

経過報告

田中 隆己

2012年1月26日

1 やった・やっていること

- RCNP での実験の proposal 提出
- HV クレート・モジュールのチェック
- DAQ 構築 (未完)
- anaroot で NEBULA の解析 (現在コードを変更中)

2 やること

- 今後、NEBULA に付くのか、HIME に尽力するのか。
- pac トラペ, Radiation Control
- タイミング計算
- 必要な回路具体数見積って RCNP と相談
- HIME 購入物整理
- 予算最終調整
- 2n 評価
- NEBULA の解析、anaroot
- 数値計算レポート

3 HV のチェック

HIME 用の HV として借りる理研で壊れていた HV64ch(SY403)x2 台に刺さっていたモジュールの全ての ch の出力をチェックした。具体的には、1ch ずつ SNEUT に接続し、HV を 1400V かけ、出力シグナルをオシロスコープで確認、また液晶パネルに表示される電圧と電流値のチェックを行った。

結果、HV クレート 2 台ともクレートに問題があるだけで、モジュールは全て正常な出力を示した。クレートの症状は以下の通り。

電源 ON → 電源ランプ,+5,+12,-12,+70,reset,interlock 等のランプは点灯。液晶画面は初めに横に縞模様？がでたあと、CAEN SY403 と出るべきところでも何も表示されない。その後もボタンを押しても無反応。また、HV ENABLE の on/off にかかわらず HV がかかっていることを示すモジュールの ch ごとのランプが全て点

灯。

一つヒューズが飛んでいるモジュールがあったが、新しいヒューズ (5A→3A に変更) をもらってつけてみたところ正常に動作。また、テープで塞がれていた ch もオシロで見るか限りは問題がなかった。



図 3.1 efficiency の energy 依存性

以上から、クレートのみ修理に出す予定である。

4 HIME のアッセンブリ

ジーテックから HIME のアッセンブリしたものが届いたが、梱包不十分で PMT が軒並み (31/36) 取れていた。返送して再度やってもらうことにした。

5 TIPS: 相対論的ヤコビアン

$\text{Li}(p,n)\text{Be}$ の断面積を計算するのに cm 系と lab 系の立体角の変換を行う必要がある。p の進む向きを z 軸とし、Li は静止しているとする。ローレンツ変換から、

$$p_n^{lab} \cos \theta^{lab} = -\gamma \beta E_n^{cm} + \gamma p_n^{cm} \cos \theta^{cm} \quad (5.1)$$

$$p_n^{lab} \sin \theta^{lab} = p_n^{cm} \sin \theta^{cm} \quad (5.2)$$

$$\text{ただし、} \beta = \frac{p_p}{E_p + m_{Li}}, \gamma = 1/\sqrt{1 - \beta^2} \quad (5.3)$$

これを整理すると、

$$\tan \theta^{lab} = \frac{\sin \theta^{cm}}{\gamma(1 - \beta/\beta_n^{cm})} \quad (5.4)$$

となる。前方では、

$$\theta^{lab} = \frac{\theta^{cm}}{\gamma(1 - \beta/\beta_n^{cm})} \quad (5.5)$$

が成り立つ。また、 θ に対して立体角は $2\pi(1 - \cos\theta)$ となり、前方では $2\pi\theta^2$ と近似できる。以上から、前方ではヤコビアンが、

$$\frac{\Omega^{cm}}{\Omega^{lab}} = (\gamma(1 - \beta/\beta_n^{cm}))^2 \quad (5.6)$$

と求まる。

ちなみに 250MeV では非相対論とほとんど違いがない (p と Li の質量差が大きいのので β が小さいことも要因)。

6 TIPS:mobaxterm

windows 用の mobile な xterm 風なもの (板橋さんが発見)。インストール不要で、native な terminal が立ち上がり ls,cp,grep,vi 等親しみ?のあるコマンドが軒並み使える。ssh ももちろんできて、一回ログインするとパスワードを勝手に覚えておいてくれる。さらに自動で sftp が走ったりとかなり造り込みが激しい。唯一日本語の入力 (utf-8 を使って表示はできる) ができないのが欠点だが、安定しており、かなり使えるソフトである。

7 TIPS:c++11

最近 c++ の新しいバージョン c++11(通称 c++0x) が正式に策定?されました。後方互換はあるそうです。wiki を見るとはっきり言ってほとんどわからないですが、vector など初期化リストが使える用になったり、関数オブジェクトをその場で生成 (ラムダ式) できるようになったりと、部分的にパツと使えるものがあります。メルセンヌツイスターも含まれるみたいです。あとは pthread とかも。でも、実際にはまだ gcc でも完全に対応していなかったり (4.5 以上必須、コンパイルには -std=c++0x とオプションが必要)、きっと原子核の業界ではまず使われないと思われるのであまり使う機会はないかもしれません。

8 TIPS:iterator と algorithm

iterator(反復子?) が定義されているクラスは STL の algorithm が適応可能である。オブジェクト指向うんぬんというよりは c++ のすごさが実感できる。簡単な例は for_each で、最初と最後の iterator の間にある要素に対して、任意の処理を適応できる。これだけ聞くと単に for 文を省略しただけの用に見えるが、for 文を使うためにはその構造を知らないと適切に回せない、つまり、他の関数に渡して処理をさせたりすることもできなければ構造の変更をすると for 文の中身も変更の必要が出る。しかし for_each による処理の場合は呼出側は中身の構造を知らなくていいので、どの関数 (実装を知ることができない関数等) からでも呼べる (開発者が知らなくていいというのとは意味が違う)。また、実装を知らないということは実装とは無関係であり、そこでクラスの独立性が保たれる。

オブジェクト指向とは、for 文の様に呼ぶ側が実装するのではなく、for_each の様に呼ばれる側が実装するというのが本質であり、理解されがたいところだと思う。

こういうよくある思想のお話は抜きにして、ソートや検索のアルゴリズムが全てのクラスに適応できることにオブジェクト指向の凄さを体感してほしい (僕が使いこなせているわけではないですが)。

http://www.geocities.jp/ky_webid/cpp/library/016.html

等。

ちなみにおもしろい例だと、

```
std::copy(v.begin(), v.end(), std::ostream_iterator(std::cout, ""));
```

で要素をまとめて cout できる。