

経過報告

田中 隆己

2012年9月6日

1 TOF 補正

解析するにあたって、対称としている粒子の TOF を位置情報などから予想して差引き、0 ns に peak がくるようにした時間情報は解析において便利である。例えば、slew correction や不要な event の cut、各種自動計算などに有効である。では、具体的に TOF の補正量がどうなるか示す。

1.1 gamma 線の場合

gamma 線の場合は beta がかならず 1 なので、Hit した位置と beta=1 を仮定して TOF を計算することで、gamma 線が 0 ns にくるようにできる。具体的には、

$$tcal = coef(tcal) * traw \quad (1.1)$$

$$flightlength = \sqrt{hitX^2 + hitY^2 + hitZ^2} \quad (1.2)$$

$$tcalflcor = tcal - \frac{flightlength}{(beta = 1) * lightvelocity} \quad (1.3)$$

と求められる。

1.2 単色中性子の場合

Li(p,n)Be(g.s.+0.43 MeV) によって生成される準単色 neutron の場合、energy は散乱角に依存する。これは純粋に kinematics で決まるので、event ごとに決まる lab 系の散乱角から neutron のエネルギーを求めれば良い。これを直接保存則から求めようとするとは難解な方程式がでてくるので、一旦 cm 系を介すことにより計算する。...

... 実際の解析において毎回この計算を行っているとは非常に計算時間を用するので、lab 系の散乱角と neutron のエネルギーの関係を多項式で fit して利用すると便利である。

2 SBT の cut

SBT で pileup しているイベントは TOF がずれることがわかった。これを図 2.1 に示す。SBT の Q がちょうど二倍程度になっているところでは TOF の mean が約 300ps ずれているため、 $qave=(sbt1+sbt)/2$ に対して (pileup の peak)- 2σ より上をカットした。gamma の場合は peak を中心に $\pm 2\sigma$ 以上を cut した。

Q の cut に加えて、SBT の T 情報が変な event は切った。ただし、SBT の TSum で切ってしまうと、こ

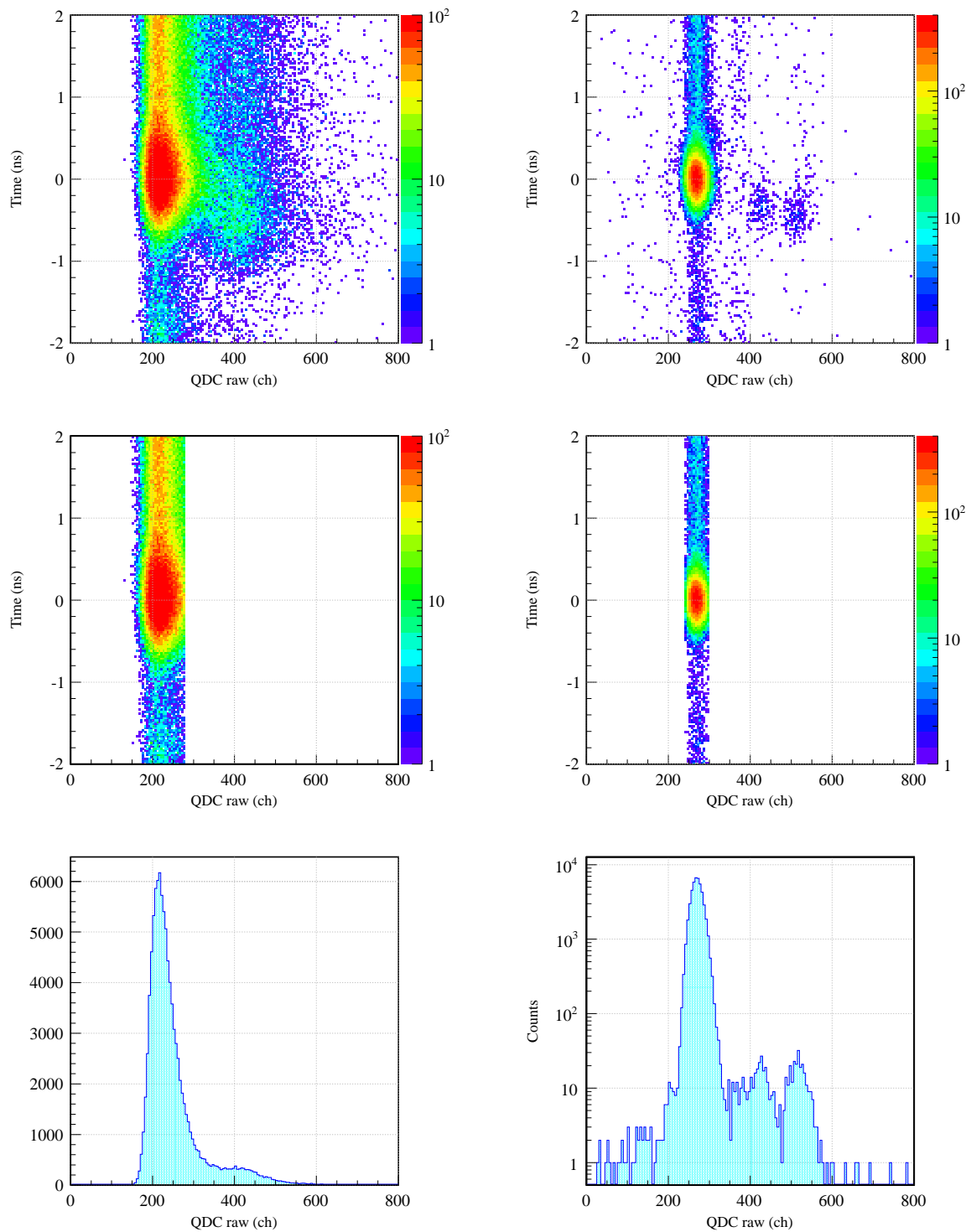


図 2.1 SBT の pileup による NEBULA の TOF の shift。(左列) proton 200 MeV の run。(右列) gamma 線の run。(上行) cut 前。(中行) cut 後。(下行) (-1,1) を x 軸に projection。SBT 自身では時間のシフトは見れないので、何かしらの TOF を見る必要があるが、特に neutron は beam の影響を受けにくいので NEBULA との TOF を見ることにより問題が発見しやすい。proton beam の場合は 400 ch 付近の peak に対して gaussian で fit し、QDC の ch が peak-2 σ より上は cut した。gamma 線の run の場合は peak を中心に $\pm 2\sigma$ を用いた。

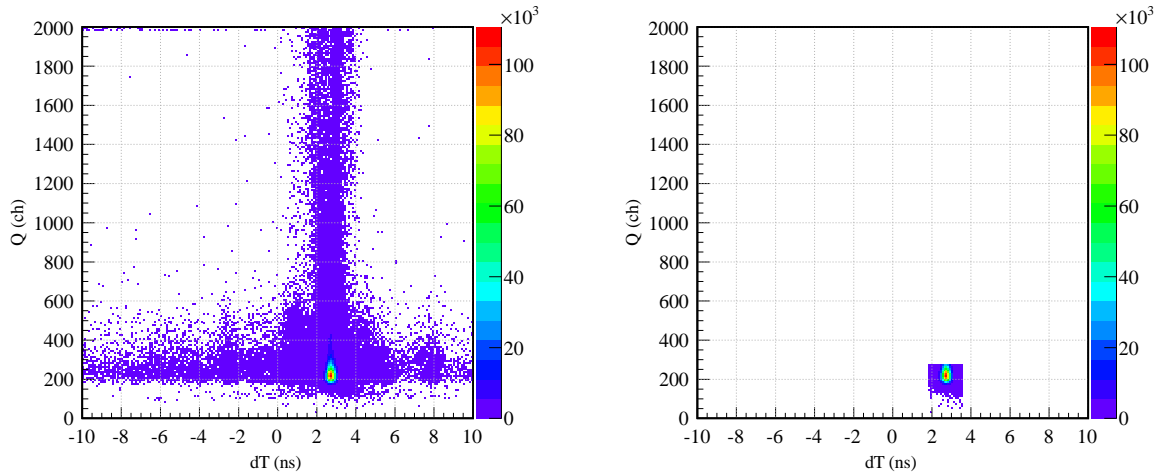


図 2.2 SBT の T と Q のカット。NEBULA trigger のときの SBT の $dT=sbt1-sbt2$ と $Q=(sbt1+sbt2)/2$ 。(左) cut 前。(右) cut 後。

