

# ナノ物性計測シミュレータのための境界マッチング 密度汎関数法プログラムの高速化とその性能評価

宋 応文<sup>†</sup>, 古家 真之介<sup>†</sup>, 渡 邊 聡<sup>†</sup>

ナノメートルスケールでの電気的な物性計測における計測量を第 1 原理から評価するために、我々のグループでは境界マッチング密度汎関数法を開発した。この手法は標準的な密度汎関数法計算より計算規模がいっそう大きくなるにもかかわらず、新しい手法であるためにまだ並列化などの高速化が十分ではない。本研究では、開発済みの計算プログラムに対し Hitachi SR8000 上で MPI を用いた並列化を施し、さらにロードバランスや並列粒度の調整、非同期通信の使用などの対策を実施した。その結果、ノード内に対しては自動要素並列化を用いた場合に比べ、ノード内も MPI を用いて並列化したうえで上記の対策を実施した場合には約 8 倍の性能改善を達成した。

## Performance Improvement and Evaluation of Boundary-matching Scattering-state Density Functional Method Program for Simulators of Measurements of Nanoscale Materials Properties

YINGWEN SONG<sup>,†</sup>, SHINNOSUKE FURUYA<sup>,†</sup>  
and SATOSHI WATANABE<sup>,†</sup>

In order to estimate measured quantities in the measurements of nanoscale materials properties, we proposed Boundary-matching Scattering-state Density Functional method and developed a program based on the method. This method requires more expensive computation than the standard density functional method, and the performance of our program has not been tuned sufficiently yet. In this work, we have parallelized our program using MPI on Hitachi SR8000. Further, we have evaluated its performance, and improved it by changing intra-node parallelization method from the automatic element parallelization to MPI, adjusting load balance, increasing granularity, using non-blocking communication, etc. The performance of the final program has been improved about 8 times compared with the original parallel one.

### 1. はじめに

次世代の電子デバイスの実現を目指した微細構造加工技術は最近急速に進歩しており、次の段階として作製されたナノメートルスケール構造の物性を計測すること（これを以下ナノ物性計測と呼ぶ）が重要である。しかしナノ物性計測では、対象とプローブとの相互作用が強いことや計測時に局所的に強い外場（電場や磁場など）が印加されることなどの理由により、従来の計測に比べて結果の解釈が一段と難しくなる。このため、計測量を信頼性高く予測でき、実験データと詳細に比較検討できるようなシミュレーションが望ま

れる。しかしながら、物質科学の広い分野ですでに活用されている分子軌道法や密度汎関数法などの第 1 原理計算手法は、この要請に応えることができない。これは、既存の標準的な方法論が孤立系または完全に周期的な系を対象とするのに対し、ナノ物性計測シミュレーションの対象は電極との間に電子が行き来する開放系であることや、非常に強い外場が印加された状態に対して標準的な方法では様々な不具合が生じることなどの理由による。

そこで我々のグループでは、境界マッチング密度汎関数法<sup>1)</sup>〔以下では BSDF (Boundary-matching Scattering-state Density Functional) 法とする〕を新たに開発した。さらに、この方法を用いてナノ物性計測の計測量を評価する「ナノ物性計測シミュレータ<sup>2)</sup>」の開発に取り組んでいる。BSDF 法と同様の機能を持つ手法としては、Lippmann-Schwinger 方程式

<sup>†</sup> 科学技術振興機構 CREST  
CREST, Japan Science and Technology Agency  
東京大学大学院  
Graduate School, The University of Tokyo

に基づく解法<sup>3)</sup>や第1原理リカージョン伝達行列法<sup>4)</sup>があげられるが, BSDF法はこれらの方法論よりアルゴリズムが簡明でありながら同等の精度を持つという特徴を有し, ナノ構造からの電界電子放出<sup>5)</sup>, 電極間原子鎖の電気特性の解析<sup>6)</sup>, ナノ構造の静電容量<sup>7)</sup>などの解析にすでに応用されている.

