

EXP1803 - bugs #140

Developing # 134 (Открыта): Симуляция эксперимента - первое приближение
bugs # 170 (Открыта): Устранение ошибок

Энергия пучка

02/17/2018 06:09 PM - Ivan Muzalevsky

Status:	Закрыта	Start date:	02/17/2018
Priority:	Нормальный	Due date:	02/23/2018
Assignee:	Ivan Muzalevsky	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:			
Description В макросе симуляции exp1803_full.C есть такие строки, относящиеся к созданию пучка <pre>Double_t kinE_MevPerNucleon = 40.; // Int_t Z = 1, A = 3, Q = 1; // TString ionName = "3H"; Int_t Z = 2, A = 6, Q = 2; TString ionName = "6He"; ERIonMixGenerator* generator = new ERIonMixGenerator(ionName, Z, A, Q, 1); Double32_t kin_energy = kinE_MevPerNucleon * 1e-3 * A; //GeV generator->SetKinE(kin_energy); generator->SetPSigmaOverP(0);</pre> Я ожидаю, в full.root файле кинетическая энергия гелия-6 будет равна 0.24 ГэВ. При выводе такого выражения: he6.fEnergy - he6.fMass, я вижу 0.240989 для каждого события. Вопрос: откуда ещё дополнительный МэВ энергии? наверное пока что это не сильно повлияет на результат, но всё таки это в этом нужно разобраться.			
Related issues: Related to Developing #143: Спектр возбуждений 5H			
		Закрыта	02/18/2018 02/23/2018

History

#1 - 02/18/2018 03:42 PM - Ivan Muzalevsky

- Due date set to 02/23/2018

#2 - 02/18/2018 03:49 PM - Sergey Belogurov

- Assignee changed from Mikhail Kozlov to Vitaliy Schetinin

#3 - 02/19/2018 10:56 AM - Vitaliy Schetinin

Проблема:

Величиной, характеризующей трек, является импульс, а не кинетическая энергия. Для пересчета нужна масса. И тут начинаются танцы. Мы для пересчет используем формулу массы из FairIon: kProtonMass * A;, а когда формируется трек FairPrimaryGenerator-у не сообщается масса, а только PDG и он лезет за массой в Root TPdgDatabase и она там вполне может быть другая. Как и случилось в этот раз.

Лечение:

1) При формировании трека массу в FairPrimaryGenerator надо передавать, а не давать ему считать ее самому. То есть указать в AddTrack параметры по умолчанию.

```
virtual void AddTrack(Int_t pdgid, Double_t px, Double_t py, Double_t pz,  
    Double_t vx, Double_t vy, Double_t vz,  
    Int_t parent = -1, Bool_t wanttracking = true,  
    Double_t e = -9e9, Double_t tof = 0.,  
    Double_t weight = 0., TMCProcess proc = kPPPrimary);
```

2) Брать массу иона сразу из TPdgDatabase.

3) Если пользователь имеет дело с экзотикой, то должен массу иона задавать сам на всякий случай. Для этого есть метод `ERIonGenerator::SetMass(mass)`

#4 - 02/20/2018 10:53 AM - Ivan Muzalevsky

Vitaliy Schetinin wrote:

Проблема:

Величиной, характеризующей трек, является импульс, а не кинетическая энергия. Для пересчета нужна масса. И тут начинаются танцы. Мы для пересчет используем формулу массы из FairIon: $kProtonMass * A$., а когда формируется трек FairPrimaryGenerator-у не сообщается масса, а только PDG и он лезет за массой в Root TPdgDatabase и она там вполне может быть другая. Как и случилось в этот раз.

Лечение:

1) При формировании трека массу в FairPrimaryGenerator надо передавать, а не давать ему считать ее самому. То есть указать в AddTrack параметры по умолчанию.

```
virtual void AddTrack(Int_t pdgid, Double_t px, Double_t py, Double_t pz,  
    Double_t vx, Double_t vy, Double_t vz,  
    Int_t parent = -1, Bool_t wanttracking = true,  
    Double_t e = -9e9, Double_t tof = 0.,  
    Double_t weight = 0., TMCProcess proc = kPPPrimary);
```

2) Брать массу иона сразу из TPdgDatabase.

3) Если пользователь имеет дело с экзотикой, то должен массу иона задавать сам на всякий случай. Для этого есть метод `ERIonGenerator::SetMass(mass)`

Виталик, спасибо большое за развернутый ответ. так вышло, что я почему-то не понял что и где поменять. буду рад если ты сделаешь соответствующий комит, лечащий эту проблему. А я посмотрю, что ты имел ввиду. спасибо

#5 - 02/21/2018 09:59 AM - Vitaliy Schetinin

Иван: это пока информация для понимания как это закодировать, а не как пользоваться). А теперь еще немного такой информации:

Расследование показало

- 1) Табличка pdg database у рута находится тут: https://github.com/root-project/root/blob/master/etc/pdg_table.txt
- 2) Как видно там вообще нет ничего про ионы. Но если принтануть эту таблицу после нашего `run->Init()`, то в конце вижу

Cherenkov	50000050	Mass: 0.0000	Width (GeV): 0.0000e+00	Charge: 0.0
FeedbackPhoton	50000051	Mass: 0.0000	Width (GeV): 0.0000e+00	Charge: 0.0
Genericlon	50000060	Mass: 0.9383	Width (GeV): Stable	Charge: 1.0
Deuteron	1000010020	Mass: 1.8711	Width (GeV): Stable	Charge: 3.0
Triton	1000010030	Mass: 2.8094	Width (GeV): 1.6916e-33	Charge: 3.0
Alpha	1000020040	Mass: 3.7284	Width (GeV): Stable	Charge: 6.0
HE3	1000020030	Mass: 2.8094	Width (GeV): 0.0000e+00	Charge: 6.0
AntiDeuteron	-1000010020	Mass: 1.8756	Width (GeV): Stable	Charge: -3.0
AntiTriton	-1000010030	Mass: 2.8089	Width (GeV): Stable	Charge: -3.0
AntiAlpha	-1000020040	Mass: 3.7274	Width (GeV): Stable	Charge: -6.0
AntiHE3	-1000020030	Mass: 2.8084	Width (GeV): Stable	Charge: -6.0
28S	1000160280	Mass: 26.0777	Width (GeV): 0.0000e+00	Charge: 48.0

- 3) 28S это то, что добавил наш генератор ионов. Все что выше, как бы это не было удивительно, добавляет

`FairEventManager::AddParticlesToPdgDataBase`.

