

東京大学 大学院新領域創成科学研究科  
基盤科学研究系  
先端エネルギー工学専攻

平成19年度  
修士論文

希薄気体における形状近似DSMC法と  
遺伝的アルゴリズムを用いた飛行体形状最適化

2008年2月提出  
指導教員 鈴木 宏二郎 准教授

66203 板橋 直亮

# 目次

第 1 章 序論	2
1.1 背景及び目的	2
第 2 章 形状近似 DSMC 法	5
2.1 DSMC 法の概要	5
2.1.1 モンテカルロ直接法	10
2.1.2 分子境界条件	14
2.2 DSMC 法計算の流れ	16
2.3 解適合格子と形状近似	17
2.3.1 解適合格子のアルゴリズム	17
2.3.2 分子の移動, 反射	22
第 3 章 形状近似 DSMC 法の検証	29
3.1 形状近似 DSMC 法	29
3.1.1 平板周りの流れ	29
3.1.2 円柱周りの流れ	38
第 4 章 形状近似 DSMC 法による希薄気体中の飛行体形状最適化	42
4.1 遺伝的アルゴリズム	42
4.2 エアロキャプチャ概要及び最適化条件	44
4.3 最適化	52
第 5 章 結言	64

# 第1章 序論

## 1.1 背景及び目的

我々が日常経験する気体の振る舞いは、ナビエ・ストークス方程で表されるいわゆる流体力学によってよく記述される。しかし、航空宇宙工学などで重要な低圧気体、あるいはマイクロマシンやエアロゾルで問題となる小さい系の気体では、気体分子の平均自由行程が系の代表長に比べて無視出来ず、従来の流体力学ではその振る舞いを正しく記述出来ない。このような場合でも、普通は密度、流速などの巨視的変数が滑らかな関数として定義出来る程度に巨視的にも多くの分子が存在し、この点では通常の気体と変わりはない。しかしこの気体の振る舞いを記述するには、密度、温度、及び流速という巨視的物質量だけでは不十分であり、系が種々の速度の分子で構成されていることを表現出来る微視的な取り扱いが必要となる。宇宙工学やナノテクノロジーの発達した現代では、このような希薄な領域となる場合も多くなる。例えば急速に進化を続ける微小電気機械システム (Micro-Electro-Mechanical systems; MEMS) では、構造物が微細なために雰囲気の影響を受けやすい。そのため MEMS の開発において今以上の高性能化を実現するためには、内部の流れを正確にとらえ、流体の及ぼす影響を考慮する必要がある。しかし、MEMS では構造物が非常に微細なために測定を行えるのはシンプルな構造の物に限定される。このような背景からコンピュータシミュレーションの重要性が増してきている。MEMS のようなマイクロスケールの流れでは標準大気状態においても、系の代表長さと分子の平均自由行程とが同程度となるため希薄流解析が必要となる。宇宙工学では、衛星を例としても、低軌道では少なからず抵抗を受けている。それらは長い年月で考えた場合には無視出来ない影響を与える。

今後 MEMS や宇宙開発が更に進むことに疑う余地はない。宇宙工学においては宇宙機の大気圏突入や衛星、エアロブレーキ、エアロキャプチャなど様々な場面で希薄な領域となる。エアロキャプチャとは、惑星到着時に惑星の周回軌道に衛星を投入する技術であり、従来の燃料による減速の代わりに、惑星の大気の空気抵抗を用いて減速を行う、エアロアシストと呼ばれる技術のうちの一つである [3]。エアロキャプチャは大気を持つすべての惑星に適用することができ、1回の突入で数 km/s の減速を行う。その利点としては以下のようなものがあると考えられている。

- パイロードを増大させることにより、探査機の小型化やより多くの観測機器を搭載することが可能.
- 重量削減、探査機の小型化により打ち上げ時に安価なロケットを使用することが可能.
- 一度の大気への突入で周回軌道への投入ができるため、軌道投入に数週間から数ヶ月かかると言われるエアロブレーキよりも短時間でミッションを行うことが可能.
- 軌道投入時の重量面からの  $\Delta V$  の制約が緩和されることにより惑星間移行の際タイミングを図る必要が少なくなりミッションを行える機会が増える.

このようにメリットは多いが突入経路角の許容角度が狭いことが問題となっている. 突入経路角の許容角度を広げるには高い揚抗比が必要となることが知られており、そのためにも空力性能の向上が必要となる. しかし、連続流の領域では旅客機を例にあげても 767 ではマッハ数 0.80, 777 で 0.84, A380 で 0.83, 787 では 0.85 と燃費の向上と共に空力性能を改善し高速化が図られてきたが、希薄気体領域での最適化を行った研究例は少ない. 希薄気体領域での最適化を行った研究例が少ない理由として次の事が上げられる.

- 通常の流体力学で言うところのレイノルズ数が非常に小さい値となるため、粘性の影響が大きく飛行体としては高い L/D を得ることは難しい.
- 希薄であるために形状の変化に対して空力性能の変化は鈍感である.
- 希薄気体の解析に適している direct simulation Monte Carlo (DSMC) 法では計算コストが大きい.
- 最適化を行う事自体の計算コストが大きい.

これらのことから、希薄流の領域での最適化はメリットに対し、計算コストが高く行われてこなかった. しかし、近年のコンピュータの CPU の性能や搭載するメモリ容量は MEMS の発達に伴い、劇的な進化を続けて、従来のパーソナルコンピュータ (PC) では計算コストが大きく、スーパーコンピュータを用いない限りは不可能であった計算が、PC で可能となってきた. 近年の PC はマルチコア CPU を備え、かなりの容量のメモリを搭載した共有メモリシステムへの進化を遂げた. 今後もコア数とメモリ容量は増加の一途をたどると予想される. これらのことから PC を用いて DSMC 法で最適化を行える様になる事は、今後の開発において更なる飛躍の一步となると言える. よって、本研究では

- 形状を近似する事で、あらゆる形状に対して同一の格子での解析を可能とし、計算コードの開発コストの削減

- PC による希薄気体における DSMC 法と遺伝的アルゴリズムを用いた形状最適化
- 最適化の高速化

を目的とし、解適合格子と形状近似 DSMC 法により計算の高速化、同一の格子において様々な形状の対象の流れ場解析を行った。

## 第2章 形状近似DSMC法

この章では希薄流について述べ、希薄流解析に適した DSMC 法の説明を行い、DSMC 法に用いた形状近似方法、解適合格子について述べる。

### 2.1 DSMC 法の概要

高高度を飛行する衛星を考え、その空気力学を調べるために希薄流の解析を行う。希薄流の領域では、流れが連続と見なせないため、ナビエ・ストークス方程式が成立しなくなる。このような場合には、支配方程式としてボルツマン方程式を用いて流体の解析を行わなくてはならない。本研究では、ボルツマン方程式の確率解法として広く知られるモンテカルロ直接法 [4] を用い、以下の仮定を用いて解析を行った。

- 剛体球分子
- 内部自由度は無視（粒子運動は並進のみを考える）
- 化学反応なし

一般に、宇宙空間や高層大気中では流れの圧力が低下し希薄流と見なされる。希薄気体となる条件は、気体中の分子が衝突無しに進める平均距離がシステムの代表長さに対して無視できなくなった場合と定義される。希薄流の解析はボルツマン方程式を基礎式とした分子運動の取り扱いが必要となる。

#### 希薄気体となる条件

希薄流となる条件は一般的にクヌーセン数  $Kn$  によって定義される。クヌーセン数は平均自由行程と流れの代表長さの比で、

$$Kn = \frac{\lambda}{L} \quad (2.1)$$

と定義される。 $L$  は物体の代表長さ、 $\lambda$  は平均自由行程で、剛体球モデルを用いると

$$\lambda = \frac{1}{\sqrt{2}\pi d^2 n} \quad (2.2)$$

表 2.1: 流れの分類

$Kn$	$Kn < 0.01$	$0.01 < Kn < 10$	$10 < Kn$
流れ	連続流	遷移流	自由分子流

となる。ここで、 $d$  は分子直径、 $n$  は分子の数密度（単位体積中の分子数）である。表 2.1 に示すように、 $Kn$  が 0.01 より小さいとき、流れは連続であると見なされ、ナビエ・ストークス方程式を用いて解析することができる。また  $Kn$  が 10 より大きい場合には、流れは自由分子流と見なすことができ自由分子としての解析が可能となる。どちらの条件も満たされない場合、 $Kn$  が 0.01 以上 10 以下の流れはその中間の遷移流れの領域となる。この領域に関して DSMC 法が適用される。

#### 速度分布関数

気体中の分子すべてが同じ速度で運動しているわけではないため、分子運動を取り扱う際にはどの速度を持った分子がどのくらい存在するかを表す速度分布関数という概念が必要となる。位置ベクトルを  $x$  とすると、その点での数密度は  $n(x)$  と書ける。流れ場の一点  $x$  を含む体積  $dx = dx_1 dx_2 dx_3$  の中にある分子の個数は  $n(x)dx$  となる。分子の速度  $c$  の直角座標成分を  $c_1, c_2, c_3$  とり、 $c_1, c_2, c_3$  を座標軸とした速度空間を考えたときに、ある速度空間のベクトル  $c$  の先端に体積  $dc = dc_1 dc_2 dc_3$  の直方体を考える。この中にある分子数は速度分布関数  $f$  を用いて  $fndc$  と書くことができ、 $c \sim c + dc$  の速度を持つ分子の数を表す。 $f$  は速度空間の位置によって変わるため  $c$  の関数であり、速度空間のすべての分子数を加えると全体の分子数に等しくなければならないので、

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} n f dc_1 dc_2 dc_3 = n \quad (2.3)$$

となる。

一般に  $n$  は時間  $t$  によっても変わるので、 $x$  と  $t$  の関数で  $n(x, t)$  となるので、 $n$  が  $x$  と  $t$  に依存すれば、当然  $f$  も  $x$  と  $t$  に依存する。すなわち  $f(c, x, t)$  と書ける。まとめると、 $x(x, t)f(c, x, t)dc dx dt$  は時刻  $t$  で物理空間の点  $x$  を含む体積  $dx$  のなかであり、速度ベクトルが速度空間の点  $c$  を含む体積  $dc$  に入る分子の数である。このような数を支配する方程式が、次のボルツマン方程式である。

$$\frac{\partial n f}{\partial t} + c \frac{\partial n f}{\partial x} = n^2 \iint [f(c', x, t)f(\zeta', x, t) - f(c, x, t)f(\zeta, x, t)] g \sigma d\Omega d\zeta \quad (2.4)$$

ここで速度  $c', \zeta'$  は速度  $c, \zeta$  を持つ一対の分子の衝突後の速度で、これらは衝突前の速度  $c, \zeta$  と 2 つの角  $c, \zeta$  の関数となり、次の式で表される [8].

$$c' = c + \frac{1}{2} [g(1 - \cos \chi + h \sin \chi)] \quad (2.5)$$

$$\zeta' = \zeta - \frac{1}{2} [g(1 - \cos \chi + h \sin \chi)] \quad (2.6)$$

ここで  $g$  は衝突前の相対速度で  $g = \zeta - c$  であり、 $h$  の直角座標成分は

$$h_1 = \sqrt{g^2 - g_1^2} \cos \epsilon \quad (2.7)$$

$$h_2 = -\frac{g_1 g_2 \cos \epsilon + g g_3 \sin \epsilon}{\sqrt{g^2 - g_1^2}} \quad (2.8)$$

$$h_3 = -\frac{g_1 g_3 \cos \epsilon + g g_2 \sin \epsilon}{\sqrt{g^2 - g_1^2}} \quad (2.9)$$

となる.  $\chi$  は  $g$  と  $g'$  のなす角であり、分子間に斥力が働くときは  $\chi$  の範囲は  $0 \sim \pi$  である. 重心系でみると 2 つの分子はともに軌道を  $\chi$  だけ曲げられる.  $\epsilon$  は分子の軌道が描かれる平面 (衝突面) を規定する角度で、範囲は  $0 \sim 2\pi$  である. また立体角  $d\Omega$  は  $\sin \chi d\chi d\epsilon$  を表す.

(2.4) の  $\sigma d\Omega$  は微分断面積であり、 $\sigma$  は

$$\sigma = \frac{b}{\sin \chi} \left| \frac{\partial b}{\partial \chi} \right| \quad (2.10)$$

で定義され、 $b$  は衝突係数と呼ばれる. 一般に  $\chi$  は  $b$  と  $g$  の関数である. 関数形は 2 つの分子間に働く力の法則によって決まり、 $g$  が増すと、分子間の働く時間が短くなるため  $\chi$  は小さくなる. 図に示す様に直径  $d$  の剛体分子球が衝突する場合には

$$\chi = \pi - 2\psi, b = \sin \psi \quad (2.11)$$

となるので式 (2.10) は

$$\sigma = \frac{d^2}{4} \quad (2.12)$$

となる.

### 流れ場の諸量

$nf$  を  $F$  とおくとボルツマン方程式から  $n$  が消え、 $F$  に関する式になる. 今  $F(c, x, t)$  が求まったとすると、このとき、流れ場の数密度  $n$ 、流速  $v$ 、温度  $T$  は以下の式から求まる.

$$n = \int F dc \quad (2.13)$$



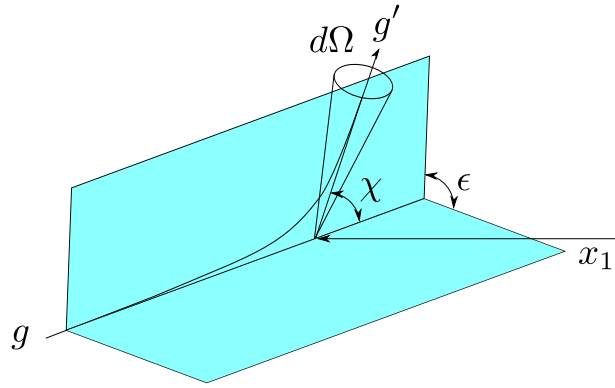


図 2.1: 衝突角と立体角  $d\Omega$

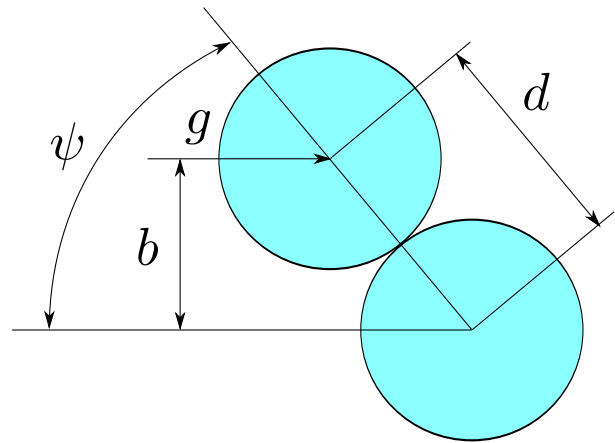


図 2.2: 剛体球分子の衝突係数  $b$

密度  $\rho$  は  $\rho = mn$ , また  $f = F/n$  を用いて

$$V = \int c f dc \quad (2.14)$$

$$T = \frac{1}{3R} \left( \int c^2 f dc - V^2 \right) \quad (2.15)$$

$F$  がわかれば流れ場は決定される. 分子の速度  $c$  から  $V$  を差し引いた

$$C = c - V \quad (2.16)$$

を熱速度といい, 流れからみた分子のランダムな速度である. これを用いると式 (2.15) は

$$T = \frac{1}{3R} \int C^2 f dc \quad (2.17)$$

とも表される. すなわち  $V$  は分子の平均速度であり,  $T$  は分子の乱雑さを表す.

#### マックスウェル分布

平衡状態では速度分布関数  $f$  は  $t$  にも  $x$  にも依存しない. このときのボルツマン方程式の解はマックスウェル分布となり  $V, T$  を一定とした場合には

$$f(c) = \frac{1}{(2\pi RT)^{3/2}} \exp \left[ -\frac{(c - V)^2}{2RT} \right] \quad (2.18)$$

として与えられる. 熱速度の大きさ  $C$  が  $C \sim C + dC$  の間にある分子の数を  $n\chi(C)dC$  とおくと, 平衡状態では

$$\chi(C) = \frac{1}{(2\pi RT)^{3/2}} C^2 \exp \left[ -\frac{C^2}{2RT} \right] \quad (2.19)$$

となり  $\chi(C)$  をスピード分布関数という.  $\chi(C)$  は

$$C_{mp} = \sqrt{2RT} \quad (2.20)$$

で極大値をとる. この  $C_{mp}$  を最確速度という. また  $C$  の平均値は

$$\bar{C} = \int_0^\infty C \chi(C) dC = \sqrt{\frac{8RT}{\pi}} \quad (2.21)$$

$C^2$  の平均値の平方根  $C_{rms} = \sqrt{\overline{C^2}}$  は

$$C_{rms} = \sqrt{3RT} \quad (2.22)$$

となる. 平均自由行程  $\lambda$  を  $\bar{C}$  で除した  $\tau$

$$\tau = \frac{\lambda}{\bar{C}} \quad (2.23)$$

を平均自由時間といい、分子はこの時間だけ他の分子と衝突せずに自由に飛行出来る。

マックスウェル分布をしている分子集団から任意の2個を取り出し、この分子間の相対速度の大きさ  $g = |\zeta - c|$  を調べる。  $g$  が  $g \sim g + dg$  にある確率を  $G(g)dg$  とすると

$$G(g) = \frac{4}{\sqrt{\pi}(4RT)^{3/2}} g^2 \exp\left(-\frac{g^2}{4RT}\right) \quad (2.24)$$

$g$  の平均値は

$$\bar{g} = \int_0^\infty gG(g)dg = 4\sqrt{\frac{RT}{\pi}} \quad (2.25)$$

となり、これは  $\sqrt{2}\bar{C}$  に等しい。  $G(g)$  は  $g_{mp} = 2\sqrt{RT}$  で最大値とる。 確率解法を用いる際に  $g$  の上限  $g_{max}$  が必要となる。 もちろん、厳密にはこのような値は存在しないが

$$g_{max} = 2.5g_{mp} = 5\sqrt{RT} \quad (2.26)$$

と置いても、  $g > g_{max}$  となる確率が無視しうるほど小さいため問題はない。

### 2.1.1 モンテカルロ直接法

モンテカルロ直接法 (DSMC 法) はボルツマン方程式の確率解法で、速度分布関数  $f$  の代わりに  $f$  から抽出したサンプルの集合を用いる解法である。 サンプルの抽出法は一様乱数  $U$  を用いて

$$F(x) = \int_{-\infty}^x f(x')dx' = U \quad (2.27)$$

により無作為にサンプル  $x$  を決定するランダムサンプリングである。

#### 分離の原理

DSMC 法では  $\Delta t \ll 1$  にとると、分子の衝突と移動を分離して考えることができる。  $F = nf$  とおくと

$$F(c, x, \Delta t) = F(c, x, 0) + \Delta t \left( \frac{\partial F}{\partial t} \right)_{t=0} \quad (2.28)$$

$$F(c, x, \Delta t) = (1 - \Delta t D + \Delta t J)F(c, x, 0) \quad (2.29)$$

ただし  $D = c \cdot (\partial/\partial x)$  で、

$$JF = \iint [F(c')F(\zeta') - F(c)F(\zeta)] g\sigma d\Omega d\zeta \quad (2.30)$$

$D$  は線形,  $J$  は非線形演算子である.  $(\Delta t)^2$  はを無視すると式 (2.29) は

$$F(c, x, \Delta t) = (1 - \Delta t D)(1 + \Delta t J)F(c, x, 0) \quad (2.31)$$

となる. したがって  $F(c, x, \Delta t)$  を求めるには

$$F^*(c, x) = (1 + \Delta t J)F(c, x, 0) \quad (2.32)$$

$$F(c, x, \Delta t) = (1 + \Delta t J)F^*(c, x) \quad (2.33)$$

とすればよい. 式 (2.32) は  $\Delta t$  の間におこる分子間衝突による変化を表し, 式 (2.33) はこれらの分子が無衝突で  $x$  から  $x + c\Delta t$  に移動することを表している. この分離の原理が成立するためには,  $\Delta t|JF| \ll F$  かつ  $\Delta t|DF| \ll F$  が必要となる. これらを書き直すと,

$$\Delta t \ll \lambda/\bar{C}, \quad \bar{c}_x \Delta t \leq \Delta x \quad (2.34)$$

となる. ここで  $\bar{c}_x$  は  $c_x$  の代表値である. 最初の条件は  $\Delta t$  は平均自由時間より十分小さくとらなければならないことを表し, 2 番目の条件はクーラン条件であり, 分子が小区間  $\Delta x$  を飛び越えないように  $\Delta t$  を選択しなければならないことを表す.

