

モンテカルロ法によるシミュレーションの  
高効率並列計算に関する研究

2003年3月

北海道工業大学大学院

鈴木 卓真

# 目 次

|       |                             |    |
|-------|-----------------------------|----|
| 第1章   | 序論                          | 1  |
| 1-1   | まえがき                        | 1  |
| 1-2   | 並列計算に関する従来の研究               | 2  |
| 1-3   | プラズマ・プロセス・シミュレーションに関する従来の研究 | 4  |
| 1-4   | 本論文の目的と構成                   | 5  |
| 第2章   | シミュレーション方法                  | 7  |
| 2-1   | まえがき                        | 7  |
| 2-2   | モンテカルロ法によるシミュレーション          | 7  |
| 2-2-1 | シミュレーション方法                  | 7  |
| 2-2-2 | シミュレーションモデル                 | 8  |
| 2-3   | サンプリング方法                    | 10 |
| 2-4   | 飛行時間の決定                     | 12 |
| 2-5   | 衝突の判定                       | 13 |
| 2-6   | 軌道の計算                       | 14 |
| 第3章   | 並列計算手法                      | 15 |
| 3-1   | まえがき                        | 15 |
| 3-2   | 並列計算環境の構築                   | 15 |
| 3-3   | 同時実行制御および同期制御方法             | 20 |
| 3-4   | 高効率並列計算                     | 26 |
| 3-4-1 | 一般的な並列計算速度効率                | 26 |
| 3-4-2 | 高効率並列計算の理論                  | 27 |

|       |  |     |
|-------|--|-----|
| 第4章   | 電磁界での荷電粒子の運動                                   | 30  |
| 4-1   | まえがき   | 30  |
| 4-2   | 電界と磁界による荷電粒子の運動                                | 30  |
| 4-2-1 | 単一粒子の電界中の運動                                    | 30  |
| 4-2-2 | 単一粒子の磁界中の運動                                    | 32  |
| 4-2-3 | 単一粒子の直交電磁界中の運動                                 | 33  |
| 4-2-4 | 任意の電界と磁界の角度における軌道計算                            | 34  |
| 4-2-5 | カusp磁場における軌道計算                                 | 36  |
| 4-3   | シミュレーション結果および考察                                | 38  |
| 4-3-1 | 電界と磁界が直交するプラズマ・パラメータ                           | 38  |
| 4-3-2 | SiH <sub>4</sub> ガス中の電界と磁界の成す角の違いによるプラズマ・パラメータ | 46  |
| 4-3-3 | Cl <sub>2</sub> ガス中の電界と磁界の成す角の違いによるプラズマ・パラメータ  | 57  |
| 4-3-4 | Ar ガス中の電界と磁界の成す角の違いによるプラズマ・パラメータ               | 70  |
| 4-3-5 | カusp磁界中での粒子の運動                                 | 79  |
| 4-4   | 並列計算結果および考察                                    | 100 |
| 4-4-1 | 並列計算結果   | 100 |
| 4-4-2 | 高効率並列計算理論との比較                                  | 102 |
| 第5章   | 結論   | 104 |
|       | 参考文献   | 107 |
|       | 謝辞   | 110 |

# 第1章 序論

## 1-1 まえがき

近年、半導体メモリやマイクロプロセッサは急速に高密度、高速化しつつあり、nm単位の半導体素子を作るナノテクノロジーの時代に突入しており、半導体薄膜製造におけるプラズマ・プロセスの分野において、高精度な微細加工技術が要求されている。プラズマ化学反応により生じた生成物を固体表面に堆積させ薄膜を形成するデポジションや、弱電離プラズマ中で励起された活性化中性粒子（ラジカル）やイオンにより、固体表面の原子を削り取るエッチングによる半導体薄膜の加工技術においては、電子、イオン密度、エネルギーおよびイオンの入射角が重要であり、電界に磁界を加えた弱電離プラズマ（以降プラズマと記す）の精緻な制御が必要とされる<sup>[1],[2]</sup>。極小領域において低温度で高密度のプラズマを生成することが可能なカスプ磁場中においては、電磁界が複雑に交差するため、プラズマの構造を決定づける粒子レベルの運動、反応過程を理解することが重要となる。プラズマ構造を把握するためにはプラズマ中の粒子エネルギーや粒子密度が指標となり、流体モデル、粒子モデルの数値シミュレーションモデルが提案されている。流体モデルでは、位置空間三次元および速度空間三次元の六次元を扱うボルツマン方程式を用いるモデルによる解析法、プラズマ現象をマクロ的に捉え、係数として速度、電離および拡散から構成される連続の方程式を用いるモデルによるシミュレーションがある。粒子モデルのシミュレーションとして、個々の粒子の運動方程式を解き乱数により衝突判定を行うモンテカルロ法によるモデルがある。モンテカルロ・シミュレーションは他のシミュレーション技法に比較し、仮定が少なくより詳細に物理現象を取り扱うことができるが、精度の高い解を得るためには試行粒子数を増加させなくてはならない。さらに、ガス分子と電子との電離衝突による粒子数の指数関数的増加現象が起こるため膨大な計算時間を要する。電界と磁界を考慮したシミュレーションにおいては高磁界になるほど収束解を得るまでの時間を必要とする。より高速

な計算を行うために、従来からスーパーコンピュータを用いたプロセッシング・エレメント（Processing Element：PE）の並列化による計算がなされており、林および堀口<sup>[3]</sup>は、分子動力学の手法によりシミュレーションを行っている。最近ではパーソナルコンピュータ（Personal Computer：PC）の演算速度の進歩および LAN やインターネットの整備が進み、大規模なスーパーコンピュータを用いなくとも、ネットワーク上で PC 同士の相互通信を行うことにより仮想的に 1 つの計算機を構成するクラスタ化、分散環境における大域的シミュレーションおよびグリッド・コンピューティング技術により、多くの計算がより高速に行える並列計算環境が整備されつつある<sup>[4]-[9]</sup>。しかし、並列計算の効率は PE または PC 間における通信時間、スケジューリングおよび同期処理性能に依存するため、PE および PC の増加に伴う線形的な台数効果を得ることができない<sup>[10]-[12]</sup>。本論文では、モンテカルロ法によるプラズマ・プロセス・シミュレーションを対象とし、分散環境を基盤とする高効率並列計算手法を提案する。

## 1-2 並列計算に関する従来の研究

コンピュータによる並列処理は 1960 年頃より実験が行われており、1976 年の Cray-1 スーパーコンピュータは全く新しいレベルの性能を実現していた。1982 年に登場した Cray X-MP では、世界で初めてマルチプロセッサが搭載され、その性能はますます高速となった。その後継機である Cray-SV1 は、シングルプロセッサの最大性能で 4.8GFLOPS を達成し、スーパークラスタ構成により 1TFLOPS までスケールアップすることができ、現在に至るまでより高速なスーパーコンピュータによる並列計算が行われている。林および堀口<sup>[3]</sup>は、分子動力学法によるドメイン分割法を用いた CM-5 スーパーコンピュータによる並列計算を行っており、位置空間を分割し各 PE に計算を割り当て、そのときに生じる粒子の不均一性が計算時間の不均一性を招き全体の計算速度に影響を及ぼすため、粒子計算の負荷分散手法について動的な負荷分散法を提案している。

最近では、蔡ら<sup>[5]</sup>は大規模なスーパーコンピュータを用いずに、小規模 LAN 内でベアウルフ・クラスタ・システムと呼ばれる PC レベルの並列計算環境を構築し、Particle-in-Cell (PIC) 法による粒子シミュレーションの高速化について研究しており、OS に Linux、ネットワークに 100Base/TX を使用し、PC 間のメッセージ・パッシングを PVM (Parallel Virtual Machine)、OpenMP の組み合わせと HPF (High Performance Fortran) を用いたものとを比較している。シミュレーションの空間領域を各 CPU にグリッドとして分配し、グリッド間での粒子の移動が起こるとき、メッセージ・パッシング、すなわち、PC 間でのデータ通信が行われる。このことが並列計算の性能を大きく左右することに着目し計算を行った結果として、PVM、OpenMP の組み合わせにおいて最大 32 台の PC で 13 倍程度、これに比較し HPF を利用した環境においては、10%~60%遅延する結果を報告している。また、分散環境における大域的シミュレーションとして、高木ら<sup>[6]</sup>は Ninf (Network Based Information Library for High Performance Computing) を提案している。Ninf はインターネットを活用し不特定多数のユーザに並列環境を提供し、世界規模の共同利用計算環境を実現することを目的としている。そのため、ネットワーク上でのセキュリティに着目し Java 言語により開発を行っている。PC 間の通信方法として Remote Procedure Call (RPC) を用いている。ハードウェア構成として Alpha ワークステーション 32 台、ネットワークに 100Base/TX を使用し、MatMult (行列の掛け算) を実行させ最大で 10 倍程度の速度効率を報告している。また、IBM ではスーパーコンピュータの開発から派生したグリッド・コンピューティング技術により、多くの計算がより高速に行える環境を提供するために Open Grid Service Architecture (OGSA) を公開しており、サーバサイド Java である J2EE (Java2 Enterprise Edition) に統合されている。グリッドはコンピュータ・リソースを結集して、多数のコンピュータで遺伝子情報の解読や核爆発シミュレーションなどのタスクを共有するものである<sup>[7],[8]</sup>。より実用的な粒子シミュレーションの並列計算として Kawamura ら<sup>[9]</sup>は、PIC 法による RF 電界下での

計算に関する研究を行っており、性能評価および 1CPU で行った結果と比較した。並列化はシミュレーションの空間領域を各 CPU にグリッドとして分配することで行われている。ハードウェア環境としては、2CPU の並列化、4CPU の並列化、1CPU の PC を 8 台用いる三つの構成であり、CPU 間の通信に MPI(Message Passing Interface)を用いている。この結果として 2CPU で構成された環境で最大 2 倍、4CPU で構成された環境で最大 3.95 倍、1CPU の PC を 8 台で構成した環境で最大 6.8 倍の結果を報告している。

### 1-3 プラズマ・プロセス・シミュレーションに関する従来の研究

材料の微細加工や物質創製の分野において、プラズマ・プロセス技術が広く用いられている。プラズマの化学反応性を利用し、技術が先端的電子デバイスである IC や LSI の製造に薄膜のデポジションやエッチングが不可欠となっている。これらの物理現象を詳細に把握するために、位置空間三次元および速度空間三次元の六次元を扱ったボルツマン方程式、速度項、電離項および拡散項を扱う連続の方程式を用いるシミュレーション、乱数を用い位置空間三次元および速度空間三次元の六次元を扱い、境界条件の考慮が可能なモンテカルロ・シミュレーション等、様々なシミュレーション技法を用いた計算が提案されている。モンテカルロ・シミュレーションにおいて、より精度の高い解を得るためには試行回数の増加、すなわちサンプル粒子数を増加させなくてはならない。さらに、ガス分子と電子との電離衝突による試行粒子数の指数関数的増加現象が起こるため、シミュレーションの高速化が必要とされている。また、半導体薄膜の微細加工においては、電子、イオン密度、エネルギーおよびイオンの入射角が重要であり、電界と磁界をもちいてプラズマの制御が行われている。磁界を扱うシミュレーションのうち、流体やハイブリッドモデルでは荷電粒子の輸送に関し電子輸送パラメータを取り扱うが、位置により複雑に交差する電磁界下での電子輸送パラメータの取り扱いが容易ではない。モンテカルロ・シミュレーションにおいては高磁界になるにつれ電子がエネルギーを得にくくなり収束解を得るまでの時間を要し、そ

れに伴い計算時間が増す。磁界を考慮した研究は数多くなされており、Nakamura ら<sup>[13]</sup>はモンテカルロ法にて直流電界下において直角に磁界が交差する  $E \times B$  フィールドでのシミュレーションを行っている。  $E/N$ ,  $B/p$  を変更し電子輸送パラメータとなるドリフト速度を計算し、Tunability Range と呼ばれるプラズマ制御に適した領域をメタンとアルゴンガスについて導き出している。White ら<sup>[14],[15]</sup>はボルツマン方程式を解き、直流電界下で任意の角度における電界と磁界についてシミュレーションを行っており、メタンガスにて速度分布関数を報告している。Ness および Makabe ら<sup>[16]</sup>は誘導結合高周波プラズマにおける電磁界のある特定の値に対し、電子スウォームの平均エネルギーが減少する領域および減少する領域が存在することを報告している。Kondo ら<sup>[17],[18]</sup>は、Ar ガス、誘導結合高周波プラズマにおける電界と磁界が交差する場における電子輸送パラメータ、 $E \times B$  ドリフトおよび電子エネルギーの増加と損失機構について述べている。

#### 1-4 本論文の目的と構成

本論文では、Java 言語によりモンテカルロ・シミュレーションに特化したシミュレーションの開発を行い、分散環境で並列計算を実現し、通信時間、スケジューリング、同時実行制御および同期制御の手法について述べる。並列計算の性能評価を行う上で、計算時間を要する問題として、初めに電界と磁界が任意の角度で交差する場において、シランガス ( $\text{SiH}_4$ )、クロラインガス ( $\text{Cl}_2$ )、アルゴンガス (Ar) を対象としたモンテカルロ・シミュレーションを行い、その結果として電子輸送パラメータの報告を行う。次に電磁界を利用した、より現実的なシミュレーションとして、位置により複雑に電界と磁界とが交差するカusp磁場でのモンテカルロ・シミュレーションを行い、その結果について述べる。最後に、本論文の目的とする高効率並列計算の性能評価を行い、その結果について述べる。

第1章は序論であり、スーパーコンピュータによる並列計算からネットワークを利用した様々な並列計算の性能および磁界を考慮したプラズマ・プロ



セス・シミュレーションに関する従来の研究について述べる。第2章は、モンテカルロ法によるシミュレーション技法およびプラズマ・プロセス・シミュレーションにおけるモンテカルロ法の適用方法について述べる。第3章は、並列計算手法として、Java言語による分散環境の構築方法、同時実行制御および同期制御方法、および本手法による高効率な並列計算を実現する理論について述べる。第4章は、電磁界での荷電粒子の運動を取り上げ、シミュレーション結果としてシランガス、クロラインガス、アルゴンガスを用いて、電界と磁界が任意の角度で交差する場における荷電粒子の輸送パラメータに関する結果について述べ、また、シミュレーションの応用分野として、電界と磁界が場所により複雑に交差するカusp磁場中でのモンテカルロ・シミュレーションを行い、その結果について論考する。これらのシミュレーションを通して並列計算の速度効率に関する結果および高効率並列計算の理論値との比較を行い、その結果について考察を行う。

## 第2章 シミュレーション方法

### 2-1 まえがき

モンテカルロ・シミュレーションは、乱数を取り扱うシミュレーション技法の総称である。乱数とはある指定された確率分布をもつ数列であるが、乱数の利用がほとんどの電子計算機で行われる現在、乱数は算術的に計算機内部で随時発生させる疑似乱数を意味するものとなっている。モンテカルロ法が対象とする問題は、確率的事象と確率的要素を含まない決定論的事象とからなる。決定論的事象は、確率とは無関係であっても問題となる方程式に確率事象を見出せば、モンテカルロ法で数値的に解くことが可能となる。モンテカルロ・シミュレーションは、現象の素過程を支配する確率法則が既知であると仮定できるとき、これら素過程をあたかも実際に生起しているかのように計算機で試行させ、全体の挙動を調べる方法である。

本章では、モンテカルロ法によるシミュレーションとして、プラズマ・プロセスを対象とし、シミュレーション方法およびモデルについて述べる。本シミュレーションにおいては、マイクロなレベルにおける分子間の衝突が存在することを確率現象として取り扱って計算を行い、粒子の飛行時間、軌道計算および衝突判定方法について述べる。また、シミュレーション結果としてエネルギー分布を表すために用いる LPWS 法によるサンプリング方法について述べる。

### 2-2 モンテカルロ法によるシミュレーション

#### 2-2-1 シミュレーション方法

モンテカルロ法によりプラズマ・プロセスにおける現象を取り扱う場合、位置空間三次元と速度空間三次元の六次元を扱うことができ、境界条件を考慮できるため、他のシミュレーション方法よりもより仮定が少なく、より詳細に物理現象を取り扱うことができる。しかし、電子を一つ一つ追跡することとなるため膨大な計算時間を要し、少ない試行回数では統計変動の影響を

受け正確な解を求めることが困難となる。さらに、電子の電離現象により指数関数的に電子およびイオンが増加するため、追跡粒子数が増し膨大な計算時間を要することとなる。本論文で取り扱うプラズマ・プロセスのシミュレーションに関し、全体の流れを以下に示す。

シミュレーションにおいては、一つずつ電子の振る舞いを追跡することとなり、一つの電子に対し、

- 1) シミュレーションの初期動作となる粒子の出発
- 2) 電子がガス分子に衝突するまでの時間となる飛行時間の決定
- 3) 現在時刻から飛行時間を進めた時間までの電子の軌道計算
- 4) 3) に関わる時間における電磁界の計算
- 5) 現在時刻から飛行時間を進めた時間における電子の衝突種類の判定

を行い、これをシミュレーションの終了時刻まで繰り返す。本論文においては、シミュレーション時間の中に複数のサンプリング時刻を設定し、電子の挙動をサンプリングしている。

## 2-2-2 シミュレーションモデル

並列計算を含んだシミュレーション物理モデルを UML(Unified Modeling Language) にて図 2-1 に示す。UML はオブジェクト指向におけるクラスやオブジェクトの関連性を示すモデリング法である。クラスとは、データと機能を一对として併せ持つものである。シミュレーションの開始および並列化機構を持つ `simMain` は RMI を実装したクラスであり、その相互通信の明確化を保つためにインターフェースとしての `simMainImpl` を提供している。RMI の実装はサーバ側の役割を行うスケルトン (`simMain_skel`) クラスとクライアント側の役割を行うスタブ (`simMain_stub`) クラスに示されており、インターフェースを介すことによりネットワークを意識することなくオブジェクトの相互通信およびメソッドの起動を行うことが可能である。このことは、並列化モデルと、物理モデルを明確に切り離すことができるため非常に有効的な手法といえる。

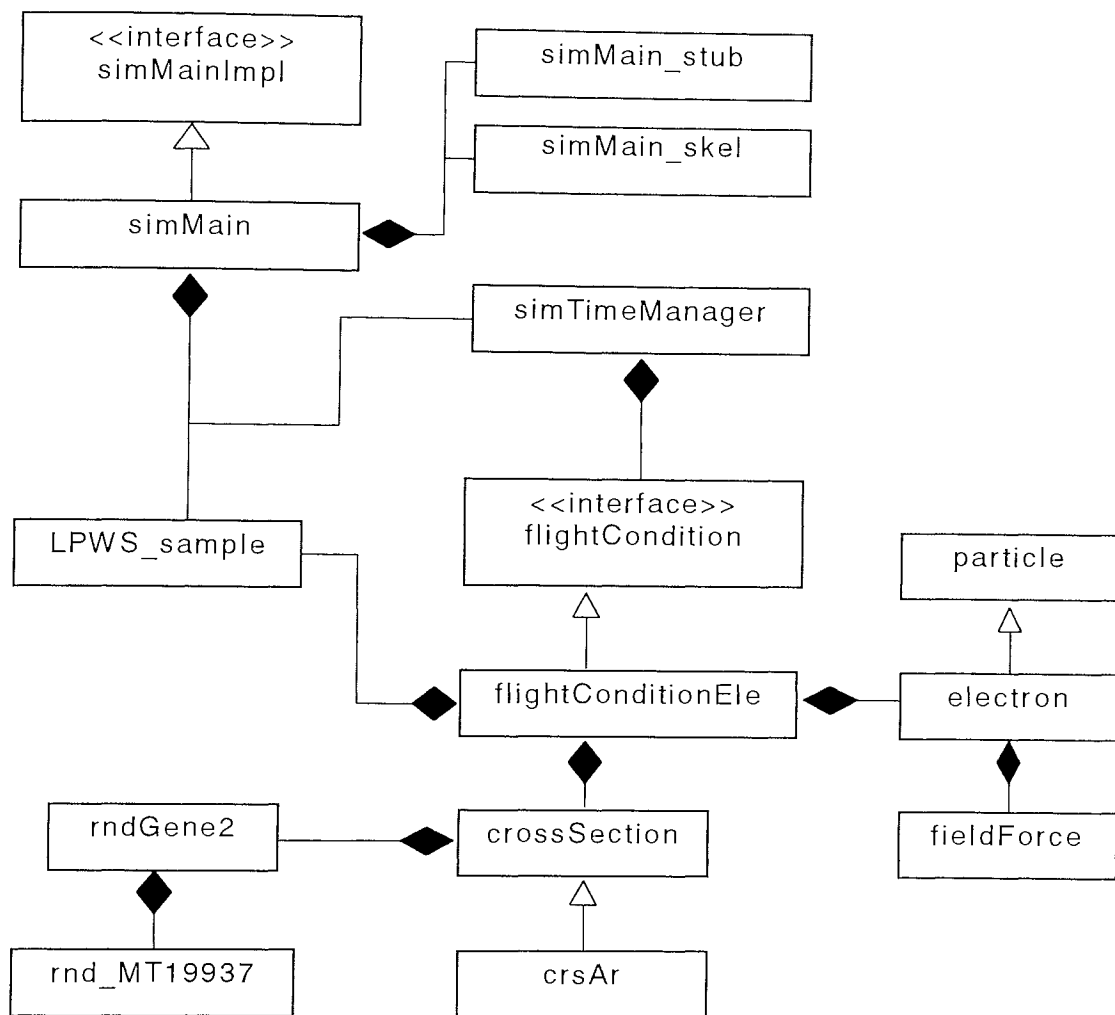


図 2-1 シミュレーションクラス図

本論文におけるシミュレーションはモンテカルロ法により任意角度の電磁界におけるプラズマ・プロセスを扱うものであり、電磁界およびその角度を扱う fieldForce、粒子を表す particle、particle を継承し電子およびその移動を扱う electron、衝突断面積を計算する crossSection、アルゴンガス断面積の crsAr、サンプリングを行う LPWS\_sample、これらを管理する flightConditionEle およびそのインターフェースとなる flightCondition、シミュレーション中の時間に関わる事項を統括管理する simTimeManager および、乱数発生器となる rnd\_MT19937、rndGene2 から構成される。

flightCondition インターフェースは、電子だけではなくイオンを取り扱うようにモデルが変更されたとき、容易に拡張可能となるため採用されている。また、シミュレーションの対象となるガスが変更されたときには、crsAr を変更する仕組みとなっており、オブジェクト指向設計が生かされている。rnd\_MT19937 は Matsumoto and Nishimura<sup>[19]</sup>により開発された Mersenne Twister を採用しており、乱数発生に関し長周期性、高速性および再現性に優れている。LPWS\_sample では電子輸送パラメータのサンプリングを行う。

### 2-3 サンプリング方法

本シミュレーションにおいては、初期条件で与えられたシミュレーション時間内においてサンプリング時間を設定し、各サンプリング時間において、電子のエネルギーおよび速度分布を 1 次元の LPWS<sup>[20]</sup>法によりサンプリングしている。また、電子の平均エネルギー、平均速度、平均位置、各種衝突の平均衝突周波数を度数分布にて表す。各種衝突の平均衝突周波数はガスの種類により異なり、Ar ガスの場合は電離衝突周波数および衝突の種類に関わらない全衝突周波数をサンプリングし、Cl<sub>2</sub> ガスの場合は、電離、付着衝突周波数および全衝突周波数をサンプリングし、SiH<sub>4</sub> ガスの場合は、電離、付着、解離衝突周波数および全衝突周波数のサンプリングを行う。

LPWS<sup>[20]</sup>法によるサンプリングはサンプル区間内においてルジャンドル多項式の高次の重みを付けてサンプルを行うことにより、区間内をルジャンドル展開したことと同様の効果があり、サンプル区間内の密度変化を滑らかかつ詳細に表すことができる。そのため、連続の方程式などで扱われる電子輸送パラメータとしての電子密度分布や速度分布のサンプルに対し非常に有効であると考えられる。また、LPWS 法によるサンプリングは、度数分布と同様に位置空間を分割してサンプルすることもできるが、1 区間でもサンプリングが可能であり、より少ない区間でサンプリングを行うことにより従来の度数分布を用いたサンプリング手法に比較し、同様の粒子数でシミュレーションを行っても統計変動の影響を受け難いという利点がある。

区間内での密度変化  $n(x)$  はルジャンドル多項式を用いて以下のように表される。

$$n(x) = \sum_{i=0}^{\infty} \tilde{n}_i P_i(x) \quad (2-1)$$

$$\tilde{n}_i = \frac{2i+1}{2} \int_{-1}^1 P_i(x_{\xi}) dx \quad (2-2)$$

ここで、 $i$  はルジャンドル多項式の項数を示しており、本論文においては第 6 項目までを用いている。また、サンプリングの重みとなる  $P_i(x)$  の値は以下の式から求まる。

$$P_{n+1}(x) = \frac{2n+1}{n+1} x P_n(x) - \frac{n}{n+1} P_{n-1}(x) \quad (2-3a)$$

$$P'_{n+1}(x) - P'_{n-1}(x) = (2n+1) P_n(x) \quad (2-3b)$$

$$P_n(1) = 1 \quad (2-3c)$$

$$P_n(-1) = (-1)^n \quad (2-3d)$$

以上より、第 6 項までのサンプリングの重みは次式となる。

$$P_0(x) = 1 \quad (2-4a)$$

$$P_1(x) = x \quad (2-4b)$$

$$P_2(x) = \frac{1}{2}(3x^2 - 1) \quad (2-4c)$$

$$P_3(x) = \frac{1}{2}(5x^3 - 3x) \quad (2-4d)$$

$$P_4(x) = \frac{1}{8}(35x^4 - 30x^2 + 3) \quad (2-4e)$$

$$P_5(x) = \frac{1}{8}(63x^5 - 70x^3 + 15x) \quad (2-4f)$$

$$P_6(x) = \frac{1}{48}(693x^6 - 945x^4 + 315x^2 - 15) \quad (2-4g)$$

したがって式(2-4a)からも、ルジャンドル多項式の 0 項目のみを利用したサンプリングは、重みが常に 1 となり度数分布と一致することがわかる。

## 2-4 飛行時間の決定

モンテカルロ・シミュレーションにおいて試行される荷電粒子の挙動として、まず初めに飛行時間、すなわち、荷電粒子とガス分子が衝突するまでの時間を決定する。荷電粒子は衝突から衝突の間（飛行時間内）に電界からエネルギーを得ながらその位置を変化させていく。飛行時間の終わりを衝突時とし、弾性、励起、電離、付着、解離等のいずれかの衝突を起こし、位置を変化させずにエネルギー損失が起こるものとする。

密度  $N$  の気体中を速度  $v$  で飛行しようとする電子は、衝突断面積を  $q(v)$  として、微小時間  $\Delta t$  間で、 $Nvq(v)\Delta t$  の衝突確率をとる。衝突確率は衝突周波数を  $f=Nvq(v)\Delta t$  とすると、 $f\Delta t$  として表せる。時刻  $t$  までに衝突しない確率は時刻  $t$  までに少なくとも 1 回は衝突する確率を  $P(t)$  とすると、 $1-P(t)$  で表され、飛行時間が時刻  $t$  と  $t+\Delta t$  間となる確率は、 $(1-P(t))f\Delta t$  となる。確率  $P(t+\Delta t)$  は、確率  $P(t)$  との確率  $(1-P(t))f\Delta t$  の和で表され、 $\Delta t \rightarrow 0$  とすることにより、微分方程式が得られる。

$$\left. \begin{aligned} P(t+\Delta t) &= P(t) + (1-P)f\Delta t \\ P(t) &= 1 - \exp\left(-\int_0^t f dt'\right) \end{aligned} \right\} \quad (2-5)$$

$P(0)=0$  を与えると、飛行時間  $t$  の確率密度  $p(t)$  を得ることができる。

$$p(t) = f \exp\left(-\int_0^t f dt'\right) \quad (2-6)$$

式(2-6)を積分することで、飛行時間確率密度  $p(t)$  となる飛行時間  $t$  を決定でき、式(2-5)と同形となる。確率  $P(t)$  は、0から1までの値を取り、これに一様乱数  $\xi$  を対応させることで  $t$  が求まるが、このとき新たに仮想の衝突断面積を加え、 $f$  を一定値とすることが可能となり、

$$t = -\ln(1-\xi)/f \quad (2-7)$$

として粒子の飛行時間  $t$  が求められる。

飛行時間の決定時に現在時間と飛行時間の和となる時間がシミュレーショ

ン時間を超えていた場合、粒子の追跡は終了する。また、現在時間と飛行時間を足した時間がサンプル時間を超えていた場合は、サンプル時間までの粒子の軌道を計算し、サンプルを行い、残りの時間の軌道計算後に衝突判定を行う。

## 2-5 衝突の判定

電子とガス分子との衝突現象は、電子が保持するエネルギーおよびガスの種類により異なる。衝突の判定は、エネルギーの関数となるガス分子が持つ衝突断面積を用いて行われ、衝突断面積の値が大きければ衝突しやすく、衝突断面積の値が小さければ衝突し難い。衝突断面積の値は、ガス分子の種類により異なり、電子が持つエネルギーにより衝突断面積の値が決定される。本シミュレーションにおける衝突判定は、Null Collision法<sup>[9]</sup>により行われる。Null Collision法はガス分子が持つ衝突断面積から衝突周波数を求め、乱数を用いて衝突の種類を判定する手法である。式(2-7)は仮想の衝突断面積を加えることにより得られたが、仮想の断面積を含め $f$ を一定値とすることでエネルギーに依らず無効衝突を含む衝突が必ず起こると考えることができるため、効率よく衝突判定が行えるのがNull Collision法の特徴である。

0 から 1 までの一様乱数を $\xi$ 、ガス分子が持つ衝突周波数を $f_i$ 、無効衝突を加えた各衝突周波数の合計を $f_{max}$ 、電子のエネルギーを $\epsilon$ とすると電子の衝突判定は以下の式で行える。

$$\xi \leq f_1(\epsilon) / f_{max} \quad (\text{衝突 1})$$

$$f_1(\epsilon) / f_{max} < \xi \leq [f_1(\epsilon) + f_2(\epsilon)] / f_{max} \quad (\text{衝突 2})$$

...

$$\sum_{j=1, N} f_j(\epsilon) / f_{max} < \xi \quad (\text{無効衝突}) \quad (2-8)$$

無効衝突が選択されたとき、飛行時間に衝突が起こらなかったこととして処理される。



## 2-6 軌道の計算

飛行時間が決定され衝突判定後、無効衝突ではない場合に電子の散乱方向を決定することとなる。

電子のエネルギー損失が生じる場合エネルギーの再計算を行い、それに伴い速度を

$$v = \sqrt{\varepsilon \frac{2e}{M_e}} \quad (2-9)$$

として求める。ここで、 $\varepsilon$ はエネルギー損失後の電子のエネルギー、 $e$ は電荷 ( $1.6021892 \times 10^{-19} \text{C}$ )、 $M_e$ は電子の質量 ( $9.109534 \times 10^{-31} \text{kg}$ ) である。

電子の散乱は

$$\cos \theta = 1 - 2\xi_1 \quad (2-10)$$

$$\sin \theta = \sqrt{1 - (\cos \theta)^2} \quad (2-11)$$

$$\cos \phi = \cos(2\pi\xi_2) \quad (2-12)$$

$$\sin \phi = \sin(2\pi\xi_2) \quad (2-13)$$

により方向が決定される。ここで、 $\xi_1$ ,  $\xi_2$ は0から1までの一様乱数である。

また、散乱後の各方向の速度ベクトルは、

$$v_x = v \cos \theta \quad (2-14)$$

$$v_y = v \sin \theta \cos \phi \quad (2-15)$$

$$v_z = v \sin^2 \theta \quad (2-16)$$

により求まる。

## 第3章 並列計算手法

### 3-1 まえがき

モンテカルロ・シミュレーションは、試行回数が少ない場合統計変動の影響を受け、結果に対する信頼性が低下する。一方、試行回数を多くすると計算時間が膨大になる問題がある。この問題を解決するために、本論文においては並列計算を提案する。並列計算は、短時間に高精度の解が得られるばかりではなく、遺伝子情報の解読等、解くことが困難なほど膨大な計算時間を必要とする問題にも応用が可能である。

並列計算を行うための環境はハードウェア、ソフトウェアが複雑に関係するため容易には構築できない。特にネットワークを介した分散環境上における並列計算環境はスーパーコンピュータの研究と同時に発生した共有メモリという概念を越えたものであり、個々のネットワーク PC の制御が求められる。本シミュレーションはオブジェクト指向言語である Java 言語を使用して開発を行った。Java 言語が持つ分散オブジェクト技術としての Remote Method Invocation (RMI) <sup>[21]</sup>および同期実行制御としてのマルチスレッド技術<sup>[22]</sup>を用い、LAN 上に配置したパーソナルコンピュータを制御し、分散環境の構築および並列計算の実装を行い、その方法について述べるとともに、高効率並列計算の理論について述べる。

### 3-2 並列計算環境の構築

本並列計算は、モンテカルロ・シミュレーションの試行回数を各分散 PC に振り分け、その各 PC で計算された統計情報を集計することにより結果を得るものである。従来の粒子シミュレーションにおいて、位置空間を分割し PC や PE に分配する並列計算が行われていた。本論文では、シミュレーションのプロセスの並列化ではなく、モンテカルロ法によるプラズマ・プロセスのシミュレーションは、個々の粒子の挙動を独立事象として扱うことが可能であることに着目し、一連の粒子の挙動に関する計算の並列化を行うモデル

とした。位置空間の分割では、粒子の空間移動により PE または PC 間に通信が必要となり、移動回数により計算時間の低下を招くことが予測されるが、本並列計算では、PC 間の通信は集計時に発生するのみとなるため、通信回数を減らすことが可能である。また、並列計算を行うためのコンピュータはスーパーコンピュータやワークステーション等、高性能なコンピュータではなく、一般的な PC を用いることができ、マルチ CPU 環境等も必要としない。インターネットや LAN に代表される、TCP/IP プロトコル上のネットワークに接続されているだけで十分であり、特別な環境ではなくとも並列化計算が可能である。また、このような分散環境では CPU パワーを分散させるだけでなく、消費メモリも分散させるメリットがあり、接続する PC の数に限界はない。

本並列計算手法は、オブジェクト指向言語である Java 言語によりネットワーク上に分散環境の構築を行い CPU とメモリについて負荷分散し、シミュレーション環境を実現するものである。環境の構築および各 PC 間でのデータの送受信に、分散オブジェクト技術を用いている。分散オブジェクト技術はネットワーク・コンピューティングに関し、オブジェクト技術を適用したものであり、この技術はネットワーク技術とオブジェクト技術の統合した技術といえ、従来用いられていた RPC のオブジェクト指向化を計った技術である。ネットワーク技術が情報理論や帯域圧縮技術といった定量的な方法論を保有し、技術基盤が確立したものであるのに対し、オブジェクト技術は必ずしも明確な方法論が確立していない発展過程の技術である。しかし、データと機能を併せ持つオブジェクトの使用により、ネットワークを介した情報伝達を明確にすることが可能であり、また、再利用性を強く持つことから開発効率が向上するという利点がある。

分散オブジェクトを実現する方法として、OMG による IDL, CORBA、Microsoft による COM/DCOM、SUN Microsystems による Java IDL, Java RMI 等がある<sup>[23]</sup>。本論文における分散環境の構築は Java RMI を用いて実現している。RMI はネットワークを介して他のマシン上にあるオブジェクト

のメソッドを呼び出すための仕組みであり、ネットワークを介したメソッドの呼び出しが可能なオブジェクトは、分散オブジェクトとして振る舞うことが可能となる。Java や C++などのオブジェクト指向の言語では、処理の多くは他のオブジェクトのメソッド呼び出しとして記述される。分散オブジェクトを構成することにより、ネットワークをあまり意識しなくても自然な形でネットワーク上にシミュレーション計算サーバを点在させることが実現可能となる。RMIをはじめとする多くの分散オブジェクトシステムは、ネットワークの取り扱いに関する部分はブラックボックスとして覆い隠し、特別の知識がなくても開発が行える仕組みとなっている。また、他の PC 上のオブジェクトを制御するためには、あらかじめ特定のプロトコルに基づいて通信を行う必要がある、分散オブジェクト技術を使用しなければ TCP のコネクションに関するコードを記述しなくてはならない。RMI システムは TCP のコネクションによる通信の上に、さらに独自のプロトコルを構築して実現でき、より柔軟な分散環境を構築可能とするため、計算の開始、PC 間におけるパラメータの伝達に代表される様々な機能の呼び出しが必要となるシミュレーション環境の構築に適したものといえる。

RMI による分散環境上で動作するシミュレーションは、サーバとクライアントという二つのプログラムから構成され、サーバアプリケーションは、数多くのリモートオブジェクトを作成し、それらのリモートオブジェクトへの参照をアクセス可能にし、クライアントがそれらのリモートオブジェクトのメソッドを呼び出すのを待つ。クライアントアプリケーションは、サーバ内の一つまたは複数のリモートオブジェクトへのリモート参照を取得し、それらのオブジェクトのメソッドを呼び出す。RMI は、サーバおよびクライアント間で通信や情報の受け渡しを行う機構を提供する。図 3-1 は、リモートオブジェクトへの参照の取得にレジストリを使用する RMI 分散アプリケーションを示し、サーバはリモートオブジェクトに名前を関連付けるために RMI レジストリを呼び出す。クライアントはサーバのレジストリ内の名前によってリモートオブジェクトを検索し、そのオブジェクトのメソッドを呼び出す。

RMIにより構成されたシステムで、既存の Web サーバを利用して、サーバからクライアント、およびクライアントからサーバに、必要に応じてオブジェクトの Java クラスバイトコードをロードする状態を示し、RMI ではシステムのサポートする任意の URL プロトコル (HTTP、FTP、ファイルなど) を利用し、クラスバイトコードをロードする。

