

Garfield++拡張用クラス TrackTrimSQLite の使用法

村田求基

1 概要

Garfield++が提供するイオンの飛跡計算クラス TrackSrim を用いるとイオンの飛跡のストリング効果が過大評価され、生成される飛跡の形状が現実の飛跡形状を再現しないことが以前から知られていた。そのため TrackSrim の代替となるクラス TrackTrimSQLite を開発した。TrackTrimSQLite は外部プログラム (TRIM) を用いて計算した荷電粒子の飛跡のデータを入力とし、入力をもとにして新たな飛跡を生成する。TRIM で計算した飛跡データを SQLite のデータベースに保存しておき、飛跡生成で必要になるたびにデータベースから衝突事象が読み込むように実装することでプログラムのメモリ消費を抑えた。また、有限に飛跡サンプルから多様なサンプルを生成するためにいくつかの工夫をした。

2 目的

Garfield++ [1] は C++ で書かれたガス検出器中の荷電粒子飛跡シミュレーションライブラリであり媒質中での荷電粒子のエネルギー損失過程から電子のドリフト、ガス増幅といった検出器の信号形成に至る一連のプロセスをモンテカルロ法を用いて計算することができる。

Garfield++が提供するクラスの一つに低エネルギーイオンのエネルギー損失過程を計算するためクラス TrackSrim がある。このクラスは外部ソフトウェアの SRIM [2] によって計算したイオンのエネルギー損失やイオン飛跡の統計的な広がりの効果を入力とし、それをもとにイオンによる電離によって生成される電子の三次元的な分布を出力する。しかし、TrackSrim が計算結果として出力する電子分布は実際に観測される電子分布と比べて空間的に広がった形状となることがわかった。この理由は SRIM の計算結果から飛跡を生成するためのアルゴリズムの誤りによるものと思われる。

2.1 Garfield++の TrackSrim の飛跡生成アルゴリズムと問題点

Garfield++では飛跡生成を Track クラスで行う。TrackSrim は Track の派生クラスで SRIM (後述) の計算結果のテーブルをもとにしてイオンの飛跡上に生成される電子の分布を計算する。

Listing 1: Garfield++の TrackSrim の Generate 関数 (2019/05/19 版) の一部。

```
1 bool TrackSrim::NewTrack(const double x0, const double y0, const double z0,  
2                          const double t0, const double dx0, const double dy0,  
3                          const double dz0)  
4 {  
5  
6     // ...
```

```

7
8 // Initial situation: starting position
9 double x = x0;
10 double y = y0;
11 double z = z0;
12
13 // ...
14
15 // Loop generating clusters
16 int iter = 0;
17 while (iter < m_maxclusters || m_maxclusters < 0) {
18     // Work out what the energy loss per cm, straggling and projected range are
19     // at the start of the step.
20     const double dedxem = DedxEM(e) * m_density;
21     const double dedxhd = DedxHD(e) * m_density;
22     const double prange = Interpolate(e, m_ekin, m_range);
23     double strlon = Interpolate(e, m_ekin, m_longstraggle);
24     double strlat = Interpolate(e, m_ekin, m_transstraggle);
25
26     // Add a cluster.
27     cluster newcluster;
28     newcluster.x = x;
29     newcluster.y = y;
30     newcluster.z = z;
31     newcluster.t = t0;
32
33     // ...
34
35     m_clusters.push_back(newcluster);
36
37     // ...
38
39     // Draw scattering distances
40     const double scale = sqrt(step / prange);
41     const double sigt1 = RndmGaussian(0., scale * strlat);
42     const double sigt2 = RndmGaussian(0., scale * strlat);
43     const double sigl = RndmGaussian(0., scale * strlon);
44     if (m_debug) std::cout << hdr << "sigma_l,t1,t2:"
45         << sigl << ", " << sigt1 << ", " << sigt2 << "\n"
46         ;
47     // Rotation angles to bring z-axis in line
48     double theta, phi;
49     if (xdir * xdir + zdir * zdir <= 0) {
50         if (ydir < 0) {
51             theta = -HalfPi;
52         } else if (ydir > 0) {
53             theta = +HalfPi;
54         } else {
55             std::cerr << hdr << "\nZero step length; clustering abandoned.\n";
56             return false;
57         }
58     }
59 }

```

```

56         }
57         phi = 0;
58     } else {
59         phi = atan2(xdir, zdir);
60         theta = atan2(ydir, sqrt(xdir * xdir + zdir * zdir));
61     }
62
63     // Update position
64     const double cp = cos(phi);
65     const double ct = cos(theta);
66     const double sp = sin(phi);
67     const double st = sin(theta);
68     x += step * xdir + cp * sigt1 - sp * st * sigt2 + sp * ct * sigl;
69     y += step * ydir + ct * sigt2 + st * sigl;
70     z += step * zdir - sp * sigt1 - cp * st * sigt2 + cp * ct * sigl;
71
72     // ...
73 }
74
75 // ...
76
77 }

```

TrackSrim ではメンバ関数の NewTrack() のなかで飛跡を計算している。NewTrack() 関数の一部をリスト 1 に示した。NewTrack() 関数ではリストの 17 行目のループ中でイオンのエネルギー損失のステップを計算している。そしてステップごとに位置やエネルギーといった情報を電子クラスターを表すクラスの cluster に記録し、TrackSrim クラスのメンバーである cluster の可変長配列の m_clusters に保存している。

各ステップごとにストラグリングによる位置の SRIM の計算結果のテーブルには、あるエネルギーのイオンが停止するまでのストラグリングの大きさが収められており、23 行目と 24 行目で現在のイオンの運動エネルギーのイオンのストラグリングの大きさをテーブルの補間により計算している。

そして、40 行目でイオンのこのステップで進む距離と止まるまでに進む飛程の比率を計算し scale に代入し、前に求めた全エネルギー喪失までのストラグリングと scale の積をこのステップの平均ストラグリングとしステップの平均ストラグリングを標準偏差とする正規分布で乱数を生成することでこのステップでの飛跡の直線からのずれの大きさをランダムに計算している。

Garfield++ が採用しているステップ内のストラグリングを全飛程とステップ長の比率から全ストラグリングから計算するという処理では、ストラグリングと飛程の間に比例関係が成立するということが念頭におかれている。もしこの仮定が正しければ同じ方向に同じエネルギーで放出されたイオンの集団の飛跡は、平均的に、飛跡の出発点から広がる円錐状の領域の内部に分布する。しかし、これは一般に正しくないと思われる。実際のイオンの飛跡のストラグリングは、むしろイオンのエネルギーが高いうちは非常に小さく、イオンのエネルギーが 0 に近づくと急激に大きくなるため、同条件の飛跡の集団は漏斗のような形状の領域に分布する。

Garfield++ のように単純な比例計算によりストラグリングを計算した場合、イオンがある程度大きい運動エネルギーを持っている領域でストラグリングの大きさが過大評価されることになる。その結果として、飛跡のうちイオンの運動エネルギーが大きい領域で飛跡が入射方向から大きくはずれるイベントが実際より大きな確率で発生し、また飛跡全体としても空間的な広がりが現実

より大きくなってしまふ。

2.2 SRIM と TRIM

SRIM はイオンの物質中でのエネルギー損失過程を計算するための計算ソフトウェア群である。詳細は付録を参照のこと。SRIM をインストールすると、同時に TRIM というソフトウェアもインストールされる。以下では、SRIM と TRIM の違いについて簡単に述べる。

SRIM はイオンの種類とエネルギーを入力として、設定した物質中で停止するまでの飛跡の長さや広がりなどの平均値を記録した表を出力する。表には物質中で停止しイオンの平均的な最終到達地点とイオンの入射時のエネルギー損失率 (dE/dx) が記録されている。出力される情報は同一条件の多数の飛跡の平均値であり、個々の飛跡の形状に関する情報は失われている。

一方、TRIM は SRIM に組み込まれたもうひとつの飛跡計算用ソフトウェアであり、SRIM よりも高機能である。入射イオンの種類と運動エネルギーに関する入力に加えて、物質の形状を SRIM よりも細かく設定することができる。TRIM ではイオンによる電離過程に加えて、物質中の原子の移動や電離電子による二次的な電離過程も計算することができる。そして、個々の飛跡の形状に関する情報を出力することができる。

2.3 Garfield++ の誤りを訂正する方法の方針について

Garfield++ では SRIM の簡便な出力ファイルを用いて、個々の飛跡イベントを統計的に生成するという方針を取っている。しかし、SRIM の出力する表から実際の物理過程を再現する方法には前述の通り誤りがある。

Garfield++ の誤りを修正する方法には 2 つの方針が考えられる。一つは Garfield++ 同様 SRIM の出力ファイルを用いて計算方法を正しく修正するという方針、もう一つは TRIM の出力する個々の飛跡の形状を反映するという方針である。

前者の場合、Garfield++ で各ステップのストラグリングを飛跡長でスケーリングして計算するところを置き換えることになる。今、イオンの運動エネルギーが E から $E - dE$ に減少する過程を 1 ステップとする。SRIM の出力は運動エネルギー E のイオンの停止までのストラグリングの平均値 $\Sigma(E)$ であり、飛跡を生成するにはここからこのステップのストラグリングの平均値 $\sigma(E, dE)$ を計算すればよい。なお、SRIM の出力にはエネルギー E のイオンの飛程 $\Lambda(E)$ とエネルギー損失率 $((dE/dx)(E))$ も既知である。Garfield++ の TrackSrim では

$$\begin{aligned} (\sigma(E, dE))_G &\equiv \Sigma(E) \times \left(\frac{dL(E, dE)}{\Lambda(E)} \right) \\ dL(E, dE) &\equiv \frac{dE}{\left(\frac{dE}{dx}(E) \right)} \end{aligned} \quad (1)$$

と計算している。ただし、 $dL(E, dE)$ はステップの移動距離である。しかし、これは正しくない。むしろ、

$$(\sigma(E, dE))_{G'} \equiv \sqrt{(\Sigma(E))^2 - (\Sigma(E - dE))^2} \quad (2)$$

とすべきである。

この式は、エネルギー E のイオンの静止までのストラグリングの過程を、現在のステップ ($E \rightarrow E - dE$) とそれ以降の飛跡 ($E - dE \rightarrow 0$) とに分割した際に、停止までの全飛跡のストラグリングが飛跡の前半部分と後半部分の独立な試行によって決定されるとしたときの式である。平均 μ 、標準

偏差 σ の正規分布を $N(x; \mu, \sigma^2)$ と書くことにすると、飛跡全体のストラグリングは $N(x; 0, \Sigma(E)^2)$ に従い、飛跡の後半部分のストラグリングは $N(x; 0, \Sigma(E - dE)^2)$ に従う。このとき、飛跡の前半部分 (現在のステップ) のストラグリングの分布 $P(x)$ を求める問題を考える。

あるイベントでの前半部分 (ステップ) のストラグリングを x_s 、同様に後半部分のストラグリングを x_r 、全体のストラグリングを X とすると

$$X = x_s + x_r \quad (3)$$

である。前半と後半のストラグリングは独立な試行とみなせるので、これらの分布の間には以下の式が成り立つ。

$$\begin{aligned} N(X; 0, \Sigma(E)^2) &= \int \int dx_s dx_r \delta(X - x_s - x_r) P(x_s) N(x_r; 0, \Sigma(E - dE)^2) \\ &= \int dx_s P(x_s) N(X - x_s; 0, \Sigma(E - dE)^2) \end{aligned} \quad (4)$$

この式と正規分布の再現性

$$N(X; \mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2) = \int dx N(x; \mu_1, \sigma_1^2) N(X - x; \mu_2, \sigma_2^2) \quad (5)$$

より、 $P(x_s)$ は正規分布であり、

$$P(x_s) = N(x_s; 0, \Sigma(E)^2 - \Sigma(E - dE)^2) \quad (6)$$

となる。

もうひとつの Garfield++ を修正する方針は、TRIM を用いて計算した飛跡の形状をそのまま飛跡生成のために用いるというものである。TRIM の出力ファイルには、計算結果のすべての飛跡の上でおこった衝突の座標やそのときの運動エネルギーが記録されている。これを用いることで、Garfield++ の他の Track クラスと同様に、飛跡上に生成される電子の空間分布を生成することができる。

これらふたつの方針のうち、今回は後者の TRIM の出力を用いるという方針を採用した。その理由は、SRIM の出力するテーブルを用いる方法では SRIM の出力するストラグリングの平均値がエネルギーの関数として滑らかでないことがありその場合テーブルごとに個別のモデル化が必要であるがそれが煩雑であること、TRIM の出力する飛跡は衝突する原子の種類による散乱角度分布の違いなどが考慮された現実的な形状であることなどである。TRIM を用いて十分多くの飛跡サンプルを計算しておけばサンプルからランダムに一つを抽出することで比較的容易に飛跡を生成するクラスを実装することができる。しかし、事前に用意したサンプル数が生成する飛跡の数に比べて十分多くなければ、同一のサンプルが何度も生成されるため生成される飛跡にバイアスがかかってしまうことになる。次章ではこれを克服するための手法と飛跡を生成するクラスの実装について述べる。

3 TrackTrimSQLite の仕様

3.1 飛跡生成の流れ

TrackTrimSQLite の飛跡生成過程は、まず TRIM を用いて計算した飛跡を生成するためには TRIM による計算結果を記録したデータベースが必要である。そのため、TrackTrimSQLite によ

