

実空間差分法による第一原理計算ソフトウェア：RSDFTと超並列計算

岩田 潤一*

A Real-Space Finite-Difference First-Principles Calculation Software : RSDFT and Massively-Parallel Computations

Jun-Ichi Iwata*

実空間差分法を用いた第一原理計算ソフト「RSDFT」の特徴と、「京」に向けて行ったチューニングおよびベンチマークテストの結果を紹介する。また「RSDFT」+「京」によって、ナノスケールの材料開発がどこまで可能かも議論する。

Key word: スーパーコンピューター、シミュレーション、ソフトウェア、実空間、差分法、密度汎関数、第一原理計算、並列化

1. はじめに

1.1. スーパーコンピューター

「コンピューターは驚くほど進化した」。これと似たようなセリフを誰もが一度は見たり、聞いたり、あるいは言ったりしたことがあるかもしれない。だがこの言葉を聞いて頭の中に思い浮かべる「コンピューターの進化」は人それぞれであろう。ちなみに私の頭に浮かんだものをフリーのイラスト素材の中から探してみたところ、図 1 が最も



図 1 コンピューターの進化

イメージに近かった。今現在というよりは、少しだけ未来という感じだろうか。昔観たアニメや映画の影響があるような気もする。

そんなコンピューターの親玉とも言うべき存在として「スーパーコンピューター」というものがある。またイメージの話をしよう。進化し過ぎたコンピューターによって逆に人間が支配されるようになり、そんな世界で反乱を起こし、次から次へと襲い来るコンピューター軍団の攻撃をかわしながら、やっとの思いでたどり着いた敵の本拠地。そこで主人公ジュンが見たものは、高さ 10 メートル、横幅 100 メートルはあろうかという超巨大なコンピューター。こいつが世界の全てに命令し、世界の全てを支配するラスボス・・・そいつがいま目の前に！

ジュン「行くぞ！みんな！」

みんな「おおおおーっ！！」

ジュン「これはクリリンの分だーっ！」

(ドカーンッ！)

しかし、この事態をあらかじめ予測していたスーパーコンピューターは、タイムマシンで過去に刺客を送り込み、ジュンの祖先を直接亡き者にすることによって未来におけるジュンの存在そのものを消し去ろうと企てる。そして送り込まれた刺

*アドバンスソフト株式会社 ナノシミュレーション
研究開発センター

R&D Center (Computational Materials Science),
AdvanceSoft Corporation

客「ターミネーター」は、素っ裸でハーレーにまたがり、ショットガンをぶっ放しながら溶鉱炉に飛び込むという、手が付けられない暴れん坊であったが、最終的にはカリフォルニアの州知事にまで登り詰めることになる・・・。

以上は、「おそらく世間一般の人々はそんな風に思っているに違いない」と筆者が勝手に思っているスーパーコンピューターのイメージである。全て妄想であるが、その巨大な見た目という点については、実際のスーパーコンピューターと上記のイメージはそれほどかけ離れていない。例えば神戸ポートアイランドに設置されているスーパーコンピューター「京」は、体育館のようなところにビッシリと、大きめの冷蔵庫のような筐体が864台並べられており、その一台一台はさらに24台のコンピューターで構成されているというのであるから、まさに「スーパー」の名に恥じない存在感である（図2）。

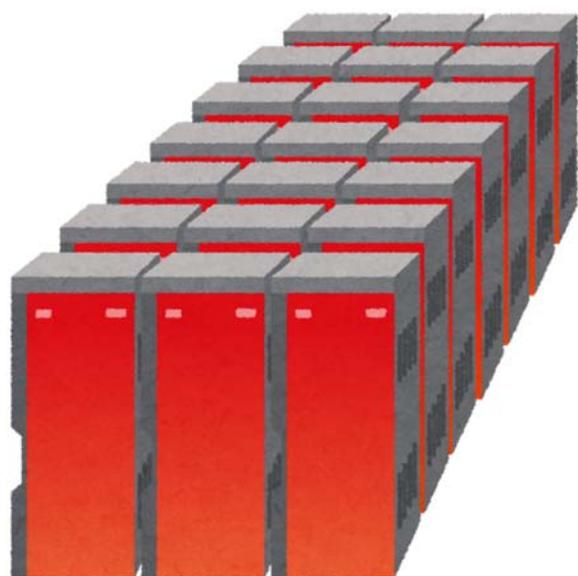


図2 ブラットと並んだスーパーコンピューターの計算ラック

さらに海外では、ブラットと並んだ筐体の側面を一枚のキャンバスに見立て、そこにデカデカと一匹のジャガーを描くという大胆なデザインが施されたものもあり、その様相はさながら、敵対するジャパニーズマフィアの組長と直接話を付けに来たジュンが、喉元に長ドスを突き付けられた和