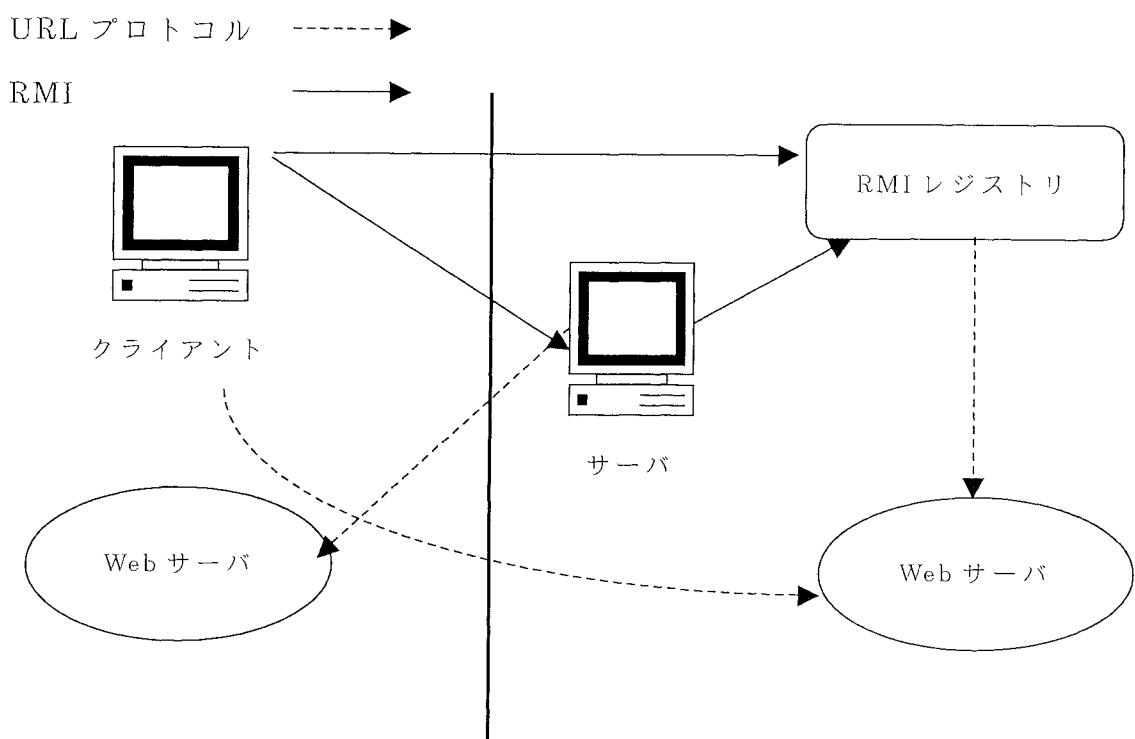


図 3-1 RMI システム

RMI では、リモートオブジェクトであるクライアント上のスタブオブジェクトおよびサーバ上のスケルトンオブジェクトとの通信に、RPC システムで用いられる標準の機構が使われる。スタブは、リモートオブジェクトに対するクライアントのローカルでの代理、つまりプロキシの役割を果たす。呼び出し側は、リモートオブジェクトに対するメソッド呼び出しを行うローカルスタブ上のメソッドを呼び出す。RMI では、リモートオブジェクトのスタブは、リモートオブジェクトが実装している一連のリモートインタフェースと

同じインタフェースを実装する。

スタブのメソッドは、呼び出されると、次の動作を順次実行する。

- 1) リモートオブジェクトを含むリモート VM との通信の開始
- 2) リモート VM へのパラメータの整列化 (書き込みおよび転送)
- 3) メソッド呼び出しの結果の待機
- 4) 戻り値または発行された例外の非整列化 (読み取り)
- 5) 呼び出し側に値を返す

スタブが、ネットワーク上での通信において必要とされるパラメータの直列化およびネットワークレベルの通信を隠すことにより、通信に必要となる手続きを単純化させることが可能となる。直列化とは、ネットワーク上に情報を伝達させる時に必要とされるデータの形式であり、最終的にデータ伝送を行う時にはデータをエンコードしなくてはならないが、利用者はそれを意識することなくスタブが自動的に行ってくれる。リモート VM では、各リモートオブジェクトに対応するスケルトンがある場合がある。スタブからの呼び出しは、スケルトンにより中継され実際のリモートオブジェクトへ処理が依頼される。スケルトンは、メソッド呼び出しを受け取ると次の動作を順次実行する。

- 1) リモートメソッド用パラメータの非整列化 (読み取り)
- 2) 実際のリモートオブジェクトの実装上にあるメソッドの呼び出し
- 3) 呼び出し側に対する結果 (戻り値または例外) の整列化 (書き込みおよび転送)

スタブおよびスケルトンは、Java 言語に含まれる `rmic` コンパイラによって容易に生成できる。

本論文で構築した分散環境は図 3-2 のように 1 台の PC に 2 台の PC が接続される階層化構造で各分散 PC の配置により構築されている。分散 PC の配置は、初期起動パラメータにより、サーバ 1 に対してクライアントが複数存在するような形態に変更が可能である。階層化構造で構築した分散環境は、上位層の PC が下位層の PC を制御する仕組みとなっている。

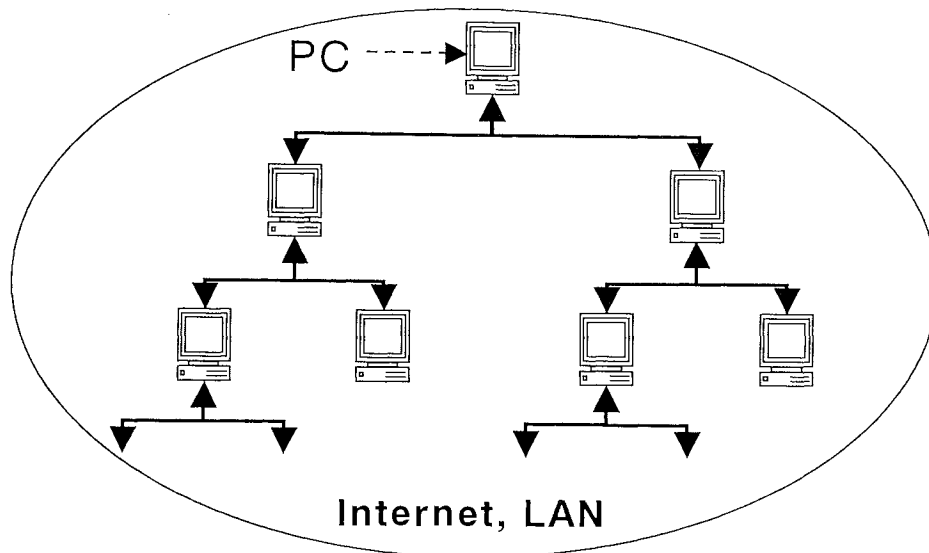


図 3-2 分散コンピュータの配置

### 3-3 同時実行制御および同期制御方法

シミュレーションにおける同時実行制御、同期制御はスレッド処理により行われる。起動パラメータの受け渡しおよび結果の集計における RMI によるメソッドの呼び出しをスレッドを用いて同期制御することにより、各分散 PC の制御を行っている。モデルが単純なため、スタンドアローンのシミュレーションモデルを並列計算モデルへ容易に拡張可能である。

図 3-3 は並列計算のスレッド化、同期化メカニズムを状態遷移図で表している。Client-Server フェーズはスレッドの生成および並列計算を開始するフェーズであり、シミュレーション条件の通知および開始依頼をする。開始依頼を受けた PC はそれぞれモンテカルロ・シミュレーションを行う。シミュレーション終了時刻に各 PC は依頼を行った PC に対してシミュレーション結果となるサンプルオブジェクトを送信する。このとき、Server-Client フェーズとなり、受信側の PC は発行したスレッドの同期を取りサンプルオブジェクトを待ち、受信したサンプルオブジェクトの合計処理を随時行う。

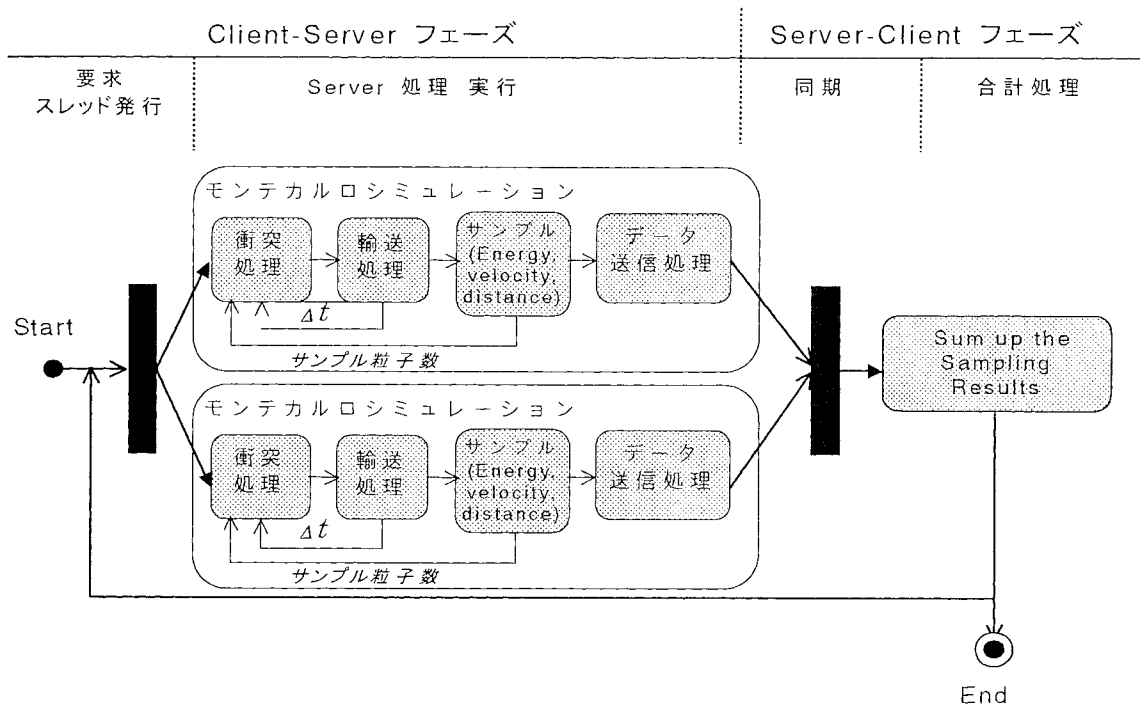
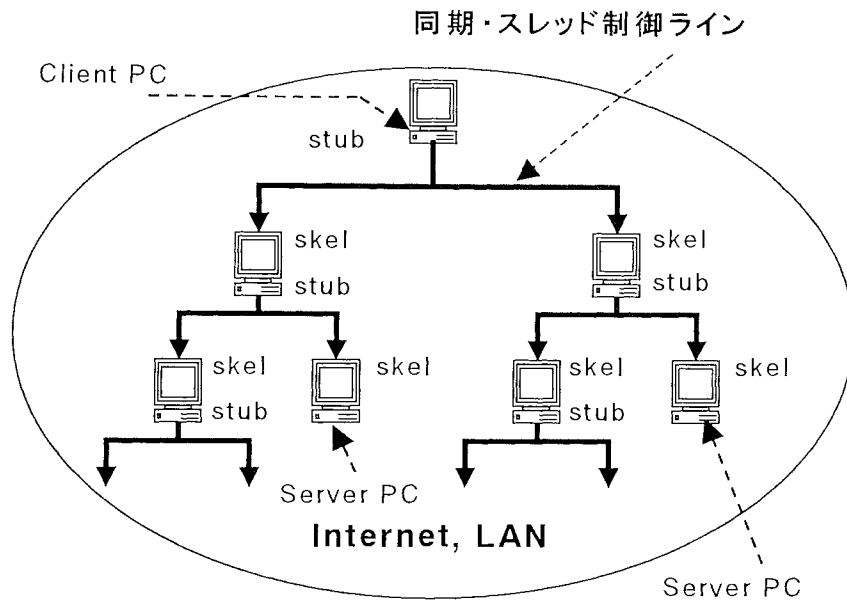


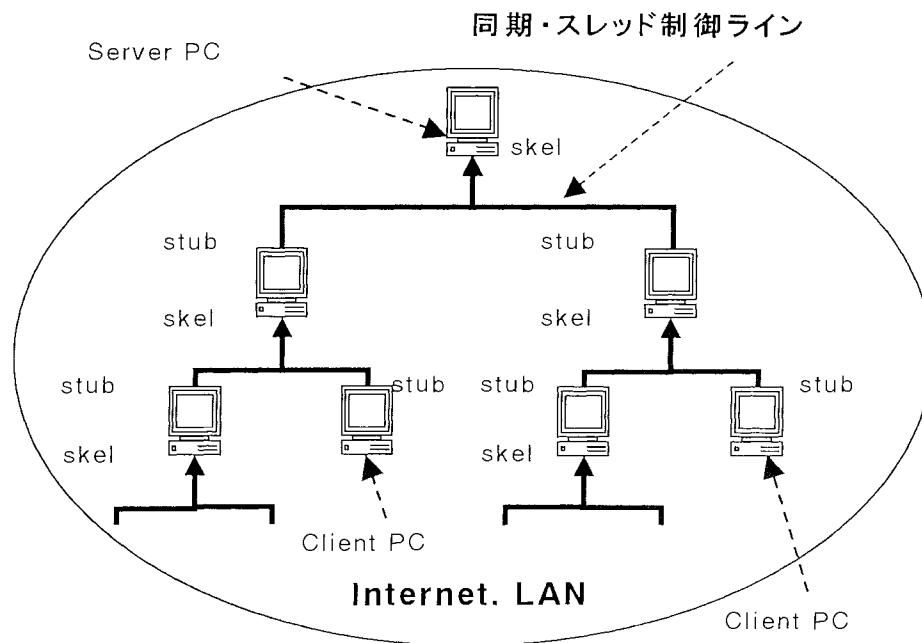
図 3-3 並列計算の状態遷移図

図 3-4 はネットワーク上に配置した PC の構造および PC 間の関連による 2 つのフェーズを示している。図 3-4 (a) は階層構造の上位の PC がクライアントであり依頼する側の PC となることを示している。逆に図 3-4 (b) では上位の PC がサーバであり依頼を受ける側の PC となることを示している。各フェーズは、厳密には RMI を利用する上で発生するスタブおよびスケルトンの関係により決定される。Client-Server フェーズでは下位層の PC がサーバの役割を行い、上位層の PC がクライアントとなり、パラメータの送信および計算の開始を依頼する。その後、各 PC はモンテカルロ・シミュレーションを実行する。計算終了後、Server-Client フェーズへ移行し、上位層の PC がサーバの役割を行い、下位層の PC がクライアントとなり同期化および集計処理が行われる。これらのフェーズは、スケルトンおよびスタブの役割により決定されている。計算速度は個々の PC におけるシミュレーションの終了時刻に関連する同期タイミングにより決定される。





(a) Client-Server フェーズ



(b) Server-Client フェーズ

図 3-4 ネットワーク上に配置した PC 構造と二つのフェーズ

Java 言語が持つマルチスレッド機能により並行プログラムを実現可能である。スレッドとは、1つのアドレス空間の中で実現された、並行プログラムのプロセスである。また、マルチスレッドとは、複数のスレッドを並行に実行することにより並行プログラムを実現する方法である。

マルチスレッドには以下のような特徴が挙げられる。

- ・ 実行時にプロセスの生成、削除が可能
- ・ 並行に実行される複数のスレッドから同じオブジェクトにアクセス可能
- ・ モニターの同期の仕組みが利用可能

Java のオブジェクトは自ら動作を起こす実体ではなく、生成したスレッドに処理内容の定義を与えるものである（図 3-5）。

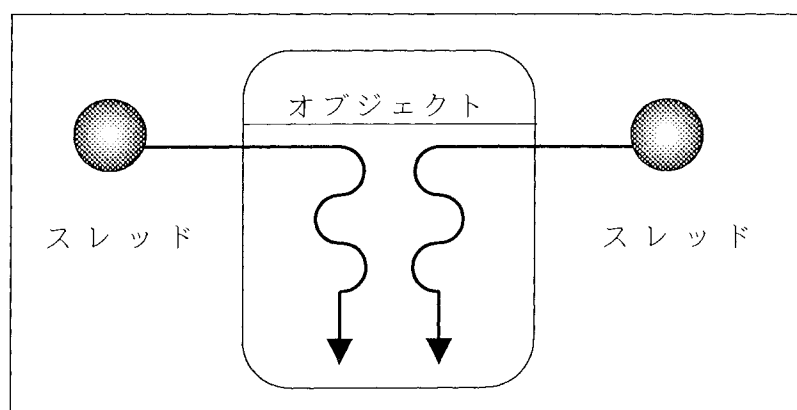


図 3-5 スレッドとオブジェクト

また Java のオブジェクトは、複数のスレッドから並行的に参照される共有データとしての役割を持つ。個々のオブジェクトは同期を実現するための機能を保持している（図 3-6）。

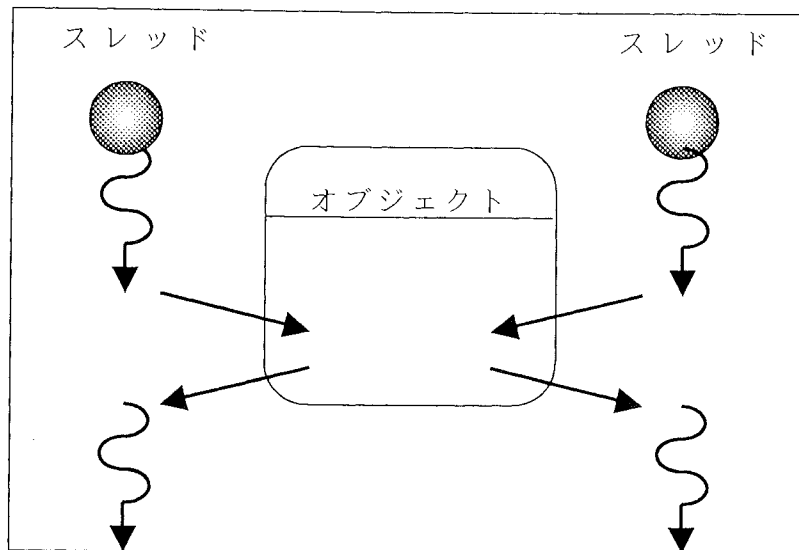


図 3-6 共有データとしてのオブジェクト

Java のスレッドを制御するには、Thread オブジェクトに対する操作を行う。Thread オブジェクトは、常に次に示す 4 つの状態のいずれかである。

- ・初期状態

Thread オブジェクトが生成されたがまだ実行されていない状態。

- ・実行可能状態

スレッドの処理を実行中であるか、あるいはスケジューリングの都合により実行を開始したりするのを待っている状態。

- ・ブロック状態

スレッドの処理を先に進めることができないような理由があり、実行可能状態に戻るのを待っている状態。

- ・終了状態

スレッドの処理を実行し終わった状態。

スレッドの処理が行われるのは、実行可能状態にあるときだけである。実行可能状態のスレッドが複数ある場合は、スケジューラがスレッドを選択し次々実行させる。Thread オブジェクトに対して start メソッドを起動すると、そのインスタンスの run メソッドを実行するような新しいスレッドが実行可

能状態になる。run メソッドは通常のメソッドと同様にプログラム中から明示的に呼び出すこともできるが、通常はユーザが直接呼び出す必要はなく、スレッドの処理が始まる時に自動的に呼ばれる。run メソッドの実行が終わると、スレッドの処理が終わる。実行中のスレッドに対して、明示的に stop メソッドを呼んでも終了状態になる。I/O の処理などのため実行を一時的に中断せざるを得ない場合、スレッドはブロック状態になることがある。また、明示的に suspend メソッドや sleep メソッドや wait メソッドを呼び、ブロック状態にすることも可能である。suspend メソッドによって、ブロック状態になったスレッドは、resume メソッドにより実行可能状態に戻る。また、sleep メソッドによってブロック状態になったスレッドは、待ち時間を消化すれば自動的に実行可能状態に戻る。wait メソッドによってブロック状態になったスレッドは、notify メソッドや notifyAll メソッドによって実行可能状態に戻る。

Java 言語の同期のメカニズムは、オブジェクトごとのロックによって実現される (図 3-7)。Java のオブジェクトはロック状態の場合に、別のスレッドがそのオブジェクトをロックしようとする、そのスレッドはオブジェクトのロックが解除されるまでブロック状態になる。おのおののオブジェクトは、それぞれをロックしようとして待っているスレッドの集合を暗黙のうちに管理している。オブジェクトのロックが解除されるとそのオブジェクトが管理しているブロック状態のスレッドの 1 つが集合から外されて実行状態に戻り、そのオブジェクトは再びロックされる (図 3-8)。

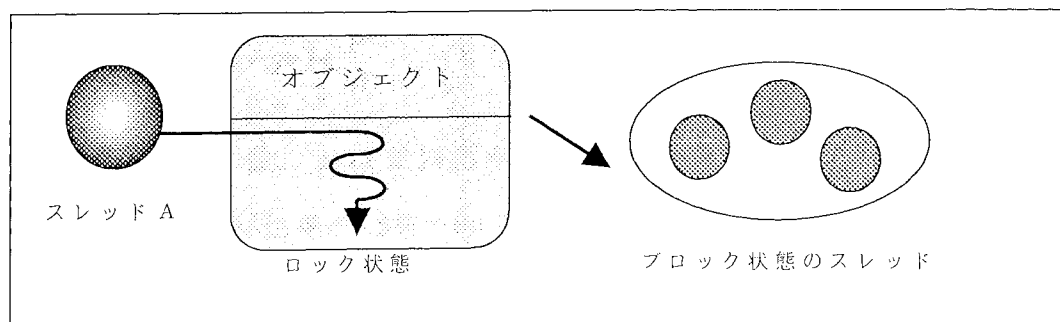


図 3-7 オブジェクトのロックとブロック状態のスレッド

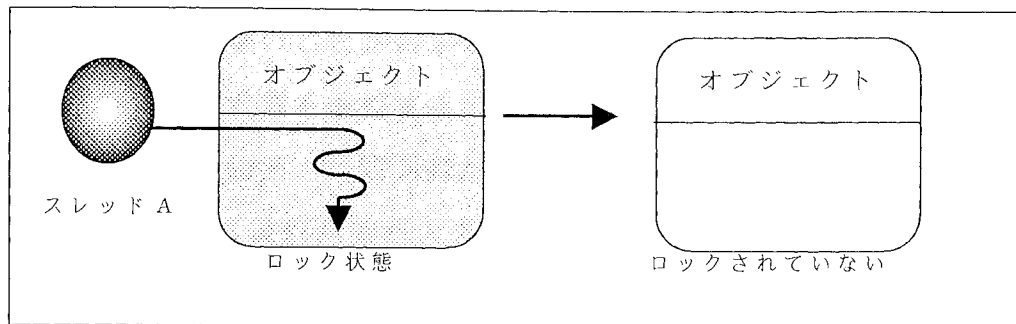


図 3-8 オブジェクトのロックと参照

同期を実現するために、Java では `synchronized` メソッドを用意しており、同じオブジェクトに対して同時に操作を行わないことを保証している。`synchronized` メソッドが起動されるとそのオブジェクトはロックされ、メソッドの処理が完了するとオブジェクトのロックが解除される。別のスレッドが同じオブジェクトに対して `synchronized` メソッドを起動すると、ロックが解除されるまでブロック状態になる。

### 3-4 高効率並列計算

#### 3-4-1 一般的な並列計算速度効率

並列計算を行った場合、一般的にはコンピュータの台数を増加させても、ある程度の台数で速度が上昇しなくなる。このことはアムダールの法則により明らかにされており次式で表される<sup>[24]</sup>。

$$speedup(n) = \frac{1}{(1 - p_r) + \frac{p_r}{n}} \quad (3-1)$$

ここで、 $n$  は CPU 数、 $p_r$  は並列化率を表しており、両条件下での計算速度倍率を導き出す。並列化率 99.5%~80.0%までの台数効果について図 3-9 に示す。わずか 0.5%の並列化率の低下により、顕著に台数効果が失われているのがわかる。並列化率 96.5%の場合 CPU 数 30 において、半分の性能しか引き出すことができず、また、並列化率 90%以下においては 10 以上の CPU を用いて

も、ほとんど効果がないことがわかる。このように、一般的には台数が増加しても高効率な並列計算を維持するのは非常に困難とされており、これらの速度低下は同時実行制御、同期制御だけではなく、CPU間またはネットワーク上のPC間における通信時間が大きく左右する。

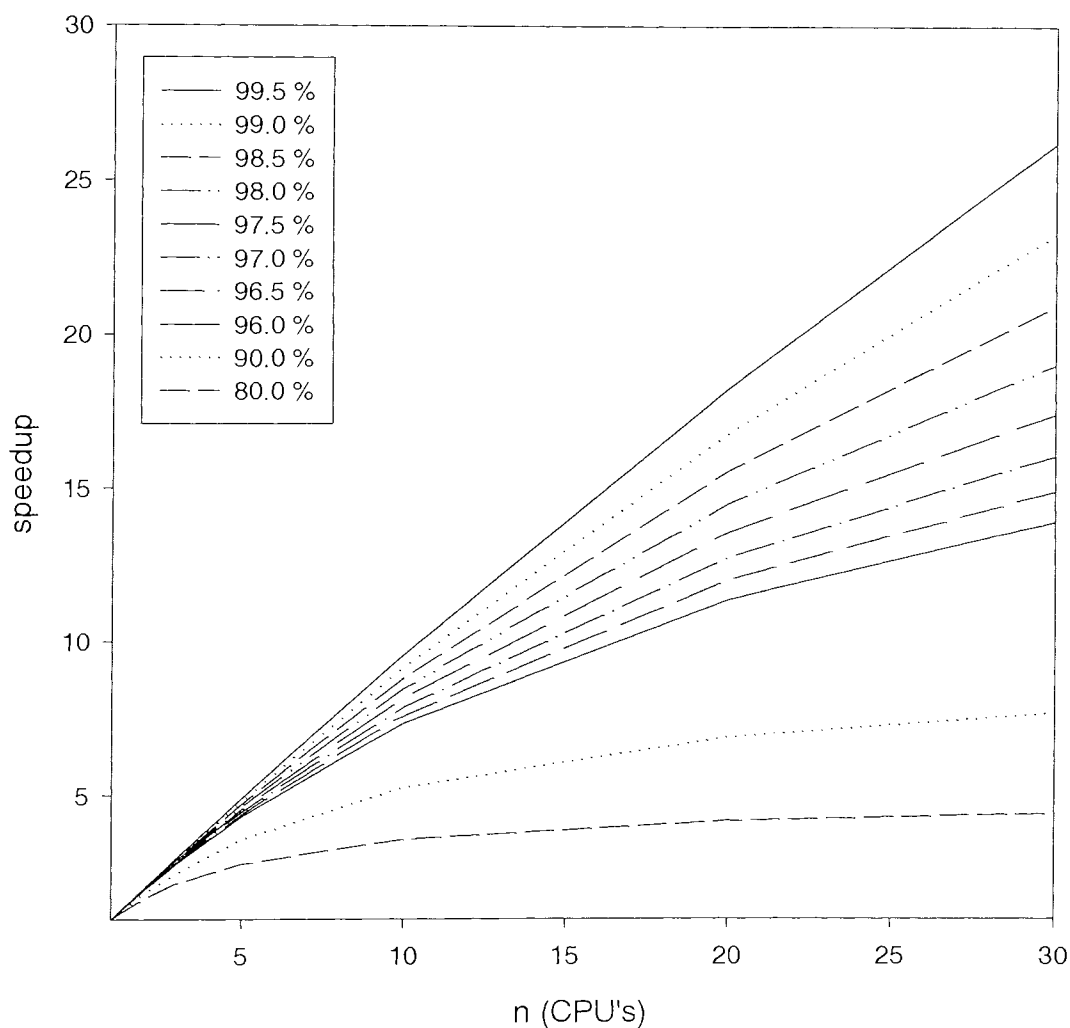


図 3-9 アムダールの法則による台数効果

### 3-4-2 高効率並列計算の理論

モンテカルロ法によるプラズマ・プロセス・シミュレーションは、一つ一つの粒子に対して試行し、最終的に全ての粒子に対する計算が終了するまで繰り返される。一つの粒子に対する計算の流れは、粒子の出発、衝突時刻の

決定、衝突種類の判定、軌道計算、サンプリングの順に、シミュレーション終了時刻に達するまで繰り返される。本並列計算では、シミュレーションを行う PC に試行する粒子を振り分け、その結果を集計することにより並列計算を行う。このとき、各 PC のシミュレーション結果を同期制御し、結果を得ることになるが、各 PC のシミュレーション終了時刻は、粒子の衝突回数に大きく起因し、計算時間に影響を及ぼす。これは、シミュレーションの終了時刻までの間に、粒子の軌道計算およびサンプル回数は変化しないが、粒子が衝突するほど衝突種類の判定を行う回数が増加するためである。

シミュレーション時間を  $T_{sim}$ 、シミュレーションを行う粒子数を  $n$ 、 $i$  番目の粒子を  $P_i$ 、粒子  $P_i$  の衝突回数を  $N_i$  とすると、 $T_{sim}$  中の平均衝突回数は、

$$\bar{N} = \frac{1}{n} \sum_{i=1}^n N_i \quad (3-2)$$

となり、 $n$  個の粒子  $P_1 \sim P_n$  を計算するのに要する全 CPU 時間  $T_{cal}$  は、

$$T_{cal} = t_1 \sum_{i=1}^n N_i = n t_1 \bar{N} \quad (3-3)$$

となる。このとき、 $t_1$  は、1 回の衝突に要する計算時間である。

次に、以下のように、粒子  $P_1 \sim P_n$  を分散環境上の  $m$  台の PC に均等に  $k$  個ずつ振り分ける。

$$P_1 \sim P_k, P_{k+1} \sim P_{2k}, P_{2k+1} \sim P_{3k}, \dots, P_{(j-1)k+1} \sim P_{jk}, \dots, P_{(m-1)k+1} \sim P_{mk}$$

$j$  組目の PC の計算時間は、

$$T_{cal, j} = t_1 \sum_{i=(j-1)k+1}^{jk} N_i = k t_1 \bar{N}'_j \quad (3-4)$$

で表され、 $k$  個の粒子の平均衝突回数は、

$$\bar{N}'_j = \frac{1}{k} \sum_{i=(j-1)k+1}^{jk} N_i \quad j=1, 2, \dots, m \quad (3-5)$$

となる。このとき、 $k$  が非常に大きな場合、平均の衝突回数の関係は、

$$\bar{N} \cong \bar{N}' \quad (3-6)$$

となり、計算時間の関係は以下のようなになる。

$$\frac{1}{m} n t_1 \bar{N} = \frac{1}{m} T_{cal} \quad (3-7)$$

$$T_{cal, j} \cong \frac{1}{m} T_{cal} \quad (3-8)$$

したがって、1台のPCに対して十分な粒子数を割り当てることで衝突回数に対し統計変動が押さえられ高効率な並列が行えると考えられる。また、並列計算の速度に起因する通信時間については、1台のPCに対して十分な粒子数を割り当てることにより、無視できるほど少なくなることが考えられる。



## 第4章 電磁界中の荷電粒子の運動

### 4-1 まえがき

半導体薄膜製造におけるプラズマの加工技術においては、高い精度が要求されており、プラズマの構造を十分理解し、制御しなければならない<sup>[1],[2]</sup>。プラズマの精密な制御を行うためには、低気圧で高密度のプラズマが必要であり、磁界を用いたプラズマの生成法が提案されている。磁界を用いた場合、電界のみの場合よりも荷電粒子の運動は複雑になり、高度な制御を行うためには、プラズマ構造を十分理解するためのプラズマの生成方法における様々なシミュレーションが行われている<sup>[25]-[28]</sup>。特に磁界を印加した場合の基礎となる電界と磁界が直交した  $E \times B$  フィールドにおいて、種々のガスについて報告されている<sup>[13]-[16], [29]-[36]</sup>。しかし、ECR プラズマやマグネトロンプラズマなどのプラズマ生成法では、場所により電界と磁界が直交しておらず、任意の電界と磁界の角度によるプラズマ構造を知ることが重要である<sup>[37]-[39]</sup>。本論文では、Nakamura ら<sup>[13]</sup>の研究に基づき Ar ガスについて、より実用的なシミュレーションとして、薄膜製造に用いられる  $\text{SiH}_4$  ガスおよびエッチングに用いられる  $\text{Cl}_2$  ガスを対象とし、電界と磁界の任意の角度に対して、速度、平均エネルギー、エネルギー分布および衝突周波数などのパラメータについて報告する。また、より現実的な応用範囲として、電磁界が複雑に作用するカusp磁場のモンテカルロ・シミュレーションの結果について述べる。本章の最後には、これら計算時間を要する磁界を考慮したモンテカルロ・シミュレーションについて、並列計算を行った結果を述べるとともに、第3章で述べた高効率並列計算の理論の検証を行った結果について述べる。

### 4-2 電界と磁界による荷電粒子の運動

#### 4-2-1 単一粒子の電界中の運動

荷電粒子は、衝突拡散のほかに電界によってもある方向に移動する。電界

がない場合には、これらの荷電粒子は密度の薄い方向に拡散する。電界が印加された場合、荷電粒子は密度に関係なく加速され、電子は負の電荷を持っているので電界と逆方向に移動し、イオンは正の電荷を持っているので電界方向に移動する。ただし、ガスの種類によって負イオンが生成される場合があり電子と同じ方向に移動する。このとき、これらの荷電粒子は中性分子と衝突し、ジグザグ運動をしながら電界によってある一定方向に進む。このことをドリフトと呼んでいる。電界  $E$  の方向に  $x$  軸をとり、質量  $m$ 、電荷  $q$  (電子のとき  $-e$ 、イオンのとき  $e$ ) の荷電粒子の運動は、初速度を  $0$  とし、 $x$  方向のニュートンの運動方程式

$$m \frac{dv}{dt} = qE \quad (4-1)$$

を積分することで速度が次のように求まる。

$$v = \frac{qE}{m} t \quad (4-2)$$

式(4-2)より、速度が時間  $t$  に正比例して増加することがわかる。しかし、実際には荷電粒子は、中性粒子と衝突するため式(4-1)の右辺に衝突項を入れたランジュヴァン方程式

$$m \frac{dv}{dt} = qE - mvf \quad (4-3)$$

が用いられ、荷電粒子が衝突時に運動量  $mv$  を失うとし、1秒間に  $f$  回衝突すると式(4-3)のように表される。式(4-3)の右辺の速度  $v$  にドリフト速度  $v_D$  を代入すると、

$$v_D = \frac{qE}{mf} \quad (4-4)$$

となる。

#### 4-2-2 単一粒子の磁界中の運動

電界がなく磁界  $B$  のみが存在する場合、電荷を  $q$ 、速度を  $v$  とすると荷電粒子に  $v$  と  $B$  に垂直方向のローレンツ力を受ける。したがって運動方程式は、

$$m \frac{dv}{dt} = qv \times B \quad (4-5)$$

となる。いま、 $z$  軸方向に磁界  $B$  が印加されているとし、荷電粒子が  $xy$  平面上を速度  $v$  で直進しようとした場合、 $z$  軸方向と進行方向に対して直角のローレンツ力を受け曲げられる。したがって、これを繰り返すことによって  $z$  軸方向と垂直な  $xy$  平面上で荷電粒子は円軌道を描くサイクロトロン運動を行い図 4-1 のようになる。

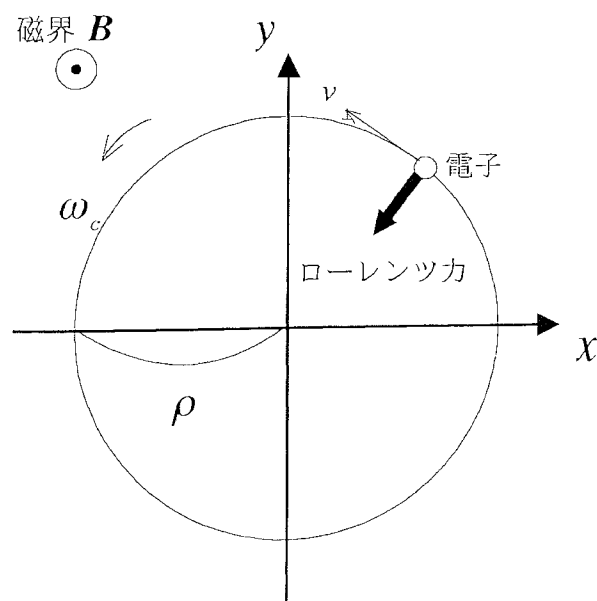


図 4-1 磁界中での電子のサイクロトロン運動

サイクロトロン角周波数  $\omega_c$  およびラーモア半径  $\rho$  は

$$\omega_c = \frac{eB}{m} \quad (4.6)$$

$$\rho = \frac{v_{\perp}}{\omega_c} \quad (4.7)$$

で求められる。

#### 4-2-3 単一粒子の直交電磁界中の運動

電界  $\mathbf{E}$  と磁界  $\mathbf{B}$  が印加された場合の荷電粒子の運動は、質量を  $m$ 、電荷を  $q$  とすると、4-2-2 で述べたように磁界  $\mathbf{B}$  によりローレンツ力を受けるので運動方程式は、

$$m \frac{d\mathbf{v}}{dt} = q(\mathbf{E} + \mathbf{v} \times \mathbf{B}) \quad (4.8)$$

となる。また、速度  $\mathbf{v}$  は  $(x, y, z)$  座標系において、

$$v_x = \frac{dx}{dt}, \quad v_y = \frac{dy}{dt}, \quad v_z = \frac{dz}{dt} \quad (4.9)$$

