Application Of Advanced Diagonalization Methods To Quantum Spin Systems

By

Jieyu Wang

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science (MSc) in Computational Sciences

The School of Graduate Studies
Laurentian University
Sudbury, Ontario, Canada

Thesis Defence Committee/COMITÉ DE SOUTENANCE DE THÈSE

Laurentian Université/Université Laurentienne
School of Graduate Studies/École des études supérieures

Title of Thesis Titre de la thèse	Application Of Advanced Diagonalization Methods To Quantum Spin Systems
Name of Candidate Nom du candidat	Wang, Jieyu
Degree Diplôme	Master of Science
Department/Program Département/Programme	Computational Sciences
Date of Defence Date de la soutenance	March 28, 2014

APPROVED/APPROUVÉ

Thesis Examiners/Examinateurs de thèse:

Dr. Ralf Meyer
(Supervisor/Directeur de thèse)

Dr. Gennady Chitov
(Committee member/Membre du comité)

Dr. Abdellatif Serghini
(Committee member/Membre du comité)

Dr. William Atkinson
(External Examiner/Examinateur externe)

Approved for the School of Graduate Studies
Approuvé pour l'École des études supérieures
Dr. David Lesbarrères
M. David Lesbarrères
Director, School of Graduate Studies
Directeur, École des études supérieures

ACCESSIBILITY CLAUSE AND PERMISSION TO USE

I, **Jieyu Wang**, hereby grant to Laurentian University and/or its agents the non-exclusive license to archive and make accessible my thesis, dissertation, or project report in whole or in part in all forms of media, now or for the duration of my copyright ownership. I retain all other ownership rights to the copyright of the thesis, dissertation or project report. I also reserve the right to use in future works (such as articles or books) all or part of this thesis, dissertation, or project report. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that this copy is being made available in this form by the authority of the copyright owner solely for the purpose of private study and research and may not be copied or reproduced except as permitted by the copyright laws without written authority from the copyright owner.

Abstract

Quantum spin models play an important role in theoretical condensed matter physics and quantum information theory. One numerical technique that is frequently used in studies of quantum spin systems is exact diagonalization. In this approach, numerical methods are used to find the lowest eigenvalues and associated eigenvectors of the Hamilton matrix of the quantum system. The computational problem is thus to determine the lowest eigenpairs of an extremely large, sparse matrix.

Although many sophisticated iterative techniques for the determination of a small number of lowest eigenpairs can be found in the literature, most exact diagonalization studies of quantum spin systems have employed the Lanczos algorithm. In contrast to this, other methods have been applied very successfully to the similar problem of electronic structure calculations. The well known VASP code for example uses a Block Davidson method as well as the residual-minimization – direct inversion of the iterative subspace algorithm (RMM-DIIS).

The Davidson algorithm is closely related to the Lanczos method but usually needs less iterations. The RMM-DIIS method was originally proposed by Pulay and later modified by Wood and Zunger. The RMM-DIIS method is particularly interesting if more than one eigenpair is sought since it does not require orthogonalization of the trial vectors at each step.

In this work I study the efficiency of the Lanczos, Block Davidson and RMM-DIIS method when applied to basic quantum spin models like the spin-1/2 Heisenberg chain, ladder and dimerized ladder. I have implemented all three methods and are currently applying the methods to the different models. In our presentation I will compare the three algorithms based on the number of

iterations to achieve convergence, the required computational time.

An Intel's Many-Integrated Core architecture with Intel Xeon Phi coprocessor 5110P integrates 60 cores with 4 hardware threads per core was used for RMM-DIIS method, the achieved parallel speedups were compared with those obtained on a conventional multi-core system.

Acknowledgements

At first, I would like to acknowledge my supervisor Dr. Ralf Meyer, with whom I worked together on this research. He led me to this scientific field, and helped me to improve program. The algorithms became more efficient through his guidance.

This work has been financially supported by Laurentian University and the *Natural Science and Engineering Research Council of Canada* (NSERC). I gratefully acknowledge generous allocation of computer time on the facilities of the *Shared Hierarchical Academic Research Network* (SHARCNET) and Compute/Calcul Canada (https://computecanada.ca/)

Table of Contents

Abbreviations

COO	Coordinate format
CSR	Compressed Sparse Row format
NNZ	Number of Non-Zero element(s)
MGS	Modified Gram-Schmidt orthogonalization procedure
BKDV	Block Davidson
RMM-DIIS/ RMM	Residual Minimization Method - Direct Inversion in the Iterative Subspace
Chain(*N*)	Heisenberg Chain model (with *N* spins)
Ladder2/ Ladder(*N*)	Two-leg Heisenberg ladder model (with *N* spins)
Ladder 215C(*N*)	Two-leg Dimerized Heisenberg ladder COLUMNAR model with dimerization parameter = 0.5 (with *N* spins)
Ladder 215S(*N*)	Two-leg Dimerized Heisenberg ladder STAGGERED model with dimerization parameter = 0.5 (with *N* spins)
Ladder 315C(*N*)	Three-leg Dimerized Heisenberg ladder COLUMNAR model with dimerization parameter = 0.5 (with *N* spins)
Ladder 315S(*N*)	Three-leg Dimerized Heisenberg ladder STAGGERED model with dimerization parameter = 0.5 (with *N* spins)
Xeon Phi	An Intel's Many-Integrated Core architecture with Intel Xeon Phi coprocessor 5110P integrates 60 cores with 4 hardware threads per core.
Odin	A machine with 2×Intel X5650 Xeon processor（2.67GHz） and 12GB of DDR3-1333 RAM,

List of Tables

List of Figures

1 Introduction

The invention of electronic computers has revolutionized all areas of sciences. The application of computational techniques has opened the door to new approaches and made it possible to study systems of unprecedented complexity. Many important discoveries in biology, chemistry, physics, materials science, etc. have been made with the help of modern computers. This development is ongoing and the invention of new computing technology and better computational techniques will fuel scientific discoveries for the foreseeable future.

One area of science that has immensely benefited from computational techniques is quantum mechanics. Very few quantum problems can be solved exactly with analytical methods. Other models can only be studied with the help of approximations or with computational techniques.

Quantum spin models are a large class of quantum models that play an important role in theoretical condensed matter physics and quantum information theory. These models can be used to study certain magnetic materials as well as general properties of quantum systems. Improved computational techniques might make it possible to study larger systems or systems that currently cannot be studied computationally and could lead to new insights in this fascinating branch of physics.

One computational technique that is often used to study quantum spin models is called exact diagonalization. In this approach, numerical methods are used to determine the

lowest eigenvalues and associated eigenvectors of the Hamilton matrix describing the model Ref. [30]. What makes this approach computationally very challenging is the fact that the Hamilton matrix of quantum spin models growth exponentially with the size of the system. The main computational problem is thus to find a few of the lowest eigenvalues and associated eigenvectors (eigenpairs). Eigenpairs rather than eigenvalues are required in many cases since they allow the calculation of a wider spectrum of physical properties.

The exact diagonalization method has been used to study the properties of a variety of different quantum spin models. Examples of studies of one-dimensional models can be found in Ref. [11][14][18][21][24][25]. Two-dimensional quantum models have also been studied with this method (see e.g. Ref. [1][2][13][15]).

In earlier studies (e.g. Ref. [11][14]) the size of the systems that could be studied was extremely low (less than 20 spins) since technical limitations prohibited the studying of systems with larger Hamilton matrices. More recently, system with 28 – 40 spins have been studied Ref. [1][2][13][15].

Many exact diagonalization studies of quantum spin systems use the Lanczos method Ref. [10] for the determination of the lowest eigenvalues and eigenvectors. Many other methods for this problem can however be found in the literature (see e.g. Ref. [10]). Some of these methods might be advantageous, in particular if one needs to determine not only the lowest eigenvalue but several of the low-lying eigenpairs.

The first goal of this work is to test whether the Block-Davidson algorithm Ref. [6] and the Residual Minimization — Direct Inversion of the Iterative Subspace (RMM-DIIS)

method Ref. [23] can be employed successfully in exact diagonalization studies of quantum spin systems. Both methods have been used successfully in electronic structure calculations, which require the determination of the lowest eigenvalues of large matrices. The Block-Davidson method and the RMM-DIIS algorithm are both implemented in the well-known VASP code Ref. [16].

The Block-Davidson algorithm Ref. [6] is closely related to the Lanczos method but it usually requires less iterations. Conceptually, both of these methods search for the eigenvector with the lowest eigenvalue. If several of the lowest eigenvalues are sought, the trial states must be reorthogonalized at every step. Otherwise, the different trial vectors may converge towards each other. For large matrices, this reorthogonalization procedure is computationally rather intensive and may dominate the total computing time.

The RMM-DIIS method does not attempt to minimize the eigenvalue but the norm of the residual vector. For this reason, trial vectors will not converge towards each other if the initial states are sufficiently separated in the state space. The RMM-DIIS method does not require a reorthogonalization at every step. Even if this method requires more iterations, this method may take substantially less computing time Ref. [16]. For quantum spin systems this effect may even be larger than for electronic structure calculations due to the larger matrix sizes in this case.

In order to test the Block-Davidson method and the RMM-DIIS algorithm in the context of quantum spin systems, computer programs for these two methods have been developed and applied to a number of spin-1/2 Heisenberg systems. The aim of these calculations was to find whether these methods do work properly for such typical

problems and to assess their potential. The goal was not immediately to perform calculations of larger systems than was possible previously.

As pointed out above, scientific progress is fueled by advances in computational methods as well as in computing technology. In recent times, increases in computing power were mainly achieved through the introduction of multi-core processors that integrate several processor cores on a single chip. In order to take advantage of modern computers, programs must be able to run efficiently in parallel on such processors. This can be problematic if a problem uses large amounts of memory like it is the case for the quantum spin systems studied in this thesis. In such cases, the computing time may be limited by memory access speed and the usage of more CPU core might not result in substantial time savings.

The next generation of processors for computing problems will be so-called many-core processors that integrate hundreds or even thousands of CPU cores on a single chip. An early example of this forthcoming technology is the Intel Xeon Phi coprocessor Ref. [3]. Current Xeon Phi processors provide 60 CPU cores with 4 hardware threads per core and a high memory band width on a single chip. The second goal of this thesis is to test whether this new architecture provides a suitable basis for exact diagonalization studies of quantum spin systems. To this end, the program for RMM-DIIS method has been employed on an Xeon Phi coprocessor and the achieved parallel speedups were compared with those obtained on a conventional multi-core system.

2 Backgrounds

2.1 Quantum Spin Systems

2.1.1 Introduction

The study of quantum spin systems or quantum spin models is an important branch of theoretical physics. The origins of these systems lies in the field of magnetism. The first quantum spin systems where invented as models to study the properties of insulating magnetic materials Ref. [22].

Today, quantum spin models are still used to study the behavior of magnetic materials. (see e.g Ref. [1][2][13][11][15][14][18][21][24][25])In addition to this, they are also used as basic models to understand fundamental phenomena of quantum mechanics, like quantum phase transitions. Finally, quantum spin systems play a role in the field of quantum computing since interacting spin systems can be used to realize quantum computing devices.

In physics, the spin is an intrinsic property of elementary particles, for example electrons. The spin of a particle behaves like a tiny angular momentum and it has an associated magnetic moment. An important property of spins is that it is quantized. Each spin has an associated quantum number *S*. which is a positive integer or half-integer number. A spin can only be in one of $2S + 1$ states that are often characterized by $S^z = -S, -S + 1, \ldots, S - 1, S$. An important case is $S = ½$ when the spin can only take one

of two states: $S^z = +½, -½$.

A quantum spin system is a system that contains an ensemble of *N* spins that interact with each other through some kind of interaction. An example for a class of quantum spin models that I use in this work is the antiferromagnetic Heisenberg model with spin $S = ½$. There are however many more types of quantum spin models.

In the Heisenberg model, the spins of the system are immobile and they interact through an exchange interaction whose strength is determined by an interaction constant *J*. In principle each spin in a Heisenberg model might interact with all other spins. In practice, however, one often uses the so-called nearest neighbor approximation. This approximation restricts the interaction so that each spin interacts only with its closest neighbours. If the interaction between two spins is not zero, one says that there is a bond between the two spins.

2.1.2 Numerical Methods for Quantum Models

Theoretical physicists use many different methods to study quantum spin systems. In some cases, it is possible to obtain exact information about the properties of a model through analytical calculations. If this is not possible, approximative analytical methods or numerical methods can be employed.

Three important numerical techniques for the studying of quantum spin models are Density Matrix Renormalization Group Ref. [27], Quantum Monte Carlo Simulation (see, e.g., Ref. [9]), and Exact Diagonalization (see, e.g., Ref. [30]). This work focuses

on the latter method which is described in the following paragraphs.

In quantum mechanics, a system is characterized by a linear operator (Hamilton operator) which acts on elements of a Hilbert space. The elements of the Hilbert space correspond to the possible states of the system. For a spin system with a finite number of spins *N*, the Hilbert space becomes a finite-dimensional vector space and the Hamilton operator a quadratic matrix.

The exact diagonalization method uses numerical techniques to find the eigenvalues and eigenvectors of the Hamilton matrix of a quantum system. The reason for this is that the physical properties of the model described by the Hamilton matrix can be understood from the eigenvalues and eigenvectors of the matrix. For many applications a sufficient understanding of the systems can be obtained from the lowest eigenvalues and their eigenvectors.

While conceptually easy to understand, the practical use of the exact diagonalization method faces enormous technical problems. The primary reason for these difficulties is the large size of the Hamilton matrices even for relatively small systems. For a spin $-½$ Heisenberg model, i.e. a Heisenberg model with $S = ½$, with *N* spins, the dimension of the state vector space is 2^N. This means that even for a modest system with 20 spins the vector space is of dimension 1,048,576 and the Hamilton matrix has more than a trillion elements.

A numerical treatment of the Hamilton matrices of quantum spin systems is only possible since only a small fraction of the matrix elements of these matrices are non-zero. It is therefore possible to use so-called sparse-matrix techniques that store only the

non-zero elements of the matrix.

A spin-½ Heisenberg model is determined by the number of bonds. Each bond, i.e. each pair of spins with a non-zero interaction constant *J*, adds 2^{N-1} off-diagonal elements to the matrix. In addition to this, most of the diagonal elements are non-zero. If one neglects the small number of diagonal elements that might be zero, the fraction of matrix elements that are non-zero in a spin$-½$ Heisenberg model with b bonds is given by $(b+2)2^{-(N+1)}$.

The size of eigenvalue problem to be handled in the exact diagonalization method can be further reduced if one makes use of the symmetries of the problem. Heisenberg models contain a number of symmetries that can be used to block diagonalize the Hamilton matrix. Each of the blocks can be treated separately which leads to a substantial reduction of the computational effort.

In this work the only symmetry that is used to reduce the problem size is the conservation of the total spin. This symmetry is related to the fact that when the Hamilton matrix is applied, the number of spins N_+ that are in the $+½$ state and the number of spins N_- that are in the $-½$ state does not change. This symmetry makes it possible to classify the eigenstates of the Hamilton matrix by the difference $M = N_+ - N_-$. In this work, only the subspace of states with $M = 0$ has been considered. The dimension of this this subspace is given by equation $\left(\frac{N!}{(N/2)!(N/2)!}\right)$

2.1.3 Quantum Spin Systems Considered in this Work

Three different types of antiferromagnetic Heisenberg models with $S = ½$ are used in this work. The term antiferromagnetic means that the interaction constant between two spins favors the antiparallel alignment: one spin is in the $+½$ state while the other is in the $-½$ state. The four models are called the Heisenberg chain, the Heisenberg ladder and the dimerized Heisenberg ladder.

The Heisenberg chain is a simple one-dimensional arrangement of the spins. Each spin interacts only with its immediate neighbours to the left and right.

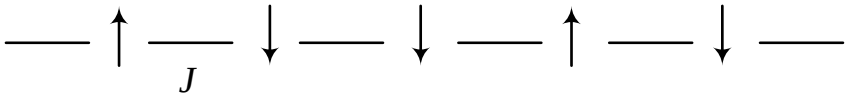


Figure 2.1 Heisenberg Chain model

Example for exact diagonalization studies of spin chains can be found in Ref. [11][14]. In this work, periodic boundary conditions are applied to all models. This means that the spin at the left end of the chain interacts with the spin at the right end of the chain so that the chain effectively has no end. All bonds in the chain use the same interaction strength $J = 1$.

The Heisenberg ladder is a generalization of the Heisenberg chain [2]. A k-leg ladder consists of k parallel Heisenberg chains. These chains are called the legs of the ladder. In addition to the interaction along the legs, each spin interacts with its neighbours in the adjacent chains.

As in the case of the chain, periodic boundary conditions are applied. The Heisenberg ladder, is characterized by two interaction constants. Interactions among neighbours along the legs use $J_{\|}$ whereas the bonds along the rungs of the ladder use $J\perp$. In this work $J_{\|} = 1$ and $J\perp = \kappa$ is used.

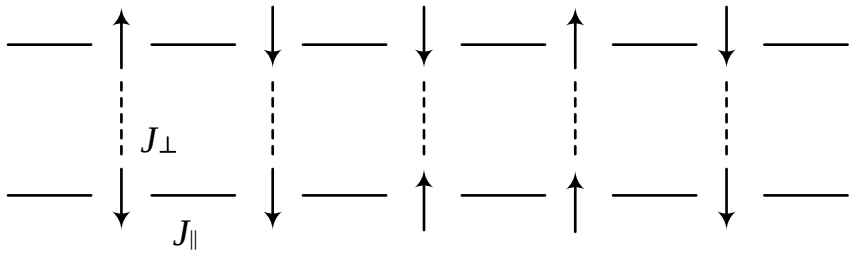


Figure 2.2 Heisenberg Ladder model

The dimerized Heisenberg ladder model Ref.[19][29] uses the same geometric arrangement of the spins as the ladder model. The difference is that the interaction strength of the bonds along the ladder legs is modulated. Two different values of the interaction constant are used in an alternating pattern. The two values used in this work are $J_{||,1} = 3/2$ and $J_{||,2} = 1/2$.

The modulation of the bond strength along the ladder legs allows for two different subtypes of the dimerized ladder. In the columnar model, the strength of the corresponding bonds in all legs is the same. In the staggered model, the pattern is shifted by one bond between the legs.

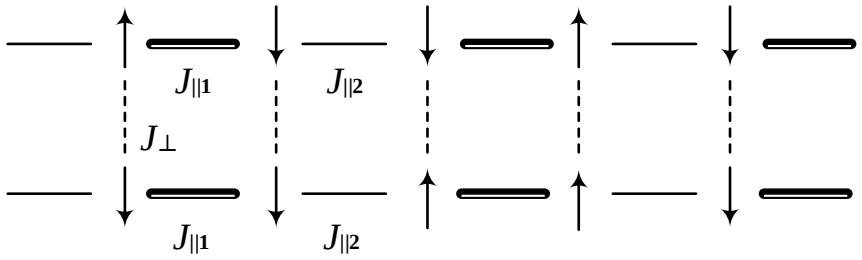


Figure 2.3 Dimerized Heisenberg ladder columnar model

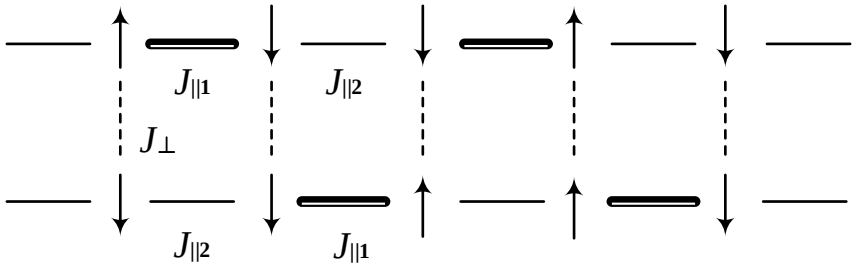


Figure 2.4 Dimerized Heisenberg ladder staggered model

2.2 Sparse matrix storage

2.2.1 Block-diagonalization

Fortunately, however, it is possible to reduce the size of the matrices. The Heisenberg models possess a number of symmetries that can be exploited to perform a block diagonalization of the Hamilton matrix. The block diagonalization divides the Hilbert space into a number of subspaces that do not interact with each other. It is then possible to diagonalize each of these subspaces independently from the others. This reduces the size of the computational problem effectively to the size of the largest subspace.

Depending on the details of the Heisenberg model, multiple symmetries can be exploited to reduce the system size. Some of these symmetries are more complicated than others. For simplicity, in this work only the conservation of the z-component of the total spin S^z has been exploited. After the block diagonalization under this symmetry, the dimension of the largest subspace ($S^z = 0$) is given by $\frac{N!}{(N/2)!(N/2)!}$.The calculations in this work have always been done in the subspace $S^z = 0$.

In Table 2.1 the full systems size and the size of the $S^z = 0$ subspace are compared. The third column (percentage) shows the fraction of the reduced system size compared to the total size. For example, in the case of $N = 26$, the dimension of the $S^z = 0$ subspace is less than 16% of the dimension of the full Hamilton matrix.

Table 2.1 Full size dimension and reduced dimension in different size

N	Full size dimension = N^2	Reduced dimension = $\frac{N!}{(N/2)!(N/2)!}$	Percentage
2	4	2	50.0000%
4	16	6	37.5000%
6	64	20	31.2500%
8	256	70	27.3438%
10	1024	252	24.6094%
12	4096	924	22.5586%
14	16384	3432	20.9473%
16	65536	12870	19.6381%
18	262144	48620	18.5471%
20	1048576	184756	17.6197%
22	4194304	705432	16.8188%
24	16777216	2704156	16.1180%
26	67108864	10400600	15.4981%

2.2.2 Sparse matrix storage (CSR)

The huge size of the Hamilton matrices describing quantum spin systems makes it impossible to directly store the complete matrices except for very small system sizes N. Fortunately, these matrices are very sparse, i.e. the vast majority of elements is zero.

For very sparse matrices, different storage techniques can be used that store only the non-zero elements of the matrix. These techniques require that in addition to the value of the matrix elements, the position of the element in the matrix (row and column index) is stored.

In this work, I use the Compressed Sparse Row (CSR) format Ref. [26]. This is a common scheme that is supported by many libraries. The CSR format stores all elements that belong to the same row in subsequent addresses. This makes is unnecessary to store the row index for each individual element.

The CSR format uses three arrays: value, column and row. The value array stores the values of all non-zero elements of the matrix. The column array stores for each elements its column index. Column and value have clearly the same size. Finally, the row array stores for each row the index of its first non-zero element in the column and value arrays. This array needs only one entry per row.

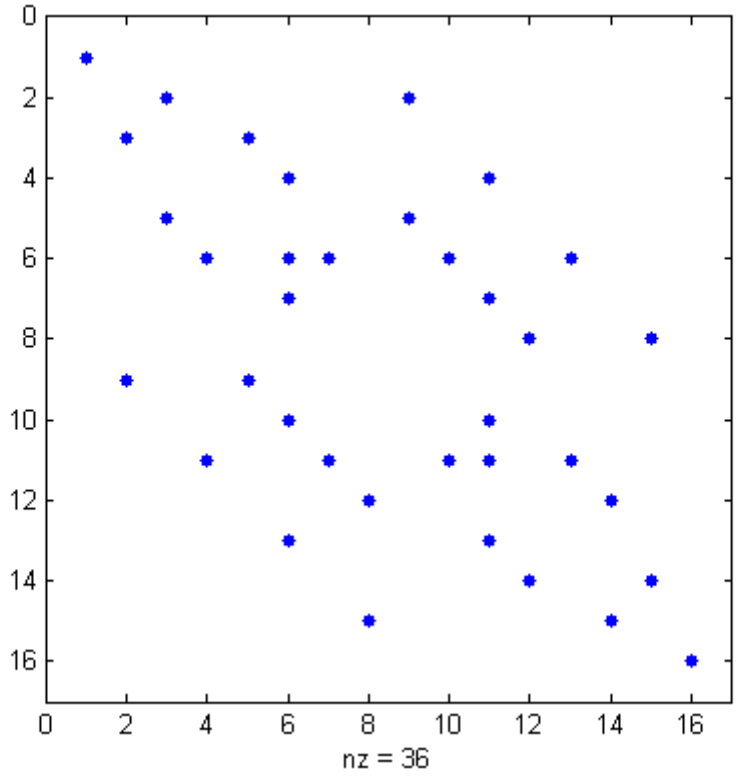


Figure 2.5 Visualize sparsity pattern of Chain(4) model

For example, Chain(4) is a 16×16 matrix, the expression is shown in Eq.(2.1). Figure 2.5 shows how sparse the matrix it is. There are 36 non-zero elements in Chain (4) model, if use full size, it needs store 16×16=196 values and their positions. According to Table 2.2, it needs $36 \times 2 + 16 = 88$ values (include indices) to store Chain(4) model, save rate =88/(196×3) ×100% ≈ 14.97%.

Chain(4)=

$$\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.5 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.5 & 0 & -1 & 0.5 & 0 & 0 & 0.5 & 0 & 0 & 0.5 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0.5 & 0 \\
0 & 0.5 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.5 & 0 & 0 & 0.5 & 0 & 0 & 0.5 & -1 & 0 & 0.5 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0.5 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix} \quad (2.1)$$

Table 2.2 CSR format of chain (4) model

Value	1	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Row	1	2		4		6		8	
Column	1	3	9	2	5	6	11	3	9
Value	0.5	-1	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Row	10					15		17	
Column	4	6	7	10	13	6	11	12	15
Value	0.5	0.5	0.5	0.5	0.5	0.5	0.5	-1	0.5
Row	19		21		23				
Column	2	5	6	11	4	7	10	11	13
Value	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	1
Row	28		30		32		34		36
Column	8	14	6	11	12	15	8	14	16

Table 2.3 Number of non-zero elements with different sizes in Chain, Ladder2, Ladder215C, Ladder 215S, Ladder315C and Ladder315S when $M = 0$

Model(N)	$M = 0$		
size N	Chain	Two-leg Ladder: Ladder2 Ladder215C Ladder215S	Three-leg Ladder: Ladder315C Ladder315S
2	4	4	-
4	22	22	-
6	92	128	104
8	390	550	-
10	1652	2352	-
12	6972	9996	11004
14	29304	42240	-
16	122694	177606	-
18	511940	743600	820820
20	2129556	3101956	-
22	8834696	12899328	-
24	36564892	53495260	59138716
26	151016712	221324768	-

Table 2.3 shows how many non-zero elements in different models with different size when $M = 0$. Number of non-zero elements of Chain model is the least, two-leg Heisenberg Ladder (Ladder2, Ladder215C and Ladder215S) models have same Number of non-zero elements, and three-leg Heisenberg (Ladder315C and Ladder315S) models have the most Number of non-zero elements in these six models.

Form Figure 2.6，we can get there are 212695496 non-zero elements in this model, $\frac{\text{number of non-zero elements}}{\text{number of all elements}} = \frac{212695496}{2^{24}\times 2^{24}} \approx 7.6 \times 10^{-7}$, this ratio shows how sparse the Chain(24) model is, hence using Compressed Sparse Row (CSR) format to compress matrix and store it that is necessary, because CSR can save so much storage.

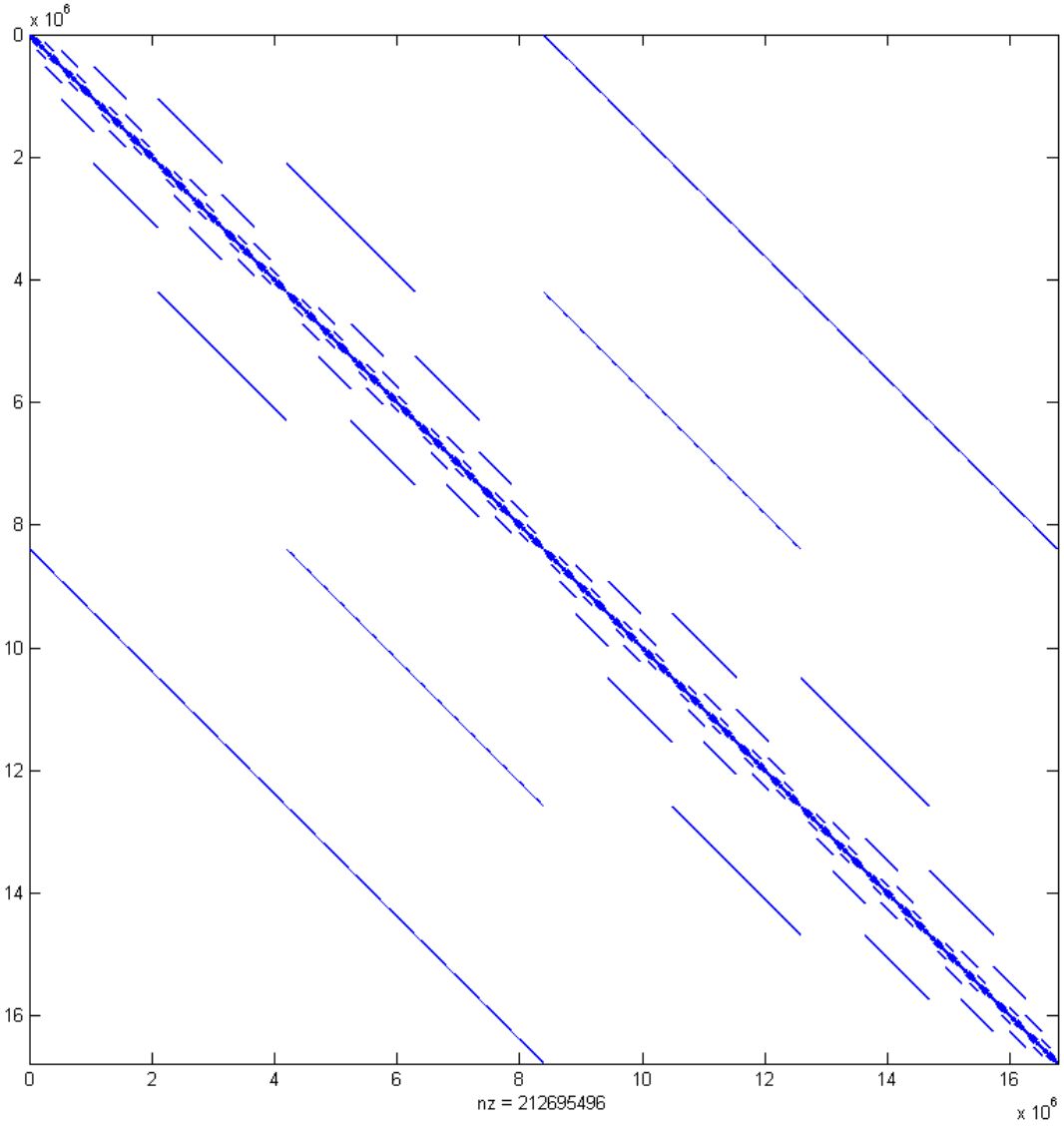


Figure 2.6 Visualize sparsity pattern of Chain(24) model

2.3 **Lanczos method**

Lanczos developed the Lanczos theory in the 1950s Ref. [10], it is suitable for solving the eigenvalues problem of large symmetrical sparse matrices.

The Lanczos algorithm is an iterative method which transforms the matrix A into a series of symmetric (Hermitian) tridiagonal matrices

$$T^k = \begin{bmatrix} \alpha_1 & \beta_1 & 0 & \cdots & 0 \\ \beta_1 & \alpha_2 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \beta_{k-1} \\ 0 & \cdots & 0 & \beta_{k-1} & \alpha_k \end{bmatrix} \tag{2.2}$$

with the help of the orthogonal transformations

$$T^k = {Q^k}^T A Q^k \tag{2.3}$$

where = $Q^k = [q_1, q_2, \ldots q_k]$ is an orthogonal matrix. As *k* increases, the eigenvalues of T^k approximate the eigenvalues of *A*. In the limiting case $k = N$, T^N is similar to *A* so that the eigenvalue spectrums of the two matrices are identical (apart from numerical errors).

The algorithm starts with a randomly chosen normalized vector q_1. At the k^{th} iteration, q_{k+1} is found by applying the matrix *A* to the vector q_k and orthogonalization of the result with respect to the previous vectors. This orthogonalization procedure would normally be a computationally expensive procedure and it would require the storage of all previous vectors. The reason for the success of

the Lanczos method is, however, that it can be shown that for symmetric (Hermitian) matrices it is sufficient to orthogonalize with respect to the last two vectors. An effective method to do this is shown in Algorithm 1 Ref. [10]. It should be mentioned however, that numerical errors lead to a degradation of the orthogonality of the vectors.

In practice, the number of iterations performed is much smaller than the dimension of the matrix A. When the algorithm stops, at iteration k the lowest eigenvalues of the matrix T^{k+1} are used as approximations for the lowest eigenvalues of A.

Since the vectors q_k are obtained by successive application of the matrix A and subsequent orthogonalization, the space spanned by the vectors $[q_1, q_2, \dots q_k]$ is the Krylov subspace

$$K(A, q_1, k) = span\{q_1, Aq_1, A^2q_1, \dots, A^{k-1}q_1\} \tag{2.4}$$

The eigenvalues found by the Lanczos algorithm are therefore the lowest eigenvalues of the matrix A projected into the subspace $K(A, q_1, k)$.

Algorithm 1 Lanczos method

Input: matrix A, where A is a $n \times n$ symmetrical sparse matrix.
vector w, where w has n elements with rand [-1,1].

Output: The lowest k eigenvalues λ_m (for $m = 1{:}\,k$)

1. $v_{1:n} = 0;\ \beta_1 = 1; k = 1; |w>= \frac{|w>}{||\,|w>\,||_2}$
2. **while** $\beta_k \neq 1$
3. **if** $k \neq 1$
4. **for** $i = 1{:}\,n$
5. $t = w_i;$
6. $w_i = \frac{v_i}{\beta_k};$
7. $v_i = -\beta_k t;$
8. **end for**
9. **end if**
10. $|v>= |v> + A|w>$
11. $k = k + 1;$
12. $\alpha_k =< w|v >;$
13. $|v>= |v> - \alpha_k|w>;$
14. $\beta_k = \left|\right|\,|v>\,\left|\right|_2$
15. **end while**
16. Make $T = \begin{bmatrix} \alpha_1 & \beta_1 & 0 & \cdots & 0 \\ \beta_1 & \alpha_2 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \beta_{k-1} \\ 0 & \cdots & 0 & \beta_{k-1} & \alpha_k \end{bmatrix}$
17. Compute eigenvalues of T ;

2.4 Davidson and Block Davidson method

The Davidson method is closely related to the Lanczos algorithm. Like the latter method, they are iterative methods that project the matrix A onto a sequence of subspaces of increasing dimensions Ref. [23]. The Davidson method is particularly suited for diagonally dominant matrices. Contrary to the Lanczos algorithm, the Davidson method requires at each step an explicit orthogonalization of the new vector with respect to the previous iterative subspace.

The Block Davidson method is a generalization of the Davidson method that allows the simultaneous search for the l lowest eigenpairs.

2.4.1 Davidson method

Davidson originally published his method in 1975 as a method to find the lowest eigenstates in quantum chemistry problems Ref. [8]. Later, the method has been improved by Liu Ref. [17] and Murray et al. Ref. [20].

At each iteration, the Davidson method performs a Rayleigh-Ritz step that determines the lowest eigenpair of the projected matrix in the current iterative subspace (Step 3-7 in Algorithm 2). The algorithm than calculates the residual vector $|r^k>= (A-\lambda^k I)|z>$ for this eigenpair. If the norm of this vector is below the given convergence criterion “epsilon”, the algorithm stops.

If the residual vector is above the convergence criterion, a pre-conditioner C^k is applied to the residual vector. The resulting vector is than orthogonalized with respect to all previous vectors and added to the iteration space V^{k+1}. It is the preconditioning step which separates the Davidson method from the Lanczos algorithm Ref.[7]. In this work, I use the diagonal pre-conditioner proposed in Ref. [8] (step 11 in Algorithm 2). Other pre-conditioners could be used here and might give better results. The application of the pre-conditioner destroys the orthogonality properties of Lanczos algorithm. This is the reason at each step the new search vector must be orthogonalized with respect to the complete search space.

Algorithm 2 Davidson method

Input: matrix A, where A is a $n \times n$ symmetrical sparse matrix.
matrix D, where D is main diagonal of A, same size as A.
integer m, where m is maximum number of iteration.
vector $|v^1>$, an initial eigenvector with rand [-1,1].
tolerance ε, where is a threshold, if $||\,|r>||_2 < \varepsilon$ then break.

Output: The lowest eigenpair $\lambda^k, |a^k>$

1. Pick up an initial unit vector $|v^1>$
2. $|v^1> = \dfrac{|v^1>}{||\,|v^1>||_2}$; $V^1 = |v^1>$;
3. **for** $k = 1{:}\,m$
4. Compute the Rayleigh matrix $H^k = {V^k}^T A V^k$;
5. Compute the lowest eigenpairs $(\lambda^k, |a^k>)$ of H^k ;
6. Compute the Ritz vector $|z> = V^k|a^k>$;
7. Compute the residual vector $|r^k> = (A - \lambda^k I)|z>$;
8. **if** $||\,|r^k>\,||_2 < \varepsilon$
9. $convergence = 1$ and exit
10. **end if**
11. Compute $C^k = (D - \lambda^k I)^{-1}$;
12. Compute the new directions $|t^k> = C^k|r^k>$
13. $V^{k+1} = MGS([V^k, |t^k>])$
14. **end for**
15. Set $|v^1> = |z>$, go to 2

2.4.2 Block Davidson

The Block Davidson method is a generalized formulation of Davidson's method. This version of the algorithm attempts to determine simultaneously the lowest l eigenpairs of A.

In this work, I follow the formulation of the Block Davidson algorithm described by Crouzeix et al. in Ref. [6] (see algorithm 3). The principle difference between algorithm 2 and algorithm 3 is that the blocked version calculates in each iteration the lowest l eigenpairs and residual vectors of the matrix projected onto the search space. After the convergence check, if the norm of residual vector is below the given convergence criterion ε, the algorithm stops, else all l residual vectors are preconditioned and added to the subspace.

Another difference to the original Davidson method is a limitation of the dimension of the subspace. Since it is too costly to store all previous vectors, the algorithm enforces a maximum dimension b for the iterative subspace. If adding of the new vectors exceeds this size, the algorithm restarts with the lowest l Ritz vectors and the preconditioned residual vectors.

Algorithm 3 Block Davidson

Input: matrix A, where A is a $n \times n$ symmetrical sparse matrix.
matrix D, where D is main diagonal of A, same size as A.
integer l, where l is number of the lowest eigenpairs need to be calculated.
matrix V^1, where V^1 is a random $n \times l$ matrix, all elements of V^1 within range [-1,1], make $V^1 = MGS(V^1)$ as initial eigenvectors of A.
integer m, where m is maximum number of iteration.
integer b, which b limits the dimension of the basis.
tolerance ε, where is a threshold, if $|| \, |r> \, ||_2 < \varepsilon$ then break.

Output: the lowest l eigenpairs $\lambda_j, |a_j>$ (j = 1,2,…,l)

1. Pick up the initial orthonormal basis V^1;
2. **for** $k = 1{:}m$
3. Compute the matrix $W^k = AV^k$;
4. Compute the Rayleigh matrix $H^k = {V^k}^T W^k$;
5. Compute the lowest l eigenpairs (λ_i^k, a_i^k) of H^k; for $i = 1{:}l$
6. Compute the Ritz vector $|z_i^k> = V^k |a_i^k>$; for $i = 1{:}l$
7. The residual vector $|r_i^k> = \lambda_i^k |a_i^k> - W^k |a_i^k>$; for $i = 1{:}l$
8. **If** $\max_{i=1:l} || \, |r_i^k> \, ||_2 < \varepsilon$
9. $convergence = 1$ and exit
10. **end if**
11. Compute $C_i^k = (D - \lambda_i^k I)^{-1}$; for $i = 1{:}l$
12. Compute the new directions $|t_i^k> = C_i^k |r_i^k>$ for $i = 1{:}l$
13. **If** $dim(V^k) < b$
14. $V^{k+1} = MGS(V^k, |t_1^k>, |t_2^k> \cdots |t_l^k>)$
15. **else**
16. $V^{k+1} = MGS(|z_1^k>, |z_2^k> \cdots |z_l^k>, |t_1^k>, |t_2^k> \cdots |t_l^k>)$
17. **end if**
18. **end for**

2.5 RMM-DIIS method

The Residual Minimization Method - Direct Inversion in the Iterative Subspace method (RMM-DIIS method) was proposed by P éter Pulay in 1980 Ref. [23], Wood and Zunger were the first to apply the method to the eigenvalue problem in electronic structure calculations Ref. [31].

The residual minimization scheme is based on the idea to minimize the norm of the residual vector in the iterative subspace generated by the previous iterations. The advantage of this is that if the trial vectors are sufficiently separated, they will converge to different eigenstates independently. This is in contrast to methods which attempt to minimize the Rayleigh quotient (like the Davidson algorithms) where the trial states must be explicitly orthogonalized in order to avoid that all trial vectors converge towards the eigenvector of the lowest eigenvalue. This is an important difference since the Gram-Schmidt orthonorrmalization procedure which requires $C(n) = \Theta(n^3)$ operations, dominates the execution time of the Block Davidson algorithm for larger matrices. Since the residual minimization method does not require explicit orthogonalization at every step it has the potential to be faster than the Block Davidson method even if the method itself is more complex.

In the RMM-DIIS method, the residual vector is used:

$$|r^k >= (A - \lambda^k I)\,|a^k > \tag{2.5}$$

With each iteration, the residual vector $|r^k >$ will converge towards zero. The λ^k is the eigenvalue of A when the residual vector$|| \, |r^k > ||_2 = 0$, or when $|| \, |r^k > ||_2$ is less than a threshold ε ($\varepsilon = 0$ ideally).

λ^k is an approximate eigenvalue, it can be calculated directly by the Rayleigh quotient Eq.(2.6), using Hamiltonian matrix A and approximate eigenvector $|a^k>$.

$$\lambda^k = \frac{< a^k|A|a^k >}{< a^k|a^k >} \tag{2.6}$$

Most iterative methods use an increment vector $|\delta x^k>$ to obtain the approximate eigenvector at the next iteration.

$$|a^{k+1}> = |a^k> + |\delta x^k> \tag{2.7}$$

Ideally, the Eq. (2.8)would bring the residual vector to zero, getting $|r^{k+1}>$ to zero yields.

$$\begin{aligned} |r(\lambda, |a^k + \delta x>)> &= (A - \lambda I)|a^k + \delta x^k> \\ &= (A - \lambda I)|a^k> + (A - \lambda I)\,|\delta x^k> \\ &= 0 \end{aligned} \tag{2.8}$$

The formal solution of this equation is:

$$|\delta x^k> = \frac{-(A - \lambda I)|a^k>}{(A - \lambda I)} \tag{2.9}$$

Unfortunately, however there is no straightforward way solve Eq.(2.9), because $(A - \lambda I)^{-1}$ needs a matrix inversion operating, which may consume more time than traditional way to solve eigenproblem and λ is not necessarily the exact eigenvalue. Instead, the RMM-DIIS method sets the increment vector $|\delta x^k>$ along the direction of the residual vector $|r^{k-1}>$ ($|\delta x^0> = |a^0>$).

$$|\delta x^k> = \begin{cases} |r^{k-1}>, & k > 0 \\ |a^k>, & k = 0 \end{cases} \tag{2.10}$$

The vector at the next iteration is written as a linear combination of increment vector at the current iterations:

$$|a^{k+1}>=\sum_{j=0}^{k}\alpha_j|\delta x^j> \tag{2.11}$$

The coefficient α_j are obtained from the condition that the square of the norm of the residual vector $\rho^2 = ||\ |r^{k+1}>\ ||_2^{\ 2}$ is minimal. This leads to the condition Ref.[31].

$$\begin{aligned}\frac{\delta\rho^2}{\delta\alpha_j^*} &= \frac{\delta}{\delta\alpha_j^*}\frac{\sum_{g,h=0}^{k}\alpha_g^*\alpha_h < r^g|r^h>}{\sum_{g,h=0}^{k}\alpha_g^*\alpha_h < \delta x^g|I|\delta x^h>}\\ &= \frac{\delta}{\delta\alpha_j^*}\frac{\sum_{g,h=0}^{k}\alpha_g^*\alpha_h < \delta x^g|(A-\lambda^{old}I)|(A-\lambda^{old}I)|\delta x^h>}{\sum_{g,h=0}^{k}\alpha_g^*\alpha_h < \delta x^g|I|\delta x^h>}\\ &= 0\end{aligned} \tag{2.12}$$

α_j (for $j = 0,1,2,\dots,k$) can be regarded as the components of a $(k+1)$dimensional vector $|\alpha>$. The problem of finding α_j is equivalent to finding the eigenvector corresponding to the lowest eigenvalue ρ^2 of the generalized Hermitian eigenproblem:

$$P|\alpha>=\rho^2 Q|\alpha> \tag{2.13}$$

where

$$P_{(g,h)} =< \delta x_i^g\big|\big(A-\lambda_i^k I\big)\big|\big(A-\lambda_i^k I\big)|\delta x_i^h> \tag{2.14}$$

and

$$Q_{(g,h)} =< \delta x_i^g|I|\delta x_i^h> \tag{2.15}$$

The lowest eigenvalue of this problem corresponds to the minim value of ρ^2. Since

P and Q are small $(k+1)\times(k+1)$ symmetric matrices the solution of this generalized eigenproblem requires only a negligible amount of time.

The lowest eigenvalue ρ^2 corresponds to the minimum value of the square of the norm of the residual vector. The components of the corresponding eigenvector $|\alpha>$ can then be used to obtain the next trial eigenvector $|a^{k+1}>$ according to Eq.(2.11). Finally, the new approximate eigenvalue is obtained as in Eq.(2.6):

$$\lambda_{\mathrm{i}}^{k+1} = \frac{< a_i^{k+1}|A|a_i^{k+1} >}{< a_i^{k+1}|a_i^{k+1} >} \tag{2.16}$$

If the new residual vector $|r^{k+1}>$ fulfills the convergence criterion, in other word, if $||\,|r_i^{k+1}>||_2 < \varepsilon$ (for $i = 1{:}\,l$) then the program is stopped. If one of the residual vector $|r_i^{k+1}>$ does not satisfy the criterion, k is increased by one and the procedure is repeated starting with Eq.(2.10).

Algorithm 4 RMM-DIIS method

Input: matrix A, where A is a $n \times n$ symmetrical sparse matrix.
integer l, where l is number of the lowest eigenpairs need to be calculated.
matrix a^0, where $a^0 = [|a_1^0>, |a_2^0>, \dots, |a_l^0>]$,is a random $n \times l$ matrix, and contains l initial eigenvectors of A.
vector λ^0, where $\lambda^0 = [\lambda_1^0, \lambda_2^0, \dots \lambda_l^0]$,$\lambda$ is the eigenvalues corresponding to the eigenvectors in V.
integer m, where m is maximum number of iterations.
tolerance ε, where is a threshold, if $|| \, |r> \, ||_2 < \varepsilon$ then break.

Output: the lowest l eigenpairs $\lambda_i^j, |a_i^j> \; (j = 1,2, \dots, l)$

1. pick input eigenpairs $\lambda_i^0, |a_i^0>$ for $i = 1{:}\, l$
2. let $|\delta x_i^0> = |a_i^0>$
3. Compute the $A|a_i^0>$ and store it;
4. **for** $k = 0{:}\, m$
5. Compute residual vector $|\delta x_i^{k+1}> = A|a_i^k> - \lambda_i^k |a_i^k>$
6. **if** $|| \, |r_i^k> \, ||_2 < \varepsilon$
7. $convergence = 1$ and **exit**
8. **end if**
9. Compute $P_{g.h} = < \delta x_i^g | \left(A - \lambda_i^k I\right) | \left(A - \lambda_i^k I\right) | \delta x_i^h >$ for $g = 0{:}\, k+1$ for $h = 0{:}\, k+1$
10. Compute $Q_{g,h} = < \delta x_i^g | I | \delta x_i^h >$ for $g = 0{:}\, k+1$ for $h = 0{:}\, k+1$
11. Compute the lowest eigenvector $|\alpha>$ in generalized eigenproblem of P and Q;
12. Compute new eigenvector $|a_i^{k+1}> = \sum_{j=0}^{k+1} \alpha_j |\delta x_i^j>$;
13. Compute new eigenvalue $\lambda_i^{k+1} = \dfrac{< a_i^{k+1} | A | a_i^{k+1} >}{< a_i^{k+1} | a_i^{k+1} >}$;
14. store $A|a_i^{k+1}>$
15. **end for**

3 Implementations

In this section, there are four algorithms listed (Block Davidson, RMM-RITZ, and two combination methods: BK-RMM and RMM-RMM) which are implemented. All implementations are achieved by C++, in order to get results more quickly, Intel's Math Kernel Library (MKL) is used in this work frequently, and OpenMP is used to parallelize some vector operations. There are also some calculations and graphs by Matlab.

3.1 Block Davidson

I implemented the Block Davidson algorithm same as Algorithm 3 in page 25.

3.2 RMM-RITZ

The dimension of P and Q are not limited in the original RMM-DIIS method, when the number of iteration grows up, size of P and Q becomes larger, therefore I set a block number b to limit the dimension in my implementation. On the hand setting a block number can save calculations, execution time and reduce memory requirement, on the other hand using block RMM-DIIS may lost some information and obtain a degenerated eigenvector, but normally, it also can reach convergent at final.

Similar to the Block Davidson method, implementation of the RMM-DIIS method require a blocking scheme for practical reasons. Although in theory the dimensions of the matrices P and Q are not limited, it is impractical to store more than a few of

vectors $|\delta x_i>$.

Since the trial vectors are iterated independently (without the need for an orthogonalization step), the RMM-DIIS blocking scheme works a bit different. The algorithm picks one trial vectors and iterates it until the size of the subspace for this vector has reached the block size. Then the procedure is repeated with a different trial vector. After all states have been iterated, a Ritz projection is performed that finds the l lowest eigenvectors in the combined iterative subspace.

In addition to a reduction of the memory size, the Ritz projection has a number of other advantages. First, the Ritz projection drives the trial states towards the lowest eigenvalues and it makes sure that it is not possible that by accident two trial vectors converge towards the same eigenvector. In addition to this, it separates and orthogonalizes trial states converging towards eigenstates with the same eigenvalue (degenerate eigenvectors).

About the Ritz projection: I made two versions of the Ritz projection: One includes all intermediate vectors in the Ritz matrix = $V^T A V$ where A is the matrix whose eigenvalues are sought, and $V = |r_1^0> \cdots |r_1^{block}>, |r_2^0> \cdots |r_2^{block}>, \ldots |r_l^1> \cdots |r_l^{block}>$ (I call this variant of the algorithm maximum V, because V include all Ritz information this version, it makes V in a maximum size). This enlarges the search space of the Ritz projection and leads in all cases to a convergence towards the lowest eigenvalues. In RMM-Ritz maximum V version, convergence is guaranteed for random initial eigenvectors, where I set the elements in initial eigenvectors with a range from 1 to -1. The other version of the Ritz projection, uses only the latest eigenvectors. (I call this variant of the algorithm minimum V.) $V = |a_1^{block}>, |a_2^{block}> \cdots |a_l^{block}>$. The

minimum V version uses less memory and less calculations steps than maximum V version, but this version needs good enough initial eigenvectors to start.

Set $M_{g,h} =< V^g|AV^h >$ and $N_{g,h} =< V^g|V^h >$ for $g,h = 0: l \times block - 1$. M and N are $(l \times block) \times (l \times block)$ symmetric matrices with small size, the generalized eigenproblem in a short time. After solving the generalized eigenproblem between M and N, the eigenvectors β which are result in generalized eigenproblem of M and N can be put into Eq.(3.1) to new eigenvalue Eq.(2.6):

$$|a_i^0 >_s = \sum_{j=0}^{d-1} \beta_{j,s} V_{i,j} \tag{3.1}$$

The new residual vector $|r^0 >$ fulfills the convergence criterion, if $|| \, |r_i^0 > \, ||_2 < \varepsilon$, for $i = 1: l$, then stop the program is stop, if the residual vector $|r_i^0 >$ does not satisfy the criterion, let $k = k + 1$ and go back to RMM step.

Algorithm 5 RMM- RITZ

Input: matrix A, where A is a $n \times n$ symmetrical sparse matrix.
integer l, where l is number of the lowest eigenpairs need to be calculated.
matrix a, where $a = [|a_1^0>, |a_2^0>, \dots, |a_l^0>]$,is a random $n \times l$ matrix, and contains l initial eigenvectors of A.
integer m, where m is maximum number of iterations.
integer k, where $k = 1$ as initial value.
integer $block$, which limits the dimension of the block.
tolerance ε, where is a threshold, if norm of residual vector smaller than ε then break.
maximum *V* or minimum *V*

Output: the lowest l eigenpairs $\lambda_i^j, |a_i^j> \ (j = 1,2, \dots, l)$

RMM – DIIS step

1. **for** $i = 1{:}\, l$
2. pick an input eigenvector $|a_i^0>$, let $|r_i^0> = |a_i^0>$
3. Compute $\lambda_i^0 = \dfrac{< a_i^0|A|a_i^0 >}{< a_i^0|a_i^0 >}$
4. **for** $b = 0 : block$
5. Compute the $A|r_i^b>$ and store it;
6. Compute the $|r_i^{b+1}> = A|a_i^b> - \lambda_i^b|a_i^b>$
7. Let $P_{g.h} =< r_i^g|\left(A - \lambda_i^b I\right)|\left(A - \lambda_i^b I\right)|r_i^h >$ for $g = 0{:}\, b+1$ for $h = 0{:}\, b+1$
8. Let $Q_{g,h} =< r_i^g|I|r_i^h >$ for $g = 0{:}\, b+1$ for $h = 0{:}\, b+1$
9. Compute the lowest eigenvector $|\alpha>$in generalized eigenproblem of P and Q;
10. Compute new eigenvector $|a_i^{b+1}> = \sum_{j=0}^{b+1} \alpha_j |r_i^j>$;
11. Compute new eigenvalue $\frac{<a_i^{b+1}|A|a_i^{b+1}>}{<a_i^{b+1}|a_i^{b+1}>}$; Store $|A|a_i^{b+1}>$;

end for

12. **end for**
13. **end RMM-DIIS step**

RITZ step

14. **if** Using maximum V
15. $d = l \times block$;
16. $V = |r_1^0> \cdots |r_1^{block}>, |r_2^0> \cdots |r_2^{block}>, \ldots |r_l^1> \cdots |r_l^{block}>$
17. $AV = A|r_1^0> \cdots A|r_1^{block}>, A|r_2^0> \cdots A|r_2^{block}> \cdots A|r_l^1> \cdots A|r_l^{block}>$
18. **else** Using minimum size V
19. $d = l$;
20. $V = |a_1^{block}>, |a_2^{block}> \cdots |a_l^{block}>$;
21. $AV = A|a_1^{block}>, A|a_2^{block}> \cdots A|a_l^{block}>$;
22. **end if**
23. Compute $M_{g,h} =< V^g|AV^h >$ for $g = 0: d-1$ for $h = 0: d-1$
24. Compute $N_{g,h} =< V^g|V^h >$ for $g = 0: d-1$ for $h = 0: d-1$
25. Compute eigenvector β in generalized eigenproblem of M and N;
26. Compute new eigenvectors(s) $|a_i^0>_s = \sum_{j=0}^{d-1} \beta_{j,s} V_{i,j}$ for $i = 1: l$ for $s = 1: dim(A)$
27. Compute new eigenvalue(s) $\lambda_i^0 = \frac{< a_i^0|A|a_i^0 >}{< a_i^0|a_i^0 >}$ for $i = 1: l$
28. Compute residual vector $|r_i^0> = A|a_i^0> - \lambda_i^0|a_i^0>$ for $i = 1: l$
29. **if** $\max_{i=1:l-2} || |r_i^0> ||_2 < \varepsilon$
30. $convergence = 1$ and exit
31. **else if** $k < m$
32. $k = k + 1$ and back to RMM-DIIS step 1
33. **end if**
34. **end RITZ step**

3.3 Block Davidson and RMM-RITZ V_{min}

1. Set an integer m which limits the number of iteration of the Block Davidson algorithm, and a tolerance ε_1;
2. Set an integer r which limits the number of iteration of the RMM-RITZ V_{min} algorithm, and a tolerance ε_2;
3. Operate the Block Davidson algorithm;
4. Store the eigenvectors which are the results of the Block Davidson;
5. Operate the RMM-RITZ V_{min} algorithm using the eigenvector obtained with the Block Davidson algorithm;

3.4 RMM-RITZ V_{max} and RMM-RITZ V_{min}

1. Set an integer m which limits number of iteration of the RMM-RITZ V_{max} algorithm, and a tolerance ε_1;
2. Set an integer r which limits number of iteration of the RMM-RITZ V_{min} algorithm, and a tolerance ε_2;
3. Operate the RMM-RITZ V_{max} algorithm;
4. Store the eigenvectors;
5. Operate the RMM-RITZ V_{min} algorithm using the eigenvector obtained with the RMM-RITZ V_{max} algorithm;

3.5 Parallelization

In my Block Davidson and RMM-DIIS algorithms can be run in parallel on shared-memory computers. To this end the programs make heavy use of Intel's Math Kernel Library (MKL). In particular, multiplications of the sparse Hamilton matrix with a vector are performed with the help of MKL. Additionally, OpenMP is used to parallelize some vector operations. Two computers were used for the calculations in this program: one is a dual hex-core machine with Intel 2×Intel X5650 Xeon processor（2.67GHz） and 12GB of DDR3-1333 RAM, 12 cores with 12 threads. The second machine is an Intel's Many-Integrated Core architecture with Intel Xeon Phi coprocessor 5110P integrates 60 cores with 4 hardware threads per core. Table 3.1 compares MKL and OpenMP on several operations, in same operation have different elapsed times.

Table 3.1 Comparison of MKL and OpenMP

MKL	Faster?	OpenMP	Faster?
dnrm2_	✗	#pragma omp parallel for reduction(+:norm2)	✓
ddot_	✗	#pragma omp parallel for reduction(+:dot)	✓
daxpy_	✗	#pragma omp parallel for	✓
dscal_	✗	#pragma omp parallel for	✓
dgemv_	✗	#pragma omp parallel for	✓

According to Table 3.1, obviously, OpenMP has a better than MKL on some operations; hence I used OpenMP instead of MKL to make these programs obtain acceleration on elapsed time, and improve speedup.

4 Results

In this chapter, I will show my results, include comparison of different block, comparison of different algorithms, and parallel speedup on two different machines.

I find out the top 6 eigenvalues can all reach to convergence easily in all 6 models in this project, but when I tried to calculate out top 8 eigenpairs, there were some different result among the different models and different size, I want to compare them so that's the reason why I determined to calculated top 8 eigenvalues.

4.1 Comparison of different block size

In this section I describe result of calculation to study the influence of the block size on the convergence speed. There are 3 reports to compare different among 5 block sizes in 4 methods on two-leg Heisenberg Ladder staggered model, I abbreviated it Ladder 215S, where N = 24, M = 0 (All results in section 4.1 are calculated by an Intel Xeon Phi coprocessor 5110P.)

In total I have implement four approaches: the first one is the Block Davidson method only; the second one is the RMM-DIIS V_{max} method only, the third one is started the calculation with the Block Davidson method and switched to the RMM-DIIS V_{min} after 60 iterations or when the norms of all residual vectors was below 5×10^{-2}; the last one is the RMM-DIIS V_{max} and RMM-DIIS V_{min} combination method, the switch conditions is same as the third method: reach to 60 iterations or when the norms of all

residual vector s was below 5×10^{-2}; For all of 4 methods, exit program if number of iterations exceed 240 or when the norms of all residual vectors was below 5×10^{-4}. The number 5×10^{-2} and 5×10^{-4} were chosen so that convergence could be achieved is a reasonable time in most cases for all system sizes. Practical applications might require more stringent criterions.

The minimum size of block is 2, hence I started from 2, and increased the size of the block until 6. There are 20 items in each report, they used same vectors as initial eigenvectors. It can compare the result objectively. In order to proof the result doesn't happen by accident, I tested three different initial eigenvectors in each report, and they have revealed convergence.