式 (2.18) は分子の速度  $c$  の 3 次元確率分布であり, 規格化の条件より確率密度の条件を満たしている. 以下  $V = 0$  とする. このときのサンプルの速度を  $c$  とすると  $V \neq 0$  のときのサンプルの速度は  $c + V$  になる.  $f(c)$  は

$$f dc = f_1 dc_1 \cdot f_2 dc_2 \cdot f_3 dc_3 \quad (2.35)$$

$$f_i = \frac{1}{(2\pi R)^3 / 2} \exp\left(-\frac{c_i^2}{2RT}\right) \quad (-\infty < c_i < \infty) \quad (2.36)$$

と書ける ( $i = 1 \sim 3$ ). 3 つの  $f_i$  は同型の確率密度である. ここで変換

$$c_1 = c_r \cos \theta, \quad c_2 = c_r \sin \theta \quad (2.37)$$

を用いて,  $\eta = cr^2$  とおくと

$$f_1 f_2 dc_1 dc_2 = F(\eta) d\eta \cdot G(\theta) d\theta \quad (2.38)$$

ただし,  $F = 1/2RT \cdot \exp(-\eta/2RT)$ ,  $G = 1/2\pi$  である. 関数  $F, G$  はそれぞれ  $\eta = 0 \sim \infty$ ,  $\theta = 0 \sim 2\pi$  で定義された確率密度である. したがって,  $\eta, \theta$  のサンプルは

$$c_r = \sqrt{-2RT \ln U}, \quad \theta = 2\pi U \quad (2.39)$$

と求まる. これを式 (2.37) に代入したものが  $c_1, c_2$  のサンプルとなる.  $c_3$  のサンプルも同様に考えて

$$c_3 = c_r \cos \theta \quad (2.40)$$

この式の  $c_r, \theta$  は式 (2.39) で与えられるが, 式中の  $U$  は  $c_1, c_2$  のサンプルを作ったときの  $U$  とは別のものである. ここで

## 最大衝突法

最大衝突法 [5] は計算時間が少なく効率が良いことで知られていることから、本研究ではこれを用いた。  $N$  分子系で  $\Delta t$  時間に起きる平均衝突回数  $\bar{M}$  は

$$\bar{M} = \frac{n\Delta t}{n} \sum_{i < j} g_{ij} \sigma_T(g_{ij}) \quad (2.41)$$

ここで  $g_{ij} = |c_j - c_i|$  及び、  $\sigma_T$  は全衝突断面積で剛体球分子では式 (2.12) より

$$G_{max} = 5\pi d^2 f \sqrt{RT} \quad (2.42)$$

として、全衝突の平均衝突数 (最大衝突数) は次のようになる。

$$\bar{M}_{max} = \frac{1}{2} n \Delta t (N - 1) G_{max} \quad (2.43)$$

最大衝突法の手順は、次のようになる。

1.  $G_{max}$  を予想し、最大衝突数  $\bar{M}_{max}$  を求める。  $\bar{M}_{max}$  は乱数  $U$  を用いて整数にする。
2.  $N$  個の分子の中から任意の2個の (分子  $i, j$ ) を抽出する。  $U < G_{i,j}/G_{max}$  なら実衝突なので3へ行き、そうでないなら無衝突なので何もしないで全衝突が1回終了する。
3.  $\chi, \epsilon$  のサンプルを抽出し、ランダムな向きをもつ単ベクトル  $R$  をもちいて  $(c_i, c_j)$  を次の  $(c'_i, c'_j)$  に置き換える。

$$c'_i = \frac{1}{2} (c_i + c_j - g_{ij} R) \quad (2.44)$$

$$c'_j = \frac{1}{2} (c_i + c_j + g_{ij} R) \quad (2.45)$$

ここでランダムな向きを持つ単位ベクトル  $R$  の定義は

$$\text{Prob} \{R \in d\Omega\} = \frac{d\Omega}{4} \quad (2.46)$$

$R$  が立体角  $d\Omega$  に含まれる確率は、  $d\Omega$  をどの方向にとるかに依存しない。  $\theta$  を極角にとると  $d\Omega = \sin \theta d\theta d\phi$  であるので、これにより

$$\text{Prob} \{R \in d\Omega\} = f(\theta) d\theta \cdot g(\phi) d\phi \quad (2.47)$$

ただし  $f(\theta) = \sin \theta/2$ ,  $g(\theta) = 1/2\pi$  で、  $f$  は  $\theta = 0 \sim \pi$ ,  $g$  は  $\phi = 0 \sim 2\pi$  で定義された確率密度である。式 (2.27) を  $f, g$  に対して用いると、サンプル  $\theta, \phi$  は

$$\cos \theta = 1 - 2U, \quad \phi = 2\pi U \quad (2.48)$$

となる。  $x_1$  軸を極軸にとると  $R$  の成分は

$$R_1 = \cos \theta, R_2 = \sin \theta \cos \phi, R_3 = \sin \theta \sin \phi \quad (2.49)$$

である。ただし  $0 < \theta < \pi$  より、  $\sin \theta = \sqrt{1 - \cos^2 \theta}$  である。式 (2.48) を式 (2.49) に代入するとランダムな向きを持つ単位ベクトルが得られる。

### 流れ場の決定

流れ場中にある全サンプル分子の座標  $x_i$  と速度  $c_i$  から、流れ場の状態量が決定される。流れ場中に一様流などの状態が既知の領域があれば、ここに基準体積  $v_\infty$  をとり、この中に  $N_\infty$  個のサンプル分子を置く。数密度  $n_\infty$  は  $N_\infty/v_\infty$  に比例する。体積  $v$  の任意のセルに  $N$  個のサンプル分子があれば、数密度  $n$  は  $N/v$  に比例するので、

$$\frac{n}{n_\infty} = \frac{Nv_\infty}{N_\infty v} \quad (2.50)$$

から求められる。あるセルに入っている  $N$  個のサンプル分子の速度を  $c_1, c_2, \dots, c_N$  とすると、このセルでの流速  $V$ 、温度  $T$  は

$$V = \frac{1}{N} \sum_{i=1}^N c_i \quad (2.51)$$

$$T = \frac{N}{3R(N-1)} \left( \frac{1}{N} \sum_{i=1}^N c_i^2 - V^2 \right) \quad (2.52)$$

から求められる。サンプルの分散の期待値を母集団の分散に一致させるために、式 (2.52) に因子  $N/(N-1)$  が付加してある。

### 時間平均

定常流の計算では、  $t = t_\infty$  で流れが定常になった後も計算を続行し、1組の  $c_i^{(0)}$  から出発した  $t = t_s + (m-1)\tau$ , ( $m = 1 \sim M$ ) における解  $c_i^{(m)}$  を求め、次の式からゆらぎの小さい流れ場の狩猟を求める。

$$\frac{n}{n_\infty} = \frac{N_M v_\infty}{M N_0}, \quad V = \frac{A}{N_M}, \quad T = \frac{N_M}{3R(N_M - 1) \left( \frac{B}{N_M} - V^2 \right)} \quad (2.53)$$

$$N_M = \sum_{m=1}^M N^{(m)}, \quad A = \sum_{m=1}^M \sum_{i=1}^{(m)} c_i^{(m)}, \quad B = \sum_{m=1}^M \sum_{i=1}^{N^{(m)}} \left( c_i^{(m)} \right)^2 \quad (2.54)$$

これを時間平均といい、 $\tau$  をサンプリング間隔という。データ間の相関を小さくするため、 $\tau$  は平均自由行程より大きくとる。ゆらぎは  $N_M^{-1/2}$  のオーダーとなり、 $N_M = 10^4$  なら得られた解は十分に実用的であるといえる。

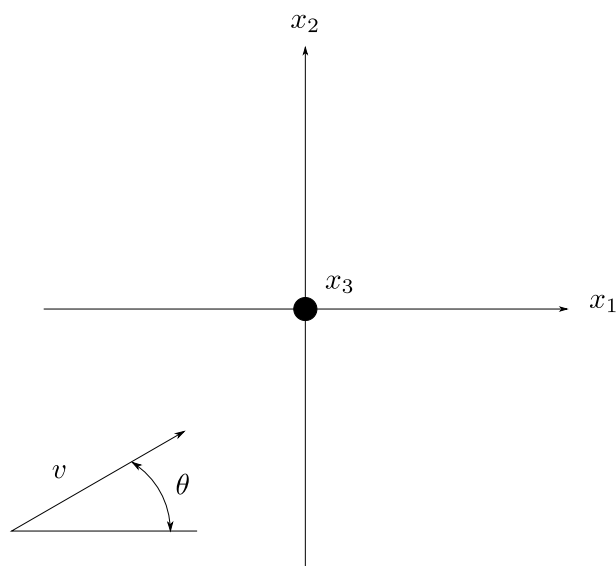


図 2.3:  $x_2x_3$  面への分子の流入

### 2.1.2 分子境界条件

確率解法では，計算領域に流入する分子，壁反射する分子の取り扱いが必要になるので，それらについて示す．

#### 流入分子

単位時間に単位面積に入射する分子の数を数流速という．図 2.3 のような  $x_2x_3$  面に流入する流速  $V$  の平衡流を考える． $x_1$  軸を回転軸として  $x_2x_3$  面を適当に回転すると  $V_3 = 0$  に出来るので， $V_3 = 0$  の場合を考えても一般性を失わない．そこで，

$$v_3 = v \cos \theta, V_2 = v \sin \theta, V_3 = 0 \quad (2.55)$$

とする． $\theta$  は  $v$  と  $x_1$  軸のなす角である．

$x_2x_3$  に流入する数流速は

$$F = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_0^{\infty} f dc_1 dc_2 dc_3 \quad (2.56)$$

で与えられる． $n$  は平衡流の数密度であり，式 (2.18) を代入して積分すると

$$F = n \sqrt{\frac{RT}{2\pi}} K(s \cos \theta) \quad (2.57)$$

ただし， $s = V/\sqrt{2RT}$  は速度比， $K(x)$  は

$$K(x) = \exp(-x^2) + \sqrt{\pi}x(1 + \operatorname{erf}(x)) \quad (2.58)$$

(2.57) の  $F$  は  $x_2x_3$  面を  $x_1$  の正方向に横切る数流速である. これを  $F^+$  とおく. 同様に  $x_2x_3$  面を  $x_1$  の負方向に横切る数流速を  $F^-$  とおくと

$$F^- = n\sqrt{\frac{RT}{2\pi}}K(-s \cos \theta) \quad (2.59)$$

$\text{erf}(x)$  は奇関数なので

$$F^+ - F^- = nV \cos \theta \quad (2.60)$$

これは,  $x_1$  方向への正味の数流速である.

面に入射する分子の速度分布関数 (確率密度)  $f^*$  は式 (2.56) の被積分関数を式 (2.57) で除したものとなり,

$$f^* = \frac{c_1}{2\pi(RT)^2 K(s \cos \theta)} \exp \left[ -\frac{(c_1 - V_1)^2 + (c_2 - V_2)^2 + c_3^2}{2RT} \right] \quad (2.61)$$

ただし,  $c_2 > 0$ ,  $\theta = 0$  の場合には

$$f^* = f_{\parallel}^* f_{\perp}^* \quad (2.62)$$

$$f_{\parallel}^* = \frac{1}{RTK(s)} c_1 \exp \left[ -\frac{(c_1 - V)^2}{2RT} \right] \quad (2.63)$$

$$f_{\perp}^* = \frac{1}{2\pi RT} c_1 \exp \left[ -\frac{c_2^2 + c_3^2}{2RT} \right] \quad (2.64)$$

式 (2.63) は  $c_1$  の, 式 (2.64) は  $c_2, c_3$  の確率密度を表す.

確率密度が式 (2.62) で与えられる場合,  $c_2, c_3$  のサンプルは式 (2.37) と同じになる. 式 (2.63) から  $c_1$  のサンプルを棄却法により決定する.  $f_{\parallel}^*$  は

$$c_1 = \frac{\sqrt{V^2 + 4RT} + V}{2} \quad (2.65)$$

で最大値をとる.  $f_{\parallel}^*$  を最大値で除したものを  $F^*$  とおくと,  $F^*$  は 1 より小さい.  $F^*(c_1)$  を大きくするほど  $c_1$  ほどサンプルになりやすい. また,  $c_1 \rightarrow \infty$  で  $f_{\parallel}^*$  なので,  $\epsilon$  を十分小さい数として

$$\int_{c_{\infty}}^{\infty} f_{\parallel}^* dc_1 < \epsilon \quad (2.66)$$

となるように  $c_1$  に上限  $c_{\infty}$  を設け,  $c_1$  のサンプルを  $0 \sim c_{\infty}$  の間から探す.

## 分子の反射

分子は壁で反射される. 固体面での分子の反射模型を図 2.4 に示す.

鏡面反射では, 壁に垂直方向の速度成分のみに反射後に符号が変わる.

$$c_1' = -c_1, \quad c_2' = c_2, \quad c_3' = c_3 \quad (2.67)$$



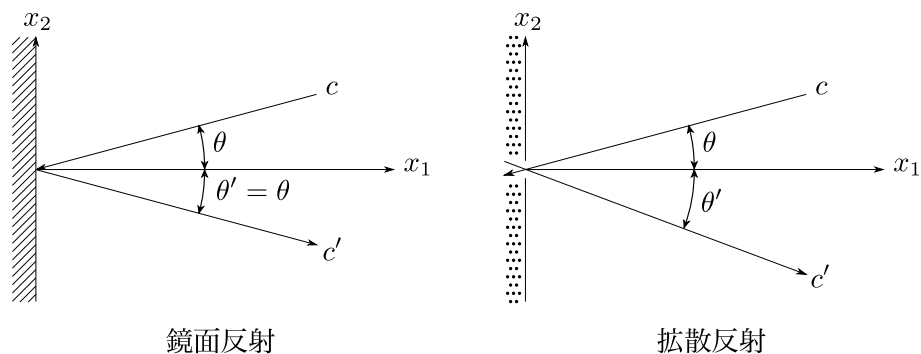


図 2.4: 個体面での分子の反射模型

この条件は境界で対象な流れが生じている場合に使用する。

膜面上での分子の反射には拡散反射の条件を用いた。拡散反射では、反射直後の速度  $c'$  は入射速度  $c$  に無関係で、壁の温度  $T$  に依存する。この模型では、壁の左側に経気温と同じ温度の仮想の平衡流があると考え、反射後の速度は仮想気体中の 1 つの分子が仮想気体中から壁に入射する速度なので、その速度分布関数は式 (2.62) で与えられる。 $c_1$  のサンプルは式 (2.63) から求まり

$$c_1 = \sqrt{-2RT \ln U} \quad (2.68)$$

ここで  $T$  は壁の温度である。 $c_2, c_3$  のサンプルは式 (2.37) の  $c_1, c_2$  に一致する。

## 2.2 DSMC 法計算の流れ

DSMC 法の計算の流れは図 2.5 に示した通りである。計算では  $t = 0$  で計算領域全体に一様に分子を配置した。分子の速度はマクスウェル分布から抽出した。時間ステップごとに分子の流入、分子の移動、衝突を計算する。時間ステップ  $\Delta t$  は平均自由時間  $\tau$  より小さくする必要があるが、本計算では  $\Delta t = 0.5\tau$  とした。この  $\Delta t$  はクーラン条件を満たしている。分子の流入は式 (2.41) を用いて  $x$  方向に 0 から  $u\Delta t$  の範囲でランダムに座標を与え、速度はマクスウェル分布を仮定して与えた。分子の移動を計算して解析対象に衝突する場合は拡散反射、また計算領域から出る場合には、各境界ではそれぞれの境界条件、反射、又は流出を適用した。分子間衝突は最大衝突法を用いて解析した。

これらの過程を繰り返し計算を行い、定常流を得た後に計算格子の分割、再結合を行い解適合格子を生成する操作を行う。その後、格子の変化が少なくなった後に、格子に対するこれらの操作を止め、統計的なゆらぎを少なくするためその後平均自由時間を経過するごとにサンプリングを行い、各セルの重心での

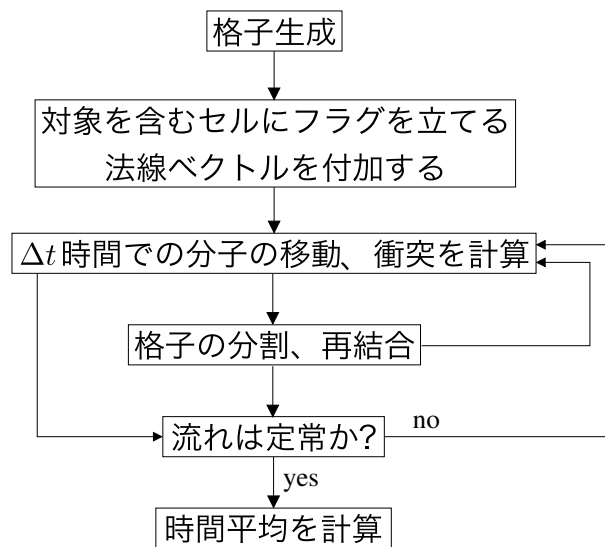


図 2.5: 計算の流れ

密度, 速度, 温度の時間平均を計算した. 計算には, 一様流の密度  $\rho_e$ , 温度  $T_e$ , 膜面の直径  $L$  として  $\rho_0 = \rho_e$ ,  $T_0 = T_e$ ,  $v_0 = \sqrt{2RT_e}$ ,  $L_0 = L$ ,  $t_0 = L_0/v_0$  を用いて無次元化を行った.

## 2.3 解適合格子と形状近似

本節では解適合格子に用いたアルゴリズムと形状近似法について述べる.

本研究では最適化を目的の 1 つとしているため, 最適解を求める過程において解析対象の形状が変化する場合においても適用可能である必要がある. 形状が変化する度に格子を生成し直すのでは非常に労力がかかるため, 形状を近似し同一の直行格子で全ての計算を行った. 図 2.6 に円柱周りの解析の際に生成された格子を示す. 図 2.6 のように格子は階層構造により解に適合すうよう変化させた.

解析対象の形状は格子を用いて表現し, その格子に実際の形状の法線ベクトルを持たせ近似を行った. これにより, 形状は滑らかではないが, セル表面での法線ベクトルの変化は連続的となる.

### 2.3.1 解適合格子のアルゴリズム

図 2.7 は, 図 2.6 のような階層構造を持つセルを 1 つだけ取り出した図である. そのセルを  $(i, j)$  とし, セル  $(i, j)$  は赤のラインで区切られる領域とする. 内部の  $8 \times 8$  の領域 1 つ 1 つはサブセルで, サブセルそれぞれの座標

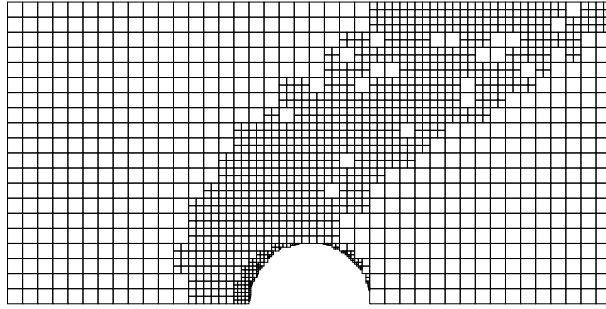


図 2.6: 解適合格子の例

は  $(i, j, ii, jj)$  として与えた.  $ii, jj$  の上限値は階層の最大値  $lv_{max}$  によって定まる. 階層が上がるにつれて領域を 2 等分するため, セル内部の分割数は  $2^{lv_{max}} \times 2^{lv_{max}}$  となり,  $ii, jj$  の範囲は  $0 \sim 2^{lv_{max}} - 1$  となる. 図 2.7 では  $lv_{max} = 3$  としたため,  $ii, jj$  の範囲は  $0 \sim 7$  となる. 横軸, 縦軸には座標を 10 進数と 2 進数で表記してある.

サブセル内部の数字はサブセル 1 つ 1 つに割り当てた階層  $lv$  を表している. 階層を基にサブセルのグループ化を行うため全てのサブセルに階層を割り当てている. 階層毎に領域の大きさが異なり, 黒のラインがグループの区切りとなる. その大きさは 1 つのサブセルの  $2^{lv_{max}-lv} \times 2^{lv_{max}-lv}$  倍として与えられる. つまり,  $lv = lv_{max}$  では 1 つのサブセルで処理を行い,  $lv = 0$  ではセル  $(i, j)$  内の全てのサブセルを一つのグループとして扱う事になる.