と定義される。電界  $\mathbf{E}$  と磁界  $\mathbf{B}$  が時間と空間において一定で直交させ、電界  $\mathbf{E} = (E_x, 0, 0)$ 、磁界  $\mathbf{B} = (0, 0, B_z)$  のように印加した場合式(4.8)は、

$$\frac{dv_x}{dt} = \frac{q}{m} (E_x + v_y B_z) \quad (4.10a)$$

$$\frac{dv_y}{dt} = -\frac{q}{m} v_x B_z \quad (4.10b)$$

$$\frac{dv_z}{dt} = 0 \quad (4-10c)$$

となる。式(4-9)の關係を用いると、

$$\frac{d^2x}{dt^2} = \frac{q}{m} \left( E_x + \frac{dy}{dt} B_z \right) \quad (4-11a)$$

$$\frac{d^2y}{dt^2} = -\frac{q}{m} \left( \frac{dx}{dt} B_z \right) \quad (4-11b)$$

となる。

#### 4-2-4 任意の電界と磁界の角度における軌道計算

磁界を用いたプラズマの生成において、ICPの境界付近やマグネトロンプラズマのように電界と磁界が直交していない場合があり、任意の電界と磁界の角度におけるプラズマパラメータを知ることは重要である。以下に軌道計算を示す。式(4-8)を $(x, y, z)$ 座標系で表すと

$$m \frac{dv_x}{dt} = q(E_x + v_y B_z - v_z B_y) \quad (4-12a)$$

$$m \frac{dv_y}{dt} = q(E_y + v_z B_x - v_x B_z) \quad (4-12b)$$

$$m \frac{dv_z}{dt} = q(E_z + v_x B_y - v_y B_x) \quad (4-12c)$$

と書ける。したがって、

$$v_x^0 = -(E_x + v_y B_z - v_z B_y) \Delta t + v_x \quad (4-13a)$$

$$v_y^0 = -(E_y + v_z B_x - v_x B_z) \Delta t + v_y \quad (4-13b)$$

$$v_z^0 = -(E_z + v_x B_y - v_y B_x) \Delta t + v_z \quad (4-13c)$$

となり、 $x, y, z$ はそれぞれ、

$$x = \frac{1}{2}(v_x + v_x^0) \Delta t + x^0 \quad (4-14a)$$

$$y = \frac{1}{2}(v_y + v_y^0) \Delta t + y^0 \quad (4-14b)$$

$$z = \frac{1}{2}(v_z + v_z^0) \Delta t + z^0 \quad (4-14c)$$

で求まる。しかし、実際の計算では、磁界と電界の変化をより詳細に扱うために  $\Delta t$  を小さくとり、式(4-13)を、

$$v_x^{00} = - \left( E_x + \frac{v_y + v_y^0}{2} \frac{B_z + B_z^0}{2} - \frac{v_z + v_z^0}{2} \frac{B_y + B_y^0}{2} \right) \Delta t + v_x \quad (4-15a)$$

$$v_y^{00} = - \left( E_y + \frac{v_z + v_z^0}{2} \frac{B_x + B_x^0}{2} - \frac{v_x + v_x^0}{2} \frac{B_z + B_z^0}{2} \right) \Delta t + v_y \quad (4-15b)$$

$$v_z^{00} = - \left( E_z + \frac{v_x + v_x^0}{2} \frac{B_y + B_y^0}{2} - \frac{v_y + v_y^0}{2} \frac{B_x + B_x^0}{2} \right) \Delta t + v_z \quad (4-15c)$$

$$x = \frac{1}{2}(v_x + v_x^{00}) \Delta t + x^0 \quad (4-16a)$$

$$y = \frac{1}{2}(v_y + v_y^{00}) \Delta t + y^0 \quad (4-16b)$$

$$z = \frac{1}{2}(v_z + v_z^{00}) \Delta t + z^0 \quad (4-16c)$$

として計算を行う。

#### 4-2-5 カスプ磁場における軌道計算

一般に、低気圧では高密度のプラズマは得られないため、磁界を用いることで電子の平均自由行程を長くして高密度のプラズマを得る方法が開発されている。電界と磁界が直交したフィールドにおいて、電子はサイクロトロン運動を行うので、直流放電においても陰極から出発した電子が陽極にたどり着くまでに、直交している磁界により電界と磁界に対して垂直方向に進む  $E \times B$  ドリフトを行い、ガス分子との衝突回数が増加し、高密度のプラズマが得られる。特に薄膜の大型化が容易で、薄膜の形成速度が速いことが知られているマグネトロンプラズマによるスパッタリングにおいては、図 4-2 に示すように、平板上に磁極を配置することにより、磁力線が図のようになり、これはカスプ磁場と呼ばれ、場所により電界と磁界の角度が異なることから粒子の運動はより複雑になる。

いま、図 4-2 に示すように磁極を配置し  $yz$  平面から  $b$  [cm] 離れた点  $A$  の場所で  $bZ$  [G] ( $[10^{-4}T]$ ) となるようにした場合、

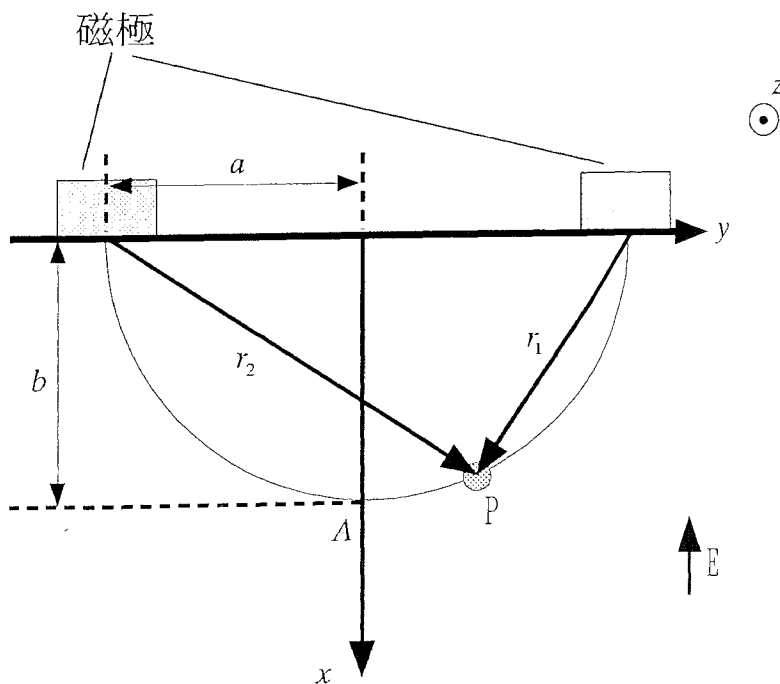


図 4-2 磁極のカスプ配置による磁界

$$r_0 = \sqrt{a^2 + b^2} \quad (4-17)$$

$$A_c = \frac{bZ \cdot r_0^2}{2a} \quad (4-18)$$

とすると、 $r_1, r_2$ はそれぞれ、

$$r_1 = \sqrt{x^2 + (y-a)^2} \quad (4-19a)$$

$$r_2 = \sqrt{x^2 + (y+a)^2} \quad (4-19b)$$

であり、したがって磁界の大きさは、

$$B_x = A \cdot x \left( \frac{1}{r_1^2} - \frac{1}{r_2^2} \right) \quad (4-20a)$$

$$B_y = A \left( \frac{y-a}{r_1^2} - \frac{y+a}{r_2^2} \right) \quad (4-20b)$$

で求められる。



### 4-3 シミュレーション結果および考察

#### 4-3-1 電界と磁界が直交するプラズマ・パラメータ

最近の半導体技術の進歩に伴って、プラズマ・プロセスに対する超微細加工技術の要求が強まり、そのひとつに低気圧でのプラズマの生成法が求められている。低気圧ではガス分子密度が低いことから荷電粒子との衝突が少なくなり制御性はよくなるが、衝突回数が減ることにより高密度プラズマが得られない。この問題を解決するために、電界と磁界を組み合わせたプラズマの生成法が開発されている。

磁界が印加されることで荷電粒子の動きが複雑になり、より詳細なプラズマ構造の理解が必要となっている。電界と磁界が直交した  $E \times B$  フィールドにおいて、高磁界下では電界の影響が妨げられ、電子へエネルギーが供給されなくなり、初期エネルギーの状態から変わらないフローズンインと呼ばれる状態が指摘され、Nakamura ら<sup>[13]</sup>は Ar ガスと  $\text{CH}_4$  ガスについて調べ、初期エネルギーのまま存在することはないことが確かめられている。

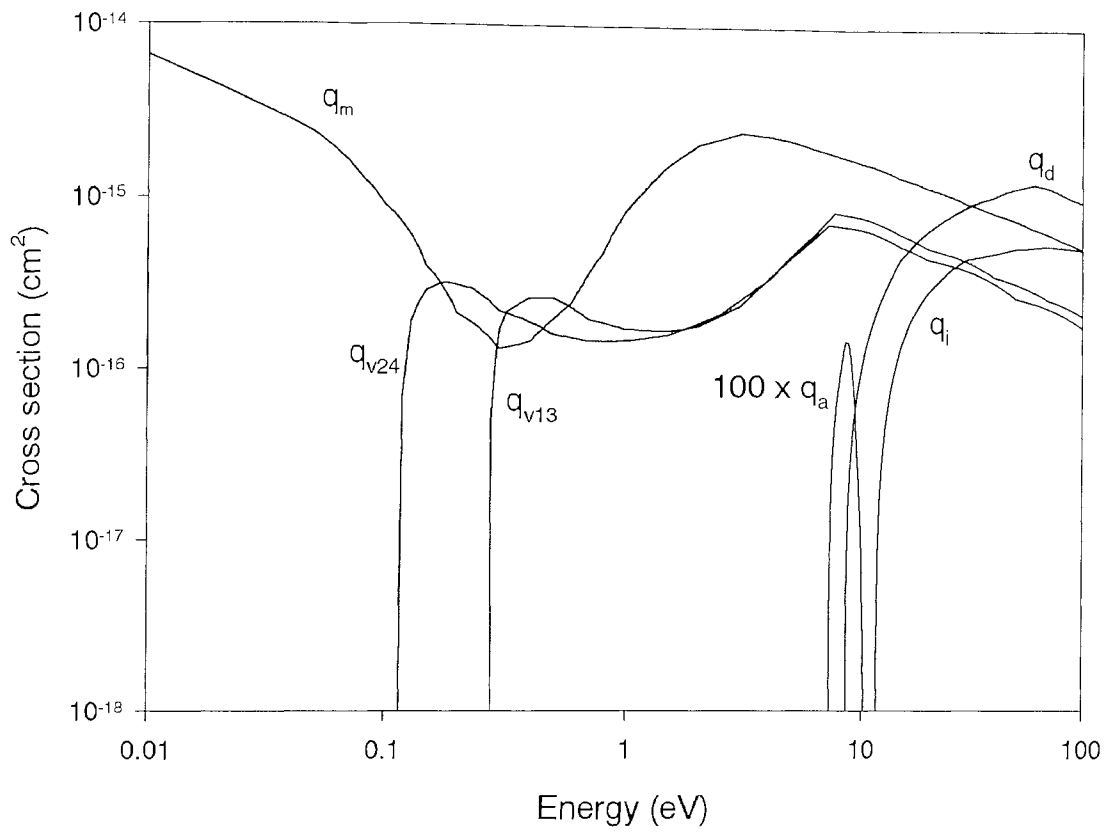
本論文では、 $\text{SiH}_4$  ガス、 $\text{Cl}_2$  ガスおよび Ar ガスについて電界と磁界の角度を変えてシミュレーションを行った。図 4-3(a),(b),(c)は計算に用いたガスの衝突断面積である。図 4-4 は、 $E \times B$  フィールドにおいて電界と磁界の与え方による平均エネルギーの違いをプロットしたグラフである。このグラフが示すように、磁界を大きくしていくとどのような電界を与えても最終的にある一定の値になっているのが示されている。これは、 $E \times B$  フィールドにおいては、常に磁界は電界に対して直角に働くので、電界によって加速される電子は電界方向には加速されず、磁界によって曲げられるため磁界が強くなればさらにエネルギーを得られなくなるために、最終的にどのような電界を与えても同じ値になると考えられる。

図 4-5 は  $\text{SiH}_4$  ガスにおいて電界 0、磁界 0、ガス熱運動なし、弾性衝突の損失エネルギーなしとしてシミュレーションを行った結果である。このグラフでは、最終的な平均エネルギーが初期のエネルギーに依存していることを示している。これは、図 4-3 に示す断面積のデータと関係しており、非弾性

衝突が始まるエネルギーは約 0.1eV 付近からであり、それ以下のエネルギーでは弾性衝突しか起こらないので、内部エネルギーの変化はない。したがって、初期エネルギーを 0.1eV 以下で与えた計算では、エネルギーの変化がないことが示されている。また、0.1eV 以上で与えた計算では、非弾性衝突によりエネルギー損失が起こるが、電界 0 のためにエネルギーを得ることがなく平均エネルギーは下がっていくだけであり、初期エネルギーと各衝突によるエネルギー損失と衝突時のエネルギーの組み合わせによって最終的な平均エネルギーの違いが起こると考えられる。

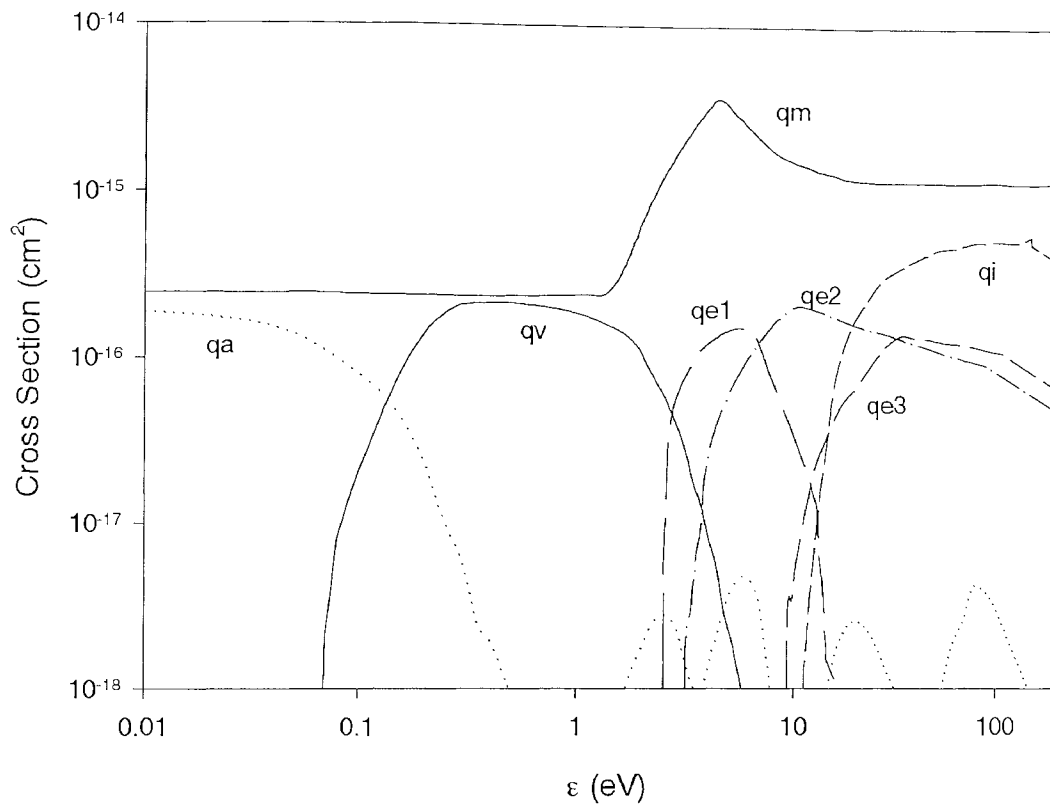
このことは、 $E \times B$  フィールドにおいても、高磁界になれば、電界の影響が打ち消され、電界によりエネルギーが得られなくなり、同じような現象が起こるのではないかと考えられる。 $E \times B$  フィールドにおける、初期エネルギーの違いによる平均エネルギーの緩和過程を示した図 4-6 において、十分時間が経過すれば初期エネルギーに関係なく平均エネルギーは同じ値となる。

プラズマ・プロセスにおけるエッチングプロセスでは磁界を印加した方法が用いられており、加工する基板表面の反応を知る上で  $E \times B$  フィールドにおける平均エネルギーが、重要なプラズマ・パラメータの一つとなる。また、プラズマの生成および制御には、磁界の与え方も電界と直交して与えるだけでなくカスプ磁場で行われる場合があり、異なる電界と磁界の角度においてプラズマ・パラメータを知ることは重要である。



$q_m$  : 弾性衝突、 $q_a$  : 付着衝突、 $q_i$  : 電離衝突、  
 $q_d$  : 解離衝突、 $q_{v12}$ ,  $q_{v24}$  : 振動励起衝突  
 (a)  $\text{SiH}_4$  ガス

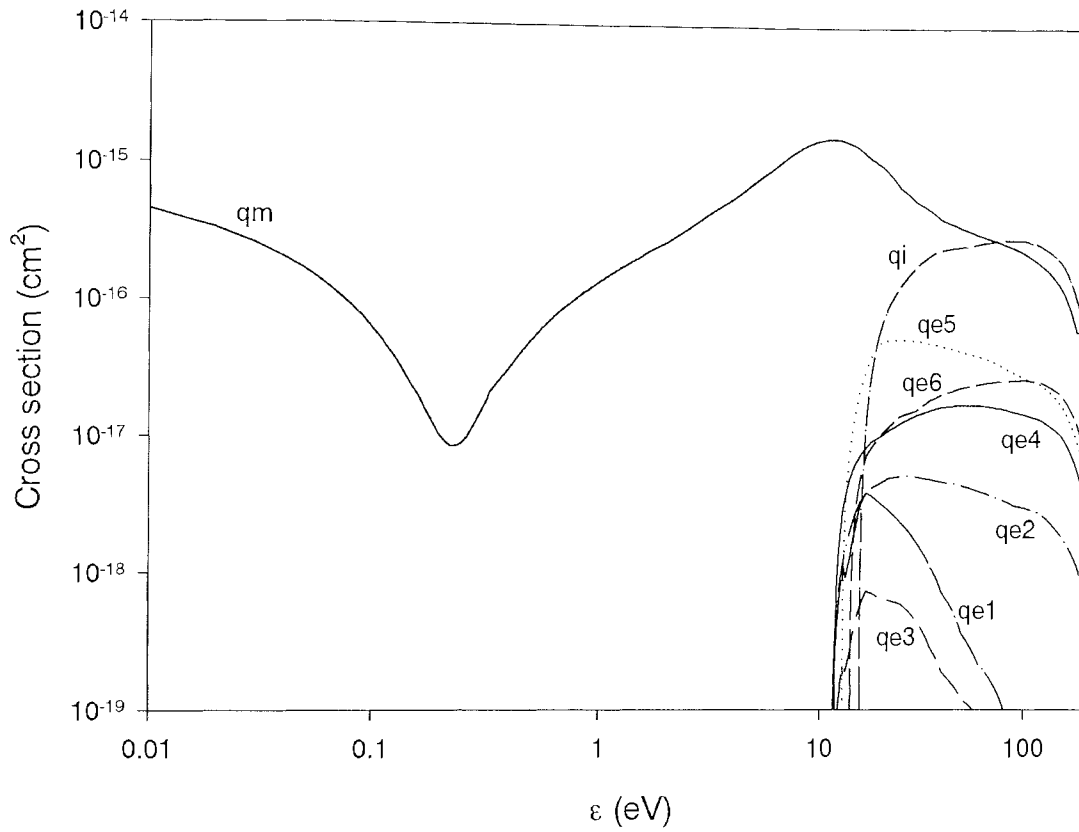
図 4-3 衝突断面積



$q_m$  : 弾性衝突、 $q_a$  : 付着衝突、 $q_i$  : 電離衝突、  
 $q_{e1}, q_{e2}, q_{e3}$  : 励起衝突、 $q_v$  : 振動励起衝突

(b)  $Cl_2$  ガス

図 4-3 衝突断面積



$q_m$  : 弾性衝突、 $q_i$  : 電離衝突、 $q_{e1}$ ,  $q_{e2}$ ,  $q_{e3}$ ,  $q_{e4}$ ,  $q_{e5}$ ,  $q_{e6}$  : 励起衝突

(c) Ar ガス

図 4-3 衝突断面積

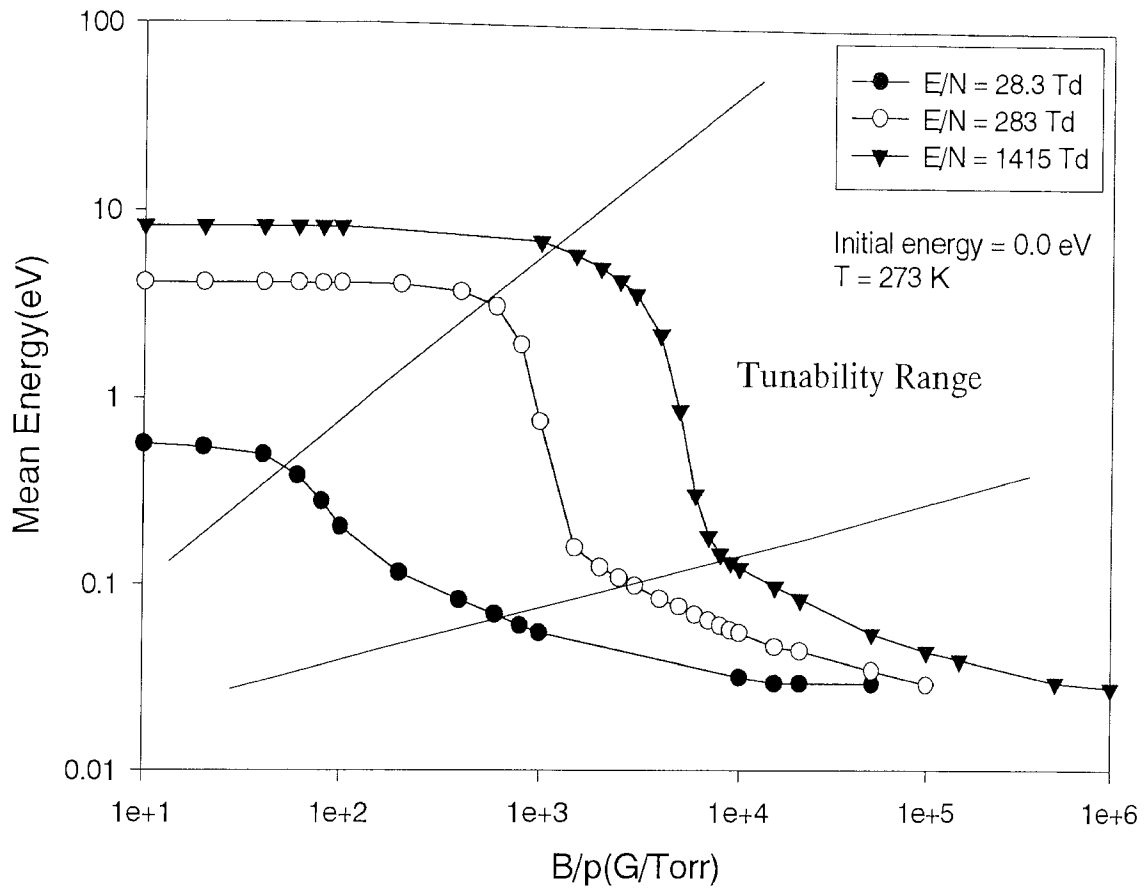


図 4-4  $E \times B$  フィールドにおける平均エネルギー ( $\text{SiH}_4$ )

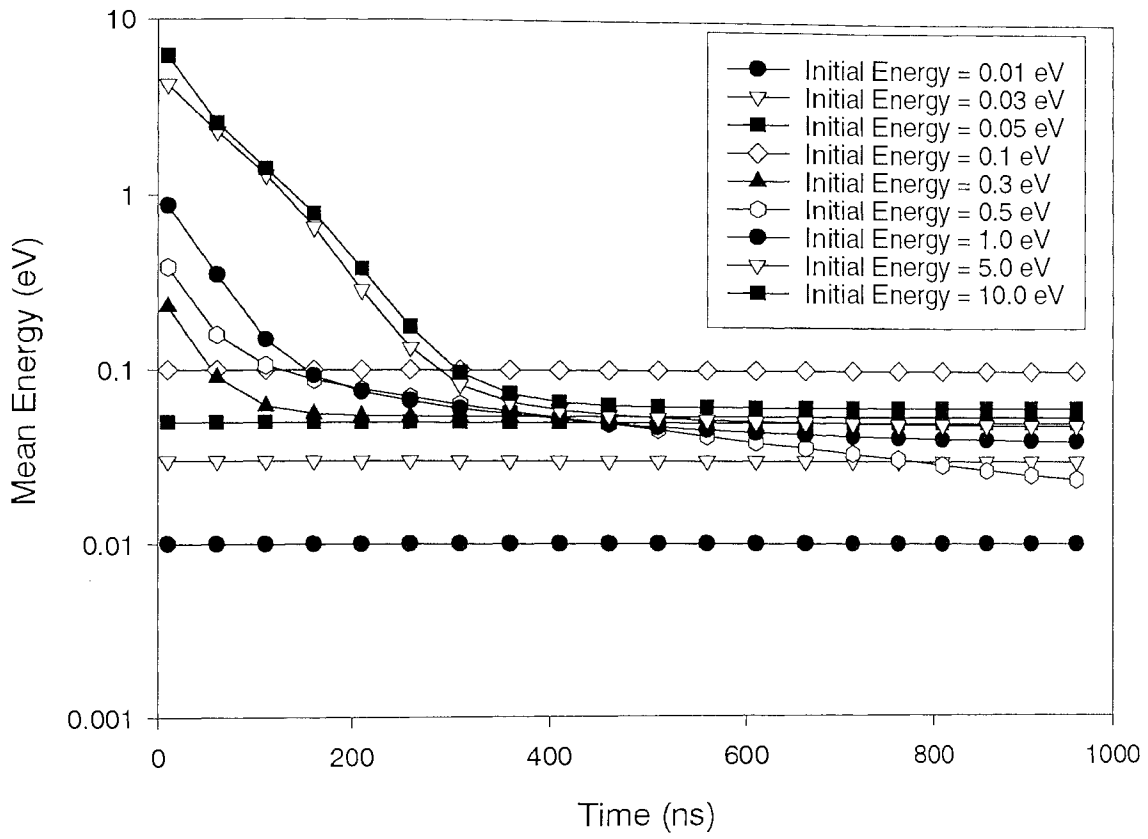
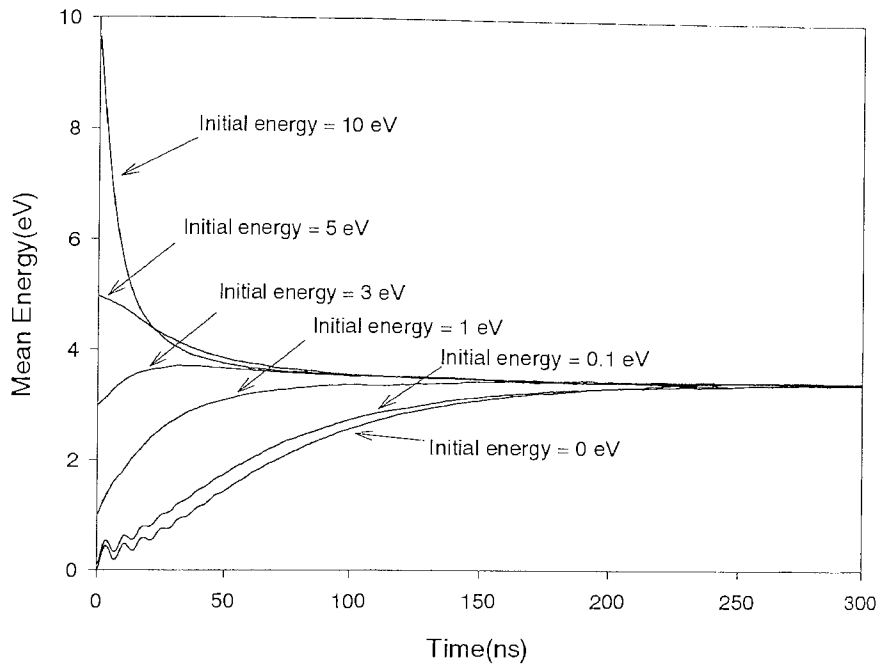
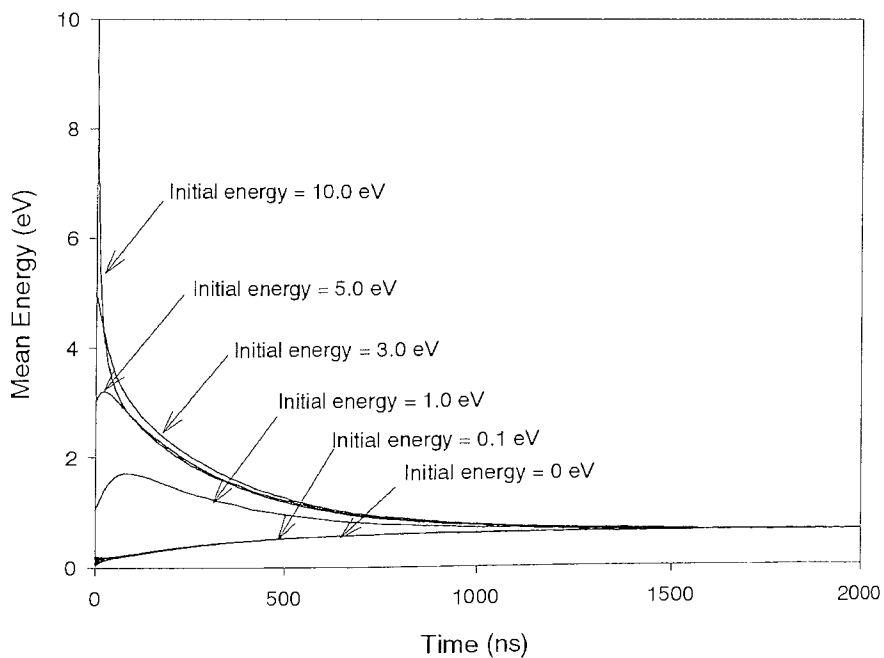


図 4-5 電界 0、磁界 0 における平均エネルギーの緩和過程 (SiH<sub>4</sub>)



(a) Mean electron energy relaxation in a crossed 500 G/Torr, 283 Td field.



(b) Mean electron energy relaxation in a crossed 1000 G/Torr, 283 Td field.

図 4-6  $E \times B$  フィールドにおける異なる初期エネルギーと平均エネルギーの緩和 ( $\text{SiH}_4$ )



#### 4-3-2 SiH<sub>4</sub> ガス中の電界と磁界の成す角の違いによるプラズマ・パラメータ

図 4-7 は電界と磁界の成す角度を変えて与えたときの平均エネルギーを示したグラフである。0 度のときは、電界と磁界が同じ方向となるので、磁界は電界によって電子の加速を妨げないので、このとき平均エネルギーが最大になり、角度が大きくなるにしたがって電子は磁界によって曲げられ、エネルギーを得にくくなり角度とともに平均エネルギーが下がっていく。

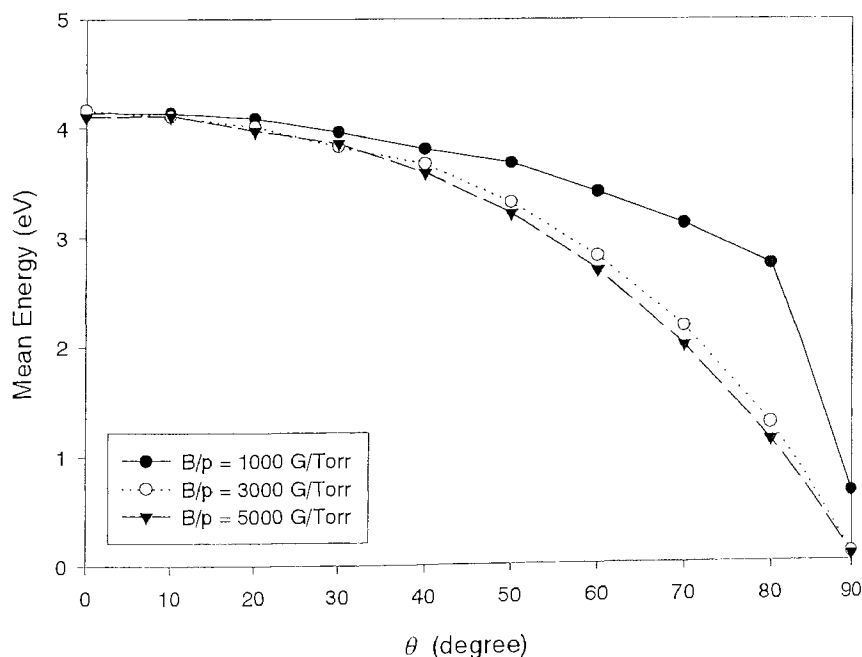
図 4-8 は各方向の平均エネルギーを示している。基本的には同じ傾向を示しているが、 $x$  軸方向に磁界が印加されているので  $x$  軸方向の平均エネルギーが  $y$  軸方向と  $z$  軸方向よりも大きい。この各方向の平均エネルギーは、衝突後の広がる力を意味している。したがって、各方向のエネルギーが同じような値になっているので等方的に散乱していると考えられる。また、図 4-8(b) および (c) にあるように電界が大きい場合は、角度を大きくしても平均エネルギーが小さくならず電界と磁界との角度が 90 度でも磁界によって曲げられるよりも電界によって加速される影響の方が大きいと考えられる。

図 4-9 は異なる電界と磁界の角度による各方向の平均速度を示しており、電子スオームがどの方向にドリフトするかを知ることができる。電界と磁界の角度が 0 度のときは、電界のみと考えられ、電子スオームは電界方向と逆方向にドリフトすると考えられる。このシミュレーションでは、電界は  $x$  軸のプラスの方向に与えられているために、電子スオームは  $x$  軸のマイナス方向にドリフトすると考えられる。電界と磁界の角度が 0 度のときに  $y$  軸と  $z$  軸の平均速度は 0 で等方に拡散し、 $x$  軸のマイナス方向にドリフトしていることが示されている。したがって、電界と磁界の角度が大きくなってくると磁界によって電子が  $y$  軸方向に曲げられ  $x$  軸方向へのドリフトが妨げられると考えられる。すなわち、 $x$  軸方向の平均速度は、角度が大きくなるにつれて値は小さくなり、逆に  $y$  軸方向の平均速度は大きくなっている。 $z$  軸方向のドリフトに関しては、電界と磁界の角度が小さいときには、磁界は  $x$  軸方向に倒れ方が大きいので電子が  $y$  軸方向にドリフトしたときに流れる電流と  $x$  軸方向の磁界とによって  $z$  軸方向へのドリフトが生じると考えられる。ま

た、角度が大きくなってくると  $x$  軸方向よりも  $z$  軸方向の磁界成分が大きくなるので  $x$  軸と  $z$  軸で生じる  $E \times B$  フィールドによって  $y$  軸方向にドリフトするため、 $z$  軸方向のドリフト成分が小さくなり 50 度付近で下に凸となる 2 次曲線的になっている。

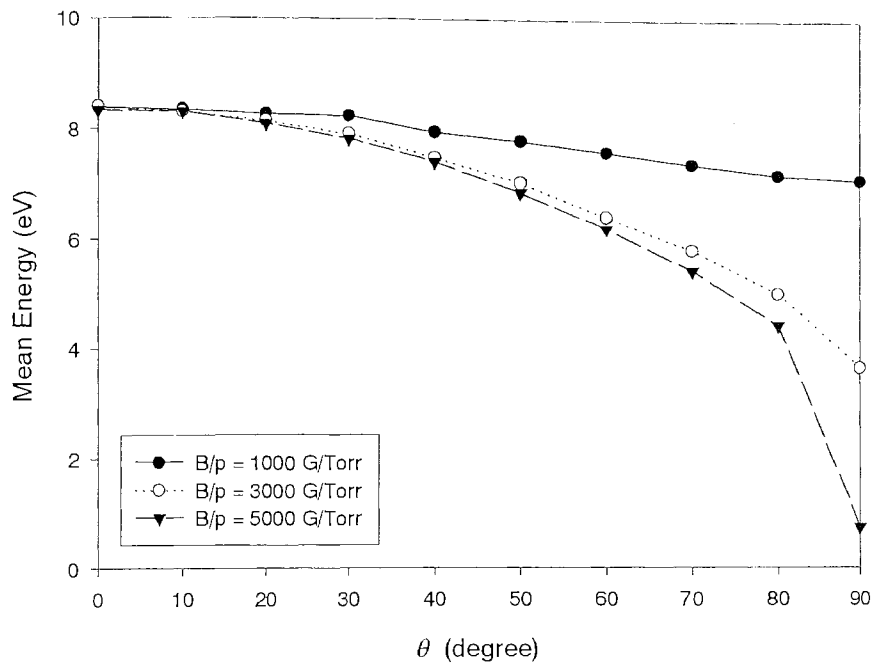
図 4-10 は平均衝突周波数を求めたものである。図 4-10 (a),(c),(d)の衝突周波数については電界と磁界の角度が大きくなるにつれて電界の影響が妨げられ、衝突回数が減少していくことが示されている。一方、図 4-10 (b)の付着衝突は 70 度付近に極大値が存在している。70 度付近は図 4-7 (c)から見るとエネルギーが 8eV 付近を示しており、このエネルギー値の衝突断面積 (図 4-3(a)) から付着衝突が起きる確率が高くなっていることがわかる。したがって、図 4-10 (b)に示すようには 70 度付近に極大値が存在することとなる。

図 4-11 は LPWS 法によりエネルギー分布を求めたものであり、密度が 1 となるように正規化して示している。低磁界 (図 4-11(a)) においては角度が異なる場合においてもエネルギー分布に変化はないが、高磁界 (図 4-11 (c),(d)) になるにつれて、60 度でも磁界が効くことがわかる。

