

# 半導体デバイス・シミュレータ・プログラム使用マニュアルおよび計算事例

Advanced Algorithm & Systems

吾妻広夫

2013年3月31日

本書類は、「半導体デバイス緩和時間シミュレータ」および「半導体 pn 接合デバイス・シミュレータ」の、使用方法、計算事例、プログラム上の問題点を、まとめたものである。

## 半導体デバイス緩和時間シミュレータの使用方法和計算事例について

### 【概要】

ソースコード simulator-05.f90 は、一様電界が印加された半導体中を、キャリア電子・正孔が、音響フォノン散乱を受けながら運動する際の、緩和時間を評価することを目的としたプログラムである。

プログラムは、モンテカルロ法を使って緩和時間の期待値を求める仕組みになっており、数値計算を実行する際は、計算時間と計算精度の釣り合いを考えて、入力パラメータを慎重に設定することがユーザに求められる。

### 【コンパイル方法】

ソースコード simulator-05.f90 は、Fortran90 の一般的な文法に従って書かれている。また、このプログラムは、Linux 上で使われることを前提に書かれている。従って、ソースコード simulator-05.f90 は、Linux マシン上において、GNU コンパイラ gcc に含まれる gfortran、または、Intel の Fortran コンパイラによって、コンパイルされることが望ましい。

GNU の gfortran でコンパイルする場合のコマンドは、以下のとおりとなる。

```
%> gfortran -O3 -o simulator-05 simulator-05.f90
```

### 【初期パラメータ設定方法】

実行プログラム simulator-05 の初期パラメータ設定方法は、以下のとおりである。

まず、ソースコード simulator-05.f90 内に書き込まれたパラメータを、次の要領で設定する。

(9 行目、および、284 行目)

```
double precision :: mass_effective_ratio = 0.33D0
```

キャリア電子・正孔の有効質量の比率を設定する。

(288 行目)

```
double precision :: deformation_potential = 7.81D0 ![eV]
```

半導体材料の deformation potential の値を設定する。

(289 行目)

```
double precision :: Young_modulus = 51.1D9 ![N/m^2]
```

半導体材料の弾性率の値を設定する。

このように、ソースコード simulator-05.f90 内のパラメータ値を書き換え、前に説明した要領でコンパイルを行う。

また、これらとは別に、input.dat ファイルを用意する。input.dat ファイルは、以下の 4 行からなるテキストファイルである。

```
electric_field_external -4.0D6
```

```
time_total 36.0D-12
```

```
temperatruure 300.0D0
```

```
number_particles 32500000
```

- 1 行目は外部印加電場の強度で、単位は[V/m]である。
- 2 行目はキャリア電子・正孔が運動する時間の長さで、単位は[s]である。
- 3 行目は系全体の温度で、単位は[K]である。
- 4 行目はモンテカルロ法による計算で発生される全粒子数である。

#### 【プログラム実行方法、および、計算事例】

緩和時間を求めるための、具体的な手順は以下のとおりである。

1)

まず作業ディレクトリを 1 個用意する。ここでは例として、`exam_05` という名前の作業ディレクトリを用意したとする。

2)

次に、作業ディレクトリ `exam_05` 内に、`test_*` という名前の計算用ディレクトリを複数作成し、これらのディレクトリ一つ一つの中で、1 回のシミュレーション計算を実行することとする。ただし、\*の部分には数桁の数字を入れて、ディレクトリ名を区別するものとする。また、計算用ディレクトリ以外の用途で、名前の先頭が"`test_`"で始まるディレクトリを、`exam_05` 内に作らないようにする。

例えば、`test_20` においては、`input.dat` が次で与えられるシミュレーション計算を実行するものとする。

```
electric_field_external -4.0D6  
time_total 36.0D-12  
temperatrure 300.0D0  
number_particles 32500000
```

また、`test_81` においては、`input.dat` が次で与えられるシミュレーション計算を実行するものとする。

```
electric_field_external -1.5D9  
time_total 7.5D-12  
temperatrure 300.0D0  
number_particles 3200000
```

計算を効果的に進めるコツとしては、電界強度を、 $-4.0\text{E}+6[\text{V/m}]$  から  $-1.5\text{E}+9[\text{V/m}]$  程度の範囲で、適当な幅で分散させて、シミュレーションを行う点が挙げられる。

また、電界強度の絶対値が大きくなるにつれて、キャリア電子・正孔の音響フォノン散乱確率が増大し、単位時間当たりの散乱回数が急激に増大する。これに伴って、緩和時間が減少するので、`input.dat` における、項目 `electric_field_external` の絶対値が大きくなるにつれて、項目 `time_total` の時間長さを小さくするのが望ましい。

さらには、電界強度の絶対値が大きくなるにつれて、単位時間当たりの散乱回数が急激に増大することから、モンテカルロ・シミュレーションに必要な計算量も、それに応じて増大することが予想される。そのため、項目 `electric_field_external` の絶対値が大きくなるにつれて、項目 `number_particles` の値を小さく設定して、シミュレーションにかかる総計算時間を適宜調節することが望ましい。

