

密度行列繰り込み群法における大規模固有値計算の並列化 Parallel solver for large eigenvalue problems on density matrix renormalization group method

山田進^{1,3}, 今村俊幸^{2,3}, 奥村雅彦^{1,3}, 五十嵐亮^{1,3}, 町田昌彦^{1,3}

¹ 日本原子力研究開発機構 システム計算科学センター

² 電気通信大学 情報工学科

³CREST(JST)

Susumu Yamada^{1,3}, Toshiyuki Imamura^{2,3}, Masahiko Okumura^{1,3},

Ryo Igarashi^{1,3}, Masahiko Machida^{1,3}

¹Center for Computational Science and e-Systems, Japan Atomic Energy Agency

²Department of Computer Science, The University of Electro-Communications

³CREST(JST)

1 はじめに

近年注目を集めている高温超伝導体や強磁性体などの固体材料が示す興味深い現象の多くは、電子間に強い相関が働く強相関量子多体効果によるものと考えられている。しかしながら、量子多体問題を解析的に計算することは一般に困難である。そのため、数値シミュレーションによるアプローチが注目されており、1つのベンチマーク問題として、ハバードモデル (図 1 参照) [1, 2] と呼ばれる格子点上に存在する電子 (フェルミ粒子) 間の短距離相互作用のみを考慮したモデルの高精度シミュレーションがある。このシミュレーションでは、ハバードモデルを数式で表したハミルトニアン

$$H = -t \sum_{i,j,\sigma} (a_{i,\sigma}^+ a_{j,\sigma} + H.C.) + U \sum_i (n_{i\uparrow} n_{i\downarrow}) \quad (1)$$

の基底状態 (最小固有値とその固有ベクトル) を計算するが、このハミルトニアンの次元はモデルのサイズの拡大に伴って指数関数的に増加する。ここで、 t は運動エネルギー、 U は反発力であり、 $a_{i,\sigma}^+$, $a_{i,\sigma}$ は i 番目の格子点における (アップスピンまたはダウンスピン) の電子の生成、消滅演算子、 $n_{i\uparrow}$ ($n_{i\downarrow}$) は i 番目の格子点におけるアップスピン (ダウンスピン) の電子に対する個数演算子である。式 (1) の第一項は、電子が隣接する格子点 (j は i から見た隣接格子点) へ移動することにより、 t の運動エネルギーの利得を得ることを明示している。一方、第二項は、 i 番目の格子点におけるアップスピン (ダウンスピン) の電子の密度 (存在確率) を表しており、スピンの異なる電子がある格子点で出会うと反発力 U のエネルギー損失が生じることを表している。

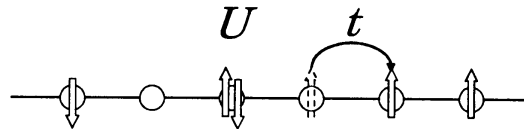


図 1: 1次元ハバードモデルの模式図。○は格子点、矢印はスピン状態の異なる電子 (フェルミ粒子) を表している。1つの格子点に同じ向きのフェルミ粒子は1つしか存在できない。 U および t はそれぞれ1つの格子点に異なる向きのスピンの粒子が存在する場合の斥力および粒子のホッピングパラメータである。

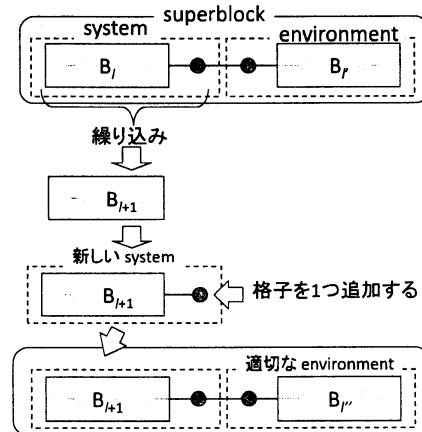


図 2: 1次元モデルに対する密度行列繰り込み群 (DMRG) 法の模式図. system と environment から構成されており, 全体を superblock と呼ぶ. system の状態を繰り込むことで格子を1つ追加しても全体の状態数が増加しないようにしている.