In the following reports, I have recorded 5 attributions; The total number of iterations and the number of blocks the in RMM-RITZ algorithm if applicable (in brackets); The convergence status for the lowest eight eigenvalues (if the norm of the residual is vector is below the tolerance); 'count system' shows how many $A|v>$ operation were performed where A is the Hamilton matrix, $|v>$ is a vector with same column size as A; 'clock time' is the execution time in seconds; 'Memory' recorded the memory usage, I checked memory with the help of the 'top' command of Ubuntu.

Table 4.1 Report 1 Convergence of Ladder 215S(24) by Block Davidson

Block size	**Iteration**	**convergent**	**Count Sys**	**Clock time (sec.)**	**Memory**
BKDV 2	240	no	5752	956.5	18062 MB
BKDV 3	240	no	6424	956.2	18227 MB
BKDV 4	125	yes	3976	521.5	18392 MB
BKDV 5	94	yes	3352	395.5	18557 MB
BKDV 6	75	yes	2960	321.7	18723 MB

Table 4.2 Report 1 Convergence of Ladder 215S(24) by Block Davidson and RMM-RITZ V_{min}

Block size	**Iteration**	**convergent**	**Count Sys**	**Clock time (sec.)**		**Memory**
BKDV 2	60	no	1432	239.6	261.3	18161 MB
RMM 4 V_{min}	78(26)	yes	865	21.7		18141 MB
RMM 3	60	no	1664	250.5	300.4	18498 MB
RMM 4 V_{min}	180(60)	no	1987	50.0		18133 MB
BKDV 4	39	yes	1224	160.3	183.7	18814 MB
RMM 4 V_{min}	84(28)	yes	931	23.4		18133 MB
BKDV 5	38	yes	1336	159.8	179.8	19160 MB
RMM 4 V_{min}	72(24)	yes	799	20.1		18149 MB
BKDV 6	32	yes	1240	138.6	157.0	19491 MB
RMM 4 V_{min}	66(22)	yes	733	18.42		18149 MB

Table 4.3 Report 1 Convergence of Ladder 215S(24) by RMM-RITZ V_{max}

Block size	**Iteration**	**convergent**	**Count Sys**	**Clock time (sec.)**	**Memory**
RMM 2 V_{max}	240(240)	no	4087	122.0	18161 MB
RMM 3 V_{max}	240(120)	no	3007	103.8	18514 MB
RMM 4 V_{max}	201(67)	yes	2218	86.1	18846 MB
RMM 5 V_{max}	160(40)	yes	1647	71.7	19176 MB
RMM 6 V_{max}	125(25)	yes	1134	60.9	19516 MB

Table 4.4 Report 1 Convergence of Ladder 215S(24) by RMM-RITZ V_{max} and V_{min}

Block size	**iteration**	**convergent**	**Count Sys**	**Clock time (sec.)**		**Memory**
RMM 2 V_{max}	60(60)	no	1027	30.5	80.4	18145 MB
RMM 4 V_{min}	180(60)	no	1987	50.0		18125 MB
RMM 3 V_{max}	60(30)	no	757	26.0	76.1	18498 MB
RMM 4 V_{min}	180(60)	no	1987	50.0		18133 MB
RMM 4 V_{max}	60(20)	no	667	25.8	43.4	18814 MB
RMM 4 V_{min}	63(21)	yes	700	17.6		18133 MB
RMM 5 V_{max}	60(15)	no	622	26.7	54.3	19160 MB
RMM 4 V_{min}	99(33)	yes	1096	27.6		18149 MB
RMM 6 V_{max}	60(12)	no	595	28.1	44.9	19491 MB
RMM 4 V_{min}	60(20)	yes	667	16.8		18149 MB

Table 4.5 Report 2 Convergence of Ladder 215S(24) by Block Davidson

Block size	**Iteration**	**convergent**	**Count Sys**	**Clock time (sec.)**	**Memory**
BKDV 2	240	no	4087	954.9	18161 MB
BKDV 3	220	yes	2757	564.2	18514 MB
BKDV 4	157	yes	1723	504.1	18846 MB
BKDV 5	124	yes	1278	417.9	19176 MB
BKDV 6	105	yes	1036	318.9	19516 MB

Table 4.6 Report 2 Convergence of Ladder 215S(24) by Block Davidson and RMM-RITZ V_{min}

Block size	**iteration**	**convergent**	**Count Sys**	**Clock time (sec.)**		**Memory**
BKDV 2	60	no	1432	239.1	263.3	18161 MB
RMM 4 V_{min}	87(29)	yes	964	24.2		18141 MB
RMM 3	44	yes	1216	180.5	198.1	18498 MB
RMM 4 V_{min}	63(21)	yes	700	17.6		18133 MB
BKDV 4	47	yes	1480	194.2	214.3	18814 MB
RMM 4 V_{min}	72(24)	yes	799	20.1		18133 MB
BKDV 5	38	yes	1336	159.9	183.3	19160 MB
RMM 4 V_{min}	84(28)	yes	931	23.4		18149 MB
BKDV 6	30	yes	1160	126.2	147.1	19491 MB
RMM 4 V_{min}	75(25)	yes	832	20.9		18149 MB

Table 4.7 Report 2 Convergence of Ladder 215S(24) by RMM-RITZ V_{max}

Block size	**Iteration**	**convergent**	**Count Sys**	**Clock time (sec.)**	**Memory**
RMM 2 V_{max}	240(240)	no	4087	122.1	18161 MB
RMM 3 V_{max}	240(120)	no	3007	103.7	18514 MB
RMM 4 V_{max}	195(65)	yes	2152	83.2	18846 MB
RMM 5 V_{max}	148(37)	yes	1524	65.7	19176 MB
RMM 6 V_{max}	115(23)	yes	1134	53.8	19516 MB

Table 4.8 Report 2 Convergence of Ladder 215S(24) by RMM-RITZ V_{max} and V_{min}

Block size	**iteration**	**convergent**	**Count Sys**	**Clock time (sec.)**		**Memory**
RMM 2 V_{max}	60(60)	no	1027	30.6	80.7	18145 MB
RMM 4 V_{min}	180(60)	no	1987	50.1		18125 MB
RMM 3 V_{max}	60(30)	no	757	26.0	75.9	18498 MB
RMM 4 V_{min}	180(60)	no	1987	49.9		18133 MB
RMM 4 V_{max}	60(20)	no	667	25.7	49.9	18814 MB
RMM 4 V_{min}	87(29)	yes	964	24.2		18133 MB
RMM 5 V_{max}	60(15)	no	662	26.7	51.7	19160 MB
RMM 4 V_{min}	90(30)	yes	997	25.0		18149 MB
RMM 6 V_{max}	55(11)	yes	546	25.7	41.6	19491 MB
RMM 4 V_{min}	57(19)	yes	634	15.9		18149 MB

Table 4.9 Report 3 Convergence of Ladder 215S(24) by Block Davidson

Block size	**Iteration**	**convergent**	**Count Sys**	**Clock time (sec.)**	**Memory**
BKDV 2	232	yes	5560	920.5	18161 MB
BKDV 3	164	yes	4576	681.8	18514 MB
BKDV 4	107	yes	3400	416.1	18846 MB
BKDV 5	86	yes	3064	366.3	19176 MB
BKDV 6	67	yes	2640	276.8	19516 MB

Table 4.10 Report 3 Convergence of Ladder 215S(24) by Block Davidson and RMM-RITZ V_{min}

Block size	**iteration**	**convergent**	**Count Sys**	**Clock time (sec.)**		**Memory**
BKDV 2	60	no	1432	238.9	256.5	18161 MB
RMM 4 V_{min}	63(21)	yes	700	17.6		18141 MB
RMM 3	44	yes	1216	179.9	201.6	18498 MB
RMM 4 V_{min}	78(26)	yes	865	21.7		18133 MB
BKDV 4	38	yes	1192	145.8	162.5	18814 MB
RMM 4 V_{min}	60(20)	yes	667	16.7		18133 MB
BKDV 5	30	yes	1048	125.8	146.7	19160 MB
RMM 4 V_{min}	75(25)	yes	832	20.9		18149 MB
BKDV 6	28	yes	1072	117.9	133.8	19491 MB
RMM 4 V_{min}	57(19)	yes	634	15.9		18149 MB

Table 4.11 Report 3 Convergence of Ladder 215S(24) by RMM-RITZ V_{max}

Block size	**Iteration**	**convergent**	**Count Sys**	**Clock time (sec.)**	**Memory**
RMM 2 V_{max}	240(240)	no	4087	122.3	18161 MB
RMM 3 V_{max}	220(110)	yes	2757	95.1	18514 MB
RMM 4 V_{max}	157(52)	yes	1723	66.7	18846 MB
RMM 5 V_{max}	124(31)	yes	1278	55.2	19176 MB
RMM 6 V_{max}	105(21)	yes	1036	49.2	19516 MB

Table 4.12 Report 3 Convergence of Ladder 215S(24) by RMM-RITZ V_{max} and V_{min}

Block size	**iteration**	**convergent**	**Count Sys**	**Clock time (sec.)**		**Memory**
RMM 2 V_{max}	60(60)	no	1027	30.5	42.3	18145 MB
RMM 4 V_{min}	42(14)	yes	469	11.8		18125 MB
RMM 3 V_{max}	60(30)	no	757	26.0	42.7	18498 MB
RMM 4 V_{min}	60(20)	yes	667	16.7		18133 MB
RMM 4 V_{max}	51(17)	yes	568	21.9	32.0	18814 MB
RMM 4 V_{min}	36(12)	yes	403	10.1		18133 MB
RMM 5 V_{max}	44(11)	yes	458	19.6	41.3	19160 MB
RMM 4 V_{min}	78(26)	yes	865	21.7		18149 MB
RMM 6 V_{max}	35(7)	yes	350	16.4	36.5	19491 MB
RMM 4 V_{min}	72(24)	yes	799	20.1		18149 MB

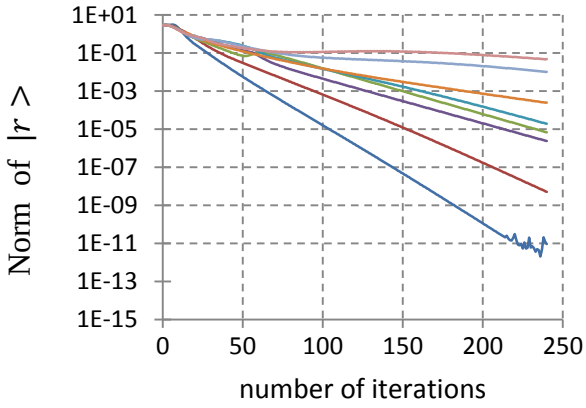


Figure 4.1 Results of Ladder 215S(24),

block = 2 by **RMM V_{max}**

iteration = 240,

time = 122.0s

on an Xeon Phi in report 1

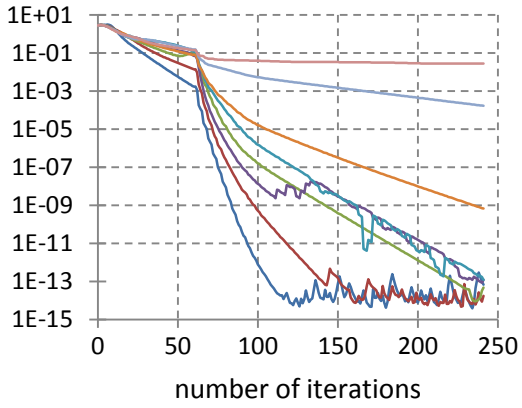


Figure 4.2 Results of Ladder 215S(24),

block = 2/4 by **RMM V_{max}**& **V_{min}**

iteration = 60(60)/180(60),

time = 30.5/50.0(80.5)s

on an Xon Phi in report 1

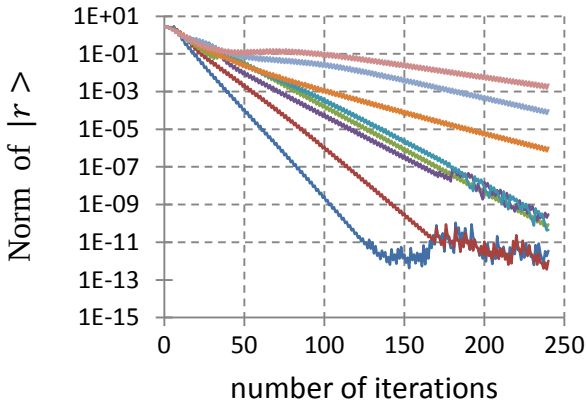


Figure 4.3 Results of Ladder 215S(24),

block = 3 by **RMM V_{max}**

iteration = 240(120),

time = 103.8s

on an Xeon Phi in report 1

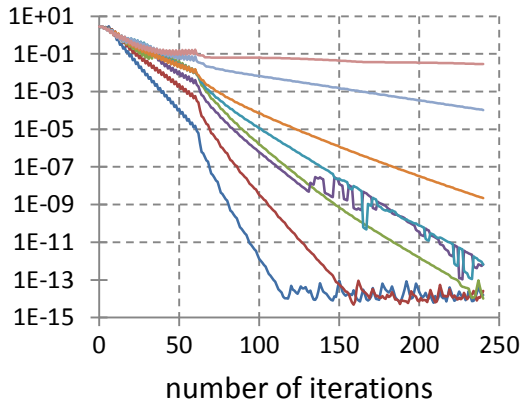


Figure 4.4 Results of Ladder 215S(24),

block = 3/4 by **RMM V_{max}**& **V_{min}**

iteration = 60(30)/180(60),

time = 26.0/ 50.01(76.1) s

on an Xeon Phi in report 1

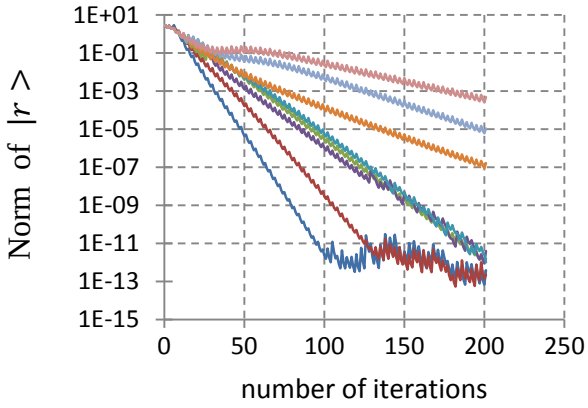


Figure 4.5 Results of Ladder 215S(24),

block = 4 by **RMM V_{max}**

iteration = 201(67),

time = 86.1s

on an Xeon Phi in report 1

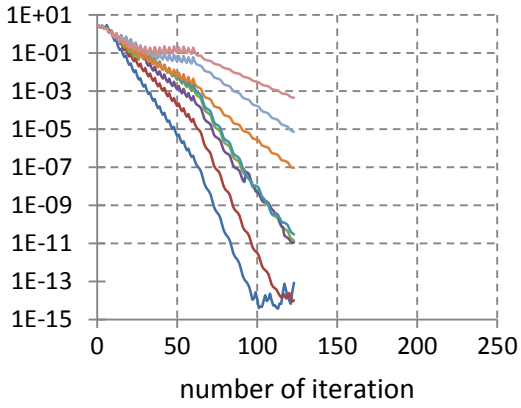


Figure 4.6 Results of Ladder 215S(24),

block = 4/4 by **RMM V_{max}**& **V_{min}**

iteration = 60(20)/63(21),

time = 25.8 /17.6(43.4)s

on an Xeon Phi in report 1

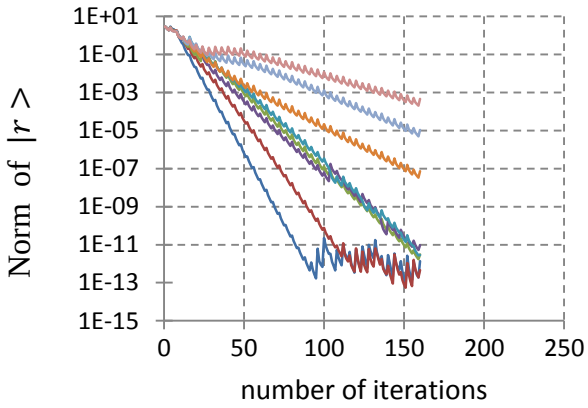


Figure 4.7 Results of Ladder 215S(24),

block = 5 by **RMM V_{max}**

iteration = 160(40),

time = 71.7s

on an Xeon Phi in report 1

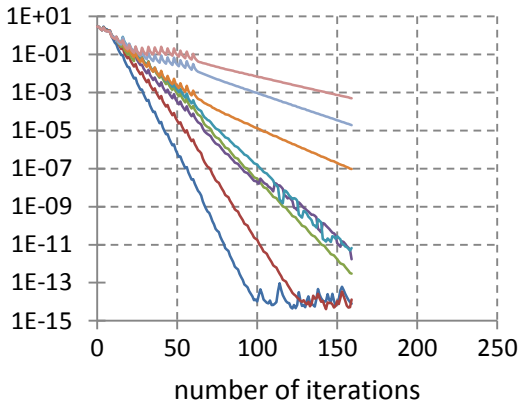


Figure 4.8 Results of Ladder 215S(24),

block = 5/4 by **RMM V_{max}**& **V_{min}**

iteration = 60(15)/99(33),

time = 26.7 /27.6 (54.3)s

on an Xeon Phi in report 1

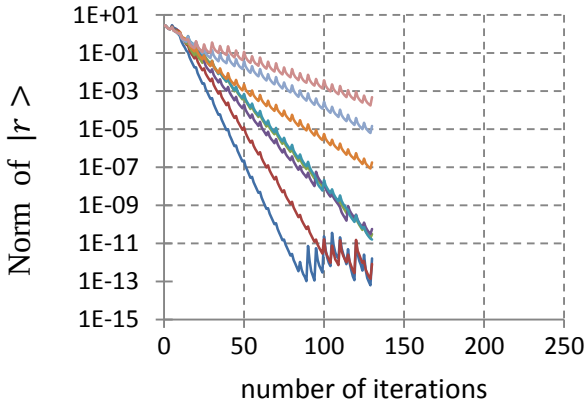


Figure 4.9 Results of Ladder 215S(24),

block = 6 by **RMM V_{max}**

iteration = 125(25),

time = 60.9s

on an Xeon Phi in report 1

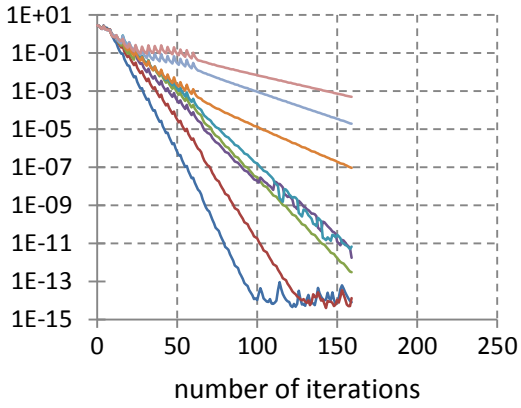


Figure 4.10 Results of Ladder 215S(24),

block = 6/4 by **RMM V_{max}**& **V_{min}**

iteration = 60(12)/60(20),

time = 28.1 /16.8 (44.9)s

on an Xeon Phi in report 1

The error lines shown in the figures are the norm of residual vectors, because there are 8 eigenvalues were calculated, so there are 8 curves plotted in each figure.

Figure 4.1, Figure 4.3, Figure 4.5, Figure 4.7 and Figure 4.9 show results of RMM V_{max} method with different block size from 2 to 6. As a measure for the error of the trial vectors, these figures show the norms of the residual vector as a function of the iteration number. When the block size increases, the time to reaching convergence shorter, but it required more memory. Just several error of eigenvalues can reach 5×10^{-4} with block = 2 and 3, after block size larger than 3, all eight eigenvalues can convergence at 5×10^{-4}.

Figure 4.2, Figure 4.4, Figure 4.6, Figure 4.8 and Figure 4.10 show results of RMM V_{max} and RMM V_{min} combination method with different block size from 2 to 6 in RMM V_{max}, and block size = 4 in RMM V_{min}. They use the same initial eigenvectors as the ones in RMM V_{max} method. For this reason the first part of the graphs in the RMM combination methods are the same as RMM V_{max} when they have same block size. After the block size increases to 4 or bigger in RMM V_{max} and RMM V_{min} combination method, all eigenvalues finally reach the tolerance level.

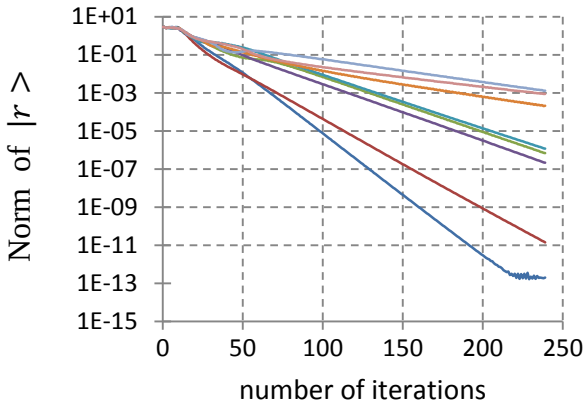


Figure 4.11 Results of Ladder 215S(24),
block = 2 by **BKDV**
iteration = 240,
time = 956.5s
on an Xeon Phi in report 1

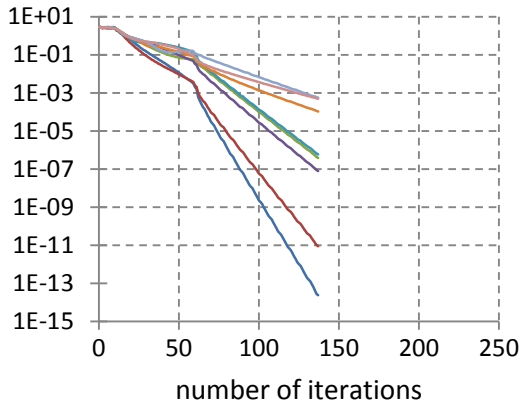


Figure 4.12 Results of Ladder 215S(24),
block = 2/4 by **BKDV** and **RMM V_{min}**
iteration = 60(60)/78(26),
time = 239.6 /21.7(261,3)s
on an Xeon Phi in report 1

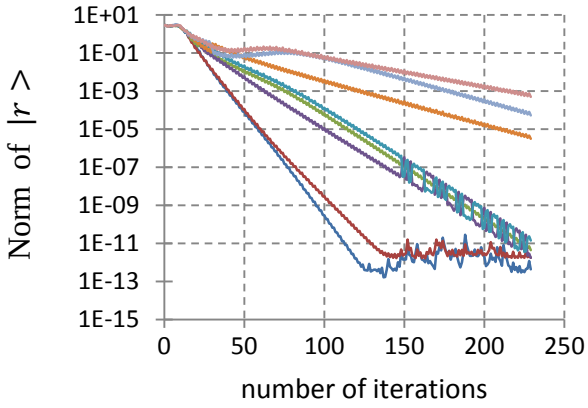


Figure 4.13 Results of Ladder 215S(24),

block = 3 by **BKDV**

iteration = 240,

time = 956.2 s

on an Xeon Phi in report 1

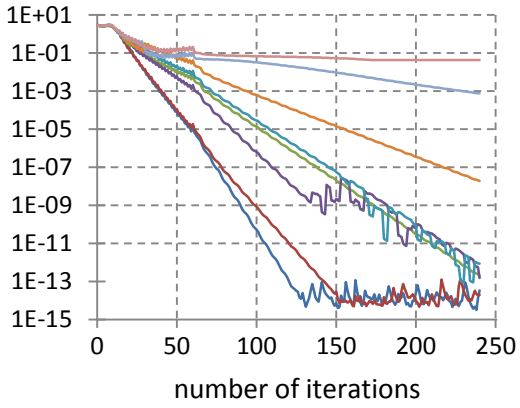


Figure 4.14 Results of Ladder 215S(24),

block = 3/4 by **BKDV** and **RMM** V_{min}

iteration = 60/180(60),

time = 250.5/50.0 (300.5) s

on an Xeon Phi in report 1

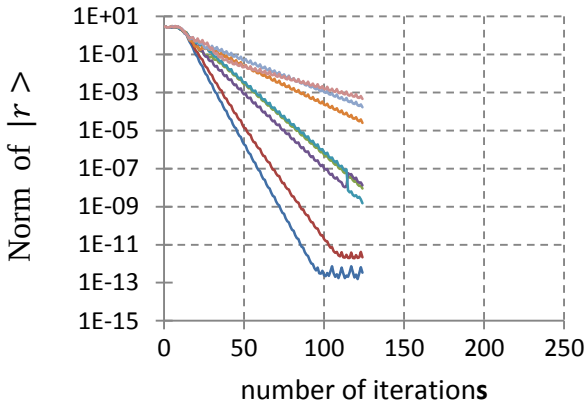


Figure 4.15 Results of Ladder 215S(24),

block = 4 by **BKDV**

iteration = 125,

time = 521.5s

on an Xeon Phi in report 1

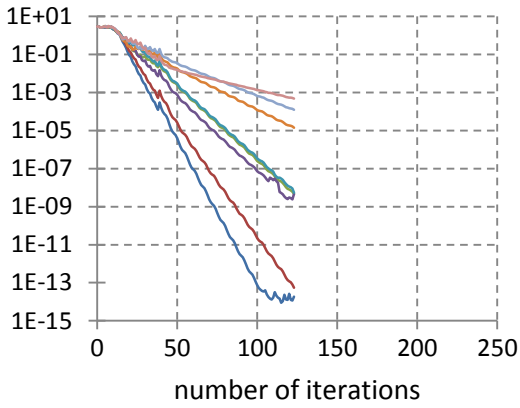


Figure 4.16 Results of Ladder 215S(24),

block = 4/4 by **BKDV** and **RMM** V_{min}

iteration = 39/84(28),

time = 160.3 /23.4 (183.7)s

on an Xeon Phi in report 1

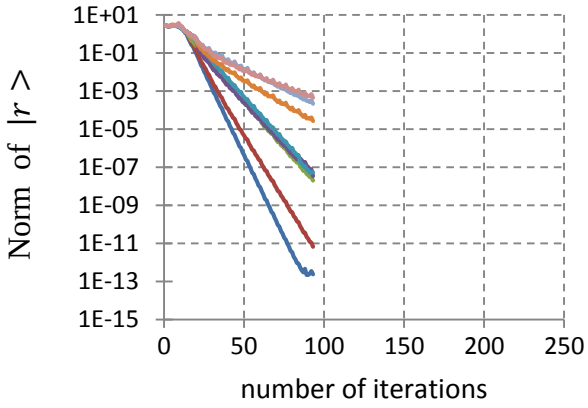


Figure 4.17 Results of Ladder 215S(24),

block = 5 by **BKDV**

iteration = 94,

time = 395.5 s

on an Xeon Phi in report 1

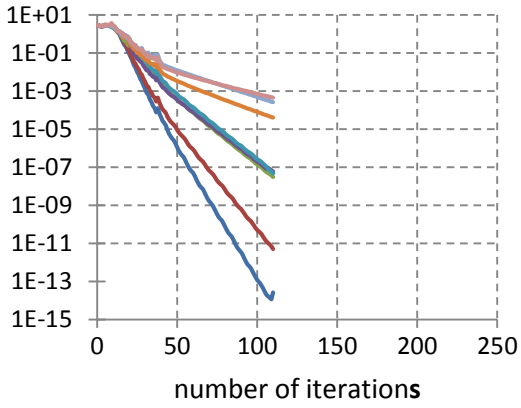


Figure 4.18 Results of Ladder 215S(24),

block = 5/4 by **BKDV and RMM** V_{min}

iteration = 38/72(24),

time = 159.8/20.1 (179.8)s

on an Xeon Phi in report 1

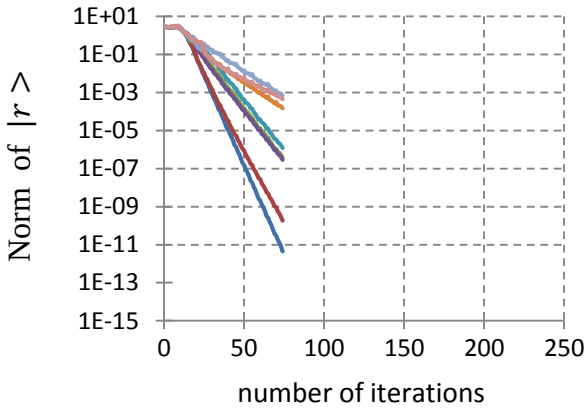


Figure 4.19 Results of Ladder 215S(24),

block = 6 by **BKDV**

iteration = 75,

time = 321.7s

on an Xeon Phi in report 1

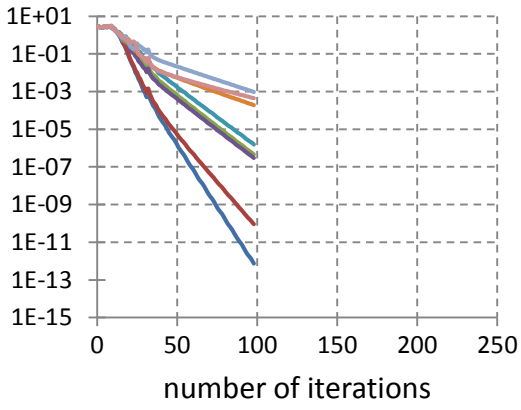


Figure 4.20 Results of Ladder 215S(24),

block = 6/4 by **BKDV and RMM** V_{min}

iteration = 32/66(22),

time = 138.6 / 156.977 (295.530.9)s

on an Xeon Phi in report 1

Figure 4.11, Figure 4.13, Figure 4.15, Figure 4.17 and Figure 4.19 show the results of the Block Davidson method with block sizes of 2, 3 ,4, 5 and 6. When the block size = 4, 8 all eigenvalues can reach convergence.

Figure 4.12, Figure 4.14, Figure 4.16, Figure 4.18 and Figure 4.20 show results of the Block Davidson and RMM V_{min} combination method with different block size from 2 to 6, when the block size = 4, 5 and 6, the eigenvectors calculated by the Block Davidson are good enough, after calculation switch to RMM V_{min} , all eigenvalues can convergence at final.

According to the reports, we observed that Block Davidson needs much more sys operations ($A\,|v>$) than RMM-DIIS, this operation needs a certain time to perform. When number of block size was increasing, greater memory was required and less iterations were needed to reach convergent. In most cases, when the block size = 4 or is over 4, it could obtain appropriate eigenvectors and the corresponding eigenvalues were going to convergent. Therefore, in following programs, default block size is set by 4.

4.2 Results of Block Davidson and RMM-DIIS

In this section I tried to calculate the lowest 8 eigenvalues via 4 different algorithms on these six different models with size N = 18 and 24 on Odin with setting MPI and OMP using 8 threads. (As before four algorithms are 1. RMM-DIIS algorithm, 2. RMM-DIIS V_{max} and V_{min} combination method, 3. Block Davidson algorithm and 4. Block Davidson and RMM-DIIS V_{min} combination method) I find out the top 6 eigenvalues can all reach to convergence easily in all 6 models in this project, reason for why try to calculate the lowest 8 eigenvalues is that not all lowest 8 eigenvalues can convergence at final in all models, therefore, I calculated the lowest 8 eigenpairs to compare which model can easy achieve required tolerance for all top 8 eigenvalues. All calculations block size = 4, iterations were stopped when the norms of all residual vectors were blow 5×10^{-4}. In order to make sure that the result do not depend on the initial vectors, the calculations were repeated five times with a different set of random staring vectors (Report 1-5). For each report the four algorithms were run with the same starting vectors and the execution time and the number of iterations were measured.

In following results table in this section, I recorded 4 attributions, included total number of iterations and iterations number of block in RMM-RITZ algorithm if applicable (in the brackets); convergence status for the lowest eight eigenvalues (if norm of residual vector below the tolerance); 'clock time' is the execution time in seconds; 'Memory' recorded the memory usage, I checked memory by 'top' command in terminal of Ubuntu.

4.2.1 Heisenberg Chain

Table 4.13 shows the result about the number of iterations and the execution time of Chain(24) model by 4 different algorithms. Not all eight errors of eigenvalues can reach to 5×10^{-4} in this model. In most reports, top seven eigenvalues can be convergence.

Table 4.14 shows the result about Chain(18) model by 4 different algorithms. There are two reports show all errors below 5×10^{-4} in RMM-DIIS V_{max} and V_{min} combination method. The lowest 8^{th} eigenvalue did not satisfy required convergent another methods.

Table 4.13 Results of Chain(24) model on Odin

Chain(24)	Method name	Number of iteration (block)	Conv ergent	Clock time(second)		Memory
Report 1	**BKDV 4**	240	7/8		810.50	2327MB
	BKDV 4	38	**8/8**	125.97	253.23	2327MB
	RMM V_{min} 4	180(60)	7/8	127.26		2100MB
	RMM V_{max} 4	240(80)	7/8		396.91	2760MB
	RMM V_{max} 4	60(20)	6/8	99.60	247.14	2780MB
	RMM V_{min} 4	180(60)	7/8	147.54		2100MB
Report 2	**BKDV 4**	240	7/8		813.23	2327MB
	BKDV 4	57	**8/8**	189.12	334.55	2327MB
	RMM V_{min} 4	180(60)	7/8	145.43		2100MB
	RMM V_{max} 4	240(80)	7/8		395.94	2760MB
	RMM V_{max} 4	57(19)	**8/8**	94.47	239.35	2780MB
	RMM V_{min} 4	180(60)	7/8	144.88		2100MB
Report 3	**BKDV 4**	240	7/8		820.20	2327MB
	BKDV 4	35	**8/8**	116.64	265.58	2327MB
	RMM V_{min} 4	180(60)	7/8	148.94		2100MB
	RMM V_{max} 4	240(80)	7/8		401.13	2760MB
	RMM V_{max} 4	45(15)	**8/8**	75.63	224.99	2780MB
	RMM V_{min} 4	180(60)	7/8	149.36		2100MB
Report 4	**BKDV 4**	240	7/8		825.19	2327MB
	BKDV 4	50	**8/8**	168.45	317.34	2327MB
	RMM V_{min} 4	180(60)	7/8	148.89		2100MB
	RMM V_{max} 4	240(80)	7/8		402.33	2760MB
	RMM V_{max} 4	39(13)	**8/8**	65.55	214.91	2780MB
	RMM V_{min} 4	180(60)	7/8	149.36		2100MB
Report 5	**BKDV 4**	240	7/8		813.00	2327MB
	BKDV 4	57	**8/8**	191.21	338.52	2327MB
	RMM V_{min} 4	180(60)	7/8	147.31		2100MB
	RMM V_{max} 4	192(64)	**8/8**		318.60	2760MB
	RMM V_{max} 4	36(12)	**8/8**	95.11	124.83	2780MB
	RMM V_{min} 4	180(60)	7/8	29.72		2100MB

Table 4.14 Results of Chain(18) model on Odin

Chain(18)	Method name	Number of iteration (block)	Convergent	Clock time(second)		Memory
Report 1	**BKDV 4**	240	7/8		4.29	689MB
	BKDV 4	29	**8/8**	0.54	1.14	689MB
	RMM V_{min} 4	180(60)	7/8	0.60		689MB
	RMM V_{max} 4	240(80)	7/8		1.93	694MB
	RMM V_{max} 4	48(16)	**8/8**	0.34	0.93	694MB
	RMM V_{min} 4	180(60)	7/8	0.59		689MB
Report 2	**BKDV 4**	240	7/8		4.63	689MB
	BKDV 4	47	**8/8**	0.89	1.50	689MB
	RMM V_{min} 4	180(60)	7/8	0.61		689MB
	RMM V_{max} 4	240(80)	7/8		1.93	694MB
	RMM V_{max} 4	48(16)	**8/8**	0.37	0.75	694MB
	RMM V_{min} 4	114(38)	**8/8**	0.38		689MB
Report 3	**BKDV 4**	240	7/8		4.67	689MB
	BKDV 4	32	**8/8**	0.61	1.21	689MB
	RMM V_{min} 4	180(60)	7/8	0.60		689MB
	RMM V_{max} 4	240(80)	7/8		1.93	694MB
	RMM V_{max} 4	45(15)	**8/8**	0.33	0.94	689MB
	RMM V_{min} 4	114(38)	**8/8**	0.60		689MB
Report 4	**BKDV 4**	240	7/8		4.53	689MB
	BKDV 4	30	**8/8**	0.55	1.13	689MB
	RMM V_{min} 4	180(60)	7/8	0.58		689MB
	RMM V_{max} 4	240(80)	7/8		1.93	694MB
	RMM V_{max} 4	39(13)	**8/8**	0.29	0.87	689MB
	RMM V_{min} 4	180(60)	7/8	0.58		689MB
Report 5	**BKDV 4**	240	7/8		4.69	689MB
	BKDV 4	28	**8/8**	0.55	1.16	689MB
	RMM V_{min} 4	180(60)	7/8	0.61		689MB
	RMM V_{max} 4	240(80)	7/8		1.96	694MB
	RMM V_{max} 4	36(12)	**8/8**	0.27	0.87	694MB
	RMM V_{min} 4	180(60)	7/8	0.61		689MB

4.2.2 Heisenberg Ladder

Table 4.15 Results of Ladder2(24) model on Odin

Ladder2(24)	Method name	Number of iteration (block)	Convergent	Clock time(second)		Memory
Report 1	**BKDV 4**	127	**8/8**		485.35	2520MB
	BKDV 4	60	7/8	228.20	348.47	2520MB
	RMM V_{min} 4	120(40)	**8/8**	120.26		2293MB
	RMM V_{max} 4	165(55)	**8/8**		300.04	2954MB
	RMM V_{max} 4	60(20)	**8/8**	109.29	144.65	2974MB
	RMM V_{min} 4	36(12)	**8/8**	35.36		2293MB
Report 2	**BKDV 4**	143	**8/8**		549.74	2520MB
	BKDV 4	53	**8/8**	200.87	274.38	2520MB
	RMM V_{min} 4	75(25)	**8/8**	73.51		2293MB
	RMM V_{max} 4	153(51)	**8/8**		284.16	2954MB
	RMM V_{max} 4	60(20)	7/8	109.85	142.35	2974MB
	RMM V_{min} 4	33(11)	**8/8**	32.50		2293MB
Report 3	**BKDV 4**	149	**8/8**		568.20	2520MB
	BKDV 4	60	**8/8**	226.18	304.82	2520MB
	RMM V_{min} 4	81(27)	**8/8**	78.64		2293MB
	RMM V_{max} 4	201(67)	**8/8**		363.33	2954MB
	RMM V_{max} 4	60(20)	7/8	108.88	172.93	2974MB
	RMM V_{min} 4	66(22)	**8/8**	64.05		2293MB
Report 4	**BKDV 4**	152	**8/8**		586.33	2520MB
	BKDV 4	60	**8/8**	230.41	312.76	2520MB
	RMM V_{min} 4	84(28)	**8/8**	82.35		2293MB
	RMM V_{max} 4	159(53)	**8/8**		290.39	2954MB
	RMM V_{max} 4	60(20)	7/8	109.97	148.30	2974MB
	RMM V_{min} 4	39(13)	**8/8**	38.33		2293MB
Report 5	**BKDV 4**	164	**8/8**		629.16	2520MB
	BKDV 4	60	7/8	229.66	324.15	2520MB
	RMM V_{min} 4	96(32)	**8/8**	94.50		2293MB
	RMM V_{max} 4	159(53)	**8/8**		285.49	2954MB
	RMM V_{max} 4	57(19)	**8/8**	100.99	133.71	2974MB
	RMM V_{min} 4	33(11)	**8/8**	32.72		2293MB

Table 4.16 Results of Ladder2(18) model on Odin

Ladder(18)	Method name	Number of iteration (block)	Convergent	Clock time(second)		Memory
Report 1	**RMM V_{max} 4**	240(80)	6/8		2.05	699MB
	RMM V_{max} 4	39(13)	**8/8**	0.32	1.01	699MB
	RMM V_{min} 4	180(60)	6/8	0.69		692MB
	BKDV 4	240	6/8		5.32	692MB
	BKDV 4	29	**8/8**	0.62	1.31	692MB
	RMM V_{min} 4	180(60)	6/8	0.69		692MB
Report 2	**RMM V_{max} 4**	240(80)	6/8		2.06	699MB
	RMM V_{max} 4	39(13)	**8/8**	0.31	0.99	699MB
	RMM V_{min} 4	180(60)	6/8	0.69		692MB
	BKDV 4	240	6/8		5.28	692MB
	BKDV 4	29	**8/8**	0.62	1.31	692MB
	RMM V_{min} 4	180(60)	6/8	0.69		692MB
Report 3	**RMM V_{max} 4**	240(80)	6/8		2.04	699MB
	RMM V_{max} 4	33(11)	**8/8**	0.25	0.92	699MB
	RMM V_{min} 4	180(60)	6/8	0.67		692MB
	BKDV 4	240	6/8		5.11	692MB
	BKDV 4	26	**8/8**	0.53	1.21	692MB
	RMM V_{min} 4	180(60)	6/8	0.68		692MB
Report 4	**RMM V_{max} 4**	240(80)	6/8		2.01	699MB
	RMM V_{max} 4	39(13)	**8/8**	0.31	0.99	699MB
	RMM V_{min} 4	180(60)	6/8	0.69		692MB
	BKDV 4	240	6/8		5.17	692MB
	BKDV 4	26	**8/8**	0.54	1.23	692MB
	RMM V_{min} 4	180(60)	6/8	0.69		692MB
Report 5	**RMM V_{max} 4**	240(80)	6/8		2.00	699MB
	RMM V_{max} 4	39(13)	**8/8**	0.30	0.97	699MB
	RMM V_{min} 4	180(60)	6/8	0.67		692MB
	BKDV 4	240	6/8		5.11	692MB
	BKDV 4	26	**8/8**	0.53	1.21	692MB
	RMM V_{min} 4	180(60)	6/8	0.67		692MB

Table 4.15 shows the result about the number of iterations and the execution time of the Ladder2(24) model by 4 different algorithms. All five reports show Ladder2(24) model can easy reach convergence form random star vectors. The variants RMM-DIIS (RMM-DIIS V_{max} and V_{max}) method is the fastest of all, the number of iterations of it is also the least one. The execution time of the RMM-DIIS only and the Block Davidson and the RMM-DIIS combination method are not much different, they are much faster than the Block Davidson.

From Table 4.16 , we can find that all reports show the lowest six eigenvalues of Ladder2(18) can be convergence by assort algorithms, and all eight eigenvalues are convergence to 5×10^{-2} in combination method.

4.2.3 Heisenberg Ladder 215C

Table 4.17 shows the result about the number of iterations and the execution time of the Ladder215C(18) model by 4 different algorithms. All reports show the lowest seven eigenvalues can be convergence.

Table 4.18 shows the results about the number of iterations and the execution time of the Ladder215C(18) model by 4 different algorithms. In four out of the five reports at least one of the RMM-DIIS variants (RMM-DIIS V_{max} and V_{max}) reached convergence for all eight eigenvalues. I think that the initial eight vectors in the last report were badly chosen, so that the 8^{th} eigenvalue converged only slowly.

Table 4.17 Results of Ladder215C(24) model on Odin

Ladder215C(24)	Method name	Number of iteration (block)	Convergent	Clock time(second)		Memory
Report 1	**BKDV 4**	240	7/8		932.50	2520MB
	BKDV 4	47	**8/8**	177.77	354.25	2520MB
	RMM V_{min} 4	180(60)	7/8	176.48		2293MB
	RMM V_{max} 4	240(80)	7/8		438.45	2954MB
	RMM V_{max} 4	54(18)	**8/8**	99.38	275.03	2974MB
	RMM V_{min} 4	180(60)	7/8	175.65		2293MB
Report 2	**BKDV 4**	240	7/8		927.53	2520MB
	BKDV 4	47	**8/8**	96.74	272.49	2520MB
	RMM V_{min} 4	180(60)	7/8	175.76		2293MB
	RMM V_{max} 4	240(80)	7/8		435.83	2954MB
	RMM V_{max} 4	54(18)	**8/8**	104.50	280.20	2974MB
	RMM V_{min} 4	180(60)	7/8	175.70		2293MB
Report 3	**BKDV 4**	240	7/8		919.23	2520MB
	BKDV 4	27	**8/8**	99.82	274.78	2520MB
	RMM V_{min} 4	180(60)	7/8	174.96		2293MB
	RMM V_{max} 4	240(80)	7/8		436.77	2954MB
	RMM V_{max} 4	57(19)	**8/8**	103.81	279.04	2974MB
	RMM V_{min} 4	180(60)	7/8	175.23		2293MB
Report 4	**BKDV 4**	240	7/8		921.35	2520MB
	BKDV 4	30	**8/8**	111.26	286.21	2520MB
	RMM V_{min} 4	180(60)	7/8	174.94		2293MB
	RMM V_{max} 4	240(80)	7/8		435.58	2954MB
	RMM V_{max} 4	60(20)	7/8	109.23	284.23	2974MB
	RMM V_{min} 4	180(60)	7/8	175.00		2293MB
Report 5	**BKDV 4**	240	7/8		858.31	2520MB
	BKDV 4	40	**8/8**	144.18	304.01	2520MB
	RMM V_{min} 4	180(60)	7/8	159.83		2293MB
	RMM V_{max} 4	240(80)	7/8		397.89	2954MB
	RMM V_{max} 4	51(17)	**8/8**	89.30	249.48	2974MB
	RMM V_{min} 4	180(60)	7/8	160.19		2293MB

Table 4.18 Results of Ladder215C(18) model on Odin

Ladder215C(18)	Method name	Number of iteration (block)	Conv ergent	Clock time(second)		Memory
Report 1	**BKDV 4**	240	7/8		5.15	692MB
	BKDV 4	60	7/8	1.28	1.96	692MB
	RMM V_{min} 4	180(60)	7/8	0.68		692MB
	RMM V_{max} 4	240(60)	7/8		2.02	699MB
	RMM V_{max} 4	60(20)	7/8	0.46	0.76	699MB
	RMM V_{min} 4	78(26)	**8/8**	0.30		692MB
Report 2	**BKDV 4**	160	**8/8**		3.45	692MB
	BKDV 4	41	**8/8**	0.86	1.34	692MB
	RMM V_{min} 4	123(41)	**8/8**	0.47		692MB
	RMM V_{max} 4	213(70)	**8/8**		1.87	699MB
	RMM V_{max} 4	60(20)	7/8	0.47	0.65	699MB
	RMM V_{min} 4	48(16)	**8/8**	0.18		692MB
Report 3	**BKDV 4**	143	**8/8**		3.06	692MB
	BKDV 4	59	**8/8**	1.25	1.53	692MB
	RMM V_{min} 4	75(25)	**8/8**	0.28		692MB
	RMM V_{max} 4	240(80)	7/8		1.99	699MB
	RMM V_{max} 4	60(20)	**8/8**	0.46	0.83	699MB
	RMM V_{min} 4	99(33)	**8/8**	0.37		692MB
Report 4	**BKDV 4**	240	7/8		5.27	692MB
	BKDV 4	60	7/8	1.30	2.00	692MB
	RMM V_{min} 4	180(60)	7/8	0.70		692MB
	RMM V_{max} 4	135(45)	**8/8**		1.25	699MB
	RMM V_{max} 4	48(16)	**8/8**	0.37	0.47	699MB
	RMM V_{min} 4	27(9)	**8/8**	0.10		692MB
Report 5	**BKDV 4**	240	7/8		5.20	692MB
	BKDV 4	60	7/8	1.29	1.98	692MB
	RMM V_{min} 4	180(60)	7/8	0.69		692MB
	RMM V_{max} 4	240(80）	7/8		2.07	699MB
	RMM V_{max} 4	60(20)	7/8	0.47	1.15	699MB
	RMM V_{min} 4	180(60)	7/8	0.68		692MB

4.2.4 Heisenberg Ladder 215S

Table 4.19 Results of Ladder215S(24) model on Odin

Ladder215S(24)	Method name	Number of iteration (block)	Conv ergent	Clock time(second)		Memory
Report 1	**BKDV 4**	141	**8/8**		538.96	2520MB
	BKDV 4	159(53)	**8/8**	200.11	276.40	2520MB
	RMM V_{min} 4	78(26)	**8/8**	76.28		2293MB
	RMM V_{max} 4	162(54)	**8/8**		287.18	2954MB
	RMM V_{max} 4	60(20)	7/8	106.37	141.59	2974MB
	RMM V_{min} 4	36(12)	**8/8**	35.22		2293MB
Report 2	**BKDV 4**	98	**8/8**		372.53	2520MB
	BKDV 4	105(35)	**8/8**	130.89	186.61	2520MB
	RMM V_{min} 4	57(19)	**8/8**	55.73		2293MB
	RMM V_{max} 4	153(51)	**8/8**		272.27	2954MB
	RMM V_{max} 4	60(20)	**8/8**	106.31	135.85	2974MB
	RMM V_{min} 4	30(10)	**8/8**	29.54		2293MB
Report 3	**BKDV 4**	140	**8/8**		538.37	2520MB
	BKDV 4	159(53)	**8/8**	201.74	278.52	2520MB
	RMM V_{min} 4	78(26)	**8/8**	76.77		2293MB
	RMM V_{max} 4	147(49)	**8/8**		262.30	2954MB
	RMM V_{max} 4	51(17)	**8/8**	91.03	126.78	2974MB
	RMM V_{min} 4	36(12)	**8/8**	35.75		2293MB
Report 4	**BKDV 4**	180	**8/8**		463.73	2520MB
	BKDV 4	150(50)	**8/8**	126.47	223.69	2520MB
	RMM V_{min} 4	132(44)	**8/8**	97.22		2293MB
	RMM V_{max} 4	240(80)	7/8		290.50	2954MB
	RMM V_{max} 4	60(20)	7/8	73.16	134.79	2974MB
	RMM V_{min} 4	84(28)	**8/8**	61.63		2293MB
Report 5	**BKDV 4**	128	**8/8**		490.24	2520MB
	BKDV 4	132(44)	**8/8**	165.76	244.67	2520MB
	RMM V_{min} 4	81(27)	**8/8**	78.91		2293MB
	RMM V_{max} 4	210(70)	**8/8**		373.92	2954MB
	RMM V_{max} 4	60(20)	7/8	105.33	216.22	2974MB
	RMM V_{min} 4	114(38)	**8/8**	110.89		2293MB

Table 4.20 Results of Ladder215S(18) model on Odin

Ladder215S(18)	Method name	Number of iteration (block)	Convergent	Clock time(second)		Memory
Report 1	**BKDV 4**	76	**8/8**		1.66	692MB
	BKDV 4	105(35)	**8/8**	0.74	0.91	692MB
	RMM V_{min} 4	42(13)	**8/8**	0.16		692MB
	RMM V_{max} 4	93(31)	**8/8**		0.85	699MB
	RMM V_{max} 4	39(13)	**8/8**	0.30	0.38	699MB
	RMM V_{min} 4	21(7)	**8/8**	0.08		692MB
Report 2	**BKDV 4**	96	**8/8**		2.21	692MB
	BKDV 4	36	**8/8**	0.81	1.04	692MB
	RMM V_{min} 4	54(18)	**8/8**	0.23		692MB
	RMM V_{max} 4	120(40)	**8/8**		1.27	699MB
	RMM V_{max} 4	36(12)	**8/8**	0.29	0.48	699MB
	RMM V_{min} 4	45(15)	**8/8**	0.19		692MB
Report 3	**BKDV 4**	95	**8/8**		2.17	692MB
	BKDV 4	99(33)	**8/8**	0.74	0.99	692MB
	RMM V_{min} 4	60(20)	**8/8**	0.26		692MB
	RMM V_{max} 4	144(48)	**8/8**		1.33	699MB
	RMM V_{max} 4	39(13)	**8/8**	0.31		699MB
	RMM V_{min} 4	48(16)	**8/8**	0.20		692MB
Report 4	**BKDV 4**	172	**8/8**		3.94	692MB
	BKDV 4	105(35)	**8/8**	0.78	1.06	692MB
	RMM V_{min} 4	66(22)	**8/8**	0.28		692MB
	RMM V_{max} 4	165(55)	**8/8**		1.55	699MB
	RMM V_{max} 4	54(18)	**8/8**	0.44	0.59	699MB
	RMM V_{min} 4	36(12)	**8/8**	0.15		692MB
Report 5	**BKDV 4**	114	**8/8**		2.65	692MB
	BKDV 4	114(38)	**8/8**	0.87	1.16	692MB
	RMM V_{min} 4	69(23)	**8/8**	0.30		692MB
	RMM V_{max} 4	183(61)			1.67	699MB
	RMM V_{max} 4	60(20)	**8/8**	0.48	0.63	699MB
	RMM V_{min} 4	36(12)	**8/8**	0.15		692MB

According to Table 4.19, we can get results about the number of iterations and the execution time of the Ladder215S(24) model by 4 different algorithms. Almost all result can reach convergence, the variants RMM-DIIS (RMM-DIIS V_{max} and V_{max}) method is the fastest of all, Block Davidson and RMM-DIIS combination method is the second fastest in most reports， the slowest one always Block Davidson method.

Table 4.20 shows the result about the number of iterations and the execution time of the Ladder215S(18) model. Results of all reports without exception can reach convergence at final Almost without exception, the variants RMM-DIIS (RMM-DIIS V_{max} and V_{max}) method is the fastest of all.

4.2.5 Heisenberg Ladder 315C

Table 4.21 shows the result about the number of iterations and the execution time of the Ladder315C(24) model, most results in these five report show Ladder315C(24) the lowest seven eigenvalues can reach convergence. Three of the all five reports show all eight eigenvalues can reach convergence via the combination RMM-DIIS.method (RMM-DIIS V_{max} and V_{max}).

Table 4.42 shows the result about the number of iterations and the execution time of the Ladder315C(18) model, all results in five reports can reach convergence, the combination RMM-DIIS (RMM V_{max} and V_{min}) is the fastest one of all, in the report5 of Table 4.22 the combination RMM-DIIS method is much faster than Block Davidson, even more than 6 times faster in this case.

Table 4.21 Comparison of results of Ladder315C(24) model on Odin

Ladder315C(24)	Method name	Number of iteration (block)	Conv ergent	Clock time(second)		Memory
Report 1	**BKDV 4**	231	**8/8**		931.91	2585MB
	BKDV 4	60	7/8	241.47	429.32	2585MB
	RMM V_{min} 4	180(60)	7/8	187.85		2358MB
	RMM V_{max} 4	240(80)	7/8		433.38	3018MB
	RMM V_{max} 4	60(20)	7/8	108.12	295.88	3038MB
	RMM V_{min} 4	180(60)	7/8	187.76		2358MB
Report 2	**BKDV 4**	240	7/8		971.18	2585MB
	BKDV 4	60	7/8	240.97	428.46	2585MB
	RMM V_{min} 4	177(59)	7/8	187.49		2358MB
	RMM V_{max} 4	240(80)	7/8		446.26	3018MB
	RMM V_{max} 4	60(20)	7/8	111.76	299.94	3038MB
	RMM V_{min} 4	180(60)	7/8	188.18		2358MB
Report 3	**BKDV 4**	240	7/8		971.48	2585MB
	BKDV 4	60	7/8	241.10	429.17	2585MB
	RMM V_{min} 4	180(60)	7/8	188.07		2358MB
	RMM V_{max} 4	240(80)	7/8		447.37	3018MB
	RMM V_{max} 4	60(20)	7/8	112.02	225.18	3038MB
	RMM V_{min} 4	108(36)	**8/8**	113.16		2358MB
Report 4	**BKDV 4**	240	**8/8**		972.21	2585MB
	BKDV 4	60	7/8	239.61	426.95	2585MB
	RMM V_{min} 4	180(60)	7/8	187.34		2358MB
	RMM V_{max} 4	231(77)	**8/8**		428.67	3018MB
	RMM V_{max} 4	60(20)	**8/8**	111.39	180.10	3038MB
	RMM V_{min} 4	66(22)	**8/8**	68.72		2358MB
Report 5	**BKDV 4**	240	7/8		971.33	2585MB
	BKDV 4	60	7/8	241.00	429.11	2585MB
	RMM V_{min} 4	180(60)	7/8	188.11		2358MB
	RMM V_{max} 4	240(80)	7/8		446.85	3018MB
	RMM V_{max} 4	60(20)	7/8	112.07	218.68	3038MB
	RMM V_{min} 4	102(34)	**8/8**	106.61		2358MB

Table 4.22 Comparison of results of Ladder315C(18) model on Odin

Ladder315(18)	Method name	Number of iteration (block)	Conv ergent	Clock time(second)		Memory
Report 1	**BKDV 4**	129	**8/8**		2.99	692MB
	BKDV 4	66(22)	**8/8**	0.72	1.10	692MB
	RMM V_{min} 4	90(30)	**8/8**	0.38		692MB
	RMM V_{max} 4	153(51)	**8/8**		1.36	699MB
	RMM V_{max} 4	45(15)	**8/8**	0.36	0.54	699MB
	RMM V_{min} 4	42(14)	**8/8**	0.18		692MB
Report 2	**BKDV 4**	116	**8/8**		2.73	692MB
	BKDV 4	114(38)	**8/8**	0.89	1.20	692MB
	RMM V_{min} 4	72(24)	**8/8**	0.31		692MB
	RMM V_{max} 4	165(55)	**8/8**		1.47	699MB
	RMM V_{max} 4	57(19)	**8/8**	0.46	0.61	699MB
	RMM V_{min} 4	36(12)	**8/8**	0.15		692MB
Report 3	**BKDV 4**	174	**8/8**		4.08	692MB
	BKDV 4	60	7/8	1.40	1.89	692MB
	RMM V_{min} 4	114(38)	**8/8**	0.50		692MB
	RMM V_{max} 4	171(57)	**8/8**		1.53	699MB
	RMM V_{max} 4	57(19)	**8/8**	0.46	0.62	699MB
	RMM V_{min} 4	39(13)	**8/8**	0.17		692MB
Report 4	**BKDV 4**	129	**8/8**		2.99	692MB
	BKDV 4	114(38)	**8/8**	0.87	1.22	692MB
	RMM V_{min} 4	84(28)	**8/8**	0.36		692MB
	RMM V_{max} 4	168(66)	**8/8**		1.52	699MB
	RMM V_{max} 4	57(19)	**8/8**	0.46	0.64	699MB
	RMM V_{min} 4	42(14)	**8/8**	0.18		692MB
Report 5	**BKDV 4**	141	**8/8**		3.24	692MB
	BKDV 4	41	**8/8**	0.93	1.31	692MB
	RMM V_{min} 4	90(30)	**8/8**	0.38		692MB
	RMM V_{max} 4	147(49)	**8/8**		1.37	699MB
	RMM V_{max} 4	39(13)	**8/8**	0.31	0.53	699MB
	RMM V_{min} 4	51(17)	**8/8**	0.22		692MB

4.2.6 Heisenberg Ladder 315S

Table 4.23 Comparison of results of Ladder315S(24) model on Odin

Ladder315S(24)	Method name	Number of iteration (block)	Convergent	Clock time(second)		Memory
Report 1	**BKDV 4**	240	6/8		648.39	2585MB
	BKDV 4	41	**8/8**	108.66	248.22	2585MB
	RMM V_{min} 4	180(60)	6/8	139.57		2358MB
	RMM V_{max} 4	240(80)	6/8		300.93	3018MB
	RMM V_{max} 4	60(20)	6/8	75.76	178.34	3038MB
	RMM V_{min} 4	132(44)	**8/8**	102.59		2358MB
Report 2	**BKDV 4**	115	**8/8**		461.22	2585MB
	BKDV 4	60	7/8	240.46	328.16	2585MB
	RMM V_{min} 4	81(27)	**8/8**	87.71		2358MB
	RMM V_{max} 4	150(50)	**8/8**		280.08	3018MB
	RMM V_{max} 4	57(19)	**8/8**	106.33	144.11	3038MB
	RMM V_{min} 4	36(12)	**8/8**	37.78		2358MB
Report 3	**BKDV 4**	240	7/8		643.39	2585MB
	BKDV 4	41	7/8	107.39	245.86	2585MB
	RMM V_{min} 4	84(28)	**8/8**	138.47		2358MB
	RMM V_{max} 4	240(80)	7/8		298.50	3018MB
	RMM V_{max} 4	60(20)	7/8	74.73	210.36	3038MB
	RMM V_{min} 4	177(59)	**8/8**	135.63		2358MB
Report 4	**BKDV 4**	127	**8/8**		511.56	2585MB
	BKDV 4	60	7/8	241.19	429.26	2585MB
	RMM V_{min} 4	180(60)	7/8	188.07		2358MB
	RMM V_{max} 4	186(62)	7/8		348.11	3018MB
	RMM V_{max} 4	60(20)	7/8	111.85	165.40	3038MB
	RMM V_{min} 4	51(17)	**8/8**	53.55		2358MB
Report 5	**BKDV 4**	143	**8/8**		530.75	2585MB
	BKDV 4	60	7/8	221.25	299.88	2585MB
	RMM V_{min} 4	84(28)	**8/8**	78.63		2358MB
	RMM V_{max} 4	135(45)	**8/8**		228.15	3018MB
	RMM V_{max} 4	60(20)	**8/8**	107.47	135.58	3038MB
	RMM V_{min} 4	30(10)	**8/8**	28.10		2358MB

Table 4.24 Comparison of results of Ladder315S(18) model on Odin

Ladder315S(18)	Method name	Number of iteration (block)	Conv ergent	Clock time (second)		Memory
Report 1	**BKDV 4**	89	**8/8**		2.08	692MB
	BKDV 4	87(29)	**8/8**	0.66	0.90	692MB
	RMM V_{min} 4	57(19)	**8/8**	0.24		692MB
	RMM V_{max} 4	123(41）	**8/8**		1.18	700MB
	RMM V_{max} 4	45(15)	**8/8**	0.37	0.51	700MB
	RMM V_{min} 4	33(11)	**8/8**	0.14		692MB
Report 2	**BKDV 4**	96	**8/8**		2.21	692MB
	BKDV 4	108(35)	**8/8**	0.81	1.04	692MB
	RMM V_{min} 4	54(18)	**8/8**	0.23		692MB
	RMM V_{max} 4	129(43)	**8/8**		1.27	700MB
	RMM V_{max} 4	36(12)	**8/8**	0.29	0.48	700MB
	RMM V_{min} 4	45(15)	**8/8**	0.19		692MB
Report 3	**BKDV 4**	95	**8/8**		2.17	692MB
	BKDV 4	99(33)	**8/8**	0.74	0.99	692MB
	RMM V_{min} 4	60(20)	**8/8**	0.26		692MB
	RMM V_{max} 4	144(48)	**8/8**		1.33	700MB
	RMM V_{max} 4	39(13)	**8/8**	0.31	0.52	700MB
	RMM V_{min} 4	48(16)	**8/8**	0.20		692MB
Report 4	**BKDV 4**	172	**8/8**		3.94	692MB
	BKDV 4	105(35)	**8/8**	0.78	1.06	692MB
	RMM V_{min} 4	66(22)	**8/8**	0.28		692MB
	RMM V_{max} 4	165(55)	**8/8**		1.55	700MB
	RMM V_{max} 4	54(18)	**8/8**	0.44	0.59	700MB
	RMM V_{min} 4	36(12)	**8/8**	0.15		692MB
Report 5	**BKDV 4**	114	**8/8**		2.65	692MB
	BKDV 4	114(38)	**8/8**	0.87	1.16	692MB
	RMM V_{min} 4	69(23)	**8/8**	0.30		692MB
	RMM V_{max} 4	183(61)	**8/8**		1.67	700MB
	RMM V_{max} 4	60(20)	**8/8**	0.48	0.63	700MB
	RMM V_{min} 4	36(12)	**8/8**	0.15		692MB

Table 4.23 shows the result about the number of iterations and the execution time of the Ladder315S(24) model, the lowest 8 eigenvalues can reach convergence in most cases, the results are all convergence via the variants RMM-DIIS can all reach convergence.