室の襖絵のようである。ちなみに「京」が置かれている理化学研究所計算科学研究センターの最寄り駅の名前が、ある時期を境に「京コンピュータ前」に変わっており、スーパーコンピューターによる人間界の支配の始まりを垣間見せるような事態も起こっている。

1.2. スーパーコンピューター用ソフトウェア

また勝手なイメージだけで話をすると、かようにスーパーなコンピューターであるから、その業界もさぞかしスーパーで、例えば図1のような状況はもはや当たり前のように実現されており、最先端のテクノロジーを駆使して開発されたソフトウェアも普段目にする Windows やゲームなどとは比較にならない代物で、一たびそれを動かさうものなら森羅万象ありとあらゆることは人間の意のままとなり、それ故に神々の怒りを買ひ、大地は裂け、火を噴き、海は割れ・・・と思われることだろう。しかし現実はそうではない。スーパーコンピューターの世界は、本体の外観以外は驚くほど地味である。

地味さの原因の1つは、スーパーコンピューターの主な仕事が数値計算であるという点だろう。すなわち実行されるプログラムは、その大半が四則演算を行うものであり、やっていることは電卓とほとんど変わらない。当然、実行結果も数字が出てくるだけである。使用されるプログラム言語も、最近ではCやC++も増えてきたが、やはり今でもぶっちぎりの一位は Fortran である。これは、Formula の translation 以外、複雑なことは一切やらないという決意の表れであろう。計算結果をグラフにしたりアニメにしたりという段になって、ようやく図1に近い雰囲気が出てくるが、それはもはやスーパーコンピューターの仕事ではなく、進化したパソコンの仕事である。

地味さのもう1つの原因は、スーパーコンピューターを利用する人種にあると思われる。スーパーコンピューター用のプログラムを開発し、それを実際に使う人の多くは、物理、化学、工学といった、自然科学の研究者である。これらの人々は、興味の対象を記述する方程式を数値的に解く際

のスピードと、大規模なモデルを扱うためのメモリ容量以外のことにはほとんど関心がない。そのため、作られたプログラムは目的の計算を実行する以外の機能を持たないストイックな作りになっており、出てくる数字に科学的価値さえあれば使い勝手は二の次となり、そうして README すらない高速電卓が出来上がる。

あるプログラムを、開発した当事者以外が使えるようにするためには、それ相応の時間をかけてマニュアルやユーザーインターフェースを整備する必要がある。チュートリアルや講習会を行わなければならない場合もあるだろう。しかし、繰り返しになるが、スーパーコンピューター用のプログラムを開発しているのは、そのほとんどが自然科学の研究者である。そしてプログラムの普及のための活動は、そもそも面倒な上に、本業である「研究」の成果とは認められない風潮があるので、モチベーションを保つのが大変難しい。アカデミアにおいてスーパーコンピューター用に開発されたプログラムの中には当然、産業応用的にも非常に役立つと思われるものが数多く存在する。しかしながら、現状、それらは思ったほど産業界に普及していない。その理由は明らかであるが、永続的にソフトウェアのサポートを行い、さらには営業や宣伝活動まで含めて実施できるようところが現れない限り、この状況を改善するのは難しいように思われる。

本節で、スーパーコンピューターの世界はとにかく地味だと繰り返してきたが、それはよく言えばストイック、あるいはアスリート的な世界であるとも言える。スーパーコンピューターのハードウェアは、例えるなら相当に鍛え上げられた肉体のようなものである。しかしそれだけではボディービルダーにしかねない。その肉体を活かし、さまざまな競技種目の選手として活躍させるのはソフトウェアの仕事である。選手としての活躍後、雑誌やテレビで引っ張りだことなり、さらには政界への進出を果たしたりするようなケースが人間界ではしばしば見られる。しかしそれには選手としての能力とはまた別のもの、その人が持つ「魅力」や「華」といったものも大きく影響し

