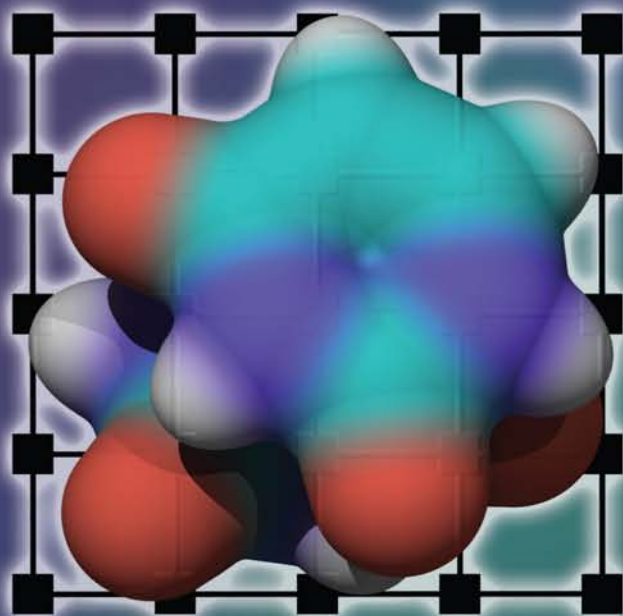


# Parallel Computing *in* Quantum Chemistry

---



---

Curtis L. Janssen

Ida M. B. Nielsen

Parallel Computing  
*in*  
Quantum Chemistry



# Parallel Computing *in* Quantum Chemistry

Curtis L. Janssen

Ida M. B. Nielsen



**CRC Press**

Taylor & Francis Group

Boca Raton London New York

---

CRC Press is an imprint of the  
Taylor & Francis Group, an **informa** business

This manuscript has been authored by Sandia National Laboratories, Contract No. DE-AC04-94AL85000 with the U.S. Department of Energy and Limit Point Systems, Inc. subcontract LF-5537 with Sandia. The publisher, by accepting the manuscript for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

CRC Press  
Taylor & Francis Group  
6000 Broken Sound Parkway NW, Suite 300  
Boca Raton, FL 33487-2742

© 2008 by Taylor & Francis Group, LLC  
CRC Press is an imprint of Taylor & Francis Group, an Informa business

No claim to original U.S. Government works  
Printed in the United States of America on acid-free paper  
10 9 8 7 6 5 4 3 2 1

International Standard Book Number-13: 978-1-4200-5164-3 (Hardcover)

This book contains information obtained from authentic and highly regarded sources. Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The Authors and Publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Except as permitted under U.S. Copyright Law, no part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, please access [www.copyright.com](http://www.copyright.com) (<http://www.copyright.com/>) or contact the Copyright Clearance Center, Inc. (CCC) 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. CCC is a not-for-profit organization that provides licenses and registration for a variety of users. For organizations that have been granted a photocopy license by the CCC, a separate system of payment has been arranged.

**Trademark Notice:** Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

---

#### Library of Congress Cataloging-in-Publication Data

---

Janssen, Curtis L.

Parallel computing in quantum chemistry / Curtis L. Janssen and Ida M.B. Nielsen.

p. cm.

Includes bibliographical references and index.

ISBN 978-1-4200-5164-3 (acid-free paper)

1. Quantum chemistry--Data processing. 2. Parallel programming (Computer science) 3. Parallel processing (Electronic computers) I. Nielsen, Ida M. B. II. Title.