Table 4.24 shows the result about the number of iterations and the execution time of the Ladder315S(18) model, all results in 5 reports can reach convergence, the variants RMM-DIIS is the fastest one of all, in report4 of Table 4.24, the variants RMM-DIIS(RMM-DIIS V_{max} and V_{max}) is much faster than Block Davidson, even more than 6 times faster in this case.

4.2.7 Comparison of results

Table 4.25 and Table 4.26 show the result about the number of iterations and the execution time of all six models (Chain, Ladder2, Ladder215C, Ladder215S, Ladder315C and Ladder315S) with N = 24 on Odin. Table 4.27, Table 4.28, Table 4.29, Table 4.30, Table 4.31 and Table 4.32 are figures tables, show result of all 6 different models result, how did they reach convergence. Table 4.33 and Table 4.34 show the result about the number of iterations and the execution time of models with size N = 18 on Odin. According these tables and figures, we can find out some interesting things about the different models and different algorithms.

Comparison of different models

Convergent: For calculating the lowest eight eigenvalues, the Ladder2, Ladder215S and Ladder315S models, errors easily reach convergence at 5×10^{-4}, top 7 eigenvalues of Chain, Ladder215C and Ladder315C models can reach convergence when size N = 24. The lowest 8 eigenvalues of all models can reach convergence with error blow 5×10^{-4}, except for the Chain model and Ladder2 when size N =18. The models with smaller size N reach convergence faster than the models with larger size N, STAGGERED models are easier reach convergence faster than the COLUMNAR models.

Clock time: Calculations by RMM V_{max} did not reach convergence at maximum iteration in Chain, Ladder215C and Ladder315C models with N = 24, we can get the execution times of these 3 models are 395.94s, 438.45s, 447.37s form Table 4.25 and

Table 4.26, that means execution time of Chain < Ladder215C < Ladder315C when their number of iteration are all 240.

Memory: The Chain model uses less memory than the other 5 models; The required memory are same when two-leg Heisenberg Ladder models (Ladder2, Ladder215C and Ladder215S) were calculated, and their required memory are more than Chain models, but less than three-leg Heisenberg Ladder models (Ladder315C and Ladder315S). Three-leg Heisenberg Ladder models need same memory when they were calculated, their required memory is the largest among the all models. We can review Table 2.3 and find out Chain model has 36564892 non-zero elements, two-leg Heisenberg Ladder models have 53495260 non-zero elements，three-leg Heisenberg Ladder models have 59138716 non-zero elements when size N =24. The reason for why do three-leg Heisenberg Ladder models need more memory is there are more non-zero elements in these two models.

Comparison of different algorithms

Convergent: Each of Table 4.27, Table 4.28, Table 4.29, Table 4.30, Table 4.31 and Table 4.32 has 4 figures which show results of 4 different algorithms respectively. In most case, the error of top several eigenvalues can reach to 1×10^{-11} via Block Davidson; RMM-DIIS $\boldsymbol{V_{max}}$ algorithm can calculate results whose error can reach convergence around 1×10^{-13}; the errors are convergence to 1×10^{-14} via the variants RMM-DIIS and Block Davidson RMM V_{min} combination method. Hence these two combination methods can calculate out eigenvalues more precise than single algorithm.

Clock time: By checking all results, we can find out that the two variants RMM-DIIS (RMM V_{max} and RMM V_{min}) are the fastest of all, no matter which model. It is much faster than another 3 algorithms, particularly compare with the Block Davidson method. The combination method of RMM V_{max} and RMM V_{min} are often 3-4 times faster than the Block Davidson method, even in some cases is more than 4 times faster, like the results of Ladder315C(24) in Table 4.26.

We can find out that RMM V_{max} is around 2 times faster than the Block Davidson, although the RMM V_{max} algorithm need more iterations to reach convergence. The reason why RMM-DIIS is faster than Block Davidson is that RMM-DIIS does not require a computationally expensive explicit orthogonalization step at each iteration. Since the method is based on the minimization of the residual vector, the trial vectors converge to the eigenvectors with eigenvalues closets to the current trial eigenvalues.

Memory: As an algorithm which can find out good enough initial eigenvectors in a combination method, RMM-DIIS V_{max} demands around 450MB more memory than Block Davidson method in all models, compare the required memory of Block Davidson (approximate 2300MB – 2500MB), 450MB is not too much. RMM V_{max} is much faster than Block Davidson method, although it need more memory than Block Davidson method.

The order of required memory is RMM V_{max}> Block Davidson > RMM V_{min}, when used with the same block number.

Table 4.25 Comparison of the results of Chain, Ladder2, Ladder215C and Ladder215S models with N = 24 on Odin

Methods name	Number of iteration (block)	Conv ergent	Clock time (second)		Memory
Heisenberg Chain (24)					
BKDV 4	240	7/8		813.23	2327MB
BKDV 4	57	**8/8**	189.12	334.55	2327MB
RMM V_{min} 4	180(60)	7/8	145.43		2100MB
RMM V_{max} 4	240(80)	7/8		395.94	2760MB
RMM V_{max} 4	57(19)	**8/8**	94.47	239.35	2780MB
RMM V_{min} 4	180(60)	7/8	144.88		2100MB
Heisenberg Ladder2(24);					
BKDV 4	143	**8/8**		549.74	2520MB
BKDV 4	53	**8/8**	200.87	274.38	2520MB
RMM V_{min} 4	75(25)	**8/8**	73.51		2293MB
RMM V_{max} 4	153(51)	**8/8**		284.16	2954MB
RMM V_{max} 4	60(20)	7/8	109.85	142.35	2974MB
RMM V_{min} 4	33(11)	**8/8**	32.50		2293MB
Heisenberg Ladderr215C(24)					
BKDV 4	240	7/8		932.50	2520MB
BKDV 4	47	**8/8**	177.77	354.25	2520MB
RMM V_{min} 4	180(60)	7/8	176.48		2293MB
RMM V_{max} 4	240(80)	7/8		438.45	2954MB
RMM V_{max} 4	54(18)	**8/8**	99.38	275.03	2974MB
RMM V_{min} 4	180(60)	7/8	175.65		2293MB
Heisenberg Ladderr215S(24)					
BKDV 4	98	**8/8**		372.53	2520MB
BKDV 4	35	**8/8**	130.89	186.61	2520MB
RMM V_{min} 4	57(19)	**8/8**	55.73		2293MB
RMM V_{max} 4	153(51)	**8/8**		272.27	2954MB
RMM V_{max} 4	60(20)	**8/8**	106.31	135.85	2974MB
RMM V_{min} 4	30(10)	**8/8**	29.54		2293MB
maximum iteration m = 240, combination method: the 1st part max iteration = 60, tolerance = 5×10^{-2} the 2nd part max iteration = 180, tolerance = 5×10^{-4}					

Table 4.26 Comparison of the results of Ladder315C and Ladder315S models with N = 24 on Odin

Methods name	Number of iteration (block)	Convergent	Clock time (second)		Memory
Heisenberg Ladderr315C(24)					
BKDV 4	240	7/8		971.48	2585MB
BKDV 4	40	7/8	241.10	429.17	2585MB
RMM V_{min} 4	180(60)	7/8	188.07		2358MB
RMM V_{max} 4	240(80)	7/8		447.37	3018MB
RMM V_{max} 4	60(20)	7/8	112.02	225.18	3038MB
RMM V_{min} 4	108(36)	**8/8**	113.16		2358MB
Heisenberg Ladderr315S(24)					
BKDV 4	115	**8/8**		461.22	2585MB
BKDV 4	60	7/8	240.46	328.16	2585MB
RMM V_{min} 4	81(27)	**8/8**	87.71		2358MB
RMM V_{max} 4	150(50)	**8/8**		280.08	3018MB
RMM V_{max} 4	57(19)	**8/8**	106.33	144.11	3038MB
RMM V_{min} 4	36(12)	**8/8**	37.78		2358MB
maximum iteration m = 240, combination method: the 1st part max iteration = 60, tolerance = 5×10^{-2} the 2nd part max iteration = 180, tolerance = 5×10^{-4}					

Table 4.27 Calculations of Chain(24) model by 4 different methods

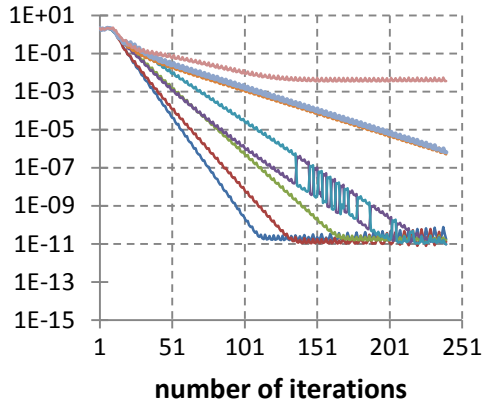


Figure 4.21 Results of Chain(24) by **BKDV**

total time = 813.23s,

iteration = 240

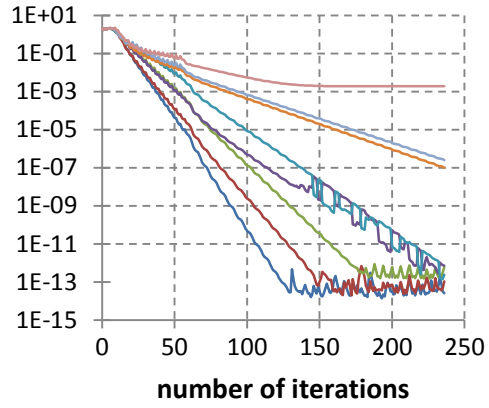


Figure 4.22 Results of Chain(24) by **BKDV** and **RMM** $\boldsymbol{V_{min}}$

total time = 334.55s,

iteration = 57/180

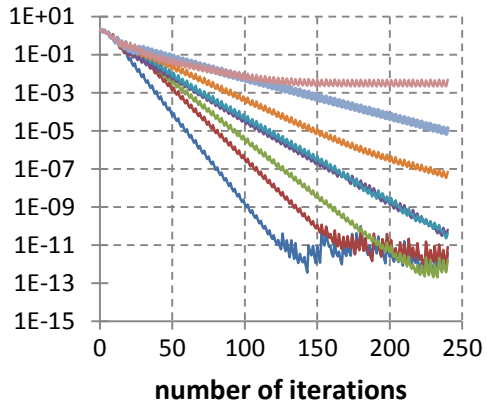


Figure 4.23 Results of Chain(24) by **RMM** $\boldsymbol{V_{max}}$

total time = 395.94s,

iteration = 240(80) on Odin

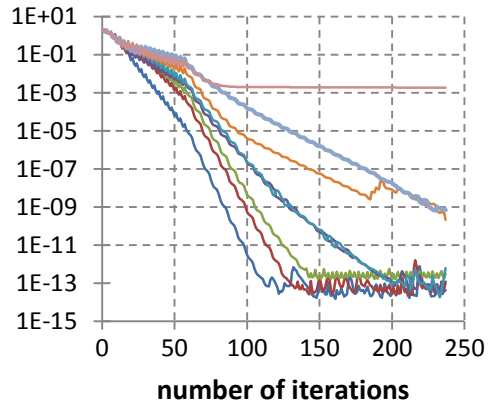


Figure 4.24 Results of Chain(24) by **RMM** $\boldsymbol{V_{max}}$ and **RMM** $\boldsymbol{V_{min}}$

total time = 239.35s,

iteration = 57/180 on Odin

Table 4.28 Calculations of Ladder2(24) model by 4 different methods

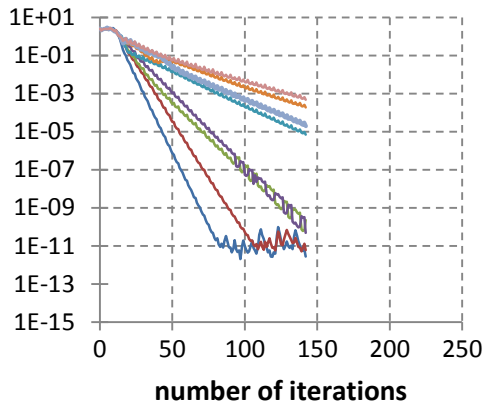
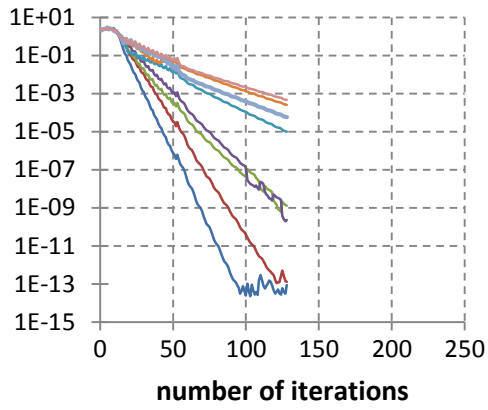
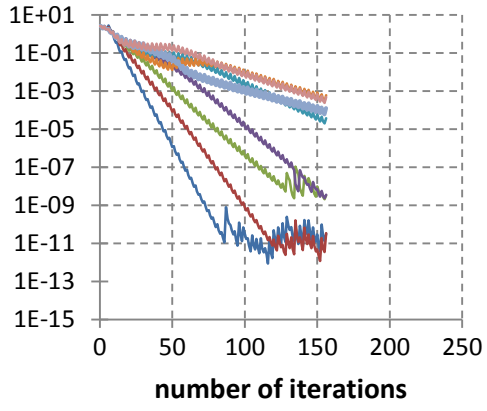
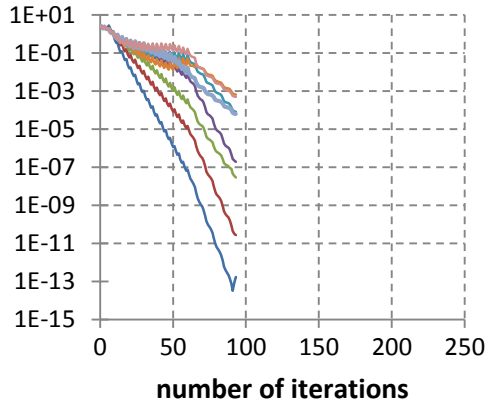
Norm of $\|r>$		
	Figure 4.25 Results of Ladder2(24) by **BKDV** total time = 549.74s, iteration = 143 on Odin	Figure 4.26 Results of Ladder2(24) by **BKDV** and **RMM** V_{min} total time = 274.38, iteration = 53/75 on Odin
Norm of $\|r>$		
	Figure 4.27 Results of Ladder2(24) by **RMM** V_{max} total time = 284.16s, iteration = 153 on Odin	Figure 4.28 Results of Ladder2(24) by **RMM** V_{max} and **RMM** V_{min} total time =,142.35 iteration = 60/33 on Odin

Table 4.29 Calculations of Ladder215C(24) model by 4 different methods

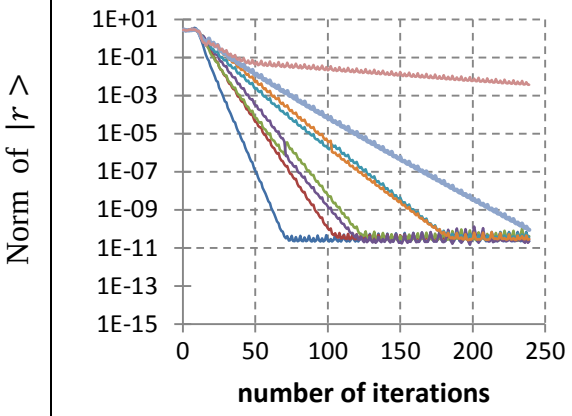


Figure 4.29 Results of Ladder215C(24) by **BKDV**

total time = 932.03s,

iteration = 240 on Odin

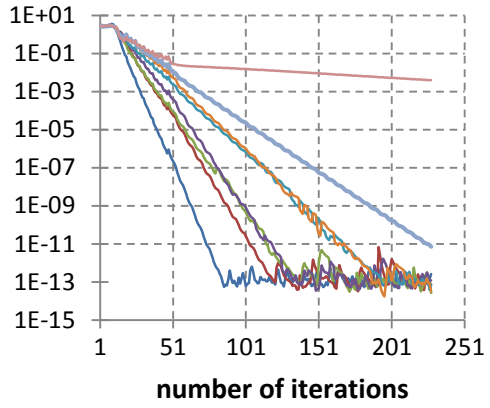


Figure 4.30 Results of Ladder215C(24) model by **BKDV** and **RMM** $\boldsymbol{V_{min}}$

total time = 354.25s,

iteration = 47/180 on Odin

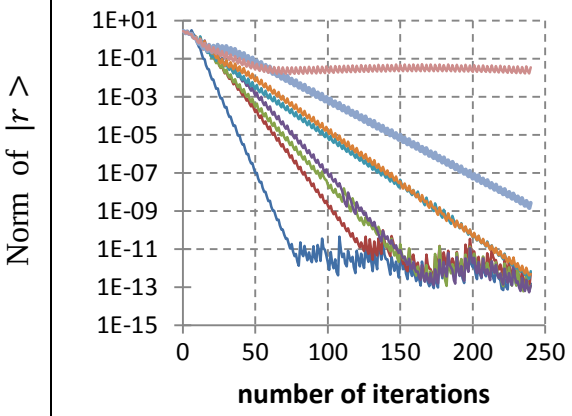


Figure 4.31 Results of Ladder215C(24) by **RMM** $\boldsymbol{V_{max}}$

total time = 438.45s,

iteration = 240 on Odin

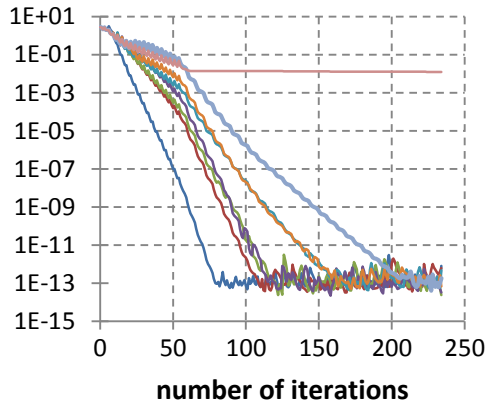


Figure 4.32 Results of Ladder215C(24) by **RMM** $\boldsymbol{V_{max}}$ and **RMM** $\boldsymbol{V_{min}}$

total time = 275.03s,

iteration = 54/180 on Odin

Table 4.30 Calculations of Ladder215S(24) model by 4 different methods

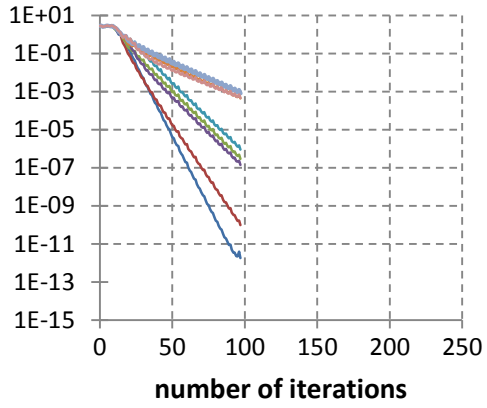
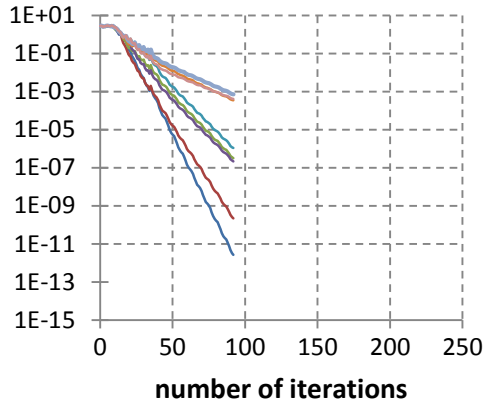
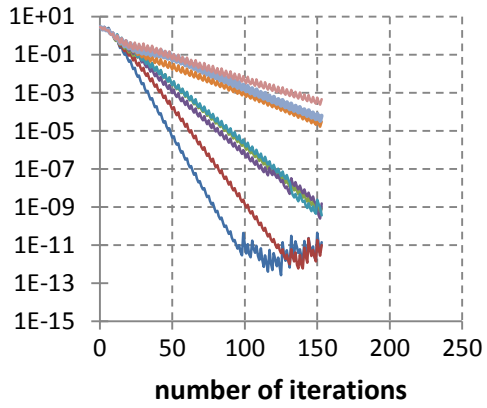
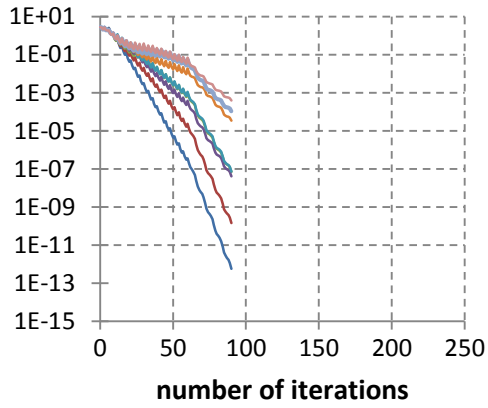
Norm of $\|r>$		
	Figure 4.33 Results of Ladder215S(24) by **BKDV** total time = 372.53s, iteration = 98 on Odin	Figure 4.34 Results of Ladder215S(24) by **BKDV** and **RMM** $\boldsymbol{V_{min}}$ total time =186.61s, iteration =35/57 on Odin
Norm of $\|r>$		
	Figure 4.35 Results of Ladder215S(24) by **RMM** $\boldsymbol{V_{max}}$ total time = 272.27s, iteration = 153 on Odin	Figure 4.36 Results of Ladder215S(24) by **RMM** $\boldsymbol{V_{max}}$ and **RMM** $\boldsymbol{V_{min}}$ total time = 135.85s, iteration = 60/30 on Odin

Table 4.31 Calculations of Ladder315C(24) model by 4 different methods

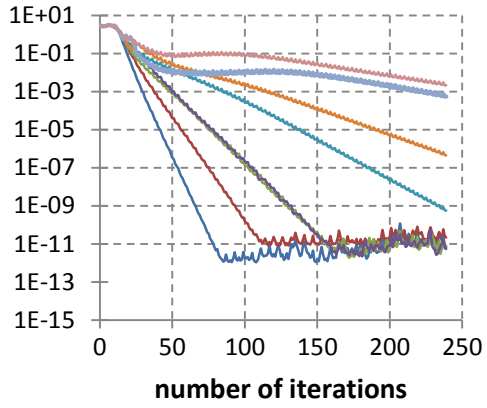
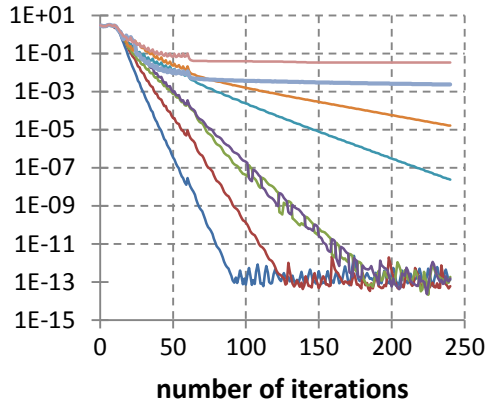
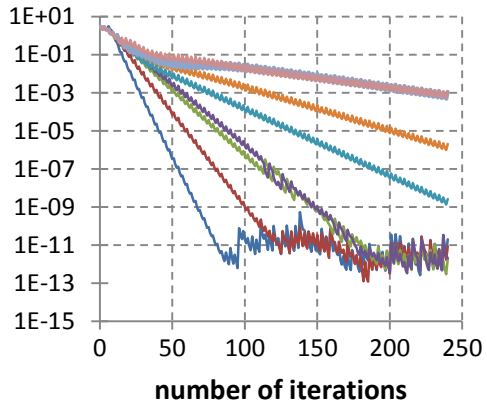
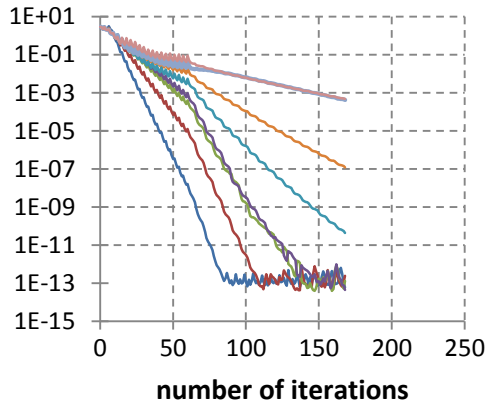
Norm of $\|r>$		
	Figure 4.37 Results of Ladder315C(24) by **BKDV** total time = 971.48s, iteration =240 on Odin	Figure 4.38 Results of Ladder315C(24) by **BKDV** and **RMM** $\boldsymbol{V_{min}}$ total time = 429.17s, iteration = 60/180 on Odin
Norm of $\|r>$		
	Figure 4.39 Results of Ladder315C(24) by **RMM** $\boldsymbol{V_{max}}$ total time = 447.37s, iteration = 240 on Odin	Figure 4.40 Results of Ladder315C(24) by **RMM** $\boldsymbol{V_{max}}$ and **RMM** $\boldsymbol{V_{min}}$ total time = 225.18s, iteration = 60/108 on Odin

Table 4.32 Calculations of Ladder315S(24) model by 4 different methods

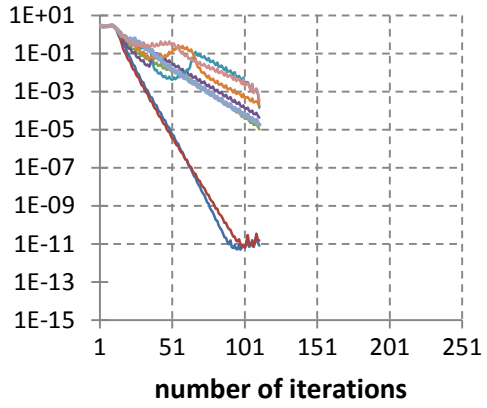


Figure 4.41 results of Ladder315C(24) by **BKDV**

total time = 461.22s,

iteration =115 on Odin

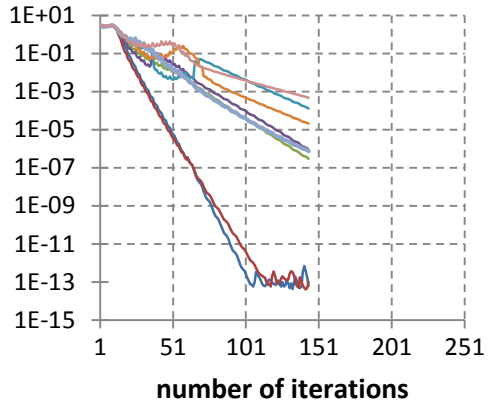


Figure 4.42 results of Ladder315S(24) by **BKDV** and **RMM** V_{min}

total time = 328.16s,

iteration =60/81 on Odin

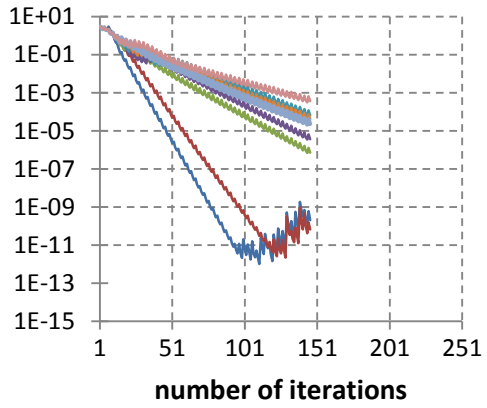


Figure 4.43 Results of Ladder31SC(24) by **RMM** V_{max}

total time = 280.08s,

iteration = 150 on Odin

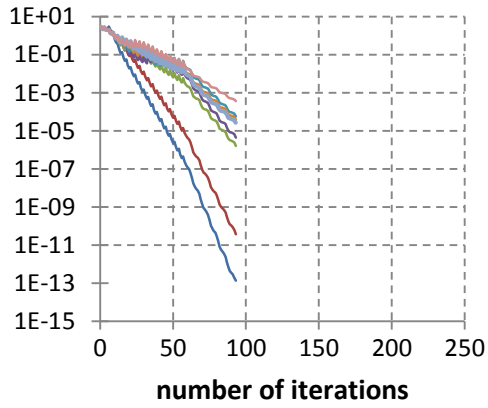


Figure 4.44 Results of Ladder315S(24) by **RMM** V_{max} and **RMM** V_{min}

total time = 144.11s,

iteration =57/36 on Odin

Table 4.33 Comparison of the results of Chain, Ladder2, Ladder215C and Ladder215S models with N = 18 on Odin

Methods	Number of iteration (block)	Conv ergent	Clock time (second)		Memory
Heisenberg Chain (18)					
BKDV 4	240	7/8		4.29	689MB
BKDV 4	29	**8/8**	0.54	1.14	689MB
RMM V_{min} 4	180(60)	7/8	0.60		689MB
RMM V_{max} 4	240(80)	7/8		1.93	694MB
RMM V_{max} 4	48(16)	**8/8**	0.34	0.93	694MB
RMM V_{min} 4	180(60)	7/8	0.59		689MB
Heisenberg Ladderr2(18)					
BKDV 4	240	6/8		5.32	692MB
BKDV 4	29	**8/8**	0.62	1.31	692MB
RMM V_{min} 4	180(60)	6/8	0.69		692MB
RMM V_{max} 4	240(80)	6/8		2.05	699MB
RMM V_{max} 4	39(13)	**8/8**	0.32	1.01	699MB
RMM V_{min} 4	180(60)	6/8	0.69		692MB
Heisenberg Ladderr215C(18)					
BKDV 4	240	7/8		5.15	692MB
BKDV 4	60	7/8	1.28	1.96	692MB
RMM V_{min} 4	180(60)	7/8	0.68		692MB
RMM V_{max} 4	240(60)	7/8		2.02	699MB
RMM V_{max} 4	60(20)	7/8	0.46	0.76	699MB
RMM V_{min} 4	78(26)	**8/8**	0.30		692MB
Heisenberg Ladderr215S(18)					
BKDV 4	76	**8/8**		1.66	692MB
BKDV 4	105(35)	**8/8**	0.74	0.91	692MB
RMM V_{min} 4	42(13)	**8/8**	0.16		692MB
RMM V_{max} 4	93(31)	**8/8**		0.85	699MB
RMM V_{max} 4	39(13)	**8/8**	0.30	0.38	699MB
RMM V_{min} 4	21(7)	**8/8**	0.08		692MB
maximum iteration m = 240, combination method: the 1st part max iteration = 60, tolerance = 5×10^{-2} the 2nd part max iteration = 180, tolerance = 5×10^{-4}					

Table 4.34 Comparison of the results of Ladder315C and Ladder315S models with N = 18 on Odin

Methods	Number of iteration (block)	Convergent	Clock time (second)		Memory
Heisenberg Ladderr315C(18)					
BKDV 4	129	**8/8**	2.99		692MB
BKDV 4	66(22)	**8/8**	0.72	1.10	692MB
RMM V_{min} 4	90(30)	**8/8**	0.38		692MB
RMM V_{max} 4	153(51)	**8/8**	1.36		700MB
RMM V_{max} 4	45(15)	**8/8**	0.36	0.54	700MB
RMM V_{min} 4	42(14)	**8/8**	0.18		692MB
Heisenberg Ladderr315S(18)					
BKDV 4	89	**8/8**	2.08		692MB
BKDV 4	87(29)	**8/8**	0.66	0.90	692MB
RMM V_{min} 4	57(19)	**8/8**	0.24		692MB
RMM V_{max} 4	123(41）	**8/8**	1.18		700MB
RMM V_{max} 4	45(15)	**8/8**	0.37	0.51	700MB
RMM V_{min} 4	33(11)	**8/8**	0.14		692MB
maximum iteration m = 240, combination method: the 1st part max iteration = 60, tolerance = 5×10^{-2} the 2nd part max iteration = 180, tolerance = 5×10^{-4}					

4.3 Parallel Speedup on Odin

All results in section 4.3 are calculated with the RMM-DIIS V_{max} and V_{min} method on a dual hex-core machine with 2×Intel X5650 Xeon processor（2.67GHz） and 12 GB of DDR3-1333 RAM, we called this machine Odin. The processor of Odin has 12 cores. I tested 6 different models with size *N* = 26, 24, 22, 20, 18 and 16.

Table 4.35 Speedup of Chain Ladder2 Ladder215C and Ladder215S models with different size *N* = 26, 24, and 22 on Odin

Model		Chain(*N*)		Ladder2(*N*)		Ladder215C(*N*)		Ladder215S(*N*)	
Size	Thread	time	speedup	time	speedup	time	speedup	time	speedup
N = 26	1	1169.72	1.00	1488.73	1.00	1450.58	1.00	1486.58	1.00
	2	701.79	1.67	848.05	1.76	860.30	1.69	866.39	1.72
	4	580.98	2.01	673.45	2.21	677.05	2.14	692.59	2.15
	6	541.21	2.16	628.30	2.37	624.19	2.32	626.17	2.37
	8	540.50	2.16	613.73	2.43	601.43	2.41	604.53	2.46
	10	541.37	2.16	605.72	2.46	605.47	2.40	603.95	2.46
	12	536.10	2.18	618.36	2.41	605.73	2.39	623.32	2.38
N = 24	1	256.06	1.00	307.08	1.00	314.33	1.00	317.40	1.00
	2	181.14	1.41	218.76	1.40	217.28	1.45	213.90	1.48
	4	152.09	1.68	181.19	1.69	180.05	1.75	175.99	1.80
	6	147.76	1.73	170.64	1.80	169.13	1.86	169.10	1.88
	8	146.09	1.75	169.46	1.81	166.61	1.89	170.62	1.86
	10	150.71	1.70	174.97	1.76	167.96	1.87	176.56	1.80
	12	153.24	1.67	177.76	1.73	170.44	1.84	176.25	1.80
N = 22	1	60.15	1.00	70.89	1.00	70.92	1.00	70.19	1.00
	2	38.33	1.57	43.87	1.62	45.10	1.57	44.89	1.56
	4	29.97	2.01	34.85	2.03	36.80	1.93	35.91	1.95
	6	26.81	2.24	32.65	2.17	34.20	2.07	33.15	2.12
	8	26.38	2.28	32.20	2.20	33.75	2.10	33.29	2.11
	10	27.14	2.22	32.19	2.20	34.22	2.07	34.11	2.06
	12	26.55	2.27	32.43	2.19	34.54	2.05	34.58	2.03

(time in seconds)

Table 4.36 Speedup of Chain Ladder2 Ladder215C and Ladder215S models with different size N = 20, 18 and 16 on Odin

Model		Chain(N)		Ladder2(N)		Ladder215C(N)		Ladder215S(N)	
Size	Thread	time	speedup	time	speedup	time	speedup	time	speedup
N = 20	1	13.75	1.00	15.49	1.00	15.88	1.00	15.40	1.00
	2	8.13	1.69	9.63	1.61	9.53	1.67	9.66	1.59
	4	6.09	2.26	7.49	2.07	7.37	2.16	7.53	2.05
	6	5.54	2.48	6.64	2.33	6.53	2.43	6.80	2.27
	8	5.55	2.48	6.54	2.37	6.42	2.47	6.73	2.29
	10	5.48	2.51	6.66	2.33	6.42	2.47	6.86	2.25
	12	5.62	2.45	6.91	2.24	6.58	2.41	6.98	2.21
N = 18	1	2.98	1.00	3.53	1.00	3.52	1.00	3.41	1.00
	2	1.65	1.81	1.85	1.91	1.96	1.80	1.95	1.75
	4	0.95	3.13	1.15	3.07	1.15	3.07	1.05	3.27
	6	0.77	3.89	0.87	4.08	1.01	3.47	0.84	4.08
	8	0.74	4.04	0.79	4.48	0.79	4.44	0.73	4.67
	10	0.77	3.87	0.86	4.09	0.87	4.06	0.76	4.48
	12	0.95	3.14	1.02	3.47	1.01	3.47	0.96	3.54
N = 16	1	0.70	1.00	0.82	1.00	0.81	1.00	0.69	1.00
	2	0.41	1.70	0.47	1.74	0.47	1.72	0.43	1.60
	4	0.30	2.33	0.33	2.46	0.32	2.51	0.30	2.25
	6	0.31	2.25	0.33	2.46	0.33	2.49	0.31	2.23
	8	0.31	2.24	0.32	2.53	0.32	2.54	0.30	2.29
	10	0.40	1.73	0.38	2.16	0.40	2.04	0.39	1.74
	12	0.45	1.56	0.47	1.75	0.46	1.76	0.45	1.52

(time in seconds)

Table 4.37 Speedup of Ladder 315C and Ladder 315S models with different size $N = 24$ and 18 on an Intel Xeon Phi

Size	$N = 24$				$N = 18$			
Model	Ladder315C(N)		Ladder315S(N)		Ladder315C(N)		Ladder315S(N)	
Threads	time	speedup	time	speedup	time	speedup	time	speedup
1	351.83	1.00	361.07	1.00	3.92	1.00	3.92	1.00
2	249.30	1.41	247.25	1.46	2.20	1.78	2.39	1.64
4	198.73	1.77	191.09	1.89	1.23	3.17	1.29	3.04
6	187.16	1.88	180.32	2.00	0.98	3.99	1.03	3.80
8	180.08	1.95	177.01	2.04	0.91	4.31	1.00	3.93
10	183.46	1.92	181.74	1.99	0.98	4.01	0.96	4.10
12	187.74	1.87	186.23	1.94	1.02	3.84	1.07	3.65

(time in seconds)

According the Table 4.35, Table 4.36 and Table 4.37, the speedup coefficient are very similar for 8, 10 and 12 threads, In most case, the speedup can reach peak when 8 cores were used in calculations. The number of speedup is larger than 4 in most modes with $N = 18$; the maximum number of speedup of 4.67 occurred in the Ladder215S (18) model with 8 threads. Compare using single thread, each core assumes 4.67/8 = 0.58375 workload. Except when $N = 18$, the best speedup for each model and size is around 2. As before, best speedup were obtained when using 8 threads. The speedup of the Ladder models is a little better than for the Chain model's on Odin.

4.4 Parallel Speedup on the Intel Xeon Phi

In this section, all calculations were made with the RMM-DIIS V_{max} and V_{min} method on an Intel Xeon Phi coprocessor 5110P. The Intel Xeon Phi co-processor is a many-core processor based on Intel's Many-integrated Core architecture. It integrates 60 cores with 4 hardware threads per core on one chip. On this machine, the speedup depends on the placement of the threads on the cores, the so-called thread affinity type. The thread affinity can be selected by setting the environment variable KMP_THREAD_AFFINITY.

There are 3 affinity types supported on Intel Many Integrated Core Architecture Ref.[3]: `scatter`, `compact` and `balanced`. The threads are placed on separate cores until all cores have at least one thread under `scatter` allocation type. The threads are placed to one core until this core is full, and go to next one under `compact` allocation type. The thread allocation under `balanced` type is balanced over the cores and the threads allocated to a core are neighbors of each other.

Figure 4.45, Figure 4.46 and Figure 4.47 are simulations of threads allocation under `scatter`, `compact` and `balanced` type. The graphs illustrate the allocation in a 3-cores system, in each core allows 4 threads across.

Figure 4.45 shows `scatter` type, when the threads are placed on separate cores. After each core has a thread, the following threads will be placed to the core which has the least threads. `Scatter` type uses as many cores as possible.

In Figure 4.46, the threads allocate in one core, after allocating 4 threads, HT0 is full, and then the threads were placed to next core HT1, `compact` type lets threads across to one core, and allocates maximum number threads of this core, this method of allocation makes workload of each core is unbalanced when there is no enough workload, it uses cores as less as possible.

Figure 4.47 diagrams the `balanced` type, every two consecutive threads are placed to each core when there are 6 threads need to be placed. Each core will have 3 threads if there are 9 threads need to be placed. The threads allocated to a core are neighbors of each other. Hence cache utilization should be efficient if the threads access data that is near in store. The `balanced` type always gives an average number of threads to each core.

In this section, each result based on the average values of 5 tests, all tests table are in section 6.2 Speedup data on the Intel Xeon Phi.

I planned to record 6 different models with size 26, 24, 22, 20, 18, 16 by 3 different affinity types. There were some problems with Xeon Phi machine at the end of tests, hence, the reports with size $N = 26$ were not tested, except Ladder215C models. That's the reason why some data are missing. The first column of each table shows the number of threads from 1 to 240 (1, 10 to 240 in steps of 10).

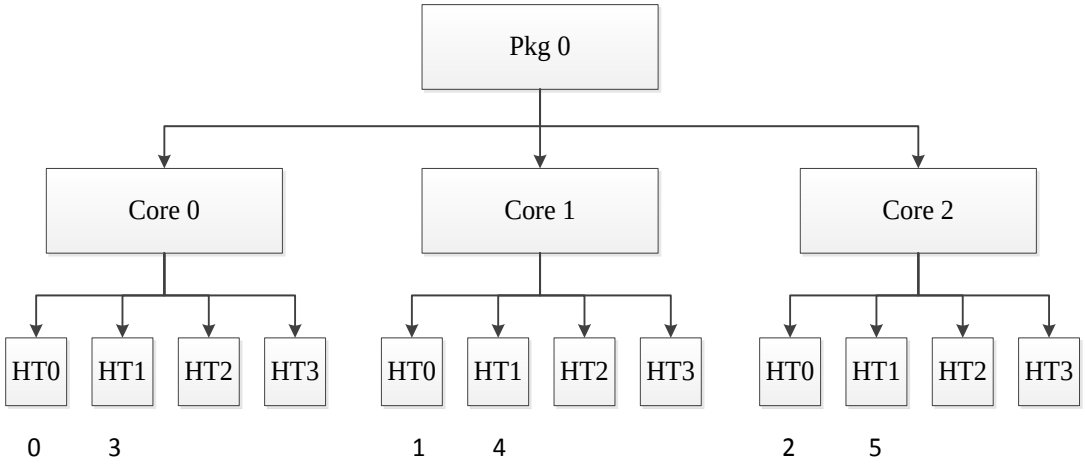


Figure 4.45 Allocation with **`scatter`** affinity type

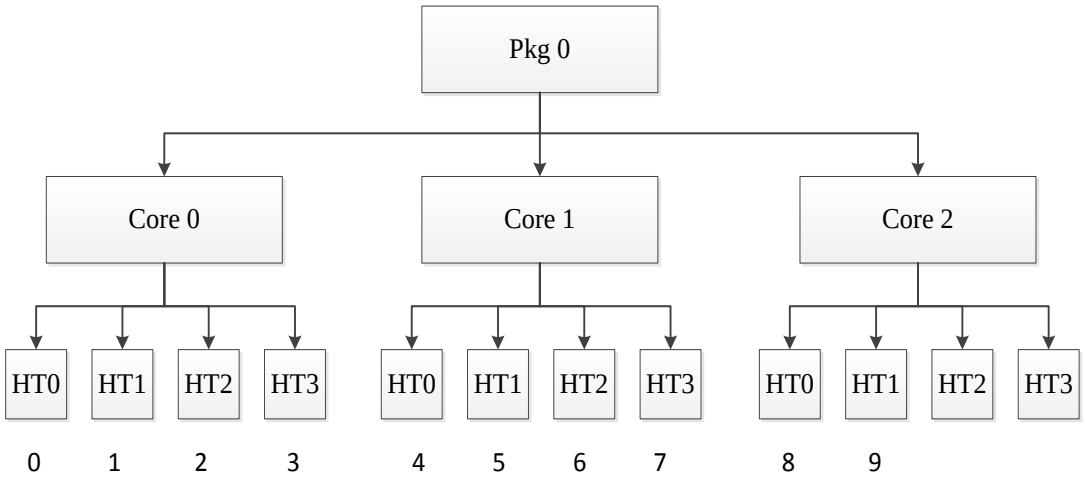


Figure 4.46 Allocation with **`compact`** affinity type

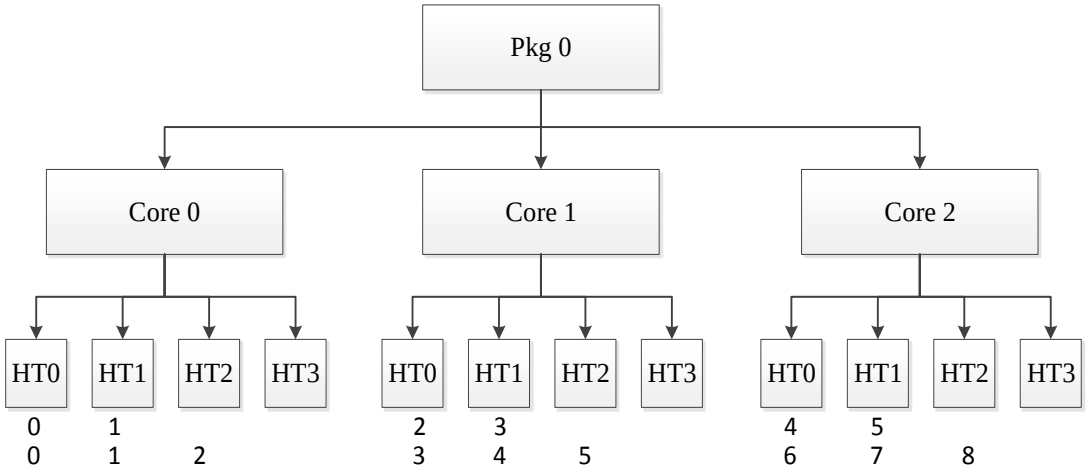


Figure 4.47 Allocation with **`balanced`** affinity type

(Line 1 has 6 threads, line 2 has 9 threads)

4.4.1 Heisenberg Chain

Table 4.38 Speedup of Chain(*N*) with *N* = 26, 24 and 22 on an Intel Xeon Phi

Thread	*N* = 26			*N* = 24			*N* = 22		
affinity type	Scatter	Compact	Balanced	Scatter	Compact	Balanced	Scatter	Compact	Balanced
1	–	–	–	1.00	1.00	1.00	1.00	1.00	1.00
10	–	–	–	4.34	3.11	4.10	4.24	3.11	3.47
20	–	–	–	7.23	5.94	6.86	7.29	5.70	5.82
30	–	–	–	9.98	8.63	9.56	10.09	8.12	8.00
40	–	–	–	13.27	11.28	12.37	13.79	10.47	10.67
50	–	–	–	16.08	13.93	15.55	16.57	12.54	12.91
60	–	–	–	18.90	16.14	18.11	18.92	14.56	14.76
70	–	–	–	20.72	18.51	20.15	20.44	16.57	16.11
80	–	–	–	23.82	21.12	22.65	22.97	19.17	18.92
90	–	–	–	25.50	22.90	24.34	24.23	20.96	19.69
100	–	–	–	27.39	25.02	25.73	26.17	23.48	21.34
110	–	–	–	28.66	26.89	27.56	28.02	25.61	22.44
120	–	–	–	29.96	28.58	29.25	29.34	27.38	23.70
130	–	–	–	30.24	30.26	30.28	29.77	28.86	24.50
140	–	–	–	31.29	31.89	31.78	30.56	30.53	25.88
150	–	–	–	33.21	33.59	33.37	25.61	26.15	6.80
160	–	–	–	33.57	35.62	35.09	26.06	26.36	6.15
170	–	–	–	33.94	35.48	35.38	26.07	25.81	5.43
180	–	–	–	35.99	38.57	37.66	25.46	25.70	4.76
190	–	–	–	36.89	40.44	38.63	25.35	25.50	4.79
200	–	–	–	37.67	41.53	40.36	24.80	24.85	3.84
210	–	–	–	39.35	42.67	41.72	24.45	24.18	3.54
220	–	–	–	40.63	43.18	41.90	23.99	23.38	3.75
230	–	–	–	43.06	44.89	44.96	23.73	22.52	3.36
240	–	–	–	43.36	45.93	45.99	22.33	21.63	3.08

Table 4.39 Speedup of Chain(*N*) with *N* = 20, 18 and 16 on an Intel Xeon Phi

Thread	*N* = 20			*N* = 18			*N* = 16		
affinity type	Scatter	Compact	Balanced	Scatter	Compact	Balanced	Scatter	Compact	Balanced
1	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
10	4.61	2.98	13.78	3.87	2.66	3.52	2.36	1.76	2.17
20	7.46	5.29	15.94	4.97	3.48	4.07	2.34	1.88	2.02
30	9.87	7.90	20.13	7.00	4.23	5.14	2.07	1.76	1.66
40	10.87	9.04	16.79	6.73	4.20	4.29	1.75	1.62	1.28
50	11.74	9.81	13.39	6.18	4.07	3.42	1.44	1.48	0.98
60	12.27	10.25	10.48	5.51	3.91	2.68	1.21	1.36	0.76
70	12.21	10.53	6.90	4.89	3.73	1.76	1.02	1.25	0.61
80	12.48	10.88	6.37	4.47	3.52	1.63	0.92	1.16	0.55
90	14.73	12.03	5.83	4.15	3.33	1.49	0.84	1.07	0.50
100	13.95	11.61	5.42	3.88	3.19	1.38	0.77	0.99	0.46
110	13.38	11.27	4.87	3.61	3.03	1.24	0.71	0.93	0.41
120	12.82	10.83	4.62	3.32	2.84	1.18	0.66	0.88	0.38
130	11.90	10.42	4.16	3.05	2.69	1.06	0.61	0.82	0.35
140	11.28	10.02	3.85	2.88	2.57	0.98	0.57	0.76	0.32
150	10.74	9.66	3.64	2.73	2.49	0.93	0.54	0.73	0.30
160	10.29	9.31	3.41	2.58	2.35	0.87	0.51	0.68	0.27
170	9.87	8.97	3.21	2.46	2.24	0.82	0.48	0.64	0.26
180	9.49	8.64	3.04	2.34	2.15	0.78	0.47	0.61	0.24
190	8.73	8.34	3.01	2.20	2.06	0.77	0.44	0.58	0.23
200	8.44	8.04	2.85	2.11	1.98	0.73	0.42	0.55	0.22
210	8.14	7.74	2.68	2.03	1.90	0.69	0.40	0.52	0.20
220	7.90	7.49	2.52	1.95	1.84	0.64	0.38	0.50	0.19
230	7.68	7.22	2.36	1.88	1.76	0.60	0.37	0.47	0.18
240	7.36	6.95	2.27	1.81	1.70	0.58	0.35	0.44	0.17

The Table 4.38 and Table 4.39 summarize 18 tables: from Table 6.10 to Table 6.27

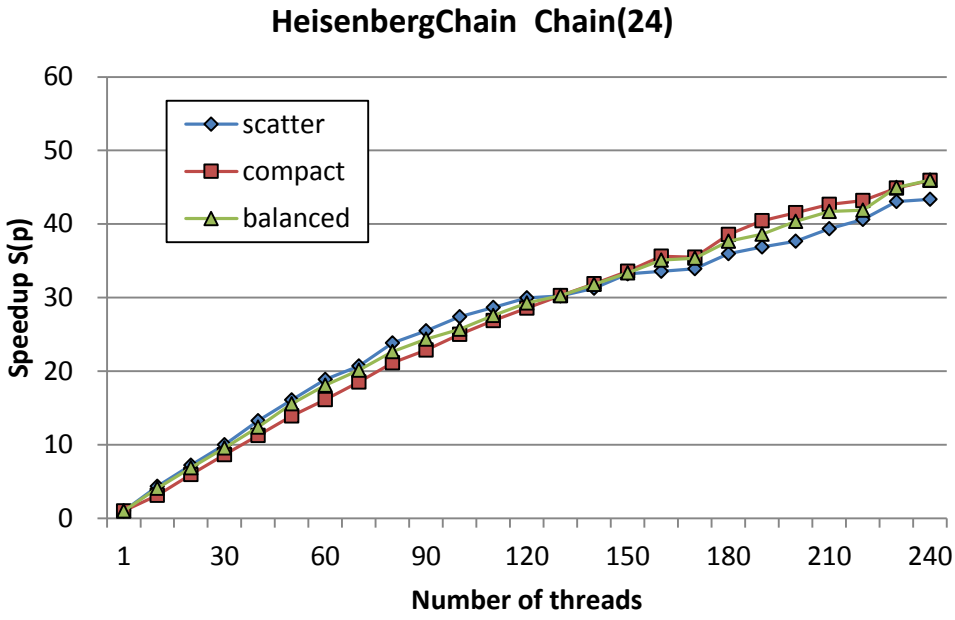


Figure 4.48 Parallel speedup of Chain(24) model on an Intel Xeon Phi with KMP_THREAD_AFFINITY set to scatter, compact and balanced

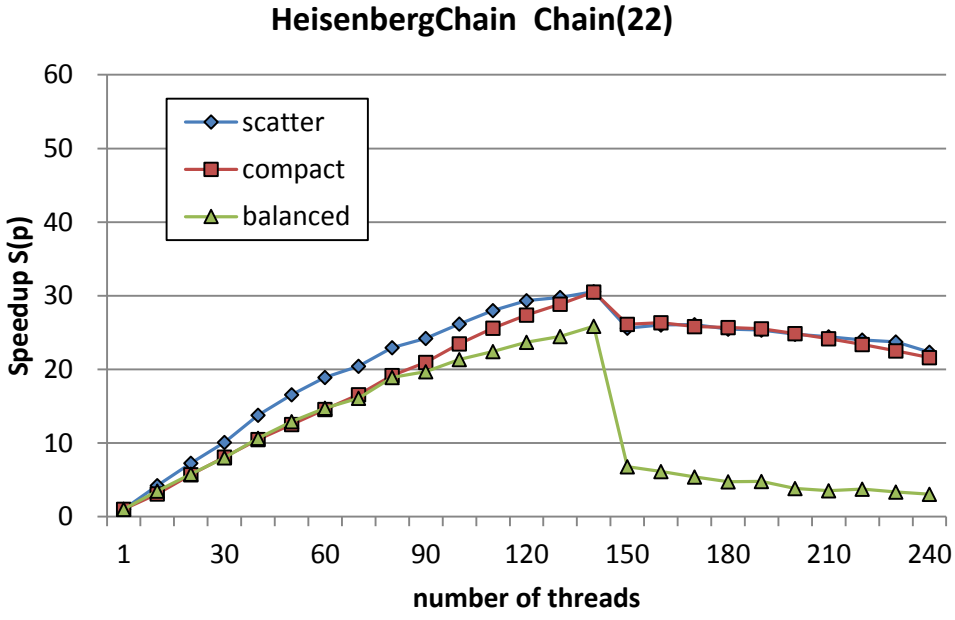


Figure 4.49 Parallel speedup of Chain(22) model on an Intel Xeon Phi with KMP_THREAD_AFFINITY set to scatter, compact and balanced

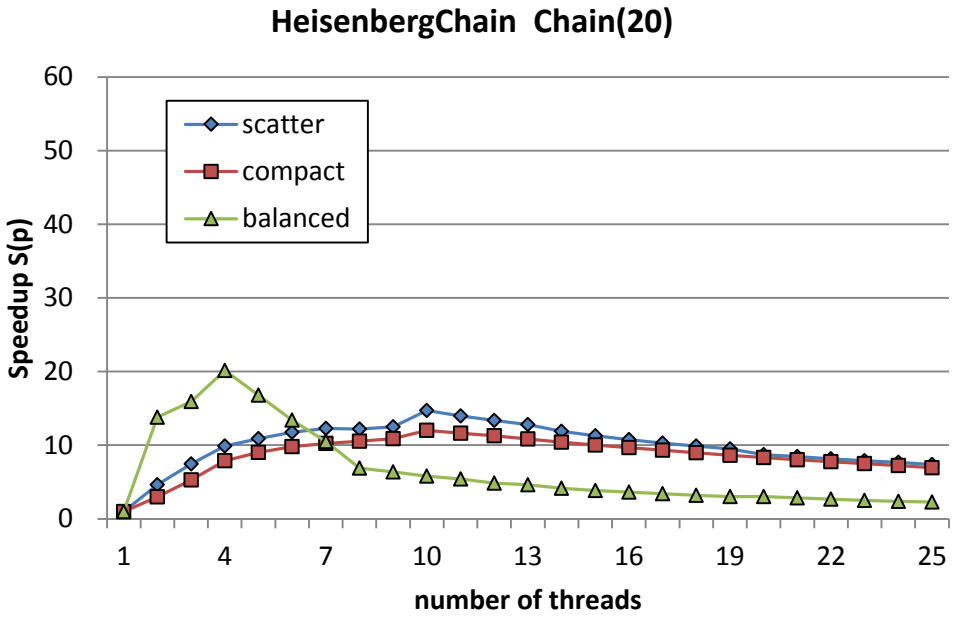


Figure 4.50 Parallel speedup of Chain(24) model on an Intel Xeon Phi with KMP_THREAD_AFFINITY set to scatter, compact and balanced

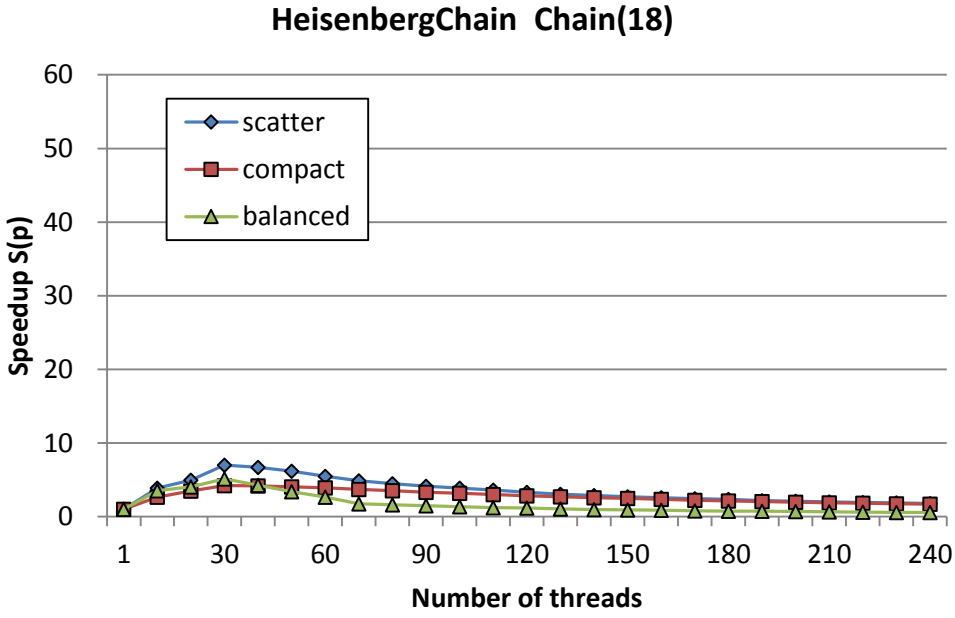


Figure 4.51 Parallel speedup of Chain(18) model on an Intel Xeon Phi with KMP_THREAD_AFFINITY set to scatter, compact and balanced

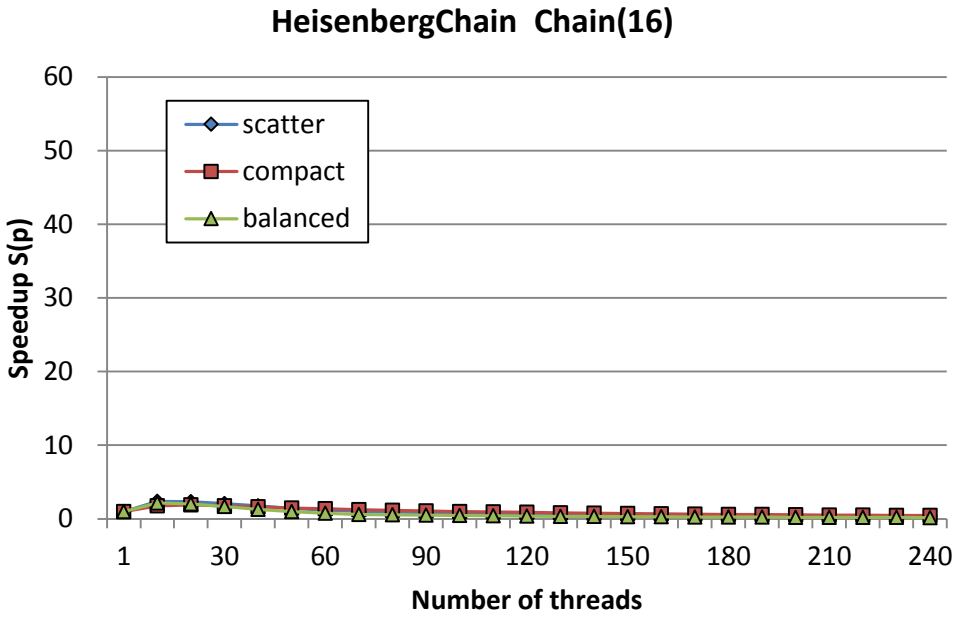


Figure 4.52 Parallel speedup of Chain(16) model on an Intel Xeon Phi with KMP_THREAD_AFFINITY set to scatter, compact and balanced

According the Figure 4.48, Figure 4.49, Figure 4.50, Figure 4.51 and Figure 4.52 as well as Table 4.38 and Table 4.39, we can find the largest speedup of the Chain model for $N = 24$. It is approximate 46 times faster than the execution time using a single thread.