ていると思われる。そしてこの「華」の部分が、おそらく、現在のスーパーコンピューター用のソフトウェアには圧倒的に足りていない。すなわち、ただ計算の性能が素晴らしいというだけでなく、それが使いやすいこと、マニュアルやサポート、各種ツール等が充実していることなど、ソフトウェアとしての魅力が十分に備わっていることが、産業界で引っ張りだこになれるかどうかの重要なカギであると考えられる。

ボディービルダーからターミネーター、そして州知事へと登り詰めたコンピューター軍団の刺客は、必ずしも一流ではない、どちらかと言えば記録よりも記憶に残るタイプのターミネーターであった。しかし、ジュンの祖先暗殺の任務中に見せた珍プレーの数々は、逆に多くの人々に愛される切っ掛けとなり、その新たな魅力が、ターミネーター失業後、州知事として第二の人生を歩むことにつながるのである。

1.3. 物質科学と異分野交流

本業ではないが、必要に迫られて異分野の技術を導入しなければならなくなったとき、その分野がビジネスとして成立しているものであれば、外注に出すというのが最も手っ取り早い方法であろう。あるいは学術分野として確立しているものであれば、共同研究として話を進めるという方法もある。

ようやく本題に入ろう。筆者は長年、RSDFT という第一原理計算ソフトウェアの開発を行ってきた[1]。その過程で、並列計算への対応や、メモリアクセスの最適化、「京」の設計段階での RSDFT のチューニングなど、とても物理の人間だけでは対応しきれない開発を計算機科学の研究者との共同研究という形で進めてきた。これらの共同研究の成果は、2011 年度 ACM ゴードンベル賞受賞という形で実を結んだ。それ以後、「京」戦略分野、ポスト「京」やその他のプロジェクトを通じ、物質科学分野では、計算機科学の研究者あるいは線形計算やその他の数値アルゴリズムの研究者との共同研究が急激に増えた。

近年では「マテリアルズインフォマティクス」

の旗印のもと、物質科学とデータ科学の共同研究が活発化しつつある。また、今後ますます計算科学のソフトウェアが大規模化することを考えると、複数人での開発や新しいメンバーの参加がスムーズに行えるように、プログラムの書き方や開発方法論といったソフトウェア工学的な技術も重要になってくると思われる。さらに、データベースやクラウドなど、IT 関連技術の中にも、科学の発展に貢献し得るものが数多くあるようにも思われる。

技術の発達に伴い、物質科学の人間だけでは、それらを有効に活用することが難しい時代になっている。今後さらなる異分野交流が進むことを願い、本稿ではその先駆的な例となった、計算機科学の研究者と物質科学の研究者との共同研究の成果である、RSDFT の開発に関するお話を紹介したい。

2. ナノ、量子力学、第一原理計算

物理系の数値シミュレーションを行うにあたり、その基礎となる方程式には、微視的レベルから巨視的レベルまでさまざまなものがある。量子力学の基礎方程式に基づくアプローチは、おそらく最も微視的なレベルから物事を記述する仕方であろう。物質は原子の集まりであり、原子は電子と原子核から構成されている。そしてそれらの系の振る舞いは量子力学によって支配されているという、自然の最も基礎的な原理に立ち返って計算を行うことを、われわれは「第一原理計算」と呼んでいる。

現在、生体分子や電子デバイスなど、物質科学の広範な分野においてナノメートルサイズの系に大きな注目が集まっている。また、より大きなサイズでの材料開発を行うような場合でも、ナノレベルの知見を得ることで性能や機能の向上につながる事例はごまんとある。

ナノスケールの世界においては、電子の量子力学的振る舞いがその物性に本質的な影響を及ぼすため、第一原理計算による数値シミュレーションの重要性が非常に大きくなっている。量子力学の基礎方程式はシュレーディンガー方程式であ

る。電子系であれば相互作用ポテンシャルの形も正確に分かっているので、方程式自体は簡単に書き下すことができる。しかし、例えば N 電子系のシュレーディンガー方程式は、空間変数 $3N$ 次元の関数を決定する方程式となっており、これは現在の計算機能力をもってしても高々 $N=10$ 電子程度の系を計算するのがやっとである。従って応用上興味を持たれる多電子系の第一原理計算を、シュレーディンガー方程式をそのまま解くというやり方で実行するのは、ほとんど絶望的ということになる。

