空気シャワー粒子の
横分布シミュレーション

宇宙粒子研究室

学籍番号　　11061061

氏名　　柳澤　壽利

目次

1. 目的

宇宙から飛来する宇宙線が空気シャワー現象を経て、地上付近で観測される粒子の分布と、距離に対する粒子密度を、CORSIKA を用いてシミュレーションする。また、実際に測定した空気シャワー粒子の横分布をシミュレーション結果と比較すること。

横分布とは：空気シャワー主軸からの距離と粒子密度の関係を表したもの。

2 原理

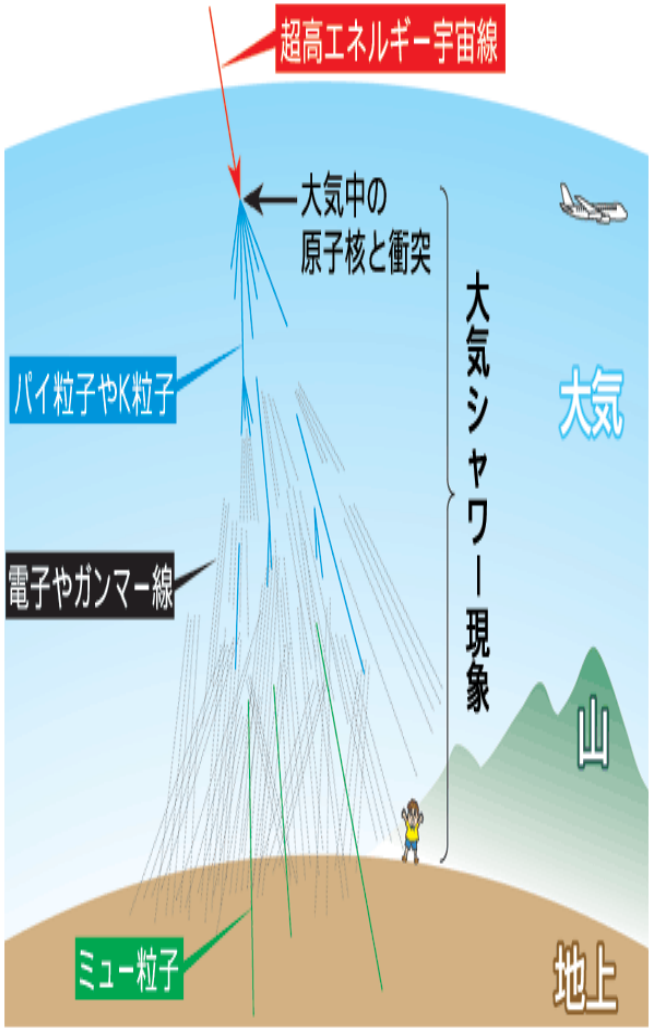
2-1　宇宙線と空気シャワー

宇宙線はほとんどが超新星爆発や太陽などで起こる爆発などから発生する高エネルギーの粒子とされており、これは一次宇宙線と呼ばれている。約９０％が陽子、約８％がアルファ粒子（ヘリウムの原子核)、その他の粒子が約１％含まれる。これらの粒子が地球の大気圏に入ると、高度数十㎞で空気中の窒素や酸素などの原子核と衝突して、中性子や陽子をはじき飛ばし、パイ中間子や K 粒子を発生させ、さらに衝突を繰り返していくと、電子やγ線、μ粒子などが発生する。この様子のことを空気シャワーという。

大気を進むにつれて空気シャワーは発達し、空気シャワー中の粒子数が増加するが、それに伴って１粒子当たりのエネルギーは低くなっていく。やがて、エネルギーの低くなった粒子は新たに粒子を生成出来なくなり、空気シャワーは減衰する。 生成された粒子のうち、寿命の短いものは崩壊し、残ったガンマ線・電子・ミュー粒子などの粒子が地表に複数同時に到来する。

下図は空気シャワーの様子を表している。

今回は地表付近（海抜 110m）での宇宙線の分布分布を解析した。



2-2　ORSIKA プログラムとは

CORSIKA とは宇宙から飛来する粒子の個数や入射する角度、粒子のもつエネルギーなどの情報を入力することで大気中での空気シャワー現象のシミュレーションを行うものである。今回は 10^{15}eV のエネルギーを持つ 1 つの粒子が地上に対して天頂角 20 度で入射したと仮定してシミュレーションを行った。