れは下流に制限をかけることになるのでやらず、 $dT=sbt1-sbt2$ にのみ cut をかけた。この cut の様子を図 2.2 に示す。

以上の sbt の T と Q の cut をかけたときの DSB beam trigger の数は $2338357/2976674=78.56\%$ まで減少した。

SBT の Q のカットにより TOF の分解能はトータルで 1 割程度改善したため、NEBULA の intrinsic な時間分解能は大きく改善したことになる。

3 NEBULA 解析

4 t vs q spectrum

基本となる t vs q spectrum を図 4.1 に示す。

5 本物より早いイベント

単色 neutron よりも早く到達するイベントがあると、最速 HIT のアルゴリズムでは誤判定する可能性がある。そこで、あらかじめタイミングの早い event は切った上でアルゴリズムを適応する方が正しい結果が得られる。実際には、 $Li(p,n)Be$ の反応においてはターゲットからの gamma は少なく、また、ヨークからの gamma は neutron とコインシデンスしないので結果に違いはあらわれない。時間 cut する前の時間スペクトルは図 4.1 に示した。解析では neutron の peak から 2 ns 以上早い event を切った。

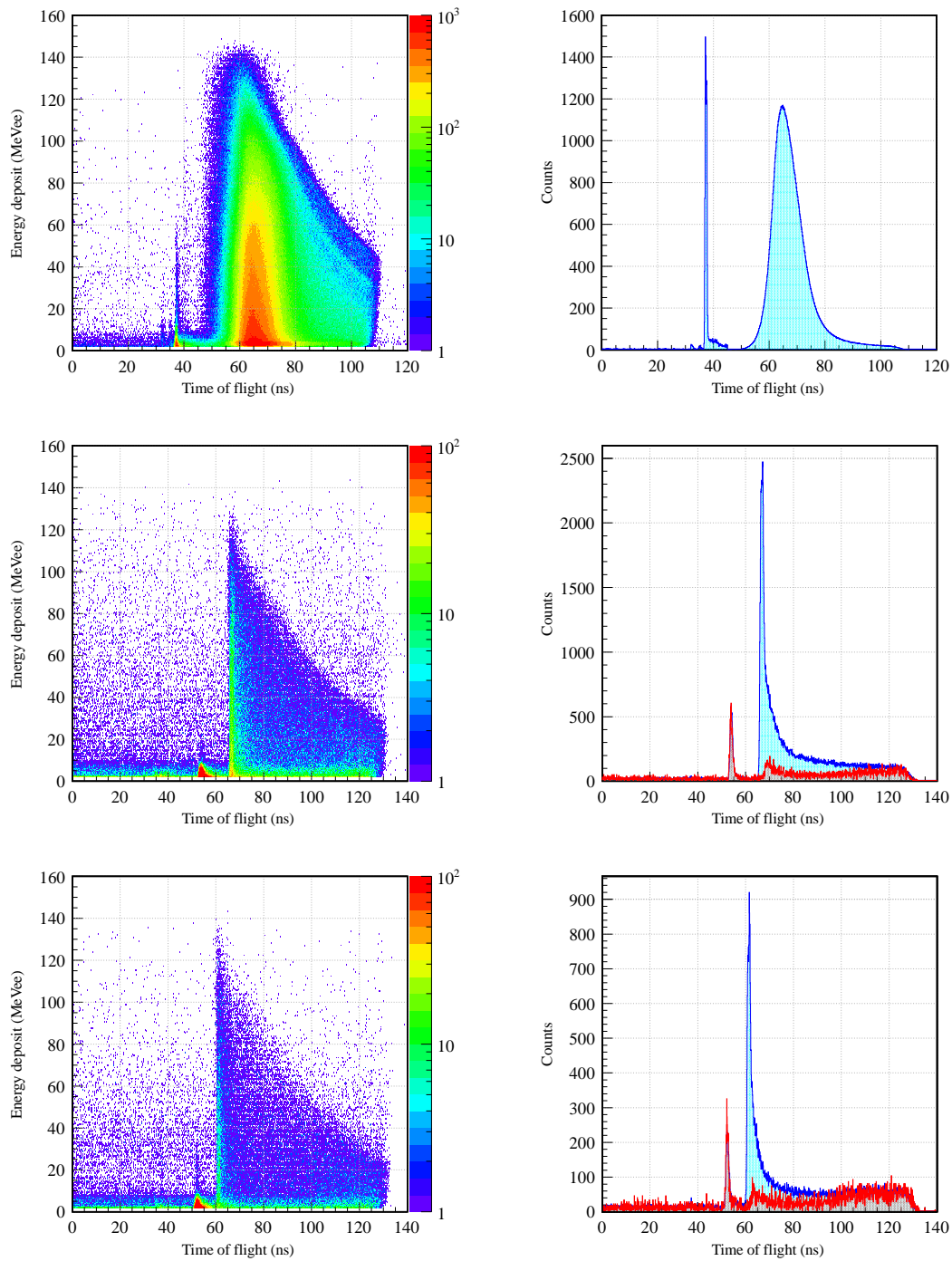


図 4.1 nebula の t と q の相関。(上) gamma の場合。(中) neutron 200 MeV の場合。(下) neutron 250 MeV の場合。2D は $\text{threshold}=2\text{MeVee}$, 1D は $\text{threshold}=6\text{MeVee}$ としている。他の条件は、 $\theta < 0.08$, 1st layer only, sbt t, dt, q cut, としている。gamma の 1D については 45 ns 以上を $1/50$ に scale している。slew 補正はしているが TOF 補正はしていない。neutron のスペクトルに見える 55 ns 付近の peak は proton が SAMURAI magnet の york に当たった際に発生した gamma 線と思われる。200 MeV のときの back ground の scale factor は 6.501。250 MeV のときは 5.813。

6 運動量とエネルギーの時間の関係

覚書。

$$\frac{dE}{E} = \beta^2 \frac{dp}{p} \quad (6.1)$$

$$\frac{dK}{K} = \left(1 + \frac{1}{\gamma}\right) \frac{dp}{p} \quad (6.2)$$

$$\frac{dK}{K} = -\gamma(1 + \gamma) \frac{dt}{t} \quad (6.3)$$

$$\frac{dp}{p} = -(1 + \beta^2 \gamma^2) \frac{dt}{t} \quad (6.4)$$

7 neutron のエネルギー

F7 から neutron のエネルギーを表 7.1 に求める。

	15C (240MeV)	p (200 MeV)	p (250 MeV)
F7 前	5.9895 Tm	2.1590 Tm	2.4490 Tm
F7 後	5.9479 Tm	2.1506 Tm	2.4423 Tm
TGT in	5.8832 Tm	2.1376 Tm	2.4320 Tm
TGT out	4.6806 Tm	2.1130 Tm	2.4125 Tm
F7 前	244.615 AMeV	201.501 MeV	253.002 MeV
F7 後	241.577 AMeV	200.074 MeV	251.765 MeV
TGT in	236.880 AMeV	197.872 MeV	249.869 MeV
TGT out	155.941 AMeV	193.731 MeV	246.292 MeV
Average	195.803 AMeV	195.802 MeV	248.081 MeV
neutron		194.158 MeV	246.437 MeV
F7 前	0.61030	0.56762	0.61604
F7 後	0.60768	0.56612	0.61500
TGT in	0.60349	0.56379	0.61338
TGT out	0.51584	0.55934	0.61030
Average	0.56307	0.56158	0.61189
neutron		0.55980	0.61042
TOF SBT-TGT	12.60 ns	13.49 ns	12.40 ns