る飛跡シミュレーションの前に別のプログラムを用いて TRIM の出力したイオンの衝突事象をデータベースに保存しておく。飛跡の生成はデータベースの衝突事象をランダムに抽出することで行う。基本的にはランダムに一つの衝突の系列を選択し、その系列を入力された条件に適合させて出力する。

3.2 SQLite のデータベース構成

TRIM の計算結果は SQLite [3] のデータベースに記録しておき、TrackTrimSQLite ではそのデータベースにアクセスして飛跡の情報を取得し、それをもとに新しい飛跡を生成する。以下ではそのデータベースの構成について述べる。

TrackTrimSQLite で用いるデータベースは 1 つのテーブル (collisions) からなる。collisions テーブルの構成を表 1 に示す。

表 1: SQLite のデータベースの collisions テーブルの構成。一つのレコードは一回の衝突に対応する。track_id と collision_id の組み合わせが collisions テーブルの主キーに設定されている。

カラム名	内容	型	主キー制約
track_id	飛跡の通し番号	INTEGER	collision_id との組み合わせ
collision_id	飛跡内での衝突の通し番号	INTEGER	track_id との組み合わせ
e_inc	衝突前のイオンの運動エネルギー	REAL	-
incident_ion	イオンの元素記号	TEXT	-
incident_ion_mass	イオンの質量数	INTEGER	-
recoil_ion	反跳した原子の元素記号	TEXT	-
e_rec	反跳した原子の運動エネルギー	REAL	-
x	衝突の x 座標	REAL	-
y	衝突の y 座標	REAL	-
z	衝突の z 座標	REAL	-
dx0	衝突前の方向ベクトルの x 成分	REAL	-
dy0	衝突前の方向ベクトルの y 成分	REAL	-
dz0	衝突前の方向ベクトルの z 成分	REAL	-
dx1	衝突後の方向ベクトルの x 成分	REAL	-
dy1	衝突後の方向ベクトルの y 成分	REAL	-
dz1	衝突後の方向ベクトルの z 成分	REAL	-
dr	次の衝突までの距離	REAL	-
de	次の衝突までの間でのエネルギー損失	REAL	-

collisions テーブルは飛跡上で起こった原子との衝突がレコードとして記録しており、TRIM の出力ファイル COLLISION.txt と RANGE_3D.txt をもとに生成する。

データベースの作成にあたり、各飛跡は COLLISION.txt の衝突履歴を順に辿った後に RANGE_3D.txt に示された最終到達地点で全エネルギーを失うものと想定した。また、イオンは原子との衝突による離散的なエネルギー移行と原子との衝突の間の連続的なエネルギー移行のふたつの過程でエネルギーを失うものと想定した。これは、COLLISION.txt に記録されている連続する 2 つの衝突時の運動エネルギーを比較すると、それらの差が衝突によって反跳粒子に与えた運動エネルギーよ

りも大きいと、記録されている衝突の間に別の過程によって粒子がエネルギーを失ったと考えるべきであるからである。テーブルの `e_rec` は衝突により原子に与えたエネルギー、`de` は記録された衝突点の間でのエネルギー損失を表している。`de` は `COLLISON.txt` の各衝突点での運動エネルギーから、その点での反跳原子のエネルギーと次の衝突点での運動エネルギーの差から計算した。

`collisions` テーブルでは、`track_id` と `collision_id` の組み合わせが主キーとして設定されており、2つの値がともに同じレコードは存在しない。

方向ベクトル $(dx0, dy0, dz0)$ および $(dx1, dy1, dz1)$ はノルムが1に規格化されている。しかし、TRIM の計算結果においてはしばしば連続する2つの散乱点が同一の座標を持つことがある。その場合方向ベクトルは零ベクトルになっている。飛跡の最後の点では、イオンがすべてのエネルギー次のステップが存在しないので $(dx1, dy1, dz1)$ は零ベクトルである。

3.3 飛跡生成のアルゴリズム

基本的に飛跡の生成にはデータベースの飛跡サンプルをランダムに抽出して用いる。もっとも単純な実装では、ランダムに抽出されたサンプル飛跡の一連の衝突列をそのまま生成される飛跡とする方法が考えられる。

一般に入力イオンの運動エネルギーとサンプルの衝突の運動エネルギーは異なるため、サンプルを適当に切り詰めて入力に合うように衝突列を処理する必要がある。このクラスではサンプルの衝突点間のエネルギー損失が一様であると仮定し、線形補間によってサンプルの運動エネルギーが入力の運動エネルギーと等しくなる点の座標を求める。そしてその点を飛跡サンプルの始点として再定義し、飛跡の入力の方向とサンプルの方向ベクトルが平行になるようにサンプルの飛跡全体を三次元的に回転させ、それを生成される飛跡とする。

実際のシミュレーションでは少数のサンプルからそれ以上の数のイベントを偏りなく生成することが望ましい。単にサンプル列を抽出するだけであっても同一条件で生成される飛跡の個数がデータベース中のサンプル数に対して十分少なければ生成される飛跡の分布の偏りは問題にならない。しかし、サンプル数を超えてイベント生成を行った場合、平均的に同一のサンプルが複数回の生成に使用されることになり、生成される飛跡の形状に偏りが生じる。この問題を回避するために、有限のサンプルから可能な限り多様な飛跡を生成するための処方を実装した。

3.3.1 散乱角度の回転

多様な飛跡を生成するための処方の一つは、サンプルの散乱角度を衝突時のイオンの入射方向軸回りにランダムに回転させることである。

イオンの原子の衝突でイオンが散乱される角度は微分散乱断面積によって決まる。入射ビーム方向が z 軸と平行であるとき、三次元球面座標で考えると、一般に微分散乱断面積は極角 (θ) と方位角 (ϕ) の両方についての関数となる。

しかし始状態に偏極がない場合、散乱断面積は θ のみの関数となる。今イオンの統計的な振る舞いのみに興味がある。そのため微分散乱断面積が ϕ 依存性を持たないという仮定をおいて計算しても問題ないと思われる。したがって、サンプルにある衝突事象は散乱方向の ϕ をランダムに決定したイベントと全く同一の確率でおこると考える。

以上の考察のもと、飛跡を計算する際にはすべての散乱点で一様乱数を用いて角度 $\tilde{\phi}$ を生成し、散乱角度の方位角 ϕ を $\phi \rightarrow \phi + \tilde{\phi}$ と更新する処方を実装した。この処方によって、イオンの飛跡の空間上の位置に多様性を生じさせることができる。

3.3.2 飛跡サンプルの乗り換え

多様な飛跡を生成するための処方のもう一つは、異なる飛跡サンプル間で乗り換えを行うことである。散乱角度の方位角を回転する処方は、飛跡の空間的な多様性を生じさせるがこの処方ではそれに加えて運動エネルギーの空間で多様性を生じさせることを目指す。

物理的にはあるエネルギーで生成されたイオンが次に原子と衝突するまでに失うエネルギーの大きさや距離は、イオンと原子の散乱断面積によって決まり、その値は指数分布に従う。しかし TRIM で計算した 1 つの飛跡サンプルをエネルギーが大きい順番に追跡すると、引き続いて起こる 2 つの衝突の間のエネルギー損失や衝突点間の距離は定まったひとつの値をとる。本来、いつ次の散乱が起こるかは確率的な事象であるため、飛跡生成のアルゴリズムとしては、衝突点の間のエネルギー損失や衝突転換の距離が確率的に決定されることが望ましい。したがって、本処方ではこの過程を近似的に取り込む。

ある運動エネルギーイオンに対して、原子との衝突の断面積を知っていれば、ある散乱点の次の散乱点を適切に確率的に生成することができる。しかし、TRIM の出力ファイルには断面積についての直接的な情報はない。そのため、確率的に散乱を記述するためには何らかの近似が必要となる。

このクラスでは、散乱断面積の大きさによって衝突頻度が変わる効果を近似的に取り入れるためにイオンの運動エネルギーが近い衝突事象をいくつかサンプリングすることにした。散乱断面積のエネルギー依存性が緩やかであれば、運動エネルギーが近似する衝突事象サンプルの衝突頻度は近似的に同一の指数分布に従うと想定できる。こうしてサンプリングされた近似的に指数分布に従う集団から、ランダムに乗り換え先の衝突事象を抽出することで、物理的に決定される衝突頻度に従いつつ飛跡形状の多様性を実現することができる。

具体的な処理の流れは以下のとおりである。まず乗り換え前の衝突後の運動エネルギーが E 、次の衝突までのエネルギー損失が dE であったとき、データベースから衝突後の運動エネルギーが $E - RdE$ 以上 $E + RdE$ 以下の衝突事象を乗り換え候補として抽出する。ただし R は運動エネルギーを同一視する範囲の指定するパラメータで、1 以下の正の実数である。次に乗り換え候補からランダムに一つの衝突を抽出する。抽出された衝突の後の運動エネルギーが \tilde{E} 、その次の衝突までのエネルギー損失が \tilde{dE} であったとする。乗り換え処理では、乗り換え前のエネルギー E で起こった衝突の次に運動エネルギーが $\tilde{E} - \tilde{dE}$ で衝突が起きたものとして飛跡を加工する。すなわち、乗り換え先として選出された衝突のひとつ後の衝突を乗り換え前の衝突事象に接続する。その際、衝突間の距離はエネルギー損失率が一定であると仮定して更新を行う。そして、それ以降の飛跡では乗り換え後の飛跡系列にしたがって処理を続行する。

乗り換え処理では、散乱断面積が同じとみなせるエネルギーの範囲と衝突点ごとの乗り換え処理を実行する確率の 2 つがパラメータとなる。エネルギー範囲は、乗り換え前の衝突が次に衝突を起こすまでの連続的なエネルギー損失 (de) に対する割合で指定する。デフォルトではこの値は 0.5 としている。また、乗り換えの確率は 0.1 としている。

生成される飛跡の品質を向上させるためには、エネルギー範囲の割合を適当に小さく、乗り換え確率を高くとるべきである。その場合、断面積の効果がより正確に取り入れられ、より多様な飛跡が生成される。しかしエネルギー範囲を小さく取りすぎた場合、乗り換え先が見つからない可能性がある。その場合乗り換えの確率が、指定した値より小さくなる可能性があることには注意が必要である。また、計算速度の観点からは乗り換え確率をある程度低い状態で保っておいたほうが良い。飛跡生成の過程はデータベースへのアクセスが律速しているため、乗り換えを試みるたびに発生するデータベースへのアクセス頻度に比例して計算速度が低下してしまう。

3.4 電子分布生成のアルゴリズム

前節のようにして生成した衝突の系列から、Garfield++の Track クラスの出力と同じ形で飛跡に沿って生じた電子の分布を生成する過程では以下の仮定をおいた。

- イオンがガス中でガスのw値にあたるエネルギーを損失すると、必ずひとつの電子が生成される
- 原子との衝突の間で起こるエネルギー損失は、引き続いて起こる衝突点を結ぶ直線上で一様に起こる
- イオンによって反跳された原子は、衝突地点で全エネルギーを損失する

生成される電子の数は w 値によってきまる。w 値はガス中で一つの電子を電離するために必要となる平均エネルギーとして定義され、様々な単一組成ガスについての値が知られている。具体的な値については Appendix H を参照せよ。

複数の種類のガスを混合する場合単一組成ガスの w 値から混合ガスの w 値を計算する必要がある。イオン化断面積が気体分子の電子数に比例するとすれば、ガス i の w 値を w_i 、1 分子あたりの電子数を z_i 、混合比率を P_i と書くことにすると、混合ガスの w 値 w_{comp} は

$$w_{comp} = \frac{\sum_i P_i z_i w_i}{\sum_i P_i z_i} \quad (7)$$

と計算すればよいと思われる。

4 使用方法

4.1 実行環境の準備

新たな飛跡シミュレーションを行う場合、まず TRIM の動作する環境が必要である。TRIM は Windows 上で動作するソフトであるが、Linux などでも仮想的に Windows を動作させるソフトである Wine を使用すれば TRIM による計算を実行することができる。

そして、TRIM の結果を SQLite のデータベースに変換するためには SQLite 3 の C 言語 API とある程度新しい C++ 規格 (C++14 ?) に対応したコンパイラが必要である。

最後に TRIM の計算結果をもとに TrackTrimSQLite を動作させるためには、Garfield++ と SQLite3 の API が動作する環境が必要である。また Garfield++ のコンパイルには ROOT のインストールが必要である。

4.2 TRIM による計算

飛跡を生成したいイオンと媒質のもとでの飛跡サンプルを TRIM を用いて計算する。TRIM の使用方法の詳細については付録にまとめた。

イオンの運動エネルギーは、生成したいイオンの最大運動エネルギーより十分高い値に設定しておかなければならないことに注意が必要である。計算結果のファイルは SRIM のインストールされたディレクトリ配下にある "SRIM Outputs" に保存されるので、そのディレクトリから COLLISON.txt と RANGE.3D.txt の両方を次のデータベース生成用プログラムが参照できる、同一のディレクトリに保存しておく必要がある。

4.3 データベースの準備

TRIM の計算結果のうち COLLISON.txt と RANGE_3D.txt の 2 つを用いてデータベースを作成する。同じ計算によって生成されたこれら 2 つのファイルを同一のディレクトリにおいた上で、リスト 2 に示したような、クラス TRIM2SQLite を用いた以下のようなプログラムを用いてデータベースに変換する。このプログラムは、コマンドライン引数から、TRIM の計算結果のディレクトリと生成されるデータベースのファイル名を指定して実行する。