実行プログラムを走らせるには、`input.dat` が入っている `test_*` ディレクトリ内で、以下の

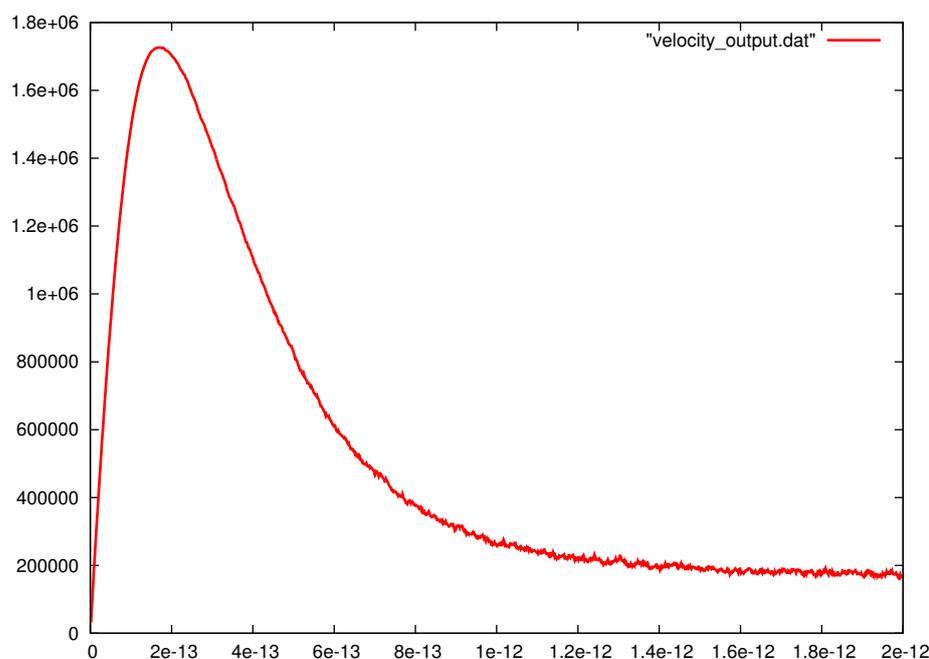
コマンドを打てば良い。

```
%> ./simulator-05
```

引数等は特に必要ない点に注意する。

以下に、実行プログラム `simulator-05` の計算結果の例を示す。

図 1



上の図は、計算用ディレクトリ `test_43` 内のデータファイル `velocity_output.dat` をプロットしたグラフである。このグラフは、シリコン半導体中での、温度  $300[\text{K}]$ 、外部電界強度  $-4.0\text{E}+7[\text{V/m}]$  とした際の、キャリア電子速度の時間変化を表している。縦軸は、キャリア電子速度  $[\text{m/s}]$ 、横軸は時間  $[\text{s}]$  となっている。このグラフは、いわゆる、半導体中でのキャリア電子のオーバーシュート現象を表していると考えられる。

3)

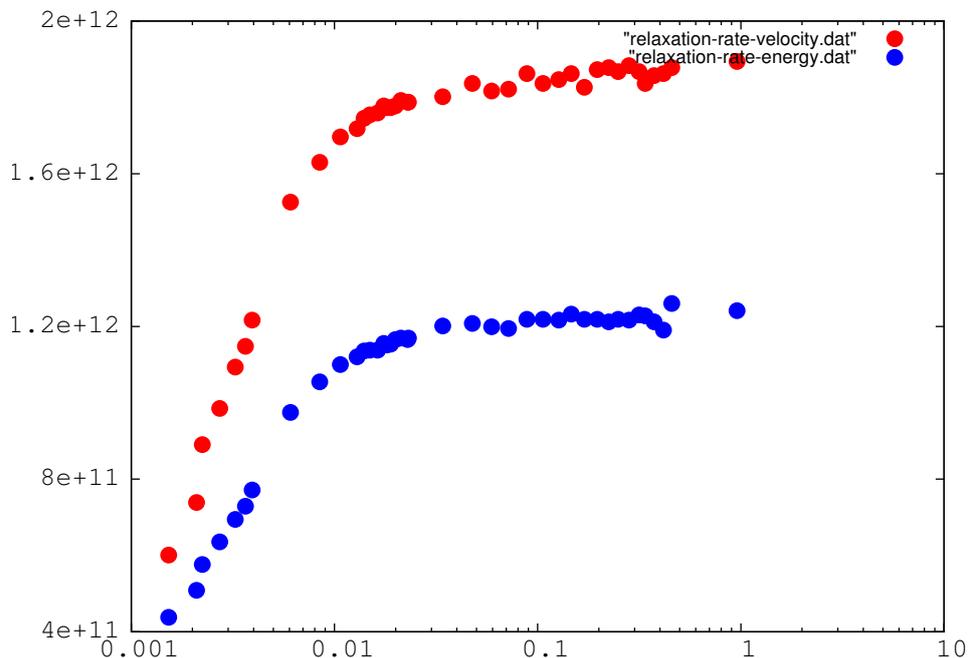
`test_*` という名前の全ての計算ディレクトリ内のモンテカルロ・シミュレーションが終了したら、作業ディレクトリ `exam_05` 内において、シェルスクリプト `data_processing.sh` を実行する。これにより、`relaxation-rate_energy.dat`、および、`relaxation-velocity.dat` という二つのファイルが生成される。

データファイル `relaxation-rate_energy.dat` では、 $x$  座標がキャリア電子・正孔の運動エネルギー  $[\text{eV}]$ 、 $y$  座標がキャリア電子・正孔の運動エネルギー緩和レート  $[\text{1/s}]$  を表している。

データファイル `relaxation-velocity.dat` では、 $x$  座標がキャリア電子・正孔の運動エネルギー  $[\text{eV}]$ 、 $y$  座標がキャリア電子・正孔の速度緩和レート  $[\text{1/s}]$  を表している。

以下に、`relaxation-rate_energy.dat`、および、`relaxation-velocity.dat` の数値データを、グラフとして示すことにする。

図 2



上の図は、温度 300[K]の半導体シリコン中での、キャリア電子の速度緩和レート、および、運動エネルギー緩和レートをプロットしたグラフである。赤い丸は速度緩和レート、青い丸は運動エネルギー緩和レートを表している。グラフの縦軸は緩和レート[1/s]、横軸はキャリア電子の運動エネルギー[eV]となっている。さらには、グラフの横軸は対数目盛りが設定されている。このグラフでは、速度緩和レートは運動エネルギー緩和レートより大きな値を取る結果となっている。この事実は、半導体物理において一般的に知られている事実であり、シミュレーションは、その事実を良く再現していると言える。

## 半導体 pn 接合デバイス・シミュレータの使用法、計算事例、および、問題点について

### 【概要】

ソースコード `beq_simulator-26.f90` は、半導体 pn 接合デバイスの電流・電圧特性をシミュレートすることを目的としたプログラムである。このプログラムは、ボルツマン方程式を簡略化したバランス方程式と呼ばれる、古典的な輸送方程式を解くことによって、解を得る点に特徴がある。

バランス方程式は、連立一階微分方程式の形をしており、物理的に安定した平衡状態の解に到達するのが非常に難しい。適切な初期値を各種の物理量に与えて、微分方程式を離散化した方程式に従って時間変化を追って行けば、原理的には、安定した解に到達し、例えば、pn 接合のダイオードとしての整流特性が得られるべきである。しかし、本プログラム `beq_simulator-26` では、ダイオードを流れる電流の時間変化を追っても、平衡状態になかなか到達せず、解が物理的に正しい値に収束しない結果となっている。この点は、今後の課題である。

