

Informatica – Prova di laboratorio, 21 luglio 2022

INTERAZIONI FONDAMENTALI

Preparazione. Un *evento* è definito come la misura effettuata da un esperimento di collisioni tra particelle elementari. Ogni evento è definito dai quadrimpulsi (energia e componenti lungo i tre assi del vettore momento) delle due particelle in collisione:

$$p_1 = \begin{bmatrix} E_1 \\ p_{1,x} \\ p_{1,y} \\ p_{1,z} \end{bmatrix}, \quad p_2 = \begin{bmatrix} E_2 \\ p_{2,x} \\ p_{2,y} \\ p_{2,z} \end{bmatrix} \quad (1)$$

dove E_i corrisponde all'energia della particella $i \in \{1, 2\}$, mentre $p_{i,j}$ alla componente $j \in \{x, y, z\}$ dell'impulso della particella $i \in \{1, 2\}$. Naturalmente, vista la natura sperimentale degli eventi, questi possono essere classificati in due categorie: segnale effettivo o rumore di background.

Frequentemente, le collisioni tra particelle producono nuove particelle neutre che si disintegrano velocemente, come per esempio il bosone di Higgs oppure i bosoni elettrodeboli Z/γ^* . Queste particelle, siccome non hanno carica elettrica, non interagiscono con i detector degli esperimenti, quindi per determinare la loro presenza si studiano le particelle risultanti dal processo di disintegrazione.

La *massa invariante* è il primo osservabile rilevante in questo contesto, serve a definire il valore della massa delle particelle neutre e, per ogni coppia di particelle, si ricava come:

$$m_{\text{inv}} = \sqrt{(E_1 + E_2)^2 - [(p_{1,x} + p_{2,x})^2 + (p_{1,y} + p_{2,y})^2 + (p_{1,z} + p_{2,z})^2]}. \quad (2)$$

Ovviamente, trattandosi di una misura sperimentale i valori ottenuti saranno soggetti ad incertezze statistiche e quindi a fluttuazioni.

La *pseudo-rapidità* η è una seconda osservabile utile per caratterizzare la particella neutra, nel nostro caso è definita come:

$$\eta = \frac{1}{2} \log \left[\frac{(E_1 + E_2) + (p_{1,z} + p_{2,z})}{(E_1 + E_2) - (p_{1,z} + p_{2,z})} \right]. \quad (3)$$

Caricamento eventi. Sono stati raccolti un numero imprecisato di eventi misurati (vedi (1)) che contengono i quadrimpulsi delle particelle derivate dal processo di disintegrazione. Ognuno di questi eventi è già stato classificato e identificato come *segnale effettivo* o *rumore di background*.

Obiettivo. Per caratterizzare le particelle neutre, il software di analisi dovrà stimare il valore centrale e incertezze per la massa invariante e η definite nelle equazioni (2) e (3).

Il progetto *alla pagina seguente* chiede di realizzare un software che sia in grado di determinare le osservabili fisiche presentate nei paragrafi precedenti a partire da un dataset di eventi di tipo segnale effettivo e rumore di background.

Specifiche del progetto, leggete attentamente \Rightarrow

SPECIFICHE DEL PROGETTO

Il file `eventi.dat` sulla macchina `tolab.fisica.unimi.it` nella cartella `/home/comune/20220721_Dati/` contiene, riga per riga, un numero imprecisato di *eventi* derivati da collisioni di particelle. Ciascun evento è descritto dai quadrimpulsi (definiti all'equazione (1)) di due particelle e da un campo booleano **segnale** che vale 1 (cioè **true**) se l'evento corrisponde a un segnale effettivo oppure vale 0 (cioè **false**) se l'evento è rumore di background. Dunque, ogni riga del file `eventi.dat` contiene 9 dati di cui i primi 8 di tipo `double` e l'ultimo di tipo `bool` (1 oppure 0).

1. Caricare tutti gli eventi descritti nel file `eventi.dat` in un array di `evento` allocato dinamicamente e definito dalla struttura:

```
struct evento {
    double p1[4];    // energia-momento particella 1 (E1, p1x, p1y, p1z)
    double p2[4];    // energia-momento particella 2 (E2, p2x, p2y, p2z)
    bool segnale;    // Segnale se 1|true, background se 0|false
    double massainv; // Massa invariante
    double eta;      // Pseudo-rapidita'
};
```

I campi `massainv` e `eta` verranno riempiti in seguito. Stampare a video:

- (i) il numero di eventi letti,
 - (ii) il numero di eventi di tipo **segnale** e **background**,
 - (iii) le percentuali di eventi **segnale** e **background** rispetto al numero totale di eventi.
2. Implementare una funzione che calcola, per ogni evento, la massa invariante definita dalla formula all'equazione (2). Per ogni evento caricato al Punto 1:
 - (i) calcolare la massa invariante e aggiungere il campo `massainv`.
 - (ii) stampare a video le masse invarianti dei primi e degli ultimi 5 eventi.
 3. Ordinare l'array con gli eventi in ordine di massa invariante crescente e stampare a video:
 - (i) la descrizione completa dei primi e degli ultimi 4 eventi.
 - (ii) la *media* e *deviazione standard* ($1-\sigma$) per gli eventi di tipo **segnale**.
 4. Implementare una funzione che calcola la pseudo-rapidità definita nell'equazione (3). Per ogni evento caricato al Punto 1:
 - (i) calcolare la pseudo-rapidità e aggiornare il campo `eta`.
 - (ii) stampare a video la *media* e *deviazione standard* della pseudo-rapidità di tutti gli eventi di tipo **segnale** aventi massa invariante entro $1-\sigma$ del valore medio determinato al punto precedente.

ATTENZIONE! Tutti i risultati, oltre che stampati a video *con opportune diciture*, devono essere salvati in un file `risultati.dat` corredati dalle stesse diciture.

Istruzioni per la consegna del progetto \Rightarrow

ISTRUZIONI PER LA CONSEGNA DEL PROGETTO

Il vostro software deve essere predisposto in una cartella denominata `cognome_matricola` che deve essere copiata in `/home/comune/20220721_Risultati` sulla macchina `tolab.fisica.unimi.it`

Nella cartella `cognome_matricola` devono essere inclusi:

- un `makefile` che tramite i comandi `make compila` e `make esegui` consenta rispettivamente di compilare e di eseguire il programma,
- il file `eventi.dat` di input del progetto,
- il file `risultati.dat` prodotti dal programma,
- tutti e soli i `.C` `.cpp` `.cxx` e `.h` `.hpp` utili alla soluzione del problema.

Valutazione del progetto. *La valutazione terrà conto sia della qualità dei risultati sia della struttura e dell'organizzazione del codice; per chiarire, sono graditi uso di funzioni e compilazione separata, mentre non è gradito un `main` onnicomprensivo. I progetti che non compilano o che entrano in loop dopo il lancio verranno immediatamente classificati come insufficienti.*