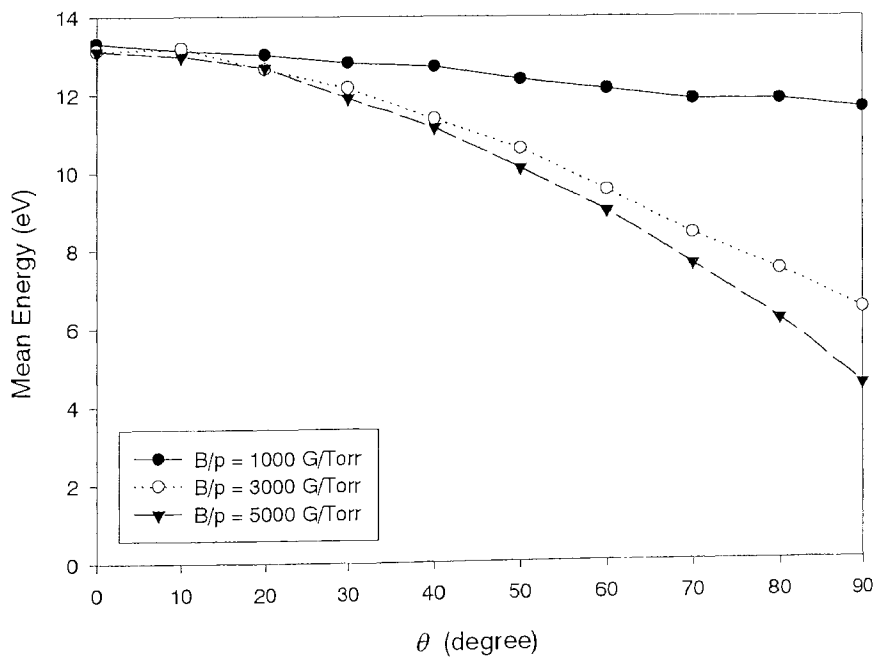


(a)  $E/N=283$ Td,  $p=0.1$ Torr

図 4-7 電界と磁界の異なる角度における平均エネルギー

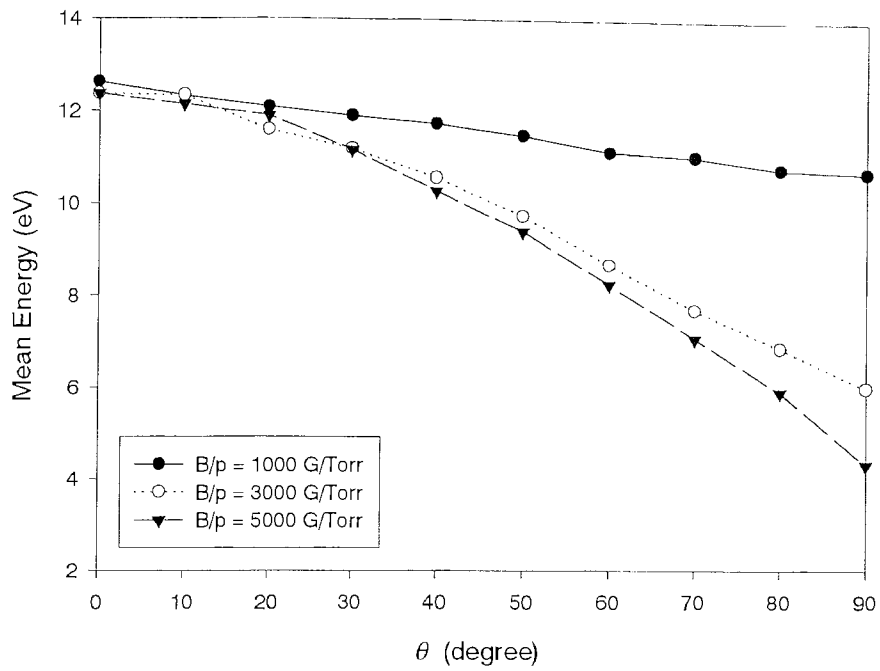


(b)  $E/N=1415\text{Td}$ ,  $p=0.1\text{Torr}$

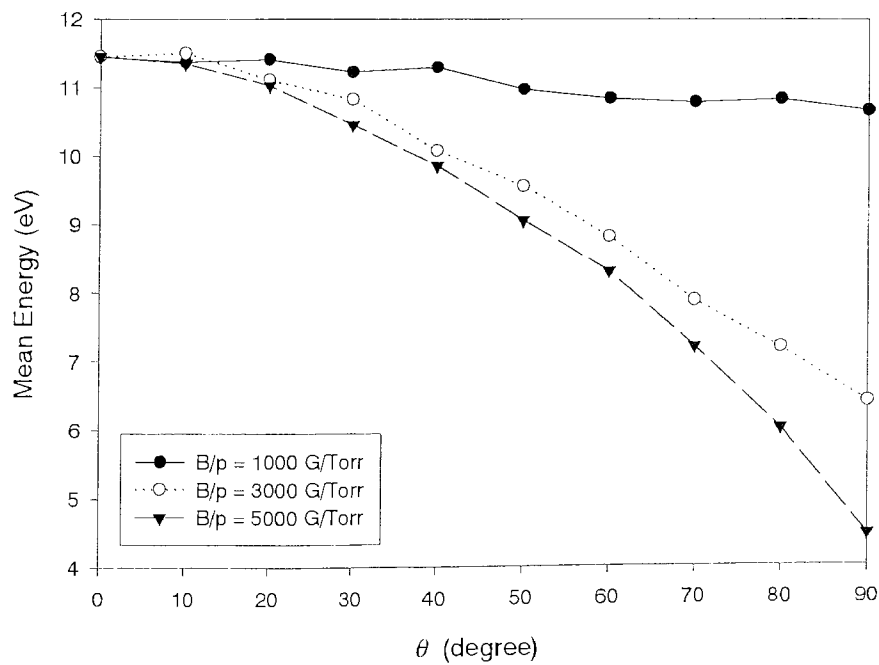


(c)  $E/N=2830\text{Td}$ ,  $p=0.1\text{Torr}$

図 4-7 電界と磁界の異なる角度における平均エネルギー

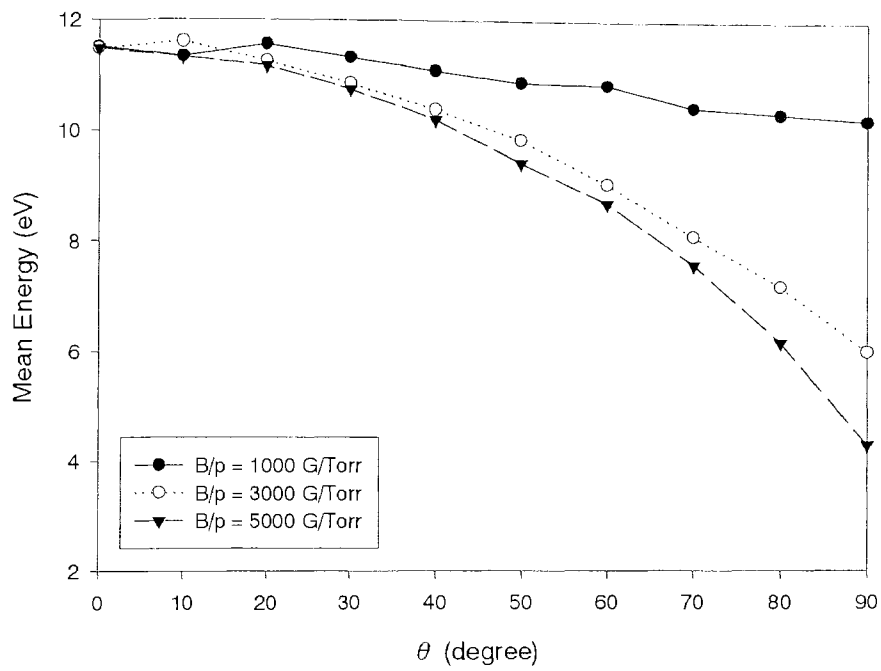


(a) x 成分,  $E/N=2830\text{Td}$ ,  $p=0.1\text{Torr}$



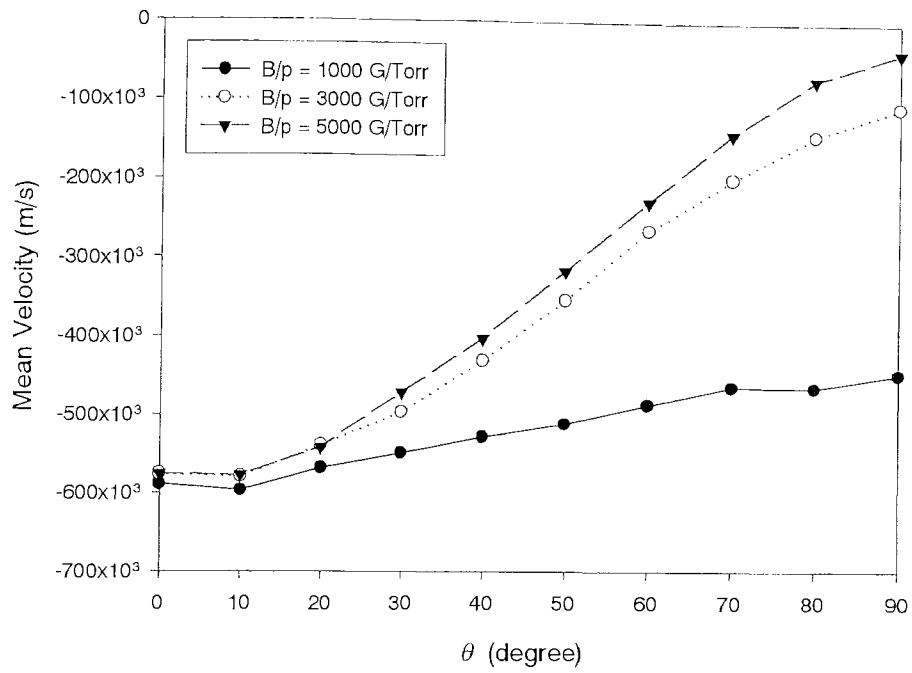
(b) y 成分,  $E/N=2830\text{Td}$ ,  $p=0.1\text{Torr}$

図 4-8 電界と磁界の異なる角度における各方向のエネルギー

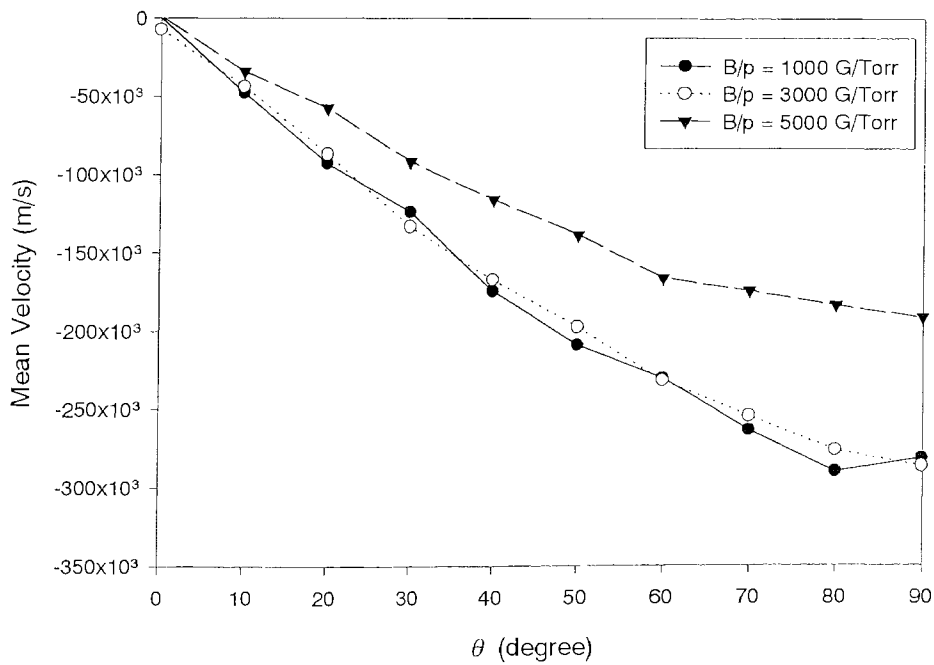


(c)  $z$  成分,  $E/N=2830\text{Td}$ ,  $p=0.1\text{Torr}$

図 4-8 電界と磁界の異なる角度における各方向のエネルギー

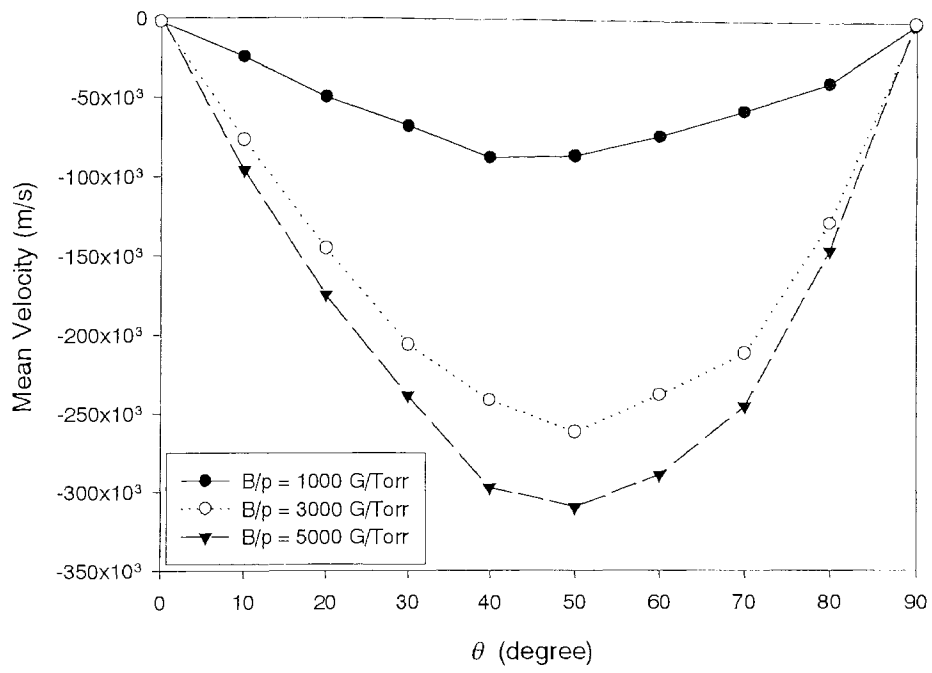


(a)  $x$  成分,  $E/N=2830$ Td,  $p=0.1$ Torr



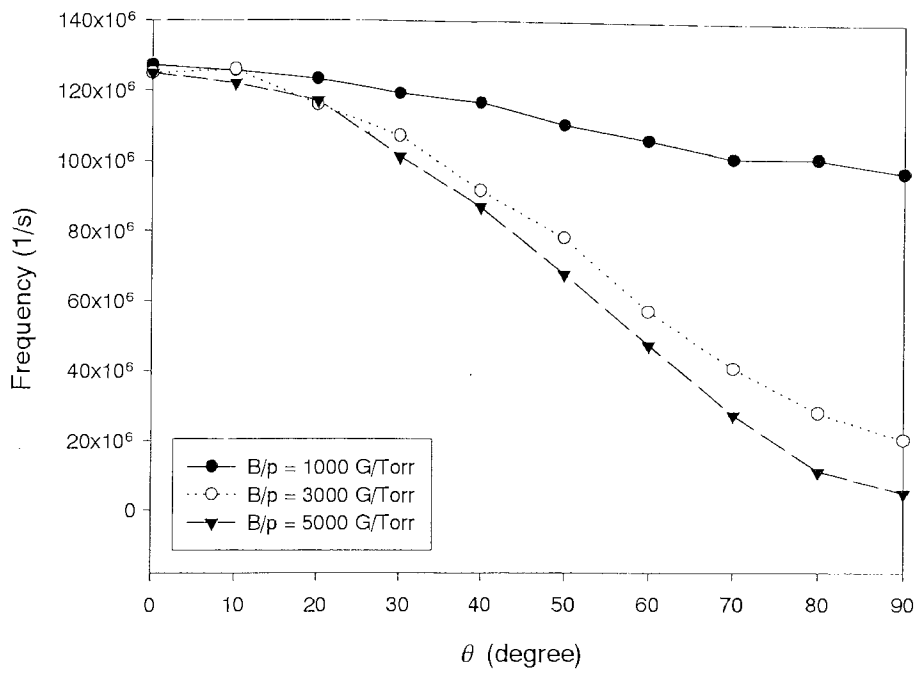
(b)  $y$  成分,  $E/N=2830$ Td,  $p=0.1$ Torr

図 4-9 電界と磁界の異なる角度における各方向の平均速度

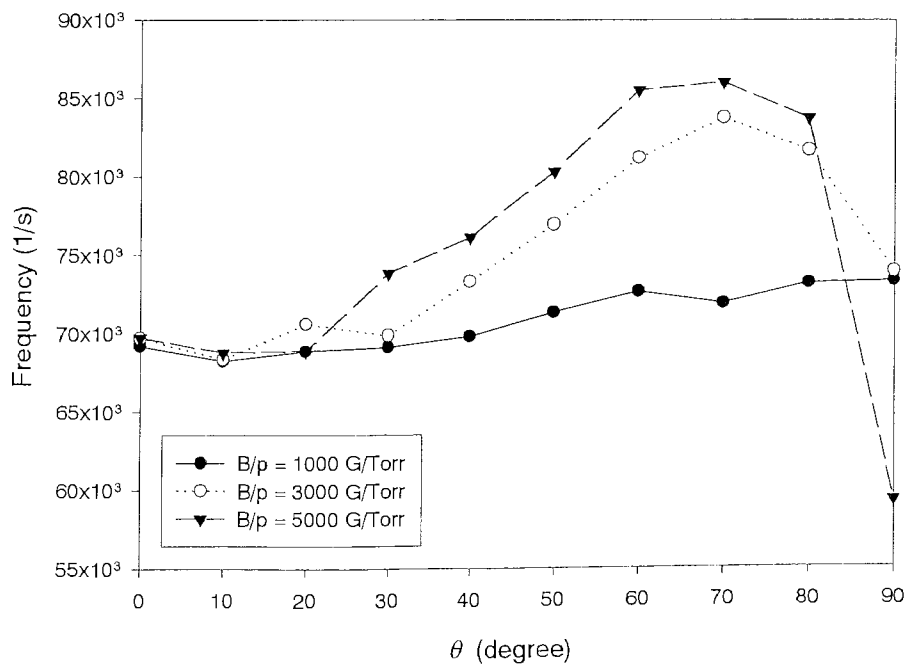


(c) z 成分,  $E/N=2830$ Td,  $p=0.1$ Torr

図 4-9 電界と磁界の異なる角度における各方向の平均速度



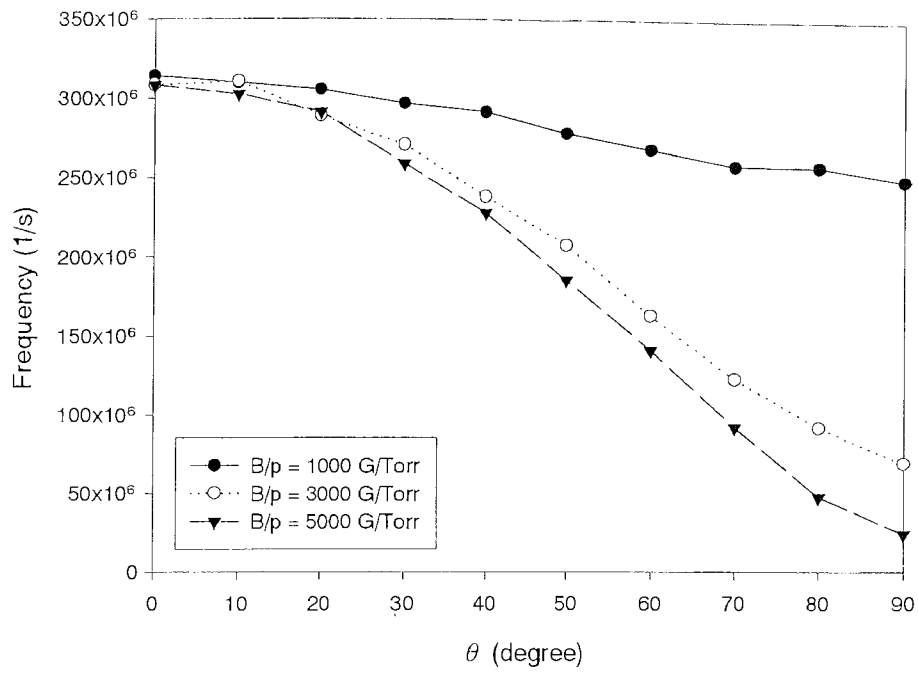
(a) 電離衝突周波数



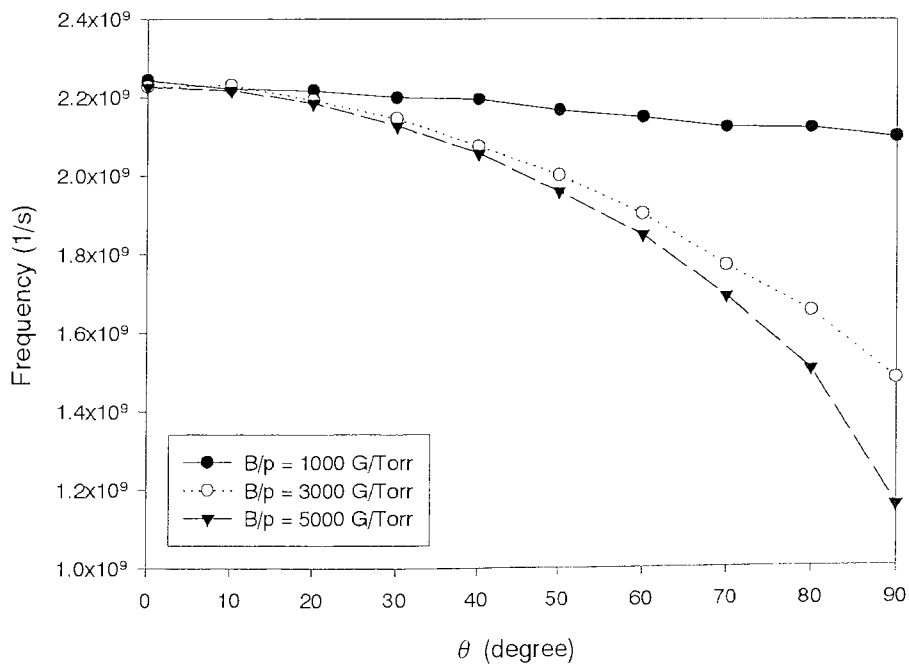
(b) 付着衝突周波数

図 4-10 電界と磁界の異なる角度におけ衝突周波数



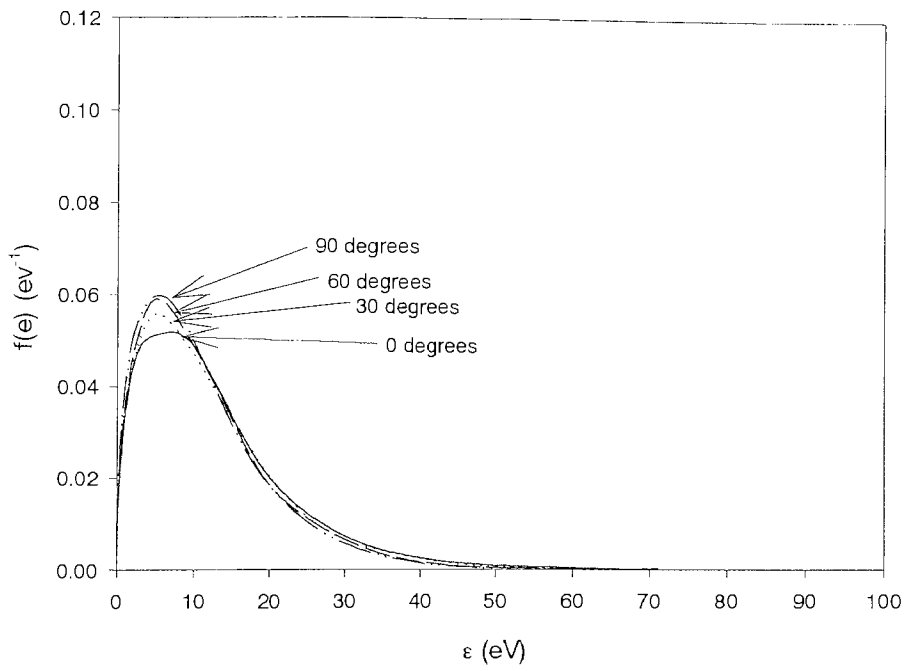


(c) 解離衝突周波数

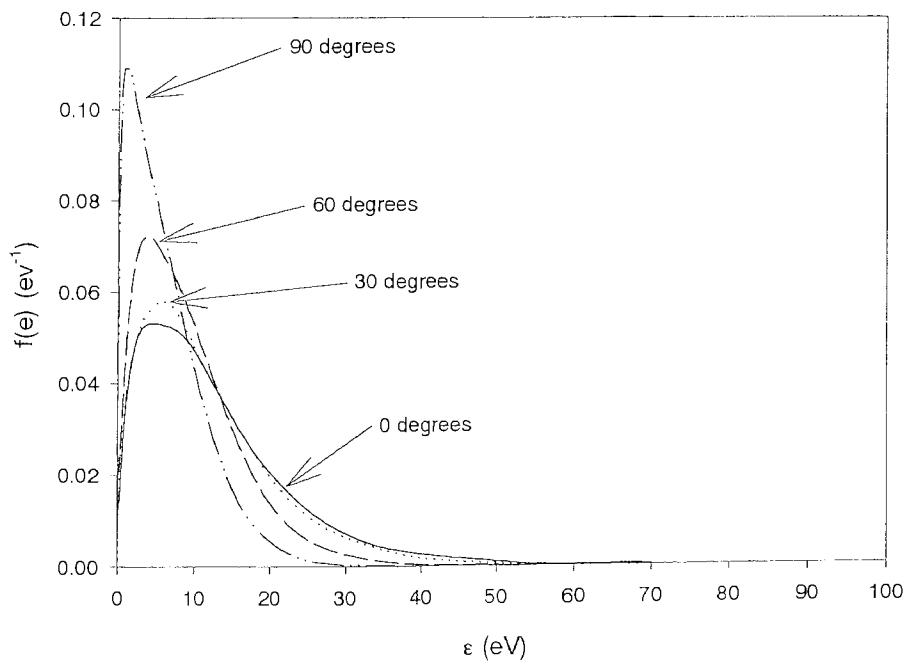


(d) 全衝突周波数

図 4-10 電界と磁界の異なる角度におけ衝突周波数

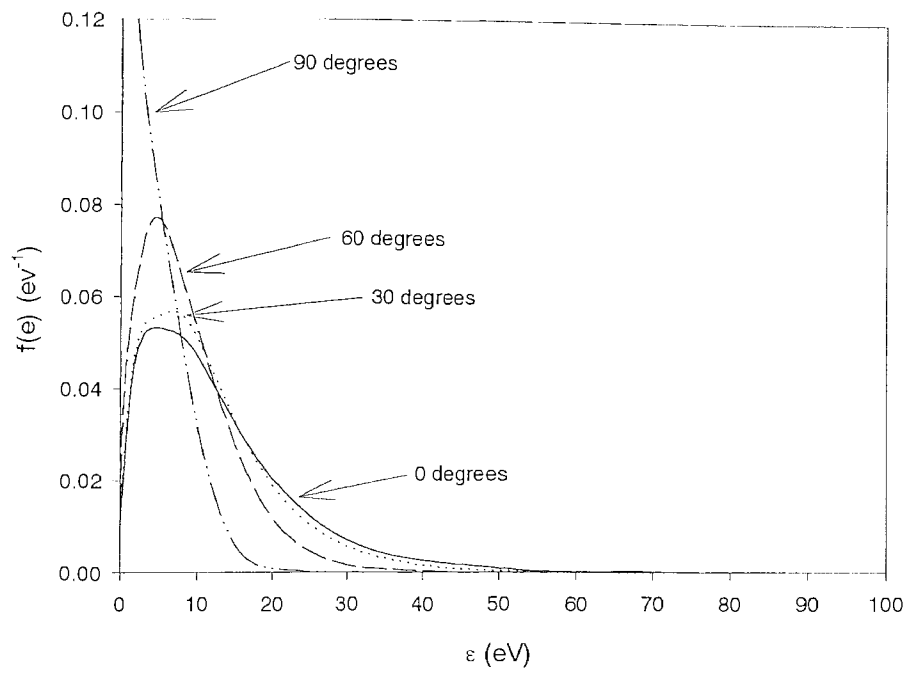


(a)  $E/N=2830\text{Td}$ ,  $B/p=1000\text{G/Torr}$



(b)  $E/N=2830\text{Td}$ ,  $B/p=3000\text{G/Torr}$

図 4-11 電界と磁界の異なる角度におけるエネルギー分布



(c)  $E/N=2830\text{Td}$ ,  $B/p=5000\text{G/Torr}$

図 4-11 電界と磁界の異なる角度におけるエネルギー分布

#### 4-3-3 Cl<sub>2</sub> ガス中の電界と磁界の成す角の違いによるプラズマ・パラメータ

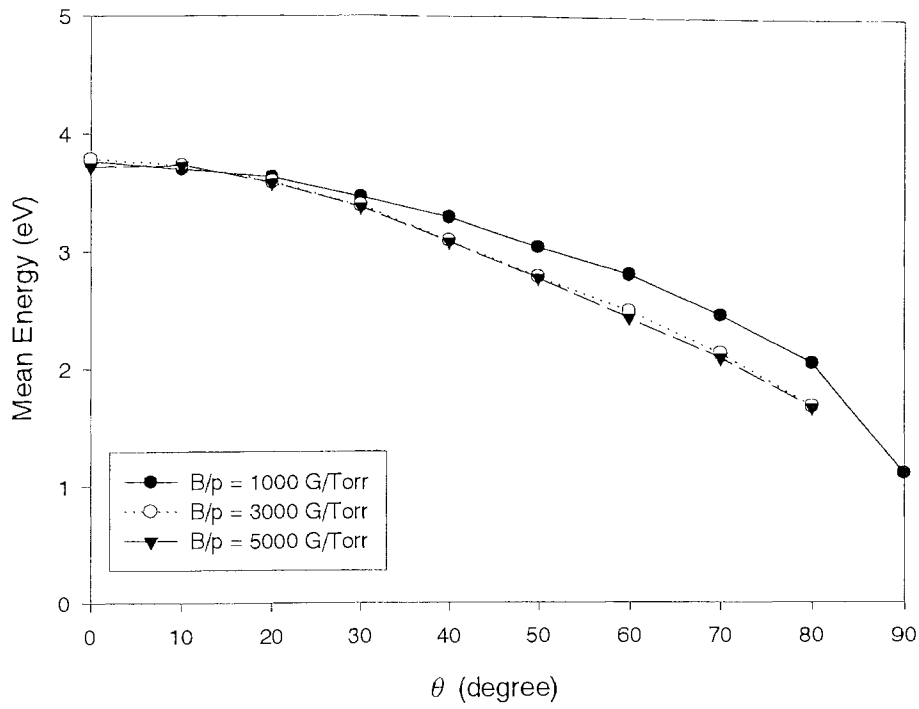
図 4-12 は電界と磁界の成す角度を変えて与えたときの平均エネルギーを示したグラフである。SiH<sub>4</sub> ガス同様に、0 度のとき、平均エネルギーが最大になり、角度が大きくなるにしたがって電子は磁界によって曲げられ、エネルギーを得にくくなり角度とともに平均エネルギーが下がっていく。

図 4-13 は各方向の平均エネルギーを示している。SiH<sub>4</sub> ガス同様に、 $x$  軸方向に磁界が印加されているので  $x$  軸方向の平均エネルギーが  $y$  軸方向と  $z$  軸方向よりも大きくなっている。

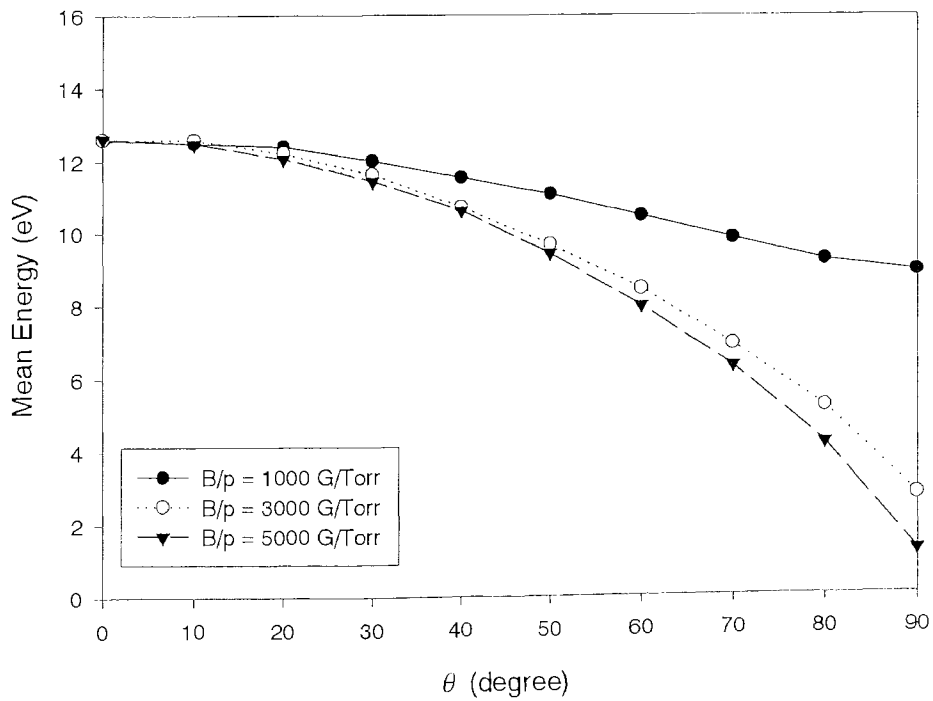
図 4-14 は異なる電界と磁界の角度による各方向の平均速度を示している。SiH<sub>4</sub> ガス同様に  $z$  軸方向の速度成分は 50 度付近に極大値が存在している。

図 4-15 は平均衝突周波数を求めたものである。図 4-15 (a),(b),(c),(e),(g) の衝突周波数については電界と磁界の角度が大きくなるにつれて電界の影響が妨げられ、衝突回数が減少していくことが示されている。一方、図 4-15 (d) の励起衝突( $q_{e2}$ )は 80 度付近に極大値が存在している。80 度付近は図 4-12 (c) から見るとエネルギーが 7~10eV 付近を示しており、このエネルギー値の衝突断面積 (図 4-3(b)) から、電離衝突が起こる前で、かつ励起衝突( $q_{e2}$ )が起こる確率が高くなっていることがわかる。したがって、図 4-15 (d) は 80 度付近に極大値が存在することとなる。また、図 4-15 (f) の振動励起衝突は 90 度付近に極大値が存在している。最も顕著な 5000G/Torr について見ると、90 度付近は図 4-12 (c) から見るとエネルギーが 4eV 付近を示しており、このエネルギー値の衝突断面積 (図 4-3(b)) から、振動励起衝突が起こる確率が高くなっていることがわかる。したがって、図 4-15 (f) は高磁界になるにつれて振動励起衝突の起きる確率が高まることとなる。

図 4-11 はエネルギー分布を求めたものであり、密度が 1 となるように正規化して示している。SiH<sub>4</sub> ガスに比べ全体的にエネルギー値が高いことがわかる。