表 7.1 エネルギーのまとめ。Average は TGT in と TGT out の平均。中村さんとかは物理的な TGT の中心でのエネルギーを使うらしい。

7.1 Target

Target 情報を表 7.2 にまとめる。

	Li	6Li	7Li	Cu
存在比		7.59%	92.41%	
原子量		6.015	7.016	
質量比		6.58%	93.42%	
密度	0.534 g/cm ³	0.0351 g/cm ³	0.499 g/cm ³	8.96 g/cm ³
厚み	20.40 mm	1.34 mm	19.06 mm	12.173 mm
厚み	1.089 g/cm ²	0.0717 g/cm ²	1.018 g/cm ²	10.907 g/cm ²

表 7.2 Target の情報。平成 23 年度の理科年表から。

7.2 back ground 差っ引きと beam 量, Live Time

back ground のスケールファクターを求めるには ungated DSB trigger の比を取れば良い。efficiency を出すのに必要になるので LiveTime を表 7.3 にまとめる。ということで、200 MeV のときの empty の差っ引き

	trigger	gated	ungated	ratio	SBT cut
200 MeV	neutron	4010851	4780093	83.9%	—
	DSB	2976674	3251921	91.5%	2338357
200 MeV emp	neutron	521025	625934	83.2%	—
	DSB	469261	510866	91.9%	359802
250 MeV	neutron	1950656	2507558	77.8%	—
	DSB	1172037	1346776	87.0%	701674
250 MeV emp	neutron	226793	262608	86.4%	—
	DSB	177247	190452	93.1%	120720

表 7.3 live time. gated は coincidence register から、ungated は scaler から。最後の行は SBT にゲートをかけたときの DSB の数。DSF はいずれも 2000

は、empty の hist に $(2338357/91.5 * 83.9)/(359802/91.9 * 83.2) = 6.582$ をかければよい。250 MeV のときは、 $(701674/87.0 * 77.8)/(120720/93.1 * 86.4) = 5.601$ となる。

back ground だが、peak の部分の S/N は 6% 程度であった。

7.3 分解能

今までとちがうのは、

- SBT での beam の選択 (pile up 除去, dT の cut($\pm 5\sigma$))
- theta < 40 mrad
- 6Li(4.4%) と target holder の寄与 (11.5%)、励起状態も含めた fit

を行ったこと。

7Li のゼロ度における断面積は cm 系で 26 mb/sr で、6Li の場合は約 14 mb/sr(論文の図から測定) である。7Li:6Li=92.41:7.59(理科年表) とのことなので、結局、4.4% が 6Li からの寄与となる。また、beam の広がり が 5mrad ほどあり、beam を SBV で定義しても一部 target holder に当たっていることがわかった ($\phi 25\text{mm}@\text{SBV}$, $\phi 30\text{mm}@\text{Target}$, SBV-Target 間隔 70cm, よって beam の像は $\phi 32\text{mm}@\text{Target}$)。BDC のトラックの精度は評価できていないが、beam の run を解析すると 8.1%(200 MeV), 4.9%(250 MeV) が holder に当たっていた。以上から、7Li から来るメインの gaussian と 6Li からくる gaussian、target holder にあたっていった成分からくる gaussian、さらに残りは半分 gaussian+ 半分 const を仮定した。これら四つの重ね合わせから peak 近傍の histogram は構成されていると考え、resolution は統一し、6Li の peak は相対論の計算から 7Li の peak より 3.432 MeV 小さいところにとりとした。また、target holder に当たった成分は Al2mm 厚のエネルギーロスである 1.93 MeV(196 MeV), 1.67 (248MeV) を仮定した。timing の resolution を求めるときも同様の fit を行ったが、6Li などの各 peak のエネルギーと時間の変換には非相対論の関係式 $dE/E=-2dt/t$ を使った。

時間分布とエネルギー分布の fit の様子を図 16.1 に示す。

実験的に得られた resolution の結果を表 15.1 にまとめる。ひとつまだ気になるのは、7Li と 6Li の ground

	200 MeV	250 MeV
thr=6 MeV	335(5) ps	300(7) ps
	2.59(4) MeV	3.49(7) MeV
500 ch to 2000 ch	309(4) ps*	272(5) ps*

表 7.4 NEBULA の分解能。plastic の厚みや beam の広がり、SBT の時間分解能などは込み込み。fit の仕方 (カイ二乗検定, likelihood) や fit の範囲によって resolution と efficiency とともに最大で 5% 程度変化する。resolution の評価において、fit の方法はカイ二乗検定、fit の範囲は $\pm 3\sigma$ に固定した。500 ch to 2000 ch に絞ると Hit する位置の不確かさが小さくなるために分解能が上がる。(*) バックグラウンド成分の立ち上がり時間が収束しなかったため、thr=6MeV の fit 結果である 0.878 ns(200 MeV), 0.780 ns(250 MeV) に固定した。

state 以外の成分を gaussian と const で無理やり fit したが、この peak の位置が状況によっては収束しなかったりするので、もう少し安定する方法を考える必要がある。

gamma 線の時間分布の fit の様子を図 7.2 に示す。

最後に、NEBULA 以外の成分の寄与などを考慮した neutron と gamma の分解能を表 7.5 に示す。

8 eff

考慮すべきこと。6Li, target holder, 見えない ch((n,pX) 以外の反応) による散乱 200 MeV, 250 MeV target, window, air, 隙間をもとにもどす補正する。角度