When size N is equal to 24 or larger than 24, the speedup graph appears to be a linear which looks like monotonically increasing function, that shows, the number of speedup is growing when the number of thread is increasing. Before the number of thread increase to 130, speedup of `Scatter` is better than the other two. After thread number reaches to 140, `Compact` and `Balanced` type is better than `Scatter`.

4.4.2 Heisenberg Ladder

Table 4.40 Speedup of Ladde2(*N*) with *N* = 26, 24 and 22 on an Intel Xeon Phi

Thread	*N* = 26			*N* = 24			*N* = 22		
affinity type	Scatter	Compact	Balanced	Scatter	Compact	Balanced	Scatter	Compact	Balanced
1	–	–	–	1.00	1.00	1.00	1.00	1.00	1.00
10	–	–	–	4.25	3.37	4.25	4.05	3.26	4.03
20	–	–	–	7.43	6.30	7.10	6.96	5.89	7.24
30	–	–	–	9.77	9.24	10.41	10.20	8.44	10.49
40	–	–	–	13.09	12.00	12.70	13.35	10.91	13.10
50	–	–	–	15.50	14.77	15.34	15.94	12.99	16.13
60	–	–	–	18.06	17.20	18.07	18.41	15.26	18.68
70	–	–	–	20.25	19.88	20.07	20.41	17.46	20.36
80	–	–	–	22.88	22.55	22.86	23.19	20.09	23.08
90	–	–	–	25.16	24.59	25.12	23.85	22.07	24.32
100	–	–	–	26.52	26.77	27.59	26.16	24.47	26.56
110	–	–	–	28.47	28.91	29.36	28.40	26.87	28.30
120	–	–	–	30.19	30.82	31.27	29.71	28.88	29.37
130	–	–	–	31.80	32.67	32.39	30.52	30.39	30.18
140	–	–	–	33.81	34.69	34.58	31.83	32.49	32.04
150	–	–	–	35.00	36.66	35.44	27.96	28.44	8.45
160	–	–	–	35.86	38.83	37.76	28.31	28.84	7.61
170	–	–	–	35.78	39.01	38.15	28.15	28.44	6.72
180	–	–	–	37.93	42.01	40.76	27.82	28.32	5.88
190	–	–	–	38.11	44.10	41.19	27.39	28.18	5.94
200	–	–	–	40.22	45.36	42.99	26.97	27.42	4.72
210	–	–	–	42.18	46.68	45.16	26.71	26.73	4.35
220	–	–	–	42.70	47.50	45.51	26.30	25.90	4.63
230	–	–	–	46.51	49.37	49.46	26.03	25.06	4.15
240	–	–	–	47.29	50.86	50.86	25.50	24.20	3.78

Table 4.41 Speedup of Ladder2(*N*) with *N* = 20, 18 and 16 on an Intel Xeon Phi

Thread	*N* = 20			*N* = 18			*N* = 16		
affinity type	Scatter	Compact	Balanced	Scatter	Compact	Balanced	Scatter	Compact	Balanced
1	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
10	4.43	3.11	4.51	4.01	2.92	3.61	2.65	1.93	2.43
20	7.63	5.55	7.45	5.22	3.98	4.52	2.65	2.07	2.38
30	10.18	8.26	10.06	7.45	4.82	6.02	2.40	1.98	2.01
40	11.45	9.71	8.74	6.88	4.90	5.32	2.06	1.83	1.62
50	12.57	10.64	8.81	6.03	4.73	4.54	1.72	1.65	1.29
60	13.32	11.35	8.48	5.11	4.59	3.73	1.43	1.52	1.02
70	13.56	11.81	4.68	4.56	4.32	2.41	1.28	1.39	0.82
80	13.90	12.23	4.09	4.18	4.10	2.23	1.15	1.28	0.74
90	13.51	12.06	3.18	3.84	3.83	2.07	1.04	1.18	0.66
100	13.34	11.90	2.74	3.53	3.61	1.93	0.95	1.10	0.60
110	13.06	11.69	2.70	3.27	3.38	1.78	0.88	1.02	0.54
120	12.65	11.50	2.66	3.03	3.16	1.67	0.82	0.94	0.50
130	13.78	12.19	2.37	2.78	2.97	1.50	0.75	0.87	0.46
140	13.24	11.72	2.72	2.61	2.80	1.40	0.70	0.81	0.42
150	12.65	11.37	2.30	2.47	2.63	1.31	0.67	0.75	0.39
160	12.11	10.96	2.03	2.34	2.48	1.21	0.63	0.70	0.36
170	11.57	10.60	1.98	2.22	2.31	1.14	0.60	0.66	0.33
180	11.29	10.22	1.95	2.11	2.17	1.08	0.57	0.61	0.31
190	10.68	9.88	2.08	1.99	2.03	1.07	0.54	0.57	0.30
200	10.26	9.55	2.07	1.90	1.91	1.02	0.51	0.54	0.28
210	9.97	9.20	1.85	1.83	1.78	0.97	0.49	0.50	0.27
220	9.68	8.90	1.69	1.75	1.68	0.91	0.47	0.47	0.25
230	9.33	8.59	1.64	1.69	1.58	0.86	0.45	0.43	0.23
240	9.05	8.30	1.60	1.62	1.50	0.82	0.44	0.41	0.22

The Table 4.40 and Table 4.41 summarize 18 tables: from Table 6.28 to Table 6.45 in section 6.2

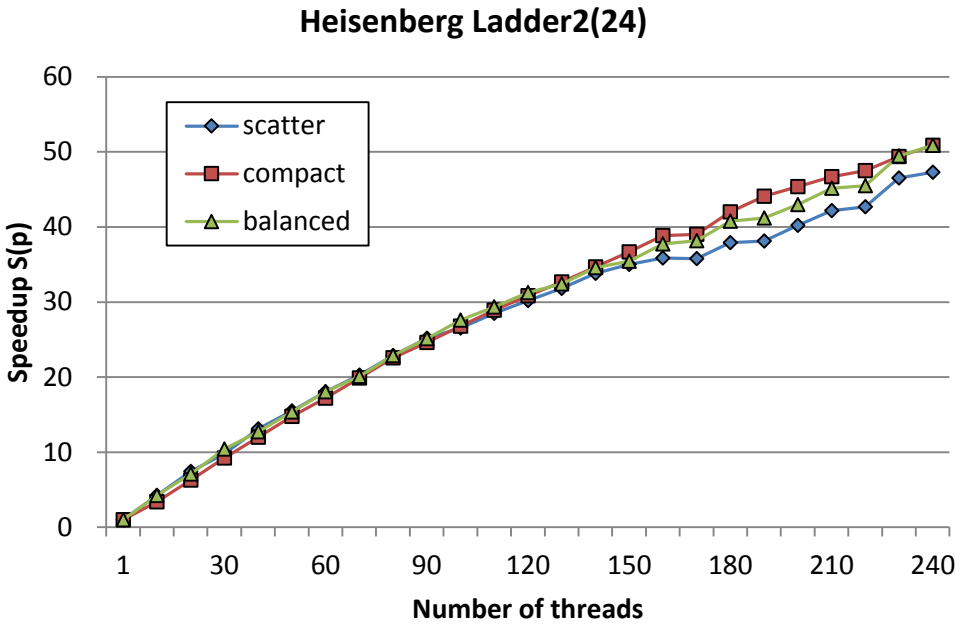


Figure 4.53 Parallel speedup of Ladder2(24) model on an Intel Xeon Phi with KMP_THREAD_AFFINITY set to scatter, compact and balanced

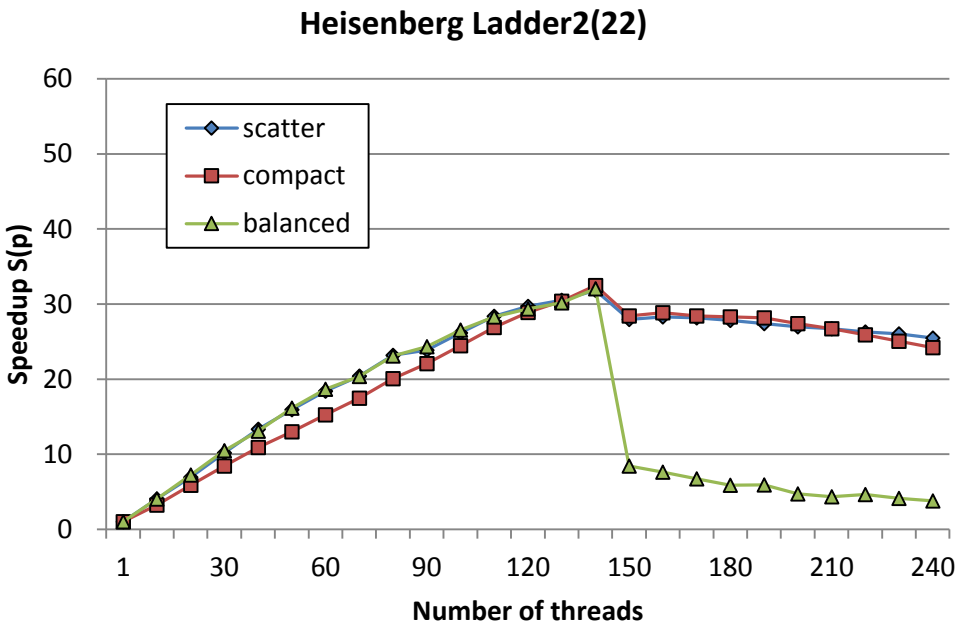


Figure 4.54 Parallel speedup of Ladder2(22) model on an Intel Xeon Phi with KMP_THREAD_AFFINITY set to scatter, compact and balanced

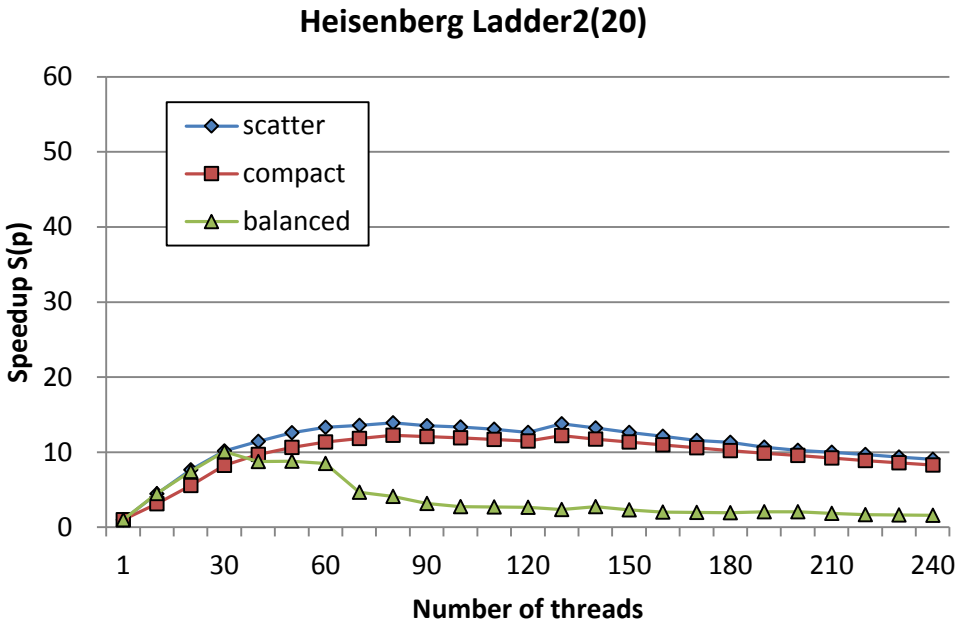


Figure 4.55 Parallel speedup of Ladder2(20) model on an Intel Xeon Phi with KMP_THREAD_AFFINITY set to scatter, compact and balanced

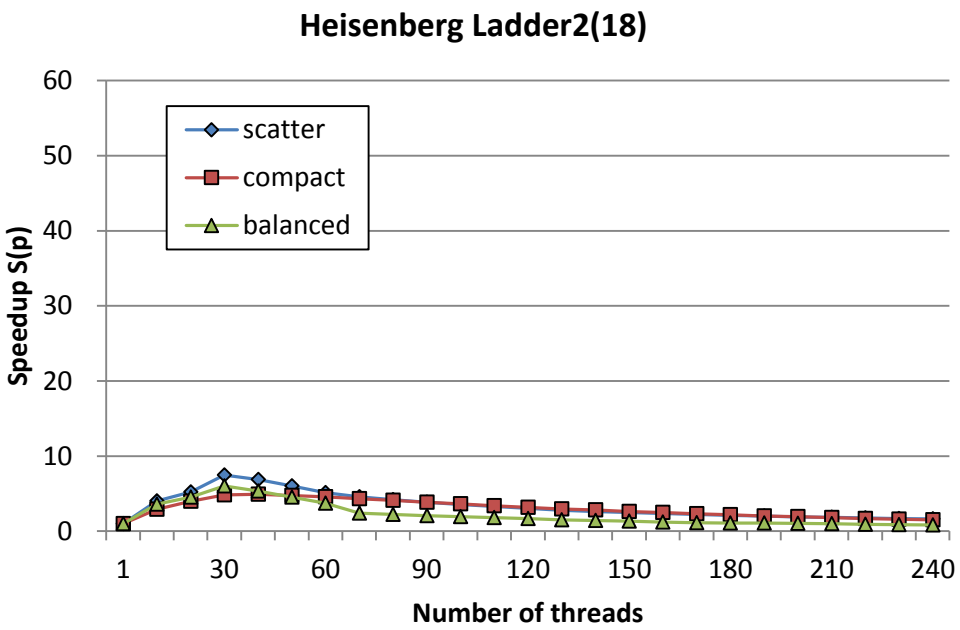


Figure 4.56 Parallel speedup of Ladder2(18) mode on an Intel Xeon Phi with KMP_THREAD_AFFINITY set to scatter, compact and balanced

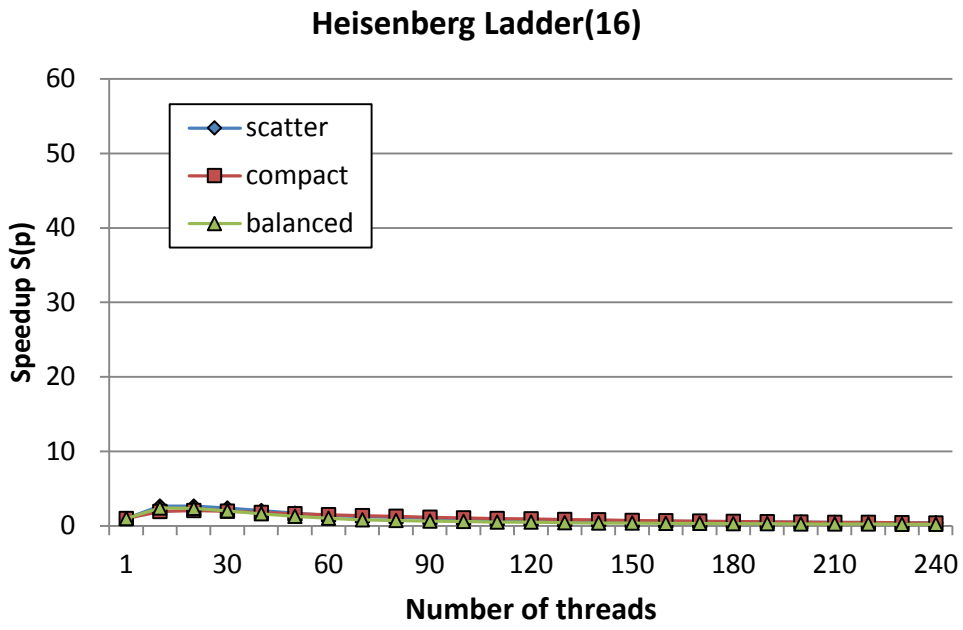


Figure 4.57 Parallel speedup of Ladder2(16) model on an Intel Xeon Phi with KMP_THREAD_AFFINITY set to scatter, compact and balanced

See above the Figure 4.53, Figure 4.54, Figure 4.55, Figure 4.56 and Figure 4.57, review Table 4.40 and Table 4.41, we can find out the largest number of speedup of Ladder2 model is happened when $N = 24$, it is 50.86 times faster than the execution time using single thread.

When $N \geq 24$, the speedup graph appears a linear which looks like monotonically increasing function, that shows, the number of speedup is growing when the number of thread is increasing.

Before the number of thread increase to 130, speedup of `Scatter` is better than the other two. After thread number reaches to 140, `Compact` and `Balanced` type is better than `Scatter`.

4.4.3 Heisenberg Ladder 215C

Table 4.42 Speedup of Ladder 215C(*N*) with *N* = 26, 24 and 22 on Xeon Phi

Thread	*N* = 26			*N* = 24			*N* = 22		
affinity	Scatter	Compact	Balanced	Scatter	Compact	Balanced	Scatter	Compact	Balanced
1	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
10	4.16	3.44	4.07	4.18	3.38	4.34	4.40	3.26	4.22
20	7.11	6.54	7.21	7.29	6.29	7.30	7.21	5.89	7.38
30	9.92	9.50	10.29	9.92	9.21	9.97	9.99	8.43	10.27
40	13.26	12.61	13.27	12.68	12.00	12.97	12.97	10.90	13.27
50	15.62	15.65	15.91	15.51	14.77	15.71	16.04	13.00	15.99
60	18.39	18.43	18.43	18.30	17.20	18.12	18.87	15.26	17.85
70	20.67	21.40	20.64	20.57	19.88	20.32	20.88	17.47	19.84
80	23.36	24.35	23.49	22.85	22.55	23.27	22.81	20.10	22.54
90	25.35	26.75	25.56	25.28	24.58	24.88	24.30	22.04	24.29
100	26.55	29.28	28.18	27.45	26.76	27.04	26.40	24.46	26.44
110	29.57	31.88	31.09	29.44	28.91	29.05	28.27	26.81	28.14
120	30.98	34.11	32.88	30.14	30.85	30.85	29.76	28.79	29.50
130	32.54	36.39	34.13	31.62	32.71	32.06	30.78	30.38	30.42
140	34.81	38.91	36.43	33.88	34.71	34.19	32.17	32.45	32.03
150	35.18	41.21	38.21	35.19	36.65	36.01	28.34	28.59	8.45
160	37.89	43.81	40.49	36.14	38.88	38.39	28.09	29.00	7.61
170	39.14	45.69	41.55	36.66	38.99	38.59	27.93	28.66	6.69
180	40.43	47.68	43.27	37.94	42.06	40.54	27.94	28.65	5.87
190	41.52	50.27	45.33	37.84	44.17	41.51	27.54	28.63	5.92
200	42.12	51.82	47.37	39.98	45.40	43.13	27.12	27.96	4.73
210	44.30	53.65	49.12	42.49	46.72	45.27	26.82	27.38	4.36
220	46.42	55.11	50.42	43.88	47.59	45.55	26.55	26.66	4.64
230	50.18	56.83	56.79	46.66	49.39	49.51	26.25	25.91	4.14
240	50.81	58.37	58.21	47.27	50.89	50.92	25.68	25.23	3.79

Table 4.43 Speedup of Ladder 215C(N) with N = 20, 18 and 16 on an Intel Xeon Phi

Thread	N = 20			N = 18			N = 16		
affinity type	Scatter	Compact	Balanced	Scatter	Compact	Balanced	Scatter	Compact	Balanced
1	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
10	4.69	3.14	4.59	3.98	2.95	3.66	2.71	1.97	2.41
20	7.53	5.60	7.53	5.24	4.02	4.46	2.74	2.12	2.33
30	9.91	8.33	10.03	7.57	4.87	5.76	2.47	2.02	1.91
40	11.45	9.78	8.74	7.06	5.01	5.03	2.12	1.88	1.52
50	12.34	10.71	8.71	6.21	4.82	4.11	1.78	1.71	1.20
60	12.90	11.43	8.22	5.26	4.73	3.25	1.49	1.57	0.93
70	12.99	11.89	4.64	4.64	4.50	2.13	1.31	1.45	0.75
80	13.08	12.35	3.96	4.24	4.27	1.99	1.19	1.34	0.67
90	12.55	12.25	3.09	3.89	4.06	1.85	1.07	1.22	0.60
100	12.16	12.10	2.67	3.58	3.84	1.69	0.98	1.14	0.55
110	11.77	11.87	2.62	3.32	3.63	1.55	0.91	1.05	0.49
120	11.38	11.62	2.58	3.08	3.45	1.46	0.84	0.98	0.46
130	12.08	12.36	2.27	2.84	3.27	1.34	0.78	0.91	0.42
140	11.47	11.93	2.58	2.65	3.11	1.24	0.73	0.86	0.38
150	10.82	11.58	2.18	2.51	2.97	1.16	0.69	0.81	0.36
160	10.32	11.19	1.94	2.37	2.84	1.08	0.65	0.76	0.33
170	9.90	10.84	1.88	2.26	2.71	1.02	0.62	0.71	0.31
180	9.72	10.45	1.84	2.15	2.59	0.98	0.59	0.67	0.29
190	9.15	10.12	1.98	2.02	2.49	0.96	0.55	0.63	0.27
200	8.84	9.81	1.93	1.93	2.38	0.92	0.53	0.60	0.26
210	8.50	9.47	1.72	1.85	2.29	0.86	0.51	0.56	0.24
220	8.22	9.16	1.58	1.78	2.20	0.81	0.49	0.53	0.23
230	7.96	8.85	1.54	1.71	2.12	0.76	0.47	0.50	0.22
240	7.68	8.57	1.51	1.64	2.03	0.73	0.45	0.47	0.21

The Table 4.42 and Table 4.43 summarize 18 tables: from Table 6.46 to Table 6.63 in section 6.2

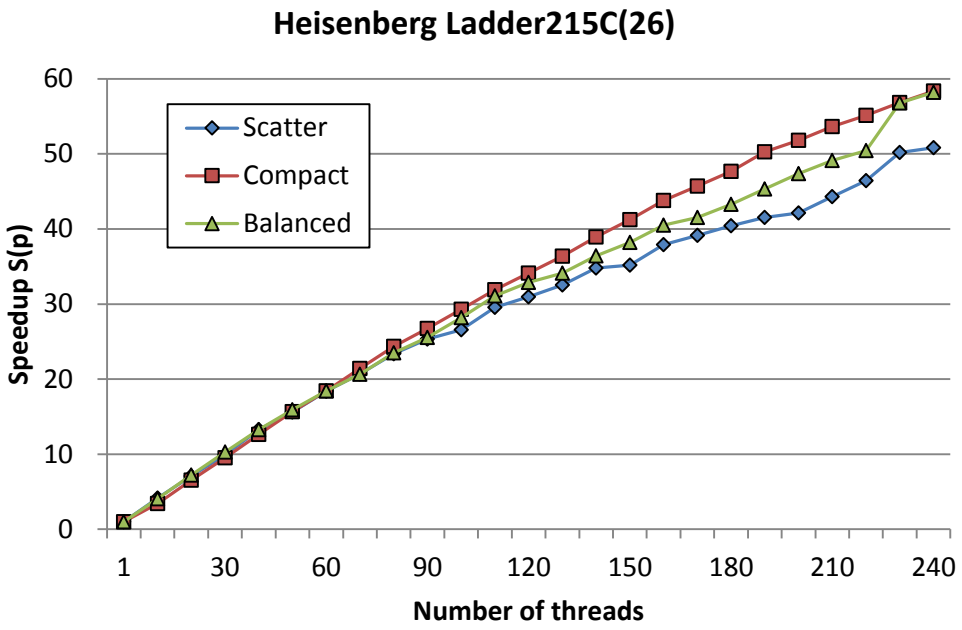


Figure 4.58 Parallel speedup of Ladder215C(26) model on an Intel Xeon Phi with KMP_THREAD_AFFINITY set to scatter, compact and balanced

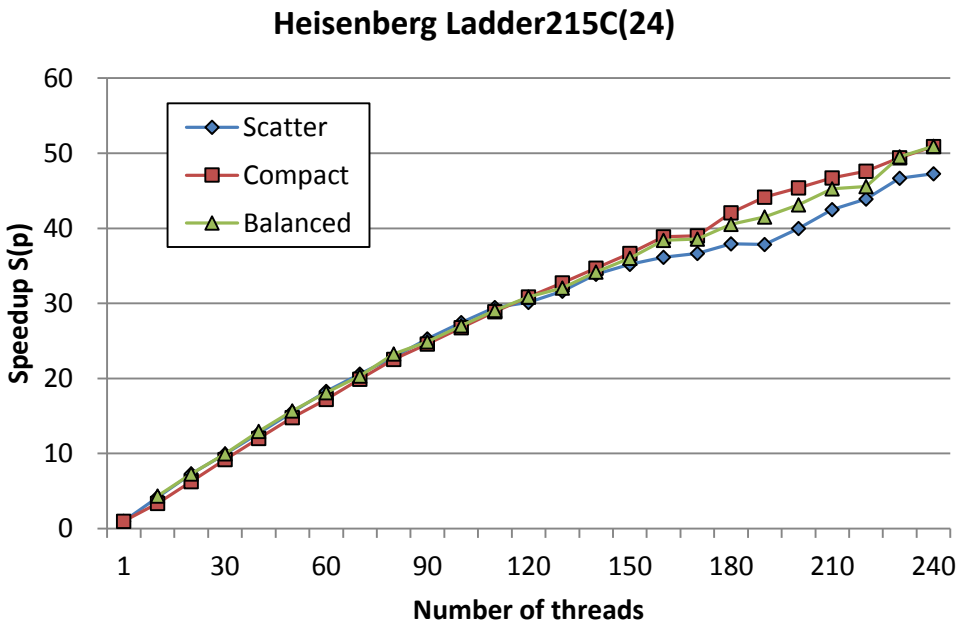


Figure 4.59 Parallel speedup of Ladder215C(24) model on an Intel Xeon Phi with KMP_THREAD_AFFINITY set to scatter, compact and balanced

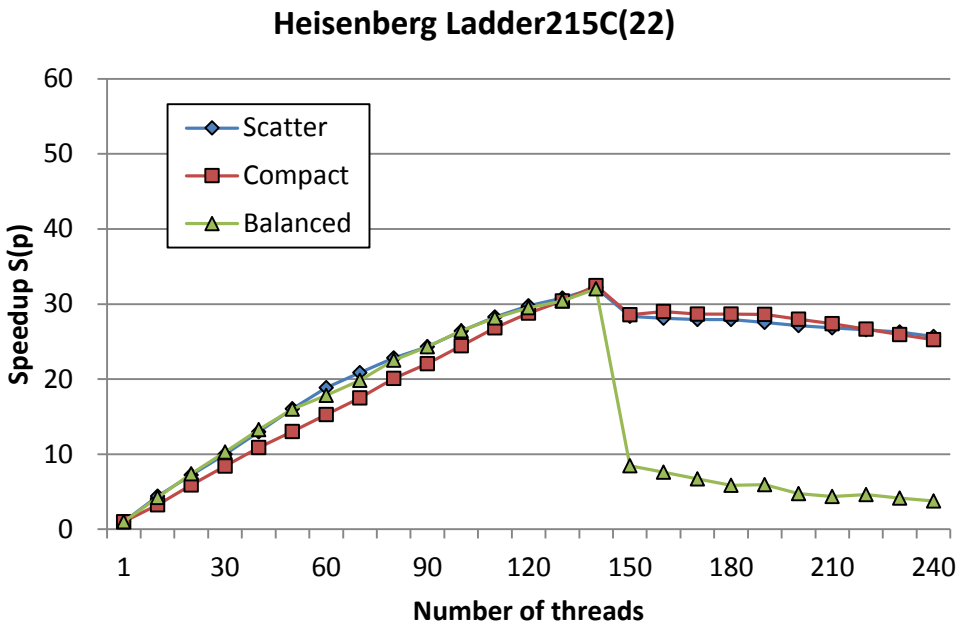


Figure 4.60 Parallel speedup of Ladder215C(22) model on an Intel Xeon Phi with KMP_THREAD_AFFINITY set to scatter, compact and balanced

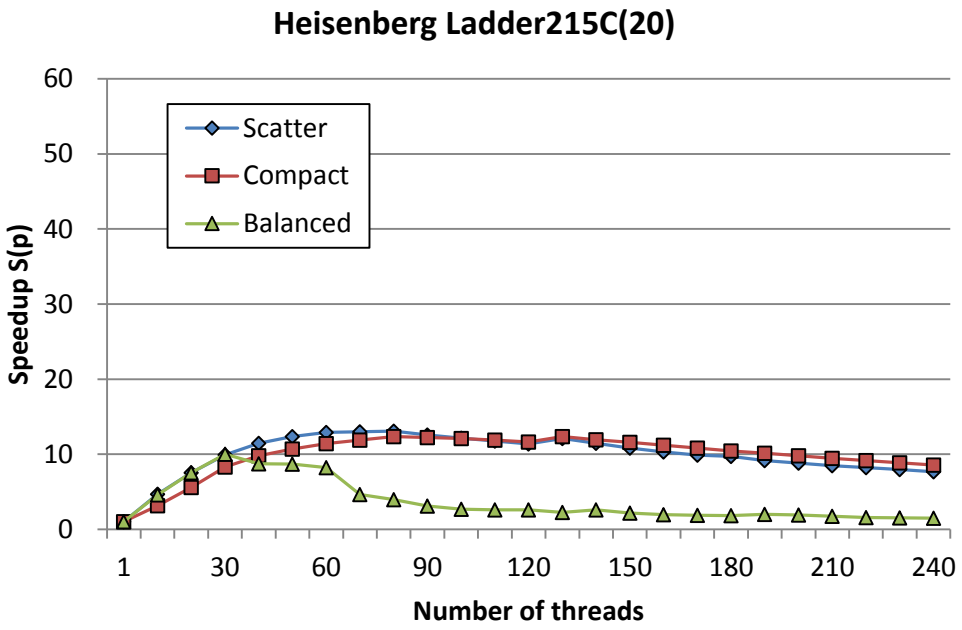


Figure 4.61 Parallel speedup of Ladder215C(20) model on an Intel Xeon Phi with KMP_THREAD_AFFINITY set to scatter, compact and balanced

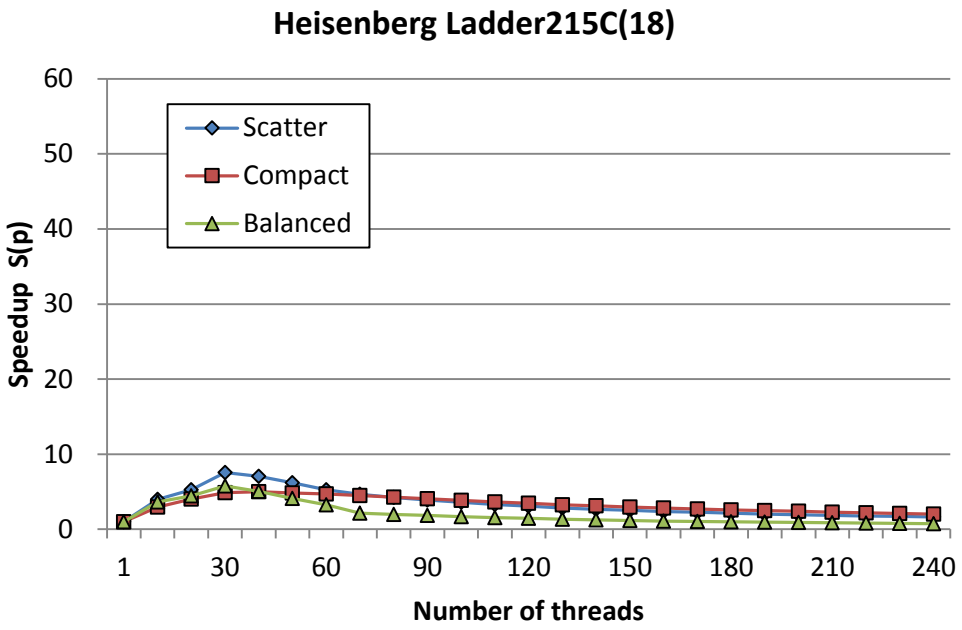


Figure 4.62 Parallel speedup of Ladder215C(18) model on an Intel Xeon Phi with KMP_THREAD_AFFINITY set to scatter, compact and balanced

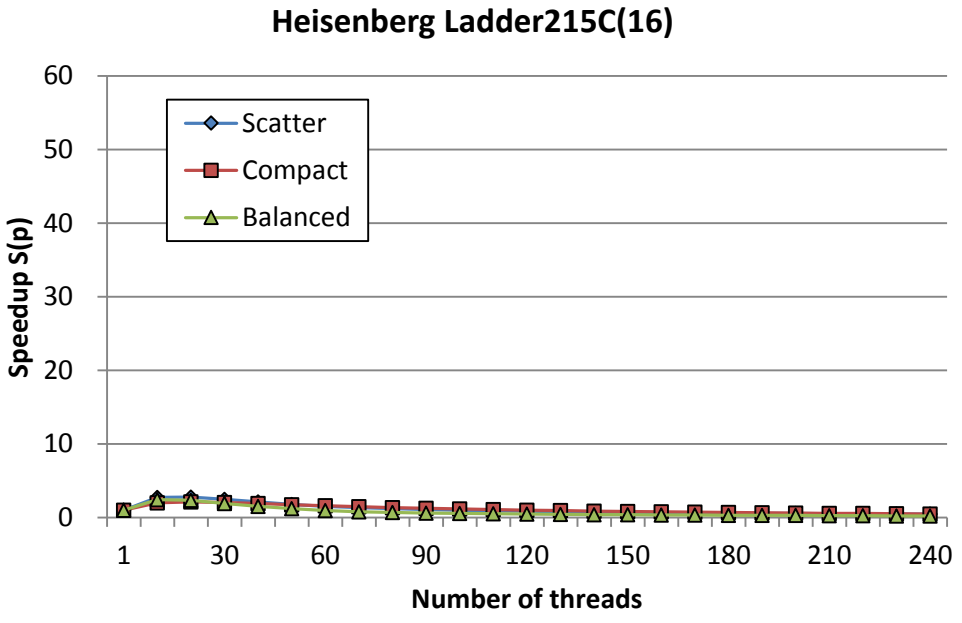


Figure 4.63 Parallel speedup of Ladder215C(16) model on an Intel Xeon Phi with KMP_THREAD_AFFINITY set to scatter, compact and balanced

See above the Figure 4.58, Figure 4.59, Figure 4.60, Figure 4.61, Figure 4.62 and Figure 4.63, review Table 4.42 and Table 4.43, we can find out the largest number of speedup of Ladder215C model is happened when N = 26, it is 58.37 times faster than the execution time using single thread. This set of data with N = 26 is the fastest one of all. The maximum speedups for N = 24 and 26 range from 50 to 60. I can also be seen that for the larger systems `compact` core binding gives better speedups.

The speedups of `scatter, compact` and `balanced` are very similar when N = 24 and 26, if N is less than 20, the coprocessor is inefficient, and these speedup numbers of Ladder215C are similar to Ladder2 model.

4.4.4 Heisenberg Ladder 215S

Table 4.44 Speedup of Ladder 215S(*N*) with *N* = 26, 24 and 22 on an Intel Xeon Phi

Thread	*N* = 26			*N* = 24			*N* = 22		
affinity type	Scatter	Compact	Balanced	Scatter	Compact	Balanced	Scatter	Compact	Balanced
1	–	–	–	1.00	1.00	1.00	1.00	1.00	1.00
10	–	–	–	4.13	3.38	4.29	4.23	3.26	4.31
20	–	–	–	7.46	6.28	7.23	6.88	5.90	7.22
30	–	–	–	10.16	9.22	9.79	10.04	8.45	9.93
40	–	–	–	13.06	12.00	12.87	13.12	10.92	13.25
50	–	–	–	15.80	14.77	15.34	15.98	13.02	16.02
60	–	–	–	18.08	17.47	18.17	18.65	15.28	18.32
70	–	–	–	20.48	19.89	20.55	20.51	17.50	19.99
80	–	–	–	23.32	22.56	23.28	23.20	20.12	23.48
90	–	–	–	24.93	24.79	25.22	24.67	22.07	24.44
100	–	–	–	27.93	26.77	27.89	26.51	24.50	26.49
110	–	–	–	30.12	28.93	29.43	28.23	26.85	27.85
120	–	–	–	30.26	30.84	31.15	29.81	28.82	29.41
130	–	–	–	31.81	32.73	32.36	30.91	30.36	30.41
140	–	–	–	33.91	34.72	34.11	32.47	32.40	32.12
150	–	–	–	34.99	36.68	36.37	28.07	28.60	8.44
160	–	–	–	36.54	39.01	38.31	28.08	29.14	7.64
170	–	–	–	36.87	39.03	38.41	27.32	28.83	6.74
180	–	–	–	38.44	42.06	40.55	26.91	28.94	5.90
190	–	–	–	39.34	44.15	41.13	26.38	29.15	5.94
200	–	–	–	40.47	45.38	43.25	25.65	28.65	4.77
210	–	–	–	43.17	46.71	45.33	25.13	28.32	4.39
220	–	–	–	44.03	47.56	45.62	24.74	27.61	4.66
230	–	–	–	46.54	49.37	49.48	24.22	27.05	4.17
240	–	–	–	47.44	50.91	50.94	23.58	26.50	3.82

Table 4.45 Speedup of Ladder 215S(*N*) with *N* = 20, 18 and 16 on an Intel Xeon Phi

Thread	*N* = 20			*N* = 18			*N* = 16		
affinity type	Scatter	Compact	Balanced	Scatter	Compact	Balanced	Scatter	Compact	Balanced
1	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
10	4.70	3.13	4.18	4.09	2.93	3.66	2.66	1.95	2.42
20	7.57	5.58	7.35	5.40	3.98	4.49	2.67	2.10	2.35
30	10.02	8.30	10.17	7.91	4.82	5.94	2.47	2.01	1.94
40	11.47	9.77	8.83	7.71	4.90	5.25	2.23	1.86	1.54
50	12.56	10.68	8.80	7.19	4.72	4.45	1.95	1.70	1.20
60	13.22	11.34	8.35	6.50	4.55	3.65	1.69	1.56	0.94
70	13.52	11.74	4.65	5.83	4.30	2.32	1.51	1.43	0.75
80	13.90	12.08	3.98	5.39	4.04	2.15	1.39	1.33	0.67
90	13.53	11.84	3.13	4.92	3.81	1.99	1.25	1.21	0.61
100	13.39	11.64	2.70	4.53	3.61	1.85	1.16	1.12	0.54
110	13.12	11.40	2.64	4.31	3.37	1.70	1.03	1.04	0.49
120	12.74	11.12	2.60	3.99	3.15	1.60	0.96	0.97	0.45
130	13.71	11.58	2.30	3.69	2.95	1.46	0.90	0.90	0.41
140	13.09	11.07	2.66	3.48	2.78	1.35	0.85	0.84	0.38
150	12.56	10.55	2.24	3.31	2.60	1.27	0.80	0.78	0.36
160	12.01	10.01	1.99	3.15	2.45	1.17	0.76	0.73	0.33
170	11.58	9.51	1.94	3.01	2.28	1.11	0.73	0.69	0.31
180	11.24	8.99	1.89	2.95	2.14	1.06	0.69	0.65	0.29
190	10.62	8.53	2.04	2.75	2.00	1.05	0.65	0.61	0.27
200	10.32	8.04	2.01	2.63	1.88	1.01	0.63	0.57	0.26
210	9.94	7.54	1.79	2.54	1.75	0.96	0.60	0.53	0.25
220	9.65	7.14	1.64	2.45	1.65	0.90	0.58	0.50	0.23
230	9.36	6.70	1.60	2.36	1.54	0.85	0.56	0.46	0.22
240	9.01	6.40	1.56	2.27	1.47	0.81	0.54	0.44	0.21

The Table 4.44 and Table 4.45 summarize 18 tables: from Table 6.64 to Table 6.81 in section 6.2

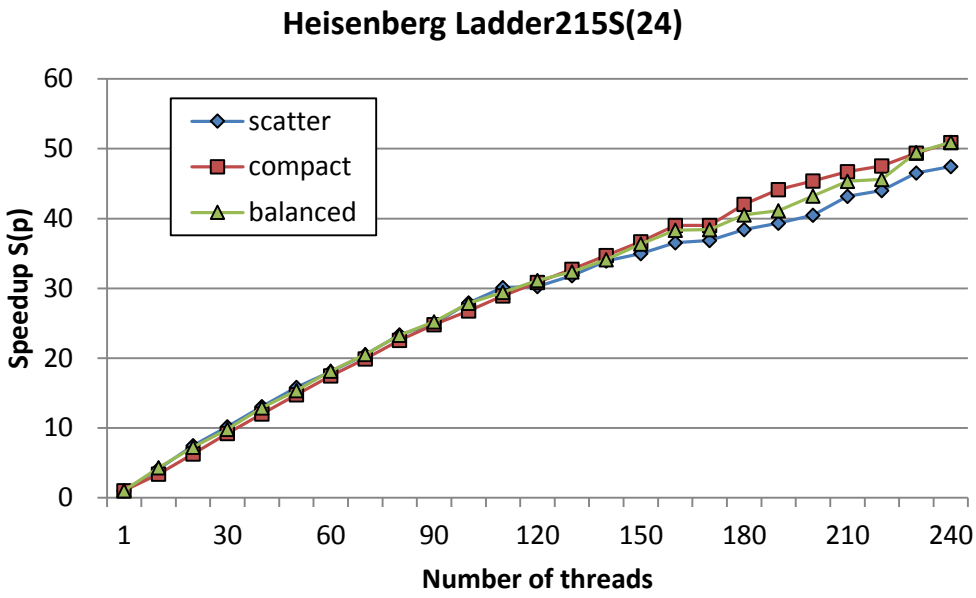


Figure 4.64 Parallel speedup of Ladder215S(24) model on an Intel Xeon Phi with KMP_THREAD_AFFINITY set to scatter, compact and balanced

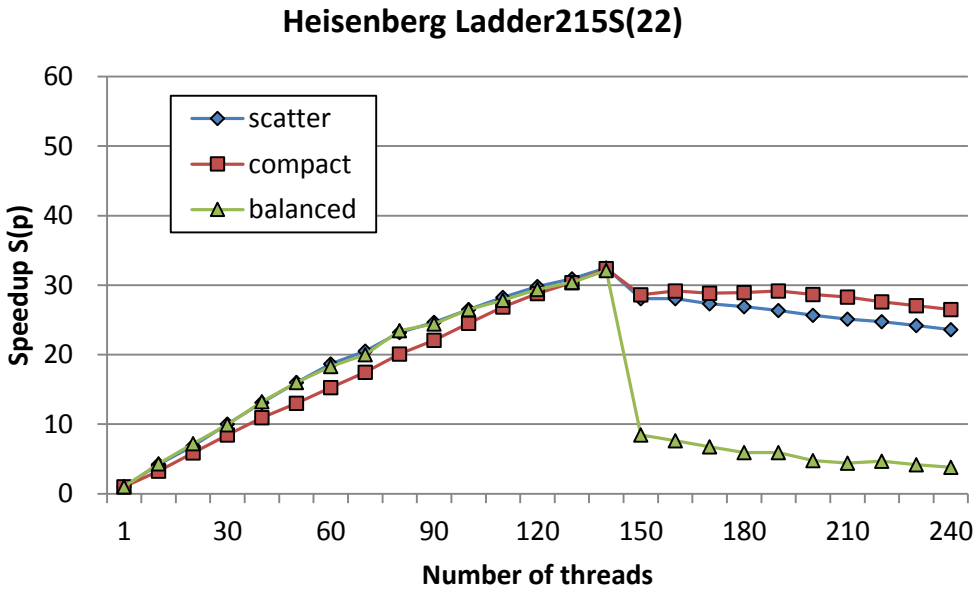


Figure 4.65 Parallel speedup of Ladder215S(22) model on an Intel Xeon Phi with KMP_THREAD_AFFINITY set to scatter, compact and balanced

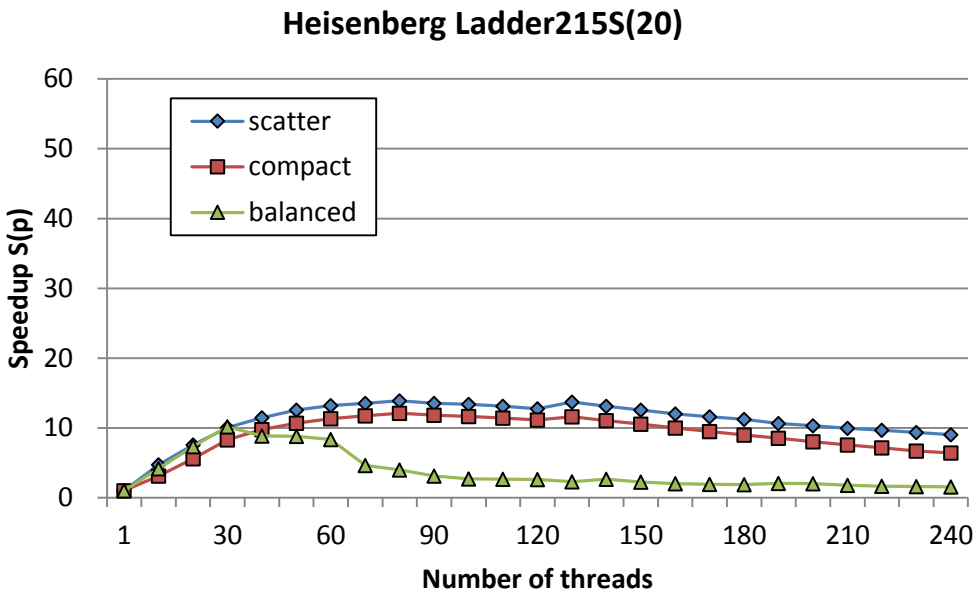


Figure 4.66 Parallel speedup of Ladder215S(20) model on an Intel Xeon Phi with KMP_THREAD_AFFINITY set to scatter, compact and balanced

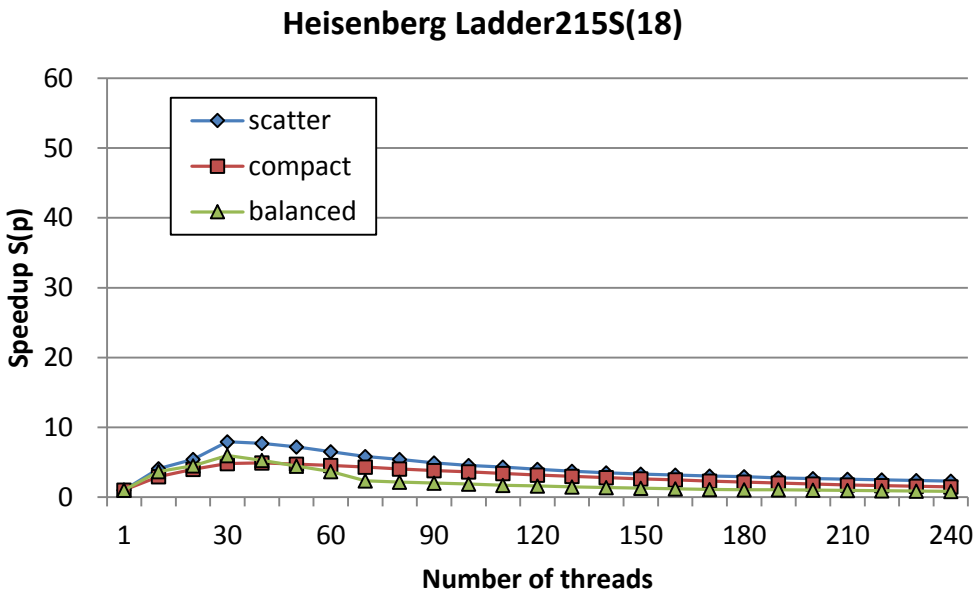


Figure 4.67 Parallel speedup of Ladder215S(18) model on an Intel Xeon Phi with KMP_THREAD_AFFINITY set to scatter, compact and balanced

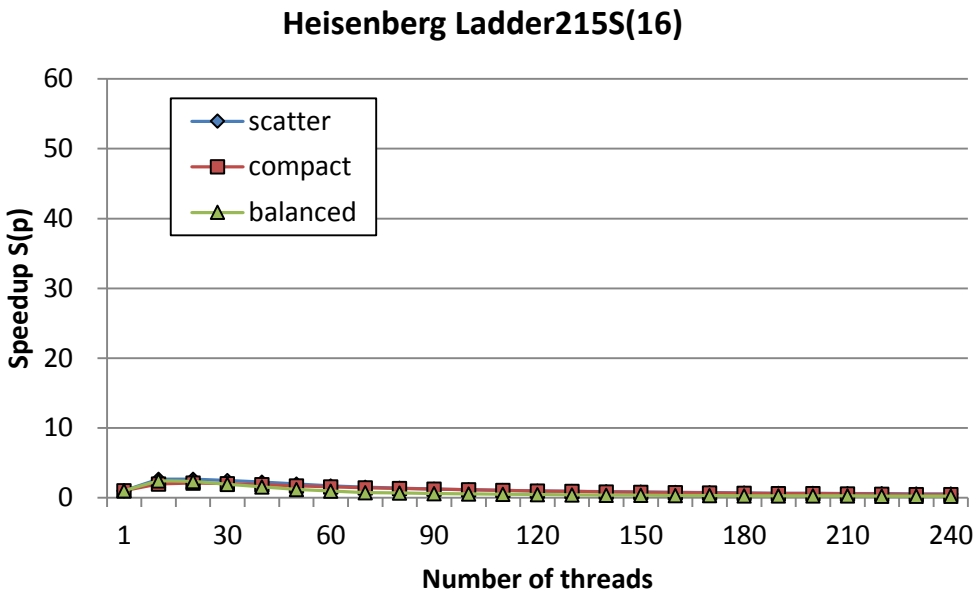


Figure 4.68 Parallel speedup of Ladder215S(16) model on an Intel Xeon Phi with KMP_THREAD_AFFINITY set to scatter, compact and balanced

The speedup of Ladder215S model is almost same as the one of Ladder215C model, I do not have data about speedup of the Ladder215S with $N = 26$ because of the Xeon Phi machine had some problem, but I can predict the speedup of Ladder215S model has almost same as Ladder215C, because the speedup of Ladder215S with size $N = 24$, 22, 20, 18 and 16 are rather similar with speedup of Ladder215C models. Actually, we can observe all two-leg Heisenberg Ladder models (Ladder2, Ladder215C and Ladder215S) has almost same speedup.