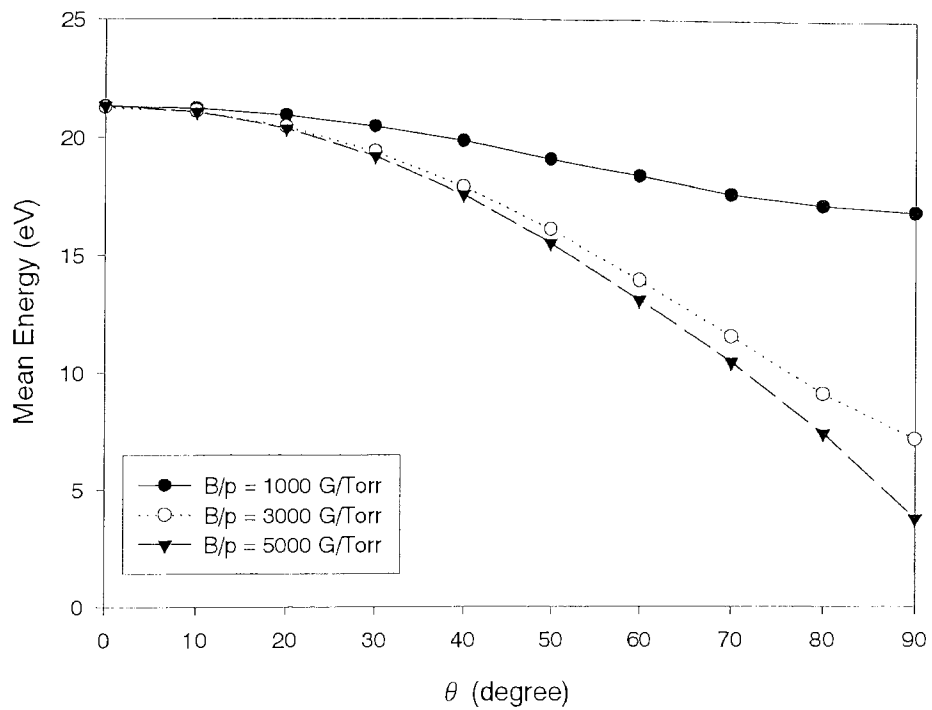


(a)  $E/N=283\text{Td}$ ,  $p=0.1\text{Torr}$



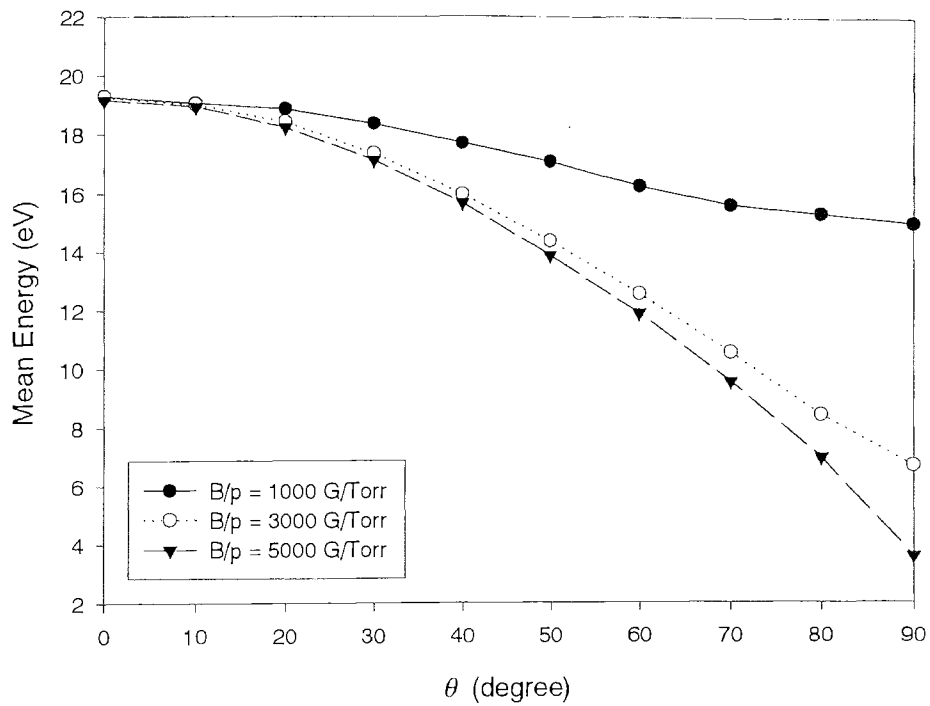
(b)  $E/N=1415\text{Td}$ ,  $p=0.1\text{Torr}$

図 4-12 電界と磁界の異なる角度における平均エネルギー

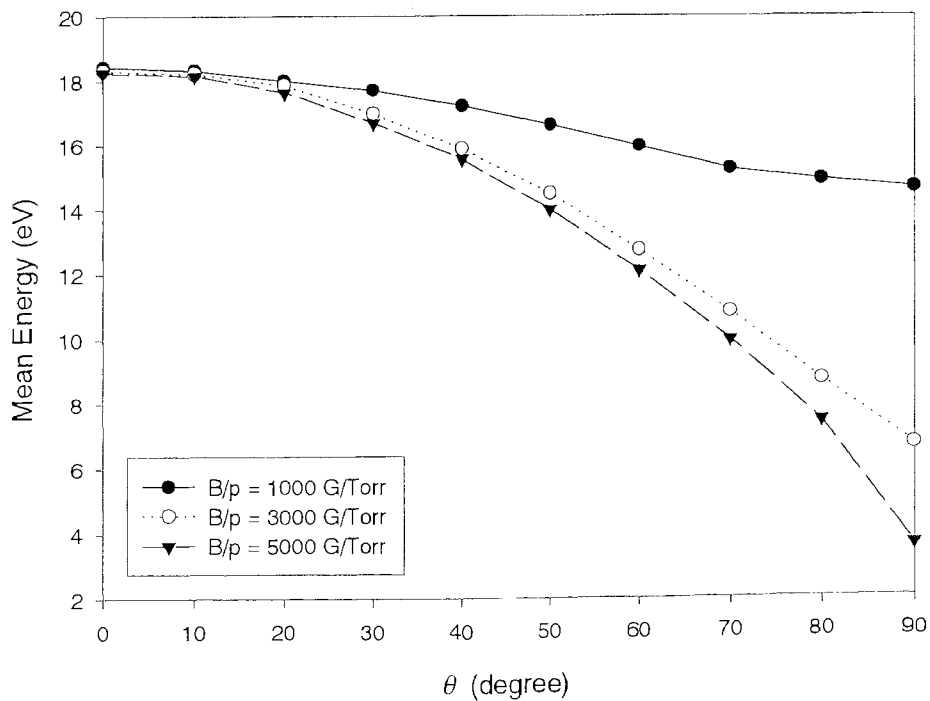


(c)  $E/N=2830\text{Td}$ ,  $p=0.1\text{Torr}$

図 4-12 電界と磁界の異なる角度における平均エネルギー

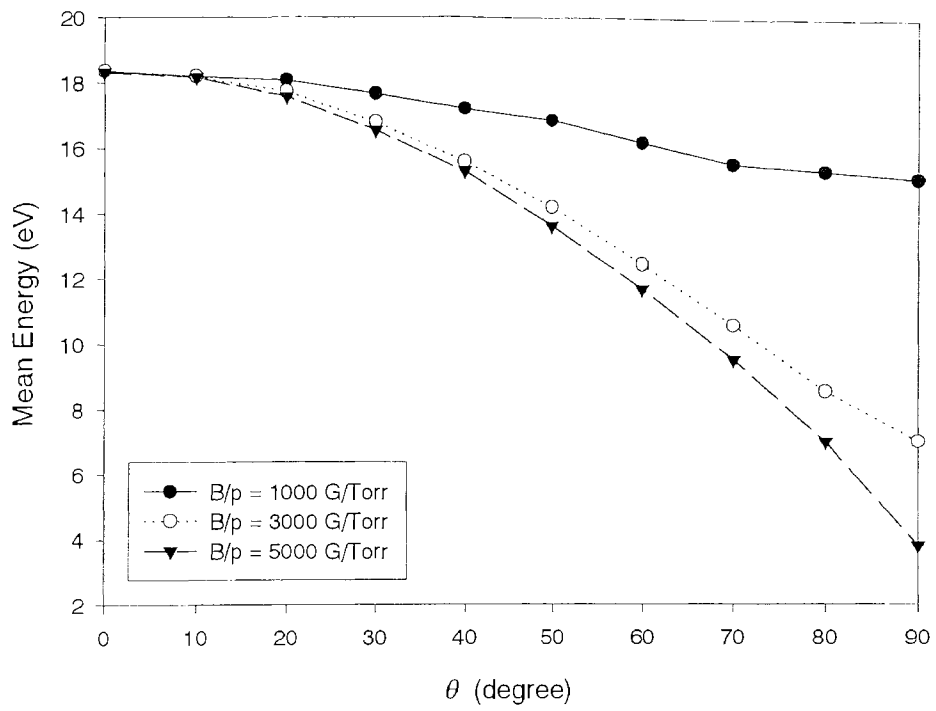


(a) x 成分,  $E/N=2830$ Td,  $p=0.1$ Torr



(b) y 成分,  $E/N=2830$ Td,  $p=0.1$ Torr

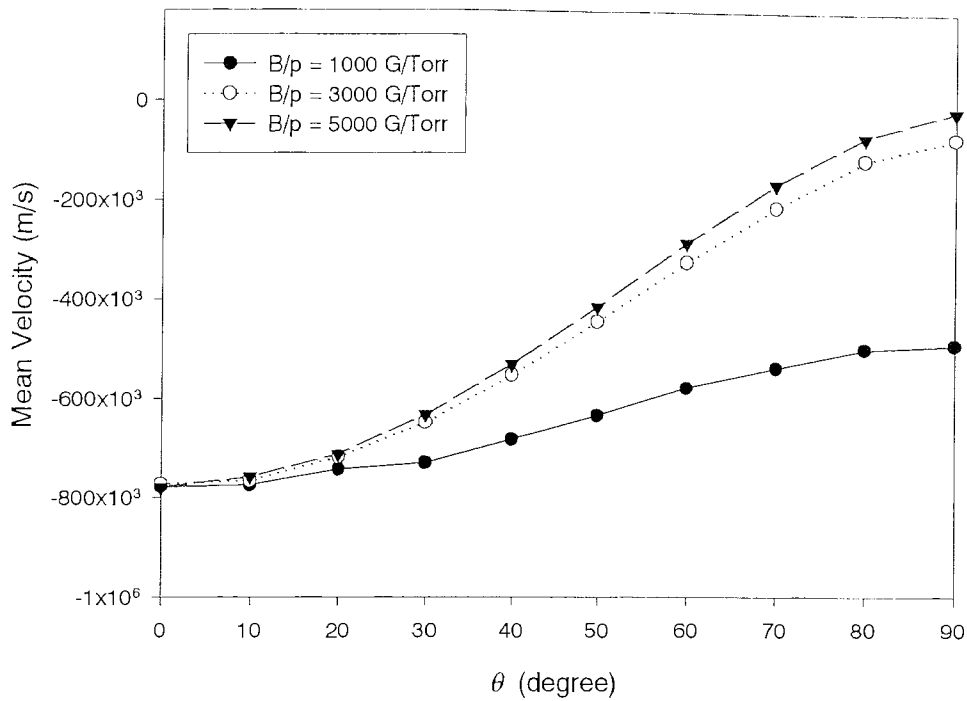
図 4-13 電界と磁界の異なる角度における各方向のエネルギー



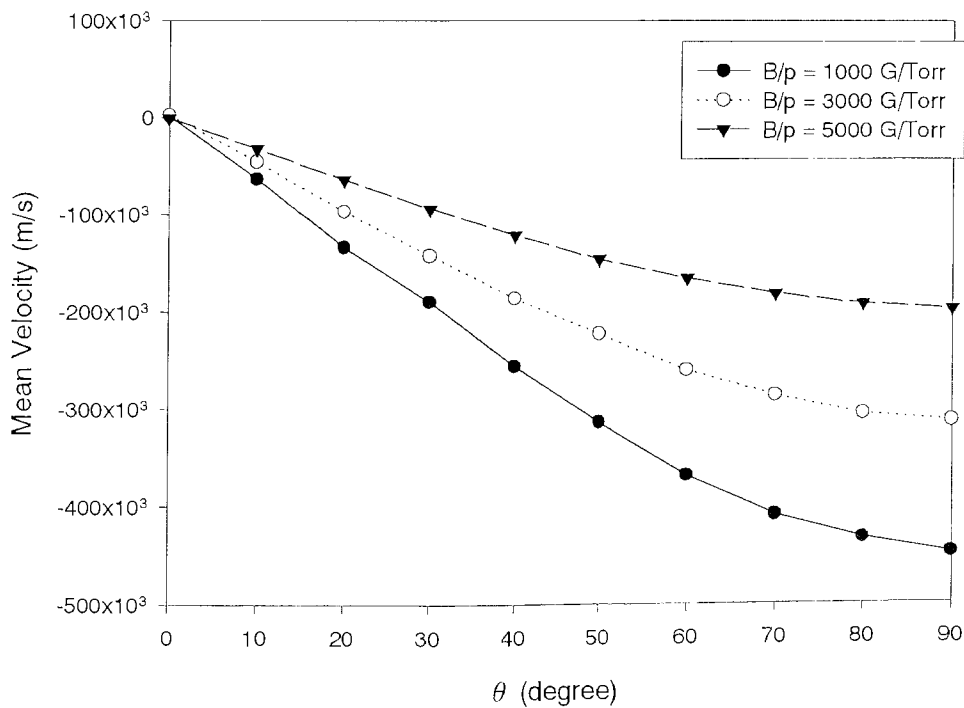
(c)  $z$  成分,  $E/N=2830$ Td,  $p=0.1$ Torr

図 4-13 電界と磁界の異なる角度における各方向のエネルギー



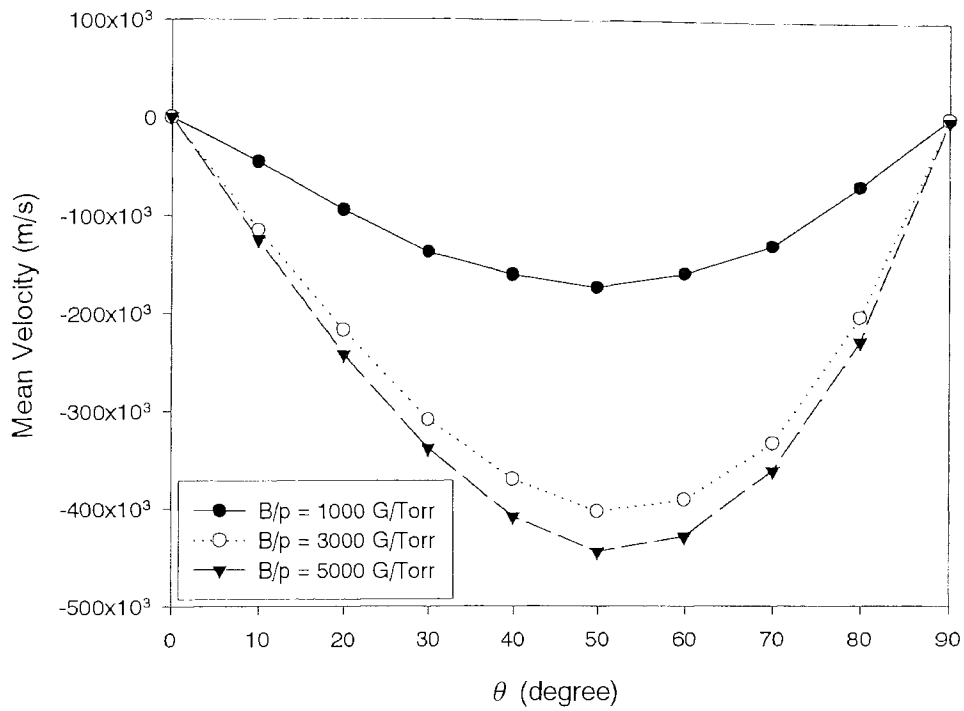


(a)  $x$  成分,  $E/N=2830$ Td,  $p=0.1$ Torr



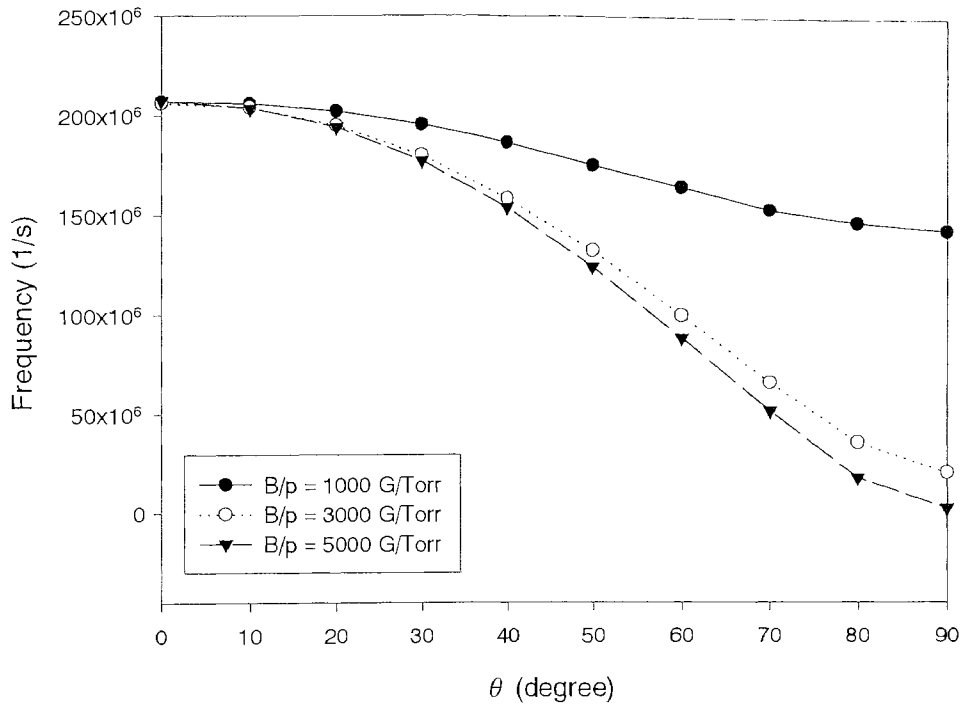
(b)  $y$  成分,  $E/N=2830$ Td,  $p=0.1$ Torr

図 4-14 電界と磁界の異なる角度における各方向の平均速度

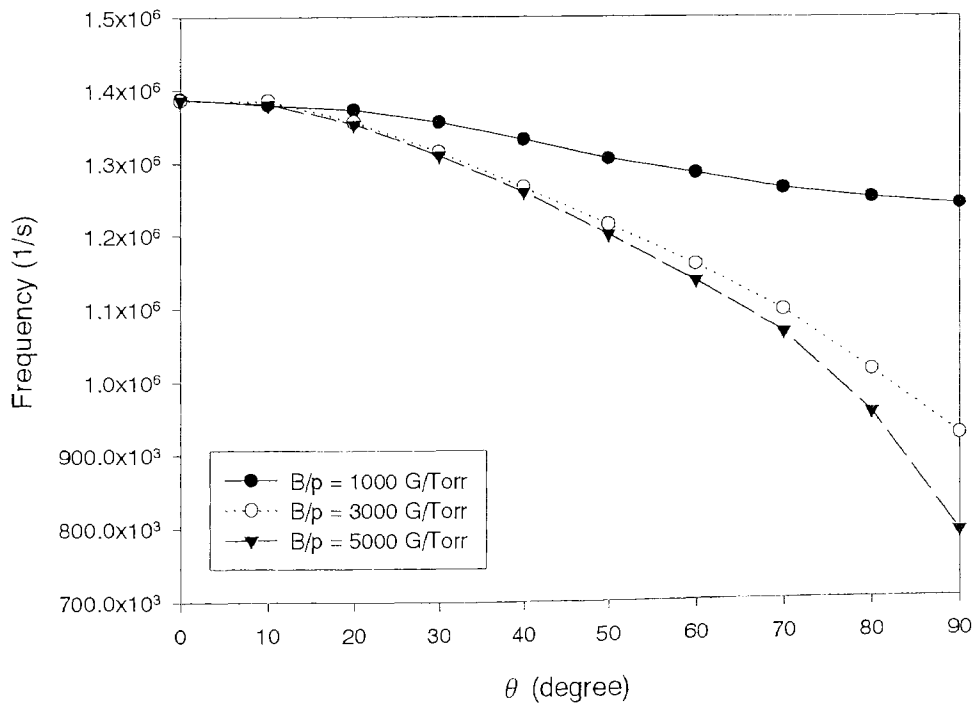


(c)  $z$  成分,  $E/N=2830$ Td,  $p=0.1$ Torr

図 4-14 電界と磁界の異なる角度における各方向の平均速度

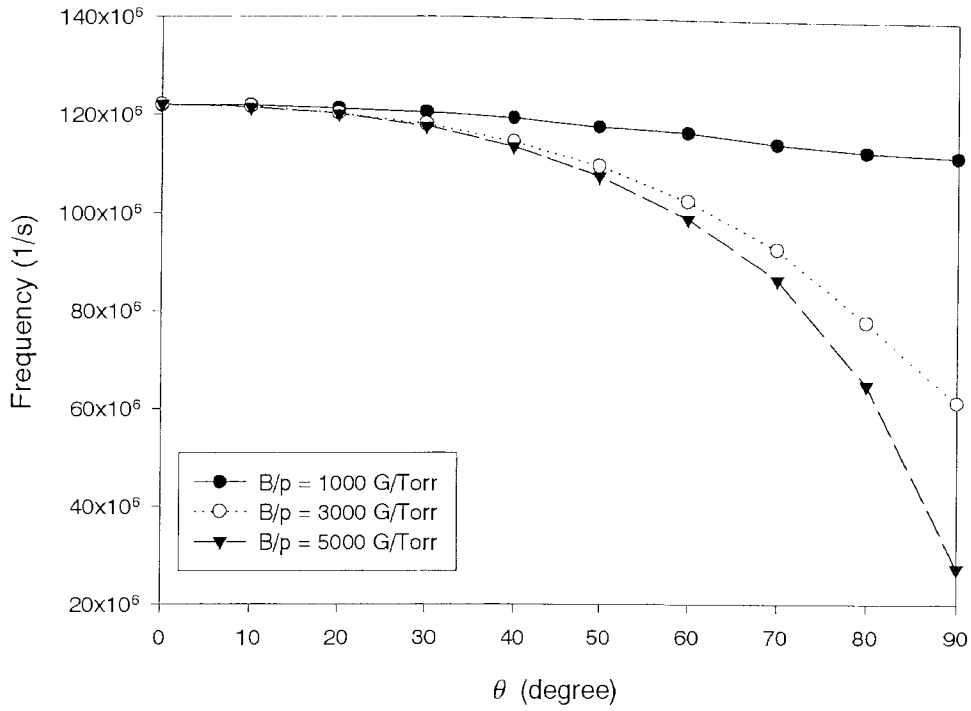


(a) 電離衝突周波数

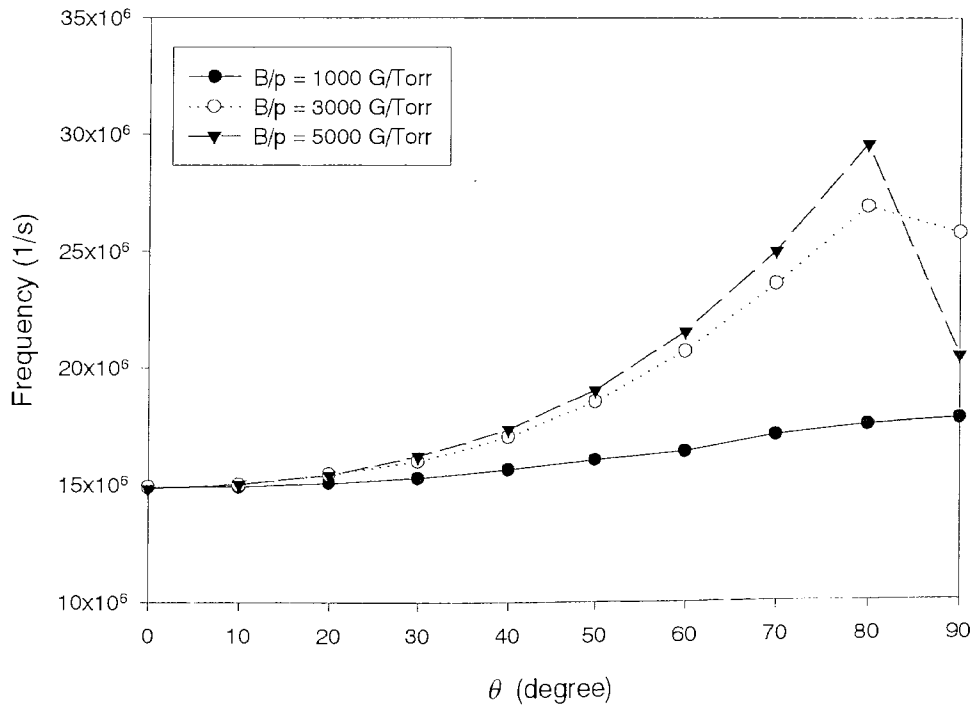


(b) 付着衝突周波数

図 4-15 電界と磁界の異なる角度におけ衝突周波数

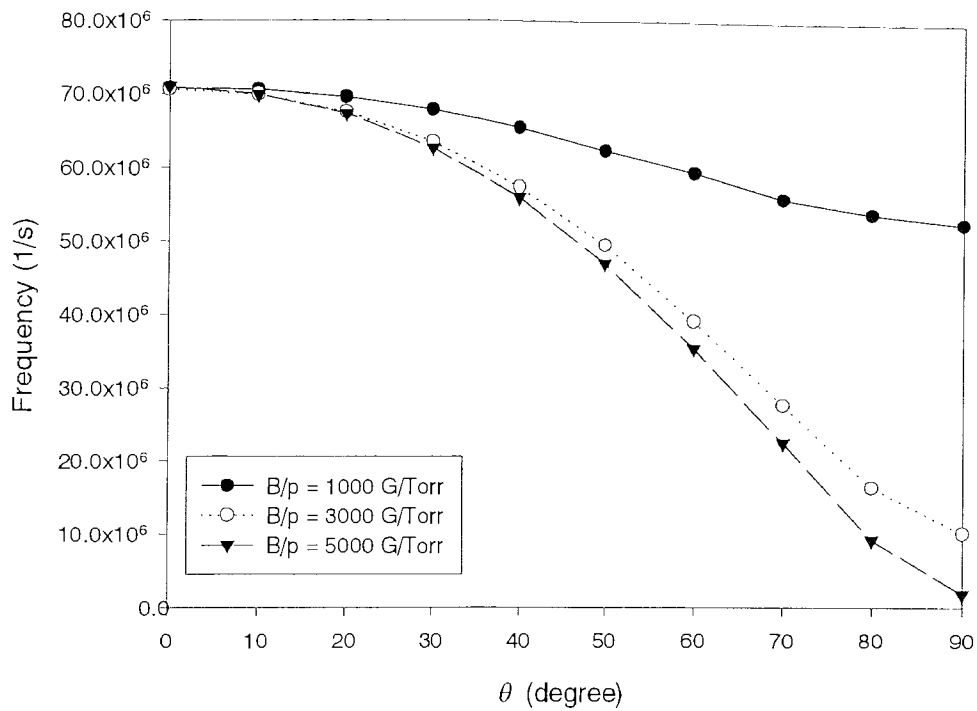


(c) 励起衝突周波数 ( $q_{e1}$ )

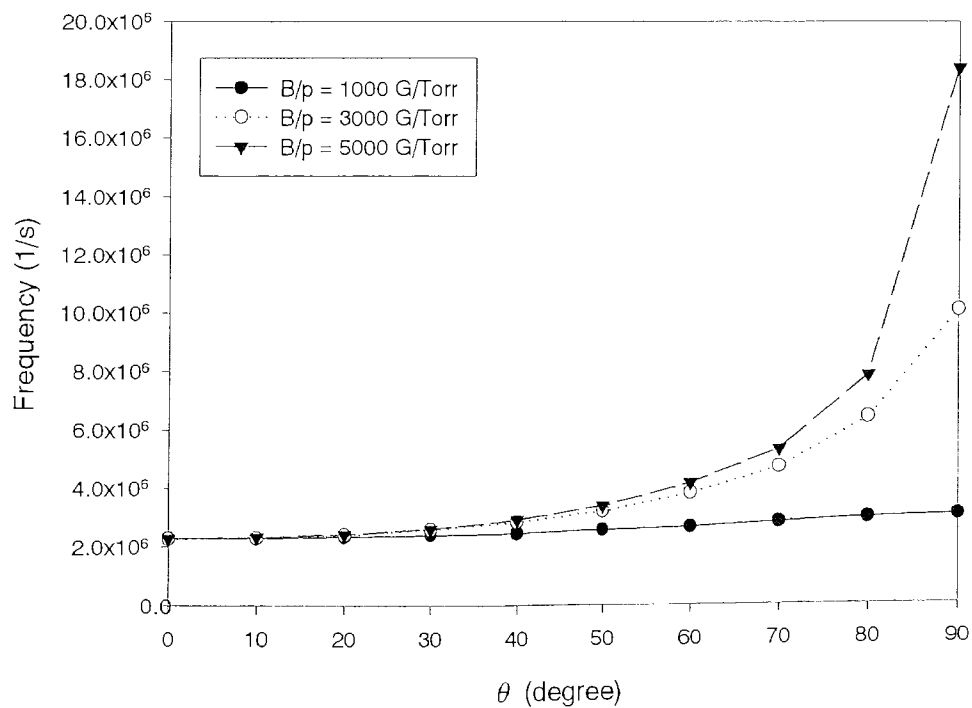


(d) 励起衝突周波数 ( $q_{e2}$ )

図 4-15 電界と磁界の異なる角度におけ衝突周波数

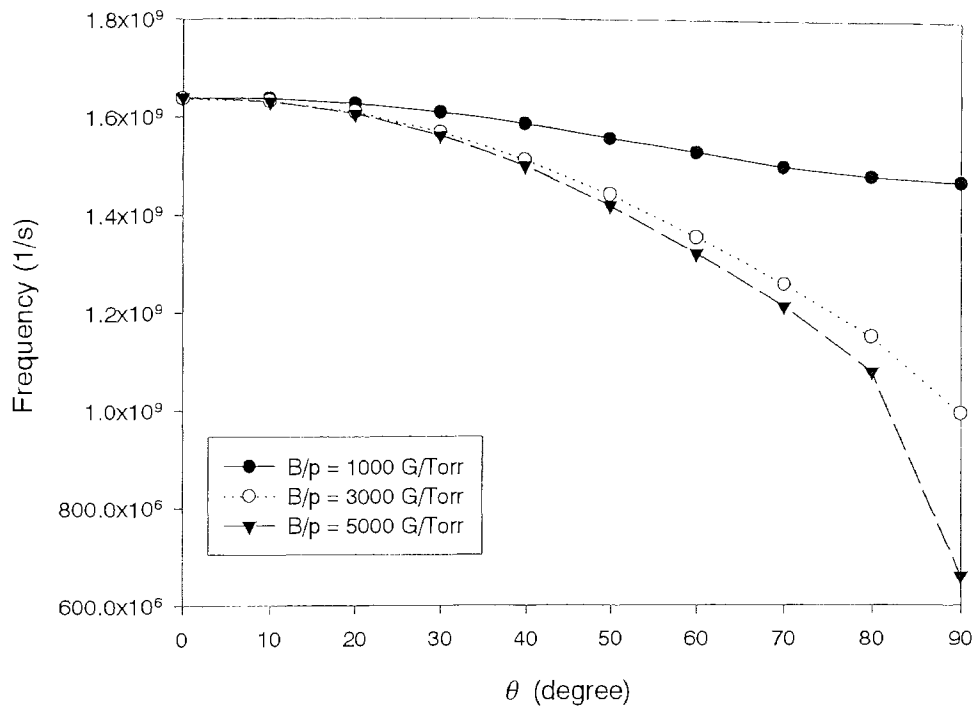


(e) 励起衝突周波数 ( $q_{e3}$ )



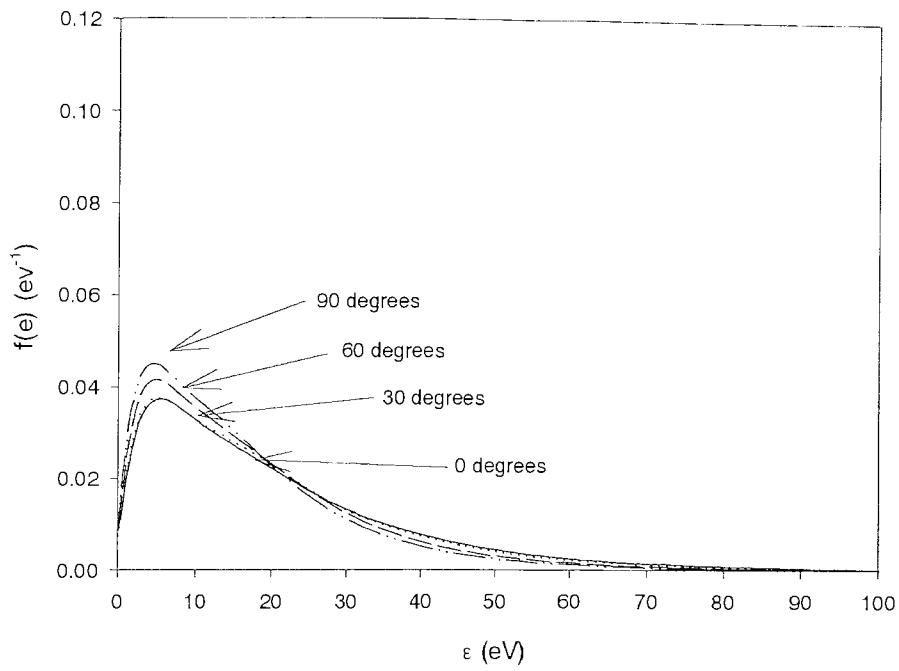
(f) 振動励起衝突周波数

図 4-15 電界と磁界の異なる角度におけ衝突周波数

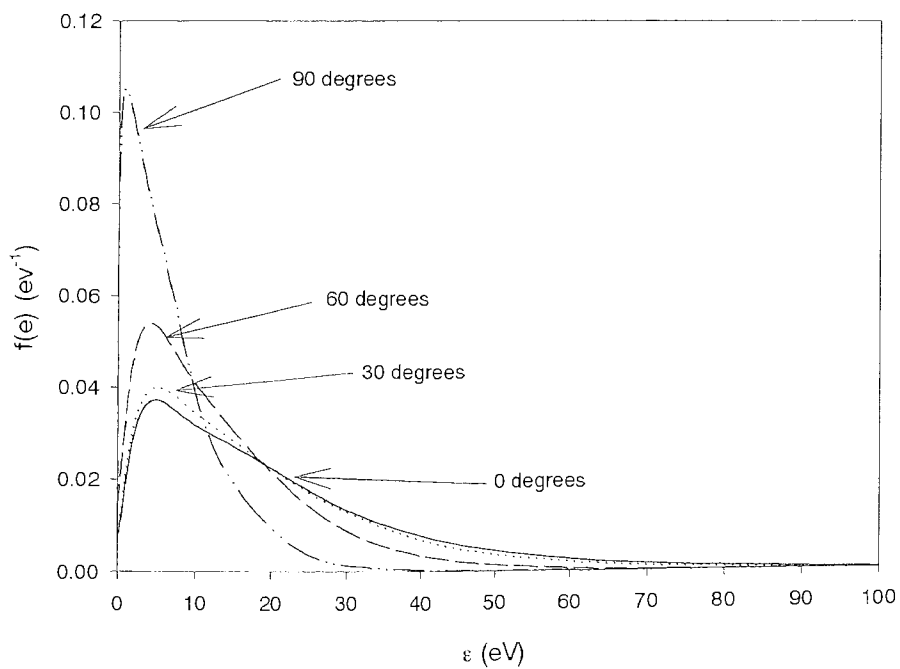


(g) 全衝突周波数

図 4-15 電界と磁界の異なる角度におけ衝突周波数

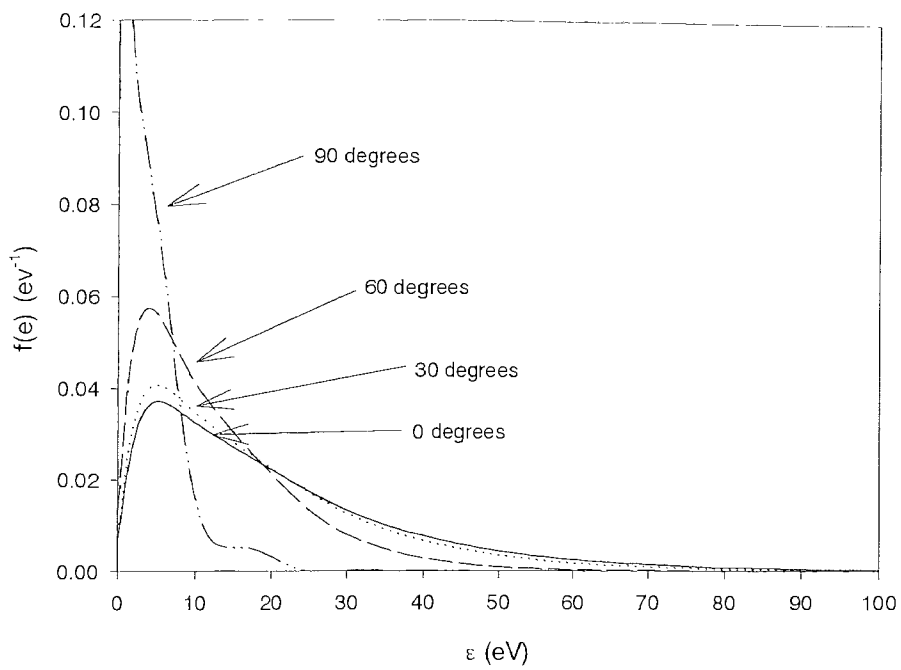


(a)  $E/N=2830\text{Td}$ ,  $B/p=1000\text{G/Torr}$



(b)  $E/N=2830\text{Td}$ ,  $B/p=3000\text{G/Torr}$

図 4-16 電界と磁界の異なる角度におけるエネルギー分布



(c)  $E/N=2830\text{Td}$ ,  $B/p=5000\text{G/Torr}$

図 4-16 電界と磁界の異なる角度におけるエネルギー分布



#### 4-3-4 Ar ガス中の電界と磁界の成す角の違いによるプラズマ・パラメータ

図 4-17 は電界と磁界の成す角度を変えて与えたときの平均エネルギーを示したグラフである。SiH<sub>4</sub> ガス、Cl<sub>2</sub> ガス同様に、0 度のとき、平均エネルギーが最大になり、角度が大きくなるにしたがって電子は磁界によって曲げられ、エネルギーを得にくくなり、角度とともに平均エネルギーが下がっていく。

図 4-18 は各方向の平均エネルギーを示している。SiH<sub>4</sub> ガス、Cl<sub>2</sub> ガス同様に、 $x$  軸方向に磁界が印加されているので  $x$  軸方向の平均エネルギーが  $y$  軸方向と  $z$  軸方向よりも大きくなっている。また、50 度付近から急に  $z$  軸方向のエネルギーが下がっている。

図 4-19 は異なる電界と磁界の角度による各方向の平均速度を示している。 $z$  軸方向の速度成分は 60 度付近に極大値が存在している。また、 $y$  軸方向については 300,500 Gauss/Torr においてほぼ同様の速度となっている。

図 4-20 は Ar ガスについて、平均衝突周波数を求めたものである。電界と磁界の角度が大きくなるにつれて電界の影響が妨げられ、衝突回数が減少していくことが示されている。

図 4-21 はエネルギー分布を求めたものであり、密度が 1 となるように正規化して示している。0 度と 30 度においては高磁界においてもほぼ変化がなく、60 度付近で分布に若干変化が見られる結果となっている。

