DEA physique des particules,physique mathématique et modélisation

STAGE INFORMATIQUE

Centre de Physique des Particules de Marseille

Interface COMPHEP - PYTHIA

Jessica LEVEQUE

Idée originale : Cristinel DIACONU

janvier 2000

Sujet :

Ecriture d'une interface entre un programme de calcul des sections efficaces d'interaction entre particules élémentaires et un logiciel de reconstruction des événements complets dans le détecteur H1.

Table des matières

Chapitre 1

Simulations en physique des particules expérimentale

Une expérience de physique des particules nécéssite à plusieurs niveaux l'utilisation de moyens informatiques importants : étude des processus de collision, complexité des détecteurs utilisés et confrontation des données avec la théorie.

Ce stage a été effectué au sein du groupe H1 qui étudie les collisions électron-proton à HERA (Hambourg).

1.1 Rôle des simulations

- **Génération Monté - Carlo du processus fondamental**

  Cette première étape permet de calculer la section efficace d'une interaction entre constituants élémentaires : quarks,leptons,bosons de jauge.Ceci permet de générer un grand nombre de processus (particules finales,énergie et impulsions de ces particules) selon le taux de réalisation donné par la section efficace.

- **Génération de l'événement complet**

  Le processus réel doit tenir compte du confinement des quarks dans des hadrons.Après l'interaction,les constituants des particules lourdes forment des jets de hadrons qu'il faut modéliser.D'autre part,il se peut que les particules produites soient instables,il faut alors simuler leur désintégration.

  La génération des processus permet d'avoir une idée du taux d'événements attendu et de leurs caractéristiques moyennes.Elle fournit aussi des renseignements sur le bruit de fond susceptible de gêner la physique recherchée. La connaissance des processus permet d'anticiper sur les éléments qui seront nécéssaires à leur étude : prévisions pour la construction d'un détecteur,stratégie d'analyse des données...

- **Simulation du détecteur**

  Une fois les processus modélisés,les particules résultantes doivent passer dans le détecteur.Cette étape est importante pour plusieurs raisons :

  - Prise en compte de l'acceptance géométrique du détecteur
    $\longrightarrow$ corrections sur le taux d'événements observé puisque le nombre réel de processus simulés est connu.
  - Modélisation de l'interaction des particules incidentes avec la matière du détecteur : ces interactions peuvent être à l'origine de décalages entre les valeurs réelles des variables cinématiques des particules incidentes et les mesures.
    $\longrightarrow$ la connaissance des valeurs initiales permet encore une fois de calculer des facteurs correctifs.

- **Reconstruction**

  La reconstruction des trajectoires et des dépots d'énergie laissés par les particules dans le détecteur permet d'obtenir une image finale au même format que les données réelles pour comparer données expérimentales et théories.

1.2 Motivations du stage

Jusqu'ici toutes les étapes précédentes étaient réalisées par le programme PYTHIA. Nous avons voulu remplacer la partie de calcul des sections efficaces par un logiciel beaucoup plus performant ,COMPHEP, qui permet de calculer les diagrammes de n'importe quel processus à partir d'un modèle théorique choisi : modèle standard,supersymétrique... ou bien implémenté par l'utilisateur.

Chapitre 2

Interface COMPHEP-PYTHIA

2.1 Comphep

Comphep est un logiciel qui permet de calculer des diagrammes de Feynman pour un processus choisi par l'utilisateur.Il est ensuite possible de générer un grand nombre d'événements (impulsion et énergie des particules finales) à partir de ces diagrammes,en ajoutant éventuellement des contraintes sur les variables cinématiques.

La génération Monté-Carlo est présentée sous forme d'un fichier texte.

