL HADRON(1,MQUA,1)
IF(WEIGHT.LT.1.E-30) GO TO 215
IF(NEVENT.LE.NMUMP) CALL DUMPEV
IFIREF=1
CALL FILLH
GO TO 210
215 CONTINUE
NEVENT=NEVENT-1
TENTRS=TENTRS-1
WGT=WGT-MGALQ(IALQ,IPACH,2)
WEIGHT=WEIGOR
GO TO 210
200 CONTINUE
IF(IFIRE.EQ.0) GO TO 1
IF(IFIREF.EQ.0) GO TO 700
RETURN
C
C-- MINIBIAS EVENTS
300 CONTINUE
NEVENT=NEVENT+1
QSQ=1.E-10
PTHARD=1.E-10
NQUA=0
WEIGHT=1.
WEIGHT=WEIGHT*(1.-PROPTW)/(1.-FRAGEN)
TWEIGH=TWEIGH+WEIGHT
NPART=0
IF(KHADRO.EQ.1)
*CALL HADRON(1,0,1)
IF(WEIGHT.LT.1.E-30) GO TO 310
IF(NEVENT.LE.NMUMP) CALL DUMPEV
CALL FILLH
RETURN
310 CONTINUE
NEVENT=NEVENT-1
GO TO 300
END
C
SUBROUTINE GEQSQ(ALQ,IALQ)
*****
C-- GENERATES HARD QSQ
COMMON/ALQGEN/MGALQ(127),MGALQ(127,5,3),Q2VECT(128),NALQ,DALQ
1 ,ALAMB,ALLAM2,PTHIN,PTHAX,PTZINT,QZEV,ALQZM,ALQF,ALQF,YF
2 ,IPBAR,MCASC,SIGINH,PROPT,IPWGT,PROPTH,PTWGT,FRHARD,FRAGEN,FRAT00
3 ,ZMCUT,ZLH,ZUP,XMIN
COMMON/FORGEN/PTHGE,MGGE,SIGHRD
INTEGER IHALF(4)
DATA IHALF/32,16,8,4,2,1/
VR=MGGE*(1.-MGGE)*RNDM(0)
IF(VR.LT.MGALQ(1)) GO TO 30
IC=64
DO 10 IN=1,6
10 ICL=IC
ID=IHAF(IN)
IF(VR.LT.MGALQ(IC)) ID=-ID
IC=ICL+ID
CONTINUE
IF(ABS(MGALQ(IC)-MGALQ(ICL)).LT.1.E-30) GO TO 20
ALQ1=ALQ+DALQ*FLOAT(IC)
ALQ2=ALQ+DALQ*FLOAT(ICL)

```

```

      ALQ=((VR-WGALQT(IC))*ALQ2+(WGALQT(ICL)-VR)*ALQ3)
      1 / (WGALQT(ICL)-WGALQT(IC))
      IALQ=MAXO(IC,ICL)
      RETURN
20  IALQ=MINO(IC,ICL)
      ALQ=ALQ+DALQ*FLOAT(IALQ)
      RETURN
30  ALQ=ALQ+DALQ*VR/WGALQT(1)
      IALQ=1
      RETURN
      END
C
SUBROUTINE GEPAIR
*****
C-- GENERATES PAIR OF INITIAL PARTONS - CALLED BY GEVENT
COMMON/QCDS/CF,CA,BALPH,PI,PI2,PROGL,PROFLA(2),NFLAVS,LGLUS,IVAL
COMMON/TABQRK/TABQRK(127,7),NBIN,DBIN,F2XQV(7),WGQV(7),AGLU,HDBIN
1 ,XV(127),AL1,AL2,IFCODE(5)
2 ,TGLIAS(127),LGLIAS(127),TGLIAS(127),LGLIAS(127)
COMMON/ALQGEN/WGALQT(127),WGALQ(127,5,3),Q2VECT(128),NALQ,DALQ
1 ,ALAMB,ALLAM2,PTMIN,PTMAX,PT2INT,QZEV,ALQZM,ALQI,ALQF,YF
2 ,IPBAR,MCASC,SIGINL,PROPT,IPWTGT,PROPTW,PTWGTW,FRHARD,FRAGEN,FRATOO
3 ,ZHCUT,ZLW,ZUP,XMIN
COMMON/ENTREV/S,QSQ,YQSQ,IPACH,XING(2),XMING(2)
1 ,PXING(2),PYING(2),IFLING(2),IFLING(2),ICHING(2)
COMMON/EVENT/NEVENT,ECM,PARHAD(401,7),NPART,MPART,WEIGHT
COMMON/TWEIGH/TWEIGH,TENTRS,TWEIMN
REAL WQ(6),WQAQ(6)
INTEGER INV(2)
DATA ICALL/0/,WQAQ/6*0/,INV/2,1/
C-- INITIALIZATION
IF(ICALL.GT.0) GO TO 9
ICALL=1
DO 8 J=1,7
DO 7 IX=2,NBIN
7  TABQRK(IX,J)=TABQRK(IX-1,J)+TABQRK(IX,J)
WQV(J)=TABQRK(NBIN,J)
8  CONTINUE
WQV(5)=WQV(5)*.5
WQV(6)=WQV(5)
DO 1 J=1,6
1  WQ(J)=WQV(J)*.2
DO 2 J=2,6
2  WQ(J)=WQ(J)+WQ(J-1)
DO 3 J=1,6
3  WQ(J)=WQ(J)/WQ(6)
DO 6 J=1,6,2
6  WQAQ(J)=WQV(J)+WQV(J+1)
WQAQ(2)=WQAQ(1)+WQAQ(2)
WQAQ(3)=WQAQ(2)+WQAQ(3)
WQAQ(4)=WQAQ(3)+WQAQ(4)
WQAQ(5)=WQAQ(4)+WQAQ(5)
WQAQ(6)=WQAQ(5)+WQAQ(6)
DO 4 J=2,6
4  WQV(J)=WQV(J)+WQV(J-1)
DO 5 J=1,6
5  WQV(J)=WQV(J)/WQV(6)
XST=XMIN
ALST=ALOG(XST)
IXST=INT(XST/DBIN)+1
ALST1=ALOG(FLOAT(IXST)*DBIN)
9  CONTINUE
C-- CHANNEL BRANCHING - TRANSLATE INTO PARTON FLAVORS
C-- IPACH = 1 Q-Q OR Q-QB DIFFERENT FLAVORS, 2 Q-Q SAME FLAVORS,
C-- 3 Q-QB SAME FLAVORS, 4 Q-G, 5 G-G
IF(IPACH.EQ.1) GO TO 10
IF(IPACH.EQ.2) GO TO 20
IF(IPACH.EQ.3) GO TO 30
IF(IPACH.EQ.4) GO TO 40
IF(IPACH.EQ.5) GO TO 50
10 DO 11 L=1,2
14 RR=RNOM(L)
DO 12 J=1,6
12 JOUT=J
IF(RR.LT.WGQV(J)) GO TO 13
CONTINUE
13 ICHING(L)=JOUT
IF(L.EQ.1) GO TO 11
CALL GECODE(ICHING,IFLING)
IF(IABS(IFLING(1)).EQ.IABS(IFLING(2))) GO TO 14
11 CONTINUE
GO TO 60
20 IF(IPBAR.EQ.1) GO TO 35
CONTINUE
RR=RNOM(0)
DO 21 J=1,6
21 JOUT=J
IF(RR.LT.WGQV(J)) GO TO 22
001761 21 CONTINUE
001762 22 ICHING(1)=JOUT
001763 ICHING(2)=JOUT
001764 CALL GECODE(ICHING,IFLING)
001765 GO TO 60
001766 30 IF(IPBAR.EQ.1) GO TO 25
001767 35 CONTINUE
001768 L1=1.5+RNOM(0)
001769 L2=INV(L1)
001770 RR=RNOM(0)
001771 DO 31 J=1,3
001772 JOUT=J
001773 IF(RR.LT.WQAQ(J)) GO TO 32
001774 CONTINUE
001775 31 ICHING(L1)=2+JOUT-1
001776 32 ICHING(L2)=ICHING(L1)+1
001777 CALL GECODE(ICHING,IFLING)
001778 GO TO 60
001779 40 L1=1.5+RNOM(0)
001780 L2=INV(L1)
001781 ICHING(L1)=7
001782 RR=RNOM(0)
001783 DO 41 J=1,6
001784 JOUT=J
001785 IF(RR.LT.WGQV(J)) GO TO 42
001786 41 CONTINUE
001787 42 ICHING(L2)=JOUT
001788 CALL GECODE(ICHING,IFLING)
001789 GO TO 60
001790 50 ICHING(1)=7
001791 ICHING(2)=7
001792 IFLING(1)=LGLUS
001793 IFLING(2)=LGLUS
C-- GENERATE X'S AND PT'S FOR INITIAL PARTON PAIR
001794 CONTINUE
001795 DO 71 L=1,2
001796 CALL GEXIN(ICHING(L),XST,IXST,ALST,ALST1,XING(L),XMING(L),
001797 1 ,PXING(L),PYING(L))
001798 CONTINUE
001799 71 CONTINUE
C-- CHECK WHETHER SHAT IS ENOUGH FOR QSQ
001800 SHAT=(XING(1)*XING(2)+XMING(1)*XMING(2))*S
001801 1 -2.*(PXING(1)*PXING(2)+PYING(1)*PYING(2))
001802 IF(SHAT.LT.3.*QSQ) GO TO 9
001803 RETURN
001804 END
001805
001806 SUBROUTINE GECODE(ICHING,IFLING)
001807 *****
C-- CHANNEL -- PARTON PAIR FLAVORS CODE
001808 COMMON/ALQGEN/WGALQT(127),WGALQ(127,5,3),Q2VECT(128),NALQ,DALQ
001809 1 ,ALAMB,ALLAM2,PTMIN,PTMAX,PT2INT,QZEV,ALQZM,ALQI,ALQF,YF
001810 2 ,IPBAR,MCASC,SIGINL,PROPT,IPWTGT,PROPTW,PTWGTW,FRHARD,FRAGEN,FRATOO
001811 3 ,ZHCUT,ZLW,ZUP,XMIN
001812 COMMON/TABQRK/TABQRK(127,7),NBIN,DBIN,F2XQV(7),WGQV(7),AGLU,HDBIN
001813 1 ,XV(127),AL1,AL2,IFCODE(5)
001814 2 ,TGLIAS(127),LGLIAS(127),TGLIAS(127),LGLIAS(127)
001815 COMMON/ENTREV/S,QSQ,YQSQ,IPACH,XING(2),XMING(2)
001816 1 ,PXING(2),PYING(2),IFLING(2),IFLING(2),ICHING(2)
001817 ICHR=ICH
001818 IF(ICH.EQ.6) ICHR=5
001819 CALL FINDX(ICH,XST,X,IX)
001820 IF(IX.GT.IXST) GO TO 300
001821 F2X=F2IN(XST,ICHR)
001822 X=EXP(RNOM(0))*(ALST1-ALST)+ALST
001823 IF(RNOM(0).GT.F2IN(X,ICHR)/F2X) GO TO 105
001824 PT2=PT2INT+ALOG(RNOM(0))
001825 XM=PT2/(X+S)
001826 IF(XM.GT.X) GO TO 300
001827 PT=SQRT(PT2)
001849 001849
001850 001850
001851 001851
001852 001852
001853 001853
001854 001854
001855 001855
001856 001856
001857 001857
001858 001858
001859 001859
001860 001860
001861 001861
001862 001862
001863 001863
001864 001864
001865 001865
001866 001866
001867 001867
001868 001868
001869 001869
001870 001870
001871 001871
001872 001872
001873 001873
001874 001874
001875 001875
001876 001876
001877 001877
001878 001878
001879 001879
001880 001880
001881 001881
001882 001882
001883 001883
001884 001884
001885 001885
001886 001886
001887 001887
001888 001888
001889 001889
001890 001890
001891 001891
001892 001892
001893 001893
001894 001894
001895 001895
001896 001896
001897 001897
001898 001898
001899 001899
001900 001900
001901 001901
001902 001902
001903 001903
001904 001904
001905 001905
001906 001906
001907 001907
001908 001908
001909 001909
001910 001910
001911 001911
001912 001912
001913 001913
001914 001914
001915 001915
001916 001916
001917 001917
001918 001918
001919 001919
001920 001920
001921 001921
001922 001922
001923 001923
001924 001924
001925 001925
001926 001926
001927 001927
001928 001928
001929 001929
001930 001930
001931 001931
001932 001932
001933 001933
001934 001934
001935 001935
001936 001936

```



```

COMMON/ALQGEN/NGALQT(127),NGALQ(127,5,3),Q2VECT(128),NALQ,DALQ 002113 201 CALL ZGENS(1,Z,IFLAQ) 002201
1 ,ALAMB,ALLAM2,PTMIN,PTMAX,PTZINT,QZEV,ALQZM,ALQI,ALQF,YF 002114 Z1=1.-Z 002202
2 ,IPBAR,NCASC,SIGINL,PROPT,IPWTGT,PROPTH,PTMGTH,FRHARD,FRAGEN,FRAT 002115 IFLA1=IFLA 002203
3 ,ZMCUT,ZLW,ZUP,XMIN 002116 X1=X*Z 002204
C 002117 IFLA2=LGLUS 002205
C-- QCD CONSTANTS 002118 X2=X*Z1 002206
PI=4.*ATAN(1.) 002119 GO TO 300 002207
PI2=PI*2 002120 C 002208
COLORS=3 002121 C-- GLUON BRANCHING. GENERATE DECAY CHANNEL 002209
CA=COLORS 002122 210 CONTINUE 002210
CF=(COLORS**2-1.)/(2.*COLORS) 002123 IF(RNDM(0.).GT.PROGL) GO TO 230 002211
BALPH=(11.*COLORS-2.*FLOAT(NFLAVS))/(12.*PI) 002124 C 002212
C 002125 C-- GLUON BRANCHES INTO TWO GLUONS 002213
C-- BASIC CONSTANTS FOR EMISSION PROBABILITIES - SPACELIKE EVOLUTION 002126 CALL ZGENS(2,Z,IFLAQ) 002214
PRO1=PAPINT(1,ZUP)-PAPINT(1,ZLW) 002127 Z1=1.-Z 002215
PRO2=PAPINT(2,ZUP)-PAPINT(2,ZLW) 002128 IFLA1=LGLUS 002216
PRO3=PAPINT(3,ZUP)-PAPINT(3,ZLW) 002129 IFLA2=LGLUS 002217
PRO3=PRO3+FLOAT(NFLAVS) 002130 X1=X*Z 002218
PROGL=PRO2/(PRO2+PRO3) 002131 X2=X*Z1 002219
FACT=1./(2.*PI*BALPH) 002132 GO TO 300 002220
PROFLA(1)=FACT*PRO1 002133 C 002221
PROFLA(2)=FACT*(PRO2+PRO3) 002134 C-- GLUON BRANCHES INTO Q-QB 002222
C 002135 230 CONTINUE 002223
C 002136 CALL ZGENS(3,Z,IFLAQ) 002224
C 002137 Z1=1.-Z 002225
C 002138 IFLA1=IFLAQ 002226
C 002139 IFLA2=-IFLAQ 002227
C 002140 X1=X*Z 002228
C-- GENERATES INITIAL LLA CASCADES 002141 X2=X*Z1 002229
C 002142 C 002230
COMMON/QCDS/CF,CA,BALPH,PI,PI2,PROGL,PROFLA(2),NFLAVS,LGLUS,IVAL 002143 C-- GENERATE MASSES AND PT OF SECONDARIES AND BOOK 002231
COMMON/PARQ/PARACT(47,7,2),MXACT,IBEAM,NBOOK(2),JBOOK,MBOOK(2) 002144 C 002232
1 ,PARSPE(39,6,2),MXSPE,SHARD 002145 300 CONTINUE 002233
COMMON/ALQGEN/NGALQT(127),NGALQ(127,5,3),Q2VECT(128),NALQ,DALQ 002146 S=ECH**2 002234
1 ,ALAMB,ALLAM2,PTMIN,PTMAX,PTZINT,QZEV,ALQZM,ALQI,ALQF,YF 002147 IF(X1.LT.XMIN) GO TO 301 002235
2 ,IPBAR,NCASC,SIGINL,PROPT,IPWTGT,PROPTH,PTMGTH,FRHARD,FRAGEN,FRAT 002148 CALL Q2GEN(IFLA1,Y,Y1,QSQ1) 002236
3 ,ZMCUT,ZLW,ZUP,XMIN 002149 TSPQ1=Z1*QSQ 002237
COMMON/EVENT/NEVENT,ECH,PARHAD(401,7),NPART,MXPART,WEIGHT 002150 PTR1=Z1*QSQ 002238
COMMON/TRANSL/ALAMBT,NFLATR,LGLUT,QSQMKT 002151 PTR1=SQRT(PTR1) 002239
COMMON/NEVOL/NEVOL,ISTOP(2),IBACK(2),MBOOKS(2) 002152 IF(KEYBRE.GE.1) PTR1=0. 002240
COMMON/CANC/ICANC(2),NBKCN(47,2) 002153 PH1=PI2*RNOM(0) 002241
COMMON/MAXN/MBOOK,MBOOK,MNPART,MNQUA,MNCUE,MNCUEJ,MNSJO 002154 PX1=PTR1*COS(PH1) 002242
COMMON/KEYBRE/KEYBRE 002155 PY1=PTR1*SIN(PH1) 002243
REAL CUE(41,8,2) 002156 PX1S=PX1*Z*PX 002244
INTEGER NCUE(2) 002157 PY1S=PY1*Z*PY 002245
DATA MNXCUE/41/ 002158 TMSQ1=PX1S**2+PY1S**2 002246
DATA ICALL/0/ 002159 XMI=TMSQ1/(S*X1) 002247
C 002160 IF(JBOOK.EQ.1) GO TO 10 002248
C 002161 IF(ICALL.GT.0) GO TO 10 002249
C 002162 ICALL=1 002250
C 002163 IBOOK=1 002251
C 002164 IBOOK1=1 002252
C 002165 IBOOK2=1 002253
10 CONTINUE 002166 TSPQ2=Z1*QSQ 002254
C 002167 PTR2=Z1*QSQ 002255
C 002168 PTR2=SQRT(PTR2) 002256
C 002169 IF(KEYBRE.GE.1) PTR2=0. 002257
C 002170 PH2=PI2*RNOM(0) 002258
C-- RESET 002171 PX2=PTR2*COS(PH2) 002259
C 002172 PY2=PTR2*SIN(PH2) 002260
C 002173 PX2S=PX2*Z1*PX 002261
C-- SETTING PARAMETERS OF INITIAL LEG 002174 PY2S=PY2*Z1*PY 002262
IF(KEYBRE.GE.3) GO TO 100 002175 TMSQ2=PX2S**2+PY2S**2 002263
Y=PARACT(NBK,4,IBEAM) 002176 XM2=TMSQ2/(S*X2) 002264
IF(Y.GT.YQSQ) GO TO 100 002177 IF(JBOOK.EQ.1) GO TO 305 002265
IFLA=PARACT(NBK,1,IBEAM) 002178 PX2T=-PX2*Z*PX 002266
IBOOK=PARACT(NBK,2,IBEAM) 002179 PY2T=-PY2*Z*PY 002267
QSQ=EXP(EXP(Y)*ALQZM+ALLAM2) 002180 305 CONTINUE 002268
X=PARACT(NBK,5,IBEAM) 002181 C 002269
PX=PARACT(NBK,6,IBEAM) 002182 C-- DECIDE WHICH IS NEXT TO CASCADE, ADD TO CUE 002270
PY=PARACT(NBK,7,IBEAM) 002183 C 002271
IF(JBOOK.EQ.2) GO TO 200 002184 400 CONTINUE 002272
ICANC(IBEAM)=ICANC(IBEAM)+1 002185 IF(X1.LT.XMIN) GO TO 452 002273
NBKCN(ICANC(IBEAM),IBEAM)=NBK 002186 IF(X1.LT.XMI) GO TO 452 002274
GO TO 200 002187 IF(X2.LT.XMIN) GO TO 450 002275
C 002188 IF(X2.LT.XM2) GO TO 450 002276
100 IBACK(IBEAM)=1 002189 NCUE(IBEAM)=NCUE(IBEAM)+1 002277
RETURN 002190 MNUE=MAX0(MNCUE,NCUE(IBEAM)) 002278
C-- BRANCHING LOOP 002191 NC=NCUE(IBEAM) 002279
C 002192 IF(NC.GE.MNXCUE) GO TO 500 002280
C ***** 002193 IF(JBOOK.EQ.2) CALL TBOOK(IBOOK,IFLA1,TSPQ2,X1,PX2T,PY2T,IBOOK2) 002281
C 002194 CUE(1,IBEAM)=IFLA2 002282
C 002195 CUE(2,IBEAM)=X2 002283
C 002196 CUE(3,IBEAM)=X2 002284
C-- GENERATE Z AND MASS -- BOOK 002197 CUE(4,IBEAM)=IBOOK2 002285
C ***** 002198 CUE(5,IBEAM)=PX2S 002286
200 CONTINUE 002199 CUE(6,IBEAM)=PY2S 002287
IF(IFLA.EQ.LGLUS) GO TO 210 002200 CUE(7,IBEAM)=QSQ2 002288
C-- QUARK BRANCHING

```



