John von Neumann Institute for Computing

High Performance Computing in Chemistry

edited by

Johannes Grotendorst

University of Karlsruhe

University of Stuttgart

Research Centre Jülich

Report

Central Institute for Applied Mathematics

Publication Series of the John von Neumann Institute for Computing (NIC)

NIC Series Volume 25

John von Neumann Institute for Computing (NIC)

High Performance Computing in Chemistry

edited by
Johannes Grotendorst

Report of the Joint Research Project:
High Performance Computing in Chemistry – HPC-Chem

Funded by the Federal Ministry for Education and Research (BMBF)
Grand Number: 01 IR A17 A – C
Period: 1 March 2001 – 29 February 2004

NIC Series Volume 25 ISBN 3-00-013618-5

Central Institute for Applied Mathematics

Die Deutsche Bibliothek – CIP-Cataloguing-in-Publication-Data
A catalogue record for this publication is available from Die Deutsche Bibliothek

Publisher: NIC-Directors
Distributor: NIC-Secretariat
Research Centre Jülich
52425 Jülich
Germany
Internet: www.fz-juelich.de/nic
Printer: Graphische Betriebe, Forschungszentrum Jülich

NIC Series Volume 25 ISBN 3-00-013618-5

Preface

Over the last three decades the methods of quantum chemistry have shown an impressive development: a large number of reliable and efficient approximations to the solution of the non-relativistic Schrödinger and the relativistic Dirac equation, respectively, are available. This is complemented by the availability of a number of well-developed computer programs which allow of the treatment of chemical problems as a matter of routine. This progress has been acknowledged by the Nobel prize in chemistry 1998 to John Pople and Walter Kohn for the development of quantum chemical methods.

Nowadays, Theoretical Chemistry is widely accepted as an essential ingredient to research in a wide field of applications ranging from chemistry over biochemistry/biophysics to different flavors of material science: quantum chemical methods are indeed one standard tool at universities and research centres as well as in industrial research. The progress in experimental techniques is invariably complemented by an increasing demand for accurate quantum mechanical models as a means to analyze and interpret experimental data as well as to provide a deeper understanding of the results. On its own, the prediction of structures and properties of materials and individual chemical compounds or complexes is of great importance - either because the targets are experimentally inaccessible at sufficient accuracy or experiments are too expensive or impractical.

Currently quantum chemical methods are on the verge of being applied to realistic problems. Many research topics of considerable economical interest have quite demanding constraints: they require to model large numbers of particles (because the interesting properties require a certain minimum size of the model to be of use), the requested level of accuracy is achievable only within the realm of electronic structure methods or requires the time-resolved dynamics of the process in question. Additionally, it is observed that neighboring disciplines such as chemistry, biochemistry, biophysics, solid state physics and material science are gradually merging and in fact are sharing similar challenges and closely related methodologies. In view of today's complexity of software engineering and computer hardware these disciplines depend heavily on the support of computer science and applied mathematics. Thus, in the field of computational science an increasing amount of multidisciplinarity is not only beneficial but essential for solving complex problems.

Finally, we have to anticipate the tremendous development in the area of information technology both from the side of software as well as hardware development. In particular the emerging parallel computer and cluster systems open the road to tackle challenges of unprecedented complexity. However, method development must not only respond to the need of ever better and computationally less expensive (linear scaling) models but as well to the requirements of the underlying computer system in terms of parallel scalability and efficient usage of the (ever-changing) hardware.

Having in mind the wishes and requirements of the researchers in the NIC community and in the German chemical industry the most promising methodologies and quantum chemistry codes were chosen in order to push forward the development. The selected program packages TURBOMOLE, QUICKSTEP, and MOLPRO cover complementary models and aspects of the whole range of quantum chemical methods. Within the project High Performance Computing in Chemistry (HPC-Chem) the functionality of these codes was extended, several important methods with linear scaling behavior with respect to the molecular size were developed and implemented, and last but not least the parallel scalability on modern supercomputers and cluster systems was substantially improved. In addition, for the treatment of solute-solvent interactions in quantum mechanical calculations the continuum model COSMO has been integrated into the aforementioned programs. This is of great relevance for the range of use since most practical problems are dealing with liquid phase chemistry.