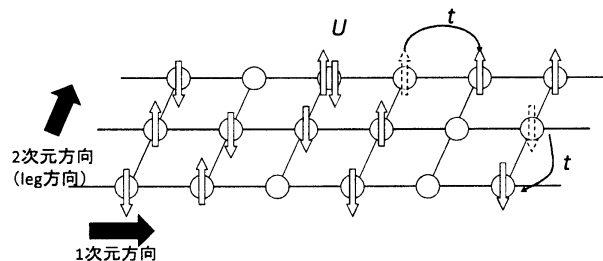


図 3: 2次元ハバードモデルの模式図. 2次元方向の格子数を leg 数と呼ぶ.

この大規模なハバードモデル (1) のハミルトニアンを計算する方法の1つに, 基底状態に関する重要な成分のみを適切に考慮して計算を行う密度行列繰り込み群 (Density-MatrixRenormalization Group, 以下 DMRG) 法がある (図 2 参照) [3, 4]. DMRG 法は本来 1 次元モデルのエネルギーを表現するハミルトニアン行列の基底状態を高精度で計算するために開発された方法であり, 実際 1 次元モデルに対しては, 数十から数千までの格子モデルの計算 (ほぼ, 実の物質の物性が理解できる範囲) を可能にし, 一次元強相関量子多体系の理解に対して多くの成功を収めてきた.

しかし, 実際の固体材料では, その 2 次元構造に高温超伝導などの興味深い現象の発現機構の要因が含まれていると考えられており [5], 図 3 のような 2 次元ハバードモデルを高精度で計算することが強く望まれている. しかし, DMRG 法は本来 1 次元のモデル用に開発されたものであるため, そのままでは 2 次元モデルに適用することができない. そのため, 2 次元モデル用に拡張するための方法が考案されている. 代表的な方法として図 4 のように 2 次元モデルをジグザクの 1 次元モデルとみなして計算する multichain 法と, 図 5 ように DMRG 法を直接 2 次元方向に拡張する direct extension DMRG (dex-DMRG) 法がある [6, 7]. multichain 法は原理的には 1 次元モデルの枠内で計算しているため, メモリ量や計算量は 1 次元モデルに対する DMRG 法とほぼ同じになるが, 本来存在していない長距離の相互作用を考慮する必要があり, その精度に関して問題があることが報告されている [8, 9]. 一方, dex-DMRG 法は短距離の相互作用を保ったまま拡張しているため, 高精度シミュレーションが期待できるが, 2 次元方向への拡張に伴って計算量が指数

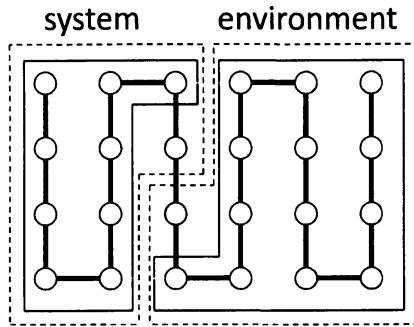


図 4: 2次元ハバードモデルに対する multichain 法の模式図. 2次元モデルを実線で示したジグザクの1次元モデルとして扱うため、破線で示した格子間の相関を長距離として扱う必要がある。

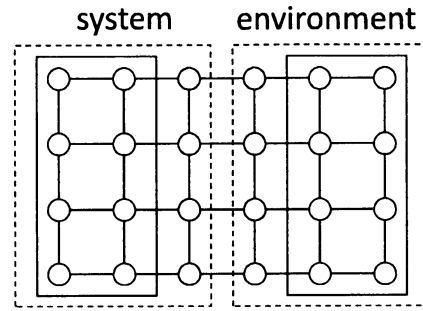


図 5: 2次元ハバードモデルに対する dex-DMRG 法の模式図. 直接2次元モデルに拡張しているため、全ての格子間の相関は短距離のみであり、高精度の結果が期待できる。