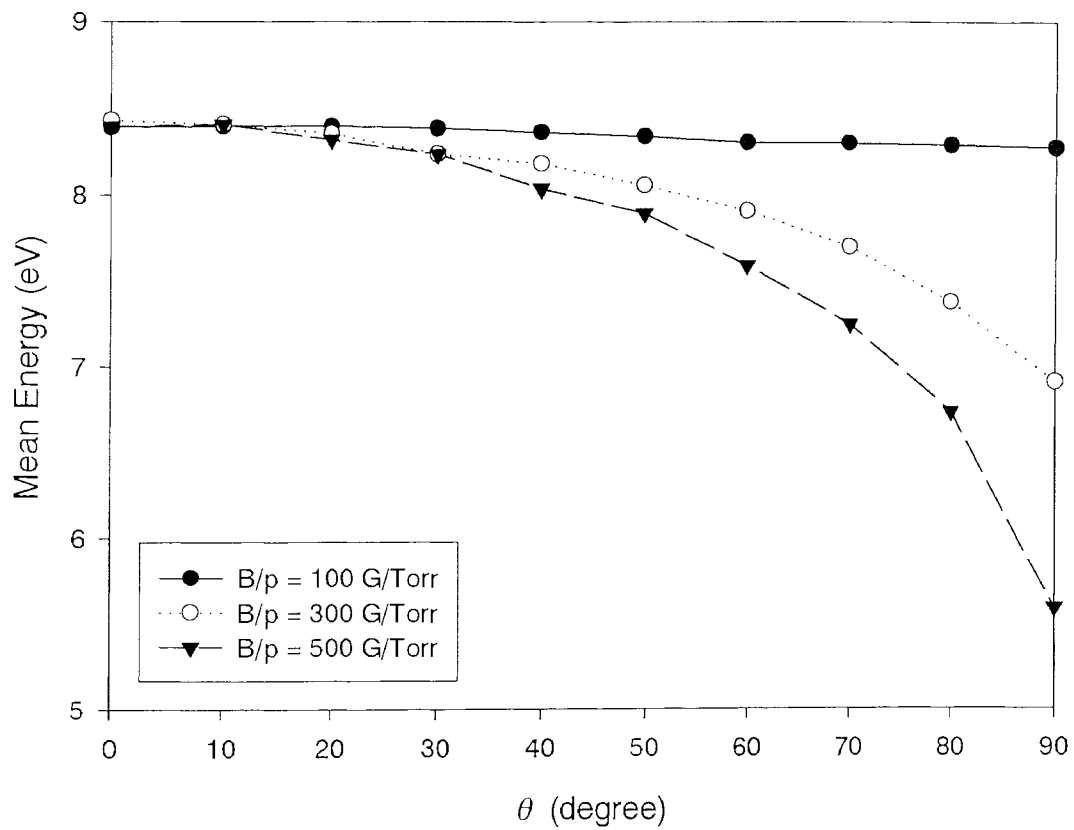
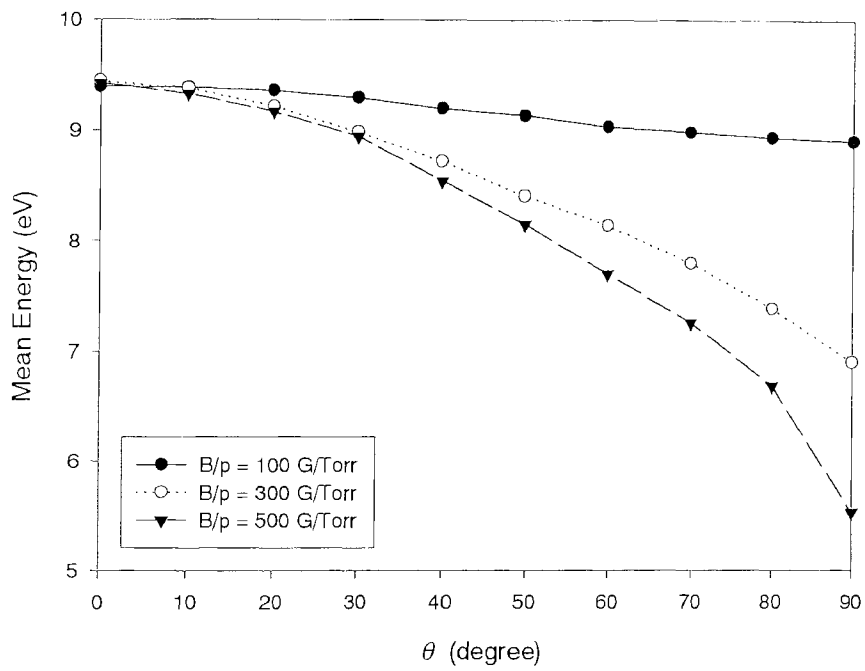
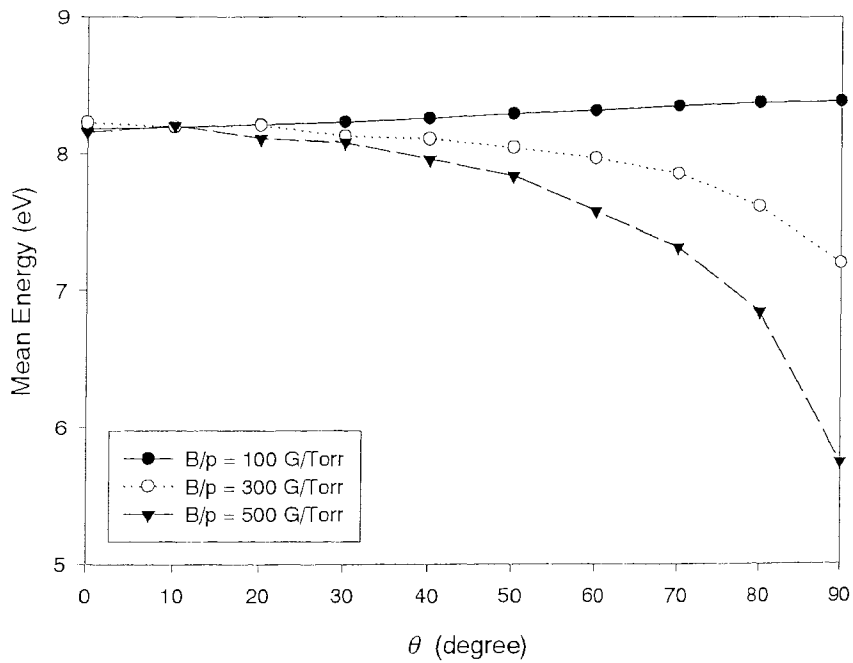


図 4-17 電界と磁界の異なる角度における平均エネルギー :  $E/N=283\text{Td}$

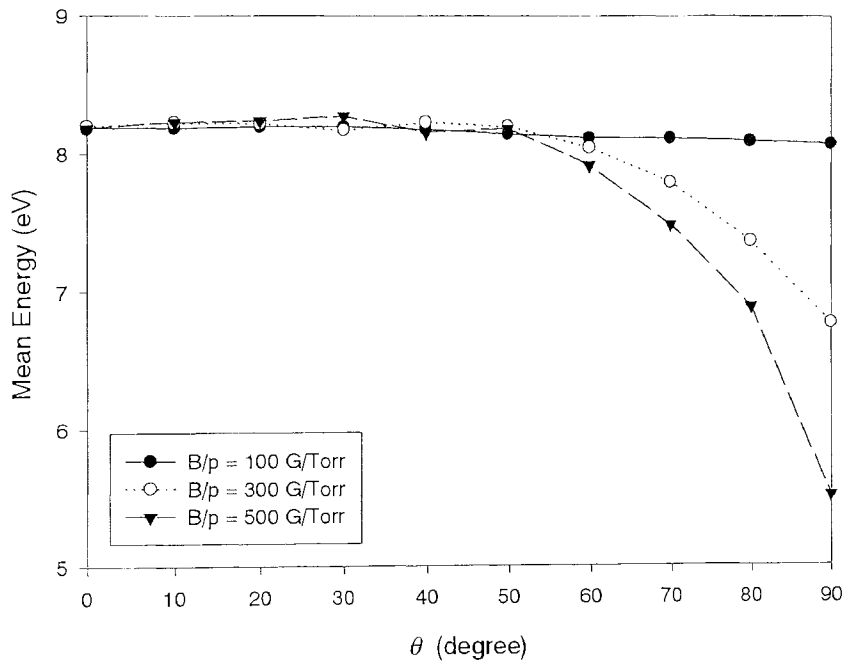


(a) x 成分, E/N=283Td, p=0.1Torr



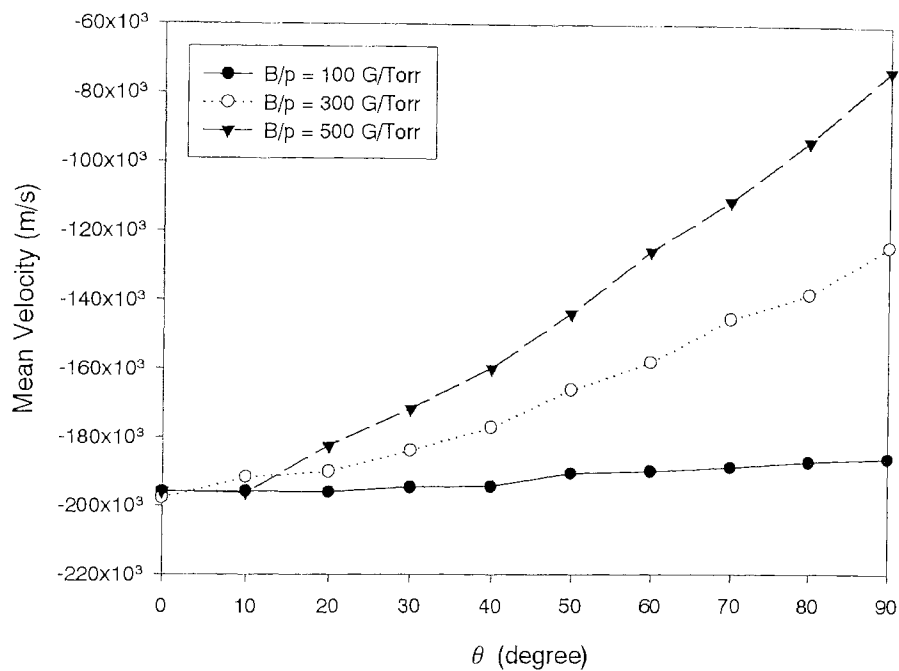
(b) y 成分, E/N=283Td, p=0.1Torr

図 4-18 電界と磁界の異なる角度における各方向のエネルギー

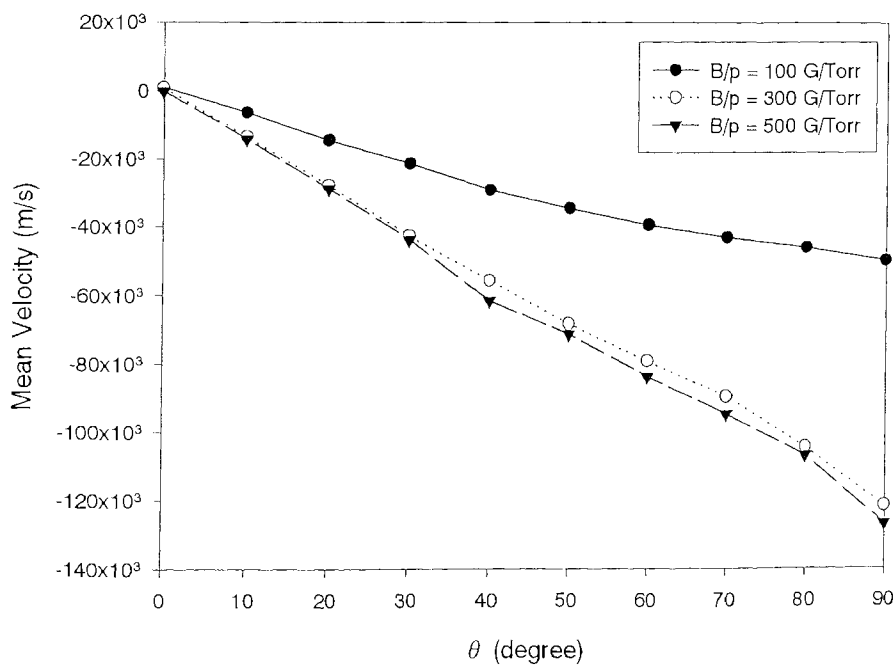


(c) z 成分,  $E/N=283\text{Td}$ ,  $p=0.1\text{Torr}$

図 4-18 電界と磁界の異なる角度における各方向のエネルギー

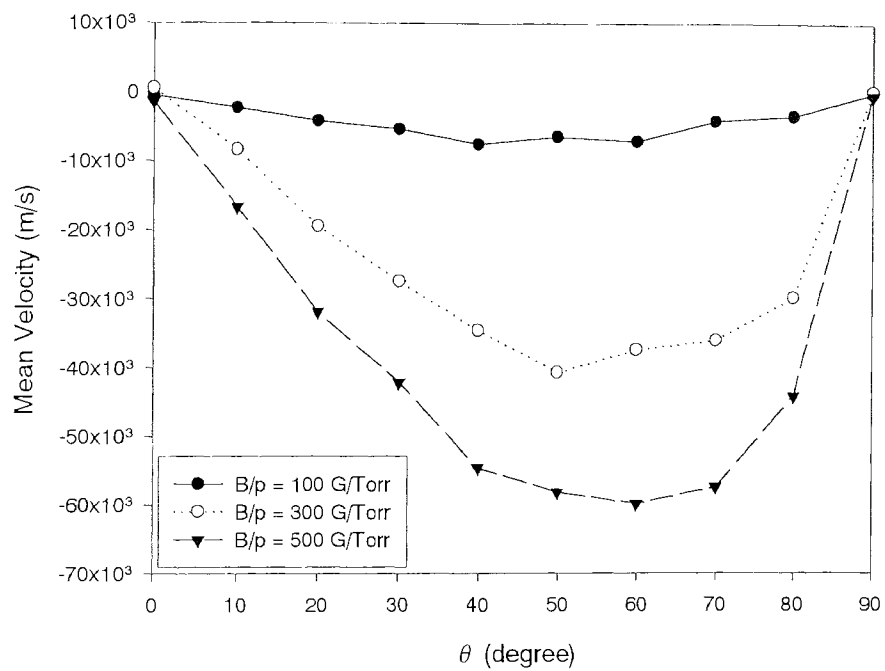


(a)  $x$  成分,  $E/N=283Td$ ,  $p=0.1Torr$



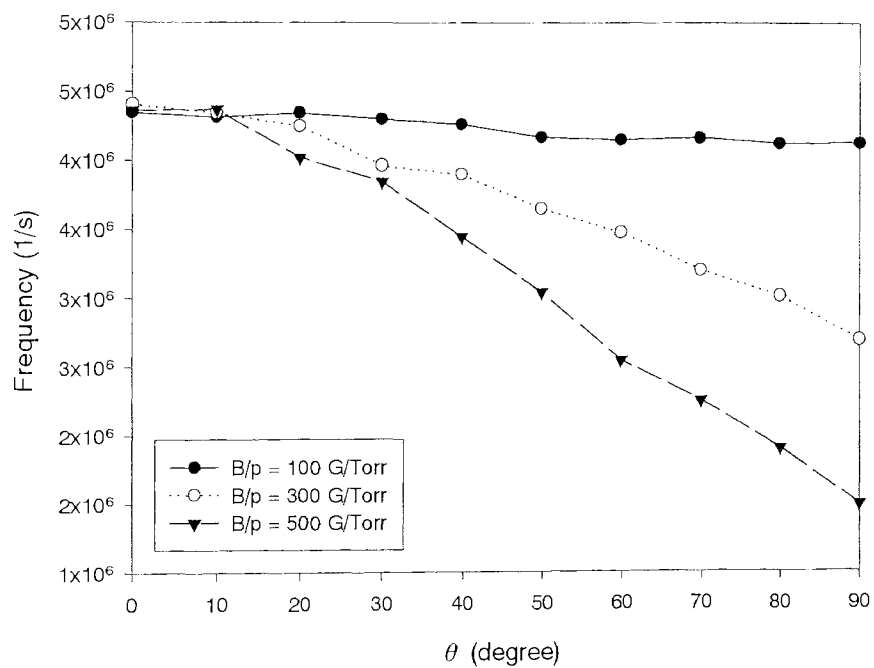
(b)  $y$  成分,  $E/N=283Td$ ,  $p=0.1Torr$

図 4-19 電界と磁界の異なる角度における各方向の平均速度

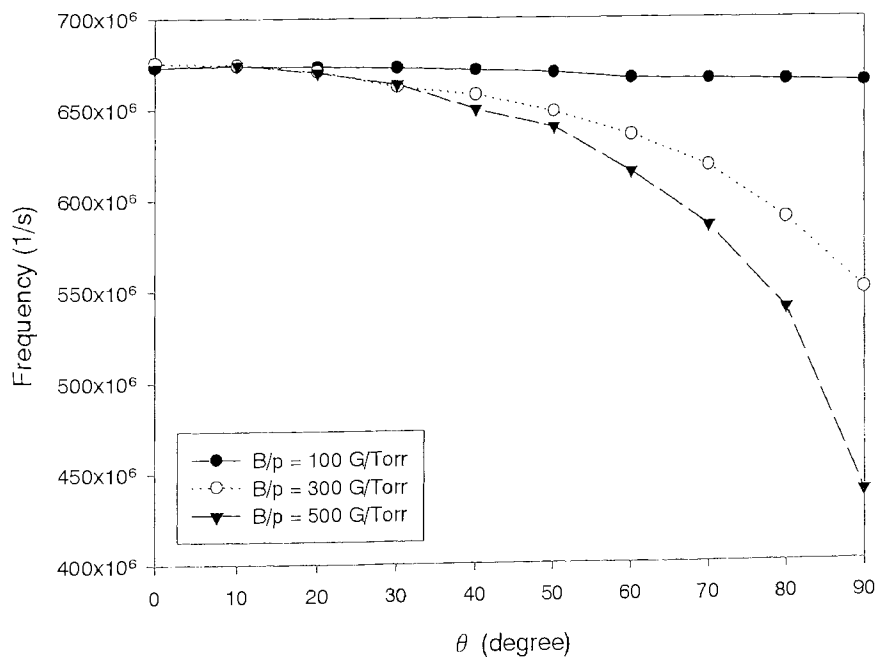


(c)  $z$  成分,  $E/N=283\text{Td}$ ,  $p=0.1\text{Torr}$

図 4-19 電界と磁界の異なる角度における各方向の平均速度

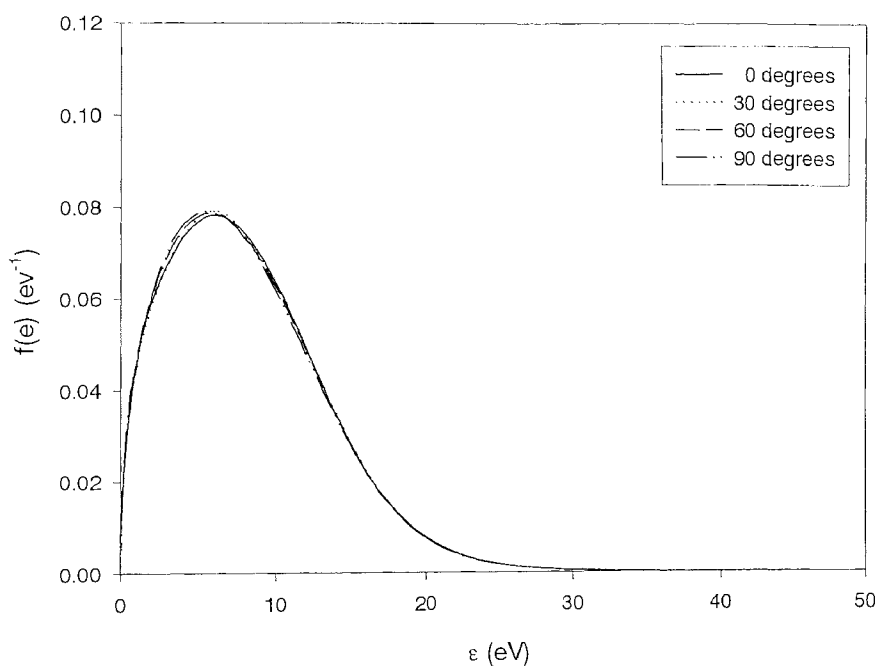


(a) 電離突周波数

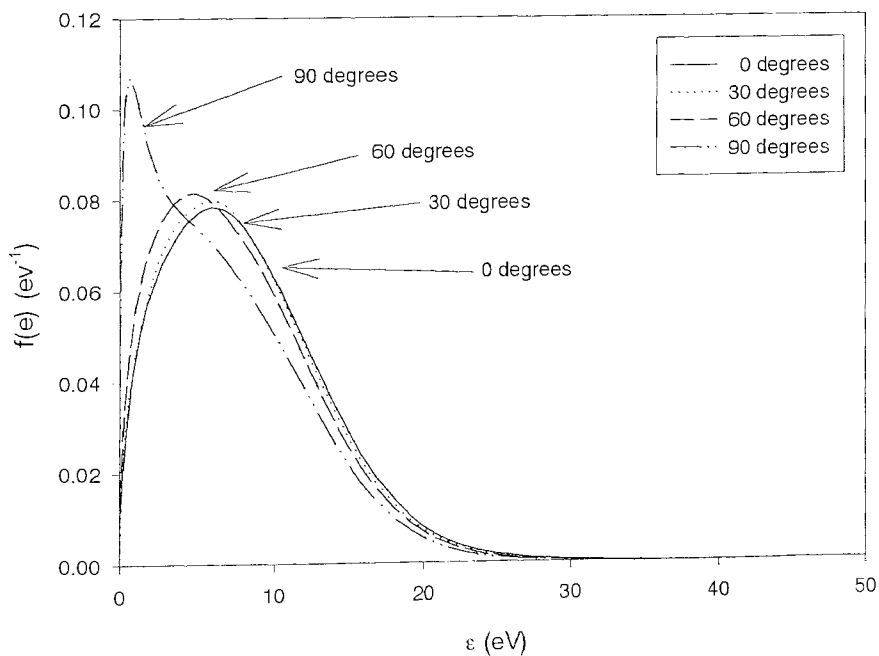


(b) 全衝突周波数

図 4-20 電界と磁界の異なる角度におけ衝突周波数



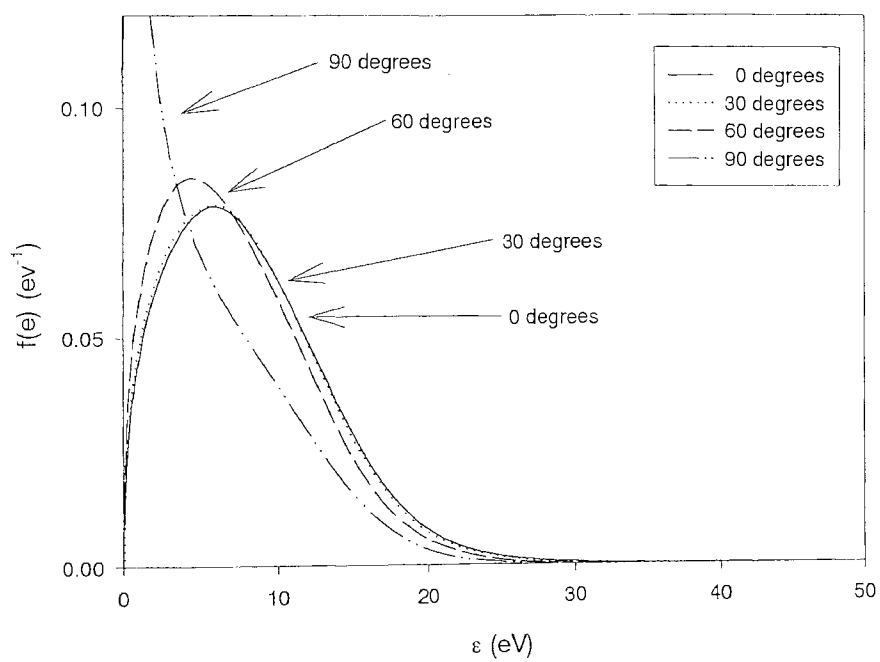
(a)  $E/N=283\text{Td}$ ,  $B/p=100\text{G/Torr}$



(b)  $E/N=283\text{Td}$ ,  $B/p=300\text{G/Torr}$

図 4-21 電界と磁界の異なる角度におけるエネルギー分布





(c)  $E/N=283\text{Td}$ ,  $B/p=500\text{G/Torr}$

図 4-21 電界と磁界の異なる角度におけるエネルギー分布

#### 4-3-5 カスプ磁界中での粒子の運動

##### (I) 磁極間の中央から電子を供給した場合

プラズマの生成・制御において、電界と磁界を直交させて用いる場合の他に、エッチングやスパッタリングなどのプロセスにおいて、カスプ磁界を用いる方法がある。

図 4-22 は、幅 1cm の磁石を中心の距離が 3cm になるように同じ面に並べて配置し  $x$  軸 0.5cm、 $y$  軸 0cm のところで 100G となるようなカスプ磁界を発生させたときの磁界を示したグラフである。カスプ磁界において、電界によって加速される電子の運動は、場所によって電界と磁界の角度が異なり、さらに電界と磁界の強さも異なり、これに衝突が加わるため電子の運動は  $E \times B$  フィールドよりも複雑になる。

本論文では、図 4-23 のように直流電界によるカスプ磁界において計算を行った。図 4-24(a)は、直流の電界を  $x$  軸のマイナス方向に 80V/cm で与え、幅 1cm の磁石を中心の距離が 3cm になるように電極面に並べて配置し、 $x$  軸 0.5cm、 $y$  軸 0cm のところで 80G となるようにカスプ磁界を与え、また、二つの磁極の中心から電子を  $10 \text{ cm}^3/\text{ns}$  の割合で発生させたときの密度分布を示している。

図 4-24(a)の上段のグラフは、 $x$  軸と  $y$  軸での密度分布であり下段のグラフは  $x$  軸と  $z$  軸での密度分布である。いま、 $x$  軸のマイナス方向に直流電界であり、磁界は図 4-22 のようにカスプ磁界が与えられているので、磁極間の中央では電界と磁界は直角になり、電子スオームのドリフト方向は  $z$  軸方向となる。したがって、図 4-24(a),(b)に示すように、電界によって  $x$  軸のプラス方向に加速されて進むが、磁界の影響によって  $z$  軸方向に曲げられて電極方向に戻され、再び電界によって  $x$  軸のプラス方向に加速されて進んでいくサイクロトロン運動が示されている。図 4-24(c)のグラフはエネルギー分布であり、 $x$  軸のプラス方向へのドリフトとともにエネルギーが大きくなっている。しかし、図 4-24(a)のグラフで示されているように、電子スオームが電極に戻される場合、電子に対する電界の影響は、電子の動きを妨げるように

働くために電極付近では、エネルギーの高い電子がないことを示していると考えられる。また、図 4-24(d)のグラフは全衝突周波数であり、 $x$  軸のプラス方向へのドリフトが最大となるところで衝突周波数が最大となっていることを示している。

図 4-25 は電界を  $x$  軸のマイナス方向に  $160\text{V/cm}$  と大きくしたときの結果であり、電界により  $x$  軸のプラス方向へのドリフトが大きくなっていることを示している。このとき、電子は磁極間の中心から供給しているので、 $\mathbf{E} \times \mathbf{B}$  フィールドとして考えることができる。したがって、電界を  $(\mathbf{E}, 0, 0)$ 、磁界を  $(0, \mathbf{B}, 0)$  とすると、運動方程式は

$$m \frac{dv_x}{dt} = e(E - v_z B) \quad (4-21a)$$

$$m \frac{dv_y}{dt} = 0 \quad (4-21b)$$

$$m \frac{dv_z}{dt} = ev_x B \quad (4-21c)$$

となる。 $v_x$  と  $x$  はそれぞれ、

$$v_x = \frac{eE}{m\omega^2} \sin \omega t \quad (4-22)$$

$$x = \frac{eE}{m\omega^2} (1 - \cos \omega t) \quad (4-23)$$

である。ここで、 $e$  は電荷、 $m$  は質量、 $\omega$  をサイクロトロン周波数とすると

$$\omega = eB/m \quad (4-24)$$

$$t_1 = \frac{\pi}{\omega} \quad (4-25)$$

で位置が最大となるので、式(4-23)より

$$x_{\max} = \frac{2eE}{m\omega^2} = \frac{2mE}{eB^2} \quad (4-26)$$

となり最大の位置が求まる。

たとえば、図 4-25(a)では、 $E = 160V/cm$ 、 $x = 0$ から折り返し地点までの平均の磁界を  $B_{avg} = 65.9G$  とすると  $x_{max} = 0.42cm$  となり、図 4-26 で示す結果とよく一致している。これらの結果から、磁極間の中心から電子を供給した場合は、磁界は常に電界の方向と垂直となる  $E \times B$  方向へドリフトし、磁界方向への拡散およびドリフトはないことを示している。

### (II) 磁極付近から電子を供給した場合 (1点出発)

図 4-27(a),(b)は、中心から  $-9mm$ 、図 4-27(c),(d)は、中心から  $+9mm$  の位置から電子を供給した場合の結果である。磁極付近では、磁界は電界と平行になるため電界は磁界によって妨げられず、電子は電界と平行に加速される。ところが、磁界は図 4-22 に示したように、電極から離れるにしたがって電界と磁界の角度が大きくなり  $z$  軸方向にドリフトし、次には電界により  $x$  軸方向にドリフトしたのち  $z$  軸方向にドリフトするという過程を繰り返すことにより、 $z$  軸方向にドリフトしながら  $x$  軸方向にドリフトしているのを示している。

### (III) 磁極付近から電子を供給した場合 (2点出発)

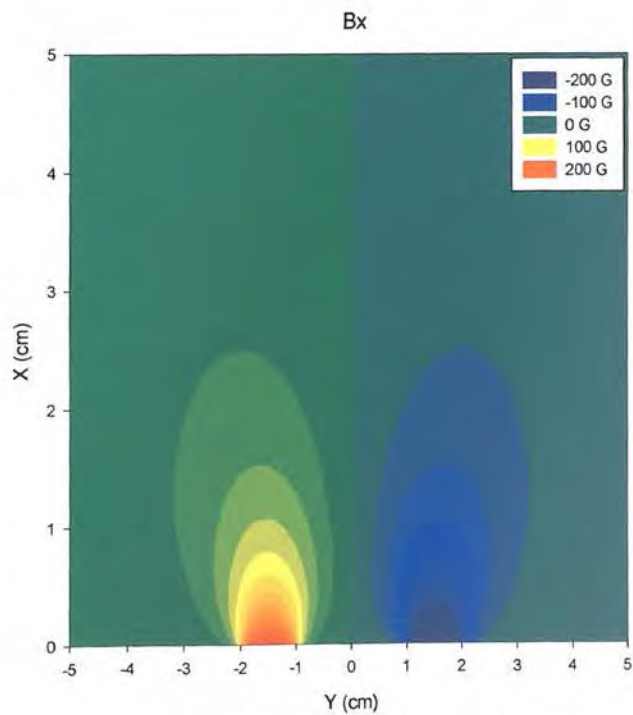
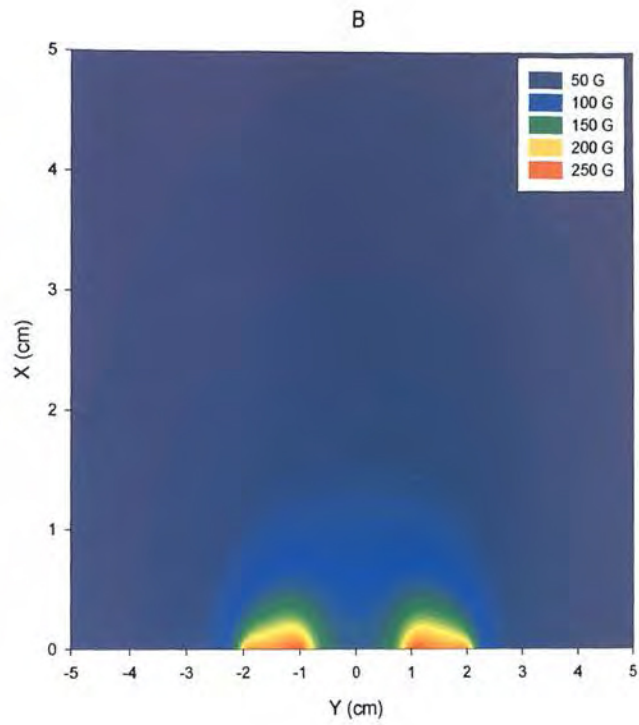
図 4-28(a),(b),(c)は、それぞれ二つの場所から電子を供給した場合の結果である。出発位置を磁極から遠ざけるにしたがい、図 4-22 に示したように、電界と磁界との角度は大きくなるが磁界は小さくなっているため、磁極から離れている方が  $x$  軸方向へドリフトすると考えられ、また、逆に磁極に近い方は、磁界が大きく中央へ曲げられ、さらに  $z$  軸方向にドリフトするために、 $x$  軸方向へのドリフトは小さくなると考えられる。したがって、図 4-28(a)で示す結果よりも図 4-28(c)で示した結果の方が、 $x$  軸方向へのドリフトが大きい。また、 $z$  軸方向ドリフトの大きさは磁極から離れた方が大きくなっている。これは、磁極付近よりも中央の方が電界と磁界の角度は大きく、 $z$  軸方向への  $E \times B$  ドリフトとなるからであろうと考えられる。

図 4-29 は全衝突周波数を示しており、磁極付近から出発させた方が、電

界と磁界の角度は小さく、主に電界によって加速されるために衝突が多くなると考えられる。逆に中央付近から出発させた場合は、電界と磁界の角度は大きくなり電界の影響が小さくなるために電子が加速されず、衝突回数が減ると考えられる。したがって、図 4-29(c)に示されているように中央付近から出発させた場合は、電極付近では電子は十分に加速されないため衝突が少なく、電極から離れるにしたがってエネルギーを得ると考えられるので、 $x$  軸において最大となる場所付近で衝突周波数が高くなっていると考えられる。

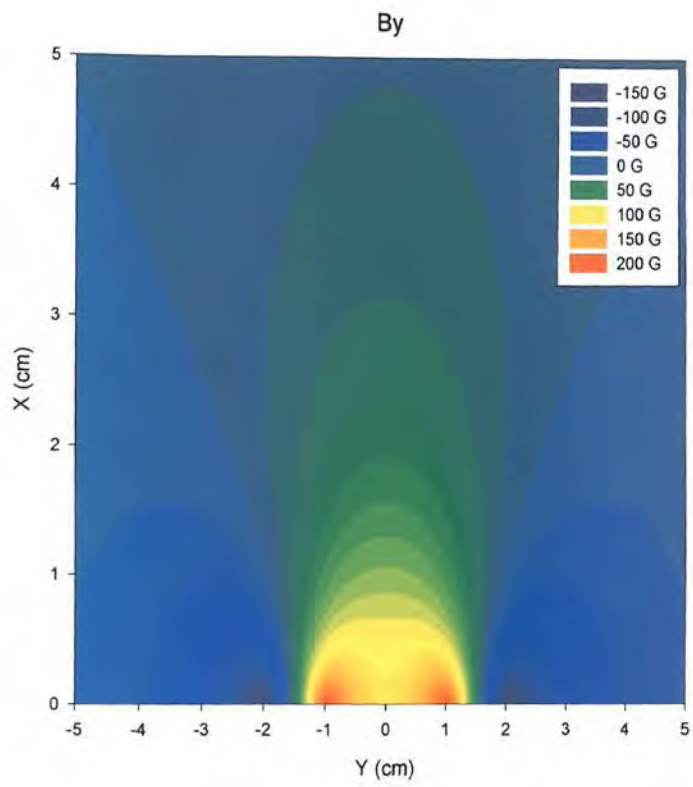
#### (IV) 磁極間で一様に電子を供給した場合

図 4-30 は、磁極間を間を一様に電子を供給した場合の密度分布を示しており、磁界の強さを変えてシミュレーションを行った結果である。この結果は、上述した(I)、(II)、(III)の現象が同時に起こっていることを示している。したがって、密度分布は中央が最も  $x$  軸方向にドリフトし、磁極に近づくと内側に曲げられている。また、磁界を強めれば、 $x$  軸方向のドリフトは抑えられ、 $z$  軸方向にドリフトしていくと考えられる。しかし、図 4-30(e),(f)より  $x$  軸方向のドリフト範囲が狭くなると、折り返してきた電子が電極に取り込まれてしまい、往復運動ができずに  $z$  軸方向にドリフトできないことがわかる。



