

修士論文

古典計算と量子計算による
新たなボルツマンソルバー
(Solving the Boltzmann equation with classical
and quantum computers)

2024年1月

東京大学大学院 理学系研究科物理学専攻

山崎壮一郎

目次

第 1 章 Boltzmann 方程式	3
1.1 Boltzmann 方程式の導出	3
1.2 Boltzmann 方程式の応用	5
1.3 Boltzmann 方程式のシミュレーション手法	6
1.3.1 N 体シミュレーション	6
1.3.2 Direct Simulation Monte Carlo 法	7
1.3.3 格子ボルツマン法	8
1.3.4 直接計算	8
第 2 章 量子計算	10
2.1 Quantum bits	10
2.1.1 1 qubit system	10
2.1.2 Multiple qubits system	12
2.2 量子ゲート	13
2.2.1 1 qubit gate	13
2.2.2 Multiple qubits gate	15
2.2.3 ゲートの分解	17
2.3 量子アルゴリズム	18
2.3.1 インクリメントおよびデクリメント	18
2.3.2 量子フーリエ変換	19
2.4 量子トモグラフィ	21
2.4.1 1 量子状態への量子トモグラフィ	21
2.4.2 n 量子状態への量子トモグラフィ	23
2.5 量子コンピューターの実機	23
2.6 QRAM	25
第 3 章 Vlasov 方程式の量子計算による加速	27
3.1 先行研究	27
3.1.1 微分方程式の量子シミュレーション手法	27
3.1.2 Vlasov 方程式の量子シミュレーション手法	27
3.1.3 Reservoir 法	28
3.1.4 Reservoir 法を用いた量子アルゴリズム	29
3.2 Methods	32
3.2.1 空間依存する力場 $F(x)$ におけるシミュレーション	32
3.2.2 空間および時間依存する力場 $F(x, t)$ におけるシミュレーション	34

3.2.3	自己重力系におけるシミュレーション	34
3.2.4	3次元のシミュレーション	36
3.3	テスト計算	37
3.3.1	Free streaming	37
3.3.2	Jeans 不安定性の成長モード	38
3.3.3	Jeans 不安定性の減衰モード	40
3.3.4	情報の引き出しの効率	43
3.4	計算量	47
第4章	Boltzmann 方程式の実装	49
4.1	直接シミュレーションの例	49
4.1.1	スペクトル法	49
4.1.2	先行研究における高速スペクトル法	50
4.2	高速化の方針	51
第5章	結論	54
付録A	古典的な論理ゲート	63
A.1	NOT	63
A.2	AND	63
A.3	OR	63
A.4	XOR	64
A.5	NAND、NOR、XNOR	64
付録B	移流方程式の性質と解法	66
B.1	移流方程式の性質	66
B.2	移流方程式の解法	66
B.2.1	Forward in Time and Centered Difference in Space スキーム	67
B.2.2	風上差分法	67
B.2.3	その他の解法	67
付録C	自己重力系における重力計算	68
付録D	Jeans 不安定性における揺らぎの成長と減衰	71
付録E	特異値分解	74

第1章 Boltzmann 方程式

本研究のタイトルにも登場するボルツマンというのは、希薄気体の運動を記述する方程式である Boltzmann 方程式からくるもので、1872 年に Ludwig Boltzmann によって提出された論文 [7] に掲載されたものである。現在この方程式は様々な応用範囲を持ち、流体計算はもちろんのこと、物性分野や量子論、そして宇宙分野にも多数の応用が存在する。宇宙分野では長らく N 体法と呼ばれるシミュレーション手法が使用されてきており、様々な成果を挙げている。 N 体法は、シミュレーション内でのマクロな範囲では正しい結果を示すことが歴史的にわかっているが、速度についてミクロな範囲で正しくシミュレーションすることはできていない。その一方、本研究も含まれる直接計算と呼ばれる手法はミクロな部分まで正しくシミュレーションすることができる。本研究の方法を用いることで、より精度の高いシミュレーションができるようになることが期待されるだけでなく、 N 体法のシミュレーション結果がなぜマクロな範囲で成り立つのかをミクロな視点から解析することができるようになることも期待される。

1.1 Boltzmann 方程式の導出

質量 m の粒子が位相空間 (\mathbf{x}, \mathbf{p}) 内に無数に存在し、その分布を連続的な分布関数 $f(\mathbf{x}, \mathbf{p}, t)$ で近似できるとする。この f は速度分布関数と呼ばれる。

空間全体が力場 $\mathbf{F}(\mathbf{x}, t)$ に支配されている時、粒子は

$$\begin{cases} \frac{d\mathbf{x}}{dt} = \mathbf{p}/m \\ \frac{d\mathbf{p}}{dt} = \mathbf{F}(\mathbf{x}, t) \end{cases} \quad (1.1)$$

に従って時間発展していく。よって (\mathbf{x}, \mathbf{p}) にいた粒子は微小時間 dt 後に

$$(\tilde{\mathbf{x}}, \tilde{\mathbf{p}}) = \left(\mathbf{x} + \frac{1}{m} \mathbf{p} dt, \mathbf{p} + \mathbf{F} dt \right) \quad (1.2)$$

に移動する。 (\mathbf{x}, \mathbf{p}) から $(\tilde{\mathbf{x}}, \tilde{\mathbf{p}})$ への変化を座標の変換と捉えたと、ヤコビアンは 1 なので

$$d\tilde{\mathbf{x}} d\tilde{\mathbf{p}} = d\mathbf{x} d\mathbf{p} \quad (1.3)$$

が成り立つ。また、位相空間内のある点 (\mathbf{x}, \mathbf{p}) における近傍 $d\mathbf{x} d\mathbf{p}$ 内の粒子数は

$$f(\mathbf{x}, \mathbf{p}, t) d\mathbf{x} d\mathbf{p} \quad (1.4)$$

であるため、 $(\tilde{\mathbf{x}}, \tilde{\mathbf{p}})$ の近傍 $d\tilde{\mathbf{x}} d\tilde{\mathbf{p}}$ 内の粒子数は

$$\left[f + \left(\frac{\partial f}{\partial t} + \frac{1}{m} \mathbf{p} \cdot \frac{\partial f}{\partial \mathbf{x}} + \mathbf{F} \cdot \frac{\partial f}{\partial \mathbf{p}} \right) dt \right]_{(\mathbf{x}, \mathbf{p}, t)} d\mathbf{x} d\mathbf{p} \quad (1.5)$$

となる。衝突が無ければ $d\mathbf{x} d\mathbf{p}$ にいた粒子は dt 後には $d\tilde{\mathbf{x}} d\tilde{\mathbf{p}}$ にいるはずなので、

$$\frac{\partial f}{\partial t} + \frac{1}{m} \mathbf{p} \cdot \frac{\partial f}{\partial \mathbf{x}} + \mathbf{F} \cdot \frac{\partial f}{\partial \mathbf{p}} = 0 \quad (1.6)$$

が成り立つ。式 (1.6) は無衝突 Boltzmann 方程式または Vlasov 方程式と呼ばれる。

以後衝突のある場合を議論するが、Boltzmann によって Stosszahlansatz (衝突数仮定) とも訳された分子的混沌と呼ばれる仮定をおく。分子的混沌は衝突数の算出の際に登場するもので、衝突する粒子の速度に依存性が無いという仮定である。以下の議論の結果登場する Boltzmann 方程式は不可逆性を備えているが、古典力学は可逆な体系を持っている。そのギャップを埋めるのが分子的混沌となっている。

衝突がある場合、 dt の間に元々 $d\mathbf{x} d\mathbf{p}$ にいた粒子が流出する場合と、その外から $d\tilde{\mathbf{x}} d\tilde{\mathbf{p}}$ に流入してくる場合があり、その2パターンによって式 (1.6) からのズレが発生する。前者の粒子の数を $J_L(\mathbf{x}, \mathbf{p}, t) d\mathbf{x} d\mathbf{p} dt$ 、後者の粒子の数を $J_G(\tilde{\mathbf{x}}, \tilde{\mathbf{p}}, t + dt) d\tilde{\mathbf{x}} d\tilde{\mathbf{p}} dt$ で表すと、以下の式が成り立つ。

$$\frac{\partial f}{\partial t} + \frac{1}{m} \mathbf{p} \cdot \frac{\partial f}{\partial \mathbf{x}} + \mathbf{F} \cdot \frac{\partial f}{\partial \mathbf{p}} = J_G - J_L \quad (1.7)$$

J_G, J_L はそれぞれ衝突による発生項、消滅項と呼ばれている。Boltzmann の課した仮定として、粒子の密度は十分に低く衝突は2体散乱までを考えればよく、3体以上の散乱は無視できるとする。

衝突がなければ $d\mathbf{x} d\mathbf{p}$ の中にいた粒子は dt の間に $d\tilde{\mathbf{x}} d\tilde{\mathbf{p}}$ の中に移動する。 $J_L d\mathbf{x} d\mathbf{p} dt$ は、 $d\mathbf{x} d\mathbf{p}$ の中にいた粒子のうち dt の間に衝突により $d\tilde{\mathbf{x}} d\tilde{\mathbf{p}}$ に入ることでできなかった数である。 $d\mathbf{x} d\mathbf{p} \rightarrow 0$ の極限では浅い角度の衝突でも $d\tilde{\mathbf{x}} d\tilde{\mathbf{p}}$ から出ていくので、 $J_L d\mathbf{x} d\mathbf{p} dt$ は $d\mathbf{x} d\mathbf{p}$ 内の粒子が dt の間に衝突する回数と一致する。ある $d\mathbf{x} d\mathbf{p}$ 内の粒子一つに注目すると、この粒子の位置を単位時間あたりに通り過ぎる粒子の数密度は

$$\frac{1}{m} \int_{\mathbb{R}^3} |\mathbf{p} - \mathbf{p}_*| f(\mathbf{x}, \mathbf{p}_*, t) d\mathbf{p}_* \quad (1.8)$$

で表される。ただし、 m は粒子の質量で、積分変数 \mathbf{p}_* は他の粒子の運動量と対応している。

よって、分子的混沌から、粒子の微分散乱断面積を $d\sigma/d\Omega(p_0, \cos \theta)$ とすると、

$$J_L(\mathbf{x}, \mathbf{p}, t) = f(\mathbf{x}, \mathbf{p}, t) \times \int_{\mathbb{R}^3} \int_{S^2} \frac{d\sigma}{d\Omega} (|\mathbf{p} - \mathbf{p}_*|, \cos \theta) \times \frac{1}{m} |\mathbf{p} - \mathbf{p}_*| f(\mathbf{x}, \mathbf{p}_*, t) d\Omega d\mathbf{p}_* \quad (1.9)$$

である。ただし、 θ は散乱角であり、入射粒子が散乱によってずれた角度である。

一方、 $J_G d\tilde{\mathbf{x}} d\tilde{\mathbf{p}} dt$ は dt の間に $d\mathbf{x} d\mathbf{p}$ の外から衝突により $d\tilde{\mathbf{x}} d\tilde{\mathbf{p}}$ の中へ流入してくる粒子の数である。流入後の2粒子の運動量を $\tilde{\mathbf{p}}, \tilde{\mathbf{p}}_*$ と置くと、流入前の2粒子の運動量は

$$\mathbf{p}' = \frac{\tilde{\mathbf{p}} + \tilde{\mathbf{p}}_*}{2} + \left| \frac{\tilde{\mathbf{p}} - \tilde{\mathbf{p}}_*}{2} \right| \boldsymbol{\omega} \quad (1.10)$$

$$\mathbf{p}'_* = \frac{\tilde{\mathbf{p}} + \tilde{\mathbf{p}}_*}{2} - \left| \frac{\tilde{\mathbf{p}} - \tilde{\mathbf{p}}_*}{2} \right| \boldsymbol{\omega} \quad (1.11)$$

と表される。ただし、 $\boldsymbol{\omega}$ は単位ベクトルである。 $\mathbf{p}', \mathbf{p}'_*$ が与えられたとき、この衝突が起こる確率分布は微分散乱断面積

$$\frac{d\sigma}{d\Omega} \left(|\mathbf{p}' - \mathbf{p}'_*|, \frac{\boldsymbol{\omega} \cdot (\tilde{\mathbf{p}} - \tilde{\mathbf{p}}_*)}{|\tilde{\mathbf{p}} - \tilde{\mathbf{p}}_*|} \right) \quad (1.12)$$

で与えられる。よって消滅項の時と同様に、ある粒子の位置を単位時間あたりに通り過ぎる粒子の数密度を考えると

$$\begin{aligned}
& J_G(\tilde{\mathbf{x}}, \tilde{\mathbf{p}}, t + dt) \\
&= \frac{1}{m} \int_{\mathbb{R}^3} \int_{S^2} \frac{d\sigma}{d\Omega} \left(|\mathbf{p}' - \mathbf{p}'_*|, \frac{\boldsymbol{\omega} \cdot (\tilde{\mathbf{p}} - \tilde{\mathbf{p}}_*)}{|\tilde{\mathbf{p}} - \tilde{\mathbf{p}}_*|} \right) |\mathbf{p}' - \mathbf{p}'_*| f(\tilde{\mathbf{x}}, \mathbf{p}', t) f(\tilde{\mathbf{x}}, \mathbf{p}'_*, t) d\Omega d\tilde{\mathbf{p}}_*
\end{aligned} \tag{1.13}$$

となる。

$dt \rightarrow 0$ を考え $\tilde{\mathbf{x}}, \tilde{\mathbf{p}}, \tilde{\mathbf{p}}_*$ を $\mathbf{x}, \mathbf{p}, \mathbf{p}_*$ と同一視する。式 (1.9), (1.13) をまとめて式 (1.7) に代入すると

$$\frac{\partial f}{\partial t} + \frac{1}{m} \mathbf{p} \cdot \frac{\partial f}{\partial \mathbf{x}} + \mathbf{F} \cdot \frac{\partial f}{\partial \mathbf{p}} = C[f] \tag{1.14}$$

$$\begin{aligned}
C[f] = \frac{1}{m} \int_{\mathbb{R}^3} \int_{S^2} \frac{d\sigma}{d\Omega} \left(|\mathbf{p} - \mathbf{p}_*|, \frac{\boldsymbol{\omega} \cdot (\mathbf{p} - \mathbf{p}_*)}{|\mathbf{p} - \mathbf{p}_*|} \right) |\mathbf{p} - \mathbf{p}_*| \\
(f(\mathbf{x}, \mathbf{p}', t) f(\mathbf{x}, \mathbf{p}'_*, t) - f(\mathbf{x}, \mathbf{p}, t) f(\mathbf{x}, \mathbf{p}_*, t)) d\Omega d\mathbf{p}_*
\end{aligned} \tag{1.15}$$

を得る。この式は Boltzmann 方程式と呼ばれる。

実際には、位相空間内の座標を表すには (\mathbf{x}, \mathbf{p}) の代わりにより扱いやすい (\mathbf{x}, \mathbf{v}) が用いられることが多い。それに伴い、今まで登場した f や $d\sigma/d\Omega$ などの \mathbf{p} を引数に持つ関数も代わりに \mathbf{v} が引数になるなど変更を受けるが、混同の恐れがない限り同じ記号を用いる。それに伴い、力場 \mathbf{F} も本来は単位質量あたりの力場 \mathbf{F}/m と変更する必要があるが、便宜上力場 \mathbf{F} として使い続ける。

1.2 Boltzmann 方程式の応用

Boltzmann 方程式には様々な応用先がある。まず挙げられるのは、直接的な流体計算への応用であり、希薄気体の動きの解析およびシミュレーションに用いることができる。後述の宇宙プラズマも希薄気体の一例であり、他にも惑星の大気 (e.g. [10]) や熱勾配のある壁面に沿って希薄気体が這うように移動する熱ほふく流と呼ばれる現象 (e.g. [53]) など、希薄気体への Boltzmann 方程式の応用例は多い。それ以外にも、物性分野においては水と水蒸気の混合した流体など複数の相を持つ流体 (e.g. [50]) および半導体内のフォノンや電子 (e.g. [29, 59]) のシミュレーションが挙げられ、量子論の文脈では量子電磁気学における熱場の量子論と Boltzmann 方程式の等価性 (e.g. [8]) などが挙げられる。

たくさんの応用がある中、本研究で最も注目するのは宇宙分野への応用である。まず Vlasov 方程式から考えると、銀河サイズのダークマターハロー形成 [34, 58]、ダークマターの大規模構造の形成 [57] やニュートリノの大規模な分布形成 [62] などが挙げられる。他にも宇宙プラズマの研究に活用されており、高温プラズマにおけるイオンホール形成 [14] やプラズマの乱流ダイナモ不安定性による磁場の成長 [46] など様々な場所で Vlasov 方程式を目にすることができる。

その一方 Boltzmann 方程式は衝突項をそのまま計算しようとするると計算量が膨大になってしまうことから、Vlasov 方程式に比べるとシミュレーションの実施例は少ない。そんな中でも使用例は多数存在し、高エネルギーの電子が宇宙マイクロ波背景放射 (CMB) と

相互作用することで生じる Sunyaev-Zeldovich (SZ) シグナルの計算 [12] や中性子星連星合体におけるニュートリノのフレーバーの変化 [37]、太陽圏における太陽風と星間物質の相互作用 [33] に自己相互作用のあるダークマターの構造形成 [16] などさまざまな計算が知られている。

このように、Vlasov, Boltzmann 方程式は銀河スケールから銀河団、そして宇宙の大規模構造のスケールのシミュレーションにも用いられるなど幅広いレンジで活躍しており、宇宙を支配する方程式の一つといえる。

1.3 Boltzmann 方程式のシミュレーション手法

Boltzmann 方程式は $3 + 3 = 6$ 次元の位相空間内の方程式のため、直接の数値シミュレーションには無衝突のケースでシミュレーション内の時間あたり少なくとも $O(N_x^3 N_v^3)$ の計算コストが必要になる。ただし、 N_x, N_v は空間および速度空間を一方向あたりいくつかのメッシュに切り分けたかを表している。さらに、速度および角度依存性のある一般の微分断面積を持った粒子に対する有衝突のケースでは、式 (1.15) のように位相空間 1 セルあたり 5 次の積分を行う必要がある。そのため、衝突項を計算するたびに少なくとも見かけ上は $6 + 5 = 11$ 次元の計算を行うことになり、現実的な計算ではなくなってしまふ。(4.1.1 で詳細は述べるが、実際には 9 次元の計算となる。) 無衝突の 6 次元の場合ですら直接数値シミュレーションを行うには計算資源が足りず、様々な近似解法が登場した。

1.3.1 N 体シミュレーション

現在最も一般的なシミュレーション手法は N 体シミュレーションである。この手法では、速度分布関数 $f(\mathbf{x}, \mathbf{p}, t)$ を N 個の超粒子と呼ばれる速度の情報を持つ仮想上の粒子の分布で近似する。ただし、速度分布関数 f はもともと多数の粒子を均してできる分布であるが、このもとの粒子の数に比べ超粒子の数 N は圧倒的に少ない。

シミュレーションにおいては、力場や衝突などにより超粒子が保有している速度を変えながら、その速度に従って超粒子が移動していく。本来多数の粒子で起こるはずだった二体緩和を人工的に加えたり、位置の近い超粒子の計算では時間刻みを細かく取るなど細かい工夫により N 体シミュレーションは一定の精度を保っている。また、愚直に N 体シミュレーションを実装すると、重力などの遠距離力を扱う際にかかる力を計算する際に $O(N^2)$ の計算量がかかってしまう。しかし、ツリー法と呼ばれる離れた粒子をまとめて取り扱うことによる近似により、力の計算が $O(N \log N)$ で行われる。

衝突を考える際には、二つの超粒子 a, b の間に働く drag force と呼ばれる架空の力を考える [16]。Drag force は a, b の相対速度と平行な力で、摩擦のように a と b の相対速度が小さくなる方向に働く。大きさは微分散乱断面積 $d\sigma/d\Omega$ と数密度分布 ρ および a, b の相対速度 \mathbf{v}_0 から決まり、シミュレーションでは

$$\frac{d|\mathbf{v}_0|}{dt} = \rho |\mathbf{v}_0|^2 \sigma_{\bar{T}} \quad (1.16)$$

で計算される。ただし、

$$\sigma_{\bar{T}} = 4\pi \int_0^1 \frac{d\sigma}{d\Omega} (1 - \cos\theta) d\cos\theta \quad (1.17)$$

は modified momentum transfer cross-section と呼ばれ、より大きな変化をもたらす角度の大きな散乱の寄与を通常の散乱断面積と比べ大きく取ったものである [47]。

Drag force を反映させる過程を追うと、運動量は保存されているがエネルギーは保存されていない。衝突による運動方向の速度の減衰はそのままに、運動量を保ったまま失ったエネルギーを補給するには、a, b に対して \mathbf{v}_0 に垂直な速度を逆方向に与えることが考えられる。

そこで、drag force を考える前の a, b の速度を $\mathbf{v}_a, \mathbf{v}_b$ とすると ($\mathbf{v}_0 = \mathbf{v}_a - \mathbf{v}_b$)、drag force により失うエネルギー ΔE は

$$\begin{aligned} \Delta E &= \frac{m_{\text{sp}}}{2} \left((|\mathbf{v}_a|^2 + |\mathbf{v}_b|^2) - \left(\left| \mathbf{v}_a - \frac{d\mathbf{v}_0}{2} \right|^2 + \left| \mathbf{v}_b + \frac{d\mathbf{v}_0}{2} \right|^2 \right) \right) \\ &= \frac{m_{\text{sp}}}{2} d\mathbf{v}_0 \cdot \left(\mathbf{v}_0 - \frac{d\mathbf{v}_0}{2} \right) \end{aligned} \quad (1.18)$$

となる。そのため、a, b に追加する速度の大きさは

$$\sqrt{\frac{2\Delta E}{m}} \quad (1.19)$$

である。衝突には方位角依存性がないので、追加する速度の向きは向きはランダムに取る。

以上をまとめると、衝突を考える際には以下の過程を経ることになる。

1. drag force により、 $d|\mathbf{v}_0|$ を相対速度と平行に、エネルギーを失う方向に加える。
2. 式 (1.19) の大きさの速度を、相対速度と垂直になる方向からランダムに選び加える。

1.3.2 Direct Simulation Monte Carlo 法

Direct Simulation Monte Carlo (DSMC) 法は基本的には N 体シミュレーションと同じ方法で進んでいくが、衝突の計算の仕方が異なる。手法の名前に Monte Carlo と入っていることから分かるように確率的な過程を経て進めていくのだが、衝突の際にある区切られた空間 $d\mathbf{X}^*$ にいる dN^* 個の粒子からなるペア全てに対し以下の確率分布で衝突する方向を確率的に指定する。

$$P(\boldsymbol{\alpha}) = \frac{m_{\text{sp}}}{m} \frac{dt}{d\mathbf{X}^*} |\mathbf{v}_0| \frac{d\sigma}{d\Omega} \left(|\mathbf{v}_0|, \frac{\mathbf{v}_0 \cdot \boldsymbol{\alpha}}{|\mathbf{v}_0|} \right) \quad (1.20)$$

ただし、 m_{sp} は超粒子の質量であり、時間の刻み幅 dt は確率が 1 を超えないように設定するパラメーターである。

この確率 $P(\boldsymbol{\alpha})$ によって $\boldsymbol{\alpha}$ が選ばれた時、二つの超粒子の速度 $\mathbf{v}_a, \mathbf{v}_b$ をそれぞれ以下の $\tilde{\mathbf{v}}_a, \tilde{\mathbf{v}}_b$ に更新する。

$$\tilde{\mathbf{v}}_a = \mathbf{v}_a - (\mathbf{v}_0 \cdot \boldsymbol{\alpha}) \boldsymbol{\alpha} \quad (1.21)$$

$$\tilde{\mathbf{v}}_b = \mathbf{v}_b + (\mathbf{v}_0 \cdot \boldsymbol{\alpha}) \boldsymbol{\alpha} \quad (1.22)$$

$P(\alpha)$ は 1 に規格化されているわけではないのでどの α も選択されない可能性があり、その場合は速度のパラメーターに変化を加えない。これは衝突が起こらなかったことに対応している。

実際には、 dt は

$$dN^*P(\alpha) \ll 1 \quad (1.23)$$

を満たすように選ぶ必要がある [64]。なぜなら、 dt の間に衝突が 2 回発生してしまうということが頻繁に発生するようなシミュレーションは、3 体以上の散乱が無視できるとした Boltzmann 方程式の仮定を犯していることになるからである。実際に衝突が 2 回起こってしまった時には 2 回目以降の衝突を無視する場合や、そのまま衝突させる場合などなんらかの対応を行うが、衝突が 2 回発生してしまう頻度は非常に小さいはずなので、その対応がシミュレーション結果を大きく変えることはない。(変えてしまう場合には、 dt の取り方が大きすぎる。)

1.3.3 格子ボルツマン法

位相空間内において十分に細かいメッシュを考え、そのメッシュ上で方程式を離散化する差分法がベースとなった方法である。空間方向については細かいメッシュを維持するものの、速度方向については 0 に近い数値を持った少ない格子点のみを考える。速度がどの格子点上に取られるかについては D3Q15 (3 次元上に 15 種類の速度) や D2Q21 (2 次元上に 21 種類の速度) など様々なモデルが存在しており、扱いたい次元や細かさなどによってモデルが選択される。