プログラム上では全てのサブセルが独立した領域となる場合を想定して, 一つ一つのサブセルに独立した変数を割り当てることになる. 例えばセル毎の粒子数  $lc$  は  $lc(i, j, ii, jj)$  として表される. よって, 同一のグループに複数のサブセルを含む場合には, そのグループ内には複数の変数を持つ事になる. DSMC 法では同一の領域内の粒子の衝突を計算するために領域内の粒子に連続した番号を割り当てるため, 同一グループ内の異なるサブセルに粒子数をそれぞれカウントし, 後に合算するのでは都合が悪い. そのため同一グループ内では同一の変数を用いて処理を行った. その領域を代表する変数として用いるのは, それぞれの領域内で最も  $ii, jj$  が小さい位置とした. アルゴリズムを図 2.8 に示す. プログラム内の処理の概略及びパラメータの意味は次の通りである.

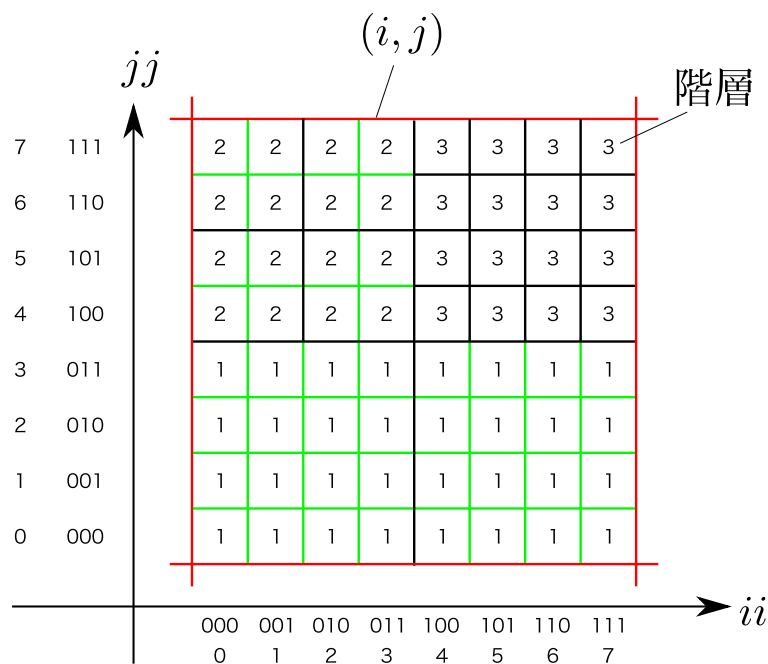


図 2.7: セル  $(i, j)$  内の座標  $(ii, jj)$  と階層

- icel, jcel : m 番目の粒子の属するセル番号
- i2, j2 : 2 進数で表した m 番目の粒子の属するサブセル番号
- ii, jj : 10 進数で表した m 番目の粒子の属するサブセル番号
- xcl, ycl : 格子の座標
- lv : 階層
- dx, dy : 階層毎の格子幅
- lc : 粒子数の合計
- 02~04 : セル及びサブセルの座標 ( icel, jcel, ii, jj) を求める処理.
- 07~14 : 先に求めたサブセル番号 ii, jj を 2 進数 i2, j2 へ変換.
- 16~19 : i2, j2 をその粒子の属するサブセルの階層 lv に合わせ座標変換を行う. lvmax - lv 桁を 0 で置換する.
- 23~28 : 2 進数から 10 進数へ復元.
- 29~31 : 変換した座標を m 番目の属するサブセルとし, 粒子数の合計 lc に 1 を足す.

図 2.8 のプログラムによって, 図 2.7 の, 例えば (1,2) や (3,3) はどちらも (0,0) となり, 同様に (0,5), (1,4), (1,5) は (0,4) と変換される. サブセルの分割数は, 先に述べたように  $2^{lv_{max}}$  なので 2 進数では  $lv_{max}$  桁で表すことが出来るので図 2.8 の i2, j2 の宣言は

```
character*lvmax i2, j2
```

```

01      do m = 1, nmol
02      icel(m) = x( m) / dx(0)
03      jcel(m) = y( m) / dy(0)
04      ii = ( x(m) - xcl( icel(m))) / dx(lvmax)
05      jj = ( y(m) - ycl( jcel(m))) / dy(lvmax)
06 c
07      do l = 1,lvmax
08      iii = int( ii / 2**( lvmax - l))
09      jjj = int( jj / 2**( lvmax - l))
10      write( i2(1:1), "(i1)") iii
11      write( j2(1:1), "(i1)") jjj
12      ii = ii - 2**( lvmax - l) * iii
13      jj = jj - 2**( lvmax - l) * jjj
14      enddo
15 c
16      do l = 1, lvmax - lv( icel(m), jcel(m), ii, jj)
17      write( i2( lvmax - l + 1: lvmax - l + 1), "(a)") "0"
18      write( j2( lvmax - l + 1: lvmax - l + 1), "(a)") "0"
19      enddo
20 c
21      ii = 0
22      jj = 0
23      do l = 1, lvmax
24      read( i2( lvmax - l + 1: lvmax - l + 1), "(i1)") iii
25      read( j2( lvmax - l + 1: lvmax - l + 1), "(i1)") jjj
26      ii = ii + 2**( l - 1) * iii
27      jj = jj + 2**( l - 1) * jjj
28      enddo
29      iicel(m) = ii
30      jjcel(m) = jj
31      lc( icel(m), jcel(m), ii, jj) = lc( icel(m), jcel(m), ii, jj) + 1
32      enddo

```

図 2.8: 座標変換プログラム (fortran)

```

00  do i = 0, mx - 1
01  do j = 0, my - 1
02  ii = 0
03  do while( ii .le. 2**lvmax - 1)
04  jj = 0
05  l = 0
06  do while( jj .le. 2**lvmax - 1)
07  if( mod( ii, 2**( lvmax - lv( i, j, ii, jj))) .eq. 0
08 & .and. mod( jj, 2**( lvmax - lv( i, j, ii, jj))) .eq. 0) then
09  .....
10  l = max( l, lv( i, j, ii, jj))
11  endif
12  jj = jj + 2**( lvmax - lv( i, j, ii, jj))
13  enddo
14  ii = ii + 2**( lvmax - 1)
15  enddo
16  enddo
17  enddo

```

図 2.9: 逐次計算

である。また  $lc$  のように配列の要素に  $ii, jj$  を用いる変数の宣言はセル数を  $mx, my$  とすると

```
dimension lc(0:mx-1, 0:my-1, 0:2**lvmax-1, 0:2**lvmax-1)
```

として宣言を行っている。

図 2.8 プログラム内の処理はまず粒子の位置する座標求めることから始まる。  $dx, dy$  は階層毎の格子幅で  $dx = dx(lv), dy = dy(lv)$  である。つまり、  $dx(0), dy(0)$  はセルの幅となる。これにより 2~3 行目で  $i, j$  を求め、4~5 行目で  $dx(lvmax), dy(lvmax)$  を用いて  $ii, jj$  を求める。  $dx(lvmax), dy(lvmax)$  は最も細かいサブセルの幅である。16~19 行目で求めた位置での階層  $lv(i, j, ii, jj)$  に応じ、  $ii, jj$  の変換を行う。  $lvmax$  桁の 2 進数の  $lvmax - lv(i, j, ii, jj)$  桁を 0 で置換を行っている。  $lvmax = 3$  の図 2.7 では  $lv = 1$  の座標は 2 進数の下 2 桁が 0 で置換され、  $lv = 2$  では下 1 桁が 0 で置換される。  $lv = lvmax$  では何も行わない。これらの処理によりグループ内で 1 つの変数のみを使用することになる。

セルの階層は分子数に基づき変化させた。その基準としては各グループ内の粒子数の最大値の 7 割以上の粒子が存在するセルは階層を上げ、階層を上げた事によりセル内の粒子数が少なくなりすぎるようであれば階層を下げる処理を行った。この階層を変化させる処理や、粒子同士の衝突などセルを逐次

的に移動し計算する。先に述べたように階層が  $lv_{max}$  でない限りは使用していない変数があるため、粒子数が0となるセルが存在する。そのため全ての位置について計算を行ったのでは計算コストが増加するだけとなる。よって、逐次計算は図 2.9 に示したように処理を行った。  $jj$  は 12 行目の処理で階層  $lv(i, j, ii, jj)$  により増加量変化させている。  $ii$  の変化量は最も  $j, jj$  方向のループ毎に最も高い階層により定める。 10 行目, 14 行目がその処理となる。

階層を変化させる場合には図 2.9 のプログラム内 09 行目でループを作成する。 図 2.7 から分かるように同一のグループ内のサブセルには同一の階層を与える必要があるためである。階層を上げる場合にはそのループのスタートとなる位置と図 2.9 のプログラム 09 行目での  $ii, jj$  とが一致するため、

```
do iii = ii, ii + 2** ( lvmax - lv( i, j, ii, jj) ) - 1
do jjj = jj, jj + 2** ( lvmax - lv( i, j, ii, jj) ) - 1
.....
enddo
endoo
```

のようになる。しかし、階層を下げる場合には先に述べた座標変換プログラム図 2.8 を用いて階層を下げた状態でのループの初期位置を決定する必要がある。その初期位置を  $is, js$  とすると

```
do iii = is, is + 2** ( lvmax - lv( i, j, ii, jj) + 1 ) - 1
do jjj = js, js + 2** ( lvmax - lv( i, j, ii, jj) + 1 ) - 1
.....
enddo
endoo
```

となり、以上の領域において階層を下げる必要がある。今回はセルの階層に方向性を持たせてはいないが、3次元計算などでセルを分割する際に分割したセルの体積が小さくなりすぎる場合など、階層を表す変数を増やし方向性を持たせれば同様のアルゴリズムで方向性を持ったセル分割も可能となる。

### 2.3.2 分子の移動, 反射

本研究では図 2.10 のように実際の境界を含むセルを計算上の粒子が反射する境界として、形状を近似し計算を行った。よって、実際の境界と計算上の境界が一致しているとは限らない。そのままでは、分子の反射方向の分布が実際の境界で反射した場合とあまりに異なり、正しく解析が行えないため、計算上の境界面には実際の境界面の法線ベクトルを与えた。これにより、計算上の境界面で反射した分子の分布は実際の境界面で反射した場合と等しい分布をとることになる。

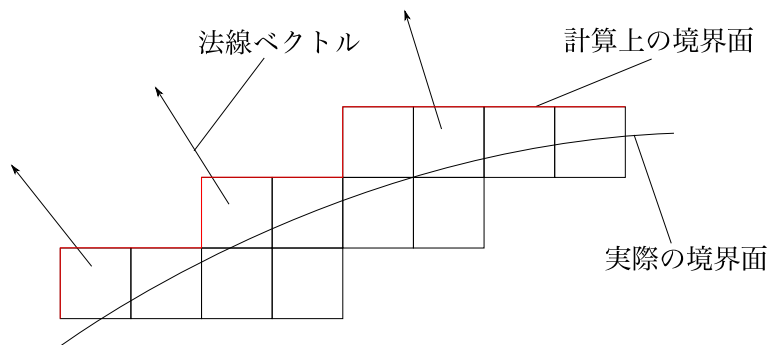


図 2.10: 境界の定義

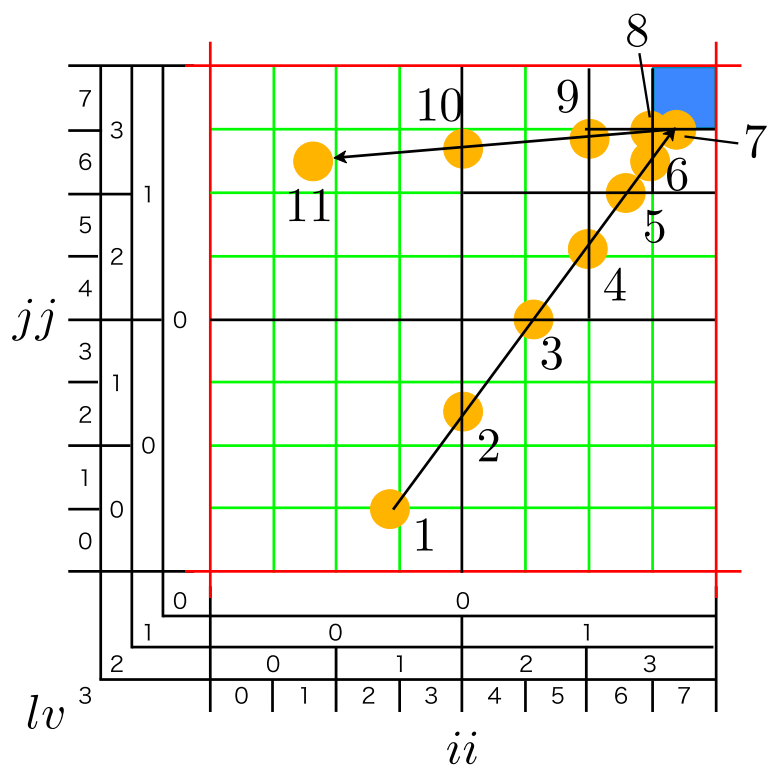


図 2.11: 境界の判別と分子の移動



DSMC 法では粒子を平均自由時間移動させ、計算対象と衝突するか判別し、衝突するのであれば反射をさせる行程が必要となる。形状を近似していない場合の形状は分子の初期位置と速度、形状を表す方程式より衝突の判定を行えば良いが、形状を近似した場合には形状を方程式で表した場合には複雑かつ膨大となる。よって実際の境界が含まれているセルにはフラグを立て、移動時にはそのフラグにより判別を行った。

図 2.11 には粒子の移動のイメージを示した。粒子の移動は 1 ~ 11 の順に移動する。図 2.11 は 1 つのセルで内部はサブセルである。 $lv_{max}$  は 3 で  $2^{lv_{max}} \times 2^{lv_{max}}$  の領域に区切られ、青塗りの領域は実際の境界を含むサブセルである。縦軸、横軸はセル  $(i, j)$  内のサブセルの座標  $(ii, jj)$  を表している。4 通りで表記してあるのは階層により座標の与え方が異なるためで、 $lv_{max} = 3$  としたため  $lv = 0 \sim 3$  の 4 通りである。例えば粒子 2 の座標は  $(lv, i, j, ii, jj)$  として表すと、

$$\begin{cases} (0, i, j, 0, 0) \\ (1, i, j, 1, 0) \\ (2, i, j, 2, 2) \\ (3, i, j, 4, 3) \end{cases}$$

の 4 通りで表される。プログラムでのフラグの宣言は

```
character*1 flag(0:lvmax,0:mx-1,0:my-1,0:2**lvmax-1,0:2**lvmax-1)
```

として宣言を行い、実際の境界を含む領域は  $flag = "y"$  とする。つまり、図 2.11 のサブセルのフラグは "y" となるので、

$$flag(3, i, j, 7, 7) = "y"$$

となる。同様に階層を下げた場合に  $flag = y$  となった  $(3, i, j, 7, 7)$  を含む領域のフラグも "y" となるので、

$$\begin{cases} flag(0, i, j, 0, 0) = "y" \\ flag(1, i, j, 1, 1) = "y" \\ flag(2, i, j, 3, 3) = "y" \end{cases}$$

となる。それ以外のセルでは  $flag = "n"$  である。

このフラグに基づき粒子を移動させる。計算ではセルを移動する毎にセルのフラグを確認する必要がある。そのため、図 2.11 で示した様にセルを移動する度に一度停止し、次に進むセルのフラグを参照し、移動もしくは反射を行う。階層は低い程粒子の 1 度の移動量が大きいため、移動にかかる計算は早く終わる事が出来る。そのため、出来る限り低い階層で計算を行うことが計算の高速化の条件である。図 2.11 では境界となるセルに近づくにつれて移動の間隔が小さくなっている。これは階層を変化させながら移動を行うためである。階層が変化するのは粒子番号 1, 3, 5, 9, 10 の位置である。1 では階層

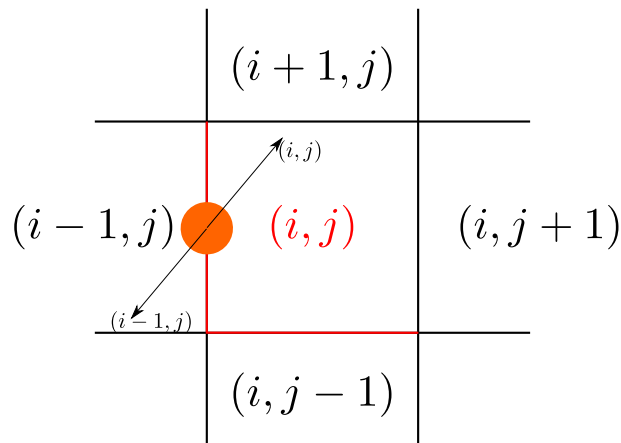


図 2.12: 境界上の分子

が0から1へ, 3では1から2へ, 5では3から2へ, 9では3から2, 10では2から1へと変化して移動を行っている。つまり, 先に定めた  $flag = "y"$  の領域の境界において階層を変化させ, その階層毎の  $ii, jj$  が変化する位置まで移動をさせている。その位置は図 2.11 内の黒及び赤のラインで示した位置である。階層を変化させる際に問題となるのが 8, 9, 10 のように速度成分が負である場合である。図 2.12 に示したような境界上の粒子は計算上セル  $(i, j)$  内に存在するとして扱われる。速度成分が正の場合には次に進むセルも粒子の属するセル  $(i, j)$  と一致する。しかし, 速度成分が負の値を持つ場合には次に進むセル  $(i-1, j)$  となり, 粒子が属するセルとは異なる。よって速度ベクトルに応じてフラグを参照する相対的な位置を変える必要がある。

粒子の移動を行うサンプルプログラムは次のようになる。

```

01 初期位置 i, j を求める。lv = 0 なので ii, jj はともに 0 としておく。
02     lv = 0
03     do while (dtfly .gt. 0)
04         do while ( lv .ge. 1 .and. flag( lv-1, i, j, int( ii / 2),
& int( jj / 2)) .eq. "n")
05             lv = lv - 1
06             ii = int( ii / 2d0)
07             jj = int( jj / 2d0)
08         enddo
09     do while (flag( lv, i, j, ii, jj) .ne. "n" .and. lv .lt. maxlv)
10         lv = lv + 1
11         ii = int((x(n) - xcl(0, i, 0)) / dx(lv))
12         jj = int((y(n) - ycl(0, j, 0)) / dy(lv))

```

```

13     if( ii .gt. 2**lv - 1) ii = 2**lv - 1
14     if( jj .gt. 2**lv - 1) jj = 2**lv - 1
15     end do
16     if ( cx(n) .lt. 0) then
17         dthit1 = ( xcl(lv, i, ii) - x(n)) / cx(n)
18     else
19         dthit1 = ( xcl(lv, i, ii) + dx(lv) - x(n)) / cx(n)
20     endif
21     if ( cy(n) .lt. 0) then
22         dthit2 = ( ycl(lv, j, jj) - y(n)) / cy(n)
23     else
24         dthit2 = ( ycl(lv, j, jj) + dy(lv) - y(n)) / cy(n)
25     endif
26     dthit = amin1(dthit1, dthit2)
27     if( dthit .eq. 0) then
28         if (dt_hit1 == 0) then
29             if ( ii .ne. 0) then
30                 ii = ii - 1
31             else if ( i .ne. 0) then
32                 ii = 2**lv - 1
33                 i = i - 1
34             endif
35             if ( flag( lv, i, j, ii, jj) .eq. "n") then
36                 dthit1 = abs( dx(lv) / cx(n))
37             else
38                 dt_hit1 = 0
39             endif
40         endif
41         if (dthit2 == 0) then
42             if (jj .ne. 0) then
43                 jj = jj - 1
44             else if ( j .ne. 0) then
45                 jj = 2**lv - 1
46                 j = j - 1
47             endif
48             if (flag(lv, i, j, ii, jj) .eq. "n") then
49                 dthit2 = abs(dy(lv) / cy(n))
50             else
51                 dthit2 = 0

```

```

52     endif
53     endif
54     endif
55     dthit = amin1(dthit1, dthit2, dtfly)
56     if (flag(lv, i, j, ii, jj) .eq. "y"
& .and. lv .eq. maxlv) then
57     反射.....
58     end if
59     if( dthit .ne. 0d0) then
60     粒子を dthit 時間分移動させる
61     if ( 移動により領域外に粒子が流出) then
62     粒子を削除する.
63     goto 111
64     endif
65     粒子の移動により変化した i, j, ii, jj を求める.
66     dtfly = dtfly - dthit
61     endif
62     enddo
63 111 continue

```

- 04~08 : 階層を下げる処理. 階層を下げた場合にも  $flag = "n"$  でれば下げる.
- 09~15 : 粒子のいる座標の  $flag \neq "n"$  であれば階層を上げる.
- 16~56 : 次の境界までの移動時間を求める処理. 0 となる場合は速度成分に負の値を持つ場合なので, その場合は次に進むセルが, この時点で粒子が位置するセルとは異なる. よって, それに応じて  $i, j, ii, jj$  を変更する.  $flag(i, j, ii, jj) = "n"$  であれば現在の階層の幅となる距離分の移動時間を求める.
- 57~62 : 粒子を移動させる.  $flag(lv, i, j, ii, jj) = "y"$  かつ  $lv = lv_{max}$  であれば反射をさせ, 計算領域外に出た粒子に関しては削除する.
- 65~ : 次に進むセルの位置  $i, j, ii, jj$  を求める. 残りの時間計算し, これが 0 となった場合には次の操作に移る.