I thank the HPC-Chem project partners and the industrial collaborators for their cooperativeness and the authors from the different research groups for their contributions to this book. Special thanks are due to Monika Marx, who invested time and effort defining the layout, correcting the figures, and designing the cover. The beauty of this volume is entirely her merit.

Jülich, October 2004

Johannes Grotendorst

Contents

Goals of the Project

Further development of quantum chemistry codes still involves both methodological development and issues of parallelization in order to make these highly advanced techniques applicable to problems of so far unprecedented size. In particular the more closely hardware related parallelization and optimization benefits largely from the contributing disciplines computer science and applied mathematics. It is not possible to simply scale up existing methods or numerical procedures to arbitrary problem sizes including parallelization. The aim is to reduce the complexity of the algorithms and to enhance the parallel scalability. We need to understand that moving to ever larger system sizes or applying even more accurate methods we will find ourselves confronted with not yet anticipated problems. On the one hand it is important to decouple hardware and software development on the other hand it is essential to exploit modern parallel computer architectures.

The goal of the reported joint research project between Research Centre Jülich (Parallelization, Linear Algebra, CFMM), University of Karlsruhe (TURBOMOLE), ETH Zürich (QUICKSTEP), University of Stuttgart (MOLPRO) and COSMOlogic (COSMO) was to join forces and to focus on the improvement of the most promising methodologies and application codes that will have substantial impact on future research capabilities in academia and industry in Germany. The selected programs and methodologies present diverse though complementary aspects of quantum chemistry and their combination was aimed at synergetic effects among the different development groups. This was a distinct feature of the multidisciplinary HPC-Chem project. The ultimate target of all development efforts was to increase the range of applicability of some of the most important electronic structure methods to system sizes which arise naturally from many application areas in the natural sciences.

Methods and programs developed within this project have been evaluated by the industrial collaborators BASF AG and Infracor GmbH, and are being tested in NIC projects on the Jülich supercomputer.

A brief overview covering the selected methodologies and quantum chemistry codes is given in the following sections. Detailed discussions of the work carried out by the project partners are found in the corresponding subsequent chapters.

DFT Functionality in TURBOMOLE
University of Karlsruhe

Density functional theory (DFT) based methods employing non-hybrid exchange-correlation functionals are not only more accurate than standard Hartree-Fock (HF) methods and applicable to a much wider class of chemical compounds, they are also faster by orders of magnitudes compared to HF implementations. This remarkable feature arises from the separate treatment of the Coulomb and exchange contributions to the Kohn-Sham matrix, which allows to exploit more efficient techniques for their evaluation. With DFT employing hybrid exchange-correlation functionals this advantage is lost and only the (slower) traditional direct HF procedures are applicable. Thus, non-hybrid DFT is the natural choice for electronic structure calculations on very extended systems, which are otherwise intractable by quantum mechanical methods. However, as the exchange-correlation functional is unknown, DFT suffers from the distinct disadvantage that, in contrast to more traditional quantum chemistry methods, there is no systematic way to improve and to assess the accuracy of a calculation. Fortunately, extensive experience shows which classes of chemical compounds can be modeled with good success.

TURBOMOLE's competitiveness is primarily due to (i) the exploitation of molecular symmetry for all point groups in most modules, giving rise to savings roughly by the order of the symmetry group, (ii) the resolution of identity (RI) technique which typically offers savings of about a factor of hundred, and finally (iii) very efficient implementations of integral evaluation and quadrature algorithms.

Within this project a multipole approximation to the RI technique has been implemented for the energy as well as gradients with respect to a displacement of the coordinates of the nuclei named Multipole Assisted RI-J procedure (MARI-J). This method decreases the effective scaling of the evaluation of the Coulomb term to approximately $N^{1.6}$, where N is a measure of the system size, resulting in substantially reduced effort for structure optimizations. Another important aspect of DFT calculations is the implementation of (analytical) second derivatives with respect to the nuclear coordinates carried out in this project. Infrared and Raman spectra are experimentally fairly readily accessible and contain a great deal of information about the structure of the compound in question. The actual assignment of the spectrum is often difficult and requires its simulation. The CPU time consumption mostly stems from the evaluation of the Coulomb contribution to the coupled perturbed Kohn-Sham equations. The RI-J approximation has been implemented for the second derivatives with respect to the nuclear coordinates reducing the computation time by roughly a factor of 2.5.