基本的に衝突項は、局所平衡分布関数 $f_i^{(0)}(\mathbf{x}, t)$ を用いて

$$-\frac{1}{\tau} (f_i(\mathbf{x}, t) - f_i^{(0)}(\mathbf{x}, t)) \quad (1.24)$$

と簡略化される。ただし、分布関数の速度依存性は添字によって表され、緩和時間係数 τ は分布が衝突により局所平衡分布に近づく時間を表している。Boltzmann 方程式 (1.14), (1.24) を 1 次元風上差分法により離散化すると、

$$f_i(\mathbf{x} + \mathbf{c}_i \Delta t, t + \Delta t) = f_i(\mathbf{x}, t) - \frac{1}{\tau} \left(f_i(\mathbf{x}, t) - f_i^{(0)}(\mathbf{x}, t) \right) \quad (1.25)$$

を得る。この時、時間刻み幅を 1、空間刻み幅を $|\mathbf{c}_i|$ としている。この式によってシミュレーションを進めていくのが格子ボルツマン法で、式 (1.25) は格子 BGK 方程式と呼ばれている [65]。

局所平衡分布関数 $f_i^{(0)}(\mathbf{x}, t)$ を適切に設定して計算を行うことで、2 次精度の Navier-Stokes 方程式のシミュレーションを行うことができる [52]。

1.3.4 直接計算

位相空間を主に直交座標によって区切り、差分法などにより速度分布関数 $f(\mathbf{x}, \mathbf{v}, t)$ そのものの時間発展を追う方法である。Vlasov 方程式に対し、演算子分離と呼ばれるシミュ

レーションの枠組みを用いることができる [11]。演算子分離においては、Vlasov 方程式を \mathbf{x} 空間に対する移流と \mathbf{v} 空間に対する移流に分離することができる。

$$\begin{cases} \frac{\partial f}{\partial t} + \mathbf{p} \cdot \frac{\partial f}{\partial \mathbf{x}} = 0 \\ \frac{\partial f}{\partial t} + \mathbf{F} \cdot \frac{\partial f}{\partial \mathbf{p}} = 0 \end{cases} \quad (1.26)$$

また、衝突のある場合は

$$\frac{\partial f}{\partial t} = C[f] \quad (1.27)$$

も考慮する。これらを風上差分法などを用いて離散化させると、シミュレーションを進めることができる。

例えば、1+1 次元位相空間を考え、1 次精度風上差分法により幅 Δt , Δx , Δv で離散化すると式 (1.26) は

$$f_{k;j}^{n+1} = f_{k;j}^n - v_k \frac{\Delta t}{\Delta x} \begin{cases} f_{k;j}^n - f_{k;j-1}^n & (v_k > 0) \\ f_{k;j+1}^n - f_{k;j}^n & (v_k < 0) \end{cases} \quad (1.28)$$

$$f_{k;j}^{n+1} = f_{k;j}^n - F_j \frac{\Delta t}{\Delta v} \begin{cases} f_{k;j}^n - f_{k-1;j}^n & (F_j > 0) \\ f_{k+1;j}^n - f_{k;j}^n & (F_j < 0) \end{cases} \quad (1.29)$$

となる。ただし、表記を簡便化して $f_{k;j}^n = f(x_j, v_k, t_n)$, $t_n = \sum_{i=0}^{n-1} \Delta t$, $x_j = j\Delta x$, $v_k = k\Delta v$ としている。また、式 (1.27) はオイラー法を用いて離散化すると

$$f_{k;j}^{n+1} = f_{k;j}^n + \Delta t C[f^n] \quad (1.30)$$

となる。他にも様々な離散化の方法があり (付録 B 参照)、より高い精度の風上差分法や 3.1.3 で述べる Reservoir 法などでも離散化を行うことができる。

衝突項については本来、位相空間の 6 次元と積分 (1.15) の計算 5 次元を合わせて少なくとも見かけ上は 11 次元の計算を毎タイムステップごとに行わなければならない。しかし、4.1.1 において後述するスペクトル法と呼ばれる方法では、11 次元の前計算は必要なもののシミュレーション中の衝突項の計算を 9 次元に抑えることができる [41]。ただし、衝突のモデルによっては前計算の積分を解析的に求めることができる場合があり、例えば Variable Hard Sphere (VHS) モデルと呼ばれるモデルでは 7 次元の前計算で済む。

さらに 4.1.2 に詳細は述べるが、Fourier 空間でのメッシュの取り方を直交座標から 3 次元球座標に切り替えることで、速度空間の積分を角度と直線の積分に切り替えることができる。これにより、少ない計算回数で同程度の計算精度を出すことができる [18]。

第2章 量子計算

量子計算は量子力学の数ある応用の一つだが、その量子計算も複数の側面を持つ。例えば、捉えどころの無い量子力学をよりはっきりと理解するための道具という側面を持つ。1980年代初頭には量子効果によって光よりも早く情報を伝えることができるのではないかという相対性理論の因果律を破る可能性が議論された。もちろん否定的な結論に至ったが、その際には、同時期に証明された no-cloning theorem という量子計算理論の定理が中心的な役割を担った [38]。このように理論的にも重要な量子計算であるが、本研究においては量子計算の持つ別の側面に注目する。

量子計算は、複数の計算を同時に進行させる量子並行性という性質を持つが、これにより古典的な計算では再現できない小さな計算量を持つ可能性（量子優位性）が示唆されている。量子優位性については情報理論の $P \neq NP$ 問題が関わっており未だ証明されていないものの、現状大多数の研究者が真であると考えている。本研究では量子計算の量子優位性に注目し、前述の Boltzmann 方程式の時間発展を追うシミュレーションを効率的に行う量子アルゴリズムを開発した。

2.1 Quantum bits

古典計算において bit は情報量を表す単位であり、1 bit が持つ情報は 0 と 1 の二つの状態のうちどちらの状態なのかというものである。量子計算において bit に対応する概念が quantum bit (qubit) であり、これがどのようなものを数学的な概念としてまずは述べる。

2.1.1 1 qubit system

bit が 0 と 1 という二つの状態から構成されていたように、qubit も二つの基本的な状態 $|0\rangle$ と $|1\rangle$ を持つ。状態についている $|\rangle$ は \mathbb{C}^2 空間内のベクトルであることを意味しており、重ね合わせと呼ばれる状態の線型結合

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (2.1)$$

も qubit の持つ状態である。ただし、 α や β は振幅と呼ばれる複素数である。 $|0\rangle$ はノルム 1 の状態ベクトルであり、 $|1\rangle$ はそれに直交するノルム 1 の状態ベクトルである。これらを、計算上の都合により本論文では以下のベクトルに割り当てる。

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad (2.2)$$

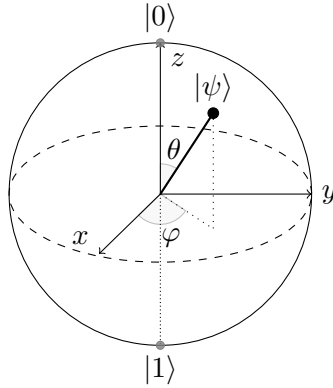


図 2.1: Bloch sphere

$$|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad (2.3)$$

また、量子力学的要請から qubit には測定という概念が登場する。ある qubit に測定を行うと、元々 2 つの状態の重ね合わせだった量子状態が、古典的な 0 か 1 のどちらかの状態となる。ただし、0 と 1 のどちらになるかは振幅に依存した確率で決まり、0 が得られる確率は

$$\langle\psi|\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}|\psi\rangle = |\alpha|^2 \quad (2.4)$$

であり、1 が得られる確率は

$$\langle\psi|\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}|\psi\rangle = |\beta|^2 \quad (2.5)$$

となる。以上より、振幅 α, β には

$$|\alpha|^2 + |\beta|^2 = 1 \quad (2.6)$$

という規格化条件が課される。代表的な 1 qubit 系には $|0\rangle$ や $|1\rangle$ の他に $|+\rangle$ や $|-\rangle$ で表される状態が存在し、それぞれ

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad (2.7)$$

$$|-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \quad (2.8)$$

である。これらは 0 と 1 が測定される確率がどちらも 50 % であり、量子並行性を考える上で非常に重要な状態となっている。

ところで、3 つの実数 θ, φ, γ を用いることで、式 (2.1) を以下のように変形することができる。

$$|\psi\rangle = e^{i\gamma} \left(\cos \frac{\theta}{2} |0\rangle + e^{i\varphi} \sin \frac{\theta}{2} |1\rangle \right) \quad (2.9)$$

この式のうち、 γ の寄与である $e^{i\gamma}$ は測定に一切影響を及ぼさないため、この節の中では考慮しない。残された2つの実数 θ, φ を用いると、3次元の単位球面上の位置を指定する形で幾何学的な情報に変換することができる (図 2.1)。この球は Bloch sphere と呼ばれ、1 qubit 系の有用な視覚化を与える。

2.1.2 Multiple qubits system

n qubits の系を考える。1 qubit における基本的な基底は $|0\rangle$ と $|1\rangle$ だったが、 n qubits にも同様の基底 $|0\rangle, |1\rangle, \dots, |2^n - 1\rangle$ が存在する。これらは $|0\rangle$ と $|1\rangle$ のテンソル積

$$|i\rangle = |i_0 i_1 \dots i_{n-1}\rangle = |i_0\rangle \otimes |i_1\rangle \otimes \dots \otimes |i_{n-1}\rangle = |i_0\rangle |i_1\rangle \dots |i_{n-1}\rangle \quad (2.10)$$

で表される。ただし、 i_0, i_1, \dots, i_{n-1} は i の二進数表記 $i_0 i_1 \dots i_{n-1}^{(2)}$ の格桁に対応している。この基底を組み合わせ、 n qubits の系は振幅 α_i により

$$|\psi\rangle = \sum_{i=0}^{n-1} \alpha_i |i\rangle \quad (2.11)$$

と表すことができる。振幅には 1 qubit の時と同様に

$$\sum_{i=0}^{n-1} |\alpha_i|^2 = 1 \quad (2.12)$$

という制限があるものの、 n qubits の系には 2^n 個の複素数の情報を入れることができる。 $n = 500$ の場合には、蓄えることのできる複素数の数が宇宙全体に存在すると考えられる原子の数よりも大きくなってしまったため、古典的に 500 qubits の系を完全にシミュレーションすることは不可能である。

2 qubits の場合には基底は $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ の4つで、「Bell 状態」あるいは「EPR ペア」と呼ばれる重要な状態

$$\frac{|00\rangle + |11\rangle}{\sqrt{2}} \quad (2.13)$$

がある。Bell 状態において、1つ目の qubit を測定すると 50% の確率で 0 または 1 が得られる。同様に、2つ目の qubit を測定しても 50% の確率で 0 または 1 が得られる。この性質から考えると、Bell 状態は $|++\rangle = |+\rangle|+\rangle$ と本質的には同じ状態だと推測することもできるが、実際には1回目の測定後の状態が異なるため異なる状態である。例えば、1つ目の qubit を測定した結果が 0 であったとする。その時、Bell 状態の場合は $|00\rangle$ になっているが、 $|++\rangle$ の場合は $|0\rangle|+\rangle$ となる。この後に2つ目の qubit を測定すると、Bell 状態の場合は 100% の確率で 0 が得られるが、 $|++\rangle$ の場合は 50% の確率で 0 または 1 が得られる。このように複数の qubits が紐づけられることをエンタングルといい、量子テレポーテーションなど古典では考えられない量子の不思議な挙動の元になっている。

2.2 量子ゲート

古典計算において論理ゲートと言え、入力と出力により決まる真理値表から定義される論理演算を行う電子回路のことで、NOT、AND、OR などがある (付録 A 参照)。これら基本的な論理ゲートを組み合わせることで任意の論理演算を行うことができ、古典計算における様々なアルゴリズムは元を辿ると基本的な数種類の論理ゲートに分解することができる。

古典計算と同様に、量子計算にも演算を行う仕組みがあり、量子ゲートと呼ばれている。量子ゲートも複数組み合わせることで様々な演算を行うことができるが、その演算には制約が存在する。 n qubits の系を \mathbb{C}^{2^n} のベクトルとして捉えた際には、量子力学から演算は線形に作用することが要請されるので、 $\mathbb{C}^{2^n \times 2^n}$ の行列で演算を表現する。量子ゲート U を任意の n qubits 状態 $|\psi\rangle$ に作用させた後にも規格化条件が成り立つので、

$$\langle\psi|U^\dagger U|\psi\rangle = 1 \quad (2.14)$$

から $U^\dagger U$ が単位行列になることが分かる。つまり、 U はユニタリ行列である。ユニタリ行列の集合は積演算に対して閉じているので、いくらゲートを組み合わせたとしても量子計算では本質的にユニタリな変換しか行うことができない。この制約から量子情報をコピーできないという no-cloning theorem や計算の可逆性が導かれるが、これらが量子アルゴリズムの開発を難化させている一因である。逆にユニタリな変換であれば、以下に紹介する量子ゲートを組み合わせることで、任意のユニタリな変換を行うことができる。

量子ゲートは数多くの種類が存在するが、本研究において登場するものに限っていくつか紹介し、関連する定理を導入する。

2.2.1 1 qubit gate

量子ゲートには様々な種類があるが、ゲートが作用する範囲が n qubits に渡っている場合、 n qubit gate と呼ばれる。そのうち、この小節では 1 qubit gate に着目する。

まず古典的な 1 bit の論理ゲートは、何もしないゲートか NOT gate の二つである。これら二つに対応する量子ゲートはそれぞれ identity gate (I gate) と X ゲートと呼ばれ、identity gate は古典から変わらず何もしないゲートで、 X ゲートは

$$X(\alpha|0\rangle + \beta|1\rangle) = (\alpha|1\rangle + \beta|0\rangle) \quad (2.15)$$

このように $|0\rangle$ と $|1\rangle$ を入れ替えるゲートである。この操作は、行列で表現すると

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad (2.16)$$

であり、図 2.1 の Bloch sphere 上だと x 軸まわりの 180 度回転に相当する。

1 qubit gate には古典的なアナロジーが存在しないゲートも存在する。その中でも重要なものは、 Z ゲートと H ゲート (Hadamard gate) である。それぞれ行列だと

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad (2.17)$$

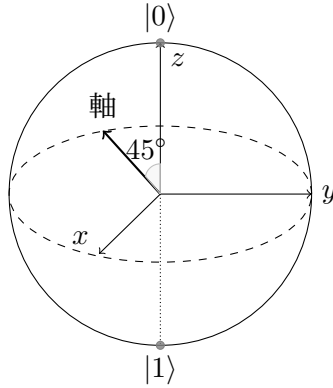


図 2.2: H ゲートで回転させる軸

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad (2.18)$$

と表され、Bloch sphere 上だと Z ゲートは z 軸まわりの 180 度回転に対応し、 H ゲートは x 軸と z 軸の間の軸 (図 2.2) まわりの 180 度回転に対応している。 H ゲートは $|0\rangle$ や $|1\rangle$ を $|+\rangle$ や $|-\rangle$ にするゲートで、かつ $H^2 = I$ となる。

また、Bloch sphere 内で x 軸まわりに θ 回転させるゲートは R_x ゲートと呼ばれており、

$$R_x(\theta) = \begin{pmatrix} \cos \frac{\theta}{2} & -i \sin \frac{\theta}{2} \\ -i \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{pmatrix} \quad (2.19)$$

で表される。同様に R_y, R_z ゲートも Bloch sphere 内で y, z 軸まわりに θ 回転させる量子ゲートであり、

$$R_y(\theta) = \begin{pmatrix} \cos \frac{\theta}{2} & -\sin \frac{\theta}{2} \\ \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{pmatrix}, \quad (2.20)$$

$$R_z(\theta) = \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix} \quad (2.21)$$

で表される。その他にも、Bloch Sphere 内で z 軸まわりに 90 度回転させる量子ゲート

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \quad (2.22)$$

や、その逆

$$S^\dagger = \begin{pmatrix} 1 & 0 \\ 0 & -i \end{pmatrix} \quad (2.23)$$

も有用な量子ゲートである。

入力		出力	
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

表 2.1: AND と OR を返す古典論理ゲートの真理値表。この論理ゲートは入力が 01 の時と 10 の時を出力から区別することができない。すなわち、不可逆な演算を行うため対応する量子ゲートを作ることができない。

入力		出力	
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

表 2.2: 対応する量子ゲートが存在する古典論理ゲートの真理値表。この論理ゲートは一つ目の出力では一つ目の入力をそのまま返し、二つ目の出力では二つの入力の XOR を返している。

2.2.2 Multiple qubits gate

1 qubit gate と同様に古典的なアナロジーから考えたいが、古典論理ゲートと量子ゲートは大きく異なる点がある。それは、古典論理ゲートは入力と出力の bit 数が異なることが許されるが、量子ゲートは許されない点である。そのため、入力 2 bits 出力 1 bit である AND ゲートのような論理ゲートに対応するような量子ゲートは存在しない。さらに、仮に論理ゲートの入出力の bit 数が同じであっても、量子ゲートに対応するものが存在するのについてはユニタリ行列として実現可能かどうかを検討しなければならない。例えば、入出力が 2 bits で AND と OR を返すような古典論理ゲート (表 2.1) は、対応する量子ゲートを作ることができない。

一方、対応する量子ゲートが存在する例を一つ挙げると表 2.2 が挙げられる。この論理ゲートに対応する量子ゲートは

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (2.24)$$

この行列で表され、もし 1 qubit 目が $|0\rangle$ なら何もせず、 $|1\rangle$ なら 2 qubit 目に X ゲートをかけることに相当している。1 qubit 目が 2 qubit 目に X ゲートをかけるかどうかを制御しているように見えることから 1 qubit 目は制御 qubit と呼ばれ、Controlled-NOT を略して CNOT (または CX ゲート) と呼ばれている。また、1 qubit 目が重ね合わさった状態に対しては、以下のように作用する。

$$\text{CNOT} (\alpha |0\rangle + \beta |1\rangle) |\psi\rangle = \alpha |0\rangle |\psi\rangle + \beta |1\rangle (X |\psi\rangle) \quad (2.25)$$

$\alpha = \beta = 1/\sqrt{2}, |\psi\rangle = |0\rangle$ の時できあがる状態は Bell 状態であることから、 H ゲートと合わせて CNOT は $|00\rangle$ から Bell 状態を作り出すことができる。制御 qubit を含む量子ゲートは他にも存在し、一般化して $C(U)$ ゲートと書かれる。 $C(U)$ ゲートは、1 qubit 目が 2 qubit 目に 1 qubit ゲート U がかかるかどうかを制御しているゲートである。

一方、multiple qubits gate には制御 qubit を持たないものも存在する。一例を挙げると、SWAP ゲートと呼ばれる 2 qubits ゲートがある。これは

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.26)$$

という行列で表され、2つの qubits を交換することに対応している。

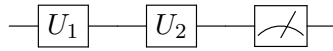


図 2.3: 量子回路図の例。ある qubit に U_1, U_2 をこの順に作用させた後測定する回路図。

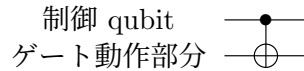


図 2.4: CNOT

ここで、量子ゲートを組み合わせてできる量子回路を表現する上で便利な手法を導入する。図 2.3 は回路図と呼ばれる図で、横線が qubit、箱が中に書いてあるゲートに対応している。(ただし、 X ゲートだけは \oplus を用いて表されることがある。) 回路図は左から右に読んでいく図で、図 2.3 はある qubit に U_1 がかった後に U_2 がかかり、最後に測定を行うという回路になっている。さらに、制御 qubit は黒丸で表される。例えば、CNOT は図 2.4 で表されるが、二本あるうちの上の横線に対応する qubit が制御 qubit となり下の横線に対応する qubit にかかるゲートを制御している。また、制御 qubit における $|0\rangle$ と $|1\rangle$ の役割を逆転させたものを白丸で表す。

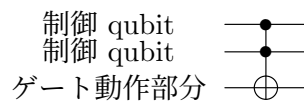
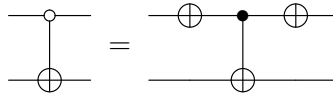


図 2.5: Toffoli ゲート

回路図を用いて、重要なゲートである Toffoli ゲートを定義する。図 2.5 が Toffoli ゲートを表す回路図で、 X ゲートに制御 qubit が二つついている。これは、上 2 本の横線に対応する qubits が両方 $|1\rangle$ のときのみ下の横線に対応する qubit に X ゲートがかかるというもので、制御 qubits が二つあることから CCX ゲートと呼ばれることもある。制御 qubits の数に上限はなく、 n 個の制御 qubits を持つ一般化された Toffoli ゲートは $C^n(X)$ ゲートとも書かれる。また、 $C(U)$ ゲートと同様に、複数の制御 qubits を持つゲートは $C^n(U)$ ゲートと書かれる。

2.2.3 ゲートの分解

量子コンピューターの実機では、使いたいゲートが直接実装されていないということが多々ある。このような時、使いたいゲートを実装されているゲートに分解することで、同等の働きをする量子回路を作り上げる。

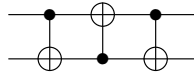


図 2.6: SWAP ゲートを表す量子回路

例えば、SWAP ゲートは図 2.6 に分解することができる。他にも、一般化された Toffoli ゲートは実機には実装されていないゲートであるが、1 qubit ゲートと CNOT に分解できることが知られている。例えば、 $C^n(X)$ ゲートは $O(n^2)$ 個の $C(R_x(\theta))$ ゲートに分解することができる [49]。さらに、一般の 1 qubit ゲート U に対して、 $C(U)$ ゲートは全体の位相を除いて以下のように 1 qubit ゲートと CNOT に分解できる。証明内に登場するパラメータである 4 つの実数に $\alpha = 0, \beta = -\pi/2, \gamma = \theta, \delta = \pi/2$ を指定することで $R_x(\theta)$ を作ることができ、 $\alpha = 0$ なので全体の位相も一致している。これらを組み合わせることで、 $O(n^2)$ 個の CNOT と 1 qubit ゲートで $C^n(X)$ を作り出すことができている。

Proof. U は

$$U = \begin{pmatrix} e^{i(\alpha-\beta/2-\delta/2)} \cos \frac{\gamma}{2} & -e^{i(\alpha-\beta/2+\delta/2)} \sin \frac{\gamma}{2} \\ e^{i(\alpha+\beta/2-\delta/2)} \sin \frac{\gamma}{2} & e^{i(\alpha+\beta/2+\delta/2)} \cos \frac{\gamma}{2} \end{pmatrix} \quad (2.27)$$

このように 4 つの実数 $\alpha, \beta, \gamma, \delta$ で表現でき、これを用いて

$$U = e^{i\alpha} AXBXC \quad (2.28)$$

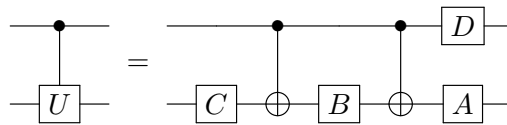


図 2.7: $C(U)$ ゲートの分解

と分解できる。ただし、

$$A = R_z(\beta)R_y(\gamma/2), B = R_y(-\gamma/2)R_z(-(\delta + \beta)/2), C = R_z((\delta - \beta)/2) \quad (2.29)$$

である。さらに D を

$$D = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{pmatrix} \quad (2.30)$$

と定義すると、図 2.7 のような $C(U)$ ゲートの分解を得ることができる。□

2.3 量子アルゴリズム

有名な量子アルゴリズムには、Shor のアルゴリズム [51] や Glover のアルゴリズム [22] などがある。ここでは、本研究で使用する量子アルゴリズムをいくつか列挙する。

2.3.1 インクリメントおよびデクリメント

インクリメントやデクリメントというのは、古典的には整数に 1 を足し引きすることである。量子の場合、 n qubits の系に対するインクリメントは以下の行列によって定義される。

$$U_{\text{inc}} = \begin{pmatrix} 0 & 0 & & 0 & 0 & 1 \\ 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & & 0 & 0 & 0 \\ \vdots & & \ddots & & \vdots & \\ 0 & 0 & & 1 & 0 & 0 \\ 0 & 0 & \cdots & 0 & 1 & 0 \end{pmatrix} \quad (2.31)$$

U_{inc} は $i \in \{0, 1, \dots, 2^n - 2\}$ に対し $|i\rangle$ を $|i + 1\rangle$ に移し、 $|2^n - 1\rangle$ を $|0\rangle$ に移す演算である。

一方デクリメントは

$$U_{\text{dec}} = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & & 0 & 0 \\ \vdots & & \ddots & & \vdots & \\ 0 & 0 & 0 & & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & 0 & & 0 & 0 \end{pmatrix} \quad (2.32)$$



図 2.8: 4 qubits の系に対するインクリメント (左) およびデクリメント (右)



図 2.9: 4 qubits の系に対する $|i\rangle \rightarrow |i+2^2\rangle$ (左) および $|i\rangle \rightarrow |i-2^2\rangle$ (右) を行う量子回路。

で定義され、 U_{dec} は U_{inc} の逆行列となっている。

U_{dec} および U_{inc} は 4 qubits の場合は図 2.8 によって再現される。他の qubits の系に対しても、同様の三角形に近い形の量子回路によって再現することができる。

これらの回路は少し回路を変更するだけで、2 冪の足し引き $|i\rangle \rightarrow |i \pm 2^p\rangle$ に変えることができる。これを実現するには、末尾から p 個の一般化 Toffoli ゲートを削除すれば良い。例えば、4 qubits の系で $p = 2$ のものは回路図にすると図 2.9 になる。

2.3.2 量子フーリエ変換

N 個の複素数 $c_x (x = 0, 1, \dots, N-1)$ に対し、離散フーリエ変換 $\tilde{c}_k (k = 0, 1, \dots, N-1)$ は以下のように定義される。

$$\tilde{c}_k = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} \exp \left[i \frac{2\pi kx}{N} \right] c_x \quad (2.33)$$