### 【コンパイル方法】

ソースコード `beq_simulator-26.f90` は、Fortran90 の一般的な文法に従って書かれている。また、このプログラムは、Linux 上で使われることを前提に書かれている。

しかし、実数の浮動小数点計算において、四倍精度を使用しており、この点が、やや特殊な仕様となっている。四倍精度のオプションを正しくコンパイルするには、以下の比較的新しいバージョンのコンパイラが必要である。

- GNU コンパイラ `gcc` 中の `gfortran` を使用する場合、`gcc` のバージョンは、4.6.2 以降とする。
- Intel の Fortran コンパイラの場合、バージョン 9.1 以降とする。

GNU の `gfortran` でコンパイルする場合のコマンドは、以下のとおりとなる。

```
%> gfortran -O3 -o beq_simulator-26 beq_simulator-26.f90
```

### 【初期パラメータ設定方法】

実行プログラム `beq_simulator-26` の初期パラメータ設定方法は、以下のとおりである。ソースコード `beq_simulator-26.f90` 内に書き込まれたパラメータを、次の要領で設定する。

(5 行目)

```
integer(8), parameter :: NN = 4000
```

デバイスを x 方向に離散化する場合の分割数を指定する。

(20,21,22,23 行目)

```
real(kind=16) :: Eg = 1.11 ![eV]: E_{C}-E_{V}
```

```
real(kind=16) :: mass_electron_effective_ratio = 0.33Q0
```

```
real(kind=16) :: mass_hole_effective_ratio = 0.52Q0
```

```
real(kind=16) :: e_relative_permittivity = 12.0Q0
```

半導体のバンドギャップ、キャリア電子の有効質量比、キャリア正孔の有効質量比、半導体の比誘電率を設定する。

(26 行目)

```
real(kind=16) :: N_DN = 1.5Q17 ![1/cm^{3}]: impurity density
```

n 型半導体中にドーパされた不純物イオンの密度[1/cm<sup>3</sup>]を設定する。

(29 行目)

```
real(kind=16) :: N_AP = 1.5Q17 ![1/cm{3}]: impurity density
```

p 型半導体中にドーパされた不純物イオンの密度[1/cm<sup>3</sup>]を設定する。

(31 行目)

```
real(kind=16) :: temperature = 300.0Q0
```

温度[K]を設定する。

(33 行目)

```
real(kind=16) :: length_x_device = 0.4Q-4 ![cm]: total length of device (pn-junction)
```

pn 接合デバイスの x 方向の長さ[cm]を設定する。

(35 行目)

```
real(kind=16) :: Vconst = -0.025Q0 ![V]: external constant voltage
```

外部から印加する電圧[V]を設定する。ダイオードの順方向の電圧を負とする。

(37 行目)

```
real(kind=16) :: delta_t = 1.0Q-18 ![s]
```

時間変数を差分化する際の時間幅[s]を設定する。

(39 行目)

```
real(kind=16) :: factor_heat_conduction = 0.0Q0
```

バランス方程式を差分化する際に、二次の微分量となる熱伝導項を有効とする。この値は、本来は 0.0 とすべきではあるが、これを 1.0 にして差分式の中に入れると、計算結果が安定化する傾向があることが知られている。

(42 行目)

```
integer(8), parameter :: time_step_total = 8000
```

```
integer(8), parameter :: time_step_interval = 20
```

シミュレーションの時間総ステップ数と、計算結果を出力する間隔を指定する。この場合、総ステップ数は 8000 で、20 ステップ毎に計算結果を出力する設定になる。

(552 行目)

```
chi(i) = 4.05Q0 * eV_to_erg ![erg]: electron affinity
```

半導体の電子親和力[eV]を設定する。この場合、シリコン半導体の電子親和力  $\chi=4.05$ [eV] が設定されている。

(582~593 行目)

```
if(Vconst < 0.0Q0) then
```

```
  do i = 0, NN + 1
```

```
    v(i) = -1.00Q7
```

```
    v_hole(i) = 1.00Q7
```

```
  end do
```

```
end if
```

```
if(Vconst > 0.0Q0) then
```

```
  do i = 0, NN + 1
```

```
    v(i) = 1.00Q7
```

```
    v_hole(i) = -1.00Q7
```

```
end do
end if
```

キャリア電子・正孔の初期速度[cm/s]が設定されている。この場合、初期速度として、 $1.0E+7$ [cm/s]が設定されている。初期速度を指定する箇所が、全部で 4 個あることに注意する。

(1033 行目)

```
relaxation_rate_density = sqrt(energy_kinetic / w0) * (1.0Q0 / 6.0Q-6) ![1/s]
```

キャリア電子密度の緩和係数[1/s]を設定する。この場合、キャリア電子密度緩和時間として、 $\tau=6.0E-6$ [s]が設定されている。

(1065 行目)

```
relaxation_rate_density_hole = sqrt(energy_kinetic_hole / w_hole0) * (1.0Q0 / 6.0Q-6) ![1/s]
```

キャリア正孔密度の緩和係数[1/s]を設定する。この場合、キャリア正孔密度緩和時間として、 $\tau=6.0E-6$ [s]が設定されている。

(1096 行目)

```
relaxation_rate_velocity = sqrt(energy_kinetic / w0) * 2.5Q0 * 1.8Q+13 ![1/s]
```

キャリア電子速度の緩和係数[1/s]を設定する。この場合、キャリア電子速度緩和係数として、 $2.5 \times 1.8E+13=4.5E+13$ [1/s]が設定されている。

(1124 行目)

```
relaxation_rate_velocity_hole = sqrt(energy_kinetic_hole / w_hole0) * 2.5Q0 * 1.8Q+13 ![1/s]
```

キャリア正孔速度の緩和係数[1/s]を設定する。この場合、キャリア正孔速度緩和係数として、 $2.5 \times 1.8E+13=4.5E+13$ [1/s]が設定されている。

(1152 行目)

```
relaxation_rate_kinetic_energy = sqrt(energy_kinetic / w0) * 2.5Q0 * 1.2Q+13 ![1/s]
```