BSDF法を含め, 上記のナノ物性計測シミュレーションのための計算手法は, 開放系を陽に扱うために標準的な第1原理手法よりも多くの計算量を要するので, スーパーコンピュータ, クラスタコンピュータ, あるいはグリッドを用いた並列計算<sup>8)</sup>が多くの場合必要となる. しかし, 方法論が開発されてから日が浅いこともあり, 並列計算の最適化にはまだあまり踏み込まれていない. たとえば, 有限要素法の並列化において問題が生じやすいと指摘されている<sup>9),10)</sup>, 並列粒度, 負荷バランス, 通信レイテンシなどの点について, 詳しく検討した研究報告はない. そこで本研究では, BSDF法計算プログラムを Hitachi SR8000において並列実装し, その性能評価に基づいてチューニングした結果を報告する.

## 2. 境界マッチング密度汎関数法の概要とアルゴリズム

BSDF法で解くべき基本方程式は, 通常密度汎関数法と同じ Kohn-Sham 方程式である. しかし, 孤立系や完全周期系でなく半無限の電極を含んだ系を扱う点と, エネルギーが計算の結果得られる量でなく入力パラメータである点が標準的な密度汎関数法計算と異なっている. BSDF法では, ある軸( $z$ 方向とする)に垂直な2つの平面で全空間を3つの領域に分ける.  $z = -\infty$ を含む部分を領域 I, 中間部分を領域 II,  $z = +\infty$ を含む部分を領域 III とすると, 領域 I は1つの電極(これを左電極とする)の内部であり, 領域 III は計算対象によりもう1つの電極(これを右電極とする)の内部または真空領域となる. 領域 II は電極の表面部分と表面上(ないし近傍)のナノ構造を含む. 境界面に平行な方向( $r_{||}$ 方向とする)に周期境界条件を用い, 波動関数を平面波で展開すると, 波動関数  $\Psi_{E, k_{||}}(r_{||}, z)$  と有効ポテンシャル  $V_{\text{eff}}(r_{||}, z)$  は

$$\Psi_{E, k_{||}}(r_{||}, z) = \sum_j \Psi(G_{||}^j, z) e^{-i(k_{||} + G_{||}^j) \cdot r_{||}}, \quad (1)$$

$$V_{\text{eff}}(r_{||}, z) = \sum_j V_{\text{eff}}(G_{||}^j, z) e^{-iG_{||}^j \cdot r_{||}} \quad (2)$$

となる. すると解くべき Kohn-Sham 方程式は

$$\frac{1}{2} \frac{d^2}{dz^2} \Psi^j(z) = \sum_{j'} V_{jj'}(z) \Psi^{j'}(z) \quad (3)$$

の形になる. ここで

$$\Psi^j(z) = \Psi(G_{||}^j, z), \quad (4)$$

$$V_{jj'}(z) = V_{\text{eff}}(G_{||}^j - G_{||}^{j'}, z) + \left[ \frac{1}{2} |k_{||} + G_{||}^j|^2 - E \right] \delta_{jj'} \quad (5)$$

であり,  $E$  はエネルギー,  $\delta_{jj'}$  はクロネッカーのデルタである.

さて, BSDF法では, 領域 I と領域 III で任意の  $E$  に対する波動関数が既知である場合を扱う. 現在のプログラムでは電極としてイオン芯の正電荷を一様にならしたジェリウム模型を用いており, この場合には波動関数は解析的に求められる. 考えるべき全系の波動関数としては, 一方の電極内部から領域 II に向かって入射し, 一部が領域 II を透過し残りが反射される, という形のものと考えれば十分であることが知られているので, 求めるべきものは透過係数  $T$ , 反射係数  $R$  および領域 II における波動関数である. BSDF法では, 波動関数の連続条件とその微分に対する連続条件とから未知数  $R, T$  を消去した境界条件を導き, 式(3)の微分を差分化して領域 II に対する閉じた連立方程式を得, これを解くことによって所望の量を得る. このアイデアは, 初等的な1次元ポテンシャル障壁の問題を解く際には標準的に用いられているものであるが, 3次元の問題にも応用可能であることはあまり気づかれていなかった. なお, 波動関数から電流は容易に求めることができ, 印加電圧の影響も電極間のフェルミ準位の差として容易に考慮することができる.

