

イオンエンジンにおける昇華性推進剤のイオン計測 に向けたトムソンパラボラの開発

白木, 僚

九州大学大学院総合理工学府総合理工学専攻プラズマ・量子理工学メジャー

<https://hdl.handle.net/2324/6792854>

出版情報 : Kyushu University, 2022, 修士, 修士
バージョン :
権利関係 :



令和 4 年度
九州大学大学院 総合理工学府
総合理工学理工学専攻
プラズマ・量子理工学メジャー

修 士 論 文

論文名

イオンエンジンにおける昇華性推進剤のイオン計測に向けた
トムソンパラボラの開発

氏 名

白木 僚

指導教員名

山本 直嗣 教授

目次

第1章 序論.....	1
1-1 研究背景.....	1
1-2 代替推進剤.....	2
1-2-1 これまで検討された推進剤.....	3
1-2-2 昇華性推進剤.....	3
1-3 研究目的.....	4
第2章 イオンエンジンの概要.....	6
2-1 電気推進の種類.....	6
2-2 イオンエンジンの作動原理.....	7
2-3 性能の評価方法.....	9
第3章 トムソンパラボラの設計.....	11
3-1 解離度の測定方法.....	11
3-2 トムソンパラボラの原理.....	13
3-2 ピンホールによる角度の推定.....	16
3-3 磁場領域の設計.....	17
3-4 電場領域の設計.....	19
3-5 検出部.....	20
3-6 検出器での放物線軌道の計算.....	21
第4章 トムソンパラボラによる計測.....	23
4-1 共通の実験体系.....	23
4-2 レンズのピント調整およびゼロ点補正.....	25
4-3 アルゴンおよび窒素での計測.....	26
4-3-1 使用したイオンエンジン.....	26
4-3-2 実験条件.....	27
4-3-3 DCC1545M での実験結果.....	28
4-3-4 ASI533MM Pro での実験結果.....	29
(1)アルゴン.....	30
(2)窒素.....	32
4-4 昇華性推進剤での計測.....	36
4-4-1 使用したイオンエンジン.....	36
4-4-2 流量制御体系.....	37
4-4-4 実験条件.....	40
4-4-5 実験結果.....	41
第5章 結論.....	43

参考文献	44
謝辞.....	46
付録.....	47
ヘッダファイル.....	47
ソースファイル.....	50

第 1 章 序論

1-1 研究背景

2014 年に, JAXA (Japan Aerospace Exploration Agency) が, イオンエンジンを主推進器として搭載した小惑星探査機「はやぶさ 2」を打ち上げ, 2019 年に小惑星「リュウグウ」にタッチダウンし, 2020 年 12 月にカプセルの回収を成功させた.¹ この偉業は日本だけでなく世界からも注目を浴び, 今後のさらなる宇宙開発が期待される.

現在, JAXA では, 宇宙太陽光発電システム (SSPS)² の実証に向けた研究が行われている. SSPS は静止軌道上に太陽光電池とマイクロ波送電アンテナを設置し太陽から得たエネルギーを電気に変換, さらにマイクロ波に変換し, 地球上に設置した受信アンテナに送電し電力に再変換する構想である. SSPS は, 昼夜を問わず発電することが可能で, 地上に比べ 1.4 倍太陽光を有効に使うことが可能である. 図 1.1 に SSPS の概要図を示す. SSPS の建造には約 1 万トンと大量の物資を静止軌道に輸送する必要がある. そのため, 物資を低軌道上まで化学推進によって運び, 低軌道から静止軌道上まで比推力の高い電気推進によって輸送する軌道間輸送が提唱されている.³ 図 1.2 は, 軌道間輸送を用いた SSPS 建造物資の輸送の概念図である. 比推力の高い電気推進ではあるが, SSPS の建造には推進剤が数千トン必要となると考えられている. しかし, これは年間の生産量 50 トンを上回るものであり, キセノンに代わる推進剤が必要とされる.

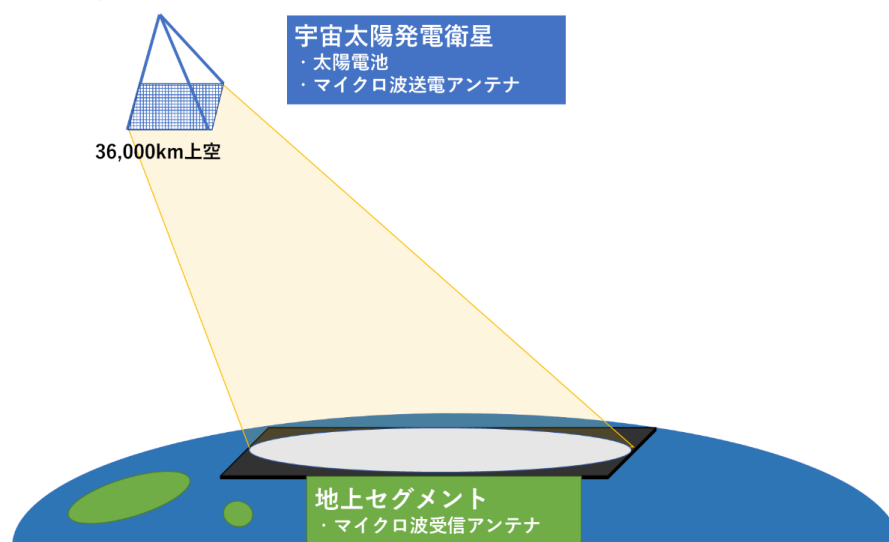


図 1.1 SSPS の概略図

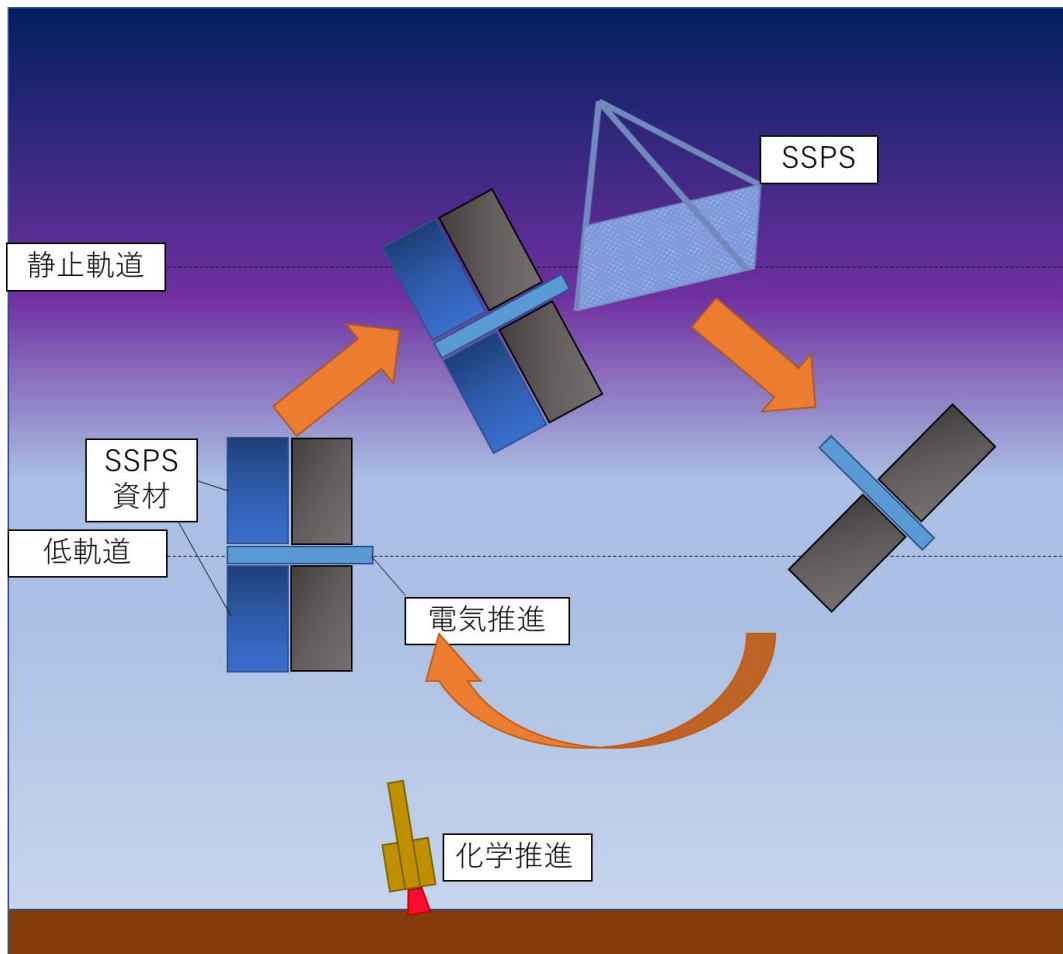


図 1.2 軌道間輸送を用いた SSPS 建造物資の輸送の概念図

1-2 代替推進剤

現在, 主に使用される推進剤はキセノンである. キセノンは, 希ガスであるため安定で, その中でも電離しやすいという特徴を持つ.⁴ しかし, キセノンは, 空気分離装置の副産物として得られるが, 空気中に 0.087 ppm しか存在しないため, 効率的な生産のためには超大型の空気分離装置が必要である. そのため, キセノンは生産量が少なく, 高価である. また, キセノンは貯蔵に際して高圧タンクが必要であり, 電気推進システムの小型化の障壁となっている. 以上の点からもキセノンの代替推進剤の検討が必要とされる.

代替推進剤には, キセノンと比較して以下のような特徴が求められる. 1) 入手性, 2) 低価格, 3) 取り扱い性, 4) 貯蔵性, 以上 4 つの観点から推進剤を選んだ.

1-2-1 これまで検討された推進剤

キセノンの代替推進剤としては、クリプトン⁵、アルゴン⁶などの希ガス、大気中に多く存在する窒素⁷、マグネシウム⁸、ビスマス⁹、水銀¹⁰などの金属、ヨウ素分子^{11,12}など様々な物質が考えられている。キセノンと同じ希ガスのクリプトンは、キセノンと特性が近いものの、希少性が高い点から課題の解決には至らない。アルゴンはキセノンより比推力が高く大気中に多く含まれるが、電離断面積が小さくイオン性成功率が低いため、キセノンと同じ推力にするには多くの電力が必要である。さらに、分子量が小さいため、タンク係数(推進剤の質量/タンク質量)が大きく貯蔵性の観点からも不十分である。窒素は大気吸込式イオンエンジンとして開発が進められている。しかし、窒素もアルゴンと同様に分子量の小さく推力の低下や必要電力の上昇が見込まれる。金属系推進剤は、固体または液体であるため、高压タンクが不要である。しかし、推進剤として使うには加熱が必要で、特にビスマスは約700℃まで加熱する必要がある。また、水銀は毒性があり、取り扱いが難しい。ヨウ素は固体で昇華性があり、高压タンクが不要であるが、水銀と同様に毒性があり、太陽電池パネルなどの衛星表面を汚染する可能性がある。また、負電荷のイオンが発生し、推力効率が低下する可能性がある。以上より、新たな推進剤の検討が必要とされる。

1-2-2 昇華性推進剤

本研究では、昇華性を有するアダマンタンや樟脳といった有機物質を代替推進剤として用いた。これらの物質は、染料、防虫剤、医薬品などに使われており、キセノンと比較して、安価で入手しやすいという利点がある。昇華性推進剤は、常温常圧で固体であり、蒸気圧が低いため、高压タンクを必要としない。これらの利点により、輸送コストの削減が見込まれる。また、分子量が100~200とキセノンと同等もしくはそれ以上であるため、推力が大幅に小さくなる可能性は低く、推進剤によっては推力の増加も期待される。表1に候補物質の名称、分子式、分子量、25度での蒸気圧、価格をキセノンと合わせて示す。表に示した推進剤はキセノンと比較して価格を10分の1から100分の1に抑えながら、推力を維持もしくは増加させることが可能である。

表 1.1 キセノンと昇華性推進剤の比較

Propellant	Molecular Weight	Price[yen/100g]	Vapor pressure [Pa] @25°C
Xenon	131.29	61827	-
Camphor	152.33	1848	29.5
Adamantane	136.23	5000	21
L-menthol	156.27	3960	8.5
Naphthalene	128.18	540	11
Anthraquinone	208.21	1980	1.55

1-3 研究目的

前節で述べたように、昇華性推進剤はこれまで課題であった推進剤の安定供給、衛星システムの小型化・コストダウンを担うことのできる高いポテンシャルを有するが、課題も存在する。

昇華性推進剤における主な課題として解離による分子量の変化が挙げられる。中野らの研究より昇華性推進剤において推力とビーム電流の関係からビーム電流が増加するに伴い推力が飽和する傾向が見られた。¹³ このことは、スクリーン電流の増加とともにイオンの分子量が減少することを示している。さらに、この分子量の減少は、解離がビーム電流の増加とともに進んでいることを示す。解離による分子量の不明瞭化は、グリッドの最適化を難化させている。解離が進むことは一見するとイオンエンジンの作動において、不利なことのようと思われる、しかし、必ずしも解離が進むことはデメリットばかりではない。イオンエンジンではイオンビームの加速噴射により推力が得られ、次式で表される。

$$F = \dot{m}_i v_z = J_b \sqrt{\frac{2m_i V_b}{e}} \quad (1-1)$$

ここで、 $\dot{m}_i$ は推進剤流量、 J_b はイオンビーム電流、 m_i はイオンの質量、 V_b は引出し

電圧, e は電気素量を表す. さらに, 推進機の燃費を示すパラメータである比推力は次のように定義される.

$$I_{sp} = \frac{F}{\dot{m}g} \quad (1-2)$$

このように比推力 I_{sp} は推進剤流量に対する推力の大小を表す指標であり, 単位は [秒] である. ここで, g は地表面における重力加速度 [m/s^2] である. 以上より解離により分子量の低下が起こると推力が低下するものの比推力は向上する. プラズマの状態を調整することにより, 深宇宙探査など比推力が重要なミッションでは解離が進むようにし, 軌道間輸送のような推力が必要とするミッションでは, 解離を抑制するような運用が可能となる.

以上より, イオンエンジンのグリッド設計と制御の両面から, 解離の度合いを推定することが不可欠である. そのために解離とプラズマ諸量 (プラズマ密度, 温度) の定量的な関係を明らかにし, 昇華性推進剤に最適なプラズマ条件を模索することが目的である. すでに本研究室においてレーザートムソン散乱計測によるプラズマ諸量 (電子温度, 電子密度) の計測手法は確立されている.¹⁴ 解離度の測定に関しては, E×B プローブなどが用いられたが十分な分解能を確保することができなかった.

そこで, 本研究では, トムソンパラボラによる解離度測定を行い, 解離度合いの推定に必要なインプットとアウトプット両方をそろえることを目的とする.

第2章 イオンエンジンの概要

2-1 電気推進の種類

電気推進機は電力を推進剤に与えることで反力を得る推進機構で、真空の宇宙空間で用いられるロケットエンジンシステムの一つである。電気推進で得られる推力は化学推進と比較して著しく小さいが、比推力が非常に大きいことが特徴で、推進剤の消費量を抑えペイロード重量の増加が可能となる。電気推進機は推進剤の加速に用いられる力の種類により分類され、イオンエンジンに代表される静電加速型、ホールスラスタに代表される電磁加速型、アークジェットスラスタに代表される電熱加速型の3つに分類される。¹⁵ ミッションの目的によって最適なエンジンを選定する必要がある。

■ 静電加速型

系が加速する方向に静的な電場を作り、クーロン力によって推進剤を加速するタイプ。イオン加速を空間電位に頼っているため、宇宙機が帯電しないよう中和器を装備する必要がある。

Ex)イオンエンジン,ホールスラスタ

■ 電磁加速型

ローレンツ力を用いているもののほか、電場と系の加速の方向が互いに異なるタイプの推進系は電磁加速型と呼ばれる。プラズマの性質を利用して加速するため、中和器が必要ない。

Ex)ホールスラスタ,MPD アークジェット

■ 電熱加速型

電気エネルギーで推進剤を加熱し、ノズルなどを用いて熱エネルギーを運動エネルギーに変換して推力を得るタイプ。ノズルは固体材料を用いたもののほか、磁場などを制御してノズルの役割を担わせるものもある。

Ex)DC アークジェット,レジストジェット

2-2 イオンエンジンの作動原理

イオンエンジンの概念図を図 2-1 に示す。イオンエンジンは主にプラズマ生成部、イオン加速(静電加速)部、イオンビームの中和部の 3 つの部分で構成されている。プラズマ生成部において推進剤を電離させ、イオン加速(静電加速)部において 2~3 枚のグリッドを用いてイオンだけを加速し、下流にて電子源から放出される電子を混合して電気的中性なプラズマジェットとして噴射し、その反力によって推進力を得る。以下に各領域の詳しい説明を述べる。

I. プラズマ生成部

イオンエンジンにおいてプラズマの生成はイオン源が担う。プラズマの生成方法に違いはあるが、プラズマ生成は次の過程を踏む。

- (1)電子を生成・加速して電離能力を持った高エネルギー一次電子を生成
- (2)電子を損失させないように閉じ込める
- (3)高エネルギーの電子を中性粒子に衝突させてイオン化
- (4)静電加速部に供給

プラズマの生成のためのイオン源は、主に直流放電式イオン源、高周波放電式イオン源、マイクロ波放電式イオン源に分類される。

II. イオン加速(静電加速)部

図 2.2 に、3 枚グリッドシステムとイオン流の概略を示す。イオン加速部は 2~3 枚の多孔電極(グリッド)から構成されており、一般的にグリッドシステムと呼ばれる。上流からスクリーングリッド、アクセルグリッド、ディセルグリッドと称する。ディセルグリッドを省略する場合もある。1mm 以下の間隔で近接設置した 3 枚のグリッドに、スクリーン電源やアクセル電源から、+1kV, -300V, 0V といった電位を与え、図 2-2 に示すような空間電位を形成する。この電位差を利用してイオン源からスクリーングリッドの孔を通じてイオンが引き出される。スクリーングリッドの孔を通過したイオンはスクリーングリッドとアクセルグリッドの間の電界によって加速され、アクセルグリッドとディセルグリッドの間の逆電位勾配によって多少減速し噴射される。アクセルグリッドとディセルグリッドの間の逆電位勾配は下流からの電子の侵入を阻止する役割もある。

Ⅲ. 中和部

中和部の主な役割はイオン源からグリッドを介し加速噴射されたイオン電流に等価な電子電流を空間に噴射し,人工衛星の電気的中性を維持することである.イオンエンジンがイオンのみを噴射すればイオンエンジンを含め宇宙機全体が負に帯電するため,噴射したイオンが再び引き戻され推力が発生しない.代表的な中和器としては,フィラメントカソード, ホローカソード, マイクロ波放電式カソード, 電界放出カソードが挙げられる.