4.4.5 Heisenberg Ladder 315C

Table 4.46 Speedup of Ladder 315C(*N*) with *N* = 24 and 18 on an Intel Xeon Phi

Thread	*N* = 24			*N* = 18		
affinity type	Scatter	Compact	Balanced	Scatter	Compact	Balanced
1	1.00	1.00	1.00	1.00	1.00	1.00
10	4.12	3.27	4.13	4.27	3.02	3.72
20	7.18	6.37	7.16	5.67	4.19	4.73
30	10.08	9.32	9.76	6.01	4.59	4.66
40	12.58	12.14	12.52	7.58	5.18	5.17
50	15.77	15.09	15.52	6.68	5.10	4.22
60	17.88	17.52	18.03	5.71	4.86	3.36
70	20.09	20.05	20.49	5.06	4.62	2.17
80	22.69	22.65	23.15	4.57	4.41	2.01
90	24.63	25.03	25.68	4.20	4.17	1.86
100	27.42	27.38	27.85	3.89	3.91	1.73
110	29.17	29.35	29.98	3.62	3.63	1.58
120	30.41	31.47	31.59	3.34	3.44	1.49
130	31.45	33.43	32.72	3.08	3.22	1.37
140	33.67	35.28	34.68	2.89	3.02	1.30
150	35.43	37.22	36.50	2.74	2.83	1.20
160	36.14	39.48	37.97	2.60	2.66	1.11
170	37.15	39.98	38.71	2.47	2.48	1.06
180	38.58	43.18	41.29	2.36	2.33	1.01
190	39.52	44.87	41.64	2.22	2.18	1.00
200	41.79	46.54	43.70	2.12	2.05	0.95
210	43.69	48.02	46.01	2.04	1.91	0.91
220	44.69	48.85	46.51	1.97	1.84	0.84
230	47.87	50.89	50.77	1.94	1.69	0.80
240	48.36	52.40	52.31	1.81	1.60	0.75

The Table 4.46 summarize 18 tables: from Table 6.82 to Table 6.87 in section 6.2

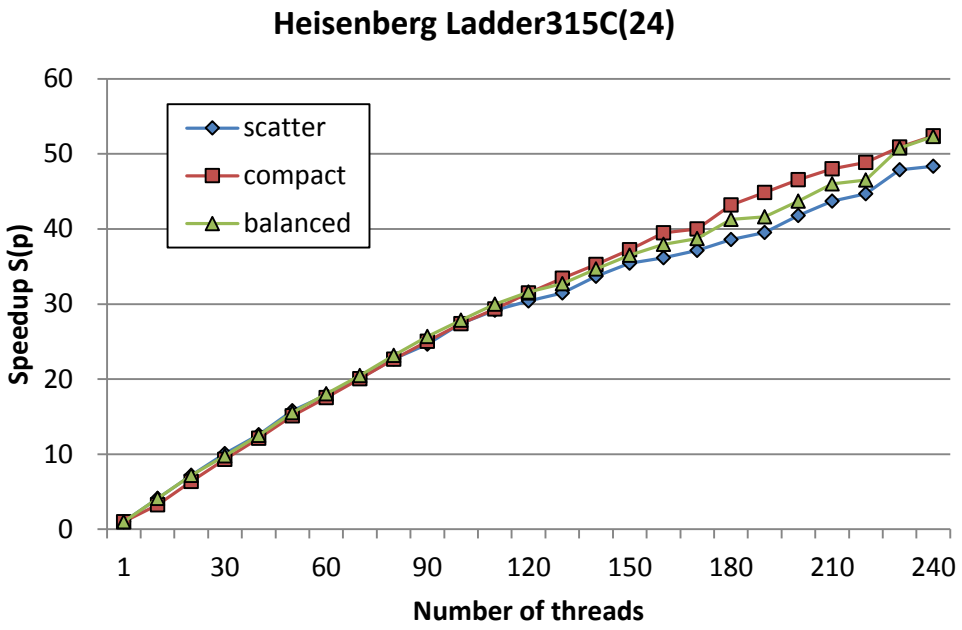


Figure 4.69 Parallel speedup of Ladder315C(24) model on an Intel Xeon Phi with KMP_THREAD_AFFINITY set to scatter, compact and balanced

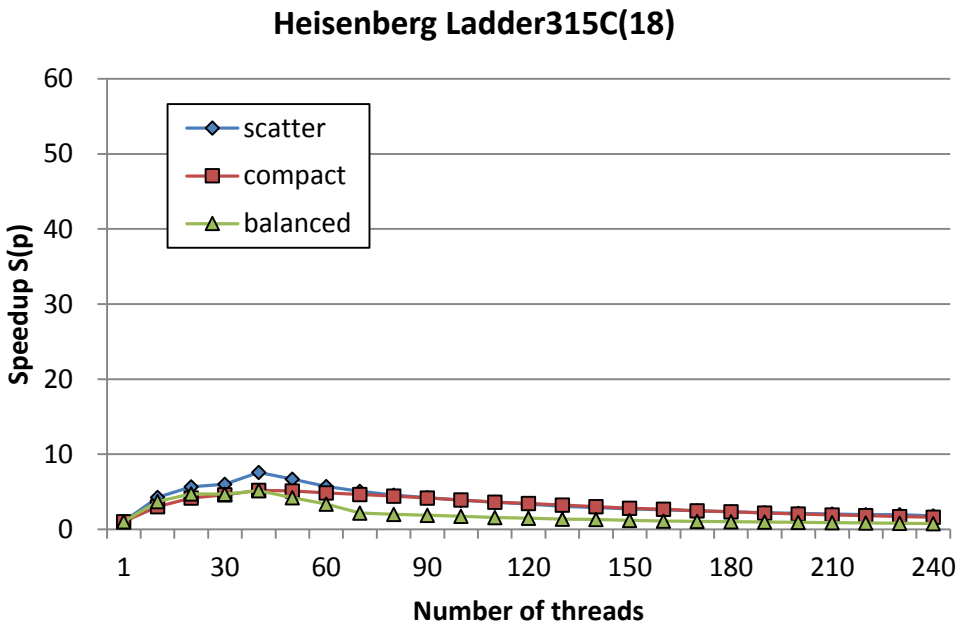


Figure 4.70 Parallel speedup of Ladder315C(18) model on an Intel Xeon Phi with KMP_THREAD_AFFINITY set to scatter, compact and balanced

4.4.6 Heisenberg Ladder 315S

Table 4.47 Speedup of Ladder 315S(*N*) with *N* = 24 and 18 on an Intel Xeon Phi

Thread	*N* = 24			*N* = 18		
affinity type	Scatter	Compact	Balanced	Scatter	Compact	Balanced
1	1.00	1.00	1.00	1.00	1.00	1.00
10	4.15	3.32	4.15	4.24	3.02	3.98
20	7.05	6.37	7.24	5.58	4.29	5.12
30	9.78	9.38	10.02	6.15	4.62	4.95
40	12.40	12.13	12.41	8.35	5.28	5.51
50	15.75	15.09	15.84	8.12	5.14	4.50
60	17.83	17.51	18.07	7.45	5.05	3.60
70	19.93	20.06	20.06	6.55	4.71	2.31
80	22.18	22.75	23.31	6.27	4.45	2.15
90	24.97	25.02	25.31	5.56	4.32	1.95
100	27.22	27.39	27.86	5.36	4.00	1.81
110	29.36	29.36	29.52	5.01	3.81	1.66
120	30.41	31.46	31.49	4.58	3.66	1.56
130	31.23	33.42	32.76	4.15	3.50	1.45
140	33.40	35.27	34.60	4.01	3.23	1.33
150	35.09	37.24	36.86	3.73	3.05	1.23
160	34.94	39.48	38.84	3.59	2.86	1.15
170	37.25	39.79	38.79	3.40	2.69	1.09
180	38.86	43.18	41.19	3.30	2.61	1.03
190	40.48	44.91	41.82	3.05	2.56	1.01
200	41.59	46.54	44.01	2.94	2.41	0.97
210	43.76	48.39	46.18	2.97	2.17	0.91
220	45.02	48.82	46.91	2.74	2.11	0.85
230	47.67	50.90	50.87	2.69	2.00	0.84
240	48.34	52.71	52.31	2.54	2.03	0.77

The Table 4.47 summarize 18 tables: from Table 6.88 to Table 6.93 in section 6.2

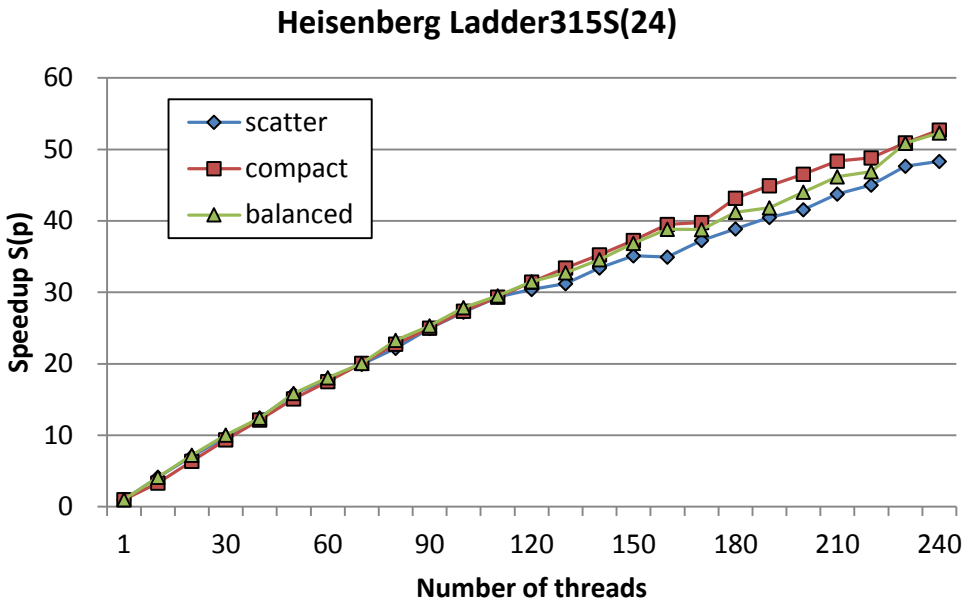


Figure 4.71 Parallel speedup of Ladder315S(24) model on an Intel Xeon Phi with KMP_THREAD_AFFINITY set to scatter, compact and balanced

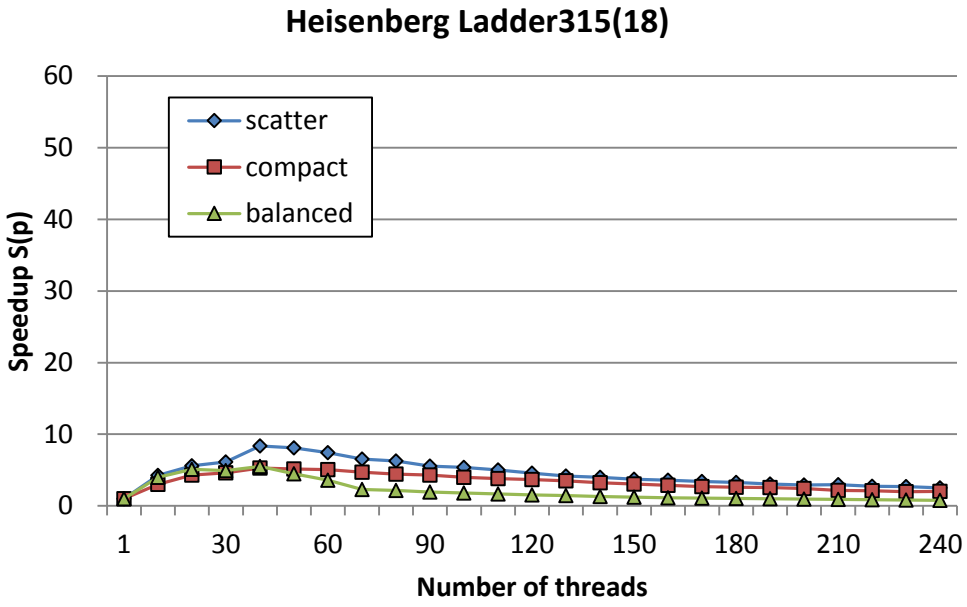


Figure 4.72 Parallel speedup of Ladder315S(24) model on an Intel Xeon Phi with KMP_THREAD_AFFINITY set to scatter, compact and balanced

The dimension of three-leg Heisenberg Ladder models require that size is divided exactly by 6, I tested speedup of three-leg Ladder models with N = 24 and 18. In Table 4.46, Figure 4.69 and Figure 4.70 show the speedup of the Ladder315C and Ladder315S are very similar, or even almost same, as before the speedups rise linearly when N = 24, but speedup of three-leg Ladder models is a bit better than the ones of two-leg Ladder models. In Table 4.47, Figure 4.71 and Figure 4.72 the speedup with N = 18 in three-leg Ladder models is low as before.

4.4.7 Time and speedup of different models on Xeon Phi

In Figure 4.73, Figure 4.74 and Figure 4.75, I show the parallel speedup obtained with the variants RMM-DIIS (RMM-DIIS V_{max} and V_{max}) method on the Intel Xeon Phi coprocessor for the dimerized two-leg Heisenberg ladder215C model for various system size *N*, I tested three settings of the environment variable KMP_THREAD_ AFFINITY `scatter`, `compact` and `balanced` . This variable affects the binding of threads to processor cores for OpenMP and the MKL on the coprocessor. If set to `scatter` the runtime system distributed the threads over as many cores as possible whereas `compact` uses as few cores as possible, `balanced` uses cores as groups. The speedups of `compact` and `balanced` are very similar when *N* = 24 and 26, if *N* is less than 24, the speedups of balanced are much lower than speedups of `scatter` and `compact` specially in the system with small *N*, almost all speedups of `balanced` are the lowest one among three types. The figures show that for small system with *N* = 16, 18 and 20, the coprocessor is inefficient with maximum speedups far below what be expected form a 60 core device, even some time speedup is less than 1. For *N* = 22 a maximum speedup of 30 is reached at 130 threads, the speedups declined, especially in `balanced` type, the speedup from 30 drop to 8 (see Figure 4.60 in page 103), whereas for *N* =24 and 26 the speedup rises continuously until *p* = 240 threads. The maximum speedups for *N* = 24 and 26 range from 50 to 60. I can also be seen that for the larger systems `compact` core binding gives better speedups.

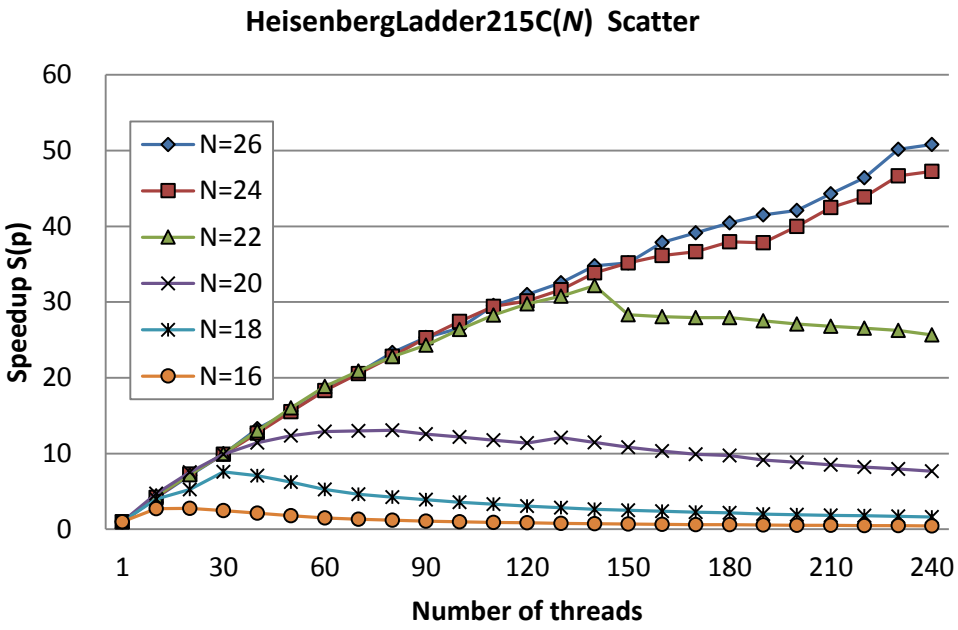


Figure 4.73 Parallel speedup of Ladder215C(*N*) with *N* = 16, 18, 20, 22, 24 and 26 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY is `Scatter`

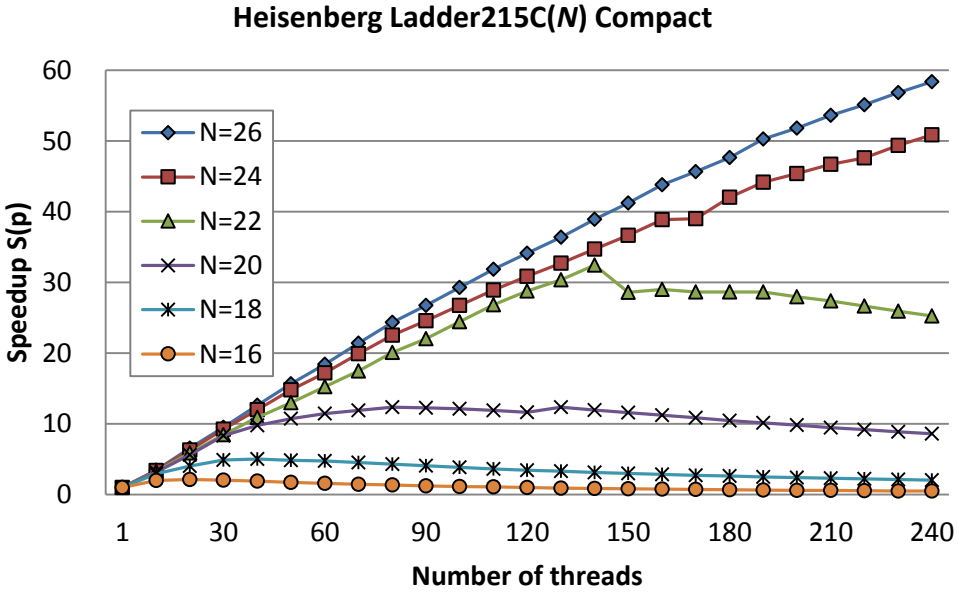


Figure 4.74 Parallel speedup of Ladder215C(*N*) with *N* = 16, 18, 20, 22, 24 and 26 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY is `compact`

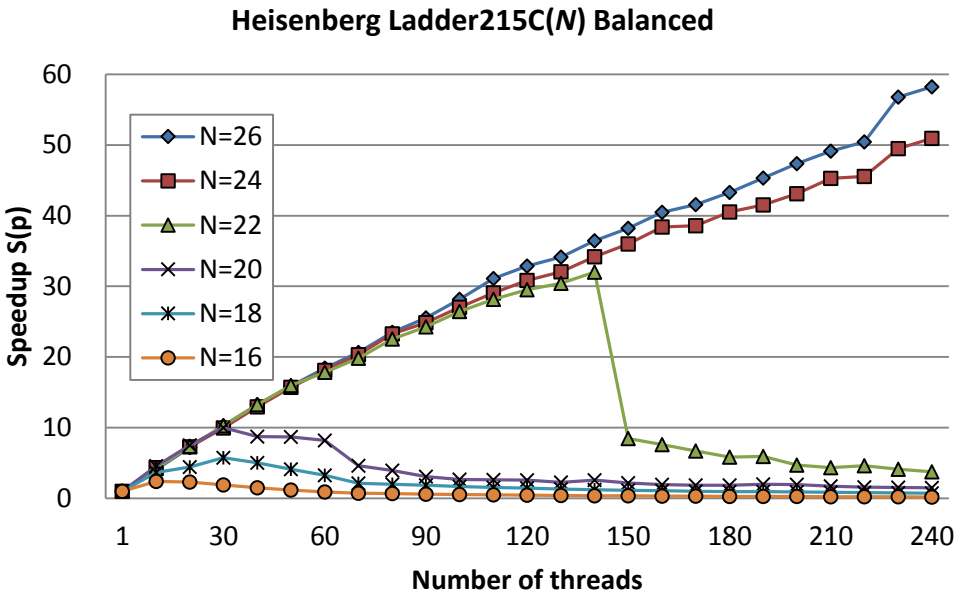


Figure 4.75 Parallel speedup of Ladder215C(*N*) with *N* = 16, 18, 20, 22, 24 and 26 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY is `Balanced`

In Table 4.48, Table 4.49, Table 4.50, Table 4.51, Table 4.52and Table 4.53 we compare the parallel speedups obtained for all our benchmark models with *N* = 24 and 18 on the Xeon Phi coprocessor for different number of threads. As in Figure 4.73, Figure 4.74 and Figure 4.75, I have considered three types of thread binding. For thread numbers *p* up to sixty, the speedups are similar for all models and correspond to a parallel efficiency of about 33%. As the number of threads *p* goes beyond 60 (i.e. more than one thread per core), the speedup continues to increase but differences between the models appear. This is best seen at *p* = 240. The highest speedup is obtained by the three-leg dimerized ladder systems. All two-leg ladder modes have slightly lower speedup follow by the chain system. The data show that the speedup is determined by the number of legs of the model, all two leg ladders have (nearly) the same speedups independent of parameters. The same is true for the three-leg models. This indicates that the speedup is determined

by the sparsity of matrix. Execution time of Chain system is the fastest one of all, although its speedup is the lowest of all. Three-leg dimerized ladder systems has the best speedup, however the execution time of it is more than two-leg dimerized ladder system and Chain system. The reason for why three-leg dimerized ladder systems has good speedup but slower is there are more non-zero elements in three-leg dimerized ladder system, hence it need more time to calculate. The number of non-zero entries per row is 13, 19, and 21 for the chain, two-leg and three-leg models, respectively.

Table 4.48 Comparison of time and speedup of Chain, Ladder and Ladder 215C with *N* = 24 on an Intel Xeon Phi with KMP_THREAD_AFFINITY set to `scatter`

`Scatter` *N* = 24 (time in seconds)						
model	Chain(*N*)		Ladder(*N*)		Ladder 215C(*N*)	
threads	time	speedup	time	speedup	time	speedup
1	1797.89	1.00	2354.22	1.00	2354.32	1.00
10	414.00	4.34	553.43	4.25	563.25	4.18
20	248.57	7.23	316.80	7.43	322.80	7.29
30	180.10	9.98	241.08	9.77	237.32	9.92
40	135.48	13.27	179.88	13.09	185.61	12.68
50	111.78	16.08	151.88	15.50	151.79	15.51
60	95.14	18.90	130.34	18.06	128.63	18.30
70	86.75	20.72	116.26	20.25	114.44	20.57
80	75.47	23.82	102.89	22.88	103.02	22.85
90	70.51	25.50	93.56	25.16	93.12	25.28
100	65.64	27.39	88.78	26.52	85.77	27.45
110	62.73	28.66	82.68	28.47	79.96	29.44
120	60.01	29.96	77.99	30.19	78.13	30.14
130	59.45	30.24	74.04	31.80	74.45	31.62
140	57.45	31.29	69.64	33.81	69.50	33.88
150	54.14	33.21	67.26	35.00	66.90	35.19
160	53.56	33.57	65.66	35.86	65.15	36.14
170	52.98	33.94	65.79	35.78	64.22	36.66
180	49.96	35.99	62.07	37.93	62.05	37.94
190	48.74	36.89	61.78	38.11	62.22	37.84
200	47.73	37.67	58.54	40.22	58.89	39.98
210	45.69	39.35	55.81	42.18	55.40	42.49
220	44.26	40.63	55.13	42.70	53.65	43.88
230	41.75	43.06	50.61	46.51	50.45	46.66
240	41.46	43.36	49.78	47.29	49.80	47.27

Table 4.49 Comparison of time and speedup of Ladder215S, Ladder315C and Ladder315S with N = 24 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `scatter`

`Scatter` N = 24 (time in seconds)						
model	Ladder 215S(N)		Ladder 315C(N)		Ladder 315S(N)	
threads	time	speedup	time	speedup	time	speedup
1	2355.34	1.00	2568.94	1.00	2568.89	1.00
10	570.38	4.13	623.26	4.12	619.72	4.15
20	315.66	7.46	357.64	7.18	364.62	7.05
30	231.80	10.16	254.79	10.08	262.78	9.78
40	180.33	13.06	204.19	12.58	207.23	12.40
50	149.11	15.80	162.92	15.77	163.13	15.75
60	130.28	18.08	143.68	17.88	144.07	17.83
70	115.02	20.48	127.89	20.09	128.89	19.93
80	101.02	23.32	113.24	22.69	115.80	22.18
90	94.48	24.93	104.28	24.63	102.88	24.97
100	84.32	27.93	93.69	27.42	94.36	27.22
110	78.20	30.12	88.07	29.17	87.50	29.36
120	77.85	30.26	84.48	30.41	84.49	30.41
130	74.05	31.81	81.67	31.45	82.26	31.23
140	69.46	33.91	76.29	33.67	76.90	33.40
150	67.32	34.99	72.51	35.43	73.20	35.09
160	64.46	36.54	71.09	36.14	73.52	34.94
170	63.88	36.87	69.15	37.15	68.97	37.25
180	61.28	38.44	66.59	38.58	66.11	38.86
190	59.87	39.34	65.00	39.52	63.46	40.48
200	58.20	40.47	61.48	41.79	61.77	41.59
210	54.57	43.17	58.80	43.69	58.70	43.76
220	53.50	44.03	57.48	44.69	57.06	45.02
230	50.60	46.54	53.66	47.87	53.89	47.67
240	49.65	47.44	53.12	48.36	53.14	48.34

Table 4.50 Comparison of time and speedup of Chain, Ladder and Ladder 215C with *N* = 24 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `compact`

`Compact` *N* = 24 (time in seconds)						
model	Chain(*N*)		Ladder(*N*)		Ladder 215C(*N*)	
threads	time	speedup	time	speedup	time	speedup
1	1797.89	1.00	2354.22	1.00	2354.32	1.00
10	577.55	3.11	698.39	3.37	696.48	3.38
20	302.90	5.94	373.96	6.30	374.54	6.29
30	208.23	8.63	254.70	9.24	255.51	9.21
40	159.40	11.28	196.21	12.00	196.13	12.00
50	129.06	13.93	159.40	14.77	159.40	14.77
60	111.40	16.14	136.87	17.20	136.85	17.20
70	97.12	18.51	118.39	19.88	118.40	19.88
80	85.11	21.12	104.39	22.55	104.41	22.55
90	78.52	22.90	95.73	24.59	95.78	24.58
100	71.86	25.02	87.95	26.77	87.98	26.76
110	66.86	26.89	81.45	28.91	81.43	28.91
120	62.91	28.58	76.40	30.82	76.31	30.85
130	59.42	30.26	72.05	32.67	71.96	32.71
140	56.38	31.89	67.87	34.69	67.82	34.71
150	53.53	33.59	64.22	36.66	64.23	36.65
160	50.48	35.62	60.63	38.83	60.56	38.88
170	50.67	35.48	60.35	39.01	60.38	38.99
180	46.61	38.57	56.04	42.01	55.98	42.06
190	44.45	40.44	53.38	44.10	53.31	44.17
200	43.29	41.53	51.90	45.36	51.85	45.40
210	42.13	42.67	50.44	46.68	50.40	46.72
220	41.63	43.18	49.56	47.50	49.47	47.59
230	40.05	44.89	47.69	49.37	47.67	49.39
240	39.14	45.93	46.29	50.86	46.26	50.89

Table 4.51 Comparison of time and speedup of Ladder215S, Ladder315C and Ladder315S with N = 24 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `compact`

`Compact` N = 24 (time in seconds)						
model	Ladder 215S(N)		Ladder 315C(N)		Ladder 315S(N)	
threads	time	speedup	time	speedup	time	speedup
1	2355.34	1.00	2568.94	1.00	2568.89	1.00
10	696.70	3.38	785.06	3.27	774.36	3.32
20	374.76	6.28	403.20	6.37	403.12	6.37
30	255.56	9.22	275.59	9.32	273.91	9.38
40	196.31	12.00	211.64	12.14	211.72	12.13
50	159.52	14.77	170.30	15.09	170.23	15.09
60	134.78	17.47	146.65	17.52	146.67	17.51
70	118.39	19.89	128.09	20.05	128.06	20.06
80	104.40	22.56	113.39	22.65	112.93	22.75
90	95.01	24.79	102.64	25.03	102.69	25.02
100	87.97	26.77	93.81	27.38	93.79	27.39
110	81.43	28.93	87.52	29.35	87.51	29.36
120	76.37	30.84	81.64	31.47	81.66	31.46
130	71.97	32.73	76.84	33.43	76.87	33.42
140	67.84	34.72	72.82	35.28	72.83	35.27
150	64.21	36.68	69.01	37.22	68.97	37.24
160	60.38	39.01	65.07	39.48	65.06	39.48
170	60.35	39.03	64.26	39.98	64.55	39.79
180	56.01	42.06	59.49	43.18	59.49	43.18
190	53.35	44.15	57.25	44.87	57.20	44.91
200	51.90	45.38	55.20	46.54	55.20	46.54
210	50.43	46.71	53.49	48.02	53.09	48.39
220	49.53	47.56	52.59	48.85	52.62	48.82
230	47.71	49.37	50.48	50.89	50.47	50.90
240	46.26	50.91	49.02	52.40	48.74	52.71

Table 4.52 Comparison of time and speedup of Chain, Ladder and Ladder 215C with *N* = 24 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `balanced`

`Balanced` *N* = 24 (time in seconds)						
model	Chain(*N*)		Ladder(*N*)		Ladder 215C(*N*)	
threads	time	speedup	time	speedup	time	speedup
1	1797.89	1.00	2354.22	1.00	2354.32	1.00
10	438.57	4.10	554.35	4.25	541.91	4.34
20	262.23	6.86	331.76	7.10	322.58	7.30
30	187.99	9.56	226.18	10.41	236.08	9.97
40	145.30	12.37	185.39	12.70	181.50	12.97
50	115.59	15.55	153.42	15.34	149.90	15.71
60	99.26	18.11	130.32	18.07	129.96	18.12
70	89.24	20.15	117.29	20.07	115.86	20.32
80	79.38	22.65	102.97	22.86	101.16	23.27
90	73.86	24.34	93.70	25.12	94.63	24.88
100	69.89	25.73	85.32	27.59	87.06	27.04
110	65.23	27.56	80.18	29.36	81.03	29.05
120	61.47	29.25	75.29	31.27	76.30	30.85
130	59.37	30.28	72.67	32.39	73.43	32.06
140	56.58	31.78	68.08	34.58	68.85	34.19
150	53.88	33.37	66.43	35.44	65.37	36.01
160	51.24	35.09	62.35	37.76	61.32	38.39
170	50.82	35.38	61.71	38.15	61.01	38.59
180	47.74	37.66	57.76	40.76	58.07	40.54
190	46.54	38.63	57.16	41.19	56.72	41.51
200	44.55	40.36	54.76	42.99	54.59	43.13
210	43.09	41.72	52.13	45.16	52.01	45.27
220	42.91	41.90	51.74	45.51	51.69	45.55
230	39.99	44.96	47.60	49.46	47.55	49.51
240	39.10	45.99	46.29	50.86	46.24	50.92

Table 4.53 Comparison of time and speedup of Ladder215S, Ladder315C and Ladder315S with N = 24 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `balanced`

`Balanced` N = 24 (time in seconds)						
model	Ladder 215S(N)		Ladder 315C(N)		Ladder 315S(N)	
threads	time	speedup	time	speedup	time	speedup
1	2355.34	1.00	2568.94	1.00	2568.89	1.00
10	549.57	4.29	622.22	4.13	619.44	4.15
20	325.76	7.23	358.71	7.16	354.97	7.24
30	240.50	9.79	263.17	9.76	256.50	10.02
40	183.06	12.87	205.25	12.52	206.93	12.41
50	153.58	15.34	165.51	15.52	162.17	15.84
60	129.63	18.17	142.44	18.03	142.15	18.07
70	114.59	20.55	125.40	20.49	128.03	20.06
80	101.17	23.28	110.97	23.15	110.19	23.31
90	93.39	25.22	100.03	25.68	101.48	25.31
100	84.45	27.89	92.25	27.85	92.21	27.86
110	80.03	29.43	85.70	29.98	87.02	29.52
120	75.62	31.15	81.33	31.59	81.57	31.49
130	72.78	32.36	78.51	32.72	78.42	32.76
140	69.06	34.11	74.08	34.68	74.25	34.60
150	64.76	36.37	70.38	36.50	69.69	36.86
160	61.47	38.31	67.66	37.97	66.15	38.84
170	61.33	38.41	66.37	38.71	66.22	38.79
180	58.09	40.55	62.21	41.29	62.36	41.19
190	57.26	41.13	61.70	41.64	61.43	41.82
200	54.46	43.25	58.78	43.70	58.37	44.01
210	51.96	45.33	55.83	46.01	55.63	46.18
220	51.63	45.62	55.23	46.51	54.76	46.91
230	47.60	49.48	50.60	50.77	50.50	50.87
240	46.24	50.94	49.11	52.31	49.11	52.31

Table 4.54 Comparison of time and speedup of Chain, Ladder and Ladder 215C with *N* = 18 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `scatter`

`Scatter` *N* = 18 (time in seconds)						
model	Chain(*N*)		Ladder(*N*)		Ladder 215C(*N*)	
threads	time	speedup	time	speedup	time	speedup
1	27.67	1.00	32.94	1.00	33.01	1.00
10	7.15	3.87	8.21	4.01	8.29	3.98
20	5.56	4.97	6.31	5.22	6.30	5.24
30	3.95	7.00	4.42	7.45	4.36	7.57
40	4.11	6.73	4.79	6.88	4.68	7.06
50	4.48	6.18	5.47	6.03	5.32	6.21
60	5.02	5.51	6.44	5.11	6.28	5.26
70	5.66	4.89	7.23	4.56	7.11	4.64
80	6.19	4.47	7.88	4.18	7.79	4.24
90	6.67	4.15	8.59	3.84	8.49	3.89
100	7.13	3.88	9.33	3.53	9.21	3.58
110	7.66	3.61	10.07	3.27	9.95	3.32
120	8.35	3.32	10.87	3.03	10.73	3.08
130	9.09	3.05	11.87	2.78	11.64	2.84
140	9.59	2.88	12.64	2.61	12.44	2.65
150	10.14	2.73	13.34	2.47	13.15	2.51
160	10.72	2.58	14.10	2.34	13.90	2.37
170	11.26	2.46	14.86	2.22	14.63	2.26
180	11.80	2.34	15.58	2.11	15.33	2.15
190	12.60	2.20	16.55	1.99	16.33	2.02
200	13.11	2.11	17.30	1.90	17.09	1.93
210	13.66	2.03	18.03	1.83	17.81	1.85
220	14.22	1.95	18.78	1.75	18.54	1.78
230	14.71	1.88	19.52	1.69	19.29	1.71
240	15.29	1.81	20.35	1.62	20.10	1.64

Table 4.55 Comparison of time and speedup of Ladder215S, Ladder315C and Ladder315S with N = 18 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `scatter`

`Scatter` N = 18 (time in seconds)						
model	Ladder 215S(N)		Ladder 315C(N)		Ladder 315S(N)	
threads	time	speedup	time	speedup	time	speedup
1	32.93	1.00	35.73	1.00	35.78	1.00
10	8.06	4.09	8.38	4.27	8.43	4.24
20	6.10	5.40	6.30	5.67	6.41	5.58
30	4.16	7.91	5.95	6.01	5.82	6.15
40	4.27	7.71	4.72	7.58	4.28	8.35
50	4.58	7.19	5.35	6.68	4.40	8.12
60	5.07	6.50	6.26	5.71	4.80	7.45
70	5.65	5.83	7.06	5.06	5.46	6.55
80	6.12	5.39	7.82	4.57	5.71	6.27
90	6.69	4.92	8.50	4.20	6.44	5.56
100	7.27	4.53	9.18	3.89	6.68	5.36
110	7.64	4.31	9.88	3.62	7.14	5.01
120	8.26	3.99	10.68	3.34	7.82	4.58
130	8.92	3.69	11.59	3.08	8.61	4.15
140	9.47	3.48	12.36	2.89	8.92	4.01
150	9.94	3.31	13.06	2.74	9.60	3.73
160	10.47	3.15	13.76	2.60	9.95	3.59
170	10.94	3.01	14.48	2.47	10.51	3.40
180	11.17	2.95	15.14	2.36	10.84	3.30
190	11.99	2.75	16.13	2.22	11.73	3.05
200	12.50	2.63	16.84	2.12	12.17	2.94
210	12.97	2.54	17.53	2.04	12.04	2.97
220	13.46	2.45	18.12	1.97	13.07	2.74
230	13.95	2.36	18.37	1.94	13.31	2.69
240	14.53	2.27	19.76	1.81	14.09	2.54

Table 4.56 Comparison of time and speedup of Chain, Ladder and Ladder 215C with *N* = 18 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `compact`

`Compact` *N* = 18 (time in seconds)						
model	Chain(*N*)		Ladder(*N*)		Ladder 215C(*N*)	
threads	time	speedup	time	speedup	time	speedup
1	26.60	1.00	32.94	1.00	33.01	1.00
10	10.02	2.66	11.27	2.92	11.20	2.95
20	7.64	3.48	8.29	3.98	8.21	4.02
30	6.28	4.23	6.84	4.82	6.78	4.87
40	6.33	4.20	6.72	4.90	6.59	5.01
50	6.53	4.07	6.96	4.73	6.85	4.82
60	6.80	3.91	7.18	4.59	6.98	4.73
70	7.13	3.73	7.63	4.32	7.33	4.50
80	7.55	3.52	8.03	4.10	7.73	4.27
90	8.00	3.33	8.61	3.83	8.14	4.06
100	8.34	3.19	9.12	3.61	8.60	3.84
110	8.77	3.03	9.76	3.38	9.10	3.63
120	9.37	2.84	10.43	3.16	9.57	3.45
130	9.88	2.69	11.11	2.97	10.09	3.27
140	10.35	2.57	11.76	2.80	10.61	3.11
150	10.70	2.49	12.54	2.63	11.11	2.97
160	11.34	2.35	13.30	2.48	11.64	2.84
170	11.86	2.24	14.25	2.31	12.17	2.71
180	12.39	2.15	15.20	2.17	12.74	2.59
190	12.89	2.06	16.22	2.03	13.28	2.49
200	13.42	1.98	17.26	1.91	13.85	2.38
210	13.98	1.90	18.46	1.78	14.42	2.29
220	14.44	1.84	19.58	1.68	14.98	2.20
230	15.08	1.76	20.89	1.58	15.58	2.12
240	15.68	1.70	21.97	1.50	16.25	2.03

Table 4.57 Comparison of time and speedup of Ladder215S, Ladder315C and Ladder315S with N = 18 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `compact`

`Compact` N = 18 (time in seconds)						
model	Ladder 215S(N)		Ladder 315C(N)		Ladder 315S(N)	
threads	time	speedup	time	speedup	time	Speedup
1	32.93	1.00	35.73	1.00	35.78	1.00
10	11.26	2.93	11.84	3.02	11.85	3.02
20	8.27	3.98	8.54	4.19	8.34	4.29
30	6.84	4.82	7.78	4.59	7.75	4.62
40	6.72	4.90	6.90	5.18	6.77	5.28
50	6.98	4.72	7.01	5.10	6.96	5.14
60	7.24	4.55	7.35	4.86	7.09	5.05
70	7.66	4.30	7.73	4.62	7.60	4.71
80	8.15	4.04	8.10	4.41	8.04	4.45
90	8.64	3.81	8.58	4.17	8.29	4.32
100	9.13	3.61	9.14	3.91	8.94	4.00
110	9.76	3.37	9.85	3.63	9.38	3.81
120	10.45	3.15	10.40	3.44	9.78	3.66
130	11.15	2.95	11.11	3.22	10.21	3.50
140	11.83	2.78	11.83	3.02	11.09	3.23
150	12.64	2.60	12.63	2.83	11.75	3.05
160	13.43	2.45	13.42	2.66	12.50	2.86
170	14.41	2.28	14.38	2.48	13.28	2.69
180	15.39	2.14	15.32	2.33	13.73	2.61
190	16.47	2.00	16.36	2.18	13.99	2.56
200	17.52	1.88	17.41	2.05	14.85	2.41
210	18.81	1.75	18.68	1.91	16.51	2.17
220	19.97	1.65	19.39	1.84	16.95	2.11
230	21.32	1.54	21.17	1.69	17.91	2.00
240	22.47	1.47	22.30	1.60	17.61	2.03

Table 4.58 Comparison of time and speedup of Chain, Ladder and Ladder 215C with *N* = 18 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `balanced`

`Balanced` *N* = 18 (time in seconds)						
model	Chain(*N*)		Ladder(*N*)		Ladder 215C(*N*)	
threads	time(sec.)	speedup	time(sec.)	speedup	time(sec.)	speedup
1	27.67	1.00	32.94	1.00	33.01	1.00
10	7.86	3.52	9.13	3.61	9.02	3.66
20	6.79	4.07	7.28	4.52	7.41	4.46
30	5.38	5.14	5.47	6.02	5.73	5.76
40	6.45	4.29	6.19	5.32	6.56	5.03
50	8.09	3.42	7.25	4.54	8.03	4.11
60	10.33	2.68	8.83	3.73	10.15	3.25
70	15.69	1.76	13.69	2.41	15.46	2.13
80	16.99	1.63	14.79	2.23	16.58	1.99
90	18.59	1.49	15.94	2.07	17.87	1.85
100	19.98	1.38	17.03	1.93	19.47	1.69
110	22.23	1.24	18.50	1.78	21.30	1.55
120	23.45	1.18	19.77	1.67	22.54	1.46
130	26.04	1.06	21.92	1.50	24.60	1.34
140	28.14	0.98	23.55	1.40	26.58	1.24
150	29.77	0.93	25.15	1.31	28.41	1.16
160	31.78	0.87	27.32	1.21	30.53	1.08
170	33.71	0.82	29.00	1.14	32.36	1.02
180	35.58	0.78	30.44	1.08	33.81	0.98
190	35.97	0.77	30.82	1.07	34.35	0.96
200	38.04	0.73	32.16	1.02	35.90	0.92
210	40.38	0.69	33.98	0.97	38.23	0.86
220	42.95	0.64	36.03	0.91	40.91	0.81
230	45.91	0.60	38.29	0.86	43.22	0.76
240	47.75	0.58	40.16	0.82	45.41	0.73

Table 4.59 Comparison of time and speedup of Ladder215S, Ladder315C and Ladder315S with N = 18 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `balanced`

`Balanced` N = 18 (time in seconds)						
model	Ladder 215S(N)		Ladder 315C(N)		Ladder 315S(N)	
threads	time	speedup	time	speedup	time	speedup
1	32.93	1.00	35.73	1.00	35.78	1.00
10	9.00	3.66	9.61	3.72	9.00	3.98
20	7.33	4.49	7.56	4.73	6.99	5.12
30	5.55	5.94	7.66	4.66	7.23	4.95
40	6.28	5.25	6.92	5.17	6.50	5.51
50	7.40	4.45	8.46	4.22	7.95	4.50
60	9.02	3.65	10.64	3.36	9.93	3.60
70	14.19	2.32	16.50	2.17	15.49	2.31
80	15.35	2.15	17.80	2.01	16.67	2.15
90	16.51	1.99	19.17	1.86	18.33	1.95
100	17.76	1.85	20.67	1.73	19.73	1.81
110	19.43	1.70	22.64	1.58	21.57	1.66
120	20.59	1.60	24.03	1.49	22.96	1.56
130	22.60	1.46	26.17	1.37	24.76	1.45
140	24.46	1.35	27.42	1.30	26.98	1.33
150	25.93	1.27	29.72	1.20	28.98	1.23
160	28.12	1.17	32.11	1.11	31.11	1.15
170	29.69	1.11	33.76	1.06	32.97	1.09
180	31.10	1.06	35.41	1.01	34.59	1.03
190	31.45	1.05	35.80	1.00	35.34	1.01
200	32.67	1.01	37.69	0.95	37.01	0.97
210	34.48	0.96	39.47	0.91	39.41	0.91
220	36.43	0.90	42.79	0.84	42.09	0.85
230	38.92	0.85	44.66	0.80	42.72	0.84
240	40.70	0.81	47.60	0.75	46.35	0.77

4.4.8 Comparison of Speedup between Xeon Phi and Odin

To sum up the above arguments, `Compact` type with 240 threads when $N = 24$ can get a better speedup on Intel Xeon Phi, with the same size $N = 24$, set 8 threads on Odin can obtain a good speedup.

Table 4.60 the execution speed of the same problem on Odin and the Xeon Phi coprocessor is computed. See that the Xeon Phi is around 3.6 times faster than Odin for all 6 models. $\boldsymbol{Efficiency = Speedup/amount\ of\ cores}$, on the Xeon Phi, the efficiency of Ladder models is up to 0.88, it is very close to 1, therefore, when $N = 24$, Xeon Phi has a rather good speedup. A good speedup can be got when Odin set 8 threads, I picked the data form the Table 4.35, (time in seconds)

Table 4.36 and Table 4.37 where threads $p = 8$, $N = 24$. The efficiency of each core is very low, just around 0.16, in this case, speed of Odin is disappointing.

Table 4.60 Comparison of time and speedup between Xeon Phi and Odin with $N = 24$

N =24	Xeon Phi `Compact` (240 threads)			Odin (8 threads)		
models name	time(sec.)	speedup	efficiency	time(sec.)	speedup	efficiency
Chain	39.14	45.93	0.77	146.09	1.75	0.15
Ladder2	46.29	50.86	0.85	169.46	1.81	0.15
Ladder215C	46.26	50.89	0.85	166.61	1.89	0.16
Ladder215S	46.26	50.91	0.85	170.62	1.86	0.16
Ladder315C	49.02	52.40	0.87	180.08	1.95	0.16
Ladder315S	48.74	52.71	0.88	177.01	2.04	0.17

Table 4.61 Comparison of time and speedup between Xeon Phi and Odin with $N = 18$

$N = 18$	Xeon Phi `Compact` (40 threads)			Odin (8 threads)		
models name	time(sec.)	speedup	efficiency	time(sec.)	speedup	efficiency
Chain	6.33	4.20	0.07	0.74	4.04	0.34
Ladder2	6.72	4.90	0.08	0.79	4.48	0.37
Ladder215C	6.59	5.01	0.08	0.79	4.44	0.37
Ladder215S	6.72	4.90	0.08	0.73	4.67	0.39
Ladder315C	6.90	5.18	0.09	0.91	4.31	0.36
Ladder315S	6.77	5.28	0.09	1.00	3.93	0.33

Table 4.61 compared time and speedup between Intel Xeon and Odin with $N = 18$. `Compact` affinity type with 40 threads can get the best speedup on Xeon Phi, so I pick this set of data as the object of comparison; In Odin, I also use the data with number of threads $p = 8$. Odin and Xeon Phi have a similar speedup in this size, but the result of Odin is faster than Xeon Phi. I have to admit that is Xeon Phi do not fit on a calculation with small size N, the smaller the slower. Hence, the calculations with small size is better executed on Odin than on the Xeon Phi.

Table 4.62 Comparison Ladder 215C(*N*) with *N* = 16,18,20,22,24,26 on two machines

Ladder 215C(*N*)	Xeon Phi `Compact`		Odin	
N	Time (sec.)	Speedup	Time (sec.)	Speedup
16	17.73	0.47	0.32	2.54
18	16.25	2.03	0.79	4.44
20	15.75	8.57	6.42	2.47
22	21.31	25.23	33.75	2.10
24	46.26	50.89	166.61	1.89
26	95.00	94.98	601.43	2.41

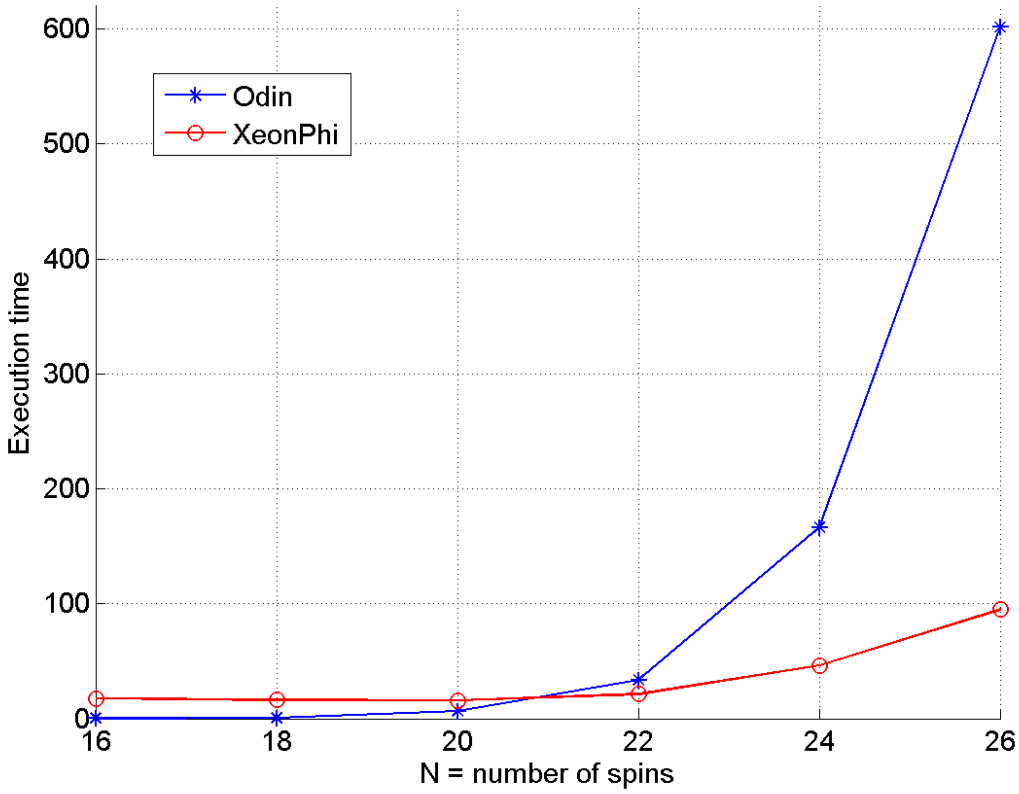


Figure 4.76 Comparison of time between Odin and Xeon Phi on Ladder215C(24)

Table 4.62 shows the execution times and parallel speedups obtained on odin and the Intel Xeon Phi machine for the Ladder 215C model. The execution times are also shown graphically by Figure 4.76. The data show clearly that for smaller N, the conventional multi-core architecture of odin yields much better results. For larger N's however the situation changes and the Intel Xeon Phi becomes the faster machine.

These results can be understood by the architectural differences between the two machines. Modern multi-core CPU's (like those in odin) have an out-of-order core architecture that has a superior performance of a single core than the in-order architecture of the Xeon-Phi cores. For smaller system sizes where memory access speed is not the limiting factor, the better performance of the individual cores gives the multi-core CPU an advantage. For larger system sizes, however, the execution speed of the program becomes more and more limited by memory access time and the out-of-order core architecture loses its advantage. The Xeon-Phi however has not only a large number of cores on the chip but also provides a large memory bandwidth of 320 GB/s. This large memory bandwidth explains the better performance of the Xeon Phi co-processor in case of the larger N's.

5 Summary and Conclusions

The work described in this thesis has two goals. The first goal is to study the performance and possible advantages of the Block-Davidson and RMM-DIIS algorithms for the exact diagonalization of quantum spin systems. In order to achieve this, programs were developed for each of these algorithms and applied to typical spin-1/2 Heisenberg models.

The second goal of this thesis is to test whether the new Intel Many-Integrated-Core technology provides a suitable architecture to carry out exact diagonalization studies. To this end, the RMM-DIIS code developed in the first part of the thesis was tested extensively on an Intel Xeon Phi coprocessor as well as a conventional multi-core machine.

5.1 Block Davidson and RMM-DIIS Algorithms

The results presented in chapter 4 show that the Block Davidson method as well as the RMM-DIIS algorithms can be used for the determination of the lowest eigenpairs of the Hamilton matrices describing typical quantum spin systems. Repeated tests with different initial trial vectors did not indicate any specific convergence problems for either method.

As expected, the RMM-DIIS method reaches the same level of convergence in less computing time than the Block Davidson method due to the reduced need for

reorthogonalization. As shown by the results in Sec. 4.3, the difference is sometimes rather large. The reason for this is the extreme size of the matrices which makes the reorthogonalization computationally very costly.

In electronic structure calculations it has been found Ref. [3] that one has to be careful with the initialization of the RMM-DIIS algorithm. The reason is that this algorithm will find the eigenvectors closest to the initial trial vectors and there is no guarantee that the states found will be the lowest vectors. In addition to this, there are two possibilities to perform the intermediate Ritz projections. One method (V_{max}) uses the full iterative subspace obtained for all trial states in the Ritz step. The other method (V_{min}) uses only the latest iteration of each trial vector in the Ritz step.

In total, I have developed and tested three different versions of the RMM-DIIS algorithm. The first version (BKDV-V_{min}) starts the calculations with the Block-Davidson algorithm and then switches to the V_{min} version of RMM-DIIS. The second variant (V_{max}) uses exclusively the V_{max} version of the algorithm. The third variant (V_{max}-V_{min}) starts the algorithm with V_{max} and then switches to the Vmin version of the algorithm.

Surprisingly, the tests indicated no initialization problems for any of the three variants of the RMM-DIIS method. The most efficient method was the third variant V_{max}-V_{min}. It is not clear why in our tests, the lowest eigenvectors were found even for the second and third variant which use the RMM-DIIS method exclusively. Possible reasons are the fact that compared to typical electronic structure calculations, only very few eigenvectors were sought. It might also play a role that the electronic structure calculations in Ref. [3] are self-consistent calculations in which the Hamilton matrix

depends on the eigenvectors. This dependency might compound the initialization problem.

A severe practical problem of exact diagonalization studies is the extreme size of even single vectors in the problem. This usually imposes severe limits on the maximum size of the system. In this thesis, this problem was not directly addressed. The RMM-DIIS algorithm would nevertheless be a promising candidate for a memory saving scheme. The reason is that between the Ritz steps, this algorithm iterates only individual eigenstates. It is therefore conceivable to design a program that stores the vectors on different nodes and communicates them during the Ritz step.

5.2 Parallel Speedup on Intel Xeon Phi

In order to test the performance of the Intel Many-Integrated Core technology for the diagonalization of quantum spin systems, I ran the RMM-DIIS algorithm on an Intel Xeon Phi coprocessor and a convention multi-core machine (Odin).

The maximum parallel speedups obtained on the conventional 12-core machine was less than 5. This was not unexpected due to the large size of the matrices and vectors. In addition to this, operations on sparse matrices often access the memory in a non-ideal manner.

On the Intel Xeon Phi, the maximum parallel speedup and the performance in general depends strongly on the size of the problem. As shown in Sec. 4.4, the coprocessor is

not working efficiently at all for the smaller systems (N = 16 - 20). For the large systems (N = 24, 26) however, the Xeon Phi coprocessor achieves excellent speedups in the range 50 - 60. The speedups vary slightly between the different models. The controlling factor appears to be the number of elements per row in the matrix. The more elements there are (i.e. the less sparse the matrix is) the better the performance.

The speedup of the Xeon Phi coprocessor has to be seen in relation to the performance of one of its cores. While the device has 60 cores, a single core is much less powerful than a core on a typical multi-core machine like odin. Nevertheless, for the larger systems, the Xeon Phi coprocessor outperforms odin by a factor of three to four.

The results obtained in this thesis indicate that the Xeon Phi coprocessor is an excellent architecture for exact diagonalization problems. Most likely it is its large memory bandwidth that is responsible for this. One might ask, however, why the speedup of the Xeon Phi coprocessor is below 60 although each of the coprocessor cores can manage four hardware threads Ref.[12]. Most likely this is due to the fact that the program remains limited by memory access speed. In addition to this, the four hardware threads still must share some of the cores physical units

5.3 Comparison with other exact diagonalization studies

In this work only relatively small systems with $N \leq 26$ have been considered. This is in contrast to recent applications of the exact diagonalization method which uses system sizes N in the range from 28 to 40 Ref. [2][15][24][25]. There are two reasons for this discrepancy.

On the one hand, the calculations in this work have focused on the case where not only the lowest eigenvalue is sought but several of the lowest eigenvalues. This increases the memory requirements and reduces the size of the attainable systems. By lowering the number of eigenvalues from eight to four and by using some memory saving strategies it might well be possible to treat systems with $N = 28$ with the program developed in this thesis.

On the other hand, system sizes of $N = 30$ and above can usually only be treated if all symmetries of the quantum models are exploited. This is a cumbersome and time consuming task. The goal of this thesis was however not to reach the largest possible system sizes but to study the properties of the Block Davidson and RMM-DIIS method in general. For this reason, only the relatively simple conservation of S^z was used in this work. By using more symmetries it should be possible to extend the size of the systems considerably.

A conclusion from this work is that for exact diagonalization studies of quantum spin systems that require more than the lowest eigenvalue, the application of the RMM-DIIS method and the Block Davidson method is possible. In particular, the RMM-DIIS method might lead to shorter computing times due to the reduced need for orthogonalization of the trial states. In addition to this, the benchmark calculations show that the Xeon Phi co-processor is an interesting architecture for studies with system sizes above $N = 24$.

