

有限要素法に基づく第一原理分子動力学法について

土田 英二

産業技術総合研究所 計算科学研究部門

E-mail: Eiji.Tsuchida@aist.go.jp

1 はじめに

人間の眼に見えるようなマクロな世界は通常ニュートン方程式やマクスウェル方程式によって記述されるが、電子の波動性が重要な意味を持つようなミクロ・ナノ系の振る舞いは量子物理学の基礎方程式であるシュレーディンガー方程式によって決定されている [1]。この方程式自体は電子の質量などの自然定数のみを含み、経験的なパラメータを一切必要としない。従って、もしある系のシュレーディンガー方程式を十分な精度で解くことができれば、任意の物理・化学的性質を非常に正確に知ることが出来る。このようなアプローチは入力パラメータとして実験値を一切使用しないことから、第一原理計算と呼ばれる [2]。一昔前までは小さな分子や完全結晶を扱うことができる程度であったが、現在では計算機およびアルゴリズムの発展のお陰で、1000 原子を超える系を十分な精度で調べることができるようになった。当然ながら第一原理計算の応用範囲は広く、孤立分子に始まり、固体・液体やそれらの表面・界面等の複雑な系まで適用され、他の方法では得難い情報を与えてくれる [3]。特に最近では材料科学 [4] や地球科学 [5]、生命科学 [6, 7] など、より応用的な分野へと適用範囲を広げつつある [8]。

実際にシュレーディンガー方程式、もしくはその近似式を解く場合、何らかの方法で離散化して数値的に扱う必要があるが、通常は平面波基底 [2] や原子基底 [9] のような（物理的に意味のある）基底関数を使用するのが標準手法となっている。歴史的に言えば、平面波は周期性が本質的な意味を持つような系、即ち完全結晶の電子状態計算において最も自然な選択であり、計算機の進歩と共に適用対象を広げてきた。また、原子軌道は少数の原子から成るような孤立分子の電子状態を効率よく表現するために広く使われてきた。

一方、有限要素法は主に流体解析や構造解析といった用途に適用されることから「マクロな系の解法」というイメージが定着しており [10, 11]、上述のようなミクロ系における数値解法として採用されることは非常に稀であった。しかしここ 10 年の間に、こういった系に対しても有限要素法や差分法のようなマクロ系における標準手法が有用であるということが認められるようになってきた [12, 13]。これは主に、大規模な並列計算機との相性の良さや、オーダー N 法 [14, 15]・ハイブリッド法 [16] 等の新しいアルゴリズムへの拡張性が期待されているためである。以下では、特に有限要素法に基づく第一原理計算の現状について紹介する。

ここで、まず手短かに量子物理学における有限要素法の歴史を振り返っておきたい。一・二次元問題における有限要素法の登場は少なくとも 1970 年代に遡る [12, 13]。一方、応用上重要な系は対称性を持たないことが多いため、三次元問題として扱わざるを得ない。

当然ながら三次元の場合は一・二次元の場合と比べて遥かに計算量が大きく、より洗練された手法が必要とされる。三次元問題における本格的な応用が現れたのは80年代であり、Whiteらによりハートリー近似 [9] に基づく計算がヘリウム原子や水素分子等の系について行われた [17]。この計算では一様な格子を用いていたが、場所によって基底関数の数を変えることで計算の効率を上げていた。

90年代に入ると、より有限要素法の特長を生かした応用が行われるようになった。まず、四面体要素と局所的分割を組み合わせた手法により、いくつかの一電子系について計算が行われた [18]。また、我々のグループでは、六面体要素と局所的分割を用いて水素分子やシリコン結晶の第一原理計算を行い、期待通りの結果を得ることができた [19]。ただし、計算コストの点から見てこの時点での有限要素法の性能は平面波や原子基底を用いた場合と比べると不十分であった。この点を改善するために、我々は一次微分まで連続な滑らかな基底関数と曲線座標系を組み合わせた方法を提案した [20]。更に、スーパーコンピュータ向けにこの方法を実装したコード FEMTECK の開発を続けている [21]。以下ではこのアプローチについて他の方法との比較も交えて紹介したい。なお、我々以外にも国内外で多くの研究者が現在有限要素法を用いた第一原理計算の研究を活発に進めている [22, 23, 24, 25, 26, 27]。

2 問題の定式化

まず、本稿で扱う問題をここで一通り紹介する。今考えている系の中に含まれる原子の数を N 、それぞれの三次元座標を $\mathbf{R}_1, \mathbf{R}_2, \dots, \mathbf{R}_N$ で表す。そうすると、原子座標 $(\mathbf{R}_1, \mathbf{R}_2, \dots, \mathbf{R}_N)$ の関数として系の全エネルギー E_{total} 及び各原子に働く力

$$\mathbf{F}_\alpha = -\frac{\partial E_{\text{total}}}{\partial \mathbf{R}_\alpha}, \quad \alpha = 1, 2, \dots, N$$

を求めることが最も基本的な作業となる。実際に E_{total} を計算する手続きは使用するアルゴリズムによって大きく異なるが、いずれにしても \mathbf{F}_α を求めるためには \mathbf{R}_α の滑らかな関数でなくてはならない。ひとたび E_{total} と \mathbf{F}_α が得られれば後は同じであり、それらの情報を利用して系の安定構造を決定したり、分子動力学計算を行うことができる。また、安定構造に対して振動数や振動モードを求めることも可能である。

例えばアルゴンのような希ガス系の振る舞いはレナード・ジョーンズ型と呼ばれる古典的な二体ポテンシャルで比較的正確に記述されることが知られており、

$$E_{\text{total}} = \sum_{\alpha < \beta} \phi(|\mathbf{R}_\alpha - \mathbf{R}_\beta|)$$

のように、 E_{total} を $\mathbf{R}_1, \dots, \mathbf{R}_N$ の関数として陽に与えることができる [28]。この場合には \mathbf{F}_α も容易に計算することができることは明らかであろう。共有結合のようなより複雑な相互作用で記述される系では球対称でなく方向性を持つポテンシャルがしばしば使われるが、その場合でもやはり E_{total} は \mathbf{R}_α の陽な関数として与えられることが多い [28]。

一方、第一原理計算の場合はもっと複雑であり、以下に述べるように原子が移動する度に電子状態を計算する必要がある。原理的には、シュレーディンガー方程式を正確に解けば E_{total} や F_α 等必要な情報は全て得られるのであるが、複数の原子から成り立つような系、特に数十原子以上を含むような系において、これは事実上不可能である。従って、何らかの近似を導入する必要がある訳であるが、大雑把に言って一電子近似に基づく密度汎関数法 [29, 30, 31] と、ハートリー・フォック近似を基本とする伝統的な量子化学的手法 [9] が二本柱として普及している。それぞれ様々な長所・短所があって使い分けられる傾向があるが、応用する立場から見ると、周期境界条件の下で液体や固体などの系を扱う場合には、密度汎関数法の方がかなりコストパフォーマンスが良い。一方、量子化学的な手法は一電子近似では不十分な場合に多体効果を段階的に取り入れることが可能であり、小分子であれば極めて高精度の計算を行うことができる。我々はできる限り大規模な系を一電子近似の範囲で扱うことを目指しているため、以下では密度汎関数法のみを扱うこととする。

まず密度汎関数法における全エネルギーの表式を原子単位系で書き下すと、

$$E_{\text{total}} = \sum_{i=1}^M \int \psi_i(\mathbf{r}) \left[-\frac{1}{2} \nabla^2 + V(\mathbf{r}) \right] \psi_i(\mathbf{r}) d\mathbf{r} + \frac{1}{2} \int \frac{n(\mathbf{r})n(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r}d\mathbf{r}' \\ + \int n(\mathbf{r}) \epsilon_{xc}(n(\mathbf{r})) d\mathbf{r} + \sum_{\alpha < \beta} \frac{Z_\alpha Z_\beta}{|\mathbf{R}_\alpha - \mathbf{R}_\beta|} \quad (1)$$

のようになる [2]。ただし M は電子数、 $\psi_1, \psi_2, \dots, \psi_M$ は各電子の波動関数であり、正規直交条件

$$\int \psi_i(\mathbf{r}) \cdot \psi_j(\mathbf{r}) d\mathbf{r} = \delta_{ij}, \quad i, j = 1, \dots, M \quad (2)$$