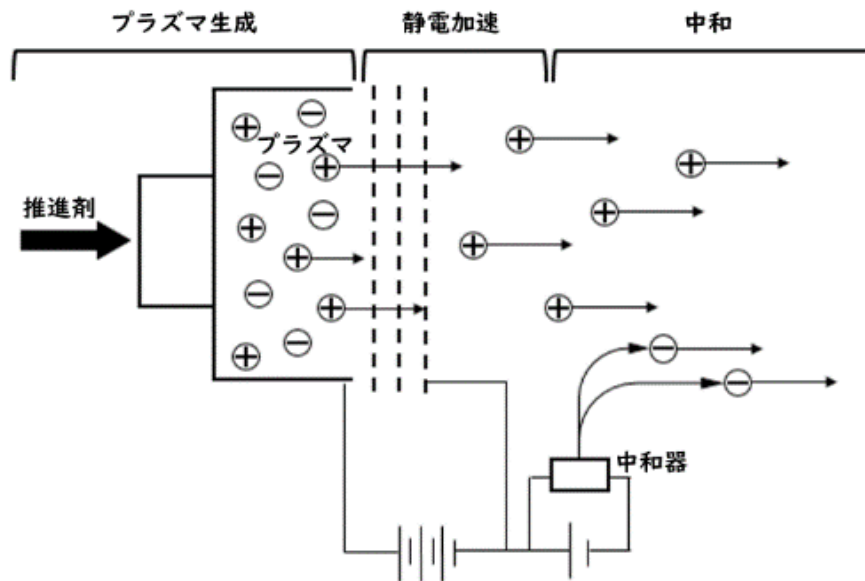


図 2.1 イオンエンジンの作動概略

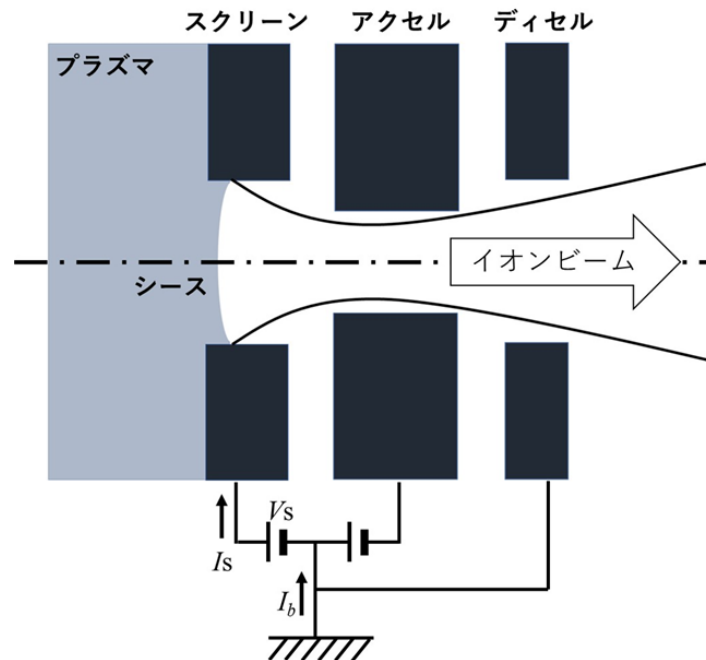


図 2.2 グリッドによる静電加速の概略図

2-3 性能の評価方法

イオンエンジンの性能はイオン引き出し電流や加速電圧、マイクロ波投入電力や推進剤流量から算出し評価することができる。式(2-1)から式(2-6)に、イオンエンジンの性能を評価するための主なパラメータの説明とその定義を示す。

■ イオン生成コスト ε_c

1 個のイオンまたは 1 A のイオン電流を得るために必要なエネルギーであり、単位は[eV]もしくは[W/A]である。直流放電式では、

$$C_i = \frac{I_D V_D}{I_s} \quad (2-1)$$

I_D は放電電源に流れる電流[A]、 V_D は放電電圧[V]、 I_s は静電加速部へのイオン電流すなわちスクリーン電流[A]である。また、マイクロ波放電式のイオン生成コストは、

$$C_i = \frac{P_u}{I_s} \quad (2-2)$$

ここで、 P_u はマイクロ波投入電力[W]である。

■ 推進剤利用効率 η_u

推進剤として供給される全粒子の中で、イオンビームとして排出される推力に直接寄与する粒子の割合であり、

$$\eta_u = \frac{\dot{m}_{ex}}{\dot{m}} \quad (2-3)$$

ここで、 $\dot{m}$ は推進剤流量、推力に直接寄与した推進剤流量である。推力に直接寄与した推進剤は、電流で評価するため推進剤効率は次の式のようにも表される。

$$\eta_u = \frac{I_b}{\dot{m}_{Aeq}} \quad (2-4)$$

ここで、 I_b は噴射されたビーム電流[A]， $\dot{m}_{Aeq}$ は推進剤流量 $\dot{m}$ で流入した全ての粒子が1価のイオンに電離したと仮定して電流値換算した値[A]である．

■ 推進効率 η

投入した電力がどれだけの推力発生に運動エネルギーに変換されたかを表すものであり，次式で与えられる．

$$\eta = \frac{F^2}{2\dot{m}P} \quad (2-5)$$

■ 推力電力比 F/P

推力電力比は、得られた推力と投入電力の比のことであり，次式で表される．

$$\frac{F}{P} = \frac{2\eta}{gI_{sp}} \quad (2-6)$$

第 3 章 トムソンパラボラの設計

3-1 解離度の測定方法

解離度の測定には,イオンを磁場や電場を用いて曲げることにより質量電荷比を計測する四重極型質量分析装置(QMS), $E \times B$ プローブ, トムソンパラボラを用いることができる.これらについて以下に説明する.

1) 四重極型質量分析装置(QMS)

QMS は,試料導入部,イオン化部(イオン源),質量分離部(アナライザー),検出部(検出器),真空排気部(真空ポンプ),装置制御部・データ処理部(データシステム)等からなる.QMS のアナライザーは以下の図 3.1 に示す通り基本的には 4 本の平行なロッド状の電極からなり,相対する電極の極性を同じにして直流電圧と高周波交流電圧を重ね合わせた電圧を印加し,四重極電場を形成している.低い加速電圧で四重極ロッドに沿ってイオンを入射すると,イオンは上下左右 に振動しながら進む.しかし, $2U/V$ を一定に保ちつつ,電圧を変化させることにより,ある瞬間に特定の m/z の値のイオンのみが検出器に達する.その一方で,そのほかの m/z を持つイオンは振幅が大きくなり,電極に衝突する.先行研究では,QMS は中性粒子もイオン化することからイオンエンジンによって加速されたイオン以外の水分やタンクの原料などの物質が検出され解離によってどのイオン種が生成されたのかが不明であった.¹⁶ 以上より, イオンビーム内のイオンのみを計測し, 雰囲気中の中性粒子を計測しない装置の開発が必要である.

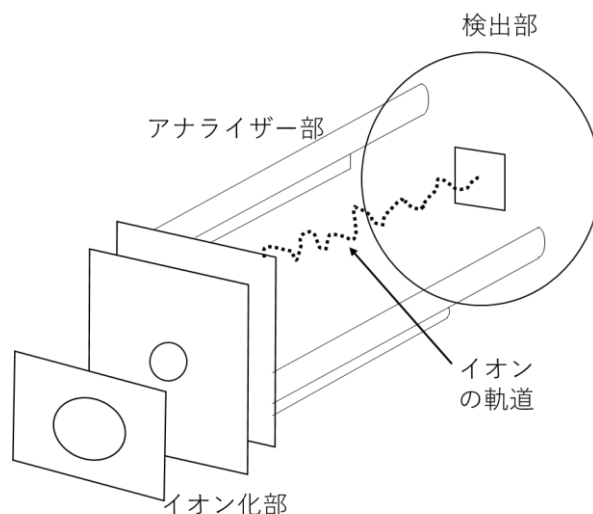


図 3.1 QMS の概念図

2) E×B プローブ

E×B プローブの概念図を図 3.2 に示す。E×B プローブは入射したイオンに電場と磁場をかけることにより分離して計測できる装置である。コリメータ部により単一のベクトルを持ったイオンビームのみが偏向部に入射される。次に、イオンビームは電場と磁場がかかっている偏向部を通過する。図 3.1 の赤の矢印は磁場の正の向きを示す。磁場は一定に保ち、電場は電極間に掛ける電圧を変えることで偏向部を通るイオンが受ける力を変化させる。磁場により受ける力と電場により受ける力が釣り合う時、イオンは力を受けずに直進する。2つの力が釣り合う条件は、

$$\mathbf{E} = \mathbf{v} \times \mathbf{B} \quad (3-1)$$

である。それぞれのイオンのエネルギーが同じとき、イオンの分子量 M とイオンの速度 v には、 $M \propto 1/v^2$ の関係がある。そのため、分子量の小さいイオンが直進するとき、分子量の大きいイオンは速度が大きいため磁場の影響を大きく受け曲がる。つまり、掃引電圧が小さいとき分子量の大きいイオンが検出され、掃引電圧が大きいとき分子量の小さいイオンが検出される。

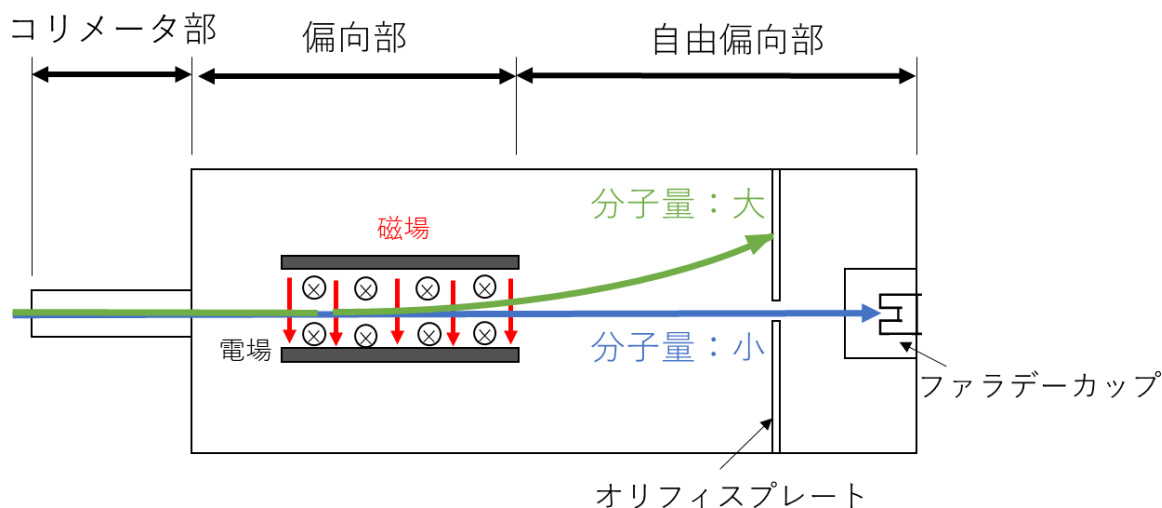


図 3.2 E×B プローブの概念図

中野らが E×B プローブを用いて解離度の計測を行ったところ、分子量の小さいものと大きいものが混在しており、解離が起こっていることが示唆された。¹⁷ さらにイオン種を分け、分子量を測定するには分解能を上

げることが必要とされる．本研究室でも以前，キセノンにおける 2 価イオンの存在比計測のために $E \times B$ プローブの開発が行われた．¹⁸ その際に得られた電流値は pA のオーダーであった．pA のオーダーであれば存在比の小さいイオン種はノイズに埋もれることが予想される．また，様々なイオン種のピークが重なることによりイオン種を見分けることが難しくなる恐れがある．そのため $E \times B$ プローブによる分子量の判別は難しく，分子量の判別ができる装置の開発が必要である．

3) トムソンパラボラ

トムソンパラボラは、静電場と静磁場を用いて入射イオンを質量電荷比とエネルギーによって入射方向から偏向させる働きを持つ．主にトムソンパラボラは、複数種のイオンが排出されるレーザー生成プラズマの研究で用いられている．^{19,20}

本研究では、解離度測定のための分子量が 12-150 の範囲のトムソンパラボラの開発を行った．

3-2 トムソンパラボラの原理

図 3.3 にトムソンパラボラの概略図を示す． x を入射方向， y を磁場による偏向方向， z を電場による偏向方向とする． y 方向にかかる静電場・静磁場中に z 方向からあるエネルギー分布を持つイオンを入射させて、放物線上の軌跡をえることで、エネルギー・質量電荷比 (m/z) を分析するものである．磁場を B ，電場を E ，磁場をかけている領域の長さを L_B ，磁石出口から検出器までの距離を D_B ，電場をかけている領域の長さを L_E ，電極出口から検出器までの距離を D_E ，イオンの質量を m ，電荷を q ，エネルギーを K とすると粒子は検出面に入射軸方向から、

$$y = \frac{qBL_B}{\sqrt{2mK}} \left(D_B + \frac{1}{2}L_B \right) \quad (3-2)$$

$$z = \frac{qEL_E}{2K} \left(D_E + \frac{1}{2}L_E \right) \quad (3-3)$$

また、式(3-2)、式(3-3)より

$$y^2 = \frac{qB^2L_B^2(L_B/2 + D_B)^2}{mEL_E(L_E/2 + D_E)} z = \alpha z \quad (3-4)$$

この式より入射イオンは検出器にて質量電荷比に従う放物線を描くことがわかる.

図 3.4 に今回作成したトムソンパラボラの図を示す. また, 作成したトムソンパラボラの諸元を表 3.1 にまとめた.

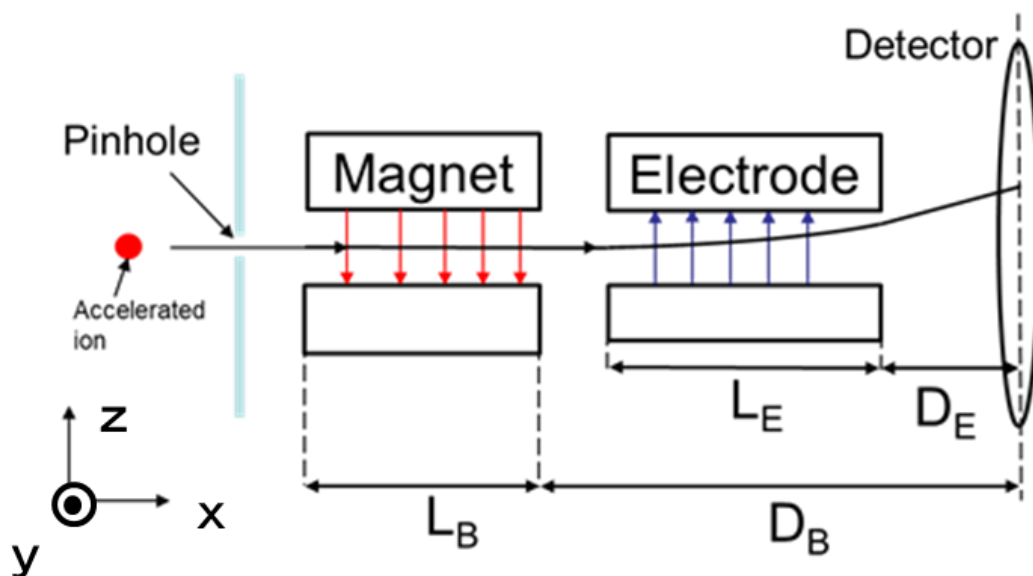
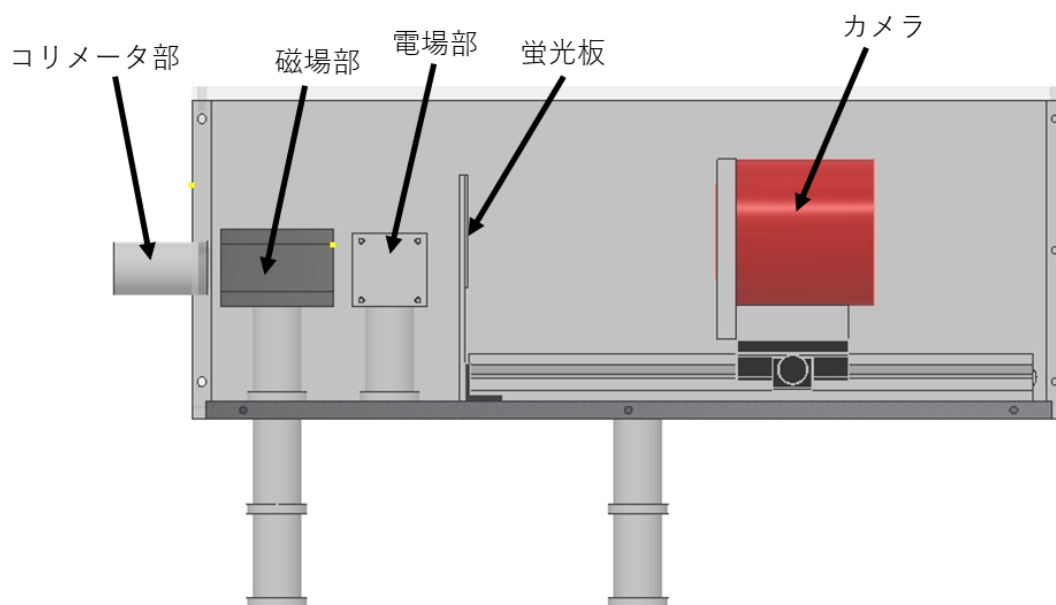
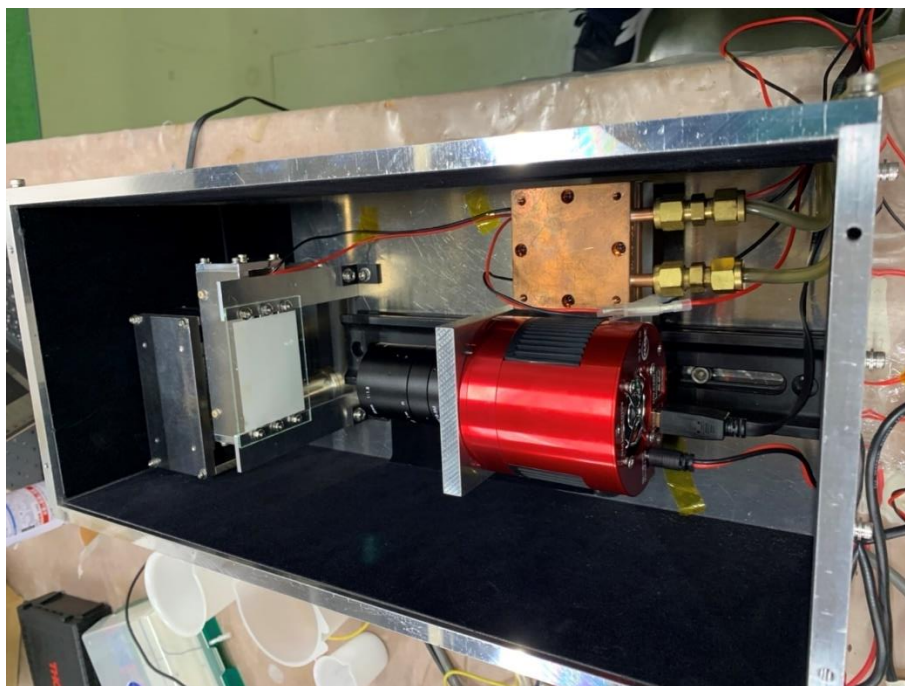


図 3.3 トムソンパラボラの概略図



(a) CAD データ



(b) 上から見た図



(c) 外観

図 3.4 作製したトムソンパラボラ

表 3.1 トムソンパラボラのパラメータ

最大磁場強度	0.301 T
L_B	0.03 m, 0.06 m
D_B	0.07 m
最大電場強度	2000 V/m
L_E	0.04 m
D_E	0.02 m

3-2 ピンホールによる角度の推定

今回作成したコリメータ部は、イオンエンジン側が $100\ \mu\text{m}$ 、トムソンパラボラ側が $150\ \mu\text{m}$ の 2 つのソーラボ製のピンホールをレンズチューブに $22.2\ \text{mm}$ 間隔で設置したものを使用した．このピンホールによる発散角は、 0.32° である．図 3.5 にこの発散角での蛍光スクリーン上での広がり軌道計算によって求めた結果を示す．図 3.5 より同一種、同エネルギーのイオンは蛍光スクリーン上で直径 $1.2\ \text{mm}$ の円として観測されることがわかる．

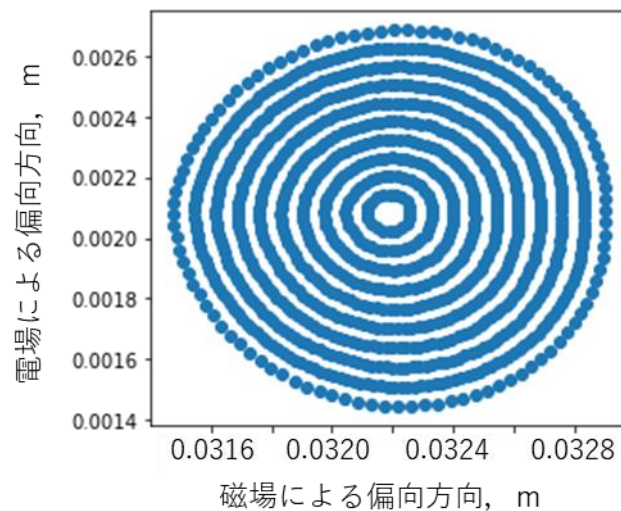


図 3.5 検出面での同一種、同エネルギーのイオンの図

3-3 磁場領域の設計

コリメータ部を通過したイオンは、次に磁場領域を通過する．図 3.6 は今回作成した磁場発生部分の図である．灰色がヨーク (鉄) であり，黒の部分が磁石 (フェライト磁石) である．各磁石の大きさは 30 mm×60 mm，厚さ 10 mm (磁化方向) であり，4 つ使用した．上下の磁石間の距離は 5 mm であり，イオンはその中心を通過する．磁石を 4 つ使用することで，分子量の大きいものは，図 3.3 でいうところの L_B を 60 mm にすることができ，分子量の小さい範囲を測定する際には， L_B が 30 mm となるように磁石を上下に一つずつ配置ことで測定範囲を変えられるようにした．図 3.7 にイオン通過面での磁束密度のシミュレーション結果を示す．イオン通過平面での磁束密度の最大値は，磁石が 2 個では 0.300 T，磁石が 4 個では，0.301 T であった．

