A MONTE CARLO PROGRAM FOR LEPTOPRODUCTION

LU TP 80-12

29.10.1980

Gunnar Ingelman and Torbjörn Sjöstrand

Department of Theoretical Physics
University of Lund
Lund
Sweden

Abstract: A FORTRAN program to simulate deep inelastic scattering of muons, or electrons, is described. It is based on a model where perturbative QCD is used to calculate the cross sections to order α_S and the Lund jet model is used for the soft hadronization process.

Contents

1. Introduction 1

2. The model for leptoproduction 1

3. The program components 2

3.1. PROGRAM QCDWEIGHTS 2

3.2. LEPTO-routines 3
3.2.1. SUBROUTINE LEPBGN 3
3.2.2. SUBROUTINE LEPPRO(ELEP) 3
3.2.3. SUBROUTINE LEPGEN(ANU,X) 4
3.2.4. SUBROUTINE LEPQCD(X,W,QG,QQB,QGMX,QQBMX) 5
3.2.5. BLOCK DATA 5
3.2.6. Commonblocks 5

4. Structure functions 6

5. Examples of the users program 6

References 8

Appendix 1 : Listing of program QCDWEIGHTS 9

Appendix 2 : Listing of LEPTO routines 15

Appendix 3 : Listing of structure function routines 20

1. Introduction

In this paper we will describe a program that simulates leptoproduction at high energies. Our basic assumption is that there are two space-time scales involved. On the short scale, characterized by the QCD parameter $1/\Lambda \cong 0.4$ fm, the coloured objects behave as essentially free particles and perturbative QCD is applicable. The long scale corresponds to the soft hadronization where the coloured objects are connected by colour force fields.

Except for the "ordinary" single quark (or antiquark) jets (q-events) there are two dynamically different 2 jet event situations in first order QCD:

(a) the quark (antiquark) radiates off a gluon (qg events)

(b) a gluon is split by the virtual photon into a quark-antiquark pair ($q\bar{q}$ events)

The cross sections for these different processes are given by Altarelli, Martinelli [1]. For the soft hadronization process we are using the Lund model for quark and gluon jets [2]. The details of how this is implemented in terms of a Monte Carlo program is described in ref. [3]. This model represents a relativistically invariant and causal way of partitioning the energy and momentum of a colour force field among the final state mesons. The dynamics of the force field is that of the massless relativistic string, without any excited transverse degrees of freedom, stretched between a quark and an antiquark. In the case where there is also a gluon present the string is stretched from the quark to the antiquark via a kink which is a representation of the gluon. This means that we do not have three jets fragmenting independently of each other, as in some other models. Hereby we also avoid the problem of joining the jets in the center. In fact, we do not have to select a fixed point in space in any particular Lorentz-frame to connect the jets. This is instead provided in a natural way by the string model we use.

It is important to note that this program conserves energy, momentum (both longitudinal and transverse) and flavour exactly.

2. The model for leptoproduction

In the following we will often refer to the case of an incoming muon (in the Monte Carlo even assumed to be a μ^+) but the formalism is obviously equally applicable to electroproduction.

This program relies heavily on the use of the programs developed earlier for e^+e^--annihilation [3]. When applying these programs for a leptoproduction event we note that, for q and qg events, if the virtual photon hits a quark the target remnant is a colour antitriplet. Apart from the particles in the target fragmentation region we expect the same results as if it were an actual antiquark. Thus we can use the e^+e^- programs by letting the target remnant be represented by an antiquark. For qq events the outgoing quark and antiquark are both connected to the target remnant which is now in a colour octet state, just like a gluon. Again, if we are not studying the target fragmentation region, we can use the same programs with the gluon taking the place of the target remnant.

No electromagnetic radiative corrections are included and the longitudinal structure function is considered to be only a small correction and hence neglected.

The QCD matrix elements for qg and $q\bar{q}$ events give rise to infrared divergencies for soft and collinear gluons. However, the string model we employ provides a natural cut off for these divergencies. In case of a collinear gluon the energy in the field between the gluon and the quark is too small to break and produce quark-antiquark pairs. Hence, the leading meson will contain energy associated both with the quark and the gluon. For a more detailed description of this and the general features of the model for leptoproduction we refer to [4] where also some results, produced with this Monte Carlo program, are presented.

3. The program components

Below we describe the different elements of the program, which is in reality divided into two separate programs. In the first, the QCD matrix elements are used to calculate the relative probabilities for qg and $q\bar{q}$ events. These probabilities,or weights, P_{qg} and $P_{q\bar{q}}$ which are functions of W (the mass of the hadronic system) and the Bjorken x-variable are then written on a data file. The second program, the LEPTO-routines, read this data file and simulate complete events.

The advantage of this procedure lies in saving computing time. For a given set of structure functions, $F_2(x,Q^2)$ and $xg(x,Q^2)$, the QCD matrix elements are integrated once and for all.

3.1. PROGRAM QCDWEIGHTS

First order QCD matrix elements [1]are used to calculate the probabilities for qg and $q\bar{q}$ events in leptoproduction. These probabilities, or weights, are functions of x and W and are therefore calculated on a grid in the x-W plane and stored in the arrays PQG and PQQB respectively. (Actually the factor $\alpha/2\pi$ is not included here but added in the LEPGEN routine so that the QCD parameter Λ can be easily changed, without having to recalculate the weights).

The weights are obtained by normalizing the integrated matrix elements to the total cross section represented by $F_2(x,Q^2)$. For each (x,W) point the maximum value of the matrix elements in the z-c plane is also found and stored in QGMAX and QQBMAX; z and c represent the two degrees of freedom in the energy and angle of the gluon. These maximum values are needed later on in the simulation process.

All these values are written (unformatted) on logical file number 20. A listing of the probabilities (including $\alpha_s/2\pi$ with Λ =0.4) is also produced.

Note that the structure functions $F_2(x,Q^2)$ and $xg(x,Q^2)$ must be supplied by the user as FUNCTION subroutines with the names F2(X,Q2) and XG(X,Q2) respectively. The standard range of x and W values covered is $0.01<x<1.$ and $5<W<45$ GeV. If one later on in the simulation program goes outside this range a warning is given and an ordinary q jet event is generated. This range can of course be changed by the user simply by changing the values given to the arrays XX and WW. If the dimension is also changed, then the same changes must, of course, be made in commonblock QCD in the LEPTO routines. The integration of the matrix elements is made with an absolute accuracy determined by the parameter ERROR (=0.001 by default). For Q^2 < Q2MIN (4. GeV^2 by default) the weights are set to zero. This kind of cut off is needed since most parametrizations of structure functions have a lower bound on Q^2 .

3.2. LEPTO-routines

These consist of a set of subroutines that, based on the already calculated QCD-weights, simulate leptoproduction events.

3.2.1. SUBROUTINE LEPBGN

A call to LEPBGN is made to initialize the routines, i.e. to read the QCD-weights from logical file number 20 into the arrays PQG etc. A few values are also set, e.g. PAR(25)=0.8 gives the width of the Gaussian primordial $k_\perp$ distribution ($\langle k_\perp^2 \rangle$=PAR(25)**2).