計算は, Kohn-Sham 方程式を解く際に用いられる有効ポテンシャルが解いた結果得られる電子密度に基づく有効ポテンシャルと許容誤差の範囲で一致するまで繰り返し行う. 1回の計算の中にエネルギー, Brillouin 域内の  $k$  点, 逆格子ベクトルに対するループ計算があるが, プロセッサごとに異なる逆格子ベクトルを振り分けるとアルゴリズムが複雑になってしまい, Brillouin 域内の  $k$  点数は大規模系ほど少数になるため, 各プロセッサに異なるエネルギーを割り振る形で並列化を行う. なお, 計算プログラムは Fortran 90 を用いて作成している.

## 3. プログラムの性能解析とチューニング

### 3.1 ノード間 MPI 並列化の性能評価

計算プログラムの性能評価とチューニングには Hi-

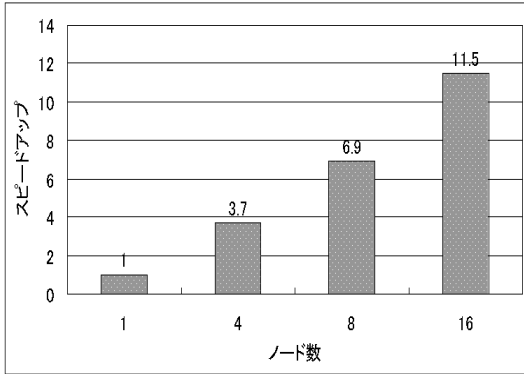


図 1 SR8000 におけるオリジナルプログラムの性能

Fig. 1 Performance of the original program on SR8000.

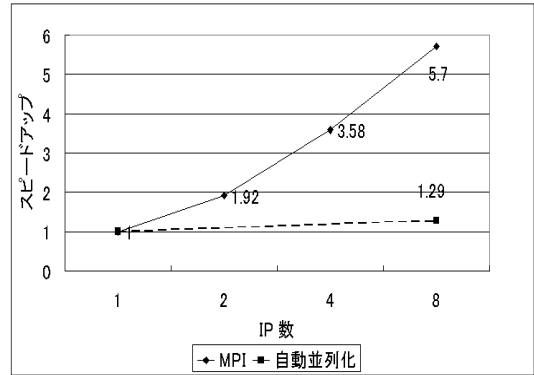


図 2 ノード内 MPI 並列化と自動要素並列化の性能比較

Fig. 2 Performance comparison of intra-node MPI and automatic parallelization.

tachi SR8000 (60 ノード (ノードあたり 8 プロセッサ)) を用いた。はじめにノード間 MPI 並列化, ノード内自動要素並列化, IP (Instruction Processor) 内ベクトル化の形で並列実装した場合について性能を解析した。並列プログラムの性能解析には多数のツールがすでに開発されているが, 管理者権限のない一般ユーザが SR8000 にインストールするのは難しいものが多い。また, SR8000 では MPI の初期化と資源解放に数十秒かかることがあるので, time コマンドによる時間計測ではプログラムの性能を正確に測れない。そこで本研究では, ソースコードの適当な場所にタイム関数を挿入して実行時間を計測し, 性能解析の一部には SR8000 の解析ツールと Vampir<sup>11)</sup> も用いた。また性能評価に用いたテストデータでは,  $z$  方向のメッシュ数 300,  $k$  点数 36, エネルギー値の数 5170, 逆格子ベクトル数 4 であり, Kohn-Sham 行列の次元数は 7500 である。

プログラムには多数回計算を行うループ部分のほかに 1 回だけ実行する前処理および後処理部分があるが, これらの部分の実行時間比率は全体の 2.4% と非常に小さいため無視できる。また, ループ中で root ノードのみが処理を行う部分の実行時間比率はループ全体の 2.7% であるため, 並列化率は高いといえる。実際, 図 1 に示した計算速度のノード数依存性を見てみると, 16 ノードで 11.5 倍の速度向上を示しており, ノード間並列化の性能は満足できるレベルといえる。

### 3.2 ノード内 MPI による並列化

本計算プログラムでは, 4 次元以上の配列を使用しており, ループ階層が深い。このため, コンパイラによる自動要素並列化では十分な性能を得られないことが考えられる。そこで 1 個の IP のみを用いた場合と 1 ノード (8 IP) 全体を用いた場合の実行時間を比較