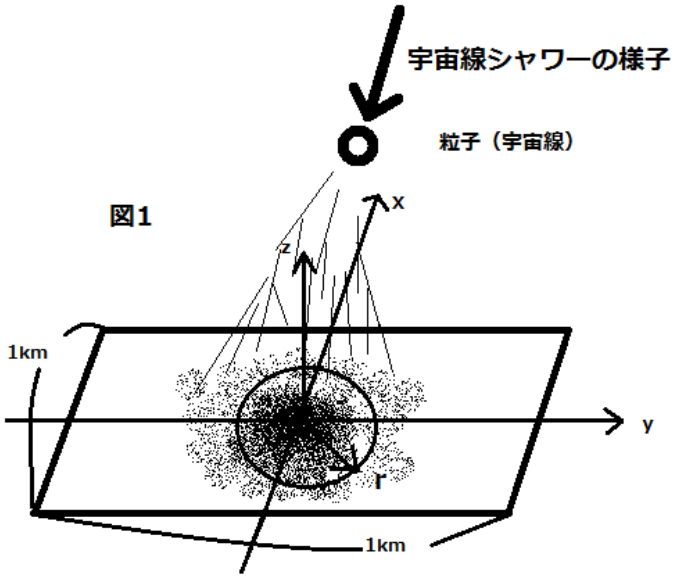


図 1 は宇宙線が空気シャワー現象を繰り返し、地上（測定高度面）まで到達する様子を表している。

2-3　空気シャワー粒子の相互作用

ガンマと電子は空気シャワー中で相互作用を引き起こす。
主に、制動放射と電子対生成である。

$\gamma \Longrightarrow e^{+} + e^{-}$　（電子対生成）

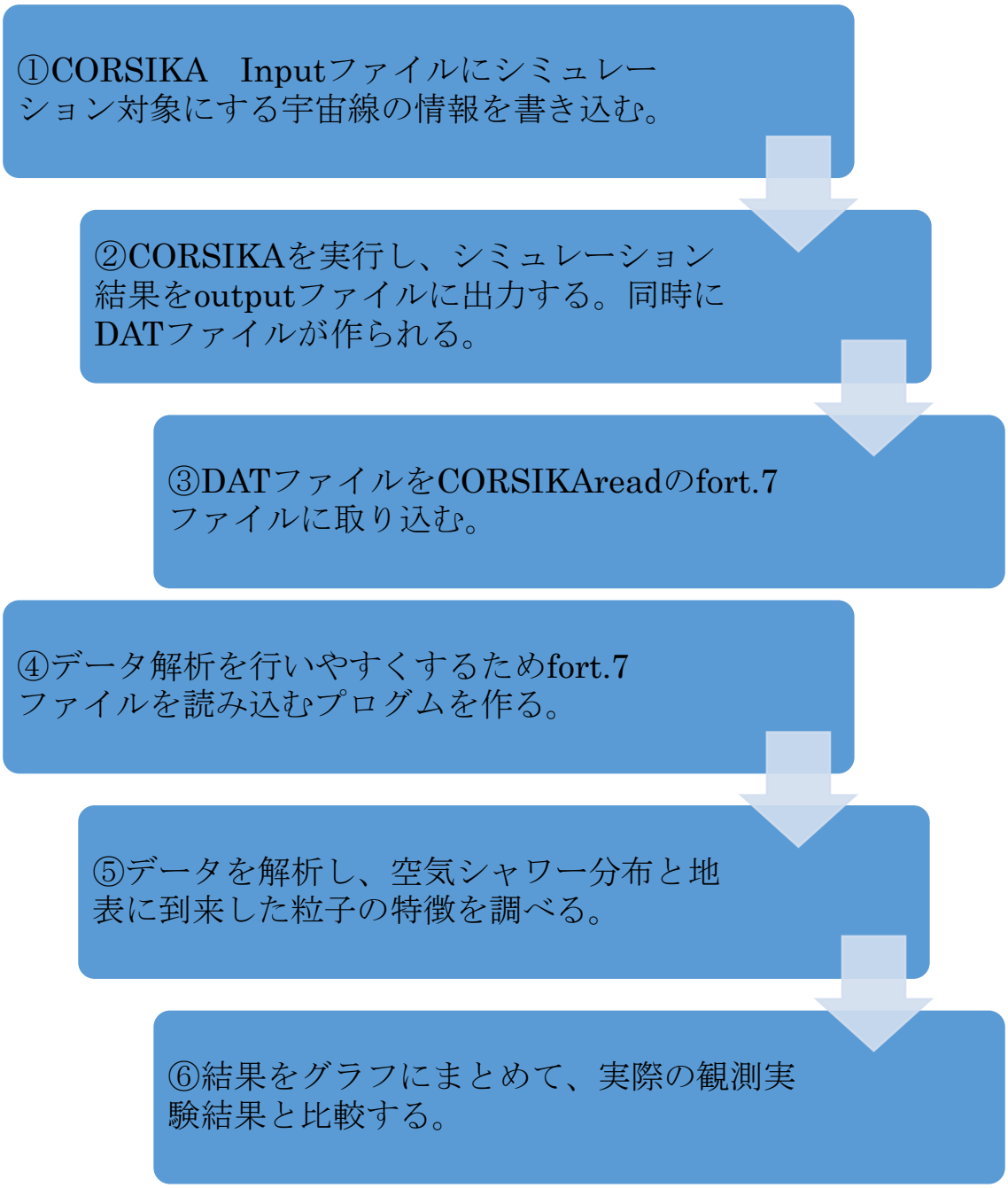
$e^{\pm} \Longrightarrow e^{\pm} + \gamma$　（制動放射）