粒子の移動のプログラムは基本的には階層毎の境界までの移動を繰り返し行う. 高速化のために 02 行目で書いたように階層は 0 からスタートを行う. 04 行目から 08 行目では階層を下げた場合にもフラグが "n" であれば階層を下げる処理を行っている. 階層が 0 であれば何も行わない. 09 行目から 15 行目はフラグが "y" となった場合に階層を上げる処理を行う. 階層が  $lv_{max}$  の場合には何も行わない. 13 行目, 14 行目の処理は後の処理により条件によっては  $(i, j)$  の境界上の粒子の位置  $(i, j)$  から  $(i - 1, j)$  もしくは  $(i, j - 1)$

と変換しているため、配列の要素の上限値をオーバーする可能性がある。そのためこのような処理を行っている。16行目から26行目まででは  $x, y$  方向それぞれの次の境界の位置までの移動時間を計算し、小さい方を選択する。25行目から55行目は移動時間が0となる場合に移動時間を計算し直すための処理である。先に述べたように速度成分が負の場合には進むセルの座標と属するセルの座標がことなるため、16行目から26行目の処理では移動時間は0となる。そのような場合には次に進む領域のフラグを参照するために座標を変換する。dthit1が0となる場合には  $i-1$  と  $i$  との境界上に位置し、dthit2が0となる場合には  $j-1$  と  $j$  との境界上に位置することになるので、それぞれに応じて変換する。その変換後のフラグが "n" である場合には移動時間を計算し直し、フラグが "y" となる場合には09行目から15行目で階層を上げるため移動時間は0のまましている。56行目で先に求めた移動時間と残りの平均自由時間を比較し最も小さいものを移動時間とする。56行目から58行目では  $lv = lv_{max}$  かつ  $flag = "y"$  となった場合に反射をさせる。この場合には速度が変化するためこのif節の中で移動時間を計算し直している。60行目からでは粒子を dthit 時間分の移動を行い、その移動した結果が計算領域の外に出た場合には粒子を削除し次の操作に移る。そうで無い場合には移動により変化した位置の座標  $(i, j, ii, jj)$  を求める。そしてまたループの最初に戻り、その位置のフラグを参照し、階層を変化させながら移動を繰り返し進める。

このサンプルコードからもわかるように、反射の判定は基本的には56行目1行であるため、計算対象の形状がどのように複雑化してもほとんど書きかえることなく使えると思われる。

## 第3章 形状近似 DSMC 法の検証

### 3.1 形状近似 DSMC 法

形状を格子で近似しているため、形状の表現は格子幅に依存している。そのため、近似が及ぼす影響、格子幅の影響を調べる事は非常に重要となる。よって、平板と円柱を用いて検証を行った。

#### 3.1.1 平板周りの流れ

bird による平板 [4] の計算と同様の条件を用いて検証問題として解いた。計算条件はクヌーセン数 0.0324, マッハ数 5, 平板の上面温度は一樣流温度, 下面は淀み点温度, 平板の迎角は  $\pi/6$  [rad] である。その結果の温度分布を図 3.1, 密度分布を図 3.2 に示す。この場合, 計算上の反射を行う境界と格子とが平行なため, 実際の格子の法線ベクトルと実際の境界の法線ベクトルが一致するため, 形状近似はしていない。

図 3.4 から図 3.13 には同様の条件, クヌーセン数 0.0324, マッハ数 5, 平板の上面温度は一樣流温度, 下面は淀み点温度, 平板の迎角は  $\pi/6$  [rad] で格子幅を変化させた結果を示す。図 3.1, 図 3.2 とは座標変換を行っただけであり, 幾何学的には同様の問題を解いている, 座標変換をした結果, セルで形状を近似しているため平板表面の形状は連続的ではなくなっている。しかし, 粒子の反射には実際の境界の法線ベクトルを用いるため, 法線ベクトルの変化は連続的である。

図 3.3 にこの平板周りの流れの解析を行ったときに生成された格子を示す。図 3.4, 図 3.5 に最小格子幅を 0.635, 図 3.6, 図 3.7 に最小格子幅を 0.0313, 図 3.8, 図 3.9 に最小格子幅を 0.0156, 図 3.10, 図 3.11 に最小格子幅を 0.00781, 図 3.12, 図 3.13 は最小格子幅を 0.00390 とした計算結果の温度分布, 及び密度分布である。温度分布はどの格子幅をとっても下面, 上面とも良く一致していた。値としては淀み点で 9.4~9.6 の間に範囲にあり, 近似を行っていない図 3.1 と比較しても DSMC 法の統計処理によるゆらぎの範囲内であった。温度分布は先に仮定した条件のように上面に一樣流温度, 下面共には淀み点温度と一樣の値を与えているため差が出にくいと考えられる。しかし, 下面側の高温の領域の広がりを見ると最小格子幅が大きいほど広く分布しているのがわかる。密度分では更に顕著に差が現れている。最も差が大きい

が前縁で、格子幅が大きいほど下面側で密度が高い部分が生じ、その領域が広く分布している。これはまずは格子の最小格子幅分の厚みを持つことが原因と考えられる。近似前の平板は厚みが0であるが、セルを用いて形状を近似しているために生じる厚みがあるため、それにより淀み点での密度が上がる事が考えられる。次に、密度が高くなったことにより、その領域の平均自由行程が小さくなる事が原因と考えられる。最も密度の高い淀み点での平均自由行程と最小格子幅の比は図 3.4, 図 3.5 で 6.7, 図 3.6, 図 3.7 で 3.8, 図 3.8, 図 3.9 で 1.9, 図 3.10, 図 3.11 で 1.0, 図 3.12, 図 3.13 で 0.5 であった。

格子幅と比較して平均自由行程が小さくなっているセルにおいては、流入する粒子数は風上の影響を強く受けほぼ一定値であるが、平均自由行程が小さくなる分、平均自由行程以上のサイズを持つセルから流出出来る粒子は減少する。よって、前縁下面付近での粒子数は増加することとなる。図 3.14 は平板の下面の密度分布をプロットしたものである。赤のラインが基準となる図 3.2 の密度を示している。その他のラインは計算対象の形状を表す最小格子幅と流れ場の最も小さい平均自由行程との比をとってあらわした。格子サイズの方が平均自由行程より充分に小さい場合には良く一致している。どのラインにおいても、密度の低くなる後縁側にいくにつれ前縁と比べ、近い値となっているため、平均自由行程と最小格子幅の関係は非常に重要である事がわかる。

次に同様の計算を計算上の境界となるセルに実際の境界の法線ベクトルを与えずに行った。計算格子は図 3.12, 図 3.13 と同様の最小格子幅を 0.00390 としたものをを用いた。しかし、淀み点付近だけでなく下面側一面に密度の高い領域が生じ平均自由行程の非常に小さい、連続流の領域が生成された。その密度の高い領域での粒子の平均自由行程が非常に小さくなるため、その領域から流出する粒子数は相対的に減少し、全体としては粒子数が増加を続け、定常解を得ることが出来なかった。形状を近似する場合には元の形状の法線ベクトルを正しく与えること、平均自由行程が最小格子幅を下回らないことが重要であると言える。