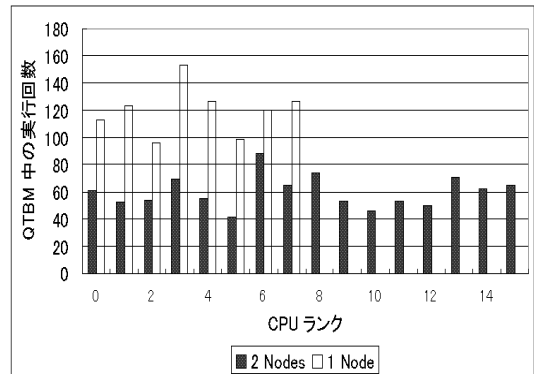


図 3 各 IP におけるサブルーチン QTBM (最も時間のかかるサブルーチン) の実行回数

Fig. 3 Execution times in the inner-most loop of QTBM (hotspot routine).

してみると, 1 IP の 225 分に対し 8 IP では 172 分とわずか 1.3 倍速い程度であることが分かった。またノード内の各 IP の負荷分散率が悪いことも判明した。これを改善する方法としてまず考えられるのは要素並列化のチューニングであるが, 本プログラムではすでに述べたように MPI の性能が良いため, ノード内に対しても MPI による並列化を行った。図 2 にノード内を自動要素並列化した場合と MPI 並列化した場合との性能の比較を示す。ソースコードの修正は特に行ってないにもかかわらず, MPI 並列化により大幅な性能改善が見られることが分かる。このことは, 自動要素並列化より IP 間のロードバランスが良くなったことを示している。

### 3.3 ロードバランスの調整

ノード内 MPI 並列化により大幅な性能向上が見られたが, スケーラビリティはまだ良くない。この原因としては通信量が多いこととロードバランスが悪いこ

```

.....
LOOPNO = 0      !This Work
DO LLL = LLLS, LLLS, LLEL    !Energy Loop
  !The energy range is divided into blocks in the original program,
  ! and each block is mapped to a CPU.
  .....
  DO L=1,NK
    !Conditional CYCLE Statement
    .....
    DO LN=1,NGY+1
      DO LM=1,NGX+1
        !Conditional CYCLE Statement
        DO LL=1,NKZ    !!!Kz loop
          !Conditional EXIT Statement
          !Conditional CYCLE Statement
          IF ((MOD(LOOPNO, NUMCPUS) == MYRANK) )THEN !This Work
            !Expensive computation start from here
            .....
            END IF      !This Work
            LOOPNO = LOOPNO +1
          END DO LL
        END DO LM
      END DO LN
    END DO L
  END DO LLL
  .....

```

図 4 本プログラムホットスポット部分の擬似コード

Fig. 4 Pseudo codes of the hotspot of this program.

とが考えられるが、本プログラムでは IP 間の通信量は少ないため、ロードバランスが問題であろうと推測される。この推測が正しいことは、プログラム中で最も計算時間のかかるサブルーチンの各 IP での実行回数(図 3 参照)から裏付けられた。図 3 からは CPU 数が増えたとこの状況がやや改善される傾向も読み取れるが、依然無視できないばらつきが見られる。このロードバランスの悪さは、エネルギー値をブロック分割して各 IP に割り当てたためにループ中の CYCLE 文や EXIT 文の実行回数が異なることから生じる。各計算ループ中の計算量は大体同じなので、実行回数の違いはロードバランスの悪さに直結する。この問題を解決するため、中に CYCLE 文や EXIT 文を含まないループに対してサイクリック分割を行った(図 4)。これにより各 IP のループ実行回数の差は最大でも 1 となり、各プロセスのホットスポット部分の計算時間もほぼ一致するようになった(図 5)。性能改善の程度はテストデータにより異なるが、前記のテストデータの場合には計算時間が 3 分の 2 に短縮された。

### 3.4 その他のチューニング

ロードバランスの調整で分割点を移動したため、旧分割点と新分割点の間の部分は非並列部分となり、計算粒度が少し小さくなってしまった。そこで、非並列部分に対し、無駄な計算の削除、重複計算のループ外移動、キャッシュを意識した配列インデックスやルー

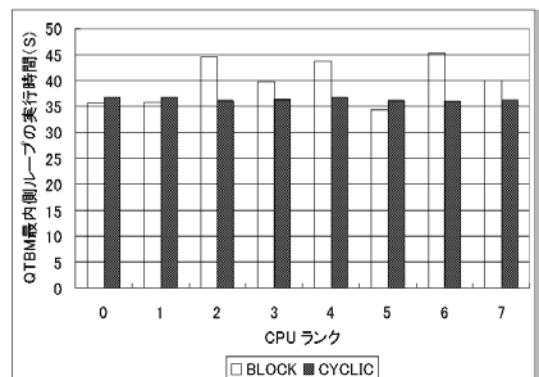


図 5 ロードバランス調整前(ブロック分割)と後(サイクリック分割)の実効負荷比較

Fig. 5 Load comparison before (block) and after (cyclic) load balance tuning.