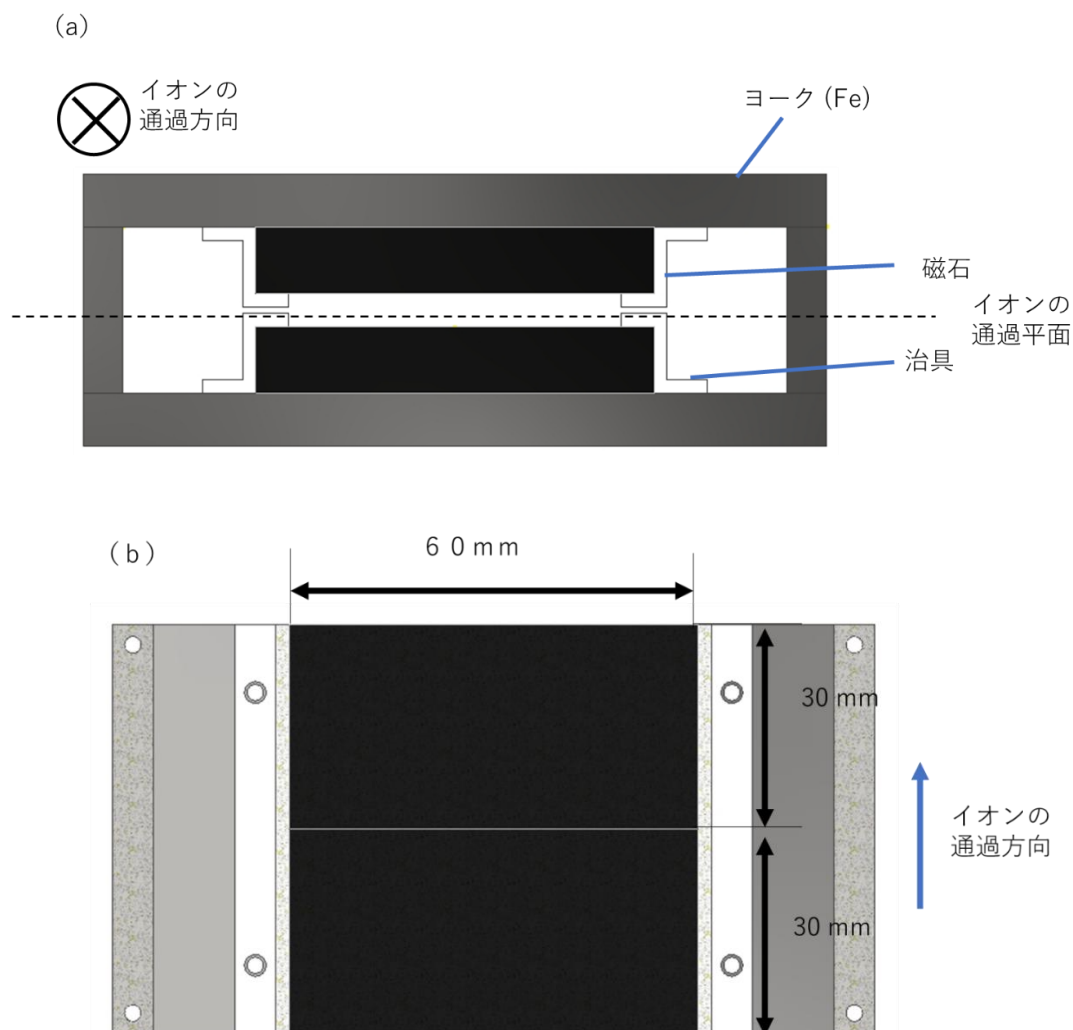
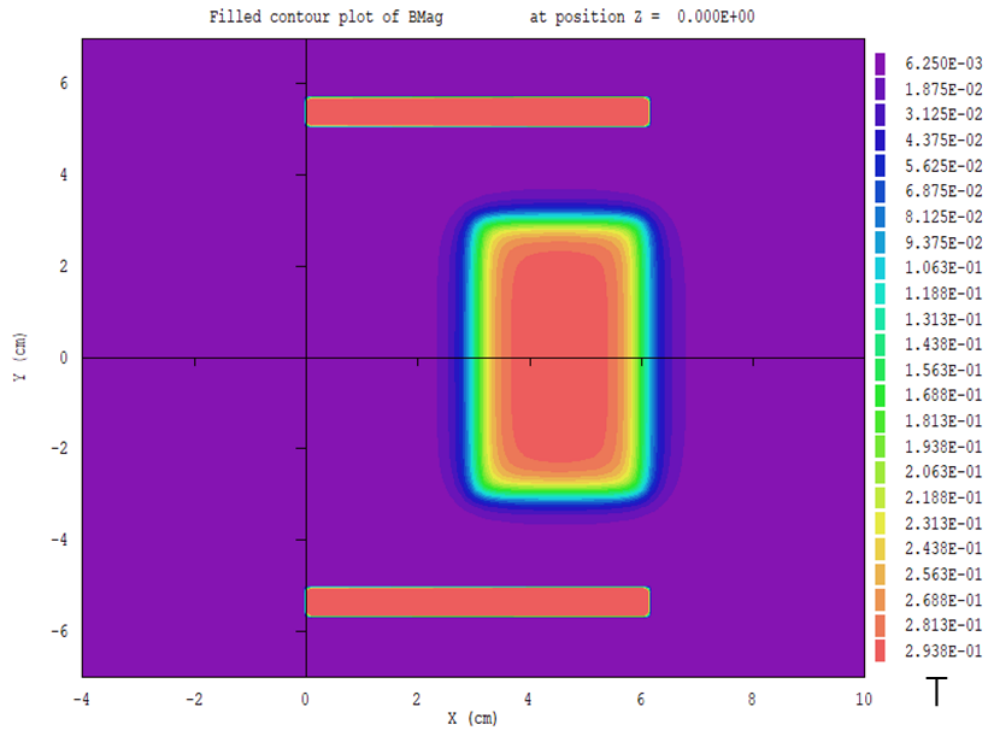


図 3.6 磁場部分の概略図

(a) 入射方向から見た図，(b) イオン通過へ面での断面図

(a)



(b)

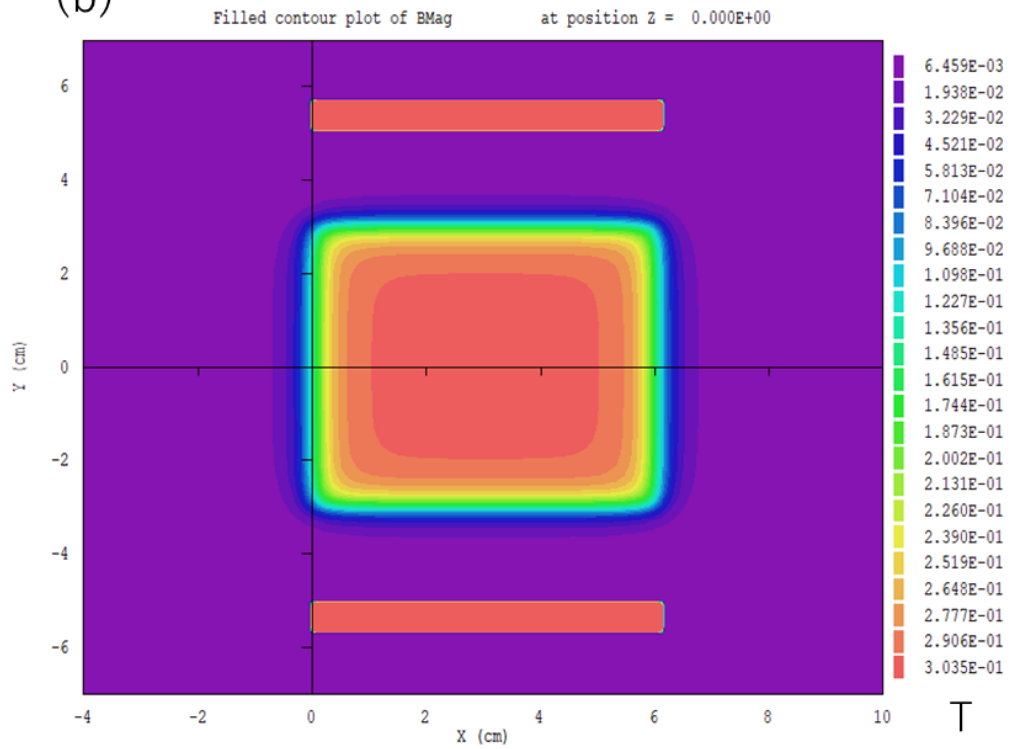


図 3.7 イオン通過平面での磁場強度

(a)磁石 2 個, (b)磁石 4 個

3-4 電場領域の設計

磁場によって曲げられたイオンは、次に電場領域を通過する．以下では、電極の形状や構成について説明する．図 3.8 は、電場領域の概略図である．電極の大きさは、40 mm×90 mm，厚さ 3 mm の純アルミ製の板を 2 枚使用した．電極間距離は 5 mm であり，その中心からイオンは入射する．また電極は外部と絶縁するためにセラミック製のスペーサーを固定部との間に配置し，プラスチック製のねじで固定した．図 3.9 に x-z 平面の断面での等電位線と電界のシミュレーション結果を示す．上の電極を接地し，下の電極に 10 V の電圧をかけた．

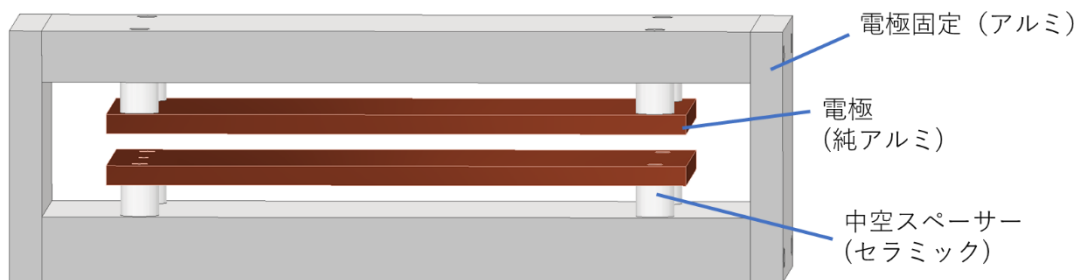
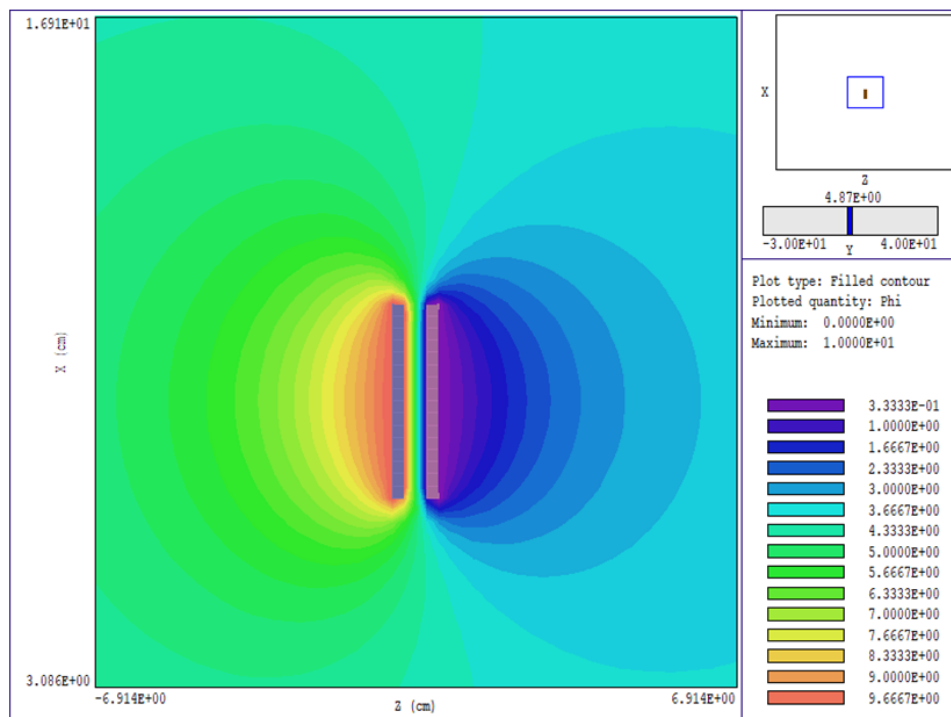


図 3.8 電場領域の概略図

(a)



V

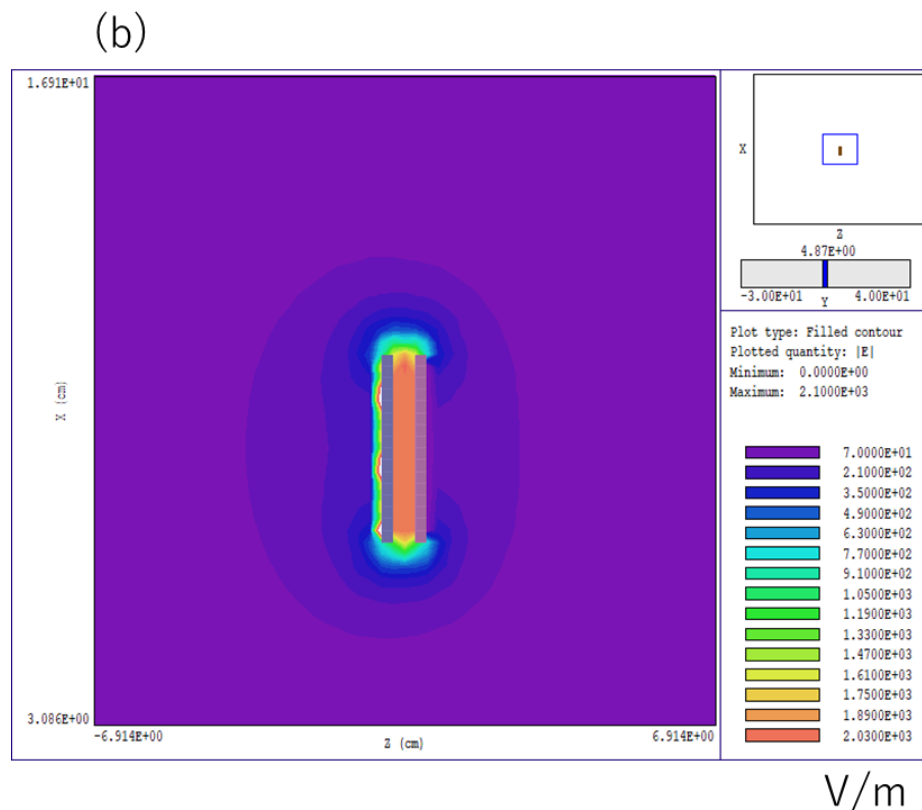


図 3.9 x-z 平面の電場のシミュレーション結果

(a)等電位面, (b)電場強度

3-5 検出部

イオンの検出にはシィアイテクノ社製の ITO ガラス蛍光体印刷基板を用いた。使用した蛍光体の発光ピークは 505 nm である。また、蛍光体成膜面の大きさは 50 mm×66 mm である。蛍光体の発光の撮像には、ソーラボ社製の DCC 1545M と ZWO 社製の ASI533MM Pro を用いた。表 3.2 に使用したカメラの性能を示す。レンズはリコー社製の FL-BC1618-9M を使用した。レンズの仕様を表 3.3 に示す。

表 3.2 カメラの性能

カメラ	DCC 1545M	ASI533MM Pro
解像度	1208×1024	3008×3008
量子効率(@505 nm)	61 %	89 %
センサーサイズ	6.66 mm×5.32 mm	11.31 mm×11.31 mm
読み出しノイズ	<25e	1.0-3.8e
最大露光時間	0.983 秒	2000 秒

表 3.3 レンズの仕様

焦点距離	16 mm
絞り範囲	F1.8-16
最短撮影距離	0.08 m
画角 (1 インチ)	43.8 ° (水平)
マウント	C-mount

3-6 検出器での放物線軌道の計算

イオンの軌道の計算では、先述した漏れ磁場と漏れ電場の影響を考慮し、タイムステップごとにイオンにかかる力を計算し、運動方程式を解いた。

計算に使用する基礎方程式について示す。以下の方程式は直交座標系および MKSA 単位系で記述する。粒子として取り扱うイオンの運動方程式は次式で表される。

磁場領域

$$m_i \frac{d\mathbf{v}_i}{dt} = Ze(\mathbf{v}_i \times \mathbf{B}) \quad (3-5)$$

電場領域

$$m_i \frac{d\mathbf{v}_i}{dt} = Ze\mathbf{E} \quad (3-6)$$

$$\mathbf{v}_i = \left(\frac{dx_i}{dt}, \frac{dy_i}{dt}, \frac{dz_i}{dt} \right) \quad (3-7)$$

ここで、 m_i はイオンの質量、 $\mathbf{v}_i$ はイオンの速度、 Z はイオンの価数、 e は電気素量、 $\mathbf{E}$ と $\mathbf{B}$ はそれぞれ電場、磁束密度を表している。

計算方法は、以下の手順で行った。

- 1) 磁場および電場、入射イオンの角度、速度の初期設定を行う。
- 2) 磁場と電場の値を取得する。
- 3) 取得した電場と磁場から式(3-5)のイオンの運動方程式をもちいてイオンの新たな位置と速度を求める。

以上 2) と 3) を繰り返しイオンの時間的变化を追う。

イオンの初期速度は、発散角より三次元極座標系から三次元直交座標系に変換することにより求めた。

図 3.10 は、磁石 2 個での 1keV のエネルギーを持った $m/z=40$ のイオンの軌道の計算結果である。磁場と電場により曲げられていることがわかる。さらに軌道計算により漏れ磁場，電場による効果の検証も行った。図 3.11 は、 $m/z=40$ のイオンにおける平坦な磁場と電場とシミュレーション結果を考慮したものとの比較である。この結果により漏れ磁場と電場の影響を考慮することが重要であることがわかる。