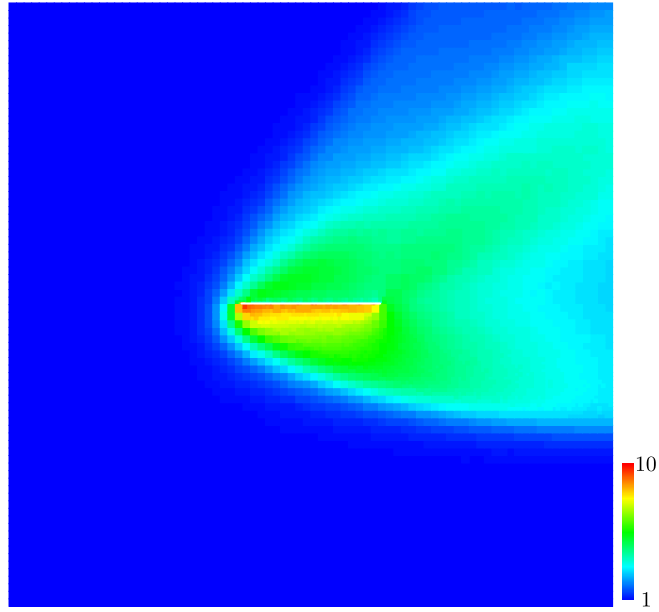


図 3.1: 平板周りの温度分布

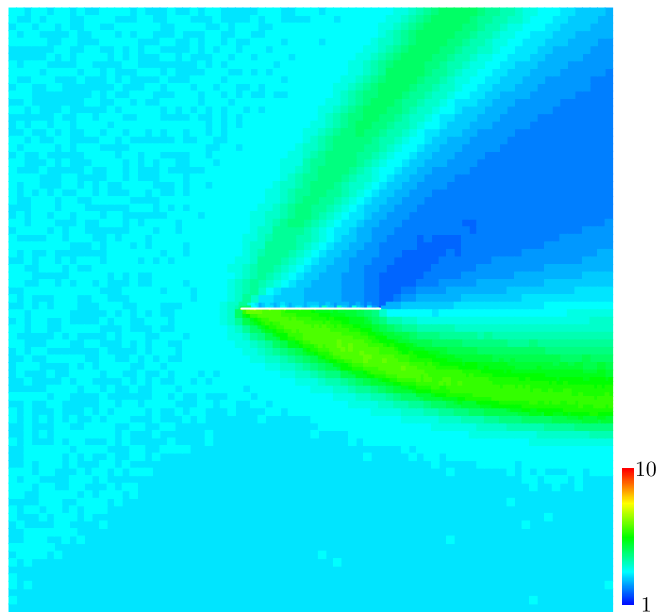


図 3.2: 平板周りの密度分布



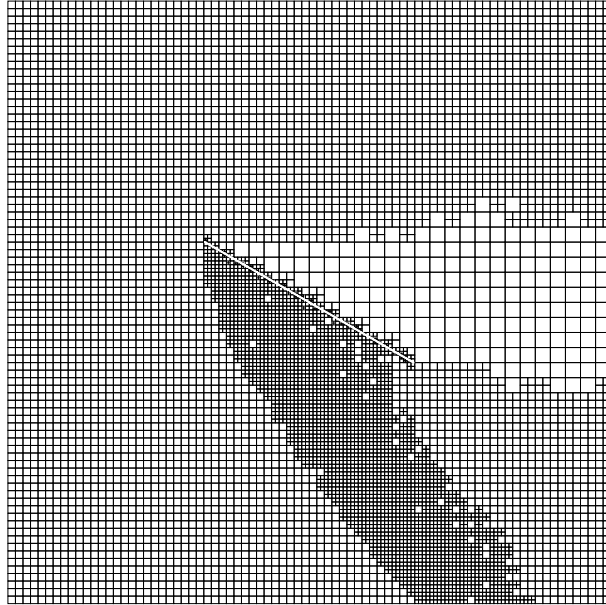


図 3.3: 平板の計算に用いた格子

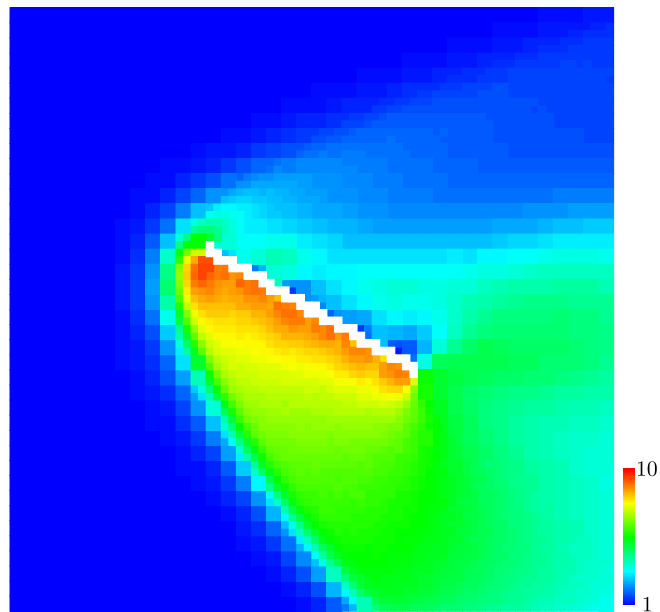


図 3.4: 最小格子幅 0.125 の温度分布

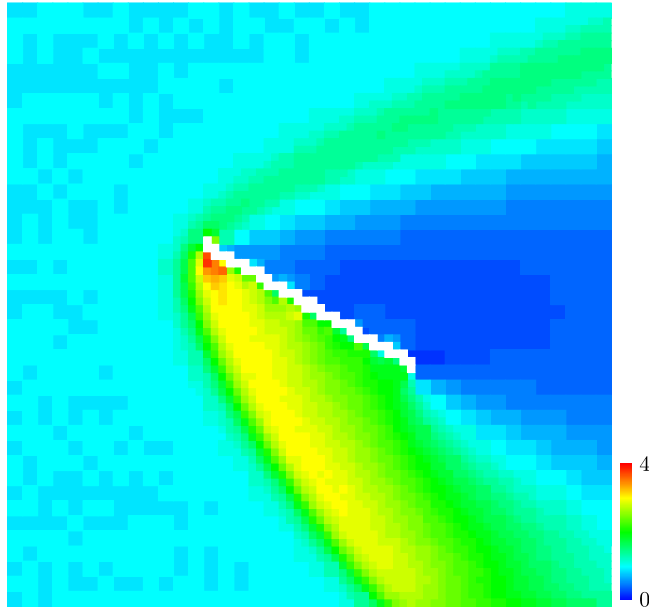


図 3.5: 最小格子幅 0.125 の密度分布

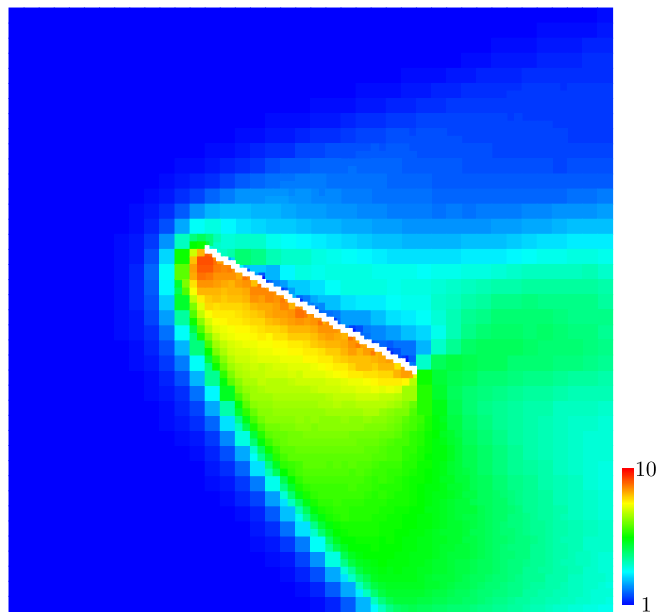


図 3.6: 最小格子幅 0.0625 の温度分布

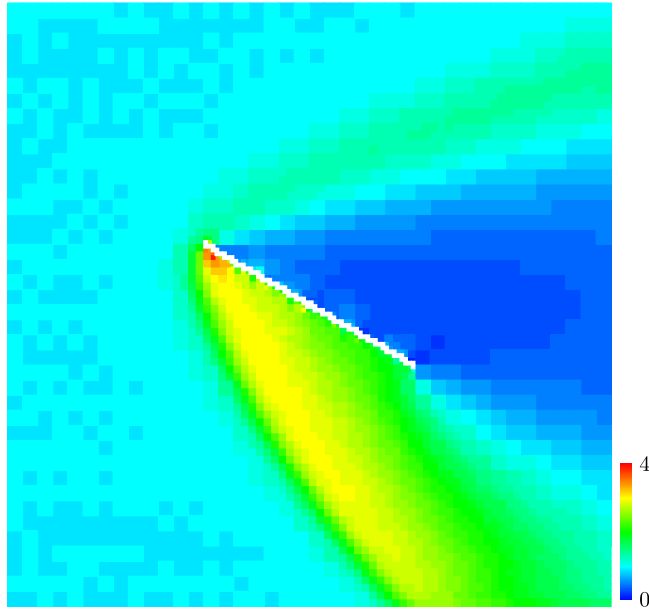


図 3.7: 最小格子幅 0.0625 の密度分布

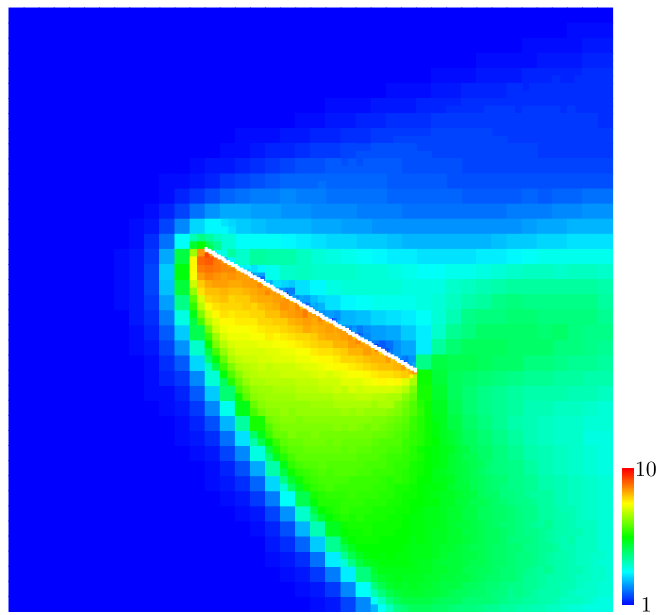


図 3.8: 最小格子幅 0.0313 の温度分布

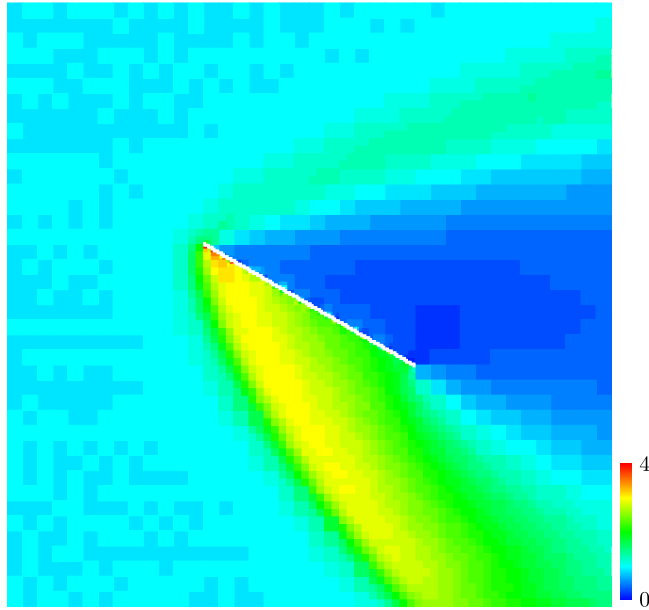


図 3.9: 最小格子幅 0.0313 の密度分布

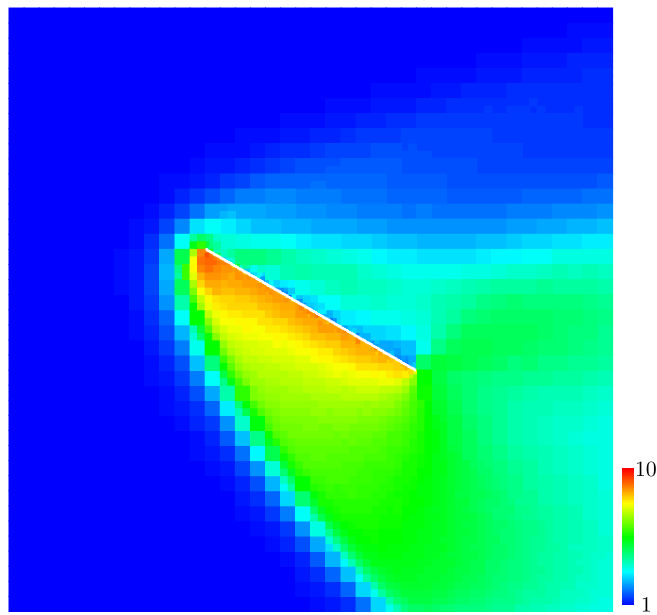


図 3.10: 最小格子幅 0.0156 の温度分布

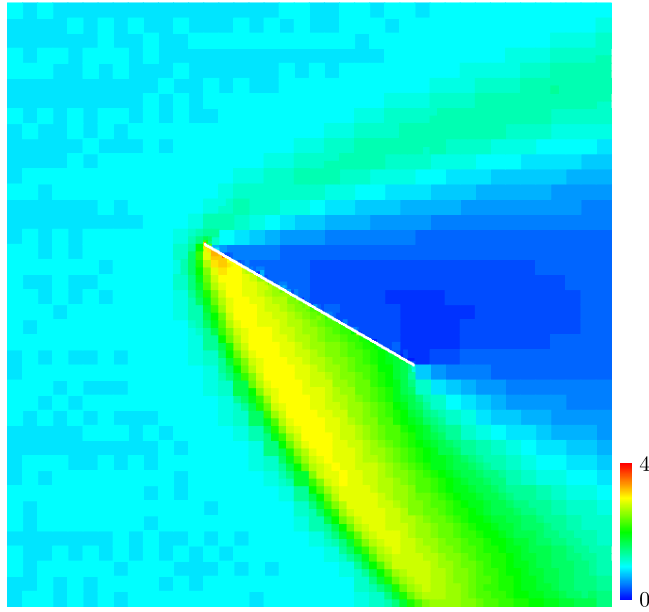


図 3.11: 最小格子幅 0.0156 の密度分布

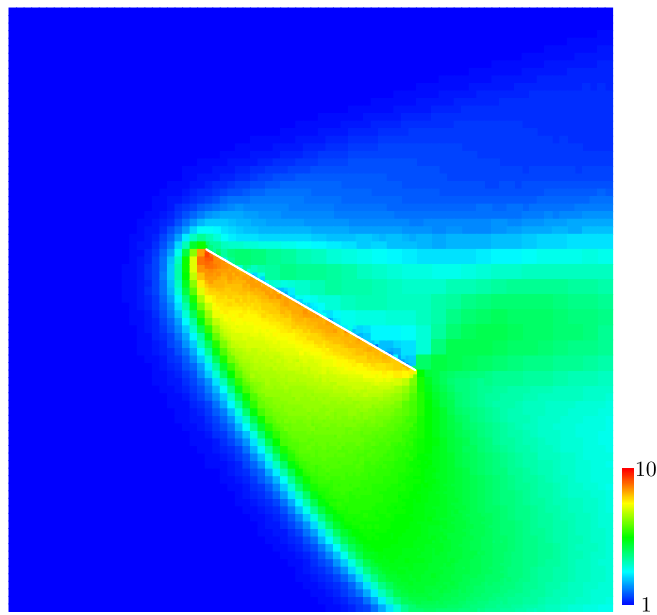


図 3.12: 最小格子幅 0.00390 の温度分布

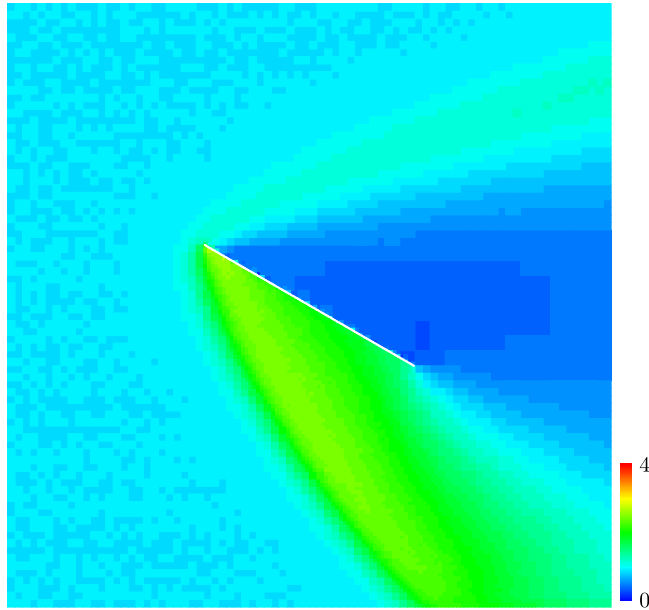


図 3.13: 最小格子幅 0.00390 の密度分布

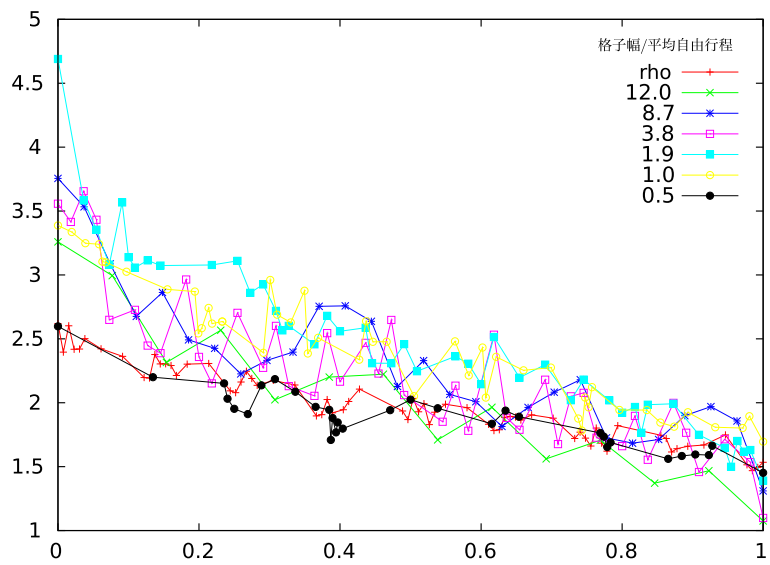


図 3.14: 平板下面の密度分布

### 3.1.2 円柱回りの流れ

3.1.1 では平板の周りの流れを解き格子幅、境界に与える法線ベクトルが重要であることがわかった。平板であったので法線ベクトルは位置によらず一定である。法線ベクトルが連続的に変化した場合に解に及ぼす影響を調べるため、円柱を用いて計算を行った。格子は図 3.15 に示した例のように、それぞれの計算ケース毎にプログラム内で自動的に解に適合する格子を生成し解析を行った。

計算条件はマッハ数 5、クヌーセン数は 0.5 とした。結果の密度分布を図 3.16, 図 3.17, 図 3.18, 図 3.19 に示す。形状を近似せずに計算を行った結果が図 3.16 である。図 3.17, 図 3.18, 図 3.19 は形状を近似し、境界となるセルには実際の境界の法線ベクトルを与えた。それぞれ格子幅が異なり、淀み点での平均自由行程と格子幅の比は 0.74, 0.4, 0.25 であり、3.1.1 より得られた格子幅と平均自由行程の関係の条件を満たしている。近似した結果それぞれの淀み点での密度は近似していない結果と比較して 1%程度の範囲内にあり良く一致していた。しかし、格子幅と平均自由行程の比が 0.75, 0.4 の場合では図 3.20 に示した赤の領域のように先端ではない領域が最も密度が高くなった。図 3.20 は円柱の先端を拡大したもので、青で示したセルは円柱を含むため、そのセルの表面が境界となる。最も密度の高くなったのは最も淀み点に近い凹面となる位置であった。その凹面となる位置での密度は図 3.17 では 10%, 図 3.18 では 5%淀み点より高かった。しかし、平均自由行程と最小格子幅の比が 0.25 の場合図 3.19 では近似していない結果図 3.16 と良く一致しているため、連続的に法線ベクトルが変化する場合においても本研究で用いた近似は有効であると言える。

凹面となる位置で密度が高くなった原因は法線ベクトルの向きの影響が大きく、連続的に変化するためではないと考えられる。先頭付近では近似して与えた法線ベクトルと格子とがほぼ平行になる。そのため、3.1.1 で境界となるセルに法線ベクトルを与えなかった場合と条件に近い領域が淀み点付近に生じることになり、それにより密度が高くなったと考えられる。この現象も格子幅を小さくすることで、徐々に見られなくなった。

計算時間は近似しなかった場合とした場合を比較すると平板、円柱とともに 2/3 程度の時間であった。これは、近似しなかった場合には衝突するかどうかの判断が簡単になっていること、反射する際の法線ベクトルはメモリを参照するのみで計算を行わないこと、プログラムで時間のかかる 2 乗やルートなどの関数の計算がないことが理由と考えられる。時間の面からもこの近似は有用であると言える。

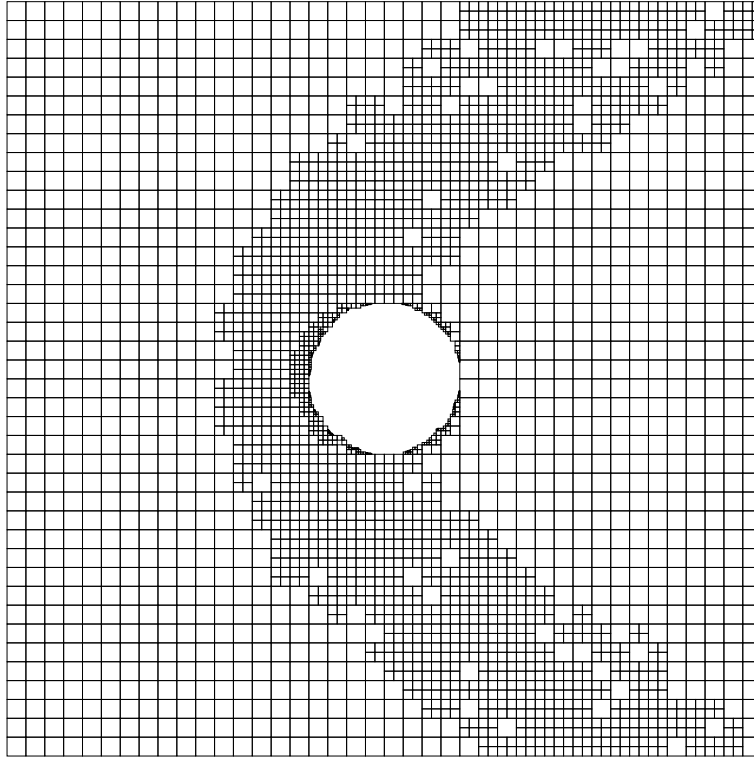


図 3.15: 円柱周りの流れに用いた格子の例

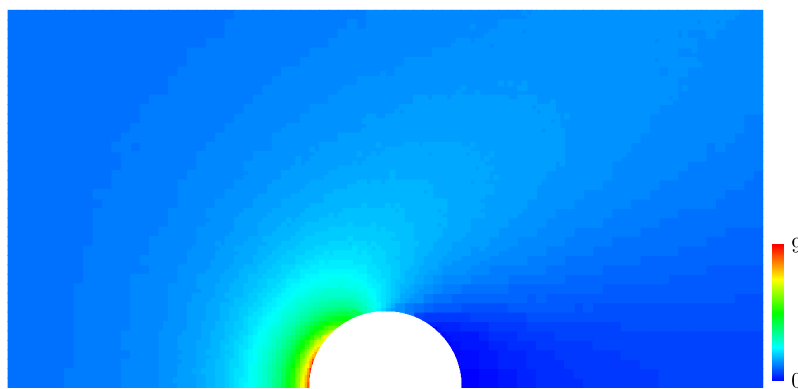


図 3.16: 形状を近似せず行った結果の密度分布



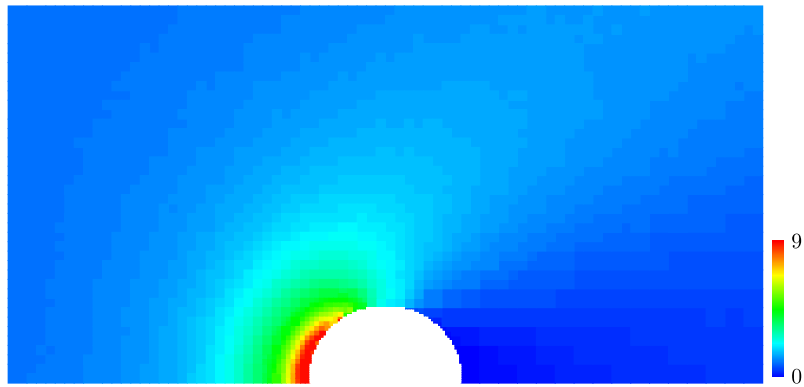


図 3.17: 密度分布 (格子幅と平均自由行程比 : 0.74)

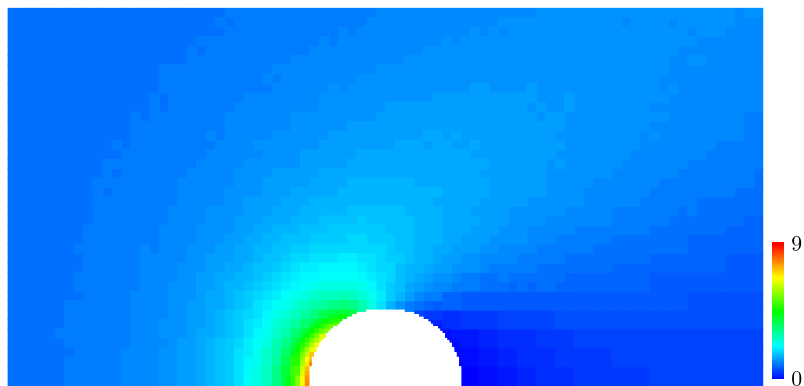


図 3.18: 密度分布 (格子幅と平均自由行程比 : 0.4)

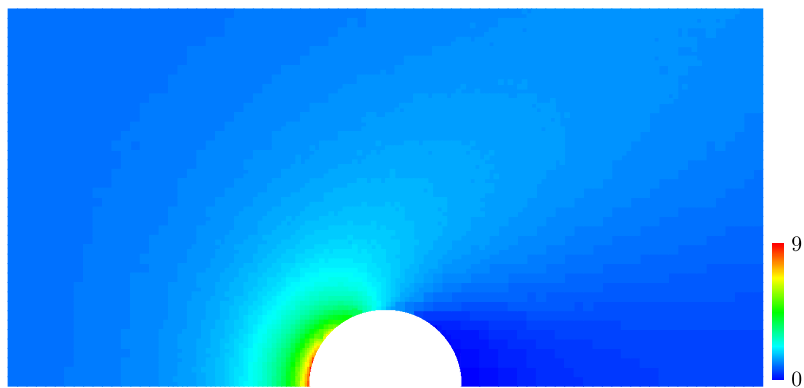


図 3.19: 密度分布 (格子幅と平均自由行程比 : 0.25)

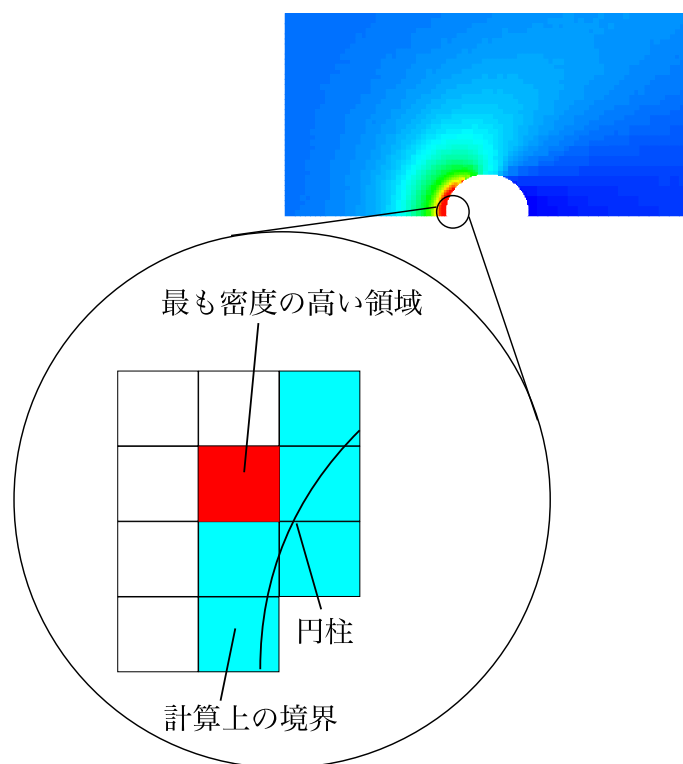


図 3.20: 粒子の流れが滞る地点)

## 第4章 形状近似DSMC法による 希薄気体中の飛行体形状最適化

### 4.1 遺伝的アルゴリズム

遺伝的アルゴリズム (genetic algorithm : GA) とは遺伝子で表現した「個体」を複数用意し、適応度の高い個体を優先的に選択して交叉や突然変異などの操作を繰り返しながら解を探索する手法である。遺伝的アルゴリズムを用いて進化させる際の原則は、適合度の大きい遺伝子ほど次世代に反映させることである。

一般に遺伝的アルゴリズムのパラメータとなる遺伝子の集団のサイズを選ぶとき、出来るだけ大きい方が良いとされている。集団が小さい場合には局所最適解に収束する可能性が高くなるためである。しかし、集団が大きすぎる場合には準最適解は求まるが過度の適合度評価が必要となる。そこで本研究ではシンプレックスアルゴリズムを用いた。

シンプレックスアルゴリズムの流れを図 4.2 に示す。図 4.2 の各操作は以下の通りである。

#### 1. 初期集団の生成

初期集団のために  $N$  個の遺伝子を生成する。ここでの  $N$  は小さな奇数である。本研究では遺伝子のコード化は問題空間の値を 2 進数に直し、染色体を 0 と 1 の遺伝子に変換する方法を用いた。

#### 2. 並べ替え

各遺伝子の適合度を評価する。適合度に基づいて最良解を選び、 $N$  番目の位置にその遺伝子を配置し、次世代へと渡す。

#### 3. 選択及び交叉

集団中の  $2n$  番目と  $2n-1$  番目の遺伝子の適合度を比較し、より高い遺伝子を交叉場へと移動させる。この操作を  $N-2$  番目の遺伝子と  $N-1$  番目の遺伝子との適合度の比較まで行い、交叉場内の  $(N-1)/2$  個の遺伝子を交叉し次世代へと渡す。 ( $n = 1, 2, \dots, (N+1)/2$ )

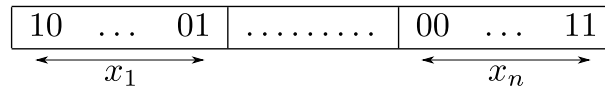


図 4.1: n 変数の遺伝子例

#### 4. 突然変異

ここまでの操作により次世代へは  $(N - 1)/2$  個の遺伝子が渡されている。残りの  $(N - 1)/2$  個の遺伝子は最良解を突然変異させ生成する。

#### 5. 適合度の評価

世代が終了世代にいたるまで 2 へ戻り繰り返し操作を行う。

個体を表す文字列としては 0,1 の並びである文字列を用いた。すなわちビット列によるコード化を行った。数値としては 2 進数と同じような扱いをするために、遺伝子の長さが精度に依存する。つまり、 $10^{-6}$  までの精度を得るためには 2 進数により 1 を  $10^{-6}$  まで分割可能な遺伝子長が必要となる。その長さは

$$2^{21} < 10^6 \leq 2^{22} \quad (4.1)$$

であるため、少なくとも 22 ビット必要である。このビット列を実数値へ戻すデコード化も、文字列に現れる 2 進数を基数 2 から基数 10 に変換し、分割された領域の大きさを乗じれば容易に得られる。ここまで 1 変数について述べたが多変数になった場合にも各変数に必要なだけビット数を与え、図 4.1 順につなぎ合わせれば問題無く行える。

交叉は生物が交配によって子孫を残すことをモデル化したもので、個体の遺伝子の一部を入れ換える操作である。交叉はその性質上、最も重要な遺伝的操作とすることができる。交叉の中にも一点交叉、二点交叉、多点交叉、一様交叉などがあるが本研究では図 4.3 に例を示した一様交叉を用いた。一様交叉とは各要素ごと独立に 1/2 の確率で入れ換える交叉である。図 4.3 の場合で言えば、乱数を用いて 0,1 の文字列を発生させ乱数が 0 ならば、親 A の遺伝子を、1 ならば親 B の遺伝子を受け継ぐ。図 4.3 では a~k といった文字を遺伝子として用いているが、実際のプログラム内での遺伝子は 0,1 のビット列であるために、親となる 2 つ遺伝子と比較したときに等しい部分では交叉を行っても同じ遺伝子となるため、あたかも交叉は行われていないかのようになる。よって突然変異のような操作が必要となる。

突然変異とは突然変異は生物に見られる遺伝子の突然変異をモデル化したもので、個体の遺伝子の一部を変化させる操作である。図 4.4 に示した例のように、遺伝子型がビット列の場合はある遺伝子座の 0 と 1 を入れ換える。数字の場合は乱数と置き換える。他にも遺伝子座の位置を変更するなどの方法がとられる。本研究では遺伝子がビット列なので 0 と 1 入れ替える方法をとった。プログラムとしては、まず  $P_{mutation}$  を決めておき、交叉を終えた

