

EXP1803 - bugs #164

Выполнение условий для распада в PhaseSpace .

03/07/2018 08:11 PM - Mikhail Kozlov

Status:	Закрыта	Start date:	03/07/2018
Priority:	Неотложный	Due date:	03/09/2018
Assignee:	Ivan Muzalevsky	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:			
Description			
Метод TGenPhaseSpace::SetDecay() возвращает false, если суммарная энергия He6+H2 в ЦМ меньше суммы масс H5 и He3. В нашем случае это происходит при увеличении массы H5.			
<small>Я пока что не уверен, что это баг, т.к. при увеличении энергии пучка все срабатывает. Нужно с точки зрения физики проверить с конкретными значениями.</small>			
<small>Но я не совсем понимаю, почему мы не бустим векторы в систему ЦМ, а считаем, что за нас это делает PhaseSpace. И в каких случаях это нужно делать самому. Прошу, объяснить, пожалуйста.</small>			
Related issues:			
Related to Developing #143: Спектр возбуждений 5H		Закрыта	02/18/2018 02/23/2018

History

#1 - 03/07/2018 08:11 PM - Mikhail Kozlov

- Related to Developing #143: Спектр возбуждений 5H added

#2 - 03/08/2018 04:54 PM - Ivan Muzalevsky

Проблему понял, TGetPhaseSpace действительно ломается при этом условии, что описано в его методах:

```
fTeCmTm = P.Mag(); // total energy in C.M. minus the sum of the masses
for (n=0;nfNt;n++) {
    fMass[n] = mass[n];
    fTeCmTm -= mass[n];
}

if (fTeCmTm < 0) return kFALSE; // not enough energy for this decay
```

Можно решить проблему по аналогии с тем, как это было устроено в SIMONE: при условии, что продуктов реакции 2 кинематика рассчитывается одним способом, при условии что 3/5... другим.

В связи с этим, скорее всего, нужно немного переделать механизм разыгрывания кинематики (создания 4-векторов) продуктов реакции.

```
Double_t mass5H = (fIs5HUserMassSet) ? f5HMass : f5H->Mass();
LOG(DEBUG)
```

```
Double_t reactMasses[2];
reactMasses[0] = f3He->Mass();
reactMasses[1] = mass5H;
```

```
fReactionPhaseSpace->SetDecay(IvReaction, 2, reactMasses);
fReactionPhaseSpace->Generate();
```

```
TLorentzVector *lv3He = fReactionPhaseSpace->GetDecay(0);
TLorentzVector *lv5H = fReactionPhaseSpace->GetDecay(1);
```

эти строки нужно переделать:

#3 - 03/09/2018 05:28 PM - Ivan Muzalevsky

- File ERDecayEXP1803.cxx added

- % Done changed from 0 to 90

#4 - 03/13/2018 07:22 PM - Sergey Belogurov

С точки зрения физики это правильно: "Метод TGenPhaseSpace::SetDecay() возвращает false, если суммарная энергия He6+H2 в ЦМ меньше суммы масс H5 и He3." Вас, Иван, попросили объяснить с т.з. физики, а вы о поправке кода. В чем физический смысл ваших изменений, и почему вы считаете, что указанное условие неверно?

#5 - 03/14/2018 04:26 PM - Ivan Muzalevsky

- Status changed from Открыта to Закрыта

- % Done changed from 90 to 100

Проблема была в том, что при разыгрывании энергии возбуждения 5H из функции, состоявшей из двух гауссов, появлялись случаи, когда масса 5H оказывалась слишком большой для протекания бинарной реакции. Задача была решена механически: если масса 5H превысила критическую, делается пересчёт энергии возбуждения заново, итерации продолжаются, пока реакция не сможет протекать. Возможно в будущем можно будет сделать так, чтобы границы, в которых разыгрывается энергия возбуждения рассчитывается автоматически из энергии пучка.

Files

ERDecayEXP1803.cxx	9.5 KB	03/09/2018	Ivan Muzalevsky
--------------------	--------	------------	-----------------