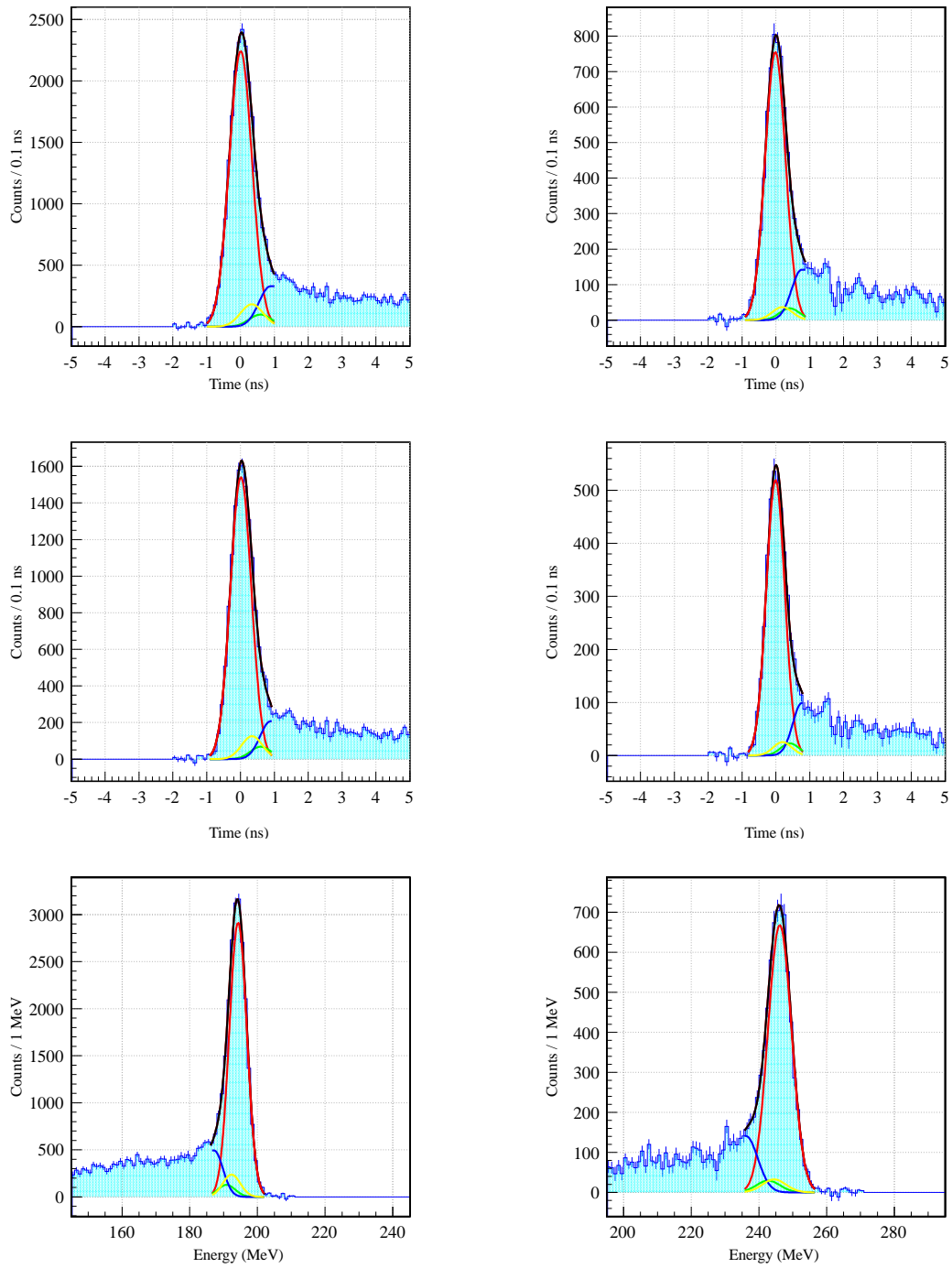


図 7.1 timing and energy resolution. 左の列が 200 MeV のときで、右の列が 250 MeV のとき。一段目は、時間分布で thr=6 MeV としたとき。二段目も時間分布だが、(500 ch, 2000 ch) を切り出したとき。この場合は 1 割ほど時間分解能が改善する。三段目はエネルギー分布で thr=6 MeV としたとき。結果は表 15.1, 表 7.5 にまとめて示す。fit の範囲は $\pm 3\sigma$ となるように再帰的に fit した。時間の場合 peak が 0 ns にくるように TOF を差っ引いている。エネルギーの場合はエネルギーの散乱角依存性を補正している。acceptance はいずれも 40 mrad に絞っている。測定位置などに対して乱数は振っていない。

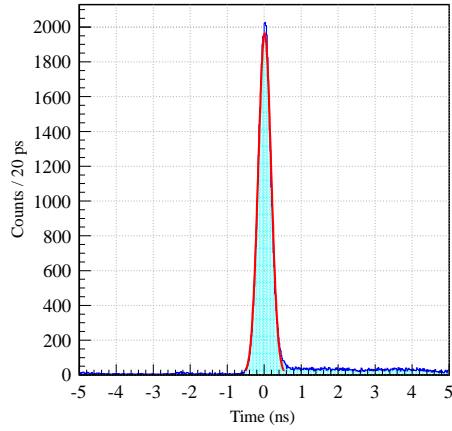


図 7.2 gamma 線に対する時間分解能。thr=6 MeV としている。theta の制限はかけていない (theta を前方等に絞っても resolution は変わらない)。fit の範囲は $\pm 3\sigma$ としている。結果は表 7.5 にまとめて示す。

	200 MeV	250 MeV	gamma
F3-F13	114 ps	85 ps	22 ps
SBT	84 ps	93 ps	25 ps
Target	151 ps	88 ps	—
Total	207 ps	154 ps	33 ps
TOF reso	335(5) ps	300(7) ps	174(1) ps
NEBULA only	263(6) ps	257(8) ps	171(1) ps
	2.03(5) MeV	2.99(9) MeV	—
Hit position	206 ps	189 ps	116 ps
NEBULA intrinsic	164(6) ps	175(8) ps	126(1) ps

表 7.5 NEBULA 以外の分解能とそれを差引いたときの NEBULA の分解能。Hit 位置の不確定性は単純に 12cm に均一にあたると仮定した。時間から Energy への変換は表 15.1 の測定結果を用いてスケールした

8.1 小角度の eff(40mrad 積分)

8.2 層ごとの eff

8.3 角度層ごとの eff

8.4 Li(p,n)Be の断面積

前方のみであれば約 26mb だが、それだと少し統計が少ないので、もう少し広い領域まで広げるために計算した (昔作ったコードを回すだけ)。ちなみに 6Li の断面積の角度依存性は 7Li と異なるはずだが、6Li と 7Li

の比は一定と仮定した。

9 角度分布

neutron, proton

10 cross talk

2枚目の veto を完全に on にしたときの cut off 1n event 数 ($\theta < 40$ mrad)

	all	1st layer only
1st veto	19482.5	10735.1
2nd veto	16506.7	8195.1
ratio	84.7%	76.3%

表 10.1 2nd veto の有り無しによる 1st hit の殺す割合

角度制限なしのときの 1n に対する diff cross talk。

角度制限なしのときの 1n に対する same cross talk。

	theta 全部	theta < 0.04
diff evapo	285/107687. = 0.26 %	65/19875. = 0.33%
戻り beta=1	153/107687. = 0.14 %	26/19875. = 0.13%
進み beta=1	101/107687. = 0.09 %	9/19875. = 0.05%
diff cross	7068/107687. = 6.56 %	1556/19875. = 7.83%
total	7614/107687. = 7.07 %	1656/19875. = 8.33%
diff proton	8807/107687. = 8.18 %	1832/19875. = 9.22%
dbeta/beta>1	123/107687.= 0.11 %	
same beta=1	262/107687. = 0.24 %	29/19875. = 0.15%
same cross	4533/107687. = 4.21 %	939/19875. = 4.72%
same proton	32550/107687. = 30.2 %	5966/19875. = 30.1%

表 10.2 cross talk の統計。beta=1 だが、低エネルギー deposit は gamma で、高エネルギー deposit はに宇宙線と思われる。宇宙線の場合は分母が 1n は意味が無くて、beam の数を分母とする必要があるが、アクセプトする時間はばに強く影響するので何とも評価しづらい (アクセプトする時間幅あたりにすればよいのか)。same は cluster 解析に強く依存する。ここでは時空間の距離 ($x, y, z, t^* \text{beta}=0.6$ の二乗和) が 24cm 以下の event を同一粒子由来とした。same は 2 層の和を表している。

進み gamma が少ないのは veto の掛け方の問題？エネルギーロスの大きい beta=1 は宇宙線と思われる。つまり、宇宙線を veto でうまく cut できていない。beta=1 を全て宇宙線とおもうと、分母は 1n ではなく beam でないと意味が無い。だけど、1st hit の energy(time でも同じ) の絞り方によってだいぶかわるので、どう定義するか微妙。