この離散フーリエ変換を用いると、量子フーリエ変換 (Quantum Fourier Transform, QFT) は、入力

$$|\psi\rangle = \sum_{x=0}^{N-1} c_x |x\rangle \quad (2.34)$$

に対して

$$|\tilde{\psi}\rangle = \sum_{k=0}^{N-1} \tilde{c}_k |k\rangle \quad (2.35)$$

で定義される。

$0 \leq j < N$ を整数として、入力 $|j\rangle$ の量子フーリエ変換は以下ようになる。

$$|\tilde{j}\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \exp \left[i \frac{2\pi kj}{N} \right] |k\rangle \quad (2.36)$$

これにより、量子フーリエ変換は測定で情報を得ることができる部分と、得ることができない部分を入れ替える操作と見ることもできる。以下に述べる量子フーリエ変換アルゴリズムを用いることで量子位相推定を行うことができ、簡単には得ることができない位相の情報に古典的にアクセスできるようになる。 $n = \log_2 N$ とし、 N が 2 冪の時に量子フーリエ変換を $O(n^2)$ の計算量で計算する量子アルゴリズムを導入する。

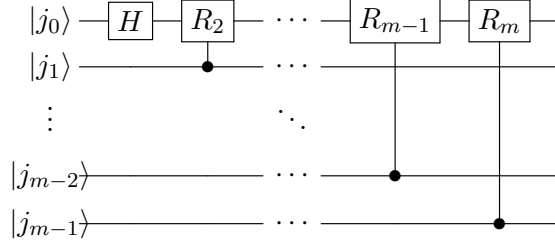


図 2.10: QFT のサブルーチン。 m qubits の系を考えている。

図 2.10 は QFT のサブルーチンで、 m qubits の系を考えている。ただし、 R_k は

$$R_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{pmatrix} \quad (2.37)$$

で定義される量子ゲートである。このサブルーチンを S_m とする。 $|j_0\rangle, \dots, |j_{m-1}\rangle$ が全て $|0\rangle$ もしくは $|1\rangle$ である場合を考え、サブルーチン S_m が $|j_0\rangle$ にどのような作用をもたらすかを追う。

まず H ゲートがかかった後は以下ようになる。

$$|j_0\rangle \rightarrow \frac{1}{\sqrt{2}} \left(|0\rangle + e^{2\pi i(0.j_0)} |1\rangle \right) \quad (2.38)$$

ただし、 $0.j_0$ は二進数の小数で、 $j_0 = 0$ なら 0 、 $j_0 = 1$ なら $1/2$ を表す。さらに、制御 R_2 ゲートがかかると

$$\rightarrow \frac{1}{\sqrt{2}} \left(|0\rangle + e^{2\pi i(0.j_0j_1)} |1\rangle \right) \quad (2.39)$$

このように変化する。さらに制御 R ゲートをかけていくと、最終的に以下の量子状態が得られる。

$$\frac{1}{\sqrt{2}} \left(|0\rangle + e^{2\pi i(0.j_0j_1\dots j_{m-1})} |1\rangle \right) \quad (2.40)$$

式 (2.36) を変形すると

$$\begin{aligned} |\tilde{j}\rangle &= \frac{1}{\sqrt{2^n}} \sum_{k_0=0}^1 \cdots \sum_{k_{n-1}=0}^1 \exp \left[\pi i j \left(\sum_{l=0}^{n-1} k_l 2^{-l} \right) \right] |k_0 \cdots k_{n-1}\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{k_0=0}^1 \cdots \sum_{k_{n-1}=0}^1 \bigotimes_{l=0}^{n-1} e^{\pi i j k_l 2^{-l}} |k_l\rangle \\ &= \frac{1}{\sqrt{2^n}} \bigotimes_{l=0}^{n-1} \left(\sum_{k_l=0}^1 e^{\pi i j k_l 2^{-l}} |k_l\rangle \right) \\ &= \frac{1}{\sqrt{2^n}} \bigotimes_{l=0}^{n-1} \left(|0\rangle + e^{\pi i j 2^{-l}} |1\rangle \right) \\ &= \frac{1}{\sqrt{2^n}} \bigotimes_{l=0}^{n-1} \left(|0\rangle + e^{2\pi i(0.j_{n-1-l}\dots j_{n-1})} |1\rangle \right) \end{aligned} \quad (2.41)$$

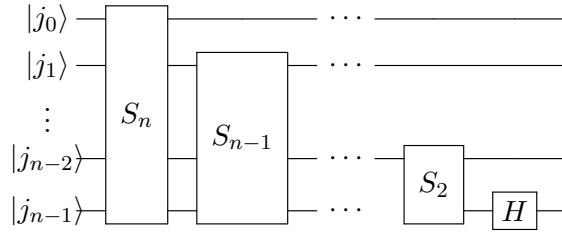


図 2.11: swap のない QFT の量子回路

となるため、 $m = n$ の場合を考えると式 (2.39) は $|\tilde{j}\rangle$ の最後の $n - 1$ 番目の qubit を再現できていることが分かる。また、サブルーチン S_m は 0 番目の qubit 以外に影響を及ぼさず、式 (2.41) は l に関して再起的な構造をしている。

このことを踏まえると、図 2.11 の量子回路により $|j\rangle$ から

$$\frac{1}{\sqrt{2^n}} \bigotimes_{l=0}^{n-1} \left(|0\rangle + e^{2\pi i(0 \cdot j_l \cdots j_{n-1})} |1\rangle \right) \quad (2.42)$$

が得られる。この量子状態は式 (2.41) の l 番目の qubit と $n - 1 - l$ 番目の qubit を交換したものになっている。よって、図 2.11 の量子回路と SWAP ゲートを組み合わせることで QFT を行う量子アルゴリズムが出来上がる。サブルーチン S_m の計算量が $O(n)$ であり、SWAP ゲートは $O(n)$ 回登場するので、全体として $O(n^2)$ の計算量となっている。

2.4 量子トモグラフィ

n qubits の系において、 $|0\rangle$ からある量子状態 $|\psi\rangle$ を与えるユニタリーな演算 U_ψ があるとす。 $|\psi\rangle$ の振幅を知りたい時、 U_ψ を用いて $|\psi\rangle$ に対する測定を繰り返しても、振幅の位相の情報は得ることができない。量子トモグラフィは、測定の前に工夫を施すことで位相の情報まで得ることができる測定の方法である。まずは 1 qubit の系での考察を行い、後から n qubits への一般化を行う。

2.4.1 1 量子状態への量子トモグラフィ

Pauli 行列を導入する。

$$\sigma_0 = I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \sigma_1 = X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \sigma_2 = Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \sigma_3 = Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad (2.43)$$

Pauli 行列は全てエルミート行列なので、量子力学よりそれぞれを物理量として捉えることができる。その解釈から、Pauli 行列の期待値を考えることができる。

1 量子状態 $|\psi\rangle$ を式 (2.9) により θ, φ を用いて表現する。

$$|\psi\rangle = e^{i\gamma} \left(\cos \frac{\theta}{2} |0\rangle + e^{i\varphi} \sin \frac{\theta}{2} |1\rangle \right) \quad (2.44)$$

σ_0 の期待値は規格化条件から必ず

$$\langle \psi | \sigma_0 | \psi \rangle = \cos^2 \frac{\theta}{2} + \sin^2 \frac{\theta}{2} = 1 \quad (2.45)$$

が得られるが、それ以外の Pauli 行列の期待値は

$$\langle \psi | \sigma_1 | \psi \rangle = \sin \theta \cos \varphi \quad (2.46)$$

$$\langle \psi | \sigma_2 | \psi \rangle = \sin \theta \sin \varphi \quad (2.47)$$

$$\langle \psi | \sigma_3 | \psi \rangle = \cos \theta \quad (2.48)$$

となる。 σ_3 の期待値に注目すると、これは Bloch sphere において z 軸と量子状態の成す角度 θ の \cos 値になっている。他の σ_1 および σ_2 についても、 x, y 軸と量子状態の成す角度の \cos 値になっており、3 つの Pauli 行列それぞれが各軸と深い関係があることが窺える。

さて、式 (2.46) ~ (2.48) がもし量子コンピューターから得られたとすると、 θ および φ が完全に決定されるため量子状態が一意に定まることになる。実際、

$$|0\rangle \text{---} \boxed{U_\psi} \text{---} \boxed{H} \text{---} \boxed{\text{Measurement}} \text{---} \quad (2.49)$$

により 0 が測定される確率は

$$\frac{1}{2} (1 + \sin \theta \cos \varphi) = \frac{1}{2} (1 + \langle \psi | \sigma_1 | \psi \rangle) \quad (2.50)$$

なので、量子回路 (2.49) を繰り返し行うことで式 (2.46) を求めることができる。他も同様に、

$$|0\rangle \text{---} \boxed{U_\psi} \text{---} \boxed{S^\dagger} \text{---} \boxed{H} \text{---} \boxed{\text{Measurement}} \text{---} \quad (2.51)$$

$$|0\rangle \text{---} \boxed{U_\psi} \text{---} \boxed{\text{Measurement}} \text{---} \quad (2.52)$$

という量子回路を考えると、(2.51) による 0 が測定される確率は

$$\frac{1}{2} (1 + \sin \theta \sin \varphi) = \frac{1}{2} (1 + \langle \psi | \sigma_2 | \psi \rangle) \quad (2.53)$$

で、(2.52) による 0 が測定される確率は

$$\frac{1}{2} (1 + \cos \theta) = \frac{1}{2} (1 + \langle \psi | \sigma_3 | \psi \rangle) \quad (2.54)$$

となる。

これらの量子回路は、以下の 3 つの点を押さえることから理解することができる。① H ゲートは Bloch sphere 内での x 軸と z 軸の中間の軸まわりの 180 度回転に対応し、 S, S^\dagger ゲートは z 軸まわりの 90, -90 度回転に対応している (2.2.1 参照)。② 量子状態の測定が Z の期待値を得ることに相当している。③ 関係式

$$HZH = \sigma_1, SHZHS^\dagger = \sigma_2, Z = \sigma_3 \quad (2.55)$$

が成り立つ。

以上をまとめると、 $|\psi\rangle$ を古典的な情報として得られていることがわかる。

2.4.2 n 量子状態への量子トモグラフィ

n 量子状態

$$|\psi\rangle = \sum_{j=0}^{2^n-1} c_j |j\rangle \quad (2.56)$$

を考える。この時、 4^n 個のエルミート行列

$$\sigma_{\mathbf{i}} = \sigma_{i_0, i_1, \dots, i_{n-1}} = \bigotimes_{j=0}^{n-1} \sigma_{i_j} \quad (2.57)$$

の実数係数線型結合により任意の $2^n \times 2^n$ エルミート行列を表現できるため、すべての $\sigma_{\mathbf{i}}$ の $|\psi\rangle$ による期待値がわかれば $|\psi\rangle$ を同定することができる。式 (2.55) から、式 (2.49), (2.51), (2.52) を適切な順番で並べれば $\sigma_{\mathbf{i}}$ の $|\psi\rangle$ による期待値がわかるため、最終的には $|\psi\rangle$ の情報が計算できる。

量子トモグラフィのやり方を上記に記したが、結局この方法は原理的には振幅を推定できるということを言っているだけで、効率のよい推定ができるというわけでは無い。そのため、振幅の情報が必要になった際にむやみに量子トモグラフィを利用するのはあまり良い選択肢ではなく、取り出す情報を少なくするアルゴリズムの設計における工夫が必要になってくる。

2.5 量子コンピューターの実機

量子コンピューターには量子アニーリングと呼ばれる種類のマシンも存在するが、本研究では万能型と呼ばれる量子万能ゲートを用いるものを取り扱う。(以下量子コンピューターと言う際には万能量子コンピューターを指すものとする。) 量子コンピューターの実機開発の研究は日進月歩であり、物理系として今まで様々なものが提案されてきた。今までの量子コンピューター開発の歴史はいかに量子状態を環境から守りながら qubit の数を増やすかの歴史であり、物理系によって特徴が異なりそれぞれの得意不得意がある。そのうちいくつかを例に挙げ、特徴を見ていく。

1 つ目は超伝導量子コンピューターである。このタイプの量子コンピューターでは、Cooper 対を Josephson 接合で隔離された超伝導体の箱に閉じ込めることで離散的なエネルギー準位を構成する。その固有状態をエネルギーの低い順に $|q\rangle = |0\rangle, |1\rangle, |2\rangle, \dots$ とした時、 $|0\rangle \leftrightarrow |1\rangle$ の遷移エネルギーと $|1\rangle \leftrightarrow |2\rangle$ の遷移エネルギーは大きく離れている。よって 2 準位系として振る舞うため、量子ビットとみなすことができる。これを応用して電荷ノイズへの耐性をつけた量子ビットは Transmon 量子ビット [26] と呼ばれ、現在の主流になっている。また、装置に対して電荷やパルスをかける操作が 1 qubit ゲートに相当し、量子ビットのキャパシターと呼ばれる部位間を直接繋ぐことで 2 qubits ゲートの実装も可能にしている。ノイズ耐性を上げる上で重要な量子誤り訂正の方法はいくつか提案されてきたが、その中でも表面符号と呼ばれる方法が有力な方法として挙げられる [21]。この方法は、近接した qubit にエンタングルメントを形成し、測定をすることで誤りを検知し修正することができる。

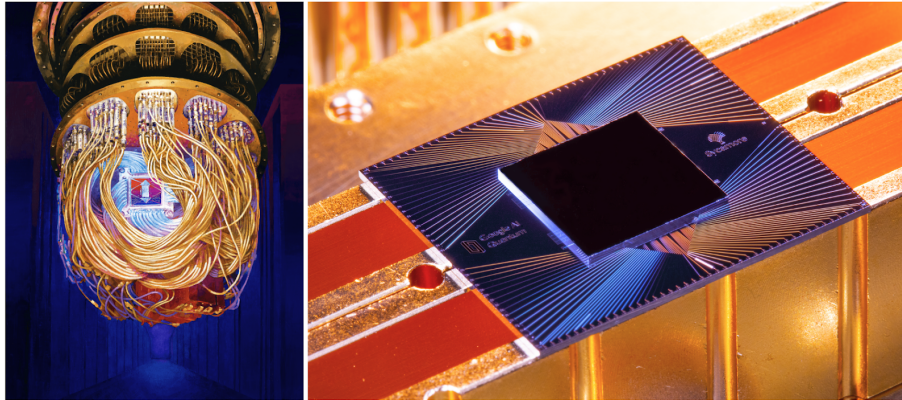


図 2.12: Google が量子超越性を示す [2] 際に使用した「Sycamore」という名前の量子コンピューター。左はプロセッサが冷凍機に入れられた様子を撮影したもの (Forest Stearns, Google AI Quantum Artist in Residence)。右はプロセッサの写真 (Erik Lucero, Research Scientist and Lead Production Quantum Hardware)。

さて、超伝導体を用いるメリットの一つとして挙げられるのが、量子ビットの集積化が可能な点である。限られた範囲に多数の量子ビットを構成することができるので量子ビットの数を増やしやすく、また狭い範囲に集まっているので 2 qubits ゲートの実装もしやすい。その一方、超伝導状態を作るために巨大な冷凍機が必要であり、ノイズ耐性が低いという欠点もある。超伝導量子コンピューターは上記のような欠点はあるつつも、メリットが大きいため現状一番有力視されている方法である。例えば、Google は 2019 年に 53 qubits の超伝導量子コンピューター (図 2.12) を用いて、古典的には最新のスーパーコンピューターを用いた時に 10000 年かかると推測される課題を約 200 s で解くことに成功した [2]。IBM から、一部の主張は過激であるなど批判は受けたものの、量子超越性を初めて実証したのは超伝導量子コンピューターだった。

続いて、2つ目は光量子コンピューターである。光量子コンピューターは長い間光子を量子ビットとして扱う KLM スキーム [25] と呼ばれる方法論が研究されてきた。しかし、この方法は欲しいときに必ず光子を作ることのできる光子源が必要であり、現状 20 個の光子を同時に作るのが限界である [60]。KLM スキームは小規模では成功を収めたものの、現状大きな規模で実現する目処は付いていない。

他にもいくつかスキームが存在する中、近年注目を集めているのは量子テレポーテーションを応用したループ型光量子コンピューターである。このスキームでは、時間多重化したスクイーズ光パルス量子ビットとして用い、このパルスの振幅を光学素子を用いて調整していくことで量子ゲートをかけていくことができる。これを用いて、1000 個以上の光パルスの大規模な量子エンタングルメントを合成する [54] など計算規模や自由度が大きく向上している。ループ型光量子コンピューターには、光のロスのエラーによる量子ビット数の制限や、一部の演算に必要な特殊な光パルスが未だ砂上の楼閣であることなど課題が残っている。その一方、極低温や真空などといった日常とかけ離れた環境を必要としないことや、将来的に量子インターネットが実現した際にスムーズに接続できることなどのメリットが挙げられる。2022 年には Xanadu が古典的には 9000 年かかるとされる課題

を $36 \mu\text{s}$ で解くことのできる量子超越性のある光量子コンピューターを開発した [32]。

3つ目はイオントラップによる光量子コンピューターである。この方法では、電氣的に運動を制御できるイオンを量子ビットとして扱う。また、イオンにレーザーなどを照射して量子状態を操作することで量子ゲートを作成する。レーザー冷却によって $\mu\text{K} \sim \text{mK}$ 温度に冷却することができ、イオン周辺の電極に電圧を印加することで真空中にイオンを一直線上に浮遊させることができる。これらの過程から高いコヒーレンスを保つことができエラー耐性は高く、集団運動の量子状態を経由することで任意のイオン間に量子もつれを生成することができる。その一方、20 量子ビット [17] を超えると制御が困難になるなど、スケールアップに課題が残っている他、量子状態の操作が遅いなどの問題も抱えている。

以上3つの量子コンピューターの実現方法の候補を紹介したが、他にも量子ドット [61] や核磁気共鳴 [48]、近年登場したトポロジカル素子 [30] といった手法もある。古典コンピューター開発は途中まで真空管を用いたものが一般的だったものの、途中からトランジスタを用いたコンピューターに取って代わられることになった。このように、現状は量子ビット数の観点で数百 qubits を誇るものも登場している超伝導型量子コンピューターが一大勢力となっているが、最終的にどの手法が覇権を握るのかは未だ不明である。

2.6 QRAM

古典コンピュータには Random-Access Memory (RAM) と呼ばれる記憶装置が存在する。RAM はメモリ内に割り当てられたアドレス i に対応するデータ x_i をセットで格納し、アドレス i を指定すると i の大きさによらず一定の非常に短い時間でデータ x_i を引き出すことができる装置である。

この装置の量子版が Quantum RAM (QRAM) で、高速な読み出しに特化した装置になっている。 n_r bit のバイナリデータ x_i ($i = 0, \dots, 2^{n_a} - 1$) が与えられたとき

$$|i\rangle_a |0\rangle_r \xrightarrow{\text{QRAM}} |i\rangle_a |x_i\rangle_r \quad (2.58)$$

という動作を行う。ただし、 $|i\rangle_a$ はアドレスを格納する n_a qubits のレジスタで、 $|i\rangle_r$ はバイナリデータが入る n_r qubits のレジスタである。重要な点は QRAM の $|i\rangle_a$ には重ね合わせ状態も入れることができるということである。具体的には、

$$\sum_{i=0}^{2^{n_a}-1} a_i |i\rangle_a |0\rangle_r \xrightarrow{\text{QRAM}} \sum_{i=0}^{2^{n_a}-1} a_i |i\rangle_a |x_i\rangle_r \quad (2.59)$$

このように重ね合わせ状態として与えられたアドレスそれぞれに対して x_i が格納される。

QRAM にはいくつかの仕組みが提案されており、Bucket-Brigade QRAM [20]、Fanout QRAM [19]、Flip-Flop QRAM [42]、Qudits-based memory [3]、Approximate PQC-based [44]、EQGAN QRAM [39] などが知られている。これらの性能は文献 [43] の TABLE IV で詳しく比較されている。

No-cloning theorem による制約が理論的に存在することに加え、提案手法のノイズ耐性の低さや構造の不安定さなどから、大規模な QRAM の実現は遅れている。しかしそんな中でも、Bucket-Brigade 法において古典データと併用した時の性能の向上の可能性を追

うことや [36]、ノイズ耐性を上げる方法を探索する [1, 40] など、最新の研究は進められている。量子コンピューターの実機と同様に、いずれはノイズ耐性のある大規模な QRAM が実現されることが期待されている。

QRAM の存在を仮定し、データをレジスタではなく振幅に入れる方法がある [45]。文献 [23] によって提案された古典的な木構造を応用した振幅へのエンコードを QRAM によって効率的に行う方法であり [66]、以下の状態を $O(\text{poly}(n_a))$ で作り出すことができる。

$$\frac{1}{M} \sum_{i=0}^{2^{n_a}-1} x_i |i\rangle_a, \quad M^2 = \sum_{i=0}^{2^{n_a}-1} x_i^2 \quad (2.60)$$

第3章 Vlasov 方程式の量子計算による加速

3.1 先行研究

Vlasov 方程式の時間発展を古典的に効率的にシミュレーションする手段は、Graphics Processing Unit (GPU) を用いたもの [13] やスーパーコンピューターを用いたもの [62, 63] などが存在しており、より大規模なシミュレーションを行うためにしのぎを削っている。これらは、Vlasov 方程式の直接シミュレーションを用いており、物理的要請により精度の落ちない範囲で N 体シミュレーションを一部用いるなど計算量が少なくなる工夫をしながらシミュレーションを進めている。しかし、直接シミュレーションを扱う以上 6 次元の計算から逃れることはできていない。

6 次元計算から解放される可能性として、量子コンピューターを用いることが挙げられる。微分方程式の時間発展を追うシミュレーションを行う量子アルゴリズムは既に模索されており、主に 2 つの方法に分けられる。

3.1.1 微分方程式の量子シミュレーション手法

1 つは変分法を用いる方法である [9, 31]。この方法は主に Noisy Intermediate Scale Quantum device (NISQ) での運用を目的に開発された古典量子ハイブリットアルゴリズムで、量子コンピュータの出すノイズへの耐性が高く、現在の量子コンピュータでも実機での運用を見込める。その一方、計算量が高く、未来に登場すると考えられている Fault Tolerant Quantum Computer (FTQC) を仮定した際に後述のもう一つの方法に比べると価値が薄くなってしまいう傾向にある。

もう一つは、何らかの形で時間発展させる量を量子コンピュータ内に持たせる方法である。こちらはノイズ耐性が低いことが多く、FTQC 向けのアルゴリズムになるが、その分計算量が低くなっている。本研究もこのやり方に則った方法になっている。

3.1.2 Vlasov 方程式の量子シミュレーション手法

Vlasov 方程式のシミュレーション手法は、これまで主に前小節 3.1.1 の後者の方法で進められてきた。基本的な方法は、与えられた力場を用いるか Vlasov 方程式を線形化した上で式変形をすることで量子コンピュータに載せられる形にする方法である。式変形によりハミルトニアンを作り出してハミルトニアンシミュレーションアルゴリズム [4, 15] と呼ばれるシミュレーション手法を行う方法 [56] や、大規模な連立一次方程式に変形することで Harrow-Hassidim-Lloyd (HHL) アルゴリズム [24] と呼ばれる連立一次方程式求解アルゴリズムに帰着させる方法 [35] などが知られている。

これらのシミュレーションは高速に動くことが期待されているが、特定の状況や線形効果を見ることに特化しているため、自己重力系など非線形な形をした力場の効果を直接見ることはできない。これを改善するために、Carleman 行列という C^∞ 級関数の微分を並べた行列を用いた非線形効果も取り込める量子シミュレーションも考案された [27]。しかし、線形な部分が非線形な部分よりも十分に大きい場合にのみ成立するシミュレーションとなっているため、非線形部分の大きな自己重力系の Vlasov 方程式には適していなかった。

他にも、Reservoir 法と呼ばれる特殊な離散化法を用いた Vlasov 方程式のシミュレーション手法がある [55]。この手法は決められた境界での反射を取り込むなど新たな取り組みが見られたものの、力場の全く無い $F = 0$ の状況をベースとした手法であった。この条件であれば、古典的には時間発展させることなく任意の時間の結果を得ることができるので、実用上は意味のない手法となってしまっている。

本研究はこの Reservoir 法を用いた手法を改善し、最終的に自己重力系における Vlasov 方程式のシミュレーションアルゴリズムを得た。そのため、以下 Reservoir 法およびそれを用いた量子アルゴリズム [55] について述べる。

3.1.3 Reservoir 法

3つの実変数 x, v, t を用いて、移流方程式

$$\frac{\partial f(x, v; t)}{\partial t} + v \frac{\partial f(x, v; t)}{\partial x} = 0 \quad (3.1)$$

を考える。この式の解は、 $t = 0$ の時の初期値 $f_0(x, v)$ を用いて

$$f(x, v; t) = f_0(x - vt, v) \quad (3.2)$$

と書ける。これは、移流方程式に用いられている変数のうち x が位置、 v が速度、 t が時刻に相当するとして、初期値 $f_0(x, v)$ が速度 v に従って形を変えずに動いていくことに対応している。

x を一定距離 $\Delta x > 0$ で離散化し、離散化された点を $x_j = j\Delta x$ と表す。また、 v には限界値 $V > 0$ を設け、離散化点の数 N_v により

$$v_k = \frac{(2k+1)V}{N_v} - V, \quad k = 0, 1, \dots, N_v - 1 \quad (3.3)$$

と離散化する。ここで、 V は $\forall |v| > V, f(x, v; t) \simeq 0$ が成り立つような V である。さらに、 t を離散化した点を t_n として $\Delta t_n = t_{n+1} - t_n$ と置く。

これらを用いて式 (3.1) を以下のように離散化することができる。

$$f_{k;j}^{n+1} = f_{k;j}^n - v_k \frac{\Delta t_n}{\Delta x} \begin{cases} f_{k;j}^n - f_{k;j-1}^n & (v_k > 0) \\ f_{k;j+1}^n - f_{k;j}^n & (v_k < 0) \end{cases} \quad (3.4)$$