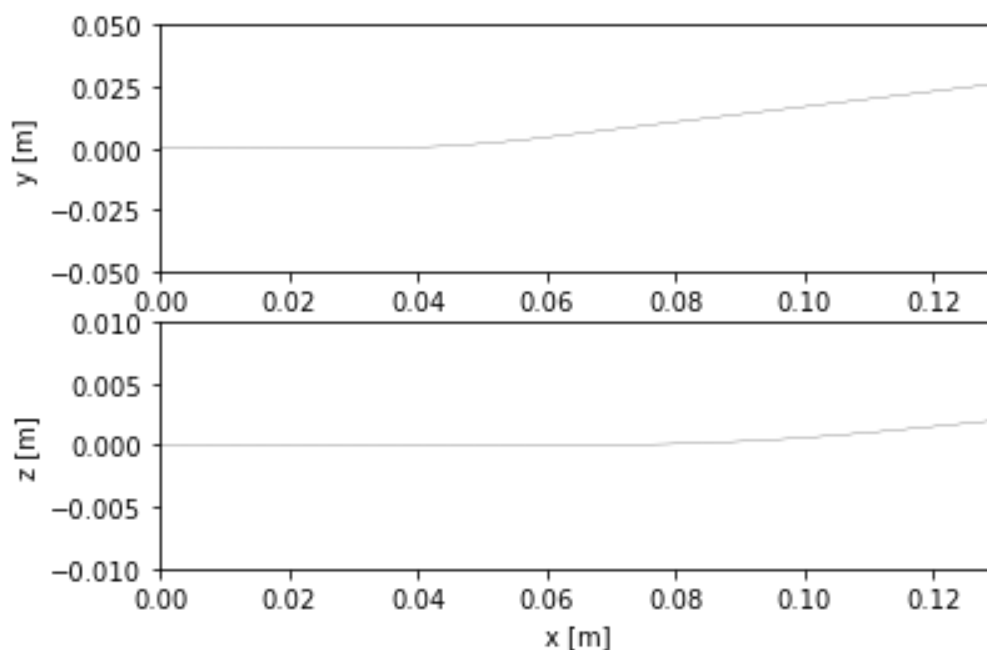


図 3.10 磁石 2 個での 1keV の $m/z=40$ のイオンの軌道

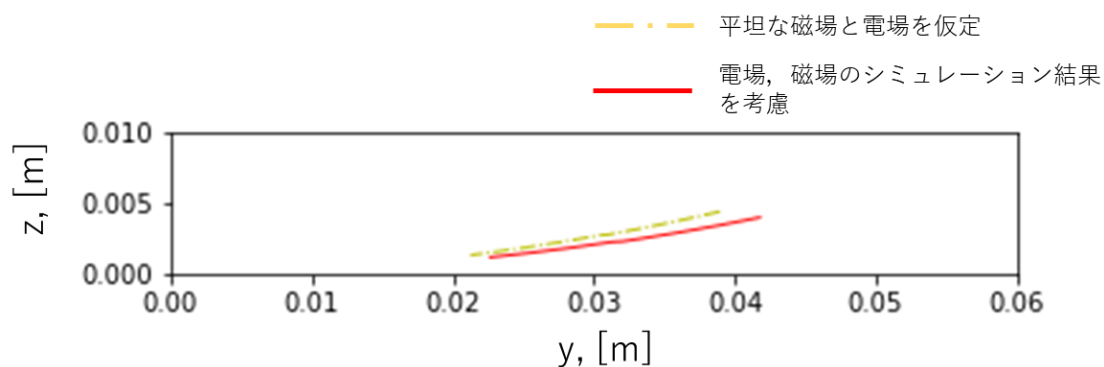


図 3.11 $m/z=40$ のイオンにおける平坦な磁場と電場とシミュレーション結果を考慮したものとの比較

第 4 章 トムソンパラボラによる計測

本章では、キセノンに代わる代替推進剤の候補として挙げられているアルゴンおよび窒素，さらに昇華性推進剤としてアダマンタンでの計測結果について述べる．

4-1 共通の実験体系

以下にトムソンパラボラでの計測における共通の実験体系について記載する．

図 4.1 は，本実験で使用した真空容器である．本実験で使用した真空容器は直径 60 cm，長さ 120 cm の円筒形である．この真空容器では 2 種類の真空排気装置を使用した．ロータリーポンプ(排気速度 1.5×10^{-2} [l/s])は粗排気用，ターボ分子ポンプ(排気速度 5.2×10^{-1} [l/s])は高真空用である．

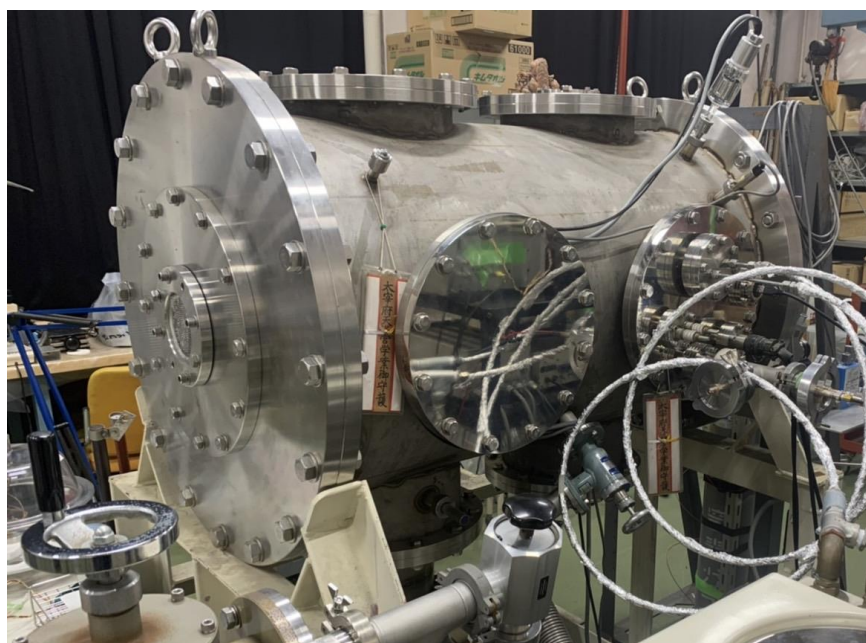


図 4.1 真空チャンバー

イオンエンジンのマイクロ波系統としては，周波数帯域 2.45 GHz の固体発信機と固体増幅器 T161-48DDA（サムウェイ製）を使用した．マイクロ波電力はスラスタに発振されるまで各種ケーブルやコネクタ等を経由するため，実際のマイクロ波のアンプが出力する値と異なることがある．そのためパワーメータを使用し，イオンエンジンに発振されているマイクロ波の電力を測定する校正実験を行い，マイクロ波のアンプの出力値と校正している．

今回の実験では、中和器を使用しないためコレクタをイオンエンジンの下流 200 mm の位置に設置した．図 4.2 に今回作成したコレクタを示す．コレクタは純アルミ製で、大きさは 260 mm×350 mm 厚さ 5 mm である．コレクタの後方にトムソンパラボラを設置するため、中心にトムソンパラボラヘイオンが入射するための直径 5 mm の穴が開いている．コレクタはグランドに接続されている．さらにこのコレクタは昇華性推進剤を使用する際の推進剤の回収の役割も担う．この板には高木製作所製の水冷ペルチェユニット PU-200W が取り付けられている．昇華性推進剤を使用した際にチャンバー内に推進剤がチャンバー内に残ったままとなり、真空度の低下(5.0×10^{-3} Pa 以上)が起こることが問題であった．そのため、チャンバー内の真空度を 3.0×10^{-3} Pa 以下にするためにコレクタの温度を -25°C (3.0×10^{-5} Pa でのアダマンタンの昇華温度)以下になるようにペルチェユニットで冷却した．

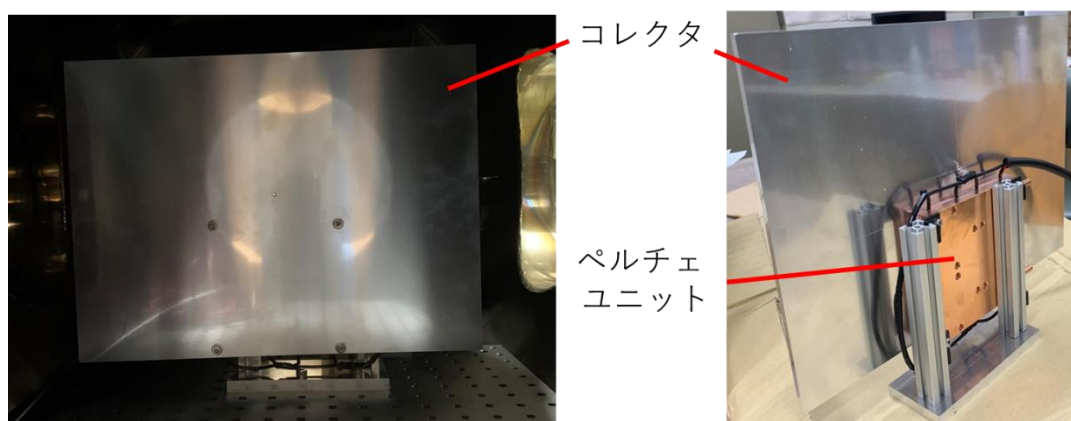


図 4.2 作製した冷却機能付きコレクタ

蛍光板を撮像する際に校正用のイオンの入射していない画像を取得する必要があるため、コレクタとトムソンパラボラの間にシャッターを設置した．シャッターは駿河精機社製の電磁シャッターF77-4を使用した．図 4.3 にイオンエンジン、コレクタ、シャッター、トムソンパラボラの位置関係の図を示す．

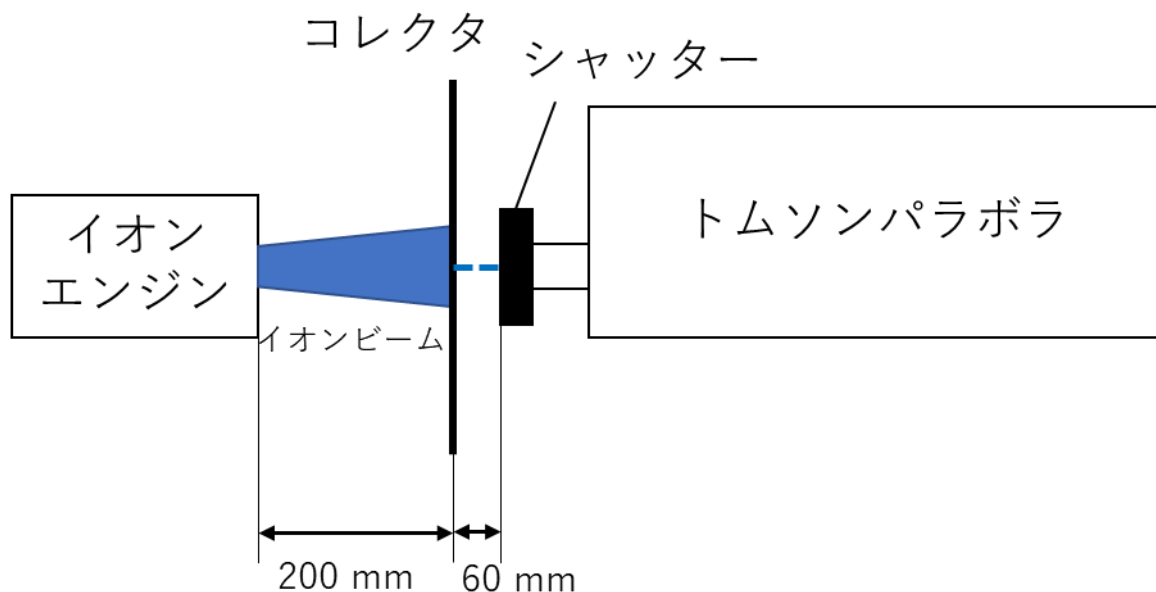


図 4.3 実験体系の全体図

4-2 レンズのピント調整およびゼロ点補正

レンズのピント調整および蛍光板における磁場と電場による偏向がない場合のイオンの入射位置（ゼロ点）をアライメント用のレーザーで行った．レーザーの波長は蛍光体の発光ピークに近い 532 nm である，まず，レーザーがトムソンパラボラにまっすぐ入るように固定する．その後，蛍光板に当たった光が最も明るく小さくなるようにピントを調整し，その点を検出面のゼロ点とした．図 4.4 は，533MM Pro のカメラを使用し校正した際の画像である．この時，レーザー光の大きさは，コリメータによる発散角から求められる広がり幅と同じ直径 1.2 mm の円であった．

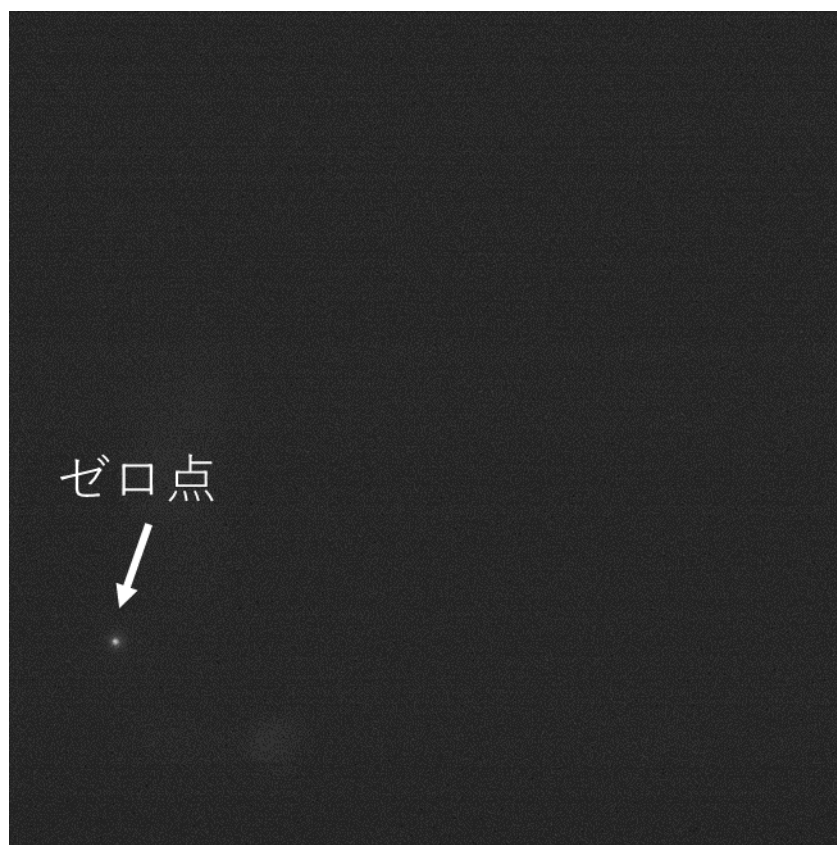


図 4.4 ゼロ点補正のための画像

4-3 アルゴンおよび窒素での計測

昇華性推進剤での実験の前にアルゴンおよび窒素を用いて、トムソンパラボラにおける検出位置の校正および性能の測定を行った。アルゴンや窒素を用いた理由としては、解離による平均分子量の低下を引き起こす主な原因と考えられる分子量の小さいイオンの計測可否を調べるためである。

4-3-1 使用したイオンエンジン

窒素およびアルゴンの実験には、本研究室の池田氏によって製作された 100 μN 級イオンエンジンを使用した。²¹ その外観を図 4.5 に示す。本イオンエンジンは 3 枚グリッドを採用しており、グリッドと同じ C/C 製のリングとセラミック製のインシュレータでグリッド間隔を保持している。放電室は内径 24 mm 高さ 19 mm の軟鉄製であり、高さ方向に磁化されたリング型サマリウムコバルト磁石（内径 3 mm，外径 8 mm，高さ 3 mm）1 つを上流，周方向磁化のセグメント型サマリウムコバルト磁石（内半径 9 mm，外半径 12 mm，高さ 2 mm，中心角

90°) 4つを下流に取り付けて磁場を構築している．アンテナはモリブデン製の星形を採用しており，放電室中央に SMA コネクタで固定されている．グリッドのパラメータを表 4.1 に示す．

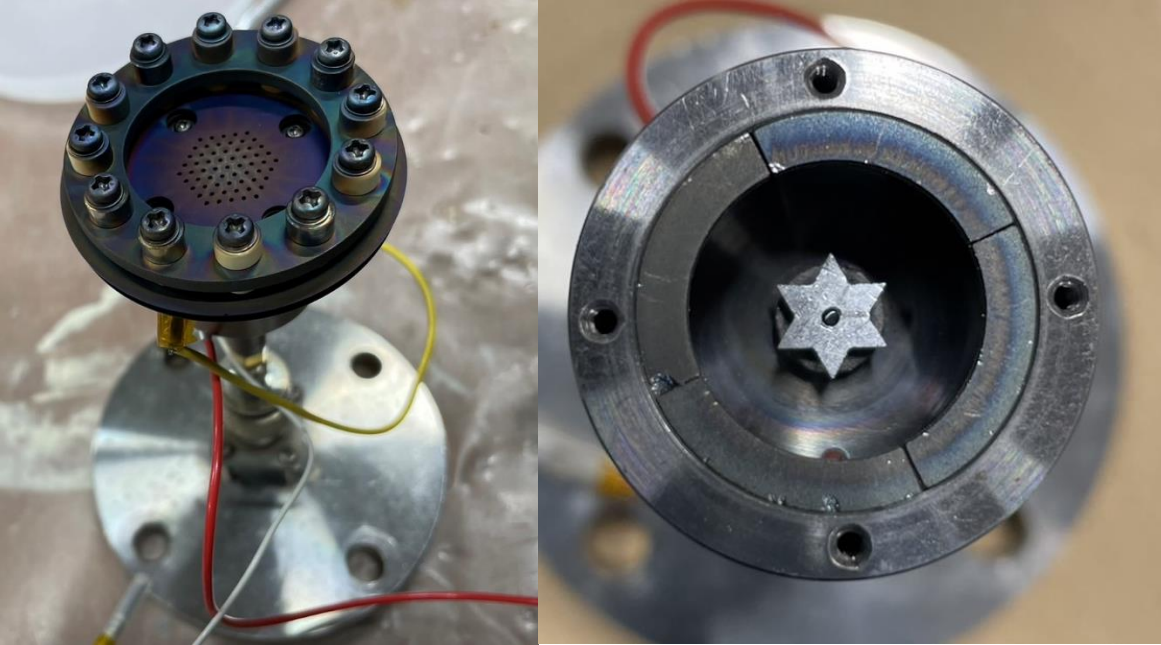


図 4.5 イオンエンジン(池田モデル)

表 4.1 グリッドのパラメータ

Parameter	Screen	Accelerate	Decel
Hole diameter, mm	1.5	0.7	0.7
Potential, V	600~1000	-150	0
Thickness, mm	0.3	0.9	0.3
Hole pitch, mm		2.0	
Grid gap, mm	0.5		1.0
Hole number		61	
Material		DFP3-2	

4-3-2 実験条件

ガスの供給には，KOFLOC 社製のスタンダードマスフローコントローラ 3660 を使用した．こちらのマスフローコントローラはアルゴンで校正されているた

め、窒素を推進剤として供給する際には窒素に対するアルゴンのコンバージョンファクターを 1.44 として流量を調整した.

実験体系の回路図を図 4.6 に示す.

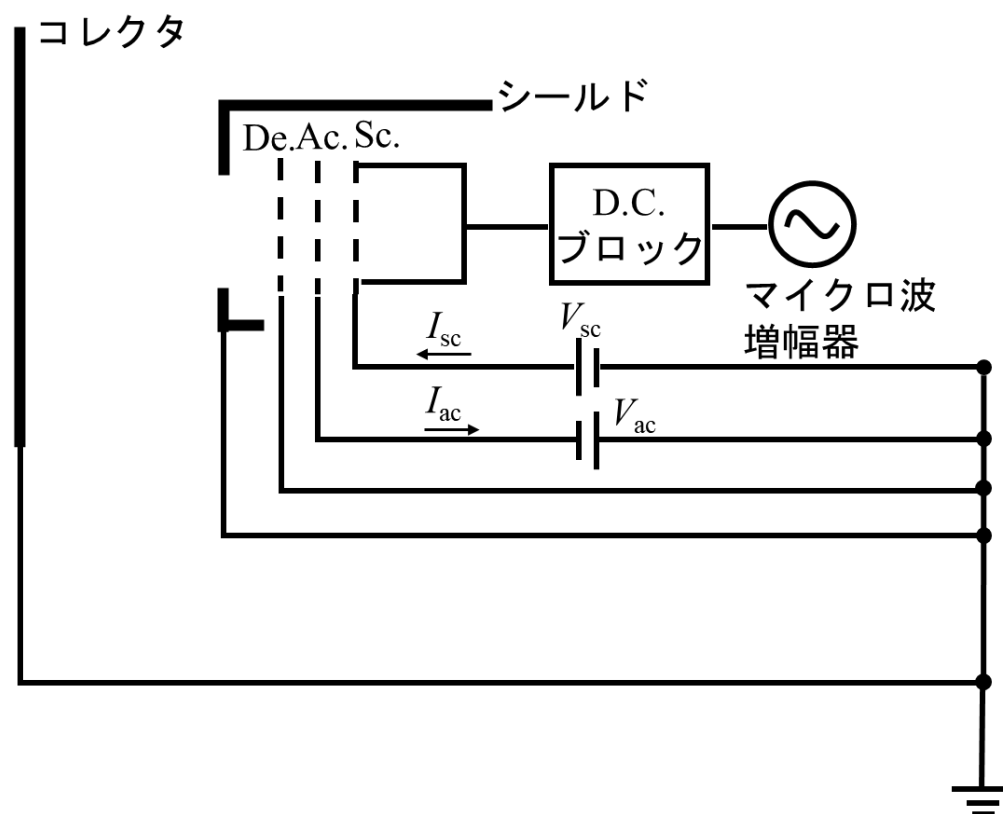


図 4.6 実験体系回路図

4-3-3 DCC1545M での実験結果

DCC1545M のカメラを使用した際のアルゴンでの結果について述べる. こちらのカメラの感度と露光時間では, $100\ \mu\text{m}$ と $150\ \mu\text{m}$ のピンホールを使用した際にイオンの軌道を撮像することができなかった. そこで, 長さ $50\ \text{mm}$ のレンズチューブの前後に直径 $1\ \text{mm}$ まで絞ることのできるアイリスを 1 つずつ固定し, 実験を行った. この時のコリメータ部による発散角は, 1.15° である. 推進剤流量は $0.5\ \text{sccm}$, 投入電力は $10\ \text{W}$, スクリーン電圧 $1\ \text{kV}$ にて実験を行った. 露光時間は $983\ \text{ms}$ で設定し, 同じ条件で撮像した画像を 10 枚重ね合わせたのち, バックグラウンドとして撮影したシャッターを閉じた状態の画像 10 枚を差し引いた.