を満たすものとする。また、 $n(\mathbf{r})$ は電子密度であり

$$n(\mathbf{r}) = \sum_{i=1}^M |\psi_i(\mathbf{r})|^2 \quad (3)$$

で与えられる。 Z_α は各原子の電荷数を表し、 $V(\mathbf{r})$ は各原子が電子に及ぼすポテンシャルの総和である。なお、 ϵ_{xc} は交換相関エネルギー密度と呼ばれる $n(\mathbf{r})$ の関数であるが、具体的な関数形は分かっていない。従って実際に第一原理計算を行う際には何らかの近似式を使用する必要がある。良く使われるものとしては、局所密度近似あるいは密度勾配近似が挙げられる [2, 3]。

上記の E_{total} は一見 ψ_i と \mathbf{R}_α の両方に依存しているように見えるが、実際には拘束条件 (2) の下で E_{total} を最小にする ψ_i を求め、その ψ_i に対する E_{total} を系の全エネルギーと見なす。従って、 E_{total} は \mathbf{R}_α のみの関数として一意に定まる。又、こうして得られた ψ_i を使って F_α も容易に計算できる。この手順を以下では電子状態計算と呼び、具体的な計算方法については 3.3 章で説明する。

この時、各波動関数は以下のコーン・シャム方程式を満たすことが知られている [29, 30] :

$$H_{\text{KS}} \psi_i = \epsilon_i \psi_i, \quad i = 1, \dots, M \quad (4)$$

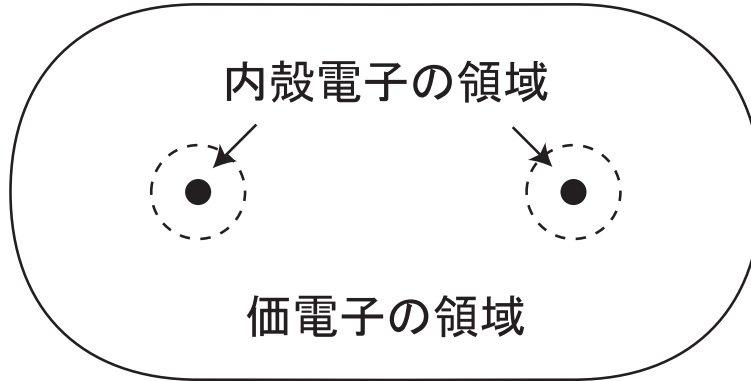


図 1: 内殻電子の波動関数は各原子核 (●) のごく近傍に局在しているが、価電子の波動関数は主にその外側に分布している。

ただし、

$$H_{\text{KS}} = -\frac{1}{2}\nabla^2 + V(\mathbf{r}) + \int \frac{n(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r}' + \mu_{xc}(\mathbf{r}) \quad (5)$$

及び

$$\mu_{xc}(\mathbf{r}) = \frac{\partial}{\partial n(\mathbf{r})} (n(\mathbf{r})\epsilon_{xc}(n(\mathbf{r}))) \quad (6)$$

であり、 ϵ_i は各波動関数の固有エネルギーとする [2]。この方程式は一電子に対するシュレーディンガー方程式と同じ形をしているが、他の電子の影響を全てポテンシャルの中に取り込んでいるため、 H_{KS} が ($n(\mathbf{r})$ 経由で) 解である ψ_i にも依存している点が異なる。

境界条件については通常、孤立境界条件 (遠方で $\psi_i(\mathbf{r}) \rightarrow 0$) か、周期境界条件 (L を計算領域のサイズとして $\psi_i(\mathbf{r} + \mathbf{L}) = \psi_i(\mathbf{r})$) が使われる。前者は分子等の孤立した系向けであり、後者は固体等の凝縮系に向いているが、十分計算領域を大きく取れば (場合によっては補正を加えることにより) 周期境界条件で孤立系を扱うことも可能である。数値計算上都合が良いため、以下では常に周期境界条件を満たすものとする。また、周期境界条件の場合には波動関数に位相因子 $\exp(i\mathbf{k}\mathbf{r})$ が付くが、大規模系への応用を重視しているため、ガンマ点 ($\mathbf{k} = 0$) のみを考え、かつ波動関数は実数であるものとして扱う。

更に、ここでもう一つの重要な近似について説明しておく。 E_{total} の表式中に現れる $V(\mathbf{r})$ について考察すると、これは各原子が電子に及ぼす影響を表しており、近似しない場合には $-\sum_{\alpha} Z_{\alpha}/|\mathbf{r} - \mathbf{R}_{\alpha}|$ という形をしている。すなわち、生のクーロン・ポテンシャルの和で与えられる。この形のまま計算を進める場合、全電子計算と呼ばれる。このような相互作用を受けた結果、各電子は原子核の近傍に局在する傾向があるが、その中でも特に強く原子核に束縛された内殻電子と、その外側にいて他の原子との結合に寄与する価電子に分けることができる (図 1)。

通常、内殻電子は非常に安定であり、原子のいる環境にはほとんど影響を受けない [2, 32]。従って、価電子のみを陽に扱い、 $V(\mathbf{r})$ の式を「原子核 + 内殻電子」が価電子に及ぼすポテンシャルに変更することで内殻電子を計算から除外することが良い近似として成り立

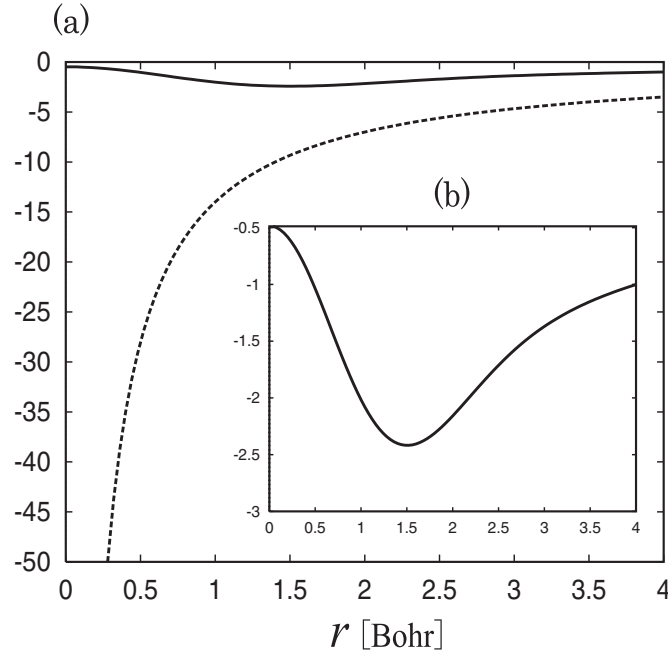


図 2: (a) シリコンに対する擬ポテンシャル [33] (実線) 及び生のクーロン・ポテンシャル (破線)。 (b) 擬ポテンシャルを拡大したもの。

つ。この変更後の $V(r)$ を擬ポテンシャルと呼ぶ。擬ポテンシャルを使うことにより、全電子計算より遥かに少ない数の要素で十分収束した解を得ることができる。また、内殻電子を計算から除くことができるため、陽に扱う電子の数を大幅に減らすことができる。従って、ある程度大きな系の第一原理計算を行う場合 (特に重元素を含む時) には擬ポテンシャルは必須の道具であり、以下でも特に明示しない限り常に擬ポテンシャルを使用するものとする。すなわち、全エネルギー表式中の $V(r)$ は擬ポテンシャルであり、また価電子の波動関数のみを考えるものとする。図 2 に擬ポテンシャルの例を示す。

なお、実際に高精度でかつ扱いやすい擬ポテンシャルを作成するのはかなり職人芸的な知識と経験を必要とする作業である。最近は多くの元素について十分なチェックを受けた擬ポテンシャルのリストが出版されており [34]、我々のコードもこれを利用している。

3 FEMTECK の構成

3.1 基底関数

計算機上で電子状態計算を行うためには何らかの基底関数を用いて波動関数を展開する必要がある。我々はエルミート補間式 [35] と呼ばれる区分多項式を基底関数として使用している。この補間式は任意の奇数次に対して定義できるが、FEMTECK では三次のも