この離散化の方法は一次精度風上差分法と呼ばれている。この式の $v_k \frac{\Delta t_n}{\Delta x}$ の部分に着目し、新たに Courant-Friedrichs-Lewy (CFL) カウンターと呼ばれる変数 C_k^n を考える。 C_k^n は以下の式に従って更新されていく。

$$C_k^{n+1} = C_k^n + |v_k| \frac{\Delta t_n}{\Delta x} \quad (3.5)$$

さらに、実数の小数部分を表す関数 $\text{dec}()$ を用いて

$$\Delta t_n = \min_k \left([1 - \text{dec}(C_k^n)] \frac{\Delta x}{|v_k|} \right) \quad (3.6)$$

を逐次的に定義していく。すると、どの n においても

$$\exists k \text{ s.t. } \text{dec}(C_k^n) = 0 \quad (3.7)$$

が成り立つ。

一次精度風上差分法 (3.4) では毎ステップ全ての k, j に対して $f_{k;j}$ の更新を行うが、 $\text{dec}(C_k^n) \neq 0$ を満たす k については $f_{k;j}$ の更新を行わないのが Reservoir 法である。その代わりに、 $\text{dec}(C_k^n) = 0$ を満たす k については、前回 k についての更新があった時から溜まっていた $v_k \Delta t_n / \Delta x$ を一度にまとめて更新する。 C_k^n の小数部分が 0 の時にのみ更新するため、溜まった $v_k \Delta t_n / \Delta x$ は合計で ± 1 になっている。そのため、以上をまとめると

$$f_{k;j}^{n+1} = \begin{cases} f_{k;j-1}^n & (v_k > 0 \text{ かつ } \text{dec}(C_k^n) = 0) \\ f_{k;j}^n & (\text{dec}(|C_k^n|) \neq 0) \\ f_{k;j+1}^n & (v_k < 0 \text{ かつ } \text{dec}(C_k^n) = 0) \end{cases} \quad (3.8)$$

という更新を行うことになる。

このように Δt_n を調整しつつ $f_{k;j}$ の更新を管理することで、 $f_{k;j}$ 同士の数値を足し引きすることなく $f_{k;j}$ の数値を $f_{k;j\pm 1}$ に移動させるだけで移流を表現している。そのため、風上差分法の課題の一つである数値散逸の発生を未然に防ぐことができている。

3.1.4 Reservoir 法を用いた量子アルゴリズム

x 方向には周期的境界条件を課し、 $N_x = 2^{n_x}$ 個に区切る。 $N_v = 2^{n_v}$ も 2 冪になるように取る。この時、量子コンピューター内に

$$|\psi\rangle = |q_x\rangle |q_v\rangle |q_a\rangle \quad (3.9)$$

を準備する。ただし、 $|q_x\rangle$ は n_x qubits の量子レジスタで、 $|q_v\rangle$ は n_v qubits の量子レジスタである。 $|q_a\rangle$ は ancilla qubits と呼ばれる量子レジスタで、QRAM による振幅エンコーディングの方法や一般化 Toffoli ゲートの実現方法など細かい部分で必要な大きさが変わってくる領域となっている。 $|q_x\rangle |q_v\rangle$ には常に $f_{k;j}$ の情報が位相や振幅の大きさなど何らかの方法で格納されているが、ancilla qubits は Toffoli ゲートの実行中など細かい動作中を除いて基本的に $|0\rangle$ の状態が保たれる。そのため、これ以降 $|q_a\rangle$ は省略する。

$|q_x\rangle$ や $|q_v\rangle$ には以下のように情報を入れる。例えば、離散化された初期値 $f_{0;k;j}$ が与えられていて位相に情報を入れるとする。その時、以下の量子状態が初期状態となる。

$$|\psi_0\rangle = \frac{1}{\sqrt{N_x N_v}} \sum_{j=0}^{N_x-1} \sum_{k=0}^{N_v-1} \exp \left[2\pi i \frac{f_{0;k;j}}{M} \right] |j\rangle |k\rangle, \quad M > \max_{j,k} f_{0;k;j} \quad (3.10)$$

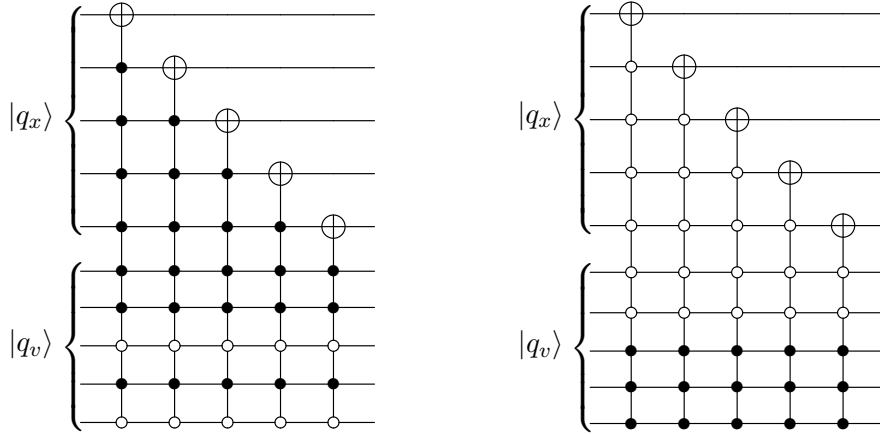


図 3.1: $N_x = N_v = 5$ の時 $k = 26$ (左) および $k = 7$ (右) における移流を表現する量子回路。

この状態に対して、例えば $N_x = N_v = 5$ での $k = 26, 7$ における $f_{k;j}$ の更新が行われる時を考える。 $k = 26$ の移流は $v_{26} > 0$ なので

$$f_{26;j} \leftarrow f_{26;j-1} \quad (3.11)$$

である。対応する量子回路は図 3.1 (左) である。 $|q_x\rangle$ の部分がインクリメント (図 2.8 左) に対応し、 $|q_v\rangle$ の部分が $|26\rangle = |11010\rangle$ の場合のみ作動する制御 qubits となっているため、合わせて式 (3.11) を表現できている。

同様に $k = 7$ の移流は $v_7 < 0$ なので

$$f_{7;j} \leftarrow f_{7;j+1} \quad (3.12)$$

である。図 3.1 (右) が対応し、 $|q_x\rangle$ の部分がデクリメント (図 2.8 右) で $|q_v\rangle$ が $|7\rangle = |00111\rangle$ でのみ起動する制御 qubits となっているので式 (3.12) を表現できる。

このように、ある $k = k_0$ での移流を表現する量子回路は、 $|q_v\rangle$ において $|k_0\rangle$ のみに反応する制御 qubits を持ち、 $|q_x\rangle$ においては $v_{k_0} > 0$ の時インクリメント、 $v_{k_0} < 0$ の時デクリメントを作用させるような回路になっている。これらの量子回路を QA_k と置く。

初期状態 $|\psi_0\rangle$ に対し、 QA_k を適当な順番で並べると Reservoir 法を実行した後の $f_{k;j}$ と同じものが量子レジスタ $|\psi\rangle$ に格納されている。具体的には、 Δt_n は $f_{k;j}$ の値とは無関係に決まる数値なので古典的に事前計算することができる。そこから、各 v_k に対してどの t_n で CFL カウンター C_k の小数部分が 0 になるのかを計算できるため、各 t_n でどの v_k に対する移流を行うことになるかが $f_{k;j}$ によらずに事前に判明する。よって、 t_n が小さい方から順番に、移流する v_k に合わせて QA_k を量子レジスタ $|\psi\rangle$ にかけていけば Reservoir 法を量子コンピューター上で再現することができる。

以上の議論から分かるように、 Δx 一定の条件下では Δt_n は v_k の取り方、すなわち V と N_v によって決まる量となっている。 v_k の取り方は式 (3.3) により決定されているため、 t_n は以下の集合を昇順で並べたものになる。

$$\{t_n | n = 0, 1, 2, \dots\} = \left\{ l \frac{N_v \Delta x}{V(2k+1)} \mid k = 0, 1, \dots, N_v - 1, l = 0, 1, 2, \dots \right\} \quad (3.13)$$

N_v	N_{cycle}	Δv	T_{cycle}	Ave. Δt
16	49	1.000	2.0	0.04167
32	213	0.500	4.0	0.01887
64	825	0.250	8.0	0.00971
128	3327	0.125	16.0	0.00481

表 3.1: $\Delta x = 1$ 下で $V = 8$ とした時の N_{cycle} : 1 周期に含まれる Δt_n の数、 Δv : 離散化速度の間隔、 T_{cycle} : 1 周期の長さ、Ave. Δt : Δt_n の平均値を各 N_v について計算した値。なお、 N_{cycle} は Δx や V によらず、 N_v のみに依存する量である。

このように、 Δt_n は実際に CFL カウンター C_k を変数に持たなくとも計算できるため、実装上は頭に C_k を保持する必要はない。

また、 Δt_n は一定にはならず複雑な変化をするが、周期

$$T_{\text{cycle}} = \frac{N_v \Delta x}{V} \quad (3.14)$$

で同じ数値を繰り返す。周期を含む Reservoir 法に関するパラメーターをまとめたのが表 3.1 である。 $N_v = 16$ のケースでは 1 周期あたりのタイムステップの数 N_{cycle} は 49 で、そこから N_v が 2 倍になるたびに N_{cycle} は約 4 倍になっている。このように、 N_{cycle} は $O(N_v^2)$ の値を持っている。その一方、 T_{cycle} は式 (3.14) の形をしているため N_v と線形関係にある。その結果、 Δt_n の平均値 Ave. $\Delta t = T_{\text{cycle}}/N_{\text{cycle}}$ は N_v と $O(1/N_v)$ の関係にある。

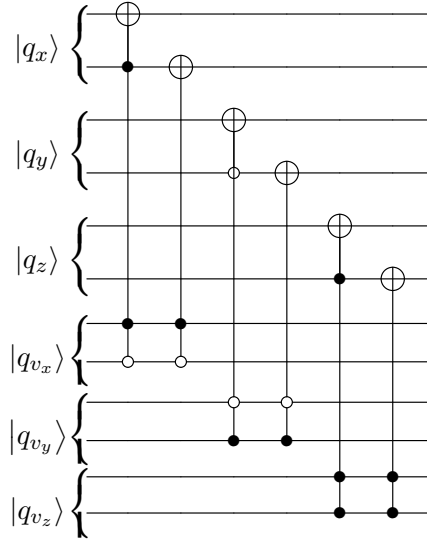


図 3.2: $N_x = N_v = 2$ の時 $k_x = 2, k_y = 1, k_z = 3$ における移流を連続で行う量子回路。

量子レジスタを 3 倍に増やしてそれぞれを x, y, z 座標に割り当てれば x, v が 3 次元の

ベクトル \mathbf{x}, \mathbf{v} だった場合でも微分方程式

$$\frac{\partial f(\mathbf{x}, \mathbf{v}; t)}{\partial t} + \mathbf{v} \cdot \frac{\partial f(\mathbf{x}, \mathbf{v}; t)}{\partial \mathbf{x}} = 0 \quad (3.15)$$

の数値シミュレーションを行うことができる。この場合でも \mathbf{v} の離散化を全ての方向に対して同じ大きさ Δv で行えば、 t_n は1次元のケースと変わらない。また、 QA_k は3次元になった影響で整数値 k が整数のベクトル \mathbf{k} になる。この時図 3.2 のような量子回路を用意すれば、1次元のケースと同様の t_n (式 3.13) を用いることで3次元における Reservoir 法を量子コンピューター上で実行できる。

式 (3.15) は Vlasov 方程式 (1.6) において力場が $\mathbf{F} = 0$ のケースに対応している。 $t = 0 \sim t_{\text{sim}}$ の間のシミュレーションを終えた後にデータ点全ての座標に対して以下の変換

$$(\mathbf{x}, \mathbf{v}) \leftarrow \left(\mathbf{x} + \frac{1}{2} t_{\text{sim}}^2 \mathbf{F}, \mathbf{v} + t_{\text{sim}} \mathbf{F} \right) \quad (3.16)$$

を施すことで、0 でない定数 \mathbf{F} を力場として持った Vlasov 方程式のシミュレーション結果が得られる。

3.2 Methods

本研究では前節で述べた Reservoir 法を用いた量子アルゴリズムを改善し、力場が時間や空間に依存するケースでもシミュレーションを行うことができるようにした。また、そのアルゴリズムをさらに改善し、自己重力系のような力場が速度分布関数 $f(\mathbf{x}, \mathbf{v}; t)$ に依存する場合においても計算できるようにした。

これ以降のケースでは、式 (3.16) のような座標変換のみでは力場の計算を済ませることはできない。そのため、無視できない大きさを持った $f(\mathbf{x}, \mathbf{v}; t)$ の要素が $\mathbb{R}^3 \times [-V, V]^3$ を出ないように、 V を十分大きく取る必要がある。以下、十分大きな V を取っていることを仮定する。

また、この節でも先に1次元でシミュレーションアルゴリズムを考える。自己重力系のシミュレーションを行う量子アルゴリズムまで考えた後、最後に3次元への拡張を検討する。

3.2.1 空間依存する力場 $F(x)$ におけるシミュレーション

力場が空間依存するケース $F(x)$ においては、Vlasov 方程式を一次精度風上差分法により離散化した結果、式 (3.4) に加え

$$f_{k;j}^{n+1} = f_{k;j}^n - F_j \frac{\Delta t_n}{\Delta v} \begin{cases} f_{k;j}^n - f_{k-1;j}^n & (F_j > 0) \\ f_{k+1;j}^n - f_{k;j}^n & (F_j < 0) \end{cases} \quad (3.17)$$

も必要になる。ただし、 F_j は $F(x)$ を空間方向に等間隔で離散化したもので、

$$\Delta v = \frac{2V}{N_v} \quad (3.18)$$

である。

式 (3.17) の使用に伴い、新たな CFL カウンター D_j^n が必要になってくる。 D_j^n は C_k^n と同様に Δt_n を決定するためのパラメータであり、

$$D_j^{n+1} = D_j^n + F_j \frac{\Delta t_n}{\Delta v} \quad (3.19)$$

によって更新されていく。これに伴い、 Δt_n は式 (3.6) から以下の式に変更される。

$$\Delta t_n = \min_{j,k} \left([1 - \text{dec}(C_k^n)] \frac{\Delta x}{|v_k|}, [1 - \text{dec}(|D_j^n|)] \frac{\Delta v}{|F_j|} \right) \quad (3.20)$$

すると、どの n においても

$$(\exists k \text{ s.t. } \text{dec}(C_k^n) = 0) \quad \text{または} \quad (\exists j \text{ s.t. } \text{dec}(|D_j^n|) = 0) \quad (3.21)$$

が成り立つようになる。これにより、各 n で行う $f_{k;j}$ への更新は

$$f_{k;j}^{n+1} = \begin{cases} f_{k;j-1}^n & (v_k > 0 \text{ かつ } \text{dec}(C_k^n) = 0) \\ f_{k;j+1}^n & (v_k < 0 \text{ かつ } \text{dec}(C_k^n) = 0) \\ f_{k-1;j}^n & (F_j > 0 \text{ かつ } \text{dec}(|D_j^n|) = 0) \\ f_{k+1;j}^n & (F_j < 0 \text{ かつ } \text{dec}(|D_j^n|) = 0) \\ f_{k;j}^n & (\text{dec}(C_k^n) \neq 0 \text{ かつ } \text{dec}(|D_j^n|) \neq 0) \end{cases} \quad (3.22)$$

となる。ただし、 F_j の値は実数値で与えられ、同一の n において $\text{dec}(C_k^n) = \text{dec}(|D_j^n|) = 0$ とならないことを仮定している。実際、 F_j がある开区間を台とする確率分布関数からランダムに与えられるとした時、ある n において $\text{dec}(C_k^n) = \text{dec}(|D_j^n|) = 0$ が成り立つというのは確率 0 の事象であるため、仮定は妥当である。

以上の工程で空間依存する力場 $F(x)$ 下におけるシミュレーションを行うことができる。しかし、このままだと Δt_n の計算に力場 $F(x)$ が絡んでより複雑になってしまう。そのため、先行研究の方法 (式 3.13) では頭に持たずに済んでいた CFL カウンター C_k^n を直接持つ必要が出てきてしまう。また、 C_k^n は小数部分が 0 になるタイミングがある程度揃っていたため t_n の数が 1 周期あたり $O(N_v^2)$ に抑えられていたが、 D_j^n は力場がばらつきのある実数値を取るため t_n の数が 1 周期あたり $O(N_v(N_v + N_x^3))$ と増えてしまい、全体として計算量を大きく増やすことになってしまう。

そこで、 Δt_n には先行研究の式 (3.13) を用いつつ、CFL カウンター D_j^n まわりの更新方法を変更することで、 C_k^n を直接保持する必要がなくなり、 t_n の数も増えることがないので計算量を抑えることができる。具体的には、

$$ND_j^n = \begin{cases} \left\lfloor D_j^n + F_j \frac{\Delta t_n}{\Delta v} \right\rfloor & (F_j > 0) \\ - \left\lceil D_j^n + F_j \frac{\Delta t_n}{\Delta v} \right\rceil & (F_j \leq 0) \end{cases} \quad (3.23)$$

を計算した後に CFL カウンター D_j^n を以下のように更新し、

$$D_j^{n+1} = D_j^n - ND_j^n \quad (3.24)$$

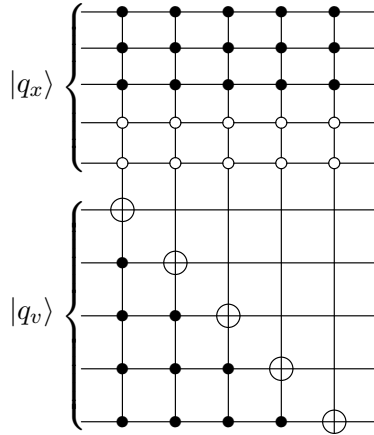


図 3.3: $j = 28$ における $ND_j = 1$ に対応する量子回路。

その後

$$f_{k;j}^{n+1} = f_{(k+ND_j^n);j}^n \quad (3.25)$$

と $f_{k;j}$ を速度方向に移流させ、最後に空間方向への移流 (式 3.8) を行う。以上の操作を各タイムステップで繰り返す。ただし、 $[\cdot]$ は整数部分を取り出す床関数である。以上の ND_j^n を用いる方法は式 (3.22) による更新の方法と比べ、 C_k^n を直接持つことなく全く同じ結果を得ることに成功している。

この方法を量子コンピューターに乗せるには、新たな量子回路を考える必要がある。速度方向の移流では、 $ND_j > 0$ の場合には図 3.3 のような量子回路を各 j に対して ND_j 回使い、 $ND_j < 0$ の場合には図 3.3 のインクリメント部分をデクリメントに変更した量子回路を各 j に対して $-ND_j$ 回用いる。空間方向の移流では図 3.1 に代表される量子回路を用いる。古典的に計算した順番で以上の量子回路を量子コンピューター上に並べると、空間方向のみに依存した力場 $F(x)$ における Vlasov 方程式の時間発展を量子レジスタ内で計算することができる。

3.2.2 空間および時間依存する力場 $F(x, t)$ におけるシミュレーション

何らかの形で古典的に各時間における力場が得られるとして考える。(e.g. 時間および空間の簡単に計算できる関数として力場が与えられる。) このケースでは、前小節 3.2.1 において導入した方法のうち F_j を時間依存するように置き換えるだけで無矛盾にシミュレーションを進めることができる。

3.2.3 自己重力系におけるシミュレーション

前小節までに考えていた力場 $F(x, t)$ を考えているうちは、Vlasov 方程式は線形であった。また、量子コンピューター内部にある速度分布関数 $f(x, v, t)$ の情報と古典的に移流

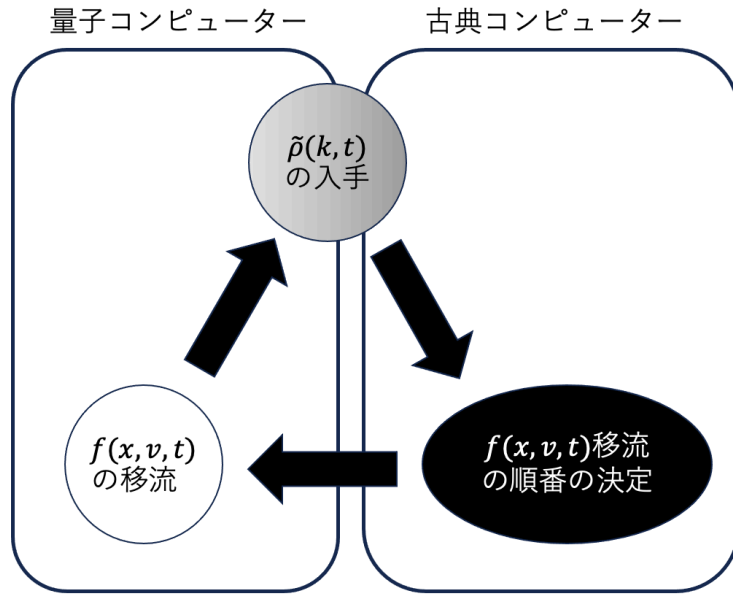


図 3.4: 自己重力系をシミュレーションするアルゴリズムの概要。大きく3つの動作があり、それらを繰り返すことでシミュレーションを進める。 $\tilde{\rho}(k, t)$ の入手に関しては、 $f(x, v, t)$ から $\tilde{\rho}(k, t)$ を作り出し、それを取り出す操作をするところまでを量子コンピューターが担当し、情報をまとめて $\tilde{\rho}(k, t)$ を格納するところを古典コンピューターが担当している。

を制御する部分は完全に切り離すことができていたため、古典的に前計算を行うだけで Vlasov 方程式のシミュレーションを行う量子アルゴリズムを構成することができた。

しかし、自己重力系では力場 $F(x, t)$ は速度分布関数 $f(x, v, t)$ に依存する汎函数項となっており、Vlasov 方程式 (1.6) は f に関する非線形微分方程式となる。qubit が線形な変換であるユニタリ変換 (量子ゲート) を通してのみ演算を行うため、非線形な項を含む微分方程式を量子コンピューターで扱うことは一般に非常に難しい。そこで、今回は非線形な計算だけは古典計算に押し付けそれ以外は量子計算で行うという、古典量子ハイブリッドアルゴリズムの形をとった。

付録 C で詳細は述べるが、周期的境界条件を x 空間に課す時

$$\rho(x, t) = \int f(x, v, t) dv \quad (3.26)$$

により定義される数密度のフーリエ係数 $\tilde{\rho}(k, t)$ が何らかの手段で得られれば、古典的に重力を計算することができる。本研究では、古典的な重力計算と量子計算による速度分布関数 $f(x, v, t)$ の移流を組み合わせることでシミュレーションを実現することに成功した。

図 3.4 は本アルゴリズムの概要を示している。大きく分けて3つの動作が存在し、それらを繰り返し用いていくことでアルゴリズムは進んでいく。1つ目の動作は量子コンピューター内で行われる $f(x, v, t)$ の移流で、2つ目の動作は古典コンピューター上で行われる速度分布関数 $f(x, v, t)$ の移流の順番の決定である。これらは、自己重力の情報が古典的に手に入るなら、前小節の空間および時間に依存する力場 $F(x, t)$ におけるシミュレーションでの動作と全く変わらない。そのため、以下では3つ目の動作である $\tilde{\rho}(k, t)$ の入手に

ついでのみ述べる。

量子コンピューター上に宛がわれた $f(x, v, t)$ の情報を $\tilde{\rho}(k, t)$ に変換していくには、① $f(x, v, t)$ の積分による $\rho(x, t)$ の入手、② $\rho(x, t)$ の離散フーリエ変換の2つのステップが必要である。これを古典コンピューター上で行うことはできるが、そのためには $f(x, v, t)$ の6次元の情報を毎回取り出す必要が出てきてしまい、計算量が多すぎ現実的ではない。そこで、2つのステップを両方量子コンピューターで行うことを考える。

まず、以下のように量子状態の振幅に情報を入れていく。

$$|\psi\rangle = |q_x\rangle |q_v\rangle = \frac{1}{M} \sum_{j=0}^{N_x-1} \sum_{k=0}^{N_v-1} f_{k;j} |j\rangle |k\rangle, \quad M^2 = \sum_{j=0}^{N_x-1} \sum_{k=0}^{N_v-1} f_{k;j}^2 \quad (3.27)$$

さらに、 $I^{\otimes n_x} \otimes H^{\otimes n_v}$ を掛ける。

$$I^{\otimes n_x} \otimes H^{\otimes n_v} |\psi\rangle = \frac{1}{\sqrt{N_v} M} \sum_{j=0}^{N_x-1} \sum_{k=0}^{N_v-1} \sum_{l=0}^{N_v-1} (-1)^{k \cdot l} f_{k;j} |j\rangle |l\rangle \quad (3.28)$$

$$= \frac{1}{\sqrt{N_v} M} \sum_{j=0}^{N_x-1} \sum_{k=0}^{N_v-1} \left(\sum_{l=0}^{N_v-1} (-1)^{k \cdot l} f_{l;j} \right) |j\rangle |k\rangle \quad (3.29)$$

ただし、演算子 \cdot は bitwise AND の後に2進数の各桁の和をとる関数である。(例えば、 $14 \cdot 23$ は $14 = 01110_{(2)}$ と $23 = 10111_{(2)}$ の bitwise AND が $6 = 00110_{(2)}$ なので、そのバイナリ表記の桁和をとって2である。) 最右辺のカッコの内側に注目すると、 $k=0$ の場合に

$$\rho_j = \sum_{l=0}^{N_v-1} f_{l;j} \quad (3.30)$$

が登場している。この値は $f_{k;j}$ を速度空間において周辺化した値であり、 $\rho(x, t)$ の近似値となっている。つまり、ステップ①の $f(x, v, t)$ の積分を行うことができています。