キャリア電子エネルギーの緩和係数[1/s]を設定する。この場合、キャリア電子エネルギー緩和係数として、 $2.5 \times 1.2E+13=3.0E+13$ [1/s]が設定されている。

(1180 行目)

```
relaxation_rate_kinetic_energy_hole = sqrt(energy_kinetic_hole / w_hole0) * 2.5Q0 * 1.2Q+13 ![1/s]
```

キャリア正孔エネルギーの緩和係数[1/s]を設定する。この場合、キャリア正孔エネルギー緩和係数として、 $2.5 \times 1.2E+13=3.0E+13$ [1/s]が設定されている。

#### 【プログラム実行方法、および、計算事例】

プログラムの実行方法は、作業ディレクトリを 1 個用意して、その中で以下のコマンドを打てば良い。

```
%> ./beq_simulator-26
```

オプションは何も必要ない点に注意する。

実行プログラム `beq_simulator-26` は、一定の時間ステップ間隔毎に、以下の 8 種類のデータファイルを生成する。

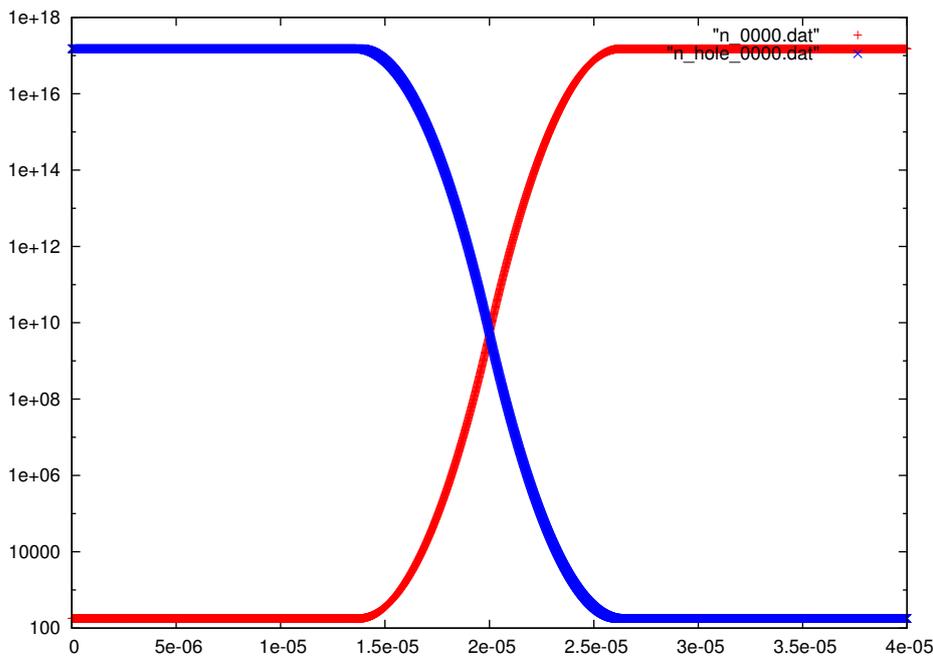
- `w_*.dat` : デバイス内部での、キャリア電子の運動エネルギー分布
- `v_*.dat` : デバイス内部での、キャリア電子の速度分布

- v\_hole\_\*.dat : デバイス内部での、キャリア正孔の速度分布
- n\_\*.dat : デバイス内部での、キャリア電子の密度分布
- n\_hole\_\*.dat : デバイス内部での、キャリア正孔の密度分布
- phi\_\*.dat : デバイス内部での、電位の x 方向の変化
- F\_el\_\*.dat : デバイス内部での、電界の x 方向の変化
- current\_\*.dat : デバイス内部での、電流密度の分布

以下に実行プログラム beq\_simulator-26 の計算結果の例を示す。計算に必要なパラメータは、上で例として説明した初期パラメータの値を、そのまま使っているとする。特に、キャリア電子・正孔の初期速度が  $1.0E+7$ [cm/s] に設定されている点に注意する。

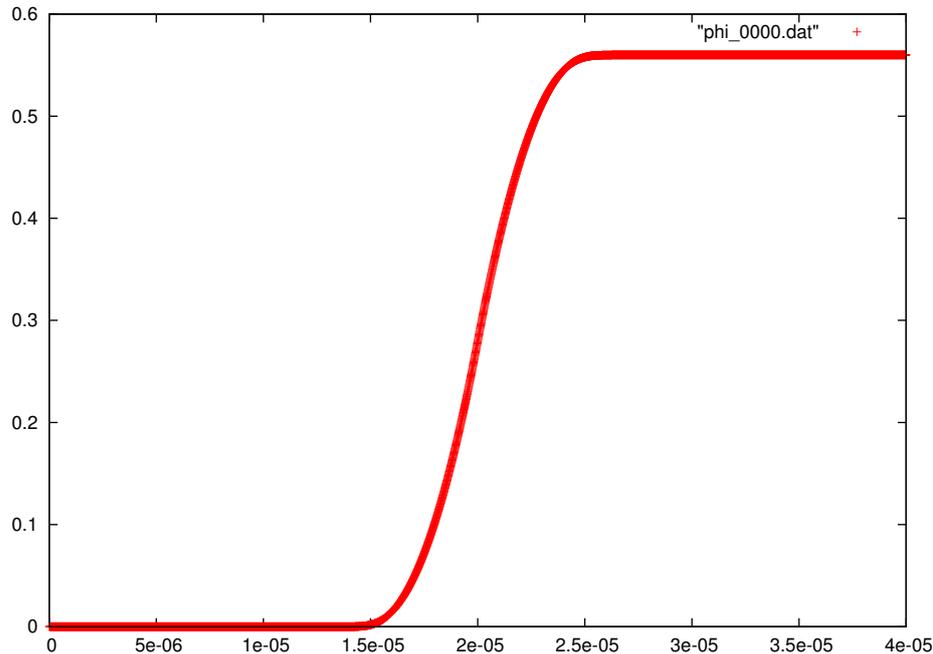
なお、次の点にも注意する。実行プログラム beq\_simulator-26 内部における数値計算では、長さの単位を[cm]、重さの単位を[g]としている。さらに、電子の質量  $m_e$  を 1、電子の素電化  $e$  を 1 としている。そのため、出力結果に関して、適切な単位の換算を行わなくてはならない。