$\pi^{+} \Longrightarrow \mu^{+} + \nu_{\mu}$

$\pi^{-} \Longrightarrow \mu^{-} + \nu_{\mu}$

3 方法

3-1 方法チャート



3-2　Input File の内容

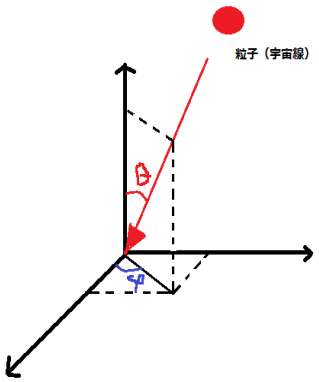
①で書いた Input　file を下記に記す。

Input

```
   RUNNR      12                                  run number
＊EVTNR      1                                   number of first shower event
＊NSHOW      1                                   number of showers to generate
＊PRMPAR     14                                  particle type of prim. particle
＊ERANGE     1.E6    1.E6                        energy range of primary particle
＊THETAP     20.     20.                         range of zenith angle (degree)
＊PHIP       -180.   180.                        range of azimuth angle (degree)
   SEED       1      0      0                    seed for 1. random number sequence
   SEED       2      0      0                    seed for 2. random number sequence
＊OBSLEV     110.E2                              observation level (in cm)
   FIXCHI     0.                                 starting altitude (g/cm**2)
   MAGNET     20.0    42.8                       magnetic field centr. Europe
   HADFLG     0    0    0    0    0    2         flags hadr.interact.&fragmentation
   ECUTS      0.3    0.3    0.003    0.003       energy cuts for particles
   MUADDI     T                                  additional info for muons
   MUMULT     T                                  muon multiple scattering angle
   ELMFLG     T      T                           em. interaction flags (NKG,EGS)
   STEPFC     1.0                                mult. scattering step length fact.
   RADNKG     200.E2                             outer radius for NKG lat.dens.distr.
   LONGI      T    10.    T    T                 longit.distr. & step size & fit & out
   ECTMAP     1.E4                               cut on gamma factor for printout
   MAXPRT     1                                  max. number of printed events
   DIRECT     ./                                 output directory
   USER       you                                user
   DEBUG      F    6    F    1000000             debug flag and log.unit for out
   EXIT                                          terminates input
```

以上で、今回シミュレーションを行った粒子の情報を書き込んだ。

水色で表した部分は、粒子情報の入力で特に重要であるため補足すると、
EVTNR とは元となる粒子の数であり、1 つに設定した。
NSHOW とはシミュレーション対象とする粒子の数であり、1 つに設定した。
PRMPAR とは入射粒子の種類であり、陽子（particle:14）に設定した。
ERANGE とは粒子の持つエネルギーの幅であり、10^15ev に設定した。
THETAP とは入射する粒子と z 軸との間の角度（天頂角）θ にあたり、20 度に設定した。
PHIP とは入射する粒子と x 軸との間の角度 φ にあたり、-180～180 度に設定した。
OBSLEV とは観測地の標高のことであり、110m に設定した。



Particle ID 表		
ID		Particle(粒子名)
1		γ（ガンマ）
2		e^+（陽電子）
3		e^-（負電子）
5		μ^+
6		μ^-
14		プロトン（陽子）

3-3 データの解析

Fort.7 にいれたデータは以下のように実数で表示される。
なお、ここに書いたのは始めの数行分だけだが、実際には数十万行のデータが存在する。

5.53100E+03 -1.03814E+01 6.60524E-01 2.70060E+01 -4.96633E+04 3.31609E+03 8.42724E+04
7.65310E+04 -2.30723E+01 2.04295E+00 6.24492E+01 7.86502E+05 -6.34608E+04 2.17937E+06
6.53100E+03 -2.18950E+01 2.25325E+00 5.95433E+01 -1.34657E+04 1.17421E+04 8.38544E+04
7.65510E+04 -4.59461E+00 1.99518E-01 1.27945E+01 4.33445E+05 -3.61116E+04 1.20475E+06
6.55100E+03 -3.86128E+00 1.52886E-01 1.05621E+01 3.51444E+03 -1.36726E+04 8.36400E+04
7.55510E+04 -2.00510E+01 1.82139E+00 5.51431E+01 4.41051E+05 -3.53648E+04 1.22664E+06
5.55100E+03 -1.91375E+01 1.64143E+00 5.26950E+01 -8.09365E+02 2.89992E+03 8.37003E+04