振幅に情報が入った状態を作り出すことができたので、ステップ②として QFT (2.3.2) を用いて ρ_j を離散フーリエ変換する。

$$\text{QFT} \sum_{j=0}^{N_x-1} \rho_j |j\rangle = \sum_{k=0}^{N_x-1} \tilde{\rho}_k |k\rangle \quad (3.31)$$

以上の2ステップにより、量子コンピューター内に $\tilde{\rho}(k, t)$ の近似値を作り出すことができた。この $\tilde{\rho}(k, t)$ を量子トモグラフィ (2.4) により必要な分だけ取り出すことで古典コンピューターに必要な情報を与え、自己重力を計算させることができる。

さらに、重力の計算を行うタイミングを考える。重力の計算を毎タイムステップ t_n ごとに行った場合と $O(\Delta x/V)$ ごとに行った場合でシミュレーション結果の精度に変化はない。なぜなら、そもそも空間方向の移流を時間刻み $(\Delta x/V)$ と同程度の精度で行なっているからである。

3.2.4 3次元のシミュレーション

上記のシミュレーション手法は、基本的に数を増やすだけで1次元から3次元に拡張できる。量子コンピューター内での $f_{k;j}$ の移流は、空間方向の移流はもちろん速度方向の

移流に関しても 3.1.4 で述べたのと同様に拡張すれば良い。CFL カウンター D_j を 3 次元量 D_j に変更すると、移流のタイミングの決定を 3 次元のケースにおいて行うことができる。最後に、積分に用いた H ゲートを $I^{\otimes n_x} \otimes H^{\otimes n_v}$ から $I^{\otimes 3n_x} \otimes H^{\otimes 3n_v}$ に変更し、 $|q_v\rangle$ にだけかけていた QFT を $|q_{v_x}\rangle, |q_{v_y}\rangle, |q_{v_z}\rangle$ の 3 つにかけるように変更すれば、3 次元量 $\tilde{\rho}_k$ を作り出すことができる。よってこれらを組み合わせれば 3 次元の系に対してシミュレーションを行うことができる。

3.3 テスト計算

本研究の方法をそのまま量子コンピューターに乗せることは、まだ現実味を帯びていない。なぜなら、必要なゲートの数が多すぎて、計算を実行しているうちに結果がノイズに埋もれてしまうからである。そのため、本研究の方法をノイズのない量子コンピューターに乗せた場合にどのような結果が得られるのかを古典的に検証したのが本節のテスト計算の結果である。また、実機を用いた運用をできない以上大きなケースでの計算を行うことができないため、 N_x や N_v といったパラメーターは小さな値に設定されている。同様の理由から 3 次元のシミュレーションはサイズが大きくなってしまっていて実行することが難しかったため、1 次元のシミュレーションを行った。

全ての Reservoir 法のテスト計算は $\Delta x = 1, V = 8$ にて実施した。3.3.4 以外では量子コンピューターから情報を取り出すことに全く制限がないと仮定し、 $\tilde{\rho}(k, t)$ の情報を全て用いることができるとして数値計算を行い、3.3.4 では取り出す情報を制限したときにどのような効果が現れるかを議論する。

3.3.1 Free streaming

簡単な移流をきちんと表現できているかを確認するために、まずは $F = 0$ で計算を行った。その結果を示したのが図 3.5 である。シミュレーションは $n_x = n_v = 6$ で行われ、表 3.1 における $N_v = 64$ での $T_{\text{cycle}} = 8.0$ を 1 cycle として、左上、右上、左下、右下の順に 0, 1, 2, 3 cycle 時点での結果を示している。黒色の領域が $f_{k;j} = 0$ 、白色の領域が $f_{k;j} = 1$ に対応しており、初期状態として与えた「正方形」が時間の経過とともに剪断変形したことがわかる。図は縦軸が速度空間、横軸が位置空間に対応しており、速度の速い上側と下側は右方向と左方向に素早く移動し、速度の遅い中央部はゆっくりと左右に移動している。

この系のハミルトニアン $H = p^2/2m$ は時間に依存しておらず、初期条件は「正方形」領域内のみに一様に割り振られた分布のため、解析力学の Liouville の定理を考えることができる。Liouville の定理によれば、ハミルトニアンから得られる正準方程式に従う運動(移流)をしながら形を変えつつも、位相空間内の体積を保存しながら時間発展するはずである。図 3.5 は位相空間内の体積を保存しながら時間発展の様子を示しており、本方法が Liouville の定理を破らないシミュレーション手法であることが分かる。

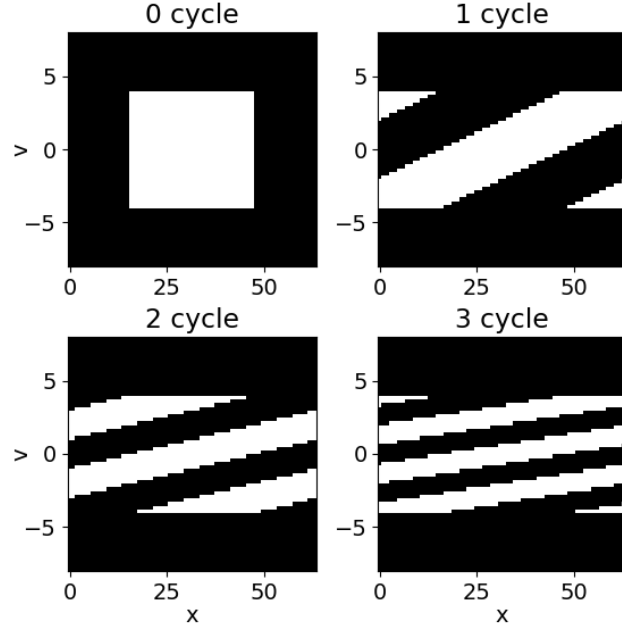


図 3.5: $F = 0, n_x = n_v = 6$ におけるシミュレーション結果。図中の cycle は表 3.1 の $N_v = 64$ における T_{cycle} と対応しており、黒色の領域が $f_{k;j} = 0$ 、白色の領域が $f_{k;j} = 1$ に対応している。

3.3.2 Jeans 不安定性の成長モード

次に、自己重力の効果をきちんと計算できているかを確認するために、自己重力系における定常状態からの摂動の揺らぎを計算した。まず、Vlasov 方程式に関し、自己重力系において Maxwell 分布と呼ばれる以下の定常状態が知られている。

$$f_M(x, v) = \frac{\rho_{\text{ref}}}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{v^2}{2\sigma^2}\right) \quad (3.32)$$

ただし、 ρ_{ref} は全体の平均密度を決めるパラメーターで、 σ は $f_M(x, v)$ がどの程度大きな速度まで値を持つかを決めるパラメーターである。この Maxwell 分布に対して、以下のように摂動を加えたものを初期条件として与える。

$$f_0(x, v) = f_M(x, v)(1 + A \cos kx) \quad (3.33)$$

A の大きさは 1 より小さくなるように取り、 k は $f_0(x, v)$ が x に関して周期的境界条件を守るように取る。

以上の初期条件 $f_0(x, v)$ の設定に登場したパラメーターによって、摂動が大きく成長または減衰していく。その様子を決定付ける Jeans 波数と呼ばれるパラメーター k_J があり、以下で定義される。

$$k_J = \frac{\sqrt{4\pi G \rho_{\text{ref}}}}{\sigma} \quad (3.34)$$

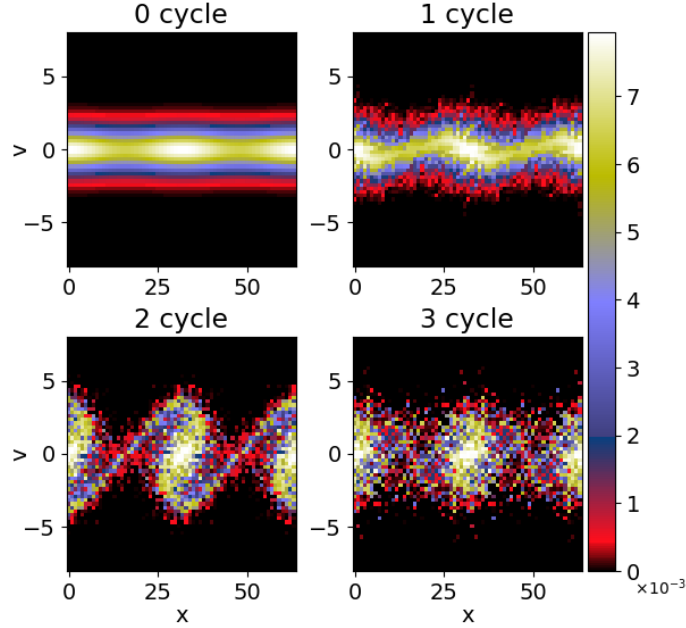


図 3.6: 自己重力系において Maxwell 分布に成長モードの摂動を加え、その時間発展を本アルゴリズムで追った結果。 $n_x = n_v = 6, k/k_J = 0.5, \sigma = 1, k = 4\pi/N_x$ でシミュレーションは行った。

この Jeans 波数 k_J が摂動の波数 k と比べて大きいときは摂動が成長し、小さいときは摂動が減衰することが解析的な計算からわかる (付録 D 参照)。例えば、揺らぎの成長を見たいときに $k/k_J = 0.5$ と設定したとすると、初期条件である $\rho_{\text{ref}}, \sigma, k$ は

$$\frac{\sqrt{4\pi G \rho_{\text{ref}}}}{\sigma} = 2k\sigma \quad (3.35)$$

という関係が成り立つように設定する必要がある。

まず、この $k/k_J = 0.5$ という条件下で摂動を成長させるよう設定した結果が図 3.6 である。この図は $n_x = n_v = 6, k/k_J = 0.5, \sigma = 1, k = 4\pi/N_x$ で行われたシミュレーションであり、関係式 (3.35) から $\rho_{\text{ref}} \simeq 0.01839$ である。また、重力の計算を行う関係上、位相空間内の微小領域における速度分布関数 $f(x, v, t)$ の総量が正しいことが求められる。そのため、初期条件 $f_0(x, v)$ は以下のように離散化した。

$$f_0 \text{ } k;j = \frac{\rho_{\text{ref}}}{2\sqrt{2\pi}\sigma^2} \left(\exp \left[-\frac{(v_k - \Delta v/2)^2}{2\sigma^2} \right] + \exp \left[-\frac{(v_k + \Delta v/2)^2}{2\sigma^2} \right] \right) \quad (3.36)$$

この計算には高精度な古典計算の例があり、文献 [63] の Figure 6 に掲載されている。図 3.6 と高精度計算の結果は共に重力によって密度の大きな部分に集まっていき、揺らぎが大きくなっていく同様の過程を辿っており、本シミュレーション結果がうまく計算を進められていることが窺える。

以上の結果を解析的に調べるために、密度の揺らぎ $\delta(x, t) = \rho(x, t) - \rho_{\text{ref}}$ のフーリエ

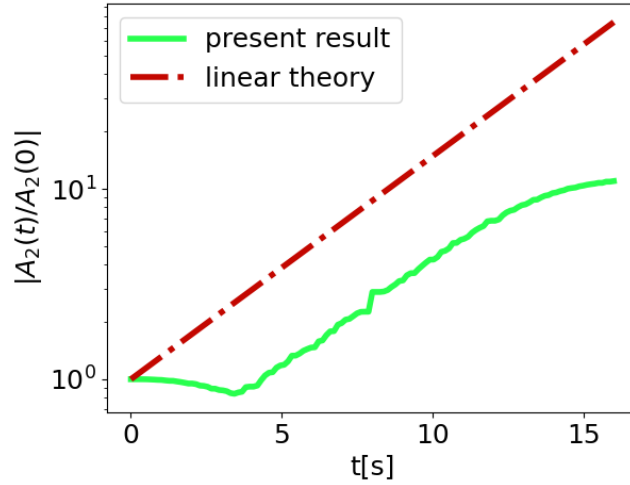


図 3.7: 実線：図 3.6 のシミュレーションと同じセッティングのシミュレーションでの、密度の揺らぎの第 2 フーリエ成分 $A_2(t)$ を $A_2(0)$ で規格化したもの。破線：図 3.6 のセッティングにおける線形成長部分の成長率 γ を計算し、 $\exp[\gamma t]$ を描いたもの。

成分 $A_k(t)$ に注目する。

$$\delta(x, t) = \sum_{k \geq 0} A_k(t) \exp\left(ik \frac{2\pi}{N_x} x\right) \quad (3.37)$$

この量の線形成長部分において、成長率 γ を解析的に計算することができる (付録 D 参照)。図 3.6 のシミュレーションでは k の値を空間方向に対して 2 回並進対称になるよう設定したため、 $A_k(t)$ の値のうち $A_2(t)$ が最も顕著な結果となることが予想される。そこで、図 3.6 のシミュレーションの $A_2(t)/A_2(0)$ を成長率の解析解とともに表示したものが図 3.7 である。シミュレーション結果の密度揺らぎのフーリエ成分 (実線) は $t = 0s$ から $t = 3.5s$ 付近まで減衰するものの、そこから $t = 13s$ 付近まで線形成長を行い、その後は非線形効果により複雑な挙動を示す。確かに線形成長部分は解析解 (破線) と同じ速度で成長しており、正しく自己重力を本アルゴリズムでシミュレーションできていることが分かる。

ところで、最初から $A_2(t)$ が成長しないのには 2 つの理由がある。1 つ目の理由は、成長モードだけでなく減衰モードも含まれるからである (付録 D 参照)。これはシミュレーション手法によらず見られるものであり、問題はない。2 つ目の理由は、力場の効果が反映されるまでにタイムラグがあるからである。これは本アルゴリズム特有の問題である。具体的には、シミュレーションスタートから CFL カウンター D_j が ± 1 に到達するまでは力場の影響は一切シミュレーションに反映されない。そのため、移流のみが進んだことで過度に phase mixing が進み $A_2(t)$ が減衰してしまったのである。詳細は 3.3.3 で述べるが、この問題は N_v を大きく取ることによって解決される。

3.3.3 Jeans 不安定性の減衰モード

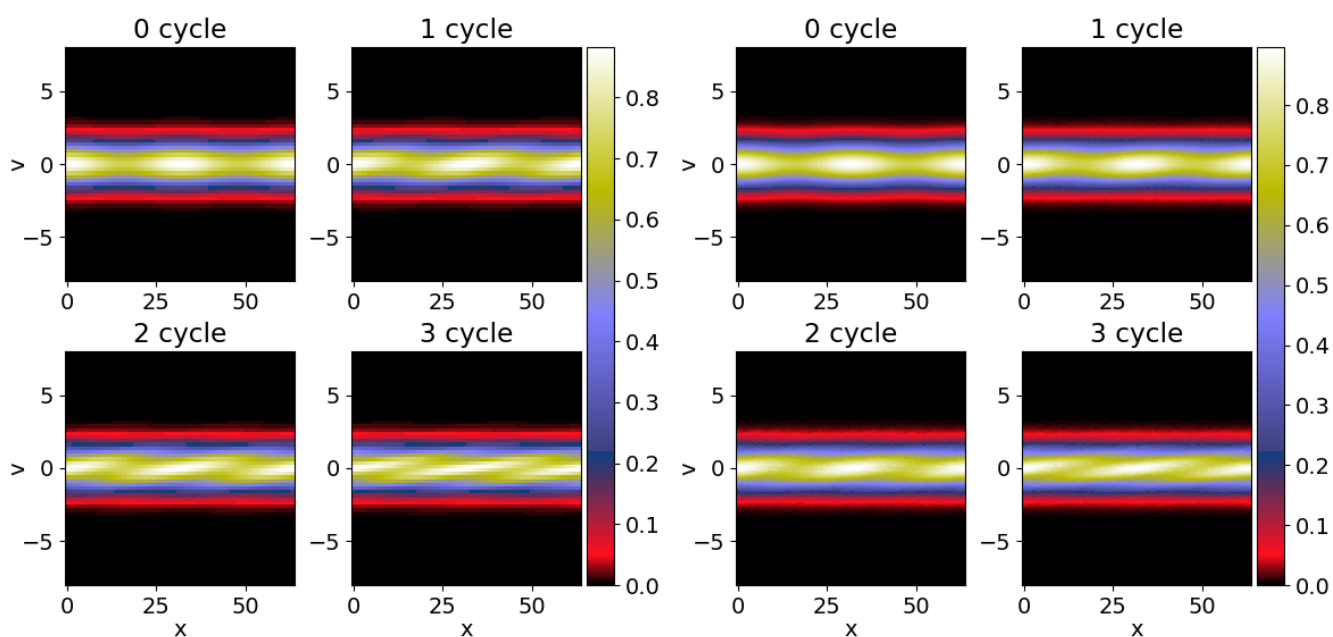


図 3.8: 自己重力系において Maxwell 分布に減衰モードの摂動を加え、その時間発展を本アルゴリズムで追った結果。 $n_x = 6, k/k_J = 1.5, \sigma = 1, k = 4\pi/N_x$ で、 $n_v = 6$ (左図)、 $n_v = 11$ (右図) でシミュレーションは行った。ただし、左右どちらの図も左上、右上、左下、右下の順にスタートから 0, 8, 18, 24 s 経過後の状態を表している。

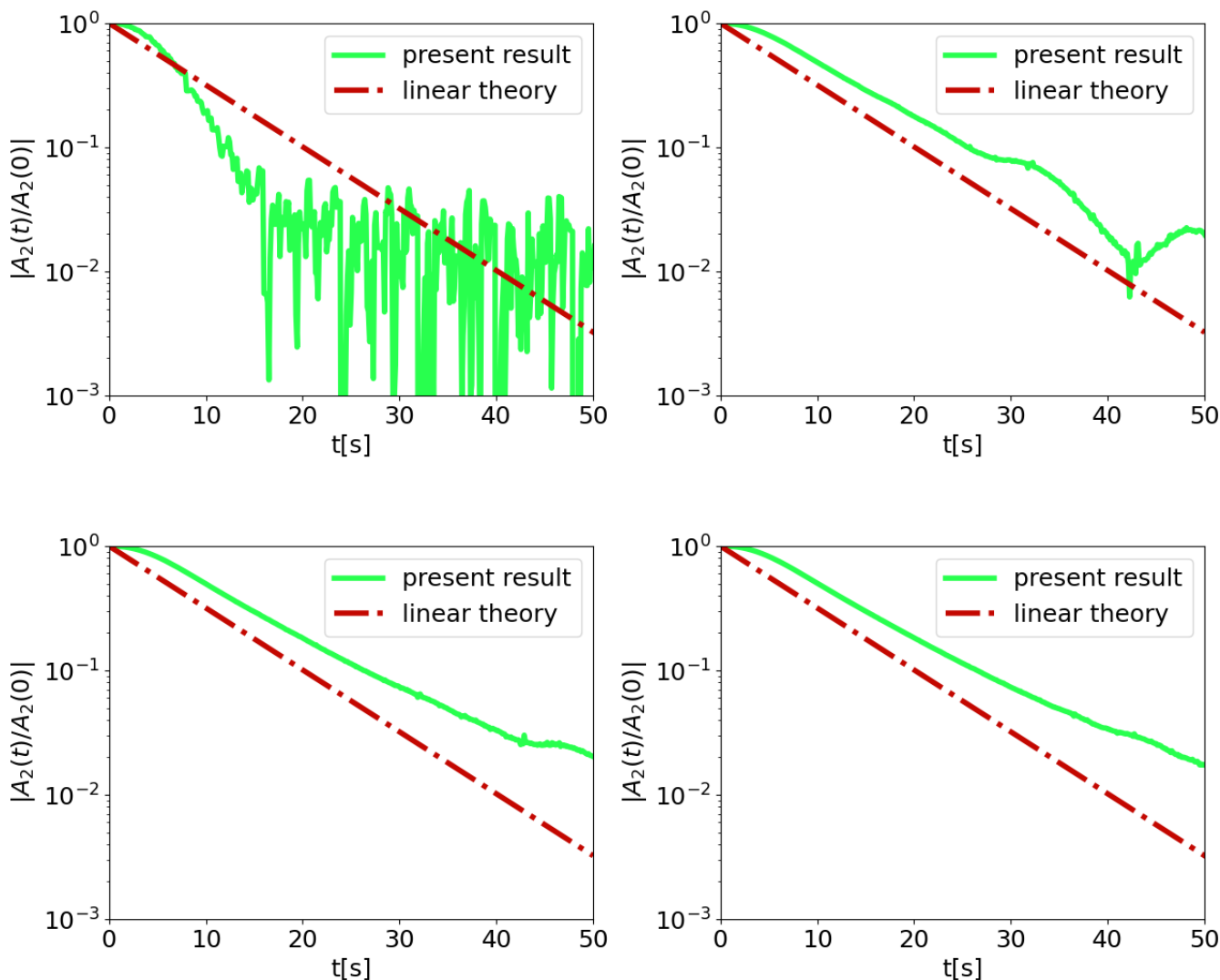


図 3.9: 実線：図 3.8 のシミュレーションと同じセッティングで、 $n_v = 6$ (左上図) $n_v = 11$ (右上図) $n_v = 12$ (左下図) $n_v = 13$ (右下図) における密度揺らぎの第 2 フーリエ成分 $A_2(t)$ の時間発展を追ったもの。破線：線形成長部分の成長率 γ の解析解に基づき、 $\exp[\gamma t]$ を描いたもの。

次に、摂動を減衰させるよう設定した結果が図 3.8 である。 $k/k_J = 1.5, \sigma = 1, k = 4\pi/N_x$ と初期条件のパラメータを設定し、空間方向の解像度は $n_x = 6$ 、速度方向の解像度は左図が $n_v = 6$ で右図が $n_v = 11$ である。以上の設定と式 (3.35) から、 $\rho_{\text{ref}} \simeq 0.002043$ でシミュレーションを行った。これも成長を追った場合と同様に高精度な古典計算の例がある (文献 [63] の Figure 7 参照)。双方 phase mixing が進んでいるためうまく計算が進められていると推測できる。

一方、密度揺らぎのフーリエ成分 $A_2(t)$ の時間発展を追ったものが図 3.9 である。左上、右上の図が図 3.8 の左右のシミュレーションに対応しており、実線が本シミュレーションの結果で、破線が成長率 γ の解析解 (付録 D 参照) を $e^{\gamma t}$ という形で描いたものである。このように $A_2(t)$ の時間発展を追うことで、 $f_{k,j}$ の大きさを見るだけではわからなかった速

度方向の解像度による違いが出てきている。また、左下、右下の図はそれぞれ $n_v = 12, 13$ とした場合のシミュレーションに対応している。線形減衰部分の減衰する速度を解析解と比較すると、 $n_v = 11, 12, 13$ については正しく計算できているものの、 $n_v = 6$ の方は明らかに失敗していることがわかる。線形減衰部分の先の部分にも注目すると、線形減衰を正しく計算できていた $n_v = 11, 12, 13$ の間でも異なる結果が得られていることがわかる。これらの線形減衰部分とそうでない部分での計算結果のずれは、実は性質の異なるものである。というのも、この計算を $n_v = 6$ で重力が存在しないとした $F = 0$ のシミュレーションでも図 3.9 左上図と全く同じ結果が得られる。つまり、 $n_v = 6$ では自己重力系をシミュレーションすることに失敗し、ただ free streaming を行っていたため自己重力による線形減衰を行っていなかったということである。一方線形減衰を終えた後のずれに関しては、線形減衰を追うことができていたことから重力は計算に盛り込まれているものの、非線形領域を追うためにはまだ解像度が足りなかったということである。以下、前者の重力が全くシミュレーションに反映されないという本シミュレーション特有の状況がどのようなケースで発生するのかを見ていく。

系の典型的な力場の大きさを F_s とする。Reservoir 法において力場がシミュレーションに反映されるには、「それなりの頻度」で速度方向の移流が行われなければならない。これは言い換えると、CFL カウンター D_j が「それなりの頻度」で ± 1 に到達する必要があるということである。Reservoir 法の元になった方法である風上差分においてはあるセルに注目した時の空間方向と速度方向の値の更新頻度は等しいので、「それなりの頻度」というのは速度方向の更新の回数が空間方向の更新の回数と釣り合うくらいの頻度と設定できる。速度方向の更新の回数は式 (3.19) より、1 周期 $T_{\text{cycle}} = N_v \Delta x / V$ あたり

$$O\left(\frac{F_s}{\Delta v} \times T_{\text{cycle}}\right) = O\left(\frac{F_s N_v^2 \Delta x}{V^2}\right) \quad (3.38)$$

である。空間方向の更新の回数 $O(N_v)$ と比較し、

$$N_v \geq O\left(\frac{V^2}{F_s \Delta x}\right) \quad (3.39)$$

という結果に至る。

3.3.4 情報の引き出しの効率

情報の取り出しに関して、以下の 2 点がどの程度シミュレーションに影響を与えるのかを Jeans 不安定性の数値実験を通して見ていく。

- ① 量子コンピューターから古典コンピューターへ情報を取り出す際の $\tilde{\rho}(k, t)$ の情報 (式 3.31) への制限は、シミュレーションの精度をどの程度保つことができるのか。
- ② 式 (3.30) は $|q_v\rangle$ の中の $|0\rangle$ のみに注目しているが、 $|0\rangle$ 以外の成分はどの程度情報の取り出しの速度に影響を与えるのか。