QUICKSTEP: Make the Atoms Dance ETH Zürich

The general statements regarding DFT given in the previous section apply to QUICKSTEP as well. QUICKSTEP is a complete re-implementation of the Gaussian plane waves (GPW) method as it is defined in the framework of Kohn-Sham density functional theory. Due to the usage of plane waves, QUICKSTEP enforces periodic boundary conditions and is thus somewhat complementary to the molecular TURBOMOLE code. As such, QUICKSTEP does not make use of point group symmetry, but on the other hand it offers substantial advantages for the modeling of solids or liquids. QUICKSTEP exploits like plane wave codes the simplicity by which the time-consuming Coulomb term can be evaluated using the efficient Fast Fourier Transform (FFT) algorithm, which shows a linear scaling behavior. In that way, the Kohn-Sham matrix is calculated by QUICKSTEP with a computational cost that scales linearly with the system size. However, the expansion of Gaussian-type functions in terms of plane waves also suffers from disadvantages, as strong spatial variations of the density would lead to extremely long and uneconomic expansion lengths. This problem is alleviated like in plane wave methods by the use of atomic pseudo potentials for the inner shells.

A new, fully modular and efficiently parallelized implementation of the GPW method including gradients has been carried out. Gaussian basis sets have been specifically optimized for the pseudo potentials of Goedecker, Teter, and Hutter (GTH). Since the traditional wavefunction optimization step, which involves the diagonalization of the full Kohn-Sham matrix, constitutes a substantial bottleneck for large calculations because of its cubic scaling, two alternative schemes, pseudo diagonalization and orbital transformation, have been investigated. The resulting performance data measured on the Jülich supercomputer Jump are impressive. Turn-around times of approximately 100 seconds per molecular dynamics (MD) step for a liquid water simulation of a unit cell with 256 water molecules on 128 CPUs suggest substantial future potential. Also geometry optimizations for molecular or crystalline systems up to approximately 300 atoms have been demonstrated to be feasible within a few minutes per geometry optimization cycle on 8 to 16 CPUs.

QUICKSTEP is part of the open source project CP2K which ensures continuation of the development in the future.

Local Electron Correlation Methods with Density Fitting in MOLPRO University of Stuttgart

Local electron correlation methods recognize that electron correlation, i.e. the difference between the exact solution to the Schrödinger equation and its Hartree-Fock (mean-field) approximation, is a short-range effect (in insulators) which decreases approximately with the sixth power of the distance between two local charge distributions. The prohibitive costs of electron correlation techniques mainly originate from the use of the orthonormal, canonical and delocalized HF molecular orbitals. Thus, central to the local electron correlation techniques is the localization of the molecular orbitals and the decomposition of the localized orbitals into spatially close subsets (orbital domains and pair domains) whose size is independent of the extent of the molecule. Configuration spaces are constructed by excitations within these domains thus reducing their number to O(N^2). Introducing a hierarchical treatment depending upon the distance of the orbital domains linear scaling can be achieved. This strategy offers the possibility to enormously reduce the costs of electron correlation techniques while maintaining the well-established hierarchy of wavefunction based *ab initio* methods. This approach succeeded in the development of local MP2 and CCSD(T) methods with approximately linear scaling of the computational cost, thus dramatically extending the range of applicability of such high-level methods. Still all electron correlation methods suffer from the slow convergence of the electron correlation energy with respect to the basis set size, thus somewhat offsetting the gain obtained by the local treatment. This aspect has also been considered by implementing local r_{12} methods which substantially improve the convergence behavior. It is remarkable, that for local MP2 the preliminary HF calculation, i.e. a conceptionally much simpler procedure, is the most time-consuming step.

Within the HPC-Chem project these new local correlation methods have been parallelized, density fitting approximations to speed up the integral evaluation have been incorporated and the method has been extended by an open-shell formalism. In addition, local r_{12} methods have been implemented. The bottleneck of evaluating the Hartree-Fock exchange contribution has been much reduced by local density fitting approximations as well, leading to speedups by 1-2 orders of magnitude. All these so far unique and unprecedented methods are part of the MOLPRO package of *ab initio* programs.