QD462.6.D38J36 2008

541'.280285435--dc22

2007048052

---

Visit the Taylor & Francis Web site at  
<http://www.taylorandfrancis.com>

and the CRC Press Web site at  
<http://www.crcpress.com>

---

# *Contents*

---

## **I Parallel Computing Concepts and Terminology**

<b>1 Introduction</b> .....	3
1.1 Parallel Computing in Quantum Chemistry: Past and Present ....	4
1.2 Trends in Hardware Development .....	5
1.2.1 Moore's Law .....	5
1.2.2 Clock Speed and Performance .....	6
1.2.3 Bandwidth and Latency .....	7
1.2.4 Supercomputer Performance .....	8
1.3 Trends in Parallel Software Development .....	10
1.3.1 Responding to Changes in Hardware .....	10
1.3.2 New Algorithms and Methods .....	10
1.3.3 New Programming Models .....	12
References .....	13
<b>2 Parallel Computer Architectures</b> .....	17
2.1 Flynn's Classification Scheme .....	17
2.1.1 Single-Instruction, Single-Data .....	17
2.1.2 Single-Instruction, Multiple-Data .....	18
2.1.3 Multiple-Instruction, Multiple-Data .....	18
2.2 Network Architecture .....	19
2.2.1 Direct and Indirect Networks .....	19
2.2.2 Routing .....	20
2.2.3 Network Performance .....	23
2.2.4 Network Topology .....	25
2.2.4.1 Crossbar .....	26
2.2.4.2 Ring .....	27
2.2.4.3 Mesh and Torus .....	27
2.2.4.4 Hypercube .....	28
2.2.4.5 Fat Tree .....	28
2.2.4.6 Bus .....	30
2.2.4.7 Ad Hoc Grid .....	31
2.3 Node Architecture .....	31
2.4 MIMD System Architecture .....	34
2.4.1 Memory Hierarchy .....	35
2.4.2 Persistent Storage .....	35

2.4.2.1	Local Storage	37
2.4.2.2	Network Storage	37
2.4.2.3	Trends in Storage	38
2.4.3	Reliability	38
2.4.4	Homogeneity and Heterogeneity	39
2.4.5	Commodity versus Custom Computers	40
2.5	Further Reading	42
	References	43
<b>3</b>	<b>Communication via Message-Passing</b>	<b>45</b>
3.1	Point-to-Point Communication Operations	46
3.1.1	Blocking Point-to-Point Operations	46
3.1.2	Non-Blocking Point-to-Point Operations	47
3.2	Collective Communication Operations	49
3.2.1	One-to-All Broadcast	50
3.2.2	All-to-All Broadcast	51
3.2.3	All-to-One Reduction and All-Reduce	54
3.3	One-Sided Communication Operations	55
3.4	Further Reading	56
	References	56
<b>4</b>	<b>Multi-Threading</b>	<b>59</b>
4.1	Pitfalls of Multi-Threading	61
4.2	Thread-Safety	64
4.3	Comparison of Multi-Threading and Message-Passing	65
4.4	Hybrid Programming	66
4.5	Further Reading	69
	References	70
<b>5</b>	<b>Parallel Performance Evaluation</b>	<b>71</b>
5.1	Network Performance Characteristics	71
5.2	Performance Measures for Parallel Programs	74
5.2.1	Speedup and Efficiency	74
5.2.2	Scalability	79
5.3	Performance Modeling	80
5.3.1	Modeling the Execution Time	80
5.3.2	Performance Model Example: Matrix-Vector Multiplication	83
5.4	Presenting and Evaluating Performance Data: A Few Caveats	86
5.5	Further Reading	90
	References	90
<b>6</b>	<b>Parallel Program Design</b>	<b>93</b>
6.1	Distribution of Work	94
6.1.1	Static Task Distribution	95
6.1.1.1	Round-Robin and Recursive Task Distributions	96

6.1.2	Dynamic Task Distribution .....	99
6.1.2.1	Manager-Worker Model .....	99
6.1.2.2	Decentralized Task Distribution .....	101
6.2	Distribution of Data .....	101
6.3	Designing a Communication Scheme .....	104
6.3.1	Using Collective Communication .....	104
6.3.2	Using Point-to-Point Communication .....	105
6.4	Design Example: Matrix-Vector Multiplication .....	107
6.4.1	Using a Row-Distributed Matrix .....	108
6.4.2	Using a Block-Distributed Matrix .....	109
6.5	Summary of Key Points of Parallel Program Design .....	112
6.6	Further Reading .....	114
	References .....	114