図 4.7 は、10 枚重ね合わせた結果の画像に軌道計算の結果を重ね合わせたものである。一番左端のゼロ点に近い光は、スラスタからの光である。 $y=35\text{ mm}$, $z=4\text{ mm}$ 付近のところにある光は $m/z=40$ の軌道上にあるため Ar^+ である。同様に、 $y=55\text{ mm}$, $z=5\text{ mm}$ 付近にある光は $m/z=20$ の軌道上にあるため Ar^{2+} である。以上より磁場と電場により価数の違うイオンを分離することに成功した。しかし、イオンのエネルギーを求めるには軌道が広がりすぎており、ノイズが大きく評価が非常に難しい。そこで、DCC1545M よりも感度がよく露光時間を長く設定でき、ノイズの小さい ASI533MM を使用することで、直径の小さいピンホールでも計測することが可能ではないかと考えた。

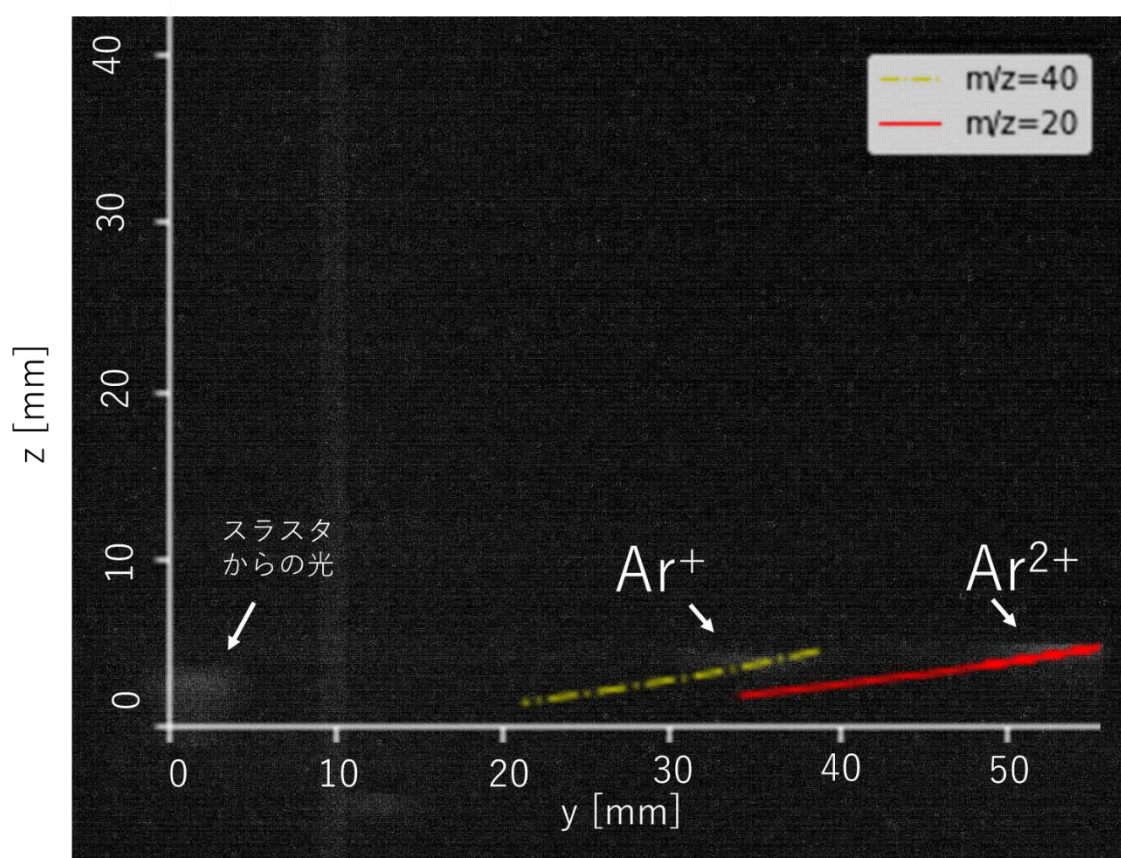


図 4.7 DCC1545M で撮像した $V_{sc}=1\text{ kV}$, $\dot{m} = 0.5\text{ sccm}$, $P=10\text{ W}$ での結果

4-3-4 ASI533MM Pro での実験結果

ASI533MM Pro は天体観測用冷却 CMOS カメラであり、二段ペルチェ冷却機能を持つ。しかし、真空中では、ファンによる排熱が不可能なため直接冷却する必要がある。本実験では、高木製作所製の水冷式ペルチェ冷却ユニット PU-50A をカメラ固定用の板（純アルミ製）に取り付けカメラを冷却した。ASI533MM

Pro を使用したところ、コリメータ部がピンホールの直径 $100\ \mu\text{m}$, $150\ \mu\text{m}$ の 2 つをレンズチューブに $22.2\ \text{mm}$ 間隔で設置したもので撮像することに成功した。以下にアルゴンと窒素の結果を示す。

(1)アルゴン

アルゴンの実験では、流量を $0.5\ \text{sccm}$, マイクロ波を $10\ \text{W}$ に固定し、スクリーン電圧を $600\ \text{V}$, $800\ \text{V}$, $1000\ \text{V}$ に変えて実験を行った。実験前の到達真空度は、 $5.2 \times 10^{-5}\ \text{torr}$ であった。

図 4.8 に実験結果を示す。(a), (b), (c) は得られた画像から、同じ露光時間でシャッターを閉じた状態で撮像した画像を差し引き、補正したものである。スクリーン電圧が $1\ \text{kV}$ の時のみ露光時間は $300\ \text{秒}$ であり、 $800\ \text{V}$ と $600\ \text{V}$ における露光時間は $180\ \text{秒}$ である。結果(d), (e), (f) は補正した画像に計算によって得られた軌道を重ね合わせたものである。ピンホールの直径を小さくしたことにより、広がりの小さい軌道の撮像に成功した。 $1\ \text{kV}$ 時には $m/z=40$ と $m/z=20$ の軌道上のイオンが検出された。つまり $m/z=40$ は Ar^+ であり、 $m/z=20$ は Ar^{2+} のイオンが検出されたことを意味する。図 4.9 は、 $E = 0\ \text{V/m}$, $V_{\text{sc}} = 600\ \text{V}$ での結果である。イオンの軌道が $z=0$ 上にあることから、ピンホールの穴の位置によるずれなどはないものと考えられる。

得られた結果より、1 価と 2 価のイオンの存在比を求めた。存在比の求め方は、イオンが飛んできた範囲内の画素値を積分することにより算出し、1 価と 2 価の相対的な関係を求めた。図 4.10 に積分した範囲を示す。結果は Ar^+ : $\text{Ar}^{2+}=4:1$ であった。

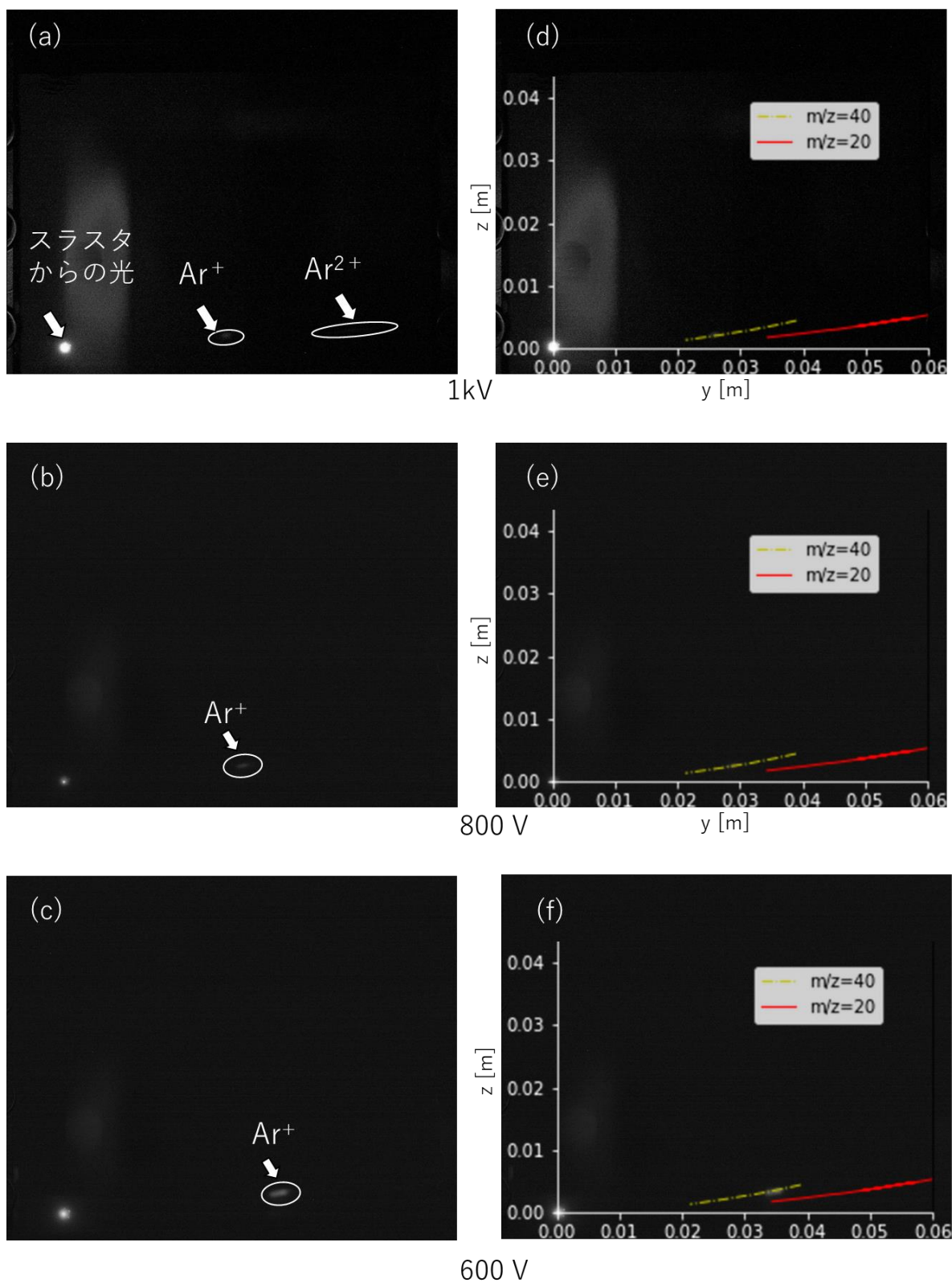


図 4.8 アルゴンでの結果

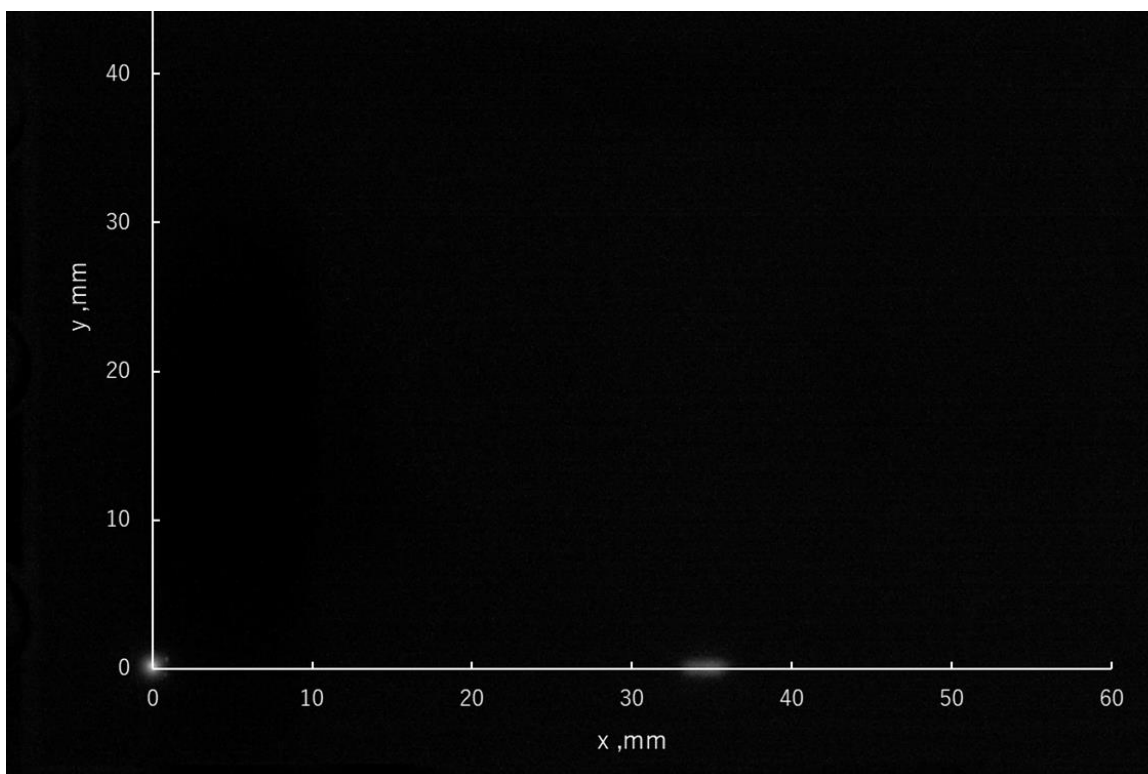


図 4.9 $E = 0 \text{ V/m}$ での結果($V_{sc} = 600 \text{ V}$)

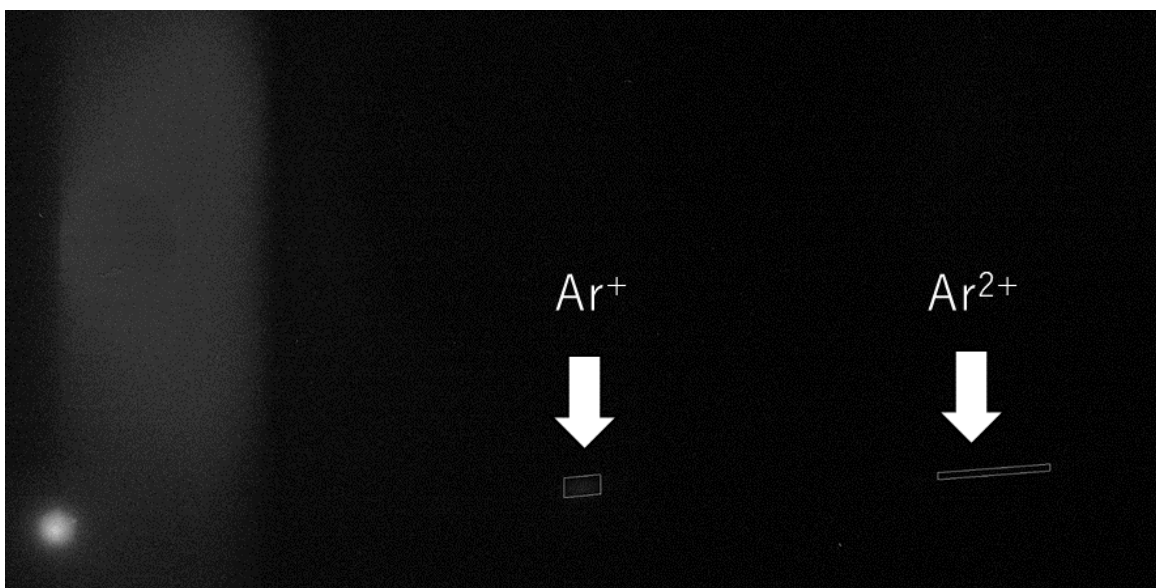


図 4.10 イオンの存在比を算出する積分範囲

(2)窒素

推進剤として窒素を使用した実験条件は、流量を 0.5 sccm , マイクロ波を 15 W に固定し、スクリーン電圧を 600 V , 800 V , 1000 V と変えて実験を行った。実験前の到達真空度は、 $3.2 \times 10^{-5} \text{ torr}$ であった。

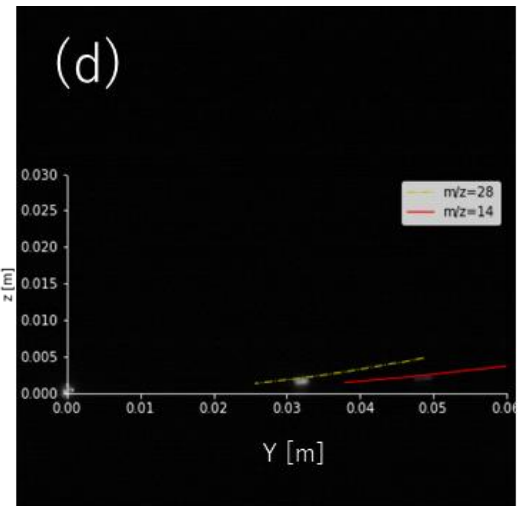
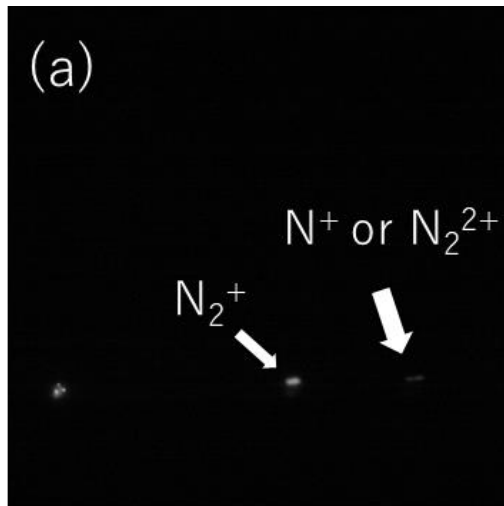
アルゴンでの実験結果には、カメラの電源ランプの光が装置内で反射した光が映っていた。そこで、電源ランプおよび蛍光板の蛍光体が塗布されていないガラス面を黒いテープで覆うことで迷光が入り込まないように対策した。さらに本実験ではビンニングを行い、短い露光時間でも光を捉えられるようにした。本実験では露光時間 30 秒で撮影した画像を 16 枚重ね合わせ、その後にバックグラウンドとしてシャッターOFF 時の同じく露光時間 30 秒、16 枚の画像を差し引くことでノイズを相対的に減少させ、計測した。

図 4.11 に窒素での結果を示す。窒素では、 $m/z = 28$ と $m/z = 14$ の軌道上にイオンが検出された。 $m/z = 28$ は N_2^+ であり、 $m/z=14$ は N_2^{2+} もしくは N^+ のイオンが検出されたことを意味する。 N_2^{2+} イオンの存在比は N^+ のイオンに比べて非常に小さいことが報告されているので、本実験で計測されたイオンは N^+ であるとする。²²

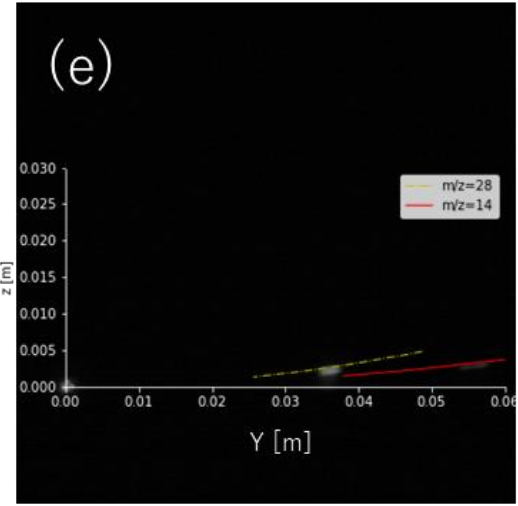
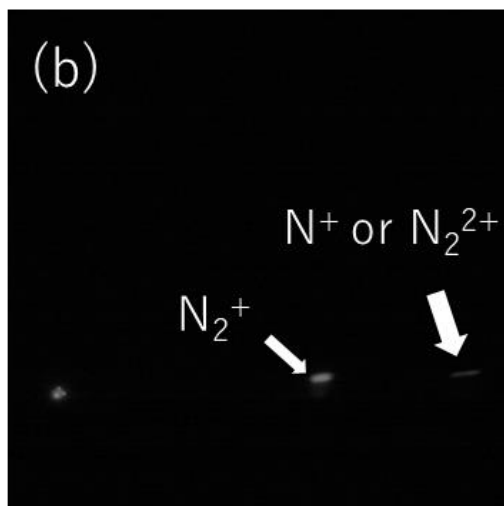
アルゴンと同様に窒素においてもイオンの存在比を算出した。スクリーン電圧 600 V では、 $m/z=14$ のイオンはみられなかったため、スクリーン電圧が 1000 V と 800 V にて算出した。スクリーン電圧 1000 V では、 $N_2^+ : N^+ = 1 : 0.186$ であった。また、800 V では、 $N_2^+ : N^+ = 1 : 0.171$ であった。