```

CUE(NC,8,IBEAM)=XM2
450 CONTINUE
IF(JBOOK.EQ.2) CALL TBOOK(BOOK,IFLA2,TPSQ1,X2,PK1T,PY1T,IBOOK1)
IF(Y1.GT.YQSQ) GO TO 402
IFLA=IFLA1
Y=Y1
X=X1
PX=PX1S
PY=PY1S
IBOOK=IBOOK1
QSQ=QSQ1
GO TO 200
402 CONTINUE
CALL BOOK(IFLA1,IBOOK1,XM1,Y1,X1,PK1S,PY1S)
IF(IBOOK.EQ.1) GO TO 462
RETURN
452 IF(X2.LT.XMIN) GO TO 462
IF(X2.LT.XM2) GO TO 462
IF(JBOOK.EQ.2) CALL TBOOK(BOOK,IFLA1,TPSQ2,X1,PK2T,PY2T,IBOOK2)
IF(Y2.GT.YQSQ) GO TO 460
IFLA=IFLA2
Y=Y2
X=X2
PX=PX2S
PY=PY2S
IBOOK=IBOOK2
QSQ=QSQ2
GO TO 200
460 CONTINUE
CALL BOOK(IFLA2,IBOOK2,XM2,Y2,X2,PK2S,PY2S)
IF(IBOOK.EQ.1) GO TO 462
RETURN
C
C-- PICK UP NEXT IN CUE
C
462 IF(NCUE(IBEAM).EQ.0) GO TO 480
NC=NCUE(IBEAM)
IFLA=CUE(NC,1,IBEAM)
Y=CUE(NC,2,IBEAM)
X=CUE(NC,3,IBEAM)
IBOOK=CUE(NC,4,IBEAM)
PX=CUE(NC,5,IBEAM)
PY=CUE(NC,6,IBEAM)
QSQ=CUE(NC,7,IBEAM)
MBOOK(IBEAM)=IBOOK
IF(JBOOK.EQ.2) MBOOKS(IBEAM)=PARSP(BOOK,4,IBEAM)
NCUE(IBEAM)=NCUE(IBEAM)+1
IF(Y.GT.YQSQ) GO TO 465
GO TO 200
465 CONTINUE
XM=CUE(NC,8,IBEAM)
CALL BOOK(IFLA,IBOOK,XM,Y,X,PX,PY)
IF(IBOOK.EQ.1) GO TO 462
RETURN
C
C-- EXIT
C
480 CONTINUE
ISTOP(IBEAM)=1
IBACK(IBEAM)=0
MBOOKS(IBEAM)=0
RETURN
C
C-- ABNORMAL EXIT
500 PRINT 501,MXNCUE,NEVOL
501 FORMAT(1H1,36HNUMBER OF QUANTA IN EVOL CUE EXCEEDS ,I10
1 //1X,13HCASCADE NO. = ,I10
2 //1X,56HINCREASE 1ST DIMENSION OF CUE( , , ) AND INCREASE MXNCUE
2 )
STOP
END
C
SUBROUTINE Q2GEN(IFLA,YIN,Y,QSQ)
C
C-- GENERATES EVOLUTION PARAMETER OF LEG (Y,GE,YIN)
COMMON/QCDS/CF,CA,BALPH,PI,PI2,PROGL,PROFLA(2),NFLAYS,LGLUS,IYAL
COMMON/ALQGEN/MGALQT(127),MGALQ(127,5,3),Q2VECT(128),NALQ,DALQ
1 ,ALAMB,ALLAM2,PTMIN,PTMAX,PT2INT,QZEV,ALQZM,ALQI,ALQF,YF
2 ,IPBAR,NCASC,SIGINL,PROPT,IPTNGT,PROPTM,PTMGTH,FRHARD,FRAGEN,FRAT
3 ,ZMCUT,ZLM,ZUP,XMIN
REAL MGTMAX(3)
DATA MGTMAX/2.,1.125,1./
DATA ICALL/0/
IF(ICALL.GT.0) GO TO 30
ICALL=1
ALLM=ALOG(1.-ZLM)
LN(0.5)
ALLMS=-.6931471
ALUP=ALOG(1.-ZUP)
DZ=ALLM-ALUP
DZG=ALLM-ALUP
CONTINUE
IF(IFUN.EQ.1) GO TO 1
IF(IFUN.EQ.2) GO TO 2
IF(IFUN.EQ.3) GO TO 3
C
C-- QUARK BRANCHING
1 CONTINUE
RR=RNOM(0.)
Z1=EXP(RR*DZ+ALUP)
Z=1.-Z1
WFUN=1.+Z**2
IF(RNOM(0.).GT.WFUN/MGTMAX(IFUN)) GO TO 11
RETURN
C
C-- GLUON BRANCHING INTO TWO GLUONS
2 CONTINUE
RR=RNOM(0.)
Z1=EXP(RR*DZG+ALUP)
Z=1.-Z1
A=Z**2
B=Z1**2
WFUN=(A+B+A*B)/Z
IF(RNOM(0.).GT.WFUN/MGTMAX(IFUN)) GO TO 21
IF(RNOM(0.).GT.0.5) Z=Z1
RETURN
C
C-- GLUON BRANCHING INTO Q-QB
3 CONTINUE
Z=ZLM+(ZUP-ZLM)*RNOM(0.)
Z1=1.-Z
WFUN=Z**2+Z1**2
IF(RNOM(0.).GT.WFUN) GO TO 3
IFLAQ=INT(FLOAT(NFLAYS)*RNOM(0.))+1.
RETURN
C
END
C
SUBROUTINE BOOK(IFLA1,IBOOK,XM1,Y1,X1,PK1S,PY1S)
C
C-- BOOKS SPACELIKE QUANTA GENERATED BY EVOL
COMMON/PARQ/PARACT(47,7,2),MXACT,IBEAM,MBOOK(2),JBOOK,MBOOK(2)
1 ,PARSP(39,6,2),MXSPE,SHARD
COMMON/NEVOL/NEVOL,ISTOP(2),IBACK(2),MBOOKS(2)
COMMON/ENTREV/S,QSQ,YQSQ,IPACH,XING(2),XING(2)
1 ,PXING(2),PYING(2),IFLING(2),ICMING(2)
NBK=MBOOK(IBEAM)
IF(JBOOK.EQ.1)
INBK=NBK+1
IF(NBK.GT.MXACT) GO TO 500
MBOOK(IBEAM)=NBK
PARACT(NBK,1,IBEAM)=IFLA1
PARACT(NBK,2,IBEAM)=IBOOK
PARACT(NBK,3,IBEAM)=XM1
PARACT(NBK,4,IBEAM)=Y1
PARACT(NBK,5,IBEAM)=X1
PARACT(NBK,6,IBEAM)=PY1S
PARACT(NBK,7,IBEAM)=PY1S
IBACK(IBEAM)=1
RETURN
C
C-- ABNORMAL EXIT
500 PRINT 501,MXACT,NEVOL
501 FORMAT(5(//),1X,44HNUMBER OF QUANTA PER BEAM IN PARACT EXCEEDS ,
110//1X,13HCASCADE NO. = ,I10

```

```

2 //IX,32HINCREASE 1ST DIMENSION OF PARACT      002465      FTST(2)=ABS(FM)      002553
3 //IX,32HINCREASE 1ST DIMENSION OF NBKCNC      002466      FTST(3)=ABS(FB)      002554
4 //IX,40HINCREASE DIMENSION OF NENT2 (SUB. MATCH) 002467      DO 100 I=2,3      002555
5 //IX,14HINCREASE MXACT )      002468      IF(FMAX,GE,FTST(I)) GO TO 100      002556
CALL OVERDM      002469      FMAX=FTST(I)      002557
RETURN      002470      CONTINUE      002558
END      002471      EST=(FA+4.*FM+FB)*DA/6.      002559
C      002472      ABSAR=(FTST(1)+4*FTST(2)+FTST(3))*DA/6.      002560
SUBROUTINE TBOOK(IBOOK,IFLA1,TPSQ1,XIT,PXIT,PYIT,IBOOK1) 002473      AEST=ABSAR      002561
C      002474      I=RECUR      002562
C-- BOOKS TIMELIKE QUANTA GENERATED BY EVOL      002475      1      002563
COMMON/PARQ/PARACT(47,7,2),MXACT,IBEAM,NBOOK(2),JBOOK,MBOOK(2) 002476      002564
1 ,PARSPE(39,6,2),MXSPE,SHARD      002477      F1=FUN(13.*A+B)/4.      002565
COMMON/NEVOL/NEVOL,ISTOP(2),IBACK(2),MBOOKS(2)      002478      F2(LVL)=FUN((A+3.*B)/4.)      002566
COMMON/EVENT/NEVENT,ECM,PARHAD(401,7),NPART,MXPART,WEIGHT      002479      EST1=5X*(FA+4.*F1+FM)      002567
MBK=MBOOK(IBEAM)      002480      FB(LVL)=FB      002568
MBK=MBK+1      002481      XB(LVL)=B      002569
IF(MBK.GT.MXSPE) GO TO 500      002482      EST2(LVL)=5X*(FM+4.*F2(LVL)+FB)      002570
MBOOK(IBEAM)=MBK      002483      SUM=EST1+EST2(LVL)      002571
IBOOK1=MBK      002484      FTST(1)=ABS(F1)      002572
PARSPE(MBK,1,IBEAM)=PXIT      002485      FTST(2)=ABS(F2(LVL))      002573
PARSPE(MBK,2,IBEAM)=PYIT      002486      FTST(3)=ABS(FM)      002574
PARSPE(MBK,3,IBEAM)=XIT*ECM      002487      AEST=5X*(ABS(FA)+4.*FTST(1)+FTST(3))      002575
PARSPE(MBK,4,IBEAM)=IBOOK      002488      AEST2(LVL)=5X*(FTST(3)+4.*FTST(2)+ABS(FB))      002576
PARSPE(MBK,5,IBEAM)=TPSQ1      002489      ABSAR=ABSAR-AEST+AEST1+AEST2(LVL)      002577
PARSPE(MBK,6,IBEAM)=IFLA1      002490      M=M+2      002578
RETURN      002491      IF(N.EQ.1) GO TO 201      002579
C      002492      IF(N.EQ.2) GO TO 200      002580
C-- ABNORMAL EXIT      002493      IF(N.EQ.3) GO TO 202      002581
500 PRINT 501,MXSPE,NEVOL      002494      DELTA=ABSAR      002582
501 FORMAT(5I/),IX,35NUMBER OF QUANTA IN PARSPE EXCEEDS ,I10      002495      GO TO 205      002583
1 //IX,13HCASCADE NO. = ,I10      002496      PRINT 39      002584
2 //IX,52HINCREASE 1ST DIMENSION OF PARSPE AND INCREASE MXSPE      002497      FORMAT(20H ERROR RETURN-N.LT.0 )      002585
2 )      002498      RETURN      002586
CALL OVERDM      002499      PRINT 40      002587
RETURN      002500      FORMAT(20H ERROR RETURN-N.GT.3 )      002588
END      002501      RETURN      002589
C      002502      DELTA=1.      002590
SUBROUTINE OVERDM      002503      GO TO 205      002591
C      002504      DO 203 I=1,2      002592
C-- TAKES ACTION IN DIMENSION OVERFLOW      002505      IF(FMAX,GE,FTST(I)) GO TO 203      002593
C-- IF KOVER.GT.0 JUST SETS WEIGHT=0. AND EXECUTION GOES ON WITH SOME      002506      203      002594
C-- DIAGNOSTICS FOR THE 1ST 100 OVERFLOWS -- IF KOVER.LT.0 STOPS      002507      FMAX=FTST(I)      002595
C-- EXECUTION BY PRODUCING AN ARITHMETIC ERROR      002508      CONTINUE      002596
COMMON/EVENT/NEVENT,ECM,PARHAD(401,7),NPART,MXPART,WEIGHT      002509      DELTA=FMAX      002597
COMMON/OVER/KOVER,SPYOVR      002510      DAFT=EST-SUM      002598
C      002511      DIFF=ABS(DAFT)      002599
DATA ICALL/0/      002512      DAFT=DAFT/15.      002600
C      002513      IF(DIFF-EPS*DELTA)6,6,3      002601
ICALL=ICALL+1      002514      IF(LVL-30)4,2,2      002602
SPYOVR=FLOAT(ICALL)      002515      IF(LVL-1)2,4,2      002603
WEIGHT=0.      002516      2=UP      002604
IF(KOVER.GT.0..AND.ICALL.LT.100) PRINT 5,ICALL      002517      A=B      002605
5 FORMAT(1X,21HTHIS EVENT IS IGNORED      002518      ESUM=ESUM+DAFT      002606
1 //IX,16HEXECUTION PROCEEDS      002519      TSUM=TSUM+SUM      002607
2 //IX,39H(KOVER.GT.0, DIMENSION OVERFLOWS SO FAR ,I10,IH) )      002520      LVL=LVL-1      002608
IF(KOVER.GT.0..AND.ICALL.LT.100) RETURN      002521      L=NRTR(LVL)      002609
C      002522      IF(L.EQ.1) GO TO 11      002610
11=NRTR(12)      002523      IF(L.EQ.2) GO TO 12      002611
PRINT 10      002524      NRTR(LVL)=1      002612
10 FORMAT(5I/IX,80(IH*))//6X,35HTHE FOLLOWING ERROR IS DELIBERATELY,      002525      EST=EST1      002613
135H PROVOCATED TO ACTIVATE MEMORY DUMP ,      002526      AEST=AEST1      002614
2 //5I/IX,80(IH*))      002527      FB=FM      002615
A=0.      002528      FM=F1      002616
B=1/A      002529      B=(A+B)/2.      002617
RETURN      002530      EPS=EPS/2.      002618
END      002531      LVL=LVL+1      002619
C      002532      GO TO 1      002620
FUNCTION ASIMP(A1,B1,EP,M,N,FUN)      002533      NRTR(LVL)=2      002621
C      002534      FA=FB      002622
C-- ADAPTIVE SIMPSON INTEGRATION ROUTINE      002535      FM=F2(LVL)      002623
REAL F2(30),FBP(30),EST2(30)      002536      FB=FBP(LVL)      002624
INTEGER NRTR(30)      002537      B=XB(LVL)      002625
REAL AEST2(30),FTST(3),XB(30)      002538      EST=EST2(LVL)      002626
IF(N.LE.0) GO TO 210      002539      AEST=AEST2(LVL)      002627
IF(N.GT.3) GO TO 211      002540      GO TO 7      002628
A=A1      002541      EPS=2.*EPS      002629
B=B1      002542      IF(LVL-1)5,5,9      002630
EPS=EP*15.      002543      ASIMP=TSUM-ESUM      002631
ESUM=0.      002544      RETURN      002632
TSUM=0.      002545      END      002633
LVL=1      002546      C-----      002634
DA=B-A      002547      C      002635
FA=FUN(A)      002548      C-- SECTION (C)      002636
FM=FUN((A+B)*0.5)      002549      C-- PACKAGE OF ROUTINES HANDLING TIMELIKE EVOLUTION OF QUANTA      002637
FB=FUN(B)      002550      C      002638
M=3      002551      C-----      002639
FMAX=ABS(FA)      002552      C      002640
FTST(1)=FMAX

```

