

量子統計物理に現れる密・スパーズ行列の 高性能並列固有値計算について

東京大学・物性研究所 坂下 達哉

Tatsuya Sakashita

Institute for Solid State Physics, The University of Tokyo

1 はじめに

量子統計物理学では、解析的に固有値・固有ベクトルを求めるのが困難な場合に数値対角化が活用される。数値対角化のパッケージとしては TITPACK2 [1, 2] をはじめ KOBEPACK [3], SPINPACK [4] が知られているが、いずれもボトルネックとなる固有値解法においてプロセス並列化が施されていない。そのため、PC クラスタやスパコンの多くの CPU を生かすことができず、このことが大規模な量子系での計算を困難にしている。そこで、我々は汎用なハイブリッド並列（スレッド&プロセス並列）の数値対角化パッケージを開発している [5, 6]。

本稿の目的は、計算機実装に触れながら、量子統計物理の固有値計算について、できるだけ物理の知識を仮定せずに簡単なレビューを行うことである。特に、量子 Heisenberg 模型のハミルトニアン行列に対する疎行列向けの固有値計算に焦点を当てる。

本稿の構成は以下のとおりである。2 節では、本稿で用いる量子力学について最小限の事柄を述べた後、量子 Heisenberg 模型の定義を行う。3 節では、密および疎行列の固有値解法について述べる。4 節では、疎行列の固有値解法の際に用いられる三重対角化の手法である Lanczos 法について述べる。5 節では、Lanczos 法で必要となる行列・ベクトル積の量子 Heisenberg 模型に特化した方法について述べる。6 節をまとめとする。

2 量子統計物理の基礎

本節では、線形代数の知識のみを仮定し、量子状態と演算子の定義を行い、量子 Heisenberg 模型の定義を紹介する。

2.1 量子状態と演算子

量子力学では、量子状態を表すベクトルをケットと呼ばれる記号で表す：

$$|0\rangle := \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |1\rangle := \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

本稿では、 $|0\rangle, |1\rangle$ の線形結合で書ける量子状態を扱う。

量子状態に対して作用する以下の演算子を考える：

$$S^z := \frac{1}{2} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \tag{1}$$

$$S^+ := \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \quad S^- := \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}. \tag{2}$$

式 (1) より、 S^z は実対称行列で、その固有値は $1/2, -1/2$ 、固有ベクトルは $|0\rangle, |1\rangle$ であることがわかる。式 (2) より、 ${}^t(S^+) = S^-$ である。また、 S^+, S^- はそれぞれケットベクトルの数ラベルを上げ下げする演算

子である：

$$S^+ |0\rangle = |1\rangle, \quad S^- |1\rangle = |0\rangle. \quad (3)$$

1次元鎖上の各点（サイトと呼ぶ）に量子状態 $|0\rangle$ または $|1\rangle$ に割り当てた合成状態を考える。図1に4サイトの場合の例を示す。図1の量子状態は、テンソル積 $|1\rangle \otimes |0\rangle \otimes |1\rangle \otimes |1\rangle$ で表される。この量子状態は、テンソル積の記号を省略して $|1011\rangle$ とも書かれる。4サイトの1次元鎖上に $|0\rangle$ または $|1\rangle$ を配置する方法は $|0000\rangle, |0001\rangle, \dots, |1111\rangle$ の 2^4 通りで、これらのケットベクトルの線形結合で任意の量子状態は書けるから、この鎖上の量子状態を表現するベクトル空間の次元は 2^4 である。これより、一般のサイト数 L の1次元鎖上の量子状態を表すベクトル空間の次元は、 2^L であることがわかる。

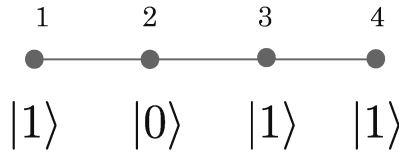


図1 1次元鎖（サイト数 $L = 4$ の場合）

2.2 量子 Heisenberg 模型

本副節では、固有値計算の題材とする量子 Heisenberg 模型の数学的な定義を行う。この模型の物理的な意味については、文献 [7, 8] 等を参照されたい。

サイト数 L の1次元鎖上の量子 Heisenberg 模型のハミルトニアン行列は、実対称行列

$$\mathcal{H} = J \sum_{\langle i,j \rangle} (S_i^z S_j^z + \frac{1}{2}(S_i^+ S_j^- + S_i^- S_j^+)) \quad (4)$$