データの見方

5.53100E+03 -1.03814E+01 6.60524E-01 2.70060E+01 -4.96633E+04 3.31609E+03 8.42724E+04

これは上にあげたデータの一番上の 1 行である。1 行に、7 つのデータ要素が書いてあるのがわかる。

1 列目の無色で示した実数 5531

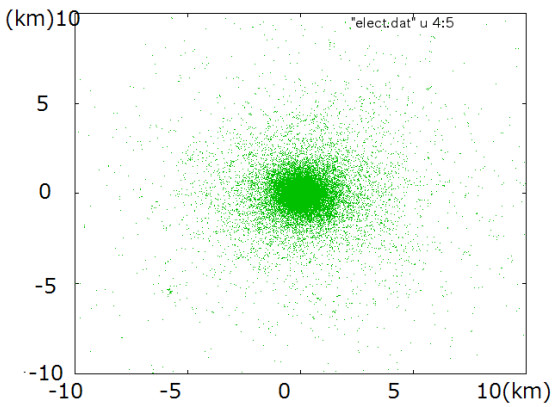
この実数を 1000 で割った整数部分が粒子の種類を表す Particle ID である。また、Particle ID を抜き 10 で割った整数部分が大気中で粒子が衝突した回数である。そして、最後に残った整数が階層を表している。よって、5531　は階層 1 で 53 回衝突した時のミューオンと読める。

2 列目の緑色で示した実数
> x 方向の運動量
3 列目の水色で示した実数
>y 方向の運動量
4 列目の紫色で示した実数
>z 方向の運動量
5 列目の赤色で示した実数
>x 方向の距離(cm)
6 列目の黄色で示した実数
>y 方向の距離(cm)
7 列目の灰色で示した実数
>経過時間