Exemple de fichier rendu par CompHep :

```
 PROCESS: e1 , u     ->  e1 , d  , W+
 MASSES:   1.000000E+00  1.000000E+00  1.000000E+00  1.000000E+00  8.034267E+01
 StructFun1: OFF
 StructFun2: OFF
 QCD constant scale:     9.12E+01 GeV,  Lambda6=    1.18E-01
 SQRT(S):   9.000E+01
 Rapidity:   0.000E+00
   Weight [pb]      P1_3 [Gev]        P2_3 [Gev]        P3_1 [Gev]
   4.957832E-09  4.5000000000E+01 -4.5000000000E+01  3.8685371881E+00
   5.362104E-10  4.5000000000E+01 -4.5000000000E+01  5.6301979964E+00
   3.230171E-09  4.5000000000E+01 -4.5000000000E+01 -6.6787359300E+00

&  P3_2 [Gev]        P3_3 [Gev]        P4_1 [Gev]        P4_2 [Gev]
&  -2.3939361744E-01 -6.6440089836E+00  1.0686241890E+00 -8.6193679217E-01
&  2.7899926123E-01 -5.9126779059E+00  5.8032859833E-02 -9.7739266321E-02
&  6.9387519977E-01 -4.8403768033E+00  8.3721671374E-01 -7.5477823002E-02

&   P4_3 [Gev]        P5_1 [Gev]        P5_2 [Gev]        P5_3 [Gev]
&   8.1261637415E-01 -4.9371613771E+00  1.1013304096E+00  5.8313926094E+00
&   1.1398123476E+00 -5.6882308562E+00 -1.8125999491E-01  4.7728655582E+00
&   6.3904079839E-01  5.8415192163E+00 -6.1839737677E-01  4.2013360049E+00 ...
```

Remarque : toutes les impulsions sont sur une seule ligne.
Les informations que nous devons récupérer sont:

- le processus.Sur l'exemple il s'agit d'une diffusion électron - quark u.

```
PROCESS: e1 , u    ->  e1 , u
```

- le nombre N de particules
- les masses des particules

```
MASSES:   0.123456E+03  0.123456E+02  0.000000E+00  0.000000E+00
```

- les impulsions et énergies de toutes les particules

```
 Weight [pb]      P1_3 [Gev]        P2_3 [Gev]        P3_1 [Gev]
5.352618E-03  4.5000000000E+01 -4.5000000000E+01  3.7594872385E+00

   P3_2 [Gev]        P3_3 [Gev]        P4_1 [Gev]        P4_2 [Gev]
-1.1087948680E+00 -4.4828973105E+01 -3.7594872385E+00  1.1087948680E+00

   P4_3 [Gev]
4.4828973105E+01
```

Ces données doivent être mises sous une forme utilisable par PYTHIA :

- les numéros des particules seront placées dans la première colonne d'un tableau K(4000,5) à deux indices. A chacune est associé un code qui permettra son identification par PYTHIA ,et ainsi la reconstruction des désintégrations des particules composites ou la formation des jets de hadrons pour les quarks.
- Un second tableau P(4000,5) à deux indices contiendra les numéros des particules ,leur masse ,les trois composantes de l'impulsion et leur énergie.

2.2 Code

L 'interface est programmée en Fortran (cf ANNEXE A).Les variables N,P,K utilisées par PYTHIA sont placées dans un COMMON :

```
COMMON/LUJETS/N,K(4000,5),P(4000,5)

     integer K,N
      real P
```

Deux subroutines principales constituent l'interface:

- Subroutine JLINIT

  Elle permet de garder les paramètres qui resteront constants pour l'ensemble des événements,c'est à dire le nombre de particules initiales et finales,leur nom et leur masse.

  Les noms des particules sont extraits de la chaine de caractères contenant le processus par la subroutine EXTRAIRE et associés à un code pour remplir le tableau K dans KTRANSL.Les codes et noms de particules ont été placés dans des tableaux :

```
DATA CCODE /'d   ','u   ','s   ','c   ','b   ','t   ',
  &         'D   ','U   ','S   ','C   ','B   ','T   ',
  &         'e1  ','n1  ','e2  ','n2  ','e3  ','n3  ',
  &         'E1  ','N1  ','E2  ','N2  ','E3  ','N3  ',
  &         'g   ','A   ','Z   ','W+  ','W-  ','H   '/

DATA KCODE / 1    ,  2   , 3    , 4    , 5    , 6    ,
  &          -1   , -2   , -3   , -4   , -5   , -6    ,
  &         11   , 12   , 13   , 14   , 15   , 16   ,
  &         -11  , -12  , -13  , -14  , -15  , -16   ,
  &         21   , 22   , 23   , 24   , -24  , 25 /
```

  Les masses sont placées dans P,puis on se positionne pour pouvoir lire les quadrivecteurs de chaque événement.

- Subroutine JLEVT

  Cette instruction est exécutée en boucle jusqu'à la fin du fichier et permet de remplir P pour chaque processus.