3.2.2. SUBROUTINE LEPPRO(ELEP)

This routine generates a complete event in the lab. frame for an incoming muon with energy ELEP GeV. Values of x and Q^2 are chosen according to the cross section formula

$$\frac{d\sigma}{dx\,dQ^2} = \text{const}\ (1-y+y^2/2)\ \frac{F_2(x,Q^2)}{x\ Q^4}$$

where

$$y = \frac{\nu}{E_\mu} = \frac{Q^2}{2ME_\mu x}$$

and assuming the following cuts :

(a)	$Q^2 > CUT(1)$	(4 GeV^2 by default)
(b)	$CUT(2) < x < CUT(3)$	(0 and 1 by default)
(c)	$CUT(4) < \nu < CUT(5)$	(1 and 1000 by default)
(d)	$W^2 > CUT(6)$	(5. GeV^2 by default)
(e)	$\theta_\mu > CUT(7)$	(0. degrees by default)
(f)	$E_\mu' > 1$ GeV	(energy of scattered lepton)

These cuts can easily be changed via the commonblock CUTS.

By a call to LEPGEN the hadron shower is generated and then a rotation of the jet axis is made from the positive z direction to the angles (θ, ϕ) of the virtual photon. Finally, the scattered muon is added as the last (N:th) particle in the arrays K and P in which information on produced particles is stored. Note that all particles from the fragmentation process are kept. To get rid of e.g. unstable and/or neutral particles one can use CALL EDIT(...), see page 40 in reference [3].

The coordinate system employed is right-handed with the incoming muon along the positive z direction. The azimuthal angle of the virtual photon is chosen at random. The azimuthal angle of the hadron event plane around the direction of the virtual photon is also randomly chosen. (Note that this is not entirely correct [5], the modifications expected should however be minor for most practical purposes.)

3.2.3. SUBROUTINE LEPGEN(ANU,X)

Given values of the arguments ν and x the hadronic jet(s) is generated. The type of event q, qg or $q\bar{q}$ is determined from the QCD-weights. For events with a gluon present the energy and angle of it is determined by the QCD matrix elements (at this point the values QGMAX and QQBMAX are needed).

For q and qg events the flavour of the struck quark is determined from the different quark structure functions. This means that not only $F2(x,Q^2)$ and $xg(x,Q^2)$ have to be supplied by the user, but also $xu(x,Q^2)$, $xd(x,Q^2)$, $x\bar{u}(x,Q^2)$, $xs(x,Q^2)$ and $xc(x,Q^2)$ have to be given as FUNCTION subroutines. Note that xu, xd are the valence distributions and $x\bar{u}$, xs, xc the sea distributions for each of the quarks $u,\bar{u},d,\bar{d},s,\bar{s},c,\bar{c}$ separately and not their sum (u-d symmetric sea assumed).

The event is generated in the quark-antiquark (or quark-antiquark-gluon) center of mass frame. A primordial $k_\perp$ is chosen according to a Gaussian distribution ($\langle k_\perp^2\rangle$=PAR(25)**2) and given to the event by swinging its axis (in the cm frame) correspondingly. Finally a boost along the z axis is made to transform to the lab frame (the rotation to the angle of the virtual photon is made after the return to LEPPRO).

3.2.4. SUBROUTINE LEPQCD(X,W,QG,QQB,QGMX,QQBMX)

To get the weights for qg and $q\bar{q}$ events for a given value of (x,W) this routine makes a linear interpolation of the values on the x-W grid.

3.2.5. BLOCK DATA

Sets default values for the cuts.

3.2.6. Commonblocks

COMMON /JET/ N,K(250,2),P(250,5),NC,KC(10),PC(10,4)

Stores the relevant data about a generated event.

(a) N is the number of particles generated and stored in the N first rows of the K and P arrays.
(b) K(I,1) describes the origin of the I:th particle. K(I,2) gives the particle type according to table 1 in [3].
(c) P(I,1), P(I,2), P(I,3) give the particle momentum p_x ,p_y ,p_z in GeV/c. P(I,4) and P(I,5) give the particle energy and mass respectively.
(d) NC is the number of primary partons (quarks and gluons) generated.
(e) KC(IC) gives the flavour of the IC:th parton.
(f) PC(IC,1),P(IC,2),P(IC,3) give the parton momentum and P(IC,4) the parton energy.

COMMON /DATA1/ IST,IFR,PUD,PS1,SIGMA,QMAS(7),PMAS(96),PAR(25)

Stores the most frequently used parameters and data.

(a) IFR (by default 2) is used to choose fragmentation scheme, for 1 it gives an "extended Field-Feynman" and for 2 it gives the Lund model.
(b) PUD (by default 0.4) is the fraction of $u\bar{u}$ pairs created in the field during the jet fragmentation and is assumed to be the same for $d\bar{d}$ pairs, with the rest being $s\bar{s}$ pairs.
(c) PS1 (by default 0.5) is the fraction of primary mesons with spin 1.
(d) SIGMA (by default 0.44) is the standard deviation of the transverse momentum (p_x and p_y components) of primary mesons formed in the field.
(e) PAR(6) , PAR(7) (by default 4., 0.4 GeV) give the number of flavours and the QCD parameter Λ to be used in the formula for α_s .
(f) PAR(24) is a flag for the type of the event generated, 1. for q jet, 2. for qg jet and 3. for $q\bar{q}$ jet.
(g) PAR(25) (by default 0.8) gives the width of the Gaussian primordial $k_\perp$ distribution

($\langle k_\perp^2\rangle$=PAR(25)**2) .

For a more detailed description of these two commonblocks we refer to ref. [3].

```
COMMON /CUTS/ F2MAX,CUT(7),X,Y,ANU,W2,Q2
```

Its purpose is to make it easy to apply cuts on the events and also make some basic kinematical variables available. For details of the cuts see section 3.2.2 . X,Y are the usual scaling variables, ANU = ν , W2 = W^2 and Q2 = Q^2 .

4. Structure functions

A set of structure functions has to be supplied to the programs. The user can have his own favourite set or use one of the two supplied standard sets. These are the Q^2-dependent parametrizations made by Buras and Gaemers [6] and Glück, Hoffmann and Reya [7] .