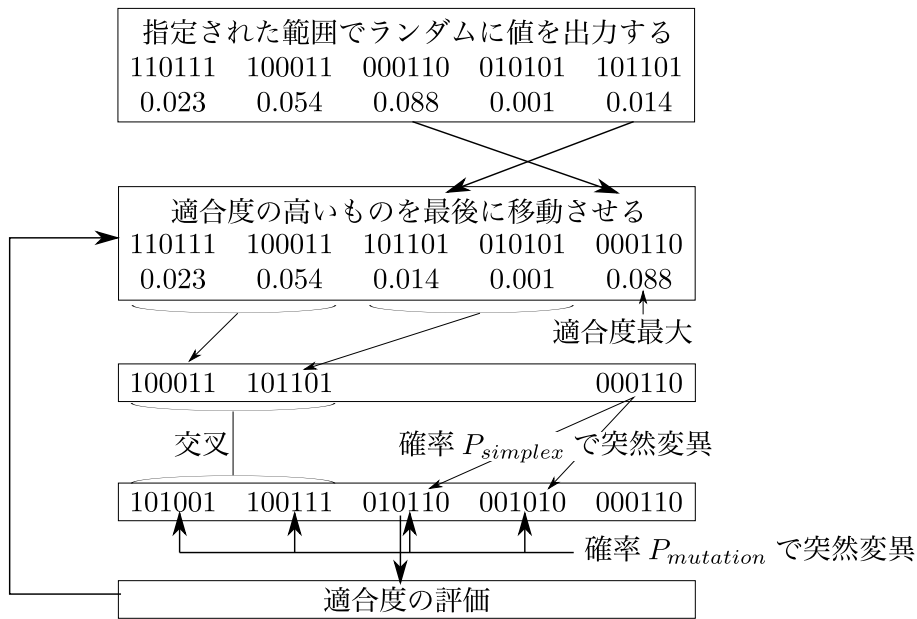


図 4.2: シンプレックスアルゴリズム

親 A	1	2	3	4	5	6	7	8	9	10
親 B	a	b	c	d	e	g	h	i	j	k
マスク	0	1	1	0	1	0	1	0	0	1
					↓					
					一様交叉					
子 A	1	b	c	4	e	6	h	8	9	k
子 B	a	2	3	d	1	g	7	i	j	10

図 4.3: 一様交叉の例

遺伝子に対して 0~1 の乱数を発生させ、乱数が  $P_{mutation}$  を下回った場合には遺伝子の 0 と 1 を入れ替え突然変異させる。図 4.4 では左から遺伝子座が  $P_{mutation}$  を下回ったので 1 から 0 へと変異している。一般的には  $P_{mutation}$  の値は非常に小さい。あまりにこの確率が大きすぎると収束性が悪くなるため、適切な値を決定する必要がある。

## 4.2 エアロキャプチャ概要及び最適化条件

最適化を行う条件として、エアロキャプチャを想定した。近年の宇宙開発に関しては、Space Ship One によって民間企業による宇宙旅行も夢ではない時代となり、宇宙へ出るために掛かるコストは大幅に削減されると予想され、惑星探査も例外ではなく、低コスト化が求められと考えられる。コスト

$P_{mutation} = 0.010$								
ビット列	:	1	0	0	<u>1</u>	1	0	1
ランダム数	:	0.283	0.785	0.194	0.003	0.6732	0.093	0.612
		↓ 突然変異						
突然変異後のビット列	:	1	0	0	<u>0</u>	1	0	1

図 4.4: 突然変異の例

を下げる為には重量当たりのペイロードを増加させる事が重要で、探査機のペイロードを増大させる方法として、オンボードの燃料重量を削減することが非常に有効であるためエアロキャプチャが注目されている。エアロキャプチャとは、惑星到着時に惑星の周回軌道に衛星を投入する技術であり、従来の燃料による減速の代わりに、惑星の大気の空気抵抗を用いて減速を行う、エアロアシストと呼ばれる技術のうちの一つである。エアロアシストと呼ばれる技術には、他に大気圏突入、エアロブレーキなどが含まれる [3]。エアロキャプチャは大気を持つすべての惑星に適用することができ、1回の突入で数 km/s の減速を行う。その利点としては以下のようなものがあると考えられている。

- ペイロードを増大させることにより、探査機の小型化やより多くの観測機器を搭載することが可能。
- 重量削減、探査機の小型化により打ち上げ時に安価なロケットを使用することが可能。
- 一度の大気への突入で周回軌道への投入ができるため、軌道投入に数週間から数ヶ月かかると言われるエアロブレーキよりも短時間でミッションを行うことが可能。
- 軌道投入時の重量面からの  $\Delta V$  の制約が緩和されることにより惑星間移行の際タイミングを図る必要が少なくなりミッションを行える機会が増える。

高層大気中を何度も衛星を通過させるエアロブレーキは既に成功を収めており、安全ではあるが長期間を要する短所がある [1]。エアロキャプチャは1回の大気圏突入で減速を終えるために期間を大幅に短縮出来、コストダウンに繋がる。しかし、1度しか大気圏突入が無いために様々な難しさを含んでいる。それは1度の突入で減速を行わなければならないために、急激な減速を行う際の厳しい空力加熱、突入角が浅過ぎた場合には逆に十分な減速が得られずに周回軌道に乗れず離脱してしまうこと、また惑星の高層大気状態の不安定性等により予想した空気力と実際の空気力との差が生じる可能性があることなどである。

最適化を行う条件を決定するために、軌道計算を行った。ミッションとしては木星から地球への帰還を想定した。木星及び木星の衛星は様々な観点から注目を集めている。まずは、太陽系の起源の解明のためである。原始太陽系星雲から惑星がどのように形成されてきたかを解明するためには、木星型惑星およびその衛星の探査から、ガス惑星の起源を明らかにすることを通じて、原始太陽系星雲の条件を制約することが必要となる。木星では、大気の組成が木星の大部分を占めるガスの組成を代表しており、それは原始太陽系星雲ガスの組成を通じて太陽系の起源につながる。たとえば重水素/水素比は太陽よりも高く、明らかに星内での重水素燃焼前の星間ガスの組成を反映している。木星のガス組成に、いわゆる氷成分がどの程度寄与しているかが問題であったが、ガリレオプローブが見出した水の欠乏は特殊なホットスポット領域に突入したためと考えられており、まだ決着がつかない。新たな突入プローブにより大気組成を確定する必要がある。サンプルを採取し解析を行うことは非常に価値の固いことである。

次に氷衛星の地質構造、イオの火山活動、非氷衛星の表面組成測定を行い、イオやエウロパといった潮汐エネルギー天体の機構を解明するために、リソスフェアの厚さをレーダやレーザ高度計の計測から求めることも大きな目的として、2020年代後半には木星探査を行う計画が立てられている [17]。木星の衛星の中で最も内側に位置するイオには数百もの活火山が存在し、その噴火による噴煙が場合によっては上空 400km 近くにまで吹き上げられているということが木星探査衛星ガリレオ (NASA) などの調査により知られている。もし、イオの火山噴煙を詳しく分析しそれが木星の磁気圏に与える影響やイオの火山活動のメカニズムを知ることができれば、小さな太陽系とも言われる木星の解明につながる。それにより、太陽系の歴史の解明や今後の宇宙開発における木星の利用にもつながる可能性がある。このような火山ガスを回収し地球に持ち帰ることにより地球上で詳細な成分分析を行うことが可能となれば、木星以遠のサンプルリターンを現実化させるため木星や木星衛星の探査を行う意義は非常に大きい。

エアロキャプチャを行う際の大气突入条件は木星から地球への帰還を仮定し、近似軌道連結法を用いて決定した。近似軌道連結法とは図 4.5 に示した様に、地球または木星のそれぞれの近傍では、地球や木星の引力が太陽の引力よりも顕著に作用する球状の範囲（影響圏）を仮定し、影響圏の中ではその引力だけで軌道を決定し、範囲の外側では太陽の引力のみで軌道が決まるとして、いくつかの軌道をつないで全体の軌道を作る方法である [6]。次の 3 つのフェーズに分けて軌道を決定した。

- 出発フェーズ（木星影響圏の範囲）：木星と宇宙機の 2 体で運動を定義し、軌道は木星を焦点とした双曲線軌道となる。
- 中間フェーズ：太陽と宇宙機のみで運動を定義し、軌道は太陽を焦点とした遷移楕円軌道となる。

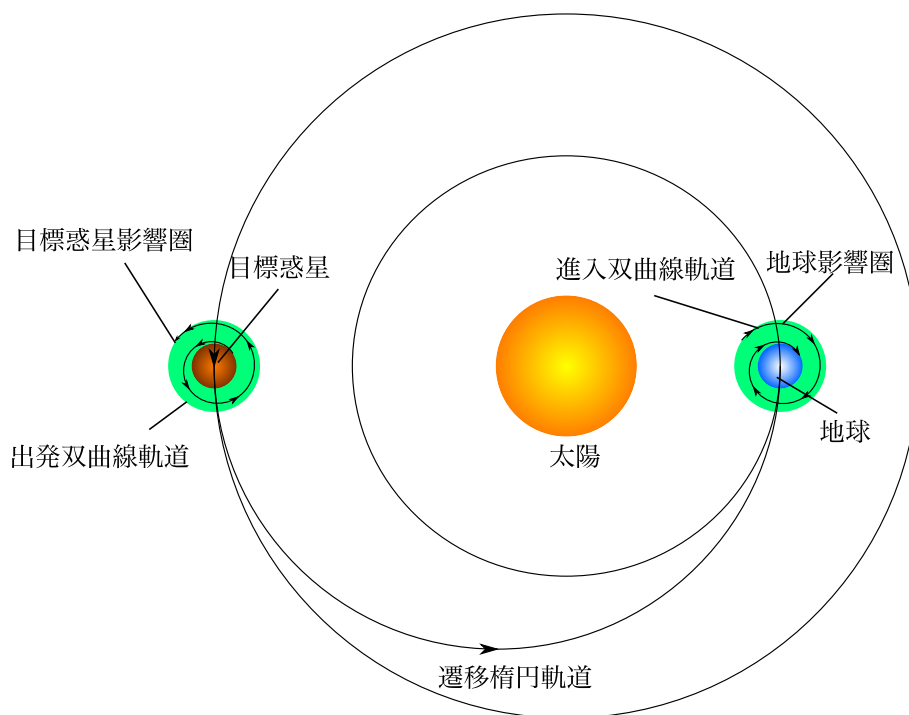


図 4.5: 近似軌道連結法

- 進入フェーズ（地球影響圏の範囲）：地球と宇宙機との2体で運動を定義し、軌道は地球を焦点とした双曲線軌道となる。

このとき、惑星間飛行の軌道は影響圏の双曲線軌道とその内側の楕円軌道を、影響圏の境界でつないでいくことによりつくる。影響圏の大きさは次の様に定義される。

$$r_e = r \left( \frac{m}{m_s} \right)^{\frac{2}{5}} \quad (4.2)$$

ここで、 $r_e$  は影響圏の半径、 $r$  は惑星や地球の平均軌道半径、 $m$  は惑星や地球の質量、 $m_s$  は太陽の質量である。

地球の影響圏は約  $0.9 \times 10^6$ 、木星で  $48.2 \times 10^6$  であり、木星、木星間のホーマン軌道に対して1%以下の範囲に過ぎないため、中間フェーズの軌道は両端の影響圏の大きさは無視し、木星の中心から地球の中心に飛行するとして差し支えないと言える。そのため、中間フェーズでは木星の中心から地球の中心までのホーマン軌道を過程した。

近日点の軌道半径  $r_p$  を地球の平均軌道半径  $149.59 \times 10^6$  km、遠日点の軌道半径  $r_a$  を木星の平均軌道半径  $778.412 \times 10^6$  km とすると、近日点での速度  $v_p$  は 38.56 km/s となる。地球の公転速度が 29.7859 km/s なので、地球に対して宇宙機が持つ相対速度  $v_\infty$  は 8.79km/s となる。双曲線軌道は進入



点での宇宙機と地球の相対的な位置関係を仮定すると一意的に決まり、大気密度が  $10^{-10}$  kg/m<sup>3</sup> となる高度 200 km[7] を大気縁とすると、その点での速度  $v$  は

$$v = \sqrt{\frac{2\mu}{R} + V_{\infty}^2} = 14.09\text{km/s} \quad (4.3)$$

となる。ここで  $\mu$  は地球重力定数、 $R$  は高度である。この速度条件で高度 200 km から地球の突入し減速を行い、再び宇宙へと離脱し周回軌道へと入る。

エアロキャプチャでは空気抵抗により大きな減速を行うので、機体は厳しい空力加熱を受ける。厳しい空力加熱から保護してエアロキャプチャを実現させる方法として主に 2 つの方法が考えられている。それらのイメージを図 4.6 を示す。一つは、空力的な形状を有する剛体のエアロシェルに衛星を収納するタイプのものである。エアロシェルは空力的な機能を果たすと同時に高速での大気飛行中に受ける厳しい空力加熱から探査機を保護する TPS (熱防御システム) となる。しかし、TPS を装備することは機体重量の増加、ペイロードを減少を意味するため、ペイロードや燃料セーブということを考えると簡単な形状であることが求められる。鈍頭の形状をしているもので、0.25 程度の揚抗比を有する。エアロシェルは外側の熱防御材、エアロシェルと熱防御材を接着する部分、構造部材から成っているが、過去に行われたエントリーブローブミッションと同様の構造であるため、現在実現に向けての技術が最も整っている機体で、残る課題としては形状の検討、アブレータの厚みの検討、高温に耐える接着剤の検討、軽量で強度を有する構造部材の検討などがある。

もう一つはパラシュートのような大面積の減速装置を展開する方法が考えられている [2]。この方法は、機体の弾道係数を下げることによりエアロシェルタイプと比べて高い高度で必要な減速を行うため、空力加熱を低減することができる。現在までに研究が進められているエアロキャプチャの機体のうち弾道係数が低いグループに属するものとしては、バリユート [9][12][13] やインフレータブルエアロシェル [10] などがある。これらの特徴としてはバリユートなどの減速装置は TPS を持たない極薄のフィルム材料によって作成され [14] 低弾道係数化がなされ、揚力を発生しないため機体は弾道飛行を行うことである。飛行制御の方法としては、探査機に搭載された制御システムにより適切なタイミングで減速装置を切り離すことにより抵抗の調節を行う。近年高温に耐える低面積密度のフィルム素材の発展が著しく、膜素材の収容性の良さや軽量であることを利用して様々な宇宙構造物に適用され、製造法や展開硬化法などの技術開発も進んでいる [11]。

しかし、どちらも突入経路角の許容範囲が狭いことが問題となっている。膜面を展開しエアロシェルとする事で、高弾道係数のグループと低弾道係数のグループのメリットを併せ持てばどのような条件であっても応用しやすいと考えられる。よって本研究ではそのような、膜面によるエアロシェルで高揚力を発生させることを目的としている。

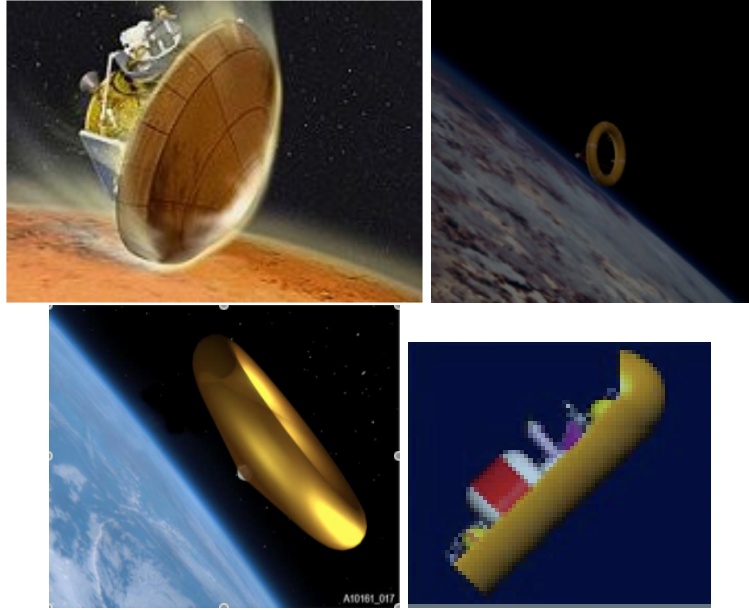


図 4.6: 鈍頭形状剛体エアロシェル (左上), 後曳型バリュート形状 (右上) [15], 付着型バリュート形状 (左上) [15], 細長形状剛体エアロシェル (下) [2]

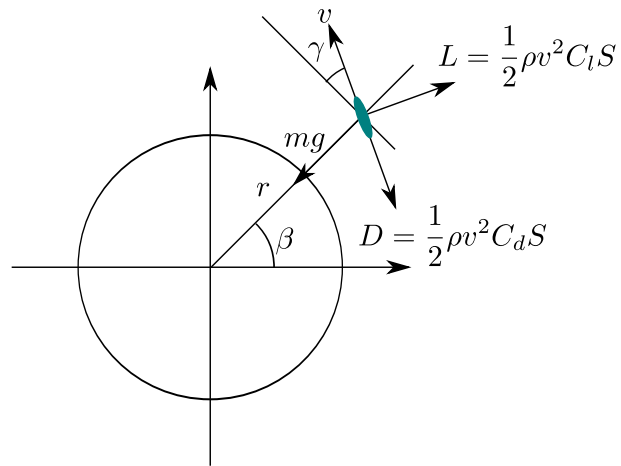


図 4.7: 軌道計算の解析モデル

先に述べたように従来用いられてきたカプセル型など高弾道係数の宇宙機では突入経路角の許される幅が狭く、高度な制御を必要とすることがエアロキャプチャの実用化を難しくさせてきた。エアロキャプチャの成功にはこの突入経路角の許される幅により、どの程度の精密な制御が必要か決定されるため重要である。大気飛行中の軌道計算は、衛星を質点と考え、2次元面内の運動を考えた。座標を図4.7のようにとると、支配方程式は以下のようになる。

$$\frac{dv}{dt} = -g \sin \gamma - \frac{1}{2} \rho v^2 \frac{1}{C_B} \quad (4.4)$$

$$v \frac{d\gamma}{dt} = -\frac{v^2}{r} \cos \gamma - g \cos \gamma + \frac{1}{2} \rho v^2 \frac{1}{C_B} \frac{C_L}{C_D} \quad (4.5)$$

$$\frac{d\beta}{dt} = \frac{v}{r} \cos \gamma \quad (4.6)$$

$$\frac{dr}{dt} = v \sin \gamma \quad (4.7)$$

ここで、 $r, \beta$ は地球の中心から宇宙機までの距離及び角度、 $m$ は宇宙機の質量、 $g$ は重力定数、 $\gamma$ は宇宙機までの位置ベクトルの法線からとった角度、 $L, C_L$ を揚力、揚力係数、 $D, C_d$ は抵抗、抵抗係数、 $\rho$ は空気密度、 $S$ は抵抗面積、 $C_B$ は弾道係数であり、

$$C_B = \frac{m}{C_d S} \quad (4.8)$$

と表される。これらの式から揚抗比が一定であるとすれば起動は初期条件と揚抗比、弾道係数のみで決定することがわかる。図4.8は弾道係数を0.018と固定し、図4.9は揚抗比を0.5と固定した場合の結果である。横軸は突入角度、縦軸は遠地点距離を示している。このように弾道係数は低いほど、揚抗比は高いほど突入の許容角度が広がり、エアロキャプチャは成功しやすい。突入の許容角度に関しては、弾道係数を0.018とかなり小さい条件で計算したにも関わらず、揚抗比が0の場合ではほぼ0となっていることからわかるように、弾道係数より揚抗比が支配的である。よって、最適化の適合度評価は揚抗比のみで評価した。

図4.8、図4.9での遠地点距離を500 kmとした場合の軌道の速度と高度を図4.10と図4.11に示す。これから高度としては130km程度から90km程度の領域で減速を行っているのがわかる。この経路をとったときの壁面空力加熱率を図4.12に示す。加熱率は

$$q = \frac{1}{2} \rho v^3 \quad (4.9)$$

膜面の温度は

$$T = \left( \frac{q}{2\sigma\epsilon} \right)^{\frac{1}{4}} \quad (4.10)$$

として推算した。ここで $\epsilon$ はStefan-Boltzmann定数、 $\epsilon$ は放射率で0.9とした。加熱率はこのようによりに低い高度まで入る経路は熱的に厳しくなる