で定義される。ここで、 $\sum_{\langle i,j \rangle}$ は、最近接サイト同士 $(i,j) = (1,2), (2,3), (3,4), \dots, (L-1,L)$ に対する和を意味する。また、式(4)に現れる $S_i^z S_j^z$ は、 $2^L \times 2^L$ サイズの行列

$$S_i^z S_j^z = I \otimes \dots \otimes I \overset{i}{\otimes S^z} \otimes I \otimes \dots \otimes I \overset{j}{\otimes S^z} \otimes I \otimes \dots \otimes I$$

を意味する。つまり、 $S_i^z S_j^z$ は、 i, j 番目の要素を式(1)の S^z とし他の要素を 2×2 サイズの単位行列 I とし、テンソルをとった行列である。式(4)の他の項 $S_i^+ S_j^-, S_i^- S_j^+$ も同様である。

式(4)は、 4×4 の実対称行列

$$\begin{aligned} \mathcal{H}_{\langle i,j \rangle} &:= \frac{1}{2} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \otimes \frac{1}{2} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} + \frac{1}{2} \left\{ \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \right\} \\ &= \begin{bmatrix} \frac{1}{4} & & & \\ & -\frac{1}{4} & & \\ & & \frac{1}{2} & \\ & & & -\frac{1}{4} \\ & & & & \frac{1}{4} \end{bmatrix} \end{aligned} \quad (5)$$

を用いて、

$$\mathcal{H} = J \sum_{\langle i,j \rangle} I \otimes \dots \otimes I \otimes \mathcal{H}_{\langle i,j \rangle} \otimes I \otimes \dots \otimes I \quad (6)$$

と書き直せる。

例えば、サイト数 $L = 4$ 、係数 $J = 1$ の場合、式 (4) のハミルトニアン行列 \mathcal{H} は、以下の $2^4 \times 2^4$ サイズの行列である：

$$\begin{bmatrix} 0.75 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.25 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & -0.25 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.25 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & -0.25 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0 & -0.75 & 0.5 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5 & -0.25 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.25 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0.25 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & -0.25 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0.5 & -0.75 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & -0.25 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0.25 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & -0.25 & 0.5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0.25 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.75 \end{bmatrix}$$

これは、ゼロ成分が非常に多い行列となっている。このような行列は疎行列 (sparse matrix) と呼ばれ、ゼロ成分を無視した効率的な計算が行える。これとは反対に、ゼロ成分が存在せず全行列成分について計算を行う必要がある行列を密行列 (dense matrix) と呼ぶ。

3 実対称行列に対する固有値解法とその並列化

本節では、実対称行列に対する密行列と疎行列向けの固有値解法について概説する。詳細については文献 [9] 等を参照されたい。

実対称行列を対角化するには、計算量を削減させるために最初に三重対角化が行われることが多い。三重対角化とは、実対称行列 A を適当な直交行列 Q を用いて、

$${}^tQAQ = \begin{bmatrix} \alpha_1 & \beta_1 & & & & \\ \beta_1 & \alpha_2 & \beta_2 & & & \\ & \beta_2 & \ddots & \ddots & & \\ & & \ddots & \alpha_{n-1} & \beta_{n-1} & \\ & & & \beta_{n-1} & \alpha_n & \end{bmatrix} \quad (7)$$

というように対角成分と副対角成分からなる三重対角対称行列に変換することである。この三重対角化は、密行列と疎行列で用いられる方法が異なる。

以下の副節で密・疎行列のそれぞれに対する固有値解法および代表的なライブラリについて述べる。

3.1 密行列向け

密行列に対する固有値解法では、Householder 法による三重対角化が用いられる。三重対角対称行列の固有値解法には、QR 法、2 分法、分割統治法がある。

これらの解法を実装した標準的な逐次版またはスレッド並列版の固有値ソルバとして LAPACK [10] が知られている。また、ハイブリッド並列 (スレッド & プロセス並列) が施された固有値ソルバには、ScaLAPACK [11] や Elemental [12] に含まれるものがある。さらに、スーパーコンピュータ「京」のような多くの CPU が搭載されたマシンでの並列実行も想定した EigenExa [13]、ELPA [14] といった固有値ソルバも開発され、今なお改良が続けられている。

ハイブリッド並列の固有値ソルバは、広く普及しているとはいえない。その原因として、これらの固有値ソルバの呼び出し方やメモリ分散の方法がおのおの異なっており、ユーザにとってわかりづらいことが考えられる。そこで、我々は密行列向けの固有値ソルバに対する統一的なインターフェース Rokko [5] を開発して公開している。