```

C
SUBROUTINE INIQCD
*****
C-- INITIALIZES PARAMETERS USED BY JETQCD
COMMON/TRANSL/ALAMBT,NFLATR,LGLUT,QSQMXT
COMMON/QCD/ALAMB0,CF,CA,BALPH,PI
COMMON/CUTOFF/QZ,QZSQ,QZMASS(6),QZFL(7),QZFLSQ(7),QTHRSQ(7)
1,NFLAVT,LGLU
COMMON/TABPSQ/TABPSQ(5,129),QSQMAX,ALMAX,ALMIN(7),DSTEP(7)
1,NPREC,NTAB,KODTAB(7),MAXTAB
DATA ICALL/0/
C
IF(ICALL.GT.0) RETURN
ICALL=1
C-- QCD CONSTANTS
PI=4.*ATAN(1.)
COLORS=3
CA=COLORS
CF=(COLORS**2-1.)/(2.*COLORS)
NFLAVT=NFLATR
BALPH=(11.*COLORS-2.*FLOAT(NFLAVT))/(12.*PI)
C
C-- CUTOFFS
QSQMAX=QSQMXT
ALAMB0=ALAMBT
QZSQ=QZ**2
DO 1 L=1,NFLAVT
  QZFL(L)=QZ+QZMASS(L)
  QZFLSQ(L)=QZFL(L)**2
1 QTHRSQ(L)=(QZ+QZFL(L))**2
LGLUT=LGLU
LGLU=LGLU
QZFL(LGLU)=QZ
QZFLSQ(LGLU)=QZSQ
QTHRSQ(LGLU)=4.*QZSQ
C
CALL TABLES
C
RETURN
END
C
SUBROUTINE TABLES
*****
C-- PREPARES TABLES FOR MASS DEGRADATION OF QUANTA
EXTERNAL FIQRK, FIGLU
COMMON/QCD/ALAMB0,CF,CA,BALPH,PI
COMMON/CUTOFF/QZ,QZSQ,QZMASS(6),QZFL(7),QZFLSQ(7),QTHRSQ(7)
1,NFLAVT,LGLU
COMMON/TABPSQ/TABPSQ(5,129),QSQMAX,ALMAX,ALMIN(7),DSTEP(7)
1,NPREC,NTAB,KODTAB(7),MAXTAB
COMMON/EVENT/NEVENT,ECH,PARHAD(401,7),NPART,MXPART,WEIGHT
COMMON/FINT/ALLAM2,LFLA
C
DO 1 L=1,NFLAVT
  IF(2.*QZ+QZMASS(L)).LT.ECH) NPLAX=L
  NFLAVT=NFLAX
  NPREC=7
  FACT=1./((2.*PI*BALPH)
  ALLAM2=ALOG(ALAMB0**2)
  NTAB=2*NPREC+1
  ALMAX=ALOG(QSQMAX)/(1.+1.E-6)
  EPSI=FLOAT(NTAB)**(-2)
  KFLA=0
  OLQTHR=0.
C
DO 10 L=1,LGLU
  ALMIN(L)=ALOG(QTHRSQ(L))
  DSTEP(L)=(ALMAX-ALMIN(L))/FLOAT(NTAB-1)
  IF(ABS(QTHRSQ(L)-OLQTHR).LT.1.E-5.AND.L.NE.LGLU) GO TO 9
  LFLA=L
  KFLA=KFLA+1
  IF(KFLA.GT.MAXTAB) GO TO 500
  ALW=ALMIN(L)-DSTEP(L)
  BUP=ALMIN(L)
  TABPSQ(KFLA,1)=0.
C
IF(L.EQ.LGLU) GO TO 12
DO 11 I=2,NTAB
  ALW=ALW+DSTEP(L)
  BUP=BUP+DSTEP(L)
  TABPSQ(KFLA,I)=FACT*ASIMP(ALW,BUP,EPSI,M,2,FIQRK)
  +TABPSQ(KFLA,I-1)
  OLQTHR=QTHRSQ(L)
  GO TO 9
11
DO 13 I=2,NTAB
  ALW=ALW+DSTEP(L)
  BUP=BUP+DSTEP(L)
  TABPSQ(KFLA,I)=FACT*ASIMP(ALW,BUP,EPSI,M,2,FIGLU)
  +TABPSQ(KFLA,I-1)
  OLQTHR=0.
  KFLA=0
  DO 501 L=1,NFLAVT
    IF(ABS(QTHRSQ(L)-OLQTHR).LT.1.E-5) GO TO 501
    OLQTHR=QTHRSQ(L)
    KFLA=KFLA+1
  501 CONTINUE
  KFLA1=KFLA+1
  PRINT 502,MAXTAB,NFLAVT,KFLA,KFLA1,KFLA1
  FORMAT(1H1,51DIMENSIONS OF TABPSQ( , ) ARE INSUFFICIENT (TABPSQ(
  11,18H, 129) AT PRESENT)
  2 /1X,12,33H FLAVORS HAVE BEEN REQUESTED WITH
  3 /1X,18H DIFFERENT CUTOFFS
  4 /1X,33H INCREASE 1ST DIMENSION TO TABPSQ(
  5 /1X,29H, 129) AND INCREASE MAXTAB TO ,12)
  STOP
  END
C
FUNCTION FIQRK(ALPSQ)
*****
C-- QUARK INTEGRATION FUNCTION USED BY TABLES
COMMON/QCD/ALAMB0,CF,CA,BALPH,PI
COMMON/CUTOFF/QZ,QZSQ,QZMASS(6),QZFL(7),QZFLSQ(7),QTHRSQ(7)
1,NFLAVT,LGLU
COMMON/FINT/ALLAM2,LFLA
C
A=ALPSQ
PSQ=EXP(A)
CALL ZLIMS(P SQ,QZSQ,ZLW,ZUP,FLAG)
IF(FLAG.GT.0.) GO TO 100
FIQRK=(PAPINT(1,ZUP)-PAPINT(1,ZLW))/(1-ALLAM2)
RETURN
100 FIQRK=0.
RETURN
END
C
FUNCTION FIGLU(ALPSQ)
*****
C-- GLUON INTEGRATION FUNCTION USED BY TABLES
COMMON/QCD/ALAMB0,CF,CA,BALPH,PI
COMMON/CUTOFF/QZ,QZSQ,QZMASS(6),QZFL(7),QZFLSQ(7),QTHRSQ(7)
1,NFLAVT,LGLU
COMMON/FINT/ALLAM2,LFLA
C
A=ALPSQ
PSQ=EXP(A)
CALL ZLIMS(P SQ,QZSQ,ZLW,ZUP,FLAG)
FIGLU=PAPINT(2,ZUP)-PAPINT(2,ZLW)
DO 1 L=1,NFLAVT
  CALL ZLIMS(P SQ,QZFLSQ(L),ZLW,ZUP,FLAG)
  IF(FLAG.GT.0) GO TO 2
  FIGLU=FIGLU+PAPINT(3,ZUP)-PAPINT(3,ZLW)
2 FIGLU=FIGLU/(1-ALLAM2)
RETURN
END
C
SUBROUTINE JETQCD(SETIN)
*****
C-- DEVELOPS TIMELIKE CASCADE OF QUANTA
COMMON/QCD/ALAMB0,CF,CA,BALPH,PI
COMMON/CUTOFF/QZ,QZSQ,QZMASS(6),QZFL(7),QZFLSQ(7),QTHRSQ(7)
1,NFLAVT,LGLU
COMMON/TABPSQ/TABPSQ(5,129),QSQMAX,ALMAX,ALMIN(7),DSTEP(7)
1,NPREC,NTAB,KODTAB(7),MAXTAB
COMMON/PARQUA/PARQUA(79,7),NQUA,MAXQUA
COMMON/EVENT/NEVENT,ECH,PARHAD(401,7),NPART,MXPART,WEIGHT
COMMON/KEYJET/IZFAST
COMMON/MAXN/MNBOOK,MNBOOK,MNPART,MNQUA,MNCUE,MNCUEJ,MNSJO
COMMON/KEYBRE/KEYBRE
C
REAL CUE(25,5)
REAL PROB(7),VZLW(7),VZUP(7),SETIN(6)
DATA NJTQCD/0/, MNQJUE/25/
C
NJTQCD=NJTQCD+1
IF(NJTQCD.GT.1) GO TO 1
CALL INIQCD
CONTINUE

```

```

C
C-- RESET
NCUE=0
C
C-- SETTING PARAMETERS OF INITIAL LEG
IFLA=SETIN(6)
IF(IFLA.EQ.0) IFLA=LGLU
MFLA=IABS(IFLA)
PX=SETIN(1)
PY=SETIN(2)
EPP=SETIN(3)
PSQ=SETIN(5)
IF(KEYBRE.GE.2) PSQ=1.E-5
IF(PSQ.LT.0.)
  ICALL PSQGEN(MFLA,-SETIN(5),PSQ,IGO)
  IF(PSQ.GT.QTHRSQ(MFLA)) GO TO 200
C
C-- NO EMISSION AT ALL. BOOK INITIAL LEG IN PARQUA
CALL JTBOOK(PX,PY,EPP,PSQ,IFLA)
RETURN
C
C-- BRANCHING LOOP
C =====
C
C-- GENERATE Z
C -----
200 CONTINUE
MFLA=IABS(IFLA)
IF(IFLA.EQ.LGLU) GO TO 210
C-- QUARK BRANCHING
CALL ZLIM(PSQ,QZFLSQ(MFLA),QZSQ,ZLM,ZUP,FLAG)
CALL ZGEN(1,ZLM,ZUP,FLAG,Z)
IFLA1=IFLA
IFLA2=LGLU
IF(IFLA.GT.0) GO TO 201
Z=1.-Z
IFLA1=LGLU
IFLA2=IFLA
201 CONTINUE
GO TO 250
C
C-- GLUON BRANCHING - GENERATE DECAY CHANNEL
210 CONTINUE
CALL ZLIMSY(PSQ,QZSQ,ZLM,ZUP,FLAG)
PROGL=PAPINT(2,ZUP)-PAPINT(2,ZLM)
PROB(1)=PROGL
VZLM(1)=ZLM
VZUP(1)=ZUP
DO 211 L=1,NFLAVT
  CALL ZLIMSY(PSQ,QZFLSQ(L),ZLM,ZUP,FLAG)
  IF(FLAG.GT.0.) GO TO 212
  PROQB=PAPINT(3,ZUP)-PAPINT(3,ZLM)
  PROB(L+1)=PROB(1)+PROQB
  VZLM(L+1)=ZLM
  VZUP(L+1)=ZUP
GO TO 211
212 PROB(L+1)=PROB(L)
211 CONTINUE
RR=RNDM(0.)
DO 213 L=1,LGLU
  LOUT=L
  IF(RR.LT.PROB(L)/PROB(LGLU)) GO TO 214
CONTINUE
213 LFLA=LOUT-1
214 FLAG=-1.
  IF(LFLA.GT.0) GO TO 230
C
C-- GLUON BRANCHES INTO TWO GLUONS
ZLM=VZLM(1)
ZUP=VZUP(1)
CALL ZGEN(2,ZLM,ZUP,FLAG,Z)
IFLA1=LGLU
IFLA2=LGLU
GO TO 250
C
C-- GLUON BRANCHES INTO Q-QB
230 ZLM=VZLM(LFLA+1)
  ZUP=VZUP(LFLA+1)
  CALL ZGEN(3,ZLM,ZUP,FLAG,Z)
  IFLA1=LFLA
  IFLA2=LFLA
  GO TO 250
C
C-- GENERATE MASSES OF SECONDARIES
C -----
250 CONTINUE
MFLA1=IABS(IFLA1)
MFLA2=IABS(IFLA2)
002017 IF(IZFAST.EQ.1) GO TO 251
002018 IF(RNDM(0.).GT..5) GO TO 270
002019 GO TO 280
002020 251 IF(Z.GT..5) GO TO 270
002021 GO TO 280
002022 CONTINUE
002023 XP1SQ=Z*(PSQ-QZFLSQ(MFLA2)/(1.-Z))
002024 CALL PSQGEN(MFLA1,XP1SQ,P1SQ,IGO1)
002025 XP2SQ=(1.-Z)*(PSQ-P1SQ/Z)
002026 CALL PSQGEN(MFLA2,XP2SQ,P2SQ,IGO2)
002027 GO TO 290
002028 280 CONTINUE
002029 XP2SQ=(1.-Z)*(PSQ-QZFLSQ(MFLA1)/Z)
002030 CALL PSQGEN(MFLA2,XP2SQ,P2SQ,IGO2)
002031 XP1SQ=Z*(PSQ-P2SQ/(1.-Z))
002032 CALL PSQGEN(MFLA1,XP1SQ,P1SQ,IGO1)
002033 GO TO 290
002034 C
002035 C-- SET TRANSVERSE VARIABLES
002036 C -----
002037 290 CONTINUE
002038 PTRSQ=Z*(1.-Z)*(PSQ-P1SQ/Z-P2SQ/(1.-Z))
002039 PTR=SQR(PTRSQ)
002040 IF(KEYBRE.GE.1) PTR=0.
002041 PHI=2.*PI*RNDM(0.)
002042 PTRX=PTR*COS(PHI)
002043 PTRY=PTR*SIN(PHI)
002044 P1X=Z*PX+PTRX
002045 P1Y=Z*PY+PTRY
002046 P2X=(1.-Z)*PX-PTRX
002047 P2Y=(1.-Z)*PY-PTRY
C
C-- BOOK SECONDARIES, DECIDE WHICH IS NEXT TO CASCADE
C -----
002051 EPP1=EPP*Z
002052 EPP2=EPP*(1.-Z)
002053 IF(IGO1.EQ.0) GO TO 291
C-- 1ST KEEPS CASCAADING
002054 IFLA=IFLA1
002055 PSQ=P1SQ
002056 EPP=EPP1
002057 PX=PIX
002058 PY=PIY
C-- 2ND IN CUE FOR FUTURE CASCAADING
002060 NCUE=NCUE+1
002061 MNUEJ=MAX0(MNUEJ,NCUE)
002062 IF(NCUE.GT.MNUEJ) GO TO 550
002063 IF(IGO2.EQ.0) PSQ=-P2SQ
002064 CUE(NCUE,1)=IFLA2
002065 CUE(NCUE,2)=P2SQ
002066 CUE(NCUE,3)=EPP2
002067 CUE(NCUE,4)=PX
002068 CUE(NCUE,5)=PY
002069 GO TO 200
C
C-- 1ST STOPS, BOOKED IN PARQUA
291 CONTINUE
CALL JTBOOK(PIX,PIY,EPP1,P1SQ,IFLA1)
IF(IGO2.EQ.0) GO TO 293
C-- 2ND KEEPS CASCAADING
IFLA=IFLA2
PSQ=P2SQ
EPP=EPP2
PX=P2X
PY=P2Y
GO TO 200
C-- 2ND STOPS, BOOKED IN PARQUA. LAST IN CUE PICKED UP FOR CASCAADING
293 CONTINUE
CALL JTBOOK(P2X,P2Y,EPP2,P2SQ,IFLA2)
C-- EXIT FROM BRANCHING LOOP, IF THERE IS NO CUE LEFT
295 IF(NCUE.EQ.0) RETURN
IFLA=CUE(NCUE,1)
PSQ=CUE(NCUE,2)
EPP=CUE(NCUE,3)
PX=CUE(NCUE,4)
PY=CUE(NCUE,5)
NCUE=NCUE-1
IF(PSQ.LT.0.) GO TO 294
GO TO 200
C-- LAST IN CUE STOPS, BOOKED IN PARQUA
294 PSQ=-PSQ
CALL JTBOOK(PX,PY,EPP,PSQ,IFLA)
GO TO 295
C
C-- ABNORMAL EXIT
550 N1=NEVENT+1
PRINT 551,MNUEJ,N1,NJTQCD
FORMAT(1H1,39HNUMBER OF QUANTA IN JETQCD CUE EXCEEDS ,I10
002905
002906
002907
002908
002909
002910
002911
002912
002913
002914
002915
002916
002917
002918
002919
002920
002921
002922
002923
002924
002925
002926
002927
002928
002929
002930
002931
002932
002933
002934
002935
002936
002937
002938
002939
002940
002941
002942
002943
002944
002945
002946
002947
002948
002949
002950
002951
002952
002953
002954
002955
002956
002957
002958
002959
002960
002961
002962
002963
002964
002965
002966
002967
002968
002969
002970
002971
002972
002973
002974
002975
002976
002977
002978
002979
002980
002981
002982
002983
002984
002985
002986
002987
002988
002989
002990
002991
002992

```