2.3 Pythia

La routine INTERFACE a été insérée dans le code de PYTHIA grâce à CMZ,qui permet de créer facilement plusieurs versions d'un même programme.La première version créée ici s'appelle CompGEN 0.0

Utilisation du programme:

Un fichier STEER contient toutes les options et paramètres d'entrée de Pythia.Il suffit donc de lui donner le nom du fichier CompHep qui contient les événements :

```
*=========================================================================
*
PON
*-------------------------------------------------------------------------
$RTX 0 / $TEXT ! GENERATOR COMMENT BANK (User's text in cols. 1-72)
 USER: J.Leveque   DATE:        CPU: marhaba
 GENERATOR:  CompGEN 0.0, JETSET 7.4,
END$
*-------------------------------------------------------------------------
POFF
GSTX 0 / $TEXTC ! GENERATOR STEERING TEXT BANK
*-------------------------------------------------------------------------
* COMPGEN STEERING
CHEV 'epair.events'   ! comphep event file
EPRO 920.0  ! Proton energy
CSI1 5 6

**********************************************************************
```

L'instruction

```
DOIT
```

(shellscript , ANNEXE B) génère les événements complets (fichier GEN) et écrit les données dans le format standard H1 : banques (système BOS ,stockage dynamique des données) et caractéristiques des particules produites dans chaque événement,ce qui permet de simuler la réponse du détecteur (fichier SIM) et de reconstruire l'image de l'événement dans le détecteur H1 (fichier REC). Ce dernier fichier est visualisable grâce à un Event Display.

Chapitre 3

Application : deux exemples

Les événements présentés sont des collisions électron - proton donnant trois leptons dans l'état final.

Pour chacun des exemples sont fournies les données rendues par CompHep (section efficaces , paramètres de la génération , coupures...) et l'image de l'événement reconstruit dans le détecteur H1.

3.1 $e^-p \longrightarrow e^+e^-e^+$ quarks

Les événements générés contiennent une paire électron-positron dans l'état final.Plusieurs contraintes ont été placées sur les impulsions et les angles entre les particules afin que toutes les particules finales soient visibles dans le détecteur:

```
 6 cuts are introduced
  5.0000000000E+00 < T4        <  1.0000000000E+03
  2.0000000000E+00 < T5        <  1.0000000000E+03
  1.0000000000E+01 < T3        <  1.0000000000E+03
  1.5000000000E+01 < A13       <  1.5500000000E+02
  1.5000000000E+01 < A14       <  1.2000000000E+02
  1.5000000000E+01 < A15       <  1.4000000000E+02
```

Calcul de la section efficace totale par CompHep:

```
 Cross section [pb]  = 0.14593E-02 +- 0.1440E-03    CHI**2/IT = 1.8517
```