量子力学的な計算でわれわれが最も知りたいことは、系の基底状態（エネルギー最安定状態）の性質である。ここから、その物質が安定に存在し得るかどうか、どのような形状か、弾性的性質はどうか、熱的性質はどうか、電子がどのように振舞うか、それによってどのような機能が発現するか、等々、化学あるいは物性物理学としての重要性はもちろん、産業応用的な観点からも重要な、数多くの知見が得られることになる。実はこのような物質の基底状態の性質だけを知りたい場合には、より数値計算に適した、シュレーディンガー方程式とは全く異なる量子力学の定式化が可能であることが、1960 年代に W. Kohn らによって証明されている [2]。その理論によれば、空間変数 3 次元の関数である電子密度だけを用いて多電子系の基底状態を厳密に計算することを可能にする。この「密度汎関数理論(DFT = Density Functional Theory)」によって、数値的に取り扱いが容易な空間 3 次元の関数のみを使って多電子問題を計算する道が開かれ、扱える系のサイズが飛躍的に大きくなった。われわれが開発した RSDFT も、この理論に基づいて第一原理計算を実行するためのプログラムである。次章で、その DFT という理論の概要を簡単に紹介する。

3. 密度汎関数法

3.1. 数学的基礎に関する現状

DFT の有名な教科書に「Density Functional Theory of Atoms and Molecules」というものがある [3]。ところが日本語訳の題名は「原子・分子の密

度汎関数法」となっている。「～理論」と「～法」の使い分けに明確なルールがあるわけではないが、個人的には、「～法」という場合は数学的に細かい話は置いておいて、とりあえず使ってみてどうなるかを議論する実用主義的な意味合いが込められていると解釈している。DFT の数学的な側面は先述のような教科書にも、ある一定レベルのところまでは書かれることが多い。しかし現時点においても、その数学的基礎に関する議論は収束しておらず、密度汎関数“法”の専門家では、中々その全貌は把握しきれないのが現状である[4]。本章も、密度汎関数“法”の、さらに大雑把な概要について書いただけのものである。しかし逆にその方が、「DFT のアイデア」自体を汲み取るのに役立つのではないかと思う。

3.2. 理論の概要

原子を空間に配置して対象とする系のモデルを組み立てる。各原子にはその原子番号の数だけ電子が付随している。水素であれば電子1個、炭素であれば電子6個、という具合である。DFT の標準的な方法では、まず系のエネルギー汎関数を以下のように与える。

$$E[\varphi] = \sum_{i=1}^N \int d\mathbf{r} \varphi_i^*(\mathbf{r}) \left(-\frac{1}{2} \nabla^2 \right) \varphi_i(\mathbf{r}) + \frac{1}{2} \iint d\mathbf{r} d\mathbf{r}' \frac{\rho(\mathbf{r})\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} + E_{xc}[\rho] + \int d\mathbf{r} v_{ext}(\mathbf{r})\rho(\mathbf{r}) \quad (1)$$

ここで $v_{ext}(\mathbf{r})$ は電子と原子核の間の相互作用を表すポテンシャルで、原子配置が決まれば容易に書き下せるものである。 $E_{xc}[\rho]$ は交換相関エネルギーと呼ばれるもので、他の3つの項で表しきれなかった、系の量子力学的効果を全て押し込めた汎関数である。 $\varphi_i(\mathbf{r})$ は一電子波動関数で、これを用いて系の電子密度 $\rho(\mathbf{r})$ は次のように表される。

$$\rho(\mathbf{r}) = \sum_{i=1}^N |\varphi_i(\mathbf{r})|^2 \quad (2)$$

波動関数が規格直交性

$$\int d\mathbf{r} \varphi_i^*(\mathbf{r})\varphi_j(\mathbf{r}) = \delta_{i,j} \quad (3)$$