```

C
C-- G --> Q + QB
3 CONTINUE
  PAPINT=(Z**3-(1.-Z)**3)/6.
  RETURN
C
C  END
C
C  SUBROUTINE ZLIM(PSQ,QZ1SQ,QZ2SQ,ZLW,ZUP,FLAG)
C  *****
C-- CALCULATES LIMITS FOR Z - INPUT, SQUARE MASSES
  FLAG=-1.
  IF(SQRT(PSQ).LT.(SQRT(QZ1SQ)+SQRT(QZ2SQ))) GO TO 10
  B=-11.*QZ1SQ/PSQ-QZ2SQ/PSQ
  C=QZ1SQ/PSQ
  DISCR=B**2/4.-C
  IF(DISCRT.LT.0.) GO TO 10
  SQDIS=SQRT(DISCRT)
  ZLW=-B/2.-SQDIS
  ZUP=-B/2.+SQDIS
  RETURN
10 FLAG=1.
  ZLW=0.5
  ZUP=0.5
  RETURN
END
C
C  SUBROUTINE ZLIMSY(PSQ,QZSQ,ZLW,ZUP,FLAG)
C  *****
C-- CALCULATES LIMITS FOR Z WHEN SECONDARY MASSES ARE EQUAL
C-- INPUT, SQUARE MASSES
  FLAG=-1.
  IF(PSQ.LT.4.*QZSQ) GO TO 10
  D=SQRT(.25-QZSQ/PSQ)
  ZLW=-5*D
  ZUP=-5*D
  RETURN
10 FLAG=1.
  ZLW=-5
  ZUP=5
  RETURN
END
C
C-----
C-- SECTION 1D)
C-- PACKAGE OF ROUTINES HANDLING THE HADRONIZATION OF QUANTA
C-- AND SPECTATOR JETS
C
C-----
C
C  SUBROUTINE HADRON(NQUA1,NQUA2,IMBIAS)
C  *****
C-- HANDLES HADRONIZATION OF FINAL STATE
  COMMON/QCD/ALAMB,CF,CA,BALPH,PI
  COMMON/CUTOFF/QZ,QZSQ,QZMASS(6),QZFL(7),QZFLSQ(7),QTNRSQ(7)
  I NFLAVT,LGLU
  COMMON/TABPSQ/TABPSQ(5,129),QSQMAX,ALMAX,ALMIN(7),DSTEP(7)
  I NPREC,NTAB,KOOTAB(7),MAXTAB
  COMMON/PARQUA/PARQUA(79,7),NQUA,MAXQUA
  COMMON/EVENT/NEVENT,ECH,PARHAD(401,7),NPART,MXPART,WEIGHT
  COMMON/PMBIAS/PMBIAS(5)
  COMMON/HADRON/LHADR(110),LCHARG(110),LCHARM(110),LBOTT(110)
  *,LTOP(110)
  COMMON/MAXN/MNBOOK,MBOOK,MNPART,MNQUA,MNCUE,MNCUEJ,MNSJO
C
C-- COMMON BLOCKS IN SJOSTRAND PROGRAM
  COMMON /PAR/ PU0, PS1, SIGMA, CX2, EBEG, WFIN, IFLBEG
  COMMON /JET/ K(399,2), P(399,5)
  COMMON /EOPAR/ ITHROW, PZMIN, PMIN, THETA, PHI, BETA(3)
  COMMON /DECPAR/ PU0D, PS1D, CND1, CND2, IOB(110)
  COMMON /DATA1/ MESO(36), CMIX(6,2), PHAS(110), QMAS(6)
  COMMON/PHASS/PHASHD(110)
  COMMON/MAXSJO/MAXSJO
  COMMON/KEYBRE/KEYBRE
C
C  DATA ICALL/0/
C
C-- PRELIMINARIES
  IF(ICALL.GT.0) GO TO 11
  ICALL=1
  CALL SBLOCK
  EPS=1.E-7
  UEPS=1.+EPS
C
003169 11 CONTINUE
003170 NPARTL=NPART
003171 NIT=0
003172 15 CONTINUE
003173 NIT=NIT+1
003174 IF(NIT.GT.10) GO TO 122
003175 NPART=NPARTL
003176 IF(NQUA1.GT.NQUA2) GO TO 125
003177 C
003178 DO 100 I=NQUA1,NQUA2
003179 C
003180 C-- PREPARE TO ENTER SJOSTRAND PROGRAM
003181 PP=SQRT(PARQUA(I,1)**2+PARQUA(I,2)**2+PARQUA(I,3)**2)
003182 E=PARQUA(I,4)
003183 EBEG=E*1000.
003184 ITHROW=1
003185 THETA=0.
003186 PHI=0.
003187 IFLBEG=PARQUA(I,6)
003188 CALL JETGEN(N)
003189 CALL EDIT(N)
003190 C
003191 C-- APPROPRIATE ROTATION IS APPLIED
003192 THETA=ACOS(PARQUA(I,3)/PP/UEPS)
003193 IF((PARQUA(I,1)**2+PARQUA(I,2)**2).LT.EPS) GO TO 21
003194 PHI=ATAN2(PARQUA(I,2),PARQUA(I,1))
003195 GO TO 22
003196 21 PHI=0.
003197 22 CONTINUE
003198 IF(PHI.LT.0.) PHI=PHI+2.*PI
003199 ITHROW=0
003200 CALL EDIT(N)
003201 C
003202 C-- BOOKING
003203 IF(N.EQ.0) GO TO 100
003204 IF(NPART.N.GT.MXPART) GO TO 500
003205 DO 41 IH=1,N
003206 IMI=NPART+IH
003207 DO 42 J=1,5
003208 PARHAD(IMI,J)=P(IH,J)/1000.
003209 PARHAD(IMI,6)=K(IH,2)
003210 PARHAD(IMI,7)=ISIGN(I,INT(PARQUA(I,7)))
003211 41 CONTINUE
003212 NPART=NPART+N
003213 100 CONTINUE
003214 IF(IMBIAS.EQ.0) RETURN
003215 C
003216 C-- PREPARE TO CALL MBIAS
003217 125 CONTINUE
003218 DO 200 J=1,3
003219 PMBIAS(J)=0.
003220 PMBIAS(4)=ECH
003221 PMBIAS(5)=PMBIAS(4)
003222 IF(NPART.EQ.0) GO TO 105
003223 DO 201 J=1,6
003224 DO 201 IMI=NPART
003225 PMBIAS(J)=PMBIAS(J)-PARHAD(IMI,J)
003226 IF(PMBIAS(4).LT.0.) GO TO 15
003227 AM2=PMBIAS(4)**2-PMBIAS(1)**2-PMBIAS(2)**2-PMBIAS(3)**2
003228 IF(AM2.LT.0.) GO TO 15
003229 PMBIAS(5)=SQRT(AM2)
003230 IF(PMBIAS(5).LT.ECH*.01.OR.PMBIAS(5).LT.5.) GO TO 15
003231 GO TO 105
003232 122 CONTINUE
003233 WEIGHT=0.
003234 GO TO 90
003235 C
003236 105 CONTINUE
003237 C
003238 CALL MBIAS
003239 C
003240 90 CONTINUE
003241 MNPART=MAX0(MNPART,NPART)
003242 C
003243 RETURN
003244 C
003245 C
003246 C-- ABNORMAL EXIT
003247 500 PRINT 501,MXPART,NEVENT,NQUA
003248 501 FORMAT(5//,1X,20HNUMBER OF PARTICLES EXCEEDS ,I10
003249 1 //1X,11HEVENT NO. = ,I10 ,10X,10HNO. JETS = ,I10
003250 2 //1X,40HINCREASE 1ST DIMENSION OF PARHAD( , ) AND MXPART
003251 3 //1X,40HINCREASE DIMENSIONS OF THV( ) AND Y( ) (SUB. KMD)
003252 4 )
003253 CALL OVERDM
003254 RETURN
003255 END
003256 C

```

```

SUBROUTINE PMBIAS                                003345 113 PARHAD(I,7)=0                                003433
C *****                                003346 110 CONTINUE                                003434
C-- GENERATES SPECTATOR JETS                                003347 RETURN                                003435
COMMON/QCDS/CF,CA,BALPH,PI,PI2,PROGL,PROFLA(2),NFLAVS,LGLUS,IVAL 003348 400 WEIGHT=0.                                003436
COMMON/ALQGEN/WGALQT(127),WGALQ(127,5,3),Q2VECT(128),NALQ,DALQ 003349 RETURN                                003437
1 ,ALAMB,ALLAM2,PTMIN,PTMAX,PT2INT,QZEV,ALQZH,ALQI,ALQF,YF 003350 C-- ABNORMAL EXIT                                003438
2 ,IPBAR,NCASC,SIGINL,PROPT,IPTMTG,PROPTH,PTNGTH,FRHAD,FRAGEN,FRAT003351 500 PRINT 501,MXPART,NEVENT,NQUA                                003439
3 ,ZHCUT,ZLW,ZUP,XMIN 003352 501 FORMAT(5(/,1X,4SHNO. OF PARTICLES DURING PMBIAS CALL EXCEEDS ,                                003440
COMMON/PARQUA/PARQUA(79,7),NQUA,MAXQUA                                003353 1110//1X,11HEVENT NO. = ,110,10X,10HNO. JETS = ,110                                003441
COMMON/ENTREV/S,QSQ,YQSQ,IPACH,XING(2),XMING(2)                                003354 2 //1X,48HINCREASE 1ST DIMENSION OF PARHAD( , ) AND MXPART                                003442
1 ,PXING(2),PYING(2),IFLING(2),ICHING(2)                                003355 3 //1X,49HINCREASE DIMENSIONS OF TMV( ) AND Y( ) (SUB. KNO)                                003443
COMMON/NEVENT/NEVENT,ECM,PARHAD(401,7),NPART,MXPART,WEIGHT                                003356 4 )                                003444
COMMON/PMBIAS/PMBIAS(5)                                003357 CALL OVERDM                                003445
COMMON/JET/ K(399,2), P(399,5)                                003358 RETURN                                003446
COMMON/PAR/ PUD, PSI, SIGMA, CX2, EBEG, WFIN, IFLBEG                                003359 END                                003447
COMMON/DECPAR/ PUDD, PSID, CND1, CND2, IDB(110)                                003360 C SUBROUTINE LOREN4(PMB,V1,V2)                                003448
COMMON/DATA1/ MESO(36), CMIX(6,2), PHAS(110), QMAS(6)                                003361 C *****                                003449
COMMON/PHAS/PHASHD(110)                                003362 C-- LORENTZ BOOST OF V1 TO V2 BY PMB                                003450
COMMON/HB/CLPRO,PT2PRO,PTTHB,FHULMB,RETPIZ,RPAIR(9)                                003363 REAL PMB(4),V1(4),V2(4),BETA(3)                                003451
COMMON/HAXSJO/HAXSJO                                003364 DO 1 J=1,3                                003452
COMMON/HAXN/HNBOOK,MNBOOK,MNPART,MNQUA,MNCUE,MNCUEJ,MNSJO 003365 1 BETA(J)=PMB(J)/PMB(4)                                003453
COMMON/KEYBRE/KEYBRE                                003366 GA=1./SQRT(1.-BETA(1)**2-BETA(2)**2-BETA(3)**2)                                003454
REAL EPRX(2),EMPR(2),PMB(4),V1(4),V2(4),EPRX(2)                                003367 BEP=BETA(1)*V1(1)+BETA(2)*V1(2)+BETA(3)*V1(3)                                003455
INTEGER NLEAD(2)                                003368 DO 2 J=1,3                                003456
C DATA ICALL/0/                                003369 V2(J)=V1(J)+GA*(GA/(1.+GA)*BEP+V1(4))*BETA(J)                                003457
C-- INITIALIZATION                                003370 3 V2(4)=GA*(V1(4)+BEP)                                003458
IF(ICALL.GT.0) GO TO 5                                003371 RETURN                                003459
ICALL=1                                003372 END                                003460
CALL SBLOCK                                003373 C FUNCTION FMUL(AN)                                003461
C=CLPRO                                003374 C *****                                003462
CLEAD=C                                003375 C-- GIVES MULTIPLICITY DISTRIBUTION OF SPECTATOR JET MESONS 003463
CINV=1./C                                003376 COMMON/FFMUL/AVNCH                                003464
EC=1.-EXP(-C)                                003377 Z=AN/AVNCH                                003465
NLEAD(1)=102                                003378 FMUL=FMULOG(Z)                                003466
IF(IPBAR.NE.1) NLEAD(1)=101                                003379 RETURN                                003467
NLEAD(2)=101                                003380 END                                003470
5 CONTINUE                                003381 C FUNCTION FMULOG(Z)                                003471
C *****                                003382 C-- KNO (DE GROOT) MULTIPLICITY FUNCTION                                003472
C-- PAIR OF LEADING NUCLEONS                                003383 FMULOG=1.333*Z/(GAMMA(Z+1))*2*EXP(-.692*Z)*2.5                                003473
EPRX(1)=PMBIAS(4)+PMBIAS(3)                                003384 C *****                                003474
EPRX(2)=PMBIAS(4)-PMBIAS(3)                                003385 C-- TOTAL MULTIPLICITY DISTRIBUTION IN SPECTATOR JETS SAMPLED ACCORDING 003482
NIT=0                                003386 FMULOG=1.333*Z/(GAMMA(Z+1))*2*EXP(-.692*Z)*2.5                                003483
NPARTH=NPART                                003387 RETURN                                003477
NH=NPARTH                                003388 END                                003478
8 CONTINUE                                003389 C SUBROUTINE KNO(M,PMB)                                003479
NPART=NPARTH                                003390 C *****                                003480
DO 10 I=1,2                                003391 C-- HELPS GENERATING SPECTATOR JETS                                003481
IF(NPART+1.GT.MXPART) GO TO 500                                003392 C-- TOTAL MULTIPLICITY DISTRIBUTION IN SPECTATOR JETS SAMPLED ACCORDING 003482
NPART=NPART+1                                003393 C-- TO KNO (DE GROOT)                                003483
9 CONTINUE                                003394 COMMON/QCDS/CF,CA,BALPH,PI,PI2,PROGL,PROFLA(2),NFLAVS,LGLUS,IVAL 003484
NIT=NIT+1                                003395 COMMON/NEVENT/NEVENT,ECM,PARHAD(401,7),NPART,MXPART,WEIGHT 003485
IF(NIT.GT.30) GO TO 400                                003396 COMMON/KEYBRE/KEYBRE                                003486
EPRX(I)=EPRX(I)*(-ALOG(1.-EC*RNOM(I))*CINV)                                003397 COMMON/PHAS/PHASHD(110)                                003487
PT2=PT2PRO+ALOG(RNOM(1+1))                                003398 COMMON/JET/ K(399,2), P(399,5)                                003488
EMPR(I)=(PHASHD(100+I)**2+PT2)/EPRX(I)                                003399 COMMON/PAR/ PUD, PSI, SIGMA, CX2, EBEG, WFIN, IFLBEG 003489
IF(EMPR(I).GT.EPRX(I)) GO TO 9                                003400 COMMON/DECPAR/ PUDD, PSID, CND1, CND2, IDB(110) 003490
PT=SQRT(PT2)                                003401 COMMON/HADQN/LHADQ(110),LCHARG(110),LCHRM(110),LBOTY(110) 003491
PHIP=PI2+RNOM(I+2)                                003402 *,LTOP(110)                                003492
PARHAD(NPART,1)=PT*COS(PHIP)                                003403 COMMON/HB/CLPRO,PT2PRO,PTTHB,FHULMB,RETPIZ,RPAIR(9) 003493
PARHAD(NPART,2)=PT*SIN(PHIP)                                003404 COMMON/HAXSJO/HAXSJO                                003494
PARHAD(NPART,3)=(-1.)*((I+1)*(EPRX(I)-EMPR(I))*5                                003405 COMMON/HAXN/HNBOOK,MNBOOK,MNPART,MNQUA,MNCUE,MNCUEJ,MNSJO 003495
PARHAD(NPART,4)=(EPRX(I)+EMPR(I))*5                                003406 REAL TBPAIR(9)                                003496
PARHAD(NPART,5)=PHASHD(NLEAD(I))                                003407 INTEGER IFTB(2,9)                                003497
PARHAD(NPART,6)=NLEAD(I)                                003408 REAL PMB(4),SUM(2),A(4),V(4),V1(4) 003498
PARHAD(NPART,7)=0                                003409 DATA VPM/0.32,6.65,5.7,5.1,4.84,4.73,4.6/ 003499
10 CONTINUE                                003410 DATA IFTB/51,51,21,22,23,24,25,26,101,102,103,104 003500
C-- DETERMINE 4-MOMENTUM OF SOFT HADRON SYSTEM                                003411 * ,105,106,107,108,109,110/ 003501
IF(IPBAR.EQ.1) PARHAD(NPART,6)=102                                003412 DATA ICALL/0/                                003502
NPARTL=NPART+1                                003413 C IF(ICALL.GT.0) GO TO 10                                003503
DO 11 J=1,4                                003414 ICALL=1                                003504
11 PMB(J)=PMBIAS(J)-(PARHAD(NPART,J)+PARHAD(NPART-1,J)) 003415 TBPAIR(1)=RPAIR(1)                                003505
IF(PMB(4).LT.0.) GO TO 8                                003416 DO 11 M=2,9                                003506
M2=PMB(4)**2-PMB(1)**2-PMB(2)**2-PMB(3)**2                                003417 TBPAIR(M)=RPAIR(M)+TBPAIR(M-1)                                003507
IF(M2.LT.1.) GO TO 8                                003418 DO 12 M=1,9                                003508
C-- CALL KNO TO GENERATE SOFT HADRON SYSTEM                                003419 12 TBPAIR(M)=TBPAIR(M)/TBPAIR(9)                                003509
CALL KNO(SQRT(M2),PMB)                                003420 10 CONTINUE                                003510
C-- LONG LORENTZ BOOST TO LAB SYSTEM                                003421 C IF(M.LT.1.) GO TO 100                                003511
IF(NPART.LT.NPARTL) RETURN                                003422 C-- MEAN MULTIPLICITY                                003512
PMB(1)=0.                                003423 NH=NPARTL                                003513
PMB(2)=0.                                003424 MN=M*1.6                                003514
DO 110 I=NPARTL,NPART                                003425 ALB=2.50+.28*ALOG(MN)+.53*(ALOG(MN))**2 003515
DO 111 J=1,4                                003426 AVN=2.2*ALB                                003516
111 V1(J)=PARHAD(I,J)                                003427 AVN=AVN*FHULMB                                003517
CALL LOREN4(PMB,V1,V2)                                003428                                003518
DO 112 J=1,4                                003429                                003519
112 PARHAD(I,J)=V2(J)                                003430                                003520

```