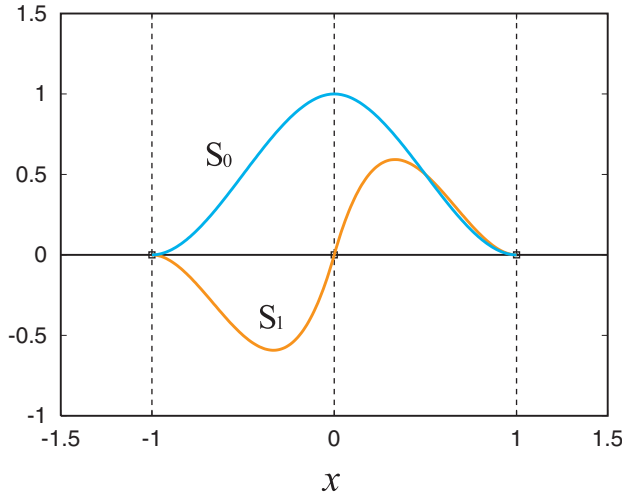


図 3: 原点に属する三次のエルミート補間式 $S_0(x), S_1(x)$ 。ただし $S_1(x)$ は見易いように拡大している。

のに特化している。まず簡単のため、一次元の格子上で 1 の間隔で節点が並んでいるものとする。この時、原点には 2 個の基底関数 $S_0(x), S_1(x)$ が割り当てられ、これらは条件

$$S_0(0) = 1, \quad S_0'(0) = 0, \quad S_0(\pm 1) = S_0'(\pm 1) = 0 \quad (7)$$

および

$$S_1(0) = 0, \quad S_1'(0) = 1, \quad S_1(\pm 1) = S_1'(\pm 1) = 0 \quad (8)$$

を満たすように定義される。これらを満たす三次の区分多項式として、

$$S_0(x) = \begin{cases} 1 - 3x^2 + 2x^3, & 0 \leq x \leq 1 \\ S_0(-x), & -1 \leq x < 0 \\ 0, & \text{その他} \end{cases} \quad (9)$$

および

$$S_1(x) = \begin{cases} x - 2x^2 + x^3, & 0 \leq x \leq 1 \\ -S_1(-x), & -1 \leq x < 0 \\ 0, & \text{その他} \end{cases} \quad (10)$$

が得られる。参考のため $S_0(x)$ 及び $S_1(x)$ を図 3 に示す。また、他の節点における基底関数も単純にこれらを平行移動することで得られる。そして、波動関数などの未知関数は全ての節点におけるこれらの基底関数の線形結合で近似される。式で表現すると、

$$\psi(x) = \sum_n (a_n \cdot S_0(x - n) + b_n \cdot S_1(x - n)) \quad (11)$$

のように展開される。ここで、 a_n および b_n は未知変数であり、それぞれ $x = n$ における波動関数の値と微分に対応している。これらの値はコーン・シャム方程式を満たすように決定される（3.3 章を参照）。

三次元問題の場合には、一次元の基底関数の積として以下のように与えられる：

$$\left\{ S_i \left(\frac{x - n_x}{h} \right) \cdot S_j \left(\frac{y - n_y}{h} \right) \cdot S_k \left(\frac{z - n_z}{h} \right) \right\}, \quad i, j, k = 0, 1 \quad (12)$$

ただし、各節点は間隔 h の一様な三次元格子にあり、それぞれの座標を (n_x, n_y, n_z) で表すものとする。これは一辺が h の立方体要素に分割していることに相当する。この時、各節点には 8 個の基底関数が割り当てられ、各基底関数は隣接する節点に属する基底関数のみと重なりを持つことになる。この結果、基底関数間の行列は行ごとに 216 個の非零要素を持つ疎行列となる。この性質は後述の曲線座標系を使用した場合にも保たれる。計算の精度を上げる場合には格子間隔 h を小さくして解像度を上げてゆけば良い。

また、実際に波動関数から E_{total} を計算する場合にはいくつかの項で数値積分を行う必要があるため [21]、多項式向けによく使われるガウス・ルジャンドル積分法 [36] を用いる。三次の基底関数の場合には通常五次の積分点を用いれば十分な精度が得られる [21]。

3.2 適合曲線座標系

2 章で説明したように、電子は原子からのポテンシャルを受けて運動している訳であるが、ポテンシャルの形から想像できるように電子の波動関数は原子付近では比較的激しく変化し、原子から離れるに従って速やかに減衰する。従って、なるべく少数の自由度で高精度の解を得るためには何らかの方法で原子付近の解像度を上げることが望ましい。このためには色々な方法があるが、良く知られたものとしては以下の三つが挙げられる [11]：

- p 法（要素高次化法）：要素自体は固定し、その代わり誤差が大きい要素から優先的に基底関数の次数を上げてゆくことで精度を上げる。
- h 法（要素細分化法）：メッシュ及び基底関数の次数は一定にし、誤差が大きい要素を分割してゆくことで精度を上げる。
- r 法（節点移動法）：基底関数の次数は要素によらず一定にし、誤差が大きい要素が相対的に小さくなるように要素の形状を連続的に変形することで精度を上げる。

初期の実装においては、p 法に従って原子付近の要素における基底関数の数を増やすか [17]、あるいは h 法のように原子付近の要素を細分化する [18, 19] という方法が提案されてきた。しかし、これらの方法で局所的に自由度を増やした場合、原子がよほど一様に分布していない限り、並列計算機上での負荷分散が悪化し、通信量も増大することが予想される。また、例えば各原子の位置が時間と共に変化する場合を考えると、細分化する領域をどこかで不連続に変更しなくてはならない。そうすると E_{total} が R_α の不連続な関数になってしまい、非常に都合が悪い。一方、r 法の場合には常に連続的に変化するため、

こういった問題に遭遇することなく原子付近の解像度を上げることが出来る。また 3.5 章で説明するように、並列計算機上での性能を損なうこともない。従って、以下では r 法を中心として説明を進める。

ここで r 法を実装する上で必要となる座標変換の方法について紹介する。このような工夫を加えることによって強く局在した軌道を持つ元素（例えば酸素やフッ素）を扱う際に基底関数の数を一桁程度減らす事ができる。我々が使用している適合曲線座標系は平面波基底と共に初めて電子状態計算に適用され [37]、次いで差分法 [38]、有限要素法 [20, 21] へと拡張された。座標変換の式としては原子座標 R_α の関数として陽に与えられる形を使用している。具体的に説明すると、ユークリッド座標を r 、曲線座標を ξ で表した時、座標変換の式は

$$\xi = r + \sum_{\alpha} (r - R_{\alpha}) g_{\alpha}(|r - R_{\alpha}|) \quad (13)$$

で与えられる。ここで、 $g_{\alpha}(r)$ は以下のような性質を持つ関数である：

1. 任意の $r \geq 0$ に対して $g_{\alpha}(r) \geq 0$.
2. 原点付近で最大値を取り、 $r \rightarrow \infty$ で速やかに減衰する
3. 少なくとも三次微分まで計算可能

このような条件を満たす関数形は色々考えられるが、実用上は少数のパラメータで指定できる形が便利である。例えば、二個のパラメータで決まる場合の形としては

$$g_{\alpha}(r) = A_{\alpha} \exp\left(-\left(\frac{r}{a_{\alpha}}\right)^2\right), \quad (14)$$

のような式が考えられる（実際にはもう少し複雑な式を採用している）。この場合には、原子 α の位置における解像度がおよそ $(1 + A_{\alpha})$ 倍に拡大され、 a_{α} が拡大する範囲を決定する。適切なパラメータの値は元素ごとに大体決まっており、例えば酸素原子のように強く局在した電子状態を持つ場合は A_{α} を大きく取る必要があるが（1程度）、アルミニウム原子のように広がった電子状態の場合には小さい値（もしくはゼロ）で十分である。

実際の計算は全て ξ 空間における一様なメッシュ上で行うため、原子が動く度に各節点（及び積分点）を r 空間に変換する必要があるが、ニュートン・ラフソン法 [36] で 2 ~ 3 回反復すれば十分な精度で式 (13) を逆変換し、 r 空間での座標を求めることができるため、特に問題はない。ただしあまり極端に解像度を上げると解が存在しなくなるため、注意が必要である。例として、窒素分子が大きな周期セルの中央で振動・回転している場合に、 ξ 空間の一様なメッシュを r 空間へ射影した結果を図 4 に示す。価電子密度（図 5）と見比べると、波動関数が最も激しく変化する原子核の近傍でほぼ等方的に解像度が拡大されていることが分かる。また、原子が動くときそれにつれてメッシュも変化するため、 F_{α} を計算する際にはこの影響も考慮に入れている。これにより、分子動力学計算を行った場合でも系の全エネルギーが高い精度で保存される [21]。

また、曲線座標系を使用した場合としない場合について窒素分子の全エネルギーを原子間距離の関数として計算したものを図 6 に示す。これから分かるように、適切な座標変換を行うことによって一様なメッシュを使用した場合と比べて精度が大幅に向上する。