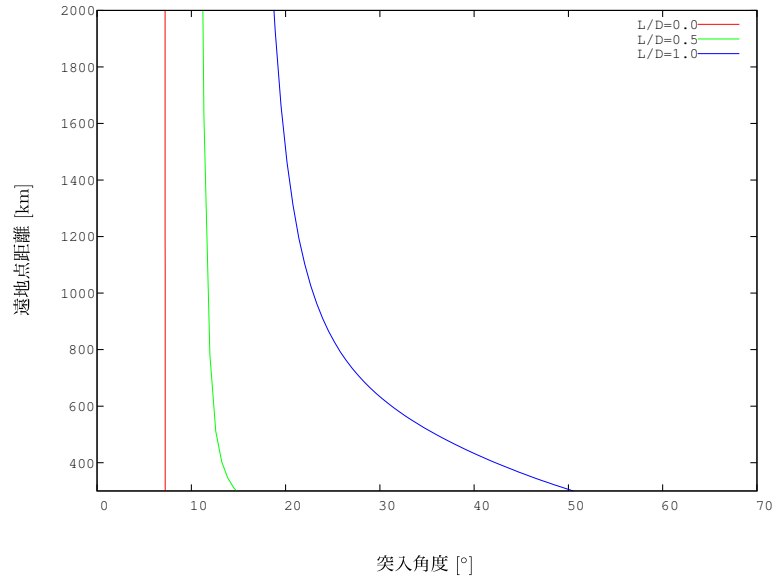


図 4.8:  $L/D$  毎の突入角と到達高度

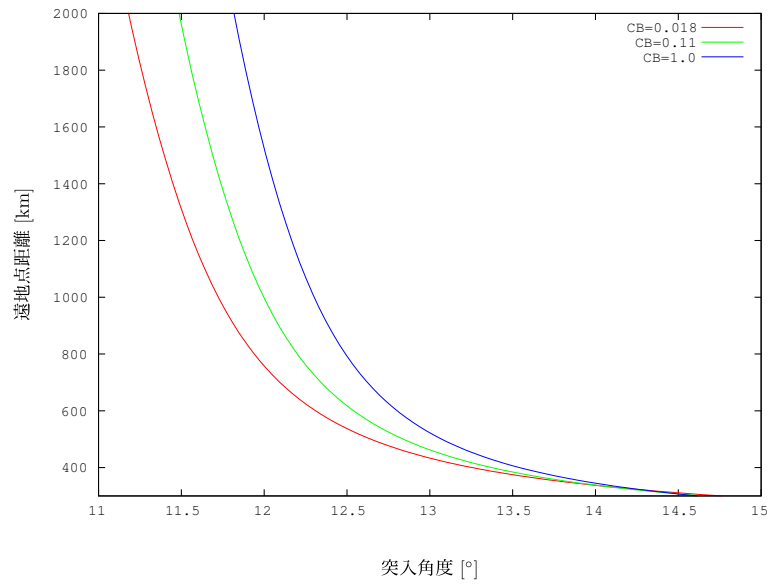


図 4.9:  $C_B$  毎の突入角と到達高度

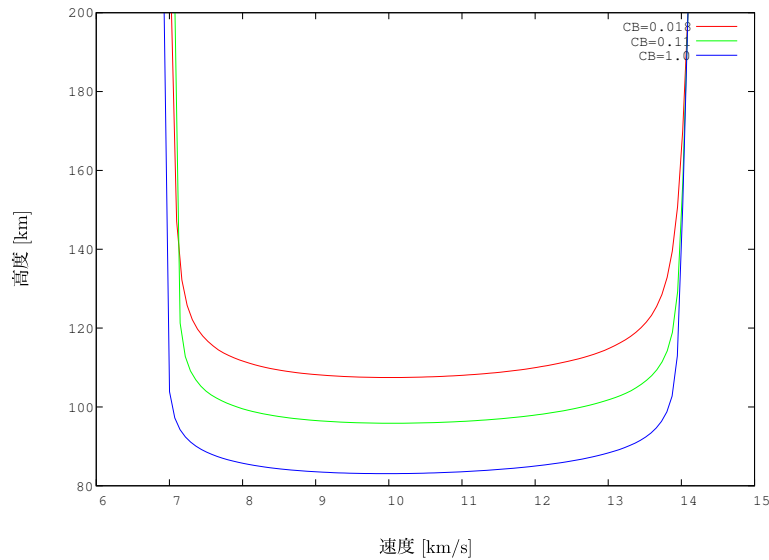


図 4.10:  $C_B$  毎の速度-高度

ため、本研究での想定した機体では高めの経路をとる事が望ましく、それにより連続流ではなく、希薄気体流れとなる。これらのグラフからどのような経路においても大きく減速を始めるのは 13 km/s 程度からであるため、最適化の条件は 13 km/s とした。高度によってクヌーセン数は異なるため、クヌーセン数は 0.05~0.5 で最適化を行った。

### 4.3 最適化

これまで検証を行った形状を近似し解析を行う DSMC 法を用いて最適化を行った。最適化条件はクヌーセン数は 2次元で 0.05~0.5, 3次元では 0.05 とした。2次元の最適化モデルは図 4.13 に示した簡単な形状を用いた。x の値は等間隔とし、y1 は固定、y2 ~ y6 までの 5 変数を用い適合度は揚抗比のみで評価した。初期形状は乱数により決定している。

最適化計算の世代毎の進化の履歴を図 4.14, 図 4.16, 図 4.18, 図 4.3, 図 4.22, 図 4.24 に、進化の途中形状を図 4.15, 図 4.17, 図 4.19, 図 4.21, 図 4.23, 図 4.25 に示す。最適化の履歴はシプレックスアルゴリズムを用いたため、最適解が更新された場合にのみプロットされる。そのため履歴の最終世代数が異なるが、全てのケースにおいて 100 世代まで計算を行った。クヌーセン数が 0.05 や 0.1 の様な比較的密度の高い領域内で最適化を行った結果では連続流に近い滑らかな形状、つまり直線に近い形状を多く有する遺伝子が最良解として得られた。翼全体で空気力を受けると揚力も増加するが、抵抗も増加するため、揚抗比としての向上は得られず全体ではなく一部を用いて

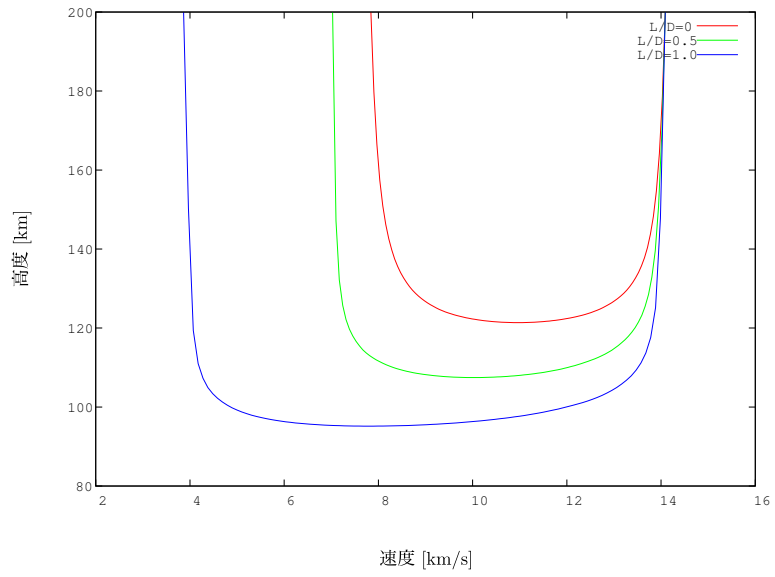


図 4.11:  $L/D$  毎の速度-高度

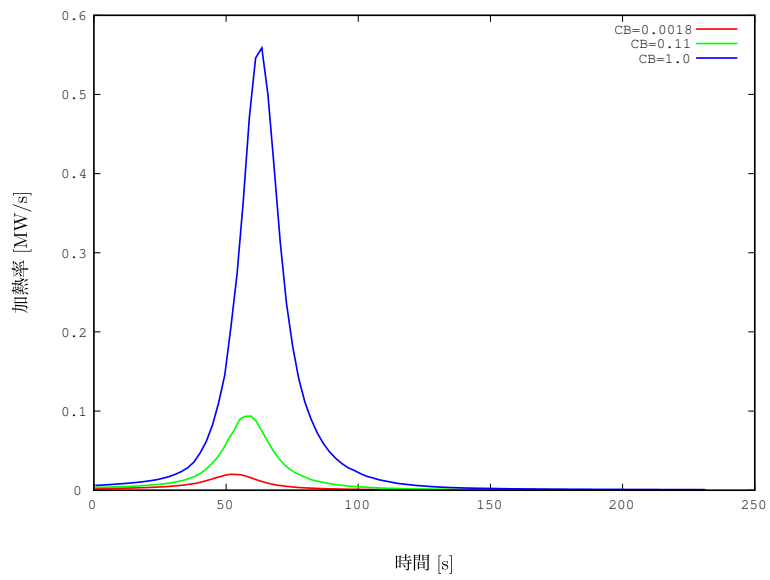


図 4.12:  $C_B$  毎の加熱率の時間履歴

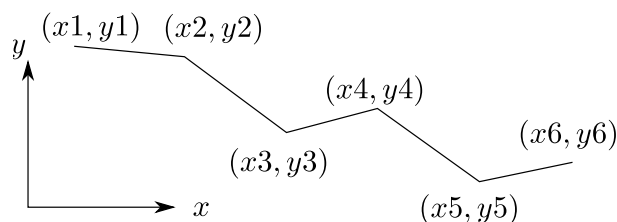


図 4.13: 2D 最適化モデル

空気力を受ける形状が最良解となった。

クヌーセン数 0.2 以上領域では希薄になるにつれ、形状の変化に対して空力係数の変化はあまり無くなっていく。そのため、滑らかでない形状であっても最良解として生き残った。これらの共通する傾向としては、受ける空気力が小さいため、前縁付近で一度圧縮し、その圧縮空気を中間付近から後縁付近で受ける形状が最良解となった。密度があがった領域ではクヌーセン数が 0.05 や 0.1 のように直線的な形状をとった。

この最適化は 4 コアを用いて並列化を行い、最適化を行った。並列化の分岐は図 4.2 の一番上の段、適合度の評価を 1 つの遺伝子に 1 つの CPU を割り当て、4 つの遺伝子を並列化した。それぞれに *flag* と法線ベクトルを独立して与えれば共通の格子で計算が行えるため、並列化を行った場合にもメモリ消費の増加は従来の格子を用いた手法と比較すると押さえることが出来る。この場合の計算時間は 1 つの遺伝子を 4 つの CPU を用いて適合度評価を行った場合のほぼ 1/4 の時間であった。DSMC 法は粒子の総数が計算の過程で変化するため、並列化をおこない辛い。それが原因となり 4 つの CPU を用いて 1 つの遺伝子の適合度評価を行った場合にも速度の向上があまり得られていない。最適化であれば 1 つの遺伝子に 1 つの CPU を割り当てることで簡単に速度の向上を得ることが出来る。今後の PC はコア数の増加、共有メモリシステムとしての進化の方向に進む事が予想されるため、このような粒度の大きい並列化が有効となることから、共通の格子で異なる形状の解析が可能であることのメリットは大きい。

図 4.26 にクヌーセン数毎に最適化を行った結果を横軸にクヌーセン数としてプロットしたグラフを示す。抵抗係数がクヌーセン数 0.3 のときに多少ラインから外れるように見えるが、クヌーセン数が小さくなるにつれ性能も上昇し傾向は正しく出ている。本形状近似 DSMC 法で正しい傾向が得ることが出来た。

3次元最適化は図 4.27 に示したモデルを用いた。  $x_1, y_1, z_1, x_3, y_3$  は固定とし最適化を行った。計算条件は 2次元最適化のクヌーセン数 0.05 と同様である。図 4.28 に 3次元最適化の履歴を、図 4.29、図 4.30 に 6 世代と 28 世代の最良解形状を示す。3次元でも 2次元同様に前縁で圧縮し、その圧縮空気を下面で受け、揚力を発生させる形状が最良解として得られた。しかし、3

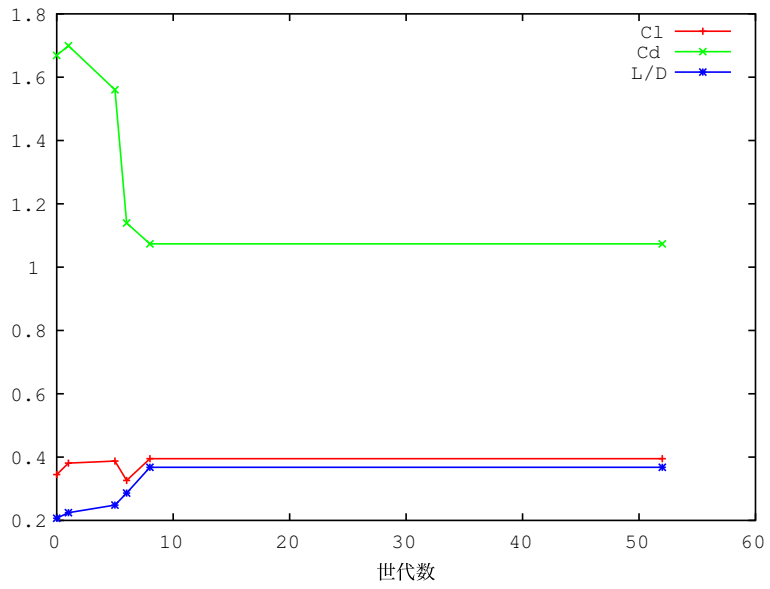


图 4.14: 最適化履歴 (Kn0.05)

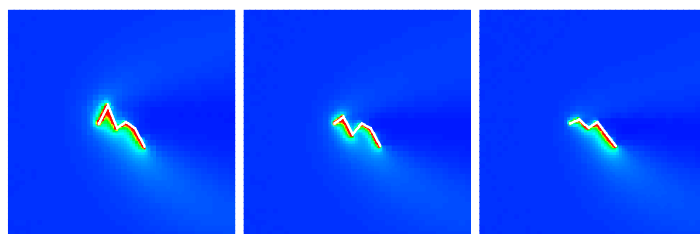


图 4.15: 密度分布 (Kn0.05) 5, 6, 51 世代



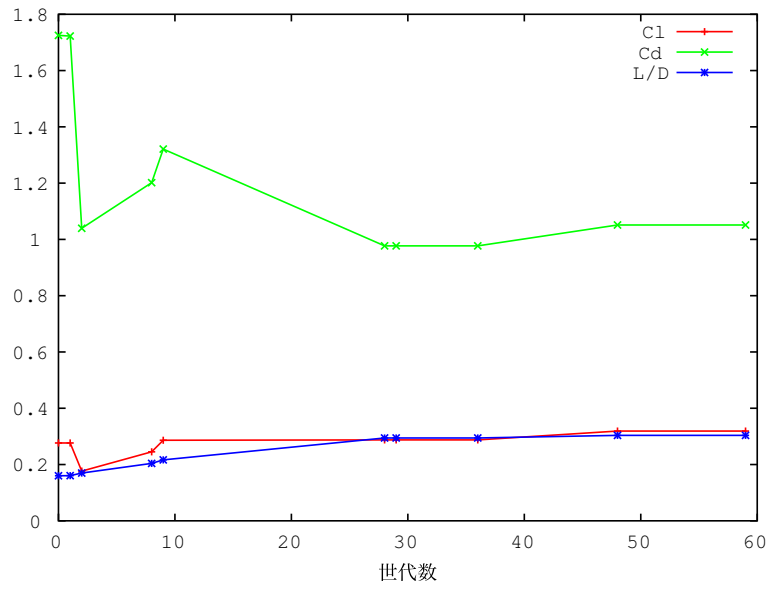


图 4.16: 最適化履歴 (Kn0.1)

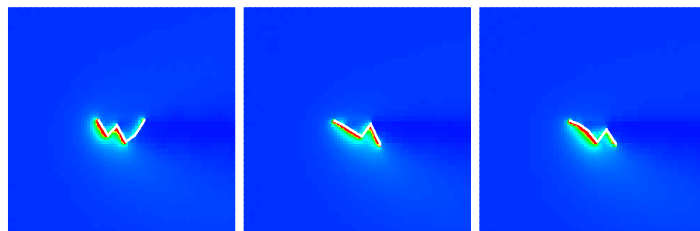


图 4.17: 密度分布 (Kn0.1) 2, 28, 59 世代

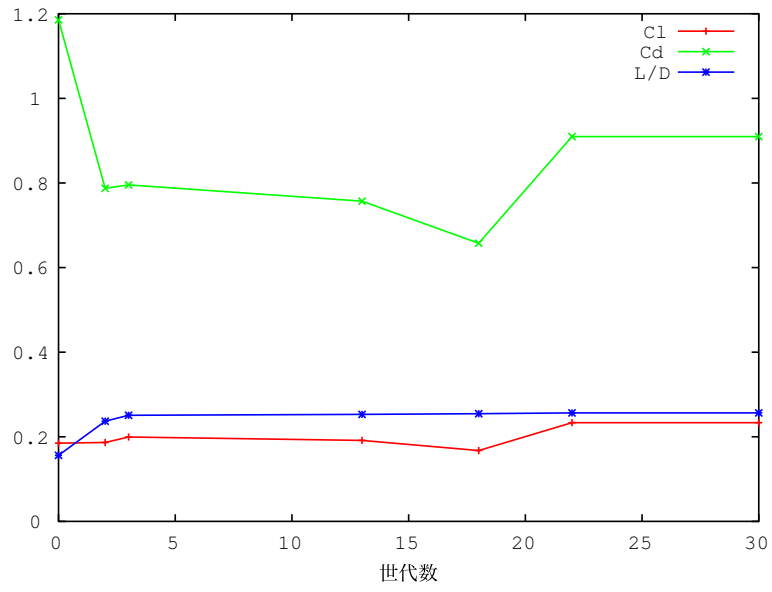


图 4.18: 最適化履歴 (Kn0.2)

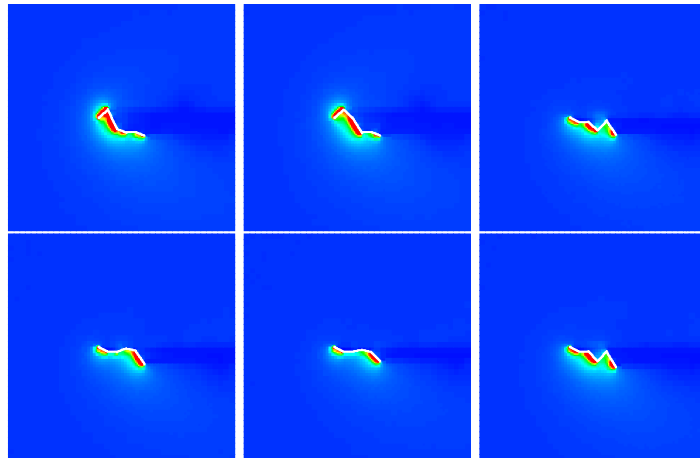


图 4.19: 密度分布 (Kn0.2) 0, 2, 3, 13, 18, 30 世代

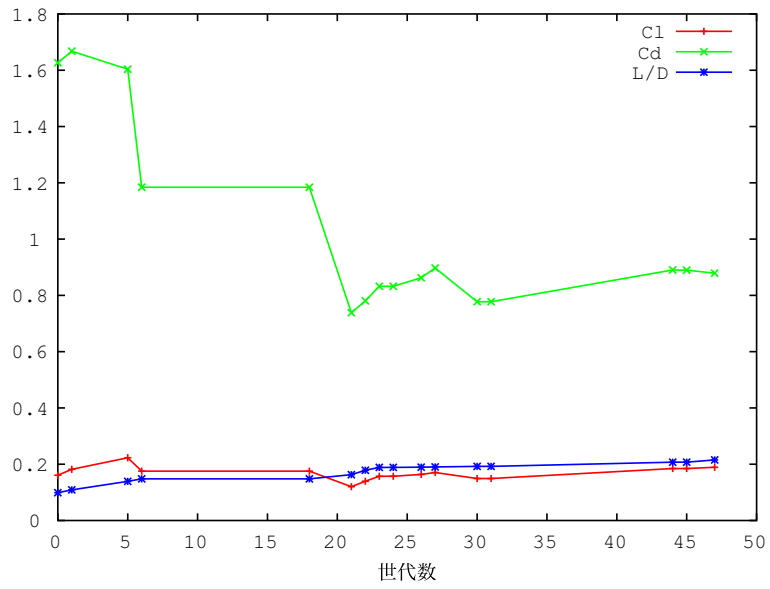


图 4.20: 最適化履歴 (Kn0.3)