Listing 2: makdb.cpp (TRIM の計算結果を SQLite のデータベースに変換するプログラム。)

```
1 #include <iostream>
2 #include "TRIM2SQLite.hpp"
3
4 int main(int argc, char *argv[])
5 {
6
7     if (argc < 3)
8     {
9         std::cerr << argv[0] << "[input_directory] [output_name]" << std::endl
10         ;
11         return 1;
12     }
13     TRIM2SQLite t2s;
14     // Example
15     // t2s.MakeSQLiteFile("../input/TRIM/1000/3H/10", "hoge.sqlite");
16     t2s.MakeSQLiteFile(argv[1], argv[2]);
17     return 0;
18 }
```

4.4 TrackTrimSQLite の使用

TrackTrimSQLite は、飛跡の多様性を作るための処理のパラメータを設定する関数以外の部分の仕様が Garfield++ の TrackSrim と同一になるように設計したので基本的に trackSrim と同様の方法で使うことができる。リスト 3 にコードの例を示す。リスト中で TrackTrimSQLite に関係する部分を赤色で示した。

飛跡の乗り換え確率を設定したい場合は、TrackTrimSQLite のメンバ関数の SetTransferProbability() を使用する。また、飛跡の乗り換え候補探索のエネルギー範囲を設定する場合、同じくメンバ関数の SetEnergyMarginRatio() を使用する。これらのパラメータはデフォルトでは、0.1、0.5 となっている。

Listing 3: testTrackTrimSQLite.cpp (TrackTrimSQLite の使用例。)

```
1 #include <iostream>
2 #include <vector>
3 #include <string>
4
5 #include <TFile.h>
6 #include <TTree.h>
7
8 #include "SolidBox.hh"
```

```

 9 #include "GeometrySimple.hh"
10 #include "ComponentConstant.hh"
11 #include "MediumMagboltz.hh"
12 #include "Sensor.hh"
13 #include "TrackSrim.hh"
14 #include "Random.hh"
15 #include "Plotting.hh"
16 #include "AvalancheMC.hh"
17 #include "TrackSrim.hh"
18 #include "Random.hh"
19
20 #include "TrackTrimSQLite.hpp"
21
22
23 int main()
24 {
25
26     const double world_size = 100;  //(cm)
27
28     auto gas = new Garfield::MediumMagboltz();
29     auto box = new Garfield::SolidBox(0, 0, 0,
30                                     world_size, world_size, world_size);
31     auto geo = new Garfield::GeometrySimple();
32     auto comp = new Garfield::ComponentConstant();
33     auto sensor = new Garfield::Sensor();
34
35     geo->AddSolid(box, gas);
36     comp->SetGeometry(geo);
37     sensor->AddComponent(comp);
38
39     //Gasfile
40     const std::string gasfile = "../input/He(90)+CH4(10)_1000.gas";
41     gas->LoadGasFile(gasfile);
42
43     // Gas property
44     double a_eff, z_eff, w_eff;
45     a_eff = 4 * 0.9 + 16 * 0.1;
46     z_eff = 2 * 0.9 + 10 * 0.1;
47     w_eff = (2 * 41.3 * 0.9 + 10 * 30 * 0.1) / z_eff;
48
49     // TrackClass
50     auto track = new GarfieldSuppl::TrackTrimSQLite();
51     track->ReadFile("./hoge.sqlite");
52     track->SetSensor(sensor);
53     track->SetWorkFunction(w_eff);
54     track->SetTargetClusterSize(1);
55     track->SetTransferProbability(0.1);
56     // track->EnableDebugging();
57
58
59     //value for Tree

```

```

60     std::vector<double> vx, vy, vz, vt;
61     std::vector<int> vn;
62     std::vector<double> ve_cls, ve_ion;
63
64     TFile fOut("tracktrim.root", "recreate");
65     TTree tr("tr", "");
66     tr.Branch("x", &vx);
67     tr.Branch("y", &vy);
68     tr.Branch("z", &vz);
69     tr.Branch("t", &vt);
70     tr.Branch("n", &vn);
71     tr.Branch("e_cls", &ve_cls);
72     tr.Branch("e_ion", &ve_ion);
73
74     for (int iTrack = 0; iTrack < 1000; ++iTrack)
75     {
76         track->SetKineticEnergy(1.0e+6);
77         std::cout << "□" << iTrack << std::endl;
78         track->NewTrack(0, 0, 0, 0, 1, 0, 0);
79
80         double xc, yc, zc, tc;
81         int ne_c;
82         double e_cls, e_ion;
83
84         vx.clear();
85         vy.clear();
86         vz.clear();
87         vt.clear();
88         vn.clear();
89         ve_cls.clear();
90         ve_ion.clear();
91
92         while (track->GetCluster(xc, yc, zc, tc, ne_c, e_cls, e_ion))
93         {
94             vx.push_back(xc);
95             vy.push_back(yc);
96             vz.push_back(zc);
97             vt.push_back(tc);
98             vn.push_back(ne_c);
99             ve_cls.push_back(e_cls);
100            ve_ion.push_back(e_ion);
101        }
102
103        tr.Fill();
104    }
105
106    tr.Write();
107
108    // delete all of Garfield objects
109    delete gas;
110    delete box;

```

```
111     delete geo;
112     delete comp;
113     delete sensor;
114     delete track;
115
116     return 0;
117 }
```

5 性能評価

1 MeV の ^3H を $\text{He}+\text{CH}_4$ (10%) 1000 hPa に x 軸の正の方向に入射させた際の飛跡を TrackTrimSQLite と従来の Garfield++ のクラス (TrackSrim) で計算しその結果を図に示した。飛跡の乗り換えに関するパラメータはデフォルト値を用いた。また、比較のためにこの計算をするにあたり TrackTrimSQLite の入力とした TRIM による計算結果も合わせて示した。

図の縦横 2×2 の 4 つのヒストグラムがそれぞれ一つの計算条件に対応する。4 つのヒストグラムはそれぞれ左上から飛跡を横から見たときの電子分布、イオンの運動エネルギーと深さ (x) の関係、飛跡終端の深さ (x) の分布、飛跡終端の横方向 (y, z) の分布を表している。そして、図の左上の 4 つのヒストグラムが TrackSrim、右上が TrackTrimSQLite、左下が TRIM の計算結果を表している。これらのうち、TRIM の計算結果では電子の分布ではなく原子との衝突の座標が示されている。これは TRIM の計算結果には電子の分布は直接含まれないためである。

図より TrackSrim では電子の分布が横から見ると円錐状になっていることがわかる。そして、飛跡終点の横方向の広がりには TRIM の計算結果と比べて大きくなっている。この結果より、TRIM の結果を信頼する立場のもとでは、TrackSrim ではイオンのストラグリングが過大評価されているという経験的に知られていた事実を再確認することができる。本来 SRIM と TRIM には同様の方法で計算されるため、TRIM の生成する飛跡の形状と SRIM の出力結果をもとに計算した飛跡の形状は一致すべきである。そしてそれらが一致しないということは、飛跡を生成する過程に誤りが含まれるということを意味する。

一方、TrackTrimSQLite の計算結果は横から見た飛跡形状や飛跡終端の位置の分布に関して TRIM の出力をよく再現している。TrackTrimSQLite の飛跡を横から見た形状を見ると、頻繁に出現する飛跡から大きくはずれた方向の飛跡が多いようにも見える。しかしこれは飛跡の生成の誤りではなく、TrackTrimSQLite と TRIM でヒストグラムに描画されているものが異なるため、TrackTrim では離れた飛跡が強調されているだけであると思われる。このことは終端座標の分布が TRIM の計算結果と同等であることからわかる。

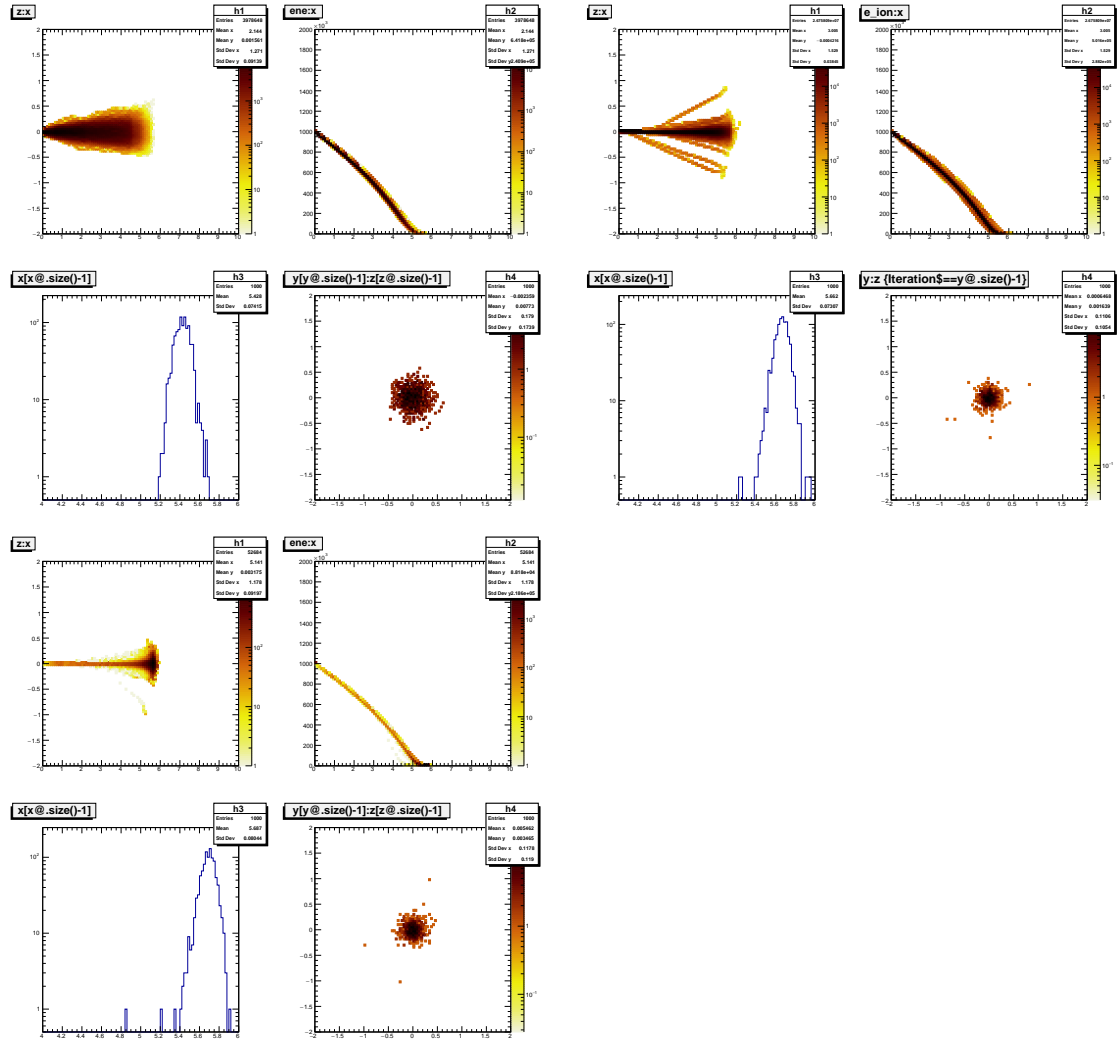


図 1: 生成した飛跡。4つのヒストグラムが一つの計算条件に対応する。左上が Garfield++ の TrackSrim、右上が今回開発した TrackTrimSQLite、左下が TrackTrimSQLite の入力に用いた TRIM の計算結果。詳細は本文参照。

6 まとめ

Garfield++の飛跡生成アルゴリズムの誤りを修正した新たなクラス TrackTrimSQLite を開発した。このクラスでは TRIM の計算した飛跡の形状データをもとに新たな飛跡を生成する。同一条件のシミュレーションを実行し、TrackTrimSQLite の計算結果は飛跡の形状が TrackSrim とは異なり正しいと思われる形状をしていること、TRIM の計算結果と同様の分布に従うことを確認した。

A SRIM (The Stopping and Range of Ions in Matter)

SRIM は物質中に侵入したイオンの阻止能や飛程を計算する一群のプログラムであり、計算の過程ではイオンと物質中の原子の衝突過程は量子力学的に取り扱われる。(以下では移動する原子をイオンと呼び、標的中の原子を原子と呼称する。) SRIM の計算は統計的アルゴリズムを用いることで効率化されている。まずイオンは連続する衝突点の間を離散的に移動する。その後、衝突点の間の衝突の効果が平均化されて取り入れられる。イオンと原子の衝突過程としては、電子の交換と重なり合う電子殻同士の相互作用などの過程を含む遮蔽されたポテンシャルによるクーロン散乱を考慮する。その他に、イオンは電子の励起やプラズモン生成を引き起こす長距離相互作用を行う。これらの過程は、標的の集団的な電子的構造と原子間結合についての情報を取り入れることで計算開始時に設定される。標的中のイオンの荷電状態は実行電荷の考え方をもとに記述される。この記述には、荷電状態の速度依存性や標的中の集団的な電子の海による長距離遮蔽効果も含まれる。