```

30  CONTINUE
    NMUL=AVN*FRAMUL(0)
    IF(RNDM(0).GT..5) NMUL=NMUL+1
    IF(NMUL.GT.MXPART) GO TO 500
    IF(NMUL.LE.1) GO TO 100
C
C-- GENERATE MULTIPLICITY AND PARTICLE Q.N.
    NCHMB=0
    IT=0
29  CONTINUE
    IT=IT+1
    IF(IT.GT.10) GO TO 30
    I=NM
    IODD=1
    IF((-1)**NMUL.GT.0) IODD=0
    IF(IOND.EQ.0) GO TO 21
    I=I+1
    PARHAD(I,6)=51
21  NPAIR=NMUL/2
    DO 20 N2=1,NPAIR
        RR=RNDM(N2)
        IF(RR.GE.TBPAIR(1)) GO TO 26
        IF(IOND.EQ.0) GO TO 27
        IF(RNDM(N2+3).LT.1./(2.*FLOAT(NMUL))) GO TO 26
27  IPAIR=1
    GO TO 23
26  CONTINUE
    DO 22 M=2,8
        IPAIR=M
        IF(RR.LT.TBPAIR(M)) GO TO 23
22  CONTINUE
    PARHAD(I+1,6)=IFTB(1,IPAIR)
    PARHAD(I+2,6)=IFTB(2,IPAIR)
    IF(IPAIR.NE.1) GO TO 24
    IF(RNDM(N2+1).LT.RETPIZ) PARHAD(I+1,6)=52
    IF(RNDM(N2+2).LT.RETPIZ) PARHAD(I+2,6)=52
24  I=I+2
20  CONTINUE
    NPART=NM+NMUL
    NH1=NM+1
    THASS=0.
    DO 25 I=NH1,NPART
        IPART=PARHAD(I,6)
        IF(1CHARG(IPART).NE.0) NCHMB=NCHMB+1
        PARHAD(I,5)=PMASHD(IPART)
25  THASS=THASS+PARHAD(I,5)
    IF(THASS.GT.W) GO TO 29
C
C-- MEAN PT RISING WITH MULTIPLICITY
    AISIG=FLOAT(NCHMB)/6.
    IF(AISIG.GE.15.) GO TO 31
    RISIG=AISIG/2.5+1.
    ISIG=RISIG
    DELSG=RISIG-FLOAT(ISIG)
    SIGMAB=(VPM(ISIG)*(1.-DELSG)+VPM(ISIG+1)*DELSG)/FPMTB
    GO TO 32
31  SIGMAB=VPM(7)*FPMTB
32  CONTINUE
C
C-- GENERATE PT'S
    DO 17 I=NH1,NPART
        K2=PARHAD(I,6)
        PERPM=X1EXP(SIGMAB*PARHAD(I,5))/SIGMAB
        TH2=PERPM**2
        PT2=TH2-PARHAD(I,5)**2
        PT=AMAX1(PT2,0.)
        PT=SQRT(PT2)
        PHI=PI2*RNDM(0)
        PARHAD(I,1)=PT*COSS(PHI)
        PARHAD(I,2)=PT*SIN(PHI)
17  CONTINUE
C
C-- PT BALANCE WITH PREVIOUS HADRONS
    DO 81 J=1,2
        SUM(J)=PMB(J)
    DO 82 I=NH1,NPART
        SUM(J)=SUM(J)+PARHAD(I,J)
    SUM(J)=SUM(J)/FLOAT(NPART-NH1+1)
    DO 83 I=NH1,NPART
        PARHAD(I,J)=PARHAD(I,J)-SUM(J)
81  CONTINUE
C
C-- TRANSVERSE MASSES
    THTOT=0.
    DO 85 I=NH1,NPART
        THV(I)=SQRT(PARHAD(I,5)**2+PARHAD(I,1)**2+PARHAD(I,2)**2)
85  THTOT=THTOT+THV(I)
003521  EW=SQRT(W**2+PMB(1)**2+PMB(2)**2)
003522  IF(THTOT.GT.EW) GO TO 30
003523  C
003524  C-- GENERATE Y'S
003525  YMAX=ALOG(W/PMASHD(21))
003526  ETOT=0.
003527  DO 86 I=NH1,NPART
003528  YV(I)=YMAX*RNDM(0)
003529  EY=EXP(YV(I))
003530  PARHAD(I,4)=.5*THV(I)*(EY+1./EY)
003531  ETOT=ETOT+PARHAD(I,4)
86  CONTINUE
003532  C
003533  C-- CORRECT Y'S ITERATIVELY SO AS TO GET RIGHT TOTAL ENERGY W
003534  IF(ETOT.LT.EW) GO TO 18
003535  C
003536  IMOVE=-1
003537  YFACT=.5
003538  YFACTX=1.
003539  INIT=1
003540  GO TO 19
003541  C
003542  18  IMOVE=1
003543  YFACT=2.
003544  YFACTX=1.
003545  INIT=2
003546  19  CONTINUE
003547  C
003548  50  CONTINUE
003549  ETOT=0.
003550  DO 51 I=NH1,NPART
003551  Y=V(I)*YFACT
003552  EY=EXP(Y)
003553  PARHAD(I,4)=.5*THV(I)*(EY+1./EY)
003554  ETOT=ETOT+PARHAD(I,4)
003555  51  CONTINUE
003556  IF(ABS(ETOT-EW)/EW.LT..01) GO TO 70
003557  IMOVE=-1
003558  IF(ETOT.LT.EW) IMOVE=1
003559  C
003560  IF(INIT.EQ.3) GO TO 60
003561  IF(INIT.EQ.2) GO TO 55
003562  C
003563  IF(IMOVE.LT.0) GO TO 52
003564  YFACT=YFACT
003565  YFACT=.5*(YFACT+YFACTX)
003566  INIT=3
003567  GO TO 50
003568  52  YFACT=YFACT*.5
003569  GO TO 50
003570  C
003571  55  IF(IMOVE.GT.0) GO TO 56
003572  YFACTX=YFACT
003573  YFACT=.5*(YFACT+YFACTX)
003574  INIT=3
003575  GO TO 50
003576  56  YFACT=YFACT*2.
003577  GO TO 50
003578  C
003579  60  IF(IMOVE.GT.0) YFACT=YFACT
003580  IF(IMOVE.LT.0) YFACTX=YFACT
003581  YFACT=.5*(YFACT+YFACTX)
003582  GO TO 50
003583  C
003584  70  CONTINUE
003585  C
003586  C-- LONGITUDINAL MOMENTUM BALANCE
003587  PLONG=0.
003588  DO 72 I=NH1,NPART
003589  PARHAD(I,3)=0.
003590  PAR3=PARHAD(I,4)**2-THV(I)**2
003591  IF(PAR3.GT.0.) PARHAD(I,3)=SQRT(PAR3)
003592  THV(I)=1.
003593  72  DO 73 IT=NH1,NPART
003594  PLX=0.
003595  DO 74 I=NH1,NPART
003596  IF(THV(I).LT.0.) GO TO 74
003597  IF(PARHAD(I,3).LT.PLX) GO TO 74
003598  IX=I
003599  PLX=PARHAD(I,3)
003600  74  CONTINUE
003601  THV(IX)=-1.
003602  IF(IT.GT.NH1) GO TO 75
003603  IF(RNDM(IT).GT..5) PARHAD(IX,3)=-PARHAD(IX,3)
003604  75  CONTINUE
003605  IF(PLONG.GT.0.) PARHAD(IX,3)=-PARHAD(IX,3)
003606  PLONG=PLONG+PARHAD(IX,3)
003607  73  CONTINUE
003608
003609
003610
003611
003612
003613
003614
003615
003616
003617
003618
003619
003620
003621
003622
003623
003624
003625
003626
003627
003628
003629
003630
003631
003632
003633
003634
003635
003636
003637
003638
003639
003640
003641
003642
003643
003644
003645
003646
003647
003648
003649
003650
003651
003652
003653
003654
003655
003656
003657
003658
003659
003660
003661
003662
003663
003664
003665
003666
003667
003668
003669
003670
003671
003672
003673
003674
003675
003676
003677
003678
003679
003680
003681
003682
003683
003684
003685
003686
003687
003688
003689
003690
003691
003692
003693
003694
003695
003696

```



```

C-- CALL CORRECTIVE ROUTINE
IP=0
00 76 I=NH1,NPART
IP=IP+1
00 76 J=1,5
76 P(IP,J)=PARHAD(I,J)
CALL P4CORR(EW,PMB,IP,IFLAG)
IF(IFLAG.NE.0) GO TO 30
IP=0
DO 77 I=NH1,NPART
IP=IP+1
DO 77 J=1,5
77 PARHAD(I,J)=P(IP,J)
GO TO 200

C
C-- VERY SMALL M, EMIT TWO P10
100 CONTINUE
SIGHAB=VPM(1)/FPTHB
EW=SQRT(W**2+PMB(1)**2+PMB(2)**2)
A(1)=PMB(1)
A(2)=PMB(2)
A(3)=0.
A(4)=EW
NH1=NH+1
NPART=NP+2
IF(NPART.GT.MXPART) GO TO 500
PERPH=XIEXP(SIGHAB*PMASHO(51))/SIGHAB
PL2=M**2*.25-PERPH**2
IF(PL2.LT.0.) GO TO 101
PL=SQRT(PL2)
PT2=PERPH**2-PMASHO(51)**2
PT=SQRT(PT2)
PHI=PI2*RNDRM(0)
DO 78 I=1,2
V(1)=PT*COS(PHI)*(-1)**I
V(2)=PT*SIN(PHI)*(-1)**I
V(3)=PL*(-1)**I
V(4)=SQRT(PMASHO(51)**2+PT2+PL2)
CALL LOREN4(A,V,V1)
DO 79 J=1,4
79 PARHAD(NH+I,J)=V(I,J)
PARHAD(NH+I,5)=PMASHO(51)
PARHAD(NH+I,6)=51
CONTINUE

78 C
C-- OPTIONAL DECAY OF P10'S
200 CONTINUE
ND1=NH1
ND2=NPART
I=0
DO 201 M=ND1,ND2
I=I+1
DO 202 J=1,5
202 P(I,J)=PARHAD(M,J)*1000.
K(I,2)=PARHAD(M,6)
201 CONTINUE
IST=I
IPD=0
203 IPD=IPD+1
IF(IGB(K(IPD,2)).GT.0) CALL DECAY(IPD,I)
IF(IPD.LT.I.AND.I.LT.(MAXSJO-31)) GO TO 203
IF(I.GT.(MAXSJO-4)) GO TO 550
MNSJO=MAX0(MNSJO,I)
IF(I.EQ.IST) GO TO 206
NPART=NH
DO 204 M=1,I
IF(IGB(K(M,2)).GT.0) GO TO 204
IF(NPART+1.GT.MXPART) GO TO 500
NPART=NPART+1
DO 205 J=1,5
205 PARHAD(NPART,J)=P(M,J)/1000.
PARHAD(NPART,6)=K(M,2)
204 CONTINUE
206 CONTINUE
RETURN

C-- ABNORMAL EXIT
500 PRINT 501,MXPART,NEVENT
501 FORMAT(5//,IX,4SHNO. OF PARTICLES DURING KNO CALL EXCEEDS ,
110//IX,11HEVENT NO. = ,I10
2 //IX,48HINCREASE 1ST DIMENSION OF PARHAD( , ) AND MXPART
3 //IX,50HINCREASE DIMENSIONS OF THV( ) AND YV( ) (SUB. KNO)
4 )
CALL OVERDM
RETURN

C
C-- ABNORMAL EXIT
550 PRINT 551,MAXSJO,NEVENT
551 FORMAT(5//,IX,4SHNO. OF PARTICLES IN SJOSTRAND P( , ) EXCEEDS ,

```

05	SUM3=0. SUM4=0. DO 85 I=1,NP SUM3=SUM3+P(I,3) P(I,4)=SQRT(P(I,3)**2+TM2(I)) SUM4=SUM4+P(I,4) F3=SUM3/SUM4 SUM3=0. DO 86 I=1,NP P(I,3)=P(I,3)-F3*P(I,4) SUM3=SUM3+ABS(P(I,3)) WMINSQ=WMINSQ FF=MSQ-WMINSQ F2=SQRT(FF)/SUM3 DO 87 IT=1,20 TEST=0. SUM3=0. DO 88 I=1,NP P(I,3)=F2*P(I,3) SUM3=SUM3+ABS(P(I,3)) P(I,4)=SQRT(P(I,3)**2+TM2(I)) TEST=TEST+P(I,4) CONTINUE IF(ABS(TEST-W).LT.0.01) GO TO 89 F2=W/TEST GG=TEST**2-WMINSQ IF(GG.GT.0.) F2=SQRT(FF/GG) CONTINUE IFLAG=1 CONTINUE RETURN END	003873 003874 003875 003876 003877 003878 003879 003880 003881 003882 003883 003884 003885 003886 003887 003888 003889 003890 003891 003892 003893 003894 003895 003896 003897 003898 003899 003900 003901 003902 003903 003904 003905 003906 003907 003908 003909 003910 003911 003912 003913 003914 003915 003916 003917 003918 003919 003920 003921 003922 003923 003924 003925 003926 003927 003928 003929 003930 003931 003932 003933 003934 003935 003936 003937 003938 003939 003940 003941 003942 003943 003944 003945 003946 003947 003948 003949 003950 003951 003952 003953 003954 003955 003956 003957 003958 003959 003960	C-- 100 2 3 4 5 C C-- 110 1 2 3 C 111 16 37 4 5 C 112 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 61
----	--	--	---