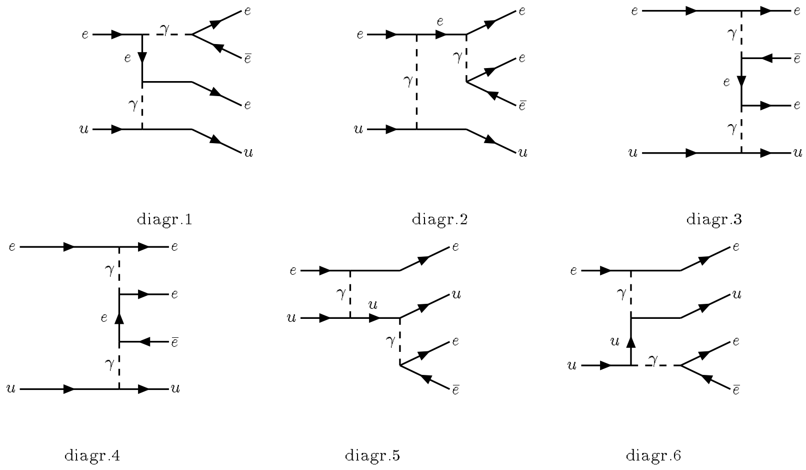

Figure 3.1: Diagrammes de Feynmann

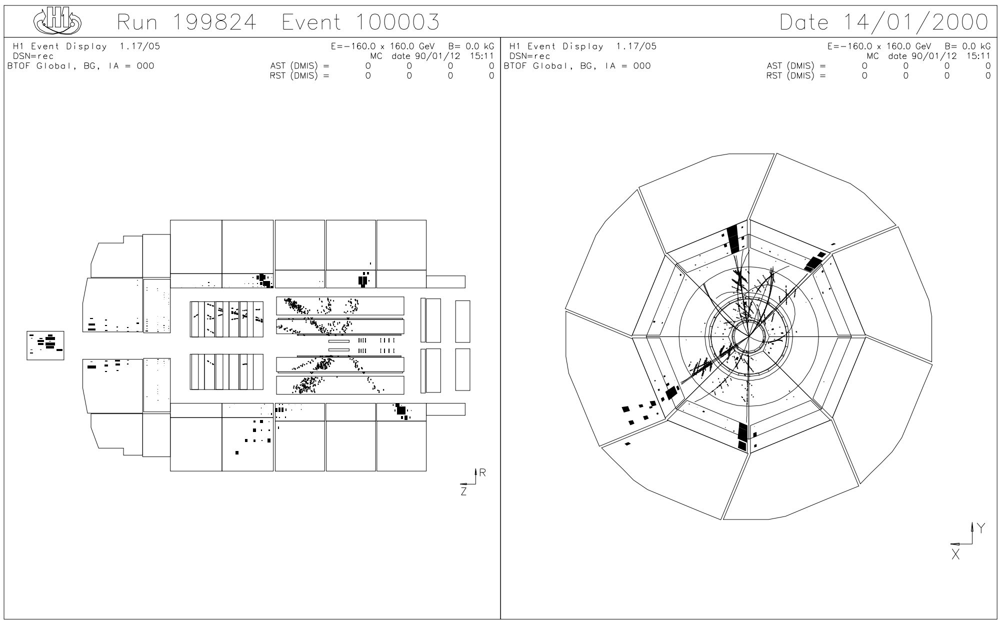

Figure 3.2: événement à 3 électrons et un jet hadronique reconstruit dans H1

Jan 21, 00 19:58 **epair2.dat** Page 1/2

```
          CompHEP kinematics module    v. 33 ( 10.12.97)
  The session parameters:
e1 ,d   -> e1 ,e1 ,E1 ,d               1
                    2 = the session number
                 3000   = NCALL
                   13   = ITMX
                    1     = MDS
 OFF
 OFF
   320.00000000000      = SQRTS
  0.00000000000000E+00 = C.M. rapidity
                    1 = number of physical model parameters
  0.31223000000000      = EE
 ===================== Kinematical scheme ( 2) -> ( 4) ========================
 Decay  1 : (+p1+p2            ) -> (+p3               ) , (+p4+p5+p6         )
 Decay  2 : (+p4+p5+p6         ) -> (+p4               ) , (+p5+p6            )
 Decay  3 : (+p5+p6            ) -> (+p5               ) , (+p6               )
  6 cuts are introduced
  5.0000000000E+00 < T4          <  1.0000000000E+03
  2.0000000000E+00 < T5          <  1.0000000000E+03
  1.0000000000E+01 < T3          <  1.0000000000E+03
  1.5000000000E+01 < A13         <  1.5500000000E+02
  1.5000000000E+01 < A14         <  1.2000000000E+02
  1.5000000000E+01 < A15         <  1.4000000000E+02
 =================== There are introduced  1 regularizations ==================
 +p4+p5              MASS= 1.0000000000E-01  WIDTH= 1.0000000000E-01    DEG=2
 ========  QCD  parameters ===========
 Q**2                 :    0.911884E+02  0 0 0 0 0 0 0 0 0 0
 Lambda(6)            :    0.1185
 F   gauge invariant widths
 F   running widths
  ===== BATCH PARAMETERS =====
     10000         5    10000        5   T
    0   0  0.0000E+00   100.0
 ==============================================================================
 INTEGRATION BY VEGAS     NCALL=   3000
  IT      CURRENT RESULTS                ACCUMULATED RESULTS        CHI**2
        Cross section [pb]  Error %    Cross section [pb]  Error %
 ------------------------------------------------------------------------------
   1   2.437302D-03        48.75      2.437302D-03        48.75    0.00
   2   1.691218D-03        35.15      1.840584D-03        28.88    0.32
   3   4.507106D-03        74.45      1.905864D-03        27.55    0.47
   4   6.962974D-03        57.77      1.990591D-03        26.15    0.83
   5   2.808113D-03        35.17      2.168376D-03        21.24    0.76
   6   6.776236D-03        75.25      2.205665D-03        20.80    0.77
   7   1.341778D-02        65.95      2.235712D-03        20.49    0.91
   8   2.292701D-03        36.72      2.248731D-03        17.89    0.78
   9   2.760495D-03        28.85      2.352779D-03        15.26    0.72
  10   2.766284D-02        85.98      2.358549D-03        15.23    0.77
  11   1.607877D-03        38.30      2.168080D-03        14.31    0.80
  12   1.024576D-03        18.01      1.323465D-03        11.98    1.64
  13   2.097088D-03        16.39      1.459271D-03         9.87    1.85
 ______________________________________________________________________________

   Cross section [pb]  = 0.14593E-02 +- 0.1440E-03     CHI**2/IT = 1.8517

  Writing events to file !
        38272 : Number of generated events
   0.1822E-02 : Max Weight
   0.1528E-06 : Mean Weight (Sum_of_weights/Num_of_events)
         0.01 : Efficiency, ( mean/max*100 )
```