計算方法に関しての完全な記述は我々の書籍”SRIM - The Stopping and Range of Ions in Solids”, by J. F. Ziegler and J. P. Biersack in 1985 (the Transport of Ions in Matter) に書かれている。この本はイオンの固体への侵入過程の物理が単純な例によって示され、その後 SRIM のソースコードが物理過程の完全な説明とともに記述されている。その後の章では SRIM の正確性と応用事例が述べられる。このウェブサイトでは、SRIM で計算して阻止能のプロットと水素とヘリウムのイオンについての入手可能なすべての実験データをダウンロードすることができる。

TRIM は SRIM に含まれる最も包括的なプログラムである。TRIM では最大 8 層の混合物からなる複雑な形状の標的について計算を行うことができる。このプログラムは終状態におけるイオンの三次元的分布およびイオンのエネルギー損失に伴って発生する標的損傷、スパッタリング、イオン化、フォノン生成といったあらゆる運動過程における現象について計算する。また標的原子の二次的な反応過程も詳細に追跡される。このプログラムは計算過程を任意のタイミングで中断し、またその後中断した時点から再開することができるように作られている。計算結果のプロットは必要に応じて保存したり、計算中に画面に表示することができる。(保存された計算結果を可視化には 5 秒を要する。)

SRIM は J. P. Biersack による飛程計算アルゴリズム (J. P. Biersack and L. Haggmark, Nucl. Instr. and Meth., vol. 174, 257, 1980) と J. F. Ziegler による阻止能に関する理論 (“The Stopping and Range of Ions in Matter”, volumes 2 - 6, Pergamon Press, 1977-1985) が元になっている。SRIM のバージョンによる違いは SRIM パッケージの VERSION というファイルの中に記述されている。

SRIM のプログラムは 1983 年に DOS のプログラムとして始まり、1989 年に Windows に移植された。

もし、あなたが SRIM のプログラムを科学論文に用いた場合、著者に論文をメールで送ってほしい。このことは SRIM の将来的なサポートのために役立つだろう。

以上、<http://www.srim.org/SRIM/SRIMINTRO.htm> の記述を翻訳した。

B SRIM のインストール方法

B.1 概要

SRIM は、SRIM.exe と TRIM.exe という 2 つの計算コードを含んでいる。後述する手順で SRIM をインストールすると SRIM.exe と TRIM.exe を利用できるようになる。

SR.exe は一様な物質中でのイオンのエネルギー損失過程の平均に関する情報を計算することができる計算コードで、TRIM.exe は多層の物質中でのイオンのエネルギー損失過程を詳細に計算するコードである。

また、これらの他に SR.exe と TIN.exe という実行ファイルも含まれている。SRIM.exe は SRIM や TRIM を起動するためのソフトウェアであるが、あまり使用することはないと思われる。TIN.exe は TRIM の入力ファイルを作成するためのソフトウェアである。

B.2 ダウンロード

SRIM は公式のウェブページ <http://www.srim.org/SRIM/SRIMLEGL.htm> からダウンロード可能である。基本的に SRIM-2013 (Professional) をダウンロードすればよい。ダウンロードされるファイルは "SRIM-2013.e" という名前である。このファイルの拡張子を exe に変更して実行すると、SRIM のファイル群が展開される。

B.3 Windows

ファイル群を展開しただけでは、SRIM を正常に動作させることができない。SRIM を使用すると、"RICHTX.OCX" などのプログラムがないことを示すポップアップウィンドウが現れアプリケーションが終了する。この問題に対処する方法は SRIM を展開したディレクトリの "SRIM Setup Message.pdf" に書かれている。このファイルの記述によれば

- MSVBvm50.exe
- ComCtl32.ocx
- ComDlg.ocx
- MSFlxGrd.ocx
- RichtX32.ocx
- TabCtl32.ocx
- LineDraw.ttf

の 7 つのプログラムが必要とのことである。

基本的には、この文章のセットアップ法 #1 に従い、SRIM を展開したディレクトリ配下の "SRIM-Setup/SRIM-AutoSetup" 内にある、"SRIM-AutoSetup.exe" を管理者として実行していけばよい。

B.4 Linux

Linux では Windows を仮想的に動作させるソフトウェアの Wine 上で SRIM を使用方法がある。そのためまず Wine をインストールする。Wine のインストール方法は OS に依存するため、ここでは述べないが基本的にパッケージマネージャからインストールすることができる。

以下で述べる手順はこのような環境でのものである。


```
[quser@oxygen SRIM]$ cat /etc/lsb-release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=18.04
DISTRIB_CODENAME=bionic
DISTRIB_DESCRIPTION="Ubuntu 18.04.5 LTS"
[quser@oxygen SRIM]$ which wine
/usr/bin/wine
wine-3.0 (Ubuntu 3.0-1ubuntu1)
```

基本的には Windows の場合と同様に必要なファイルをインストールしていくことになる。Wine がインストールされている環境ではコマンドラインで

```
$ wine (実行ファイル)
```

とすることで仮想 Windows 環境からそのファイルを実行することができる。

まず、任意の場所に SRIM をダウンロードする。以下では例えば、" quser/opt/SRIM" とする。

```
$ cd quser/opt/SRIM
$ wget http://www.srim.org/SRIM/SRIM-2013-Pro.e
$ mv SRIM-2013-Pro.e SRIM-2013-Pro.exe
$ wine SRIM-2013-Pro.exe
```

とするとポップアップウィンドウが現れるので、Extract で圧縮されているファイルを展開し、Done で終了する。

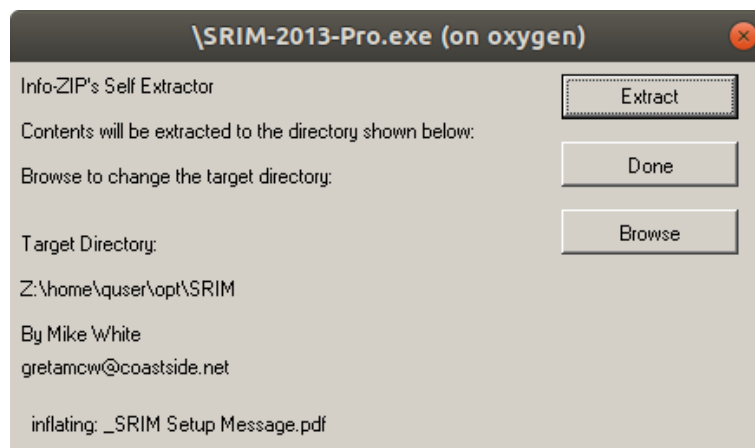


図 2: TRIM の入力ファイル生成用ソフトウェア TIN.exe の画面

その後必要なプログラムをインストールする。Windows の場合とは違って" Auto-Setup" がうまくできなかったの、手動でプログラムをインストールする。インストールすべきプログラムは、SRIM のディレクトリ配下の" SRIM-Setup" に用意されているので、以下の手順の前にそこに移動しておく。

```
$ cd SRIM-Setup
```

そして、Visual Studio のランタイムパッケージをインストールする。

```
$ wine MSVBvm50.exe
```

これを実行すると、図3のようなウインドウが現れるので、“Yes”を選択する。インストールが終わると、図4のようなウインドウが現れるので“OK”を選択し終了する。

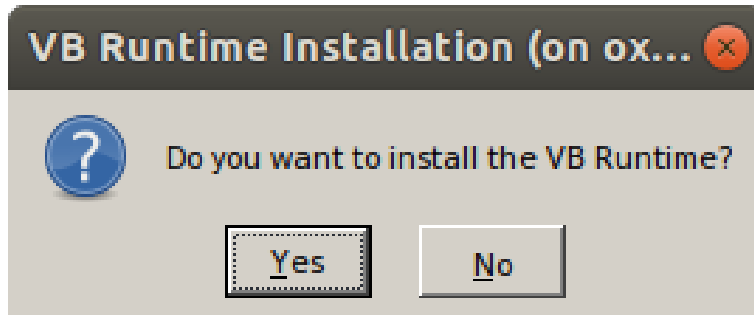


図 3: MSBvm50.exe のインストールを確認するポップアップウインドウ。

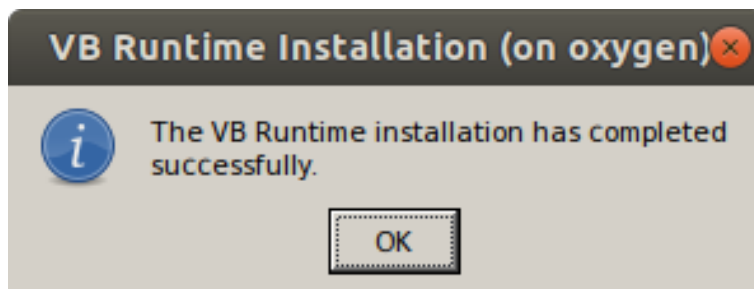


図 4: MSBvm50.exe のインストール完了時にあらわれるポップアップウインドウ。

最後に、必要な OCX ファイルをインストールする。

```
$ wine ../../.wine/drive_c/windows/syswow64/regsvr32.exe RichTx32.ocx
$ wine ../../.wine/drive_c/windows/syswow64/regsvr32.exe ComDlg32.ocx
$ wine ../../.wine/drive_c/windows/syswow64/regsvr32.exe MSFlxGrd.ocx
$ wine ../../.wine/drive_c/windows/syswow64/regsvr32.exe ComCtl32.ocx
$ wine ../../.wine/drive_c/windows/syswow64/regsvr32.exe TabCtl32.ocx
```

これらのファイルのインストールが成功すると

```
regsvr32: Successfully registered DLL 'TabCtl32.ocx'
```

というようなメッセージがかえってくる。

ここまでの手順を終えると、SRIMを展開したディレクトリで

```
$ wine SRIM.exe
```

などとすれば、SRIM や TRIM を使用することができるはずである。

C TRIM の使い方

C.1 入力ファイルの生成

TRIM の計算条件は TRIM のインストール先ディレクトリにある TRIM.IN というテキストファイルに書き込む。これを編集するためのアプリケーションが TIN.exe である。TIN.exe の画面を図 5 に示す。

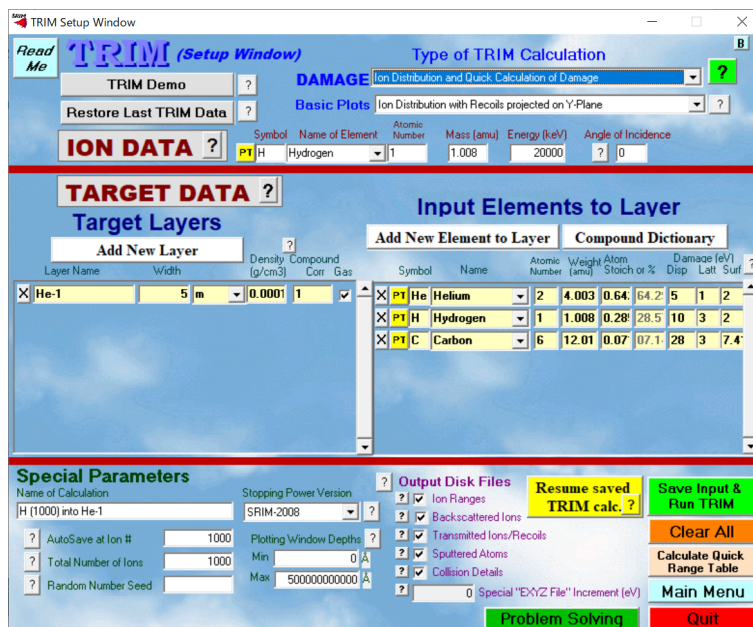


図 5: TRIM の入力ファイル生成用ソフトウェア TIN.exe の画面

TrackTrimSQLite の入力として用いる場合、TIN.exe ウィンドウの右上 "DAMAGE" の項目を "Ion Distribution and Quick Calculation of Damage" に設定して、ウィンドウ右下の "Output Disk Files" のすべてにチェックをしておく必要がある。また、媒質の層の厚さを表す "Width" を粒子の飛程に比べて十分大きくしておく必要がある。

TIN.exe は日本語の Windows 環境で頻繁に異常な動作を示す。その一例として、数値を入力するためのフィールドが点滅し操作不能になることがある。この場合、Win+d を二回続けて入力すると元の状態に復帰できる。しかしこの異常は非常に頻繁に発生するので、TIN.exe を用いて入力ファイルを作成することは推奨できない。その代わりにテキストエディタから TRIM.IN を直接編集し、動作の確認のみを TIN.exe で行うほうが素早く入力ファイルを編集することができる。

生成された TRIM.IN を以下に示す。

Listing 4: TRIM.IN の例。

```
1 ==> SRIM-2013.00 This file controls TRIM Calculations.
2 Ion: Z1 , M1, Energy (keV), Angle, Number, Bragg Corr, AutoSave Number.
3 1 1.008 20000 0 1000 1 1000
4 Cascades (1=No;2=Full;3=Sputt;4-5=Ions;6-7=Neutrons), Random Number Seed, Reminders
5 1 0
6 Diskfiles (0=no,1=yes): Ranges, Backscatt, Transmit, Sputtered, Collisions(1=Ion;2=Ion+Recoils), Special EXYZ.txt
7 file 1 1 1 1 2 0
8 Target material : Number of Elements & Layers 3 1
9 "H (1000) into He-1"
10 PlotType (0-5); Plot Depths: Xmin, Xmax(Ang.) [=0 0 for Viewing Full Target]
11 0 5E+11
12 Target Elements: Z Mass(amu)
13 Atom 1 = He = 2 4.003
14 Atom 2 = H = 1 1.008
15 Atom 3 = C = 6 12.011
16 Layer Layer Name / Width Density He(2) H(1) C(6)
17 Numb. Description (Ang) (g/cm3) Stoich Stoich Stoich
18 1 "He-1" 5E+11 .000104 .6429 .2857 .0714
19 0 Target layer phases (0=Solid, 1=Gas)
```

```

20 1
21 Target Compound Corrections (Bragg)
22 1
23 Individual target atom displacement energies (eV)
24 5 10 28
25 Individual target atom lattice binding energies (eV)
26 1 3 3
27 Individual target atom surface binding energies (eV)
28 2 2 7.41
29 Stopping Power Version (1=2011, 0=2011)
30 0