The structure functions should be in the form of FUNCTION subroutines with the following names :

(a) F2(X,Q2)
(b) XU(X,Q2) = xu valence distribution
(c) XD(X,Q2) = xd valence distribution
(d) XUB(X,Q2) = xu = $x\bar{u}$ = xd = $x\bar{d}$, i.e. u,d symmetric sea
(e) XS(X,Q2) = xs = $x\bar{s}$ strange sea distribution
(f) XC(X,Q2) = xc = $x\bar{c}$ charmed sea distribution
(g) XG(X,Q2) = xg gluon distribution

5. Examples of the users program

In this first example complete events are generated by the LEPTO routines :

```
      COMMON /JET/ N,K(250,2),P(250,5),NC,KC(10),PC(10,4)
      COMMON /CUTS/ F2MAX,CUT(7),X,Y,ANU,W2,Q2
      .
      .
      CALL LEPBGN
      DO 100 I=1,NEVENT
      CALL LEPPRO(280.)
      CALL EDIT(3,0,0.,0.) ; keep only charged,
      .                    ; stable particles
      .
      .                    ; analysis of event
      .
  100 CONTINUE
      STOP
      END
```

In this second example the kinematical variables are supplied by the user, e.g. from data, and hence LEPPRO is by-passed :

```
      COMMON /JET/ N,K(250,2),P(250,5),NC,KC(10),PC(10,4)
      .
      .
```

```
      CALL LEPBGN
      DO 100 I=1,NEVENT
      .
      ANU = ...
      X = ...
      .
      CALL LEPGEN(ANU,X)
      CALL EDIT(3,0,0.,0.)        ; keep only charged,
      .                           ; stable particles
      .
      THETA = ...                 ; angles of virtual photon
      PHI = ...
      CALL ROTBST(THETA,PHI,0.,0.,0.)    ; rotate hadron
      .                           ; shower to photon axis
      .                           ; analysis of event
      .
 100  CONTINUE
      STOP
      END
```

The program should be loaded together with the LEPTO routines, the JET routines of ref. [3] and the desired set of structure functions and the corresponding data file with QCD-weights specified as logical unit 20. As an example of needed computing time we mention that on an IBM 370/168 it takes about 30 CPU-seconds to generate 1000 events.

References

1. G. Altarelli, G. Martinelli
Phys. Lett. 76B (1978) 89

2. B. Andersson, G. Gustafson, C. Peterson
Z. Physik C1 (1979) 105
B. Andersson, G. Gustafson
Z. Physic C3 (1980) 22
B. Andersson, G. Gustafson, T. Sjöstrand
LU TP 80-1 (to be published in Z. Physik C)

3. T. Sjöstrand : A Monte Carlo program for quark and
gluon jet generation, LU TP 80-3

4. B. Andersson, G. Gustafson, G. Ingelman, T. Sjöstrand :
On high energy leptoproduction, LU TP 80-6

5. H. Georgi, H.D. Politzer Phys.Rev.Lett. 40 (1978) 3

6. A.J. Buras, K.J.F. Gaemers
Nucl. Phys. B132(1978) 249

7. M. Glück, E. Hoffmann, E. Reya :
Scaling violations and the gluon distribution of
the nucleon, Dortmund preprint DO-TH 80/13