```

PT=SQRT(PBOS(1)**2+PBOS(2)**2) 004049 IF(IAOR.NE.IQ) GO TO 45 004137
PP=SQRT(PT**2+PBOS(3)**2) 004050 M=N+1 004138
THETA=ACOS(PBOS(3)/PP)*180./PI 004051 DO 46 J=1,5 004139
Y=SIGN(-ALOG(TAN(ATANI(PT/ABS(PBOS(3)))/2.)),PBOS(3)) 004052 P(N,J)=PARHAD(I,J) 004140
PHI=ATAN2(PBOS(2),PBOS(1))*180./PI 004053 K(N,1)=0 004141
IF(PHI.LT.0.) PHI=PHI+360. 004054 K(N,2)=PARHAD(I,6) 004142
XF=PBOS(3)/(1.5*ECM) 004055 45 CONTINUE 004143
PRINT 5,WEAKON,PBOS(5),PBOS(4),(PBOS(J),J=1,3) 004056 IF(N.EQ.0) GO TO 40 004144
* ,PT,XF,Y,THETA,PHI 004057 CALL LIST(N) 004145
5 FORMAT(///1X,A2,17H BOSON PARAMETERS//5X, 004058 40 CONTINUE 004146
16HMASS =,1X,E10.3,5X,3HE =,1X,E10.3,5X,4HPX =,1X,E10.3,5X,4HPY =, 004059 C 004147
21X,E10.3 ,5X,4HPZ =,1X,E10.3//5X,4HPT =,1X,E10.3,5X,4HFX =,1X, 004060 C-- SOFT PARTICLE SYSTEM 004148
3E10.3 ,5X,3HY =,1X,E10.3,5X,7HTHETA =,1X,E10.3,5X, 004061 100 CONTINUE 004149
35HPHI =,1X,E10.3//) 004062 IF(SPYQ.GT.0.) RETURN 004150
IF(IDBI(INT(PBOS(6))).EQ.0) PRINT 7,WEAKON 004063 PRINT 101,PHIBIAS 004151
7 FORMAT(1X,A2,1X,22HREQUESTED UNDECAYED ) 004064 101 FORMAT(///1X,14HSPECTATOR JETS 004152
IF(IDBI(INT(PBOS(6))).EQ.0) GO TO 31 004065 1 //1X,44HKINEMATIC PARAMETERS OF SOFT PARTICLE SYSTEM 004153
PRINT 6,WEAKON 004066 2 ///10X,4HPX =,E10.3,10X,4HPY =,E10.3,10X,4HPZ =,E10.3 004154
6 FORMAT(1X,A2,1X,20HBOSON DECAY PRODUCTS ) 004067 3 ///10X,3HE =,E10.3,10X,16HINVARIANT MASS =,E10.3 004155
IDC=ICDDB 004068 N=0 004156
IFBOS=PBOS(6) 004069 DO 105 I=NPARTH,NPART 004157
CALL CHDECIFBOS,IDC,CH) 004070 IOR=PARHAD(I,7) 004158
PRINT 51,CHA2(IFBOS),(CH(J),J=1,2),ICDDB 004071 IF(IAOR.NE.0) GO TO 105 004159
51 FORMAT(///1X,A4,8H ---+ ,A4,2X,A4,5X,12H(CHANNEL NO. ,I3,1H) ) 004072 M=N+1 004160
IF(WEAKON.EQ.ZH.AND.ICDDB.GT.155) GO TO 30 004073 DO 106 J=1,5 004161
IF(WEAKON.EQ.WH.AND.ICDDB.GT.172) GO TO 30 004074 106 P(N,J)=PARHAD(I,J) 004162
NPARTH=NP+1 004075 K(N,1)=0 004163
DO 15 IB=1,2 004076 K(N,2)=PARHAD(I,6) 004164
N=0 004077 CONTINUE 004165
DO 10 I=1,NPART 004078 IF(N.NE.0) CALL LIST(N) 004166
IOR=PARHAD(I,7) 004079 C 004167
IF(IAOR.NE.IB) GO TO 10 004080 PRINT 107 004168
M=N+1 004081 107 FORMAT(15(/)) 004169
DO 11 J=1,5 004082 C 004170
P(N,J)=PARHAD(I,J) 004083 RETURN 004171
K(N,1)=0 004084 END 004172
K(N,2)=PARHAD(I,6) 004085 C 004173
10 CONTINUE 004086 SUBROUTINE CHDEC(IP,IDC,CH) 004174
CALL LIST(N) 004087 ***** 004175
15 CONTINUE 004088 C-- GIVES DESCRIPTION OF DECAY CHANNEL 004176
C JETS 004089 COMMON /DATA2/ CBR(180), KDP(180,3), WTCOR(10) 004177
30 IF(NQUA.EQ.0) GO TO 100 004090 COMMON /DATA3/ CHA1(36), CHA2(110), CHA3(2) 004178
IF((WEAKON.EQ.ZH.AND.ICDDB.GT.155).OR.(WEAKON.EQ.WH.AND.ICDDB.GT 004091 INTEGER JCNPAR(110),JCNQOB(36) 004179
*.172)) PRINT 34,WEAKON 004092 REAL PAIRS(36) 004180
34 FORMAT(///1X,11H(JETS FROM ,A2,12HDECAY FIRST) ) 004093 REAL CH(3) 004181
31 CONTINUE 004094 C 004182
IF(NQUA.EQ.0) GO TO 100 004095 DATA BLANK/4H / 004183
IQMX=NQUA 004096 DATA PAIRS/4HU-UB,4HU-DB,4HU-SB,4HU-CB,4HU-BB,4HU-TB 004184
DO 32 IQ=1,NQUA 004097 1 ,4HD-UB,4HD-DB,4HD-SB,4HD-CB,4HD-BB,4HD-TB 004185
IF(PARQUA(IQ,7).GT.0.) GO TO 33 004098 2 ,4HS-UB,4HS-DB,4HS-SB,4HS-CB,4HS-BB,4HS-TB 004186
32 IQMX=IQ 004099 3 ,4HC-UB,4HC-DB,4HC-SB,4HC-CB,4HC-BB,4HC-TB 004187
33 CONTINUE 004100 4 ,4HB-UB,4HB-DB,4HB-SB,4HB-CB,4HB-BB,4HB-TB 004188
DO 40 IQC=1,NQUA 004101 5 ,4HT-UB,4HT-DB,4HT-SB,4HT-CB,4HT-BB,4HT-TB/ 004189
IQ=IQC 004102 DATA ICALL/0/ 004190
IF(IPRO.EQ.2.AND.IQC.LE.NQUA-IQMX) IQ=IQMX+IQC 004103 C 004191
IF(IPRO.EQ.2.AND.IQC.GT.NQUA-IQMX) IQ=IQC-(NQUA-IQMX) 004104 IF(ICALL.GT.0) GO TO 10 004192
PT=SQRT(PARQUA(IQ,1)**2+PARQUA(IQ,2)**2) 004105 ICALL=1 004193
PP=SQRT(PT**2+PARQUA(IQ,3)**2) 004106 CALL SBLOCK 004194
THETA=ACOS(PARQUA(IQ,3)/PP)*180./PI 004107 C 004195
Y=SIGN(-ALOG(TAN(ATANI(PT/ABS(PARQUA(IQ,3)))/2.)),PARQUA(IQ,3)) 004108 DO 21 I=1,110 004196
PHI=ATAN2(PARQUA(IQ,2),PARQUA(IQ,1))*180./PI 004109 JCNPAR(I)=I 004197
IF(PHI.LT.0.) PHI=PHI+360. 004110 IF(I.LE.2.OR.(I.GE.5.AND.I.LE.6) 004198
IFLA=PARQUA(IQ,6) 004111 * .OR.(I.GE.51.AND.I.LE.60).OR.I.GE.91) GO TO 21 004199
IF(FLA.GT.0) FLA=QFLA(IFLA) 004112 IF((-1.)#MI.LT.0.) JCNPAR(I)=I+1 004200
IF(FLA.LT.0) FLA=QBFLA(-IFLA) 004113 IF((-1.)#MI.GT.0.) JCNPAR(I)=I-1 004201
IF(FLA.EQ.0) FLA=GLUON 004114 CONTINUE 004202
IGOR=PARQUA(IQ,7) 004115 DO 22 JI=1,6 004203
PRINT 35,IQ,FLA,PARQUA(IQ,4) 004116 DO 22 J2=1,6 004204
1 , (PARQUA(IQ,J),J=1,3) 004117 I=6*(JI-1)+J2 004205
35 FORMAT(///1X,8HJET NO. ,I3,15H FLAVOR = ,A2, 004118 IB=6*(J2-1)+J1 004206
1 9H E = ,E10.3 004119 22 JCNQOB(I)=IB 004207
2 //34X,10H PJX = ,E10.3,10H PJY = ,E10.3,10H PJZ =, 004120 10 CONTINUE 004208
2E10.3) 004121 C 004209
PRINT 36,PT,Y,THETA,PHI 004122 DO 6 J=1,3 004210
36 FORMAT(18X,4HET =,E10.3,5X,3HY =,E10.3,5X,7HTHETA =,E10.3 004123 JJ=KDP(IOC,J) 004211
1 ,5X,5HPHI =,E10.3) 004124 IF(IP.EQ.4.AND.JJ.GT.0) JJ=JCNPAR(JJ) 004212
IF(IGOR.LT.0) PRINT 36 004125 IF(IP.EQ.4.AND.JJ.LT.0) JJ=-JCNQOB(-JJ) 004213
IF(IPRO.EQ.1.AND.IGOR.GT.0) PRINT 37 004126 IF(JJ.EQ.0) CH(J)=BLANK 004214
IF(IPRO.EQ.2.AND.IGOR.GT.0) PRINT 37,WEAKON 004127 IF(JJ.GT.0) CH(J)=CHA2(JJ) 004215
36 FORMAT(//18X,24H(INITIAL BREMSSTRAHLUNG) ) 004128 IF(JJ.LT.0) CH(J)=PAIRS(-JJ) 004216
37 FORMAT(//18X,20H(PARTON FINAL STATE) ) 004129 6 CONTINUE 004217
39 FORMAT(//18X,1H( ,A4,2X,6HDECAY) ) 004130 C 004218
IF(SPYQ.GT.0..AND.(IPRO.NE.2.OR.PARQUA(IQ,7).LT.0.)) GO TO 40 004131 IF(IP.NE.4.OR.IOC.GT.172) GO TO 30 004219
N=0 004132 CH2=CH(1) 004220
DO 45 I=NPARTH,NPART 004133 CH1=CH(2) 004221
IOR=PARHAD(I,7) 004134 CH(1)=CH1 004222
IAOR=IABS(IOR) 004135 CH(2)=CH2 004223
004136 30 CONTINUE 004224

```

```

C
RETURN
END
C-----
C-- SECTION (D1)
C-- PACKAGE OF ROUTINES FOR QUARK JET GENERATION ACCORDING TO THE FIELD-
C-- FEYNMAN CASCADE MODEL (T.SJOSTRAND, UNIV. LUND PREPRINT LU TP 79-8
C-- (1979), WITH MINOR REVISIONS).
C-----
C
SUBROUTINE JETGEN(N)
*****
COMMON /JET/ K(399,2), P(399,5)
COMMON /PAR/ PUD, PSI, SIGMA, CX2, EBEG, WFIN, IFLBEG
COMMON /DECPAR/ PUDD, PS10, CND1, CND2, IDB(110)
COMMON /DATA1/ HESO(36), CHIX(6,2), PHAS(110), QMAS(6)
COMMON /MAXSJO/ MAXSJO
COMMON /SHEAVY/ IHEV(399)
COMMON /EVENT/ NEVENT, ECH, PARHAD(401,7), NPART, HXPART, WEIGHT
COMMON /HBOOK/ HBOOK, HBOOK, HNPART, HMQUA, HMCUE, HMCUEJ, HNSJO
COMMON /PETER/ EPSPET
REAL TG(100), TGV(100,3)
INTEGER LG(100), LSV(100,3)
DATA ICALL/0/
IF(IFLBEG.EQ.0) IFLBEG=(3.*RNDM(0)+1.)*(-1.)*INT(2.*RNDM(1)+1.)
C
C
IF(ICALL.GT.0) GO TO 50
ICALL=1
CALL SBLOCK
NG=100
DG=1./FLOAT(NG)
HOG=DG/2.
DO 10 JG=1,3
TTG=0.
RM2=(QMAS(4)/QMAS(JG+3))*2
DO 11 IG=1,NG
XG=DG*FLOAT(IG-1)+HOG
TG(IG)=1./((XG*(1.-1./XG-EPSPET*RM2/(1.-XG))*2)
11 TTG=TTG+TG(IG)
DO 12 IG=1,NG
TG(IG)=TG(IG)/TTG
CALL ALIAS(LG,TG,NG)
DO 13 IG=1,NG
TGV(IG,JG)=TG(IG)
13 LGV(IG,JG)=LG(IG)
10 CONTINUE
50 CONTINUE
C
C 1 INITIAL VALUES, PT FOR FIRST QUARK
DO 201 L=1,MAXSJO
201 IHEV(L)=0
I=0
IPD=0
H=2.*EBEG
IFL=IFLBEG
PT1=SIGMA*SQRT(-ALOG(RNDM(0)))
PHI1=6.2832*RNDM(0)
PX1=PT1*COS(PHI1)
PY1=PT1*SIN(PHI1)
C 2 FLAVOUR AND PT FOR NEXT ANTIQUARK
100 I=1
IFL2=ISIGN(1+INT(RNDM(0)/PUD),-IFL1)
PT2=SIGMA*SQRT(-ALOG(RNDM(0)))
PHI2=6.2832*RNDM(0)
PX2=PT2*COS(PHI2)
PY2=PT2*SIN(PHI2)
C 3 MESON FORMED, SPIN ADDED AND FLAVOUR MIXED
K(I,1)=HESO(6*(MAXO(IFL1,IFL2)-1)-MINO(IFL1,IFL2))
ISPIN=INT(PS1+RNDM(0))
K(I,2)=20+40*ISPIN+K(I,1)
IF(K(I,1).GE.31.AND.K(I,1).LE.33) GO TO 802
GO TO 801
802 CONTINUE
THIX=RNDM(0)
KM=K(I,1)-30+3*ISPIN
K(I,2)=51+40*ISPIN+INT(THIX*CHIX(KM,1))+INT(THIX*CHIX(KM,2))
801 CONTINUE
C 4 MESON MASS FROM TABLE, PT FROM CONSTITUENTS
PI(5)=PHAS(K(I,2))
PI(1)=PX1+PX2
PI(2)=PY1+PY2
PHTS=PI(1)*2+PI(2)*2+PI(5)*2
C 5 SCALING IN X GIVES E AND PZ, CUT IN PZ FOR FIRST MESON
110 X=RNDM(0)
004225 C
004226 IF(IABS(IFL1).GT.3) GO TO 51
004227 IF(RNDM(0).LT.CX2) X=1.-X*(1./3.)
004228 GO TO 60
004229 51 JG=IABS(IFL1)-3
004230 IXG=X*FLOAT(NG)+1.
004231 IF(RNDM(0).GT.TGV(IXG,JG)) IXG=LGV(IXG,JG)
004232 X=DG*FLOAT(IXG-1)+HOG
004233 60 CONTINUE
004234 C
004235 P(I,3)=(X*W-PHTS/(X*W))/2.
004236 P(I,4)=(X*W+PHTS/(X*W))/2.
004237 C IF(I.EQ.1.AND.P(I,3).LE.0.) GOTO 110
004238 C 6 IF UNSTABLE, DECAY CHAIN INTO STABLE PARTICLES
004239 IL=I
004240 CALL CHEAVY(IL)
004241 120 IPD=IPD+1
004242 IF(IDB(K(IPD,2)).GT.0) CALL DECAY(IPD,I)
004243 IF(IPD.LT.1.AND.I.LT.(MAXSJO-3)) GOTO 120
004244 IF(IHEV(IL).NE.1) GO TO 202
004245 DO 203 L=IL,I
004246 IHEV(L)=1
004247 202 CONTINUE
004248 C 7 FLAVOUR AND PT FOR QUARK FORMED IN PAIR WITH ANTIQUARK ABOVE
004249 IFL1=-IFL2
004250 PX1=-PX2
004251 PY1=-PY2
004252 C 8 IF ENOUGH E+PZ LEFT, GO TO 2
004253 H=(1.-X)*W
004254 C-----
004255 C-- MODIFIED -- THRON AWAY UNSTABLE PARTICLES IN INTERMEDIATE STEPS OF
004256 C-- THE JET CASCADE
004257 C-----
004258 II=0
004259 DO 210 IZ=1,I
004260 IF(IDB(K(IZ,2)).GT.0) GO TO 210
004261 II=II+1
004262 K(II,1)=K(IZ,1)
004263 K(II,2)=K(IZ,2)
004264 DO 211 J=1,5
004265 211 P(II,J)=P(IZ,J)
004266 IHEV(II)=IHEV(IZ)
004267 210 CONTINUE
004268 I=II
004269 IPD=I
004270 C
004271 IF(I.GT.(MAXSJO-4)) GO TO 500
004272 IF(M.GT.WFIN) GOTO 100
004273 N=I
004274 HNSJO=MAXO(HNSJO,N)
004275 RETURN
004276 C
004277 C-- ABNORMAL EXIT
004278 500 CONTINUE
004279 EBEG=EBEG/1000.
004280 PRINT 501,MAXSJO,IFLBEG,EBEG,NEVENT
004281 501 FORMAT(5I/,1X,4SHND. OF ALL PARTICLES IN SJOSTRAND JET EXCEEDS,
004282 1I0//1X,7HFLAVOR ,1I0,10X,4HE = ,E10.3
004283 2 //1X,1IHEVENT NO. = ,I10
004284 3//1X,5SHINCREASE 1ST DIMENSIONS OF K( , ) AND P( , ) AND MAXSJO ,
004285 4//1X,52HINCREASE DIMENSIONS OF IHEV( ) (/CANC/) AND TH2V( ) ,
004286 513H(SUB. P4CORR)
004287 CALL OVEROM
004288 RETURN
004289 END
004290 C
004291 SUBROUTINE DECAY(IPD,I)
004292 *****
004293 COMMON /JET/ K(399,2), P(399,5)
004294 COMMON /DECPAR/ PUDD, PS10, CND1, CND2, IDB(110)
004295 COMMON /DATA1/ HESO(36), CHIX(6,2), PHAS(110), QMAS(6)
004296 COMMON /DATA2/ CBR(180), KDP(180,3), WTCOR(10)
004297 COMMON /EVENT/ NEVENT, ECH, PARHAD(401,7), NPART, HXPART, WEIGHT
004298 COMMON /MAXSJO/ MAXSJO
004299 COMMON /HBOOK/ HBOOK, HBOOK, HNPART, HMQUA, HMCUE, HMCUEJ, HNSJO
004300 REAL PM(10,5), RND(10), U(3), BE(3)
004301 INTEGER IFL(4), IFLR(4)
004302 PAMT(A,B,C)=SQRT((A**2-(B+C)*2)*(A**2-(B-C)*2))/(2.*A)
004303 FOUR(I,J)=PI(I,4)*PI(J,4)-PI(I,1)*PI(J,1)-PI(I,2)*PI(J,2)-PI(I,3)*PI(J,3)
004304 C 1 DECAY CHANNEL CHOICE, GIVES DECAY PRODUCTS
004305 TBR=RNDM(0)
004306 IDC=IDB(K(IPD,2))-1
004307 100 IDC=IDC+1
004308 IF(TBR.GT.CBR(IDC)) GOTO 100
004309 NP=0
004310 NQ=0
004311 PSUM=0.
004312 IILW=I+1
004313
004314
004315
004316
004317
004318
004319
004320
004321
004322
004323
004324
004325
004326
004327
004328
004329
004330
004331
004332
004333
004334
004335
004336
004337
004338
004339
004340
004341
004342
004343
004344
004345
004346
004347
004348
004349
004350
004351
004352
004353
004354
004355
004356
004357
004358
004359
004360
004361
004362
004363
004364
004365
004366
004367
004368
004369
004370
004371
004372
004373
004374
004375
004376
004377
004378
004379
004380
004381
004382
004383
004384
004385
004386
004387
004388
004389
004390
004391
004392
004393
004394
004395
004396
004397
004398
004399
004400

```