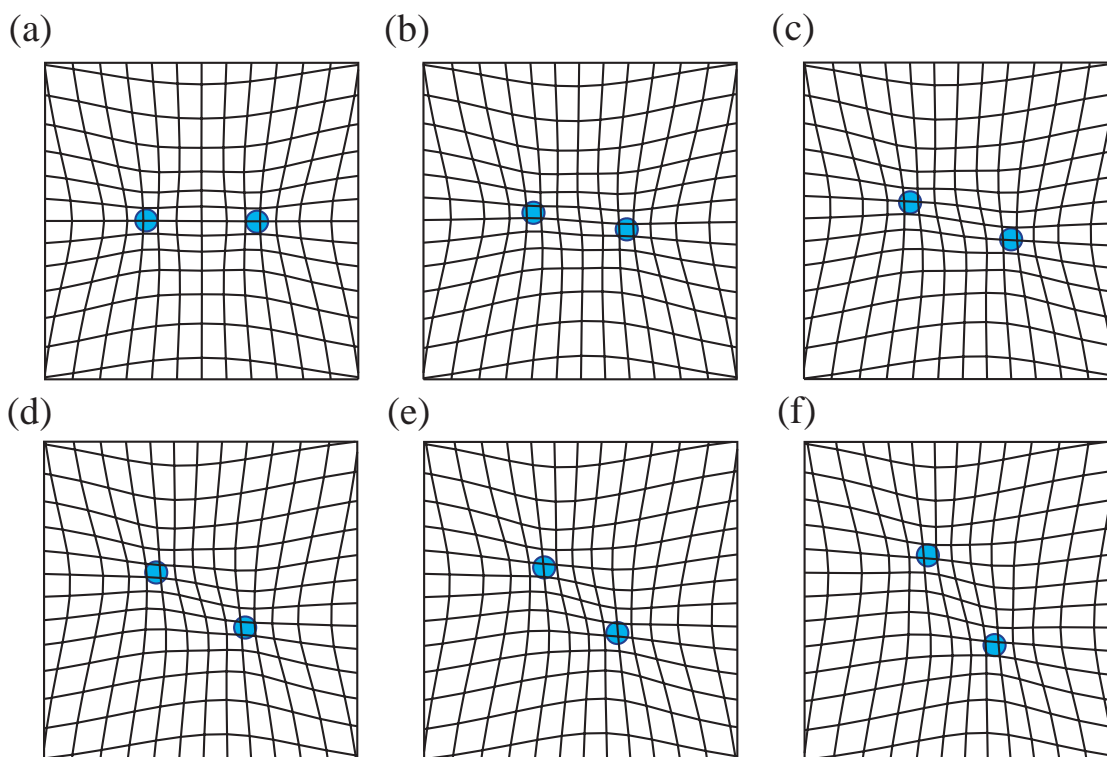


図 4: 窒素分子 (N_2) が振動・回転している場合の曲線座標系の時間変化を示す (ユニットセルの中心付近のみ表示)。

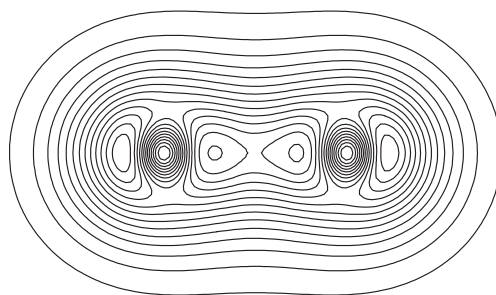


図 5: 窒素分子の価電子密度

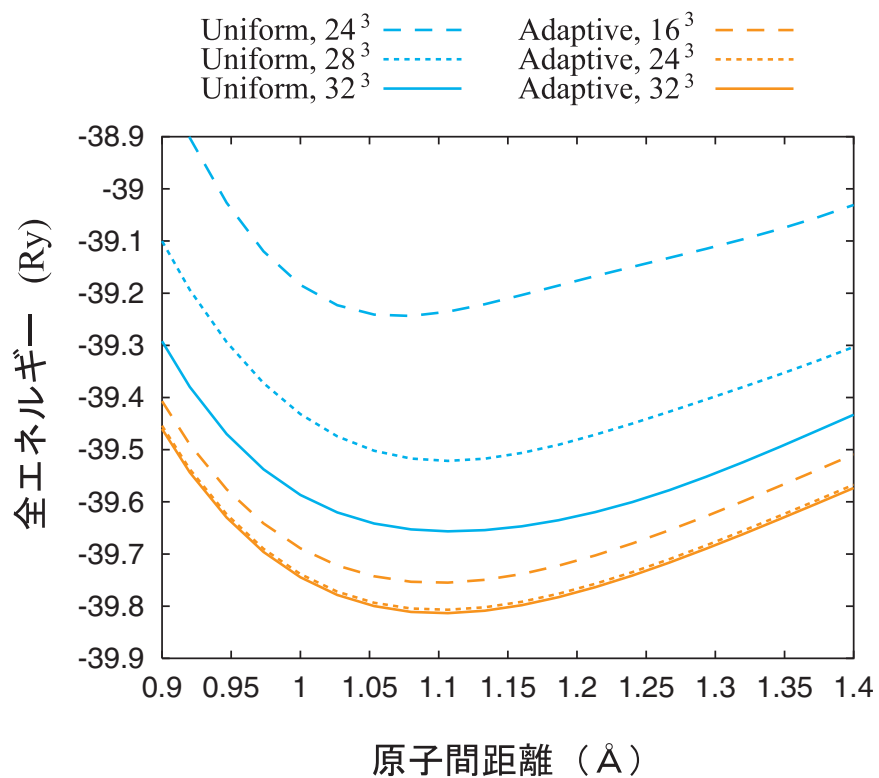


図 6: 一辺が 12 Å のスーパーセル中に置いた窒素分子のエネルギーを原子間距離の関数として計算したもの。Uniform: 一様なメッシュの場合、Adaptive: 適合曲線座標系を使用して各窒素原子の付近で 2 倍強程度解像度を上げた場合。結合長の実験値は 1.10 Å 程度であり、計算値と良く一致している。

3.3 電子状態計算の方法

ここでは、与えられた R_α に対して電子状態計算を行い、基底状態の波動関数及び E_{total} を得る方法について具体的に説明する。まず、波動関数は全て ξ 空間における一様なメッシュ上の各節点に属するエルミート補間式 (式 (12)) で展開される。すなわち、 $\psi_1, \psi_2, \dots, \psi_M$ は式 (11) を三次元に拡張した形で近似される。このとき、コーン・シャム方程式 (式 (4)) は次の式のように離散化される：

$$A \vec{\psi}_i = \epsilon_i S \vec{\psi}_i, \quad i = 1, \dots, M \quad (15)$$

ここで A は H_{KS} を基底関数で挟んで積分した行列であり、 S は基底関数間の重なり行列である。また、 $\vec{\psi}_i$ は波動関数を展開した際の係数をベクトルとして並べたものである。通常、有限要素法の基底関数は非直交であるため、離散化されたコーン・シャム方程式は一般化固有値問題になる。必要な固有状態の数 M は大体 10^2 から 10^3 程度である。また、 A 及び S は非常に規則性の高い対称な疎行列であり、その次元は M の $10^2 \sim 10^3$ 倍程度であるため、系によっては 10^6 を超える巨大な行列になることもある。更に、波動関数が増えると H_{KS} 自体も変更されるため、注意を要する。式 (15) を解く方法は色々提案されており、十分な精度で計算すればどの方法でも同じ結果が得られるが、 A や S の性質を活用する為には何らかの反復法を使うことが望ましい。

我々は固有値問題を直接解く代わりに、正規直交条件

$${}^t \vec{\psi}_i \cdot S \cdot \psi_j = \delta_{ij}$$

の下で

$$E = \sum_{i=1}^M {}^t \vec{\psi}_i \cdot A \cdot \psi_i \quad (16)$$

を最小にするような $\vec{\psi}_i$ を求めるという一種の最適化問題に変換し、これを準ニュートン法 [16, 39] で反復して解いている。FEMTECK では特に limited memory BFGS 法と呼ばれる準ニュートン法の省メモリ版 [40, 41] を実装し、大規模系でも効率良く扱うことができる。

一方、電子状態計算のコストを抑えるためには最適化の方法だけでなく、波動関数の良い初期値を用意することも非常に重要である。分子動力学を行う場合を考えると、通常各原子の位置は時間と共に滑らかに変化する。従って、過去 2 ~ 3 ステップの原子配置における波動関数の値から外挿することにより、現在の原子配置に対する波動関数の近似値を求める方法が非常に効果的である [42]。一番簡単な場合の外挿式を示すと、

$$\psi_n^{\text{ini}} = \psi_{n-1} + (\psi_{n-1} - \psi_{n-2}) = 2\psi_{n-1} - \psi_{n-2} \quad (17)$$

のようになる。ただし、 ψ_{n-1}, ψ_{n-2} はそれぞれ 1, 2 ステップ前の波動関数の (収束した) 値であり、 ψ_n^{ini} が最新の原子配置に対する波動関数の初期値である。このような外挿を行った場合、 R_α が変わるたびに大体 10 ~ 15 回程度反復すれば十分な精度を持った波動関数が得られる。