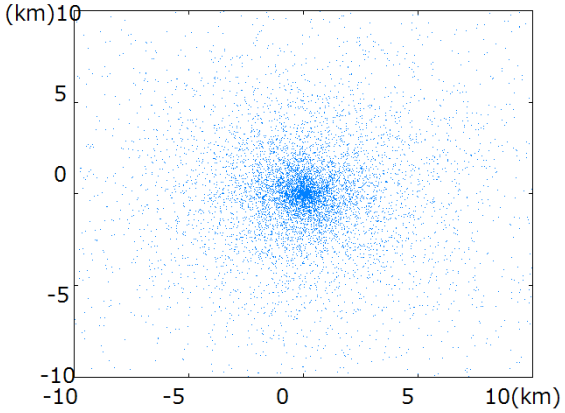
4 結果

4-1 粒子の分布図

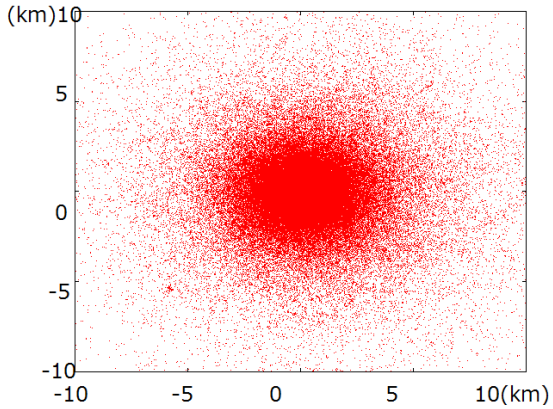
地表面で、原点を中心として縦横 10km 内での粒子の分布を表すグラフである。



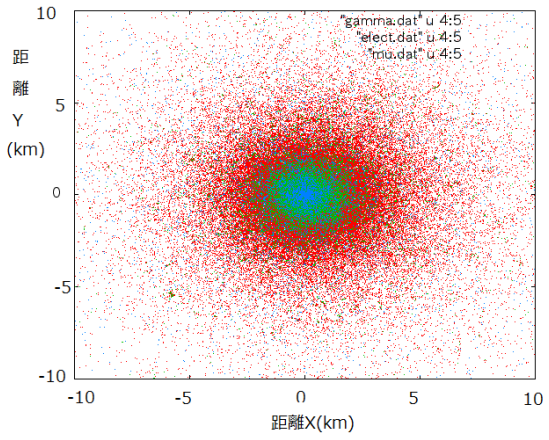
電子の分布（粒子数：40271）



ミューオンの分布（粒子数：8740）

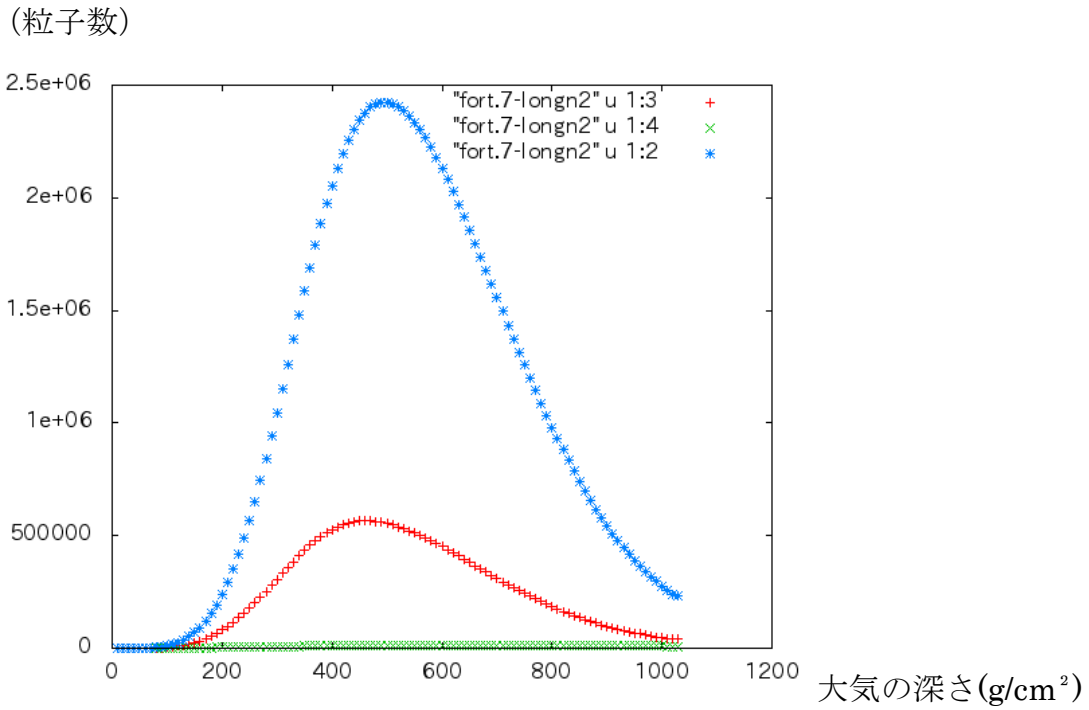


ガンマの分布（粒子数：233731）



全粒子の分布（粒子数：291798）

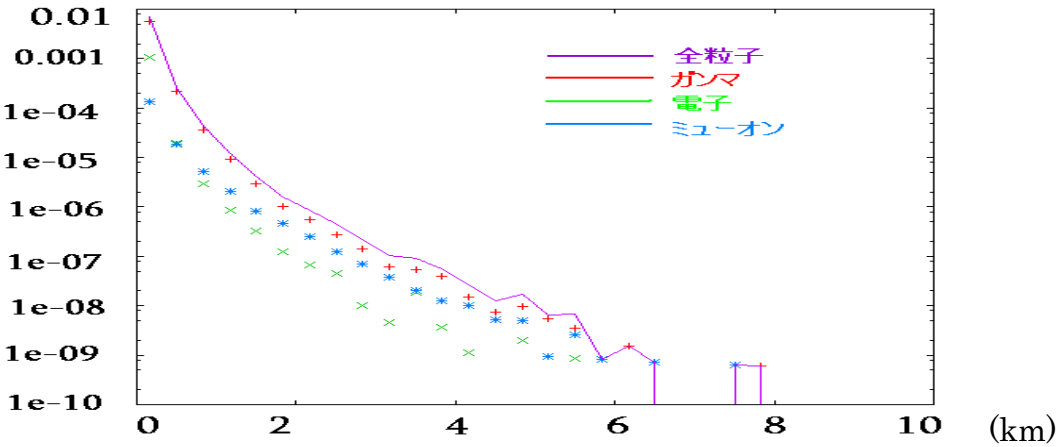
4-2 縦方向分布グラフ



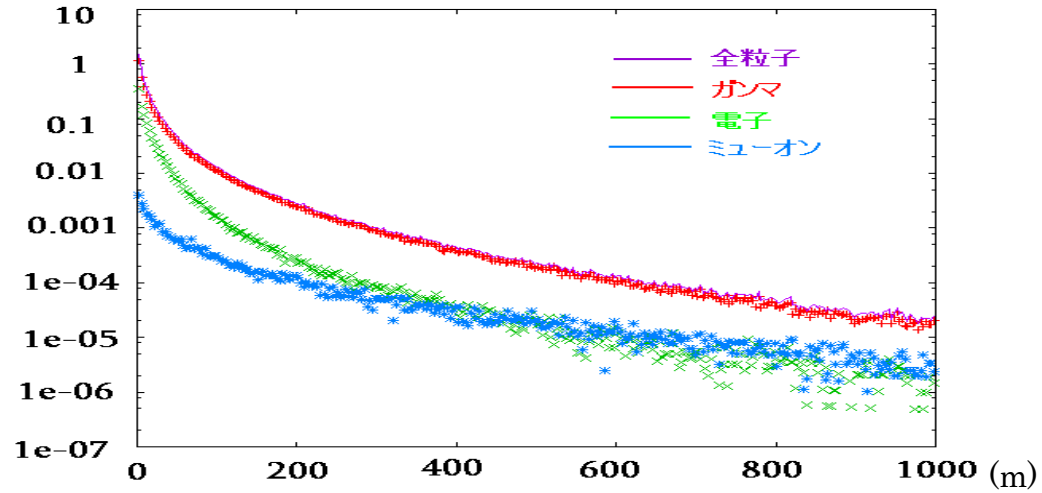
大気の深さと粒子数を表す縦方向分布グラフである。

4-3 横方向分布グラフ

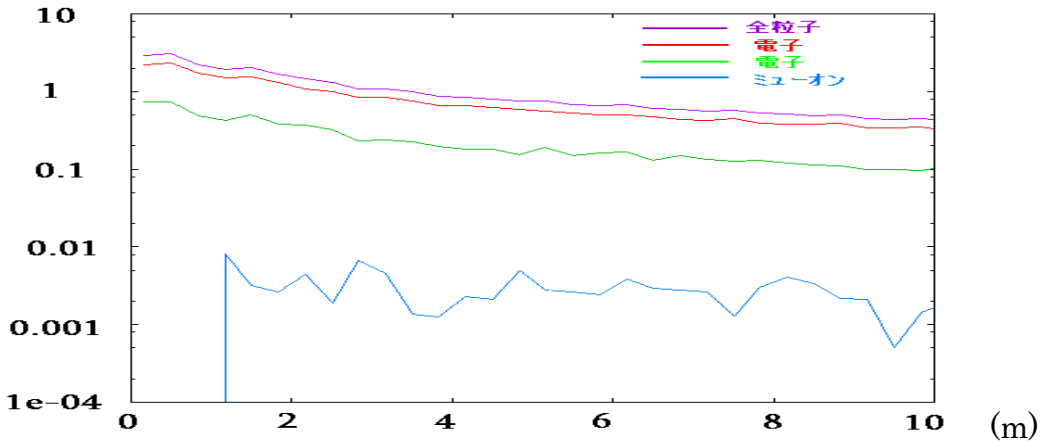
(粒子密度/m²)



(粒子密度/m²)



(粒子密度/m²)



4-4 実際の測定との比較

地表において空気シャワー粒子中の電子とミューオンの比を計算し、比較を行った。

空気シャワー中の各粒子数

μ：8.7*10³　　e：4.0*10⁴

(μ　+　e)　＝4.9*10⁴

よって（μ　+　e）中のμの割合は、18%

また、グラフより空気シャワー主軸中心付近では　約 2%

実際の測定から算出した割合は　約 8%　なので、測定では中心から近い位置の粒子を捉えたのではないかと考えられる。

5 結論

CORSIKA プログラムを用いて宇宙線の空気シャワーシミュレーションを行うことに成功した。