## II Applications of Parallel Programming in Quantum Chemistry

<b>7</b>	<b>Two-Electron Integral Evaluation .....</b>	<b>117</b>
7.1	Basics of Integral Computation .....	117
7.2	Parallel Implementation Using Static Load Balancing .....	119
7.2.1	Parallel Algorithms Distributing Shell Quartets and Pairs .....	119
7.2.2	Performance Analysis .....	121
7.2.2.1	Determination of the Load Imbalance Factor $k(p)$ .....	122
7.2.2.2	Determination of $\mu$ and $\sigma$ for Integral Computation .....	123
7.2.2.3	Predicted and Measured Efficiencies .....	124
7.3	Parallel Implementation Using Dynamic Load Balancing .....	125
7.3.1	Parallel Algorithm Distributing Shell Pairs .....	126
7.3.2	Performance Analysis .....	128
7.3.2.1	Load Imbalance .....	128
7.3.2.2	Communication Time .....	128
7.3.2.3	Predicted and Measured Efficiencies .....	129
	References .....	130
<b>8</b>	<b>The Hartree–Fock Method .....</b>	<b>131</b>
8.1	The Hartree–Fock Equations .....	131
8.2	The Hartree–Fock Procedure .....	133
8.3	Parallel Fock Matrix Formation with Replicated Data .....	135
8.4	Parallel Fock Matrix Formation with Distributed Data .....	138
8.5	Further Reading .....	145
	References .....	146
<b>9</b>	<b>Second-Order Møller–Plesset Perturbation Theory .....</b>	<b>147</b>
9.1	The Canonical MP2 Equations .....	147
9.2	A Scalar Direct MP2 Algorithm .....	149

9.3	Parallelization with Minimal Modifications .....	151
9.4	High-Performance Parallelization .....	154
9.5	Performance of the Parallel Algorithms .....	158
9.6	Further Reading .....	164
	References .....	164
<b>10</b>	<b>Local Møller–Plesset Perturbation Theory .....</b>	<b>167</b>
10.1	The LMP2 Equations .....	167
10.2	A Scalar LMP2 Algorithm .....	169
10.3	Parallel LMP2 .....	170
10.3.1	Two-Electron Integral Transformation .....	171
10.3.2	Computation of the Residual .....	173
10.3.3	Parallel Performance .....	174
	References .....	177
 <b>Appendices</b>		
<b>A</b>	<b>A Brief Introduction to MPI .....</b>	<b>181</b>
<b>B</b>	<b>Pthreads: Explicit Use of Threads .....</b>	<b>189</b>
<b>C</b>	<b>OpenMP: Compiler Extensions for Multi-Threading .....</b>	<b>195</b>
	<b>Index .....</b>	<b>205</b>

---

## *Preface*

---

This book is intended to serve as a reference for the design and implementation of parallel quantum chemistry programs. Development of efficient quantum chemistry software capable of utilizing large-scale parallel computers requires a grasp of many issues pertaining to both parallel computing hardware and parallel programming practices, as well as an understanding of the methods to be implemented. The text provides an in-depth view of parallel programming challenges from the perspective of a quantum chemist, including parallel computer architectures, message-passing, multi-threading, parallel program design and performance analysis, as well as parallel implementation of important electronic structure procedures and methods such as two-electron integral computation, Hartree–Fock and second-order Møller–Plesset perturbation (MP2) theory, and the local correlation method LMP2. Some topics relevant to parallel computing in quantum chemistry have not been included in this book. Thus, performance tools and debuggers are not treated, parallel I/O is only briefly discussed, and advanced electronic structure methods such as coupled-cluster theory and configuration interaction are not covered.

We will assume that the reader has a basic understanding of quantum chemistry, including Hartree–Fock theory and correlated electronic structure methods such as Møller–Plesset perturbation theory. Readers can find introductory discussions of these methods in, for example, Jensen<sup>1</sup> and the classic Szabo and Ostlund text.<sup>2</sup> A comprehensive and somewhat more advanced treatment of electronic structure theory can be found in Helgaker, Jørgensen, and Olsen.<sup>3</sup> No prior experience with parallel computing is required, but the reader should be familiar with computer programming and programming languages at the advanced undergraduate level. The program examples in the book are written in the C programming language, and at least a rudimentary knowledge of C will therefore be helpful. The text by Kernighan and Ritchie<sup>4</sup> covers all C features used in this book.

---

### **Scope and Organization of the Text**

This book is divided into two parts. In Part I we will discuss parallel computer architectures as well as parallel computing concepts and terminology with a focus on good parallel program design and performance

analysis. Part II contains detailed discussions and performance analyses of parallel algorithms for a number of important and widely used quantum chemistry procedures and methods.

An outline of the contents of each chapter is given below.

### ***Chapter 1: Introduction***

Here we provide a brief history of parallel computing in quantum chemistry and discuss trends in hardware as well as trends in the methods and algorithms of quantum chemistry. The impact of these trends on future quantum chemistry programs will be considered.