Parallel DFT in TURBOMOLE, Linear Algebra, and CFMM Research Centre Jülich

The (re-)parallelization of the DFT code in TURBOMOLE aims specifically at further extending its range of applicability to very large systems by means of parallelization. In fact, the implementation of the MARI-J method by the Karlsruhe group already allows for very large clusters in serial operation provided sufficient memory is available and rather long turn-around times are acceptable while still being very small compared to standard DFT or RI-J DFT. The master-slave concept is no longer adequate, memory requirements have to be reduced substantially by use of distributed data, and parallelization of a much larger number of computational steps is required. In view of the fast methodological development, serial and parallel code differ marginally in the actual quantum chemical code while a specialized set of library routines supports maintenance, parallelization or re-parallelization of existing code with little effort. The short hardware life cycle prohibits highly machine or architecture dependent implementations. The efficient exploitation of point group symmetry by the TURBOMOLE code is fully supported in the parallel implementation.

Serial linear algebra routines have to be replaced in many cases by parallel versions, either because the size of the matrices enforces distributed data or due to the cubic scaling with the problem size. In some cases, the replacement by alternative algorithms is more advantageous either due to better parallel scalability or more favorable cache usage.

The evaluation of a pairwise potential over a large number of particles is a rather widespread problem in the natural sciences. One way to avoid the quadratic scaling with the number of particles is the Fast Multipole Method (FMM) which treats a collection of distant charges as a single charge by expanding this collection of charges in a single multipole expansion. The FMM is a scheme to group the particles into a hierarchy of boxes and to manage the necessary manipulation of the associated expansions such that linear scaling is achieved.

An improved version of the FMM employing more stable recurrence relations for the Wigner rotation matrices and an improved error estimate has been implemented. The implementation is essentially parameter free: for a given requested accuracy the FMM specific parameters are determined automatically such that the computation time is minimized. The achieved accuracy is remarkable and competitive.

In addition, the Continuous Fast Multipole Method (CFMM), a generalization of the FMM for continuous charge distributions, has been implemented and incorporated into the DSCF module of the TURBOMOLE quantum chemistry package.

Conductor-like Screening Model
COSMOlogic

The treatment of solute-solvent interactions in quantum chemical calculations is an important field of application, since most practical problems are dealing with liquid phase chemistry. The explicit treatment of the solvent by placing a large number of solvent molecules around the solute requires apart from electronic also geometric relaxation of the complete solvent-solute system yielding this approach rather impractical. Continuum solvation models replace the solvent by a continuum which describes the electrostatic behavior of the solvent. The response of the solvent upon the polarization by the solute is represented by screening charges appearing on the boundary surface between continuum and solute. They, however, cannot describe orientation dependent interactions between solute and solvent. The particular advantage of the COSMO (Conductor-like Screening Model) formalism over other continuum models are the simplified boundary conditions.

Within the HPC-Chem project COSMO has been implemented for the HF and DFT methods (including energies, gradients and numerical second derivatives) as well as for the MP2 energies.

DFT Functionality in TURBOMOLE

Reinhart Ahlrichs and Klaus May

Institute for Physical Chemistry
University of Karlsruhe
Kaiserstr. 12, 76128 Kalrsruhe, Germany
E-mail: Reinhart.Ahlrichs@chemie.uni-karlsruhe.de

1 Introduction

The remarkable success of quantum chemistry, which could not have been anticipated 30 or 40 years ago, is a good example for the growing importance of scientific computing. This progress is clearly connected with the availability of computers with ever increasing performance at ever decreasing prices. Hardware is only one aspect, however, equally important for the impressive achievements of quantum chemistry have been software developments aiming at novel modeling methods and improved algorithms, which together resulted in great gains in efficiency. We thus have presently at our disposal an arsenal of computational procedures which covers very accurate calculations for small molecules (10 to 20 atoms) up to more approximate methods applicable to clusters with 1000 atoms.

Larger clusters are typically treated with DFT (density functional theory) methods employing functionals of GGA type (generalized gradient approximation), which have become available only in the late eighties [1, 2, 3]. DFT-GGA calculations are more accurate than HF (Hartree-Fock) and are applicable to a much wider class of chemical compounds, such as transition metal complexes for which HF very often fails; they are further 10 to 100 times faster than present-day HF routines and 100 to 1000 times faster than HF implementations of the 60s, i.e. before the invention of direct HF procedures (DSCF = Direct Self Consistent Field) [4], efficient integral prescreening [5] and evaluation procedures.