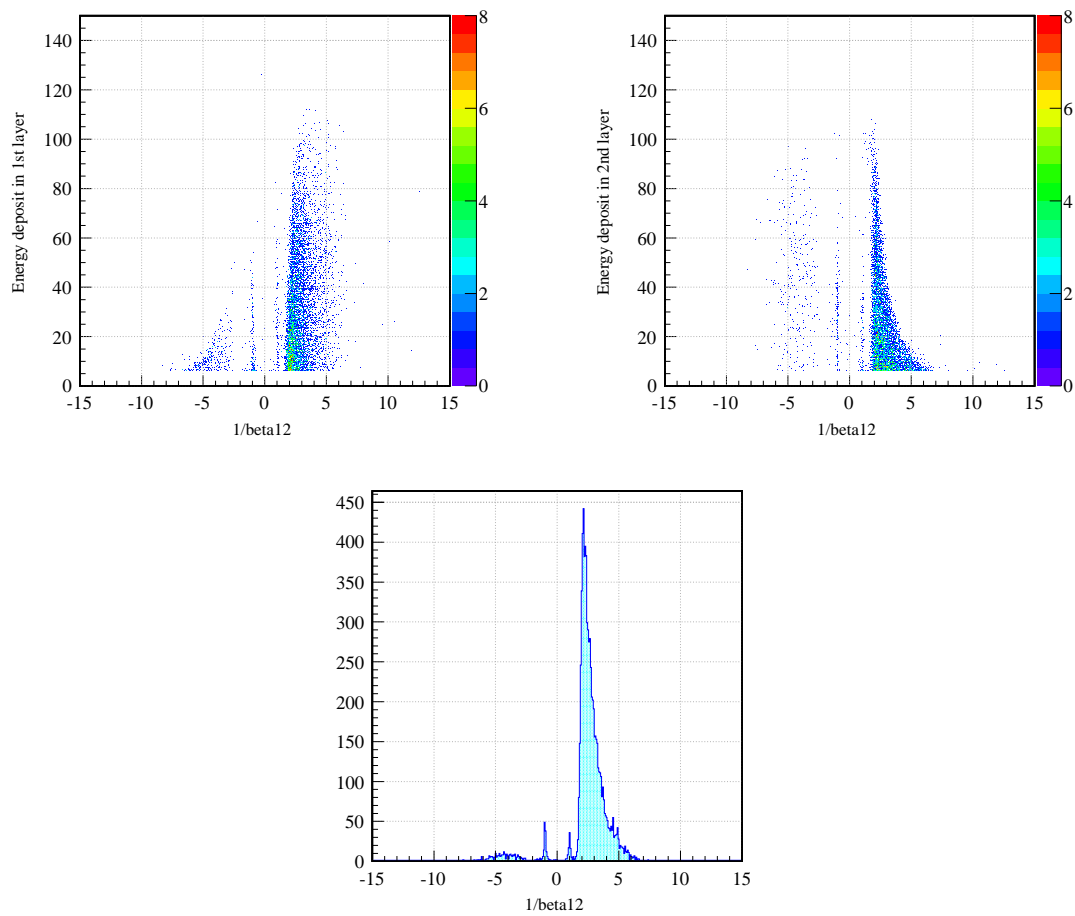


図 10.1 200 MeV のときの diff wall の beta12。最速 hit が 190 から 200MeV であることを要請している。 veto algorithm($r60cm$ & $1ns$ time6ns)

10.1 cross talk の erel

diff のときは erel が大きくなる。

11 Live Time の考察

efficiency の絶対値を出すには有効な beam 量を求める必要がある。有効な beam とはディテクターや Software で定義した beam の内、DAQ が Data 取得可能なときに来た beam の数のこと (trigger とは本質的には無関係)。もし、beam の時間構造が無視できると、

$$\text{有効な beam} = \text{beam} * \text{LiveTime}$$

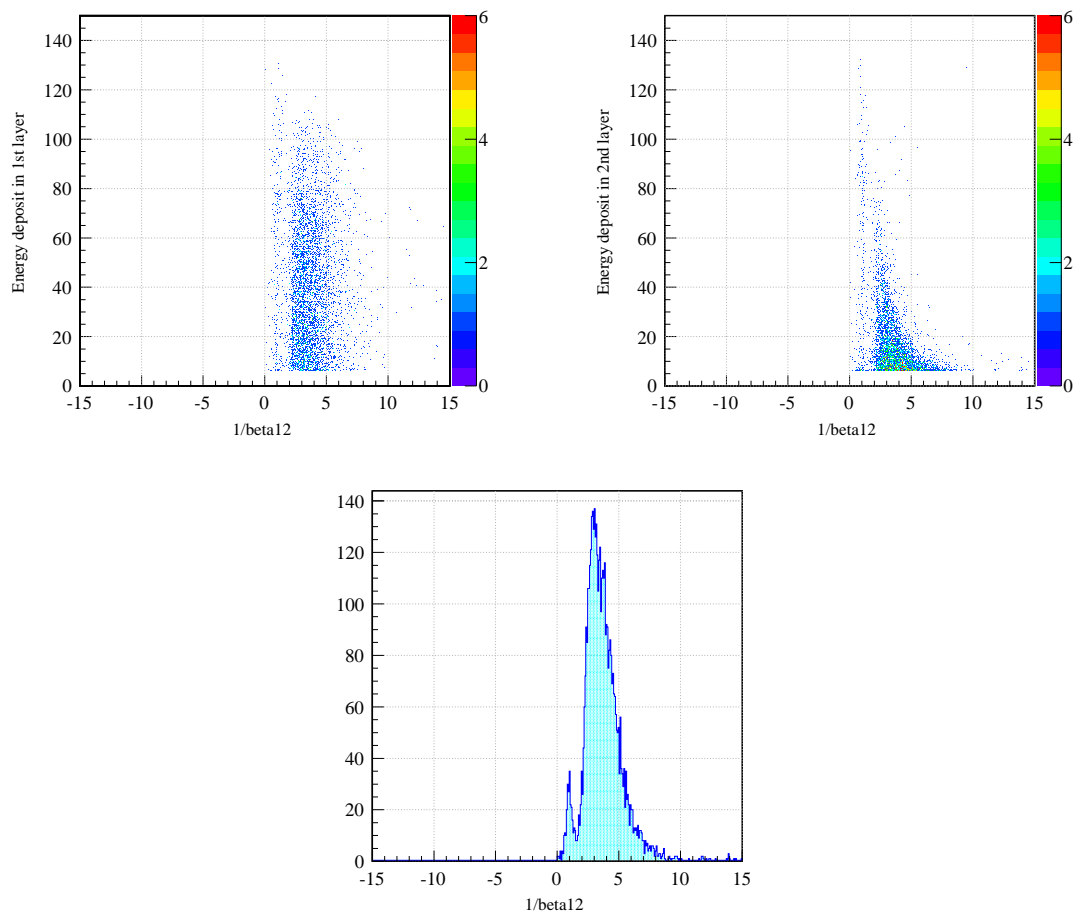


図 10.2 200 MeV のときの same wall の beta12。最速 hit が 190 から 200MeVであることを要請している。veto algorithm($r60cm$ & $1nstime6ns$)。cluster 解析は時空間の距離 ($x,y,z,t*\beta=0.6$ の二乗和) が 24cm 以下の event を同一粒子由来とした。