地表での各粒子の分布結果より、どの粒子も空気シャワー主軸に近いほど粒子数が多いと確認できた。

空気シャワー軸からの距離が大きくなるにつれ、各粒子密度は減少したが、ミューオンの減少率はガンマ・電子と比べて小さかった。よって、ミューオンは非常に広範囲にわたって分布しているといえる。

ミューオンと電子の比を計算した結果、ミューオンが（μ ＋ e)で占める割合は18%。中心付近では約 2%であるとわかった。

今後の進展

今回は 1 つの宇宙線、1 つのシンチレータの場合と仮定してシミュレーションを行ったため、実際の計測結果との比較は難しい。本研究を基礎としてより精度の高いシミュレーション結果が出る事を期待します。

6 実験に用いたプログラム

データ解析の際に用いたプログラムのソース・コードを以下に記載する。

6-1 文字変換のプログラム

```
program internal_file
   implicit none
   real*4 a
   character(4) s
   equivalence(a,s)

   s ='RUNH'
   print *, 's = "', s, '"'
   print *, "a =", a

   s ='EVTH'
   print *, 's = "', s, '"'
   print *, "a =", a

   s ='LONG'
   print *, 's = "', s, '"'
   print *, "a =", a

   s ='EVTE'
   print *, 's = "', s, '"'
   print *, "a =", a

   s ='RUNE'
   print *, 's = "', s, '"'
   print *, "a =", a
end program internal_file
```