The just given example demonstrates the benefits of software developments but it also indicates a problem: computational procedures often become obsolete after 5 to 10 years. This then does not leave sufficient time for proper software engineering (to convert 'academic code' to a product) required e.g. for parallelization. A second important aim of HPC-Chem was, therefore, to better implement the parallelization of TURBOMOLE to facilitate maintaining the code and to increase efficiency, of course. The corresponding work was carried out by project partners from Jülich and is described in the chapter IV.

The main goal of TURBOMOLE work packages within HPC-Chem was to further increase efficiency and functionality of the program as specified in the proposal. The work plan was focused on the development of procedures especially tailored to the treatment of large molecules. The results will be reported in this article. The presentation of results will be preceded by a short description of TURBOMOLE and a brief account of the theoretical background to prepare for the method developments described thereafter.

2 About TURBOMOLE

The Theoretical Chemistry group of Karlsruhe was (among) the first to seriously test and exploit the use of workstations for molecular electronic structure calculations when the new hardware became available in the late eighties. In a series of diploma and PhD theses an available HF code was adapted to UNIX workstations with the aim to do large molecules on small computers. Thanks to the algorithmic developments of the excellent students M. Bär, M. Häser, H. Horn and C. Kölmel the ambitious project was completed successfully and TURBOMOLE was announced in 1989 [6].

In the time to follow we have continuously added new features if they appeared promising for the treatment of large molecules. The present program version 5.7 covers HF, DFT [7], MP2 [8, 9] and CC2 [10] treatments of (electronic) ground state properties such as energies, optimization of structure constants, chemical shifts of NMR, and nuclear vibrations. Electronic excitations and time-dependent properties are covered by linear response procedures for DFT (usually called TD-DFT) [11], HF (RPA-HF) [12, 13] and CC2 [14]. The implementations include optimizations of molecular structure for excited states on the basis of analytical gradients. For more details the reader is referred to the user's manual (`http://www.turbomole.com`).

Let us finally mention the two essential features of TURBOMOLE which are the basis of its competitiveness and strength - in the not unbiased view of the authors. The codes exploit molecular symmetry for all point groups in most modules (exceptions are groups with complex irreps for NMR). This reduces CPU times by roughly the order of the symmetry group, i.e. by a factor of about 12 for D_{3h} or D_{3d} , and a factor of 48 for O_h. Most other programs take only advantage of Abelian groups, i.e. D_{2h} and subgroups. The other specialty concerns the RI technique [15], for resolution of the identity, which will be discussed in the following section.

3 Theoretical background: HF, DFT, and the RI technique

3.1 HF and DFT

As a preparation for the subsequent sections it is appropriate to briefly sketch relevant features of HF and DFT. These methods are of single reference type and are fully specified by the MOs ϕ_i and their occupation numbers n_i. For the sake of simplicity we consider only closed shell cases with n_i = 2 for occupied MOs ϕ_i (and n_a = 0 for virtual MOs ϕ_a). The total electronic energy of HF and DFT then includes the one-electron term $E^{(1)}$, the Coulomb interaction of electrons J, the HF exchange K and the exchange-correlation term E_{XC} of DFT

$$\begin{aligned} E_{\text{HF}} &= E^{(1)} + J - K && (1) \\ E_{\text{DFT}} &= E^{(1)} + J - E_{XC}. && (2) \end{aligned}$$

The DFT expression applies for non-hybrid functionals only, hybrid functionals include part of the HF exchange, i.e. a term -$C_X K$ with $0 < C_X < 1$. The evaluation of $E^{(1)}$ is straightforward and fast. E_{XC} is defined as a three-dimensional integral

$$\begin{aligned} E_{XC} &= \int d\tau f_{XC} \left(\rho(r), |\nabla\rho(r)|^2, ...\right) && (3) \\ \rho(r) &= 2\sum_i |\phi_i(r)|^2 && (4) \end{aligned}$$

where f_{XC} specifies the actual functional employed. Eq. (3) is evaluated by numerical integration (quadrature) and the procedure implemented in TURBOMOLE is efficient and numerically stable [7], the CPU time increases linearly with molecular size for large cases, i.e. an O(N) procedure, as demonstrated below.