関数的に増加する。そのため、大規模な並列計算機が利用できる環境であれば、並列化した dex-DMRG 法を用いてシミュレーションすることが望ましい。

そこで、本研究では、dex-DMRG 法の並列化手法を提案し、実際に並列計算機を利用したシミュレーションにより、その有効性を確認する。

2 DMRG 法の並列化

2.1 DMRG 法のアルゴリズム

本研究で対象とする DMRG 法は、図 2 のように、モデル全体 (superblock) を system と environment に分割し、system の重要な成分を繰り込むことで、ハミルトニアン行列の次元の増加を回避する方法である。1次元モデルに対する具体的な計算方法は以下の通りである。

1. superblock に対応するハミルトニアン行列を構成し、その基底状態を計算する。
2. 1. で求めた基底状態を利用し、system に対応した密度行列を構成する。
3. 密度行列の固有値を計算し、大きい方から m 個の固有値とそれに対応する固有ベクトルを計算する。
(m は考慮する状態数)
4. 3. で得られた情報 (固有ベクトル) を利用して、system の重要な情報を繰り込み、その system に新しい格子を 1 つ追加する。
5. 適切な environment を利用して、新しい superblock を構成する。

2.2 DMRG 法の並列化

2次元モデルに対する dex-DMRG 法において、最も計算量が多く、また、メモリを必要とする演算は密度行列の固有値計算と、ハミルトニアン行列の基底状態の計算である。密度行列は密行列 (正確には各ブ

ブロック行列が密行列のブロック対角行列) であるため, ScaLAPACK などの既存の固有値計算ライブラリを使用することで, 効率的な並列計算が期待できる. 一方, ハミルトニアン行列の基底状態の計算に関しては, 行列が疎行列であることと, 基底状態のみを求めることから, Lanczos 法 [10] や PCG 法 [11, 12] などの反復法を用いて計算することが望ましい. また, この反復法において最も計算量が大きい演算は行列とベクトルの掛け算であるため, この掛け算を並列化することが重要である. 疎行列とベクトルの掛け算の並列化は, 疎行列を行方向に分割して並列計算するのが一般的であるが, 疎行列内の非零要素の分布に偏りがある場合には, 演算量および通信量に偏りができてしまう. 特に並列度が大きい場合にはその効果が得られなくなる. そのため, 超並列計算機を有効に利用するためには, 問題の物理的性質を利用することで演算量, 通信量を均等に分割する計算方法を見出すことが必須である. そこで, モデル自身にひそむ並列性を利用した並列化手法を提案する.

まず, 図 6 の上図のように 4 つのブロックで構成されているブロックを図 6 の下図のように 3 つのブロックに分割し, それぞれのブロックを表現するハミルトニアンを左から H_l, H_c, H_r とする. ここで, dex-DMRG 法の電子の移動は隣接格子点に限定されるため, これらのブロックは互いに影響を与えない. そのためハミルトニアン H は

$$H = I_4 \otimes I_3 \otimes H_l + I_4 \otimes H_c \otimes I_1 + H_r \otimes I_2 \otimes I_1$$

と単位行列と行列の直積の和で表現できる. ここで, I_i は i ブロックの状態数の大きさの単位行列である. この関係から, ハミルトニアンとベクトルの掛け算 Hv は $(I_4 \otimes I_3 \otimes H_l)v, (I_4 \otimes H_c \otimes I_1)v, (H_r \otimes I_2 \otimes I_1)v$ の 3 つの行列ベクトル積を計算し, それらを足し合わせればよいことがわかる. また, これらの掛け算に単位行列との直積が含まれていることから, 単位行列との直積計算の関係を考慮し, ベクトル v を適切に密行列 V_l, V_c, V_r に変換することで 3 つの行列ベクトル積はそれぞれ $(I_4 \otimes I_3 \otimes H_l)v \rightarrow H_l V_l, (I_4 \otimes H_c \otimes I_1)v \rightarrow H_c V_c, (H_r \otimes I_2 \otimes I_1)v \rightarrow H_r V_r$ と 3 組の疎行列と密行列の積で表現できる. こうして, 全ての計算プロセスが疎行列 H_l, H_c, H_r の情報を有していれば, V_l, V_c, V_r を列方向に分割し, 各プロセッサが密行列のデータを均等に分担することで計算量の偏りがないようにこれらの行列積を並列化することができる. すなわち, 固有状態から構成される密行列 V_l, V_c, V_r の均等分割は, 非ゼロ要素がないことから簡単に均等分割され, 超並列化可能となる. ただし, 行方向に分割した V_l, V_c, V_r 間の変換を行うための通信が必要になる. この通信を含めたこの演算の具体的な並列計算アルゴリズムは