で計算できることになる。さらに、beam と trigger の時間構造が一致すれば、

$$LiveTime = \frac{\text{有効な beam}}{\text{beam}} = \frac{\text{gatedtrigger}}{\text{trigger}}$$

という関係式が成り立つ。注意すべきは各等号の成り立つ条件で、一つ目の等号は「時間構造が無視できる」こと、二つ目の等号は「beam と trigger の時間構造が一致する」ことであり、微妙に異なる。

先に時間構造が無視できない状況の極端な例にあげると、DAQ のデッドタイムが例えば 10 ns とすると、これはサイクロトロン の RF よりも短いので全ての beam は有効となる。つまり、

$$LiveTime \neq \frac{\text{有効な beam}}{\text{beam}} = 1$$

$$LiveTime \neq \frac{\text{gatedtrigger}}{\text{trigger}} = 1$$

となる。

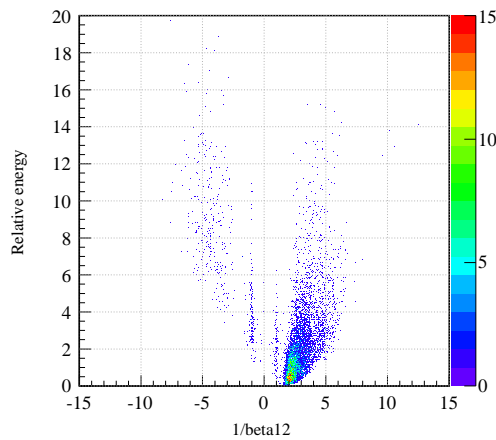


図 10.3 beta12 に対する 2n 間の relative energy. $1/\beta_{12} \sim 5$ 付近の event が evaporation で、 $erel$ は 4 以上となっている。

beam の時間構造が無視できないとすると複雑過ぎてどうしようもないので、beam の時間構造は無視できると仮定すると、時間構造のない trigger の gated-trigger/trigger を使えば良いということになる。つまり、NEBULA の解析をしているから neutron trigger の Live Time を使うのではなく、単に neutron trigger は random な trigger とみなせるから accept された beam の数を計算するのに使ってよいのである。一方、DSB trigger は時間構造が無視できないので、DSB の Live Time は使ってはいけないという結論になる。DSB-trigger は beam を例えば 1000 個ごとにアクセプトしているのでその間隔の期待値は $\sim 1 \times 10^5 / 1000 = \sim 1 \times 10^2 \text{cps} = 10 \text{ms}$ となり、明らかに DAQ の dead time (typ. 300 us) が終わった後に次の trigger がくる。つまり例えば dead time よりも長い周期の DSB-trigger のみで取っている場合は、理想的には

$$\frac{\text{gatedDSBtrigger}}{\text{DSBtrigger}} = 1$$

となる。ただし、実際には beam が random に来ていることと、DSB-trigger 以外の trigger によって random に DAQ が死ぬので、

$$\text{LiveTime} \sim \frac{\text{gatedDSBtrigger}}{\text{DSBtrigger}}$$

となる。

さらにもう一步、SBT にゲートをかけた場合にどう考えるかだが、SBT で発光量にゲートをかけてパイルアップを切ったとすると、これは beam の時間構造をいじっていることに相当する。つまり、十分 random にくる beam の内、時間の近い beam はなかったことにしている。この操作によって beam の random 性は失われない(と思われる)ので、やはり random な trigger によって求めた Live Time を使うことに間違いはない。

12 Hime 実験 memo

- SBT のパイルアップを取り除くときに、Q で切ることを考えると、QDC のゲートは 1 バンチきっちりのみを取るように調整した方がイベントの損がない(というか NEBULA の場合はパイルアップで 2

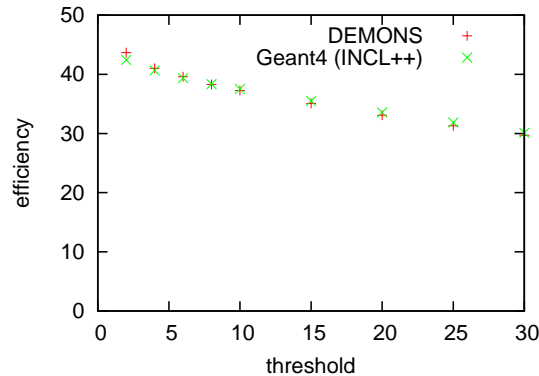


図 13.1 nebula の efficiency の評価。DEMONS と INCL++ の比較。シミュレーション条件は基本的に一緒にした ($E_{in}=250$ MeV)。

割ぐらい損している)。RCNP では beam が非常に細いこともあり、さらに window disceri を入れればハードできっちりとパイルアップを切れると思われる。

- 真の event 以外にも遅い event が大量にあるので、トリガーのタイミングを攻めることでデッドタイムを減らせる。
- beam 強度が弱いのでおそらく加速器側での微調はつらいと思われる。なので、初めは 1/7 の pulser を入れておいて、強度の調整は pulser でやれば正確にできるかもしれない。

13 Simulation

efficiency の simulation を開始した。Geant4 の version が 9.5 になったときに中エネルギーの原子核の反応を記述する INCL という物理 code が INCL++ という名前で移植された。これはまだ実験段階だそうだが、試しに NEBULA の efficiency を DEMONS と比べたところ、非常に良い一致が得られた (図 13.1) いままでは中エネルギー領域向けの物理クラスは binary cascade、もしくは bertini cascade というモデルが主流だったようだが、これに変わる物理クラスとして有望視されているそう。DEMONS と比べると非常によく一致している。INCL の適応範囲が 200 MeV と言われているので、それ以下の入射エネルギーでは合わないかもしれない。また、散乱された neutron の反応も正確に追う必要がそれも微妙かもしれない。Geant4 の Rederence Manual に INCL とは何か細かく書いてあったがまだ追えていない。とりあえずこの INCL で学会はしのぐつもり。(注: この結果を出したときの Geant4 で simulation の inelastic の全断面積はおそらく DEMONS に近いものをつかっており、より正しいと思われる断面積クラスを用いたときはこれより 1 割ほど小さくなる。)

France グループだとも思うが、MENATE_R という従来の様に個別の反応 ch を考慮した物理クラスもあって、これも試してみたが、何かと不可解なところがあり、発表するには微妙過ぎる。

14 cross section の比較

cross section の比較を行った。

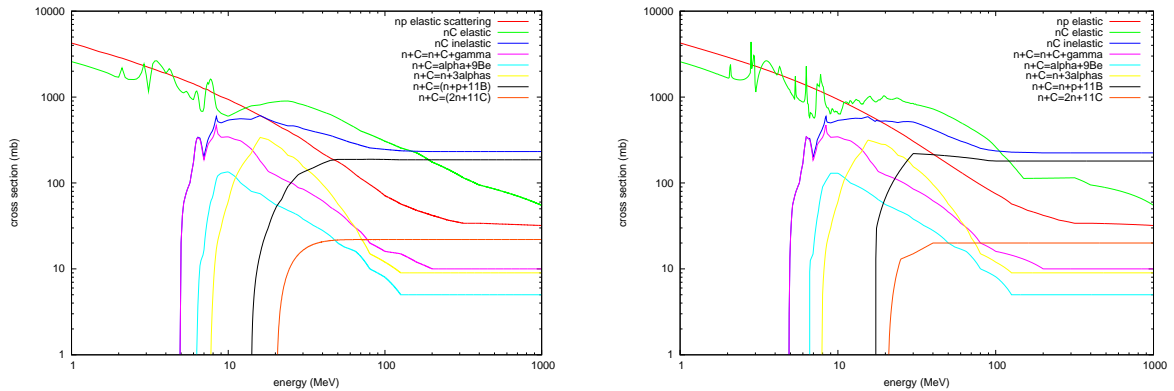


図 14.1 DEMONS と MENATE.R での cross section。MENATE.R では $^{12}\text{C}(n,p)^{12}\text{B}$ の断面積が非常に大きく、データベースや実験とあわなかったため除外した。