### ***Chapter 2: Parallel Computer Architectures***

This chapter provides an overview of parallel computer architectures, including the traditional Flynn classification scheme and a discussion of computation nodes and the networks connecting them. We also present an overall system view of a parallel computer, describing the hierarchical nature of parallel architecture, machine reliability, and the distinction between commodity and custom computers.

### ***Chapter 3: Communication via Message-Passing***

This chapter covers message-passing, one of the primary software tools required to develop parallel quantum chemistry programs for distributed memory parallel computers. Point-to-point, collective, and one-sided varieties of message-passing are also discussed.

### ***Chapter 4: Multi-Threading***

The importance of multi-threading will continue to increase due to the emergence of multicore chips. Parallelization by means of multi-threading is discussed as well as hybrid multi-threading/message-passing approaches for utilizing large-scale parallel computers.

### ***Chapter 5: Parallel Performance Evaluation***

Design and implementation of efficient parallel algorithms requires careful analysis and evaluation of their performance. This chapter introduces idealized machine models along with measures for predicting and assessing the performance of parallel algorithms.

### ***Chapter 6: Parallel Program Design***

This chapter discusses fundamental issues involved in designing and implementing parallel programs, including the distribution of tasks and data as well as schemes for interprocessor communication.

### ***Chapter 7: Two-Electron Integral Evaluation***

An important, basic step performed in most quantum chemistry programs is the computation of the two-electron integrals. Schemes for parallel computation of these integrals and detailed performance models incorporating load imbalance are discussed.

### ***Chapter 8: The Hartree–Fock Method***

The Hartree–Fock method is central to quantum chemistry, and an efficient Hartree–Fock program is an essential part of a quantum chemistry program package. We outline the Hartree–Fock procedure and present and analyze both replicated data and distributed data Fock matrix formation algorithms.

### ***Chapter 9: Second-Order Møller–Plesset Perturbation Theory***

Second-order Møller–Plesset (MP2) perturbation theory is a widely used quantum chemical method for incorporating electron correlation. This chapter considers parallel computation of MP2 energies, comparing the performance achievable with simple and more sophisticated parallelization strategies.

### ***Chapter 10: Local Møller–Plesset Perturbation Theory***

Local correlation methods represent an important new class of correlated electronic structure methods that aim at computing molecular properties with the same accuracy as their conventional counterparts but at a significantly lower computational cost. We discuss the challenges of parallelizing local correlation methods in the context of local second-order Møller–Plesset perturbation theory, illustrating a parallel implementation and presenting benchmarks as well.

### ***Appendix A: A Brief Introduction to MPI***

The Message-Passing Interface (MPI) is the primary mechanism used for explicit message-passing in scientific computing applications. This appendix briefly discusses some of the most commonly used MPI routines.

### ***Appendix B: Pthreads***

Pthreads is a standard for creating and managing multiple threads. We give a brief introduction to multi-threaded programming with Pthreads, including an example Pthreads program.

### ***Appendix C: OpenMP***

OpenMP is a set of compiler extensions to facilitate development of multi-threaded programs. We describe these compiler extensions, using example source code illustrating parallel programming with OpenMP.

---

## References

1. Jensen, F. *Introduction to Computational Chemistry*. Chichester, UK: John Wiley & Sons, 1999.
2. Szabo, A., and N. S. Ostlund. *Modern Quantum Chemistry*, 1st revised edition. New York: McGraw-Hill, 1989.
3. Helgaker, T., P. Jørgensen, and J. Olsen. *Molecular Electronic-Structure Theory*. Chichester, UK: John Wiley & Sons, 2000.
4. Kernighan, B. W., and D. M. Ritchie. *The C Programming Language*, 2nd edition. Englewood Cliffs, NJ: Prentice Hall, 1988.

---

## *Acknowledgments*

We are indebted to many people who provided support and advice in the preparation of this book. Lance Wobus of Taylor & Francis encouraged us to undertake the writing of the book, and Jennifer Smith expertly guided us through the final stages of manuscript preparation. We thank Craig Smith and Kurt Olsen at Sandia National Laboratories for facilitating the project, and we appreciate the assistance of Gregg Andreski, who prepared most of the artwork. Jan Linderberg kindly provided historical references, and Shawn Brown, Joe Kenny, and Matt Leininger made insightful comments on various drafts of the manuscript. Ron Minnich, Helgi Adalsteinsson, Jim Schutt, Ed Valeev, Daniel Crawford, Theresa Windus, and David Bernholdt provided stimulating discussions, feedback, and technical data. Finally, CLJ would like to thank Mike Colvin for introducing him to parallel computing.