Appendix 1 : Listing of program QCDWEIGHTS

```
C ********************************************************************
C
C   PROGRAM QCD-WEIGHTS
C
C   USES FIRST ORDER QCD MATRIX ELEMENTS TO CALCULATE THE PROBABILITY
C   FOR QUARK-GLUON AND QUARK-ANTIQUARK EVENTS IN LEPTOPRODUCTION.
C
C   REQUESTED FUNCTION'S :  F2(X,Q2) AND XG(X,Q2)
C
C ********************************************************************

      EXTERNAL QQBAR,QGLUON,FL,FU
      DIMENSION XX(20),WW(17),PQG(20,17),PQQB(20,17),
     & QGMAX(20,17),QQBMAX(20,17)
      COMMON/VAR/X,ANU,Q2,W2,ALS2P,TOTAL,QGMX,QQBMX,QMAS(7),CM(2),PAR9
C...GIVE X AND W VALUES TO ARRAYS XX AND WW WITH DIMENSION NXX
C...AND NWW RESPECTIVELY.
      DATA XX/.01,.025,.05,.075,.1,.125,.15,.175,.2,.25,.3,.35,.4,.45,
     &   .5,.6,.7,.8,.9,.99/
      DATA WW/5.,7.5,10.,12.5,15.,17.5,20.,22.5,25.,27.5,30.,32.5,
     & 35.,37.5,40.,42.5,45./
      DATA NXX,NWW/20,17/
C...SET REQUIRED ABSOLUTE ACCURACY OF INTEGRATION, MINIMUM Q**2
C...AND QCD-LAMBDA.
      DATA ERROR,Q2MIN,QLAM/0.001,4.,0.4/
C...CALCULATE VALUES FOR CUTS ON SOFT AND COLLINEAR GLUONS
      CM(1)=QMAS(1)**2+10.*QMAS(1)*PAR9+12.5*PAR9**2
      CM(2)=8.*PAR9**2
      F2MAX=0.
      WRITE(6,200)
      DO 100 IW=1,NWW
      W=WW(IW)
      W2=W**2
      DO 100 IX=1,NXX
      X=XX(IX)
      ANU=(W2-.88)/(2.*.938*(1.-X))
      Q2=2.*0.938*ANU*X
      Q=SQRT(Q2)
C...CALCULATE ALPHA-STRONG/2/PI
      ALS2P=3./(25.*ALOG(Q/QLAM))
      ALFA=ALS2P*2.*3.14159
      QQB=0.
      QQBMX=0.
      QG=0.
      QGMX=0.
      IF(Q2.LT.Q2MIN) GOTO 50
      F2XQ2=F2(X,Q2)
      TOTAL=F2XQ2/X
C...FIND MAXIMUM OF F2
      IF(F2XQ2.GT.F2MAX) F2MAX=F2XQ2
C...INTEGRATE MATRIX ELEMENTS
      EPS=ERROR
      CALL GADAP2(X,1.,FL,FU,QQBAR,EPS,QQB)
      EPS=ERROR
```



```
      CALL GADAP2(X,1.,FL,FU,QGLUON,EPS,QG)
   50 PQQB(IX,IW)=QQB/ALS2P
      QQBMAX(IX,IW)=QQBMX
      PQG(IX,IW)=QG/ALS2P
      QGMAX(IX,IW)=QGMX
      PQ=1.-QG-QQB
      WRITE(6,300) W,X,Q2,ANU,ALFA,PQ,QG,QQB,QGMX,QQBMX
  100 CONTINUE
      WRITE(6,400) NXX,NWW,ERROR,Q2MIN,QLAM,F2MAX
C...WRITE RESULTS (UNFORMATTED) ON LOGICAL FILE NUMBER 20
      WRITE(20) NXX,NWW,XX,WW,PQG,PQQB,QGMAX,QQBMAX,F2MAX
  200 FORMAT(1H1,////,10X,'W',7X,'X',9X,'Q**2',6X,'NY',7X,'ALFA',6X,
     & 'Q-JET',4X,'QG-JET',3X,'QQB-JET',3X,'QG-MAX',7X,'QQB-MAX',//)
  300 FORMAT(2X,F10.1,F10.3,2F10.1,4F10.4,2E12.4)
  400 FORMAT(///,T7,'NXX   NWW',T23,'ERROR    Q2MIN     QLAM    F2MAX',
     & //,T5,2I5,T20,F8.4,3F8.2,//)
      STOP
      END


      REAL FUNCTION FL(Y)
C...GIVE LOWER BOUNDARY FOR INTEGRATION IN C=COS(THETA)
      FL=-1.
      RETURN
      END


      REAL FUNCTION FU(Y)
C...GIVE UPPER BOUNDARY FOR INTEGRATION IN C...
      FU=1.
      RETURN
      END


      REAL FUNCTION QQBAR(Z,C)

C ********************************************************************
C
C   CALCULATE VALUE OF QCD MATRIX ELEMENT AS A FUNCTION OF Z AND C
C   FOR QUARK-ANTIQUARK EVENTS, I.E. PHOTON-GLUON FUSION PROCESS.
C   ALSO MAKE CUTS AGAINST COLLINEAR GLUONS.
C
C ********************************************************************

      COMMON/VAR/X,ANU,Q2,W2,ALS2P,TOTAL,QGMX,QQBMX,QMAS(7),CM(2),PAR9
      U=X/Z
      EM3=Q2*(1.-Z)/Z
      EM1=Q2*(Z-X)*(1.-Z)*(1.-C)/(X*Z*(1.+(2.*Z-1.)*C))
      EM2=Q2*(Z-X)*(1.+C)/(X*(1.+(2.*Z-1.)*C))
      IF(EM1.LT.CM(1).OR.EM2.LT.CM(1)) GOTO  100
      FGMASS = U*2.*.938*ANU + (U*.938)**2 - Q2
      SQ = 0.
      DO 20 IFL=1,4
C...CHECK THAT INVARIANT MASS OF PHOTON+GLUON IS BIG ENOUGH TO
C...PRODUCE QUARK-ANTIQUARK PAIR
      IF(FGMASS .LT. (2.*QMAS(IFL))**2)  GOTO 20
      CHARGE = 1.
      IF(IFL .EQ. 1) CHARGE = 4.
      X1 = 1. - (EM1-QMAS(IFL)**2)/W2
      X2 = 1. - (EM2-QMAS(IFL)**2)/W2
```

```
      X3=2.-X1-X2
      IF((1.-X1)*(1.-X2)*(1.-X3)/X3**2.LE.(QMAS(IFL)**2/W2)) GOTO 20
      IF(IFL .NE. 4) GOTO 10
C...MAKE EXTRA CUTS FOR THE HEAVY CHARMED QUARK BY USING THE
C...THRESHOLD (1-WTH**2/W**2) ; WTH**2 = 22. GEV**2
      IF(W2 .LT. 22.) GOTO 20
      XCM=1.-5.*PAR9*(5.*PAR9+4.*QMAS(IFL))/(2.*W2)
      IF(X2 .GT. XCM .OR. X1 .GT. XCM) GOTO 20
      CHARGE = 4.*(1.-22./W2)
   10 SQ = SQ + CHARGE
   20 CONTINUE
      P=SQ/9./Z*XG(U,Q2)/U/(1.+(2.*Z-1.)*C)**2*
     &  (Z**2*(1.+C)/(1.-C)+(1.-Z)**2*(1.-C)/(1.+C)-
     &   Z*(1.-Z)*2.*(1.+(2.*Z-1.)*C)**2/(1.-C**2) +
     &   4.*(Z*(1.-Z))**2)
      QQBAR=P*ALS2P/TOTAL
      IF(P.GT.QQBMX) QQBMX = P
      RETURN
  100 QQBAR=0.
      RETURN
      END


      REAL FUNCTION QGLUON(Z,C)

C ********************************************************************
C
C   CALCULATE VALUE OF QCD MATRIX ELEMENT AS A FUNCTION OF Z AND C
C   FOR QUARK-GLUON EVENTS, I.E. RADIATION OF A GLUON.
C   ALSO MAKE CUTS AGAINST SOFT AND COLLINEAR GLUONS.
C
C ********************************************************************

      COMMON/VAR/X,ANU,Q2,W2,ALS2P,TOTAL,QGMX,QQBMX,QMAS(7),CM(2),PAR9
      U=X/Z
      EM2=Q2*(1.-Z)/Z
      IF(EM2.LT.CM(1)) GOTO 100
      EM1=Q2*(Z-X)*(1.-Z)*(1.-C)/(X*Z*(1.+(2.*Z-1.)*C))
      EM3=Q2*(Z-X)*(1.+C)/(X*(1.+(2.*Z-1.)*C))
      IF(EM1.LT.CM(1).OR.EM2*EM1/EM3.LT.CM(2)) GOTO 100
      P=F2(U,Q2)/U*8./3./(1.+(2.*Z-1.)*C)**2*(Z*(1.+Z**2)/(1.-Z)*
     &(1.+C)/(1.-C)+(1.-Z)**2+(1.-Z)*(1.-C+2.*Z**2*(1.+C))/(1.+(2.*Z
     &-1.)*C))
      QGLUON=P*ALS2P/TOTAL
      IF(P.GT.QGMX) QGMX=P
      RETURN
  100 QGLUON=0.