Jan 21, 00 19:58 **epair2.dat** Page 2/2

```
e1 ,d   -> e1 ,e1 ,E1 ,d               1
                    3 = the session number
                 3000   = NCALL
                   13   = ITMX
                    1     = MDS
 OFF
 OFF
   320.00000000000      = SQRTS
  0.00000000000000E+00 = C.M. rapidity
                    1 = number of physical model parameters
  0.31223000000000      = EE
 ===================== Kinematical scheme ( 2) -> ( 4) ========================
 Decay  1 : (+p1+p2            ) -> (+p3               ) , (+p4+p5+p6         )
 Decay  2 : (+p4+p5+p6         ) -> (+p4               ) , (+p5+p6            )
 Decay  3 : (+p5+p6            ) -> (+p5               ) , (+p6               )
  6 cuts are introduced
  5.0000000000E+00 < T4          <  1.0000000000E+03
  2.0000000000E+00 < T5          <  1.0000000000E+03
  1.0000000000E+01 < T3          <  1.0000000000E+03
  1.5000000000E+01 < A13         <  1.5500000000E+02
  1.5000000000E+01 < A14         <  1.2000000000E+02
  1.5000000000E+01 < A15         <  1.4000000000E+02
 =================== There are introduced  1 regularizations ==================
 +p4+p5              MASS= 1.0000000000E-01  WIDTH= 1.0000000000E-01    DEG=2
 ========  QCD  parameters ===========
 Q**2                 :    0.911884E+02  0 0 0 0 0 0 0 0 0 0
 Lambda(6)            :    0.1185
 F   gauge invariant widths
 F   running widths
  ===== BATCH PARAMETERS =====
     10000         5    10000        5   T
    0   0  0.0000E+00   100.0
```

Friday January 21, 2000 epair2.dat 1/1

Figure 3.3: données de la génération Monte-Carlo “paire d'électrons “

3.2 $e^- p \longrightarrow e^+ \tau^- \tau^+$ quarks

Nous avons ici généré la production d'une paire de leptons τ. Deux coupures ont été faites :

- implusion transverse d'un des τ supérieure à 5 GeV
- angle entre le proton incident et l'un des τ compris entre 15 et 155 degrés (angle du détecteur)

```
2 cuts are introduced
1.5000000000E+01 < A14       <  1.5500000000E+02
5.0000000000E+00 < T4        <  1.0000000000E+03
```

Section efficace totale :

```
Cross section [pb]  = 0.76195E-03 +- 0.5424E-03     CHI**2/IT = 7.1937
```

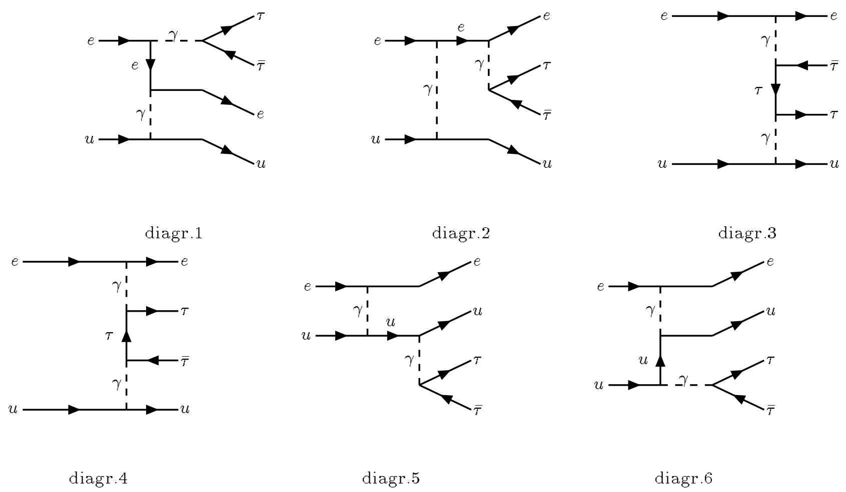

Figure 3.4: diagrammes de feynman "paire de leptons τ "