得られた軌道の丸みの評価を行った。イオンは放電室内部において数十 eV のエネルギーの幅を持って存在していると考えられる。そのため、グリッドによって加速され蛍光板に当たったイオンもそのエネルギー幅を持つと考えられる。その際のイオンの軌道は図 3.5 のような楕円状のイメージがエネルギーの幅によって引き延ばされているようなものとなる。図 4.12 はスクリーン電圧が 800V での N_2^+ の拡大画像である。同一エネルギーの N_2^+ は円状に広がっていると近似すると、半径 0.55 mm の円となった。これは、軌道計算の結果である半径 0.6 mm と同程度であり、妥当であるといえる。イオンの発散角による広がりがないと考えられる場合、この円の中心を通る軌道と計算結果は一致すると思われる。

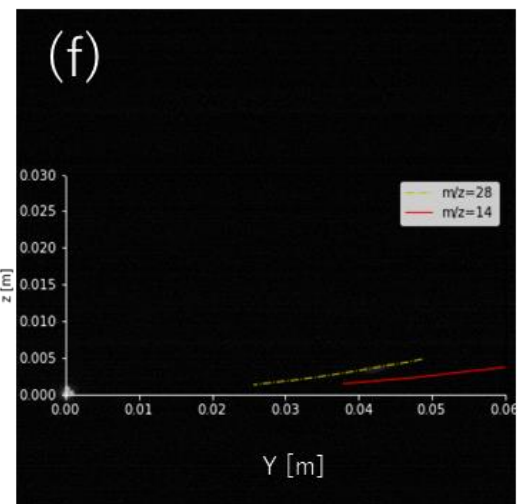
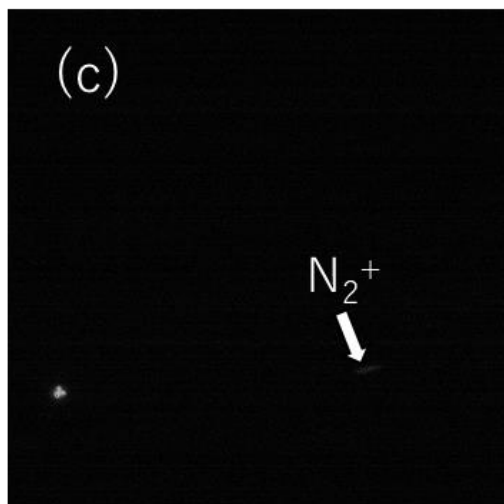
図 4.11 に示す軌道は大方、発光位置の近くにあるものの軌道の中心にあるわけではない。これは、印加電圧を 10V としているものの実際に電源から供給される電圧が 9.5~10.5 V の間にある可能性があること、ピンホールの位置がずれて角度のついたイオンが入射してきたことなどが考えられる。そこで、今回は印加電圧を 10.5 V、入射角が z マイナス方向に 0.25° 傾いているものと補正して軌道を算出した。補正結果を図 4.13 に示す。強度分布からエネルギー分布関数を再構築した。補正軌道を用いたエネルギー分布を図 4.14 に示す。



1 kV



800 V



600 V

図 4.11 窒素での結果

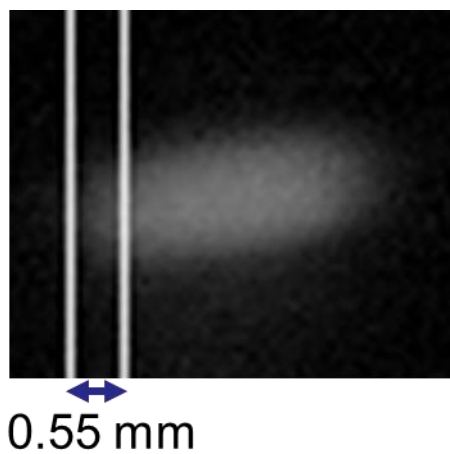
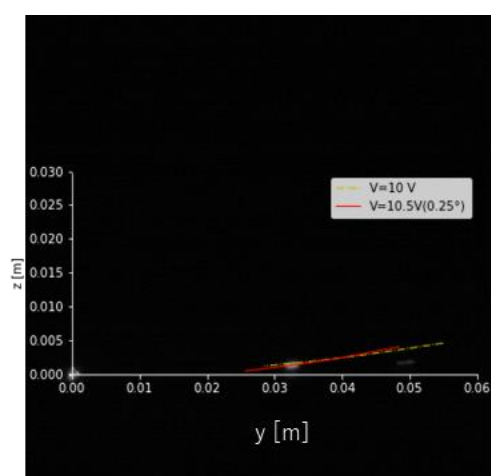
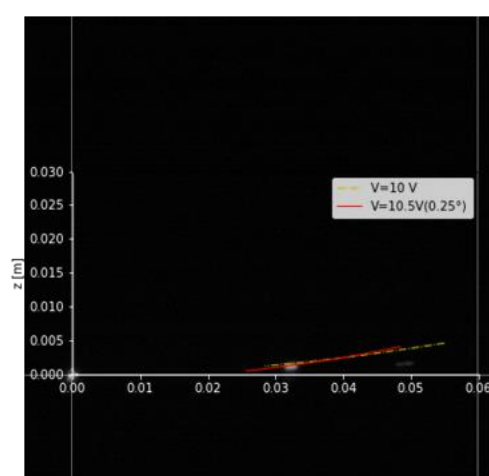


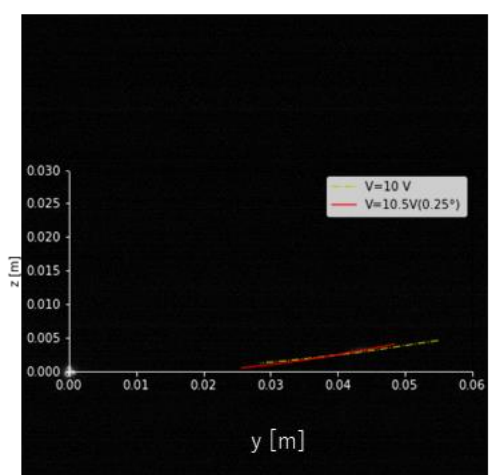
図 4.12 N_2^+ イオンのエネルギーによる広がり



1 kV



800 V



600 V

図 4.13 軌道の補正結果

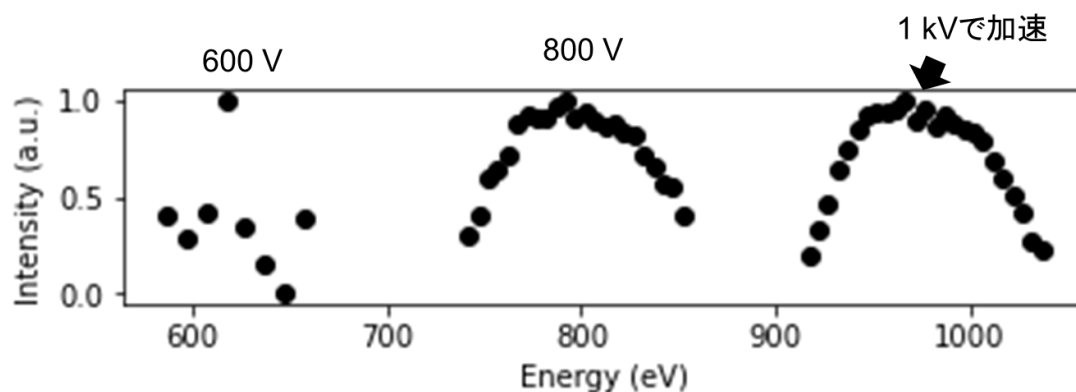


図 4.14 エネルギー分布

4-4 昇華性推進剤での計測

本実験では、昇華性推進剤としてアダマンタンを中心に実験を行った。アダマンタンは、分子量が 136.23 とキセノンよりも大きく、酸素原子を含まないため原子状酸素による材料劣化の懸念がない。さらにアダマンタンは電離電圧が低く、イオン化する際に生じることのある開裂が比較的少ないと報告されている。²³

その他、L-メントール、サリチル酸を推進剤として選択し実験を行った。

4-4-1 使用したイオンエンジン

イオンエンジンは本研究室で近岡氏によって作製されたものを使用した。その外観を図 4.15 に示す。内径 21 mm 高さ 12 mm の Al 製放電室の周囲に 4 mm × 4 mm, 高さ 12 mm (磁化方向) のサマリウムコバルト磁石を 10 個配置し、軟鉄製の 2 枚のヨークで挟み、それを外形 50 mm × 50 mm の Al 製の箱の中に収納して固定している。放電室の中央には Mo 製の星形アンテナが SMA コネクタによって固定されており、マイクロ波を放電室内に伝搬している。グリッドは 2 枚グリッドを採用している。グリッドのパラメータを表 4.2 に示す。

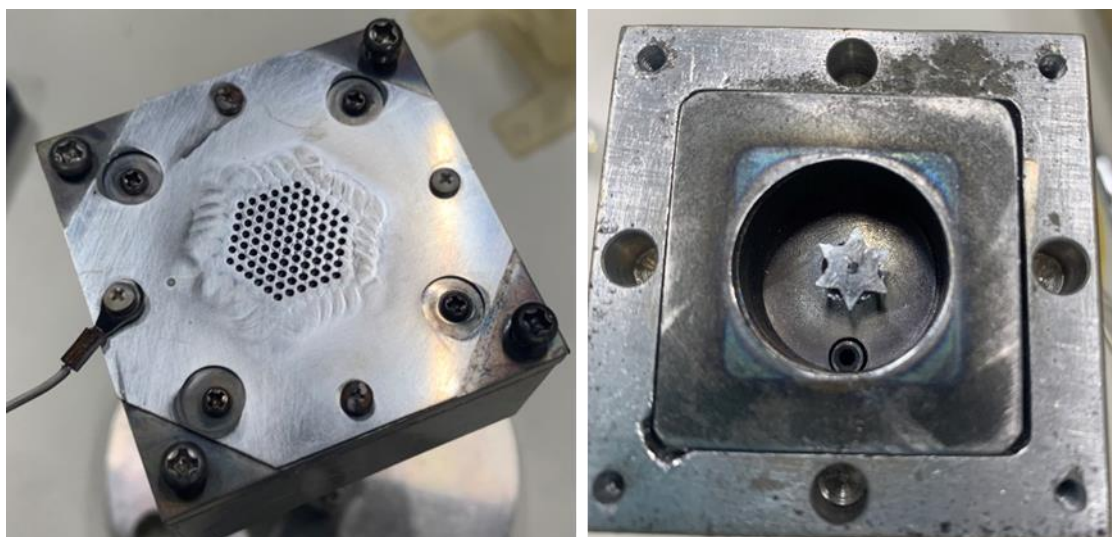


図 4.15 イオンエンジン（近岡モデル）

表 4.2 グリッドのパラメータ

	Screen	Acceleration
Hole diameter, mm	1.20	0.70
Potential, V	1000	-150
Thickness, mm		0.3
Hole pitch, mm		1.20
Grid gap, mm		0.2
Number of holes		91

4-4-2 流量制御体系

プラズマ点火時には流量を多くし、その後流量を調節できるような流量制御系が必要である。図 4.16 は昇華性推進剤における飽和蒸気圧・温度曲線²⁴⁻²⁷と温度制御用に作成したタンクと配管のコンダクタンスから流量を温度の関数としてモデル化したものである。図 4.16 より温度を制御することによりプラズマの点火時に流量を多くし、その後プラズマの状況に合わせて流量を調整できるのではないかと考えた。本実験では温度制御のために高木製作所製の水冷式ペルチェ冷却ユニット PU-50A とヒーターを用いた。プラズマ点火時など流量を大きくしたい時はヒーターを使用し、その後、流量を抑えたい時はペルチェユニットによるタンクの冷却によって行う。

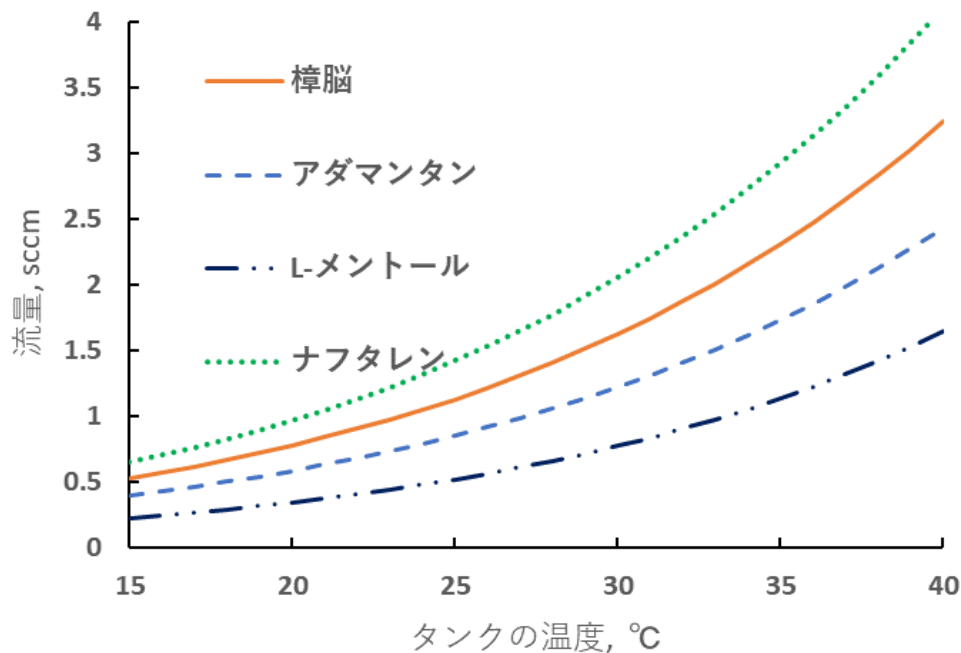


図 4.16 昇華性推進剤のタンク温度-流量グラフ

図 4.17 に今回作成したタンクの図を示す．タンク本体は，本研究室が所有している 3D プリンター (Formlabs 製の Form2) で作成した．タンクの底面にはペルチェユニットへの固定用の穴が開いており，ペルチェユニットとタンクの間に熱伝導シートを挟み底面からタンクを冷やすこととした．さらにペルチェユニットに固定する際に熱電対をタンクに取り付けこちらの温度でタンク温度をモニタリングした．蓋はステンレス製であり，タンクと蓋の間に O リングを挟むことで推進剤の漏れが起こらないようにした．蓋にはバルブ装着用のねじ山が切られており直接バルブをつけられるようになっている．バルブは SMC 社製の VDW14KA ソレノイドバルブを使用した．バルブのオリフィス径は 1.6 mm であり， $C=0.14 \text{ dm}^3/\text{s} \cdot \text{bar}$ である．バルブを使用することにより，意図しない推進剤の漏れを防いだ．図 4.18 はタンクにフタおよびバルブを取り付けた図である．図 4.19 に実験装置の概要を示す．先述の通りタンクをペルチェユニットにより底面より冷やし，ヒーターによって側面から加熱する．