      RETURN
      END


      BLOCK DATA
      COMMON/VAR/X,ANU,Q2,W2,ALS2P,TOTAL,QGMX,QQBMX,QMAS(7),CM(2),PAR9
C...GIVE QUARK MASSES. PAR9=PAR(9) IN REF. 3.
      DATA QMAS/.3,.3,.5,1.6,5.,20.,.275/,PAR9/.8/
      END
```



```
C.......................................................................
C   PURPOSE              - INTEGRATE A FUNCTION F(X,Y) OF TWO VARIABLES
C   METHOD               - ADAPTIVE GAUSSIAN IN BOTH DIRECTIONS
C   USAGE                - CALL GADAP2(A0,B0,FL,FU,F,EPS,SUM)
C   PARAMETERS     A0    - LOWER X-LIMIT (INPUT,REAL)
C                  B0    - UPPER X-LIMIT (INPUT,REAL)
C                  FL    - USER SUPPLIED FUNCTION FL(X) GIVING THE LOWER
C                          Y-LIMIT FOR A GIVEN X-VALUE
C                          (INPUT,REAL FUNCTION)
C                  FU    - USER SUPPLIED FUNCTION FU(X) GIVING THE UPPER
C                          Y-LIMIT FOR A GIVEN X-VALUE
C                          (INPUT,REAL FUNCTION)
C                  F     - USER SUPPLIED FUNCTION F(X,Y) TO BE INTEGRATED
C                          (INPUT,REAL FUNCTION)
C                  EPS   - DESIRED ACCURACY (INPUT,REAL)
C                  SUM   - CALCULATED VALUE FOR THE INTEGRAL (OUTPUT,REAL)
C   PRECISION            - SINGLE
C   REQ'D PROG'S         - FL,FU,F,FGADAP
C   AUTHOR               - THOMAS JOHANSSON, LDC,1973
C.......................................................................
      SUBROUTINE GADAP2(A0,B0,FL,FU,F,EPS,SUM)
      COMMON/GADAP1/ NUM,IFU
      EXTERNAL F,FL,FU
      DIMENSION A(300),B(300),F1(300),F2(300),F3(300),S(300),N(300)
    1 FORMAT(16H GADAP:I TOO BIG)
      DSUM(F1F,F2F,F3F,AA,BB)=5./18.*(BB-AA)*(F1F+1.6*F2F+F3F)
      IF(EPS.LT.1.0E-8) EPS=1.0E-8
      RED=1.4
      L=1
      I=1
      SUM=0.
      C=SQRT(15.)/5.
      A(1)=A0
      B(1)=B0
      X=0.5*(1+C)*A0+0.5*(1-C)*B0
      AY=FL(X)
      BY=FU(X)
      F1(1)=FGADAP(X,AY,BY,F,EPS)
      X=0.5*(A0+B0)
      AY=FL(X)
      BY=FU(X)
      F2(1)=FGADAP(X,AY,BY,F,EPS)
      X=0.5*(1-C)*A0+0.5*(1+C)*B0
      AY=FL(X)
      BY=FU(X)
      F3(1)=FGADAP(X,AY,BY,F,EPS)
      IFU=3
      S(1)=  DSUM(F1(1),F2(1),F3(1),A0,B0)
  100 CONTINUE
      L=L+1
      N(L)=3
      EPS=EPS*RED
      A(I+1)=A(I)+C*(B(I)-A(I))
      B(I+1)=B(I)
      A(I+2)=A(I)+B(I)-A(I+1)
      B(I+2)=A(I+1)
      A(I+3)=A(I)
      B(I+3)=A(I+2)
      W1=A(I)+(B(I)-A(I))/5.
      U2=2.*W1-(A(I)+A(I+2))/2.
```

```
      X=A(I)+B(I)-W1
      AY=FL(X)
      BY=FU(X)
      F1(I+1)=FGADAP(X,AY,BY,F,EPS)
      F2(I+1)=F3(I)
      X=B(I)-A(I+2)+W1
      AY=FL(X)
      BY=FU(X)
      F3(I+1)=FGADAP(X,AY,BY,F,EPS)
      X=U2
      AY=FL(X)
      BY=FU(X)
      F1(I+2)=FGADAP(X,AY,BY,F,EPS)
      F2(I+2)=F2(I)
      X=B(I+2)+A(I+2)-U2
      AY=FL(X)
      BY=FU(X)
      F3(I+2)=FGADAP(X,AY,BY,F,EPS)
      X=A(I)+A(I+2)-W1
      AY=FL(X)
      BY=FU(X)
      F1(I+3)=FGADAP(X,AY,BY,F,EPS)
      F2(I+3)=F1(I)
      X=W1
      AY=FL(X)
      BY=FU(X)
      F3(I+3)=FGADAP(X,AY,BY,F,EPS)
      IFU=IFU+6
      IF(IFU.GT.5000) GOTO 130
      S(I+1)=  DSUM(F1(I+1),F2(I+1),F3(I+1),A(I+1),B(I+1))
      S(I+2)=  DSUM(F1(I+2),F2(I+2),F3(I+2),A(I+2),B(I+2))
      S(I+3)=  DSUM(F1(I+3),F2(I+3),F3(I+3),A(I+3),B(I+3))
      SS=S(I+1)+S(I+2)+S(I+3)
      I=I+3
      IF(I.GT.300)GOTO 120
      SOLD=S(I-3)
      IF(ABS(SOLD-SS).GT.EPS*(1.+ABS(SS))/2.) GOTO 100
      SUM=SUM+SS
      I=I-4
      N(L)=0
      L=L-1
  110 CONTINUE
      IF(L.EQ.1) GOTO 130
      N(L)=N(L)-1
      EPS=EPS/RED
      IF(N(L).NE.0) GOTO 100
      I=I-1
      L=L-1
      GOTO 110
  120 WRITE(6,1)
  130 RETURN
      END

      FUNCTION FGADAP(X,A0,B0,F,EPS)
      COMMON/GADAP1/ NUM,IFU
      EXTERNAL F
      DIMENSION A(300),B(300),F1(300),F2(300),F3(300),S(300),N(300)
    1 FORMAT(16H GADAP:I TOO BIG)
      DSUM(F1F,F2F,F3F,AA,BB)=5./18.*(BB-AA)*(F1F+1.6*F2F+F3F)
      IF(EPS.LT.1.0E-8) EPS=1.0E-8
```



```
      RED=1.4
      L=1
      I=1
      SUM=0.
      C=SQRT(15.)/5.
      A(1)=A0
      B(1)=B0
      F1(1)=F(X,0.5*(1+C)*A0+0.5*(1-C)*B0)
      F2(1)=F(X,0.5*(A0+B0))
      F3(1)=F(X,0.5*(1-C)*A0+0.5*(1+C)*B0)
      IFU=3
      S(1)=  DSUM(F1(1),F2(1),F3(1),A0,B0)
  100 CONTINUE
      L=L+1
      N(L)=3
      EPS=EPS*RED
      A(I+1)=A(I)+C*(B(I)-A(I))
      B(I+1)=B(I)
      A(I+2)=A(I)+B(I)-A(I+1)
      B(I+2)=A(I+1)
      A(I+3)=A(I)
      B(I+3)=A(I+2)
      W1=A(I)+(B(I)-A(I))/5.
      U2=2.*W1-(A(I)+A(I+2))/2.
      F1(I+1)=F(X,A(I)+B(I)-W1)
      F2(I+1)=F3(I)
      F3(I+1)=F(X,B(I)-A(I+2)+W1)
      F1(I+2)=F(X,U2)
      F2(I+2)=F2(I)
      F3(I+2)=F(X,B(I+2)+A(I+2)-U2)
      F1(I+3)=F(X,A(I)+A(I+2)-W1)
      F2(I+3)=F1(I)
      F3(I+3)=F(X,W1)
      IFU=IFU+6
      IF(IFU.GT.5000) GOTO 130
      S(I+1)=  DSUM(F1(I+1),F2(I+1),F3(I+1),A(I+1),B(I+1))
      S(I+2)=  DSUM(F1(I+2),F2(I+2),F3(I+2),A(I+2),B(I+2))
      S(I+3)=  DSUM(F1(I+3),F2(I+3),F3(I+3),A(I+3),B(I+3))
      SS=S(I+1)+S(I+2)+S(I+3)
      I=I+3
      IF(I.GT.300)GOTO 120
      SOLD=S(I-3)
      IF(ABS(SOLD-SS).GT.EPS*(1.+ABS(SS))/2.) GOTO 100
      SUM=SUM+SS
      I=I-4
      N(L)=0
      L=L-1
  110 CONTINUE
      IF(L.EQ.1) GOTO 130
      N(L)=N(L)-1
      EPS=EPS/RED
      IF(N(L).NE.0) GOTO 100
      I=I-1
      L=L-1
      GOTO 110
  120 WRITE(6,1)
  130 FGADAP=SUM
      EPS=EPS/RED
      RETURN
      END
```