図 3



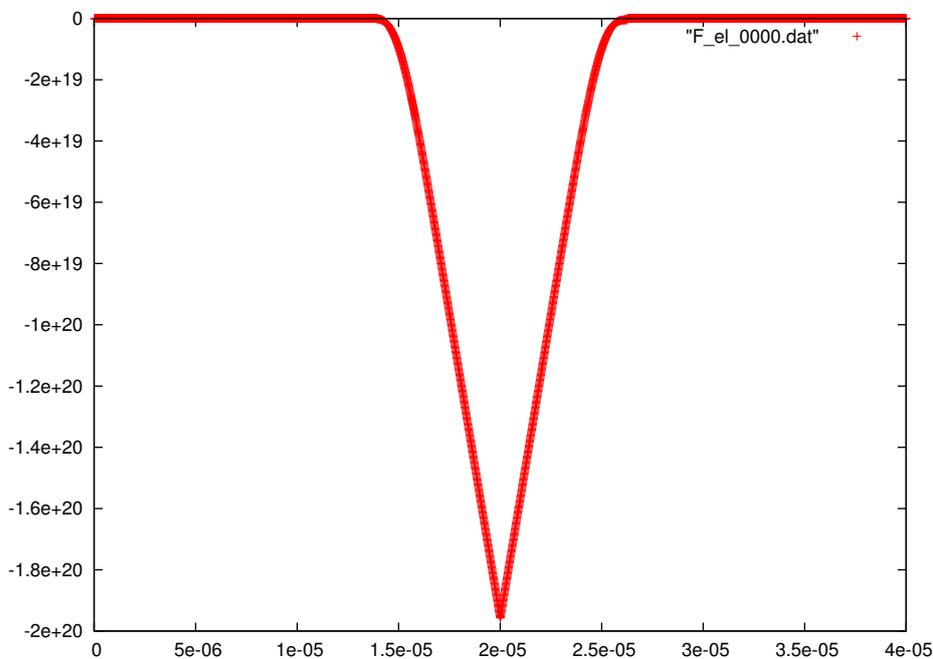
上の図は、シミュレーション初期時刻での、半導体 pn 接合におけるキャリア電子・正孔の密度分布を表したグラフである。縦軸はキャリア電子・正孔密度[1/cm<sup>3</sup>]、横軸は pn 接合デバイスの x 方向の長さ[cm]としている。また、縦軸は対数目盛が取られている。デバイスは、全長  $0.4[\mu\text{m}]$  で、左側 ( $0 \leq x \leq 2.0 \times 10^{-5}$  [cm]) が p 型半導体、右側 ( $2.0 \times 10^{-5} \leq x \leq 4.0 \times 10^{-5}$  [cm]) が n 型半導体、ちょうど中央 ( $x = 2.0 \times 10^{-5}$  [cm]) が接合面となっている。赤い曲線はキャリア電子の密度分布を、青い曲線はキャリア正孔の密度分布を表している。デバイス中央部 ( $1.5 \times 10^{-5} \leq x \leq 2.5 \times 10^{-5}$  [cm]) に、空乏層が形成されていることが見て取れる。

図 4



上の図は、半導体 pn 接合内での電位の変化を表したグラフである。縦軸は電位[V]、横軸は pn 接合デバイスの x 方向の長さ[cm]としている。このグラフを見ると、電位が変化するのは、空乏層内であることが理解される。

図 5



上の図は、半導体 pn 接合内での電界強度の変化を表したグラフである。縦軸は電界強度 [V/cm]、横軸は pn 接合デバイスの x 方向の長さ[cm]としている。ただし、縦軸の電界強度の値については、電子の質量  $m_e$  を 1、電子の素電化  $e$  を 1 とする単位系を取っているため、