まずは①について、情報を取り出す割合を変えて成長モードの時間発展を追ったものが図 3.10 であり、重力計算の際に式 (3.31) の $\tilde{\rho}_k$ を低次の成分に絞って利用したものに

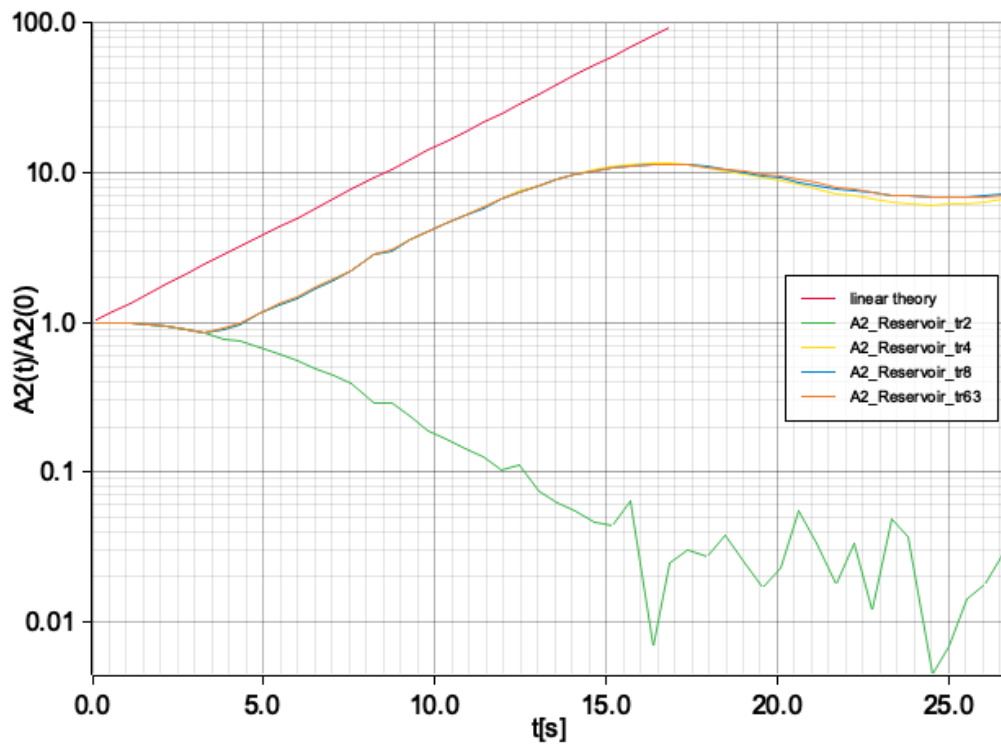


図 3.10: 図 3.6 と同じパラメーターで、重力計算の際に使う情報を制限したもの。緑、黄、青、橙の順に、64 - 1 個の意味のある情報のうち 2,4,8,63 個を取り出した場合に A_2 がどのような時間発展をするかを追ったもの。赤線は線形成長部分の成長率の解析解 γ を $e^{\gamma t}$ の形で描いたもの。青、橙線はほぼ重なってしまっている。

なっている。利用したポアソンソルバー (付録 C 参照) では $\tilde{\rho}_0$ の情報は利用しない。また、 $\tilde{\rho}_k$ のうち $k = 1, \dots, N_x/2$ はそのままフーリエ級数の k 項目の係数の近似値に対応しているが、 $k = N_x/2, \dots, N_x - 1$ についてはフーリエ級数の $k = -N_x/2, \dots, -1$ 項目の係数の近似値に対応している。よって、低次の項 a 個を使うとした場合は $k = 1, \dots, a/2, N_x - a/2, \dots, N_x - 1$ について利用し、それ以外の項は $\tilde{\rho}_k = 0$ として重力を計算した。本シミュレーションでは \cos 型の摂動を与えており、 $\tilde{\rho}_k$ と $\tilde{\rho}_{-k} = \tilde{\rho}_{N_x-k}$ は同じオーダーの大きさを持つことが考えられる。そのため、 a が奇数の場合は考えず、偶数の場合のみを計算した。

図 3.10 について、緑線 ($a = 2$) には線形成長部分が存在せず、これは十分に重力をシミュレーションで捕捉できているとは考えられない。一方、黄線 ($a = 4$)、青線 ($a = 8$) は線形成長部分での成長率が線形成長率の解析解を記した赤線と等しい。また、その後の非線形領域まで含めても、黄線は少しズレてるものの青線は橙線 ($a = 63$, 全ての情報を使ったケース) とぴったり一致しており、この時間の範囲内であれば低次の項 8 つがあれば十分に解析を進めることができる。この黄線のズレは何に起因するのかというと、3 次以上のモードによる変化を $a = 4$ では拾えていないことに起因している。しかし、黄線が線形成長部分の時間発展をは追うことができているのは、 $a = 4$ では最低限 2 次のモードを追うことができているからである。すなわち、 a の値は $a/2$ 次のモードまでは確実に追うことができるということを保証している。

②について、図 3.6 のシミュレーションの初期状態に相当する量子状態に対し、 $I^{\otimes n_x} \otimes H^{\otimes n_v}$ をかけた状態を考える。すると、量子状態の各振幅は図 3.11 のようになる。これを見ると、大きな値をもつ振幅は 64 個ある $|q_v\rangle$ のうち 4 つのみであり、明らかに大きな値を持つ振幅の数が抑えられていることがわかる。つまり、その後の QFT の後の取り出しの回数が増えただけで、レジスタ $|q_x\rangle |q_v\rangle$ が $|0\rangle |0\rangle$ の振幅を取り出すことができるということである。

基本的に、ここでの定数倍が N_x や N_v の大きさによって変わることはない。例えば、2 つの量子レジスタを考え、それらを $'$ のあるなしで区別する。ある N_x と N_v に対して $f_{k;j}$ を考えそれらを $|j\rangle |k\rangle$ に入れ、空間速度両方に対してセル数を 2 倍にした $f'_{k';j'}$ を考えそれらを $|j'\rangle |k'\rangle$ に入れ、

$$f'_{k';j'} = f_{[k'/2];[j'/2]}, \quad (j' = 0, 1, \dots, 2N_x - 1, k' = 0, 1, \dots, 2N_v - 1) \quad (3.40)$$

これらが H ゲート通過後にどのような振幅を持っているかを考える。式 (3.29) の大きな括弧内の値に注目すると、 f' を考えた時 k' が奇数の場合の振幅は 0 となっている。それ以外の k' が偶数の場合は、 $|j'\rangle |k'\rangle$ の振幅は $||j'/2\rangle ||k'/2\rangle$ の振幅の $1/\sqrt{2}$ 倍と等しいため、結果的に定数倍が変わっていないことがわかる。また、現実には解像度を上げると細かい構造が見えてくることもあるが、細かい構造は基本的に大きな構造に比べて速度分布関数の変化量が少ない。そのため、定数倍を大きく変えることはない。

上記の①および②より、本シミュレーションにおいてあらかじめ考えたい最高次数のモードが決まっていれば、それに合わせて取り出す qubit 数を制限することで効率的にシミュレーションを進めることができることが分かった。

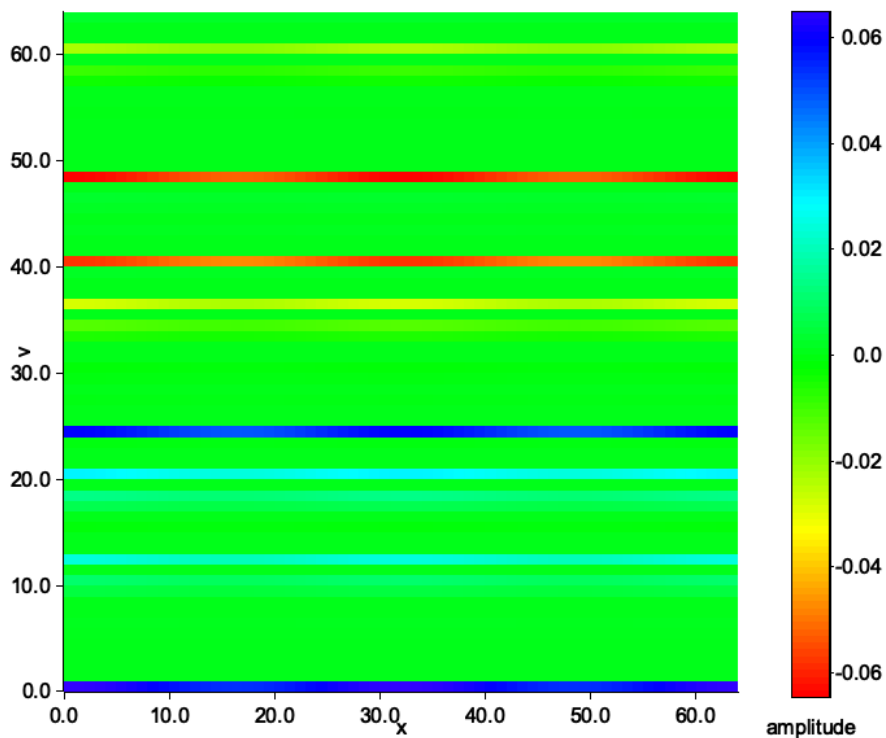


図 3.11: 図 3.6 のシミュレーションの初期状態に相当する量子状態に対し、 $I^{\otimes n_x} \otimes H^{\otimes n_v}$ をかけた状態を表した図。図内は 64×64 に区切られており、左から j 番目、下から k 番目のセルの色は $|q_x\rangle|q_v\rangle$ というレジスタの $|j\rangle|k\rangle$ に対する振幅の大きさを表している。(振幅は本来複素数値であるが、考えている量子状態には実数値の振幅しか登場しないためカラーバーによる表記が可能になっている。)

3.4 計算量

量子計算に対する時間計算量に関しては、一番 2 qubits ゲートがかかる回数が多い qubit に注目し、そのゲート数を考えることとする。なお一般化 Toffoli ゲートに関しては、制御 qubit の数が n 個のものは 2 qubits ゲート $O(n)$ 個分として扱う [49]。 $d+d$ 次元の位相空間を扱うことを考え、シミュレーション内の時間は周期を繰り返した回数 N_{cycle} もしくはその定数倍 $N_t = N_{\text{cycle}}/N_v$ で表す。 N_v は、1 セルに注目した時に空間方向の移流が 1 周期中に起こる回数の平均とオーダー記法の範囲内で一致しているため、 N_t は 1 セルあたり空間方向の移流が行われた平均的な回数とオーダー記法の範囲で一致する。また、平均的な力場の大きさは F_s で表す。

まず、空間および時間依存する力場におけるケースを考える。このケースの古典的な動作は、CFL カウンター D_j や力場 F_j の扱いが一番重い。そしてそれらは全体を通して $O(N_x^d N_t)$ の時間計算量を要する。同様に、初期条件を除いて必要とする古典的なメモリは、力場のデータを保持しておきたいため $O(N_x^d N_t)$ となる。力場のデータを保持する必要がない場合は $O(N_x^d)$ だけで済む。一方量子的な動作は、インクリメントやデクリメントを行うだけである。空間方向の移流を行うものの数は $O(N_v^2 N_{\text{cycle}})$ であり、速度方向の移流を行う回数は式 (3.38) より合計 $O((F_s N_v^2 \Delta x / V^2) N_x^d N_{\text{cycle}})$ 回使われると考えられる。ただし実際には、一回のタイムステップで複数回 (R 回とする) の速度方向の移流が行われる場合、インクリメントおよびデクリメントの末尾から一般化 Toffoli ゲートを削除していくことで添え字の移動する大きさを倍々に変化させていくことができる (2.3.1 参照) ため、インクリメントおよびデクリメントの使用回数を $\log R$ 回に抑えることができる。つまり、速度方向の移流を行う回数は $O(\min(N_v(\log(F_s N_v \Delta x / V^2) + 1), F_s N_v^2 \Delta x / V^2) N_x^d N_{\text{cycle}})$ 回に抑えることができる。インクリメントおよびデクリメントは制御 qubit $O(n_x + n_v)$ 個の一般化 Toffoli ゲートが n_x または n_v 個並んでできたものなので、量子的な時間計算量は

$$O\left((n_x + n_v) \left(n_x N_v^2 + n_v \min\left(N_v \left(\log\left(\frac{F_s N_v \Delta x}{V^2}\right) + 1\right), \frac{F_s N_v^2 \Delta x}{V^2}\right) N_x^d \right) N_{\text{cycle}}\right) \quad (3.41)$$

となる¹。量子コンピューター内で使う領域は $O(n_x + n_v)$ qubits である。

さらに、自己重力系で前述の話がどこまで変わるのかを見ていく。まず空間計算量について、必要な qubit 数は一切変わらず、古典メモリ使用量については F_j を記憶していかなければならないため力場のデータを保持するケースと一致する。時間計算量について、まずは古典の部分について重力の計算が加わる。重力の計算は iFFT でポテンシャルを得る計算が一番重いため、 $O(N_x^d N_t)$ から $O(n_x N_x^d N_t)$ に変化する。量子の部分については、まず H ゲートと QFT が加わる。これらの計算量は前述の計算量を超えないため無視する。次に、重力の計算のために使用する $\tilde{\rho}$ をどの a 次まで考えるとす。これらは量子トモグラフィで取り出すため、 $O(a^{2d})$ 回の量子計算の実行が重力計算のたびに必要になる。取り出しによって量子状態が毎回作り直しになることも考えると、量子の部分での計

¹ N_x と N_v に関する時間計算量の非対称性は、CFL カウンター D_j が d 次元量なのに対して、CFL カウンター C_k が 1 次元の量になることに起因している。というのも、 F_j は \mathbf{x} 空間内全ての点で異なる値を取りうる連続量だが、 v_k は全ての方向に対して同じ方法で離散化された量であり、各方向について分けて考える必要がないからである。

算量は

$$O\left((n_x + n_v)\left(n_x N_v^2 + n_v \min\left(N_v \left(\log\left(\frac{F_s N_v \Delta x}{V^2}\right) + 1\right), \frac{F_s N_v^2 \Delta x}{V^2}\right) N_x^d\right) N_{\text{cycle}} N_t a^{2d}\right) \quad (3.42)$$

となる。 N_v が十分大きい場合であっても量子の部分の方が大きな計算量になっていることから、量子コンピューターの性能が良くなっていくことでこのアルゴリズムの性能は上がっていくことがわかる。

古典コンピューターのみを利用したシミュレーションでは、空間計算量 $O(N_x^d N_v^d)$ 、時間計算量 $O(N_x^d N_v^d N_t)$ である。これに比べ量子コンピューターも用いた本方法は、空間計算量は $O(N_v^d)$ だけ落ちており、時間計算量は時間について2乗になってしまうものの $d=3$ のケースで対数項を無視すれば、 N_x および N_v に関して $O(N_v^3)$ 落ちていることになる。

また、量子計算を並行して行うことができればさらに計算量を落とすことが可能である。例えば $O(a^{2d})$ 回の量子計算は完全に独立であるし、初期状態から毎回やり直さずとも並行計算で直前の状態を作っておけばいいのでその部分も並行計算による恩恵を受けることができる。これにより、自己重力系であっても空間および時間依存する力場におけるシミュレーションと同じ時間計算量で実施することができる。

量子コンピューターのハードウェア開発においては、ノイズのない大規模な qubit 数を持った量子コンピューターが最終目標とされている。このようなマシンが台頭してくれば、本シミュレーションアルゴリズムによる大規模な Vlasov 方程式の計算が現実的なものとして頭角を表してくると考えている。

第4章 Boltzmann 方程式の実装

今までの研究では衝突のない Vlasov 方程式のみを考えていたが、衝突のある Boltzmann 方程式を考えることでさらにシミュレーションできる系の幅が広がる。現在 Boltzmann 方程式の直接シミュレーションは必要な計算量が大きくあまり実用的なシミュレーションは行われていない。しかし、新たに速度の速いアルゴリズムを考えることで、今後様々なシーンで応用のきくシミュレーションアルゴリズムとなることが期待される。

4.1 直接シミュレーションの例

直接シミュレーション法のなかで最も高速化の可能性が高いと考えている方法にスペクトル法がある。このスペクトル法は既に高速化の例がある。スペクトル法で速度のフーリエ空間において登場する特殊な畳み込み計算を、座標変換を用いて必要性の低い情報を遮断しながら畳み込み計算に変換するという方法である [18]。以下、まずはスペクトル法とその高速化の例についてまずは述べる。

4.1.1 スペクトル法

速度方向のメッシュ数を N_v 、フーリエ変換を $\tilde{\cdot}$ で表すとする。衝突項 $C[f]$ をフーリエ空間に持っていくと

$$\tilde{C}_k = \frac{1}{(2L)^3} \int C[f] e^{-i\frac{\pi}{L}k \cdot v} dv \approx \sum_{\substack{l, m = -\frac{N_v}{2} \\ l+m=k}}^{\frac{N_v}{2}-1} \mathcal{G}(l, m) \tilde{f}_l \tilde{f}_m \quad (4.1)$$

が得られる。ただし、フーリエ変換は離散フーリエ変換により近似し、 l, m, k は $\{-\frac{N_v}{2}, -\frac{N_v}{2} + 1, \dots, \frac{N_v}{2} - 1\}^3$ の要素で、 $\mathcal{G}(l, m) = G(l, m) - G(m, m)$ は

$$G(l, m) = \int_{B_R} \int_{S^2} |g| \frac{d\sigma}{d\Omega} \left(|g|, \frac{\boldsymbol{\omega} \cdot \mathbf{g}}{|g|} \right) \exp \left[-i \frac{\pi(l+m)}{2L} \cdot \mathbf{g} + i \frac{|g|\pi(l-m)}{2L} \cdot \boldsymbol{\omega} \right] d\Omega dg \quad (4.2)$$

により近似的に得られる。この時、 \mathbf{g} の積分は半径 R の球 B_R 内について行うとして近似し、 R と L については $S = \inf\{S > 0 \mid \text{Supp}_v(f) \subset B_S\}$ を用いて $R = 2S$ および $L \geq (3 + \sqrt{2})S/2$ として定義される。 f の台については多くの場合 \mathbb{R}^3 となってしまうため、 f の値が近似的に 0 と同一視できる領域は含まないこととする。

以上により、前計算 (4.2) および重みつき畳み込み計算 (4.1) と \tilde{C}_k の逆フーリエ変換により衝突項 $C[f]$ は計算することができる。式 (4.2) の Ω 積分の評価点数を N_Ω 、 \mathbf{g} の

半径方向の積分の評価点数を N_g とすると、前計算には $O(N_v^6 N_g N_\Omega^2)$ の計算量が必要になる。前計算以外では、一番重い計算は重みつき畳み込み計算なので、衝突項の評価が必要な時に毎回 $O(N_v^6)$ の計算が空間セル1つにつき必要になる。また、常に $G(l, m)$ を保持し続ける必要があるため、 $O(N_v^6)$ のメモリを圧迫し続けることになる。

前計算に限ってはモデルごとに計算量を落とすことが可能である。例えば Variable Hard Sphere (VHS) モデル [6] の場合

$$|\mathbf{v}| \frac{d\sigma}{d\Omega} = b_\gamma |\mathbf{v}|^\gamma, \quad b_\gamma > 0, \quad 0 \leq \gamma \leq 1 \quad (4.3)$$

となるが、この時式 (4.2) は

$$G(l, m) = 16\pi^2 b_\gamma \int_0^R r^{\gamma+2} \text{Sinc}\left(\frac{\pi r|l+m|}{2L}\right) \text{Sinc}\left(\frac{\pi r|l-m|}{2L}\right) dr \quad (4.4)$$

と書くことができ、この計算は各 l, m に対して $O(N_g)$ の計算になっている。このように事前に解析的な積分を進めておくことで、実際に計算機にかかる負担を減らすことができる。

4.1.2 先行研究における高速スペクトル法

以下、[18] において示されているスペクトル法の改善案を高速スペクトル法と称し紹介する。

上記の $G(l, m)$ を

$$G(l, m) = \int_0^R \int_{S^2} F(l+m, r, \boldsymbol{\omega}) \exp\left[i\frac{\pi r l}{2L} \cdot \boldsymbol{\omega} - i\frac{\pi r m}{2L} \cdot \boldsymbol{\omega}\right] d\Omega dr \quad (4.5)$$

$$F(l+m, r, \boldsymbol{\omega}) = r^2 \int_{S^2} r \frac{d\sigma}{d\Omega}(r, \boldsymbol{\omega} \cdot \hat{\mathbf{g}}) e^{-i\frac{\pi r(l+m)}{2L} \cdot \hat{\mathbf{g}}} d\hat{\mathbf{g}} \quad (4.6)$$

と変形する。これを Lebedev 求積法 [28] など球面上の積分の求積法を用いることで、

$$G(l, m) \approx \sum_{r, \theta, \phi} w_r w_\theta w_\phi \sin \theta F(l+m, r, \boldsymbol{\omega}) \exp\left[i\frac{\pi r l}{2L} \cdot \boldsymbol{\omega} - i\frac{\pi r m}{2L} \cdot \boldsymbol{\omega}\right] \quad (4.7)$$

という近似を得る。これにより、 \tilde{C}_k の新たな表記として

$$\tilde{C}_k = \tilde{C}_k^+ - \tilde{C}_k^- \quad (4.8)$$

$$\tilde{C}_k^+ \approx \sum_{r, \theta, \phi} w_r w_\theta w_\phi \sin \theta F(l+m, r, \boldsymbol{\omega}) \sum_{\substack{l, m = -\frac{N_v}{2} \\ l+m=k}}^{\frac{N_v}{2}-1} \left[e^{i\frac{\pi r l}{2L} \cdot \boldsymbol{\omega}} \tilde{f}_l \right] \left[e^{-i\frac{\pi r m}{2L} \cdot \boldsymbol{\omega}} \tilde{f}_m \right] \quad (4.9)$$

$$\tilde{C}_k^- = \sum_{\substack{l, m = -\frac{N_v}{2} \\ l+m=k}}^{\frac{N_v}{2}-1} G(m, m) \tilde{f}_l \tilde{f}_m \quad (4.10)$$

を得る。

式 (4.9),(4.10) より、前計算できるのは $F(k, r, \omega)$ と $G(m, m)$ である。 $F(k, r, \omega)$ の計算には $O(N_v^3 N_g N_\Omega^2)$ 、 $G(m, m)$ の計算には $O(N_v^3 N_g N_\Omega)$ の計算量が必要になる。よって全体として、前計算の計算量は $O(N_v^3)$ だけ落ちていることになる。

前計算以外の部分の計算に関しては、 \tilde{C}_k^- がただの畳み込み演算なのでフーリエ空間を経由することで $O(N_v^3 \log N_v)$ に計算量を抑えられる。一方 \tilde{C}_k^+ は、同様に畳み込み部分をフーリエ空間を経由して計算したとしても $O(N_\Omega N_g N_v^3 \log N_v)$ の計算量となるため、一見スペクトル法の $O(N_v^6)$ から悪化したように見える。ところが、実際には $N_\Omega \ll N_v^2$ としても十分に高い精度でシミュレーションを進めることができることがわかっているため [18]、結果として衝突項評価の際に必要な計算量は $O(N_v^3 / (N_\Omega N_g \log N_v))$ 落ちたと言える。また、必要なメモリ容量も $O(N_\Omega N_g N_v^3)$ とスペクトル法の $O(N_v^6)$ から減っている。

高速スペクトル法もモデルを選択することで解析的に前計算の積分を進めることができる場合がある。例えば VHS モデルの場合には

$$F(k, r) = 4\pi b_\gamma r^{\gamma+2} \text{Sinc} \left(\frac{\pi r |k|}{2L} \right) \quad (4.11)$$

$$G(m, m) = 16\pi^2 b_\gamma \int_0^R r^{\gamma+2} \text{Sinc} \left(\frac{\pi r |m|}{L} \right) dr \quad (4.12)$$

となるため、全部で $O(N_g N_v^3)$ の計算量で前計算が終了する。

4.2 高速化の方針

全ての速度分布関数 $f(\mathbf{x}, \mathbf{v}, t)$ および微分断面積 $d\sigma/d\Omega$ に対して素早く作動するようなアルゴリズムを考えるのはおそらく不可能である。これは上記の高速スペクトル法も同じで、論文内で言及されていないが、例えば角度方向に対して細かく複雑な変化をするような微分断面積 $d\sigma/d\Omega$ を考えると、 N_Ω は十分に大きくとらなければ精度が出ずこのアルゴリズムは不向きということになってしまう。逆に、VHS モデルのように速度方向に依存しない微分断面積 $d\sigma/d\Omega$ の場合には N_Ω が小さくて済む (1 で十分) ので、高速スペクトル法が適している。

高速スペクトル法は、直交座標から 3 次元極座標に座標変換することにより高速化を図った。具体的には、式 (4.1) の最右辺の特殊な畳み込みを複数の畳み込み演算に分解し FFT による加速を実現できる形に変更した。これは一般化すると、

$$G(l, m) \approx \sum_{p=1}^{N_p} \alpha_p(l+m) \beta_p(l) \gamma_p(m) \quad (4.13)$$

と変形できる時に \tilde{C}_k^+ の計算を

$$\tilde{C}_k^+ \approx \sum_{p=1}^{N_p} \alpha_p(k) \sum_{\substack{l, m = -\frac{N_v}{2} \\ l+m=k}}^{\frac{N_v}{2}-1} \left(\beta_p(l) \tilde{f}_l \right) \left(\gamma_p(m) \tilde{f}_m \right) \quad (4.14)$$