を満たすという条件の下で、式(1)のエネルギー汎関数を最小化すると、その最小値が系の基底状態の全エネルギーとなり、最小値を与える波動関数から基底状態の電子密度が得られる。具体的にエネルギー最小化を実行する変分方程式を書き下すと

$$\left(-\frac{1}{2} \nabla^2 + v_s(\mathbf{r}) \right) \varphi_i(\mathbf{r}) = \varepsilon_i \varphi_i(\mathbf{r}) \quad (4)$$

$$v_s(\mathbf{r}) = \int d\mathbf{r}' \frac{\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} + \frac{\delta E_{xc}}{\delta \rho(\mathbf{r})} + v_{ext}(\mathbf{r}) \quad (5)$$

が得られる。これを Kohn-Sham 方程式と呼ぶ。式(4)は、形の上ではポテンシャル場中に電子が1個いる場合のシュレーディンガー方程式であるが、ポテンシャル $v_s(\mathbf{r})$ が電子密度 $\rho(\mathbf{r})$ に依存しており、電子密度は式(2)により方程式の解そのものから作られるため、見た目ほど単純な式ではない。すなわち式(4)は単なる固有値問題ではなく、左辺の演算子(行列)が固有関数(固有ベクトル)に依存する非線形問題であり、電子密度、ポテンシャル、固有関数が、自己無撞着となるように解を求める必要がある。

3.3. 交換相関汎関数

このように、DFT を経由すれば、本来 3N 次元の関数の多電子シュレーディンガー方程式を解かなければ得られなかったはずの情報が3次元の関数の計算だけで得られることになり、ほとんど絶望的と思われていた第一原理計算が、今では産業界における材料開発にも利用されるほど実用的

なツールとなっている。しかし、DFTで精度よく元の問題と同じ情報が得られるかどうかは、交換相関エネルギー汎関数 $E_{xc}[\rho]$ という項の形に大きく依存する。厳密な関数形は知られておらず、実際の計算においては何らかの近似形を用いることになる。交換相関汎関数については実にさまざまな研究が行われているが、現在最も広く使用されているものとして、空間的に一様な電子密度を持つ仮想的な系についての厳密な計算結果から得られた関数形を、非一様な密度を持つ現実系の計算にもそのまま使う「局所密度近似 (LDA = Local Density Approximation)」や、電子密度とその勾配までを用いて関数形を作る「一般化勾配近似 (GGA = Generalized Gradient Approximation)」と呼ばれるものなどがある [5]。実際にこれらを用いて得られる結果は、高い定量性をもって実験値を再現することが知られている。一方で、バンドギャップを過小評価する、あるいは強相関電子系を上手く記述できないというような欠点も知られているが、多数の適用例と、確固たる理論的基礎を有することにより、信頼性の高い物性予測を得るためのツールとして、物理、化学はもちろん、産業界での研究開発においても広く利用されるようになってきている。

4. 大規模第一原理計算

4.1. ナノ

DFTにより、多体シュレーディンガー方程式をまともに解いていたのではとても手が出ないような物質も第一原理計算のターゲットに入ようになった。現状、1000原子程度の系であれば、比較的気軽に第一原理計算を実行することが可能である。しかし、数千原子、数万原子、・・・となると、第一原理計算による物性研究は断念するのが通例である。ところが、われわれの興味の対象となるナノスケールの世界は、原子数で言えば、まさに数千～数万原子からなる系である。しかも量子力学的効果が顕著に現れるため、信頼性の高い数値シミュレーションを行うためには第一原理計算が必須になるという、非常に厄介なターゲットであるといえる。しかし、「京」のような

10 PFLOPSを越える演算性能を持つ計算機が登場し、その能力をフル活用することができれば、数万原子からなる超大規模系も十分第一原理計算のターゲットとなり得る状況になってきた。

4.2. 第一原理計算と超並列計算機

時代がベクトル型スーパーコンピュータから並列型スーパーコンピュータへと移り変わりはじめた頃、われわれは、超並列計算機上での第一原理計算を念頭において、実空間差分法によるDFT計算コードRSDFTの開発に取り組んだ [1]。