For DFT it remains to consider J, which is defined as

$$J = \frac{1}{2}\int d\tau\rho(r_1)\rho(r_2)\,|r_1 - r_2|^{-1}\,. \qquad (5)$$

The evaluation of J is typically the most demanding part of DFT treatments. With the usual expansion of ϕ_i in a set of basis functions $f_\nu(r)$

$$\phi_i(r) = \sum_\nu f_\nu(r) C_{\nu i} \qquad (6)$$

one gets the density ρ and the density matrix $\mathbf{D}$

$$\begin{aligned} \rho(r) &= \sum_{\nu\mu} D_{\nu\mu} f_\nu(r) f_\mu(r) && (7) \\ D_{\nu\mu} &= 2\sum_i C_{\nu i} C_{\mu i} && (8) \\ \text{and} \\ J &= \frac{1}{2} \sum_{\nu\mu\kappa\lambda} D_{\nu\mu} D_{\kappa\lambda} (\nu\mu|\kappa\lambda) && (9) \\ K &= \frac{1}{4} \sum_{\nu\mu\kappa\lambda} D_{\nu\kappa} D_{\mu\lambda} (\nu\mu|\kappa\lambda) && (10) \\ (\nu\mu|\kappa\lambda) &= \int f_\nu(r_1) f_\mu(r_1) f_\lambda(r_2) f_\kappa(r_2) |r_1 - r_2|^{-1} d\tau && (11) \end{aligned}$$

where K is given for completeness. The MOs are now specified by the coefficients $C_{\nu i}$ and the chosen basis set, of course. Optimization of $\mathbf{C}$ within the variation principle yields the HF and Kohn-Sham (KS) equations to be solved

$$\begin{aligned} F_{\nu\mu} &= \frac{\partial E}{\partial D_{\nu\mu}} && (12) \\ \sum_\mu F_{\nu\mu} C_{\mu i} &= \epsilon_i \sum_\mu S_{\nu\mu} C_{\mu i} && (13) \end{aligned}$$

where $\mathbf{S}$ denotes the overlap matrix.

In HF one evaluates J and K together, it is a great advantage of DFT that K does not occur in (2). Since only J has to be treated other procedures - than (9) - can be considered, and this has been done from the very beginning of DFT or X$_\alpha$ theory.

3.2 RI technique

One of the successful procedures [16, 17] was to approximate ρ in terms of an auxiliary or fitting basis P

$$\rho(r) \approx \tilde{\rho}(r) = \sum_P C_P P(r). \tag{14}$$

The free parameters C_P are obtained from a least squares requirement

$$< \rho - \tilde{\rho} | \rho - \tilde{\rho} >= min \tag{15}$$

which yields

$$\sum_P < Q|P > C_P =< Q|\rho > . \tag{16}$$

It remains to specify the scalar product occurring in the last two equations. A careful analysis by Almlöf et al. has identified the best choice [18]

$$< f|g >= \int f(r_1)g(r_2)|r_1 - r_2|^{-1} d\tau. \tag{17}$$

Straightforward algebra then yields

$$J = \frac{1}{2} < \rho|\rho > \approx \tilde{J} = \frac{1}{2} \sum_{P,Q} < \rho|P >< P|Q >^{-1} < Q|\rho > \tag{18}$$

where $< P|Q >^{-1}$ denotes matrix elements of the inverse of $< P|Q >$, and all scalar products are understood to be as in (17). The form of (18) has lead to the label RI (for resolution of the identity) for this technique.

With the basis set expansion for ρ, Eq. (7), it then remains to compute as the essential term

$$< \rho|P >= \sum_{\nu\mu} D_{\nu\mu} < f_\nu f_\mu|P > . \tag{19}$$

The formal O(N^4) behavior of (9) is thus replaced by a formal O(N^3) scaling in (19) leading to considerable savings in CPU time [15]. With the usual choice of Gaussian basis functions one can neglect $f_\nu f_\mu$ if the corresponding centers are sufficiently far apart; the number of significant products $f_\nu f_\mu$ thus increases for large molecules only as O(N). This results in an asymptotic O(N^2) scaling for RI and conventional treatments - with a much smaller prefactor for the RI technique.