プ順の調整、関数やサブルーチンのインライン化、外部モジュール変数のローカライズ、ループ中での外部入出力の抑制などの従来チューニング手法も適用した<sup>12),13)</sup>。後で示すように、これらの非並列部分に対するチューニングは大規模ノード数における本プログラムのスケーラビリティ向上におおいに寄与している。また本プログラムは計算量に比べて通信量が少ないため、通信レイテンシによる性能低下は比較的小さいと考えられる。しかし、一層の性能向上のため非同期通信の使用も試みた。MPI 標準ライブラリを用いて非同

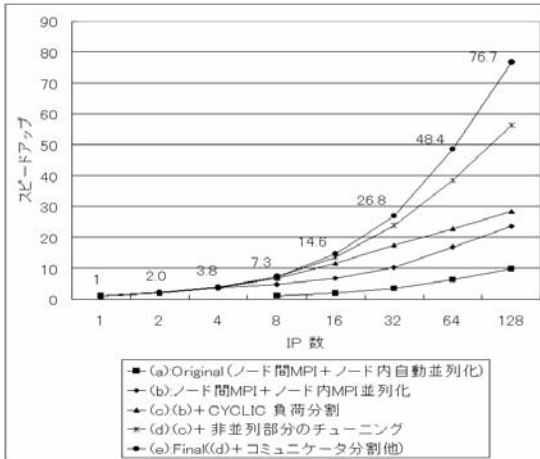


図 6 各バージョンの性能

Fig. 6 The speed up of each version.

期 BROADCAST や非同期 REDUCE サブルーチンを実装した結果、同期通信を用いた場合に比べ若干の性能改善が見られた。次に、並列粒度を大きくすることを試みた。本計算手法で電極が 2 つある系に対して計算を行う場合には入射波が左電極にある波動関数と右電極にあるものとを独立に計算することができるので、グローバルコミュニケータ中のすべてのプロセッサを 2 つのサブコミュニケータに分割することができる。これを利用して並列粒度を大きくした結果、各サブコミュニケータ内の計算量が同程度である場合には明らかな性能改善が見られたものの、電極間にバイアス電圧を印加した場合には計算量に差が生じ、逆に性能が低下することもあった。しかしこのような場合にも、各サブコミュニケータ内のプロセッサ数を調整することにより、サブコミュニケータの同期時間を明らかに短縮することができた。なお、Fortran 90 の便利な機能の 1 つとして動的なメモリ割付け・解放があるが、この機能はプラットフォーム、コンパイラおよびプログラムによっては性能の低下を招く。本研究の場合、動的メモリ管理を用いると静的メモリ管理の場合に比べ 1.5 倍程度遅かった。

### 3.5 総合評価

前節までに述べたチューニングをすべてまとめた性能比較を図 6 に示す。すでに述べたように、オリジナル版は 8 IP 時に 1 IP の約 1.3 倍の速さしか出ていなかった。プログラムの性能はチューニング作業にともない徐々に改良されたが、ノード数が比較的小数の場合には、(b) ノード内自動並列化を MPI 並列化に代えたことと、(c) CYCLIC 分割によるロードバランスの調整とによって大きく性能が向上しているのに

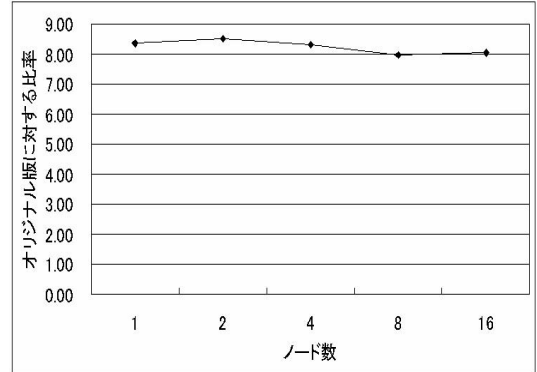


図 7 オリジナル版(ノード内自動要素並列化)に対するチューニング後の性能

Fig. 7 The performance ratio of the final version relative to the original one.