物性物理の分野で使われるDFT計算コードは、無限に広がったバルク系を扱うため、周期境界条件を課すのに適した平面波基底関数を用いる実装がほとんどである。すなわち式(4)の波動関数 $\phi_i(\mathbf{r})$ を有限個の平面波で展開し、その展開係数に対する固有値方程式を解く。その際に必要な行列-ベクトル積は、高速フーリエ変換を用いることで疎行列の演算と同等になり、これによって平面波法は非常に効率的な計算方法となっている。しかし超並列計算機全盛の現在においては、高速フーリエ変換を多用する平面波法は、全対全通信のコストがかさむために並列性能が上がらず、大規模計算に向かない手法となりつつある。これが、われわれが新たに実空間差分法による第一原理計算コードの開発を始めた背景である。

5. 実空間法

5.1. 実空間法とは

「実空間法」とは、平面波法がフーリエ変換により逆空間(波数空間)で定式化された計算手法であることに対比して使われる用語である。狭義には実空間差分法を指すが、広義には、原子軌道、有限要素法、ウェーブレット法等も実空間法と呼ぶことがある。実空間差分法は1994年に、J. R. Chelikowskyらによって、物性物理における第一原理計算に導入された方法で、3次元実空間を格子状に切って、その上でKohn-Sham方程式を差分法で解くというものである [6]。全ての計算を実空間で行うことにより、高速フーリエ変換を必要

とせず、超並列計算に非常に適した手法となっている。また周期境界条件に縛られることがなく、バルクではない分子やクラスターといった有限孤立系の計算や、周期境界条件と孤立境界条件が混在している表面・界面スラブ系、半無限系など、さまざまな境界条件に非常に柔軟に対応できるという長所も兼ね備えている。

5.2. RSDFT

実空間格子上で離散化された Kohn-Sham 方程式は、格子点数の次元の行列の固有値問題となる。ただし行列が固有ベクトルに依存する構造になっているため、非線形固有値問題として自己無撞着な解を求める必要がある。固有ベクトルは、系の電子密度を構成するのに必要な本数だけあればよい。すなわち系の電子数と同じ程度の本数でよく、これは行列の次元 (= 格子点数) よりもはるかに小さい。従って、われわれが行うべき数値計算は、大規模疎行列の固有値問題とほとんど同じである。ただし、行列要素を陽に持たず、行列-ベクトル積の結果 (ベクトル) を返すルーチンだけが利用可能だとして、アルゴリズムや前処理法を作る必要がある。この問題の解法として、われわれは共役勾配法を採用している。これは従来の平面波基底展開法による固有値問題にも利用されてきた解法である。そのアルゴリズムを図 3 にまとめる。大雑把に言えば、通常の大規模疎行列の固有値解法アルゴリズムの途中に、電子密度とポテンシャル関数の更新が加わったものになっている。密度およびポテンシャルの更新にかかる時間はほとんど無視できるので、計算のボトルネックは Gram-Schmidt の直交化と部分空間での密行列対角化という、疎行列固有値問題ソルバー部分になる。これらの計算コストは、空間格子点数を M 、固有ベクトル本数 (≒ 電子数) を N とすれば、それぞれ $O(MN^2)$ および $O(MN^2) + O(N^3)$ となる。システムサイズを原子の数で規定することにすれば、 M, N ともに原子数に比例して増大するので、RSDFT 全体の計算時間は原子数の 3 乗に比例して増大することになる。

6. チューニングと並列化

計算時間がシステムサイズ (原子数) の 3 乗に比例するボトルネック部分のチューニング、および「京」のような数万ノード、数十万コアを持つ計算機上での並列化を行うにあたり、われわれ計算物質科学の人間だけでなく、計算機科学者の力を借りることで大きな成果をあげることができた。

ボトルネックの 1 つである Gram-Schmidt 直交化は、通常、ベクトルどうしの演算を基本としてアルゴリズムが書かれており、それを素直に実装すると、BLAS 1 あるいは BLAS 2 に相当する演算だけが現れることになる。しかし Gram-Schmidt のアルゴリズムを複数本のベクトルを同時にまとめて処理するようなアルゴリズムに書き直せることが分かり、そのようにアルゴリズムを変更すると、演算の大部分が BLAS 3 に相当する行列-行列積の形で行えるようになる。行列-行列積のような BLAS 3 の演算は、例えば Intel 系の CPU であれば MKL を利用することで、ピークの 8 割から 9 割という性能を容易に達成することができる [3]。