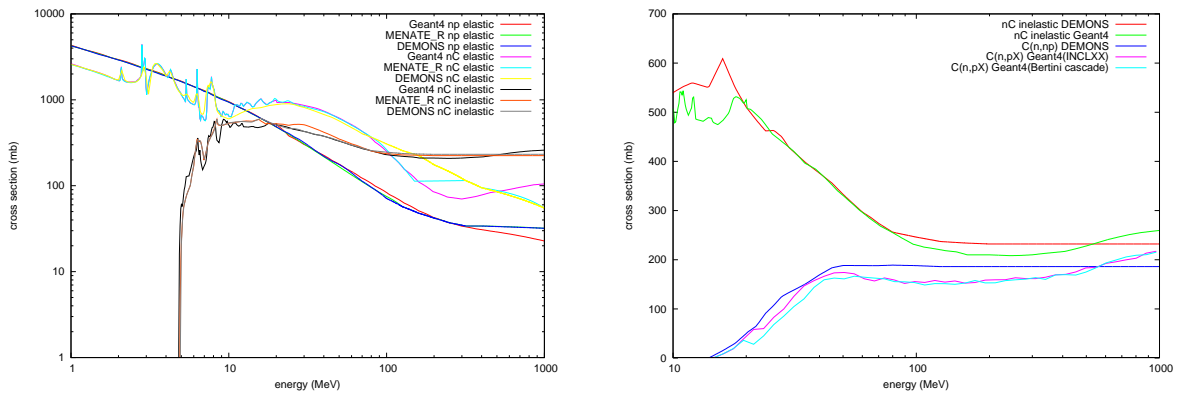


図 14.2 (左) elastic と inelastic の全断面積の比較。Geant4 で使用している断面積はおもに ENDF と JENDL が公開している断面積を使用している。(右) DEMONS と Geant4 の cross section をより細かく比較したグラフ。Geant4 では 200MeV において inelastic の全断面積が小さくなることにより efficiency が下がると言える。

中エネルギー領域の実験データがそもそもないので、どこまで合っているかをつめきめることは難しい。

15 efficiency の条件依存性

intrinsic な efficiency を考える場合は、NEBULA の隙間 (1.7%) も考慮する必要がある。Target, Window, Air (略して TWA) によって、3.5%(inclxx) ほど flux を損しているので、実験で求めた efficiency が 32% とすると、intrinsic な efficiency は 33.7% となる。

さらに、ナマることにより切られる成分は 5% ほどあり、TWA と合わせて 8.4% にのぼる。DEMONS などと比較するためにナマリ, TWA, モジュール間の隙間, モジュールの体積, a を補正したものを efficiency とすると、32% が 35.6% が増える。threshold=6MeV の場合の DEMONS の結果は 41.1% であり、13% ほど小さい結果となる。

この違いは主に inelastic の全断面積の違いで説明できる。200MeV 付近での inelastic の全断面積を

200 MeV	inclxx	bert
V	37.6 %	34.4
TWA	36.4 %	33.3
V+reso cut	35.9 %	32.6
TWA+reso cut	34.7 %	31.4
V	37.3 %	34.1
TWA	36.1 %	33.0
V+reso cut	35.6 %	32.3
TWA+reso cut	34.4 %	31.1

表 15.1 efficiency のモデル依存性と target, window, air (略して TWA) のありなし、resolution がわるいところの cut off ありなし。二段目は volume が 120*120 に比べて実際にきたモジュールは id1-30 についての平均が 120.35*118.68 ぐらいで、0.8% ほど volume が少ないので、これを補正した。

DEMONS と最近のデータベースで比較すると、209mb/232mb=90.1% と約 1 割小さいことになっている ((n,pX) を比較すると Geant4 の方が 15% ほど小さい)。

これより実験結果が DEMONS の結果に比べて 13% 小さくなったのはあらかじめ間違いではないと思われる。

16 TOF の energy deposit 依存性

slew 以外にも、z 方向の当たる位置の偏りによって TOF の energy deposit 依存性が生じる。これを正確に議論すれば分解能の改善が見込まれる。

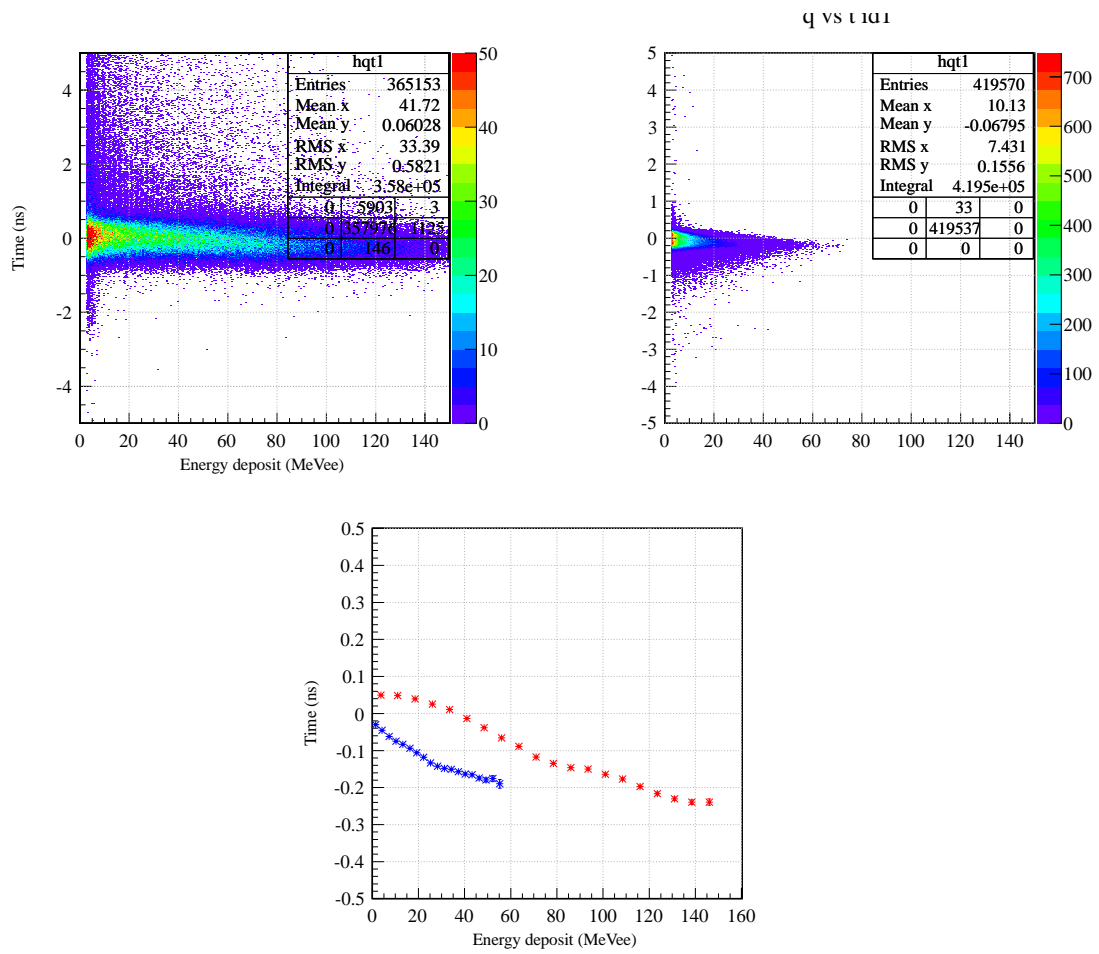


図 16.1 TOF の energy deposit 依存性。Hit した位置によって energy deposit に制限が生じるためにこのような相関が見られる。実験に反映するには energy deposit のある程度正確なキャリブレーションが必要となる。