対し、ノード数が増えてくるとこれだけでは不十分で、(d) 並列化不可能部分に対するチューニング、および (e) MPI.COMM.WORLD コミュニケータの分割などのチューニングも重要になってくることが分かる。最終版の計算速度の IP 数による変化を見ると、図 6 に見られるように、8 IP では 1 IP の場合の 7.3 倍、128 IP では 76.7 倍と良好な性能を示している。ノード間 MPI 並列化とノード内自動要素並列化のみを施したオリジナル版の場合に比べ、約 8 倍の性能向上がいずれのノード数においても見られた(図 7)が、このことは各ノード内の 8 IP を十分に利用していることを示している。

メッシュ数や k 点などのパラメータを変化させた場合にも、極端に小さいモデルと大きいモデルとを除いて、一般的に上記の性能が得られる。モデルが小さい場合には、大規模ノード数使用時の通信量が計算量と比べて無視できなくなり、非並列化部分の相対比率も高くなってしまいが、この場合には計算量自体が少ないため、少数ノードで計算すればよい。モデルが極端に大きい場合の性能向上度低下は、オリジナル版では 1 ノードに 8 スレッドが走るのに対しチューニング版では 8 プロセスが走るためにメモリの使用量が数倍になり、モデルサイズ増大にともなうメモリのページフォルトがオリジナル版より早く発生することによる。ナノ物性計測シミュレーションでは、より大きなモデルでの計算を行うこと以上に同じサイズのモデルに対する多数回の計算を効率的に行うことが強く求められるので、この点は致命的な欠点とはならないが、メモリ使用量を抑える工夫は今後試みたいと考えている。

#### 4. ま と め

ナノ物性計測に対するシミュレーションに用いられる BSDF 法の計算プログラムの並列化を行い、その性能を評価・解析してチューニングを施した。Hitachi SR8000 において本計算プログラムを実行する場合には、ノード内自動要素並列化よりも MPI 並列化が有効であり、またサイクリック分割によるロードの均一化も有効であることが分かった。非同期通信の使用、粗い並列粒度の調整、従来のチューニング手法も適用した結果、最終的にはノード間 MPI 並列化とノード内自動要素並列化のみを施したオリジナルプログラムに対し約 8 倍の性能改善を達成した。SR8000 を 16 ノード使用した場合でも、1 IP の 70 倍以上の速度向上となった。

ナノ物性計測シミュレーションにおいては、多くの場合、パラメータをいろいろ変えて多数回シミュレーションする必要がある。たとえば局所仕事関数の計測は試料表面の電荷分布や極の様子を知るための有力な計測手法と考えられるが、これをシミュレートするためには探針を試料表面平行方向に走査して各点でデータを取るのに加え、各点でのデータの計算には探針-試料間距離がわずかに異なる 2 点に対してトンネル電流を計算することが必要になる。このため、これまでの予備計算<sup>14)</sup>では表面平行方向の走査は行っていなかったが、今回の性能向上によりこれも可能になり、実験データとの詳細な比較を行えるシミュレータの素地がほぼ整ったといえよう。はじめに述べたように、ナノ物性計測シミュレータはナノ領域の物性を計測する実験データを正しく解釈するうえで大変有用なツールである。したがって今後ナノ領域の物性解明を進めるうえで、ひいてはナノマテリアルやナノデバイスの設計を進めるうえで、今回行ったプログラムチューニングは大きな意義を持つものといえる。

謝辞 有益な議論と協力をいただいた科学技術振興機構 CREST「ナノ物性計測シミュレータ」チームのメンバと BSDF 法プログラムの最初期からの開発者である合田義弘氏に感謝する。また、性能解析のために Vampir をご提供いただいた Pallas 社の Mario Deilmann 氏に感謝する。性能評価とチューニング作業には東京大学物性研究所の Hitachi SR8000 を使用した。

#### 参 考 文 献

1) Gohda, Y., Nakamura, Y., Watanabe, K. and Watanabe, S.: Self-Consistent Density Func-

tional Calculation of Field Emission Current from Metals, *Phys. Rev. Lett.*, Vol.85, No.8, pp.1750-1753 (2000).