Jan 21, 00 19:56 **taupair2.dat** Page 1/2

```
          CompHEP kinematics module   v. 33 ( 10.12.97)
  The session parameters:
e1 ,u   -> e1 ,e3 ,E3 ,u             1
                     1 = the session number
                 10000   = NCALL
                     5   = ITMX
                     1     = MDS
 OFF
 OFF
   320.00000000000     = SQRTS
  0.00000000000000E+00 = C.M. rapidity
                     2 = number of physical model parameters
  0.31223000000000     = EE
   1.7770000000000     = Mt
====================== Kinematical scheme ( 2) -> ( 4) ========================
Decay  1 : (+p1+p2           ) -> (+p3               ) , (+p4+p5+p6        )
Decay  2 : (+p4+p5+p6        ) -> (+p4               ) , (+p5+p6           )
Decay  3 : (+p5+p6           ) -> (+p5               ) , (+p6              )
  2 cuts are introduced
  1.5000000000E+01 < A14        <  1.5500000000E+02
  5.0000000000E+00 < T4         <  1.0000000000E+03
================== There are introduced  0 regularizations ==================
========  QCD  parameters ===========
Q**2                :    0.911884E+02  0 0 0 0 0 0 0 0 0 0
Lambda(6)           :    0.1185
F   gauge invariant widths
F   running widths
  ===== BATCH PARAMETERS =====
    10000        5    10000        5   T
   0   0  0.0000E+00   100.0
===============================================================================
INTEGRATION BY VEGAS     NCALL=  10000
 IT      CURRENT RESULTS              ACCUMULATED RESULTS        CHI**2
      Cross section [pb]  Error %    Cross section [pb]  Error %
-------------------------------------------------------------------------------
  1  -7.305251D+07       100.00     -7.305251D+07       100.00    0.00
  2   6.718025D-04        81.00      6.718025D-04        81.00    1.00
  3   1.145355D-02        60.08      7.387997D-04        73.43    1.72
  4   9.804634D-02        59.05      7.473407D-04        72.59    2.09
  5   4.541251D-01        21.04      7.619520D-04        71.19    7.19
_______________________________________________________________________________

  Cross section [pb]  = 0.76195E-03 +- 0.5424E-03     CHI**2/IT = 7.1937

  Writing events to file !
       43740 : Number of generated events
   0.1461E+08 : Max Weight
  -0.3340E+03 : Mean Weight (Sum_of_weights/Num_of_events)
        0.00 : Efficiency, ( mean/max*100 )
  NEGATIVE POINTS CONTRUBUTION (%)   -99.99999916927329




e1 ,u   -> e1 ,e3 ,E3 ,u             1
                     2 = the session number
                 10000   = NCALL
                     5   = ITMX
                     1     = MDS
```

Jan 21, 00 19:56 **taupair2.dat** Page 2/2

```
OFF
OFF
  320.00000000000     = SQRTS
 0.00000000000000E+00 = C.M. rapidity
                    2 = number of physical model parameters
 0.31223000000000     = EE
  1.7770000000000     = Mt
===================== Kinematical scheme ( 2) -> ( 4) ========================
Decay  1 : (+p1+p2           ) -> (+p3               ) , (+p4+p5+p6        )
Decay  2 : (+p4+p5+p6        ) -> (+p4               ) , (+p5+p6           )
Decay  3 : (+p5+p6           ) -> (+p5               ) , (+p6              )
 2 cuts are introduced
 1.5000000000E+01 < A14        <  1.5500000000E+02
 5.0000000000E+00 < T4         <  1.0000000000E+03
================== There are introduced  0 regularizations ==================
========  QCD  parameters ===========
Q**2                :    0.911884E+02  0 0 0 0 0 0 0 0 0 0
Lambda(6)           :    0.1185
F   gauge invariant widths
F   running widths
 ===== BATCH PARAMETERS =====
    10000        5    10000        5   T
   0   0  0.0000E+00   100.0
```

Friday January 21, 2000 taupair2.dat 1/1

Figure 3.5: données de la génération Monte-Carlo "paire de leptons τ "