CAL 1 $W_1^c = H_l V_l^c$

COM 1 V_l^c を V_r^c に変換するための通信

CAL 2 $Z_2^c = H_r V_r^c$

COM 2 Z_2^c を W_2^c に変換するための通信

COM 3 V_l^c を V_c^c に変換するための通信

CAL 3 $Z_3^c = H_c V_c^c$

COM 4 Z_3^c を W_3^c に変換するための通信

CAL 4 $W^c = W_1^c + W_2^c + W_3^c$

となる. ここで, 上付きの添字 c は列方向に分割した行列を表している. このときの通信の COM1-4 は全て all-to-all 通信であるため理論上は通信量も均一になる (実際のシミュレーションでは物理的に不適な状態に対するベクトル成分は無視するため, 正確には送信するデータ量はプロセスごとに若干異なる).

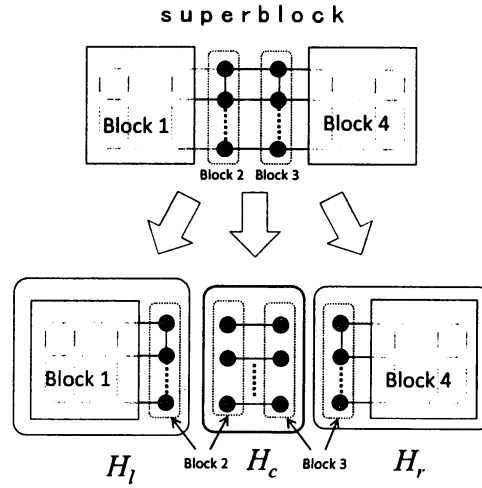


図 6: 2次元モデルの superblock の分割方法の模式図. 上図の4つのブロックを下図の3つのブロックに分割する。下図の3つのブロックに対応するハミルトニアンを左から H_l , H_c , H_r としている。

2.3 通信の最適化

前節で説明したようにハミルトニアンとベクトルの掛け算は $H_l V_l$, $H_c V_c$, $H_r V_r$ の3つの疎行列と密行列の掛け算で表せ、密行列を列方向に分割することで並列化を実現したが、分割した V_l , V_c , V_r 間でのデータのやりとりのため全プロセスでデータのやりとりを行う all-to-all 通信が必要になる。しかし、この通信はネットワークに負担がかかるため、性能の低下を引き起こす。特に、現在の並列計算機の主流であるマルチコアクラスタでは、コア当たりのネットワークのバンド幅が小さく、全プロセスでの通信は避けるのが望ましい。そのため、我々は、 V_l から V_r へのデータのやりとりを直接行うのではなく、図7のように V_c を経由させることで、全プロセスでの通信を回避する方法を提案している (詳細は文献 [13, 14] 参照)。この通信を含めたこの演算の具体的な並列計算アルゴリズムは