Appendix 2 : Listing of LEPTO routines

```
      SUBROUTINE LEPBGN

C ********************************************************************
C
C   INITIALIZE LEPTO-ROUTINES. READ QCD-WEIGHTS FROM LOGICAL FILE 20.
C   PAR(25) GIVES WIDTH OF PRIMORDIAL KT-DISTRIBUTION.
C
C ********************************************************************
      COMMON /CUTS/ F2MAX,CUT(7),X,Y,ANU,W2,Q2
      COMMON /QCD/ NXX,NWW,XX(20),WW(17),PQG(20,17),PQQB(20,17),
     & QGMAX(20,17),QQBMAX(20,17),CM(2)
      COMMON /DATA1/ IST,IFR,PUD,PS1,SIGMA,QMAS(7),PMAS(96),PAR(25)
      READ(20) NXX,NWW,XX,WW,PQG,PQQB,QGMAX,QQBMAX,F2MAX
      PAR(6)=4.
      PAR(7)=.4
      PAR(9)=.8
      PAR(25)=.8
      CM(1)=QMAS(1)**2+10.*QMAS(1)*PAR(9)+12.5*PAR(9)**2
      CM(2)=8.*PAR(9)**2
      RETURN
      END


      SUBROUTINE LEPPRO(ELEP)

C ********************************************************************
C
C   CHOOSE X,Y AND Q2 FROM CROSS SECTION FORMULA. MAKE CUTS ON
C   KINEMATICAL VARIABLES. GENERATE PARTICLES VIA CALL TO LEPGEN.
C
C ********************************************************************

      COMMON /JET/ N,K(250,2),P(250,5),NC,KC(10),PC(10,4)
      COMMON /DATA1/ IST,IFR,PUD,PS1,SIGMA,QMAS(7),PMAS(96),PAR(25)
      COMMON /CUTS/ F2MAX,CUT(7),X,Y,ANU,W2,Q2
  100 Q2=CUT(1)/RANF(0)
      XMIN=CUT(1)/(2.*.938*ELEP)
  110 X=XMIN**RANF(0)
      IF(X.LT.CUT(2).OR.X.GT.CUT(3))  GOTO 110
      X1=1.-X
      IF(F2(X,Q2).LT.F2MAX*RANF(0)) GOTO 100
      Y=Q2/(2.*.938*ELEP*X)
      IF(1.-Y+.5*Y**2.LT.RANF(0)) GOTO 100
      ELEPP = ELEP*(1.-Y)
      IF(ELEPP.LE.1.) GOTO 100
      ANU=ELEP*Y
      IF(ANU.LT.CUT(4).OR.ANU.GT.CUT(5)) GOTO 100
      Q = SQRT(Q2)
      W2 = 2.*0.938*ANU*X1 + .88
      IF(W2.LT.CUT(6)) GOTO 100
      W = SQRT(W2)
      THEL=ACOS(1.-Q2/(2.*ELEP*ELEPP))
      IF(THEL*180./3.1415927.LT.CUT(7)) GOTO 100
      CALL LEPGEN(ANU,X)
```



```
      THE=ACOS((ANU+Q2/2./ELEP)/SQRT(ANU**2+Q2))
      PHI = 6.2832*RANF(0)
      PHIL = PHI + 3.1416
C...ROTATE TO ANGLES OF VIRTUAL PHOTON
      CALL ROTBST(THE,PHI,0.,0.,0.)
C...ADD SCATTERED MUON TO THE EVENT
      N=N+1
      K(N,1)=38
      K(N,2)=12
      P(N,1)=ELEPP*SIN(THEL)*COS(PHIL)
      P(N,2)=ELEPP*SIN(THEL)*SIN(PHIL)
      P(N,3)=ELEPP*COS(THEL)
      P(N,4)=ELEPP
      P(N,5)=0.1057
      RETURN
      END


      SUBROUTINE LEPGEN(ANU,X)

C ********************************************************************
C
C   GENERATE THE HADRONIC PART OF THE EVENT FOR A GIVEN NU AND X.
C   REQUESTED FUNCTION'S : F2(X,Q2), XU(X,Q2), XD(X,Q2), XUB(X,Q2),
C                          XS(X,Q2), XC(X,Q2), XG(X,Q2)
C
C ********************************************************************

      COMMON /JET/ N,K(250,2),P(250,5),NC,KC(10),PC(10,4)
      COMMON /DATA1/ IST,IFR,PUD,PS1,SIGMA,QMAS(7),PMAS(96),PAR(25)
      COMMON /QCD/ NXX,NWW,XX(20),WW(17),PQG(20,17),PQQB(20,17),
     & QGMAX(20,17),QQBMAX(20,17),CM(2)
      Q2=2.*0.938*ANU*X
      Q=SQRT(Q2)
      W2=2.*0.938*ANU*(1.-X)+.88
      W=SQRT(W2)
      QQB=0.
      QG=0.
C...IF X AND W IS WITHIN THE COVERED RANGE GET QCD-PROBABILITIES
      IF(X.GT.XX(1).AND.X.LT.XX(NXX).AND.
     & W.GT.WW(1).AND.W.LT.WW(NWW))  GOTO 10
      IF(W.LT.WW(1)) GOTO 20
      WRITE(6,5) X,W
    5 FORMAT(/,5X,'X  (=',F6.4,') OR W (=',F6.1,') OUT OF RANGE',/)
      GOTO 20
   10 CALL LEPQCD(X,W,QG,QQB,QGMX,QQBMX)
C...INCLUDE ALPHA-STRONG/2/PI
      ALS2P=3./(33.-2.*PAR(6))/ALOG(Q/PAR(7))
      QG=QG*ALS2P
      QQB=QQB*ALS2P
   20 SRANF=RANF(0)
      JSIGN=SIGN(1.,2.*RANF(0)-1.)
      IF(SRANF.LT.QQB) GOTO 300
C...FOR Q AND QG EVENTS CHOOSE FLAVOUR
      XD1=XD(X,Q2)
      XUB10=XUB(X,Q2)*10.
      XS2=XS(X,Q2)*2.
      XC8=XC(X,Q2)*8.
      IFL=1
   30 RF2=RANF(0)*F2(X,Q2)*9.
```

```
      IF(RF2.GT.XC8) GOTO 40
C...APPLY THRESHOLD FACTOR FOR CHARMED QUARKS
      IF(1.-22./W2.LT.RANF(0)) GOTO 30
      IFL=4*JSIGN
      GOTO 70
   40 IF(RF2.GT.XC8+XS2) GOTO 50
      IFL=3*JSIGN
      GOTO 70
   50 IF(RF2.GT.XC8+XS2+XUB10) GOTO 60
      IFL=MAX(1,INT(5.*RANF(0)-2.))*JSIGN
      GOTO 70
   60 IF(RF2.GT.XC8+XS2+XUB10+XD1) GOTO 70
      IFL=2
   70 IFLA=IABS(IFL)
      IF(SRANF .LT. QG+QQB) GOTO 200
  100 JETYPE=1
C...GENERATE QUARK-JET EVENT
      CALL QQJET(IFL,W)
      GOTO 500
  200 JETYPE=2