```

実用的には同一ガス組成でいろいろな種類の粒子の飛跡を計算したいと思われる。その場合、TRIM.IN のガス組成さえ正しく設定しておけば、TRIM.IN の 3 行目で粒子の種類を、目的によっては 18 行目で粒子の変更するだけで良いと思われる。

C.2 計算

TRIM の計算は TRIM.exe を用いて行う。TRIM.exe を実行すると TRIM.IN を入力として直ちに計算が開始される。図に TRIM.exe の実行中の画面を図 6 に示す。計算が終了すると、図 7 のようにポップアップが現れる。ポップアップするウインドウを拡大したものが図 8 である。ここで”はい”を選択して計算結果を保存する。すると、保存先を尋ねる図 9 のようなウインドウが現れる。ここで”Store in SRIM directory”を選択すると結果のテキストファイルは SRIM のディレクトリの下にある”SRIM Outputs”に保存される。

計算結果ファイルの最終行で正しく終端処理を行うためには TRIM のウインドウを閉じる必要がある。TRIM のウインドウの右上のボタンを押すと、図 10 のようなウインドウが現れる。計算結果を使うだけであればどちらを選んでも差し支えない。

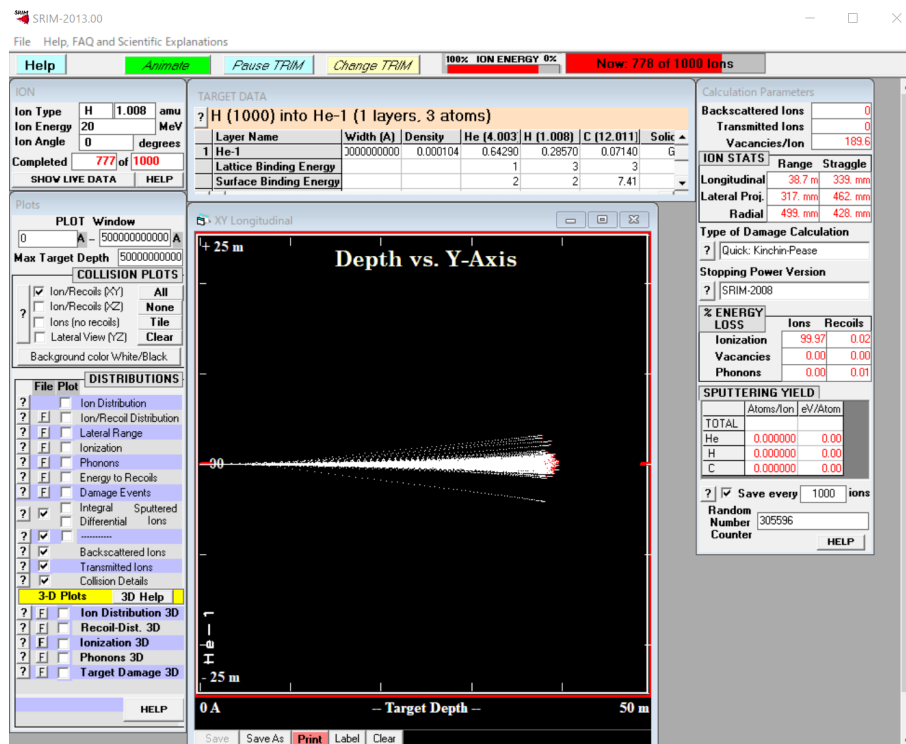


図 6: TRIM の計算途中の様子。計算の進捗とともに右上のプログレスバーが右に動いている。

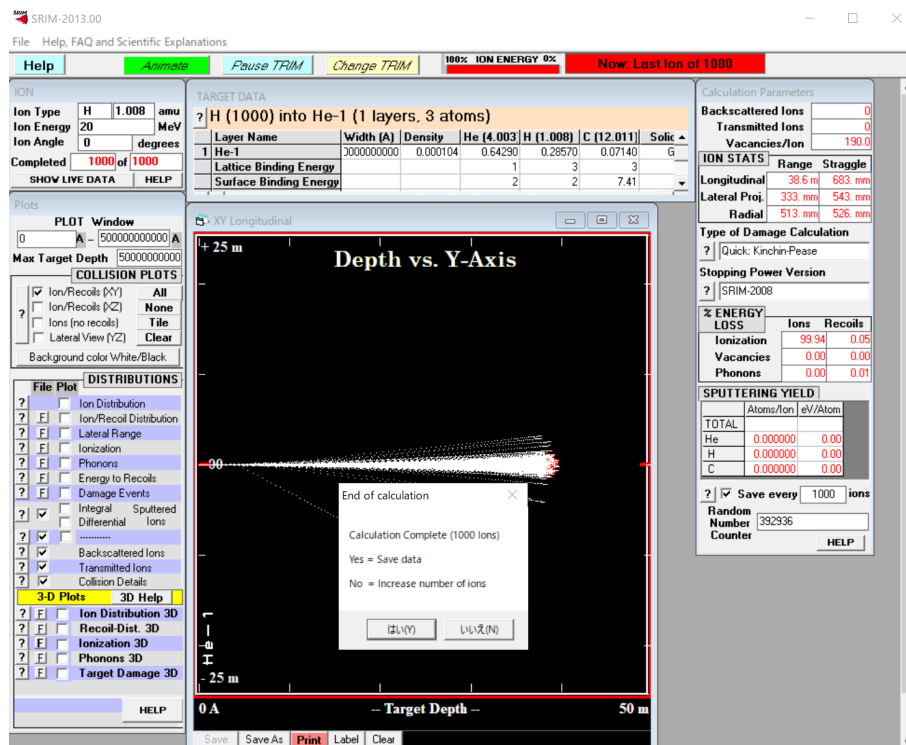


図 7: TRIM の計算終了時の様子。計算後の処理を尋ねるウィンドウがポップアップする。

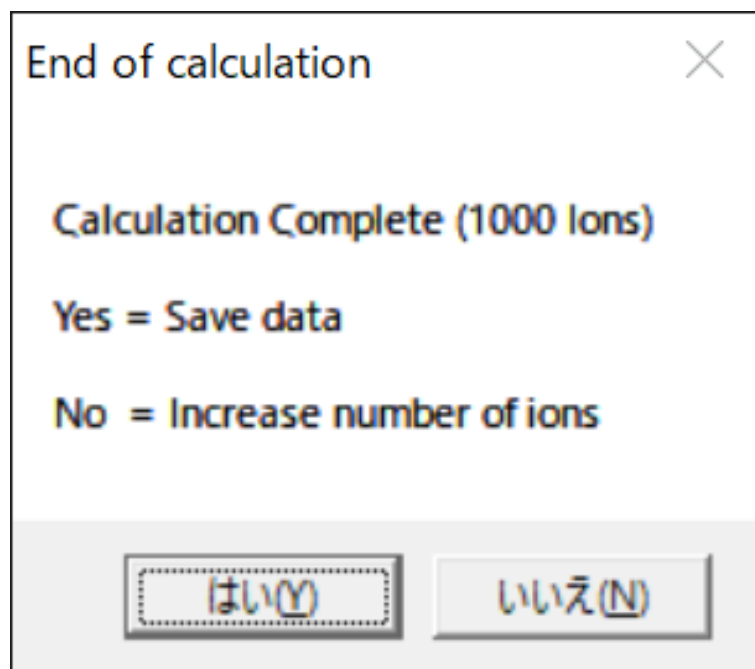


図 8: TRIM の計算終了時にポップアップするウィンドウ。”はい”を選択し、計算結果を保存する。

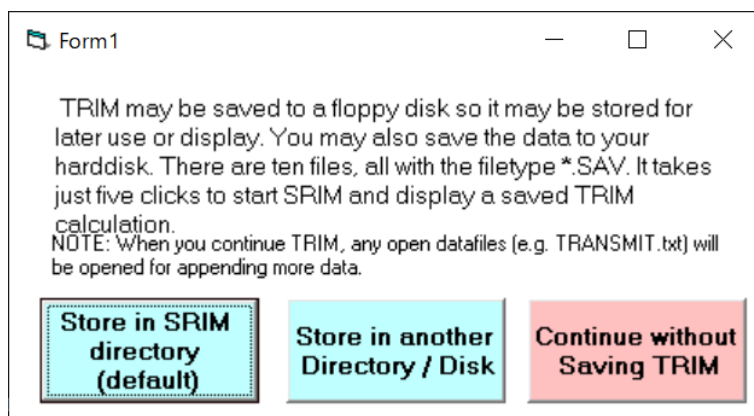


図 9: TRIM 計算結果の保存先ディレクトリを指定するためのウインドウ。図 8 で計算結果を保存することを選択するとポップアップする。

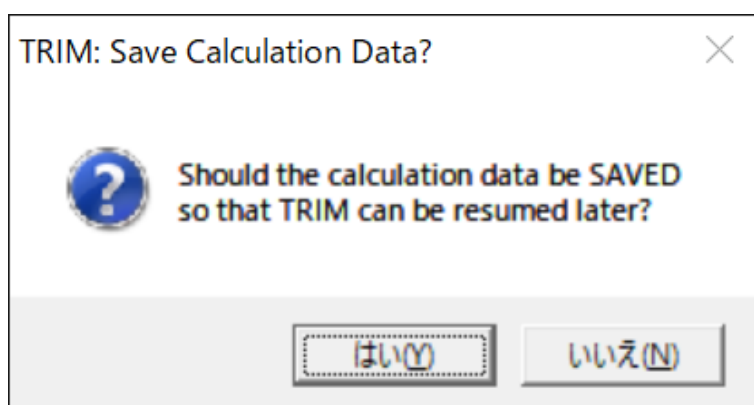


図 10: TRIM のウインドウを右上のバツボタンで閉じるときに現れるウインドウ。“いいえ”を選択して問題ない。

D TRIM の出力ファイルの形式

TRIM では計算結果をいくつかのテキストファイルに保存することができる。以下では TrackTrimSQLite の入力ファイルを作るために必要な 2 つのファイル COLLISON.txt と RANGE_3D.txt について述べる。

D.1 COLLISON.txt

COLLISON.txt はイオンの物質中での衝突事象を記録している。なお、COLLISON.txt の形式は指定した TRIM の計算モデルによって変わるが、TrackTRIMSQLite が対応する形式は “DAMAGE” として “Ion Distribution and Quick Calculation of Damage” を選んだフォーマットである。

リストに COLLISON.txt の例を示す。COLLISON.txt のはじめには計算条件を記録しているヘッダーがあり、その後各イベントの衝突履歴が示されている。リストの 35 行目以降の各行が一つの衝突に対応しており、値の凡例は 34 行目にある。各行をよく観察するとすべての行の同じ列

に数字の3がつらなっていることがわかる。これは、なぜか区切り文字が数字の3になっているためであると思われる。

衝突事象を記述する値の読み取り方を最初の衝突事象の35行目を一例として解説する。区切り文字3に注目すると、この行の衝突事象はイオン番号が1、エネルギーが98.22E+02 (keV)、深さが62396.E+04 (Å)、yが-2698.E-03 (Å)、zが1889.E-03 (Å)、Seが0 (eV/Å)、衝突した原子がHe、反跳エネルギーが80472.E-04 (eV) とわかる。a