```

      IIUP=I+3
      DO 110 I1=IILW,IIUP
      K(I1,2)=KOP(IOC,I1-I)
      IF(K(I1,2).GT.0) GO TO 812
      GO TO 811
812  CONTINUE
      NP=NP+1
      K(I1,1)=-IPD
      P(I1,5)=PMAS(K(I1,2))
      PSUMP=PSUMP+P(I1,5)
      GO TO 813
811  IF(K(I1,2).LT.0) GO TO 814
      GO TO 813
814  CONTINUE
      NQ=NQ+1
      IFLQ(2*NQ-1)=(5-K(I1,2))/6
      IFLQ(2*NQ)=K(I1,2)+6*(IFLQ(2*NQ-1)-1)
813  CONTINUE
110  CONTINUE
C 2 DECAY MULTIPLICITY
      ND=ND+ND
      IF(I+ND.GT.MAXSJO) GO TO 500
      PSUM=PSUMP
      IF(NQ.EQ.0) GOTO 170
      CND=CND1*ALOG(P(IPD,5)/CND2)
      GAUSS=SQRT(-2.*CND*ALOG(RNDM(0)))*SIN(6.2832*RNDM(0))
120  GAUSS=SQRT(-2.*CND*ALOG(RNDM(0)))*SIN(6.2832*RNDM(0))
      ND=0.5+0.5*(NP+NQ)+CND*GAUSS
      IF(I+ND.GT.MAXSJO) GO TO 500
      IF(ND.LT.NP+NQ.OR.ND.LT.2.OR.ND.GT.10) GOTO 120
      IF(NP+NQ.EQ.3.AND.K(I+2,2).GE.7.AND.K(I+2,2).LE.16) ND=3
      IF(I+ND.GT.MAXSJO) GO TO 500
C 3 GENERATION AND PAIRING OFF OF QUARK-ANTIQUARK PAIRS
      DO 130 J=1,4
130  IFLR(J)=IFLQ(J)
      IF(ND.EQ.NP+NQ) GOTO 150
      IILW=I+NP+1
      IIUP=I+ND+NQ
      DO 140 I1=IILW,IIUP
      IQ=1+INT(2*NQ*RNDM(0))
      IFLN=SIGN(1+INT(RNDM(0)/PUDD),-IFLR(IQ))
      K(I1,1)=MESO(6*(MAX(0,IFLR(IQ),IFLN)-1)-MIN(0,IFLR(IQ),IFLN))
140  IFLR(IQ)=-IFLN
150  IQ=2+2*INT(NQ*RNDM(0))
      K(I+ND-NQ+1,1)=MESO(6*(IFLR(1)-1)-IFLR(IQ))
      IF(NQ.EQ.2) K(I+ND,1)=MESO(6*(IFLR(3)-1)-IFLR(6-IQ))
C 4 MESONS ARE FORMED, IF TOTAL MASS TOO LARGE GO TO 2
      PSUM=PSUMP
      IILW=I+NP+1
      IIUP=I+ND
      DO 160 I1=IILW,IIUP
      ISPIN=INT(PSID+RNDM(0))
      K(I1,2)=20+40*ISPIN+K(I1,1)
      IF(K(I1,1).GE.31.AND.K(I1,1).LE.33) GO TO 822
      GO TO 821
822  CONTINUE
      THIX=RNDM(0)
      KM=K(I1,1)-30+3*ISPIN
      K(I1,2)=51+40*ISPIN+INT(THIX+CHIX(KM,1))*INT(THIX+CHIX(KM,2))
821  CONTINUE
      K(I1,1)=-IPD
      P(I1,5)=PMAS(K(I1,2))
160  PSUM=PSUMP+P(I1,5)
      IF(PSUM+5.*GT.P(IPD,5)) GOTO 120
C 5 PRODUCT CHARGE CONJUGATION FOR DECAYING ANTIPARTICLE
170  IF(2*(K(IPD,2)/2).EQ.K(IPD,2)) GO TO 832
      GO TO 831
832  CONTINUE
      IILW=I+1
      IIUP=I+ND
      DO 180 I1=IILW,IIUP
      KCOR=K(I1,2)-(-1)**(K(I1,2)-2*(K(I1,2)/2))
      IF(K(I1,2).GE.7.AND.K(I1,2).LE.50) K(I1,2)=KCOR
180  IF(K(I1,2).GE.61.AND.K(I1,2).LE.90) K(I1,2)=KCOR
831  CONTINUE
C 6 ONE-PARTICLE DECAY, START NO-PARTICLE DECAY
      IF(ND.EQ.1) GO TO 842
      GO TO 841
842  CONTINUE
      DO 190 J=1,4
190  P(I+1,J)=P(IPD,J)
      GOTO 330
841  CONTINUE
      DO 200 J=1,5
200  PH(I,J)=P(IPD,J)
      PH(ND,5)=P(I+ND,5)
      IF(ND.EQ.2) GOTO 260
C 7 CALCULATION OF MAXIMUM WEIGHT
      WTHAX=1./WTCOR(ND)
004401  PMAX=PH(1,5)-PSUM+P(I+ND,5)
004402  PHIN=0.
004403  MLW=1
004404  MLUP=ND-1
004405  DO 210 ML=MLW,MLUP
004406  IL=MLUP-ML+1
004407  PMAX=PMAX+P(I+IL,5)
004408  PHIN=PHIN+P(I+IL,5)
004409  210 WTHAX=WTHAX*PMT(PMAX,PHIN,P(I+IL,5))
004410  C 8 M-GENERATOR GIVES WEIGHT, IF REJECTED TRY AGAIN
004411  220 RND(1)=1.
004412  IILW=2
004413  ILUP=ND-1
004414  DO 240 I1=IILW,IIUP
004415  RAN=RNDM(0)
004416  ML2LW=1
004417  ML2UP=I1-1
004418  DO 230 ML2=ML2LW,ML2UP
004419  IL2=ML2UP-ML2+1
004420  IF(RAN.LE.RND(IL2)) GOTO 240
004421  230 RND(IL2+1)=RND(IL2)
004422  240 RND(IL2+1)=RAN
004423  RND(ND)=0.
004424  WT=1
004425  MLW=1
004426  MLUP=ND-1
004427  DO 250 ML=MLW,MLUP
004428  IL=MLUP-ML+1
004429  PHIL(5)=PH(IL+1,5)+P(I+IL,5)+(RND(IL)-RND(IL+1))*P(IPD,5)-PSUM
004430  250 WT=WT*PMT(PHIL(5),PH(IL+1,5),P(I+IL,5))
004431  IF(WT.LT.RNDM(0)*WTHAX) GOTO 220
004432  C 9 TWO-PARTICLE DECAY IN CM
004433  260 CONTINUE
004434  IILW=1
004435  ILUP=ND-1
004436  DO 280 IL=IILW,IIUP
004437  PA=PMT(PHIL(5),PH(IL+1,5),P(I+IL,5))
004438  U(3)=2.*RNDM(0)-1.
004439  PHI=6.2832*RNDM(0)
004440  U(1)=SQRT(1.-U(3)**2)*COS(PHI)
004441  U(2)=SQRT(1.-U(3)**2)*SIN(PHI)
004442  DO 270 J=1,3
004443  P(I+IL,J)=PA*U(J)
004444  270 PH(IL+1,J)=PA*U(J)
004445  P(I+IL,4)=SQRT(PA**2+P(I+IL,5)**2)
004446  280 PH(IL+1,4)=SQRT(PA**2+PH(IL+1,5)**2)
004447  C 10 DECAY PRODUCTS LORENTZ TRANSFORMED TO LAB SYSTEM
004448  DO 290 J=1,4
004449  290 P(I+ND,J)=PH(ND,J)
004450  MLW=1
004451  MLUP=ND-1
004452  DO 320 ML=MLW,MLUP
004453  IL=MLUP-ML+1
004454  DO 300 J=1,3
004455  300 BE(J)=PH(IL,J)/PH(IL,4)
004456  GA=PH(IL,4)/PH(IL,5)
004457  IILW=I+IL
004458  IIUP=I+ND
004459  DO 320 I1=IILW,IIUP
004460  BEP=BE(1)*P(I1,1)+BE(2)*P(I1,2)+BE(3)*P(I1,3)
004461  DO 310 J=1,3
004462  310 P(I1,J)=P(I1,J)+GA*(GA/(1+GA)*BEP+P(I1,4))*BE(J)
004463  320 P(I1,4)=GA*(P(I1,4)+BEP)
004464  C 11 MATRIX ELEMENTS FOR SEMILEPTONIC AND OHG AND PHI DECAYS
004465  IF(ND.EQ.3.AND.K(I+2,2).GE.7.AND.K(I+2,2).LE.16) GO TO 852
004466  GO TO 851
004467  852 CONTINUE
004468  WT=FOUR(IPD,I+1)*FOUR(I+2,I+3)
004469  IF(WT.LT.RNDM(0)*P(IPD,5)**4/WTCOR(1)) GOTO 220
004470  GO TO 853
004471  851 IF(ND.EQ.3.AND.(K(IPD,2).EQ.92.OR.K(IPD,2).EQ.93)) GO TO 854
004472  GO TO 853
004473  854 CONTINUE
004474  WT=(P(I+1,5)*P(I+2,5)*P(I+3,5))**2-(P(I+1,5)*FOUR(I+2,I+3))**2
004475  +-(P(I+2,5)*FOUR(I+1,I+3))**2-(P(I+3,5)*FOUR(I+1,I+2))**2
004476  +2.*FOUR(I+1,I+2)*FOUR(I+1,I+3)*FOUR(I+2,I+3)
004477  IF(WT.LT.RNDM(0)*P(IPD,5)**6/WTCOR(2)) GOTO 220
004478  853 CONTINUE
004479  330 CONTINUE
004480  IF(ND.EQ.0) RETURN
004481  DO 331 L=1,ND
004482  331 K(I+L,1)=-IPD
004483  I=I+ND
004484  MNSJO=MAX(0,MNSJO,I)
004485  RETURN
004486  C
004487  C-- ABNORMAL EXIT
004488  500 PRINT 501,MAXSJO,NEVENT
004489
004490
004491
004492
004493
004494
004495
004496
004497
004498
004499
004500
004501
004502
004503
004504
004505
004506
004507
004508
004509
004510
004511
004512
004513
004514
004515
004516
004517
004518
004519
004520
004521
004522
004523
004524
004525
004526
004527
004528
004529
004530
004531
004532
004533
004534
004535
004536
004537
004538
004539
004540
004541
004542
004543
004544
004545
004546
004547
004548
004549
004550
004551
004552
004553
004554
004555
004556
004557
004558
004559
004560
004561
004562
004563
004564
004565
004566
004567
004568
004569
004570
004571
004572
004573
004574
004575
004576

```

```

501  FORMAT(5(//),IX,4SHNO. OF PARTICLES IN SJOSTRAND P( , ) EXCEEDS ,
      1110//IX,31HOVERFLOW OCCURRED IN SUB. DECAY
      2 //IX,11HEVENT NO. = ,I10
      3//IX,5SHINCREASE 1ST DIMENSIONS OF K( , ) AND P( , ) AND MAXSJO,
      4//IX,52HINCREASE DIMENSIONS OF IHEV( ) (/CANC/) AND TH2V( ) ,
      513H(SUB. PACORR)
      CALL OVEROM
      RETURN
      END
C
C  SUBROUTINE LIST(N)
C  *****
COMMON /JET/ K(399,2), P(399,5)
COMMON /DECPAR/ PUDD, PSID, CND1, CND2, IDB(110)
COMMON /DATA3/ CHA1(36), CHA2(110), CHA3(2)
PRINT 110
DO 100 I=1,N
  IF(K(I,1).GT.0) C1=CHA1(K(I,1))
  IF(K(I,1).LE.0) IC1=-K(I,1)
  C2=CHA2(K(I,2))
  C3=CHA3(1)
  IF(IDB(K(I,2)).EQ.0) C3=CHA3(2)
  IF(K(I,1).GT.0) PRINT 120, I, C1, C2, C3, (P(I,J), J=1,5)
100 IF(K(I,1).LE.0) PRINT 130, -I, IC1, C2, C3, (P(I,J), J=1,5)
RETURN
110 FORMAT(//T11,IHI,T17,3HORI,T24,4HPART,T32,4HSTAB,
      +T44,2HPX,T56,2HPY,T68,2HPZ,T80,1HE,T92,1HM/)
120 FORMAT(IX,I4,4X,A2,1X,2(4X,A4),5(3X,E9.3))
130 FORMAT(IX,I4,4X,1X,I2,2(4X,A4),5(3X,E9.3))
END
C
C  SUBROUTINE EDIT(N)
C  *****
C-----
C-- MODIFIED -- DECAY PRODUCTS OF HEAVY FLAVOR PARTICLES ARE ALWAYS
C-- KEPT IF STABLE
C-----
COMMON /JET/ K(399,2), P(399,5)
COMMON /DECPAR/ PUDD, PSID, CND1, CND2, IDB(110)
COMMON /EDPAR/ ITHROW, PZMIN, PHIN, THETA, PHI, BETA(3)
COMMON /SHEAVY/ IHEV(399)
REAL ROT(3,3), PR(3)
C 1 THROW AWAY NEUTRALS OR UNSTABLE OR WITH TOO LOW PZ OR P
  IF(ITHROW.EQ.0) GOTO 120
  I1=0
  DO 110 I=1,N
    IF(ITHROW.GT.0.OR.(ITHROW.LT.0.AND.K(I,1).GT.0)) PZ=P(I,3)
    IF(IABS(ITHROW).GE.1.AND.IDB(K(I,2)).GT.0) GOTO 110
    IF(IHEV(I).EQ.1) GO TO 111
    IF(IABS(ITHROW).GE.2.AND.(K(I,2).EQ.9.OR.K(I,2).EQ.10.OR.K(I,2).
      +EQ.13.OR.K(I,2).EQ.14.OR.K(I,2).EQ.17.OR.K(I,2).EQ.18)) GOTO 110
    IF(IABS(ITHROW).GE.3.AND.(K(I,2).EQ.1.OR.K(I,2).EQ.58)) GOTO 110
    IF(PZ.LT.PZMIN.OR.(P(I,4)**2-P(I,5)**2.LT.PHIN**2) GOTO 110
111  CONTINUE
    I1=I1+1
    K(I,1)=IDIM(K(I,1),0)
    IF(ITHROW.GT.0) K(I,1)=0
    K(I,2)=K(I,2)
    DO 100 J=1,5
100  P(I,J)=P(I,J)
110  CONTINUE
    N=I1
C 2 ROTATE TO GIVE JET PRODUCED IN DIRECTION THETA, PHI
120 IF(THETA.LT.1E-4) GOTO 150
    ROT(1,1)=COS(THETA)*COS(PHI)
    ROT(1,2)=-SIN(PHI)
    ROT(1,3)=SIN(THETA)*COS(PHI)
    ROT(2,1)=COS(THETA)*SIN(PHI)
    ROT(2,2)=COS(PHI)
    ROT(2,3)=SIN(THETA)*SIN(PHI)
    ROT(3,1)=-SIN(THETA)
    ROT(3,2)=0
    ROT(3,3)=COS(THETA)
    DO 140 I=1,N
    DO 130 J=1,3
130  PR(J)=P(I,J)
    DO 140 J=1,3
140  P(I,J)=ROT(J,1)*PR(1)+ROT(J,2)*PR(2)+ROT(J,3)*PR(3)
C 3 OVERALL LORENTZ BOOST GIVEN BY BETA VECTOR
150 IF(BETA(1)**2+BETA(2)**2+BETA(3)**2.LT.1E-8) RETURN
    GA=1./SQRT(1.-BETA(1)**2-BETA(2)**2-BETA(3)**2)
    DO 170 I=1,N
    BEP=BETA(1)*P(I,1)+BETA(2)*P(I,2)+BETA(3)*P(I,3)
    DO 160 J=1,5
160  P(I,J)=P(I,J)+GA*(GA*(1.+GA)*BEP+P(I,4))*BETA(J)
170  P(I,4)=GA*(P(I,4)+BEP)
    RETURN
    END

```

```

COMMON/NEVENT/NEVENT,ECH,PARHAD(401,7),NPART,MXPART,WEIGHT
COMMON/PARQUA/PARQUA(79,7),NQUA,MAXQUA
COMMON/PAR/ PUD, PSI, SIGMA, CX2, EBEG, WFIN, IFLBEG
COMMON/DECPAR/ PUDD, PSID, CND1, CND2, IDB(110)
COMMON/JET/ K(399,2), P(399,5)
COMMON/EOPAR/ ITHROW, PZMIN, PHIN, THETA, PHI, BETA(3)
COMMON/MAXN/MNBOOK,MNBOOK,MNPART,MNGUA,MNCUE,MNCUEJ,MNSJO

C
EPS=1.E-7
UEPS=1.*EPS

C
IFLBEG=PARQUA(IQ,6)
EBEG=PARQUA(IQ,4)*1000.
PP=SQR(T(PARQUA(IQ,1)**2+PARQUA(IQ,2)**2+PARQUA(IQ,3)**2))
THETA=ACOS(PARQUA(IQ,3)/PP/UEPS)
PTQSQ=PARQUA(IQ,1)**2+PARQUA(IQ,2)**2
IF(PTQSQ.LT.EPS) GO TO 20
PTQ=SQR(PTQSQ)
PHI=ATAN2(PARQUA(IQ,2),PARQUA(IQ,1))
GO TO 21
20 PHI=0.
PTQ=0.
21 CONTINUE