И ради чего? Чтобы в event display подсвечивать лейблы этих частиц, а не pdg код.

Но в данном месте расследование не заканчивается. Потому что масса серы не та, которую я ожидал ($k \text{ProtonMass} \cdot A$).

- 4) Ion заданный нами (как `A,Z,Q,mass` (определяется по глупой формуле или задается пользователем), `ExitationE` (задается пользователем)) сначала идет в `FairMCApplication->AddIons()`

потом в `TGeant4->Definelon` и в определенный момент VMC говорит, что ему плевать на наше представление о массе иона и он сам ее определит по `A,Z,Q, ExitationE`.

Далее `vmc` в `TG4ParticlesManager::AddIon` начинает спрашивать у `geant`, знает ли он что то про ион (`A,Z,ExitationE`), если не знает то создает такой ион

и сам приписывает ему массу. Кухня того как `Geant` считает массу по (`A,Z,ExitationE`) огромна. Там есть куча `if` на легкие ионы и всякие теоритические специальные модели.

Но если уж ничего не знает, то использует это

```
G4double mass = AtomicMass(A,Z);
```

```
// atomic mass is converted to nuclear mass according formula in AME03
```

```
mass -= Z*electron_mass_c2;
```

```
mass += ( 14.4381*std::pow ( Z , 2.39 ) + 1.55468*1e-6*std::pow ( Z , 5.35 ) ) *eV;
```

Выводы:

- 1) то, что мы пытаемся указать массу иона через `Fairlon` вообще не имеет никакого эффекта. Но имеет эффект то, что мы задаем энергию возбуждения.

- 2) навязать свою массу `geant` не дает. Просто крешится, если я в `DatabasePDG` задаю ион руками (я думаю поэтому такая отсечка массы стоит в `vmc`). Так что все, что мы можем крутить у ионов - `Exitation`

- 3) Для правильного пересчета в `ERlonGenerator::SetEKin` энергии в импульс первичного трека, массу надо спрашивать не у `Fairlon`, а напрямую в `G4IonTable::GetTable()->GetIon(A,Z,Q,Exitation)`.

#6 - 02/21/2018 10:04 AM - Vitaliy Schetinin

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

#7 - 02/21/2018 11:16 AM - Vitaliy Schetinin

Поправил. Проверяйте.

Теперь метода SetMass у генератора нет. Только SetExcitationEnergy();

Но и расхождения в задаваемой кин энергии и получаемой тоже нет

#8 - 02/21/2018 11:33 AM - Sergey Belogurov

- Assignee changed from Vitaliy Schetinin to Ivan Muzalevsky

Спасибо Виталику, который сориентировал нас в проблеме. Теперь задача физиков разобраться, относительно чего считается excitation energy, По уму, это превышены массы основного состояния или полной энергии возбужденного над $A \cdot \text{AMU}$, но лучше Ивану в этом убедиться и написать в документацию ER ion generator, что ему удалось установить "экспериментально" по поводу связи excitation energy и полной энергии.

#9 - 02/21/2018 11:39 AM - Vitaliy Schetinin

1) У нас вообще бывают возбужденные пучковые ионы?

2) Сейчас я запрограммировал, что есть масса основного состояния. Она берется из Geant по A,Z. Во всех наших пересчетах между Кинетической энергией и импульсом за массу принимается эта масса + заданный пользователем excitation.

#10 - 02/21/2018 12:01 PM - Sergey Belogurov

Похоже, что масса иона в GEANT считается по хитрой формуле, а не как $A \cdot \text{Amu} + \text{Excit.}$ (это как делал СИ), поэтому Ивану стоит проанализировать, какую масс дает geant с тем, чтобы вводить правильную энергию возбуждения.

Что касается возбужденных ионов.

Нет пучковые ионы мы рассматриваем в основном состоянии, но для корректировки массы этого состояния, вероятно, придется использовать переменную excitation energy, поскольку не факт, что geant правильно считает массу экзотических ионов. Важно понимать, что никакого размазывания энергии возбуждения по гауссу для пучковых ионов не требуется!

Есть еще другая тема - распад короткоживущего иона (H-5 в нашем случае). Поскольку этот ион распадается мгновенно, он не имеет трека и не должен пропагироваться geant. По сути этот ион существует только как 4-вектор с какой-то массой (полной энергией) и есть задача о размазывании этой полной энергии по гауссу. Боюсь, что здесь возникла некоторая путаница. Все ли теперь понятно?

#11 - 02/21/2018 12:21 PM - Vitaliy Schetinin

Путаницы никакой нет. Есть генератор, есть распад. Сущности разные, с разными интерфейсами. Здесь обсуждали только генератор.

#12 - 02/27/2018 07:50 PM - Ivan Muzalevsky

- Status changed from *Открыта* to *Закрыта*

- % Done changed from 0 to 100

#13 - 03/27/2018 05:41 PM - Vratislav Chudoba

- Parent task set to #170