C...GENERATE QUARK-GLUON JET EVENT, CHOOSE Z AND C ACCORDING TO QCD
C...MATRIX ELEMENTS AND APPLY CUTS FOR SOFT AND COLLINEAR GLUONS
  210 Z=X+(1.-X)*RANF(0)
      EM2=Q2*(1.-Z)/Z
      IF(EM2.LT.CM(1)) GOTO 210
      C=2.*RANF(0)-1.
      EM1=Q2*(Z-X)*(1.-Z)*(1.-C)/(X*Z*(1.+(2.*Z-1.)*C))
      EM3=Q2*(Z-X)*(1.+C)/(X*(1.+(2.*Z-1.)*C))
      IF(EM1.LT.CM(1).OR.EM1*EM2/EM3.LT.CM(2)) GOTO 210
      SIG=F2(X/Z,Q2)*8.*
     &Z/(3.*X)/(1.+(2.*Z-1.)*C)**2*(Z*(1.+Z**2)/(1.-Z)*(1.+C)/(1.-C)+
     &(1.-Z)**2+(1.-Z)*(1.-C+2.*Z**2*(1.+C))/(1.+(2.*Z-1.)*C))
      IF(SIG.LT.QGMX*RANF(0)) GOTO 210
      X1=1.-(EM1-QMAS(IFLA)**2)/W2
      X2=1.-(EM2-QMAS(IFLA)**2)/W2
      X3=2.-X1-X2
      IF((1.-X1)*(1.-X2)*(1.-X3)/X3**2.LE.(QMAS(IFLA)/W)**2) GOTO 100
      XCUT=1.-5.*PAR(9)*(5.*PAR(9)+4.*QMAS(IFLA))/(2.*W2)
      IF(X2 .GT. XCUT .OR. X1 .GT. XCUT) GOTO 100
      CALL QQGJET(IFL,W,X1,X2)
      CALL ROTBST(ACOS(-PC(2,3)/SQRT(PC(2,3)**2+PC(2,1)**2)),
     &6.2832*RANF(0),0.,0.,0.)
      GOTO 500
  300 JETYPE=3
C...GENERATE QUARK-ANTIQUARK JET EVENT, CHOOSE Z AND X ACCORDING TO
C...QCD MATRIX ELEMENTS AND APPLY CUTS ON COLLINEAR GLUONS.
  310 Z=X+(1.-X)*RANF(0)
      EM3=Q2*(1.-Z)/Z
      C=2.*RANF(0)-1.
      EM1=Q2*(Z-X)*(1.-Z)*(1.-C)/(X*Z*(1.+(2.*Z-1.)*C))
      EM2=Q2*(Z-X)*(1.+C)/(X*(1.+(2.*Z-1.)*C))
      IF(EM1.LT.CM(1).OR.EM2.LT.CM(1)) GOTO 310
      U=X/Z
      FGMASS=U*2.*.938*ANU+(U*.938)**2-Q2
      SQ=0.
      DO 350 IFLA=1,4
      IF(FGMASS .LT. (2.*QMAS(IFLA))**2)  GOTO 350
      CHARGE = 1.
      IF(IFLA.EQ.1) CHARGE = 4.
      X1 = 1. - (EM1-QMAS(IFLA)**2)/W2
```



```
      X2 = 1. - (EM2-QMAS(IFLA)**2)/W2
      X3=2.-X1-X2
      IF((1.-X1)*(1.-X2)*(1.-X3)/X3**2.LE.(QMAS(IFLA)**2/W2)) GOTO 350
      IF(IFLA .NE. 4) GOTO 340
C...THRESHOLD FACTOR FOR CHARMED QUARK
      IF(W2 .LT. 22.) GOTO 350
      XCUT=1.-5.*PAR(9)*(5.*PAR(9)+4.*QMAS(IFLA))/(2.*W2)
      IF(X2 .GT. XCUT .OR. X1 .GT. XCUT) GOTO 350
      CHARGE = 4.*(1.-22./W2)
  340 SQ = SQ + CHARGE
  350 CONTINUE
      SIG=SQ/9./Z*XG(U,Q2)/U/(1.+(2.*Z-1.)*C)**2*
     &  (Z**2*(1.+C)/(1.-C)+(1.-Z)**2*(1.-C)/(1.+C)-
     &   Z*(1.-Z)*2.*(1.+(2.*Z-1.)*C)**2/(1.-C**2) +
     &   4.*(Z*(1.-Z))**2)
      IF(SIG.LT.QQBMX*RANF(0)) GOTO 310
      SIFL=SQ*RANF(0)
C...CHOOSE FLAVOUR
      IFLA=1
      IF(SIFL .GT. 4.) IFLA=2
      IF(SIFL .GT. 5.) IFLA=3
      IF(SIFL .GT. 6.) IFLA=4
      IFL=IFLA*JSIGN
      X1=1.-(EM1-QMAS(IFLA)**2)/W2
      X2=1.-(EM2-QMAS(IFLA)**2)/W2
      X3=2.-X1-X2
      CALL QQGJET(IFL,W,X1,X2)
      CALL ROTBST(-ACOS(-PC(3,3)/SQRT(PC(3,3)**2+PC(3,1)**2)),
     &6.2832*RANF(0),0.,0.,0.)
  500 CONTINUE
C...INTRODUCE PRIMORDIAL KT
      CALL PTDIST(PX,PY,PAR(25))
      PT=SQRT(PX**2+PY**2)
      PHI=SIGN(ACOS(PX/PT),PY)
      CALL ROTBST(ATAN(PT/ABS(PC(JETYPE,3))),PHI,0.,0.,0.)
C...MAKE BOOST ALONG Z-AXIS TO LAB.-SYSTEM
      CALL ROTBST(0.,0.,0.,0.,1./SQRT(1.+W2/(ANU**2+Q2)))
      PAR(24)=JETYPE
      RETURN
      END


      SUBROUTINE LEPQCD(X,W,QG,QQB,QGMX,QQBMX)
C ******************************************************************
C
C   TO GET THE QCD-WEIGHTS FOR GIVEN VALUES OF (X,W) MAKE A LINEAR
C   INTERPOLATION OF THE VALUES ON THE X-W GRID.
C
C ******************************************************************
      COMMON /QCD/ NXX,NWW,XX(20),WW(17),PQG(20,17),PQQB(20,17),
     & QGMAX(20,17),QQBMAX(20,17),CM(2)
      IX=0
  100 IX=IX+1
      IF(X.GT.XX(IX+1)) GOTO 100
      IW=0
  200 IW=IW+1
      IF(W.GT.WW(IW+1)) GOTO 200
      WD=(W-WW(IW))/(WW(IW+1)-WW(IW))
      XD=(X-XX(IX))/(XX(IX+1)-XX(IX))
      X1P=(PQG(IX+1,IW)-PQG(IX,IW))*XD+PQG(IX,IW)
```

```
      X2P=(PQG(IX+1,IW+1)-PQG(IX,IW+1))*XD+PQG(IX,IW+1)
      QG=(X2P-X1P)*WD+X1P
      X1P=(PQQB(IX+1,IW)-PQQB(IX,IW))*XD+PQQB(IX,IW)
      X2P=(PQQB(IX+1,IW+1)-PQQB(IX,IW+1))*XD+PQQB(IX,IW+1)
      QQB=(X2P-X1P)*WD+X1P
      X1P=(QGMAX(IX+1,IW)-QGMAX(IX,IW))*XD+QGMAX(IX,IW)
      X2P=(QGMAX(IX+1,IW+1)-QGMAX(IX,IW+1))*XD+QGMAX(IX,IW+1)
      QGMX=(X2P-X1P)*WD+X1P
      X1P=(QQBMAX(IX+1,IW)-QQBMAX(IX,IW))*XD+QQBMAX(IX,IW)
      X2P=(QQBMAX(IX+1,IW+1)-QQBMAX(IX,IW+1))*XD+QQBMAX(IX,IW+1)
      QQBMX=(X2P-X1P)*WD+X1P
      RETURN
      END


      BLOCK DATA
C...GIVE DEFAULT VALUES ON CUTS
      COMMON /CUTS/ F2MAX,CUT(7),X,Y,ANU,W2,Q2
      DATA CUT/4.,0.,1.,1.,1.E+03,5.,0./
      END
```