6 Appendix A

6.1 Speedup data on Odin

Table 6.1 Time and speedup of Chain(*N*) model with *N* = 26, 24 and 22 on Odin

Heisenberg chain(*N*) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
N = 26	8582MB	5963MB					
1	1208.36	1173.69	1083.89	1170.62	1212.05	**1169.72**	1.00
2	701.42	693.72	705.53	693.85	714.45	**701.79**	1.67
4	576.42	573.71	589.59	578.34	586.83	**580.98**	2.01
6	525.60	541.38	553.69	542.88	542.50	**541.21**	2.16
8	513.80	547.78	557.58	542.54	540.81	**540.50**	2.16
10	526.09	544.83	554.14	540.99	540.78	**541.37**	2.16
12	502.99	539.58	546.44	542.71	548.79	**536.10**	2.18
N = 24	2865MB	2184MB					
1	254.78	254.81	259.26	256.69	254.76	**256.06**	1.00
2	181.44	181.91	178.47	180.68	183.19	**181.14**	1.41
4	152.10	153.49	149.31	150.31	155.22	**152.09**	1.68
6	148.49	148.00	145.21	144.35	152.75	**147.76**	1.73
8	147.23	148.51	143.31	143.19	148.19	**146.09**	1.75
10	151.60	153.08	147.87	148.21	152.80	**150.71**	1.70
12	155.19	155.10	149.78	150.28	155.88	**153.24**	1.67
N = 22	1411MB	1234MB					
1	61.45	60.78	59.88	59.42	59.22	**60.15**	1.00
2	41.38	37.97	39.97	36.33	36.00	**38.33**	1.57
4	32.24	29.81	30.30	28.24	29.23	**29.97**	2.01
6	27.09	26.44	28.09	25.48	26.94	**26.81**	2.24
8	26.45	25.07	27.17	27.10	26.10	**26.38**	2.28
10	27.69	26.28	27.49	27.44	26.80	**27.14**	2.22
12	26.63	26.67	27.54	25.14	26.78	**26.55**	2.27

Table 6.2 Time and speedup of Chain(*N*) model with *N* = 20, 18 and16 on Odin

Heisenberg Chain(*N*) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
N = 20	1039MB	992MB					
1	13.36	13.16	15.84	13.18	13.21	**13.75**	1.00
2	8.02	8.15	8.19	8.21	8.10	**8.13**	1.69
4	5.97	6.21	5.90	6.16	6.19	**6.09**	2.26
6	5.49	5.50	5.57	5.81	5.34	**5.54**	2.48
8	5.29	5.58	5.57	5.66	5.62	**5.55**	2.48
10	5.27	5.45	5.66	5.57	5.45	**5.48**	2.51
12	5.53	5.60	5.66	5.71	5.61	**5.62**	2.45
N = 18	982MB	963MB					
1	3.02	2.95	3.04	2.95	2.95	**2.98**	1.00
2	1.70	1.52	1.58	1.87	1.58	**1.65**	1.81
4	1.00	0.90	1.06	0.90	0.89	**0.95**	3.13
6	0.71	0.70	0.93	0.76	0.73	**0.77**	3.89
8	0.74	0.69	0.73	0.77	0.77	**0.74**	4.04
10	0.75	0.75	0.76	0.79	0.80	**0.77**	3.87
12	0.91	0.97	0.96	0.92	0.99	**0.95**	3.14
N = 16	952MB						
1	0.71	0.70	0.70	0.70	0.70	**0.70**	1.00
2	0.46	0.42	0.40	0.40	0.40	**0.41**	1.70
4	0.31	0.30	0.30	0.30	0.30	**0.30**	2.33
6	0.33	0.33	0.30	0.29	0.30	**0.31**	2.25
8	0.30	0.33	0.30	0.31	0.33	**0.31**	2.24
10	0.42	0.42	0.42	0.36	0.40	**0.40**	1.73
12	0.45	0.44	0.45	0.46	0.45	**0.45**	1.56

Table 6.3 Time and speedup of Ladder2(*N*) model with *N* = 26,24 and 22, on Odin

Heisenberg Ladder2(*N*) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
N = 26	9354MB	6768MB					
1	1491.41	1485.34	1498.03	1484.01	1484.85	**1488.73**	1.00
2	845.32	844.28	865.33	845.96	839.35	**848.05**	1.76
4	656.97	686.26	660.47	688.01	675.54	**673.45**	2.21
6	630.07	623.82	639.71	616.73	631.19	**628.30**	2.37
8	589.68	622.94	633.13	602.35	620.53	**613.73**	2.43
10	584.50	604.59	594.01	628.19	617.31	**605.72**	2.46
12	630.43	601.42	633.51	623.66	602.78	**618.36**	2.41
N = 24	3058MB	2378MB					
1	307.87	308.99	304.96	304.91	308.64	**307.08**	1.00
2	219.38	219.40	218.63	218.61	217.77	**218.76**	1.40
4	184.31	179.66	181.38	180.51	180.11	**181.19**	1.69
6	172.98	160.76	173.67	172.16	173.65	**170.64**	1.80
8	173.15	157.28	173.01	171.08	172.78	**169.46**	1.81
10	175.18	174.32	175.57	174.94	174.81	**174.97**	1.76
12	179.92	175.82	179.30	177.91	175.87	**177.76**	1.73
N = 22	1458MB	1280MB					
1	70.01	70.08	70.03	70.06	74.29	**70.89**	1.00
2	43.79	44.73	45.37	43.73	41.74	**43.87**	1.62
4	34.96	35.37	35.97	34.97	32.98	**34.85**	2.03
6	32.88	33.25	33.36	33.11	30.63	**32.65**	2.17
8	32.40	33.23	32.70	32.64	30.04	**32.20**	2.20
10	33.23	33.41	31.03	33.45	29.81	**32.19**	2.20
12	34.19	33.90	30.33	34.20	29.54	**32.43**	2.19

Table 6.4 Time and speedup of Ladder2(*N*) model with *N* = 20, 18 and16 on Odin

Heisenberg Ladder2(*N*) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
N = 20	1050MB	1003MB					
1	15.52	15.53	15.25	15.49	15.66	**15.49**	1.00
2	9.66	9.74	9.72	9.55	9.49	**9.63**	1.61
4	7.62	7.50	7.51	7.49	7.32	**7.49**	2.07
6	6.73	6.68	6.58	6.81	6.40	**6.64**	2.33
8	6.47	6.72	6.64	6.57	6.31	**6.54**	2.37
10	6.64	6.61	6.86	6.64	6.53	**6.66**	2.33
12	6.93	7.00	6.73	6.95	6.93	**6.91**	2.24
N = 18	984MB	981MB					
1	3.49	3.54	3.54	3.53	3.55	**3.53**	1.00
2	1.84	1.85	1.86	1.85	1.86	**1.85**	1.91
4	1.13	1.14	1.13	1.20	1.17	**1.15**	3.07
6	0.91	0.84	0.87	0.86	0.85	**0.87**	4.08
8	0.73	0.78	0.88	0.75	0.80	**0.79**	4.48
10	0.87	0.79	0.95	0.89	0.82	**0.86**	4.09
12	0.98	0.98	1.03	1.05	1.05	**1.02**	3.47
N = 16	951MB	948MB					
1	0.82	0.82	0.82	0.82	0.82	**0.82**	1.00
2	0.46	0.46	0.46	0.51	0.46	**0.47**	1.74
4	0.33	0.33	0.34	0.34	0.33	**0.33**	2.46
6	0.33	0.32	0.34	0.35	0.32	**0.33**	2.46
8	0.33	0.32	0.33	0.32	0.32	**0.32**	2.53
10	0.37	0.36	0.38	0.34	0.44	**0.38**	2.16
12	0.48	0.46	0.48	0.46	0.46	**0.47**	1.75

Table 6.5 Time and speedup of Ladder 215C(N) with N = 26, 24 and 22 on Odin

Heisenberg Ladder215C(N) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
N = 26	9354MB	6768MB					
1	1358.32	1523.56	1501.96	1504.02	1365.02	**1450.58**	1.00
2	861.17	867.98	852.24	867.11	852.99	**860.30**	1.69
4	662.47	662.25	695.12	673.30	692.11	**677.05**	2.14
6	627.53	609.93	616.93	624.01	642.56	**624.19**	2.32
8	615.06	591.68	607.90	604.46	588.06	**601.43**	2.41
10	600.64	580.73	625.91	593.96	626.13	**605.47**	2.40
12	624.96	577.95	631.38	581.71	612.63	**605.73**	2.39
N = 24	3058MB	2378MB					
1	312.11	312.37	314.66	315.11	317.42	**314.33**	1.00
2	215.57	229.89	214.73	213.01	213.20	**217.28**	1.45
4	179.66	191.82	176.63	177.41	174.74	**180.05**	1.75
6	168.85	175.67	167.34	167.96	165.85	**169.13**	1.86
8	168.77	169.02	165.11	165.79	164.36	**166.61**	1.89
10	169.74	169.39	167.10	167.05	166.54	**167.96**	1.87
12	173.86	169.03	171.77	169.47	168.07	**170.44**	1.84
N = 22	1458MB	1280MB					
1	69.98	71.05	71.03	70.91	71.64	**70.92**	1.00
2	44.55	45.27	45.11	45.36	45.22	**45.10**	1.57
4	36.08	35.83	36.66	39.05	36.36	**36.80**	1.93
6	33.61	34.14	34.56	34.57	34.16	**34.20**	2.07
8	32.93	33.59	34.17	33.78	34.29	**33.75**	2.10
10	33.58	33.27	34.62	34.72	34.90	**34.22**	2.07
12	33.93	33.27	35.10	35.02	35.36	**34.54**	2.05

Table 6.6 Time and speedup of Ladder 215C(*N*) with *N* = 20, 18 and16 on Odin

Heisenberg Ladder215C(*N*) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
N = 20	1050MB	1003MB					
1	16.44	16.51	15.49	15.49	15.48	**15.88**	1.00
2	9.30	9.30	9.65	9.72	9.72	**9.53**	1.67
4	7.18	7.04	7.47	7.61	7.55	**7.37**	2.16
6	6.29	6.33	6.66	6.66	6.69	**6.53**	2.43
8	6.20	6.22	6.63	6.57	6.47	**6.42**	2.47
10	6.16	6.14	6.58	6.59	6.64	**6.42**	2.47
12	6.30	6.19	6.74	6.81	6.87	**6.58**	2.41
N = 18	984MB	981MB					
1	3.49	3.52	3.53	3.52	3.53	**3.52**	1.00
2	2.08	1.85	1.86	2.14	1.85	**1.96**	1.80
4	1.16	1.21	1.14	1.10	1.13	**1.15**	3.07
6	0.97	1.00	1.12	0.89	1.09	**1.01**	3.47
8	0.74	0.86	0.81	0.78	0.77	**0.79**	4.44
10	0.78	0.93	0.83	0.94	0.86	**0.87**	4.06
12	1.00	0.97	1.05	0.99	1.06	**1.01**	3.47
N = 16	951MB	948MB					
1	0.80	0.82	0.82	0.82	0.80	**0.81**	1.00
2	0.47	0.51	0.48	0.46	0.45	**0.47**	1.72
4	0.32	0.33	0.31	0.33	0.32	**0.32**	2.51
6	0.33	0.33	0.32	0.33	0.32	**0.33**	2.49
8	0.31	0.33	0.29	0.34	0.32	**0.32**	2.54
10	0.36	0.44	0.43	0.41	0.35	**0.40**	2.04
12	0.47	0.46	0.46	0.46	0.46	**0.46**	1.76

Table 6.7 Time and speedup of Ladder 215S(N) with N = 26, 24 and 22 on Odin

Heisenberg Ladder215S(N) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
N = 26	9354MB	6768MB					
1	1470.10	1534.82	1405.18	1514.15	1508.64	**1486.58**	1.00
2	841.47	869.74	882.98	888.47	849.31	**866.39**	1.72
4	684.45	690.01	731.44	680.91	676.13	**692.59**	2.15
6	614.11	619.52	640.68	624.63	631.91	**626.17**	2.37
8	589.24	596.88	621.57	601.81	613.16	**604.53**	2.46
10	589.81	603.36	624.11	589.38	613.08	**603.95**	2.46
12	623.06	608.92	659.49	578.15	646.98	**623.32**	2.38
N = 24	3058MB	2378MB					
1	306.74	327.60	304.39	305.01	343.28	**317.40**	1.00
2	218.43	206.97	218.86	220.24	205.02	**213.90**	1.48
4	180.30	171.00	181.41	182.58	164.67	**175.99**	1.80
6	171.07	171.52	174.43	174.36	154.13	**169.10**	1.88
8	179.25	173.43	172.67	172.47	155.28	**170.62**	1.86
10	198.52	176.30	176.43	176.24	155.31	**176.56**	1.80
12	185.35	181.10	179.57	177.88	157.37	**176.25**	1.80
N = 22	1458MB	1280MB					
1	70.02	71.07	70.45	70.02	69.39	**70.19**	1.00
2	44.57	45.35	45.54	44.08	44.92	**44.89**	1.56
4	35.97	35.49	36.84	35.62	35.63	**35.91**	1.95
6	33.34	31.87	33.97	33.36	33.20	**33.15**	2.12
8	32.77	33.73	34.00	32.60	33.36	**33.29**	2.11
10	33.44	34.47	34.88	34.27	33.50	**34.11**	2.06
12	34.24	33.76	35.96	34.82	34.13	**34.58**	2.03

Table 6.8 Time and speedup of Ladder 215S(N) with N = 20, 18 and 16 on Odin

Heisenberg Ladder215S(N) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
N = 20	1050MB	1003MB					
1	15.39	15.37	15.36	15.45	15.44	**15.40**	1.00
2	9.62	9.64	9.66	9.76	9.61	**9.66**	1.59
4	7.49	7.52	7.49	7.52	7.64	**7.53**	2.05
6	6.75	6.95	6.74	6.79	6.75	**6.80**	2.27
8	6.96	6.72	6.77	6.82	6.40	**6.73**	2.29
10	6.86	6.87	6.94	6.83	6.77	**6.86**	2.25
12	7.13	7.00	7.09	6.85	6.83	**6.98**	2.21
N = 18	984MB	981MB					
1	3.45	3.56	3.27	3.33	3.46	**3.41**	1.00
2	1.80	2.35	2.10	1.81	1.68	**1.95**	1.75
4	1.15	1.04	0.96	1.03	1.04	**1.05**	3.27
6	0.87	0.79	0.98	0.80	0.75	**0.84**	4.08
8	0.72	0.65	0.83	0.71	0.75	**0.73**	4.67
10	0.83	0.73	0.70	0.77	0.78	**0.76**	4.48
12	0.98	0.98	0.91	1.03	0.93	**0.96**	3.54
N = 16	951MB	948MB					
1	0.71	0.71	0.72	0.64	0.65	**0.69**	1.00
2	0.45	0.42	0.41	0.48	0.38	**0.43**	1.60
4	0.37	0.30	0.32	0.30	0.23	**0.30**	2.25
6	0.37	0.29	0.31	0.31	0.26	**0.31**	2.23
8	0.34	0.30	0.30	0.30	0.25	**0.30**	2.29
10	0.37	0.41	0.38	0.47	0.34	**0.39**	1.74
12	0.39	0.48	0.47	0.50	0.42	**0.45**	1.52

Table 6.9 Time and speedup of Ladder315C(*N*) and 315S(*N*) model with *N* = 24 and 18 on Odin

Heisenberg Ladder315C(*N*) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
N = 24	3123MB	2442MB					
1	351.96	351.86	352.01	351.70	351.60	**351.83**	1.00
2	249.88	247.97	248.97	249.33	250.35	**249.30**	1.41
4	198.04	198.05	199.62	199.23	198.70	**198.73**	1.77
6	188.00	187.12	187.62	185.86	187.19	**187.16**	1.88
8	179.09	179.01	182.60	180.49	179.24	**180.08**	1.95
10	183.33	183.73	183.54	183.47	183.22	**183.46**	1.92
12	187.43	187.45	187.66	187.97	188.19	**187.74**	1.87
N = 18	985MB	982MB					
1	3.92	3.92	3.91	3.91	3.92	**3.92**	1.00
2	2.08	2.33	2.48	2.05	2.05	**2.20**	1.78
4	1.26	1.17	1.15	1.34	1.26	**1.23**	3.17
6	1.01	0.91	1.10	0.93	0.95	**0.98**	3.99
8	0.93	0.82	1.03	0.92	0.84	**0.91**	4.31
10	1.02	0.86	1.09	1.04	0.87	**0.98**	4.01
12	1.02	1.00	1.02	1.00	1.05	**1.02**	3.84
Heisenberg Ladder315S(*N*) (recorded time in seconds)							
N = 24	3123MB	2442MB					
1	374.54	352.03	352.03	363.48	363.26	**361.07**	1.00
2	231.54	249.75	252.14	252.05	250.75	**247.25**	1.46
4	179.94	197.71	198.49	196.26	183.03	**191.09**	1.89
6	168.42	186.61	186.71	172.95	186.93	**180.32**	2.00
8	165.48	181.62	178.58	178.36	181.00	**177.01**	2.04
10	178.46	183.07	182.85	183.06	181.26	**181.74**	1.99
12	184.60	185.16	187.84	187.90	185.65	**186.23**	1.94
N = 18	985MB	982MB					
1	3.91	3.92	3.93	3.92	3.92	**3.92**	1.00
2	2.35	2.08	2.48	2.53	2.53	**2.39**	1.64
4	1.16	1.31	1.38	1.37	1.22	**1.29**	3.04
6	0.91	1.01	0.95	1.16	1.13	**1.03**	3.80
8	0.84	1.09	0.89	1.13	1.04	**1.00**	3.93
10	0.85	0.92	0.96	0.91	1.14	**0.96**	4.10
12	1.06	1.08	1.08	1.03	1.12	**1.07**	3.65

6.2 Speedup data on Xeon Phi

6.2.1 Heisenberg Chain on Xeon Phi

Table 6.10 Time and speedup of Chain(*N*) model with *N* = 26 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `scatter`

`Scatter` Heisenberg Chain(26) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	–	–	–	–	–	–	–
10	–	–	–	–	–	–	–
20	–	–	–	–	–	–	–
30	–	–	–	–	–	–	–
40	–	–	–	–	–	–	–
50	–	–	–	–	–	–	–
60	–	–	–	–	–	–	–
70	–	–	–	–	–	–	–
80	–	–	–	–	–	–	–
90	–	–	–	–	–	–	–
100	–	–	–	–	–	–	–
110	–	–	–	–	–	–	–
120	–	–	–	–	–	–	–
130	–	–	–	–	–	–	–
140	–	–	–	–	–	–	–
150	–	–	–	–	–	–	–
160	–	–	–	–	–	–	–
170	–	–	–	–	–	–	–
180	–	–	–	–	–	–	–
190	–	–	–	–	–	–	–
200	–	–	–	–	–	–	–
210	–	–	–	–	–	–	–
220	–	–	–	–	–	–	–
230	–	–	–	–	–	–	–
240	–	–	–	–	–	–	–

Table 6.11 Time and speedup of Chain(*N*) model with *N* = 24 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `scatter`

`Scatter` Heisenberg Chain(24) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	1797.55	1798.52	1797.94	1797.59	1797.87	**1797.89**	1.00
10	408.80	399.77	384.35	448.46	428.65	**414.00**	4.34
20	236.42	252.82	231.32	260.13	262.17	**248.57**	7.23
30	170.63	192.00	178.05	181.27	178.54	**180.10**	9.98
40	138.15	139.45	133.77	134.82	131.20	**135.48**	13.27
50	109.58	113.72	113.82	109.84	111.93	**111.78**	16.08
60	97.33	96.03	90.53	96.13	95.72	**95.14**	18.90
70	88.95	86.39	82.71	89.39	86.34	**86.75**	20.72
80	76.29	75.00	74.04	74.72	77.29	**75.47**	23.82
90	69.56	68.96	71.00	73.05	70.00	**70.51**	25.50
100	66.41	63.45	68.11	67.01	63.22	**65.64**	27.39
110	62.47	62.51	65.99	60.10	62.60	**62.73**	28.66
120	60.72	59.27	59.91	60.06	60.07	**60.01**	29.96
130	59.27	58.18	61.14	57.11	61.54	**59.45**	30.24
140	56.53	57.06	54.90	58.87	59.90	**57.45**	31.29
150	53.15	52.62	53.68	56.79	54.43	**54.14**	33.21
160	51.47	52.55	50.36	55.08	58.34	**53.56**	33.57
170	52.61	51.83	51.53	52.29	56.63	**52.98**	33.94
180	48.93	48.64	50.17	48.99	53.07	**49.96**	35.99
190	48.79	47.53	50.04	47.84	49.52	**48.74**	36.89
200	48.00	46.30	48.64	46.73	48.98	**47.73**	37.67
210	44.96	44.95	45.93	45.09	47.52	**45.69**	39.35
220	43.63	43.12	43.07	44.67	46.79	**44.26**	40.63
230	41.86	41.68	41.74	41.57	41.90	**41.75**	43.06
240	41.40	41.43	41.52	41.52	41.43	**41.46**	43.36

Table 6.12 Time and speedup of Chain(*N*) model with *N* = 22 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `scatter`

`Scatter` Heisenberg Chain(22) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	433.11	434.21	434.14	434.11	434.01	**433.92**	1.00
10	111.12	95.31	94.57	99.04	111.58	**102.32**	4.24
20	60.25	60.50	59.53	55.35	61.86	**59.50**	7.29
30	42.34	47.07	42.29	40.78	42.61	**43.02**	10.09
40	31.92	31.26	31.48	31.38	31.34	**31.47**	13.79
50	26.28	26.51	26.03	26.09	26.00	**26.18**	16.57
60	22.94	23.12	22.63	23.00	22.99	**22.94**	18.92
70	21.45	21.14	21.10	21.26	21.20	**21.23**	20.44
80	20.23	18.12	18.71	18.73	18.67	**18.89**	22.97
90	18.46	18.21	17.58	17.66	17.64	**17.91**	24.23
100	16.73	16.66	16.65	16.59	16.27	**16.58**	26.17
110	15.59	15.56	15.51	15.52	15.25	**15.49**	28.02
120	14.78	14.66	14.91	14.70	14.90	**14.79**	29.34
130	14.90	14.43	14.55	14.38	14.61	**14.58**	29.77
140	14.99	14.33	13.97	13.82	13.89	**14.20**	30.56
150	17.25	16.71	17.40	16.63	16.71	**16.94**	25.61
160	17.09	16.65	16.67	16.53	16.32	**16.65**	26.06
170	16.99	16.55	17.15	16.26	16.27	**16.65**	26.07
180	17.21	17.18	17.51	16.57	16.73	**17.04**	25.46
190	16.95	16.86	18.06	16.73	16.99	**17.12**	25.35
200	17.28	17.28	18.64	17.04	17.22	**17.49**	24.80
210	17.53	17.37	19.08	17.27	17.49	**17.75**	24.45
220	17.86	17.64	19.61	17.50	17.83	**18.09**	23.99
230	18.06	17.78	19.95	17.60	18.03	**18.28**	23.73
240	21.00	18.38	20.82	18.22	18.73	**19.43**	22.33

Table 6.13 Time and speedup of Chain(*N*) model with *N* = 20 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `scatter`

`Scatter` Heisenberg Chain(20) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	108.24	108.40	108.28	108.27	108.29	**108.30**	1.00
10	26.13	22.82	22.84	22.82	22.80	**23.49**	4.61
20	14.89	14.69	14.73	14.16	14.13	**14.52**	7.46
30	10.74	11.11	10.82	11.09	11.08	**10.97**	9.87
40	9.99	9.96	10.07	9.93	9.88	**9.97**	10.87
50	9.24	9.18	9.31	9.21	9.19	**9.23**	11.74
60	8.87	8.85	8.71	8.88	8.83	**8.83**	12.27
70	8.88	8.83	8.88	8.92	8.84	**8.87**	12.21
80	8.66	8.62	8.76	8.74	8.61	**8.68**	12.48
90	7.26	7.24	7.54	7.46	7.26	**7.35**	14.73
100	7.75	7.57	7.92	7.78	7.78	**7.76**	13.95
110	8.11	7.93	8.12	8.16	8.17	**8.10**	13.38
120	8.47	8.39	8.43	8.46	8.48	**8.45**	12.82
130	9.09	9.07	9.08	9.14	9.12	**9.10**	11.90
140	9.52	9.57	9.64	9.73	9.56	**9.60**	11.28
150	10.01	10.05	10.04	10.22	10.08	**10.08**	10.74
160	10.49	10.44	10.46	10.57	10.68	**10.53**	10.29
170	10.96	10.88	11.00	11.00	11.00	**10.97**	9.87
180	11.36	11.45	11.46	11.40	11.40	**11.41**	9.49
190	12.37	12.38	12.38	12.45	12.42	**12.40**	8.73
200	12.80	12.84	12.84	12.86	12.82	**12.83**	8.44
210	13.27	13.23	13.35	13.34	13.30	**13.30**	8.14
220	13.67	13.66	13.72	13.74	13.73	**13.70**	7.90
230	14.06	14.06	14.13	14.14	14.10	**14.10**	7.68
240	14.86	14.54	14.77	14.77	14.68	**14.72**	7.36

Table 6.14 Time and speedup of Chain(*N*) model with *N* = 18 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `scatter`

`Scatter` Heisenberg Chain(18) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	28.00	27.91	27.94	27.93	26.60	**27.67**	1.00
10	7.08	7.55	7.04	7.04	7.05	**7.15**	3.87
20	5.58	5.74	5.55	5.41	5.54	**5.56**	4.97
30	3.87	4.04	3.94	3.97	3.95	**3.95**	7.00
40	4.06	4.13	4.10	4.14	4.13	**4.11**	6.73
50	4.46	4.43	4.46	4.53	4.51	**4.48**	6.18
60	5.02	4.99	5.00	5.03	5.05	**5.02**	5.51
70	5.65	5.63	5.66	5.70	5.68	**5.66**	4.89
80	6.22	6.17	6.16	6.19	6.19	**6.19**	4.47
90	6.54	6.68	6.70	6.74	6.70	**6.67**	4.15
100	7.18	6.96	7.15	7.19	7.20	**7.13**	3.88
110	7.65	7.64	7.65	7.68	7.70	**7.66**	3.61
120	8.34	8.31	8.33	8.37	8.39	**8.35**	3.32
130	9.11	9.06	9.06	9.09	9.11	**9.09**	3.05
140	9.66	9.60	9.61	9.41	9.68	**9.59**	2.88
150	10.15	10.12	10.14	10.14	10.15	**10.14**	2.73
160	10.74	10.70	10.72	10.72	10.72	**10.72**	2.58
170	11.31	11.24	11.22	11.26	11.25	**11.26**	2.46
180	11.82	11.77	11.75	11.85	11.81	**11.80**	2.34
190	12.65	12.57	12.56	12.59	12.61	**12.60**	2.20
200	13.17	13.10	13.09	13.06	13.12	**13.11**	2.11
210	13.74	13.65	13.62	13.63	13.67	**13.66**	2.03
220	14.30	14.21	14.18	14.18	14.22	**14.22**	1.95
230	14.80	14.71	14.67	14.66	14.71	**14.71**	1.88
240	15.63	15.25	15.17	15.17	15.23	**15.29**	1.81

Table 6.15 Time and speedup of Chain(*N*) model with $N = 16$ on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `scatter`

`Scatter` Heisenberg Chain(16) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	7.07	6.99	7.01	7.02	7.03	**7.02**	1.00
10	3.07	2.93	3.01	2.94	2.94	**2.98**	2.36
20	2.98	3.01	3.00	3.00	3.00	**3.00**	2.34
30	3.38	3.38	3.42	3.40	3.38	**3.39**	2.07
40	4.02	4.02	4.00	4.03	4.05	**4.02**	1.75
50	4.88	4.89	4.89	4.90	4.90	**4.89**	1.44
60	5.96	5.51	6.32	5.67	5.68	**5.83**	1.21
70	6.78	6.76	7.24	6.79	6.81	**6.87**	1.02
80	7.49	7.51	7.96	7.52	7.52	**7.60**	0.92
90	8.33	8.32	8.87	8.35	7.96	**8.36**	0.84
100	8.83	9.04	9.60	9.06	9.06	**9.12**	0.77
110	9.83	9.82	9.83	9.84	9.87	**9.84**	0.71
120	10.64	10.62	10.62	10.66	10.67	**10.64**	0.66
130	11.55	11.53	11.53	11.55	11.04	**11.44**	0.61
140	12.02	12.28	12.29	12.30	12.32	**12.24**	0.57
150	13.04	13.01	12.72	13.05	13.05	**12.97**	0.54
160	13.78	13.76	13.76	13.80	13.79	**13.78**	0.51
170	14.50	14.51	14.49	14.53	14.54	**14.51**	0.48
180	15.21	14.55	14.88	14.93	15.26	**14.96**	0.47
190	16.17	16.15	16.13	16.20	15.13	**15.96**	0.44
200	16.89	16.89	16.50	16.93	16.93	**16.83**	0.42
210	17.64	17.65	17.63	17.68	17.69	**17.66**	0.40
220	18.40	18.01	18.41	18.44	18.44	**18.34**	0.38
230	19.12	19.14	19.14	19.16	19.16	**19.15**	0.37
240	20.35	19.88	19.88	19.90	19.91	**19.98**	0.35

Table 6.16 Time and speedup of Chain(*N*) model with *N* = 26 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `compact`

`Compact` Heisenberg Chain(26) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	–	–	–	–	–	–	–
10	–	–	–	–	–	–	–
20	–	–	–	–	–	–	–
30	–	–	–	–	–	–	–
40	–	–	–	–	–	–	–
50	–	–	–	–	–	–	–
60	–	–	–	–	–	–	–
70	–	–	–	–	–	–	–
80	–	–	–	–	–	–	–
90	–	–	–	–	–	–	–
100	–	–	–	–	–	–	–
110	–	–	–	–	–	–	–
120	–	–	–	–	–	–	–
130	–	–	–	–	–	–	–
140	–	–	–	–	–	–	–
150	–	–	–	–	–	–	–
160	–	–	–	–	–	–	–
170	–	–	–	–	–	–	–
180	–	–	–	–	–	–	–
190	–	–	–	–	–	–	–
200	–	–	–	–	–	–	–
210	–	–	–	–	–	–	–
220	–	–	–	–	–	–	–
230	–	–	–	–	–	–	–
240	–	–	–	–	–	–	–

Table 6.17 Time and speedup of Chain(*N*) model with *N* = 24 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `compact`

`Compact` Heisenberg Chain(24) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	1797.55	1798.52	1797.94	1797.59	1797.87	**1797.89**	1.00
10	575.70	587.43	577.94	573.34	573.34	**577.55**	3.11
20	304.72	302.29	302.37	302.55	302.55	**302.90**	5.94
30	211.64	207.18	207.83	207.27	207.27	**208.23**	8.63
40	159.52	159.40	159.21	159.44	159.44	**159.40**	11.28
50	129.09	129.10	128.99	129.05	129.05	**129.06**	13.93
60	111.45	111.44	111.39	111.37	111.37	**111.40**	16.14
70	97.13	97.15	97.14	97.08	97.08	**97.12**	18.51
80	85.09	85.12	85.07	85.14	85.14	**85.11**	21.12
90	78.55	78.55	78.53	78.49	78.49	**78.52**	22.90
100	71.88	71.91	71.82	71.85	71.85	**71.86**	25.02
110	66.87	66.91	66.82	66.84	66.84	**66.86**	26.89
120	62.93	62.97	62.92	62.87	62.87	**62.91**	28.58
130	59.44	59.53	59.36	59.39	59.39	**59.42**	30.26
140	56.40	56.47	56.38	56.33	56.33	**56.38**	31.89
150	53.52	53.61	53.46	53.54	53.54	**53.53**	33.59
160	50.49	50.55	50.44	50.46	50.46	**50.48**	35.62
170	50.83	50.80	50.62	50.56	50.56	**50.67**	35.48
180	46.65	46.74	46.58	46.54	46.54	**46.61**	38.57
190	44.46	44.55	44.45	44.41	44.41	**44.45**	40.44
200	43.31	43.41	43.26	43.23	43.23	**43.29**	41.53
210	42.15	42.26	42.09	42.08	42.08	**42.13**	42.67
220	41.64	41.76	41.61	41.57	41.57	**41.63**	43.18
230	40.08	40.18	39.98	40.00	40.00	**40.05**	44.89
240	39.19	39.22	39.15	39.08	39.08	**39.14**	45.93

Table 6.18 Time and speedup of Chain(*N*) model with *N* = 22 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `compact`

`Compact` Heisenberg Chain(22) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	433.11	434.21	434.14	434.11	434.01	**433.92**	1.00
10	139.61	139.40	139.32	140.05	139.36	**139.55**	3.11
20	76.15	76.15	76.07	76.19	76.00	**76.11**	5.70
30	53.50	53.51	53.46	53.48	53.34	**53.46**	8.12
40	41.50	41.44	41.47	41.41	41.43	**41.45**	10.47
50	34.65	34.58	34.57	34.60	34.59	**34.60**	12.54
60	29.85	29.80	29.83	29.80	29.75	**29.81**	14.56
70	26.22	26.17	26.22	26.25	26.11	**26.19**	16.57
80	22.64	22.61	22.71	22.65	22.58	**22.64**	19.17
90	20.70	20.74	20.80	20.68	20.61	**20.71**	20.96
100	18.46	18.52	18.55	18.46	18.43	**18.48**	23.48
110	16.93	16.95	17.07	16.90	16.88	**16.94**	25.61
120	15.78	15.86	15.97	15.86	15.77	**15.85**	27.38
130	15.01	15.04	15.16	15.01	14.95	**15.04**	28.86
140	14.17	14.24	14.34	14.17	14.12	**14.21**	30.53
150	16.53	16.54	16.76	16.60	16.54	**16.59**	26.15
160	16.39	16.32	16.67	16.53	16.40	**16.46**	26.36
170	16.71	16.63	17.05	16.93	16.74	**16.81**	25.81
180	16.82	16.61	17.14	17.02	16.84	**16.88**	25.70
190	16.97	16.60	17.32	17.31	16.87	**17.01**	25.50
200	17.46	16.84	17.84	17.86	17.29	**17.46**	24.85
210	18.00	17.14	18.35	18.38	17.85	**17.94**	24.18
220	18.61	17.56	19.05	19.19	18.38	**18.56**	23.38
230	19.32	18.10	19.83	19.97	19.09	**19.26**	22.52
240	20.19	18.63	20.76	20.85	19.90	**20.06**	21.63

Table 6.19 Time and speedup of Chain(*N*) model with *N* = 20 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `compact`

`Compact` Heisenberg Chain(20) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	108.24	108.40	108.28	108.27	108.29	**108.30**	1.00
10	36.33	36.35	36.35	36.25	36.25	**36.31**	2.98
20	20.51	20.51	20.51	20.41	20.40	**20.47**	5.29
30	13.71	13.72	13.74	13.69	13.66	**13.71**	7.90
40	12.00	12.01	12.01	11.96	11.94	**11.98**	9.04
50	11.05	11.05	11.06	11.01	11.01	**11.04**	9.81
60	10.59	10.57	10.57	10.55	10.54	**10.56**	10.25
70	10.31	10.28	10.33	10.27	10.24	**10.28**	10.53
80	9.97	9.96	9.98	9.92	9.93	**9.95**	10.88
90	9.00	8.97	9.02	9.01	9.01	**9.00**	12.03
100	9.32	9.33	9.32	9.34	9.33	**9.33**	11.61
110	9.59	9.57	9.63	9.62	9.63	**9.61**	11.27
120	9.99	10.00	9.99	10.00	10.00	**10.00**	10.83
130	10.40	10.38	10.39	10.38	10.40	**10.39**	10.42
140	10.83	10.81	10.83	10.80	10.79	**10.81**	10.02
150	11.24	11.21	11.21	11.19	11.21	**11.21**	9.66
160	11.64	11.63	11.64	11.61	11.63	**11.63**	9.31
170	12.04	12.08	12.08	12.10	12.07	**12.07**	8.97
180	12.50	12.53	12.54	12.56	12.52	**12.53**	8.64
190	12.98	12.99	12.98	13.01	12.99	**12.99**	8.34
200	13.46	13.48	13.48	13.47	13.45	**13.47**	8.04
210	13.99	13.99	13.98	13.98	13.99	**13.99**	7.74
220	14.47	14.46	14.46	14.47	14.46	**14.47**	7.49
230	15.00	14.99	15.00	14.99	15.00	**15.00**	7.22
240	15.81	15.51	15.54	15.52	15.50	**15.58**	6.95

Table 6.20 Time and speedup of Chain(*N*) model with *N* = 18 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `compact`

`Compact` Heisenberg Chain(18) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	28.00	27.91	27.94	27.93	26.60	**26.60**	1.00
10	10.02	10.04	10.02	10.01	9.99	**10.02**	2.66
20	7.65	7.65	7.66	7.63	7.63	**7.64**	3.48
30	6.12	6.33	6.32	6.32	6.30	**6.28**	4.23
40	6.33	6.30	6.34	6.32	6.35	**6.33**	4.20
50	6.50	6.54	6.56	6.53	6.53	**6.53**	4.07
60	6.80	6.82	6.79	6.81	6.80	**6.80**	3.91
70	7.16	7.17	7.18	6.94	7.17	**7.13**	3.73
80	7.55	7.55	7.52	7.56	7.57	**7.55**	3.52
90	8.02	8.00	7.95	8.00	8.02	**8.00**	3.33
100	8.38	8.38	8.38	8.41	8.15	**8.34**	3.19
110	8.80	8.82	8.55	8.83	8.85	**8.77**	3.03
120	9.37	9.36	9.34	9.39	9.40	**9.37**	2.84
130	9.88	9.87	9.89	9.88	9.90	**9.88**	2.69
140	10.34	10.35	10.35	10.36	10.35	**10.35**	2.57
150	10.80	10.23	10.78	10.85	10.84	**10.70**	2.49
160	11.33	11.34	11.32	11.34	11.35	**11.34**	2.35
170	11.87	11.84	11.86	11.87	11.87	**11.86**	2.24
180	12.43	12.36	12.40	12.39	12.39	**12.39**	2.15
190	12.89	12.88	12.91	12.87	12.89	**12.89**	2.06
200	13.44	13.40	13.41	13.40	13.43	**13.42**	1.98
210	13.97	13.99	14.00	13.95	13.97	**13.98**	1.90
220	14.51	14.48	14.18	14.51	14.53	**14.44**	1.84
230	15.07	15.08	15.10	15.07	15.08	**15.08**	1.76
240	15.94	15.62	15.60	15.60	15.65	**15.68**	1.70

Table 6.21 Time and speedup of Chain(*N*) model with *N* = 16 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `compact`

`Compact` Heisenberg Chain(16) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	7.07	6.99	7.01	7.02	7.03	**7.02**	1.00
10	4.01	3.99	3.90	4.04	3.99	**3.98**	1.76
20	3.79	3.65	3.77	3.77	3.72	**3.74**	1.88
30	4.00	3.97	4.00	4.00	3.97	**3.99**	1.76
40	4.34	4.32	4.32	4.35	4.32	**4.33**	1.62
50	4.71	4.69	4.74	4.81	4.78	**4.75**	1.48
60	5.12	5.16	5.14	5.23	5.22	**5.17**	1.36
70	5.39	5.52	5.53	5.81	5.77	**5.61**	1.25
80	5.83	5.98	5.99	6.32	6.27	**6.08**	1.16
90	6.43	6.44	6.26	6.84	6.83	**6.56**	1.07
100	6.90	6.88	6.91	7.32	7.30	**7.06**	0.99
110	7.39	7.35	7.39	7.87	7.86	**7.57**	0.93
120	7.66	7.86	7.47	8.43	8.41	**7.97**	0.88
130	8.38	8.37	7.74	9.12	9.04	**8.53**	0.82
140	8.88	8.88	8.87	9.71	9.67	**9.20**	0.76
150	9.42	9.39	9.39	9.89	10.35	**9.69**	0.73
160	9.92	9.90	9.90	11.04	11.05	**10.36**	0.68
170	10.41	10.41	10.40	11.80	11.79	**10.96**	0.64
180	10.96	10.96	10.97	12.51	12.51	**11.58**	0.61
190	11.51	11.48	11.50	13.34	13.04	**12.17**	0.58
200	11.49	11.72	12.01	14.13	14.14	**12.70**	0.55
210	12.60	12.60	12.56	15.05	15.05	**13.57**	0.52
220	13.12	13.10	13.09	15.96	15.24	**14.10**	0.50
230	13.61	13.69	13.65	17.10	17.06	**15.02**	0.47
240	14.54	14.19	14.44	18.32	17.98	**15.90**	0.44

Table 6.22 Time and speedup of Chain(*N*) model with *N* = 26 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `balanced`

`Balanced` Heisenberg Chain(26) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	–	–	–	–	–	–	–
10	–	–	–	–	–	–	–
20	–	–	–	–	–	–	–
30	–	–	–	–	–	–	–
40	–	–	–	–	–	–	–
50	–	–	–	–	–	–	–
60	–	–	–	–	–	–	–
70	–	–	–	–	–	–	–
80	–	–	–	–	–	–	–
90	–	–	–	–	–	–	–
100	–	–	–	–	–	–	–
110	–	–	–	–	–	–	–
120	–	–	–	–	–	–	–
130	–	–	–	–	–	–	–
140	–	–	–	–	–	–	–
150	–	–	–	–	–	–	–
160	–	–	–	–	–	–	–
170	–	–	–	–	–	–	–
180	–	–	–	–	–	–	–
190	–	–	–	–	–	–	–
200	–	–	–	–	–	–	–
210	–	–	–	–	–	–	–
220	–	–	–	–	–	–	–
230	–	–	–	–	–	–	–
240	–	–	–	–	–	–	–

Table 6.23 Time and speedup of Chain(*N*) model with *N* = 24 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to balanced

`Balanced` Heisenberg Chain(24) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	1797.55	1798.52	1797.94	1797.59	1797.87	**1797.89**	1.00
10	574.70	445.37	377.58	408.04	387.16	**438.57**	4.10
20	302.79	238.96	254.07	259.27	256.06	**262.23**	6.86
30	207.42	190.01	183.26	178.95	180.32	**187.99**	9.56
40	159.36	144.71	140.65	140.03	141.76	**145.30**	12.37
50	129.12	109.01	112.21	114.96	112.64	**115.59**	15.55
60	111.44	94.13	95.71	97.17	97.84	**99.26**	18.11
70	97.07	87.01	85.61	87.73	88.77	**89.24**	20.15
80	85.05	75.88	78.55	81.46	75.97	**79.38**	22.65
90	78.51	75.99	75.46	69.21	70.14	**73.86**	24.34
100	71.80	70.11	70.04	70.79	66.69	**69.89**	25.73
110	66.85	65.09	64.30	64.77	65.13	**65.23**	27.56
120	62.89	61.15	60.63	61.38	61.28	**61.47**	29.25
130	59.37	58.98	59.04	59.89	59.55	**59.37**	30.28
140	56.33	55.92	56.53	56.90	57.22	**56.58**	31.78
150	53.54	53.95	52.81	54.06	55.04	**53.88**	33.37
160	50.35	50.75	51.49	51.71	51.89	**51.24**	35.09
170	50.67	49.92	51.32	51.13	51.04	**50.82**	35.38
180	46.54	48.32	47.98	47.95	47.93	**47.74**	37.66
190	44.33	45.66	47.68	47.66	47.36	**46.54**	38.63
200	43.25	44.73	45.13	44.61	45.01	**44.55**	40.36
210	42.07	43.57	43.30	43.86	42.66	**43.09**	41.72
220	41.72	43.14	43.48	43.17	43.06	**42.91**	41.90
230	40.00	39.86	40.07	40.11	39.88	**39.99**	44.96
240	39.11	39.12	39.04	39.20	39.01	**39.10**	45.99

Table 6.24 Time and speedup of Chain(*N*) model with *N* = 22 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `balanced`

`Balanced` Heisenberg Chain(22) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	433.11	434.21	434.14	434.11	434.01	**433.92**	1.00
10	124.48	121.27	131.55	127.16	119.93	**124.88**	3.47
20	68.84	75.78	78.13	76.67	73.68	**74.62**	5.82
30	54.59	53.60	55.13	54.42	53.51	**54.25**	8.00
40	41.42	39.47	41.24	40.88	40.26	**40.65**	10.67
50	34.28	33.11	33.34	33.75	33.61	**33.62**	12.91
60	29.46	28.81	29.55	29.78	29.41	**29.40**	14.76
70	26.44	27.75	27.55	26.56	26.40	**26.94**	16.11
80	21.04	24.07	23.12	23.30	23.15	**22.94**	18.92
90	21.69	23.27	21.65	21.88	21.70	**22.04**	19.69
100	20.04	21.56	20.94	19.23	19.90	**20.33**	21.34
110	18.58	19.86	19.32	19.42	19.50	**19.34**	22.44
120	18.34	18.48	18.00	18.28	18.46	**18.31**	23.70
130	17.82	17.84	17.55	17.59	17.75	**17.71**	24.50
140	16.63	16.72	16.81	16.79	16.89	**16.77**	25.88
150	63.99	63.97	63.89	63.66	63.57	**63.82**	6.80
160	70.50	70.35	70.98	70.86	69.90	**70.52**	6.15
170	79.78	80.28	80.09	79.96	79.65	**79.95**	5.43
180	91.59	90.93	90.74	91.24	91.67	**91.23**	4.76
190	91.01	91.31	89.98	90.08	90.64	**90.60**	4.79
200	113.74	113.06	112.10	112.97	112.55	**112.88**	3.84
210	122.83	122.59	122.96	122.95	122.36	**122.74**	3.54
220	115.38	115.92	115.48	115.85	115.30	**115.59**	3.75
230	129.02	128.81	129.01	129.39	129.74	**129.19**	3.36
240	141.84	140.37	141.19	140.81	140.96	**141.03**	3.08

Table 6.25 Time and speedup of Chain(*N*) model with *N* = 20 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `balanced`

`Balanced` Heisenberg Chain(20) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	108.24	108.40	108.28	108.27	108.29	**108.30**	1.00
10	7.81	7.81	7.83	7.98	7.86	**7.86**	13.78
20	6.74	6.73	6.73	6.91	6.86	**6.79**	15.94
30	5.40	5.40	5.28	5.46	5.37	**5.38**	20.13
40	6.44	6.45	6.44	6.53	6.41	**6.45**	16.79
50	8.06	8.12	8.04	8.12	8.08	**8.09**	13.39
60	10.34	10.35	10.14	10.44	10.40	**10.33**	10.48
70	15.64	15.61	15.67	15.83	15.72	**15.69**	6.90
80	16.90	16.90	16.84	17.00	17.32	**16.99**	6.37
90	18.53	18.80	18.54	18.46	18.60	**18.59**	5.83
100	20.20	19.91	19.99	20.27	19.55	**19.98**	5.42
110	22.20	22.23	22.19	22.31	22.23	**22.23**	4.87
120	23.48	23.47	22.91	23.76	23.65	**23.45**	4.62
130	25.94	26.12	26.14	25.92	26.07	**26.04**	4.16
140	28.33	27.94	28.24	28.07	28.09	**28.14**	3.85
150	29.86	29.43	30.04	29.86	29.64	**29.77**	3.64
160	32.00	31.94	31.96	31.23	31.76	**31.78**	3.41
170	33.57	33.56	33.59	33.81	34.01	**33.71**	3.21
180	35.60	35.61	35.74	35.62	35.34	**35.58**	3.04
190	36.16	36.07	36.26	36.05	35.33	**35.97**	3.01
200	37.96	38.21	38.02	37.81	38.22	**38.04**	2.85
210	40.27	40.53	40.44	40.25	40.41	**40.38**	2.68
220	43.28	43.09	42.61	42.69	43.10	**42.95**	2.52
230	46.24	45.78	45.67	45.85	46.00	**45.91**	2.36
240	48.43	46.76	47.66	47.64	48.23	**47.75**	2.27

Table 6.26 Time and speedup of Chain(*N*) model with *N* = 18 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `balanced`

`Balanced` Heisenberg Chain(18) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	28.00	27.91	27.94	27.93	26.60	**27.67**	1.00
10	7.81	7.81	7.83	7.98	7.86	**7.86**	3.52
20	6.74	6.73	6.73	6.91	6.86	**6.79**	4.07
30	5.40	5.40	5.28	5.46	5.37	**5.38**	5.14
40	6.44	6.45	6.44	6.53	6.41	**6.45**	4.29
50	8.06	8.12	8.04	8.12	8.08	**8.09**	3.42
60	10.34	10.35	10.14	10.44	10.40	**10.33**	2.68
70	15.64	15.61	15.67	15.83	15.72	**15.69**	1.76
80	16.90	16.90	16.84	17.00	17.32	**16.99**	1.63
90	18.53	18.80	18.54	18.46	18.60	**18.59**	1.49
100	20.20	19.91	19.99	20.27	19.55	**19.98**	1.38
110	22.20	22.23	22.19	22.31	22.23	**22.23**	1.24
120	23.48	23.47	22.91	23.76	23.65	**23.45**	1.18
130	25.94	26.12	26.14	25.92	26.07	**26.04**	1.06
140	28.33	27.94	28.24	28.07	28.09	**28.14**	0.98
150	29.86	29.43	30.04	29.86	29.64	**29.77**	0.93
160	32.00	31.94	31.96	31.23	31.76	**31.78**	0.87
170	33.57	33.56	33.59	33.81	34.01	**33.71**	0.82
180	35.60	35.61	35.74	35.62	35.34	**35.58**	0.78
190	36.16	36.07	36.26	36.05	35.33	**35.97**	0.77
200	37.96	38.21	38.02	37.81	38.22	**38.04**	0.73
210	40.27	40.53	40.44	40.25	40.41	**40.38**	0.69
220	43.28	43.09	42.61	42.69	43.10	**42.95**	0.64
230	46.24	45.78	45.67	45.85	46.00	**45.91**	0.60
240	48.43	46.76	47.66	47.64	48.23	**47.75**	0.58

Table 6.27 Time and speedup of Chain(*N*) model with *N* = 16 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `balanced`

`Balanced` Heisenberg Chain(16) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	7.07	6.99	7.01	7.02	7.03	**7.02**	1.00
10	3.22	3.28	3.23	3.26	3.23	**3.24**	2.17
20	3.44	3.50	3.46	3.48	3.54	**3.48**	2.02
30	4.27	4.05	4.27	4.29	4.30	**4.24**	1.66
40	5.45	5.46	5.48	5.51	5.54	**5.49**	1.28
50	7.08	7.11	7.15	7.13	7.20	**7.13**	0.98
60	9.31	9.35	9.16	9.33	9.32	**9.29**	0.76
70	11.60	11.29	11.66	11.65	11.65	**11.57**	0.61
80	12.76	12.83	12.85	12.76	12.48	**12.73**	0.55
90	14.07	14.09	14.21	14.31	13.83	**14.10**	0.50
100	15.07	15.58	15.58	15.14	15.55	**15.38**	0.46
110	17.11	17.27	17.26	17.28	17.18	**17.22**	0.41
120	18.51	18.71	18.54	18.64	18.53	**18.59**	0.38
130	20.19	20.31	20.30	20.38	20.24	**20.28**	0.35
140	22.01	21.51	22.03	22.06	22.07	**21.94**	0.32
150	23.58	23.79	23.64	23.93	23.73	**23.73**	0.30
160	25.97	25.84	25.71	25.91	25.90	**25.87**	0.27
170	27.83	27.59	27.53	27.63	27.00	**27.52**	0.26
180	29.43	29.56	29.52	29.62	29.60	**29.55**	0.24
190	30.58	30.53	30.51	29.91	30.44	**30.39**	0.23
200	31.69	31.64	32.56	32.52	31.09	**31.90**	0.22
210	34.74	34.85	34.84	34.78	34.87	**34.81**	0.20
220	36.91	36.92	36.85	36.93	37.04	**36.93**	0.19
230	39.03	38.24	37.36	38.57	37.52	**38.14**	0.18
240	41.14	38.31	39.89	40.90	40.84	**40.22**	0.17

6.2.2 Heisenberg Ladder on Xeon Phi

Table 6.28 Time and speedup of Ladder2(*N*) model with *N* = 26 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `scatter`

`Scatter` Heisenberg Ladder2(26) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	–	–	–	–	–	–	–
10	–	–	–	–	–	–	–
20	–	–	–	–	–	–	–
30	–	–	–	–	–	–	–
40	–	–	–	–	–	–	–
50	–	–	–	–	–	–	–
60	–	–	–	–	–	–	–
70	–	–	–	–	–	–	–
80	–	–	–	–	–	–	–
90	–	–	–	–	–	–	–
100	–	–	–	–	–	–	–
110	–	–	–	–	–	–	–
120	–	–	–	–	–	–	–
130	–	–	–	–	–	–	–
140	–	–	–	–	–	–	–
150	–	–	–	–	–	–	–
160	–	–	–	–	–	–	–
170	–	–	–	–	–	–	–
180	–	–	–	–	–	–	–
190	–	–	–	–	–	–	–
200	–	–	–	–	–	–	–
210	–	–	–	–	–	–	–
220	–	–	–	–	–	–	–
230	–	–	–	–	–	–	–
240	–	–	–	–	–	–	–

Table 6.29 Time and speedup of Ladder2(*N*) model with *N* = 24 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `scatter`

`Scatter` Heisenberg Ladder2(24) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	2352.88	2354.23	2354.63	2355.27	2354.12	**2354.22**	1.00
10	554.40	579.01	542.70	553.20	537.86	**553.43**	4.25
20	322.49	317.95	299.73	317.82	326.01	**316.80**	7.43
30	246.32	239.08	229.18	235.76	255.04	**241.08**	9.77
40	185.69	175.71	182.13	174.23	181.62	**179.88**	13.09
50	152.52	157.67	146.06	152.02	151.11	**151.88**	15.50
60	130.45	130.59	129.63	132.26	128.77	**130.34**	18.06
70	117.04	118.71	115.49	115.66	114.41	**116.26**	20.25
80	100.40	103.29	101.93	101.65	107.20	**102.89**	22.88
90	90.80	91.14	93.64	98.43	93.78	**93.56**	25.16
100	90.77	85.09	89.40	90.78	87.89	**88.78**	26.52
110	82.38	82.19	80.04	83.35	85.44	**82.68**	28.47
120	76.77	77.19	78.30	77.85	79.82	**77.99**	30.19
130	72.68	73.57	75.77	74.29	73.87	**74.04**	31.80
140	69.30	69.72	70.44	69.04	69.69	**69.64**	33.81
150	66.61	70.65	67.66	65.16	66.21	**67.26**	35.00
160	62.66	67.48	64.46	68.40	65.31	**65.66**	35.86
170	65.80	64.93	64.54	64.79	68.89	**65.79**	35.78
180	64.00	61.06	60.90	61.11	63.32	**62.07**	37.93
190	61.38	59.62	62.10	62.43	63.34	**61.78**	38.11
200	59.89	58.08	57.75	58.61	58.36	**58.54**	40.22
210	56.13	56.55	55.10	55.25	56.03	**55.81**	42.18
220	54.24	57.53	53.36	53.27	57.27	**55.13**	42.70
230	50.48	51.12	50.55	50.40	50.52	**50.61**	46.51
240	49.70	49.89	49.83	49.66	49.85	**49.78**	47.29

Table 6.30 Time and speedup of Ladder2(*N*) model with *N* = 22 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `scatter`

`Scatter` Heisenberg Ladder2(22) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	534.20	539.41	539.06	540.01	539.19	**538.37**	1.00
10	127.81	135.78	135.87	139.20	125.51	**132.83**	4.05
20	79.92	77.81	77.73	72.80	78.24	**77.30**	6.96
30	55.09	49.34	54.87	49.48	55.10	**52.78**	10.20
40	41.26	40.86	40.57	37.98	40.91	**40.32**	13.35
50	33.77	33.78	34.39	33.40	33.57	**33.78**	15.94
60	29.44	29.14	29.38	29.59	28.69	**29.25**	18.41
70	26.23	26.72	26.06	26.50	26.37	**26.38**	20.41
80	22.86	23.33	22.91	23.24	23.73	**23.21**	23.19
90	23.84	23.19	22.53	21.70	21.61	**22.58**	23.85
100	21.86	20.57	20.25	20.37	19.83	**20.58**	26.16
110	19.69	19.06	18.45	19.15	18.45	**18.96**	28.40
120	18.06	17.96	18.29	18.02	18.28	**18.12**	29.71
130	18.19	17.33	17.61	17.46	17.59	**17.64**	30.52
140	17.26	16.71	16.70	17.23	16.69	**16.92**	31.83
150	19.86	19.02	18.98	19.53	18.90	**19.26**	27.96
160	19.00	19.27	18.78	19.31	18.73	**19.02**	28.31
170	19.27	19.34	18.87	19.26	18.87	**19.12**	28.15
180	19.28	19.43	19.27	19.30	19.50	**19.36**	27.82
190	19.56	19.51	20.02	19.55	19.65	**19.66**	27.39
200	19.89	19.93	20.20	19.86	19.95	**19.97**	26.97
210	20.07	20.09	20.41	20.07	20.13	**20.16**	26.71
220	20.39	20.43	20.73	20.41	20.39	**20.47**	26.30
230	20.68	20.65	20.71	20.67	20.69	**20.68**	26.03
240	21.31	21.07	21.11	21.03	21.03	**21.11**	25.50

Table 6.31 Time and speedup of Ladder2(*N*) model with *N* = 20 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `scatter`

`Scatter` Heisenberg Ladder2(20) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	133.70	133.85	133.96	134.29	134.28	**134.02**	1.00
10	29.42	32.05	30.37	28.52	30.84	**30.24**	4.43
20	18.72	17.56	17.25	17.17	17.17	**17.58**	7.63
30	13.22	13.14	12.94	13.66	12.87	**13.17**	10.18
40	11.85	11.51	11.80	11.61	11.76	**11.71**	11.45
50	10.76	10.76	10.74	10.43	10.63	**10.66**	12.57
60	10.05	10.13	10.09	10.05	10.02	**10.06**	13.32
70	9.90	9.85	10.03	9.83	9.81	**9.88**	13.56
80	9.64	9.62	9.82	9.55	9.58	**9.64**	13.90
90	9.92	9.92	10.05	9.85	9.86	**9.92**	13.51
100	10.04	10.03	10.12	9.96	10.09	**10.05**	13.34
110	10.30	10.24	10.29	10.21	10.25	**10.26**	13.06
120	10.62	10.63	10.60	10.56	10.55	**10.59**	12.65
130	9.66	9.68	9.72	9.80	9.76	**9.73**	13.78
140	10.05	10.14	10.32	10.05	10.07	**10.13**	13.24
150	10.68	10.59	10.65	10.52	10.53	**10.59**	12.65
160	11.13	10.98	11.16	10.96	11.12	**11.07**	12.11
170	11.63	11.48	11.67	11.54	11.60	**11.58**	11.57
180	11.92	11.81	11.82	11.92	11.87	**11.87**	11.29
190	12.53	12.57	12.55	12.53	12.56	**12.55**	10.68
200	13.03	13.09	13.08	13.06	13.05	**13.06**	10.26
210	13.42	13.46	13.43	13.45	13.43	**13.44**	9.97
220	13.83	13.87	13.86	13.84	13.83	**13.84**	9.68
230	14.35	14.38	14.40	14.38	14.32	**14.37**	9.33
240	15.08	14.72	14.79	14.78	14.69	**14.81**	9.05

Table 6.32 Time and speedup of Ladder2(*N*) model with *N* = 18 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `scatter`

`Scatter` Heisenberg Ladder2(18) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	32.89	32.85	33.03	33.01	32.93	**32.94**	1.00
10	8.12	8.83	8.09	8.02	8.00	**8.21**	4.01
20	6.23	6.38	6.18	6.41	6.37	**6.31**	5.22
30	4.38	4.38	4.34	4.58	4.43	**4.42**	7.45
40	4.75	4.71	4.72	5.02	4.75	**4.79**	6.88
50	5.39	5.40	5.34	5.78	5.42	**5.47**	6.03
60	6.40	6.34	6.33	6.79	6.36	**6.44**	5.11
70	7.21	7.22	7.16	7.32	7.24	**7.23**	4.56
80	7.89	7.88	7.84	7.90	7.90	**7.88**	4.18
90	8.57	8.62	8.54	8.59	8.61	**8.59**	3.84
100	9.31	9.32	9.30	9.36	9.36	**9.33**	3.53
110	10.07	10.07	10.01	10.09	10.10	**10.07**	3.27
120	10.89	10.87	10.83	10.87	10.89	**10.87**	3.03
130	11.89	11.88	11.80	11.89	11.87	**11.87**	2.78
140	12.64	12.65	12.58	12.66	12.66	**12.64**	2.61
150	13.34	13.34	13.29	13.39	13.35	**13.34**	2.47
160	14.10	14.10	14.09	14.11	14.11	**14.10**	2.34
170	14.87	14.85	14.82	14.87	14.88	**14.86**	2.22
180	15.59	15.57	15.50	15.59	15.65	**15.58**	2.11
190	16.56	16.54	16.49	16.56	16.56	**16.55**	1.99
200	17.33	17.32	17.25	17.30	17.32	**17.30**	1.90
210	18.05	18.04	17.98	18.04	18.05	**18.03**	1.83
220	18.78	18.79	18.74	18.79	18.79	**18.78**	1.75
230	19.53	19.53	19.47	19.52	19.53	**19.52**	1.69
240	20.65	20.29	20.22	20.29	20.30	**20.35**	1.62

Table 6.33 Time and speedup of Ladder2(*N*) model with *N* = 16 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `scatter`

`Scatter` Heisenberg Ladder2(16) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	8.28	8.20	8.22	8.24	8.25	**8.24**	1.00
10	3.11	3.10	3.10	3.11	3.13	**3.11**	2.65
20	3.10	3.10	3.10	3.11	3.11	**3.10**	2.65
30	3.44	3.43	3.44	3.45	3.44	**3.44**	2.40
40	3.98	3.99	3.99	3.99	4.00	**3.99**	2.06
50	4.77	4.78	4.77	4.80	4.85	**4.79**	1.72
60	5.74	5.73	5.74	5.75	5.76	**5.74**	1.43
70	6.44	6.48	6.46	6.48	6.46	**6.46**	1.28
80	7.14	7.14	7.17	7.14	7.16	**7.15**	1.15
90	7.91	7.92	7.92	7.93	7.95	**7.93**	1.04
100	8.66	8.66	8.67	8.67	8.68	**8.67**	0.95
110	9.38	9.37	9.38	9.39	9.41	**9.39**	0.88
120	10.07	10.07	10.09	10.09	10.10	**10.08**	0.82
130	10.92	10.91	10.92	10.94	10.94	**10.92**	0.75
140	11.68	11.68	11.70	11.70	11.70	**11.69**	0.70
150	12.38	12.38	12.37	12.39	12.39	**12.38**	0.67
160	13.05	13.05	13.06	13.07	13.07	**13.06**	0.63
170	13.76	13.76	13.78	13.78	13.78	**13.77**	0.60
180	14.44	14.44	14.44	14.45	14.46	**14.45**	0.57
190	15.34	15.31	15.32	15.35	15.34	**15.33**	0.54
200	16.04	16.03	16.04	16.04	16.05	**16.04**	0.51
210	16.74	16.72	16.73	16.71	16.73	**16.72**	0.49
220	17.45	17.43	17.45	17.43	17.45	**17.44**	0.47
230	18.14	18.12	18.12	18.12	18.13	**18.13**	0.45
240	19.16	18.81	18.81	18.81	18.85	**18.89**	0.44

Table 6.34 Time and speedup of Ladder2(*N*) model with *N* = 26 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `compact`

`Compact` Heisenberg Ladder2(26) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	–	–	–	–	–	–	–
10	–	–	–	–	–	–	–
20	–	–	–	–	–	–	–
30	–	–	–	–	–	–	–
40	–	–	–	–	–	–	–
50	–	–	–	–	–	–	–
60	–	–	–	–	–	–	–
70	–	–	–	–	–	–	–
80	–	–	–	–	–	–	–
90	–	–	–	–	–	–	–
100	–	–	–	–	–	–	–
110	–	–	–	–	–	–	–
120	–	–	–	–	–	–	–
130	–	–	–	–	–	–	–
140	–	–	–	–	–	–	–
150	–	–	–	–	–	–	–
160	–	–	–	–	–	–	–
170	–	–	–	–	–	–	–
180	–	–	–	–	–	–	–
190	–	–	–	–	–	–	–
200	–	–	–	–	–	–	–
210	–	–	–	–	–	–	–
220	–	–	–	–	–	–	–
230	–	–	–	–	–	–	–
240	–	–	–	–	–	–	–

Table 6.35 Time and speedup of Ladder2(*N*) model with *N* = 24 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `compact`

`Compact` Heisenberg Ladder2(24) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	2352.88	2354.23	2354.63	2355.27	2354.12	**2354.22**	1.00
10	698.26	697.62	698.07	700.38	697.64	**698.39**	3.37
20	373.91	373.94	373.94	373.97	374.04	**373.96**	6.30
30	254.73	254.64	254.70	254.66	254.75	**254.70**	9.24
40	196.22	196.15	196.22	196.18	196.28	**196.21**	12.00
50	159.29	159.26	159.92	159.24	159.28	**159.40**	14.77
60	136.88	136.83	136.87	136.88	136.92	**136.87**	17.20
70	118.42	118.34	118.39	118.41	118.40	**118.39**	19.88
80	104.34	104.47	104.38	104.37	104.37	**104.39**	22.55
90	95.73	95.73	95.73	95.74	95.72	**95.73**	24.59
100	87.98	87.96	87.94	87.94	87.95	**87.95**	26.77
110	81.42	81.46	81.45	81.43	81.48	**81.45**	28.91
120	76.40	76.39	76.38	76.41	76.39	**76.40**	30.82
130	72.02	72.02	72.05	72.13	72.04	**72.05**	32.67
140	67.88	67.89	67.87	67.86	67.87	**67.87**	34.69
150	64.23	64.19	64.21	64.24	64.25	**64.22**	36.66
160	60.64	60.62	60.60	60.64	60.63	**60.63**	38.83
170	60.39	60.31	60.48	60.29	60.30	**60.35**	39.01
180	56.07	56.01	56.04	56.05	56.05	**56.04**	42.01
190	53.34	53.39	53.40	53.39	53.38	**53.38**	44.10
200	51.88	51.91	51.91	51.90	51.92	**51.90**	45.36
210	50.43	50.46	50.45	50.42	50.44	**50.44**	46.68
220	49.55	49.58	49.57	49.57	49.55	**49.56**	47.50
230	47.69	47.72	47.69	47.67	47.66	**47.69**	49.37
240	46.36	46.31	46.26	46.29	46.22	**46.29**	50.86

Table 6.36 Time and speedup of Ladder2(*N*) model with *N* = 22 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `compact`

`Compact` Heisenberg Ladder2(22) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	534.20	539.41	539.06	540.01	539.19	**538.37**	1.00
10	165.25	165.26	165.31	165.28	165.32	**165.29**	3.26
20	91.37	91.39	91.42	91.37	91.38	**91.38**	5.89
30	63.76	63.80	63.78	63.80	63.81	**63.79**	8.44
40	49.35	49.37	49.36	49.38	49.36	**49.37**	10.91
50	41.46	41.46	41.47	41.43	41.43	**41.45**	12.99
60	35.26	35.28	35.27	35.30	35.26	**35.27**	15.26
70	30.83	30.84	30.86	30.83	30.83	**30.84**	17.46
80	26.81	26.80	26.80	26.81	26.77	**26.80**	20.09
90	24.41	24.40	24.39	24.38	24.39	**24.40**	22.07
100	22.00	22.04	21.99	22.00	21.97	**22.00**	24.47
110	20.04	20.01	20.07	20.03	20.03	**20.04**	26.87
120	18.65	18.65	18.64	18.64	18.62	**18.64**	28.88
130	17.74	17.71	17.72	17.71	17.69	**17.71**	30.39
140	16.60	16.56	16.57	16.57	16.56	**16.57**	32.49
150	18.96	18.92	18.92	18.93	18.92	**18.93**	28.44
160	18.70	18.64	18.67	18.66	18.66	**18.66**	28.84
170	18.95	18.93	18.92	18.93	18.92	**18.93**	28.44
180	19.02	19.00	19.01	19.01	19.01	**19.01**	28.32
190	19.16	19.09	19.11	19.11	19.07	**19.11**	28.18
200	19.65	19.64	19.66	19.63	19.58	**19.63**	27.42
210	20.15	20.13	20.15	20.13	20.13	**20.14**	26.73
220	20.81	20.76	20.76	20.82	20.79	**20.79**	25.90
230	21.52	21.48	21.46	21.51	21.45	**21.48**	25.06
240	22.46	22.18	22.17	22.20	22.22	**22.25**	24.20

Table 6.37 Time and speedup of Ladder2(*N*) model with *N* = 20 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `compact`