なお、電子状態計算を行うのに必要な計算量は原子数 N と共に

$$c_0 + c_1N + c_2N^2 + c_3N^3$$

のように増加する。 c_0, \dots, c_3 の値は実装方法に依存するが、全て正である。 N が数百の場合は $O(N)$ 及び $O(N^2)$ の項が最も計算時間を消費するが、1000 原子を超えるような大規模系では $O(N^3)$ の項が計算時間の大半を占める。また、必要なメモリ量は $O(N^2)$ のように増加する。計算時間ほど急激に増加する訳ではないが、メモリ量が足りないとそもそも動作させること自体できないため、こちらも深刻な問題である。

ところで、シミュレーションを行う系がエネルギー・ギャップを持つような場合には、精度を損ねることなく電子状態を強く局在させることが可能であることが知られている。このような性質を生かして計算量・メモリが共に $O(N)$ で済むような（近似的な）電子状態計算法が最近数多く提案されている [14, 15]。これらのオーダー N 法と呼ばれる手法には色々なバリエーションがあるが、大部分は実空間における電子状態の局在性を利用しているため、空間全体に広がった平面波基底よりも有限要素法や差分法のような実空間基底の方が適している。オーダー N 法は今まで安定性に問題があるという理由で広く使われることはなかったが、我々は最近この問題を回避できることを発見し、FEMTECK 中に実装した [43]。更にこの方法の信頼性を実証するため、エタノール₁₂₅ 分子（1125 原子）から成る大規模な溶液系で第一原理分子動力学計算を行った（図 7） [44]。十分な平衡化を行った後、300 K で 10 ピコ秒程度のシミュレーションを行い、動径分布関数や振動スペクトル等に関して十分な精度で実験値を再現することができた。また、水素結合の統計データ等、実験からは求めにくい物理量についても計算した。この系のシミュレーションでは、オーダー N 法を使用することによって従来の方法と比べて 5 倍近い高速化を実現することができた。

3.4 ポアソン方程式の解法

系の全エネルギー（式 (1)）の第二項、すなわち

$$\frac{1}{2} \int \frac{n(\mathbf{r})n(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r}d\mathbf{r}'$$

は電子間の相互作用を表しているが、これを計算するためには電子密度 $n(\mathbf{r})$ が変化する度にハートリー・ポテンシャル

$$V_H(\mathbf{r}) = \int \frac{n(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r}'$$

を求める必要がある。周期境界条件の場合には、この式をそのまま計算するよりもポアソン方程式

$$\nabla^2 V_H(\mathbf{r}) = -4\pi(n(\mathbf{r}) - \bar{n})$$

の解として求める方が都合が良い。ただし \bar{n} は電子密度の平均値である。実際には V_H を ξ 空間で基底関数展開することによって上記のポアソン方程式を離散化し、一次方程式の

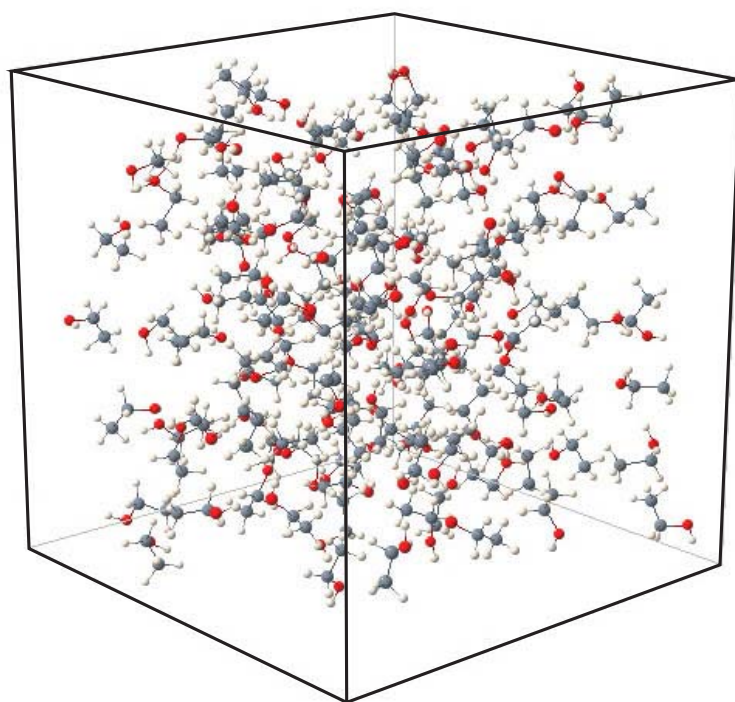


図 7: エタノール 125 分子から成るスーパーセル

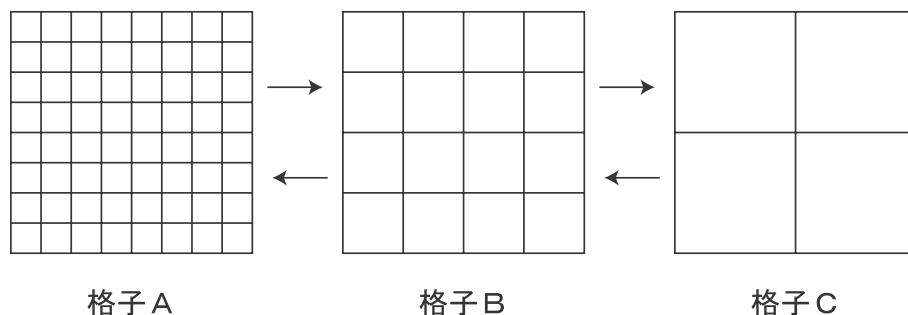


図 8: 多重格子法で格子 A 上での近似解を改善したい場合、残差を A B C B A の順に移動させ、それぞれの格子上で緩和法を適用する。

形にして解く。これを

$$Lv = n \quad (18)$$

と書くと、 L は ∇^2 に対応する行列であり、行ごとに 216 個の非零要素を持つ疎行列になる。系が小さい時はコレスキー分解のような直接法によってこの方程式を解くことが可能であるが、系が大きくなると行列の次元が非常に大きくなるために何らかの反復法に頼らざるをえない。標準的な反復法としては共役勾配法が良く知られているが、反復回数が系のサイズとともに増加するという難点がある。これは、系が大きくなるにつれて残差 ($= n - Lv$) の長波長成分の減衰がどんどん遅くなっていくことに起因している。このような場合、最も有力な解法は多重格子法と呼ばれる方法であり、系の大きさによらず一定の反復回数で同じ精度の解を与えることが知られている [45]。多重格子法のポイントは、解像度の異なる複数の格子を用意し、それぞれの格子毎に残差を減少させるような緩和法を適用することである (図 8)。これによって長波長から短波長の残差まで偏りなく減少させることができるため、系が大きくなっても一定の収束率を保つことが可能である [45]。

今の所、各格子上での緩和法としては、ベクトル化及び並列化の容易な最小二乗多項式の種類 [46] を採用し、異なる格子間の移動については V サイクルを使用している。また、主に並列化との兼ねいで、最も粗い格子上でも厳密に解くことはせずに、緩和法を数回適用している。なお、多重格子法は単独で用いても良いが、共役勾配法などの前処理として使用した方が、より効果を発揮できることが多い [21, 47]。例として図 9 にエタノール 125 分子の系でポアソン方程式の解の収束性を比較した結果を示す。共役勾配法の前処理として多重格子法を使用すると、対角行列の場合と比べて 3 倍以上の高速化が実現できることが分かる。系が大きくなるに連れてこれらの差は開くため、多重格子法は大規模計算には必須の技法である。