Appendix 3 : Listing of structure function routines

```
C ******************************************************************
C
C   STRUCTURE FUNCTIONS FROM BURAS AND GAEMERS, SEE REF. 6 .
C    (SOLUTION 1)
C   REQUESTED FUNCTION :   GAMMA(X)
C
C ******************************************************************

      FUNCTION F2(X,Q2)
      F2=5./18.*XV8(X,Q2)+1./6.*XV3(X,Q2)+
     & 12./9.*XS(X,Q2)+8./9.*XC(X,Q2)
      RETURN
      END


      FUNCTION B(X,Y)
      B=GAMMA(X)*GAMMA(Y)/GAMMA(X+Y)
      RETURN
      END


      FUNCTION XV8(X,Q2)
      S=ALOG(ALOG(Q2/.09)/2.9957)
      ETA1=.70-.176*S
      ETA2=2.60+.8*S
      XV8=3.*X**ETA1*(1.-X)**ETA2/B(ETA1,1.+ETA2)
      RETURN
      END


      FUNCTION XV3(X,Q2)
      S=ALOG(ALOG(Q2/.09)/2.9957)
      ETA3=.85-.24*S
      ETA4=3.35+.816*S
      XV3=XV8(X,Q2)-2.*X**ETA3*(1.-X)**ETA4/B(ETA3,1.+ETA4)
      RETURN
      END


      FUNCTION XU(X,Q2)
      XU=.5*(XV8(X,Q2)+XV3(X,Q2))
      RETURN
      END


      FUNCTION XD(X,Q2)
      XD=.5*(XV8(X,Q2)-XV3(X,Q2))
      RETURN
      END


      FUNCTION XUB(X,Q2)
      XUB=XS(X,Q2)
      RETURN
```

```
      END


      FUNCTION XS(X,Q2)
      S=ALOG(ALOG(Q2/.09)/2.9957)
      PS=.75*(.169*EXP(-.747*S)+.429-.488*EXP(-.427*S))
     &  +.0275*EXP(-.427*S)
      XS=(.75*(.0028145*EXP(-.747*S)+.16335*EXP(-.609*S)
     &   -.157*EXP(-.667*S))+.25*.009167*EXP(-.667*S))/PS
      XS=(PS*(1./XS-1.)*(1.-X)**(1./XS-2.))/6.
      RETURN
      END


      FUNCTION XC(X,Q2)
      S=ALOG(ALOG(Q2/.09)/2.9957)
      PC=.25*(.169*EXP(-.747*S)+.429-.488*EXP(-.427*S)-.11*EXP(-.427*S))
      XC=.25*(.0028145*EXP(-.747*S)+.16335*EXP(-.609*S)-.157*
     &  EXP(-.667*S)-.009167*EXP(-.667*S))/PC
      XC=(PC*(1./XC-1.)*(1.-X)**(1./XC-2.))/2.
      RETURN
      END


      FUNCTION XG(X,Q2)
      S=ALOG(ALOG(Q2/.09)/2.9957)
      PG=.571-.169*EXP(-.747*S)
      XG=(.0425397*EXP(-.609*S)-.0090397*EXP(-1.386*S))/PG
      XG=PG*(1./XG-1.)*(1.-X)**(1./XG-2.)
      RETURN
      END

C *********************************************************************
C
C   STRUCTURE FUNCTIONS FROM GLUCK, HOFFMANN AND REYA
C   DORTMUND PREPRINT DO-TH 80/13, MAY 1980
C   REQUESTED FUNCTION :  GAMMA(X)
C
C *********************************************************************
      FUNCTION F2(X,Q2)
      F2=(4.*XU(X,Q2)+XD(X,Q2)+10.*XUB(X,Q2)+2.*XS(X,Q2)
     & +8.*XC(X,Q2))/9.
      RETURN
      END


      FUNCTION BETA(X,Y)
      BETA=GAMMA(X)*GAMMA(Y)/GAMMA(X+Y)
      RETURN
      END


      FUNCTION XU(X,Q2)
      S=ALOG(ALOG(Q2/.16)/3.2189)
      A=.421-.0412*S
      C=2.-.6223*S**.8
      D=3.37+.4319*S
      XU=2.*C*X**A*(1.-X**C)**D/BETA(D+1.,A/C)
```



```
      RETURN
      END


      FUNCTION XD(X,Q2)
      S=ALOG(ALOG(Q2/.16)/3.2189)
      A=.364-.0368*S
      C=2.-.5414*S**.8
      D=5.09+.3463*S
      XD=C*X**A*(1.-X**C)**D/BETA(D+1.,A/C)
      RETURN
      END


      FUNCTION XUB(X,Q2)
      S=ALOG(ALOG(Q2/.16)/3.2189)
      A=.25+.088*S**1.3
      B=.8128*S-2.003*S**1.8+.0831*S**2
      C=3.97*S
      D=7.+1.666*S
      E=.2487*S**2.5
      F=27.8+59.68*S
      XUB=A*(1.+B*X+C*X**2)*(1.-X)**D + E*EXP(-F*X)
      RETURN
      END


      FUNCTION XS(X,Q2)
      S=ALOG(ALOG(Q2/.16)/3.2189)
      A=.0625+.1132*S**1.3
      B=12.64*S-51.70*S**1.8+38.02*S**2
      C=4.448*S
      D=7.+1.562*S
      E=.3081*S**2.5
      F=47.24+67.91*S
      XS=A*(1.+B*X+C*X**2)*(1.-X)**D + E*EXP(-F*X)
      RETURN
      END


      FUNCTION XC(X,Q2)
      XC=0.
      RETURN
      END


      FUNCTION XG(X,Q2)
      S=ALOG(ALOG(Q2/.16)/3.2189)
      A=.9243+2.51*S**0.5
      B=8.558-9.227*S**0.3-.655*S**1.5
      C=53.57-68.78*S**0.3+19.3*S
      D=6.+1.454*S
      E=11.29*S**2
      F=41.24+50.71*S
      XG=A*(1.+B*X+C*X**2)*(1.-X)**D + E*EXP(-F*X)
      RETURN
      END
```