Computational Physics Lab

Analysis of Large Data Sets: NTuples

04/02/2009

Outline

1. Reading Ascii Data Files
   - Units
2. Analyzing Data with NTuples
3. Particle Data Booklet

Opening Input Files

$$\gamma p \longrightarrow N^* \longrightarrow p\pi^0\pi^0$$

```
21400.000     49.000   1900.610      2.000
  282.366    -59.729    294.642   1024.923
   -7.397   -243.317   1163.580   1196.409
 -274.982    303.030    442.388    617.550

21400.000   1810.000    936.093      0.000
 -115.682    120.330    762.354   1220.410
 -215.029    -81.591    112.839    289.561
  330.688    -38.740     60.900    364.394

21400.000   1812.000   1270.843      0.000
  309.611     24.443    885.796   1327.194
 -392.629    220.538    354.049    588.527
   83.016   -244.967     30.999    293.394
      ...
```

run number	event number	E_γ	(float)
proton P_x	P_y	P_z	E
pion[1] P_x	P_y	P_z	E
pion[2] P_x	P_y	P_z	E

File Streams

# include <iostream.h>
# include <fstream.h>

ifstream fin (“input.dat”);
ofstream fout (“output.dat”);

main() {

    float P_x, P_y, P_z, E
    ...

Some kind of loop

        fin $\gg P_x \gg P_y \gg P_z \gg E$;
        fout $\ll P_x \ll P_y \ll P_z \ll E$;
        ...

}

1. The disc file input.dat in the directory from which the program is beeing run is associated with a stream called fin.
2. In a similar way, the disc file output.dat is associated with a stream called fout.

File Streams

```
# include <iostream.h>
# include <fstream.h>

ifstream fin (“input.dat”);

main() {
    float $P_x$, $P_y$, $P_z$, $E$
    ...

for (int i=1; i<NEvents; i++)  {
        fin $\gg$ $P_x$ $\gg$ $P_y$ $\gg$ $P_z$ $\gg$ $E$;
        ...
}
```

1. for loop is convenient if the number of events is known.
2. Remember for Project 10: 4 lines (and 16 values) must be read in for each iteration.

File Streams

```
# include <iostream.h>
# include <fstream.h>

ifstream fin (“input.dat”);

main() {
    float P_x, P_y, P_z, E
    ...

while (! fin.eof())  {
        fin >> P_x >> P_y >> P_z >> E;
        ...
}
```

1. while loop is convenient if the number of events is NOT known.
2. Remember for Project 10: 4 lines (and 16 values) must be read in for each iteration.

Units

The electronvolt (symbol eV) is a unit of energy. It is the amount of energy equivalent to that gained by a single unbound electron when it is accelerated through an electrostatic potential difference of one volt, in vacuum. In other words, it is equal to one volt (1 volt = 1 joule per coulomb) multiplied by the (unsigned) charge of a single electron.

One electronvolt is a very small amount of energy:

1 eV = 1.602 176 53(14) $\cdot 10^{-19}$ J (or approximately 0.160 aJ)

The proton has a mass of 0.938 GeV/c^2, making a GeV/c^2 a very convenient unit of mass for particle physics:

- 1 eV/c^2 = $1.783 \cdot 10^{-36}$ kg
- 1 GeV/c^2 = $1.783 \cdot 10^{-27}$ kg

Outline

1. Reading Ascii Data Files
   - Units
2. Analyzing Data with NTuples
3. Particle Data Booklet

NTuples

Start ROOT!

In the ROOT shell, type:

- TFile file (“basic.root”);
  (Opens NTuple file)
- ntp1->Print();
  (Shows the tree structure: the number of entries, ...)
- ntp1->Scan();
  (Shows all values of the list)
- ntp1->Draw(“y:x”);
  (Draws *x* versus *y*)
- ntp1->Draw(“y:x” , ”x < 5.0 && x > 2.0”);
  (Draws *x* versus *y* for $2.0 \leq x \leq 5.0$)

Outline

Particle Data Booklet

Baryon Summary Tables
in the 2006 Review of Particle Physics

http://pdg.lbl.gov/2006/tables/contents_tables_baryons.html