3.5 並列化の方法

ある程度大規模な第一原理計算を行う場合には、メモリ使用量及び計算コストの面から並列計算機を活用することが不可欠であるが、ここまで説明してきたアルゴリズムを並列

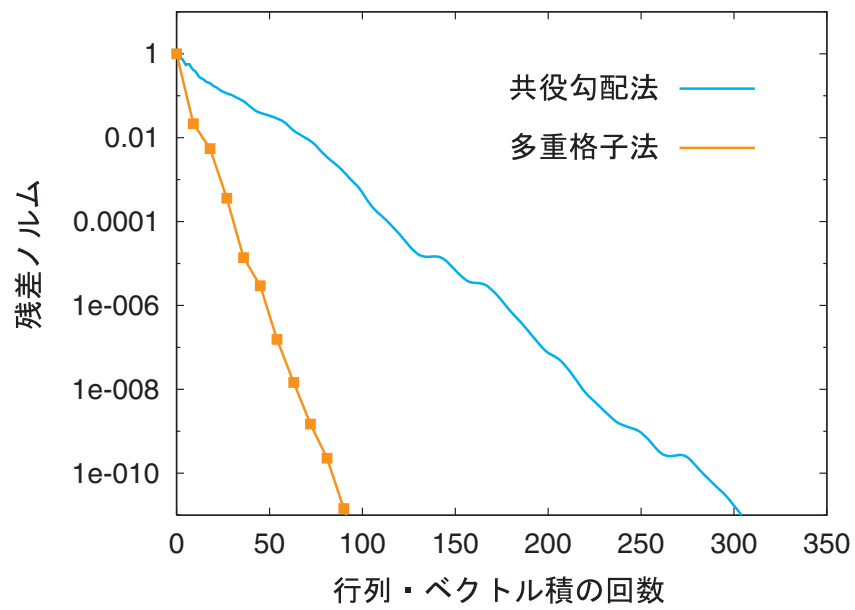


図 9: ポアソン方程式を通常の共役勾配法（前処理 = 対角行列）と多重格子法で解いた場合の収束性を示す。一反復当たりのコストが異なるため、行列・ベクトル積の回数を横軸としている。

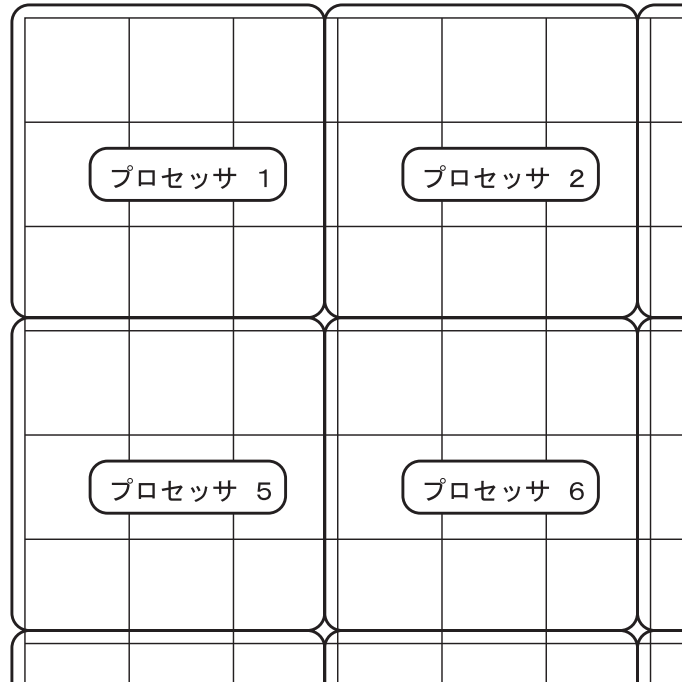


図 10: 並列化のイメージ図

化することは極めて容易である。即ち、我々の実装は r 法に基づいているため、曲線座標系 (ξ 空間) における一様なメッシュを使用している。これを同じサイズの領域に三次元分割して各プロセッサに割り振るだけで十分である (図 10)。この時各プロセッサには同数の要素が割り当てられるため、負荷分散はほぼ完全に等しくなる。また、第一原理計算の場合には古典分子動力学法と異なり、各プロセッサが全ての原子の座標を持つことが普通である。

電子状態計算を行う際に必要な通信のタイプは主として (i) 波動関数等の境界における値を隣接プロセッサ間で共有する為のローカルな通信、及び (ii) 電子数 M 程度の次元を持つ行列のグローバルな通信、の二種類に分類される。言い換えれば、最も情報量の多い波動関数の通信をローカルで済ませ、通信負荷の高いグローバルな通信はほぼ必要最小限に抑えることができる。従って、プロセッサ間の通信によるオーバーヘッドは比較的小さい。こういった事情は有限要素法に限らず差分法でも同様であり [48, 49]、実空間法に共通の利点の一つとすることができる。

FEMTECK は 1997 年頃に、本センターに設置された直後の SR-2201 上で並列化を行った。一時期、SR-8000 向けに最適化していた際には共有メモリを活用するためにマルチスレッド化を施していたが、(特に PC クラスタ上では) 色々な制約が付くため、現在は MPI 通信のみを使用して並列化している。上述の領域分割に基づく並列化の効率を HA8000 ク

ラスタシステム (東京大学情報基盤センター)¹ 上で測定した結果については6章で紹介する。

3.6 分子動力学

2章で触れたように、我々の計算では原子が移動する度に電子状態計算を行い、各原子に働く力を求めている。これは Born-Oppenheimer molecular dynamics (BOMD) 法と呼ばれる手法であり、非常に多くの系 (特にエネルギー・ギャップを持つような系) において正確な記述を与える [3]。ここでは原子核の量子性は考慮せず、全て古典的に振舞うものとする。そうすると、各原子 α の位置 R_α 及び速度 V_α はニュートン方程式

$$\frac{dR_\alpha}{dt} = V_\alpha \quad (19)$$

$$\frac{dV_\alpha}{dt} = \frac{F_\alpha}{m_\alpha} \quad (20)$$

に従って変化する。実際の計算ではこれらの式を離散化した速度ベルレ法 [50, 51]

$$R_\alpha(t + \Delta t) = R_\alpha(t) + \Delta t V_\alpha(t) + \frac{(\Delta t)^2}{2m_\alpha} F_\alpha(t) \quad (21)$$

$$V_\alpha(t + \Delta t) = V_\alpha(t) + \frac{\Delta t}{2m_\alpha} (F_\alpha(t + \Delta t) + F_\alpha(t)) \quad (22)$$

を用いて数値積分することで時間発展を追うことができる。ここで Δt は時間刻みを表す。なお、式 (21), (22) はミクロカノニカル集合に対応したものであるが、原子系の温度をコントロールする必要がある場合には Berendsen 熱浴 [52] を加えている。また、シミュレーションの初期段階などでは分子内の共有結合長を一定に保つような拘束条件の下で時間発展を追うことも多い。このような場合には RATTLE と呼ばれるアルゴリズムを実装することで対応している [53]。これにより初期配置が不自然であったり系が高温である等の理由で共有結合が切れることを防ぐ事ができ、更に時間刻みを2~3倍大きく取ることができる。

4 他の基底関数との比較

ここまで有限要素法を用いて波動関数を展開し、電子状態を計算する方法について説明してきたが、他にも様々な基底関数があり、それぞれ得手不得手がある。本章ではこのような視点から、代表的な基底関数について長所・短所を手短に考察する。

まず、いわゆる原子基底²の長所としては、以下のような点が挙げられる：

- 原子数の2~3倍の基底関数でも半定量的な計算ができる。従って、机上のパソコンでもある程度のサイズの系の第一原理計算を手軽に行うことができる。

¹ <http://www.cc.u-tokyo.ac.jp/ha8000/>

² ガウス型、スレーター型、あるいは数値型を指すことが多い。

- 豊富な機能を持つ有償・無償のプログラムが多数開発されており、専門家でなくとも容易に計算を開始することが出来る³。

一方、原子基底には次のような問題点もある：

- (特に基底関数が不十分な場合、) 基底関数重なり誤差の影響で原子間相互作用が過大評価され易い。
- 多原子系の場合、基底関数の選び方は必ずしも自明ではない。従って、いくら基底関数の数を増やしても波動関数が完全に収束していることを保証することは容易ではない。また、数を増やし過ぎると線形従属に近くなり、数値的な安定性が落ちる。

次に平面波基底について考察すると、まず長所としては

- 正規直交基底であり、また微分・積分が容易なため、 E_{total} のうち ϵ_{xc} を含む項以外は全て解析的に計算することができる。
- 原子基底と同様、完成度の高いプログラムが多い⁴。

といった点があり、短所としては

- フーリエ変換を頻繁に行う必要があり、並列化を行う際に大きな負担となる。
- 本質的に広がった基底関数であり、オーダー N 法やハイブリッド法のように局在性が重要なアルゴリズムへの適用は容易ではない。

といった点が挙げられる。一方、差分法は有限要素法と同様に実空間で定義されているため、高い並列性やオーダー N 法・ハイブリッド法への適合性など、多くの共通する特長を持つが、

- 有限要素法とは異なり、正規直交系を形成している。
- 差分法では波動関数を基底関数で展開する代わりに E_{total} 中の微分演算子を差分近似している。従って変分原理を満たさず、その結果 E_{total} が真の値より低くなり得る。