$$\text{CAL 1 } W_1^c = H_l V_l^c$$

COM 1 V_l^c を V_c^b に変換するための通信

COM 2 V_c^b を V_r^c に変換するための通信

$$\text{CAL 3 } Z_2^c = H_r V_r^c$$

COM 3 Z_2^c を W_2^b に変換するための通信

$$\text{CAL 2 } Z_3^b = W_2^b + H_c V_c^b$$

COM 4 Z_3^b を W_3^c に変換するための通信

$$\text{CAL 4 } W^c = W_1^c + W_3^c$$

となる。ここで、上付きの添字 c および b はそれぞれ列方向および2次元ブロックに分割した行列を表している。

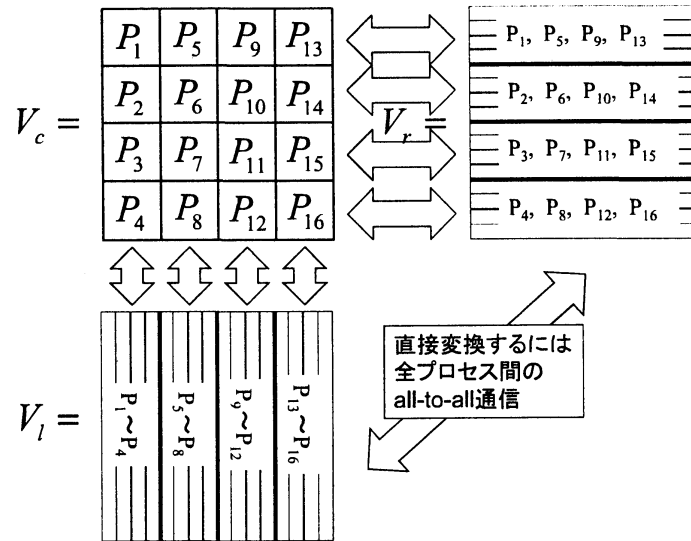
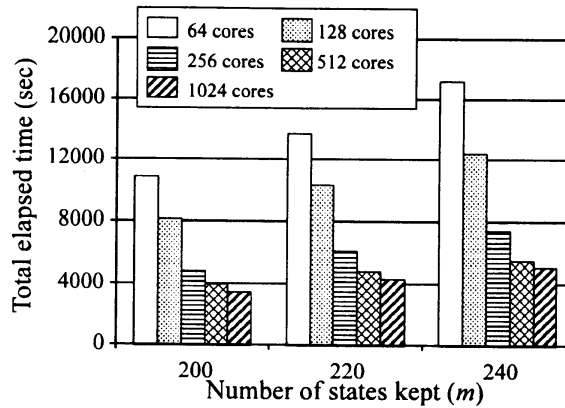


図 7: 16 プロセスの際の V_l , V_c , V_r の通信方法の模式図。 $P_1 \sim P_{16}$ は担当するプロセス番号を表している。直接 V_l , V_r 間で変換すると、all-to-all 通信が必要になる。一方、 V_l , V_c 間の変換はプロセス 1~4, 5~8, 9~12, 13~16 間の通信で実現できる。また、 V_c , V_r 間の変換はプロセス {1,5,9,13}, {2,6,10,14}, {3,7,11,15}, {4,8,12,16} 間の通信で実現できる。

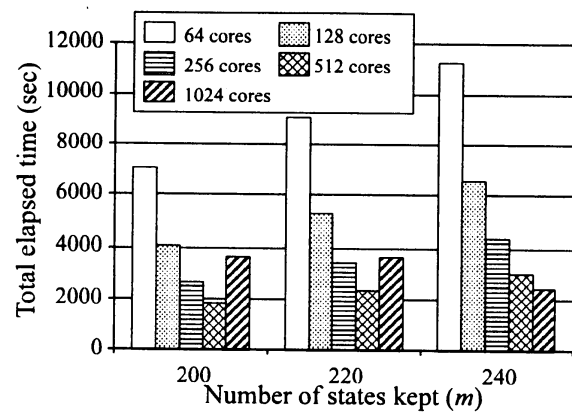
3 数値実験

ここでは、開発した並列 DMRG 法の並列化効率を確認するため、実際の問題として 4×10 サイト（アップスピン数 19 個、ダウンスピン数 19 個、 $U/t = 10$ ）のハバードモデルの基底状態を、考慮する成分数 m を変化させ、東京大学情報基盤センターの HA8000 クラスタシステム（T2K オープンスーパーコンピュータ）を利用して並列計算した際の全体の計算時間を図 8 に、行列ベクトル積の計算時間を図 9 に示す。この計算では Intel Fortran Compiler11.0 を利用し、並列化では全ての並列化を MPI 命令で実施する Flat MPI 並列を使用している。また、通信方法として、全てのプロセスでの all-to-all 通信を行うもの（all-to-all 通信）と V_c を経由するもの（2 段階通信）の 2 通りを採用する。