3.2 疎行列向け

疎行列に対する固有値解法には、Lanczos 法、Jacobi-Davidson 法、LOBPCG (Locally Optimal Block Preconditioned Conjugate Gradient) 法がある。4 節では、三重対角型を経由する解法である Lanczos 法のみをとりあげて解説する。

現在も開発・メンテナンスが行われている固有値ソルバを含む疎行列用ハイブリッド並列固有値ライブラリとしては、SLEPc [15], Trilinos [16] (多分野の数値計算のオープンソースソフトウェアを統合するフレームワーク) を構成する一つのパッケージである Anasazi がある。疎行列向けの固有値ソルバへデータを渡す方法には 2 通りある。

第 1 の方法は、疎行列の非ゼロ成分のみを求め固有値ソルバに渡す方法である。この方法で用いられる疎行列の代表的な格納方式には、CRS 方式 (Compressed Row Storage, 圧縮行格納方式) がある。CRS 方式は、各行ごとに非ゼロ成分とその列添字を格納する方法である。以下に CRS 方式の例を示す。

例．以下の疎行列を考える：

$$\begin{bmatrix} 7 & 9 & 0 & 0 \\ 0 & 0 & 0 & 6 \\ 4 & 0 & 0 & 5 \\ 0 & 0 & 7 & 0 \end{bmatrix}$$

これを CRS 方式で表したのが以下のデータである：

- 非ゼロ成分に対する行添字 = [1, 1, 2, 3, 3, 4]
- 非ゼロ成分に対する列添字 = [1, 2, 4, 1, 4, 3]
- 非ゼロ行列成分 = [7, 9, 6, 4, 5, 7]

この方法では、メモリ分散並列化は行単位で行え、この並列化もライブラリが自動でやってくれるという利点がある。欠点は、行列格納にかかる CPU 時間およびメモリを要することである。

疎行列向け固有値解法では固有値分解を行うべき行列がその行列とベクトル積の形でしか用いられない。この事に基づいて、行列そのものではなく行列・ベクトル積を行うルーチンを用意して固有値ソルバに渡すのが第 2 の方法である。この方法の利点は、行列格納にかかる CPU 時間・メモリを要せず、行列の形状に適した効率化が行えることである。一方、欠点は実装に手間がかかることである。とくに、分散メモリ向け並列化は、各プロセスが担当するベクトル成分の依存関係を考慮してデータの通信を記述する必要がある。5 節では、量子 Heisenberg 模型のハミルトニアン行列に特化した行列・ベクトル積を行う方法を紹介する。

4 Lanczos 法

本節では、疎行列向けの固有値解法として Lanczos 法を解説する。本来、Lanczos 法とは三重対角化の手法のことであるが、この三重対角化を経由する固有値解法も Lanczos 法と総称される。

Lanczos 法 (三重対角化の手法) の計算原理について述べる。任意にベクトル \mathbf{q}_1 ($\|\mathbf{q}_1\| = 1$) を選ぶ、 $A\mathbf{q}_1$ を

$$A\mathbf{q}_1 = t_{11}\mathbf{q}_1 + t_{21}\mathbf{q}_2 \quad \text{s.t. } (\mathbf{q}_1, \mathbf{q}_2) = 0, \quad \|\mathbf{q}_2\| = 1$$

と表す。さらに, Aq_2 を

$$Aq_2 = t_{12}q_1 + t_{22}q_2 + t_{32}q_3 \quad \text{s.t. } (q_1, q_3) = (q_2, q_3) = 0, \quad \|q_3\| = 1$$

と表し, これを繰り返す。

上記の操作を行列で書くと,

$$A[q_1, q_2, \dots, q_n] = [q_1, q_2, \dots, q_n] \begin{bmatrix} t_{11} & t_{12} & t_{13} & \dots & t_{1n} \\ t_{21} & t_{22} & t_{23} & \dots & t_{2n} \\ & t_{32} & t_{33} & \dots & t_{3n} \\ & & \ddots & \ddots & \vdots \\ & & & t_{n,n-1} & t_{nn} \end{bmatrix} \quad (8)$$

となる。

ここで, $Q := [q_1, q_2, \dots, q_n]$ は直交行列であり, A は実対称行列より, $T := {}^tQAQ$ も実対称行列である。よって, T は三重対角対称行列であることがわかる。この事に留意すると, 式 (8) は

$$A[q_1, q_2, \dots, q_n] = [q_1, q_2, \dots, q_n] \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \ddots & \ddots & \\ & & \ddots & \alpha_{n-1} & \beta_{n-1} \\ & & & \beta_{n-1} & \alpha_n \end{bmatrix} \quad (9)$$