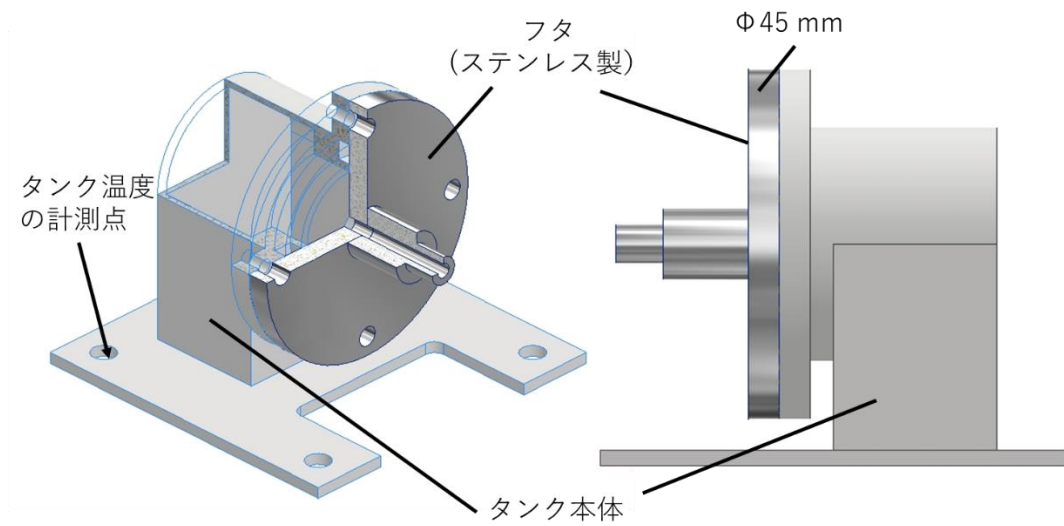


図 4.17 作製したタンク

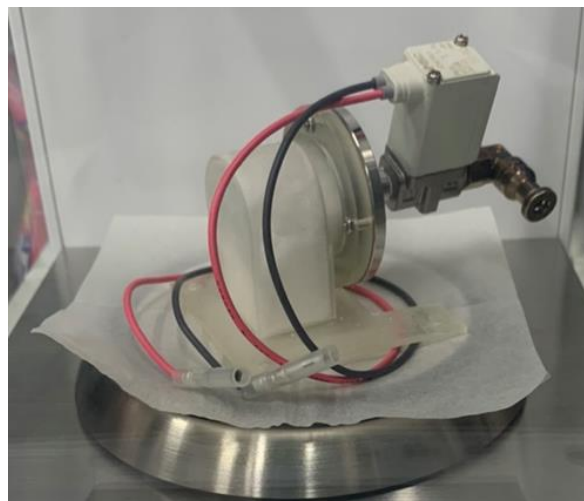


図 4.18 タンクにバルブを取り付けた図

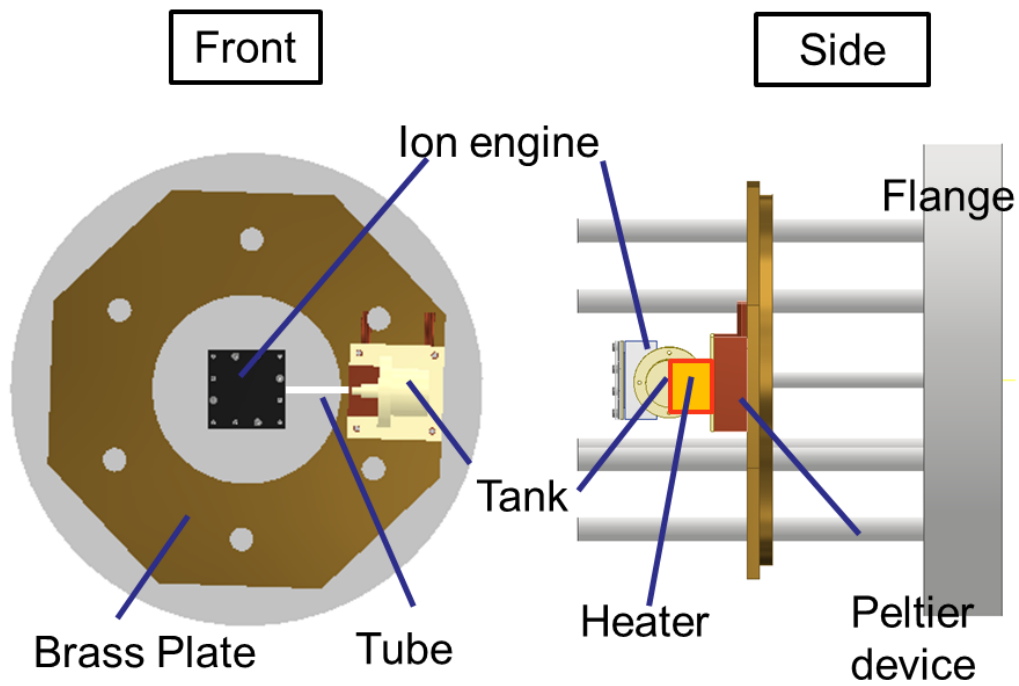


図 4.19 実験装置

4-4-4 実験条件

実験開始前の到達真空度は 3.2×10^{-5} torr であった．図 4.20 に実験体系回路図を示す．

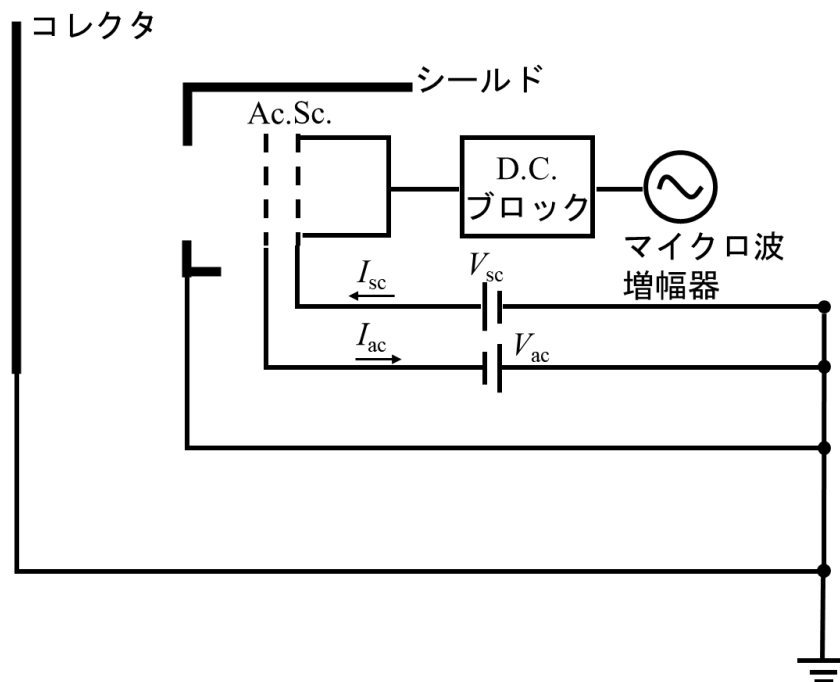


図 4.20 実験体系回路図

4-4-5 実験結果

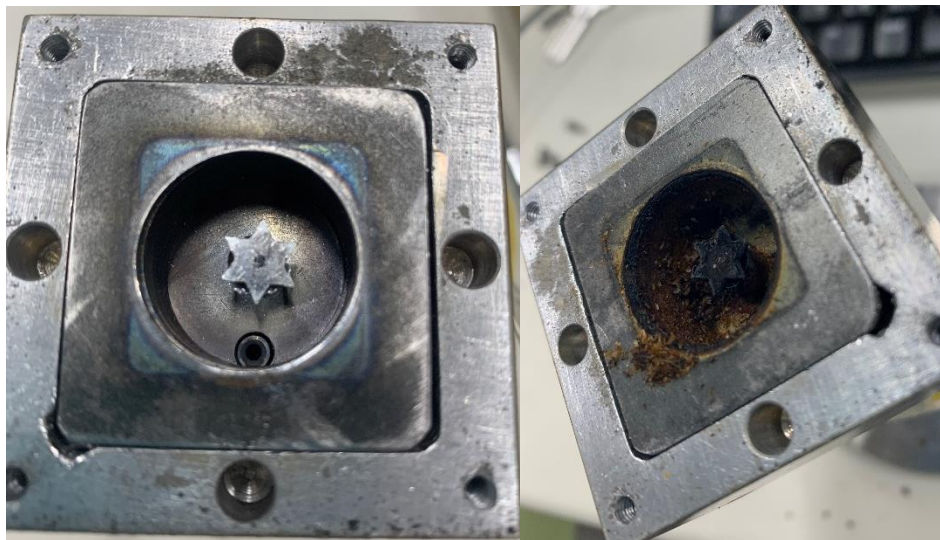
アダマンタンでの実験結果について述べる．ヒーターによる過熱を必要とせずプラズマの点火に成功した．その後，タンク温度を -2°C に保ち実験を行った．この時の平均流量は 0.06 sccm であった．この平均流量は，実験開始前と終了後のタンクの質量変化をバルブ作動時間で除することにより算出したものである．マイクロ波の投入電力は， 4 W とした．プラズマの点火および維持は問題なかったもののスクリーン電圧を 100 V まで上げたところ以上放電が起こり 100 V 以上の電圧をかけることができず，計測を行うことができなかった．流量や使用したイオンエンジンのグリッドの形状がアダマンタンに図 4.21 はアダマンタンでのスクリーン電圧 100 V 印加時のイオンエンジン作動時の写真である．

図 4.22 は実験前後のイオンエンジンの写真である．解離した物質が重合したものとみられる物質が放電室内およびグリッドに付着していた．このことから解離の度合いを調整し，重合を防ぐことの重要性が示唆される．

また，その他の L メントールおよびサリチル酸で実験を行ったが，いずれもプラズマをつけることには成功したものの，イオンビームの引き出しには至らなかった．今後は，流量の制御および放電室の形状の最適化を行い計測をする必要がある．



図 4.21 アダマンタンでのイオンエンジンの作動時



実験前

実験後

図 4.22 アダマンタンを使用した際の実験前後の放電室内

第5章 結論

解離の起こる昇華性推進剤を用いるには、イオンエンジンのグリッド設計と制御の両面から、解離の度合いを推定することが不可欠である。そのために解離とプラズマ諸量（プラズマ密度、温度）の定量的な関係を明らかにし、昇華性推進剤に最適なプラズマ条件を模索することが必要である。本研究では、イオンエンジンにおける昇華性物質を推進剤として用いた際の解離度計測に向けたトムソンパラボラの開発を行った。

- ・軌道計算をする際に漏れ磁場や電場などの端の効果によりイオンの軌道が変化することを確認した。

- ・アルゴンと窒素を推進剤として用いた際に排出されたイオンの分離に成功した。分離されたイオンの存在比を画像から見積もった。スクリーン電圧が1kVにおけるイオンの存在比を以下に示す。

$$\text{Ar}^+ : \text{Ar}^{2+} = 4 : 1, \quad \text{N}_2^+ : \text{N}^+ = 1 : 0.186$$

- ・得られた強度分布からエネルギー分布関数を再構築した。しかしながら、デコンボリューションを必要がある事が分かった。

- ・バルブ搭載型タンクにおけるアダマンタンのプラズマの点火および維持に成功した。今後は、流量やマイクロ波を調整しながら、イオンの引き出しを行いトムソンパラボラによる解離度調査を行うことが必要である。

参考文献

1. JAXA. 小惑星探査機「はやぶさ2」再突入カプセルの回収結果について.
https://www.jaxa.jp/press/2020/12/20201206-1_j.html (2020).
2. Glaser, P. E. Power from the Sun: Its Future. *Science* (1979) **162**, 857–861 (1968).
3. 中野 正勝. 大出力電気推進を用いた宇宙太陽発電衛星の輸送コスト評価. 宇宙太陽発電 **1**, 74–79 (2016).
4. 荒川 義博 西山 和孝, 國中 均, 中山 宜典 & 西山 和孝. イオンエンジンによる動力航行. (コロナ社, 2006).
5. Linnell, J. A. & Gallimore, A. D. Efficiency analysis of a hall thruster operating with krypton and xenon. *J Propuls Power* **22**, 1402–1412 (2006).
6. Yamamoto, N., Miyoshi, M., Takao, Y. & Nakashima, H. Development of a Microwave Discharge Ion Thruster using Argon. *Trans.JSASS Space Tech* **7**, pb_119-pb_124 (2009).
7. Lotz, B. Plasma physical and material physical aspects of the application of atmospheric gases as a propellant for Ion-Thruster of the RIT-Type. (Justus-Liebig-University of Giessen, 2013).
8. Hopkins, M. A. & King, L. B. Performance comparison between a magnesium and xenon-fueled 2 kilowatt hall thruster. *J Propuls Power* **32**, 1015–1021 (2016).
9. Sengupta, A. *et al.* An Overview of the VHITAL Program: A Two-Stage Bismuth Fed Very High Specific Impulse Thruster With Anode Layer. in *IEPC-2005-238* (2005).
10. Sovey, J. S. *et al.* Ion Propulsion Development Projects in U.S.:
Space Electric Rocket Test I to Deep Space I Ion Propulsion Development Projects in U.S.: Space Electric Rocket Test I to Deep Space 1. JOURNAL OF PROPULSION AND POWER vol. 17.
11. Szabo, J. J. *et al.* Iodine Propellant Space Propulsion EP Test Standards View project High Throughput Hall Thruster for Small Spacecraft View project Iodine Propellant Space Propulsion. <https://www.researchgate.net/publication/305700508>.
12. Rafalskyi, D. *et al.* In-orbit demonstration of an iodine electric propulsion system. *Nature* **599**, 411–415 (2021).
13. 中野 正勝, 山本 直嗣, 大川 恭志 & 船木 一幸. 昇華性物質をイオンエンジンに用いる上での課題. in *JSASS-2021-4357-2L08* (2021).
14. 白木 僚. 昇華性推進剤を用いたイオンエンジンのプラズマ特性把握. (九州大学, 2021).
15. 栗木 恭一 & 荒川 義博. 電気推進ロケット入門. (東京大学出版会, 2003).
16. 安達 美咲. イオンエンジンの推進剤としての昇華性物質の検討. (九州大学, 2020).
17. 中野 正勝, 山本 直嗣, 大川 恭志 & 船木 一幸. 昇華性物質を推進剤とするイオンエンジンの性能評価. 宇宙太陽発電 **6**, 1–4 (2021).
18. 平野 賢治. 様々な推進器への適用を目指したE×Bプローブの開発研究. (九州大学).

19. Olsen, J. N., Kuswa, G. W. & Jones, E. D. Ion-expansion energy spectra correlated to laser plasma parameters. *J Appl Phys* **44**, 2275–2283 (1973).
20. Sakabe, S., Mochizuki, T., Yamanaka, T. & Yamanaka, C. Modified Thomson parabola ion spectrometer of wide dynamic range. *Review of Scientific Instruments* **51**, 1314–1315 (1980).
21. 池田 凌. ドラッグフリー衛星への搭載を目指した100 μ N級イオンエンジンの開発. (九州大学, 2019).
22. Itikawa, Y. Cross sections for electron collisions with nitrogen molecules. *J Phys Chem Ref Data* **35**, 31–53 (2006).
23. Holste, K. *et al.* Ion thrusters for electric propulsion: Scientific issues developing a niche technology into a game changer. *Review of Scientific Instruments* vol. 91 Preprint at <https://doi.org/10.1063/5.0010134> (2020).
24. 疋田 晴夫, 前田 道宏 & 梅村 実. 充填物-ガス流間の物質移動. 化学工学 **28**, 214–220 (1964).
25. Nelson, C. Investigation of Vaporization Enthalpies and Vapor Pressures of Organic Compounds by Correlation Gas Chromatography. *The journal of Chemical Thermodynamics* **121**, 175–186 (2018).
26. Stl Ll, D. R. Vapor Pressure of Pure Substances ORGANIC COMPOUNDS The increasing importance of vapor pressures in the. *Industrial and Enginerring Chemistry* **39**, 517–540 (1947).
27. Morecroft, D. W., Perhydrochrysene, C. & CisHm Tricyclohexylmethane, - Dicyclohexylhexane. Vapor Pressures of Some High Molecular Weight Hydrocarbons. *JOURNAL OF CHEMICAL AND ENGINEERING DATA* **9**, 488–490 (1964).

謝辞

本研究の遂行ならびに本論文の執筆にあたって、ご指導、ご鞭撻を頂き、貴重な実験の場を当ててくださった山本直嗣教授に心より感謝いたします。先生のご指導により、論文執筆や実験において数多くの助言を頂いた森田太智助教に深く感謝いたします。また、研究生活がスムーズにいくよう、様々なサポートをしていただき、親切に接して頂いた大内田映子秘書、宮本朋子秘書に深く感謝いたします。トムソンパラボラの加工をしていただいた九州大学工学部技術部の蓮尾様に深く感謝いたします。

明るく充実した研究室での生活を送らせていただいた研究室の皆様に感謝致します。

特に同じ実験グループとして、実験準備や作業をともし助言をしてくれた川戸大誠氏に深く感謝いたします。また、卒論の忙しい時期にまで実験を手伝ってくれた学部4年の松本遼太郎君に深く感謝いたします。

最後になりましたが学生生活を支えてくれた両親をはじめとする家族の皆に心より感謝します。その他、今まで私を見守りご支援いただいたすべての方に心から感謝の意を表し、謝辞とさせていただきます。

付録

軌道計算に用いたコードを付録に示す.

ヘッダファイル

[init.h]

```
#ifndef INIT_H
```

```
#define INIT_H
```

```
#define MPEV 938.E6 // proton mass [eV/c^2]
```

```
#define MP 1.67E-27 // proton mass [kg]
```

```
#define QE 1.60E-19 // elementary charge [C]
```

```
#define VC 2.998E8 // speed of light [m/s]
```

```
#define INIT_B_FILE "Bfield.txt"
```

```
#define INIT_E_FILE "Efield.txt"
```

```
#define GRID_WID 0.0005// grid width for B field [m]
```

```
#define NX 141
```

```
#define NY 121
```

```
#define NZ 11
```

```
// NDIV 毎にイオンの飛跡を出力する
```

```
#define NDIV 55
```

```
// イオン設定
```

```
#define N_ION 10000. // number of ions
```

```
#define ION_Z 1. // 価数
```

```
#define ION_A 40. // イオンの分子量
```

```
// Point source
```

```
#define P_DIV_ANG 0.43// proton beam full divergence angle [degrees]
```

```
#define NTHETA 1 0// theta 方向の個数, phi 方向はコードの中で計算
```

```
// イオンのエネルギーから計算
```

```
#define CALC_WITH_ENERGY
```

```
#ifdef CALC_WITH_ENERGY
```

```
#define ION_E 1000 // ion energy [eV]
```

```

#endif
// 磁場データの座標の最小値
#define X_MIN_B -0 // [m]

// 電場データの座標の最小値
#define X_MIN_E 0.06 // [m]

// 検出器の場所
#define DETECT_X 0.13 // detector position [m]

// particle information
typedef struct __particle {
    int n; // particle number
    int stop; // stopped by the coil holder
    double dt;
    double x[3];
    double v[3];
} particle;

// information of each grid
typedef struct __Bgrid {
    int n;
    double x[3];
    double B[3];
} Bgrid;

// information of each grid
typedef struct __Egrid {
    int n;
    double x[3];
    double E[3];
} Egrid;

// Bfield information
typedef struct __Bfield {
    int ngrid;
    int nx;
    int ny;

```

```

        int nz;
        double min_x;
        double max_x;
        double min_y;
        double max_y;
        double min_z;
        double max_z;
        Bgrid **bgrid;
    } Bfield;

// Efield information
typedef struct __Efield {
    int ngrid;
    int nx;
    int ny;
    int nz;
    double min_x;
    double max_x;
    double min_y;
    double max_y;
    double min_z;
    double max_z;
    Egrid **egrid;
} Efield;

particle **init_particles_col(void);
particle **init_particles_point_source(void);
Bfield *init_Bfield(void);
Efield *init_Efield(void);

void delete_particles(particle **ion);
void delete_Bfield(Bfield *bfield);
void delete_Efield(Efield *bfield);

#endif

[particles.h]
#ifdef PARTICLES_H

```

```

#define PARTICLES_H

int Bcalc(Bfield *bfield, double *x, double *B);
int Ecalc(Efield *efield, double *x, double *E);
int Bcalc(Bfield *bfield, double *x, double *B);
int UpdateParticleB(particle *ion, double *B);
int UpdateParticleE(particle *ion, double *B);
int FindMaxX(particle **ion, double *max);
int MoveParticleToDetector(particle *ion);
int MoveParticleToBEdge(particle *ion);
int MoveParticleToEEdge(particle *ion);

```

```

#endif

```

[propagate.h]

```

#ifndef PROPAGATE_H
#define PROPAGATE_H
#endif

```

ソースファイル

[init.cc]

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include <float>
#include "init.h"

```

```

/**

```

```

 * 粒子の座標、速度の初期化

```

```

 */

```

```

particle **init_particles_point_source(void)

```

```

{

```

```

    particle **ion = (particle**)malloc(N_ION * sizeof(particle *));

```

```

    int nphi[NTHETA];

```

```

    // 1 度あたりの角度

```

```

double dtheta = M_PI*0.5*P_DIV_ANG/180. / NTHETA;

// 1 粒子あたりの面積
double dS = 2*M_PI * (1 - cos(M_PI*0.5*P_DIV_ANG/180.)) / N_ION;

int ntot = 0;
int ion_num = 0;
for (int i = 1; i <= NTHETA; i++) {
    nphi[i-1] = (int)(2*M_PI*sin(i*dtheta)*dtheta / dS);
//    printf("[%d] nphi : %d¥n", i, nphi[i-1]);
    ntot += nphi[i-1];
    if (i == NTHETA && ntot > N_ION) {
//        printf("total: %d¥n", ntot);
        nphi[NTHETA-1] -= (ntot - N_ION);
    }
    double dphi = 2*M_PI / nphi[i-1];
    double theta = dtheta * i;

    for (int j = 0; j < nphi[i-1]; j++) {
        double phi = dphi * j;
        ion[ion_num] = (particle*)malloc(sizeof(particle));
        ion[ion_num]->n = ion_num;

#ifdef CALC_WITH_ENERGY
        ion[ion_num]->v[0] = VC * sqrt(2*ION_E / (MPEV*ION_A)) *
cos(theta);

        printf("v0: %g¥n", ion[ion_num]->v[0]);
        ion[ion_num]->v[1] = VC * sqrt(2*ION_E / (MPEV*ION_A)) *
sin(theta)*cos(phi);

        ion[ion_num]->v[2] = VC * sqrt(2*ION_E / (MPEV*ION_A)) *
sin(theta)*sin(phi);

        ion[ion_num]->dt = 0.5*GRID_WID / ion[ion_num]->v[0];
#else
        ion[ion_num]->v[0] = ION_V * cos(theta);
        ion[ion_num]->v[1] = ION_V * sin(theta)*cos(phi);
        ion[ion_num]->v[2] = ION_V * sin(theta)*sin(phi);
        ion[ion_num]->dt = 0.5*GRID_WID / ion[ion_num]->v[0];
#endif
    }
}
#endif

```

```

        ion[ion_num]->stop = 0;

        ion[ion_num]->x[0] = -SRC_DIST;
        ion[ion_num]->x[1] = 0.;
        ion[ion_num]->x[2] = 0.;

        ion_num++;
        if (ion_num > N_ION) break;
    }
}

return ion;
}

Bfield *init_Bfield(void)
{
    char line[1024];
    int lineCount = 0;
    FILE *fp = fopen(INIT_B_FILE, "r");
    while (fgets(line, sizeof(line), fp) != NULL) {
        lineCount++;
    }
    rewind(fp);
    printf("lineCount: %d¥n", lineCount);
    Bfield *bfield = (Bfield *)malloc(sizeof(Bfield));
    bfield->ngrid = lineCount;
    bfield->min_x = DBL_MAX;
    bfield->min_y = DBL_MAX;
    bfield->min_z = DBL_MAX;
    bfield->max_x = DBL_MIN;
    bfield->max_y = DBL_MIN;
    bfield->max_z = DBL_MIN;
    bfield->bgrid = (Bgrid **)malloc(lineCount*sizeof(Bgrid *));
    int i = 0;
    while (fgets(line, sizeof(line), fp) != NULL) {
        bfield->bgrid[i] = (Bgrid *)malloc(sizeof(Bgrid));
        bfield->bgrid[i]->x[0] = strtod(strtok(line, " ¥t"), NULL);
        bfield->bgrid[i]->x[1] = strtod(strtok(NULL, " ¥t"), NULL);
        bfield->bgrid[i]->x[2] = strtod(strtok(NULL, " ¥t"), NULL);
    }
}

```

```

        bfield->bgrid[i]->B[0] = strtod(strtok(NULL, " \t"), NULL);
        bfield->bgrid[i]->B[1] = strtod(strtok(NULL, " \t"), NULL);
        bfield->bgrid[i]->B[2] = strtod(strtok(NULL, " \t"), NULL);
        //bfield->bgrid[i]->B[0] = 0;
        //bfield->bgrid[i]->B[1] = 0;
        //bfield->bgrid[i]->B[2] = 0;
        bfield->bgrid[i]->n = i;