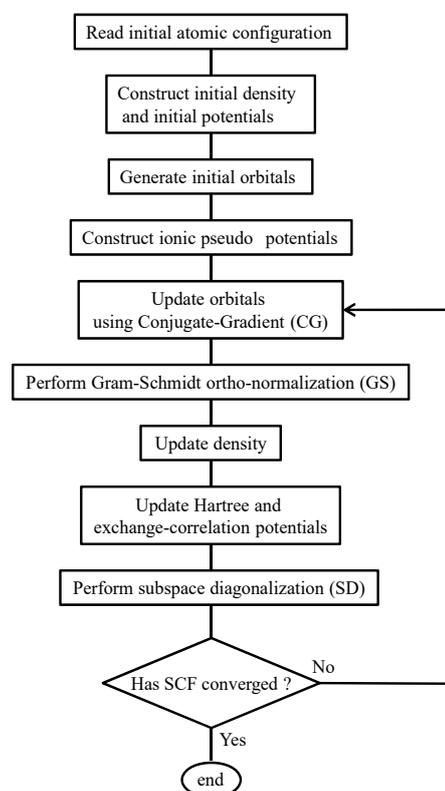


図 3 共役勾配法による Kohn-Sham 方程式の自己無撞着解法

もう1つのボトルネックである部分空間対角化でも、Gram-Schmidt 直交化で用いたのと同様、行列-行列積化のアルゴリズムが適用できる。部分空間対角化では、これに加えて $N \times N$ の密行列の対角化を行う必要がある。この演算コストも $O(N^3)$ であり、計算時間とメモリの両方の観点から並列計算が必須となる。この部分に、われわれは ScaLAPACK の固有値ソルバーを利用している。また「京」において大規模実行を行う際には ScaLAPACK の代わりに、T. Imamura らによって開発された Eigen ライブラリを利用することもある [7]。

RSDFT の元々のコードは、MPI による空間並列、すなわち 3 次元実空間を複数のブロックに分けて、各ブロックを各ノードが担当するという仕方で並列化を行っていた。主要な通信は、差分法の境界値交換のための隣接通信と、空間積分のための reduction 通信である。「京」は 8 万 CPU、64 万コアからなる超並列計算機であり、MPI による空間並列のみでこれを使い切るのは非常に困難である。特に「京」では MPI が使用するメモリ量が大きくなり、全てのコアを MPI 並列に回そうとすると、途端にメモリ不足に陥る。そこでわれわれは、ノード内は OpenMP によるスレッド並列を行い、ノード間は MPI で並列化する、いわゆるハイブリッド並列実装を行った。それでもなお 8 万ノードを空間分割だけで使用すると、特に reduction 通信の性能劣化により並列化効率が上がらなくなる。そこでわれわれは、これまでの空間並列のみの実装から、まず固有ベクトルを Q 個のグループに分割し、各グループに P 個の CPU を割り当て、その P 個の CPU で空間分割を行う方法を実装した。われわれはこれを「2次元分割」とか「2軸並列」と呼んでいたが、これにより、空間積分の reduction 通信に参加する CPU 数が $1/Q$ になるため、上記の並列化効率の劣化を軽減することができる。空間+固有ベクトルによる 2 軸並列化は、演算が $O(MN)$ のルーチンに関してはほとんど自明な並列化が可能である。一方、Gram-Schmidt のような $O(MN^2)$ のルーチンでは、自グループが計算を担当した固有ベクトルの情報を他の CPU グループに

渡す必要が生じる。このため通信パターンが複雑になり、ロードインバランスが起りやすくなるが、固有ベクトルを各 CPU グループにブロックサイクリックに割り当てることで、効率的な並列化を実現している。

7. 「京」でどこまで計算できるか

半導体デバイスの性能向上はこれまで MOSFET (Metal-Oxide-Semiconductor Field-Effect Transistor) の微細化によって達成されてきた。しかし微細化が進み、漏れ電流の抑制が困難となってきたために、従来の平面的な構造の FET から、全く異なる構造の FET を導入されるようになってきている。そのような新構造トランジスタのうち、Si ナノワイヤトランジスタは実用化が有望視されているものの1つである。Si ナノワイヤトランジスタが導入される世代では、チャネル長は 10 nm 以下となっており、またワイヤ断面のサイズは 20 nm 以下が最適になるという見積もりがある。このときのナノワイヤチャネルを構成する Si 原子の数を数えてみると、サイズにより数千~10 万個程度の原子が含まれることが分かる。すなわち、10 万原子系の第一原理計算が実行可能になれば、実際のトランジスタをほぼ丸ごと、経験的な実験パラメータを持ち込むことなく、量子力学の基礎原理だけでシミュレーションできることになる。