その因子の分だけ値が異なる。このグラフを見ると、電界強度がゼロでないのは、空乏層内であることが理解される。

図 6

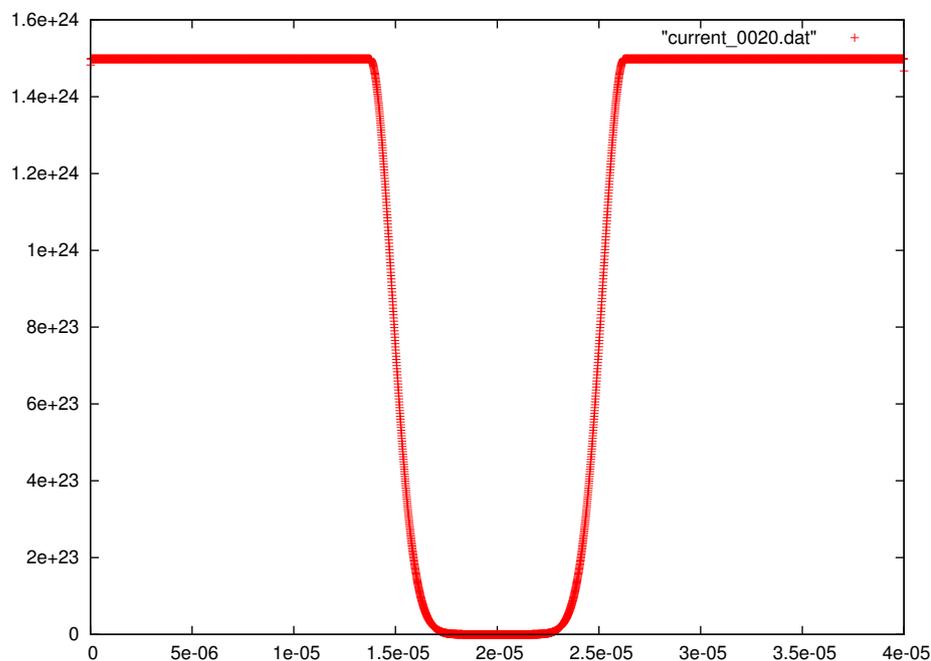


図 7

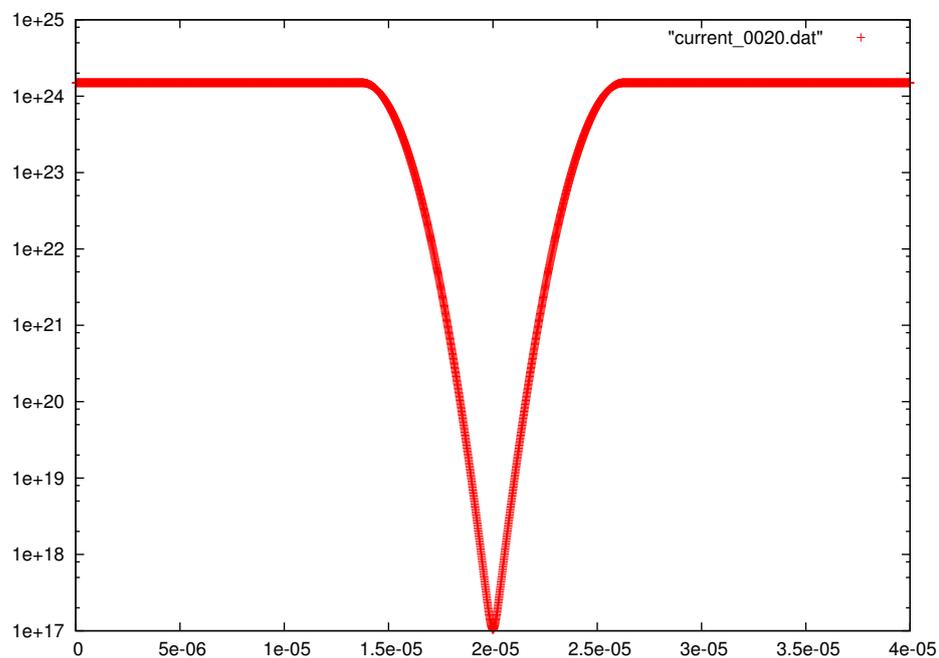


図 6,7 は、シミュレーション計算における 20 ステップ目での、半導体 pn 接合内の電流密度分布を表したグラフである。縦軸は電流密度[A/cm<sup>2</sup>]、縦軸は pn 接合デバイスの x 方向の長さ[cm]としている。ただし、図 6 は縦軸が線形目盛、図 7 は縦軸が対数目盛に設定されている。また、縦軸の電流密度の値については、電子の素電化  $e$  を 1 とする単位系を取っているため、その因子の分だけ値が異なる。具体的には、縦軸の目盛りを  $1.602 \times 10^{-19}$  倍し

た値が、実際の電流密度[A/cm<sup>2</sup>]となる。

図 6,7 の電流密度分布図では、空乏層において、電流がわずかしか流れていないことになっている。これは、空乏層で、電子と正孔の生成・再結合が起こっていることの表れと解釈できる。