6-2 データ解析のプログラム

```
#include <stdio.h>
#include <stdlib.h>

#define NUMPAT 10000000

struct PARTICL{
   int id, hadGen, level;
   float px, py, pz;
   float x, y;
   float t;
};
struct PARTICL Particl[NUMPAT];
int NumPart;
int main(void)
{
   int n;
   char buf[1024];
   double f0, f1, f2, f3, f4, f5, f6;
   int fflag;
   int itmp;

   fgets(buf, 1024, stdin);

   NumPart=0;
   fflag=0;
   while(1){

      for(n=0; n<39; n++){
         if( fgets(buf, 1024, stdin)==NULL){
           fflag=1;
           break;
         }
         if(sscanf(buf,"%le %le %le %le %le %le %le¥n",
                  &f0,&f1,&f2,&f3,&f4,&f5,&f6) != 7){
```

```
        fprintf(stderr,"error 1¥n");
        exit(1);
      }
      /**
      if(f1 == 1.0)
        printf("%d %s¥n", NumPart, buf);
      **/

      itmp = (int)f0;
      Particl[NumPart].id = itmp/1000;
      itmp = itmp - Particl[NumPart].id;
      Particl[NumPart].hadGen = itmp/10;
      Particl[NumPart].level = itmp-Particl[NumPart].hadGen;


      Particl[NumPart].px = (float)f1;
      Particl[NumPart].py = (float)f2;
      Particl[NumPart].pz = (float)f3;
      Particl[NumPart].x = (float)f4;
      Particl[NumPart].y = (float)f5;
      Particl[NumPart].t = (float)f6;
      NumPart++;

      if(NumPart>=291798){
        fflag=1; break;
        printf("%d %s¥n", NumPart, buf);
      }
    }
    if(fflag!=0) break;
}

/*****************************************************************/

//for(n=0; n<NumPart; n++){
for(n=0; n<NumPart; n++){
    /**
```

```
      if(n%39==0)
      printf("%f %f¥n",Particl[n].x, Particl[n].y);
      **/
      //if(Particl[n].id == 75 || Particl[n].id == 76)
      if(Particl[n].id == 75 || Particl[n].id == 76) continue;
         printf("%d %d %d %f %f¥n",n, n%39, Particl[n].id, Particl[n].x, Particl[n].y);
   }
}
```