17 Geant4 まわり

17.1 解析マシンに Geant4 を入れました

解析マシンに最新 (4.9.5.p01) の Geant4 を入れました。ここ最近でインストール方法に大きな変更があったけど本質的な違いはない模様。マイナーな関数の引数とか変わっていたりするので、昔のコードを使うには一部手を入れる必要があるかも。Install のメモは web に上げました。

17.2 nebula(というか neutron detector) のシミュレータの開発

neutron detector のシミュレーションの土台となる部分が大体できた気がします。シミュレータは二段階に分かれていて、

1. beam 入射から素情報取得まで: geant4 の user src。
2. 素情報から実験情報生成・解析: ROOT でちょろちょろっと

となっている。geant4 の部分はそれなりに汎用的・ポリシーを持って書きました (次の章参照)。そのまま HIME, NEBULA, LNEUT, どれでも src を変更すること無くシミュレーションできます (LNEUT と SNEUT の複合とかはいじる必要あり)。VETO の段差やモジュールの間隔も等間隔だけど再現できます。

シミュレーションで得られる素情報 (各ステップごとのエネルギーロスなど) は ROOT を使って回収します。また、これを格納する入れ物は別途のライブラリで定義します。入れ物の種類は 3 種類で、beam 情報、scintillator 中の情報、detector の parameter 情報です。これをまとめて ROOT file に入れておくことで一つのファイルで parameter 情報まで入ったシミュレーション結果が得られます。

ここでひとつ、入れ物を定義するということは、この定義を変更するとシミュレーションデータの互換性がなくなるように思われるかもしれないが、ROOT の機能で既に定義したクラスのメンバ変数を追加しても後方互換性があります。逆もおそらく可です。メンバ関数はそもそも情報として保存していないのでどのように変更しても互換性があります。互換性が明確になる様にライブラリとシミュレータのメジャー version で互換性を表すように (たぶん) します。

素情報から実験情報を生成したり、いろいろ解析する部分はシミュレーションに強く依るし、個人の趣味があるので仕様とかは決めてません。

ちなみに、physics list を少しいじれるようになりました。この neutron detector のシミュレータでは無理やり感がありますが global time(要は TOF) を 200 ns に制限したりしてます。きっと

18 Geant4 での visualize

磯部さんも使っていたが、visualize、及び UI に Qt(キョート) を使える様になった。ただ、やれることは増えてないしリモートだと無駄に遅くなるので微妙。一方、vrml の viewer のいいのをずっと探していたが、やっといいのが見つかった。このページ (<http://cic.nist.gov/vrml/vbdetect.html>) に綺麗にまとまっていて、その中の view3dscene が free で且つ linux 可、使い勝手も良好で見た目の good だった。解析 PC に入れてあるが、やはり 3D を X で飛ばすのは辛いので、生成される.wrl を local にコピーしてそれを local で見る方法をおすすめする (/usr/bin/remote-view3dscene というスクリプトを置きました)。

19 Geant4 によるシミュレーションコードの開発仕様

こんなことを意識して開発するときっと幸せになれるという妄想。以下を念頭にしてコードを書いたので、その思想を何となく引き継いで他のシミュレーションをしてほしい。

19.1 命名規則

ROOT には厳格な命名規則があり、Geant4 には何となく命名規則しかない。Geant4 によるシミュレーションを確立するにあたって命名規則を ROOT の規則に統一する。詳しくは <http://root.cern.ch/drupal/content/c-coding-conventions> を参照。といっても、完全に準拠するのは大変なので、以下を大体守ればよい。

- クラスの名前は T 始まり (url 中 Mixin は気にしなくてよい)。ただし、Geant4 のクラスを継承した場合は特につけない。TContainerView, EventAction
- メンバ変数は f はじまり。fViewLits
- グローバル関数内の static な変数とグローバル変数は g 始まり。gDeviceList
- local 変数は小文字始まり。seed, theCurrentArea
- メンバ関数は大文字始まり。DrawSelf()
- 複数単語になる名前はアンダーバーを使わずに大文字を使う。

一応規則として以下も書いておきますが、わかる人は以下にも従ってください。

- Enum の空間の名前は E 始まり。EFreezeLevel
- Enum や定数 (static const 指定された変数のこと。メンバ変数含む) は k 始まり。kMenuCommand
- static なメンバ変数は fg 始まり。fgTokenClient
- typedef した場合は末尾に_t をつける。Int_t
- struct は末尾に_t をつける。SimpleStructure_t

以上に加えて、ROOT 標準ではないが多分一般的に思われていること。

- 変数の名前は省略せずそれを読んで意味がわかるようにする (eloss ではなく energyLoss とか)。特にオブジェクトの場合は基本的に接頭辞を除いたクラスの名前をそのまま使う (TContainerView なら containerView)。
- 変数名がかぶったときは、1,2,... としたくなるなら vector などを使う。そうでない場合は意味のある単語 (x,y,z とかでも) を付け加える。

19.2 ROOT と Geant のどちらのクラスを使うか

ROOT のクラスは基本的にデータ回収用のみ使う。例えばベクトルの計算は G4ThreeVector を使い、それを格納する変数のみに TVector3 を使う。といっても複雑な beam の初期化等で ROOT のクラスを使いなくなるはずで、そういうときは混在はできるだけ避けて、ROOT のクラスを使ってまとめて書くようにする。

19.3 仕様

仕様と言えるほどちゃんとしていないが、ある程度設計思想的なものを統一したい。

- 開発者・ユーザーは中村研で閉じることを想定する。
- ユーザーがブラックボックスとしてとりあえず使えるようにする。
- 開発者にはある程度のレベルを要求する。つまり初心者用に冗長に書かない。
- 書き方・ファイル構造・クラス設計等は Geant4 の example や src を真似る (Pieter 反対!)
- Geant4 で計算するのは素情報までとし、これを専用のクラスに格納した上で tree に保存する。
- Data と Geant4 で使用した Parameter は全て単一の ROOT file に保存する。

19.4 意識してほしいテクニカルなこと

かなり具体的なこと。上の区分に入らない注意点ともいえる。

- warning は無視しない。
- 配列よりも vector を使う (vector を使うとほぼ配列と互換性のある書き方ができます)。
- segmentation fault はデバッガで追えるのでいちいち Debug 用コードを書かない。例えば無駄にポインタがゼロでないかのチェックをするとか。
- 残す必要のある DEBUG コードは `#ifdef DEBUG...#endif` で囲む。
- `#DEFINE` ではなく `static const` を使うこと。
- 行ごとにコメントを書くなどのコメントの乱用はしない。できるだけ変数名やコードの書き方で意図を伝える。
- コードの可読性は重視する。関数化したりコメント残したりしないで `memory` を直接いじるとかシフト演算するとかは死ぬ。
- その .cc 用の header file の `include` は一番最初を書く。
- `using namespace` を header file に書かない。
- pointer の宣言は `double* x;` と、アスタリスクを型側にくっつける。