---

## *Funding Statement*

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.



---

## *Authors*

---

**Curtis L. Janssen** is a Distinguished Member of the Technical Staff at Sandia National Laboratories and holds a Ph.D. in theoretical chemistry from the University of California at Berkeley. He has an extensive publication record in the areas of quantum chemistry methods development and high-performance computing, and he is the lead developer of the MPQC program suite.

Curtis L. Janssen  
Sandia National Laboratories  
7011 East Ave.  
Livermore, CA 94550  
cljanss@sandia.gov

**Ida M. B. Nielsen** is a scientist at Sandia National Laboratories and holds a Ph.D. in theoretical chemistry from Stanford University. She has published numerous articles on the development and application of quantum chemical methods and high-performance computing and is one of the core developers of the MPQC program suite.

Ida M. B. Nielsen  
Sandia National Laboratories  
7011 East Ave.  
Livermore, CA 94550  
ibniels@sandia.gov



---

## ***Disclaimer of Liability***

This work of authorship was prepared as an account of work sponsored by an agency of the United States Government. Accordingly, the United States Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so for United States Government purposes. Neither Sandia Corporation, the United States Government, nor any agency thereof, nor any of their employees makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately-owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by Sandia Corporation, the United States Government, or any agency thereof. The views and opinions expressed herein do not necessarily state or reflect those of Sandia Corporation, the United States Government, or any agency thereof.

---

### **Trademarks**

Intel<sup>®</sup> and Intel<sup>®</sup> Xeon<sup>®</sup> are trademarks of Intel Corporation. AMD, AMD Opteron<sup>™</sup>, and HyperTransport<sup>™</sup> are trademarks of Advanced Micro Devices, Inc. Motorola<sup>®</sup> AltiVec<sup>™</sup> is a trademark of Motorola, Inc. IBM POWER<sup>™</sup> is a trademark of International Business Machines Corporation. Cray X1<sup>™</sup> and Cray XT3<sup>™</sup> are trademarks of Cray Inc. The Linux<sup>®</sup> trademark is owned by Linus Torvalds. InfiniBand<sup>™</sup> is a trademark of the InfiniBand Trade Association. InfiniHost<sup>™</sup> and InfiniScale<sup>™</sup> are trademarks of Mellanox Technologies. Java<sup>™</sup> is a trademark of Sun Microsystems Inc. OpenMP<sup>™</sup> is a trademark of the OpenMP Architecture Review Board. UNIX<sup>™</sup> is a trademark of The Open Group. POSIX<sup>®</sup> is a trademark of the Institute of Electrical and Electronic Engineers (IEEE). Myrinet<sup>®</sup> is a trademark of Myricom, Inc. Quadrics<sup>®</sup> is a trademark of Quadrics Ltd.



## **Part I**

# **Parallel Computing Concepts and Terminology**



# 1

---

## *Introduction*

---

The need for parallel software for scientific computing is ever increasing. Supercomputers are not only being built with more processors, but parallel computers are also no longer limited to large machines owned and managed by high-performance computing centers; parallel desktop computers are increasingly widespread, and even laptop computers have multiple processors. Development of scientific computing software must adapt to these conditions as parallel computation becomes the norm rather than the exception. In the field of quantum chemistry, additional factors contribute to the need for parallelism. Quantum chemistry has become an indispensable tool for investigating chemical phenomena, and quantum chemical methods are employed widely in research across many chemical disciplines; this widespread use of quantum chemistry reinforces the importance of rapid turnaround computations, which can be addressed by parallel computing. Additionally, for quantum chemistry to continue to be an integral part of chemical research, quantum chemical methods must be applicable to the chemical systems of interest, including larger molecules, and parallelism can play an important role in extending the range of these methods. Parallel implementations can broaden the scope of conventional quantum chemical methods, whose computational cost scales as a high-degree polynomial in the molecular size, and enable the treatment of very large molecular systems with linear-scaling or reduced-scaling methods.

In the following, we will first give a brief historical sketch and current perspective of parallel computing in quantum chemistry. We will then discuss trends in hardware development for single-processor and parallel computers as well as trends in parallel software development, including the parallel programming challenges following the emergence of new quantum chemical methods and the changes in hardware.