6-3 CORSIKAread プログラム

```
C=============================================================
==========
C
C   c o r s i k a r e a d . f    (without THINNING)
C
================================================
C                        READ    AND    PRINT    CORSIKA    SHOWER    DATA
C
================================================
C                           f77 -fbounds-check corsikaread.f -o corsikaread
C                     gfortran -fbounds-check corsikaread.f -o corsikaread
C        Output format for particle output (blocklength = 22932+8 fixed)
C        each block consists of 21 subblocks of 273 words.
C-------------------------------------------------------------------
C        compilation:
C              gfortran -fbounds-check corsikaread.f -o corsikaread
C              f77 -fbounds-check -m32 corsikaread.f -o corsikaread
C              ifort -C corsikaread.f -o corsikaread
C-------------------------------------------------------------------
C        How to use this program:
C        1) Generate a file 'input' containing the path and name of the
C             DATnnnnnn file to be analyzed by this program.
```

```
C              The name should not contain leading blanks but filled up
C              by trailing blanks to get a total length of >70 characters.
C         2) Execute this program with the file 'input' as standard input:
C                        ./corsikaread <input >output
C         3) The file 'output' will contain a short overview of the
C              content of the DATnnnnnn file to be analyzed.
C         4) The file fort.7 will contain a detailed print out of the
C              content of DATnnnnnn.
C-------------------------------------------------------------------
C         J.Oehlschlaeger, D. Heck, 26 Apr 2013
C=================================================================
==========
          PROGRAM CORSIKAREAD
          CHARACTER CHV(5733)*4,CIDENT*4,CDAT*70,CBLK*70
          DIMENSION PDATA(5733)
          EQUIVALENCE (CHV(1),PDATA(1))
          COMMON /CHARS/CHV,CDAT,CBLK,CIDENT
          CBLK='                                                                '
          CDAT=CBLK
          IREC=0

C--READ FILE NAME-----------------------------------------------------
          READ(*,428,END=440,ERR=439) CDAT
  428     FORMAT(A)
  429     CONTINUE
          WRITE(*,430) CDAT
  430     FORMAT(1H ,'READ DATA FROM FILE = ',A)
          OPEN(UNIT=3,FILE=CDAT,STATUS='OLD',FORM='UNFORMATTED')
* - - - - - - read data records with 5733 words - - - -
    431 CONTINUE
          IREC = IREC + 1
          READ(UNIT=3,ERR=434,END=433) PDATA
          if ( mod(irec,100) .eq. 0 )
         +     WRITE(*,*)'               HAVE READ RECORD NR.',IREC
C-----------loop over subblocks---------------------------------------
          DO       LIA=1,5733,273
```

```
            CIDENT(1:1) = CHV(LIA)(1:1)
            CIDENT(2:2) = CHV(LIA)(2:2)
            CIDENT(3:3) = CHV(LIA)(3:3)
            CIDENT(4:4) = CHV(LIA)(4:4)
            IF (PDATA(LIA).GE.211284.0.AND.
     +          PDATA(LIA).LE.211286.0) THEN
               CIDENT = 'RUNH'
               WRITE(*,*)'RUNH'
            ENDIF
            IF (PDATA(LIA).GE.217432.0.AND.
     +          PDATA(LIA).LE.217434.0) THEN
               CIDENT = 'EVTH'
               WRITE(*,*)'EVTH'
            ENDIF
            IF (PDATA(LIA).GE. 52814.0.AND.
     +          PDATA(LIA).LE. 52816.0) THEN
               CIDENT = 'LONG'
               WRITE(*,*)'LONG'
            ENDIF
            IF (PDATA(LIA).GE.    3396.0.AND.
     +          PDATA(LIA).LE.    3398.0) THEN
               CIDENT = 'EVTE'
               WRITE(*,*)'EVTE'
            ENDIF
            IF (PDATA(LIA).GE.    3300.0.AND.
     +          PDATA(LIA).LE.    3302.0) THEN
               CIDENT = 'RUNE'
               WRITE(*,*)'RUNE'
            ENDIF
C-----------which kind of block is it?---------------------------------
            IF ( CIDENT.EQ.'RUNH' .OR. CIDENT.EQ.'RUNE' .OR.
     +           CIDENT.EQ.'LONG' .OR. CIDENT.EQ.'EVTH' .OR.
     +                                 CIDENT.EQ.'EVTE' ) THEN
               CHV(LIA) = CIDENT
               IF         ( CIDENT .EQ. 'RUNH' ) THEN
C----------------subblock run header-----------------------------------
```

```
                ! PDATA(LIA) = 11111111.
                DO       IL=LIA,LIA+272,7
                   WRITE(7,'(1P,7E13.5)') (PDATA(II+IL),II=0,6)
                ENDDO
             ELSEIF ( CIDENT .EQ. 'EVTH' ) THEN
C----------------subblock event header---------------------------------
                ! PDATA(LIA) = 33333333.
                DO       IL=LIA,LIA+272,7
                   WRITE(7,'(1P,7E13.5)') (PDATA(II+IL),II=0,6)
                ENDDO
C----------------subblock longitudinal data----------------------------
             ELSEIF ( CIDENT .EQ. 'LONG' ) THEN
                ! PDATA(LIA) = 55555555.
                DO       IL=LIA,LIA+272,7
                   WRITE(7,'(1P,7E13.5)') (PDATA(II+IL),II=0,6)
                ENDDO
C----------------subblock event end------------------------------------
             ELSEIF ( CIDENT .EQ. 'EVTE' ) THEN
                ! PDATA(LIA) = 77777777.
                DO       IL=LIA,LIA+272,7
                   WRITE(7,'(1P,7E13.5)') (PDATA(II+IL),II=0,6)
                ENDDO
C----------------subblock run end--------------------------------------
             ELSEIF ( CIDENT .EQ. 'RUNE' ) THEN
                ! PDATA(LIA) = 99999999.
                DO       IL=LIA,LIA+272,7
                   WRITE(7,'(1P,7E13.5)') (PDATA(II+IL),II=0,6)
                ENDDO
                GOTO 929
             ENDIF
          ELSE
C-----------subblock with particle data---------------------------------
              DO       IL=LIA,LIA+272,7
                 WRITE(7,'(1P,7E13.5)') (PDATA(II+IL),II=0,6)
              ENDDO
          ENDIF
```

```
      ENDDO
  929 CONTINUE
      GOTO 431

C--END OF TEST-------------------------------------------------------
  433 CONTINUE
      WRITE(*,*)'           LAST RECORD ',irec-1
      CLOSE(UNIT=3)
      STOP
  434 CONTINUE
      WRITE(*,*)'           READ ERROR ON UNIT 3'
*     CLOSE(UNIT=3)
      GOTO 431
  439 CONTINUE
      WRITE(*,*)'           READ ERROR ON STANDARD INPUT'
      GOTO 429
  440 CONTINUE
      WRITE(*,*)'           READ END ON STANDARD INPUT'
      STOP
      END
```

謝辞

本研究を進めるに当たり教授の方々には適切、丁寧なご指導ご協力を頂きました。プログラミングが苦手な私でしたが、梶野先生と山本先生の指導、そして同研究室の院生、学部生の協力により研究を進めることが出来ました。本当にありがとうございました。

参考文献

http://www.cfca.nao.ac.jp/~takhshhr/plasma_file/20130208Oshima.pdf#search='%E7%A9%BA%E6%B0%97%E3%82%B7%E3%83%A3%E3%83%AF%E3%83%BC'

http://corsair.ff.bg.ac.rs/CORSIKA_GUIDE69xx.pdf#search='CORSIKA+users+guide'