Although the RI procedure had been implemented in various DFT programs, its accuracy had not been systematically tested since the programs could only compute $\tilde{J}$ and not the rigorous expression (9) for J. It was also unsatisfactory that the important auxiliary functions P had not been carefully optimized.

We therefore started a major effort to carefully optimize auxiliary basis sets for atoms across the periodic table and to document the errors caused by the RI technique [15, 19]. This firmly established reliability, it also increased efficiency since optimized sets do not only guarantee more accurate results, they can often be chosen smaller than 'guessed' bases. The Karlsruhe auxiliary basis set are now available for different accuracy requirements for RI-DFT and also for RI-K[20], RI-MP2 and RI-CC2 calculations [21, 22, 23], which will not be discussed here - but these bases are made available for other projects within HPC-Chem. There appear to be no other auxiliary basis sets which are comparable in accuracy and efficiency.

3.3 Gradients

Until now we have considered so called 'single point' calculations which yield the molecular electronic structure (occupied MOs ϕ_i) and the electronic energy for given nuclear coordinates. It is not possible, however, to determine the most important molecular properties efficiently on the basis of single point calculations. As an example consider molecular equilibrium geometries, i.e. structures of one (or more) isomers of a molecule defined as

$$E^{\xi} = \frac{dE}{d\xi} = 0 \tag{20}$$

where ξ denotes structure constants, e.g. coordinates of nuclei. An attempt to locate structures by single point calculations would hardly be feasible even for small molecules with ten degrees of freedom, f=10.

A solution to this problem was achieved by analytical gradient methods, which evaluate E^{ξ} simultaneously for all degrees of freedom [24]. The computation of E^{ξ} is surprisingly simple in principle, if one recalls that E depends explicitly only on ξ (location of nuclei including the centers of basis functions) and on the density matrix, i.e. $E = E(\xi, \mathbf{D})$, where $\mathbf{D}$ depends implicitly on ξ. Thus

$$\frac{dE}{d\xi} = \frac{\partial E}{\partial \xi} + \frac{\partial E}{\partial \mathbf{D}} \cdot \frac{\partial \mathbf{D}}{\partial \xi}. \tag{21}$$

The first term can be straightforwardly treated since its structure is similar to the evaluation of E in a single HF or DFT iteration, only the effort is about three times larger. The second term can be transformed since one has solved a HF or KS equation before, i.e. one exploits that MOs ϕ_i have been optimized and are orthonormal

$$\frac{\partial E}{\partial \mathbf{D}} \cdot \frac{d\mathbf{D}}{d\xi} = -tr\mathbf{W}\mathbf{S}^{\xi} \tag{22}$$

where $\mathbf{S}^{\xi}$ denotes the derivative of the overlap matrix and $\mathbf{W}$ the 'energy weighted' density matrix

$$W_{\nu\mu} = 2\sum_i \epsilon_i c_{\nu i} c_{\mu i}. \tag{23}$$

With the capability to compute E^{ξ} it is a standard task to locate in an iterative procedure structures that fulfill (20):

1. starting from a reasonable guess ξ_0 for ξ
2. solve the HF or DFT equations to get optimized MOs
3. compute E^{ξ}
4. relax the structure $\xi_0 \rightarrow \xi$, e.g. by conjugate gradient methods
5. repeat until convergence.

The development of efficient gradient procedures together with reliable and stable relaxation methods was decisive for the success of quantum chemistry. Since convergence of the relaxation procedure is typically reached within $f/2$ cycles (often less, rarely more), and since the computation of E^{ξ} is (often much) faster than a DFT or HF calculation, structure determinations, which are the bread and butter of quantum chemistry, have become routine.

4 The MARI-J (Multipole Assisted RI-J) procedure

It has already been mentioned that the RI-J method is an $O(\mathrm{N}^2)$ procedure for large molecules, e.g. more than 100 atoms, whereas the other demanding computational task, the evaluation of E_{XC}, scales as $O(\mathrm{N})$. It was the aim of this project to increase efficiency of the RI-J procedure by exploiting the multipole expansion for the Coulomb interaction of (non-overlapping) charge distributions. Since details of the rather technical derivations have been documented in a publication [25] we will only sketch our approach.