と書け, 三重対角化の定義式 (7) と一致している。ここで, $\alpha_i := t_{i,i}$ ($i = 1, \dots, n$) および $\beta_i := t_{i,i+1} = t_{i+1,i}$ ($i = 1, \dots, n-1$) とおいた。

式 (9) を満たすように q_k, α_k, β_k ($k = 1, \dots, n$) を決める手続きが Lanczos 法のアルゴリズムとなる。これを Algorithm 1 に示す。

Algorithm 1: Lanczos 法	
Input:	実対称行列 A
Output:	三重対角化された行列 tQAQ で上書きされた A 。ここで, $Q := [q_1, q_2, \dots, q_n]$ は直交行列。
[1]	初期ベクトル q_1 を選ぶ。
[2]	$\alpha_1 \leftarrow (q_1, Aq_1)$
[3]	$\beta_1 \leftarrow \ Aq_1 - \alpha_1 q_1\ $
[4]	$q_2 \leftarrow (Aq_1 - \alpha_1 q_1) / \beta_1$
[5]	for $k \leftarrow 2, \dots, n$ do
[6]	$\alpha_k \leftarrow (q_k, Aq_k)$
[7]	$\beta_k \leftarrow \ Aq_k - \beta_{k-1} q_{k-1} - \alpha_k q_k\ $
[8]	$q_{k+1} \leftarrow (Aq_k - \beta_{k-1} q_{k-1} - \alpha_k q_k) / \beta_k$
[9]	if $\beta_k = 0$ then
[10]	q_{k+1} を $(q_1, q_k) = (q_2, q_k) = \dots = (q_{k-1}, q_k) = 0, \ q_k\ = 1$ となるように選ぶ。
[11]	

Algorithm 1 で計算量が多いのは, 2,6 行目の行列・ベクトル積 Aq_1, Aq_k である。Lanczos 法は実際の有限桁の数値計算では $Q := [q_1, q_2, \dots, q_n]$ の直交性が悪くなるという欠点があるため, 再直交化が行われる。

三重対角対称行列 T の固有値・固有ベクトルは, 密の三重対角行列向けの解法 (QR 法, 2 分法等) で求める。三重対角行列 T の固有値を $\{\lambda_i\}$, 固有ベクトルを $\{p_i\}$ とすると, 元の行列 A の固有値は $\{\lambda_i\}$, 固有ベクトルは $\{Qp_i\}$ となる。

5 量子 Heisenberg 模型のハミルトニアン行列とベクトルの積の計算

前節で述べたように、Lanczos 法において固有値分解を行う行列は、その行列とベクトルの積の形でのみ用いられる。本節では、量子 Heisenberg 模型のハミルトニアン行列に特化した行列・ベクトル積の計算法について紹介する。詳細は、文献 [17] を参照されたい。

1 次元鎖上に $|0\rangle$ または $|1\rangle$ を配置して作ったケットベクトルは量子状態を表す空間の基底をなしており、これらのケットベクトルはそのラベルのビット列で表せる。例えば、 $|1011\rangle$ はそのラベル 1011 で表せる。この事を考慮すると、式 (4) のハミルトニアン行列 \mathcal{H} とベクトル v の積は、計算機にとって効率的なビット演算を活用して計算することができる。この計算手順を Algorithm 2 で与える。

Algorithm 2: 量子 Heisenberg 模型のハミルトニアン行列とベクトルの積	
Input:	ベクトル $v = [v_0, \dots, v_{n-1}]$, 1 次元鎖のサイト数 L
Output:	ベクトル $w = [w_0, \dots, w_{n-1}] = \mathcal{H}v$
[1]	$n \leftarrow 2^L$
[2]	$w \leftarrow 0$
[3]	for $\langle i, j \rangle \leftarrow \langle 1, 2 \rangle, \langle 2, 3 \rangle, \dots, \langle L-1, L \rangle$ do
[4]	for $k \leftarrow 0, \dots, n-1$ do
[5]	if k の i, j ビット目が 01 または 10 then
[6]	$\ell \leftarrow (k$ の i, j ビット目をビット反転した値)
[7]	$w_k \leftarrow w_k + 0.5v_\ell - 0.25v_k$
[8]	else
[9]	$w_k \leftarrow w_k + 0.25v_k$