2011 年、まだ開発途中の「京」を利用し、「京」用のチューニング行ってきた RSDFT を用いて、われわれはいくつかの形状や長さの異なる Si ナノワイヤの第一原理計算を行い、それらが電子状態にどのような違いとなって現れるかを見た。物理的な議論は文献に譲ることにして [8,9,10]、ここでは計算の規模と時間についての結果だけを見ることにする。原子数にして 1 万~4 万原子程度までの系の計算を行い、自己無撞着な電子状態を得るのに、「京」の 5% 程度のリソースを用いて 1~2 日程度を要した。また、10 万原子計算の実行可能性を見るために、自己無撞着反復計算 1 ステップ分の計算時間と計算性能の評価も行った。システムサイズは、原子数 107,292、格子点数 63,700,992、固有ベクトル本数 229,824 で、計

算リソースは「京」全体の約7割にあたる 55,296 ノード (理論ピーク性能 7.07 PFLOPS) を用いた。このノード数を三分割して固有ベクトル並列を行い、各 18,432 ノードを空間並列に用いた。その結果、反復 1 ステップに約 5500 秒かかることが分かり、収束するまで反復を繰り返したとすれば約 5 日で 10 万原子系の自己無撞着な電子状態が得られることが分かった。このときの演算性能は、3.08 PFLOPS で、ピーク性能比 43.6 %を達成した。これら Si ナノワイヤの大規模計算とその演算性能により、われわれは 2011 年度 ACM ゴードンベル賞を受賞した[8]。その後、「京」の全システム (10.62 PFLOPS) を利用したベンチマークを行い、その頃には計算機システム自体のチューニングも進んでいたこともあり、同じ 10 万 Si 原子系において、反復 1 ステップに約 2200 秒、ピーク性能比 51.7 %を達成している [9]。

8. エピローグ

長かった戦いもついに最後の時を迎える。巨大な筐体の大部分は崩れ去り、あとは何の攻撃力も持たない中枢部分を残すのみとなったスーパーコンピューター。とどめをさすため歩み寄るジュン。そのとき、もはや瓦礫と化した筐体の奥から、どこか懐かしい声が聞こえてくる。

??? 「大きくなったな、ジュン・・・」

ジュン 「と、父さん？」

(おわり)



図 4 ???

参考文献

- [1] J.-I. Iwata *et al.*, J. Comp. Phys. 229, 2339 (2010).
- [2] P. Hohenberg and W. Kohn, Phys. Rev. 136, B864 (1964); W. Kohn and L. J. Sham, Phys. Rev. 140, A1133 (1965).
- [3] R. G. Parr and W. Yang, “Density-Functional Theory of Atoms and Molecules”, Oxford, New York, NY (1989).
- [4] E. S. Kryachko and E. V. Ludena, Phys. Rep. 544, 123 (2014).
- [5] J. P. Perdew and A. Zunger, Phys. Rev. B23, 5048 (1981); J. P. Perdew, K. Burke, M. Ernzerhof, Phys. Rev. Lett. 77, 3865 (1996).
- [6] J. R. Chelikowsky *et al.*, Phys. Rev. B50, 11355 (1994).
- [7] T. Imamura, S. Yamada, M. Machida, Proceedings of Joint International Conference on Supercomputing in Nuclear Applications and Monte Carlo 2010 (2010).
- [8] Y. Hasegawa *et al.*, Proceedings of SC’11, 1:1, ACM, NY, USA (2011).
- [9] Y. Hasegawa *et al.*, Int. J. High Perform. Comp. Appl. 28, 335 (2014).
- [10] S. Kyogoku, J.-I. Iwata, and A. Oshiyama, Phys. Rev. B87, 165418 (2013).

※ 技術情報誌アドバンスシミュレーションは、アドバンスソフト株式会社 ホームページのシミュレーション図書館から、PDF ファイルがダウンロードできます。(ダウンロードしていただくには、アドバンス/シミュレーションフォーラム会員登録が必要です。)