という点で異なっている。

なお、原子基底以外のもの(平面波、差分法、有限要素法)は機械的に計算精度を上げることが可能であり、系統的な基底関数と呼ばれる。例えば有限要素法であれば、実空間メッシュの間隔を小さくしていけば常に収束した波動関数が得られる。これらの基底関数の場合、原子数が一定であれば、計算量はほぼ基底関数の数に比例する。一方、原子基底の場合には基底関数を増やしてゆくと計算時間は急激に増加するため、ある程度精度の高い計算を行う場合には系統的な基底関数の方が高速であると考えられる。

以上、代表的な基底関数を幾つか紹介して来たが、実際には必要な精度・系のサイズ・使用する計算機などに応じて最適なものを選ぶのが現実的である。例えば膨大な数の原子

³ Gaussian, Q-Chem, GAMESS, SIESTA, OpenMX 等が有名である。

⁴ 代表的なものとしては CPMD, ABINIT, VASP, CASTEP 等がある。

配置（あるいは原子組成）から有望なものを選び出すような場合には、まず最小限の原子基底を用いてある程度絞り込みを行い、残ったものに対して系統的な基底関数を使用して高精度計算を行う、といった使い分けが考えられる。

5 応用例

ここでは、FEMTECK を用いた第一原理計算の応用例をいくつか紹介する。

常温における水素結合性の分子性液体は非常に複雑な構造を持ち、頻繁に水素結合の生成・消滅を繰り返している。このような系を定量的に理解するため、メタノール (CH_3OH) [54], ホルムアミド (HCONH_2) [55], エタノール ($\text{C}_2\text{H}_5\text{OH}$) [44] 等の分子性液体について数ピコ～数十ピコ秒程度の第一原理分子動力学計算を行い、構造（動径分布関数、水素結合数）や動的性質（拡散係数、振動数分布）などの物性を計算した。特に、実験と直接比べることが可能な量については非常に良い一致を示すことを確認している。

また最近、我々のグループでは燃料電池に関連する系のシミュレーションに力を注いでおり、濃リン酸 (H_3PO_4) [56], 硫酸水溶液 ($\text{H}_2\text{SO}_4 / \text{H}_2\text{O}$) [57]、ナフィオン/水界面 [58] 等の系について研究を進めている。これらの系では頻繁にプロトン移動が起きるため、古典ポテンシャルで定量的に記述することは非常に困難であり、第一原理計算で扱う意義が大きい。また、ベリー位相を利用して外部電場を加えることでプロトン伝導を促進することも行っている [59, 60]。

6 ベンチマーク計算

最後に、本センターに設置された HA8000 上における FEMTECK のベンチマーク結果について紹介したい。以下のような3種類の規模の異なる系についてそれぞれ第一原理分子動力学計算を行い、1ステップ進めるのに要する時間を測定した。これらの計算では十分な数の基底関数を使用し、またオーダーN法は使用していない。

| 名称 | 系 | 原子数 | 波動関数の数 |
|--------|--------|------|--------|
| システム A | ダイヤモンド | 64 | 128 |
| システム B | 水 | 375 | 500 |
| システム C | 水 | 1920 | 2560 |

システム A を使って基本的な部分を詰めるため、まず1ノード中の1コアのみを使用して、単体性能を測定した。コンパイラ（言語は Fortran）及び数値ライブラリ（BLAS/LAPACK）としては幾つか選択肢があるが、HA8000 上で問題無く動作した組み合わせは「日立コンパイラ + netlib」及び「インテルコンパイラ + ACML(MKL)」であった。コンパイルの際にはそれぞれ以下のようなオプションを使用した。

```
% f90 -Oss -noparallel *.F -llapack_sc -lblas_sc  
% ifort -O3 -static *.F -lacml
```

また、ACML ライブラリは

<http://developer.amd.com/cpu/Libraries/acml/downloads/Pages/default.aspx>

よりダウンロードした Intel Fortran 用の 64 ビット版 (バージョン 4.1.0) を使用した。比較対象として、普段使用している AIST スーパークラスターの P32 システム (Dual Opteron 2GHz) を 1 コアのみ使用して測定した結果を用いた。この環境では PGI Fortran 90 及び ACML ライブラリを使用した。

これらの条件の下で、システム A で実行時間を測定した結果を以下に示す：

| コンパイラ / 数値ライブラリ | システム | 実行時間 (秒) |
|-----------------|--------|----------|
| 日立+netlib | HA8000 | 112.7 |
| インテル+ACML | HA8000 | 87.1 |
| インテル+MKL | HA8000 | 87.8 |
| PGI+ACML | P32 | 112.9 |

まず分かるように、HA8000 上でのパフォーマンスは「インテル+ACML/MKL」の方がかなり速い。ただし、内訳を良く見ると、数値ライブラリを呼び出している部分で差が付いており、それ以外の自作ルーチンに対する性能は大差ない。従って、(リンクが可能であれば)「日立コンパイラ+ ACML/MKL」の組み合わせの方が速くなる可能性はある。また Goto BLAS 等も有力な選択肢であるが、LAPACK や ScaLAPACK を使う際には一手間ありそうだったため、今回は見送った。なお、「インテルコンパイラ + MKL」はやや挙動不審な面があり、また ACML と比べて特に性能上のメリットも見られなかったため、以降のベンチマークは「インテルコンパイラ + ACML」に限定する。

一方、HA8000 上で ACML を使用した場合のパフォーマンスは P32 と比べておよそ 1.3 倍程度高速化されている。HA8000 及び P32 システムの 1 コア当たりの理論ピーク性能はそれぞれ 9.2GFlops と 4GFlops であるため、HA8000 の方が 2.3 倍高速なはずであり、実測値はこれよりもかなり差が小さいことになる。この理由を知るために profile を調べた所、数値ライブラリに依存する部分は大体理論性能に比例したパフォーマンスが出ているが、それ以外の部分、特にイレギュラーな演算を行っているルーチンでほぼ同程度の時間を消費していることが分かった。これは恐らく、P32 の場合には 1 クロック当たり 2 演算を行うことで理論ピーク性能を達成できるが、HA8000 の場合には同時に 4 演算を行う必要があり、規則性が低い演算の場合には性能を出すことがより困難であるためと考えられる。

次に並列性能について考察するため、16 並列のジョブを色々な条件の下で実行し、実行時間を測定した。並列化はフラット MPI で行い、OpenMP は一切使用していない。また数値ライブラリも全てシングル・スレッドモードで使用した。まず 1 ノード 16 コアを使用し、デフォルトの設定で測定した場合と、プロセスが連続したコアに割り振られるように自分で NUMA を指定した場合を比べると、以下のようになった：

| 計算条件 | 実行時間 (秒) |
|-----------|----------|
| NUMA 指定なし | 7.85 |
| NUMA 指定あり | 7.42 |

なお、NUMA は以下のようなスクリプトに従って指定した：

```
#!/bin/bash
MYRANK=$MXMPI_ID
TEMP=$(expr $MYRANK % 16)
NODE=$(expr $TEMP / 4)
/usr/bin/numactl --membind $NODE --cpunodebind $NODE $@
```

FEMTECK の場合には、隣接するプロセス間ほど大規模な通信を行う仕様になっているため、このような指定が有効であると考えられる。

ただし、NUMA を指定した場合であっても、1 コアの場合と比べたパフォーマンスは 12 倍弱であり、並列性能としてはやや不満足である。この点を明らかにするため、(かなり贅沢な使い方であるが) 1 ノード当たり 1 コアのみを使用し、16 ノード 16 コアを使用して全く同じジョブを流して実行時間を測定した。この時の実行時間は 5.70 秒であり、15.3 倍の高速化が得られた。この結果から、1 ノード 16 コアを使用した場合には、

- 4 つのコア毎に一つのメモリを共有しているため、バンド幅が不足する
- ノード内の通信が干渉する

といった現象の影響が大きいと推測される。

とはいえ、トータルの性能ではやはり全てのコアを使用した方が速いのは確かであり、以下のベンチマークでは特に断らない限りノード毎に 16 コア全てを使用し、NUMA を指定した上で測定するものとする。このような条件で、システム B を用いて 16 コア (1 ノード) から 512 コア (32 ノード) まで並列性能を測定した：

| コア数 | 実行時間 (秒) | パフォーマンス |
|---------------|----------|---------|
| 16 | 230.2 | 1 |
| 32 | 124.0 | 1.86 |
| 64 | 64.2 | 3.59 |
| 128 | 34.5 | 6.66 |
| 128 (NUMA なし) | 47.1 | 4.89 |
| 256 | 20.0 | 11.5 |
| 512 | 12.6 | 18.3 |