この結果から、all-to-all 通信では、1024 コアまで高速化しており、64 コアと比較し計算時間は 3 分の 1 以下になっている。一方、2 段階通信では繰り込み数が 240 の場合に 1024 コアまで高速化し、64 コアと比較し計算時間が約 4.5 分の 1 になっているが、繰り込み数が 200, 220 の場合は 512 コアまでしか高速化せず、1024 コアになると急速に遅くなっていることが確認できる。しかし、この 2 ケースを除けば、2 段階通信は all-to-all 通信と比較し 2 倍程度の高速化を実現している。この遅くなる原因を調査するため、2 段階通信の $m = 200$ と 240 の通信時間を表 1 に示す。この表から、理由は不明であるが繰り込み数が少ない ($m = 200$) 時の **COM 2** および **COM 3** の通信時間が非常に大きくなっていることが確認できる。そのため、さらなる DMRG の高速化のためには、この部分の通信が遅くなる原因を調査し、それを回避する通信方法を開発することが重要であると考えている。

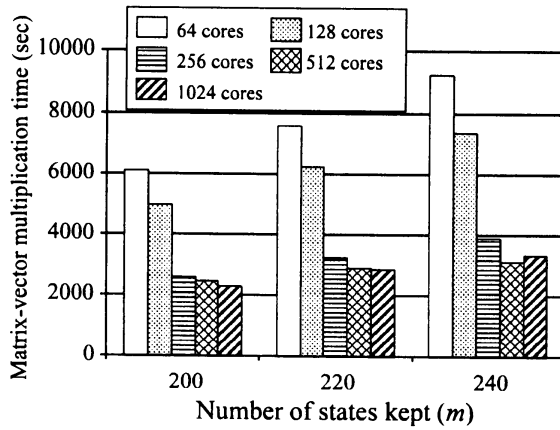


(a) all-to-all 通信を利用した dex-DMRG 法

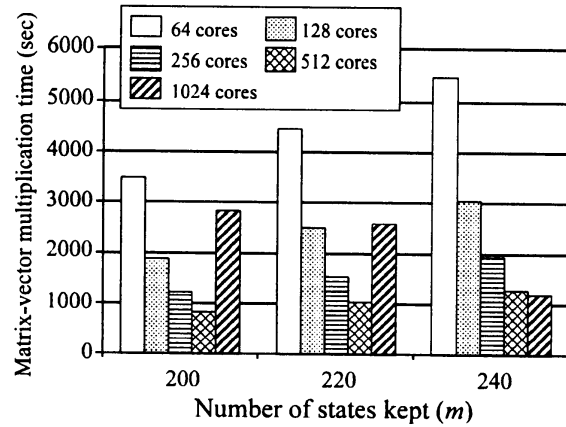


(b) 2 段階通信を利用した dex-DMRG 法

図 8: dex-DMRG 法の並列計算時間。4×10 サイトハバードモデルを dex-DMRG 法で HA8000 クラスタ (T2K スパコン) で並列計算した際の総計算時間を示している。この計算では、繰り込み数 m を 200, 220, 240 としている。



(a) all-to-all 通信を利用した dex-DMRG 法



(b) 2 段階通信を利用した dex-DMRG 法

図 9: dex-DMRG 法の行列ベクトル積の計算時間。4×10 サイトハバードモデルを dex-DMRG 法で HA8000 クラスタ (T2K スパコン) で並列計算した際のハミルトニアン行列とベクトルの掛け算部分の計算時間 (演算時間と通信時間の合計) を示している。

4 まとめ

本研究では、本来 1 次元モデル量子問題のシミュレーション手法である DMRG 法を準 2 次元モデル用に拡張するための並列化手法を提案し、東京大学 HA8000 クラスタを利用した並列計算により、繰り込み数が大きければ 1024 コアまで並列化の効果が得られることを確認した。また、データの分割方法を改良することで、現在の並列計算機の主流であるマルチコアクラスタに適していない全プロセッサ間での all-to-all 通信を行わない通信を実現し、all-to-all 通信を使用した場合と比較し、多くの場合で総計算時間が約半分になっていることを確認した。しかし、この提案した通信手法は繰り込み数が少ないと、1024 コアでの並列計算の際に急激に遅くなることが確認できた。現状ではこの理由は不明であるため、この点の解明を今後の課題の 1 つとしたい。