Algorithm 2 ではビット演算を用いるため、ベクトルの添字 k や ℓ は、1 から n ではなく、0 から $n-1$ を用いていることに注意を要する。また、Algorithm 2 の 5, 6 行目にある「 k の i, j ビット目」は、2 進数表示した k の最上位ビット（最左）を 1 ビット目として数えた番号であり、式 (5) の $\mathcal{H}_{(i,j)}$ の行添字に対応する：

$$\mathcal{H}_{(i,j)} = \begin{matrix} & |00\rangle & |01\rangle & |10\rangle & |11\rangle \\ \begin{matrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{matrix} & \begin{pmatrix} \frac{1}{4} & & & \\ & -\frac{1}{4} & \frac{1}{2} & \\ & \frac{1}{2} & -\frac{1}{4} & \\ & & & \frac{1}{4} \end{pmatrix} & & & \end{matrix}$$

これより、5 行目の if 文が真となるとき、式 (4) の和の中の第 $\langle i, j \rangle$ 項は k 行目で非対角成分を持つことがわかる。この非対角成分の列添字は、6 行目で求める ℓ である。

Algorithm 2 の正しさを検証するために、 $L=3$ の場合を例として考える。この場合のハミルトニアン行列 \mathcal{H} とベクトル v の積は、式 (4) より、

$$\mathcal{H}v = (\mathcal{H}_{(1,2)} \otimes I)v + (I \otimes \mathcal{H}_{(2,3)})v \quad (10)$$

である。式 (10) の各項の計算過程を図 2 に図示する。図 2 において灰色で記した行では $\mathcal{H}_{(1,2)} \otimes I$ および $I \otimes \mathcal{H}_{(2,3)}$ は非対角成分を持っていることがわかる。

式 (10) の第 1 項に対応する図 2(a) では $\langle i, j \rangle = \langle 1, 2 \rangle$ であるから、行添字を表すケットベクトルの上位 2 ビットが if 文の判定に用いられる。実際、上位 2 ビットが 01 となる $|010\rangle, |011\rangle$, 10 となる $|100\rangle, |101\rangle$ に対応する行で非対角成分を持っていることがわかる。また、これらの非対角成分の列添字はそれぞれ $|100\rangle, |101\rangle$

$$\begin{array}{cccccccc}
 & |000\rangle & |001\rangle & |010\rangle & |011\rangle & |100\rangle & |101\rangle & |110\rangle & |111\rangle \\
 \begin{array}{l} |000\rangle \\ |001\rangle \\ |010\rangle \\ |011\rangle \\ |100\rangle \\ |101\rangle \\ |110\rangle \\ |111\rangle \end{array} & \left[\begin{array}{cccccccc}
 1/4 & & & & & & & & \\
 & 1/4 & & & & & & & \\
 & & -1/4 & & 1/2 & & & & \\
 & & & -1/4 & & 1/2 & & & \\
 & & 1/2 & & -1/4 & & & & \\
 & & & 1/2 & & -1/4 & & & \\
 & & & & & & & 1/4 & \\
 & & & & & & & & 1/4
 \end{array} \right] \begin{array}{l} v_0 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \end{array}
 \end{array}$$

(a) 第1項 ($\mathcal{H}_{(1,2)} \otimes I$) v

$$\begin{array}{cccccccc}
 & |000\rangle & |001\rangle & |010\rangle & |011\rangle & |100\rangle & |101\rangle & |110\rangle & |111\rangle \\
 \begin{array}{l} |000\rangle \\ |001\rangle \\ |010\rangle \\ |011\rangle \\ |100\rangle \\ |101\rangle \\ |110\rangle \\ |111\rangle \end{array} & \left[\begin{array}{cccccccc}
 1/4 & & & & & & & & \\
 & -1/4 & 1/2 & & & & & & \\
 & 1/2 & -1/4 & & & & & & \\
 & & & 1/4 & & & & & \\
 & & & & 1/4 & & & & \\
 & & & & & -1/4 & 1/2 & & \\
 & & & & & 1/2 & -1/4 & & \\
 & & & & & & & & 1/4
 \end{array} \right] \begin{array}{l} v_0 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \end{array}
 \end{array}$$

(b) 第2項 ($I \otimes \mathcal{H}_{(2,3)}$) v 図2 量子 Heisenberg 模型のハミルトニアン行列とベクトルの積 ($L=3$ の場合)

および $|010\rangle, |011\rangle$ であり, 上位2ビットをビット反転した値になっていることが確認できる. 式(10)の第2項に対応する図2(b)では $(i, j) = \langle 2, 3 \rangle$ であるから, 行添字を表すケットベクトルの下位2ビットに着目して同様の考察を行えばよい.