としていることに他ならない。まず、 \tilde{C}_k^- の計算はただの畳み込みなので FFT を用いて素早く計算でき、問題にならない。残った \tilde{C}_k^+ は上記の変形を通して N_p 個の畳み込み計算に変形でき、空間 1 セルあたり $O(N_p N_v^3 \log N_v)$ で衝突項の計算を行うことができる。この計算量が通常のスเปクトル法における $O(N_v^6)$ よりも小さい場合に、高速化が見込まれることとなる。高速スเปクトル法においては $N_p = N_\Omega N_g$ であり、 N_Ω を十分に小さく取ることのできるという仮定 $N_\Omega \ll N_v^2$ のもと高速化が成り立っている。

この一般化を元に、新たに高速に動作するアルゴリズムを得られる可能性がある。例えば、 $G(l, m)$ を $N_v^3 \times N_v^3$ のサイズの行列だとみなし低ランク近似を用いることで、式 (4.13) の形に分解することができる。 $\{\mathbb{Z}/N_v\mathbb{Z}\}^3$ と $\mathbb{Z}/N_v^3\mathbb{Z}$ を繋ぐ全単射 φ を用いて l, m を 0 から $N_v^3 - 1$ までである 1 次元量とみなす。すると、 $G(l, m)$ は $N_v^3 \times N_v^3$ の行列とみなすことができるようになり、これを G で表す。 G は行列なので $N_v^3 \times N_v^3$ の直交行列 U, V および非負の対角行列 Σ により特異値分解 (付録 E 参照)

$$G = U \Sigma V^\dagger \quad (4.15)$$

することができる。ただし、 $\Sigma_{0,0} \geq \Sigma_{1,1} \geq \dots \geq \Sigma_{N_v^3-1, N_v^3-1}$ となるように特異値分解を行う。行列 M の i 列目の縦ベクトルを M_i で表すと、

$$G = \sum_{p=0}^R \Sigma_{p,p} U_p V_p^\dagger, \quad R = N_v^3 - 1 \quad (4.16)$$

と変形できる。これは先ほどの一般化において

$$\alpha_p = \Sigma_{p,p}, \quad \beta_p(l) = U_{l,p}, \quad \gamma_p(m) = V_{m,p} \quad (4.17)$$

と置くことに対応している。式 (4.16) において R は G のランクに対応しており、この R を小さく取るとは低ランク近似と呼ばれる近似に対応する。低ランク近似は画像データに対してよく用いられる手法であり、一般に行列で表すことのできる量に対する強力な近似の手法である。低ランク近似が機能し、

$$G \approx \sum_{p=0}^R \Sigma_{p,p} U_p V_p^\dagger, \quad R \ll N_v^3 - 1 \quad (4.18)$$

とできる場合に、この方法は上記の高速スเปクトル法と同等もしくはより高い性能を発揮することが期待される。これから、この方法の性能のテストおよび、新たな方法の模索をしていきたいと考えている。

さらに、これらの方法は将来的に量子コンピューターと結びつけることでさらに高速化できる可能性を秘めている。3 では Vlasov 方程式の移流部分を量子コンピューターによって高速化した、Boltzmann 方程式において移流部分は律速段階ではない。そこで、移流部分および重力の計算を古典計算で行い、衝突項を量子計算で行うことを考える。衝突項は 6 次元の量なので量子計算も最速で 6 次元の計算にはなるが、もしその速度で計算を行うことができれば上記のスเปクトル法、高速スเปクトル法よりも十分に早くなっているということができる。例えば、式 (4.1) 最右辺について、一見 \tilde{f} について線形ではない式に見えるが、 \tilde{f}_l と \tilde{f}_m を別のもつとみなせば、 $G(l, m)$ と \tilde{f}_l と \tilde{f}_m という 3 つの要素に

対して線形になる。線形な式ということは量子計算と相性が良く、QRAM などによって G を入力する準備を整えておき、情報の持ち方や掛け算や和の仕方などに工夫を凝らせば十分に計算方法が見つかる可能性があると考えている。また、衝突項は 3 次元の量を受け取って 3 次元の量を返す関数が 3 次元分集まっているとも見ることができる。これは並列処理に向いているということを意味している。大量の量子コンピューターがあれば、古典計算をスーパーコンピューターなどに並列処理させ、衝突項計算だけを量子コンピューターに計算させるような形で並列処理を進めることができる。最終的にはこのような古典量子ハイブリッドの Boltzmann 方程式ソルバーを開発したいと考えている。

第5章 結論

古典コンピューターのみでは、Vlasov 方程式の時間発展を追うために 6 次元の計算が必要となる。これを本研究では、量子コンピューターを用いることで 3 次元の計算にまで落とすことに成功した。

宇宙分野では自己重力系の Vlasov 方程式が非常に重要な存在として語られるが、その非線形性は量子コンピューターの線形性と相性が悪く、今まで量子コンピューターを用いた高速化は研究されてこなかった。そこで本研究では古典量子ハイブリッドの形式をとり、非線形領域を古典コンピューター、線形領域を量子コンピューターで処理することで自己重力系のシミュレーションを可能にした。具体的には、1. シミュレーション自体は量子コンピューターで行い、2. シミュレーションの制御および 3. 重力の計算を古典コンピューターで行うという分業を図った。

1. シミュレーション自体には Reservoir 法と呼ばれる方法を採用し、その量子コンピューターによる実装の先行研究 [55] から、インクリメントおよびデクリメントの量子回路を移流を表現するピースとして利用するというアイデアを採用した。
2. ピースである量子回路を並べる順番を制御するために、CFL カウンターと呼ばれる変数を古典的に採用した。
3. 重力の計算を古典コンピューターで行うにはその情報を量子コンピューターから取り出す必要がある。そのため振幅に入った速度分布関数の情報を H ゲートにより足し合わせ、QFT により密度のフーリエ変換を入手し、その低次の項のみを取り出すこととした。

重力の計算自体は線形な計算であるが、量子コンピューターから取り出す情報をなるべく少なく抑えるために古典計算に割り振ることとなった。

量子コンピューターから取り出す情報は密度のフーリエ変換となるが、その内低次の項のみを選んで取り出すこととなる。この時より高次の項まで取り出そうとするとそれだけシミュレーションが遅くなるというトレードオフが存在する。そのため、ユーザーはあらかじめシミュレーションによって観察したい対象のサイズを具体的に把握し、そのサイズに合わせて何次の項まで取り出すのかを考えておく必要がある。

また、いくつかのテスト計算も行った。量子コンピューターの実機はまだノイズが多く本アルゴリズムの使用には適さないため、古典コンピューターで同様の動作をシミュレートし、 64×64 の小さなケースで Free streaming の確認および Jeans 不安定性を見る数値実験を行った。これらにより、移流を正しく表現できるだけでなく重力もきちんとシミュレーションできていることが分かった。その一方、速度方向の解像度は他のパラメーターから決まるある水準以上に細かく設定しなければならず、粗い場合は重力が反映されずに

移流するだけの結果となることも分かった。さらにテスト計算に用いた速度分布関数の一例を用いて、 H ゲートをかけた後の状態の振幅がどの程度偏っているのかを調べた。その結果、64 個の基底状態のうち欲しい情報である密度の入ったものを含め 4 つの基底状態が支配的であったため、情報の取り出しに大きく支障をきたさないことが分かった。

今後の展望として、これまで衝突項のない Vlasov 方程式のシミュレーションを量子コンピューターにより加速してきたが、最終的には衝突項のある Boltzmann 方程式のシミュレーションを加速することが目標である。Boltzmann 方程式は衝突項の計算が 9 次元の計算となっており、Vlasov 方程式のシミュレーションと比べても非常に重い計算コストを誇っている。古典コンピューターのみを用いた方法であれば、低ランク近似を用いるなど既存のスペクトル法をさまざまな状況に特化させることで擬似的に計算量を落とすような改善を施すということが考えられる。一方量子コンピューターを用いる方法であれば、衝突項は離散化した時にある捉え方をすると線形な式になっており、その式を利用することで量子計算に乗せ、計算量を大きく落とすことのできる可能性がある。このように、今後は古典と量子の双方から衝突項計算の計算コストを削っていきたいと考えている。

関連図書

- [1] Srinivasan Arunachalam, Vlad Gheorghiu, Tomas Jochym-O'Connor, Michele Mosca, and Priyaa Varshinee Srinivasan. On the robustness of bucket brigade quantum RAM. *New Journal of Physics*, Vol. 17, No. 12, p. 123010, December 2015.
- [2] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, David A. Buell, Brian Burkett, Yu Chen, Zijun Chen, Ben Chiaro, Roberto Collins, William Courtney, Andrew Dunsworth, Edward Farhi, Brooks Foxen, Austin Fowler, Craig Gidney, Marissa Giustina, Rob Graff, Keith Guerin, Steve Habegger, Matthew P. Harrigan, Michael J. Hartmann, Alan Ho, Markus Hoffmann, Trent Huang, Travis S. Humble, Sergei V. Isakov, Evan Jeffrey, Zhang Jiang, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Paul V. Klimov, Sergey Knysh, Alexander Korotkov, Fedor Kostritsa, David Landhuis, Mike Lindmark, Erik Lucero, Dmitry Lyakh, Salvatore Mandrà, Jarrod R. McClean, Matthew McEwen, Anthony Megrant, Xiao Mi, Kristel Michielsen, Masoud Mohseni, Josh Mutus, Ofer Naaman, Matthew Neeley, Charles Neill, Murphy Yuezhen Niu, Eric Ostby, Andre Petukhov, John C. Platt, Chris Quintana, Eleanor G. Rieffel, Pedram Roushan, Nicholas C. Rubin, Daniel Sank, Kevin J. Satzinger, Vadim Smelyanskiy, Kevin J. Sung, Matthew D. Trevithick, Amit Vainsencher, Benjamin Villalonga, Theodore White, Z. Jamie Yao, Ping Yeh, Adam Zalcman, Hartmut Neven, and John M. Martinis. Quantum supremacy using a programmable superconducting processor. *Nature*, Vol. 574, No. 7779, pp. 505–510, October 2019.
- [3] Jonathan M. Baker, Casey Duckering, and Frederic T. Chong. Efficient Quantum Circuit Decompositions via Intermediate Qudits. *arXiv e-prints*, p. arXiv:2002.10592, February 2020.
- [4] Dominic W. Berry, Graeme Ahokas, Richard Cleve, and Barry C. Sanders. Efficient Quantum Algorithms for Simulating Sparse Hamiltonians. *Communications in Mathematical Physics*, Vol. 270, No. 2, pp. 359–371, March 2007.
- [5] James Binney and Scott Tremaine. *Galactic Dynamics: Second Edition*. 2008.
- [6] G.A. Bird. *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*. The Oxford engineering science series. Clarendon Press, 1994.

- [7] Ludwig Boltzmann. Weitere Studien über das Wärmegleichgewicht unter Gas-
molekülen. *Sitzungsberichte Akad. Wiss., Vienna*, Vol. part II, No. 66, p. 275–370,
1872.
- [8] F. T. Brandt, R. B. Ferreira, and J. F. Thurst. Classical boltzmann equation and
high-temperature qed. *Phys. Rev. D*, Vol. 91, p. 045023, Feb 2015.
- [9] M. Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C. Benjamin, Suguru Endo,
Keisuke Fujii, Jarrod R. McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cin-
cio, and Patrick J. Coles. Variational Quantum Algorithms. *arXiv e-prints*, p.
arXiv:2012.09265, December 2020.
- [10] Joseph W. Chamberlain. Planetary coroneae and atmospheric evaporation.
Planet. Space Sci., Vol. 11, No. 8, pp. 901–960, August 1963.
- [11] C. Z. Cheng and G. Knorr. The Integration of the Vlasov Equation in Configuration
Space. *Journal of Computational Physics*, Vol. 22, No. 3, pp. 330–351, November
1976.
- [12] Jens Chluba, Daisuke Nagai, Sergey Sazonov, and Kaylea Nelson. A fast and
accurate method for computing the Sunyaev-Zel’dovich signal of hot galaxy clusters.
MNRAS, Vol. 426, No. 1, pp. 510–530, October 2012.
- [13] Lukas Einkemmer. Semi-Lagrangian Vlasov simulation on GPUs. *Computer Physics
Communications*, Vol. 254, p. 107351, September 2020.
- [14] B. Eliasson, P. K. Shukla, and M. E. Dieckmann. Theoretical and simulation studies
of relativistic ion holes in astrophysical plasmas. *New Journal of Physics*, Vol. 8,
No. 4, p. 55, April 2006.
- [15] Alexander Engel, Graeme Smith, and Scott E. Parker. Quantum algorithm for the
vlasov equation. *Phys. Rev. A*, Vol. 100, p. 062315, Dec 2019.
- [16] Moritz Serenus Fischer. *Simulations of Dark Matter with Frequent and Rare Self-
Interactions*. PhD thesis, Hamburg U., 2022.
- [17] Nicolai Friis, Oliver Marty, Christine Maier, Cornelius Hempel, Milan Holzäpfel,
Petar Jurcevic, Martin B. Plenio, Marcus Huber, Christian Roos, Rainer Blatt,
and Ben Lanyon. Observation of entangled states of a fully controlled 20-qubit
system. *Phys. Rev. X*, Vol. 8, p. 021012, Apr 2018.
- [18] Irene M. Gamba, Jeffrey R. Haack, Cory D. Hauck, and Jingwei Hu. A Fast Spectral
Method for the Boltzmann Collision Operator with General Collision Kernels. *SIAM
Journal on Scientific Computing*, Vol. 39, No. 4, pp. B658–B674, January 2017.
- [19] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Architectures for a quan-
tum random access memory. *Phys. Rev. A*, Vol. 78, No. 5, p. 052310, November
2008.

- [20] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Quantum random access memory. *Phys. Rev. Lett.*, Vol. 100, p. 160501, Apr 2008.
- [21] Rajeev Google Quantum AI, Acharya, Igor Aleiner, Richard Allen, Trond I. Andersen, Markus Ansmann, Frank Arute, Kunal Arya, Abraham Asfaw, Juan Atalaya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Joao Basso, Andreas Bengtsson, Sergio Boixo, Gina Bortoli, Alexandre Bourassa, Jenna Bovaird, Leon Brill, Michael Broughton, Bob B. Buckley, David A. Buell, Tim Burger, Brian Burkett, Nicholas Bushnell, Yu Chen, Zijun Chen, Ben Chiaro, Josh Cogan, Roberto Collins, Paul Conner, William Courtney, Alexander L. Crook, Ben Curtin, Dripto M. Debroy, Alexander Del Toro Barba, Sean Demura, Andrew Dunsworth, Daniel Eppens, Catherine Erickson, Lara Faoro, Edward Farhi, Reza Fatemi, Leslie Flores Burgos, Ebrahim Forati, Austin G. Fowler, Brooks Foxen, William Giang, Craig Gidney, Dar Gilboa, Marissa Giustina, Alejandro Grajales Dau, Jonathan A. Gross, Steve Habegger, Michael C. Hamilton, Matthew P. Harrigan, Sean D. Harrington, Oscar Higgott, Jeremy Hilton, Markus Hoffmann, Sabrina Hong, Trent Huang, Ashley Huff, William J. Huggins, Lev B. Ioffe, Sergei V. Isakov, Justin Iveland, Evan Jeffrey, Zhang Jiang, Cody Jones, Pavol Juhas, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Tanuj Khattar, Mostafa Khezri, Mária Kieferová, Seon Kim, Alexei Kitaev, Paul V. Klimov, Andrey R. Klots, Alexander N. Korotkov, Fedor Kostritsa, John Mark Kreikebaum, David Landhuis, Pavel Laptev, Kim-Ming Lau, Lily Laws, Joonho Lee, Kenny Lee, Brian J. Lester, Alexander Lill, Wayne Liu, Aditya Locharla, Erik Lucero, Fionn D. Malone, Jeffrey Marshall, Orion Martin, Jarrod R. McClean, Trevor McCourt, Matt McEwen, Anthony Megrant, Bernardo Meurer Costa, Xiao Mi, Kevin C. Miao, Masoud Mohseni, Shirin Montazeri, Alexis Morvan, Emily Mount, Wojciech Mruczkiewicz, Ofer Naaman, Matthew Neeley, Charles Neill, Ani Nersisyan, Hartmut Neven, Michael Newman, Jiun How Ng, Anthony Nguyen, Murray Nguyen, Murphy Yuezhen Niu, Thomas E. O’Brien, Alex Opremcak, John Platt, Andre Petukhov, Rebecca Potter, Leonid P. Pryadko, Chris Quintana, Pedram Roushan, Nicholas C. Rubin, Negar Saei, Daniel Sank, Kannan Sankaragomathi, Kevin J. Satzinger, Henry F. Schurkus, Christopher Schuster, Michael J. Shearn, Aaron Shorter, Vladimir Shvarts, Jindra Skrzuzny, Vadim Smelyanskiy, W. Clarke Smith, George Sterling, Doug Strain, Marco Szalay, Alfredo Torres, Guifre Vidal, Benjamin Villalonga, Catherine Vollgraf Heidweiller, Theodore White, Cheng Xing, Z. Jamie Yao, Ping Yeh, Juhwan Yoo, Grayson Young, Adam Zalcman, Yaxing Zhang, and Ningfeng Zhu. Suppressing quantum errors by scaling a surface code logical qubit. *Nature*, Vol. 614, No. 7949, pp. 676–681, February 2023.
- [22] Lov K. Grover. A fast quantum mechanical algorithm for database search. *arXiv e-prints*, pp. quant-ph/9605043, May 1996.

- [23] Lov Grover and Terry Rudolph. Creating superpositions that correspond to efficiently integrable probability distributions. *arXiv e-prints*, pp. quant-ph/0208112, August 2002.
- [24] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum Algorithm for Linear Systems of Equations. *Phys. Rev. Lett.*, Vol. 103, No. 15, p. 150502, October 2009.
- [25] E. Knill, R. Laflamme, and G. J. Milburn. A scheme for efficient quantum computation with linear optics. *Nature*, Vol. 409, No. 6816, pp. 46–52, January 2001.
- [26] Jens Koch, Terri M. Yu, Jay Gambetta, A. A. Houck, D. I. Schuster, J. Majer, Alexandre Blais, M. H. Devoret, S. M. Girvin, and R. J. Schoelkopf. Charge-insensitive qubit design derived from the cooper pair box. *Phys. Rev. A*, Vol. 76, p. 042319, Oct 2007.
- [27] Hari Krovi. Improved quantum algorithms for linear and nonlinear differential equations. *Quantum*, Vol. 7, p. 913, February 2023.
- [28] V. I. Lebedev and Dimitri N. Laikov. A quadrature formula for the sphere of the 131st algebraic order of accuracy. *Doklady Mathematics*, Vol. 59, pp. 477–481, 1999.
- [29] Wu Li. Electrical transport limited by electron-phonon coupling from Boltzmann transport equation: An ab initio study of Si, Al, and MoS₂. *Phys. Rev. B*, Vol. 92, No. 7, p. 075405, August 2015.
- [30] Daniel Litinski and Felix von Oppen. Quantum computing with Majorana fermion codes. *Phys. Rev. B*, Vol. 97, No. 20, p. 205404, May 2018.
- [31] Y. Y. Liu, Z. Chen, C. Shu, S. C. Chew, B. C. Khoo, X. Zhao, and Y. D. Cui. Application of a variational hybrid quantum-classical algorithm to heat conduction equation and analysis of time complexity. *Physics of Fluids*, Vol. 34, No. 11, p. 117121, November 2022.
- [32] Lars S. Madsen, Fabian Laudenbach, Mohsen Falamarzi. Askarani, Fabien Rortais, Trevor Vincent, Jacob F. F. Bulmer, Filippo M. Miatto, Leonhard Neuhaus, Lukas G. Helt, Matthew J. Collins, Adriana E. Lita, Thomas Gerrits, Sae Woo Nam, Varun D. Vaidya, Matteo Menotti, Ish Dhand, Zachary Vernon, Nicolás Quesada, and Jonathan Lavoie. Quantum computational advantage with a programmable photonic processor. *Nature*, Vol. 606, No. 7912, pp. 75–81, June 2022.
- [33] Jr. McNutt, R. L., John Lyon, and Charles C. Goodrich. Simulation of the heliosphere: Model. *J. Geophys. Res.*, Vol. 103, No. A2, pp. 1905–1912, February 1998.

- [34] Thomas E. C. Merrall and Richard N. Henriksen. Relaxation of a Collisionless System and the Transition to a New Equilibrium Velocity Distribution. *ApJ*, Vol. 595, No. 1, pp. 43–58, September 2003.
- [35] Koichi Miyamoto, Soichiro Yamazaki, Fumio Uchida, Kotaro Fujisawa, and Naoki Yoshida. Quantum algorithm for the Vlasov simulation of the large-scale structure formation with massive neutrinos. *arXiv e-prints*, p. arXiv:2310.01832, October 2023.
- [36] Pablo Antonio Moreno Casares. Circuit implementation of bucket brigade qRAM for quantum state preparation. *arXiv e-prints*, p. arXiv:2006.11761, June 2020.
- [37] Hiroki Nagakura. Global features of fast neutrino-flavor conversion in binary neutron star mergers. *Phys. Rev. D*, Vol. 108, No. 10, p. 103014, November 2023.
- [38] Cuang I. L. Nielsen, M. A. *Quantum Computation and Quantum Information*. Cambridge University Press, 10th anniversary edition edition, 2010.
- [39] Murphy Yuezhen Niu, Alexander Zlokapa, Michael Broughton, Sergio Boixo, Masoud Mohseni, Vadim Smelyanskiy, and Hartmut Neven. Entangling Quantum Generative Adversarial Networks. *Phys. Rev. Lett.*, Vol. 128, No. 22, p. 220505, June 2022.
- [40] Alexandru Paler, Oumarou Oumarou, and Robert Basmadjian. Parallelizing the queries in a bucket-brigade quantum random access memory. *Phys. Rev. A*, Vol. 102, No. 3, p. 032608, September 2020.
- [41] Lorenzo Pareschi and Giovanni Russo. Numerical solution of the boltzmann equation I: spectrally accurate approximation of the collision operator. *SIAM J. Numer. Anal.*, Vol. 37, No. 4, pp. 1217–1245, 2000.
- [42] Daniel K. Park, Francesco Petruccione, and June-Koo Kevin Rhee. Circuit-Based Quantum Random Access Memory for Classical Data. *Scientific Reports*, Vol. 9, p. 3949, March 2019.
- [43] Koustubh Phalak, Avimita Chatterjee, and Swaroop Ghosh. Quantum Random Access Memory for Dummies. *Sensors*, Vol. 23, No. 17, p. 7462, August 2023.
- [44] Koustubh Phalak, Junde Li, and Swaroop Ghosh. Approximate Quantum Random Access Memory Architectures. *arXiv e-prints*, p. arXiv:2210.14804, October 2022.
- [45] Anupam Prakash. *Quantum Algorithms for Linear Algebra and Machine Learning*. PhD thesis, University of California, Berkeley, 2014.
- [46] François Rincon, Francesco Califano, Alexander A. Schekochihin, and Francesco Valentini. Turbulent dynamo in a collisionless plasma. *Proceedings of the National Academy of Science*, Vol. 113, No. 15, pp. 3950–3953, April 2016.

- [47] Andrew Robertson, Richard Massey, and Vincent Eke. Cosmic particle colliders: simulations of self-interacting dark matter with anisotropic scattering. *MNRAS*, Vol. 467, No. 4, pp. 4719–4730, June 2017.
- [48] C. A. Ryan, M. Laforest, and R. Laflamme. Randomized benchmarking of single- and multi-qubit control in liquid-state NMR quantum information processing. *New Journal of Physics*, Vol. 11, No. 1, p. 013034, January 2009.
- [49] Mehdi Saeedi and Massoud Pedram. Linear-depth quantum circuits for n-qubit Toffoli gates with no ancilla. *Phys. Rev. A*, Vol. 87, No. 6, p. 062318, June 2013.
- [50] Xiaowen Shan and Hudong Chen. Lattice Boltzmann model for simulating flows with multiple phases and components. *Phys. Rev. E*, Vol. 47, No. 3, pp. 1815–1819, March 1993.
- [51] P.W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pp. 124–134, 1994.
- [52] Victor Sofonea and Robert F. Sekerka. Viscosity of finite difference lattice Boltzmann models. *Journal of Computational Physics*, Vol. 184, No. 2, pp. 422–434, January 2003.
- [53] Yoshio Sone. Thermal Creep in Rarefied Gas. *Journal of the Physical Society of Japan*, Vol. 21, No. 9, pp. 1836–1837, September 1966.
- [54] Shuntaro Takeda, Kan Takase, and Akira Furusawa. On-demand photonic entanglement synthesizer. *Science Advances*, Vol. 5, No. 5, p. eaaw4530, May 2019.
- [55] Blaga N. Todorova and René Steijl. Quantum algorithm for the collisionless Boltzmann equation. *Journal of Computational Physics*, Vol. 409, p. 109347, May 2020.
- [56] Kiichiro Toyozumi, Naoki Yamamoto, and Kazuo Hoshino. Hamiltonian simulation using quantum singular value transformation: complexity analysis and application to the linearized Vlasov-Poisson equation. *arXiv e-prints*, p. arXiv:2304.08937, April 2023.
- [57] Cora Uhlemann, Michael Kopp, and Thomas Haugg. Schrödinger method as N-body double and UV completion of dust. *Phys. Rev. D*, Vol. 90, No. 2, p. 023517, July 2014.
- [58] Jan Veltmaat, Jens C. Niemeyer, and Bodo Schwabe. Formation and structure of ultralight bosonic dark matter halos. *Phys. Rev. D*, Vol. 98, No. 4, p. 043509, August 2018.
- [59] Václav Špicka and Pavel Lipavský. Quasiparticle Boltzmann equation in semiconductors. *Phys. Rev. Lett.*, Vol. 73, No. 25, pp. 3439–3442, December 1994.