表 1: 2 段階通信を利用した際の通信時間

Number of processors	Communication time (sec)							
	$m = 200$				$m = 240$			
	COM 1	COM 2	COM 3	COM 4	COM 1	COM 2	COM 3	COM 4
64	178.31	212.94	214.51	116.70	257.33	314.48	316.47	172.65
128	92.37	127.94	126.78	60.83	140.57	189.84	189.92	92.58
256	58.38	86.32	84.64	40.58	87.23	124.42	121.59	60.97
512	31.08	78.71	80.50	22.24	46.14	98.73	100.47	32.61
1024	26.86	1216.24	1102.37	26.63	49.49	186.20	186.58	49.49

謝辞

本研究の一部は文部科学省科学研究補助金 (基盤研究 (C) 一般 20500044) および JST CREST プロジェクト「超伝導新奇応用のためのマルチスケール・マルチフィジックスシミュレーションの基盤構築」の成果によるものである。

参考文献

- [1] M. Rasetti, ed., The Hubbard Model: Recent Results, World Scientific, Singapore (1991)
- [2] A. Montorsi, ed., The Hubbard Model, World Scientific, Singapore (1992).
- [3] S. R. White, Density Matrix Formulation for Quantum Renormalization Groups, Phys. Rev. Lett. 69, 1992, pp. 2863-2866.
- [4] S. R. White, Density-matrix algorithms for quantum renormalization groups, Phys. Rev. B 48, 1993, pp. 10345-10355.
- [5] 今田正俊: 酸化物高温超伝導体はどこまでわかったか理論の立場から, パリティ, 丸善株式会社, Vol. 23, No. 4, pp.28-32 (2008).
- [6] S. Yamada, M. Okumura, and M. Machida, High Performance Computing for Eigenvalue Solver in Density-Matrix Renormalization Group Method: Parallelization of the Hamiltonian Matrix-Vector Multiplication, J.M.L.M Palma et al. (Eds.):VECPAR 2008, LNCS 5336, 39-45 (2008).
- [7] S. Yamada, M. Okumura, and M. Machida, Direct Extension of Density-Matrix Renormalization Group to Two-Dimensional Quantum Lattice Systems: Studies of Parallel Algorithm, Accuracy, and Performance, J. Phys. Soc. Jpn. 78 (2009) 094004.
- [8] G. Hager, G. Wellein, E. Jackemann, and, H. Fehske, Stripe formation in dropped Hubbard ladders, Phys. Rev. B, 71, 075108 (2005)
- [9] R. M. Noack and S. R. Manmana. Diagonalization and Numerical Renormalization- Group-Based Methods for Interacting Quantum Systems, Proc. of AIP Conf., 789, 91-163 (2005)

- [10] C. Lanczos, An Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators, Research of the National Bureau of Standards 45 (1950), 225-282.
- [11] A. V. Knyazev, "Preconditioned eigensolvers - An oxymoron?", *Electronic Transactions on Numerical analysis*, Vol. 7 (1998), 104-123.
- [12] A. V. Knyazev, Toward the optimal eigensolver: Locally optimal block preconditioned conjugate gradient method, *SIAM Journal on Scientific Computing*, 23 (2001), 517-541.
- [13] 山田進, 今村俊幸, 町田昌彦, マルチコアクラスタのネットワーク構造を考慮した並列密度行列繰り込み群法の通信手法, 日本計算工学会論文集, Vol. 2009 (2009), 20090015.
- [14] 山田進, 今村俊幸, 奥村雅彦, 町田昌彦, マルチコアクラスタ向け通信手法を用いた密度行列繰り込み群法の並列化, 情報処理学会研究報告, IPSJ SIG Technical Report, Vol.2009-HPC-121, No. 38 (2009).