Listing 5: COLLISION.txt の例。1 イベントのみ抽出。

```

1 ===== SRIM-2013.00 =====
2
3 -----
4 ----- TRIM Calc. = H(10 MeV) ==> He-1( 3 m) -----
5 -----
6
7 o Ion Name = H o
8 o Ion Mass = 003.016 amu o
9 o Ion Energy = 000010000.0 keV o
10
11 > He Displacement Energy of He = 0005.00000 eV
12 Latt.Binding Energy of He = 0001.00000 eV
13 SurfaceBind. Energy of He = 0002.00000 eV
14 > H Displacement Energy of H = 0010.00000 eV
15 Latt.Binding Energy of H = 0003.00000 eV
16 SurfaceBind. Energy of H = 0002.00000 eV
17 > C Displacement Energy of C = 0028.00000 eV
18 Latt.Binding Energy of C = 0003.00000 eV
19 SurfaceBind. Energy of C = 0007.41000 eV
20
21 =====
22 3=> Recoils Calculated with Kinchin-Pease Theory (Only Vacancies Calc) <=3
23 =====
24
25 ===== COLLISION HISTORY =====
26 NOTES: Only Ion Collisions which produce Displacements are tabulated.
27 Atom Sums and Averages are incomplete if Recoil Cascades Leave Target.
28 Target Displacements = VACancies + REPLACement Collisions.
29 Target VACancies = INTERstitials + Sputtered + Transmitted Atoms.
30 Recoil Atoms which end at the surface, are not counted (see manual).
31 =====
32 Ion Energy Depth Lateral Distance (Å) Se Atom Recoil Target Target Target Target
33 Numb (keV) (Å) Y Axis Z Axis (eV/Å) Hit Energy(eV) DISP. VAC. REPLAC INTER
34
35 300001398. 44E+02362396.E+043 -2698.E-033 1889.E-0330000.003 He 380472.E-0430000000001.0003 3 3
36 300001396. 93E+02312230.E+053 5829.E+023 -2228.E+0230000.003 He 314007.E-0330000000001.0003 3 3
37 300001394. 55E+02322012.E+053 2572.E+033 1754.E+0230000.003 C 310852.E-0230000000001.1543 3 3
38 300001390. 24E+0237691.E+053 1053.E+043 1369.E+0430000.003 He 328631.E-0330000000001.8753 3 3
39 300001387. 97E+02345808.E+053 1412.E+043 2032.E+0430000.003 He 311376.E-0330000000001.0003 3 3
40 300001383. 64E+02361476.E+053 1959.E+043 3192.E+0430000.003 He 311216.E-0330000000001.0003 3 3
41 300001381. 51E+02369033.E+053 2225.E+043 3840.E+0430000.003 He 335646.E-0330000000002.3083 3 3
42 300001379. 34E+02376408.E+053 2643.E+043 4393.E+0430000.003 He 310326.E-0330000000001.0003 3 3
43 300001377. 24E+02383599.E+053 2958.E+043 4952.E+0430000.003 He 315785.E-0330000000001.0613 3 3
44 300001375. 23E+02390612.E+053 3183.E+043 5416.E+0430000.003 He 368992.E-0330000000004.2793 3 3
45 300001373. 05E+02397454.E+063 6477.E+043 1039.E+0530000.003 He 311900.E-0330000000001.0003 3 3
46 300001369. 03E+02311059.E+063 3304.E+043 6658.E+0430000.003 He 361495.E-0430000000001.0003 3 3
47 300001367. 01E+02311690.E+063 3481.E+043 6966.E+0430000.003 He 310805.E-0330000000001.0003 3 3
48 300001362. 77E+02312900.E+063 4169.E+043 7678.E+0430000.003 H 366835.E-0330000000002.0783 3 3
49 300001360. 81E+02313477.E+063 4620.E+043 8183.E+0430000.003 He 364959.E-0430000000001.0003 3 3
50 300001356. 63E+02314580.E+063 5370.E+043 9142.E+0430000.003 He 370799.E-0430000000001.0003 3 3
51 300001354. 56E+02315104.E+063 5736.E+043 9517.E+0430000.003 H 317956.E-0330000000001.0003 3 3
52 300001352. 62E+02315611.E+063 6086.E+043 9932.E+0430000.003 He 393246.E-0430000000001.0003 3 3
53 300001350. 60E+02316100.E+063 6477.E+043 1039.E+0530000.003 He 311900.E-0330000000001.0003 3 3
54 300001348. 60E+02316572.E+063 6893.E+043 1090.E+0530000.003 He 314230.E-0330000000001.0003 3 3
55 300001346. 54E+02317027.E+063 7375.E+043 1136.E+0530000.003 He 310430.E-0330000000001.0003 3 3
56 300001344. 53E+02317464.E+063 7895.E+043 1175.E+0530000.003 H 312117.E-0330000000001.0003 3 3
57 300001342. 59E+02317883.E+063 8385.E+043 1208.E+0530000.003 He 322864.E-0330000000001.5133 3 3
58 300001336. 77E+02319038.E+063 9983.E+043 1324.E+0530000.003 He 385278.E-0430000000001.0003 3 3
59 300001334. 81E+02319389.E+063 1046.E+053 1364.E+0530000.003 He 390462.E-0430000000001.0003 3 3
60 300001333. 00E+02319722.E+063 1095.E+053 1408.E+0530000.003 C 375373.E-0330000000001.0003 3 3
61 300001331. 11E+02320039.E+063 1163.E+053 1470.E+0530000.003 He 366170.E-0430000000001.0003 3 3
62 300001329. 34E+02320339.E+063 1228.E+053 1524.E+0530000.003 He 310721.E-0330000000001.0003 3 3
63 300001327. 45E+02320624.E+063 1295.E+053 1574.E+0530000.003 He 342036.E-0330000000002.6973 3 3
64 300001325. 59E+02320892.E+063 1348.E+053 1617.E+0530000.003 He 323175.E-0330000000001.5333 3 3
65 300001323. 87E+02321142.E+063 1393.E+053 1649.E+0530000.003 H 331433.E-0330000000001.0243 3 3
66 300001322. 08E+02321378.E+063 1434.E+053 1674.E+0530000.003 H 315810.E-0330000000001.0003 3 3
67 300001320. 28E+02321597.E+063 1473.E+053 1695.E+0530000.003 C 351155.E-0230000000004.1603 3 3
68 300001318. 57E+02321799.E+063 1488.E+053 1654.E+0530000.003 C 366985.E-0330000000001.0003 3 3
69 300001316. 84E+02321986.E+063 1480.E+053 1608.E+0530000.003 He 346258.E-0330000000002.9503 3 3
70 300001315. 20E+02322157.E+063 1484.E+053 1566.E+0530000.003 He 391076.E-0430000000001.0003 3 3
71 300001313. 53E+02322313.E+063 1484.E+053 1531.E+0530000.003 H 387675.E-0330000000002.6653 3 3
72 300001311. 92E+02322453.E+063 1490.E+053 1497.E+0530000.003 He 313175.E-0330000000001.0003 3 3
73 300001310. 34E+02322577.E+063 1494.E+053 1462.E+0530000.003 He 314706.E-0330000000001.0003 3 3
74 300001388. 25E+01322687.E+063 1492.E+053 1429.E+0530000.003 He 393978.E-0430000000001.0003 3 3
75 300001373. 77E+01322782.E+063 1494.E+053 1401.E+0530000.003 He 331347.E-0330000000002.0443 3 3
76 300001360. 31E+01322863.E+063 1501.E+053 1374.E+0530000.003 He 320743.E-0330000000001.3783 3 3
77 300001347. 93E+01322930.E+063 1504.E+053 1355.E+0530000.003 H 326870.E-0330000000001.0003 3 3
78 300001336. 05E+01322986.E+063 1505.E+053 1340.E+0530000.003 He 323223.E-0330000000001.5363 3 3
79 300001325. 95E+01323028.E+063 1507.E+053 1324.E+0530000.003 He 315798.E-0330000000001.0613 3 3
80 300001317. 93E+01323060.E+063 1507.E+053 1315.E+0530000.003 He 318064.E-0330000000001.2073 3 3
81 300001312. 15E+01323084.E+063 1507.E+053 1306.E+0530000.003 He 312918.E-0330000000001.0003 3 3
82 300001383. 81E+00323100.E+063 1509.E+053 1298.E+0530000.003 He 311847.E-0330000000001.0003 3 3
83 300001356. 99E+00323113.E+063 1511.E+053 1291.E+0530000.003 He 348366.E-0230000000022.3573 3 3
84 300001340. 17E+00323121.E+063 1512.E+053 1276.E+0530000.003 He 312957.E-0330000000001.0003 3 3
85 300001329. 45E+00323128.E+063 1511.E+053 1265.E+0530000.003 He 344760.E-0330000000002.8613 3 3
86 300001324. 27E+00323133.E+063 1512.E+053 1255.E+0530000.003 He 322432.E-0330000000001.4863 3 3
87 300001319. 16E+00323137.E+063 1513.E+053 1244.E+0530000.003 He 317112.E-0330000000001.1463 3 3
88 300001316. 13E+00323141.E+063 1512.E+053 1236.E+0530000.003 He 334851.E-0330000000002.2593 3 3
89 300001313. 23E+00323144.E+063 1511.E+053 1230.E+0530000.003 He 323355.E-0330000000001.5443 3 3
90 300001311. 90E+00323146.E+063 1511.E+053 1226.E+0530000.003 He 323969.E-0330000000001.5833 3 3
91 300001310. 75E+00323149.E+063 1511.E+053 1221.E+0530000.003 H 347450.E-0330000000001.5113 3 3
92 300001399. 01E-01323151.E+063 1509.E+053 1217.E+0530000.003 He 362458.E-0330000000003.9033 3 3
93 300001392. 77E-01323153.E+063 1497.E+053 1214.E+0530000.003 He 323606.E-0330000000001.5603 3 3
94 300001389. 84E-01323155.E+063 1504.E+053 1213.E+0530000.003 He 390431.E-0330000000005.4833 3 3
95 300001387. 43E-01323157.E+063 1502.E+053 1214.E+0530000.003 He 349887.E-0330000000003.1663 3 3
96 300001376. 43E-01323159.E+063 1501.E+053 1217.E+0530000.003 C 349210.E-0330000000001.0003 3 3
97 300001374. 51E-01323161.E+063 1499.E+053 1222.E+0530000.003 He 336357.E-0330000000002.3523 3 3
98 300001368. 80E-01323163.E+063 1499.E+053 1226.E+0530000.003 He 323683.E-0330000000001.5653 3 3
99 300001363. 28E-01323164.E+063 1498.E+053 1232.E+0530000.003 He 376364.E-0330000000004.6983 3 3
100 300001352. 09E-01323166.E+063 1498.E+053 1238.E+0530000.003 He 320637.E-0330000000001.3723 3 3
101 300001345. 32E-01323167.E+063 1498.E+053 1245.E+0530000.003 He 323606.E-0330000000001.5603 3 3
102 300001338. 12E-01323168.E+063 1499.E+053 1250.E+0530000.003 He 313463.E-0330000000001.0003 3 3
103 300001332. 35E-01323169.E+063 1500.E+053 1254.E+0530000.003 He 352448.E-0330000000003.3183 3 3
104 300001328. 92E-01323170.E+063 1501.E+053 1256.E+0530000.003 H 313565.E-0330000000001.0003 3 3
105 300001328. 79E-01323171.E+063 1502.E+053 1259.E+0530000.003 He 313277.E-0330000000001.0003 3 3