`Compact` Heisenberg Ladder2(20) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	133.70	133.85	133.96	134.29	134.28	**134.02**	1.00
10	42.97	43.35	43.00	42.97	42.97	**43.05**	3.11
20	24.23	24.21	24.21	24.06	24.07	**24.16**	5.55
30	16.26	16.22	16.22	16.21	16.21	**16.22**	8.26
40	13.83	13.81	13.83	13.77	13.77	**13.80**	9.71
50	12.60	12.59	12.62	12.58	12.57	**12.59**	10.64
60	11.82	11.82	11.81	11.77	11.78	**11.80**	11.35
70	11.37	11.34	11.36	11.36	11.33	**11.35**	11.81
80	10.97	10.97	10.97	10.94	10.94	**10.96**	12.23
90	11.12	11.12	11.11	11.09	11.12	**11.11**	12.06
100	11.29	11.26	11.26	11.25	11.26	**11.26**	11.90
110	11.50	11.46	11.45	11.47	11.46	**11.47**	11.69
120	11.66	11.68	11.66	11.66	11.64	**11.66**	11.50
130	10.97	11.00	11.00	11.01	11.01	**11.00**	12.19
140	11.45	11.42	11.42	11.45	11.43	**11.44**	11.72
150	11.75	11.78	11.80	11.80	11.81	**11.79**	11.37
160	12.22	12.20	12.24	12.24	12.22	**12.22**	10.96
170	12.61	12.66	12.65	12.67	12.65	**12.65**	10.60
180	13.11	13.06	13.11	13.16	13.15	**13.12**	10.22
190	13.58	13.55	13.52	13.56	13.60	**13.56**	9.88
200	14.08	14.03	14.02	14.03	14.02	**14.04**	9.55
210	14.60	14.55	14.54	14.55	14.57	**14.56**	9.20
220	15.10	15.06	15.03	15.03	15.10	**15.06**	8.90
230	15.65	15.59	15.58	15.58	15.62	**15.61**	8.59
240	16.45	16.06	16.03	16.06	16.10	**16.14**	8.30

Table 6.38 Time and speedup of Ladder2(*N*) model with *N* = 18 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `compact`

`Compact` Heisenberg Ladder2(18) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	32.89	32.85	33.03	33.01	32.93	**32.94**	1.00
10	11.31	11.27	11.29	11.25	11.24	**11.27**	2.92
20	8.35	8.29	8.29	8.26	8.24	**8.29**	3.98
30	6.88	6.87	6.89	6.78	6.78	**6.84**	4.82
40	6.75	6.71	6.73	6.69	6.74	**6.72**	4.90
50	6.99	6.96	6.97	6.94	6.94	**6.96**	4.73
60	7.19	7.18	7.20	7.16	7.19	**7.18**	4.59
70	7.67	7.61	7.65	7.61	7.61	**7.63**	4.32
80	8.05	8.05	8.04	8.02	8.01	**8.03**	4.10
90	8.62	8.59	8.62	8.60	8.60	**8.61**	3.83
100	9.14	9.12	9.14	9.11	9.10	**9.12**	3.61
110	9.80	9.75	9.79	9.73	9.73	**9.76**	3.38
120	10.47	10.42	10.42	10.43	10.41	**10.43**	3.16
130	11.14	11.10	11.12	11.09	11.09	**11.11**	2.97
140	11.77	11.73	11.77	11.77	11.78	**11.76**	2.80
150	12.57	12.51	12.53	12.55	12.53	**12.54**	2.63
160	13.32	13.26	13.32	13.29	13.30	**13.30**	2.48
170	14.27	14.21	14.27	14.27	14.26	**14.25**	2.31
180	15.25	15.14	15.18	15.20	15.21	**15.20**	2.17
190	16.29	16.15	16.24	16.20	16.22	**16.22**	2.03
200	17.32	17.17	17.23	17.31	17.28	**17.26**	1.91
210	18.54	18.40	18.38	18.50	18.50	**18.46**	1.78
220	19.64	19.55	19.51	19.60	19.59	**19.58**	1.68
230	20.95	20.83	20.82	20.92	20.94	**20.89**	1.58
240	22.29	21.82	21.83	21.95	21.96	**21.97**	1.50

Table 6.39 Time and speedup of Ladder2(*N*) model with *N* = 16 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `compact`

`Compact` Heisenberg Ladder2(16) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	8.28	8.20	8.22	8.24	8.25	**8.24**	1.00
10	4.25	4.25	4.27	4.28	4.29	**4.27**	1.93
20	3.98	3.98	3.96	3.98	3.98	**3.98**	2.07
30	4.15	4.13	4.15	4.18	4.17	**4.16**	1.98
40	4.50	4.47	4.48	4.51	4.51	**4.49**	1.83
50	4.98	4.99	4.98	4.99	5.01	**4.99**	1.65
60	5.40	5.39	5.41	5.44	5.41	**5.41**	1.52
70	5.90	5.90	5.93	5.91	5.91	**5.91**	1.39
80	6.42	6.42	6.41	6.46	6.41	**6.42**	1.28
90	6.95	6.96	6.97	6.97	6.98	**6.96**	1.18
100	7.49	7.47	7.50	7.50	7.50	**7.49**	1.10
110	8.11	8.09	8.12	8.12	8.14	**8.12**	1.02
120	8.80	8.78	8.80	8.80	8.79	**8.79**	0.94
130	9.48	9.47	9.51	9.48	9.50	**9.49**	0.87
140	10.17	10.16	10.14	10.17	10.18	**10.16**	0.81
150	10.93	10.93	10.93	10.96	10.96	**10.94**	0.75
160	11.72	11.68	11.69	11.70	11.71	**11.70**	0.70
170	12.56	12.58	12.58	12.59	12.59	**12.58**	0.66
180	13.40	13.40	13.39	13.42	13.43	**13.41**	0.61
190	14.45	14.43	14.44	14.44	14.45	**14.44**	0.57
200	15.37	15.32	15.32	15.31	15.34	**15.33**	0.54
210	16.61	16.57	16.58	16.56	16.59	**16.58**	0.50
220	17.69	17.66	17.67	17.70	17.68	**17.68**	0.47
230	18.98	18.95	18.91	18.95	18.95	**18.95**	0.43
240	20.36	20.03	20.01	20.02	20.03	**20.09**	0.41

Table 6.40 Time and speedup of Ladder2(*N*) model with *N* = 26 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `balanced`

`Balanced` Heisenberg Ladder2(26) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	–	–	–	–	–	–	–
10	–	–	–	–	–	–	–
20	–	–	–	–	–	–	–
30	–	–	–	–	–	–	–
40	–	–	–	–	–	–	–
50	–	–	–	–	–	–	–
60	–	–	–	–	–	–	–
70	–	–	–	–	–	–	–
80	–	–	–	–	–	–	–
90	–	–	–	–	–	–	–
100	–	–	–	–	–	–	–
110	–	–	–	–	–	–	–
120	–	–	–	–	–	–	–
130	–	–	–	–	–	–	–
140	–	–	–	–	–	–	–
150	–	–	–	–	–	–	–
160	–	–	–	–	–	–	–
170	–	–	–	–	–	–	–
180	–	–	–	–	–	–	–
190	–	–	–	–	–	–	–
200	–	–	–	–	–	–	–
210	–	–	–	–	–	–	–
220	–	–	–	–	–	–	–
230	–	–	–	–	–	–	–
240	–	–	–	–	–	–	–

Table 6.41 Time and speedup of Ladder2(*N*) model with *N* = 24 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `balanced`

`Balanced` Heisenberg Ladder2(24) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	2352.88	2354.23	2354.63	2355.27	2354.12	**2354.22**	1.00
10	570.37	531.96	529.47	565.59	574.36	**554.35**	4.25
20	352.77	323.24	333.92	344.63	304.25	**331.76**	7.10
30	230.39	216.88	221.68	233.30	228.63	**226.18**	10.41
40	190.24	185.76	182.14	184.46	184.36	**185.39**	12.70
50	160.14	151.48	152.97	151.61	150.93	**153.42**	15.34
60	134.38	129.75	128.54	129.22	129.69	**130.32**	18.07
70	115.42	120.86	116.21	117.65	116.32	**117.29**	20.07
80	106.82	102.17	101.95	104.03	99.90	**102.97**	22.86
90	93.92	92.17	90.67	96.93	94.80	**93.70**	25.12
100	85.45	83.18	87.40	84.60	85.96	**85.32**	27.59
110	79.51	76.49	80.94	83.73	80.23	**80.18**	29.36
120	75.10	77.33	74.66	74.36	75.01	**75.29**	31.27
130	72.89	73.69	72.82	71.64	72.35	**72.67**	32.39
140	68.46	69.39	66.88	67.17	68.52	**68.08**	34.58
150	67.02	65.78	66.50	66.86	65.98	**66.43**	35.44
160	62.68	61.53	62.15	61.51	63.89	**62.35**	37.76
170	62.09	60.91	60.94	61.50	63.14	**61.71**	38.15
180	58.39	57.59	57.45	57.67	57.69	**57.76**	40.76
190	57.24	57.20	57.08	57.08	57.20	**57.16**	41.19
200	54.97	54.78	55.01	54.78	54.27	**54.76**	42.99
210	51.82	52.03	52.50	52.35	51.95	**52.13**	45.16
220	51.17	51.74	51.94	51.92	51.92	**51.74**	45.51
230	47.61	47.60	47.58	47.58	47.65	**47.60**	49.46
240	46.34	46.31	46.24	46.28	46.26	**46.29**	50.86

Table 6.42 Time and speedup of Ladder2(*N*) model with *N* = 22 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `balanced`

`Balanced` Heisenberg Ladder2(22) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	534.20	539.41	539.06	540.01	539.19	**538.37**	1.00
10	130.34	141.66	141.54	133.68	120.60	**133.56**	4.03
20	78.49	72.05	72.49	78.92	69.75	**74.34**	7.24
30	54.29	50.96	54.91	48.22	48.23	**51.32**	10.49
40	43.25	39.89	40.50	41.55	40.26	**41.09**	13.10
50	33.79	33.28	33.20	33.61	32.96	**33.37**	16.13
60	29.20	28.58	29.11	28.65	28.54	**28.82**	18.68
70	26.11	26.32	26.49	26.10	27.22	**26.45**	20.36
80	23.12	23.18	23.16	23.03	24.16	**23.33**	23.08
90	22.40	21.56	21.53	23.57	21.63	**22.14**	24.32
100	20.66	20.68	20.27	19.89	19.85	**20.27**	26.56
110	18.45	19.37	19.44	19.37	18.49	**19.02**	28.30
120	18.36	18.50	18.34	18.22	18.24	**18.33**	29.37
130	17.90	17.87	17.59	17.90	17.94	**17.84**	30.18
140	16.94	16.63	16.54	16.71	17.20	**16.80**	32.04
150	63.69	63.25	63.93	63.86	63.79	**63.70**	8.45
160	70.80	70.72	70.69	70.96	70.36	**70.71**	7.61
170	80.37	80.53	80.21	79.86	79.61	**80.12**	6.72
180	91.94	91.29	91.90	92.08	90.96	**91.63**	5.88
190	90.70	90.60	90.29	90.98	90.79	**90.67**	5.94
200	114.24	113.82	114.13	113.57	114.10	**113.97**	4.72
210	123.48	123.83	124.46	123.50	123.35	**123.72**	4.35
220	115.42	116.56	116.27	116.94	116.01	**116.24**	4.63
230	47.61	47.60	47.58	47.58	47.65	**47.60**	49.46
240	46.34	46.31	46.24	46.28	46.26	**46.29**	50.86

Table 6.43 Time and speedup of Ladder2(*N*) model with *N* = 20 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `balanced`

`Balanced` Heisenberg Ladder2(20) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	133.70	133.85	133.96	134.29	134.28	**134.02**	1.00
10	28.71	29.31	29.58	30.70	30.27	**29.71**	4.51
20	17.21	17.64	17.97	18.74	18.45	**18.00**	7.45
30	12.94	13.11	13.23	13.73	13.58	**13.32**	10.06
40	15.45	15.41	15.34	15.08	15.40	**15.34**	8.74
50	15.20	15.29	15.09	15.23	15.26	**15.21**	8.81
60	15.78	15.81	15.83	15.83	15.81	**15.81**	8.48
70	28.62	28.48	28.56	28.75	28.76	**28.63**	4.68
80	32.61	32.98	32.74	33.16	32.45	**32.79**	4.09
90	41.97	42.60	41.95	42.19	41.75	**42.09**	3.18
100	48.82	48.56	49.13	49.07	49.11	**48.94**	2.74
110	49.98	49.20	50.08	49.52	49.58	**49.67**	2.70
120	50.43	50.58	50.35	50.25	50.30	**50.38**	2.66
130	56.80	56.92	56.39	56.06	56.54	**56.54**	2.37
140	49.36	49.12	49.30	48.86	49.42	**49.21**	2.72
150	57.91	58.27	57.92	58.41	58.55	**58.21**	2.30
160	66.40	66.00	65.59	66.06	66.38	**66.08**	2.03
170	67.88	67.66	66.98	67.83	67.52	**67.57**	1.98
180	68.78	69.21	68.62	68.75	68.62	**68.80**	1.95
190	64.16	64.64	64.48	64.32	64.25	**64.37**	2.08
200	64.55	64.49	65.00	65.12	64.58	**64.75**	2.07
210	72.44	72.47	73.15	72.57	72.43	**72.61**	1.85
220	79.47	79.55	79.31	79.55	79.61	**79.50**	1.69
230	81.54	82.35	81.55	81.97	81.54	**81.79**	1.64
240	83.52	83.44	83.83	83.04	83.88	**83.54**	1.60

Table 6.44 Time and speedup of Ladder2(*N*) model with *N* = 18 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `balanced`

`Balanced` Heisenberg Ladder2(18) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	32.89	32.85	33.03	33.01	32.93	**32.94**	1.00
10	9.03	9.11	9.50	8.71	9.29	**9.13**	3.61
20	7.21	7.21	7.48	7.30	7.21	**7.28**	4.52
30	5.49	5.42	5.53	5.52	5.39	**5.47**	6.02
40	6.27	6.12	6.22	6.24	6.07	**6.19**	5.32
50	7.15	7.31	7.25	7.27	7.29	**7.25**	4.54
60	8.79	8.83	8.89	8.78	8.88	**8.83**	3.73
70	13.56	13.84	13.63	13.69	13.73	**13.69**	2.41
80	14.57	14.84	14.75	14.85	14.94	**14.79**	2.23
90	15.79	15.99	15.98	15.95	15.99	**15.94**	2.07
100	16.93	17.05	17.17	16.92	17.09	**17.03**	1.93
110	18.60	18.44	18.39	18.53	18.54	**18.50**	1.78
120	19.64	19.94	19.61	19.90	19.76	**19.77**	1.67
130	22.05	21.93	22.05	21.81	21.76	**21.92**	1.50
140	23.49	23.38	23.53	23.72	23.63	**23.55**	1.40
150	25.23	25.11	25.05	25.16	25.18	**25.15**	1.31
160	27.09	27.28	27.35	27.57	27.31	**27.32**	1.21
170	28.89	28.82	29.24	29.18	28.87	**29.00**	1.14
180	30.46	30.26	30.70	30.26	30.54	**30.44**	1.08
190	30.74	30.95	30.94	30.71	30.79	**30.82**	1.07
200	32.12	32.31	32.09	32.25	32.04	**32.16**	1.02
210	34.24	34.12	33.77	34.00	33.75	**33.98**	0.97
220	35.89	35.81	36.01	36.16	36.27	**36.03**	0.91
230	38.37	38.34	38.09	38.44	38.18	**38.29**	0.86
240	40.35	39.98	40.31	40.00	40.14	**40.16**	0.82

Table 6.45 Time and speedup of Ladder2(*N*) model with *N* = 16 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `balanced`

`Balanced` Heisenberg Ladder2(16) (recorded time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	8.28	8.20	8.22	8.24	8.25	**8.24**	1.00
10	3.37	3.39	3.45	3.37	3.36	**3.39**	2.43
20	3.44	3.45	3.51	3.45	3.45	**3.46**	2.38
30	4.09	4.10	4.12	4.10	4.12	**4.11**	2.01
40	5.07	5.04	5.12	5.06	5.10	**5.08**	1.62
50	6.42	6.39	6.36	6.40	6.41	**6.40**	1.29
60	8.05	8.07	8.00	8.09	8.04	**8.05**	1.02
70	10.05	10.05	10.02	10.00	10.07	**10.04**	0.82
80	11.10	11.11	11.08	11.12	11.11	**11.10**	0.74
90	12.38	12.46	12.42	12.42	12.40	**12.41**	0.66
100	13.70	13.71	13.62	13.74	13.76	**13.71**	0.60
110	15.25	15.28	15.20	15.26	15.28	**15.25**	0.54
120	16.39	16.33	16.37	16.44	16.34	**16.37**	0.50
130	18.03	18.09	18.03	18.10	18.07	**18.06**	0.46
140	19.68	19.74	19.91	19.81	19.86	**19.80**	0.42
150	21.22	21.24	21.20	21.19	21.21	**21.21**	0.39
160	23.15	23.10	23.11	23.16	23.12	**23.13**	0.36
170	24.69	24.57	24.74	24.81	24.78	**24.72**	0.33
180	26.29	26.23	26.32	26.19	26.16	**26.24**	0.31
190	27.82	27.97	27.85	27.85	27.84	**27.86**	0.30
200	29.51	29.59	29.71	29.67	29.74	**29.65**	0.28
210	30.59	30.51	30.40	30.51	30.58	**30.52**	0.27
220	33.14	32.98	33.16	33.05	33.02	**33.07**	0.25
230	35.46	35.41	35.14	35.17	35.11	**35.26**	0.23
240	37.11	36.77	36.59	36.70	36.62	**36.76**	0.22

6.2.3 Heisenberg Ladder 215C on Xeon Phi

Table 6.46 Time and speedup of Ladder 215C(*N*) model with *N* = 26 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `scatter`

`Scatter` Heisenberg Ladder215C(26) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	5546.54	5544.92	5545.24	5546.00	5545.56	**5545.65**	1.00
10	1314.54	1373.46	1343.54	1312.54	1314.54	**1331.72**	4.16
20	786.67	764.15	783.30	780.39	786.45	**780.19**	7.11
30	565.87	568.16	547.49	560.44	553.37	**559.07**	9.92
40	420.16	422.09	408.09	419.36	420.67	**418.07**	13.26
50	351.58	353.10	357.10	352.40	360.99	**355.03**	15.62
60	302.64	301.46	301.68	300.60	301.29	**301.53**	18.39
70	266.54	266.33	270.66	268.53	269.33	**268.28**	20.67
80	237.81	237.73	238.00	234.84	238.67	**237.41**	23.36
90	216.61	217.73	219.83	222.26	217.54	**218.80**	25.35
100	209.00	212.83	208.42	206.45	207.66	**208.87**	26.55
110	185.39	189.26	187.38	188.05	187.57	**187.53**	29.57
120	179.40	177.99	180.26	179.76	177.68	**179.02**	30.98
130	170.63	171.43	170.33	170.32	169.47	**170.44**	32.54
140	164.04	160.28	155.17	157.68	159.43	**159.32**	34.81
150	157.59	159.45	158.54	155.09	157.45	**157.62**	35.18
160	147.34	146.42	145.33	146.89	145.78	**146.35**	37.89
170	139.21	142.44	142.73	141.77	142.31	**141.69**	39.14
180	137.45	136.74	137.17	136.76	137.66	**137.15**	40.43
190	136.11	132.45	132.01	132.45	134.86	**133.58**	41.52
200	131.25	131.51	132.88	130.89	131.74	**131.65**	42.12
210	124.39	128.81	123.52	124.55	124.60	**125.18**	44.30
220	119.08	119.68	119.58	119.67	119.34	**119.47**	46.42
230	111.96	110.63	109.69	109.45	110.80	**110.51**	50.18
240	109.06	108.16	108.72	109.83	109.96	**109.15**	50.81

Table 6.47 Time and speedup of Ladder 215C(N) model with N = 24 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `scatter`

`Scatter` Heisenberg Ladder215C(24) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	2351.55	2355.30	2354.26	2355.33	2355.14	**2354.32**	1.00
10	569.02	533.47	556.68	604.12	552.95	**563.25**	4.18
20	311.97	322.75	321.53	326.19	331.58	**322.80**	7.29
30	247.05	224.24	240.73	245.90	228.66	**237.32**	9.92
40	186.12	183.99	185.63	185.66	186.67	**185.61**	12.68
50	153.67	150.16	151.67	153.88	149.60	**151.79**	15.51
60	128.68	126.90	129.58	129.32	128.69	**128.63**	18.30
70	114.06	112.09	115.44	115.11	115.49	**114.44**	20.57
80	102.05	105.42	101.52	104.50	101.58	**103.02**	22.85
90	94.41	94.03	92.26	92.65	92.23	**93.12**	25.28
100	89.06	88.90	83.84	83.50	83.54	**85.77**	27.45
110	82.15	75.95	76.56	82.51	82.65	**79.96**	29.44
120	77.56	77.75	78.34	78.33	78.64	**78.13**	30.14
130	74.62	74.25	74.61	74.03	74.73	**74.45**	31.62
140	69.36	70.14	68.83	69.71	69.45	**69.50**	33.88
150	67.02	66.63	68.00	66.20	66.65	**66.90**	35.19
160	62.89	62.40	67.17	66.71	66.57	**65.15**	36.14
170	61.65	66.98	61.03	64.91	66.55	**64.22**	36.66
180	64.54	62.09	61.62	60.27	61.72	**62.05**	37.94
190	64.06	64.84	63.51	59.01	59.68	**62.22**	37.84
200	60.33	60.95	57.56	58.71	56.89	**58.89**	39.98
210	55.91	55.04	55.61	54.39	56.08	**55.40**	42.49
220	52.24	53.38	52.69	53.23	56.71	**53.65**	43.88
230	50.30	50.44	50.26	50.49	50.80	**50.45**	46.66
240	50.01	49.58	49.80	49.59	50.04	**49.80**	47.27

Table 6.48 Time and speedup of Ladder 215C(N) model with N = 22 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `scatter`

`Scatter` Heisenberg Ladder215C(22) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	537.74	537.83	534.50	539.50	539.20	**537.75**	1.00
10	120.58	127.88	122.16	120.74	120.13	**122.30**	4.40
20	72.74	77.01	77.08	68.84	77.39	**74.61**	7.21
30	51.73	53.79	53.48	55.20	54.85	**53.81**	9.99
40	41.84	41.53	41.21	41.11	41.65	**41.47**	12.97
50	33.60	32.90	33.89	33.43	33.76	**33.52**	16.04
60	28.43	28.63	28.61	28.39	28.43	**28.50**	18.87
70	25.70	25.77	25.80	25.76	25.76	**25.76**	20.88
80	24.06	22.77	24.18	22.80	24.08	**23.58**	22.81
90	22.34	21.85	22.57	21.44	22.47	**22.13**	24.30
100	20.47	20.51	20.66	19.71	20.50	**20.37**	26.40
110	18.85	18.94	19.12	19.11	19.09	**19.02**	28.27
120	17.94	17.96	18.20	18.12	18.12	**18.07**	29.76
130	17.37	17.39	17.57	17.51	17.50	**17.47**	30.78
140	17.09	16.54	16.68	16.63	16.63	**16.72**	32.17
150	19.38	18.86	18.92	18.86	18.86	**18.97**	28.34
160	19.20	19.26	19.88	18.73	18.65	**19.14**	28.09
170	19.04	19.28	19.14	19.46	19.36	**19.25**	27.93
180	19.07	19.22	19.30	19.29	19.34	**19.24**	27.94
190	19.31	19.37	19.39	20.07	19.48	**19.52**	27.54
200	19.67	19.77	19.73	20.18	19.78	**19.83**	27.12
210	19.99	20.02	19.97	20.23	20.04	**20.05**	26.82
220	20.20	20.24	20.28	20.29	20.26	**20.26**	26.55
230	20.47	20.47	20.48	20.48	20.51	**20.48**	26.25
240	21.09	20.92	20.86	20.95	20.87	**20.94**	25.68

Table 6.49 Time and speedup of Ladder 215C(N) model with N = 20 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `scatter`

`Scatter` Heisenberg Ladder215C(20) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	135.34	136.35	134.47	134.58	134.60	**135.07**	1.00
10	28.84	28.79	28.77	28.53	28.92	**28.77**	4.69
20	17.32	18.62	17.89	17.20	18.71	**17.95**	7.53
30	12.99	13.72	13.84	13.72	13.87	**13.63**	9.91
40	11.49	11.95	11.88	11.86	11.83	**11.80**	11.45
50	10.96	10.97	10.95	10.89	10.96	**10.95**	12.34
60	10.52	10.48	10.45	10.46	10.44	**10.47**	12.90
70	10.40	10.37	10.37	10.34	10.50	**10.40**	12.99
80	10.27	10.23	10.60	10.20	10.32	**10.32**	13.08
90	10.64	10.60	11.23	10.72	10.63	**10.76**	12.55
100	10.96	11.02	11.51	10.98	11.03	**11.10**	12.16
110	11.41	11.36	11.89	11.30	11.40	**11.47**	11.77
120	11.78	11.73	12.37	11.73	11.73	**11.87**	11.38
130	10.98	10.96	11.97	11.03	10.96	**11.18**	12.08
140	11.50	11.60	12.48	11.74	11.56	**11.77**	11.47
150	12.18	12.41	13.23	12.37	12.22	**12.48**	10.82
160	12.82	13.01	13.74	12.84	13.01	**13.08**	10.32
170	13.69	13.54	14.08	13.42	13.52	**13.65**	9.90
180	13.96	13.85	13.85	13.92	13.92	**13.90**	9.72
190	14.81	14.81	14.72	14.72	14.75	**14.76**	9.15
200	15.28	15.27	15.30	15.28	15.31	**15.29**	8.84
210	15.89	15.88	15.91	15.91	15.90	**15.90**	8.50
220	16.40	16.45	16.44	16.41	16.43	**16.43**	8.22
230	16.94	17.02	16.95	16.94	16.98	**16.97**	7.96
240	17.82	17.51	17.53	17.54	17.55	**17.59**	7.68

Table 6.50 Time and speedup of Ladder 215C(N) model with N = 18 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `scatter`

`Scatter` Heisenberg Ladder215C(18) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	32.99	32.94	33.02	32.97	33.13	**33.01**	1.00
10	8.00	8.09	8.71	8.72	7.92	**8.29**	3.98
20	6.17	6.29	6.30	6.30	6.45	**6.30**	5.24
30	4.35	4.29	4.38	4.40	4.37	**4.36**	7.57
40	4.67	4.64	4.64	4.75	4.68	**4.68**	7.06
50	5.29	5.30	5.30	5.40	5.30	**5.32**	6.21
60	6.22	6.26	6.24	6.37	6.30	**6.28**	5.26
70	7.05	7.04	7.08	7.28	7.10	**7.11**	4.64
80	7.76	7.74	7.75	7.90	7.78	**7.79**	4.24
90	8.44	8.45	8.47	8.63	8.49	**8.49**	3.89
100	9.16	9.17	9.19	9.32	9.21	**9.21**	3.58
110	9.94	9.90	9.91	10.07	9.94	**9.95**	3.32
120	10.73	10.69	10.72	10.76	10.73	**10.73**	3.08
130	11.61	11.61	11.68	11.67	11.64	**11.64**	2.84
140	12.47	12.41	12.43	12.45	12.45	**12.44**	2.65
150	13.13	13.14	13.15	13.18	13.17	**13.15**	2.51
160	13.88	13.88	13.89	13.92	13.94	**13.90**	2.37
170	14.59	14.60	14.63	14.64	14.68	**14.63**	2.26
180	15.31	15.32	15.35	15.35	15.34	**15.33**	2.15
190	16.34	16.31	16.34	16.32	16.34	**16.33**	2.02
200	17.08	17.08	17.10	17.08	17.10	**17.09**	1.93
210	17.80	17.81	17.83	17.80	17.81	**17.81**	1.85
220	18.53	18.55	18.56	18.51	18.54	**18.54**	1.78
230	19.28	19.29	19.31	19.27	19.30	**19.29**	1.71
240	20.34	20.01	20.04	20.04	20.05	**20.10**	1.64

Table 6.51 Time and speedup of Ladder 215C(N) model with N = 16 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `scatter`

`Scatter` Heisenberg Ladder215C(16) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	8.28	8.23	8.27	8.27	8.29	**8.27**	1.00
10	3.04	3.05	3.05	3.05	3.05	**3.05**	2.71
20	3.01	3.01	3.00	3.01	3.01	**3.01**	2.74
30	3.33	3.35	3.35	3.36	3.36	**3.35**	2.47
40	3.89	3.90	3.89	3.91	3.91	**3.90**	2.12
50	4.62	4.65	4.64	4.64	4.65	**4.64**	1.78
60	5.57	5.55	5.55	5.57	5.56	**5.56**	1.49
70	6.32	6.33	6.31	6.32	6.28	**6.31**	1.31
80	6.94	6.94	6.95	6.96	6.96	**6.95**	1.19
90	7.76	7.76	7.76	7.76	7.77	**7.76**	1.07
100	8.45	8.45	8.45	8.47	8.45	**8.45**	0.98
110	9.12	9.11	9.13	9.12	9.13	**9.12**	0.91
120	9.79	9.80	9.82	9.81	9.81	**9.81**	0.84
130	10.59	10.58	10.58	10.59	10.59	**10.59**	0.78
140	11.31	11.29	11.29	11.30	11.30	**11.30**	0.73
150	12.03	12.03	12.02	12.04	11.99	**12.02**	0.69
160	12.67	12.66	12.67	12.69	12.68	**12.67**	0.65
170	13.34	13.35	13.36	13.36	13.36	**13.35**	0.62
180	14.00	14.01	14.01	14.02	14.01	**14.01**	0.59
190	14.92	14.89	14.90	14.91	14.90	**14.90**	0.55
200	15.59	15.56	15.56	15.56	15.55	**15.57**	0.53
210	16.24	16.23	16.23	16.21	16.22	**16.23**	0.51
220	16.91	16.90	16.89	16.90	16.90	**16.90**	0.49
230	17.57	17.57	17.56	17.57	17.57	**17.57**	0.47
240	18.57	18.22	18.22	18.23	18.23	**18.29**	0.45

Table 6.52 Time and speedup of Ladder 215C(N) model with N = 26 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `compact`

`Compact` Heisenberg Ladder215C(26) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	5546.54	5544.92	5545.24	5546.00	5545.56	**5545.65**	5546.54
10	1591.48	1615.95	1620.90	1617.41	1616.34	**1612.42**	1591.48
20	848.94	848.35	848.39	848.67	848.66	**848.60**	848.94
30	590.84	573.56	585.34	583.45	584.33	**583.50**	590.84
40	439.94	439.65	439.77	439.53	439.55	**439.69**	439.94
50	359.70	352.95	353.01	353.02	353.00	**354.33**	359.70
60	300.94	301.04	301.11	301.02	300.67	**300.96**	300.94
70	259.29	259.05	259.11	259.24	259.11	**259.16**	259.29
80	227.82	227.68	227.81	227.63	227.90	**227.77**	227.82
90	207.29	207.41	207.36	207.34	207.29	**207.34**	207.29
100	189.46	189.35	189.40	189.30	189.33	**189.37**	189.46
110	173.97	173.86	174.00	174.02	173.92	**173.95**	173.97
120	162.69	162.73	162.37	162.44	162.56	**162.56**	162.69
130	152.31	152.42	152.33	152.55	152.35	**152.39**	152.31
140	142.54	142.55	142.49	142.47	142.54	**142.52**	142.54
150	134.42	134.45	134.77	134.61	134.54	**134.56**	134.42
160	126.73	126.51	126.53	126.64	126.55	**126.59**	126.73
170	121.15	121.12	121.06	122.36	121.25	**121.39**	121.15
180	116.28	116.38	116.23	116.27	116.44	**116.32**	116.28
190	110.40	110.32	110.29	110.28	110.24	**110.31**	110.40
200	107.14	107.00	106.99	106.96	106.97	**107.01**	107.14
210	103.46	103.34	103.59	103.44	103.01	**103.37**	103.46
220	100.68	100.54	100.60	100.67	100.69	**100.64**	100.68
230	97.74	97.66	97.62	97.33	97.56	**97.58**	97.74
240	94.98	95.57	94.79	94.80	94.88	**95.00**	94.98

Table 6.53 Time and speedup of Ladder 215C(N) model with N = 24 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `compact`

`Compact` Heisenberg Ladder215C(24) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	2351.55	2355.30	2354.26	2355.33	2355.14	**2354.32**	1.00
10	696.64	696.05	696.79	696.64	696.26	**696.48**	3.38
20	374.57	374.48	374.59	374.60	374.46	**374.54**	6.29
30	254.88	254.92	257.87	254.99	254.91	**255.51**	9.21
40	196.22	196.08	196.12	196.16	196.07	**196.13**	12.00
50	159.36	159.39	159.45	159.42	159.36	**159.40**	14.77
60	136.88	136.84	136.86	136.86	136.81	**136.85**	17.20
70	118.40	118.40	118.40	118.41	118.40	**118.40**	19.88
80	104.43	104.44	104.45	104.36	104.35	**104.41**	22.55
90	95.80	95.77	95.79	95.78	95.74	**95.78**	24.58
100	87.99	87.97	88.00	87.98	87.98	**87.98**	26.76
110	81.53	81.40	81.40	81.39	81.42	**81.43**	28.91
120	76.29	76.29	76.32	76.32	76.33	**76.31**	30.85
130	71.96	71.93	71.93	71.95	72.05	**71.96**	32.71
140	67.81	67.79	67.84	67.82	67.84	**67.82**	34.71
150	64.22	64.22	64.22	64.20	64.29	**64.23**	36.65
160	60.59	60.59	60.51	60.56	60.54	**60.56**	38.88
170	60.52	60.33	60.33	60.37	60.35	**60.38**	38.99
180	55.96	55.99	55.95	55.98	55.99	**55.98**	42.06
190	53.28	53.34	53.33	53.31	53.27	**53.31**	44.17
200	51.82	51.87	51.86	51.85	51.87	**51.85**	45.40
210	50.40	50.40	50.38	50.40	50.40	**50.40**	46.72
220]	49.47	49.46	49.46	49.47	49.48	**49.47**	47.59
230	47.68	47.67	47.66	47.67	47.65	**47.67**	49.39
240	46.40	46.20	46.21	46.26	46.23	**46.26**	50.89

Table 6.54 Time and speedup of Ladder 215C(N) model with N = 22 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `compact`

`Compact` Heisenberg Ladder215C(22) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	537.74	537.83	534.50	539.50	539.20	**537.75**	1.00
10	165.07	165.08	165.11	165.09	165.10	**165.09**	3.26
20	91.31	91.30	91.35	91.35	91.30	**91.32**	5.89
30	63.78	63.80	63.78	63.76	63.75	**63.77**	8.43
40	49.33	49.34	49.37	49.35	49.35	**49.35**	10.90
50	41.38	41.35	41.36	41.36	41.38	**41.37**	13.00
60	35.26	35.23	35.22	35.24	35.22	**35.23**	15.26
70	30.79	30.77	30.77	30.76	30.77	**30.77**	17.47
80	26.79	26.75	26.77	26.74	26.74	**26.76**	20.10
90	24.42	24.41	24.37	24.37	24.41	**24.39**	22.04
100	22.01	21.99	22.00	21.98	21.97	**21.99**	24.46
110	20.08	20.05	20.03	20.05	20.07	**20.06**	26.81
120	18.72	18.66	18.68	18.69	18.66	**18.68**	28.79
130	17.71	17.70	17.70	17.69	17.69	**17.70**	30.38
140	16.58	16.56	16.58	16.58	16.56	**16.57**	32.45
150	18.82	18.79	18.82	18.83	18.78	**18.81**	28.59
160	18.57	18.53	18.54	18.55	18.53	**18.54**	29.00
170	18.78	18.77	18.77	18.76	18.74	**18.76**	28.66
180	18.78	18.76	18.77	18.77	18.77	**18.77**	28.65
190	18.79	18.76	18.79	18.80	18.77	**18.78**	28.63
200	19.25	19.23	19.24	19.21	19.23	**19.23**	27.96
210	19.66	19.63	19.62	19.64	19.67	**19.64**	27.38
220	20.17	20.18	20.13	20.18	20.19	**20.17**	26.66
230	20.75	20.75	20.75	20.76	20.75	**20.75**	25.91
240	21.58	21.25	21.24	21.25	21.24	**21.31**	25.23

Table 6.55 Time and speedup of Ladder 215C(N) model with N = 20 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `compact`

`Compact` Heisenberg Ladder215C(20) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	135.34	136.35	134.47	134.58	134.60	**135.07**	1.00
10	42.95	42.91	42.91	43.00	43.01	**42.96**	3.14
20	24.18	24.15	24.13	24.11	24.12	**24.14**	5.60
30	16.27	16.19	16.20	16.22	16.22	**16.22**	8.33
40	13.84	13.79	13.80	13.81	13.80	**13.81**	9.78
50	12.63	12.60	12.61	12.59	12.63	**12.61**	10.71
60	11.84	11.82	11.79	11.80	11.83	**11.82**	11.43
70	11.35	11.37	11.36	11.36	11.34	**11.36**	11.89
80	10.95	10.94	10.93	10.93	10.94	**10.94**	12.35
90	11.05	11.00	11.03	11.02	11.03	**11.03**	12.25
100	11.18	11.16	11.17	11.16	11.16	**11.16**	12.10
110	11.42	11.35	11.37	11.36	11.38	**11.37**	11.87
120	11.63	11.62	11.63	11.62	11.60	**11.62**	11.62
130	10.91	10.89	10.88	10.96	10.98	**10.92**	12.36
140	11.27	11.29	11.27	11.41	11.38	**11.32**	11.93
150	11.66	11.63	11.64	11.70	11.71	**11.67**	11.58
160	12.04	12.04	12.05	12.12	12.09	**12.07**	11.19
170	12.45	12.44	12.45	12.48	12.47	**12.46**	10.84
180	12.90	12.89	12.89	12.98	12.94	**12.92**	10.45
190	13.41	13.28	13.29	13.38	13.35	**13.34**	10.12
200	13.80	13.75	13.75	13.75	13.79	**13.77**	9.81
210	14.32	14.22	14.23	14.26	14.28	**14.26**	9.47
220	14.80	14.70	14.70	14.71	14.83	**14.75**	9.16
230	15.36	15.26	15.22	15.20	15.29	**15.26**	8.85
240	16.09	15.66	15.65	15.65	15.73	**15.75**	8.57

Table 6.56 Time and speedup of Ladder 215C(N) model with N = 18 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `compact`

`Compact` Heisenberg Ladder215C(18) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	32.99	32.94	33.02	32.97	33.13	**33.01**	1.00
10	11.21	11.21	11.23	11.17	11.18	**11.20**	2.95
20	8.24	8.21	8.23	8.19	8.19	**8.21**	4.02
30	6.83	6.80	6.83	6.72	6.74	**6.78**	4.87
40	6.61	6.58	6.61	6.58	6.56	**6.59**	5.01
50	6.90	6.88	6.89	6.79	6.79	**6.85**	4.82
60	7.00	6.98	6.99	6.99	6.97	**6.98**	4.73
70	7.34	7.34	7.34	7.30	7.32	**7.33**	4.50
80	7.75	7.74	7.73	7.72	7.71	**7.73**	4.27
90	8.15	8.15	8.15	8.13	8.12	**8.14**	4.06
100	8.61	8.61	8.64	8.58	8.55	**8.60**	3.84
110	9.13	9.10	9.08	9.11	9.10	**9.10**	3.63
120	9.57	9.58	9.58	9.58	9.56	**9.57**	3.45
130	10.09	10.07	10.09	10.09	10.08	**10.09**	3.27
140	10.62	10.60	10.62	10.61	10.61	**10.61**	3.11
150	11.11	11.10	11.11	11.12	11.10	**11.11**	2.97
160	11.64	11.62	11.63	11.66	11.64	**11.64**	2.84
170	12.17	12.18	12.15	12.17	12.19	**12.17**	2.71
180	12.77	12.72	12.74	12.74	12.72	**12.74**	2.59
190	13.31	13.25	13.28	13.26	13.29	**13.28**	2.49
200	13.85	13.85	13.87	13.84	13.82	**13.85**	2.38
210	14.42	14.41	14.42	14.41	14.43	**14.42**	2.29
220	14.97	14.98	14.97	14.98	14.99	**14.98**	2.20
230	15.59	15.57	15.58	15.57	15.59	**15.58**	2.12
240	16.49	16.17	16.18	16.17	16.22	**16.25**	2.03

Table 6.57 Time and speedup of Ladder 215C(N) model with N = 16 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `compact`

`Compact` Heisenberg Ladder215C(16) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	8.28	8.23	8.27	8.27	8.29	**8.27**	1.00
10	4.17	4.15	4.16	4.28	4.24	**4.20**	1.97
20	3.87	3.86	3.86	3.98	3.95	**3.90**	2.12
30	4.04	4.07	4.04	4.17	4.13	**4.09**	2.02
40	4.38	4.36	4.37	4.46	4.43	**4.40**	1.88
50	4.81	4.82	4.82	4.87	4.86	**4.84**	1.71
60	5.25	5.23	5.21	5.31	5.28	**5.25**	1.57
70	5.70	5.70	5.68	5.78	5.72	**5.72**	1.45
80	6.16	6.13	6.14	6.24	6.20	**6.18**	1.34
90	6.75	6.74	6.76	6.79	6.79	**6.77**	1.22
100	7.25	7.23	7.23	7.31	7.31	**7.27**	1.14
110	7.84	7.83	7.84	7.87	7.85	**7.84**	1.05
120	8.42	8.39	8.39	8.46	8.44	**8.42**	0.98
130	9.03	9.01	9.01	9.09	9.07	**9.04**	0.91
140	9.62	9.57	9.59	9.70	9.69	**9.64**	0.86
150	10.24	10.19	10.21	10.31	10.29	**10.25**	0.81
160	10.90	10.85	10.85	11.00	10.97	**10.91**	0.76
170	11.61	11.56	11.57	11.69	11.66	**11.62**	0.71
180	12.32	12.26	12.27	12.45	12.35	**12.33**	0.67
190	13.06	13.00	13.02	13.20	13.13	**13.08**	0.63
200	13.86	13.78	13.79	13.99	13.91	**13.87**	0.60
210	14.83	14.74	14.75	14.95	14.85	**14.83**	0.56
220	15.63	15.48	15.48	15.76	15.70	**15.61**	0.53
230	16.66	16.53	16.54	16.80	16.76	**16.66**	0.50
240	17.92	17.45	17.44	18.17	17.69	**17.73**	0.47

Table 6.58 Time and speedup of Ladder 215C(N) model with N = 26 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `balanced`

`Balanced` Heisenberg Ladder215C(26) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	5546.54	5544.92	5545.24	5546.00	5545.56	**5545.65**	1.00
10	1355.22	1366.78	1366.92	1361.38	1366.33	**1363.33**	4.07
20	733.29	783.94	785.89	760.13	784.44	**769.54**	7.21
30	534.78	536.15	536.44	546.62	540.03	**538.81**	10.29
40	407.68	426.61	417.82	417.13	420.44	**417.94**	13.27
50	345.18	351.43	348.59	349.43	348.13	**348.55**	15.91
60	300.32	300.42	301.99	300.92	301.23	**300.98**	18.43
70	266.42	269.05	270.08	268.44	269.23	**268.64**	20.64
80	236.94	234.77	236.32	234.40	237.89	**236.06**	23.49
90	216.77	218.17	216.58	217.00	216.49	**217.00**	25.56
100	194.73	198.56	196.94	197.33	196.34	**196.78**	28.18
110	177.39	178.04	179.52	178.55	178.36	**178.37**	31.09
120	168.43	168.10	170.22	168.50	168.18	**168.69**	32.88
130	161.37	164.04	163.84	161.45	161.66	**162.47**	34.13
140	151.37	151.88	154.45	151.32	152.03	**152.21**	36.43
150	145.00	144.37	145.62	145.22	145.55	**145.15**	38.21
160	138.49	138.14	136.85	134.97	136.34	**136.96**	40.49
170	133.04	132.66	133.86	134.47	133.36	**133.48**	41.55
180	127.21	128.96	128.81	127.76	128.00	**128.15**	43.27
190	120.98	122.09	123.21	122.44	122.99	**122.34**	45.33
200	116.68	117.17	116.97	117.67	116.83	**117.07**	47.37
210	113.22	112.88	111.69	113.32	113.43	**112.91**	49.12
220	110.07	110.34	110.78	109.27	109.45	**109.98**	50.42
230	97.60	97.52	97.83	97.46	97.89	**97.66**	56.79
240	95.49	95.50	94.96	94.55	95.84	**95.27**	58.21

Table 6.59 Time and speedup of Ladder 215C(N) model with N = 24 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `balanced`

`Balanced` Heisenberg Ladder215C(24) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	2351.55	2355.30	2354.26	2355.33	2355.14	**2354.32**	1.00
10	537.24	529.52	529.76	549.94	563.11	**541.91**	4.34
20	333.32	327.72	310.97	301.85	339.03	**322.58**	7.30
30	242.93	233.59	225.93	237.99	239.96	**236.08**	9.97
40	176.10	180.54	181.38	186.06	183.42	**181.50**	12.97
50	152.09	153.17	145.13	145.77	153.33	**149.90**	15.71
60	128.39	130.69	129.76	129.97	130.98	**129.96**	18.12
70	114.72	116.04	116.52	118.53	113.49	**115.86**	20.32
80	102.94	97.58	106.08	99.15	100.05	**101.16**	23.27
90	95.64	96.18	92.26	96.71	92.37	**94.63**	24.88
100	85.61	85.58	88.32	88.18	87.59	**87.06**	27.04
110	80.90	80.16	81.74	81.25	81.11	**81.03**	29.05
120	76.35	75.05	78.04	75.77	76.31	**76.30**	30.85
130	75.38	72.23	74.20	71.48	73.86	**73.43**	32.06
140	70.20	69.03	68.77	68.37	67.91	**68.85**	34.19
150	65.23	65.65	65.74	64.29	65.94	**65.37**	36.01
160	61.91	60.72	61.30	60.88	61.82	**61.32**	38.39
170	61.54	61.84	59.33	61.22	61.11	**61.01**	38.59
180	57.56	57.97	58.26	58.38	58.17	**58.07**	40.54
190	57.18	56.40	56.54	56.96	56.54	**56.72**	41.51
200	54.77	55.38	54.08	54.95	53.77	**54.59**	43.13
210	51.89	52.10	52.22	51.95	51.87	**52.01**	45.27
220]	51.34	52.11	52.33	51.61	51.06	**51.69**	45.55
230	47.56	47.68	47.46	47.55	47.52	**47.55**	49.51
240	46.30	46.18	46.25	46.21	46.24	**46.24**	50.92

Table 6.60 Time and speedup of Ladder 215C(N) model with N = 22 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `balanced`

`Balanced` Heisenberg Ladder215C(22) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	537.74	537.83	534.50	539.50	539.20	**537.75**	1.00
10	119.87	124.44	132.22	123.49	136.63	**127.33**	4.22
20	70.99	72.28	74.65	74.18	72.36	**72.90**	7.38
30	49.90	51.60	53.15	52.06	55.15	**52.37**	10.27
40	39.89	40.85	40.17	40.40	41.31	**40.52**	13.27
50	33.65	33.39	34.16	32.61	34.34	**33.63**	15.99
60	31.04	31.49	29.29	29.04	29.79	**30.13**	17.85
70	26.49	26.26	27.88	28.38	26.51	**27.10**	19.84
80	23.16	24.76	23.13	24.80	23.45	**23.86**	22.54
90	21.68	22.94	21.55	22.96	21.57	**22.14**	24.29
100	19.89	20.81	19.97	20.21	20.81	**20.34**	26.44
110	19.67	19.42	18.54	18.50	19.40	**19.11**	28.14
120	18.18	18.22	18.44	18.19	18.10	**18.23**	29.50
130	17.60	17.82	17.52	17.79	17.66	**17.68**	30.42
140	17.06	16.53	16.77	16.94	16.64	**16.79**	32.03
150	63.28	63.66	63.72	63.87	63.62	**63.63**	8.45
160	70.41	70.75	71.16	70.47	70.71	**70.70**	7.61
170	80.32	80.41	80.48	80.50	80.35	**80.41**	6.69
180	90.98	91.84	91.91	91.72	91.72	**91.63**	5.87
190	90.99	90.64	90.74	91.18	90.64	**90.84**	5.92
200	113.68	114.41	113.43	113.72	113.40	**113.73**	4.73
210	122.18	123.51	123.20	124.56	123.25	**123.34**	4.36
220	116.07	115.84	116.35	115.99	115.40	**115.93**	4.64
230	129.81	129.72	130.59	129.37	130.24	**129.95**	4.14
240	142.34	142.49	141.47	141.47	142.28	**142.01**	3.79

Table 6.61 Time and speedup of Ladder 215C(N) model with N = 20 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `balanced`

`Balanced` Heisenberg Ladder215C(20) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	135.34	136.35	134.47	134.58	134.60	**135.07**	1.00
10	28.73	28.72	32.53	28.55	28.56	**29.42**	4.59
20	17.28	17.94	18.61	18.56	17.25	**17.93**	7.53
30	13.77	13.69	13.43	13.47	12.96	**13.47**	10.03
40	15.39	15.44	15.44	15.51	15.50	**15.45**	8.74
50	15.58	15.49	15.51	15.49	15.51	**15.52**	8.71
60	16.39	16.43	16.42	16.51	16.41	**16.43**	8.22
70	29.39	29.12	28.93	29.13	29.12	**29.14**	4.64
80	34.13	34.04	34.24	34.40	33.84	**34.13**	3.96
90	43.68	43.69	43.76	43.47	43.96	**43.71**	3.09
100	50.48	51.06	50.80	50.40	50.51	**50.65**	2.67
110	51.79	51.45	52.22	50.69	51.86	**51.60**	2.62
120	52.25	52.35	53.01	52.40	51.99	**52.40**	2.58
130	59.58	59.30	59.70	59.38	59.96	**59.59**	2.27
140	52.31	52.36	52.39	52.53	52.21	**52.36**	2.58
150	62.16	61.95	61.96	62.03	61.90	**62.00**	2.18
160	69.47	69.40	69.94	69.73	69.62	**69.63**	1.94
170	71.42	71.75	71.90	71.81	71.61	**71.70**	1.88
180	72.73	73.62	73.48	72.90	73.46	**73.24**	1.84
190	68.29	68.29	68.08	68.08	67.80	**68.11**	1.98
200	70.02	69.73	70.15	69.68	69.90	**69.90**	1.93
210	78.10	78.93	78.32	77.84	78.48	**78.33**	1.72
220	85.15	85.36	84.87	85.42	85.54	**85.27**	1.58
230	86.94	87.63	87.84	87.45	88.27	**87.63**	1.54
240	90.13	89.68	90.16	88.92	89.31	**89.64**	1.51

Table 6.62 Time and speedup of Ladder 215C(*N*) model with *N* = 18 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `balanced`

`Balanced` Heisenberg Ladder215C(18) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	32.99	32.94	33.02	32.97	33.13	**33.01**	1.00
10	8.85	9.58	8.84	8.83	8.99	**9.02**	3.66
20	7.50	7.46	7.46	7.34	7.26	**7.41**	4.46
30	5.74	5.76	5.80	5.66	5.71	**5.73**	5.76
40	6.53	6.56	6.57	6.55	6.60	**6.56**	5.03
50	7.98	8.01	8.05	8.08	8.03	**8.03**	4.11
60	10.26	10.04	10.08	10.15	10.22	**10.15**	3.25
70	15.17	15.57	15.57	15.44	15.55	**15.46**	2.13
80	16.67	16.55	16.50	16.61	16.59	**16.58**	1.99
90	17.73	17.80	18.00	17.82	18.01	**17.87**	1.85
100	19.55	19.45	19.36	19.33	19.68	**19.47**	1.69
110	21.33	21.36	21.20	21.31	21.32	**21.30**	1.55
120	22.54	22.61	22.28	22.54	22.75	**22.54**	1.46
130	24.46	24.48	24.68	24.70	24.67	**24.60**	1.34
140	26.48	26.63	26.53	26.65	26.60	**26.58**	1.24
150	28.53	28.27	28.26	28.39	28.62	**28.41**	1.16
160	30.55	30.69	30.37	30.61	30.41	**30.53**	1.08
170	32.18	32.50	32.20	32.48	32.44	**32.36**	1.02
180	33.71	33.94	33.84	33.81	33.74	**33.81**	0.98
190	34.48	34.40	34.33	34.33	34.23	**34.35**	0.96
200	35.81	35.91	35.67	36.18	35.93	**35.90**	0.92
210	38.19	38.29	38.34	38.19	38.14	**38.23**	0.86
220	40.69	41.19	40.59	40.98	41.12	**40.91**	0.81
230	43.37	43.27	43.20	43.14	43.11	**43.22**	0.76
240	45.59	45.30	45.36	45.44	45.36	**45.41**	0.73

Table 6.63 Time and speedup of Ladder 215C(*N*) model with *N* = 16 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `balanced`

`Balanced` Heisenberg Ladder215C(16) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	8.28	8.23	8.27	8.27	8.29	**8.27**	1.00
10	3.37	3.43	3.53	3.36	3.46	**3.43**	2.41
20	3.52	3.56	3.57	3.52	3.56	**3.55**	2.33
30	4.27	4.28	4.28	4.58	4.27	**4.34**	1.91
40	5.36	5.38	5.35	5.73	5.40	**5.44**	1.52
50	6.86	6.86	6.88	6.92	6.87	**6.88**	1.20
60	8.98	8.88	8.86	8.88	8.87	**8.89**	0.93
70	11.01	11.02	11.12	11.12	11.20	**11.09**	0.75
80	12.34	12.24	12.40	12.22	12.37	**12.31**	0.67
90	13.60	13.68	13.77	13.65	13.72	**13.68**	0.60
100	15.05	14.99	15.04	15.14	15.61	**15.17**	0.55
110	16.68	16.86	16.67	16.61	17.43	**16.85**	0.49
120	18.01	17.98	18.04	17.98	18.03	**18.01**	0.46
130	19.80	19.85	19.87	19.70	19.75	**19.79**	0.42
140	21.51	21.50	21.57	21.68	21.63	**21.58**	0.38
150	23.15	23.03	23.19	23.07	23.33	**23.15**	0.36
160	25.09	25.02	25.10	25.23	25.41	**25.17**	0.33
170	26.83	26.69	26.89	26.83	26.93	**26.83**	0.31
180	28.57	28.54	28.52	28.64	28.60	**28.57**	0.29
190	30.11	30.13	30.20	30.17	30.16	**30.15**	0.27
200	31.25	31.28	31.20	31.16	31.24	**31.23**	0.26
210	33.76	33.78	33.63	33.80	33.78	**33.75**	0.24
220	36.01	36.24	36.23	36.05	36.14	**36.14**	0.23
230	38.14	38.00	38.10	38.11	38.11	**38.09**	0.22
240	40.23	39.83	39.93	39.83	39.92	**39.95**	0.21

6.2.4 Heisenberg Ladder 215S on Xeon Phi

Table 6.64 Time and speedup of Ladder 215S(*N*) model with *N* = 26 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `scatter`

`Scatter` Heisenberg Ladder215S(26) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	–	–	–	–	–	–	–
10	–	–	–	–	–	–	–
20	–	–	–	–	–	–	–
30	–	–	–	–	–	–	–
40	–	–	–	–	–	–	–
50	–	–	–	–	–	–	–
60	–	–	–	–	–	–	–
70	–	–	–	–	–	–	–
80	–	–	–	–	–	–	–
90	–	–	–	–	–	–	–
100	–	–	–	–	–	–	–
110	–	–	–	–	–	–	–
120	–	–	–	–	–	–	–
130	–	–	–	–	–	–	–
140	–	–	–	–	–	–	–
150	–	–	–	–	–	–	–
160	–	–	–	–	–	–	–
170	–	–	–	–	–	–	–
180	–	–	–	–	–	–	–
190	–	–	–	–	–	–	–
200	–	–	–	–	–	–	–
210	–	–	–	–	–	–	–
220	–	–	–	–	–	–	–
230	–	–	–	–	–	–	–
240	–	–	–	–	–	–	–

Table 6.65 Time and speedup of Ladder 215S(*N*) model with *N* = 24 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `scatter`

`Scatter` Heisenberg Ladder215S(24) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	2354.56	2355.89	2356.18	2355.27	2354.79	**2355.34**	1.00
10	554.12	595.99	570.68	558.85	572.24	**570.38**	4.13
20	308.48	325.02	325.96	317.64	301.21	**315.66**	7.46
30	236.02	233.41	235.03	240.10	214.45	**231.80**	10.16
40	175.22	183.40	183.80	179.79	179.45	**180.33**	13.06
50	148.45	148.29	149.98	148.17	150.67	**149.11**	15.80
60	130.43	129.77	129.99	131.34	129.88	**130.28**	18.08
70	114.94	114.38	115.27	114.81	115.73	**115.02**	20.48
80	100.23	102.51	99.64	101.05	101.68	**101.02**	23.32
90	93.27	92.66	97.70	93.19	95.57	**94.48**	24.93
100	82.71	83.02	82.50	88.32	85.07	**84.32**	27.93
110	78.38	77.63	82.14	81.34	71.50	**78.20**	30.12
120	78.01	77.96	77.63	78.13	77.51	**77.85**	30.26
130	73.96	74.96	73.87	74.31	73.14	**74.05**	31.81
140	70.14	69.13	69.96	69.04	69.00	**69.46**	33.91
150	68.10	68.70	66.76	66.11	66.94	**67.32**	34.99
160	65.98	65.49	62.91	64.61	63.29	**64.46**	36.54
170	60.96	63.88	67.24	64.57	62.74	**63.88**	36.87
180	62.01	61.07	61.73	60.64	60.94	**61.28**	38.44
190	59.57	60.19	60.26	59.29	60.01	**59.87**	39.34
200	58.33	58.37	57.84	58.29	58.16	**58.20**	40.47
210	54.51	54.59	54.53	54.51	54.69	**54.57**	43.17
220	53.32	53.51	53.54	53.30	53.81	**53.50**	44.03
230	50.51	50.63	50.64	50.50	50.74	**50.60**	46.54
240	49.62	49.58	49.65	49.71	49.67	**49.65**	47.44

Table 6.66 Time and speedup of Ladder 215S(*N*) model with *N* = 22 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `scatter`

`Scatter` Heisenberg Ladder215S(22) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	534.98	539.45	539.39	539.47	539.60	**538.58**	1.00
10	127.11	133.64	132.48	123.07	119.89	**127.24**	4.23
20	79.67	79.76	76.60	82.03	73.36	**78.29**	6.88
30	48.19	55.14	53.75	56.84	54.30	**53.64**	10.04
40	40.99	40.78	40.95	42.57	39.91	**41.04**	13.12
50	33.59	33.82	33.47	34.45	33.16	**33.70**	15.98
60	28.71	29.40	28.64	29.14	28.47	**28.87**	18.65
70	26.17	27.49	25.78	26.12	25.75	**26.26**	20.51
80	22.80	24.32	22.67	22.97	23.31	**23.21**	23.20
90	21.37	22.56	21.29	21.41	22.53	**21.83**	24.67
100	20.67	20.69	20.60	19.91	19.72	**20.32**	26.51
110	19.01	19.05	18.93	19.14	19.25	**19.08**	28.23
120	17.04	17.99	18.36	18.72	18.21	**18.07**	29.81
130	17.38	17.43	17.55	17.28	17.48	**17.43**	30.91
140	16.53	16.61	16.71	16.44	16.64	**16.59**	32.47
150	18.98	19.03	19.10	19.48	19.34	**19.19**	28.07
160	18.33	19.29	19.10	19.53	19.65	**19.18**	28.08
170	19.35	19.83	19.89	19.63	19.87	**19.71**	27.32
180	20.16	20.05	20.04	19.90	19.92	**20.01**	26.91
190	20.43	20.56	20.48	20.29	20.33	**20.42**	26.38
200	20.90	21.33	20.95	20.93	20.86	**20.99**	25.65
210	21.28	21.64	21.38	21.32	21.54	**21.43**	25.13
220	21.70	21.86	21.75	21.80	21.74	**21.77**	24.74
230	22.17	22.29	22.28	22.22	22.24	**22.24**	24.22
240	22.95	22.78	22.85	22.81	22.80	**22.84**	23.58