### 【問題点】

ソースコード beq\_simulator-26.f90 は、以下に示す二つの問題点が有ると考えられる。

#### (問題点 1)

シミュレーションを長時間ステップ続けても、ダイオードを流れる電流の絶対値は、時間の経過と共に小さくなる一方で、一向に、平衡状態に達する傾向が見られない。

図 8

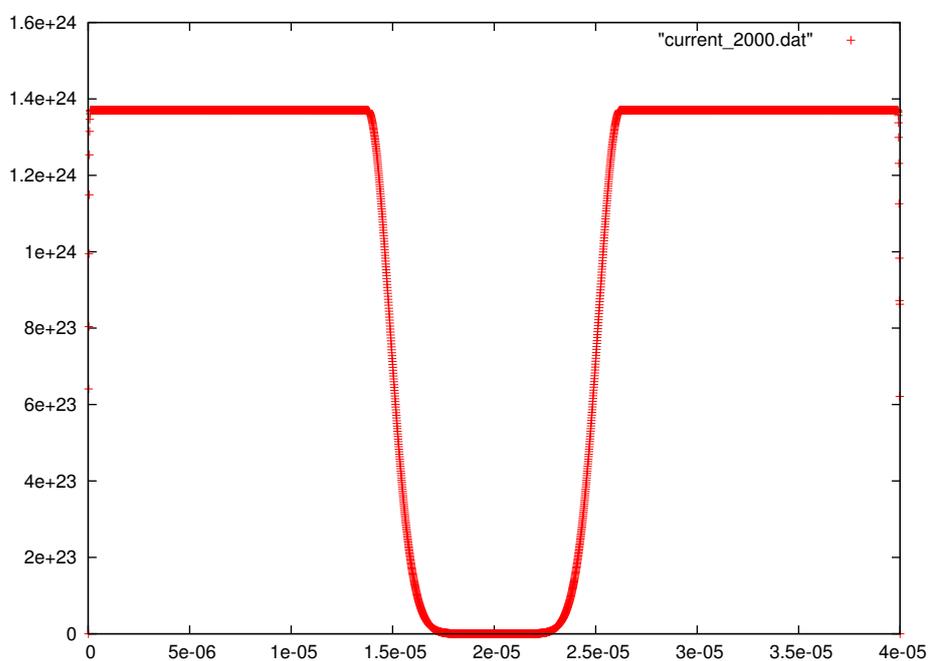


図 9

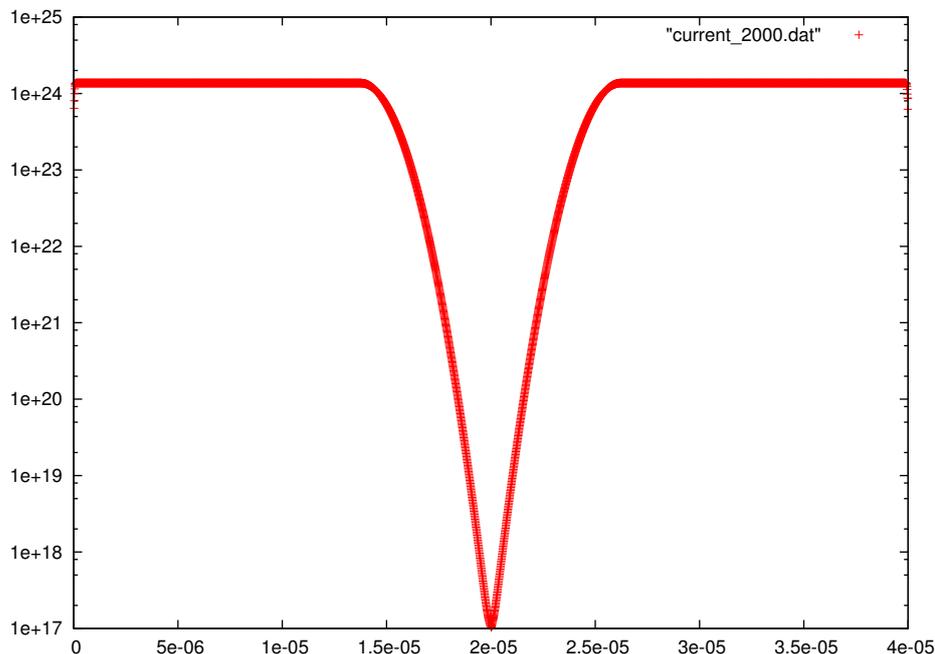


図 8,9 は、シミュレーション計算における 2000 ステップ目での、半導体 pn 接合内の電流密度分布を表したグラフである。縦軸は電流密度[A/cm<sup>2</sup>]、横軸は pn 接合デバイスの x 方向の長さ[cm]としている。ただし、図 8 は縦軸が線形目盛、図 9 は縦軸が対数目盛に設定されている。また、縦軸の電流密度の値については、電子の素電化  $e$  を 1 とする単位系を取っているため、その因子の分だけ値が異なる。具体的には、縦軸の目盛りを  $1.602 \times 10^{-19}$  倍した値が、実際の電流密度[A/cm<sup>2</sup>]となる。

図 6,7,8,9 で示されるように、計算ステップが 20 ステップ目の電流値と、2000 ステップ目の電流値を比較すると、縦軸の目盛りで  $1.5E+24$  から  $1.38E+24$  に減少していることが分かる。すなわち、計算ステップ数を増加させるにつれて、ダイオード電流は減少していることが分かる。このような傾向は、時間ステップ数にして 8000 ステップまで続いており、一定の平衡状態の電流値に収束することは起こらないのが現状である。

以上の理由から、実行プログラム beq\_simulator-26 は、最終的な電流値に収束せず、このため、ダイオードの電流-電圧特性が得られない結果となってしまう。

#### (問題点 2)

キャリア電子・正孔の初期速度によって、ダイオードを流れる電流値が変化してしまう。また、キャリア電子・正孔の初期速度をどのような値に設定するのが物理的に正しいのか、判断できない。

このことを説明するため、以下に、キャリア電子・正孔の初期速度が  $0.7E+7$ [cm/s]に設定されている以外、他の物理的パラメータは上と全く同じシミュレーションの計算結果を示す。

図 10

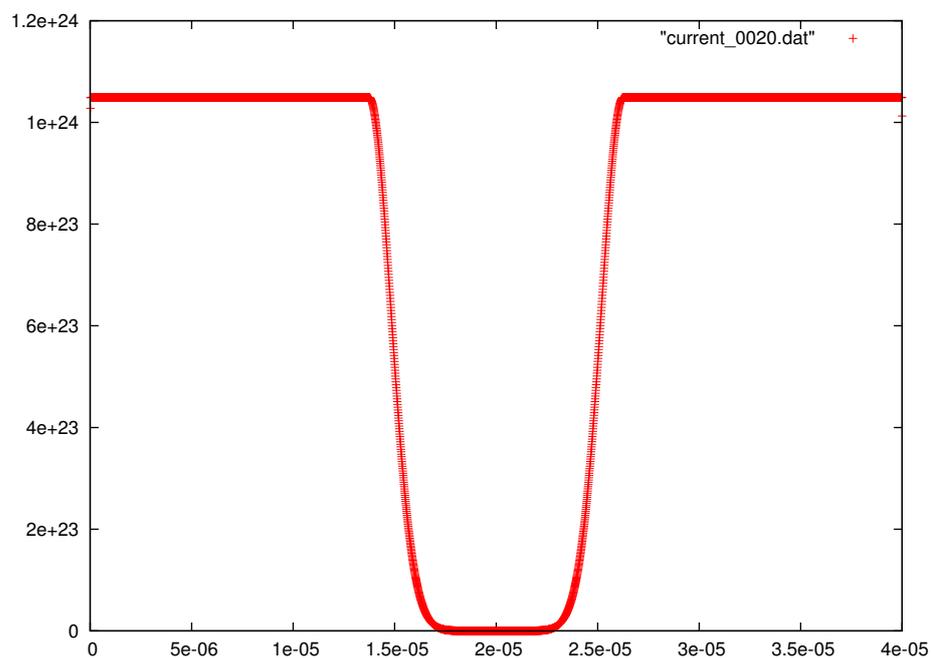


図 11

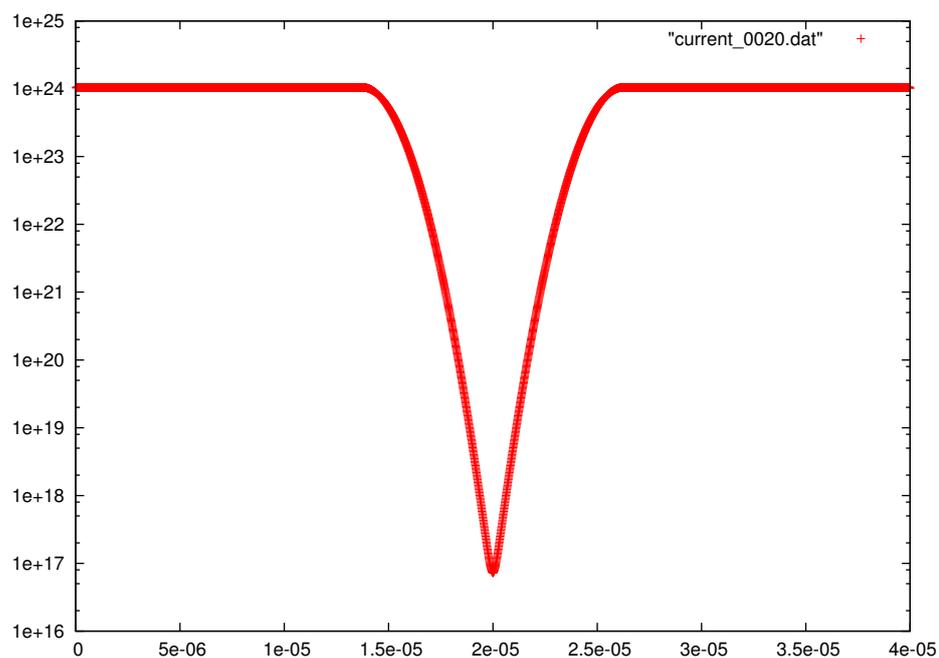


図 10,11 は、シミュレーション計算における 20 ステップ目での、半導体 pn 接合内の電流密度分布を表したグラフである。縦軸は電流密度[A/cm<sup>2</sup>]、横軸は pn 接合デバイスの x 方向の長さ[cm]としている。ただし、図 10 は縦軸が線形目盛、図 11 は縦軸が対数目盛に設定されている。