```



```

7
8 Ion = Hydrogen [1] , Mass = 3.016 amu
9
10 Target Density = 2.0800E-04 g/cm3 = 3.3680E+19 atoms/cm3
11 Target is a GAS
12 ===== Target Composition =====
13 Atom Atom Atomic Mass
14 Name Numb Percent Percent
15 ---
16 He 2 064.29 069.20
17 H 1 028.57 007.74
18 C 6 007.14 023.06
19 -----
20 Bragg Correction = 0.48%
21 Stopping Units = MeV / (mg/cm2)
22 See bottom of Table for other Stopping units
23
24 Ion dE/dx dE/dx Projected Longitudinal Lateral
25 Energy Elec. Nuclear Range Straggling Straggling
26 -----
27 10.00 keV 2.289E-01 2.616E-02 2.08 mm 421.24 um 454.76 um
28 11.00 keV 2.460E-01 2.442E-02 2.25 mm 437.90 um 480.02 um
29 12.00 keV 2.631E-01 2.291E-02 2.41 mm 452.47 um 502.82 um
30 13.00 keV 2.803E-01 2.160E-02 2.57 mm 465.28 um 523.47 um
31 14.00 keV 2.975E-01 2.044E-02 2.71 mm 476.60 um 542.24 um
32 15.00 keV 3.149E-01 1.942E-02 2.85 mm 486.66 um 559.35 um
33 16.00 keV 3.323E-01 1.850E-02 2.98 mm 495.63 um 574.99 um
34 17.00 keV 3.499E-01 1.767E-02 3.11 mm 503.67 um 589.34 um
35 18.00 keV 3.675E-01 1.692E-02 3.23 mm 510.90 um 602.53 um
36 20.00 keV 4.029E-01 1.562E-02 3.46 mm 523.79 um 625.91 um
37 22.50 keV 4.475E-01 1.427E-02 3.72 mm 537.05 um 650.55 um
38 25.00 keV 4.921E-01 1.315E-02 3.96 mm 547.71 um 671.19 um
39 27.50 keV 5.363E-01 1.221E-02 4.18 mm 556.44 um 688.70 um
40 30.00 keV 5.794E-01 1.141E-02 4.39 mm 563.70 um 703.74 um
41 32.50 keV 6.211E-01 1.071E-02 4.58 mm 569.83 um 716.80 um
42 35.00 keV 6.607E-01 1.010E-02 4.76 mm 575.09 um 728.26 um
43 37.50 keV 6.978E-01 9.566E-03 4.93 mm 579.66 um 738.44 um
44 40.00 keV 7.318E-01 9.087E-03 5.10 mm 583.68 um 747.56 um
45 45.00 keV 7.899E-01 8.271E-03 5.40 mm 591.26 um 763.33 um
46 50.00 keV 8.352E-01 7.600E-03 5.69 mm 597.58 um 776.68 um
47 55.00 keV 8.700E-01 7.037E-03 5.97 mm 603.02 um 788.27 um
48 60.00 keV 8.976E-01 6.558E-03 6.23 mm 607.84 um 798.56 um
49 65.00 keV 9.211E-01 6.144E-03 6.49 mm 612.17 um 807.83 um
50 70.00 keV 9.406E-01 5.783E-03 6.75 mm 616.13 um 816.27 um
51 80.00 keV 9.838E-01 5.184E-03 7.24 mm 625.03 um 831.19 um
52 90.00 keV 1.024E+00 4.705E-03 7.71 mm 632.78 um 844.04 um
53 100.00 keV 1.063E+00 4.312E-03 8.17 mm 639.64 um 855.31 um
54 110.00 keV 1.099E+00 3.984E-03 8.61 mm 645.78 um 865.33 um
55 120.00 keV 1.131E+00 3.705E-03 9.04 mm 651.36 um 874.35 um
56 130.00 keV 1.159E+00 3.466E-03 9.45 mm 656.48 um 882.56 um
57 140.00 keV 1.183E+00 3.257E-03 9.86 mm 661.25 um 890.12 um
58 150.00 keV 1.202E+00 3.074E-03 10.26 mm 665.72 um 897.14 um
59 160.00 keV 1.217E+00 2.912E-03 10.66 mm 669.97 um 903.72 um
60 170.00 keV 1.229E+00 2.767E-03 11.05 mm 674.02 um 909.91 um
61 180.00 keV 1.238E+00 2.636E-03 11.44 mm 677.92 um 915.80 um
62 200.00 keV 1.247E+00 2.412E-03 12.21 mm 689.68 um 926.80 um
63 225.00 keV 1.247E+00 2.182E-03 13.17 mm 706.36 um 939.48 um
64 250.00 keV 1.237E+00 1.995E-03 14.13 mm 722.58 um 951.31 um
65 275.00 keV 1.221E+00 1.839E-03 15.11 mm 738.60 um 962.57 um
66 300.00 keV 1.200E+00 1.707E-03 16.10 mm 754.62 um 973.47 um
67 325.00 keV 1.176E+00 1.594E-03 17.11 mm 770.78 um 984.02 um
68 350.00 keV 1.150E+00 1.496E-03 18.14 mm 787.16 um 994.46 um
69 375.00 keV 1.124E+00 1.410E-03 19.19 mm 803.85 um 1.00 mm
70 400.00 keV 1.098E+00 1.333E-03 20.27 mm 820.89 um 1.02 mm
71 450.00 keV 1.046E+00 1.204E-03 22.51 mm 884.92 um 1.04 mm
72 500.00 keV 9.967E-01 1.099E-03 24.86 mm 950.29 um 1.06 mm
73 550.00 keV 9.506E-01 1.012E-03 27.33 mm 1.02 mm 1.08 mm
74 600.00 keV 9.080E-01 9.386E-04 29.91 mm 1.09 mm 1.10 mm
75 650.00 keV 8.687E-01 8.754E-04 32.61 mm 1.16 mm 1.13 mm
76 700.00 keV 8.325E-01 8.206E-04 35.43 mm 1.23 mm 1.15 mm
77 800.00 keV 7.686E-01 7.303E-04 41.44 mm 1.50 mm 1.20 mm
78 900.00 keV 7.142E-01 6.587E-04 47.92 mm 1.77 mm 1.26 mm
79 1.00 MeV 6.674E-01 6.006E-04 54.88 mm 2.03 mm 1.33 mm
80 1.10 MeV 6.268E-01 5.523E-04 62.30 mm 2.30 mm 1.40 mm
81 1.20 MeV 5.913E-01 5.116E-04 70.19 mm 2.57 mm 1.47 mm
82 1.30 MeV 5.599E-01 4.768E-04 78.54 mm 2.84 mm 1.55 mm
83 1.40 MeV 5.320E-01 4.466E-04 87.34 mm 3.11 mm 1.64 mm
84 1.50 MeV 5.070E-01 4.202E-04 96.59 mm 3.38 mm 1.74 mm
85 1.60 MeV 4.845E-01 3.969E-04 106.28 mm 3.66 mm 1.83 mm
86 1.70 MeV 4.640E-01 3.761E-04 116.42 mm 3.94 mm 1.94 mm
87 1.80 MeV 4.454E-01 3.576E-04 126.98 mm 4.23 mm 2.05 mm
88 2.00 MeV 4.127E-01 3.257E-04 149.39 mm 5.31 mm 2.29 mm
89 2.25 MeV 3.784E-01 2.933E-04 179.79 mm 6.86 mm 2.62 mm
90 2.50 MeV 3.498E-01 2.671E-04 212.80 mm 8.33 mm 2.98 mm
91 2.75 MeV 3.255E-01 2.453E-04 248.40 mm 9.76 mm 3.37 mm
92 3.00 MeV 3.045E-01 2.270E-04 286.56 mm 11.19 mm 3.79 mm
93 3.25 MeV 2.874E-01 2.113E-04 327.17 mm 12.61 mm 4.24 mm
94 3.50 MeV 2.705E-01 1.978E-04 370.25 mm 14.04 mm 4.72 mm
95 3.75 MeV 2.554E-01 1.859E-04 415.95 mm 15.50 mm 5.23 mm
96 4.00 MeV 2.432E-01 1.755E-04 464.14 mm 16.97 mm 5.76 mm
97 4.50 MeV 2.221E-01 1.579E-04 567.54 mm 22.48 mm 6.91 mm
98 5.00 MeV 2.047E-01 1.437E-04 680.22 mm 27.64 mm 8.16 mm
99 5.50 MeV 1.902E-01 1.319E-04 802.00 mm 32.65 mm 9.51 mm
100 6.00 MeV 1.778E-01 1.219E-04 932.67 mm 37.61 mm 10.94 mm
101 6.50 MeV 1.670E-01 1.134E-04 1.07 m 42.56 mm 12.47 mm
102 7.00 MeV 1.577E-01 1.061E-04 1.22 m 47.54 mm 14.09 mm
103 8.00 MeV 1.420E-01 9.407E-05 1.54 m 65.95 mm 17.59 mm
104 9.00 MeV 1.294E-01 8.457E-05 1.90 m 83.04 mm 21.43 mm
105 10.00 MeV 1.191E-01 7.688E-05 2.28 m 99.69 mm 25.60 mm
106 11.00 MeV 1.104E-01 7.052E-05 2.70 m 116.23 mm 30.09 mm
107 12.00 MeV 1.029E-01 6.517E-05 3.15 m 132.83 mm 34.90 mm
108 13.00 MeV 9.653E-02 6.061E-05 3.64 m 149.58 mm 40.02 mm
109 14.00 MeV 9.093E-02 5.666E-05 4.15 m 166.54 mm 45.45 mm
110 15.00 MeV 8.600E-02 5.322E-05 4.69 m 183.74 mm 51.17 mm
111 16.00 MeV 8.162E-02 5.019E-05 5.26 m 201.20 mm 57.20 mm
112 17.00 MeV 7.770E-02 4.750E-05 5.87 m 218.92 mm 63.52 mm
113 18.00 MeV 7.417E-02 4.509E-05 6.50 m 236.91 mm 70.13 mm
114 20.00 MeV 6.806E-02 4.096E-05 7.85 m 305.30 mm 84.27 mm
115 22.50 MeV 6.180E-02 3.679E-05 9.71 m 403.28 mm 103.44 mm
116 25.00 MeV 5.668E-02 3.342E-05 11.74 m 496.05 mm 124.38 mm
117 27.50 MeV 5.241E-02 3.063E-05 13.94 m 586.97 mm 147.02 mm
118 30.00 MeV 4.878E-02 2.829E-05 16.32 m 677.54 mm 171.33 mm
119 32.50 MeV 4.566E-02 2.630E-05 18.86 m 768.51 mm 197.27 mm
120 35.00 MeV 4.295E-02 2.457E-05 21.58 m 860.32 mm 224.84 mm
121 37.50 MeV 4.057E-02 2.307E-05 24.46 m 953.22 mm 253.99 mm
122 40.00 MeV 3.846E-02 2.174E-05 27.50 m 1.05 m 284.71 mm
123 45.00 MeV 3.488E-02 1.952E-05 34.06 m 1.40 m 350.81 mm
124 50.00 MeV 3.197E-02 1.772E-05 41.26 m 1.74 m 422.96 mm
125 55.00 MeV 2.954E-02 1.623E-05 49.08 m 2.06 m 501.04 mm
126 60.00 MeV 2.749E-02 1.499E-05 57.51 m 2.39 m 584.92 mm
127 65.00 MeV 2.572E-02 1.392E-05 66.55 m 2.71 m 674.50 mm
128 70.00 MeV 2.420E-02 1.301E-05 76.18 m 3.04 m 769.66 mm
129 80.00 MeV 2.167E-02 1.150E-05 97.17 m 4.26 m 976.42 mm
130 90.00 MeV 1.967E-02 1.032E-05 120.46 m 5.39 m 1.20 m
131 100.00 MeV 1.804E-02 9.363E-06 145.98 m 6.50 m 1.45 m
132 -----

```

```

133 Multiply Stopping by          for Stopping Units
134 -----
135 2.0799E-03                    eV / Angstrom
136 2.0799E-02                    keV / micron
137 2.0799E-02                    MeV / mm
138 1.0000E+00                    keV / (ug/cm2)
139 1.0000E+00                    MeV / (mg/cm2)
140 1.0000E+03                    keV / (mg/cm2)
141 6.1756E+00                    eV / (1E15 atoms/cm2)
142 1.3108E+00                    L.S.S. reduced units
143 -----
144 (C) 1984,1989,1992,1998,2008 by J.P. Biersack and J.F. Ziegler

```

F SQLite

SQLiteは開発が続けられているライブラリであり、以下のような特徴を備えたSQLデータベースを実装している。

- 単体で使用可能
- サーバーを持たない
- 設定不要
- トランザクション処理が可能

SQLiteのコードはパブリックドメインであるため、商用、私用の別を問わずいかなる目的においても無償で使用可能である。SQLiteは、世界中で最も広く利用されているデータベースであり、いくつかの著名なプロジェクトをふくむ無数のアプリケーションに用いられている。

SQLiteは組み込み型SQLデータベースエンジンである。ほとんどのその他のSQLデータベースと異なり、SQLiteは別途サーバープロセスを必要としない。SQLiteの読み込みと書き込みは通常のデータディスクに直接行われる。複数のテーブルとインデックス、トリガー、ビューからなる完全なSQLデータベースが単一のファイルに含まれている。データベースのファイルフォーマットは環境に依存しないため、32ビット環境と64ビット環境の間やビッグエンディアンとリトルエンディアンのアーキテクチャ間であっても問題なく移植可能である。これらの特徴を有するため、SQLiteはアプリケーションで用いるファイル形式としてよく用いられる。SQLiteデータベースファイルは、米国議会図書館の推奨する記録形式でもある。すなわち、SQLiteはOracleのデータベースの代替品ではなく、`fopen()` 関数に代わるものであると考えるべきである。

SQLiteは軽量ライブラリである。すべての特徴を有効化した場合であっても、動作プラットフォームやコンパイラの最適化オプションの設定を適切にすればライブラリのサイズは600 KiB以下に抑えられる(64ビットコードの方がライブラリのサイズは大きくなる。また、積極的な関数のインライン化やループの展開などのいくつかのコンパイラの最適化によってもライブラリのサイズは大きくなる。)。メモリ使用量と実行速度はトレードオフの関係にある。SQLiteは一般的にいて実行のために割いたメモリ領域が大きいほど高速で実行できる。しかしながら、メモリが小さい環境であっても普通は十分なパフォーマンスを発揮する。使用方法にもよるが、SQLiteはファイルシステムを介した直接のI/Oよりも高速で実行できることがある。

SQLiteはすべてのリリースに先立ち細心の注意を払って試験が尽くされるため、その信頼性には定評がある。ほとんどのSQLiteのソースコードは純粹に試験と検証のために書かれている。自動化された一連のテストによって数億個のSQL文を含む何百万種類ものテストケースが実行され、すべての条件分岐が網羅される。SQLiteはメモリ確保のエラーやディスクI/Oのエラーに対して適切に対応する。トランザクション中にシステムのクラッシュや電源喪失があった場合でもトランザクションはACIDの要件を満たす。これらのことは、システム異常をシミュレーションでき

る特別なテストハーネスをもちいた自動試験によって検証されている。無論これらすべての試験を行ったとしても、バグを完全に排除することはできない。しかし他のよく似たプロジェクト (特に商用の競合プロジェクト) とは異なり、SQLite はバグを秘匿せずすべてのバグに誠実に応答し、バグの一覧と一分ごとのコードの改変の記録を提供する。

SQLite のコードベースは国際的な SQLite 専任の開発者チームにより支援されている。開発者は継続的に SQLite の機能を拡張し、信頼性と性能を向上させ続けている。また、その一方で策定されたインターフェースの仕様と SQL シンタックス、ファイル形式についての後方互換製を維持している。ソースコードは誰に対しても無料で提供されるにもかかわらず、プロによるサポートを受けることも可能である。

SQLite のプロジェクトは 2000 年 5 月 9 日に始まった。断言することはできないが、2050 年まで SQLite のサポートを続ける予定である。あらゆる設計上の決定は、この目的を念頭においた上でなされる。

我々開発者は、高速で信頼性を有し、かつ使用法が容易なより良く美しい製品を作るために、ユーザーに SQLite を活用してもらいたいと考えている。他人を赦すように、自分自身への赦しを乞いなさい。そして、ちょうどあなたが SQLite を無料で入手したように、他人に無償の施しを行い恩に報いなさい。

以上、<https://www.sqlite.org/about.html> の記述を翻訳した。

F.1 SQLite3 の対話型実行環境の使い方

F.1.1 SQLite3 の起動

適切にインストールされている環境であればコマンドラインから

```
$ sqlite3
```

とすることで SQLite3 の対話型実行環境が起動する。データベースのチェックなどは対話型実行環境から行うことができる。対話型環境を終了したい場合は

```
sqlite> .q
```