Table 6.67 Time and speedup of Ladder 215S(*N*) model with *N* = 20 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `scatter`

`Scatter` Heisenberg Ladder215S(20) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	134.13	134.28	134.34	134.30	134.31	**134.27**	1.00
10	28.71	28.65	28.64	28.49	28.44	**28.59**	4.70
20	17.27	17.21	17.25	18.45	18.50	**17.73**	7.57
30	13.06	12.94	13.49	13.75	13.73	**13.39**	10.02
40	11.83	11.84	11.41	11.73	11.74	**11.71**	11.47
50	10.81	10.81	10.51	10.80	10.50	**10.69**	12.56
60	10.16	10.14	10.16	10.18	10.15	**10.16**	13.22
70	9.95	9.97	9.92	9.90	9.91	**9.93**	13.52
80	9.65	9.81	9.62	9.65	9.59	**9.66**	13.90
90	9.92	10.01	9.92	9.89	9.88	**9.92**	13.53
100	10.00	10.07	9.99	10.12	9.98	**10.03**	13.39
110	10.26	10.25	10.18	10.29	10.18	**10.23**	13.12
120	10.55	10.54	10.54	10.53	10.51	**10.54**	12.74
130	9.80	9.66	9.77	9.87	9.86	**9.79**	13.71
140	10.16	10.24	10.37	10.23	10.27	**10.26**	13.09
150	10.63	10.70	10.67	10.71	10.72	**10.69**	12.56
160	11.04	11.26	11.14	11.32	11.14	**11.18**	12.01
170	11.50	11.73	11.52	11.71	11.54	**11.60**	11.58
180	11.93	11.94	11.88	12.02	11.99	**11.95**	11.24
190	12.61	12.60	12.70	12.66	12.66	**12.65**	10.62
200	12.98	13.00	13.01	13.04	13.04	**13.02**	10.32
210	13.51	13.45	13.48	13.50	13.57	**13.50**	9.94
220	13.89	13.89	13.91	13.90	13.99	**13.92**	9.65
230	14.36	14.28	14.31	14.34	14.46	**14.35**	9.36
240	15.14	14.74	14.81	14.83	14.95	**14.89**	9.01

Table 6.68 Time and speedup of Ladder 215S(*N*) model with *N* = 18 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `scatter`

`Scatter` Heisenberg Ladder215S(18) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	32.96	32.91	32.93	32.92	32.94	**32.93**	1.00
10	7.91	7.88	7.84	8.75	7.91	**8.06**	4.09
20	6.02	5.97	5.93	6.39	6.19	**6.10**	5.40
30	4.08	4.07	4.06	4.40	4.20	**4.16**	7.91
40	4.20	4.18	4.17	4.55	4.26	**4.27**	7.71
50	4.48	4.48	4.45	4.94	4.54	**4.58**	7.19
60	4.95	4.97	4.93	5.50	5.01	**5.07**	6.50
70	5.50	5.50	5.52	6.19	5.56	**5.65**	5.83
80	5.95	5.89	5.93	6.69	6.12	**6.12**	5.39
90	6.52	6.44	6.50	7.38	6.61	**6.69**	4.92
100	7.05	7.11	7.03	8.06	7.08	**7.27**	4.53
110	7.43	7.41	7.43	8.46	7.48	**7.64**	4.31
120	8.02	8.01	8.04	9.12	8.09	**8.26**	3.99
130	8.66	8.62	8.73	9.91	8.69	**8.92**	3.69
140	9.18	9.16	9.19	10.53	9.27	**9.47**	3.48
150	9.64	9.62	9.66	11.05	9.73	**9.94**	3.31
160	10.20	10.12	10.16	11.66	10.22	**10.47**	3.15
170	10.64	10.60	10.65	12.13	10.69	**10.94**	3.01
180	11.14	11.13	11.11	11.30	11.17	**11.17**	2.95
190	12.01	11.93	11.97	12.04	12.04	**11.99**	2.75
200	12.53	12.45	12.49	12.50	12.54	**12.50**	2.63
210	13.00	12.92	12.96	12.97	13.01	**12.97**	2.54
220	13.49	13.40	13.46	13.45	13.50	**13.46**	2.45
230	13.98	13.89	13.94	13.94	13.99	**13.95**	2.36
240	14.82	14.41	14.44	14.45	14.50	**14.53**	2.27

Table 6.69 Time and speedup of Ladder 215S(*N*) model with *N* = 16 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `scatter`

`Scatter` Heisenberg Ladder215S(16) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	8.30	8.25	8.26	8.31	8.28	**8.28**	1.00
10	3.08	3.18	2.98	3.26	3.09	**3.12**	2.66
20	3.07	3.09	3.08	3.22	3.05	**3.10**	2.67
30	3.34	3.30	3.30	3.52	3.31	**3.35**	2.47
40	3.70	3.70	3.73	3.72	3.71	**3.71**	2.23
50	4.23	4.23	4.24	4.25	4.27	**4.24**	1.95
60	4.89	4.88	4.87	4.90	4.92	**4.89**	1.69
70	5.48	5.50	5.47	5.53	5.50	**5.50**	1.51
80	5.85	6.02	6.02	5.87	6.03	**5.96**	1.39
90	6.61	6.60	6.62	6.62	6.61	**6.61**	1.25
100	7.18	7.18	7.02	7.20	7.20	**7.16**	1.16
110	7.84	8.57	7.86	7.88	7.88	**8.01**	1.03
120	8.48	9.00	8.48	8.50	8.50	**8.59**	0.96
130	9.23	9.23	9.22	9.23	9.23	**9.23**	0.90
140	9.77	9.78	9.77	9.80	9.80	**9.79**	0.85
150	10.31	10.31	10.30	10.32	10.32	**10.31**	0.80
160	10.84	10.85	10.84	10.86	10.86	**10.85**	0.76
170	11.37	11.38	11.38	11.41	11.38	**11.39**	0.73
180	11.94	11.94	11.93	11.94	11.95	**11.94**	0.69
190	12.70	12.66	12.65	12.68	12.68	**12.67**	0.65
200	13.24	13.22	13.21	13.21	13.23	**13.22**	0.63
210	13.55	13.77	13.78	13.78	13.78	**13.73**	0.60
220	14.34	14.31	14.00	14.31	14.35	**14.26**	0.58
230	14.85	14.84	14.83	14.50	14.85	**14.78**	0.56
240	15.72	15.37	15.07	15.38	15.38	**15.38**	0.54

Table 6.70 Time and speedup of Ladder 215S(*N*) model with *N* = 26 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `compact`

`Compact` Heisenberg Ladder215S(26) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	–	–	–	–	–	–	–
10	–	–	–	–	–	–	–
20	–	–	–	–	–	–	–
30	–	–	–	–	–	–	–
40	–	–	–	–	–	–	–
50	–	–	–	–	–	–	–
60	–	–	–	–	–	–	–
70	–	–	–	–	–	–	–
80	–	–	–	–	–	–	–
90	–	–	–	–	–	–	–
100	–	–	–	–	–	–	–
110	–	–	–	–	–	–	–
120	–	–	–	–	–	–	–
130	–	–	–	–	–	–	–
140	–	–	–	–	–	–	–
150	–	–	–	–	–	–	–
160	–	–	–	–	–	–	–
170	–	–	–	–	–	–	–
180	–	–	–	–	–	–	–
190	–	–	–	–	–	–	–
200	–	–	–	–	–	–	–
210	–	–	–	–	–	–	–
220	–	–	–	–	–	–	–
230	–	–	–	–	–	–	–
240	–	–	–	–	–	–	–

Table 6.71 Time and speedup of Ladder 215S(*N*) model with *N* = 24 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `compact`

`Compact` Heisenberg Ladder215S(24) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	2354.56	2355.89	2356.18	2355.27	2354.79	**2355.34**	1.00
10	696.75	696.73	696.99	696.89	696.15	**696.70**	3.38
20	374.73	374.63	374.72	374.78	374.93	**374.76**	6.28
30	255.16	257.06	255.28	255.18	255.14	**255.56**	9.22
40	196.31	196.27	196.29	196.29	196.41	**196.31**	12.00
50	159.43	159.43	159.48	159.81	159.45	**159.52**	14.77
60	136.81	136.78	136.81	131.77	131.75	**134.78**	17.47
70	118.38	118.37	118.41	118.38	118.41	**118.39**	19.89
80	104.40	104.39	104.40	104.39	104.41	**104.40**	22.56
90	95.73	95.72	95.71	92.18	95.71	**95.01**	24.79
100	87.98	87.97	87.96	87.98	87.97	**87.97**	26.77
110	81.41	81.44	81.42	81.44	81.43	**81.43**	28.93
120	76.35	76.40	76.38	76.35	76.36	**76.37**	30.84
130	71.99	71.96	71.96	71.96	71.98	**71.97**	32.73
140	67.85	67.83	67.85	67.83	67.84	**67.84**	34.72
150	64.29	64.19	64.17	64.18	64.20	**64.21**	36.68
160	60.61	59.49	60.60	60.60	60.60	**60.38**	39.01
170	60.25	60.37	60.37	60.37	60.38	**60.35**	39.03
180	55.97	56.01	56.00	56.03	56.02	**56.01**	42.06
190	53.35	53.36	53.36	53.35	53.34	**53.35**	44.15
200	51.89	51.90	51.90	51.92	51.91	**51.90**	45.38
210	50.44	50.41	50.43	50.43	50.42	**50.43**	46.71
220	49.49	49.52	49.55	49.55	49.53	**49.53**	47.56
230	47.70	47.70	47.71	47.73	47.72	**47.71**	49.37
240	46.30	46.25	46.26	46.27	46.25	**46.26**	50.91

Table 6.72 Time and speedup of Ladder 215S(*N*) model with *N* = 22 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `compact`

`Compact` Heisenberg Ladder215S(22) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	534.98	539.45	539.39	539.47	539.60	**538.58**	1.00
10	165.16	165.15	165.12	165.11	165.17	**165.14**	3.26
20	91.34	91.37	91.29	91.28	91.29	**91.32**	5.90
30	63.74	63.73	63.75	63.76	63.74	**63.74**	8.45
40	49.34	49.33	49.32	49.33	49.31	**49.33**	10.92
50	41.36	41.35	41.40	41.36	41.38	**41.37**	13.02
60	35.25	35.31	35.22	35.23	35.25	**35.25**	15.28
70	30.79	30.78	30.78	30.77	30.78	**30.78**	17.50
80	26.77	26.78	26.76	26.78	26.75	**26.77**	20.12
90	24.44	24.40	24.40	24.41	24.39	**24.41**	22.07
100	22.00	21.97	22.00	21.97	21.98	**21.99**	24.50
110	20.07	20.05	20.06	20.07	20.05	**20.06**	26.85
120	18.71	18.68	18.69	18.66	18.69	**18.69**	28.82
130	17.74	17.74	17.75	17.72	17.74	**17.74**	30.36
140	16.61	16.63	16.61	16.62	16.63	**16.62**	32.40
150	18.82	18.85	18.83	18.82	18.82	**18.83**	28.60
160	18.49	18.47	18.47	18.47	18.50	**18.48**	29.14
170	18.71	18.67	18.67	18.67	18.69	**18.68**	28.83
180	18.63	18.62	18.62	18.59	18.60	**18.61**	28.94
190	18.51	18.46	18.50	18.47	18.47	**18.48**	29.15
200	18.79	18.78	18.82	18.77	18.82	**18.80**	28.65
210	19.05	19.01	19.03	18.99	19.02	**19.02**	28.32
220	19.50	19.52	19.50	19.50	19.50	**19.50**	27.61
230	19.93	19.90	19.90	19.91	19.90	**19.91**	27.05
240	20.48	20.31	20.23	20.29	20.31	**20.32**	26.50

Table 6.73 Time and speedup of Ladder 215S(*N*) model with *N* = 20 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `compact`

`Compact` Heisenberg Ladder215S(20) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	134.13	134.28	134.34	134.30	134.31	**134.27**	1.00
10	42.83	42.80	42.78	42.88	42.89	**42.83**	3.13
20	24.12	24.05	24.06	24.03	24.03	**24.06**	5.58
30	16.20	16.14	16.12	16.19	16.21	**16.17**	8.30
40	13.77	13.72	13.71	13.75	13.76	**13.74**	9.77
50	12.58	12.54	12.55	12.59	12.58	**12.57**	10.68
60	11.87	11.82	11.80	11.86	11.85	**11.84**	11.34
70	11.45	11.43	11.39	11.42	11.47	**11.43**	11.74
80	11.17	11.11	11.08	11.11	11.11	**11.12**	12.08
90	11.39	11.33	11.31	11.33	11.34	**11.34**	11.84
100	11.57	11.53	11.52	11.53	11.53	**11.54**	11.64
110	11.86	11.75	11.74	11.79	11.78	**11.78**	11.40
120	12.11	12.04	12.03	12.09	12.09	**12.07**	11.12
130	11.56	11.54	11.55	11.64	11.66	**11.59**	11.58
140	12.11	12.09	12.10	12.18	12.18	**12.13**	11.07
150	12.72	12.69	12.70	12.76	12.76	**12.73**	10.55
160	13.38	13.38	13.37	13.48	13.46	**13.41**	10.01
170	14.12	14.10	14.06	14.18	14.16	**14.12**	9.51
180	14.99	14.89	14.91	14.96	14.95	**14.94**	8.99
190	15.80	15.73	15.71	15.74	15.71	**15.74**	8.53
200	16.76	16.66	16.67	16.67	16.73	**16.70**	8.04
210	17.88	17.77	17.79	17.77	17.81	**17.80**	7.54
220	18.92	18.79	18.81	18.61	18.85	**18.80**	7.14
230	20.14	20.00	20.00	20.00	20.06	**20.04**	6.70
240	21.34	20.90	20.90	20.89	20.94	**20.99**	6.40

Table 6.74 Time and speedup of Ladder 215S(*N*) model with *N* = 18 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `compact`

`Compact` Heisenberg Ladder215S(18) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	32.96	32.91	32.93	32.92	32.94	**32.93**	1.00
10	11.28	11.25	11.26	11.26	11.24	**11.26**	2.93
20	8.30	8.29	8.28	8.26	8.24	**8.27**	3.98
30	6.84	6.84	6.86	6.82	6.83	**6.84**	4.82
40	6.70	6.70	6.72	6.74	6.72	**6.72**	4.90
50	6.99	6.98	7.00	6.97	6.98	**6.98**	4.72
60	7.22	7.22	7.21	7.27	7.25	**7.24**	4.55
70	7.68	7.63	7.66	7.66	7.69	**7.66**	4.30
80	8.14	8.12	8.14	8.18	8.16	**8.15**	4.04
90	8.63	8.59	8.62	8.68	8.69	**8.64**	3.81
100	9.14	9.13	9.12	9.14	9.12	**9.13**	3.61
110	9.73	9.75	9.74	9.79	9.79	**9.76**	3.37
120	10.44	10.42	10.44	10.48	10.47	**10.45**	3.15
130	11.14	11.13	11.13	11.17	11.17	**11.15**	2.95
140	11.83	11.79	11.84	11.85	11.84	**11.83**	2.78
150	12.64	12.63	12.64	12.67	12.64	**12.64**	2.60
160	13.41	13.42	13.42	13.45	13.46	**13.43**	2.45
170	14.40	14.40	14.43	14.46	14.37	**14.41**	2.28
180	15.43	15.38	15.38	15.43	15.34	**15.39**	2.14
190	16.51	16.47	16.50	16.50	16.40	**16.47**	2.00
200	17.58	17.48	17.53	17.55	17.44	**17.52**	1.88
210	18.85	18.80	18.85	18.82	18.72	**18.81**	1.75
220	20.08	19.96	19.98	19.96	19.88	**19.97**	1.65
230	21.34	21.34	21.36	21.31	21.24	**21.32**	1.54
240	22.73	22.40	22.43	22.43	22.34	**22.47**	1.47

Table 6.75 Time and speedup of Ladder 215S(*N*) model with *N* = 16 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `compact`

`Compact` Heisenberg Ladder215S(16) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	8.30	8.25	8.26	8.31	8.28	**8.28**	1.00
10	4.22	4.23	4.25	4.29	4.29	**4.26**	1.95
20	3.92	3.93	3.95	4.01	3.95	**3.95**	2.10
30	4.10	4.11	4.12	4.15	4.14	**4.13**	2.01
40	4.44	4.43	4.45	4.50	4.47	**4.46**	1.86
50	4.80	4.86	4.88	4.91	4.90	**4.87**	1.70
60	5.29	5.29	5.32	5.35	5.33	**5.32**	1.56
70	5.79	5.77	5.77	5.77	5.82	**5.78**	1.43
80	6.22	6.24	6.23	6.27	6.27	**6.25**	1.33
90	6.83	6.84	6.80	6.84	6.85	**6.83**	1.21
100	7.36	7.37	7.37	7.40	7.39	**7.38**	1.12
110	7.98	7.95	7.96	7.97	7.97	**7.96**	1.04
120	8.56	8.55	8.57	8.59	8.57	**8.57**	0.97
130	9.24	9.22	9.23	9.21	9.25	**9.23**	0.90
140	9.89	9.88	9.89	9.89	9.70	**9.85**	0.84
150	10.61	10.60	10.62	10.57	10.60	**10.60**	0.78
160	11.28	11.27	11.30	11.28	11.28	**11.28**	0.73
170	12.08	12.06	12.08	12.07	12.07	**12.07**	0.69
180	12.83	12.80	12.84	12.79	12.81	**12.81**	0.65
190	13.76	13.75	13.76	13.73	13.10	**13.62**	0.61
200	14.01	14.65	14.32	14.60	14.64	**14.44**	0.57
210	15.75	15.72	15.74	15.32	15.61	**15.63**	0.53
220	16.74	16.74	16.74	16.61	16.68	**16.70**	0.50
230	17.92	17.91	17.91	17.84	17.86	**17.89**	0.46
240	19.27	18.94	18.95	18.86	18.91	**18.99**	0.44

Table 6.76 Time and speedup of Ladder 215S(*N*) model with *N* = 26 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `balanced`

`Balanced` Heisenberg Ladder215S(26) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	–	–	–	–	–	–	–
10	–	–	–	–	–	–	–
20	–	–	–	–	–	–	–
30	–	–	–	–	–	–	–
40	–	–	–	–	–	–	–
50	–	–	–	–	–	–	–
60	–	–	–	–	–	–	–
70	–	–	–	–	–	–	–
80	–	–	–	–	–	–	–
90	–	–	–	–	–	–	–
100	–	–	–	–	–	–	–
110	–	–	–	–	–	–	–
120	–	–	–	–	–	–	–
130	–	–	–	–	–	–	–
140	–	–	–	–	–	–	–
150	–	–	–	–	–	–	–
160	–	–	–	–	–	–	–
170	–	–	–	–	–	–	–
180	–	–	–	–	–	–	–
190	–	–	–	–	–	–	–
200	–	–	–	–	–	–	–
210	–	–	–	–	–	–	–
220	–	–	–	–	–	–	–
230	–	–	–	–	–	–	–
240	–	–	–	–	–	–	–

Table 6.77 Time and speedup of Ladder 215S(*N*) model with *N* = 24 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `balanced`

`Balanced` Heisenberg Ladder215S(24) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	2354.56	2355.89	2356.18	2355.27	2354.79	**2355.34**	1.00
10	544.79	547.65	536.39	555.49	563.53	**549.57**	4.29
20	337.22	326.28	327.61	305.80	331.90	**325.76**	7.23
30	234.18	242.23	245.66	240.92	239.49	**240.50**	9.79
40	177.17	178.09	185.46	186.77	187.82	**183.06**	12.87
50	152.39	152.27	153.14	152.21	157.87	**153.58**	15.34
60	129.28	129.17	129.64	130.35	129.71	**129.63**	18.17
70	112.29	113.49	116.21	116.36	114.60	**114.59**	20.55
80	95.16	101.43	102.31	101.09	105.85	**101.17**	23.28
90	91.01	95.79	95.52	91.45	93.19	**93.39**	25.22
100	81.47	88.15	83.16	81.98	87.50	**84.45**	27.89
110	79.69	79.85	80.87	79.77	79.95	**80.03**	29.43
120	76.19	76.03	75.43	75.75	74.72	**75.62**	31.15
130	72.98	75.75	73.22	67.41	74.54	**72.78**	32.36
140	68.48	69.32	69.10	67.82	70.57	**69.06**	34.11
150	65.84	63.31	66.80	63.43	64.40	**64.76**	36.37
160	60.63	61.10	61.72	61.98	61.95	**61.47**	38.31
170	62.35	61.52	61.94	59.67	61.16	**61.33**	38.41
180	58.46	58.24	58.05	57.99	57.69	**58.09**	40.55
190	56.87	57.18	58.12	56.59	57.55	**57.26**	41.13
200	54.27	54.54	54.99	53.93	54.59	**54.46**	43.25
210	51.57	52.45	51.98	52.37	51.47	**51.96**	45.33
220	51.63	51.64	51.67	51.25	51.98	**51.63**	45.62
230	47.50	47.66	47.56	47.51	47.79	**47.60**	49.48
240	46.29	46.23	46.22	46.25	46.21	**46.24**	50.94

Table 6.78 Time and speedup of Ladder 215S(*N*) model with *N* = 22 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `balanced`

`Balanced` Heisenberg Ladder215S(22) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	534.98	539.45	539.39	539.47	539.60	**538.58**	1.00
10	124.48	121.27	131.55	127.16	119.93	**124.88**	4.31
20	68.84	75.78	78.13	76.67	73.68	**74.62**	7.22
30	54.59	53.60	55.13	54.42	53.51	**54.25**	9.93
40	41.42	39.47	41.24	40.88	40.26	**40.65**	13.25
50	34.28	33.11	33.34	33.75	33.61	**33.62**	16.02
60	29.46	28.81	29.55	29.78	29.41	**29.40**	18.32
70	26.44	27.75	27.55	26.56	26.40	**26.94**	19.99
80	21.04	24.07	23.12	23.30	23.15	**22.94**	23.48
90	21.69	23.27	21.65	21.88	21.70	**22.04**	24.44
100	20.04	21.56	20.94	19.23	19.90	**20.33**	26.49
110	18.58	19.86	19.32	19.42	19.50	**19.34**	27.85
120	18.34	18.48	18.00	18.28	18.46	**18.31**	29.41
130	17.82	17.84	17.55	17.59	17.75	**17.71**	30.41
140	16.63	16.72	16.81	16.79	16.89	**16.77**	32.12
150	63.99	63.97	63.89	63.66	63.57	**63.82**	8.44
160	70.50	70.35	70.98	70.86	69.90	**70.52**	7.64
170	79.78	80.28	80.09	79.96	79.65	**79.95**	6.74
180	91.59	90.93	90.74	91.24	91.67	**91.23**	5.90
190	91.01	91.31	89.98	90.08	90.64	**90.60**	5.94
200	113.74	113.06	112.10	112.97	112.55	**112.88**	4.77
210	122.83	122.59	122.96	122.95	122.36	**122.74**	4.39
220	115.38	115.92	115.48	115.85	115.30	**115.59**	4.66
230	129.02	128.81	129.01	129.39	129.74	**129.19**	4.17
240	141.84	140.37	141.19	140.81	140.96	**141.03**	3.82

Table 6.79 Time and speedup of Ladder 215S(*N*) model with *N* = 20 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `balanced`

`Balanced` Heisenberg Ladder215S(20) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	134.13	134.28	134.34	134.30	134.31	**134.27**	1.00
10	33.62	33.52	31.11	30.65	31.62	**32.10**	4.18
20	18.69	18.68	18.05	18.54	17.42	**18.28**	7.35
30	13.67	13.13	13.25	12.87	13.09	**13.20**	10.17
40	14.99	15.29	15.31	15.27	15.17	**15.21**	8.83
50	15.23	15.27	15.22	15.30	15.27	**15.26**	8.80
60	16.12	16.07	16.07	16.04	16.08	**16.08**	8.35
70	29.06	29.12	28.63	28.56	29.17	**28.91**	4.65
80	33.37	34.14	33.77	33.57	33.79	**33.73**	3.98
90	43.54	42.66	42.75	42.89	42.68	**42.90**	3.13
100	49.92	50.04	50.30	49.16	49.46	**49.78**	2.70
110	50.64	51.02	50.91	50.63	51.28	**50.89**	2.64
120	51.61	51.82	51.67	51.35	51.45	**51.58**	2.60
130	58.18	58.78	57.71	58.41	58.52	**58.32**	2.30
140	50.32	50.78	50.21	49.67	51.03	**50.40**	2.66
150	59.73	59.87	59.67	59.86	60.01	**59.83**	2.24
160	67.27	67.14	67.70	67.38	67.72	**67.44**	1.99
170	69.03	69.23	69.29	69.30	69.08	**69.19**	1.94
180	70.23	70.73	71.02	70.96	71.46	**70.88**	1.89
190	65.19	65.88	65.69	65.80	65.99	**65.71**	2.04
200	66.38	66.65	66.48	67.37	66.71	**66.72**	2.01
210	75.60	74.88	74.89	75.03	75.52	**75.18**	1.79
220	81.43	81.53	82.35	81.62	81.72	**81.73**	1.64
230	84.52	83.82	84.47	83.75	83.69	**84.05**	1.60
240	86.32	85.91	86.19	85.81	85.78	**86.00**	1.56

Table 6.80 Time and speedup of Ladder 215S(*N*) model with *N* = 18 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to balanced

Balanced Heisenberg Ladder215S(18) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	32.96	32.91	32.93	32.92	32.94	**32.93**	1.00
10	9.24	9.49	8.83	8.74	8.71	**9.00**	3.66
20	7.48	7.37	7.25	7.19	7.36	**7.33**	4.49
30	5.62	5.59	5.46	5.51	5.55	**5.55**	5.94
40	6.25	6.36	6.25	6.27	6.26	**6.28**	5.25
50	7.38	7.40	7.35	7.41	7.46	**7.40**	4.45
60	8.95	9.06	8.98	9.13	9.00	**9.02**	3.65
70	14.49	14.03	14.12	14.15	14.16	**14.19**	2.32
80	15.22	15.33	15.38	15.31	15.53	**15.35**	2.15
90	16.70	16.60	16.41	16.46	16.39	**16.51**	1.99
100	17.63	17.75	17.76	17.82	17.86	**17.76**	1.85
110	19.50	19.28	19.32	19.71	19.34	**19.43**	1.70
120	20.66	20.45	20.55	20.70	20.57	**20.59**	1.60
130	22.51	22.67	22.74	22.59	22.51	**22.60**	1.46
140	24.23	24.48	24.54	24.46	24.58	**24.46**	1.35
150	25.76	25.80	25.77	25.85	26.49	**25.93**	1.27
160	28.28	27.99	27.91	27.90	28.54	**28.12**	1.17
170	29.95	29.63	29.63	29.65	29.60	**29.69**	1.11
180	31.09	31.05	31.14	31.03	31.20	**31.10**	1.06
190	31.45	31.52	31.18	31.67	31.44	**31.45**	1.05
200	32.89	32.55	32.66	32.58	32.65	**32.67**	1.01
210	34.25	34.81	34.54	34.40	34.39	**34.48**	0.96
220	36.45	36.45	36.36	36.38	36.50	**36.43**	0.90
230	39.17	38.92	38.98	38.82	38.71	**38.92**	0.85
240	40.83	40.76	40.39	40.65	40.87	**40.70**	0.81

Table 6.81 Time and speedup of Ladder 215S(*N*) model with *N* = 16 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to balanced

Balanced Heisenberg Ladder215S(16) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	8.30	8.25	8.26	8.31	8.28	**8.28**	1.00
10	3.46	3.37	3.38	3.37	3.49	**3.42**	2.42
20	3.49	3.49	3.51	3.51	3.59	**3.52**	2.35
30	4.24	4.24	4.24	4.30	4.30	**4.26**	1.94
40	5.36	5.39	5.36	5.45	5.39	**5.39**	1.54
50	6.88	6.96	6.72	6.96	6.89	**6.88**	1.20
60	8.92	8.75	8.93	8.95	8.71	**8.85**	0.94
70	11.08	11.16	11.09	11.13	10.89	**11.07**	0.75
80	12.42	12.27	12.26	12.45	12.34	**12.35**	0.67
90	13.70	13.65	13.62	13.69	13.70	**13.67**	0.61
100	15.13	15.16	15.22	15.07	15.76	**15.27**	0.54
110	17.03	16.87	17.30	16.86	16.84	**16.98**	0.49
120	18.23	18.22	18.26	18.22	18.22	**18.23**	0.45
130	20.12	19.99	20.07	20.00	20.10	**20.06**	0.41
140	21.83	21.91	21.88	21.94	21.83	**21.88**	0.38
150	23.32	22.79	23.43	23.31	22.83	**23.14**	0.36
160	25.50	25.35	25.42	24.91	25.44	**25.32**	0.33
170	27.10	27.19	27.11	27.19	27.10	**27.14**	0.31
180	28.96	28.95	29.15	28.95	29.02	**29.00**	0.29
190	30.27	30.28	30.27	30.28	29.66	**30.15**	0.27
200	31.76	31.77	31.86	31.84	31.72	**31.79**	0.26
210	34.02	30.79	34.16	34.18	34.19	**33.47**	0.25
220	36.29	36.21	36.28	35.59	36.50	**36.18**	0.23
230	38.45	38.40	38.42	38.40	37.61	**38.26**	0.22
240	40.34	40.01	39.43	40.23	40.11	**40.03**	0.21

6.2.5 Heisenberg Ladder 315C on Xeon Phi

Table 6.82 Time and speedup of Ladder 315C(*N*) model with *N* = 24 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to scatter

`Scatter` Heisenberg Ladder215S(24) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	2566.78	2569.76	2570.14	2568.15	2569.88	**2568.94**	1.00
10	614.93	640.37	625.17	624.50	611.34	**623.26**	4.12
20	350.78	352.20	376.72	358.77	349.76	**357.64**	7.18
30	252.83	228.58	259.25	270.46	262.84	**254.79**	10.08
40	206.77	195.42	205.11	206.07	207.56	**204.19**	12.58
50	162.46	166.52	164.73	159.60	161.28	**162.92**	15.77
60	144.32	146.04	142.27	142.51	143.28	**143.68**	17.88
70	127.30	127.16	131.05	126.79	127.17	**127.89**	20.09
80	117.72	112.35	112.89	109.83	113.40	**113.24**	22.69
90	104.20	101.01	101.67	104.51	110.03	**104.28**	24.63
100	96.68	91.72	90.98	92.65	96.41	**93.69**	27.42
110	85.32	89.89	90.45	86.10	88.58	**88.07**	29.17
120	83.32	83.66	84.93	84.97	85.50	**84.48**	30.41
130	84.45	81.31	81.43	80.70	80.46	**81.67**	31.45
140	76.70	77.09	76.78	75.99	74.88	**76.29**	33.67
150	74.31	72.08	72.25	71.31	72.58	**72.51**	35.43
160	71.78	75.82	67.85	68.42	71.56	**71.09**	36.14
170	69.87	69.90	70.30	69.52	66.16	**69.15**	37.15
180	65.68	66.69	65.82	66.02	68.75	**66.59**	38.58
190	65.22	64.83	65.25	65.24	64.45	**65.00**	39.52
200	61.12	60.50	61.53	61.88	62.35	**61.48**	41.79
210	58.41	58.56	59.01	58.95	59.08	**58.80**	43.69
220	56.98	57.09	57.26	57.59	58.49	**57.48**	44.69
230	53.54	53.67	53.76	53.61	53.71	**53.66**	47.87
240	53.15	53.11	52.92	53.08	53.32	**53.12**	48.36

Table 6.83 Time and speedup of Ladder 315C(N) model with N = 18 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `scatter`

`Scatter` Heisenberg Ladder215S(18) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	35.64	35.67	35.79	35.69	35.87	**35.73**	1.00
10	8.41	8.39	8.44	8.31	8.34	**8.38**	4.27
20	6.40	6.39	6.42	6.33	5.98	**6.30**	5.67
30	5.94	5.93	5.94	6.03	5.90	**5.95**	6.01
40	4.69	4.67	4.69	4.81	4.71	**4.72**	7.58
50	5.38	5.31	5.31	5.39	5.33	**5.35**	6.68
60	6.28	6.25	6.26	6.27	6.23	**6.26**	5.71
70	7.06	7.05	7.06	7.07	7.06	**7.06**	5.06
80	7.84	7.83	7.93	7.75	7.75	**7.82**	4.57
90	8.58	8.50	8.64	8.28	8.50	**8.50**	4.20
100	9.18	9.15	9.25	9.19	9.15	**9.18**	3.89
110	9.91	9.85	9.93	9.85	9.85	**9.88**	3.62
120	10.65	10.68	10.75	10.68	10.66	**10.68**	3.34
130	11.56	11.55	11.69	11.57	11.58	**11.59**	3.08
140	12.34	12.34	12.47	12.31	12.33	**12.36**	2.89
150	13.08	13.03	13.13	13.01	13.02	**13.06**	2.74
160	13.75	13.76	13.80	13.74	13.73	**13.76**	2.60
170	14.50	14.47	14.48	14.50	14.44	**14.48**	2.47
180	15.13	15.13	15.13	15.14	15.18	**15.14**	2.36
190	16.12	16.13	16.13	16.13	16.12	**16.13**	2.22
200	16.85	16.82	16.83	16.85	16.84	**16.84**	2.12
210	17.57	17.50	17.51	17.54	17.53	**17.53**	2.04
220	18.29	18.23	18.27	18.25	17.53	**18.12**	1.97
230	18.97	18.96	17.86	18.96	17.12	**18.37**	1.94
240	20.02	19.69	19.71	19.70	19.70	**19.76**	1.81

Table 6.84 Time and speedup of Ladder 315C(*N*) model with *N* = 24 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `compact`

`Compact` Heisenberg Ladder315C(24) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	2566.78	2569.76	2570.14	2568.15	2569.88	**2568.94**	1.00
10	784.03	784.53	784.71	785.20	786.81	**785.06**	3.27
20	403.16	403.20	403.24	403.22	403.18	**403.20**	6.37
30	277.42	277.70	274.23	274.24	274.38	**275.59**	9.32
40	211.65	211.68	211.66	211.59	211.62	**211.64**	12.14
50	170.29	170.31	170.29	170.30	170.30	**170.30**	15.09
60	146.63	146.68	146.64	146.63	146.65	**146.65**	17.52
70	128.17	128.04	128.11	128.08	128.07	**128.09**	20.05
80	113.35	113.41	113.40	113.41	113.40	**113.39**	22.65
90	102.64	102.63	102.64	102.66	102.64	**102.64**	25.03
100	93.82	93.80	93.81	93.81	93.81	**93.81**	27.38
110	87.54	87.52	87.53	87.53	87.51	**87.52**	29.35
120	81.63	81.64	81.65	81.65	81.63	**81.64**	31.47
130	76.88	76.85	76.83	76.84	76.80	**76.84**	33.43
140	72.83	72.83	72.83	72.82	72.81	**72.82**	35.28
150	69.02	69.01	69.02	69.04	68.98	**69.01**	37.22
160	65.04	65.08	65.10	65.07	65.06	**65.07**	39.48
170	64.30	64.18	64.21	64.31	64.28	**64.26**	39.98
180	59.47	59.49	59.51	59.50	59.48	**59.49**	43.18
190	57.20	57.28	57.26	57.23	57.27	**57.25**	44.87
200	55.16	55.20	55.25	55.18	55.20	**55.20**	46.54
210	53.44	53.49	53.50	53.51	53.53	**53.49**	48.02
220	52.56	52.57	52.59	52.61	52.60	**52.59**	48.85
230	50.44	50.48	50.52	50.44	50.53	**50.48**	50.89
240	49.07	49.02	48.99	49.04	49.00	**49.02**	52.40

Table 6.85 Time and speedup of Ladder 315C(N) model with N = 18 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `compact`

`Compact` Heisenberg Ladder315C(18) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	35.64	35.67	35.79	35.69	35.87	**35.73**	1.00
10	11.88	11.90	11.86	11.79	11.79	**11.84**	3.02
20	8.57	8.59	8.57	8.47	8.48	**8.54**	4.19
30	7.84	7.81	7.81	7.73	7.73	**7.78**	4.59
40	6.94	6.95	6.93	6.84	6.85	**6.90**	5.18
50	7.05	7.04	7.00	6.98	6.96	**7.01**	5.10
60	7.36	7.37	7.35	7.32	7.33	**7.35**	4.86
70	7.79	7.75	7.76	7.67	7.68	**7.73**	4.62
80	8.21	7.77	8.18	8.15	8.17	**8.10**	4.41
90	8.74	7.96	8.73	8.73	8.72	**8.58**	4.17
100	9.23	8.76	9.22	9.23	9.25	**9.14**	3.91
110	9.86	9.85	9.85	9.85	9.85	**9.85**	3.63
120	10.45	10.46	10.44	10.23	10.43	**10.40**	3.44
130	11.11	11.12	11.10	11.11	11.11	**11.11**	3.22
140	11.83	11.83	11.83	11.82	11.82	**11.83**	3.02
150	12.65	12.64	12.61	12.63	12.61	**12.63**	2.83
160	13.42	13.43	13.43	13.40	13.41	**13.42**	2.66
170	14.40	14.39	14.38	14.36	14.38	**14.38**	2.48
180	15.38	15.33	15.30	15.30	15.31	**15.32**	2.33
190	16.37	16.37	16.37	16.32	16.38	**16.36**	2.18
200	17.44	17.43	17.39	17.39	17.42	**17.41**	2.05
210	18.68	18.67	18.68	18.66	18.72	**18.68**	1.91
220	17.57	19.84	19.84	19.84	19.86	**19.39**	1.84
230	21.17	21.11	21.20	21.19	21.18	**21.17**	1.69
240	22.58	22.14	22.26	22.27	22.26	**22.30**	1.60

Table 6.86 Time and speedup of Ladder 315C(N) model with N = 24 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `balanced`

`Balanced` Heisenberg Ladder315C(24) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	2566.78	2569.76	2570.14	2568.15	2569.88	**2568.94**	1.00
10	596.15	633.40	636.07	615.49	630.02	**622.22**	4.13
20	350.59	357.93	365.96	369.76	349.31	**358.71**	7.16
30	252.62	269.55	259.16	258.60	275.90	**263.17**	9.76
40	207.07	208.64	209.85	200.92	199.76	**205.25**	12.52
50	168.33	157.19	170.30	165.52	166.21	**165.51**	15.52
60	139.29	139.73	147.62	144.22	141.36	**142.44**	18.03
70	124.46	123.70	125.36	127.71	125.76	**125.40**	20.49
80	105.77	109.38	117.61	110.90	111.20	**110.97**	23.15
90	99.18	99.15	100.23	100.79	100.80	**100.03**	25.68
100	90.78	89.29	90.61	96.26	94.30	**92.25**	27.85
110	86.28	87.58	85.01	84.92	84.69	**85.70**	29.98
120	80.59	81.46	82.53	80.90	81.17	**81.33**	31.59
130	77.65	78.81	78.58	79.70	77.82	**78.51**	32.72
140	73.57	74.46	74.70	73.17	74.48	**74.08**	34.68
150	70.27	70.38	69.11	71.53	70.61	**70.38**	36.50
160	67.43	67.70	67.06	68.21	67.89	**67.66**	37.97
170	66.25	66.22	66.09	66.81	66.48	**66.37**	38.71
180	62.67	61.79	62.09	62.58	61.94	**62.21**	41.29
190	61.83	61.90	61.49	61.46	61.82	**61.70**	41.64
200	58.06	59.20	58.59	59.16	58.90	**58.78**	43.70
210	55.98	55.54	55.65	55.96	56.02	**55.83**	46.01
220	54.98	55.57	55.36	55.05	55.18	**55.23**	46.51
230	50.70	50.42	50.75	50.66	50.47	**50.60**	50.77
240	49.19	49.08	49.10	49.08	49.09	**49.11**	52.31

Table 6.87 Time and speedup of Ladder 315C(N) model with N = 18 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `balanced`

`Balanced` Heisenberg Ladder315C(18) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	35.64	35.67	35.79	35.69	35.87	**35.73**	1.00
10	10.01	9.20	9.46	10.08	9.31	**9.61**	3.72
20	7.66	7.54	7.20	7.77	7.60	**7.56**	4.73
30	7.68	7.56	7.75	7.71	7.60	**7.66**	4.66
40	6.81	6.93	6.91	7.02	6.91	**6.92**	5.17
50	8.47	8.41	8.49	8.41	8.52	**8.46**	4.22
60	10.64	10.59	10.57	10.74	10.68	**10.64**	3.36
70	16.45	16.50	16.61	16.44	16.49	**16.50**	2.17
80	17.78	17.86	17.71	17.83	17.80	**17.80**	2.01
90	19.10	18.95	19.37	19.20	19.22	**19.17**	1.86
100	20.62	20.56	20.58	20.90	20.71	**20.67**	1.73
110	22.63	22.35	22.61	22.77	22.81	**22.64**	1.58
120	23.95	24.07	24.02	24.31	23.81	**24.03**	1.49
130	26.26	26.14	26.27	26.14	26.04	**26.17**	1.37
140	27.80	27.96	28.26	28.05	25.03	**27.42**	1.30
150	29.69	29.91	30.16	28.85	29.97	**29.72**	1.20
160	32.34	31.74	32.14	32.10	32.22	**32.11**	1.11
170	33.70	33.39	33.88	33.92	33.89	**33.76**	1.06
180	35.37	35.44	35.52	35.25	35.48	**35.41**	1.01
190	35.84	35.72	35.63	35.87	35.92	**35.80**	1.00
200	38.06	37.57	37.53	37.53	37.77	**37.69**	0.95
210	40.38	40.43	40.46	40.43	35.67	**39.47**	0.91
220	42.90	42.84	42.62	42.83	42.76	**42.79**	0.84
230	45.27	41.10	45.77	45.64	45.49	**44.66**	0.80
240	47.92	47.36	47.63	47.48	47.63	**47.60**	0.75

6.2.6 Heisenberg Ladder 315S on Xeon Phi

Table 6.88 Time and speedup of Ladder 315S(*N*) model with *N* = 24 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `scatter`

`Scatter` Heisenberg Ladder315S(24) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	2565.74	2569.90	2569.13	2569.56	2570.11	**2568.89**	1.00
10	634.26	639.13	607.93	620.80	596.48	**619.72**	4.15
20	376.11	385.95	366.75	355.90	338.40	**364.62**	7.05
30	262.39	257.53	262.53	267.41	264.03	**262.78**	9.78
40	202.56	208.22	208.06	208.26	209.03	**207.23**	12.40
50	156.74	166.11	166.98	164.72	161.10	**163.13**	15.75
60	140.80	150.12	142.74	142.53	144.15	**144.07**	17.83
70	127.88	133.37	127.35	127.86	127.99	**128.89**	19.93
80	119.86	117.38	113.98	116.31	111.49	**115.80**	22.18
90	97.99	107.04	102.54	104.67	102.16	**102.88**	24.97
100	97.07	92.93	91.55	95.22	95.03	**94.36**	27.22
110	86.42	89.37	85.07	87.30	89.35	**87.50**	29.36
120	84.68	83.71	84.22	83.72	86.11	**84.49**	30.41
130	81.39	81.86	80.09	81.00	86.99	**82.26**	31.23
140	76.46	84.02	69.73	77.06	77.25	**76.90**	33.40
150	71.35	72.13	77.31	72.55	72.66	**73.20**	35.09
160	68.38	78.77	78.80	68.86	72.77	**73.52**	34.94
170	67.46	66.58	70.87	70.59	69.36	**68.97**	37.25
180	66.24	66.92	65.38	66.26	65.76	**66.11**	38.86
190	65.13	57.42	65.01	64.55	65.22	**63.46**	40.48
200	62.26	62.58	61.17	61.49	61.37	**61.77**	41.59
210	58.69	58.95	58.70	58.16	59.03	**58.70**	43.76
220	57.74	57.46	57.49	57.55	55.04	**57.06**	45.02
230	53.87	53.95	53.88	53.97	53.79	**53.89**	47.67
240	53.21	53.00	53.23	53.08	53.19	**53.14**	48.34

Table 6.89 Time and speedup of Ladder 315S(N) model with N = 18 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `scatter`

`Scatter` Heisenberg Ladder315S(18) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	35.80	35.71	35.81	35.72	35.88	**35.78**	1.00
10	8.30	8.49	8.74	8.34	8.29	**8.43**	4.24
20	6.38	6.62	6.44	6.31	6.30	**6.41**	5.58
30	5.78	5.92	5.81	5.74	5.86	**5.82**	6.15
40	4.24	4.37	4.25	4.25	4.31	**4.28**	8.35
50	4.49	4.29	4.26	4.50	4.49	**4.40**	8.12
60	4.73	4.93	4.91	4.55	4.88	**4.80**	7.45
70	5.46	5.40	5.48	5.50	5.49	**5.46**	6.55
80	5.10	5.89	5.84	5.85	5.85	**5.71**	6.27
90	6.42	6.49	6.41	6.44	6.43	**6.44**	5.56
100	6.84	6.88	6.20	6.60	6.86	**6.68**	5.36
110	7.03	6.77	7.29	7.31	7.30	**7.14**	5.01
120	7.73	7.87	7.87	7.74	7.89	**7.82**	4.58
130	8.60	8.60	8.62	8.62	8.63	**8.61**	4.15
140	9.20	8.29	8.97	8.99	9.15	**8.92**	4.01
150	9.60	9.60	9.60	9.60	9.60	**9.60**	3.73
160	10.05	9.48	10.08	10.09	10.07	**9.95**	3.59
170	10.51	10.54	10.51	10.50	10.51	**10.51**	3.40
180	10.34	10.96	10.95	10.95	11.02	**10.84**	3.30
190	11.73	11.74	11.74	11.70	11.73	**11.73**	3.05
200	12.17	12.17	12.17	12.15	12.18	**12.17**	2.94
210	9.72	12.62	12.63	12.62	12.62	**12.04**	2.97
220	13.07	13.06	13.08	13.05	13.07	**13.07**	2.74
230	13.51	13.52	13.52	13.51	12.48	**13.31**	2.69
240	14.36	14.02	14.01	14.04	14.02	**14.09**	2.54

Table 6.90 Time and speedup of Ladder 315S(N) model with N = 24 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `compact`

`Compact` Heisenberg Ladder315S(24) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	2565.74	2569.90	2569.13	2569.56	2570.11	**2568.89**	1.00
10	763.46	762.06	769.37	790.99	785.94	**774.36**	3.32
20	402.93	402.86	402.86	403.52	403.42	**403.12**	6.37
30	273.55	273.65	273.61	274.46	274.27	**273.91**	9.38
40	211.63	211.75	211.66	211.73	211.85	**211.72**	12.13
50	170.08	170.13	170.07	170.47	170.42	**170.23**	15.09
60	146.72	146.67	146.68	146.64	146.66	**146.67**	17.51
70	127.98	128.06	127.97	128.13	128.14	**128.06**	20.06
80	113.28	113.31	113.30	111.31	113.44	**112.93**	22.75
90	102.57	102.59	102.62	102.82	102.86	**102.69**	25.02
100	93.74	93.77	93.79	93.83	93.83	**93.79**	27.39
110	87.44	87.49	87.46	87.58	87.57	**87.51**	29.36
120	81.66	81.66	81.60	81.67	81.71	**81.66**	31.46
130	76.83	76.85	76.84	76.91	76.91	**76.87**	33.42
140	72.82	72.81	72.82	72.86	72.85	**72.83**	35.27
150	68.91	68.99	68.97	69.02	68.99	**68.97**	37.24
160	65.07	65.02	65.05	65.09	65.07	**65.06**	39.48
170	64.67	64.54	64.47	64.50	64.59	**64.55**	39.79
180	59.46	59.49	59.49	59.52	59.48	**59.49**	43.18
190	57.15	57.19	57.17	57.23	57.28	**57.20**	44.91
200	55.15	55.21	55.18	55.21	55.24	**55.20**	46.54
210	53.46	51.50	53.42	53.52	53.56	**53.09**	48.39
220	52.57	52.62	52.62	52.60	52.67	**52.62**	48.82
230	50.43	50.44	50.42	50.51	50.54	**50.47**	50.90
240	49.38	48.98	48.95	47.29	49.08	**48.74**	52.71

Table 6.91 Time and speedup of Ladder 315S(N) model with N = 18 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `compact`

`Compact` Heisenberg Ladder315S(18) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	35.80	35.71	35.81	35.72	35.88	**35.78**	1.00
10	11.87	11.89	11.87	11.82	11.81	**11.85**	3.02
20	8.53	7.63	8.55	8.49	8.48	**8.34**	4.29
30	7.78	7.77	7.77	7.70	7.71	**7.75**	4.62
40	6.89	6.53	6.77	6.85	6.83	**6.77**	5.28
50	7.00	7.00	6.95	6.92	6.94	**6.96**	5.14
60	7.29	7.30	7.26	6.37	7.24	**7.09**	5.05
70	7.66	7.65	7.61	7.46	7.61	**7.60**	4.71
80	8.05	8.03	8.03	8.05	8.03	**8.04**	4.45
90	8.50	8.46	8.47	7.58	8.43	**8.29**	4.32
100	8.95	8.96	8.91	8.92	8.94	**8.94**	4.00
110	9.13	9.48	9.44	9.42	9.43	**9.38**	3.81
120	9.98	9.07	9.94	9.95	9.96	**9.78**	3.66
130	10.58	10.58	8.82	10.53	10.54	**10.21**	3.50
140	11.15	10.94	11.11	11.13	11.11	**11.09**	3.23
150	11.82	11.81	11.55	11.79	11.77	**11.75**	3.05
160	12.50	12.51	12.50	12.48	12.51	**12.50**	2.86
170	13.31	13.30	13.27	13.28	13.25	**13.28**	2.69
180	13.98	14.01	12.66	14.01	13.98	**13.73**	2.61
190	14.82	14.29	13.43	12.59	14.84	**13.99**	2.56
200	12.05	15.54	15.57	15.58	15.54	**14.85**	2.41
210	16.53	16.53	16.52	16.50	16.49	**16.51**	2.17
220	17.43	17.41	17.08	17.42	15.43	**16.95**	2.11
230	15.66	18.47	18.49	18.47	18.46	**17.91**	2.00
240	19.72	14.22	17.17	19.41	17.55	**17.61**	2.03

Table 6.92 Time and speedup of Ladder 315S(N) model with N = 24 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `balanced`

`Balanced` Heisenberg Ladder315S(24) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	2565.74	2569.90	2569.13	2569.56	2570.11	**2568.89**	1.00
10	633.32	615.96	601.85	628.27	617.80	**619.44**	4.15
20	358.79	353.39	358.15	342.32	362.17	**354.97**	7.24
30	255.15	263.30	244.04	257.84	262.17	**256.50**	10.02
40	207.99	207.69	206.36	206.84	205.77	**206.93**	12.41
50	163.20	165.28	166.02	160.07	156.31	**162.17**	15.84
60	140.84	141.97	142.81	142.21	142.94	**142.15**	18.07
70	124.50	126.09	128.79	134.83	125.94	**128.03**	20.06
80	107.91	108.08	113.08	112.78	109.10	**110.19**	23.31
90	101.81	92.49	104.68	106.45	101.98	**101.48**	25.31
100	92.60	91.05	90.78	93.13	93.47	**92.21**	27.86
110	87.55	87.07	87.02	86.50	86.97	**87.02**	29.52
120	81.90	81.04	80.99	82.28	81.65	**81.57**	31.49
130	76.94	78.16	79.32	79.00	78.67	**78.42**	32.76
140	75.09	74.35	74.49	73.73	73.62	**74.25**	34.60
150	70.59	70.00	69.48	69.29	69.08	**69.69**	36.86
160	68.61	65.82	65.27	65.21	65.82	**66.15**	38.84
170	67.51	65.78	66.26	64.86	66.69	**66.22**	38.79
180	62.95	62.34	61.35	62.88	62.29	**62.36**	41.19
190	61.48	61.67	61.24	61.03	61.74	**61.43**	41.82
200	58.00	57.88	59.11	58.99	57.88	**58.37**	44.01
210	55.71	55.56	55.66	55.32	55.90	**55.63**	46.18
220	55.08	55.23	52.48	55.46	55.57	**54.76**	46.91
230	50.52	50.43	50.56	50.45	50.52	**50.50**	50.87
240	49.07	49.19	49.14	49.06	49.12	**49.11**	52.31

Table 6.93 Time and speedup of Ladder 315S(N) model with N = 18 on an Intel Xeon Phi coprocessor with KMP_THREAD_AFFINITY set to `balanced`

`Balanced` Heisenberg Ladder315S(18) (time in seconds)							
Thread	Data 1	Data 2	Data 3	Data 4	Data 5	**Average**	Speedup
1	35.80	35.71	35.81	35.72	35.88	**35.78**	1.00
10	9.13	8.38	9.15	9.99	8.34	**9.00**	3.98
20	6.93	6.61	7.27	6.55	7.60	**6.99**	5.12
30	7.43	7.59	5.94	7.69	7.50	**7.23**	4.95
40	5.61	6.71	6.62	6.84	6.70	**6.50**	5.51
50	8.10	7.65	8.06	7.82	8.14	**7.95**	4.50
60	9.50	10.28	10.08	10.41	9.39	**9.93**	3.60
70	15.53	14.72	15.35	15.91	15.92	**15.49**	2.31
80	16.99	17.09	16.73	14.77	17.75	**16.67**	2.15
90	18.21	18.13	18.21	18.36	18.74	**18.33**	1.95
100	19.60	19.58	19.64	19.96	19.87	**19.73**	1.81
110	21.49	21.76	21.52	21.52	21.56	**21.57**	1.66
120	22.95	23.02	22.80	22.82	23.20	**22.96**	1.56
130	24.40	23.68	25.08	25.22	25.40	**24.76**	1.45
140	27.04	27.14	26.36	27.09	27.26	**26.98**	1.33
150	29.03	28.53	28.97	29.12	29.25	**28.98**	1.23
160	30.88	31.17	31.44	31.13	30.96	**31.11**	1.15
170	33.06	32.75	32.92	33.04	33.07	**32.97**	1.09
180	34.65	34.59	34.53	34.58	34.62	**34.59**	1.03
190	35.31	35.44	35.51	35.24	35.20	**35.34**	1.01
200	37.24	36.95	36.97	36.85	37.03	**37.01**	0.97
210	39.42	38.84	39.55	39.64	39.62	**39.41**	0.91
220	42.13	41.90	42.03	42.11	42.25	**42.09**	0.85
230	34.27	44.88	45.04	44.75	44.66	**42.72**	0.84
240	44.74	46.87	46.80	46.61	46.71	**46.35**	0.77

7 Reference

[1] A. Honecker, *A comparative study of the magnetization process of two-dimensional antiferromagnets*, J. Phys.: Condens. Matter **11**, 4697-4713 (1999).

[2] Andreas Lüscher and Andreas M. Läuchli, *Exact diagonalization study of the antiferromagnetic spin-1/2 Heisenberg model on the square lattice in a magnetic field*, Phys. Rev. B **79**, 195102 (2009).

[3] B: G. Kresse and J. Furthmüller, Phys. Rev. B **54**, 11169 (1996).

[4] Balanced Affinity Type, Intel® Fortran Compiler XE 13.1 User and Reference Guides, https://software.intel.com/sites/products/documentation/doclib/stdxe/2013/composerxe/compiler/fortran-lin/GUID-8FCD3720-6F73-429C-AE65-7144ED0B991A.htm retrieved November 12th, 2013.

[5] Barnes, T., Dagotto, E., Riera, J., Swanson, E.S.: Excitation spectrum of Heisenberg spin ladders. Phys. Rev. B **47**, 3196–3203 (1993).

[6] Crouzeix, M., Phillipe, B., Sadkane, M.: The Davidson method. SIAM J. Sci. Comput. 15(1), 62–76 (1994).

[7] David L. Harrar II, "On the Davidson and Jacobi-Davidson Methods for Large-Scale Eigenvalue Problems", Research Report, Centre for Mathematics and Applications, Australian National University.

[8] Davidson, E.R.: The iterative calculation of a few lowest eigenvalues and corresponding eigenvectors of large symmetric matrices. Comput. Phys. **17**, 87–94 (1975).

[9] Fehske, H. Schneider, R. and Weiße, A. (eds.), Computational Many-Particle Physics, Lect. Notes Phys. **739** (Springer-Verlag Berlin Heidelberg 2008).

[10] Gene H.G.: Matrix Computations, 3rd edition. Johns Hopkins University Press; 471-489 (1996).

http://maths-intranet.anu.edu.au/research.reports/mrr/99/047/MRR99-047.pdf

[11] J. B. Parkinson and J. C. Bonner, *Spin chain in a field: Crossover from quantum to classical behavior*, Phys. Rev. B **32**, 4703-4724 (1985).

[12] J. Jeffers and J. Reinders, Intel Xeon Phi Coprocessor High Performance Programming (Morgan Kaufman, NewYork, 2013).

[13] J. Oitmaa and D. D. Betts, The ground state of two quantum models of magnetism, Can. J. Phys. **56**, 897 (1978).

[14] Jill C. Bonner and Michael E. Fisher, *Linear Magnetic Chains with Anisotropic Coupling*, Phys. Rev. **135**, A640-A658 (1964).

[15] Johannes Richter, Jörg Schulenburg, and Andreas Honecker, *Quantum Magnetism in Two Dimensions: From Semi-classical Néel Order to Magnetic Disorder*, Lect. Notes Phys. **645**, 85-153 (2004).

[16] Kresse, G., Furthmüller, J.: Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set. Comput. Mater. Sci. 6, 15–50 (1996).

[17] Liu, B.: Numerical Algorithms in Chemistry: Algebraic method, vol. LBL-8158, chap. The simultaneous expansion for the solution of several of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices, pp. 49–53. Lawrence Berkeley Lab., CA (1978).

[18] Mamoru Usami and Sei-ichiro Suga, *Critical properties of S=1/2 Heisenberg ladders in magnetic fields*, Phys. Rev. B **58**, 14401-14406 (1998).

[19] Martín-Delgado, M.A., Shankar, R., Sierra, G.: Phase transitions in staggered spin ladders. Phys. Rev. Lett. **77**, 3443–3446 (1996).

[20] Murray, C.W., Racine, S.C., Davidson, E.R.: Improved algorithms for the lowest few eigenvalues and associated eigenvectors of large matrices. J. Comput. Phys. **103**, 382–389 (1992).

[21] N. Flocke, *Exact diagonalization study on spin-1/2 ladders as a function of two coupling*

parameters using the symmetric group approach, Phys. Rev. B **56**, 13673-13676 (1997).

[22] Parkinson, J.B., Farnell, D.J.J.: An Introduction to Quantum Spin Systems, *Lect. Notes Phys.*, **816**. Springer-Verlag, Berlin Heidelberg (2010).

[23] Pulay, P.: Convergence acceleration of iterative sequences. The case of SCF iteration. Chem. Phys. Lett. **73**, 393–398 (1980).

[24] S. Nemati, S. Batebi, and S. Mahdavifar, *Concurrence and entanglement entropy in a dimerised spin-1/2 two-leg ladder*, Eur. Phys. J. B **83**, 329-335 (2011).

[25] Sandra J. Gibson, R. Meyer and Gennady Y. Chitov, *Numerical study of critical properties and hidden orders in dimerized spin ladders*, Phys. Rev. B **83**, 104423 (2011).

[26] Sparse Matrix Wikipedia, http://en.wikipedia.org/wiki/Sparse_matrix Retrieved Novermber 16th, 2013.

[27] Steven R. White, *Density matrix formulation fro quantum renormalization groups*, Phys. Rev. Lett. **69**, 2863-2866 (1992).

[28] Thijssen, J.: Computational Physics, 2nd edition. Cambridge University Press, Cambrige (2010).

[29] Totsuka, K., Suzuki, M.: The spin-1/2 Heisenberg spin ladder with bond alternation. J. Phys: Condens. Matter **7**, 6079–6096 (1995).

[30] Weiße, A., Fehske, H.: Computational Many-Particle Physics, Lect. Notes Phys., vol. **739**, chap. Exact Diagonalization Techniques, pp. 529–544. Springer-Verlag, Berlin Heidelberg (2008).

[31] Wood, D.M., Zunger, A.: A new method for diagonalising large matrices. J. Phys. A: Math. Gen. **18**, 1343–1359 (1985).