大体この程度の系が今までの計算資源で十分な統計精度で扱うことができる限界に近いサイズであったが、HA8000 の大規模キューを使用すれば 2 ~ 3 日で 10^4 ステップ程度のデータを集めることができるようになった。また、試しに NUMA を指定せずに測定した所、かなり大きな差が付いた。並列度が高い場合の方が NUMA の影響が大きいようである。

最後に、月に一度の大規模ジョブ実行サービスを利用して、システム C で最大級のベンチマークを取った。なお、メモリの制約上、動作させるためには最低でも 512 コアが必要であった。また、1024 コアの場合は時間の都合でデータを取ることが出来なかった。ベンチマーク結果は以下ようになった：

| コア数 | 実行時間 (秒) | パフォーマンス |
|------|----------|---------|
| 512 | 559.5 | 1 |
| 2048 | 208.9 | 2.68 |
| 4096 | 161.7 | 3.46 |

基本的には系が大きい方が並列性能が出易いはずであるが、その割にはやや不本意な性能であった。更に profile を見ると、以下のような点が問題であることが分かった：

- 細かい通信を頻繁に行うルーチンのコストが大きい
- 一部、並列化していないルーチンがあり、コア数が多い場合には（相対的に）かなり時間を消費している
- 数値ライブラリに依存する部分は 512 → 4096 コアの際に行列の次元が減る分、実効 Flops が 10% 強低下する

これらの点に配慮し、重複計算している部分を並列化する等の改善を行うことにより、512 → 4096 コアの際に 5 倍程度の高速化は達成できると見積もっている。現状では HA8000 上でのプロダクション・ランで 4096 コアを使うことはあまり現実的ではないが、次世代スパコンでは通常の粒度になる可能性もあり、その場合にはかなり細かい部分まで漏れなく並列化することが重要になりそうである。

参考文献

- [1] 小出 昭一郎 著：「量子力学 I・II」 [改訂版] (裳華房, 1990).
- [2] 金森 順次郎、米沢 富美子、川村 清、寺倉 清之：「固体 - 構造と物性」(岩波書店, 2001).
- [3] R. M. Martin: *Electronic Structure: Basic Theory and Practical Methods* (Cambridge University Press, New York, 2004).
- [4] A. E. Mattsson *et al.*, *Modelling Simul. Mater. Sci. Eng.* **13** (2005) R1.
- [5] M. J. Gillan *et al.*, *Rep. Prog. Phys.* **69**, 2365 (2006).
- [6] S. Rauegi, F. L. Gervasio, and P. Carloni, *Phys. Stat. Sol. B* **243** (2006) 2500.
- [7] *Topics in Current Chemistry* **268** (2007) :「Atomistic Approaches in Modern Biology」特集号.
- [8] *固体物理* **39** (2004) 11月号: 「計算機ナノマテリアルデザイン」特集号.
- [9] A. Szabo, N. S. Ostlund : 「新しい量子化学 (上・下)」(東京大学出版会, 1987).

- [10] 矢川 元基、吉村 忍 著 : 「有限要素法」(培風館, 1991) .
- [11] 手塚 明、土田 英二 著 : 「アダプティブ有限要素法」(丸善, 2003) .
- [12] T. L. Beck: Rev. Mod. Phys. **72** (2000) 1041.
- [13] L. R. Ram-Mohan: *Finite Element and Boundary Element Applications in Quantum Mechanics* (Oxford University Press, New York, 2002).
- [14] S. Goedecker: Rev. Mod. Phys. **71** (1999) 1085.
- [15] G. E. Scuseria: J. Phys. Chem. A **103** (1999) 4782.
- [16] ナノシミュレーション技術ハンドブック委員会 編 : 「ナノシミュレーション技術ハンドブック」(共立出版, 2006).
- [17] S. R. White, J. W. Wilkins, and M. P. Teter, Phys. Rev. B **39** (1989) 5819.
- [18] J. Ackermann, B. Erdmann, and R. Roitzsch, J. Chem. Phys. **101** (1994) 7643.
- [19] E. Tsuchida and M. Tsukada, Phys. Rev. B **52** (1995) 5573.
- [20] E. Tsuchida and M. Tsukada, Phys. Rev. B **54** (1996) 7602.
- [21] E. Tsuchida and M. Tsukada, J. Phys. Soc. Jpn. **67** (1998) 3844.
- [22] S. Yamakawa and S. Hyodo, Phys. Rev. B **71** (2005) 035113.
- [23] Y. Shiihara, O. Kuwazuru, and N. Yoshikawa, J. Solid Mech. Mater. Eng. **2** (2008) 95.
- [24] 櫛田 慶幸、奥田 洋司 : 日本計算工学会論文集 No. 20040023 (2004).
- [25] J. E. Pask and P. A. Sterne: Modelling Simul. Mater. Sci. Eng. **13** (2005) R71.
- [26] T. Torsti *et al.*, Phys. Stat. Sol. B **243** (2006) 1016.
- [27] J. -L. Fattebert, R. D. Hornung, and A. M. Wissink, J. Comp. Phys. **223** (2007) 759.
- [28] 岡田 勲, 大澤 映二編 : 「分子シミュレーション入門」(海文堂, 1989).
- [29] P. Hohenberg and W. Kohn, Phys. Rev. **136** (1964) B864.
- [30] W. Kohn and L. J. Sham, Phys. Rev. **140** (1965) A1133.
- [31] R. Car and M. Parrinello, Phys. Rev. Lett. **55** (1985) 2471.
- [32] W. E. Pickett, Comp. Phys. Rep. **9** (1989) 115.

- [33] J. A. Appelbaum and D. R. Hamann, Phys. Rev. B **8** (1973) 1777.
- [34] S. Goedecker, M. Teter, and J. Hutter: Phys. Rev. B **54** (1996) 1703.
- [35] P. M. Prenter, *Splines and Variational Methods* (Wiley, New York, 1975).
- [36] W. H. Press 他 著、丹慶勝市他 訳、「ニューメリカルレシピ・イン・シー C 言語による数値計算のレシピ」(技術評論社, 1993) .
- [37] F. Gygi, Phys. Rev. B **48** (1993) 11692.
- [38] N. A. Modine, G. Zumbach, and E. Kaxiras, Phys. Rev. B **55** (1997) 10289.
- [39] 茨木俊秀、福島雅夫: 「FORTRAN77 最適化プログラミング」(岩波書店, 1991) .
- [40] D. C. Liu and J. Nocedal, Math. Prog. **45** (1989) 503.
- [41] E. Tsuchida, J. Phys. Soc. Jpn. **71** (2002) 197.
- [42] T. A. Arias, M. C. Payne and J. D. Joannopoulos: Phys. Rev. Lett. **69** (1992) 1077.
- [43] E. Tsuchida, J. Phys. Soc. Jpn. **76** (2007) 034708.
- [44] E. Tsuchida, J. Phys.: Condens. Matter **20** (2008) 294212.
- [45] 荒川 忠一 著 : 数値流体力学 (東京大学出版会, 1994).
- [46] Y. Saad, SIAM J. Sci. Stat. Comput. **6** (1985) 865.
- [47] O. Tatebe and Y. Oyanagi, in: Proceedings of Supercomputing '94 (IEEE Computer Society, 1994).
- [48] J. R. Chelikowsky, N. Troullier, K. Wu and Y. Saad, Phys. Rev. B **50** (1994) 11355.
- [49] E. L. Briggs, D. J. Sullivan, and J. Bernholc, Phys. Rev. B **54** (1996) 14362.
- [50] M. P. Allen and D. J. Tildesley: *Computer Simulation of Liquids* (Oxford Science Publications, Clarendon, Oxford, 1987).
- [51] 上田 顕 著 : 「コンピュータシミュレーション」(朝倉書店, 1990)
- [52] H. J. C. Berendsen *et al.*, J. Chem. Phys. **81** (1984) 3684.
- [53] H. C. Andersen: J. Comp. Phys. **52** (1983) 24.
- [54] E. Tsuchida, Y. Kanada, and M. Tsukada, Chem. Phys. Lett. **311** (1999) 236.
- [55] E. Tsuchida: J. Chem. Phys. **121** (2004) 4740.

- [56] E. Tsuchida: J. Phys. Soc. Jpn. **75** (2006) 054801.
- [57] Y. -K. Choe, E. Tsuchida, and T. Ikeshoji, J. Chem. Phys. **126** (2007) 154510.
- [58] Y. -K. Choe, E. Tsuchida, T. Ikeshoji, S. Yamakawa, and S. Hyodo, J. Phys. Chem. B **112** (2008) 11586.
- [59] P. Umari and A. Pasquarello, Phys. Rev. Lett. **89** (2002) 157602.
- [60] I. Souza, J. Íñiguez, and D. Vanderbilt, Phys. Rev. Lett. **89** (2002) 117602.