磁極間中心距離 3cm、磁極の幅 1cm、 $X=0.5\text{cm}$ 、 $Y=0\text{cm}$  のところで 100G

図 4-22 カスプ磁界の磁力線



(c)  $B_y$  方向

図 4-22 カスプ磁界の磁力線

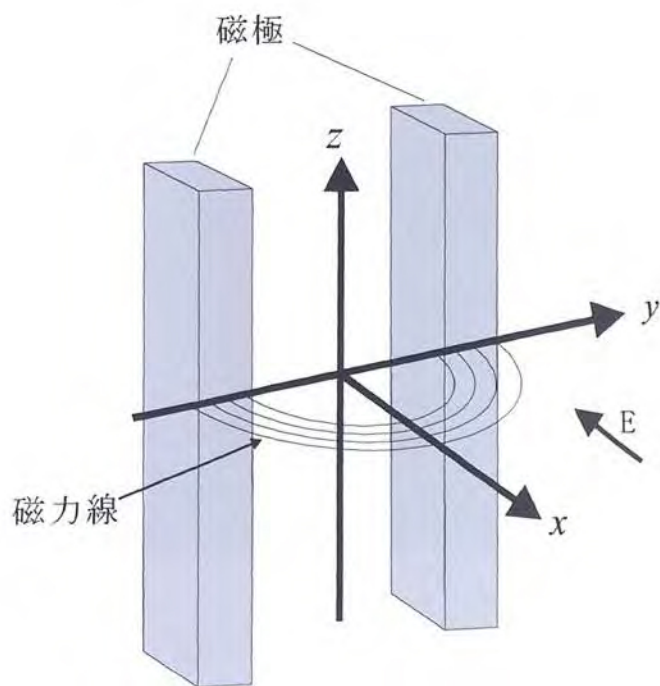


図 4-23 同じ面に二つ磁極を配置したカスプ磁場

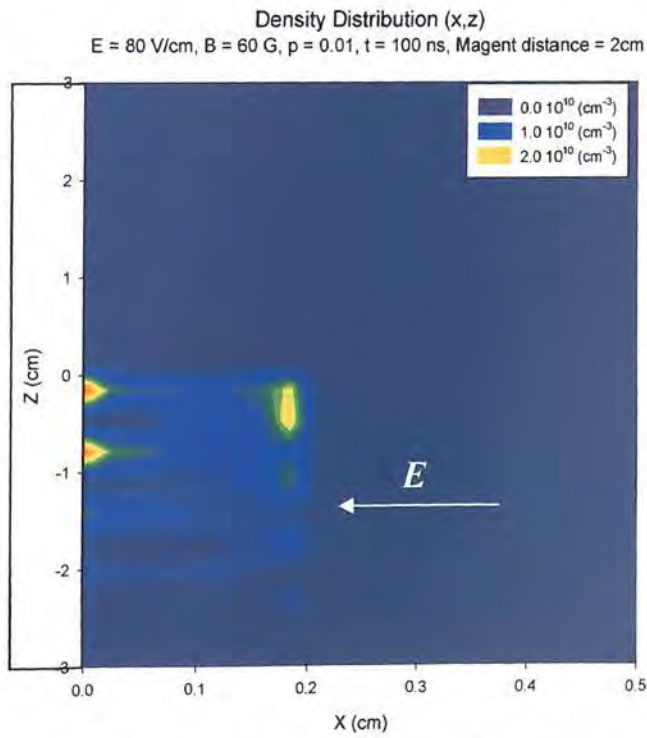
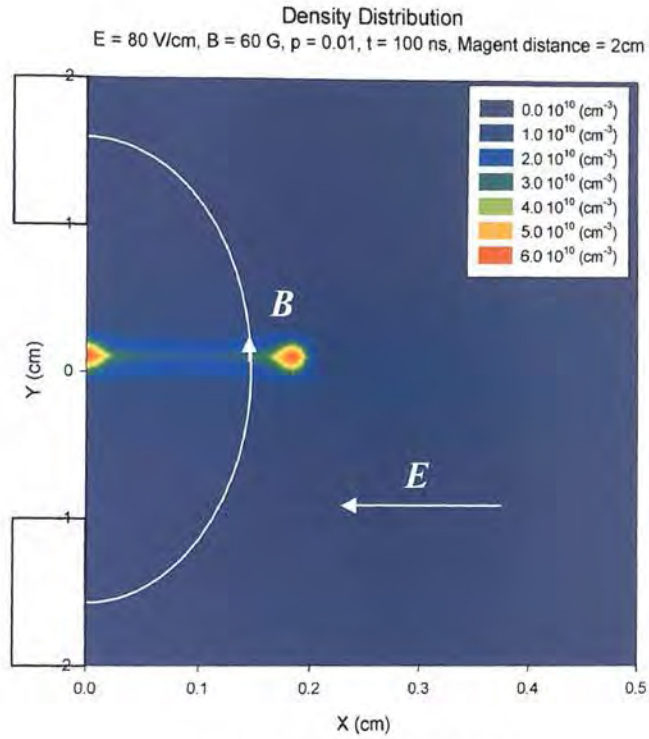
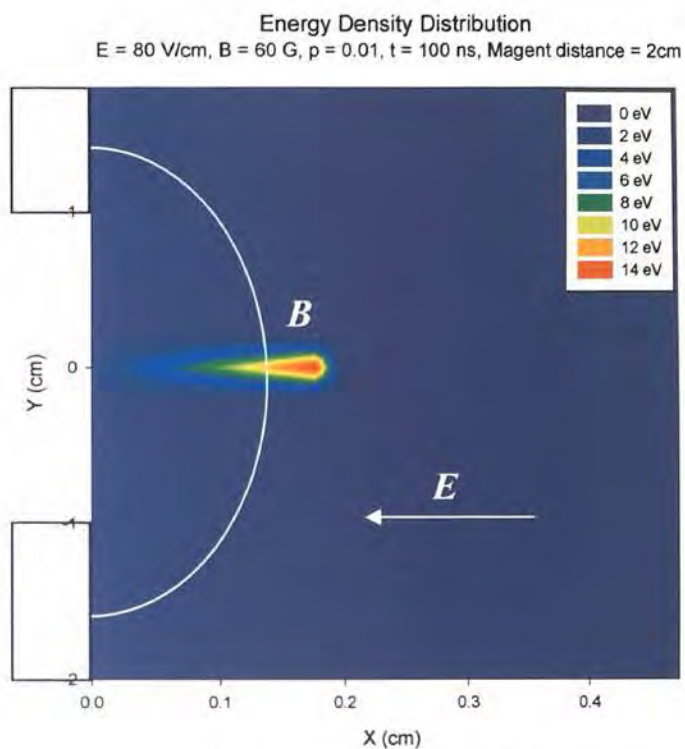
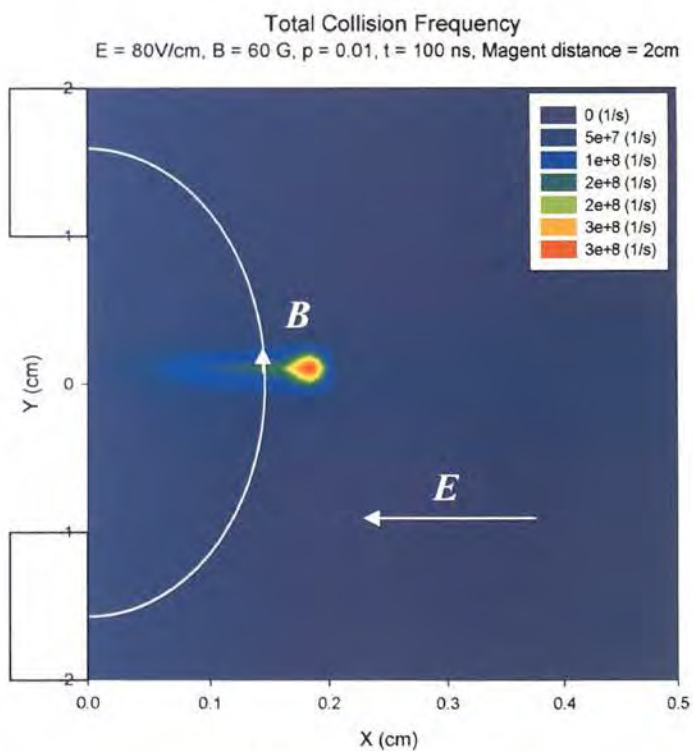


図 4-24 磁極の中央から電子を出発させたときの分布



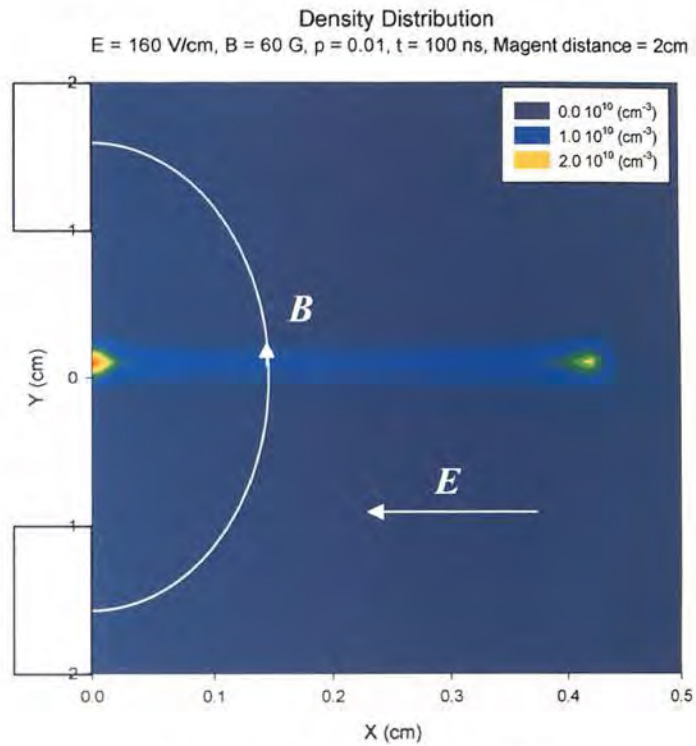


(c) エネルギー分布

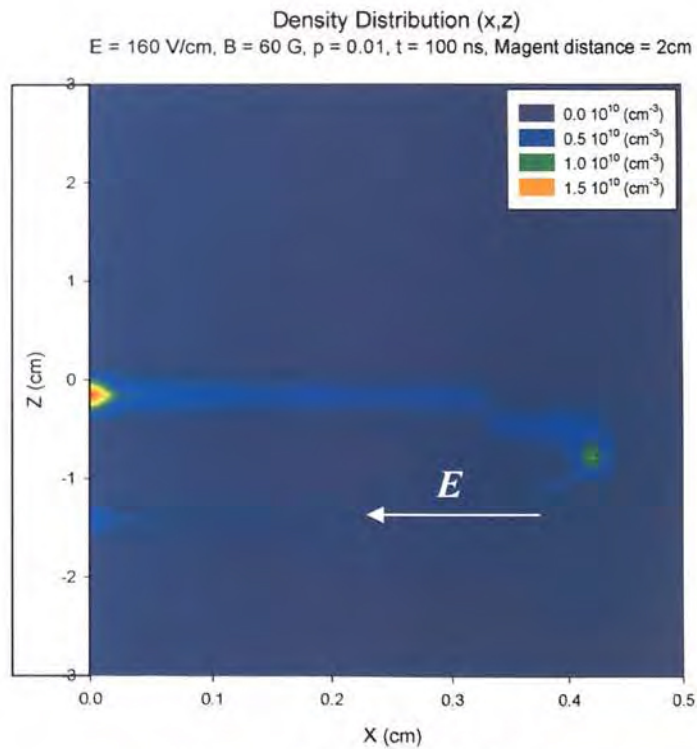


(d) 全衝突周波数

図 4-24 磁極の中央から電子を出発させたときの分布



(a) XY 平面の密度分布



(b) XZ 平面の密度分布

図 4-25 磁極の中央から電子を出発させたときの分布

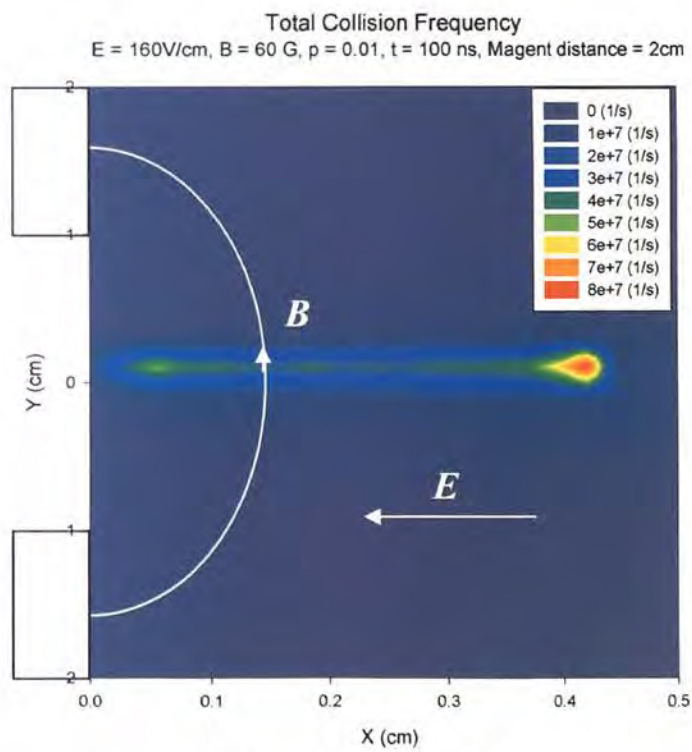
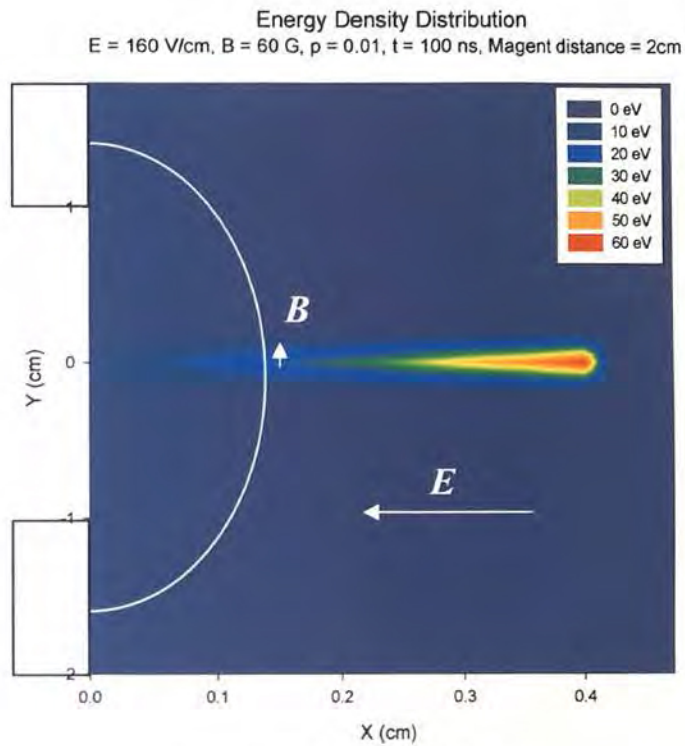
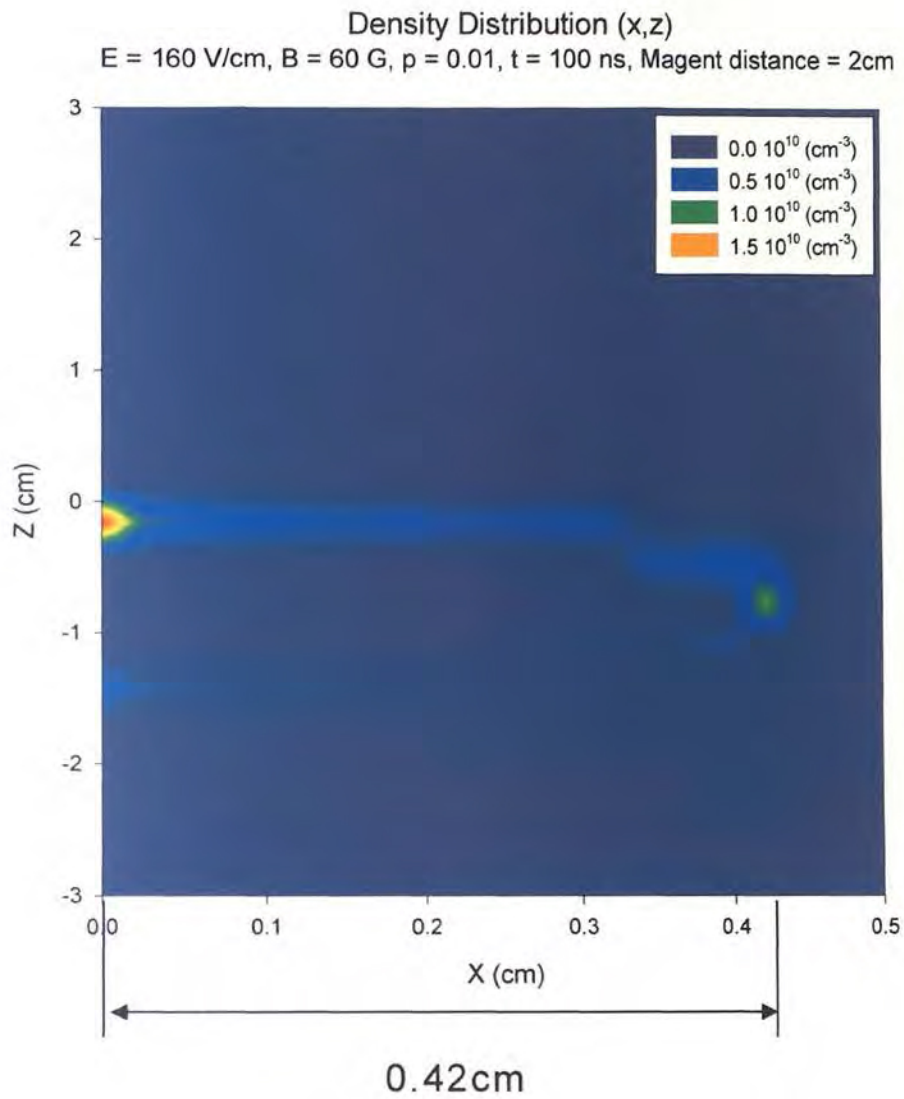


図 4-25 磁極の中央から電子を出発させたときの分布

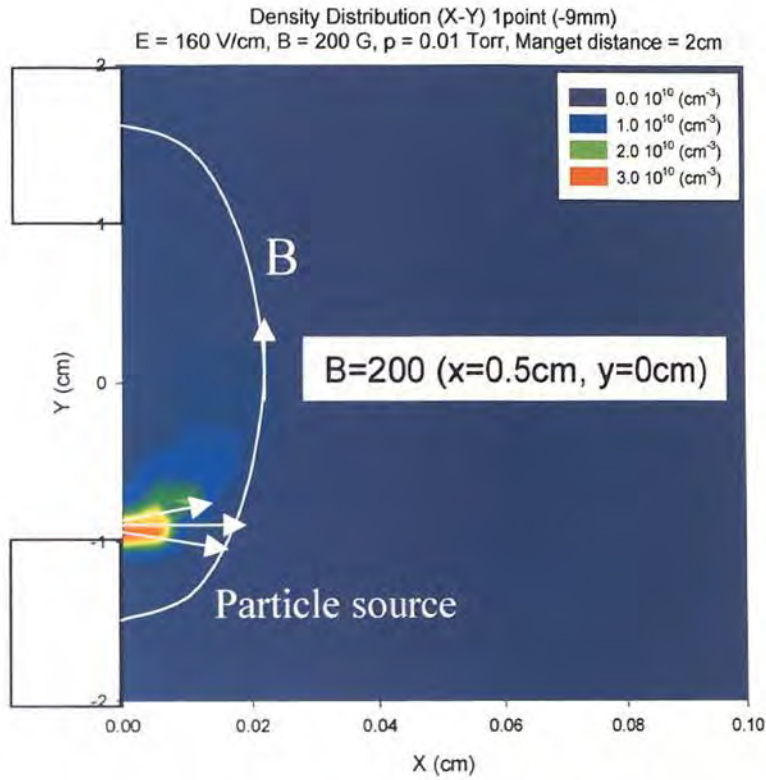


$$E = 160 \text{ V/cm} = 16000 \text{ V/m}$$

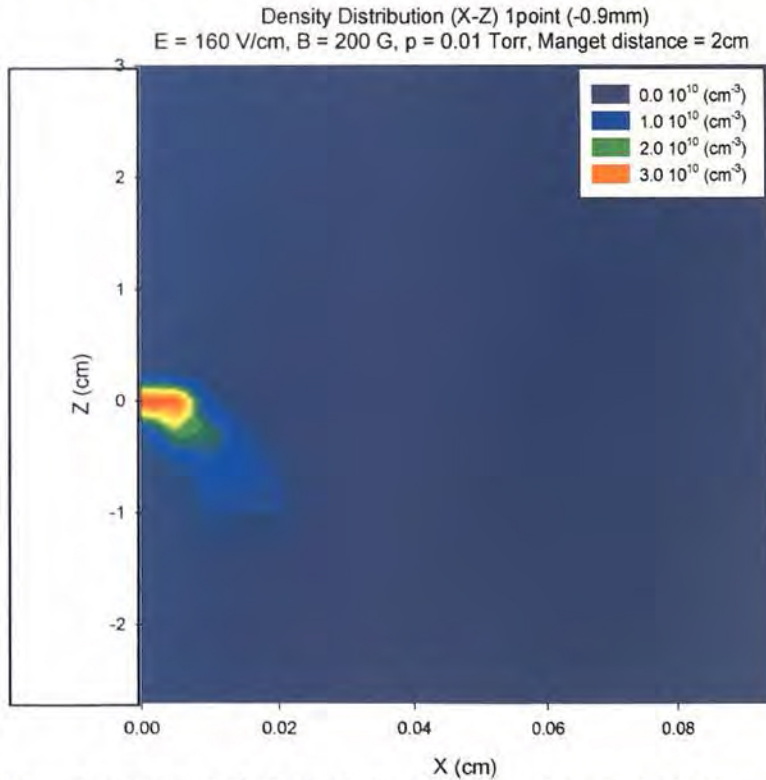
$$B_{avg} = 65.9 \text{ G} = 0.00659 \text{ T}$$

$$x_{max} = 0.42 \text{ cm}$$

図 4-26 最大のドリフト距離

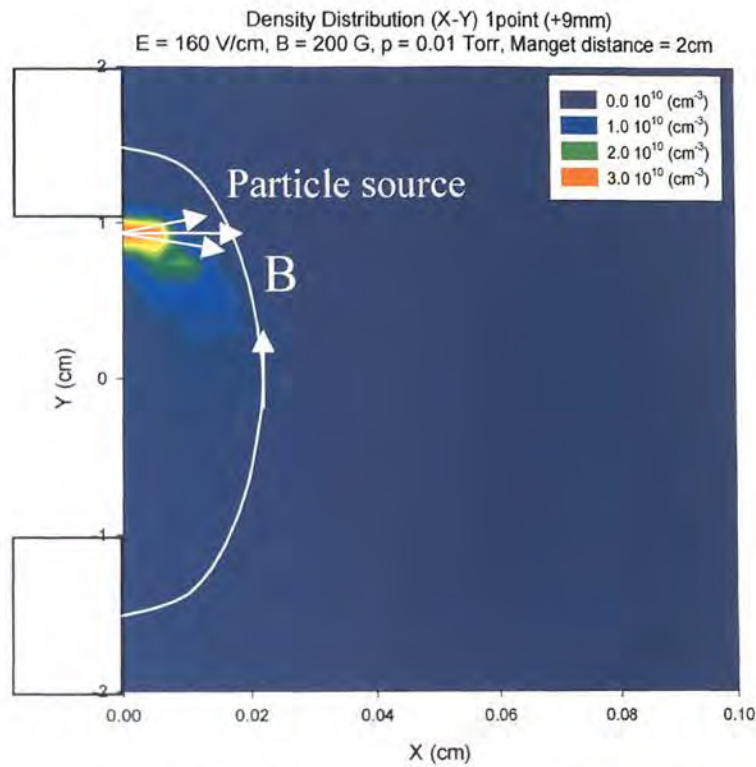


(a) XY 平面の密度分布 (中心から-9mm から出発)

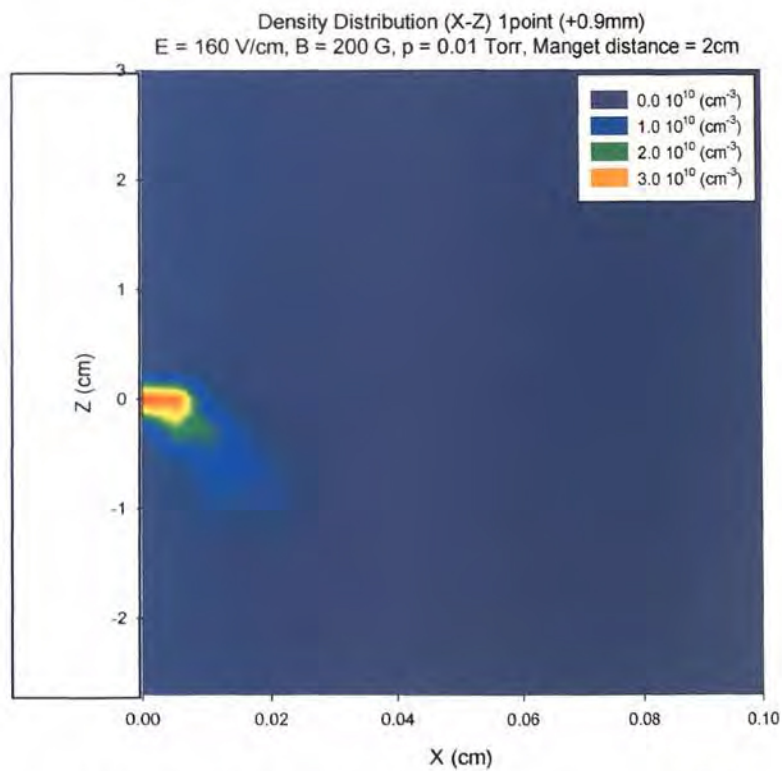


(b) XZ 平面の密度分布 (中心から-9mm から出発)

図 4-27 磁極近傍から電子を供給した場合の密度分布

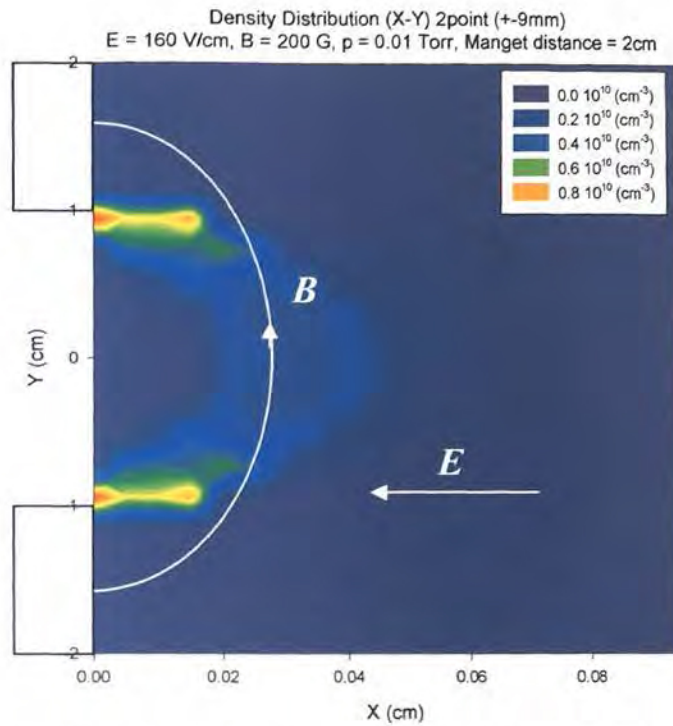


(c) XY 平面の密度分布 (中心から+9mm から出発)

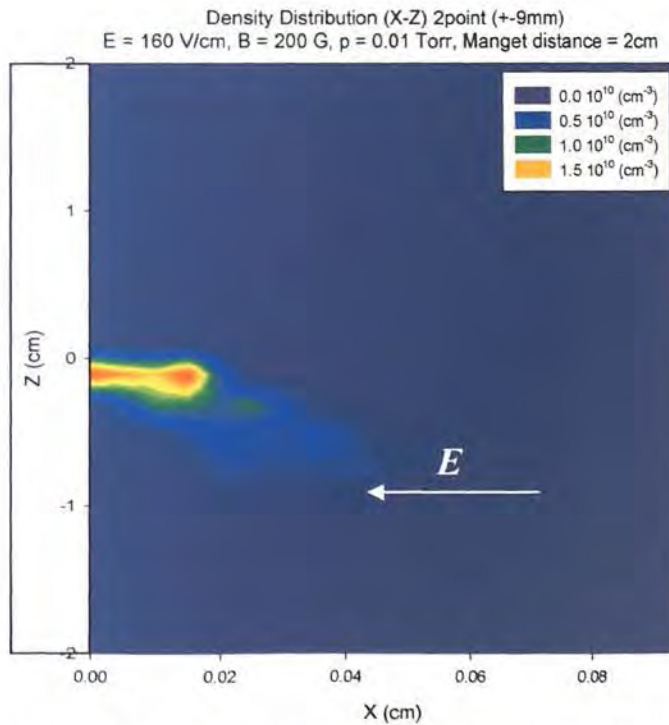


(d) XZ 平面の密度分布 (中心から+9mm から出発)

図 4-27 磁極近傍から電子を供給した場合の密度分布

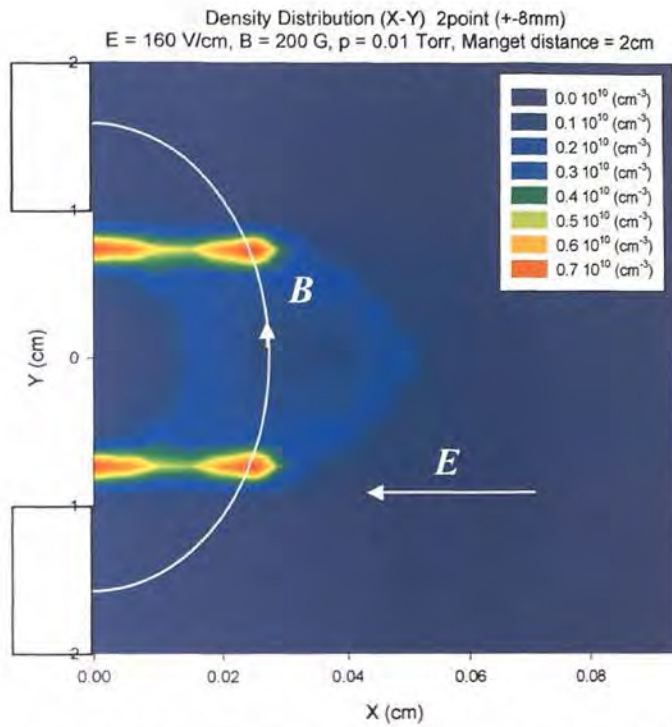


(a) XY 平面の密度分布 (中心から+9mm、-9mm から出発)

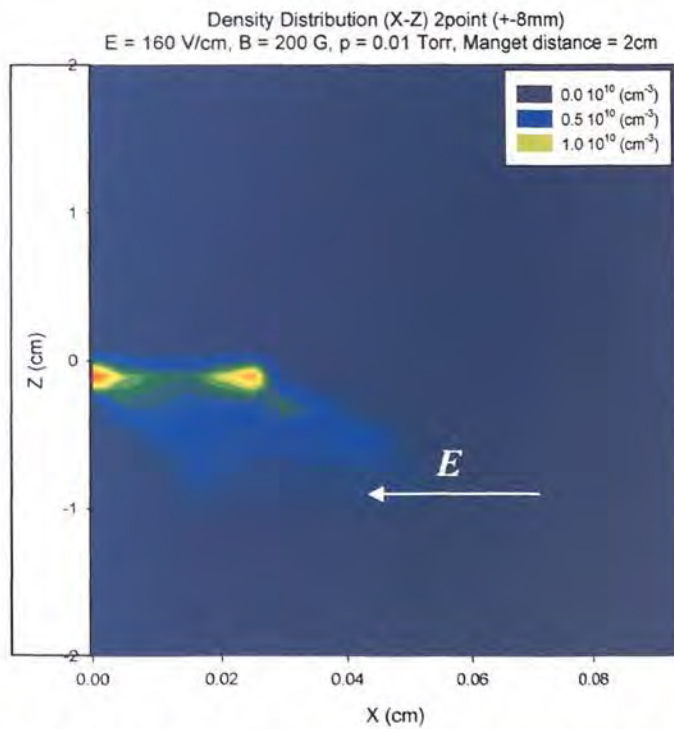


(b) XZ 平面の密度分布 (中心から+9mm、-9mm から出発)

図 4-28 2つの場所から電子を供給した場合の密度分布



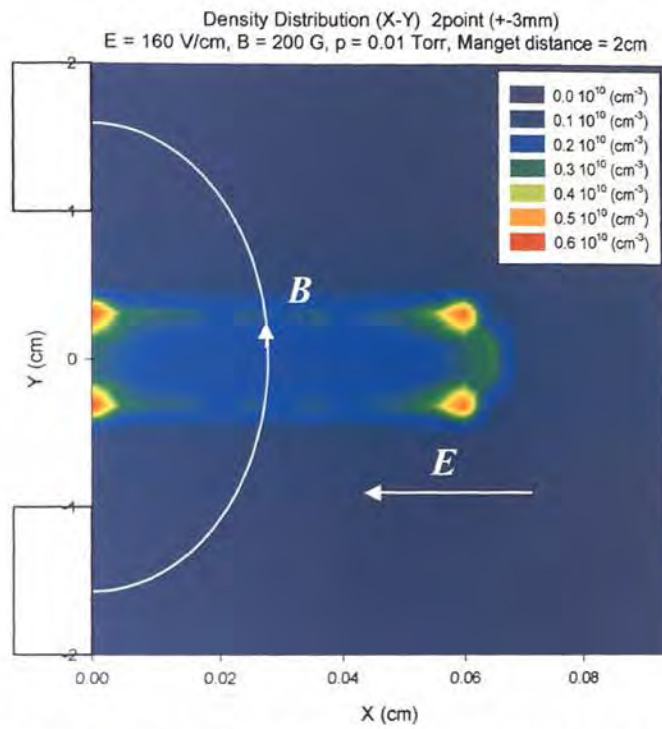
(c) XY 平面の密度分布 (中心から+8mm、-8mm から出発)



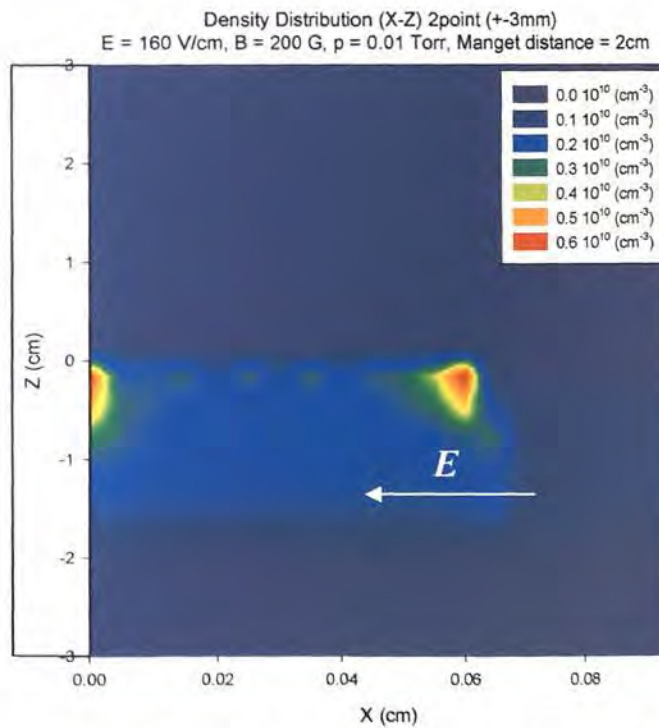
(d) XZ 平面の密度分布 (中心から+8mm、-8mm から出発)

図 4-28 2つの場所から電子を供給した場合の密度分布



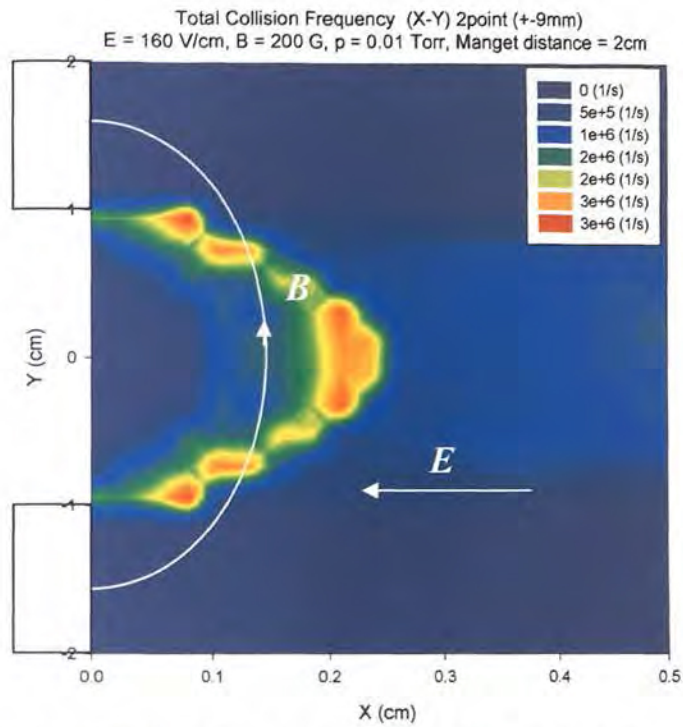


(e) XY 平面の密度分布 (中心から+3mm、-3mm から出発)

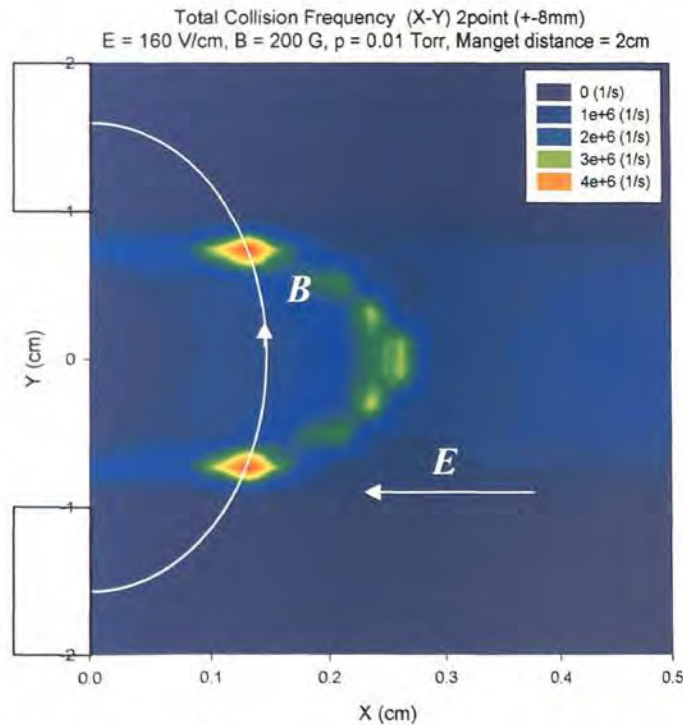


(f) XZ 平面の密度分布 (中心から+3mm、-3mm から出発)