        // find min, max of x, y, z
        if (bfield->max_x < bfield->bgrid[i]->x[0]) bfield->max_x = bfield->bgrid[i]-
>x[0];

        if (bfield->min_x > bfield->bgrid[i]->x[0]) bfield->min_x = bfield->bgrid[i]-
>x[0];

        if (bfield->max_y < bfield->bgrid[i]->x[1]) bfield->max_y = bfield->bgrid[i]-
>x[1];

        if (bfield->min_y > bfield->bgrid[i]->x[1]) bfield->min_y = bfield->bgrid[i]-
>x[1];

        if (bfield->max_z < bfield->bgrid[i]->x[2]) bfield->max_z = bfield->bgrid[i]-
>x[2];

        if (bfield->min_z > bfield->bgrid[i]->x[2]) bfield->min_z = bfield->bgrid[i]-
>x[2];

        i++;
    }
    bfield->nx = NX;
    bfield->ny = NY;
    bfield->nz = NZ;
    return bfield;
}

Efield *init_Efield(void)
{
    char line[1024];
    int lineCount = 0;
    FILE *fp = fopen(INIT_E_FILE, "r");
    while (fgets(line, sizeof(line), fp) != NULL) {
        lineCount++;
    }
    rewind(fp);

```

```

printf("lineCount: %d\n", lineCount);
Efield *efield = (Efield *)malloc(sizeof(Efield));
efield->ngrid = lineCount;
efield->min_x = DBL_MAX;
efield->min_y = DBL_MAX;
efield->min_z = DBL_MAX;
efield->max_x = DBL_MIN;
efield->max_y = DBL_MIN;
efield->max_z = DBL_MIN;
efield->egrid = (Egrid **)malloc(lineCount*sizeof(Egrid *));
int i = 0;
while (fgets(line, sizeof(line), fp) != NULL) {
    efield->egrid[i] = (Egrid *)malloc(sizeof(Egrid));
    efield->egrid[i]->x[0] = strtod(strtok(line, " \t"), NULL);
    efield->egrid[i]->x[1] = strtod(strtok(NULL, " \t"), NULL);
    efield->egrid[i]->x[2] = strtod(strtok(NULL, " \t"), NULL);
    efield->egrid[i]->E[0] = strtod(strtok(NULL, " \t"), NULL);
    efield->egrid[i]->E[1] = strtod(strtok(NULL, " \t"), NULL);
    efield->egrid[i]->E[2] = strtod(strtok(NULL, " \t"), NULL);
    //efield->egrid[i]->E[0] = 0;
    //efield->egrid[i]->E[1] = 0;
    //efield->egrid[i]->E[2] = 0;
    efield->egrid[i]->n = i;

    // find min, max of x, y, z
    if (efield->max_x < efield->egrid[i]->x[0]) efield->max_x = efield->egrid[i]-
>x[0];
    if (efield->min_x > efield->egrid[i]->x[0]) efield->min_x = efield->egrid[i]-
>x[0];
    if (efield->max_y < efield->egrid[i]->x[1]) efield->max_y = efield->egrid[i]-
>x[1];
    if (efield->min_y > efield->egrid[i]->x[1]) efield->min_y = efield->egrid[i]-
>x[1];
    if (efield->max_z < efield->egrid[i]->x[2]) efield->max_z = efield->egrid[i]-
>x[2];
    if (efield->min_z > efield->egrid[i]->x[2]) efield->min_z = efield->egrid[i]-
>x[2];

```

```

        i++;
    }
    efield->nx = NX;
    efield->ny = NY;
    efield->nz = NZ;
    return efield;
}

/**
 * 粒子用メモリの削除
 */
void delete_particles(particle **ion)
{
    for (int i = 0; i < N_ION; i++) {
        free(ion[i]);
    }
    free(ion);
}

/**
 * B field 用メモリの削除
 */
void delete_Bfield(Bfield *bfield)
{
    for (int i = 0; i < bfield->ngrid; i++) {
        free(bfield->bgrid[i]);
    }
    free(bfield->bgrid);
    free(bfield);
}

/**
 * E field 用メモリの削除
 */
void delete_Efield(Efield *efield)
{
    for (int i = 0; i < efield->ngrid; i++) {
        free(efield->egrid[i]);
    }
}

```

```

    }
    free(efield->egrid);
    free(efield);
}

```

main.cc

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "init.h"
#include "particles.h"

int main(void)
{
    int i;
    /***** Initialization *****/
    // Initialize particles
    particle **ion = init_particles_point_source();
    // Initialize B,E field
    Bfield *bfield = init_Bfield();
    Efield *efield = init_Efield();
    int nstep = 0;
    FILE *ftra = fopen("ion_trajectory.txt", "w");
    for (i = 0; i < N_ION; i++) {
        // 粒子位置を出力
        if (i % NDIV == 0) {
            fprintf(ftra, "%g %g %g ", ion[i]->x[0], ion[i]->x[1], ion[i]->x[2]);
        }
    }
    fprintf(ftra, "\n");
    nstep++;
    int ret = 0;
    for (i = 0; i < N_ION; i++) {
        ret = MoveParticleToBEdge(ion[i]);
        if (ret) {
            // 粒子位置を出力
            if (i % NDIV == 0) {

```

```

        fprintf(fttra, "%g %g %g ", ion[i]->x[0], ion[i]->x[1], ion[i]-
>x[2]);

        }

    }

    if (ret) {
        fprintf(fttra, "%n");
        nstep++;
    }

    /***** 粒子の軌道計算 *****/
    double B[3] = {0, 0, 0};
    double E[3] = {0, 0, 0};
    double v[3];
    double xmax = bfield->min_x + 0.5 * GRID_WID;
    int cnt = 0;

    // 磁場データがある範囲を伝搬
    while (xmax <= bfield->max_x) {
        cnt = 0;

        for (i = 0; i < N_ION; i++) {

            if (ion[i]->stop) {
                if (i % NDIV == 0) {
                    fprintf(fttra, "%g %g %g ", ion[i]->x[0], ion[i]->x[1],
ion[i]->x[2]);

                    cnt++;
                    if (i == 0) {
                        nstep++;
                    }
                }
                continue;
            }

            // 粒子位置の磁場を計算
            Bcalc(bfield, ion[i]->x, B);

```

```

        // 粒子位置、速度更新
        UpdateParticleB(ion[i], B);
        FindMaxX(ion, &xmax);
        /*
        // 粒子位置を出力
        if (i % NDIV == 0) {
            fprintf(fttra, "%g %g %g ", ion[i]->x[0], ion[i]->x[1], ion[i]-
>x[2]);

            cnt++;
            if (i == 0) {
                nstep++;
            }
        }

        printf("cnt: %d¥n", cnt);
        fprintf(fttra, "¥n");
        printf("maxx: %g¥n", xmax);
    }
    // この後は電場領域まで等速運動
    for (i = 0; i < N_ION; i++) {
        ret = MoveParticleToEEdge(ion[i]);
        if (ret) {
            // 粒子位置を出力
            if (i % NDIV == 0) {
                fprintf(fttra, "%g %g %g ", ion[i]->x[0], ion[i]->x[1], ion[i]-
>x[2]);

            }
        }
    }
    if (ret) {
        fprintf(fttra, "¥n");
        nstep++;
    }
    // 電場領域の計算
    while (xmax <= efield->max_x) {
        cnt = 0;
        for (i = 0; i < N_ION; i++) {
            if (ion[i]->stop) {

```

```

        if (i % NDIV == 0) {
            fprintf(ftra, "%g %g %g ", ion[i]->x[0], ion[i]->x[1],
ion[i]->x[2]);

            cnt++;
            if (i == 0) {
                nstep++;
            }
        }
        continue;
    }

    // 粒子位置の磁場を計算
    Ecalc(efield, ion[i]->x, E);
    //printf("E: %g %g %g\n", E[0], E[1], E[2]);

    // 粒子位置、速度更新
    UpdateParticleE(ion[i], E);
    FindMaxX(ion, &xmax);

    // 粒子位置を出力
    if (i % NDIV == 0) {
        fprintf(ftra, "%g %g %g ", ion[i]->x[0], ion[i]->x[1], ion[i]-
>x[2]);

        cnt++;
        if (i == 0) {
            nstep++;
        }
    }

    }
    fprintf(ftra, "\n");
}

// この後はディテクタまで等速運動
for (i = 0; i < N_ION; i++) {
    if (!ion[i]->stop) {
        MoveParticleToDetector(ion[i]);
    }
}

```

```

        // 粒子位置を出力
        if (i % NDIV == 0) {
            fprintf(fttra, "%g %g %g ", ion[i]->x[0], ion[i]->x[1], ion[i]->x[2]);
            if (i == 0) {
                nstep++;
            }
        }
    }
    fprintf(fttra, "\n");
    FILE *fout = fopen("screen_ion.txt", "w");
    int stopped = 0;
    for (i = 0; i < N_ION; i++) {
        if (!ion[i]->stop) {
            fprintf(fout, "%g %g %g\n", ion[i]->x[0], ion[i]->x[1], ion[i]->x[2]);
        } else {
            stopped++;
        }
    }
    fclose(fout);
    fclose(fttra);

    // データ情報ファイル作成
    FILE *finf = fopen("inf.txt", "w");
    fprintf(finf, "%d %d\n", cnt, nstep);
    printf("%d %d\n", cnt, nstep);
    fclose(finf);
    /*
    Bcalc(bfield, x, B);
    */

    /***** メモリ解放 *****/
    delete_particles(ion);
    delete_Bfield(bfield);
    delete_Efield(efield);
}

particles.cc
#include <stdio.h>
#include <stdlib.h>

```

```

#include <math.h>
#include "init.h"
#include "particles.h"

//
// (x, y, z) = (0, 0, 0), (1, 0, 0), (0, 1, 0), (1, 1, 0), (0, 0, 1), (1, 0, 1), (0, 1, 1), (1, 1, 1)
// n = 0, 1, 2, 3, 4, 5, 6, 7
//
int Bcalc(Bfield *bfield, double *x, double *B)
{
    // Find the nearest grids(磁場)
    int ix = (int)floor((x[0] - bfield->min_x) / GRID_WID);
    int iy = (int)floor((x[1] - bfield->min_y) / GRID_WID);
    int iz = (int)floor((x[2] - bfield->min_z) / GRID_WID);
    int nx = bfield->nx;
    int ny = bfield->ny;
    int nz = bfield->nz;
    if (ix >= bfield->nx-1 || iy >= bfield->ny-1 || iz >= bfield->nz-1
        || ix < 0 || iy < 0 || iz < 0) {
        B[0] = 0;
        B[1] = 0;
        B[2] = 0;
        return 0;
    }
    int n[8] = {-1, -1, -1, -1, -1, -1, -1, -1};
    n[0] = ix + nx * iy + nx * ny * iz;
    n[1] = ix + 1 + nx * iy + nx * ny * iz;
    n[2] = ix + nx * (iy + 1) + nx * ny * iz;
    n[3] = ix + 1 + nx * (iy + 1) + nx * ny * iz;
    n[4] = ix + nx * iy + nx * ny * (iz + 1);
    n[5] = ix + 1 + nx * iy + nx * ny * (iz + 1);
    n[6] = ix + nx * (iy + 1) + nx * ny * (iz + 1);
    n[7] = ix + 1 + nx * (iy + 1) + nx * ny * (iz + 1);
    for (int i = 0; i < 8; i++) {
        if (n[i] > NX*NY*NZ-1) {
            printf("n[%d] : %d¥n", i, n[i]);
            printf("ix, iy, iz: %d, %d, %d¥n", ix, iy, iz);
            exit(1);
        }
    }
}

```

```

    }
}

double vol[8];
int i = 0;
// 頂点の座標
double gridx[8][3];
for (i = 0; i < 8; i++) {
    gridx[i][0] = bfield->bgrid[n[i]]->x[0];
    gridx[i][1] = bfield->bgrid[n[i]]->x[1];
    gridx[i][2] = bfield->bgrid[n[i]]->x[2];
}

double x0 = bfield->bgrid[n[0]]->x[0];
double y0 = bfield->bgrid[n[0]]->x[1];
double z0 = bfield->bgrid[n[0]]->x[2];
double x1 = bfield->bgrid[n[7]]->x[0];
double y1 = bfield->bgrid[n[7]]->x[1];
double z1 = bfield->bgrid[n[7]]->x[2];
vol[0] = (x[0] - x0) * (x[1] - y0) * (x[2] - z0);
vol[1] = -(x[0] - x1) * (x[1] - y0) * (x[2] - z0);
vol[2] = -(x[0] - x0) * (x[1] - y1) * (x[2] - z0);
vol[3] = (x[0] - x1) * (x[1] - y1) * (x[2] - z0);
vol[4] = -(x[0] - x0) * (x[1] - y0) * (x[2] - z1);
vol[5] = (x[0] - x1) * (x[1] - y0) * (x[2] - z1);
vol[6] = (x[0] - x0) * (x[1] - y1) * (x[2] - z1);
vol[7] = -(x[0] - x1) * (x[1] - y1) * (x[2] - z1);
//
double tot = 0;
for (i = 0; i < 8; i++) {
    vol[i] /= (GRID_WID*GRID_WID*GRID_WID);
}
B[0] = 0;
B[1] = 0;
B[2] = 0;
for (i = 0; i < 8; i++) {
    B[0] += bfield->bgrid[n[i]]->B[0] * vol[7-i];
    B[1] += bfield->bgrid[n[i]]->B[1] * vol[7-i];
    B[2] += bfield->bgrid[n[i]]->B[2] * vol[7-i];
}
return 1;

```

```
}
```

```
int Ecalc(Efield *efield, double *x, double *E)
{
    // Find the nearest grids(電場)
    int ix = (int)floor((x[0] - efield->min_x) / GRID_WID);
    int iy = (int)floor((x[1] - efield->min_y) / GRID_WID);
    int iz = (int)floor((x[2] - efield->min_z) / GRID_WID);
    int nx = efield->nx;
    int ny = efield->ny;
    int nz = efield->nz;
    printf("%s: i: %d, %d, %d¥n", __FUNCTION__, ix, iy, iz);
    printf("minx, miny, minz: %g, %g, %g¥n", efield->min_x, efield->min_y, efield->min_z);
    printf("x, y, z: %g, %g, %g¥n", x[0], x[1], x[2]);
    if (ix >= efield->nx-1 || iy >= efield->ny-1 || iz >= efield->nz-1
        || ix < 0 || iy < 0 || iz < 0) {
        E[0] = 0;
        E[1] = 0;
        E[2] = 0;
        return 0;
    }
    int n[8] = {-1, -1, -1, -1, -1, -1, -1, -1};
    n[0] = ix + nx * iy + nx * ny * iz;
    n[1] = ix + 1 + nx * iy + nx * ny * iz;
    n[2] = ix + nx * (iy + 1) + nx * ny * iz;
    n[3] = ix + 1 + nx * (iy + 1) + nx * ny * iz;
    n[4] = ix + nx * iy + nx * ny * (iz + 1);
    n[5] = ix + 1 + nx * iy + nx * ny * (iz + 1);
    n[6] = ix + nx * (iy + 1) + nx * ny * (iz + 1);
    n[7] = ix + 1 + nx * (iy + 1) + nx * ny * (iz + 1);
    for (int i = 0; i < 8; i++) {
        if (n[i] > NX*NY*NZ-1) {
            printf("n[%d] : %d¥n", i, n[i]);
            printf("ix, iy, iz: %d, %d, %d¥n", ix, iy, iz);
            exit(1);
        }
    }
}
```

```

double vol[8];
int i = 0;
// 頂点の座標
double gridx[8][3];
for (i = 0; i < 8; i++) {
    gridx[i][0] = efield->egrid[n[i]]->x[0];
    gridx[i][1] = efield->egrid[n[i]]->x[1];
    gridx[i][2] = efield->egrid[n[i]]->x[2];
}

double x0 = efield->egrid[n[0]]->x[0];
double y0 = efield->egrid[n[0]]->x[1];
double z0 = efield->egrid[n[0]]->x[2];
double x1 = efield->egrid[n[7]]->x[0];
double y1 = efield->egrid[n[7]]->x[1];
double z1 = efield->egrid[n[7]]->x[2];
vol[0] = (x[0] - x0) * (x[1] - y0) * (x[2] - z0);
vol[1] = -(x[0] - x1) * (x[1] - y0) * (x[2] - z0);
vol[2] = -(x[0] - x0) * (x[1] - y1) * (x[2] - z0);
vol[3] = (x[0] - x1) * (x[1] - y1) * (x[2] - z0);
vol[4] = -(x[0] - x0) * (x[1] - y0) * (x[2] - z1);
vol[5] = (x[0] - x1) * (x[1] - y0) * (x[2] - z1);
vol[6] = (x[0] - x0) * (x[1] - y1) * (x[2] - z1);
vol[7] = -(x[0] - x1) * (x[1] - y1) * (x[2] - z1);
//
double tot = 0;
for (i = 0; i < 8; i++) {
    vol[i] /= (GRID_WID*GRID_WID*GRID_WID);
}
E[0] = 0;
E[1] = 0;
E[2] = 0;
for (i = 0; i < 8; i++) {
    E[0] += efield->egrid[n[i]]->E[0] * vol[7-i];
    E[1] += efield->egrid[n[i]]->E[1] * vol[7-i];
    E[2] += efield->egrid[n[i]]->E[2] * vol[7-i];
}
return 1;
}

```

```

int UpdateParticleB(particle *ion, double *B)
{
    double dv[3];
    double v[3];
    for (int i = 0; i < 3; i++) {
        v[i] = ion->v[i];
    }
    // dv = q (v x B) / m
    dv[0] = QE*ION_Z * (v[1]*B[2] - v[2]*B[1]) * ion->dt / (MP*ION_A);
    dv[1] = QE*ION_Z * (v[2]*B[0] - v[0]*B[2]) * ion->dt / (MP*ION_A);
    dv[2] = QE*ION_Z * (v[0]*B[1] - v[1]*B[0]) * ion->dt / (MP*ION_A);
    for (int i = 0; i < 3; i++) {
        ion->v[i] += dv[i];
        ion->x[i] += v[i] * ion->dt;
    }
    return 1;
}

```

```

int UpdateParticleE(particle *ion, double *E)
{
    double dv[3];
    double v[3];
    for (int i = 0; i < 3; i++) {
        v[i] = ion->v[i];
    }
    // dv = qE / m *dt
    dv[0] = QE*ION_Z * E[0] * ion->dt / (MP*ION_A);
    dv[1] = QE*ION_Z * E[1] * ion->dt / (MP*ION_A);
    dv[2] = QE*ION_Z * E[2] * ion->dt / (MP*ION_A);
    for (int i = 0; i < 3; i++) {
        ion->v[i] += dv[i];
        ion->x[i] += v[i] * ion->dt;
    }
    return 1;
}

```

```

int FindMaxX(particle **ion, double *max)
{

```

```

        *max = -10000;
        int maxn = -10000;
        for (int i = 0; i < N_ION; i++) {
            if (*max < ion[i]->x[0]) {
                *max = ion[i]->x[0];
                maxn = ion[i]->n;
            }
        }
        return maxn;
    }

    int MoveParticleToDetector(particle *ion)
    {
        double time = (DETECT_X - ion->x[0]) / ion->v[0];
        for (int i = 0; i < 3; i++) {
            ion->x[i] += ion->v[i] * time;
        }
        return 1;
    }

    int MoveParticleToBEdge(particle *ion)
    {
        printf("MoveParticleToBedge() start¥n");
        if (X_MIN_B < ion->x[0]) return 0;
        double time = (X_MIN_B - ion->x[0]) / ion->v[0];
        printf("time: %g¥n", time);
        printf("%g - %g / %g¥n", X_MIN_B, ion->x[0], ion->v[0]);
        for (int i = 0; i < 3; i++) {
            ion->x[i] += ion->v[i] * time;
        }
        printf("MoveParticleToBedge() end¥n");
        return 1;
    }

    int MoveParticleToEEdge(particle *ion)
    {
        printf("MoveParticleToEEdge() start¥n");
        if (X_MIN_E < ion->x[0]) return 0;
        double time = (X_MIN_E - ion->x[0]) / ion->v[0];
        for (int i = 0; i < 3; i++) {

```

```

        ion->x[i] += ion->v[i] * time;
    }
    printf("MoveParticleToEEdge() end¥n");
    return 1;
}

```

```

[propagate.cc]
#include "propagate.h"

```