図 12

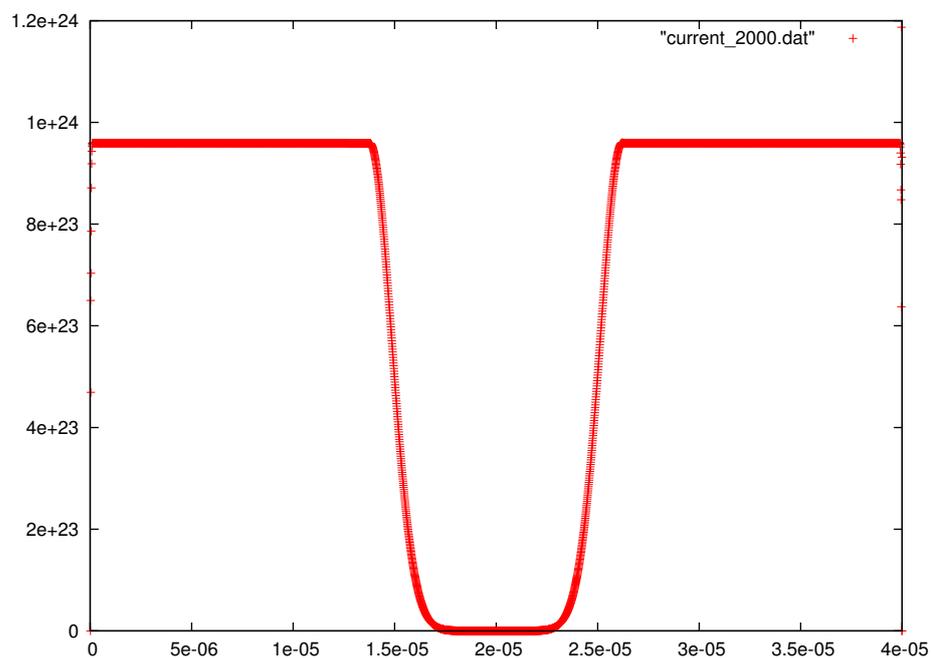


図 13

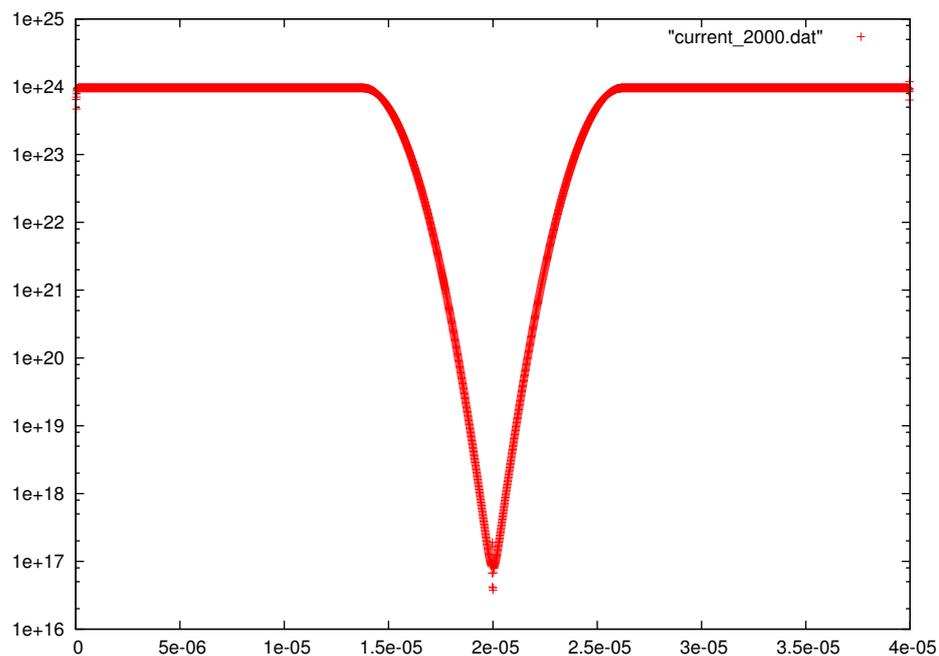


図 12,13 は、シミュレーション計算における 2000 ステップ目での、半導体 pn 接合内の電流密度分布を表したグラフである。縦軸は電流密度[A/cm<sup>2</sup>]、縦軸は pn 接合デバイスの x 方向の長さ[cm]としている。ただし、図 12 は縦軸が線形目盛、図 13 は縦軸が対数目盛に設定されている。

図 6 から図 13 のグラフを見ると分かるように、20 ステップ目、2000 ステップ目での電流値に関して、以下の結果が得られる。

キャリア電子・正孔の初期速度	0.7E+7[cm/s]	1.0E+7[cm/s]
20 ステップ目での電流値	1.05E+24	1.5E+24
2000 ステップ目での電流値	9.6E+24	1.38E+24

上の表を見ると、電流値は、ほぼ初期速度に比例していることが分かる。このことから、初期速度の値の与え方によって、ダイオードを流れる電流は、いくらでも変化してしまい、数値計算シミュレーションとしての意味が薄れてしまっていることが理解される。

これら二つの問題点は、原因として、バランス方程式の立て方に問題があると予想される。この点については、「太陽電池シミュレータに関する技術的疑問点」という書類にまとめられている。

以上