図 4-28 2つの場所から電子を供給した場合の密度分布

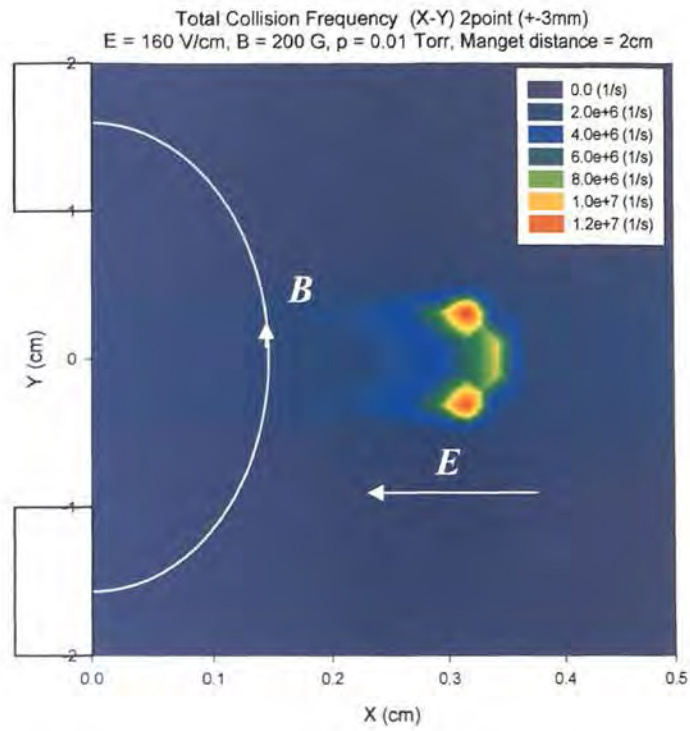


(a) 全衝突周波数 (中心から+9mm、-9mm から出発)



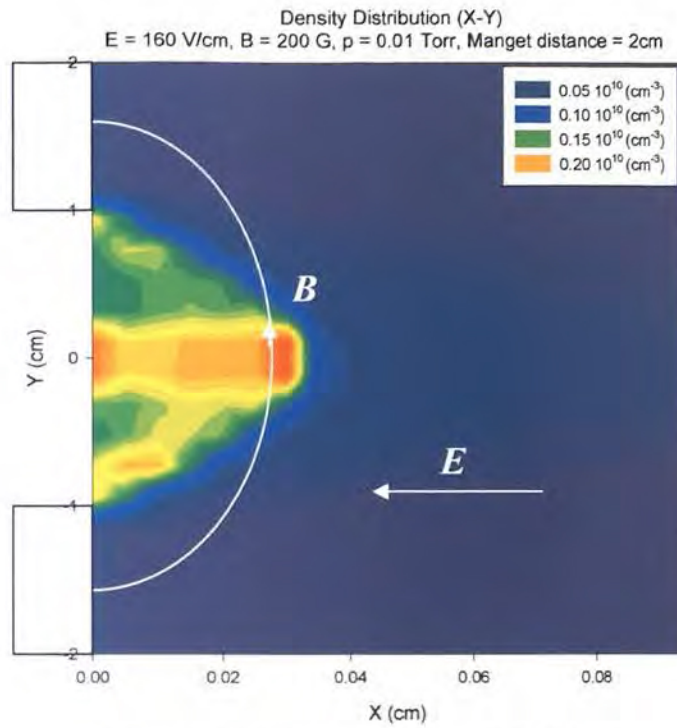
(b) 全衝突周波数 (中心から+8mm、-8mm から出発)

図 4-29 2つの場所から電子を供給した場合の全衝突周波数

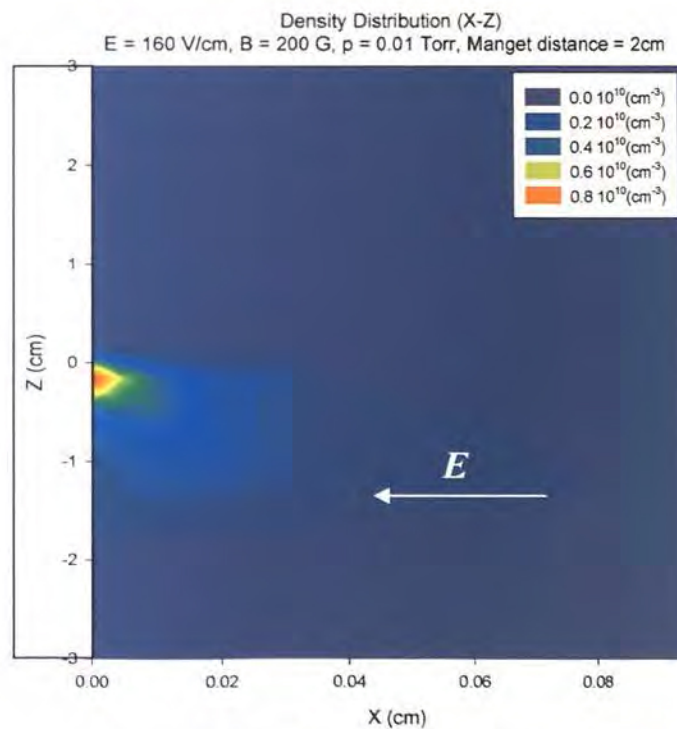


(c) 全衝突周波数 (中心から+3mm、-3mm から出発)

図 4-29 2つの場所から電子を供給した場合の全衝突周波数

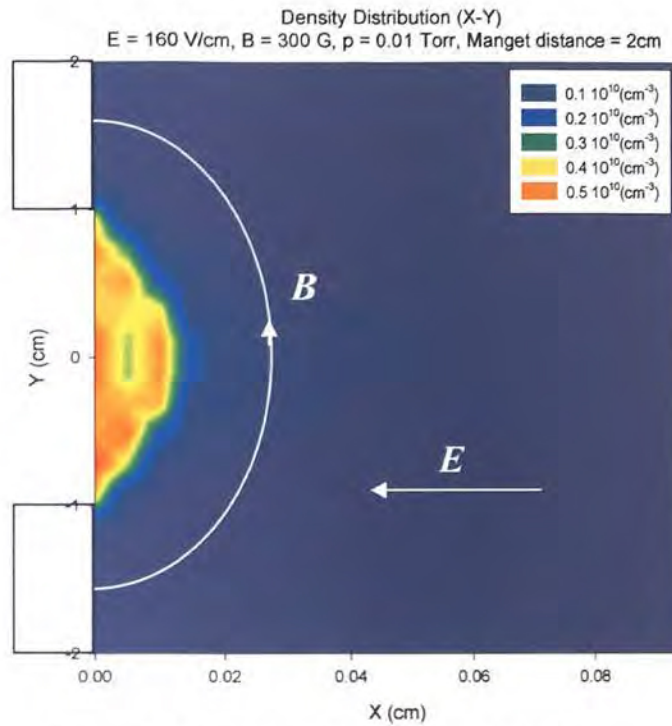


(a) XY 平面の密度分布 ( $B=200\text{G}$ )

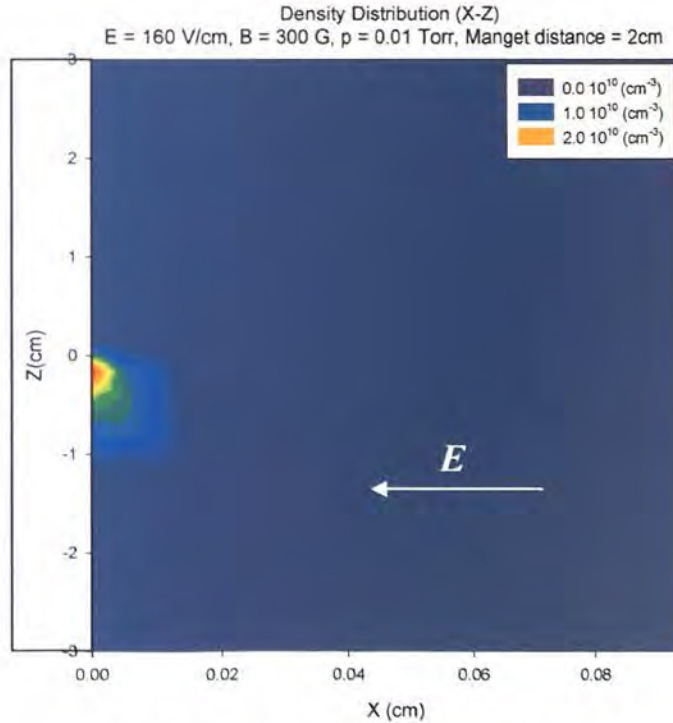


(b) XZ 平面の密度分布 ( $B=200\text{G}$ )

図 4-30 磁極間一様に電子を供給した場合の密度分布

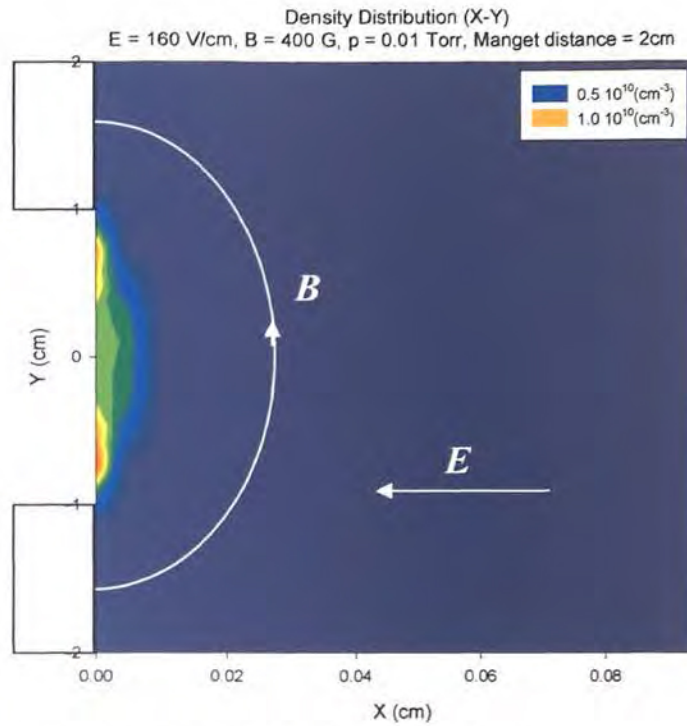


(c) XY 平面の密度分布 ( $B=300\text{G}$ )

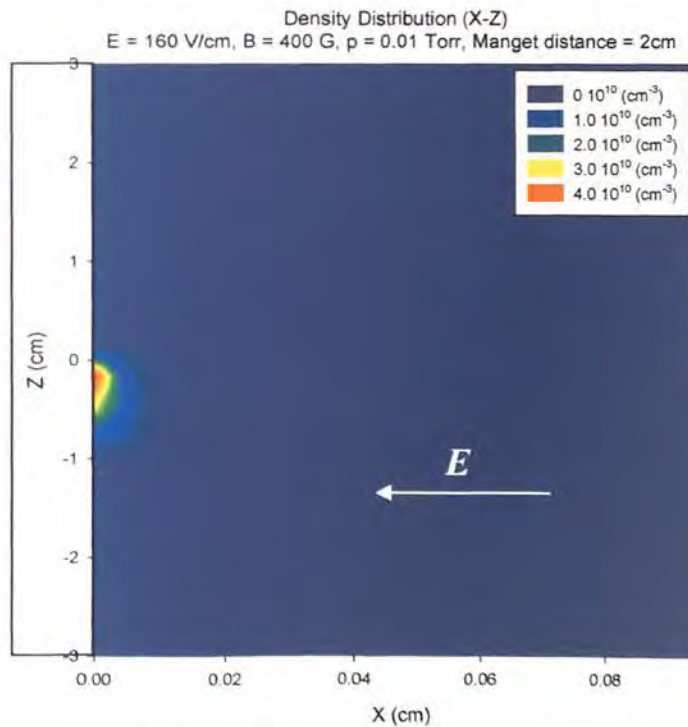


(d) XZ 平面の密度分布 ( $B=300\text{G}$ )

図 4-30 磁極間一様に電子を供給した場合の密度分布



(e) XY 平面の密度分布 ( $B=400\text{G}$ )



(f) XZ 平面の密度分布 ( $B=400\text{G}$ )

図 4-30 磁極間一様に電子を供給した場合の密度分布

#### 4-4 並列計算結果および考察

##### 4-4-1 並列計算結果

LAN ネットワーク上に JavaRMI により分散環境を構築し、最大 30 台の PC を用いた本シミュレーションの PC の構成は、PC に Fujitsu 社 Pentium III 600MHz、メモリ 256MByte、ネットワーク 100Base/TX を用いた。図 4-31 はモンテカルロ・シミュレーションにおける試行回数、すなわち粒子数を 30,000 と 60,000 で並列計算を行った結果およびアムダールの法則による並列化率との比較を行ったものである。10 台程度の PC では速度にほとんど変化が見られない。これは 1 台の PC 当たりの試行粒子数が 3,000 および 6,000 であったため、式(3-8)が成り立ったと考えられる。20 台以上では変化が見られ 30 台においては大きな違いがわかる。30 台の場合、それぞれの PC での試行粒子数は 1,000 および 2,000 となる。すなわち式(3-8)が満たされなくなりつつあると考えられる。これは、1 衝突当たりの粒子の処理にかかる計算時間を一定と考えた場合、それぞれの PC において粒子の衝突回数が統計変動を起こし、均一性を損なっているためであると考えられる。10 台程度の PC においては、十分な粒子数を試行しているため各 PC における平均の衝突回数は全体の平均衝突回数とほぼ等しくなり式(3-8)が満たされているのである。

一方、式(3-1)に示したアムダールの法則との比較からこの結果に対して評価を行うと、粒子数 60,000 の場合、並列化率 99.5%とほぼ等しく、粒子数 30,000 の場合、並列化率 99.0%とほぼ等しいことがわかる。すなわち、並列化率が 0.01%程度低くなることにより、全体の処理性能が台数を増すことにより大きく低下してしまうことがわかる。このまま 200 台までの計算を行うとすると、アムダールの法則から、粒子数 30,000 の場合およそ 67 倍、粒子数 60,000 の場合およそ 100 倍と予測され、99.5%の並列化率でも PC 数により台数効果が得られないと考えられる。また、並列化率が 98.5%の場合においては、50 倍程度となり、これは並列化率 99.5%の場合のおよそ 67 台の PC に相当する。多少の並列化率の低下がこれほどまでに全体の処理速度に影響

を与えることを考えた場合、本論文における高効率並列計算手法は、非常に有効であると考えられる。並列化計算を行う上で問題となるのは、このように並列化を行う PC 間に待ちが発生しないことが大きく影響を及ぼすだけではなく、一般的には通信時間もその要因となる。しかし、プラズマプロセスのモンテカルロ・シミュレーションでは膨大な計算時間を必要とするため、30 台程度の PC においても計算時間に 1 時間を要しており、このことから計算時間に対し通信時間が無視できるほどであるため、高効率な並列計算が行えたと考えられる。

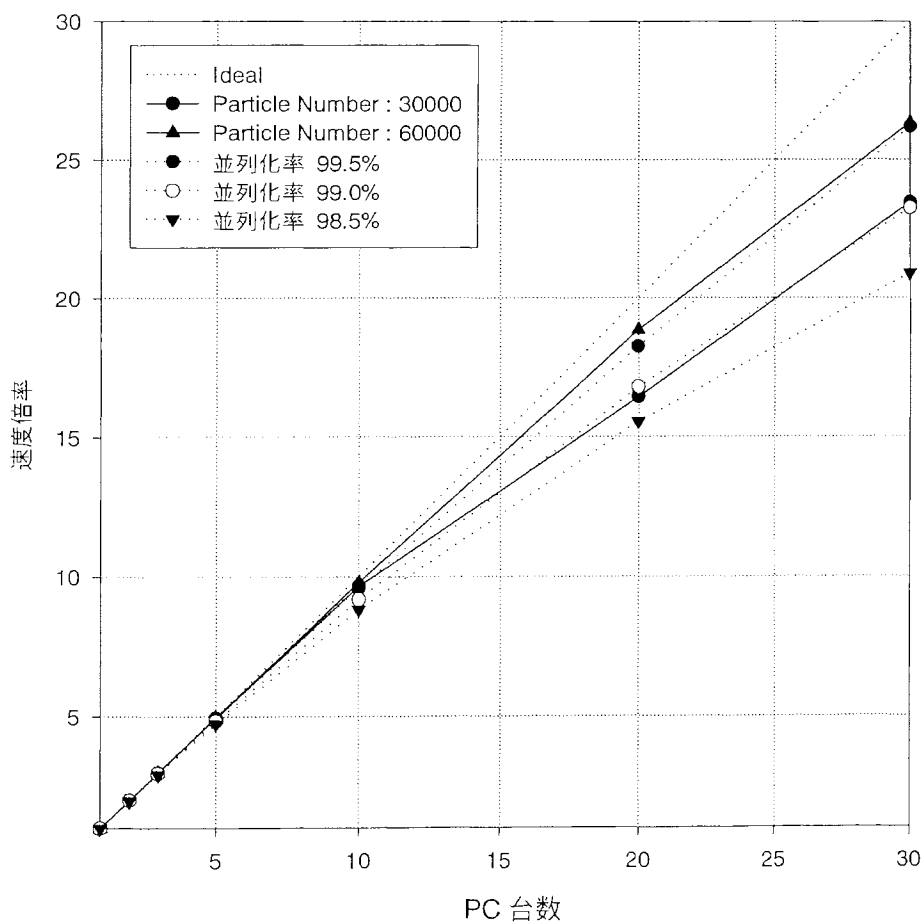


図 4-31 並列計算速度倍率



#### 4-4-2 高効率並列計算理論との比較

表 4-1(a)は 1 台の PC で計算を行った結果であり、試行粒子数を増やすに従い  $T_{cal}$  が上昇している。表 4-1(b)は 10 台の PC においてそれぞれ試行粒子数を 500、(c)は 10 台の PC でそれぞれ試行粒子数 1000、(d) は 10 台の PC でそれぞれ試行粒子数 3000、(e) は 10 台の PC でそれぞれ試行粒子数 6000 として計算を行った結果である。1 台の PC に割り当てる試行粒子数を増やすに従い、 $t_l$  および粒子の衝突回数が平均値に近づいている。特に表 4-1(e) においては、 $t_l$  が一致しており平均の衝突回数にもばらつきが少ないため、高効率で並列化計算が行われていたと予測される。また、シミュレーションの終了時刻のばらつきは(b)~(e)にさほど変わらないが、実際行ったシミュレーション時間との割合から考えると、(e)が最も効率の良い結果であった。これは、式(3-7)より得られる値が近いためであるが、その他の要因としては、粒子数を増加させても同期回数および集計回数は変化しておらず、また、シミュレーション時間に対しネットワーク上での RMI 処理時間が占める割合が非常に小さくなったためであると考えられる。以上のことより、モンテカルロ法によるシミュレーションにおいて、個々の PC が長時間独立に計算できるモデルを設計することで高効率な並列化計算が行えることが示された。

表 4-1 並列化計算効率の実測値

|                                       |        |        |         |         |
|---------------------------------------|--------|--------|---------|---------|
| Number of Particles : $i$             | 5000   | 10000  | 30000   | 60000   |
| Average of number of collisions : $N$ | 537.9  | 537.8  | 536.3   | 536.4   |
| Simulation time(ms) : $T_{col}$       | 262344 | 522892 | 1560203 | 3121890 |

(a)

Particles  $P_1 \sim P_{5000}$  divide into 10 CPU sets of 500 particles

|  | $P_1 \sim P_{500}$ | $P_{501} \sim P_{1000}$ | $P_{1001} \sim P_{1500}$ | $P_{1501} \sim P_{2000}$ | $P_{2001} \sim P_{2500}$ | $P_{2501} \sim P_{3000}$ | $P_{3001} \sim P_{3500}$ | $P_{3501} \sim P_{4000}$ | $P_{4001} \sim P_{4500}$ | $P_{4501} \sim P_{5000}$ |
|--|--------------------|-------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| Number of Particles : $k$              | 500                | 500                     | 500                      | 500                      | 500                      | 500                      | 500                      | 500                      | 500                      | 500                      |
| Number of collisions : $N$             | 268178             | 268779                  | 273235                   | 278038                   | 261768                   | 270688                   | 267326                   | 268069                   | 265335                   | 267982                   |
| CPU time : $T_{col,j}$                 | 26672              | 26578                   | 26703                    | 27032                    | 25390                    | 26282                    | 25968                    | 25985                    | 25750                    | 25984                    |
| CPU time per collisions : $t_f$        | 0.0995             | 0.0989                  | 0.0977                   | 0.0972                   | 0.0970                   | 0.0971                   | 0.0971                   | 0.0971                   | 0.0969                   | 0.0970                   |
| Average of number of collisions : $N'$ | 536.4              | 537.6                   | 546.5                    | 556.1                    | 523.5                    | 541.4                    | 534.7                    | 536.1                    | 530.7                    | 536.0                    |

(b)

Particles  $P_1 \sim P_{10000}$  divide into 10 CPU sets of 1000 particles

|  | $P_1 \sim P_{1000}$ | $P_{1001} \sim P_{2000}$ | $P_{2001} \sim P_{3000}$ | $P_{3001} \sim P_{4000}$ | $P_{4001} \sim P_{5000}$ | $P_{5001} \sim P_{6000}$ | $P_{6001} \sim P_{7000}$ | $P_{7001} \sim P_{8000}$ | $P_{8001} \sim P_{9000}$ | $P_{9001} \sim P_{10000}$ |
|--|---------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|---------------------------|
| Number of Particles : $k$              | 1000                | 1000                     | 1000                     | 1000                     | 1000                     | 1000                     | 1000                     | 1000                     | 1000                     | 1000                      |
| Number of collisions : $N$             | 537303              | 551693                   | 533601                   | 535108                   | 534218                   | 537198                   | 539576                   | 535954                   | 538582                   | 534322                    |
| CPU time : $T_{col,j}$                 | 52328               | 53703                    | 51813                    | 52047                    | 51890                    | 52188                    | 52453                    | 52094                    | 52391                    | 51985                     |
| CPU time per collisions : $t_f$        | 0.0974              | 0.0973                   | 0.0971                   | 0.0973                   | 0.0971                   | 0.0971                   | 0.0972                   | 0.0972                   | 0.0973                   | 0.0973                    |
| Average of number of collisions : $N'$ | 537.3               | 551.7                    | 533.6                    | 535.1                    | 534.2                    | 537.2                    | 539.6                    | 536.0                    | 538.6                    | 534.3                     |

(c)

Particles  $P_1 \sim P_{30000}$  divide into 10 CPU sets of 3000 particles

|  | $P_1 \sim P_{3000}$ | $P_{3001} \sim P_{6000}$ | $P_{6001} \sim P_{9000}$ | $P_{9001} \sim P_{12000}$ | $P_{12001} \sim P_{15000}$ | $P_{15001} \sim P_{18000}$ | $P_{18001} \sim P_{21000}$ | $P_{21001} \sim P_{24000}$ | $P_{24001} \sim P_{27000}$ | $P_{27001} \sim P_{30000}$ |
|--|---------------------|--------------------------|--------------------------|---------------------------|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|
| Number of Particles : $k$              | 3000                | 3000                     | 3000                     | 3000                      | 3000                       | 3000                       | 3000                       | 3000                       | 3000                       | 3000                       |
| Number of collisions : $N$             | 1623494             | 1607939                  | 1614908                  | 1607094                   | 1629980                    | 1601536                    | 1616905                    | 1589285                    | 1605445                    | 1593051                    |
| CPU time : $T_{col,j}$                 | 157625              | 156000                   | 156625                   | 156000                    | 158016                     | 155078                     | 156734                     | 153984                     | 155750                     | 154391                     |
| CPU time per collisions : $t_f$        | 0.0971              | 0.0970                   | 0.0970                   | 0.0971                    | 0.0969                     | 0.0968                     | 0.0969                     | 0.0969                     | 0.0970                     | 0.0969                     |
| Average of number of collisions : $N'$ | 541.2               | 536.0                    | 538.3                    | 535.7                     | 543.3                      | 533.8                      | 539.0                      | 529.8                      | 535.1                      | 531.0                      |

(d)

Particles  $P_1 \sim P_{60000}$  divide into 10 CPU sets of 6000 particles

|  | $P_1 \sim P_{6000}$ | $P_{6001} \sim P_{12000}$ | $P_{12001} \sim P_{18000}$ | $P_{18001} \sim P_{24000}$ | $P_{24001} \sim P_{30000}$ | $P_{30001} \sim P_{36000}$ | $P_{36001} \sim P_{42000}$ | $P_{42001} \sim P_{48000}$ | $P_{48001} \sim P_{54000}$ | $P_{54001} \sim P_{60000}$ |
|--|---------------------|---------------------------|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|
| Number of Particles : $k$              | 6000                | 6000                      | 6000                       | 6000                       | 6000                       | 6000                       | 6000                       | 6000                       | 6000                       | 6000                       |
| Number of collisions : $N$             | 3232032             | 3222265                   | 3232142                    | 3206924                    | 3199429                    | 3209169                    | 3221065                    | 3231428                    | 3198455                    | 3230115                    |
| CPU time : $T_{col,j}$                 | 313640              | 312485                    | 313406                     | 311109                     | 310485                     | 311187                     | 312437                     | 313625                     | 310110                     | 313406                     |
| CPU time per collisions : $t_f$        | 0.0970              | 0.0970                    | 0.0970                     | 0.0970                     | 0.0970                     | 0.0970                     | 0.0970                     | 0.0971                     | 0.0970                     | 0.0970                     |
| Average of number of collisions : $N'$ | 538.7               | 537.0                     | 538.7                      | 534.5                      | 533.2                      | 534.9                      | 536.8                      | 538.6                      | 533.1                      | 538.4                      |

(e)

## 第 5 章 結論

本論文は、モンテカルロ法のシミュレーションに対し、高効率な並列計算を行うためのシミュレーション環境および計算手法を提案するものであり、電磁界を考慮したプラズマ・プロセスの分野に適用し、高効率並列計算の理論およびアムダールの法則より、高効率性について検証を行った。

シミュレーションの開発において、オブジェクト指向設計として UML を用いたモデリングから Java 言語による開発を行った。RMI を利用することにより、通信処理に関するデータ伝送の仕組み、ネットワークを越えた処理の呼び出しに関する手続きを隠蔽して考えることが可能となった。また、マルチスレッドの機能により、同時実行制御、同期制御を容易に行うことが可能となった。すなわち Java 言語に含まれる、RMI およびマルチスレッドの機能を利用することにより、シミュレーションの並列計算を行うための分散環境の構築が可能となった。また、オブジェクト指向設計により開発を行う場合、オブジェクト単位でデータおよび機能が明確化できるため、開発効率が向上するといわれているが、本論文においては、並列計算モデルと物理モデルを分けて考えることができ、また、物理モデル内においては、Ar、SiH<sub>4</sub>、Cl<sub>2</sub> ガスを利用する際に容易に変更が行えシミュレーションを行うことが可能となった。今後のモデルの変更として、電界や境界条件の変更およびサンプリング手法の拡張に対しても柔軟に対応でき、シミュレーションの開発に対し、オブジェクト指向開発が有効であると考えられる。

本論文で提案する高効率並列計算手法を、電磁界を考慮したプラズマ・プロセス・シミュレーションに適用した結果として、30 台の PC において粒子数 60,000 では約 26 倍かつ並列化率 99.5%、粒子数 30,000 では約 23 倍かつ並列化率 99.0%の結果を得たため、モンテカルロ法によるシミュレーションに対し十分に実用的な並列計算手法であると考えられる。

本論文の第 1 章は序論であり、スーパーコンピュータおよび分散環境上に配置した並列計算および同期実行制御に関する従来の研究について述べると

共に、その並列計算性能について述べた。また、種々のシミュレーション技法およびガス分子を用いた、磁界を考慮したプラズマ・プロセス・シミュレーションに関する従来の研究について述べた。

第2章においては、モンテカルロ法によるプラズマ・プロセス・シミュレーションの方法について述べ、本シミュレーションモデルの説明、サンプリング方法、電子の飛行時間の決定方法、Null Collision法を用いた衝突判定方法および電子の散乱に関する計算として軌道の計算方法について述べた。

第3章においては、並列計算手法について述べ、本手法では、モンテカルロ法によるシミュレーションにおいて粒子の挙動が独立している点に着目し、粒子に関する計算を分散環境上のPCに分配しその計算結果を集計することによりシミュレーションを行うことを述べた。具体的な方法として、Java言語によるマルチスレッドおよびRMIを用いた分散環境の構築方法、および、分散環境に配置された各PCの同時実行制御および同期制御方法について述べ、アムダールの法則に従う一般的な並列計算効率および本手法による高効率な並列計算を実現する理論について述べた。

第4章においては、本手法による並列計算の信頼性を検証するために電磁界での荷電粒子の運動を取り上げた。磁界を考慮したプラズマ現象を明らかにするために必要とされる電界と磁界が任意の角度で交差する場でのシミュレーションを、Ar、SiH<sub>4</sub>、Cl<sub>2</sub>ガスについて解いており、シミュレーション結果となる荷電粒子の輸送パラメータとして、0~90度に至るまでの $x$ 、 $y$ 、 $z$ 軸方向の平均速度、平均エネルギー、平均衝突周波数およびLPWS法を用い、エネルギー分布について考察した。また、より現実的なシミュレーションとして、位置により電磁界が複雑に交差するカスプ磁場でのシミュレーションを行いその結果について報告を行った。最後に、本論文の目的となる並列計算結果および高効率並列計算の理論値との比較を行った。最大30台のPCにて、並列計算効率99.5%に到達し、各PCに対し試行回数2000程度でも計算時間に大きく依存する粒子の衝突回数が平均値に近づくことを示し、また、ネットワーク上での通信時間も無視できるほどであることから、高効率な並

列計算が実現できたことを示した。よって、プラズマ・プロセスのモンテカルロ・シミュレーションにおいて、より信頼性の高い解を得るためには多くの試行回数を必要とするが、本手法を用いることで高効率に多くの試行が行え、短時間でも信頼性の高い結果が得られることを示した。

## 参考文献

- [1] 寒川, 三重, *T. IEE Japan*, Vol.118-A, No.9, (1998) pp.916-923.
- [2] 飯塚, 佐藤, *T. IEE Japan*, Vol.118-A, No.9, (1998) pp.971-978.
- [3] 林, 堀口: 固定分配セルを用いた動的負荷分散法による並列分子動力学法シミュレーション, 情報処理学会論文誌, Vol. 40, No. 5, (1999) pp.2152-2162.
- [4] 蒲池, P. Thierry: 並列 CORBA におけるデータ分散管理手法, 情報処理学会論文誌, Vol. 43, No. 4, (2002) pp.1005-1017.
- [5] 蔡, 安室, 李, 陸, 肖: LinuxPC クラスタを用いた並列粒子シミュレーション, 情報処理学会論文誌: ハイパフォーマンスコンピューティングシステム, Vol. 42, No. SIG 12(HPS 4), (2001) pp.124-131.
- [6] 高木, 松岡, 中田, 関口, 佐藤, 長嶋: Java による大域的並列計算環境 Ninplet, 情報処理学会論文誌, Vol. 40, No. 5, (1999) pp.2203-2214.
- [7] I. Foster, C. Kesselman, J. Nick, S. Tuecke: The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration, Open Grid Service Infrastructure WG, Global Grid Forum, June 22, 2002, <http://www.globus.org/research/papers.html>
- [8] A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, S. Tuecke: The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets. *Journal of Network and Computer Applications*, 23(2001) pp.187-200.
- [9] E. Kawamura, C. K. Birdsall and V. Vahedi: *Plasma Sources Sci. Technol.* 9 (2000) pp.413.
- [10] 菅谷, 阿曾: 階層的クラスタリングを用いた非均一環境におけるタスク割り当て手法, 情報処理学会論文誌, Vol. 43, No. 7, (2002) pp.2340-2349.
- [11] 堀, 石川, 小中, 前田, 友清: 時分割空間分割スケジューリング, 情報処理学会論文誌, Vol. 37, No. 7, (1996) pp.1320-1331.

- [12] 堀, 手塚, 石川: ギャングスケジューリングの高速化技法の提案, 情報処理学会論文誌, Vol. 40, No. 5, (1999) pp.2072-2083.
- [13] S. Nakamura, P. L. G. Ventzek and K. Kitamori: J. Appl. Phys. 85 (1999) pp.2534-2539.
- [14] R.D.White, K.F.Ness and R.E.Robson: Velocity distribution functions for electron swarms in methane in electric and magnetic fields, J. Phys. D: Appl. Phys. 32(1999) pp.1842-1850.
- [15] R.D.White, M.J Brennan and K.F.Ness: Benchmark simulations for electron swarms in crossed electric and magnetic fields, J. Phys. D: Appl. Phys. 30(1997) pp.810-816.
- [16] K. Ness and T. Makabe: Electron transport in argon in crossed electric and magnetic fields, Phys. Rev. E 62 (2000) pp.4083.
- [17] K. Kondo, H. Kuroda and T. Makabe: A Study of the Sustaining Mechanism in an Inductively Coupled Plasma, Jpn. J. Appl. Phys. 33 (1994) pp.4254.
- [18] K. Kondo, H. Kuroda and T. Makabe: Spatiotemporal characteristics determined by a relaxation continuum model of an inductively coupled plasma, Appl. Phys. Lett. 65(1994) pp.31.
- [19] M. Matsumoto and T. Nishimura: ACM Trans. on Modeling and Computer Simulation. 8(1998) pp.3.
- [20] P. L. G. Ventzek and K. Kitamori: J. Appl. Phys. 75 (1994) pp.3785.
- [21] W. Grosso: *Java RMI*, (O'Reilly & Associates, Inc, Network, 2001).
- [22] D. Lea: *Concurrent Programming in Java*, (Addison-Wesley, Boston, 1999).
- [23] 河米, 中村, 大野, 飯島: 分散オブジェクトコンピューティング, (共立出版, 1999).
- [24] 田上, 坂上, 高橋: 粒子コードの並列化における通信の隠ぺい, 情報処理学会論文誌, ハイパフォーマンスコンピューティング, 82-32, (2000) pp.185-190.

- [25] S. Ido, M. Kashiwagi and M Takahashi, *Jpn. J. Appl. Phys.* Vol.38, (1999) pp.4450-4454.
- [26] D. S. Lemons and D. L. Kaufman, *IEEE Trans. Plasma Science*, Vol.27, No.5, (1999) pp.1287-1296.
- [27] Chae Hwa Shon, Jun Seong Park, Bong Koo Kang and Jae Koo, Lee, *Jpn. J. Appl. Phys.* Vol.38, (1999) pp.4440-4449.
- [28] T. M. Minea, J. Bretagne and G. Gousset, *IEEE Trans. Plasma Science*, Vol.27, No. 1, (1999) pp.94-95.
- [29] N. Shimura and T. Makabe, *Appl. Phys. Lett.*, 62(7),(1993) pp.678-680.
- [30] M. S. Dincer, *J. Phys. D: Appl. Phys.* 26, (1993) pp.1427-1431.
- [31] M. S. Dincer and A. Gokmen, *J. Phys. D: Appl. Phys.*, 25, (1992) pp.942-944.
- [32] G. R. Govinda Raju, M. S. Dincer, *IEEE Trans. Plasma Science*, Vol. 18, No. 5, (1990) .
- [33] M. J. Brennan, A. M. Garvie and L. J. Kelly, *Aust. J. Phys.*, 43, (1990) pp.27-43.
- [34] A. E. D. Heylen, *IEE Proc*, Vol.127, Pt. A, No.4, (1980) pp.221-244.
- [35] G. R. G. Raju and G. R. Gurumurthy, *INT. J. Electronics*, Vol.44, No.4, (1978) pp.355-365.
- [36] A. E. D. Heylen and C. L. Dargan, *INT. J. Electronics*, Vol.35, No.4, (1973) pp.433-451.
- [37] H. Sugawara, T. Yahata, A. Oda and Y. Sakai, *J. Phys.D: Appl. Phys.*, 33,(2000) pp.1191-1196.
- [38] R. D. White, R. E. Robson and K. F. Ness, *IEEE Trans. Plasma Science*, Vol.27, No.65, (1999) pp.1249-1253.
- [39] R. D. White, K. F. Ness and R. E. Robson, *J. Phys. D: Appl. Phys.*, 32, (1999) pp.1842-1850.



## 謝 辞

本論文を作成するにあたり、多大なご指導ご検討を賜りました北海道工業大学工学部丸山晃市教授に心から感謝の意を表します。長年にわたり、懇切丁寧なご指導ご検討を賜りました北海道工業大学工学部北守一隆教授に心から感謝の意を表します。長年にわたり、懇切丁寧なご指導ご検討を賜りました米国モトローラ P.L.G.Ventzek 博士に心から感謝の意を表します。また、機会あるごとに貴重なご意見、ご助言を頂きました、札幌大学経済学部大森義行教授、北海道職業能力開発大学校中村聡先生に心から感謝致します。また、本研究に貴重なご助言を頂きました、北海道工業大学大学院特別研究生堀江育也氏に心から感謝致します。並列計算の実験において大変な協力を頂いた、北海道工業大学大学院宮崎洋平君に心から感謝致します。