とすればよい。

対話型環境では SQL 文の他に SQLite 独自のコマンド (. から始まる) を実行することができる。独自コマンドについては、

```
sqlite> .help [コマンド名]
```

でヘルプを表示できる。コマンド名を指定せず実行するとコマンドリストが表示される。

F.1.2 ファイルを開く

コマンドラインから

```
$ sqlite3 [ファイル名]
```

あるいは SQLite3 を起動してから

```
$ sqlite3
sqlite> .open [ファイル名]
```

とする。

F.1.3 開いているファイルの内容を確認する

ファイルのデータベースが持つテーブルの一覧を表示するには、

```
sqlite> .tables
```

とすればよい。また SQLite のテーブルの構造を確認するには、

```
sqlite> .schema
```

とするとテーブルを作成したときの SQL が表示される。

F.2 データを表示する

データの表示には SELECT 文を用いる。ファイルを開いた状態で、

```
sqlite> SELECT [表示したいデータ、複数ある場合カンマ区切り] FROM [テーブル名];
```

とするとテーブル内のすべてのレコードの選択したデータが表示される。データにはカラム名やそれを四則演算で組み合わせたものなどが指定できる。なお、SELECT や FROM などのキーワードは小文字でもよい。

すべてのレコードではなく、特定の条件を満たすレコードのみを抽出する場合は、SELECT 文に WHERE 句をつけて条件を指定する。WHERE 句は WHERE の後に条件式を加えたものである。条件式には基本的な演算と AND などの論理演算が使用できる。等しいという条件を課す際には ==ではなく=を用いることに注意。

```
sqlite> SELECT (中略) FROM [テーブル名] WHERE [条件式];
```

レコードの表示順を昇順または降順にソートする場合は、SELECT 文に ORDER BY 句をつけて条件を指定する。書き方は

```
sqlite> SELECT (中略) FROM [テーブル名] ORDER BY [基準] [昇順または降順];
```

基準になるところにはテーブルのカラムやその四則演算による値が入る。昇順と降順の別は、昇順にソートしたい場合は ASC を降順にソートしたい場合は DESC を指定する。順番の指定がない場合、昇順にソートされる。また、WHERE 句と ORDER BY 句は同時に使用することもできる。

例えば、テーブル collisions から track.id が 10 のレコードの x と e.inc-e.rec の値を collision.id の降順でソートして出力したい場合には

```
sqlite> SELECT x, e.inc - e.rec FROM collisions WHERE track_id = 10
ORDER BY collision_id DESC;
```

とする。

F.3 外部ファイルにデータを書き出す

テーブルのデータを外部ファイルに書き出したい場合、

```
sqlite> .output [ファイル名]
sqlite> SELECT (略)
```

とする。そうすると、SELECT 文の結果が指定したファイルに書き込まれるようになる。.output でファイル名を指定しない場合、書き込み先は標準出力になる。

データの出力形式を変えるには、ファイルへの書き込み実行前に

```
sqlite> .mode [出力形式]
```

とする。出力形式は、list (デフォルト) の他に csv や line などから選択する。

F.4 SQLite の C 言語 API の使用法

F.4.1 前提

SQLite3 のライブラリ (libsqlite3.so) とヘッダファイル (sqlite3.h) をコンパイル時に読み込めるようにする。これらのファイルが適切に配置されている環境であれば、コンパイル時に”-lsqlite3” とオプションをつけてライブラリをリンクするだけでよい。ソースコードでは

```
1    #include <sqlite3.h>
```

としておく。

F.4.2 使用例

```
1    #include <iostream>
2    #include <sqlite3.h>
3    #include <iostream>
4    #include <fstream>
5    #include <string>
6    #include <vector>
7    #include <algorithm>
8    #include <utility>
9    #include <map>
10   #include <sstream>
11
12   #include <sqlite3.h>
13
14
15   struct Student
16   {
17       Student(){};
18       Student(int _id, const std::string& _name,
19               double _height, double _weight)
20           :id(_id), name(_name), height(_height), weight(_weight){};
21       int id;
22       std::string name;
```

```

23     double height, weight;
24 };
25
26 int main(int argc, char* argv[])
27 {
28
29     std::vector<Student> students;
30     students.emplace_back(1, "Alice", 158, 51.5);
31     students.emplace_back(2, "Bob", 183.3, 70);
32     students.emplace_back(3, "Charlie", 165, 60);
33
34     sqlite3 *db;
35     sqlite3_open_v2("test.sqlite", &db,
36                     SQLITE_OPEN_READWRITE|SQLITE_OPEN_CREATE, nullptr);
37
38     char *err;
39     sqlite3_exec(db, "CREATE_TABLE_IF_NOT_EXISTS_students(personal_id_
40                     INTEGER, name TEXT, height REAL, weight REAL);", nullptr, nullptr, &
41                     err);
42
43     //INSERT data
44     for(auto & s: students)
45     {
46         sqlite3_stmt * stmt;
47
48         std::string statement = "INSERT INTO students VALUES (";
49         statement += std::to_string(s.id) + ", " ;
50         statement += "'" + s.name + "', " ;
51         statement += std::to_string(s.height) + ", " ;
52         statement += std::to_string(s.weight);
53         statement += ");";
54
55         sqlite3_prepare_v3(db, statement.c_str(), -1, 0, &stmt, nullptr);
56
57         sqlite3_step(stmt);
58         sqlite3_finalize(stmt);
59     }
60
61     //SELECT content
62     {
63         sqlite3_stmt * stmt;
64
65         std::string statement = "SELECT name FROM students ORDER BY height;"
66         ;
67         sqlite3_prepare_v3(db, statement.c_str(), -1, 0, &stmt, nullptr);
68
69         std::vector<std::string> order;
70         auto ret = sqlite3_step(stmt);
71         while (ret == SQLITE_ROW)
72         {

```

```

71         std::ostringstream tmp;
72         tmp << sqlite3_column_text(stmt, 0);
73         order.push_back(tmp.str());
74         ret = sqlite3_step(stmt);
75     }
76
77     for(auto &name : order)
78     {
79         std::cout << name << std::endl;
80     }
81     sqlite3_finalize(stmt);
82 }
83
84
85     sqlite3_close_v2(db);
86
87 }
```

実行結果は、

```

Alice
Charlie
Bob
```

となる。

G 三次元の回転行列

プログラム中で飛跡を三次元空間で回転させる際には以下の3行3列の行列を用いた。

$$R_{\mathbf{n}}(\theta) = \begin{pmatrix} n_x^2(1 - \cos \theta) + \cos \theta & n_x n_y(1 - \cos \theta) - n_z \sin \theta & n_z n_x(1 - \cos \theta) + n_y \sin \theta \\ n_x n_y(1 - \cos \theta) + n_z \sin \theta & n_y^2(1 - \cos \theta) + \cos \theta & n_y n_z(1 - \cos \theta) - n_x \sin \theta \\ n_z n_x(1 - \cos \theta) - n_y \sin \theta & n_y n_z(1 - \cos \theta) + n_x \sin \theta & n_z^2(1 - \cos \theta) + \cos \theta \end{pmatrix} \quad (8)$$

行列 $R_{\mathbf{n}}(\theta)$ は単位ベクトル $\mathbf{n} = (n_x, n_y, n_z)$ を軸とし、そのまわりに角度 θ だけだけ回転させる操作に対応する。回転前の方向ベクトルを (x, y, z) 、回転後の方向ベクトルを (x', y', z') とすると回転の演算は $R_{\mathbf{n}}(\theta)$ を用いて以下のように書かれる。

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = R_{\mathbf{n}}(\theta) \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad (9)$$

H 検出ガスの性質の表

PDG のレビュー [4] の "35.6 Gaseous detectors" から引用した。

表 2: 摂氏 20 度、1 気圧のもとでのガスの性質。 E_x と E_I はそれぞれ最低励起エネルギーと第一イオン化エネルギーを表す。 W_I が平均イオン化エネルギー、 $dE/dx|_{min}$ 、 N_P 、 N_T はそれぞれ最小電離粒子 (MIP) に対するエネルギー損失率、1cm あたりの 1 次電離電子数、全電離電子数を表す。

Gas	Density mg cm ⁻³	E_x eV	E_I eV	W_I eV	$dE/dx _{min}$ keV cm ⁻¹	N_P cm ⁻¹	N_T cm ⁻¹
He	0.179	19.8	24.6	41.3	0.32	3.5	8
Ne	0.839	16.7	21.6	37	1.45	13	40
Ar	1.66	11.6	15.7	26	2.53	25	97
Xe	5.495	8.4	12.1	22	6.87	41	312
CH ₄	0.667	8.8	12.6	30	1.61	28	54
C ₂ H ₆	1.26	8.2	11.5	26	2.91	48	112
iC ₄ H ₁₀	2.49	6.5	10.6	26	5.67	90	220
CO ₂	1.84	7.0	13.8	34	3.35	35	100
CF ₄	3.78	10.0	16.0	54	6.38	63	120

参考文献

- [1] Garfield++. <https://garfieldpp.web.cern.ch/garfieldpp/>.
- [2] James F. Ziegler and Jochen P. Biersack. *The Stopping and Range of Ions in Matter*, pages 93–129. Springer US, Boston, MA, 1985.
- [3] SQLite. <https://www.sqlite.org/index.html>.
- [4] Particle Data Group, P A Zyla, R M Barnett, J Beringer, O Dahl, D A Dwyer, D E Groom, C J Lin, K S Lugovsky, E Pianori, D J Robinson, C G Wohl, W M Yao, K Agashe, G Aielli, B C Allanach, C Amsler, M Antonelli, E C Aschenauer, D M Asner, H Baer, Sw Banerjee, L Baudis, C W Bauer, J J Beatty, V I Belousov, S Bethke, A Bettini, O Biebel, K M Black, E Blucher, O Buchmuller, V Burkert, M A Bychkov, R N Cahn, M Carena, A Ceccucci, A Cerri, D Chakraborty, R Sekhar Chivukula, G Cowan, G D’Ambrosio, T Damour, D de Florian, A de Gouvêa, T DeGrand, P de Jong, G Dissertori, B A Dobrescu, M D’Onofrio, M Doser, M Drees, H K Dreiner, P Eerola, U Egede, S Eidelman, J Ellis, J Erler, V V Ezhela, W Fetscher, B D Fields, B Foster, A Freitas, H Gallagher, L Garren, H J Gerber, G Gerbier, T Gershon, Y Gershtein, T Gherghetta, A A Godizov, M C Gonzalez-Garcia, M Goodman, C Grab, A V Gritsan, C Grojean, M Grünewald, A Gurtu, T Gutsche, H E Haber, C Hanhart, S Hashimoto, Y Hayato, A Hebecker, S Heinemeyer, B Heltsley, J J Hernández-Rey, K Hikasa, J Hisano, A Höcker, J Holder, A Holtkamp, J Huston, T Hyodo, K F Johnson, M Kado, M Karliner, U F Katz, M Kenzie, V A Khoze, S R Klein, E Klempt, R V Kowalewski, F Krauss, M Kreps, B Krusche, Y Kwon, O Lahav, J Laiho, L P Lellouch, J Lesgourgues, A R Liddle, Z Ligeti, C Lippmann, T M Liss, L Littenberg, C Lourenço, S B Lugovsky, A Lusiani, Y Makida, F Maltoni, T Mannel, A V Manohar, W J Marciano, A Masoni, J Matthews, U G Meißner, M Mikhasenko, D J Miller, D Milstead, R E Mitchell, K Mönig, P Molaro, F Moortgat, M Moskvic, K Nakamura, M Narain, P Nason, S Navas,

M Neubert, P Nevski, Y Nir, K A Olive, C Patrignani, J A Peacock, S T Petcov, V A Petrov, A Pich, A Piepke, A Pomarol, S Profumo, A Quadt, K Rabbertz, J Rademacker, G Raffelt, H Ramani, M Ramsey-Musolf, B N Ratcliff, P Richardson, A Ringwald, S Roesler, S Rolli, A Romaniouk, L J Rosenberg, J L Rosner, G Rybka, M Ryskin, R A Ryutin, Y Sakai, G P Salam, S Sarkar, F Sauli, O Schneider, K Scholberg, A J Schwartz, J Schwiening, D Scott, V Sharma, S R Sharpe, T Shutt, M Silari, T Sjöstrand, P Skands, T Skwarnicki, G F Smoot, A Soffer, M S Sozzi, S Spanier, C Spiering, A Stahl, S L Stone, Y Sumino, T Sumiyoshi, M J Syphers, F Takahashi, M Tanabashi, J Tanaka, M Taševský, K Terashi, J Terning, U Thoma, R S Thorne, L Tiator, M Titov, N P Tkachenko, D R Tovey, K Trabelsi, P Urquijo, G Valencia, R Van de Water, N Varelas, G Venanzoni, L Verde, M G Vincter, P Vogel, W Vogelsang, A Vogt, V Vorobyev, S P Wakely, W Walkowiak, C W Walter, D Wands, M O Wascko, D H Weinberg, E J Weinberg, M White, L R Wiencke, S Willocq, C L Woody, R L Workman, M Yokoyama, R Yoshida, G Zanderighi, G P Zeller, O V Zenin, R Y Zhu, S L Zhu, F Zimmermann, J Anderson, T Basaglia, V S Lugovsky, P Schaffner, and W Zheng. Review of Particle Physics. *Progress of Theoretical and Experimental Physics*, 2020(8), 08 2020. 083C01.