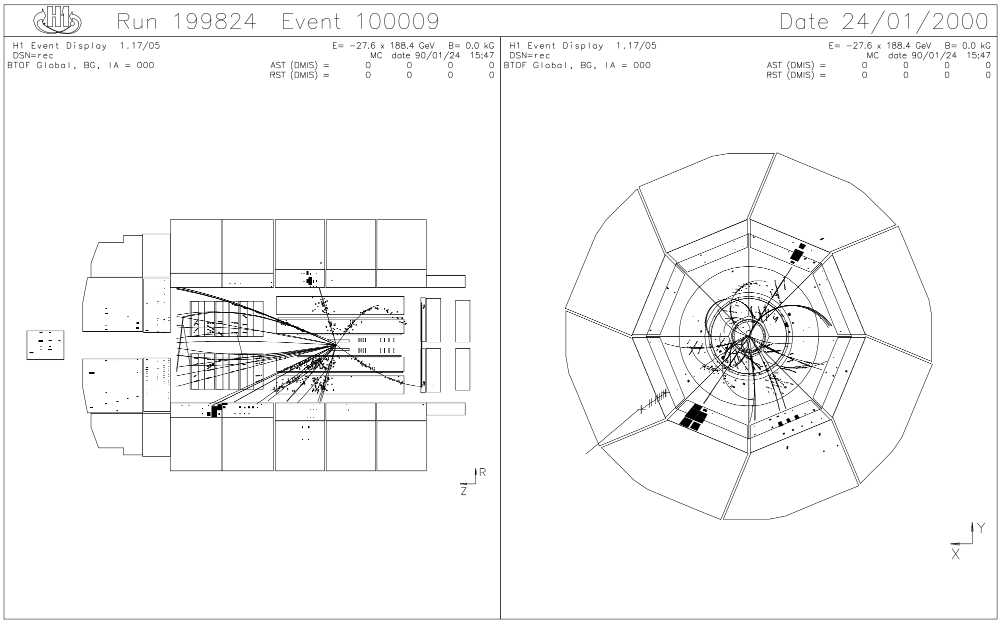

Figure 3.6: événement à 2 τ , un électron et un jet hadronique reconstruit dans H1

3.3 Conclusion

Le but de ce stage a été de remplacer le générateur d'événements de Pythia qu'utilise le groupe H1 par le logiciel CompHep.Le nouveau programme permettra de tenir compte d'un plus grand nombre de processus pour les prédictions théoriques et de simuler des événements au-delà du modèle standard.
Ce programme sera utilisé lors du stage de DEA, pour rechercher des déviations par rapport aux prédictions théoriques du modèle standard à haute énergie,en particulier dans les événements multi-leptons.

Chapitre 4

Annexes

```
        Program interface

        implicit none

        COMMON/LUJETS/N,K(4000,5),P(4000,5)

        integer K,N
        real P

cccc    Ouverture du fichier source COMPHEP

        open(unit=50,file='c')

cccc    Initialisation  N,K

        call JLINIT(50)

ccccc  Lecture des Monte Carlo

        nev=50000
        do iev=1,nev
           call JLEVT(50,kerr)
           if(kerr.eq.1) goto 200
        enddo
200     continue

        END

cccccccccccccccccccccc SUBROUTINES  cccccccccccccccccccccccccccccccccc

        SUBROUTINE JLINIT(unite)

c        implicit none

        COMMON/LUJETS/N,K(4000,5),P(4000,5)
        real P
        integer N,K

        integer kk
        common /kodepar/kk(4000,5)

        COMMON/local/ni

        integer ni,nf,unite
        integer i,j,m,compte,len_final,len_init
        logical arret
        character process*100,nom_in(100)*4,nom(100)*4
        character initstr*50,final*50
        character*9 DUM

        parameter nmemo=30
        integer KCODE(nmemo)
        character CCODE(nmemo)*4

        DATA CCODE /'d ','u ','s ','c ','b ','t ',
     &             'D ','U ','S ','C ','B ','T ',
     &             'e1 ','n1 ','e2 ','n2 ','e3 ','n3 ',
     &             'E1 ','N1 ','E2 ','N2 ','E3 ','N3 ',
     &             'g ','A ','Z ','W+ ','W- ','H '/

        DATA KCODE / 1    , 2    , 3    , 4    , 5    , 6    ,
     &               -1   , -2   , -3   , -4   , -5   , -6   ,
```



```
     &               11   , 12   , 13   , 14   , 15   , 16   ,
     &               -11  , -12  , -13  , -14  , -15  , -16  ,
     &               21   , 22   , 23   , 24   , -24  , 25 /

ccccc   Lecture du processus

        read(unite,10)process

10      format(10x,1A100)
20      format(A9,10e14.6e2)
21      format(/////A)
22      format(a4,i4)

ccccc   Determination du nombre de particules N = ni + nf
c            ni = nb particules initiales
c            nf = nb particules finales

c   tab de particules initiales

          i=1
          j=1
          do while (process(i:i).ne.'-')
             initstr(j:j)=process(i:i)
             i=i+1
             j=j+1
          end do
         len_init=i

         j=1
         i=i+3

c    tab de particules finales

         arret=.false.
         compte=0
         do while (.not.arret.and.i.lt.100)
             final(j:j)=process(i:i)
             if (process(i:i).eq.' ')compte=compte+1
             if (compte.gt.10)arret=.true.
             j=j+1
             i=i+1
         end do
         len_final=j

cccc   extraction de ni,nf,nom et N

          call  extraire(initstr,len_init,0,nom_in,ni)
          call  extraire(final,len_final,ni,nom_in,nf)
          N=ni+nf

cccc   Suppression des espaces avant les noms de particules

          do m=1,N
            call salign(nom_in(m),nom(m))
          enddo

ccccc    Remplissage du tableau K

         do i=1,N
            K(i,1)=ktransl(nom(i),CCODE,KCODE)
            KK(i,1)=k(i,1)
            write(*,22)nom(i),K(i,1)
```