6 まとめ

本稿では, 以下の解説を行った.

1. 量子 Heisenberg 模型を紹介し, このハミルトニアン行列が疎行列であることを示した.
2. 疎行列の固有値解法として, 三重対角化を経由する Lanczos 法を紹介した. この三重対角化の過程では, 固有値計算を行う行列とベクトルの積が用いられる.
3. 量子 Heisenberg 模型のハミルトニアン行列に特化した, ビット演算を用いた行列・ベクトル積の計算法を述べた.

最後に, 我々が開発しているパッケージについて触れたい. 3.1節で述べたように, 我々は密行列に対する固有値ソルバに対する統一的なインターフェース Rokko [5] を開発し, 公開している. 本稿で述べた量子 Heisenberg 模型のハミルトニアン行列に対する固有値計算は, Rokko 中のサンプルプログラムとして含まれている. また, Rokko を土台にした量子格子模型の数値対角化(厳密対角化)パッケージ Barista [6] の開発にも取り組んでいる. Barista は, 量子統計物理のシミュレーションのためのパッケージの集合体である ALPS [18] と連携し, 様々な量子格子模型のハミルトニアン行列を入力ファイルで指定できるようにして

いる。

謝辞

RIMS 研究集会「高次元量子トモグラフィにおける統計理論的なアプローチ」で発表する機会を与えていただいた研究代表者の田中冬彦氏に感謝する。

参考文献

- [1] 西森秀稔. 量子スピン系の対角化プログラム TITPACK Ver.2. 物性研究, Vol. 56, No. 5, pp. 494-565, August 1991.
- [2] 西森秀稔. TITPACK2 Ver.2 ソースコード・マニュアル. <http://www.stat.phys.titech.ac.jp/nishimori/titpack2/index-e.html>.
- [3] 利根川孝, 鎗木誠, 西野友年. $S = 1$ スピン鎖の対角化パッケージ KOBEPACK. http://quattro.phys.sci.kobe-u.ac.jp/Kobe_Pack/Kobe_Pack_J.html, 1992.
- [4] Joerg Schulenburg. SPINPACK. <http://www-e.uni-magdeburg.de/jschulen/spin/>, 2008.
- [5] 坂下達哉, 五十嵐亮, 大久保毅, 藤堂真治. 並列固有値ソルバの統一的インターフェース Rokko. <https://github.com/t-sakashita/rokko.git>, 2012.
- [6] 坂下達哉, 五十嵐亮, 大久保毅, 藤堂真治. 統一的インターフェースを用いた量子格子模型の厳密対角化パッケージ Barista. <https://github.com/t-sakashita/barista.git>, 2012.
- [7] 宮下精二. 量子スピン系. 岩波書店, 2006.
- [8] Hidetoshi Nishimori and Gerardo Ortiz. *Elements of Phase Transitions and Critical Phenomena*. Oxford University Press, 2011.
- [9] Gene H. Golub and Charles F. Van Loan. *Matrix Computation*. The Johns Hopkins University Press, 4th edition, 2012.
- [10] LAPACK (Linear Algebra PACKage). <http://www.netlib.org/lapack/>.
- [11] ScaLAPACK (Scalable Linear Algebra PACKage). <http://www.netlib.org/scalapack/>.
- [12] Jack Poulson, Matthias Petschow, and Paolo Bientinesi. Distributed-memory dense linear algebra Elemental. <http://code.google.com/p/elemental/>.
- [13] 理化学研究所計算科学研究機構. 高性能固有値ソルバ EigenExa. <http://www.aics.riken.jp/labs/lpnctr/EigenExa.html>.
- [14] ELPA Consortium. ELPA (Eigenvalue solvers for Petaflop Applications). <http://code.google.com/p/elemental/>.
- [15] SLEPc (Scalable Library for Eigenvalue Problem Computations). <http://www.grycap.upv.es/slep/>.
- [16] Trilinos project. <http://trilinos.org>.
- [17] 宮下精二. 数値計算. 朝倉書店, 2002.
- [18] ALPS project (Algorithms and Libraries for Physics Simulations). <http://alps.comp-phys.org/>.
- [19] 坂下達哉, 五十嵐亮, 大久保毅, 藤堂真治. 並列固有値計算ライブラリの統一的インターフェースの開発とベンチマーク. HPC 研究発表会, May 2013.