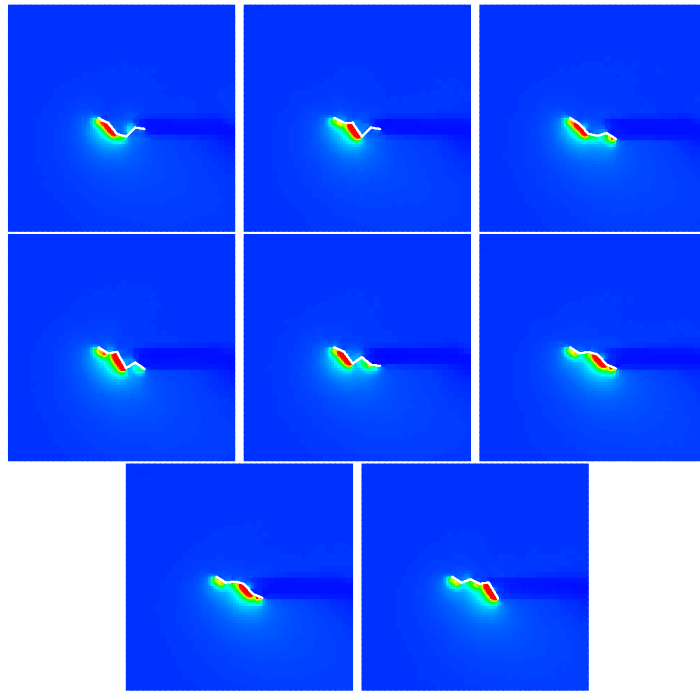


图 4.21: 密度分布 (Kn0.3) 21, 22, 23, 27, 30, 44, 45, 47 世代

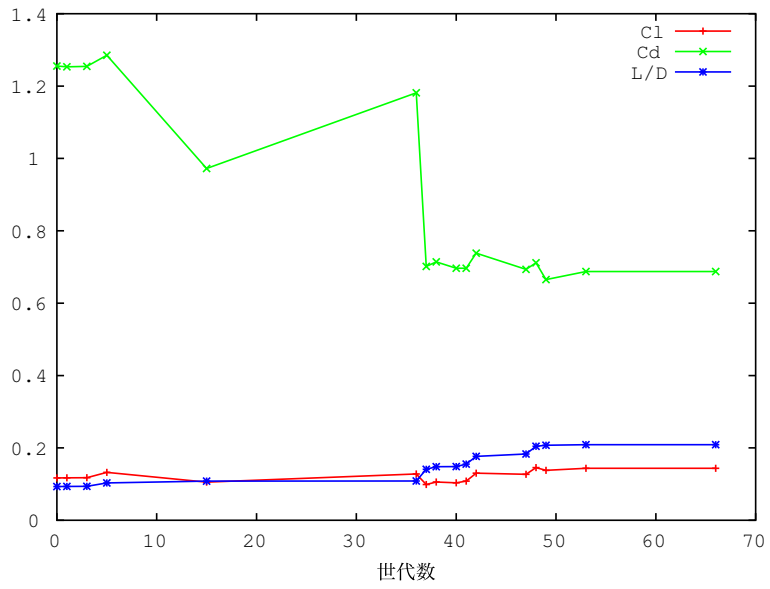


图 4.22: 最適化履歴 (Kn0.4)

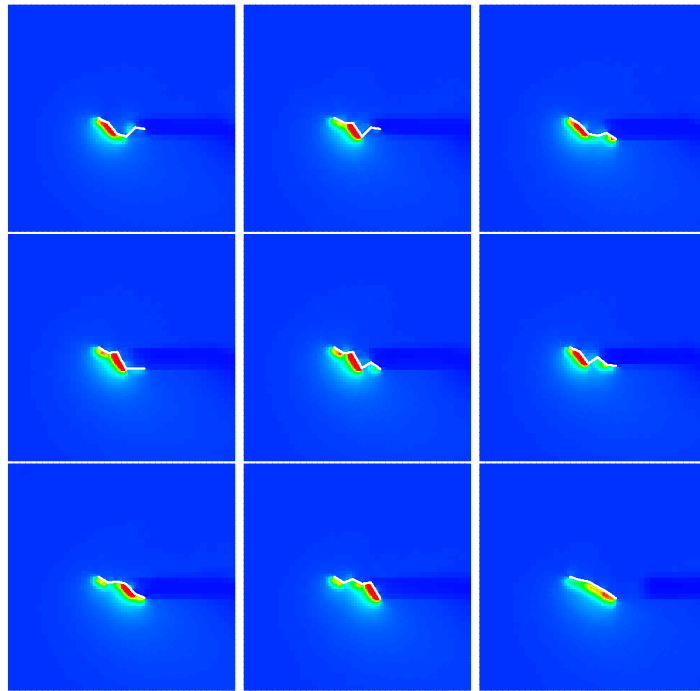


图 4.23: 密度分布 (Kn0.4) 21, 22, 23, 26, 27, 30, 45, 47, 66 世代

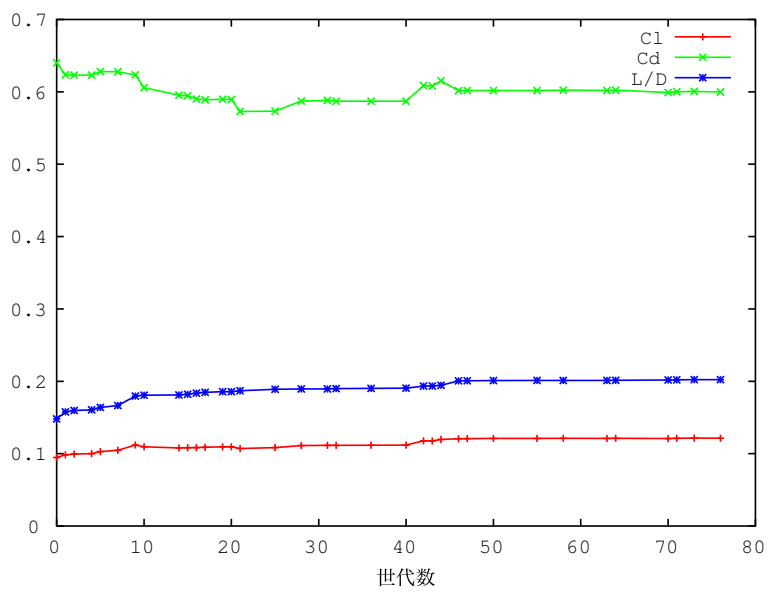


图 4.24: 最適化履歴 (Kn0.5)

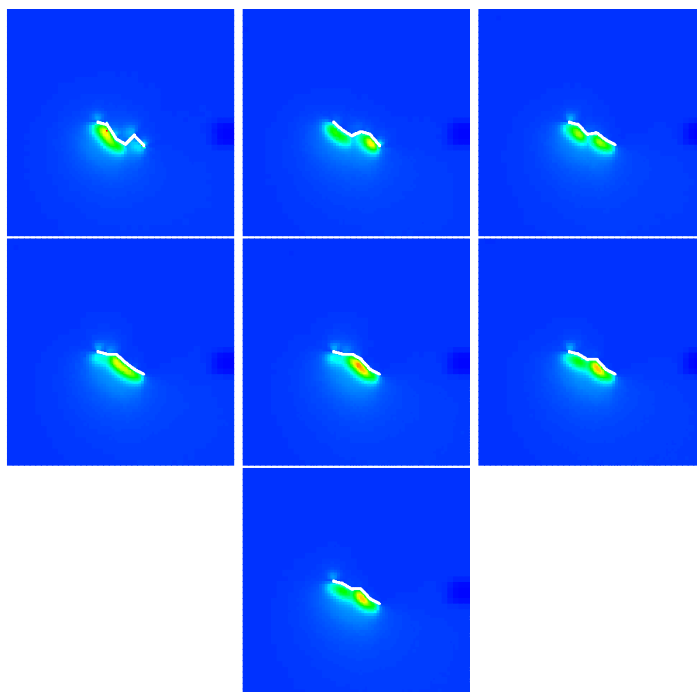


图 4.25: 密度分布 (Kn0.5) 0, 9, 32, 43, 44, 46, 76 世代

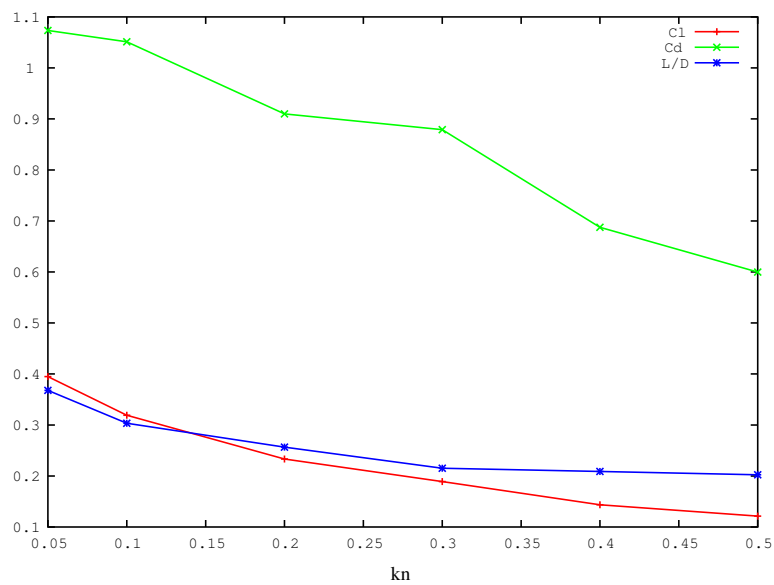


図 4.26:  $Kn$  に対する空力係数

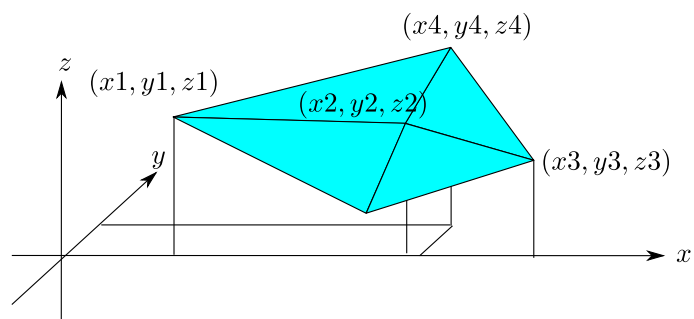


図 4.27: 3次元最適化モデル

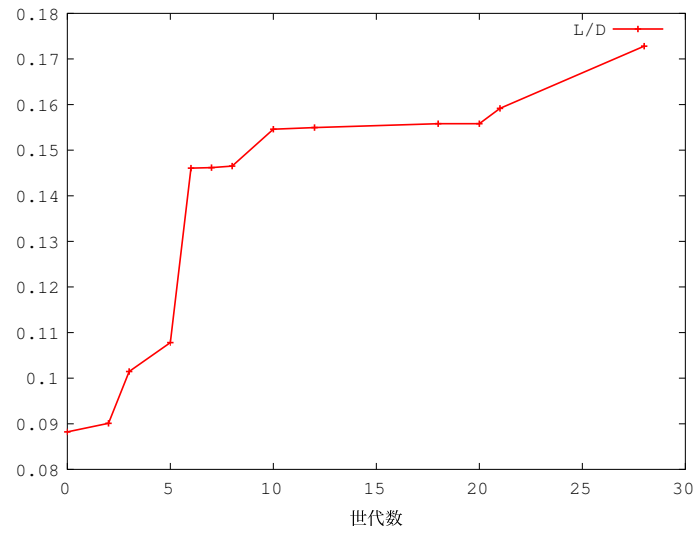


图 4.28: 3次元最適化結果

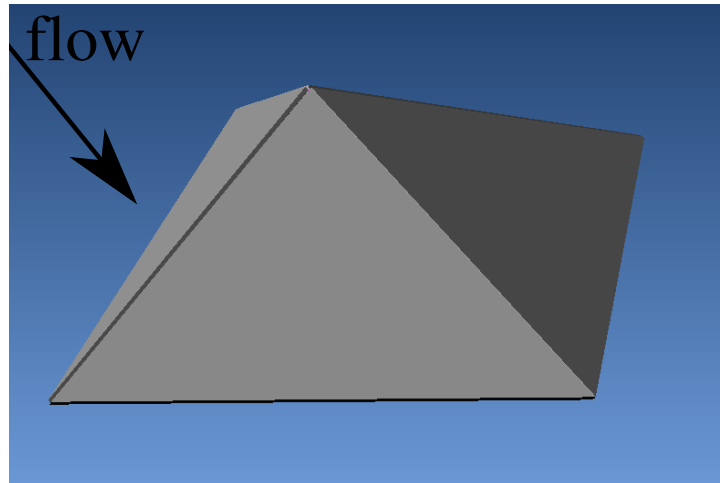


图 4.29: 3次元最適化形状 (6世代)

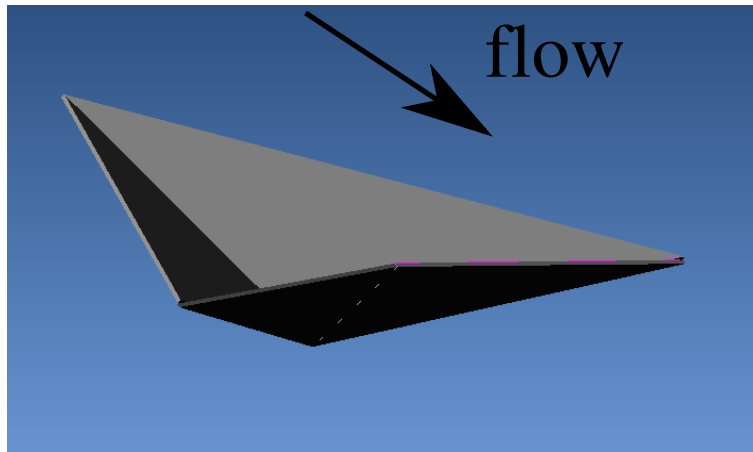


図 4.30: 3次元最適化形状 (28世代)

次元では2次元ほど下面側での圧縮が出来ないため、揚抗比は2次元程高い値は得る事が出来なかった。しかし、3次元であっても本研究で用いた近似DSMC法により、最適化を行うことが出来た。初期形状は乱数で発生させたが、この結果が得られたためあらゆる形状に適用出来ると考えられる



## 第5章 結言

実際の境界となる面を含むセルを計算上の境界とし、そのセルに実際の境界と同様の法線ベクトルを与えることにより形状を近似し計算をおこなった。最小格子幅と平均自由行程の最小値とを比較し、最小格子幅より平均自由行程の方が十分に大きければ非常に良好な解を得る事が出来た。法線ベクトルを付加しなかった場合には凹面となる領域に粒子が集まり、その結果その領域の平均自由行程が小さくなり粒子数の増加が続き定常解を得る事が出来なかった。

計算時間は形状を近似する際に予め法線ベクトルを求めているために、法線ベクトルが位置により異なる場合でも従来の DSMC 法のように新たに計算し直すことがないこと、衝突し反射を行うか否かの判定についても計算負荷を低減しているため高速化された。形状が複雑化した場合にもプログラムを実行する初期の段階でどの位置に境界とすべきセルがあるかの判別を行う過程が複雑化するのみとなるため、時間ステップを進めるときのアルゴリズムは何も変わらない。そのため複雑化した場合にも計算時間には影響が少ない。よって、更なる複雑な形状への適用も可能であろうと考えられる。

最小格子幅を決定するにあたり、平板では平均自由行程の  $1/2$  で良好な解を得る事が出来たが、円柱では  $1/4$  まで小さくしなければ良好な解は得られなかった。本研究で用いた手法は凹面との相性が悪いため、流れ方向に対して垂直では無いが垂直に近い形状が良好な解を得辛い。本研究での最適化の適合度の評価は揚抗比のみでおこなったため、抵抗の少ない形状が選択されるため最終的に得られた最良解では垂直に近い形状が出る事が無いため適用出来た。よって、本研究で用いた手法は、まず形状を吟味し計算が終わった後に平均自由行程と格子幅の関係を確認する必要があるが、どの様な形状であってもアルゴリズムを変更せずに適応出来るため有用である。乱数を用いて決定した形状でも適用可能であった。

この形状近似 DSMC 法を用いて最適化をおこなった。得られた結果としては、クヌーセン数の増加に伴い受ける空気力が減少するた多段階的に圧縮し、その圧縮空気を受けることで揚力を増加させる形状が最良解となった。クヌーセン数の小さい領域では、連続流に近いいため、滑らかな形状、直線に近い領域を多く有するものが最良解となった。

## 参考文献

- [1] 久保田 弘敏, 鈴木 宏二郎, 綿貫 忠晴, 『宇宙飛行体の熱気体力学』, 東京大学出版会, 2002
- [2] 中村久美子, 『土星における低弾道係数エアロキャプチャ生成の実現性』, 東京大学大学院修士論文, 2006
- [3] Gerald D. Walberg, “A Survey of Aeroassisted Orbit Transfer” , J. Spacecraft, Vol.22, No.1, Jan-Feb, 1985.
- [4] G. A. Bird, ”Molecular Gas Dynamics and the Direct Simulation of Gas Flow” ,Oxford,1994
- [5] 保原 充, 大宮司 久明, 『数値流体力学 基礎と応用』, 東京大学出版, 1992
- [6] 茂原 正道, 木田 隆 1, 『宇宙工学入門 II』, 培風館, 1998
- [7] 日本航空宇宙学会 編, 航空宇宙工学便覧, 丸善, 1992
- [8] W. G. Vincenti and C. H. Kruger, ”Introduction to Physical Gas Dynamics”, Wily New York, 1965
- [9] Glen J. Brown, C. Epp, C. Graves, S. Lingard, M. Darley, K. Jordan, “Hypercone Inflatable Supersonic Decelerator” , 17th AIAA Aerodynamic Decelerator Systems Technology Conference, Monterey, AIAA 2003-2167, May, 2003.
- [10] Jim Stein, Daniel T. Lyons, Richard G. Wilmoth, “Trailing Ballute Aerocapture: Concept and Feasibility Assessment” 39th AIAA/ASME/SAE/ASEE Joint Propulsion Conference, Huntsville, AIAA 2003-4655, July 2003.
- [11] R. E. Freeland, G. D. Bilyeu, M. M. Mikulas, “Inflatable Deployable Space Structures Technology Summary” , 49th International Astronautical Congress, Melbourne, IAF-98-I.5.01, September 28-October 2, 1998.
- [12] Jim Stein, Daniel T. Lyons, Richard G. Wilmoth, “Trailing Ballute Aerocapture: Concept and Feasibility Assessment” 39th AIAA/ASME/SAE/ASEE Joint Propulsion Conference, Huntsville, AIAA 2003-4655, July 2003.

- [13] A. Rasheed, K. Fujii, H. G. Hornung, J. L. Hall, “Experimental Investigation of the Flow over a Toroidal Aerocapture Ballute”, 19th Applied Aerodynamics Conference, Anaheim, AIAA 2001- 2460, June, 2001.
- [14] Rohrschneider, R.R., and Braun, R.D., “Static Aeroelastic Analysis of a Thin-Film Clamped Ballute for Titan Aerocapture” ,1st International ARA Days – Atmospheric Reentry Systems, Missions and Vehicles Conference, Arcachon, France, July 2006.
- [15] Masciarelli, James P. Lin, John K. H. Ware, Joanne S. Rohrschneider, Reuben R. Braun, Robert D. Bartels, Robert E. Moses, Robert W. Hall, Jeffery L.,”Ultra Lightweight Ballutes for Return to Earth from the Moon”, 47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference May 2006, Newport, RI.
- [16] 伊庭斉志著, 遺伝的アルゴリズムの基礎 GA の謎を解く, オーム社, 1994
- [17] 月惑星探査推進チーム・太陽系探査ロードマップ検討小委員会報告, 太陽系探査科学の進むべき方向 -我々の地球をより良く理解するために- , JAXA 月惑星探査チーム、太陽系探査ロードマップ検討小委員会, 2007

## 謝辞

本研究を行うことにあたり，二年間渡り親切丁寧なご指導を頂きました鈴木宏二郎准教授に心より感謝申し上げます。日頃より，御指導，御助言頂きました綿貫忠晴助教，今村宰助教に感謝致申し上げます。

同研究室の高間良樹氏，児島佳敬氏，松本達矢氏，若山一樹氏をはじめ，鈴木研究室の博士課程の方々，後輩の皆様には公私に渡りお世話になりました。皆様の暖かいご協力に感謝を申し上げます。

最後に研究生生活を支えてくださった家族に様々な局面にて多大なる御協力いただきました。ここに感謝の意を表します。

2008年2月13日

板橋直亮

## 本論文に関する学会発表

DSMC 法による希薄気体解析を用いたエアロキャプチャ機体形状の L/D 最適化, 第 51 回宇宙科学技術連合講演会, 2007 年 10 月 29 - 31 日