Figure 4.1: ANNEXE A : Code de l'interface COMPHEP-PYTHIA

```
      enddo

ccccc     Remplissage de P(n,5),masse de la particule n

          read(unite,20) DUM,(P(i,5),i=1,N)
          write(*,20)DUM,(P(i,5),i=1,N)

ccccc     Saut de lignes

          read(unite,21) s

          return
          END


          SUBROUTINE JLEVT(unite,kerr)

          implicit none

          COMMON/LUJETS/N,K(4000,5),P(4000,5)
          integer  K,N
          real     P

          integer kk
          common /kodepar/kk(4000,5)

          COMMON/local/ni
          integer kerr
          integer ni

          integer i,j,m,unite
          real wgt

50        format(e15.6e2,20e18.10e2)

ccccc     Remplissage des quadrivecteurs

ccc       Impulsions

          read(unite,50,err=100)
     &        WGT,(P(i,3),i=1,ni),((P(j,m),m=1,3),j=ni+1,N)
          do i=1,ni
             P(i,1)=0
             P(i,2)=0
          enddo

ccc      Energies + code

         do i=1,N
            P(i,4)=sqrt(P(i,1)*P(i,1)+P(i,2)*P(i,2)
     &                  +P(i,3)*P(i,3)+P(i,5)*P(i,5))
            k(i,1)=kk(i,1)
             write(*,22)  K(i,1)
22       format(i4)
         enddo

         kerr=0
         return
 100     continue
         kerr=1
         return
         END
```



```
          SUBROUTINE extraire(chaine,len_chaine,indice,nom,nb_part)

          implicit none

          CHARACTER chaine*(*),nom(100)*4
          INTEGER   len_chaine,indice,nb_part,p,j

          nb_part=1
          j=0
          p=1
          do while(j.lt.len_chaine)
             j=j+1
             if(chaine(j:j).eq.',') then
                 nom(nb_part+indice)=chaine(p:j-1)
                 nb_part=nb_part+1
                 p=j+1
             endif
          enddo
          nom(nb_part+indice)=chaine(p:j-1)

          END

          SUBROUTINE salign(chaine_in,chaine_out)

          implicit none

          CHARACTER chaine_in*4,chaine_out*4
          INTEGER   m,j

          chaine_out='  '
          j=0
          m=0
          do j=1,4
             if(chaine_in(j:j).ne.' ') then
                 m=m+1
                 chaine_out(m:m)=chaine_in(j:j)
             endif
          enddo

          END

         integer FUNCTION ktransl(nom,CCODE,KCODE)

          COMMON/LUJETS/N,K(4000,5),P(4000,5)
          real P
          integer N,K

          integer KCODE(*)
          character CCODE(*)*4

          character nom*4
          integer j

          j=1
          do while(nom.ne.CCODE(j))
             j=j+1
          enddo
          ktransl=KCODE(j)

          END
```


Figure 4.2: ANNEXE A : Code de l'interface COMPHEP-PYTHIA

```
echo "Simulate and Reconstruct with CompGen (JL&CD Inc.@copyright)"
rm -f compgen ; rm -f gen ; rm -f gen.out; rm -f log ; make ;
      compgen <steer > gen.out
rm -f sim ; rm -f sim.out ; /h1/h1sim/bin/h1sim30906 < sim_steer > sim.out
rm -f rec.out ; rm -f rec ; /h1/h1rec9/bin/h1rec990304 < rec_steer > rec.out
```

Figure 4.3: ANNEXE B : Shellscript de l'instruction DOIT