C
CALL JETGEN(N)
IF(N.EQ.0) RETURN
ITHROW=1
CALL EDIT(N)

C
DO 25 I=1,N
DO 25 J=1,5
P(I,J)=P(I,J)/1000.
25 CONTINUE

C
RETURN
END

C
SUBROUTINE SPECTJ
*****
C-- COMPLETES CURRENT CONTENT OF PARHAD( ,7) WITH SPECTATOR JETS
C-- CONTRIBUTION SO AS TO HAVE 4-MOMENTUM CONSERVATION
CALL HAORON(1,0,1)
RETURN
END

C
SUBROUTINE LOOSEM(IH,N)
*****
C-- CARRIES ON DECAY OF HEAVY FLAVOR HAORON STORED IN PARHAD(IH,7)
C-- ALL CASCADE HAORONS (STABLE OR UNSTABLE) ARE STORED IN P( , ) AND
C-- K( , )
C-- IABS(K(I,1)) GIVES ORDER NO. IN THE CASCADE (OF LENGTH N) OF THE
C-- PARENT OF THE HAORON WITH ORDER NO. I
C-- IF K(I,1).LT.0 PARTICLE WITH ORDER NO. I IS UNSTABLE
COMMON/NEVENT/NEVENT,ECH,PARHAD(401,7),NPART,MXPART,WEIGHT
COMMON/PAR/ PUD, PSI, SIGMA, CX2, EBEG, WFIN, IFLBEG
COMMON/DECPAR/ PUDD, PSID, CND1, CND2, IDB(110)
COMMON/JET/ K(399,2), P(399,5)
COMMON/MAXN/MNBOOK,MNBOOK,MNPART,MNGUA,MNCUE,MNCUEJ,MNSJO
COMMON/CBROR/CBROR(180),IDBOR(110)
INTEGER IDBC(110)

C
DO 100 L=1,110
IDBC(L)=IDB(L)
100 IDB(L)=IDBOR(L)

C
I=1
DO 202 J=1,5
P(I,J)=PARHAD(IH,J)*1000.
K(I,2)=PARHAD(IH,6)
K(I,1)=1
202 CONTINUE

C
IPO=0
203 IPO=IPO+1
IF(IDB(K(IPO,2)).GT.0) CALL DECAY(IPO,I)
IF(IPO.LT.I) GO TO 203
N=I
DO 204 M=1,N
IF(IDB(K(M,2)).EQ.0) K(M,1)=-K(M,1)
DO 204 J=1,5
P(M,J)=P(M,J)/1000.
204 CONTINUE

C
IF(N.GT.1) K(I,1)=0

C
DO 300 L=1,110
IDB(L)=IDBC(L)
RETURN
END

C
SUBROUTINE STABKP(N)

```

```

004753 C *****
004754 C-- ELIMINATES UNSTABLE PARTICLES FROM P( , ) AND K( , )
004755 COMMON /JET/ K(399,2), P(399,5)
004756 C
004757 IF(N.EQ.0) RETURN
004758 I1=0
004759 DO 110 I=1,N
004760 IF(K(I,1).LE.0) GO TO 110
004761 I1=I+1
004762 K(I,1)=0
004763 K(I1,2)=K(I,2)
004764 DO 100 J=1,5
004765 P(I1,J)=P(I,J)
004766 110 CONTINUE
004767 N=I1
004768 C
004769 RETURN
004770 END
004771 C
004772 SUBROUTINE SHOCAS(N)
004773 C *****
004774 C-- LISTS CASCADE PARTICLES
004775 COMMON /JET/ K(399,2), P(399,5)
004776 COMMON /DATA3/ CHA1(36), CHA2(110), CHA3(2)
004777 COMMON/CBROR/CBROR(180),IDBOR(110)
004778 PRINT 110
004779 IF(N.EQ.0) RETURN
004780 DO 100 I=1,N
004781 C2=CHA2(K(I,2))
004782 C3=CHA3(I)
004783 IF(IDBOR(K(I,2)).EQ.0) C3=CHA3(2)
004784 100 PRINT 130,I,K(I,1),C2,C3,(P(I,J),J=1,5)
004785 RETURN
004786 110 FORMAT(/T11,IH1,T17,SHORI,T24,4HPART,T32,4HSTAB,
004787 *T44,2HPX,T56,2HPY,T68,2HPZ,T80,1HE,T92,1HM/)
004788 130 FORMAT(8X,I4,2X,I5,2(4X,A4),5(3X,E9.3))
004789 END
004790 C
004791 SUBROUTINE FORSLE(HFLA,KEY)
004792 C *****
004793 C-- FORCES ELECTRON S.L. DECAY OF HEAVY FLAVORS
004794 C-- (IF KEY.GT.0 -- IF KEY.LT.0 RESTORES ORIGINAL O.R.'S)
004795 COMMON /DATA2/ CBR(180), KDP(180,3), WTCOR(10)
004796 COMMON/CBROR/CBROR(180),IDBOR(110)
004797 COMMON/SPYFOR/SPYC,SPYB,SPYT,SPYQ,SPYCE,SPYBEL,SPYTEL
004798 DATA HC,H8,HT/1HC,1HB,1HT/
004799 C
004800 CALL SBLOCK
004801 C
004802 IF(HFLA.EQ.HC) GO TO 40
004803 IF(HFLA.EQ.HB) GO TO 50
004804 IF(HFLA.EQ.HT) GO TO 60
004805 GO TO 500
004806 C
004807 C-- CHARM
004808 40 CONTINUE
004809 IF(KEY.LT.0) GO TO 45
004810 SPYCEL=1.
004811 CBR(6)=1.
004812 CBR(9)=1.
004813 CBR(12)=1.
004814 CBR(121)=1.
004815 RETURN
004816 45 CONTINUE
004817 SPYCEL=-1.
004818 CBR(6)=CBROR(6)
004819 CBR(9)=CBROR(9)
004820 CBR(12)=CBROR(12)
004821 CBR(121)=CBROR(121)
004822 RETURN
004823 C
004824 C-- BOTTOM
004825 50 CONTINUE
004826 IF(KEY.LT.0) GO TO 55
004827 SPYBEL=1.
004828 CBR(16)=1.
004829 CBR(21)=1.
004830 CBR(26)=1.
004831 CBR(32)=0.
004832 CBR(33)=1.
004833 CBR(126)=1.
004834 RETURN
004835 55 CONTINUE
004836 SPYBEL=-1.
004837 CBR(16)=CBROR(16)
004838 CBR(21)=CBROR(21)
004839 CBR(26)=CBROR(26)
004840 CBR(32)=CBROR(32)

```

```

004841
004842
004843
004844
004845
004846
004847
004848
004849
004850
004851
004852
004853
004854
004855
004856
004857
004858
004859
004860
004861
004862
004863
004864
004865
004866
004867
004868
004869
004870
004871
004872
004873
004874
004875
004876
004877
004878
004879
004880
004881
004882
004883
004884
004885
004886
004887
004888
004889
004890
004891
004892
004893
004894
004895
004896
004897
004898
004899
004900
004901
004902
004903
004904
004905
004906
004907
004908
004909
004910
004911
004912
004913
004914
004915
004916
004917
004918
004919
004920
004921
004922
004923
004924
004925
004926
004927
004928

```

```

      CBR(33)=CBROR(33)
      CBR(126)=CBROR(126)
      RETURN
C
C-- TOP
60  CONTINUE
      IF(KEY.LT.0) GO TO 65
      SPYTEL=1.
      CBR(39)=1.
      CBR(45)=1.
      CBR(50)=1.
      CBR(55)=1.
      CBR(61)=1.
      CBR(130)=1.
      RETURN
65  CONTINUE
      SPYTEL=-1.
      CBR(39)=CBROR(39)
      CBR(45)=CBROR(45)
      CBR(50)=CBROR(50)
      CBR(55)=CBROR(55)
      CBR(61)=CBROR(61)
      CBR(130)=CBROR(130)
      RETURN
C
C-- ABNORMAL EXIT
500 PRINT 501,HFLA
501 FORMAT(1H1,24HINCORRECT CALL TO FORSL
1 //1X,27HCALLED WITH 1ST ARGUMENT = ,A1
2 //1X,39HONLY C, B AND T (IN HOLLERITH) ACCEPTED
3 //1X,20HEXECUTION TERMINATED)
      STOP
      END
C
C-----
C
C-- SECTION (F)
C-- TIME LEFT FOR EXECUTION AND RANDOM NUMBER GENERATOR ROUTINES
C-- TO BE FIXED BY USER DEPENDING ON LOCAL IMPLEMENTATIONS
C-----
C
      SUBROUTINE TIMELF(TLE)
      *****
C-- GIVES TIME LEFT FOR EXECUTION
      CALL TIMEL(TLE)
      RETURN
      END
C
      FUNCTION RNDM(A)
      *****
C-- UNIFORM RANDOM NUMBER GENERATOR
C-- N.B. RNDM SHOULD NEVER BE EQUAL TO 0. OR 1.
      RNDM=РАНF()
      RETURN
      END
C
C-----
C-- SECTION (G)
C-- THE CONTENTS OF THE FOLLOWING ROUTINES CAN BE ARBITRARILY CHANGED
C-- BY USER
C-----
C*****
C
      SUBROUTINE BOOKH
      *****
C-- BOOKS HISTOGRAMS
      COMMON//B(2000)
      CALL HLMIT(2000)
C
C-- QUANTA
      CALL HBOOK1(1001,10HN QUANTA $,100,0.,100.)
      CALL HBOOK1(1002,10HOSIG/DETJ$,100,0.,100.)
C
C-- BALANCE
      CALL HBOOK1(2001,10HDE/ECH $,100,-1.,1.)
      CALL HBOOK1(2002,10HDP/P $,100,0.,1.)
C
C-- HADRONS
      CALL HBOOK1(3001,10META CH $,100,-5.,5.)
      CALL HBOOK1(3002,10HNCH $,100,0.,200.)
      CALL HBOOK1(3003,10HPT CH $,100,0.,10.)
      CALL HBOOK1(3004,10HET Y.LT.3$,100,0.,100.)
      CALL HBOOK1(3005,10HET Y.LT.14$,100,0.,100.)
CC
C-- PTHARD
004929 CALL HBOOK1(4001,10MPHARD $,100,0.,200.)
004930 C
004931 CALL HBSTAT(0)
004932 C
004933 RETURN
004934 END
004935 C
004936 SUBROUTINE FILLH
004937 *****
004938 C-- FILLS HISTOGRAMS
004939 C-- NOTE ON /HADQN/ - LHADR(M),LCHARG(M),LCHARM(M),LBOTT(M) AND LTOP(M)
004940 C-- GIVE HADRON/LEPTON QUALITY (1/0 RESPECTIVELY), ELECTRIC CHARGE,
004941 C-- CHARM Q.N., BOTTOM Q.N. AND TOP Q.N. (1,-1 OR 0), RESPECTIVELY,
004942 C-- OF PARTICLE WITH CODE NUMBER M (SEE PARTICLE TABLE)
004943 REAL PTOT(4)
004944 COMMON/PARQUA/PARQUA(79,7),NQUA,MAXQUA
004945 COMMON/EVENT/NEVENT,ECH,PARHAD(401,7),NPART,HXPART,WEIGHT
004946 COMMON/PTHARD/PTHARD
004947 COMMON/HADQN/LHADN(110),LCHARG(110),LCHARM(110),LBOTT(110)
004948 *,LTOP(110)
004949 COMMON/TOTWGT/TOTWGT
004950 COMMON/ACHTOT/ACHTOT
004951 C
004952 DATA ICALL//
004953 C
004954 IF(ICALL.GT.0) GO TO 10
004955 ICALL=1
004956 EPS=1.E-7
004957 FECH=2./ECH
004958 PI=4.*ATAN(1.)
004959 TOTWGT=0.
004960 ACHTOT=0.
004961 10 CONTINUE
004962 C
004963 C-- QUANTA
004964 IF(NQUA.EQ.0) GO TO 16
004965 CALL HFILL(1001,FLOAT(NQUA)+EPS,0.,WEIGHT)
004966 DO 13 I=1,NQUA
004967 ETSQ=PARQUA(I,1)**2+PARQUA(I,2)**2
004968 ETJ=SQRT(ETSQ)
004969 CALL HFILL(1002,ETJ,0.,WEIGHT)
004970 13 CONTINUE
004971 16 CONTINUE
004972 C
004973 IF(NPART.EQ.0) RETURN
004974 C-- BALANCE OF HADRONIC STATE
004975 DO 21 J=1,4
004976 PTOT(J)=0.
004977 21 PSUM=0.
004978 DO 22 I=1,NPART
004979 PSUM=PSUM+SQRT(PARHAD(I,1)**2+PARHAD(I,2)**2+PARHAD(I,3)**2)
004980 DO 23 J=1,4
004981 PTOT(J)=PTOT(J)+PARHAD(I,J)
004982 23 CONTINUE
004983 22 CONTINUE
004984 EBLNC=(PTOT(4)-ECH)/ECH
004985 PBLNC=SQRT(PTOT(1)**2+PTOT(2)**2+PTOT(3)**2)/PSUM
004986 CALL HFILL(2001,EBLNC,0.,WEIGHT)
004987 CALL HFILL(2002,PBLNC,0.,WEIGHT)
004988 C
004989 C-- PARTICLES
004990 IF(NPART.EQ.0) RETURN
004991 TOTWGT=TOTWGT+WEIGHT
004992 NCH=0
004993 ETY3=0.
004994 ETY1=0.
004995 DO 91 I=1,NPART
004996 PP=SQRT(PARHAD(I,1)**2+PARHAD(I,2)**2+PARHAD(I,3)**2)
004997 PTSQ=PARHAD(I,1)**2+PARHAD(I,2)**2
004998 PT=SQRT(PTSQ)
004999 ETI=SQRT(PTSQ+PARHAD(I,5)**2)
005000 Y=SIGN(-ALOG(TAN(ATAN(PT/ABS(PARHAD(I,3)))/2.)),PARHAD(I,3))
005001 IPART=PARHAD(I,6)
005002 IF(LCHARG(IPART).NE.0)
005003 * CALL HFILL(3003,PT,0.,WEIGHT)
005004 IF(LCHARG(IPART).NE.0) CALL HFILL(3001,Y,0.,WEIGHT)
005005 C-- ELIMINATE ET CONTRIBUTIONS FROM NEUTRINOS
005006 IF(IPART.EQ.9.OR.IPART.EQ.10.OR.IPART.EQ.13.OR.IPART.EQ.14
005007 *.OR.IPART.EQ.17.OR.IPART.EQ.18) ETI=0.
005008 IF(LCHARG(IPART).NE.0) NCH=NCH+1
005009 IF(ABS(Y).LT.3.) ETY3=ETY3+ETI
005010 IF(ABS(Y).LT.1.) ETY1=ETY1+ETI
005011 91 CONTINUE
005012 C
005013 ACHTOT=ACHTOT+FLOAT(NCH)*WEIGHT
005014 CALL HFILL(3002,FLOAT(NCH)+1.E-5,0.,WEIGHT)
005015 CALL HFILL(3004,ETY3,0.,WEIGHT)
005016 CALL HFILL(3005,ETY1,0.,WEIGHT)
005017
005018
005019
005020
005021
005022
005023
005024
005025
005026
005027
005028
005029
005030
005031
005032
005033
005034
005035
005036
005037
005038
005039
005040
005041
005042
005043
005044
005045
005046
005047
005048
005049
005050
005051
005052
005053
005054
005055
005056
005057
005058
005059
005060
005061
005062
005063
005064
005065
005066
005067
005068
005069
005070
005071
005072
005073
005074
005075
005076
005077
005078
005079
005080
005081
005082
005083
005084
005085
005086
005087
005088
005089
005090
005091
005092
005093
005094
005095
005096
005097
005098
005099
005100
005101
005102
005103
005104

```


C		005105
C--	PTHARD	005106
	CALL HPFILL(4001,PTHARD,0.,WEIGHT)	005107
C		005108
C	RETURN	005109
	END	005110
C		005111
	SUBROUTINE OUTH	005112
C	*****	005113
C--	OUTPUTS HISTOGRAMS	005114
	EXTERNAL FMUL	005115
	COMMON/TOTWGT/TOTWGT	005116
	COMMON/ACHTOT/ACHTOT	005117
	COMMON/FFMUL/AVNCH	005118
C		005119
	IF(TOTWGT.LT.1.E-5) GO TO 100	005120
	AVNCH=ACHTOT/TOTWGT	005121
	CALL HNCRMA(3002,AVNCH)	005122
	CALL HFUNC(3002,FMUL)	005123
	CALL HPRFUN(3002,3HYES)	005124
C		005125
	CALL MOPERA(1002,1H+,1002,1002,1./TOTWGT,0.)	005126
	CALL MOPERA(3001,1H+,3001,3001,1./TOTWGT,0.)	005127
	CALL MOPERA(3003,1H+,3003,3003,1./TOTWGT,0.)	005128
	CALL MOPERA(3004,1H+,3004,3004,1./TOTWGT,0.)	005129
	CALL MOPERA(3005,1H+,3005,3005,1./TOTWGT,0.)	005130
C		005131
100	CONTINUE	005132
	CALL HIST00	005133
C		005134
	RETURN	005135
	END	005136
		005137