- 2) 渡邊 聡, 渡辺一之: ナノ物性シミュレータの開発に向けて, *化学工業*, Vol.54, No.4, pp.30-35 (2003).
- 3) Lang, N.D.: Resistance of atomic wires, *Phys. Rev. B*, Vol.52, No.7, pp.5335-5342 (1995).
- 4) Hirose, K. and Tsukada, M.: First-principles calculation of the electronic structure for a bi-electrode junction system under strong field and current, *Phys. Rev. B*, Vol.51, No.8, pp.5278-5290 (1995).
- 5) Gohda, Y. and Watanabe, S.: Total Energy Distribution of Field-Emitted Electrons from Al (100) Surface with Single-Atom Terminated Protrusion, *Phys. Rev. Lett.*, Vol.87, No.17, pp.177601-1-177601-4 (2001).
- 6) Furuya, S., Gohda, Y., Sasaki, N. and Watanabe, S.: Ab Initio Calculation of the Electric Properties of Al Atomic Chains under Finite Bias Voltages, *Jpn. J. Appl. Phys.*, Vol.41, No.9A/B, pp.L989-L991 (2002).
- 7) Tanaka, M., Gohda, Y., Furuya, S. and Watanabe, S.: Ab Initio Calculation of Capacitance of Semi-Infinite Jellium Electrodes with a Nanoscale Gap, *Jpn. J. Appl. Phys.*, Vol.42, No.7A, pp.L766-L768 (2003).
- 8) 湯浅太一, 安村通晃, 中田登志之: はじめての並列プログラミング, 共立出版 (1999).
- 9) 矢川元基, 塩谷隆二: 超並列有限要素法, 朝倉書店 (1998).
- 10) 矢川元基, 曾根田直樹: パラレル・コンピューティング, 培風館 (1991).
- 11) Song, Y.W.: Research on the Parallelization of Existing Codes on Cluster Computers, ERCIM Fellowship Report (2002).
- 12) J.L. ベントリー (著), 武市正人, 武市しげ子 (訳): プログラム改良学, 近代科学社 (1989).
- 13) Dowd, K., Severance, C.R.: *High Performance Computing*, 2nd ed., O'Reilly & Associates (1998).
- 14) Totsuka, H., Gohda, Y., Furuya, S. and Watanabe, S.: First-Principles Study of Apparent Barrier Height, *Jpn. J. Appl. Phys.*, Vol.41, No.10B, pp.L1172-L1174 (2002).

(平成 15 年 10 月 6 日受付)

(平成 16 年 1 月 13 日採録)



宋 応文(正会員)

昭和 43 年生。平成 11 年埼玉大学大学院理工学研究科博士後期課程修了。同年民間企業に入社してソフトウェア開発に従事。平成 13 年

ERCIM (the European Research Consortium for Informatics and Mathematics) の研究員として仏国 INRIA で HPC の研究に従事。平成 14 年民間企業で上級 HPC アナリストとして勤務。平成 15 年より科学技術振興機構研究員として東京大学で並列計算機のアーキテクチャ、プログラムの並列化、最適化、チューニングの技法に関する応用、研究、およびナノシミュレーションに従事。Ph.D., 日本シミュレーション学会, 日本計算工学会各会員。



古家真之介

昭和 50 年生。東京大学大学院工学系研究科マテリアル工学専攻博士後期課程在学中。ナノ構造物性、特に電極間ナノ構造の電気特性と、その解析手法の開発に興味を持つ。日

本物理学会会員。



渡邊 聡

昭和 36 年生。平成元年東京大学大学院理学系研究科物理学専攻博士課程修了。理学博士取得。新技術事業団青野原子制御表面プロジェクト研究員、日立基礎研究所嘱託研究員

を経て平成 9 年東京大学大学院工学系研究科助教授。現在同研究科マテリアル工学専攻教授。固体表面物性、ナノ構造物性、ナノ物性計測等に関する原子レベル計算研究に従事。日本物理学会、応用物理学会、日本表面科学会、日本金属学会各会員。