- [60] Hui Wang, Jian Qin, Xing Ding, Ming-Cheng Chen, Si Chen, Xiang You, Yu-Ming He, Xiao Jiang, L. You, Z. Wang, C. Schneider, Jelmer J. Renema, Sven Höfling, Chao-Yang Lu, and Jian-Wei Pan. Boson sampling with 20 input photons and a 60-mode interferometer in a 10^{14} -dimensional hilbert space. *Phys. Rev. Lett.*, Vol. 123, p. 250503, Dec 2019.
- [61] Jun Yoneda, Kenta Takeda, Tomohiro Otsuka, Takashi Nakajima, Matthieu R. Delbecq, Giles Allison, Takumu Honda, Tetsuo Koderu, Shunri Oda, Yusuke Hoshi, Noritaka Usami, Kohei M. Itoh, and Seigo Tarucha. A quantum-dot spin qubit with coherence limited by charge noise and fidelity higher than 99.9%. *Nature Nanotechnology*, Vol. 13, No. 2, pp. 102–106, December 2018.
- [62] Kohji Yoshikawa, Satoshi Tanaka, and Naoki Yoshida. A 400 trillion-grid vlasov simulation on fugaku supercomputer: Large-scale distribution of cosmic relic neutrinos in a six-dimensional phase space. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC '21*, New York, NY, USA, 2021. Association for Computing Machinery.
- [63] Kohji Yoshikawa, Naoki Yoshida, and Masayuki Umemura. Direct Integration of the Collisionless Boltzmann Equation in Six-dimensional Phase Space: Self-gravitating Systems. *ApJ*, Vol. 762, No. 2, p. 116, January 2013.
- [64] 青木一生 著曾根良夫, 日本流体力学会編集. 分子気体力学. 流体力学シリーズ. 朝倉書店, 新装版, 2020.
- [65] 蔦原道久. 格子ボルツマン法の基礎と応用. 日本機械学会論文集, Vol. 77, No. 784, pp. 149–160, October 2011.
- [66] 義皓 (著) 情報処理学会出版委員会 (監修) 嶋田. 量子コンピューティング —基本アルゴリズムから量子機械学習まで—. オーム社, 2020.

付録 A 古典的な論理ゲート

古典的な論理ゲートは1つまたは複数の真偽値に対して行われる演算のことである。ここでは真偽値は0と1で表す。論理ゲートにはたくさんの種類があるが、そのうち重要なものについていくつか紹介する。

A.1 NOT

入力	出力
0	1
1	0

表 A.1: NOT ゲートを表す真偽値表。

NOT ゲートは1 bit の入力に対して真偽値を反転させる演算である (表 A.1)。

A.2 AND

入力	出力
0 0	0
0 1	0
1 0	0
1 1	1

表 A.2: AND ゲートを表す真偽値表。

AND ゲートは2 bits の入力に対して両方が1の時のみ1を返し、そうでない場合には0を返す演算である (表 A.2)。

A.3 OR

OR ゲートは2 bits の入力に対して両方が0の時のみ0を返し、そうでない場合には1を返す演算である (表 A.3)。

入力		出力
0	0	0
0	1	1
1	0	1
1	1	1

表 A.3: OR ゲートを表す真偽値表。

入力		出力
0	0	0
0	1	1
1	0	1
1	1	0

表 A.4: XOR ゲートを表す真偽値表。

A.4 XOR

XOR ゲートは 2 bits の入力に対して 2 つの入力が異なる時に 1 を返し、同じときに 0 を返す演算である (表 A.4)。

A.5 NAND、NOR、XNOR

入力		出力
0	0	1
0	1	1
1	0	1
1	1	0

表 A.5: NAND ゲートを表す真偽値表。

NAND / NOR / XNOR ゲートは、AND / OR / XOR ゲートの後に NOT ゲートがかかったものである (表 A.5, A.6, A.7)。

入力		出力
0	0	1
0	1	0
1	0	0
1	1	0

表 A.6: NOR ゲートを表す真偽値表。

入力		出力
0	0	1
0	1	0
1	0	0
1	1	1

表 A.7: XNOR ゲートを表す真偽値表。

付録B 移流方程式の性質と解法

x, t をそれぞれ位置と時間を表す変数としたとき、移流方程式とは

$$\frac{\partial u(x, t)}{\partial t} = -\frac{\partial cu(x, t)}{\partial x} \quad (\text{B.1})$$

という形をした微分方程式のことである。

B.1 移流方程式の性質

c が x, t によらない定数の場合には、 $u(x, t)$ の解は

$$u(x, t) = u(x - ct, 0) \quad (\text{B.2})$$

という形になる。これは、初期条件の形を維持したまま $u(x, t)$ が c という速度を持って移動していると解釈することができる。 c が x や t に依存する場合には上記のような簡単な形の解にはならないものの、 c が $u(x, t)$ の速度に相当するパラメータとなることは変わらない。

B.2 移流方程式の解法

移流方程式のシミュレーションを行う際には、まず移流方程式を離散化する必要がある。そして離散化した式に従って時間発展を追っていくことになる。離散化した $u(x, t)$ は u_j^n と書くこととし、これは時刻 $t_n = \Delta t \times n$ における位置 $x_j = \Delta x \times j$ の $u(x_j, t_n)$ の近似値である。離散化の方法はいくつも存在しており、そのうちいくつか代表的なものを以下で紹介する。ただし、どの場合においても Courant-Friedrichs-Lewy (CFL) 条件と呼ばれる

$$\frac{c\Delta t}{\Delta x} < 1 \quad (\text{B.3})$$

が満たされる必要がある。CFL 条件はシミュレーション内での情報の伝わる速度が系の速度 c を上回っているということに対応している。もし CFL 条件が満たされない場合は、シミュレーションで追うことのできる速度よりも早く系が移動するということになり、正しく時間発展を追うことができない。

B.2.1 Forward in Time and Centered Difference in Space スキーム

Forward in Time and Centered Difference in Space (FTCS) スキームは中心差分法とも呼ばれ、一番簡単な 2 次精度のケースでは以下のように離散化される。

$$u_j^{n+1} = u_j^n - \frac{c\Delta t}{2\Delta x}(u_{j+1}^n - u_{j-1}^n) \quad (\text{B.4})$$

この方法は一番簡単な離散化でも 1 次ではなく 2 次精度となるように精度が出せるという利点はあるものの、安定性に欠けるという欠点があり、得られる解が振動してしまう。

B.2.2 風上差分法

風上差分法は一番簡単な 1 次精度のケースで以下のように離散化される。

$$u_j^{n+1} = u_j^n - \frac{c\Delta t}{\Delta x} \begin{cases} u_j^n - u_{j-1}^n & (c > 0) \\ f_{j+1}^n - f_j^n & (c < 0) \end{cases} \quad (\text{B.5})$$

この方法は先ほどの FTCS スキームと違って精度は低くなってしまう。しかし、この離散化後の式は FTCS スキームのものに対して拡散項を追加したと解釈することができ、安定性がある解法となっているが数値散逸が発生してしまう。

B.2.3 その他の解法

上記の解法はそれぞれ高次精度のものが存在し、それらは $u(x, t)$ のテイラー展開から導くことができる。上記のもの以外にも、風上差分法とは異なる方法で FTCS スキームに拡散項を加えて不安定性の除去を目指した Lax-Wendroff 法 と呼ばれる方法や、風上差分法から数値散逸が発生しないよう時間刻みの取り方を工夫した Reservoir 法などがあり、様々なものが知られている。

付 録 C 自己重力系における重力計算

自己重力系では、力場が速度分布関数 $f(\mathbf{x}, \mathbf{v}, t)$ を通して与えられる。速度分布関数を速度 \mathbf{v} に関して周辺化すると数密度

$$\rho(\mathbf{x}, t) = \int f(\mathbf{x}, \mathbf{v}, t) d\mathbf{v} \quad (\text{C.1})$$

が得られ、重力定数 G を用いて重力ポテンシャル $\Phi(\mathbf{x}, t)$ が以下のように与えられる。

$$\begin{cases} \Delta\Phi(\mathbf{x}, t) = 2G\rho(\mathbf{x}, t) & (1 \text{次元}) \\ \Delta\Phi(\mathbf{x}, t) = 4\pi G\rho(\mathbf{x}, t) & (3 \text{次元}) \end{cases} \quad (\text{C.2})$$

また、重力ポテンシャルから力場 $F(\mathbf{x}, t)$ が

$$F(\mathbf{x}, t) = -\nabla\Phi(\mathbf{x}, t) \quad (\text{C.3})$$

により得られる。

グリーン関数 $\mathcal{G}(\mathbf{x}, \mathbf{x}')$ は微分作用素 L に対して

$$L\mathcal{G}(\mathbf{x}, \mathbf{x}') = -\delta(\mathbf{x} - \mathbf{x}') \quad (\text{C.4})$$

を満たす関数であるが、 $L = \Delta$ の時式 (C.4) は以下の解を持つ。

$$\begin{cases} \mathcal{G}(x, x') = \frac{|x - x'|}{2} & (1 \text{次元}) \\ \mathcal{G}(\mathbf{x}, \mathbf{x}') = \frac{1}{4\pi|\mathbf{x} - \mathbf{x}'|} & (3 \text{次元}) \end{cases} \quad (\text{C.5})$$

これを用いると、ポテンシャル $\Phi(\mathbf{x}, t)$ は密度 $\rho(\mathbf{x}, t)$ を用いて

$$\begin{cases} \Phi(x, t) = -2G \int \mathcal{G}(x, x')\rho(x', t) dx' & (1 \text{次元}) \\ \Phi(\mathbf{x}, t) = -4\pi G \int \mathcal{G}(\mathbf{x}, \mathbf{x}')\rho(\mathbf{x}', t) d\mathbf{x}' & (3 \text{次元}) \end{cases} \quad (\text{C.6})$$

と表せる。グリーン関数 $\mathcal{G}(\mathbf{x}, \mathbf{x}')$ は $\mathbf{x} - \mathbf{x}'$ を通してのみ依存関係を持つので、誤解の無い範囲で記号の濫用を恐れずに $\mathcal{G}(\mathbf{x} - \mathbf{x}') = \mathcal{G}(\mathbf{x}, \mathbf{x}')$ と書くと式 (C.6) は

$$\begin{cases} \Phi(x, t) = -2G \int \mathcal{G}(x - x')\rho(x', t) dx' & (1 \text{次元}) \\ \Phi(\mathbf{x}, t) = -4\pi G \int \mathcal{G}(\mathbf{x} - \mathbf{x}')\rho(\mathbf{x}', t) d\mathbf{x}' & (3 \text{次元}) \end{cases} \quad (\text{C.7})$$

となり畳み込みの形をしている。

よって、

$$\begin{cases} \tilde{\Phi}(k, t) = -2G \tilde{\mathcal{G}}(k) \tilde{\rho}(k, t) & (1 \text{次元}) \\ \tilde{\Phi}(\mathbf{k}, t) = -4\pi G \tilde{\mathcal{G}}(\mathbf{k}) \tilde{\rho}(\mathbf{k}, t) & (3 \text{次元}) \end{cases} \quad (\text{C.8})$$

が成り立つ。ただし、周期的境界条件が \mathbf{x} 空間に課されていた場合は $\tilde{\cdot}$ は複素フーリエ級数展開のフーリエ係数を指し、そうでない場合はフーリエ変換を指す。そのため、周期的境界条件の場合は \mathbf{k} は整数値のベクトルであり、そうでない場合は実数値のベクトルを取る。

グリーン関数のフーリエ係数 $\tilde{\mathcal{G}}(\mathbf{k})$ は、 \mathbf{x} 空間の周期の単位が L だとして

$$\begin{cases} \tilde{\mathcal{G}}(k) = \frac{L}{4\pi^2 k^2} & (1 \text{次元}) \\ \tilde{\mathcal{G}}(\mathbf{k}) = \frac{1}{4\pi^2 L \mathbf{k}^2} & (3 \text{次元}) \end{cases} \quad (\text{C.9})$$

で与えられ、フーリエ変換の場合は

$$\tilde{\mathcal{G}}(\mathbf{k}) = \frac{1}{\mathbf{k}^2} \quad (\text{C.10})$$

で与えられる。

ただし、これを離散フーリエ変換によって近似する場合には、各方向に対して 0 から $N_x - 1$ までの N_x 個の要素のうち前半分の要素が $k \geq 0$ に対応し、後ろ半分の要素が $k < 0$ に対応する。そのため、例えば周期的境界条件で 1 次元の場合には

$$\tilde{\mathcal{G}}_k = \begin{cases} \frac{L}{4\pi^2 k^2} & (0 \leq k < N_x/2) \\ \frac{L}{4\pi^2 (N_x - k)^2} & (N_x/2 \leq k < N_x) \end{cases} \quad (\text{C.11})$$

となる。

以上を周期的境界条件のケースについてまとめると、Boltzmann シミュレーションのあるタイミング t における重力 $F(\mathbf{x}, t)$ の情報が必要になった時、以下の手順で効率的に計算を進めることができる。

1. 速度分布関数 $f(\mathbf{x}, \mathbf{v}, t)$ を \mathbf{v} に関して数値積分し、密度 $\rho(\mathbf{x}, t)$ の近似値を得る。
2. Fast Fourier Transform (FFT) により密度のフーリエ係数 $\tilde{\rho}(\mathbf{k}, t)$ を得る。
3. $\tilde{\rho}$ とグリーン関数のフーリエ係数 (式 C.9) を掛け合わせ、次元に合わせて定数倍をすることで (式 C.8) 重力ポテンシャルのフーリエ係数 $\tilde{\Phi}(\mathbf{k}, t)$ を得る。
4. inverse FFT (iFFT) により、重力ポテンシャル $\Phi(\mathbf{x}, t)$ を得る。
5. 数値微分により式 (C.3) を計算し、重力 $F(\mathbf{x}, t)$ を近似的に得る。

シミュレーションを d 次元空間で行い、空間を N_x^d 個、速度空間を N_v^d 個のメッシュに切ったとすると、それぞれの過程における古典的な計算量は 1. $O(N_x^d N_v^d)$, 2. $O(N_x^d \log(N_x))$,

3. $O(N_x^d)$, 4. $O(N_x^d \log(N_x))$, 5. $O(N_x^d)$ となる。グリーン関数を用いた微分方程式の解析を通してポアソン方程式 (C.2) を畳み込みの形 (式 C.7) にすることで、高い精度を保ちつつ少ない計算量で重力計算を行うことができる。また、何らかの手段で密度のフーリエ係数 $\tilde{\rho}(\mathbf{k}, t)$ が手に入るなら、3. から計算を行うことになり、全体として $O(N_x^d \log(N_x))$ の計算になる。

付録D Jeans 不安定性における揺らぎの成長と減衰

自己重力系を考える。Vlasov 方程式 (1.6) において、Maxwell 分布 $f_M(\mathbf{v})$ は定常解である。

$$f_M(\mathbf{v}) = \frac{\rho_{\text{ref}}}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\mathbf{v}^2}{2\sigma^2}\right) \quad (\text{D.1})$$

ただし、 ρ_{ref} および σ は定数である。

この定常解の空間方向に摂動を加え、その摂動がどのように系に影響を与えるのかを見ていく。まず、加えられた摂動が $\epsilon f_1(\mathbf{x}, \mathbf{v}, t)$ という形をしているとする。 f_1 は f_M と同じ程度の大きさであり、 $0 < \epsilon \ll 1$ である。 f_M における重力ポテンシャルを Φ_M とし、摂動を加えた後の Φ_M からの誤差を $\epsilon\Phi_1(\mathbf{x}, t)$ とする。

速度分布関数 $f(\mathbf{x}, \mathbf{v}, t) = f_M(\mathbf{v}) + \epsilon f_1(\mathbf{x}, \mathbf{v}, t)$ や重力ポテンシャル $\Phi(\mathbf{x}, t) = \Phi_M + \epsilon\Phi_1(\mathbf{x}, t)$ を Vlasov 方程式やポアソン方程式 (C.2) に代入し ϵ の 1 次についての項を集めると、

$$\frac{\partial f_1}{\partial t} + \mathbf{v} \cdot \frac{\partial f_1}{\partial \mathbf{x}} - \frac{\partial \Phi_1}{\partial \mathbf{x}} \cdot \frac{\partial f_M}{\partial \mathbf{v}} = 0 \quad (\text{D.2})$$

$$\Delta \Phi_1 = 4\pi G \int f_1 d\mathbf{v} \quad (\text{D.3})$$

となる。これらを \mathbf{x} 空間に関してフーリエ変換し、フーリエ空間を \mathbf{k} で表すことにすると

$$\frac{\partial \bar{f}_1}{\partial t} + i\mathbf{k} \cdot \mathbf{v} \bar{f}_1 = i\bar{\Phi}_1 \mathbf{k} \cdot \frac{\partial f_M}{\partial \mathbf{v}} \quad (\text{D.4})$$

$$-k^2 \bar{\Phi}_1 = 4\pi G \bar{\rho}_1 \quad (\text{D.5})$$

を得る。ただし、 $\bar{\cdot}$ は \mathbf{x} 空間に関するフーリエ変換を表している。方程式 (D.4) は

$$\bar{f}_1(\mathbf{k}, \mathbf{v}, t) = i\mathbf{k} \cdot \frac{\partial f_M}{\partial \mathbf{v}} \int_{-\infty}^t e^{i\mathbf{k} \cdot \mathbf{v}(t-t')} \bar{\Phi}_1(\mathbf{k}, t') dt' \quad (\text{D.6})$$

という解を持つ。これに方程式 (D.5) も考えると、

$$\bar{\rho}_1(\mathbf{k}, t) = -\frac{4\pi G i}{k^2} \int \mathbf{k} \cdot \frac{\partial f_M}{\partial \mathbf{v}} \int_{-\infty}^t e^{i\mathbf{k} \cdot \mathbf{v}(t-t')} \bar{\rho}_1(\mathbf{k}, t') dt' d\mathbf{v} \quad (\text{D.7})$$

を得る。

さらに、時間 t に関するフーリエ変換を施し、フーリエ空間を ω と表すと

$$\tilde{\bar{\rho}}_1(\mathbf{k}, \omega) = -\frac{4\pi G}{k^2} \int \frac{1}{\mathbf{k} \cdot \mathbf{v} - \omega} \mathbf{k} \cdot \frac{\partial f_M}{\partial \mathbf{v}} \tilde{\bar{\rho}}_1(\mathbf{k}, \omega) d\mathbf{v}, \quad \text{Im}(\omega) > 0 \quad (\text{D.8})$$

を得る。ただし、 \sim は時間に関するフーリエ変換を表している。以上より、

$$-\frac{4\pi G}{k^2} \int \frac{1}{\mathbf{k} \cdot \mathbf{v} - \omega} \mathbf{k} \cdot \frac{\partial f_M}{\partial \mathbf{v}} d\mathbf{v} = 1, \quad \text{Im}(\omega) > 0 \quad (\text{D.9})$$

である。この重積分のうち \mathbf{k} と垂直な成分は積分を行い、平行な成分 v_k については $x = kv_k$ として置換積分を行い f_M に具体的な形を代入すると、

$$-\frac{4\pi G \rho_{\text{ref}}}{\sqrt{2\pi} k^3 \sigma^3} \int_{-\infty}^{\infty} \frac{x e^{-x^2/(2k^2\sigma^2)}}{x - \omega} dx = 1, \quad \text{Im}(\omega) > 0 \quad (\text{D.10})$$

となる。よってプラズマ分散関数と呼ばれる

$$Z(w) = \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} \frac{e^{-s^2}}{s - w} ds, \quad \text{Im}(w) > 0 \quad (\text{D.11})$$

を用いると、

$$\frac{k^2}{k_J^2} = 1 + wZ(w), \quad \omega = \sqrt{2}k\sigma w \quad (\text{D.12})$$

とまとめることができる。ただし、 k_J は Jeans 波数と呼ばれる量で

$$k_J = \frac{4\pi G \rho_{\text{ref}}}{\sigma^2} \quad (\text{D.13})$$

で定義される。

今まで $\text{Im}(\omega) > 0$ の範囲でのみ議論をしてきたが、 $(-\infty, \infty)$ の範囲の積分を複素積分と捉え、Landau contour と呼ばれる積分経路を考えることで $\text{Im}(\omega) \leq 0$ にも $Z(w)$ を解析接続することができる。

$$Z(w) = \begin{cases} \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} \frac{e^{-s^2}}{s - w} ds & (\text{Im}(w) > 0) \\ \frac{1}{\sqrt{\pi}} \text{P} \int_{-\infty}^{\infty} \frac{e^{-s^2}}{s - w} ds + i\sqrt{\pi} e^{-w^2} & (\text{Im}(w) = 0) \\ \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} \frac{e^{-s^2}}{s - w} ds + 2i\sqrt{\pi} e^{-w^2} & (\text{Im}(w) < 0) \end{cases} \quad (\text{D.14})$$

以下この $Z(w)$ を用いて、 k に対してありえる ω を探っていく。

まず、 $\text{Im}(\omega) > 0$ のケースを考える。 $wZ(w)$ は虚部が 0 でなければならないので、

$$\text{Im}(wZ(w)) = \frac{\text{Im}(w)}{\sqrt{\pi}} \int_{-\infty}^{\infty} \frac{se^{-s^2}}{|s - w|^2} ds = 0 \quad (\text{D.15})$$

である。 $\text{Re}(w) = 0$ の場合は、式 (D.15) の被積分関数が奇関数となるため $\text{Im}(wZ(w)) = 0$ である。それ以外の場合、 $\text{Re}(w) > 0$ の時は $\text{Im}(wZ(w)) > 0$ で、 $\text{Re}(w) < 0$ の時は $\text{Im}(wZ(w)) < 0$ である。つまり、 $\omega = \sqrt{2}k\sigma w$ は 0 もしくは純虚数なので、実数 γ を用いて $\omega = i\gamma$ と表せる。この時、式 (D.12) は

$$\frac{k^2}{k_J^2} = 1 - \frac{\sqrt{\pi}\gamma}{\sqrt{2}k\sigma} \exp\left(\frac{\gamma^2}{2k^2\sigma^2}\right) \left[1 - \text{erf}\left(\frac{\gamma}{\sqrt{2}k\sigma}\right)\right] \quad (\text{D.16})$$

となる。ただし、 erf は誤差関数と呼ばれ、以下の式で定義される関数である。

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \quad (\text{D.17})$$

以上の式から γ の値は決定されるが、 $\text{Im}(\omega) > 0$ すなわち $\gamma > 0$ という仮定があったので、 $k < k_J$ の場合のみに対応する。

次に、 $\text{Im}(\omega) = 0$ のケースを考える。 $\text{Im}(wZ(w)) = \sqrt{\pi}we^{-w^2} = 0$ なので、 $\omega = 0$ である。これは、 $k = k_J$ の場合にのみ対応している。

最後に、 $\text{Im}(\omega) < 0$ のケースを考える。このケースでは、各 $k > 0$ に対して加算無限個の $\omega \in \mathbb{C}$ の解が存在する [5]。 $\text{Im}(\omega) > 0$ のケースからの類推で ω が純虚数または 0 の解があると考え、実数 $\gamma < 0$ に対して $\omega = i\gamma$ と仮定する。この時、

$$\text{Im}(w \times i\sqrt{\pi}e^{-w^2}) = \text{Im}(-\sqrt{\pi}\gamma e^{\gamma^2}) = 0 \quad (\text{D.18})$$

となるため、分散関係は $\text{Im}(\omega) > 0$ のケースと同様の式 (D.16) で表される。この式は解くことができるので、 $\omega = i\gamma$ とした仮定は合っていた。この解は $\gamma < 0$ であることから $k > k_J$ にのみ対応している。また、文献 [5] の Figure 5.2 よりこの解は $k > k_J$ に対応する ω のうち、一番 $\text{Im}(\omega) = \gamma$ が大きな解である。

さて、 ω は時間 t のフーリエ変換のパラメーターである。そのため、それぞれの ω が $e^{-i\omega t}$ というモードに対応しており、虚部が 0 より大きい場合は成長モード、小さい場合は減衰モードとなる。また、 $\gamma = \text{Im}(\omega)$ はモードの成長率を表すこととなる。

以上の議論をまとめ、以下の事実が判明した。

1. 各 $k > 0$ に対して無限に解 ω が存在する。
2. その中で最大の虚部を持つ解、つまり一番支配的な解は式 (D.16) を解くことで得られる。
3. 全ての $k > 0$ には $\gamma < 0$ の減衰モードの解が存在する。
4. Jeans 波数 k_J より k が小さい場合にのみ $\gamma > 0$ の成長モードの解が存在する。

付録E 特異値分解

整数 $m \leq n$ に対して $m \times n$ 複素行列 M を考える。 MM^\dagger は $m \times m$ のエルミート行列なので、 $m \times m$ ユニタリ行列 U と実数を対角要素にもつ対角行列 Σ^2 により

$$MM^\dagger = U\Sigma^2U^\dagger \quad (\text{E.1})$$

このように対角化可能である。 MM^\dagger は半正定値行列なので、 Σ^2 の対角要素は全て非負であり、それらの平方根を対角要素に並べた $m \times n$ の対角行列を Σ とする。この時、 $n \times n$ 複素行列 V を

$$V = M^\dagger U \Sigma^{-1} \quad (\text{E.2})$$

により定義すると、 $VV^\dagger = V^\dagger V = I$ より V はユニタリ行列である。 $(\Sigma^{-1}$ は $n \times m$ の対角行列で、 Σ の対角成分の逆数を並べたものである。) これにより、特異値分解と呼ばれる分解

$$M = U\Sigma V^\dagger \quad (\text{E.3})$$

が定義される。また、 Σ の対角成分は特異値と呼ばれる。 M が実数行列の場合には、 U, V は直交行列となる。