The multipole expansion deals with the Coulomb interaction of two charge distributions ρ_1 and ρ_2, provided they do not overlap. Let ρ_1 be centered around A and ρ_2 around B. We then compute the moments of ρ_1 as

$$\Omega^{\mathrm{A}}_{lm} = \int \rho_1(r) O_{lm}(r - \mathrm{A}) d\tau \tag{24}$$

$$O_{lm}(x) = \frac{|x|^l}{(l+|m|)!} P_{lm}(\cos\theta) e^{-im\phi} \tag{25}$$

where P_{lm} denote associated Legendre polynomials, and similarly for Ω^{B}_{lm} referring to ρ_2. One then gets

$$< \rho_1|\rho_2 > = \sum_{ljkm} \Omega^{\mathrm{A}}_{lm} M_{l+j,m+k}(R) \Omega^{\mathrm{B}}_{jk} \tag{26}$$

$$M_{lm}(R) = \frac{(l-|m|)!}{|R|^{l+1}} P_{lm}(\cos\theta) e^{im\phi} \tag{27}$$

where R denotes the vector pointing from A to B: $R =$ B-A, and the angles θ and ϕ of respective vectors are defined in an arbitrary fixed coordinate system. Eq. (26) effects a separation of Coulomb interactions between ρ_1 and ρ_2 if they do not overlap.

The computation of $< \rho|P >$, Eq. (19), is the only demanding task within RI-J, and we apply the multipole expansion to accelerate the evaluation. For this purpose we decompose ρ into contributions associated with nuclei N, which are additionally characterized by an extension e

$$\rho = \sum_{N,e} \rho_{N,e}. \tag{28}$$

We then compute the moments $\Omega^{N,e}_{lm}$ from (24), where we have put $\rho_1 = \rho_{N,2}$ and have chosen for A the position of nucleus N. The auxiliary functions P are by construction atom-centered and are further chosen with angular behavior as spherical harmonics; the evaluation of the corresponding moment $\Omega P^{M,e}_{j,k}$ is thus trivial.

The crucial point of this procedure is the decomposition (28), which is based on a detailed consideration of products of basis functions $f_\nu f_\mu$. Depending on the actual case the product is associated with the nuclei N at which the steeper function is centered, and an appropriate extension established. One then has to look at individual $< \rho_{N,e}|P >$. If the charge distributions $\rho_{N,e}$ and P are well separated, for which we have introduced a new and strict test, one uses the multipole expansion, the so called 'far field' (FF) contribution

$$< \rho_{N,e}|P >_{\mathrm{FF}} = \sum \Omega^{N,e}_{l,m} M_{e+j,m+k}(R) \Omega P^{M,e}_{j,k} \tag{29}$$

where R points from nucleus N to P. For the remaining terms one cannot apply this formula since the charge distributions penetrate and one bas to use for this 'near field' (NF) part the conventional integral code:

$$< \rho_{N,e}|P >_{\mathrm{NF}} = \text{usual integrals.} \tag{30}$$

Our aim is to define parameter sets for the MARI-*J* method that yield the shortest CPU times while maintaining errors due to the multipole expansions at a minimum level, below the errors of the RI-*J* approximation. We specify two parameter sets, which incur the shortest CPU times while maintaining a minimum precision for each calculated molecule corresponding to 1×10^{-6} and 1×10^{-4} E_h, respectively. They are hereafter referred to simply as the high- and low-precision sets. Table 1 lists errors obtained for the largest molecules studied, specified in more details below. The errors are evaluated by comparing converged total MARI-*J* energies with results of full RI-*J* calculations. Using the high-precision parameter set yields errors below 1.0×10^{-6} E_h, which corresponds to no more than 1.0×10^{-9} E_h per atom for the largest molecules in the series. As expected, the smallest errors, less than 1×10^{-7} E_h, are observed for two-dimensional graphitic sheets and for relatively low-density zeolite clusters. Using the low-precision set gives errors lower than 1×10^{-4} E_h, which amounts to no more than 1×10^{-7} E_h per atom for the largest systems. For the insulin molecule, the high-precision MARI-*J* calculation yields total energy differing 1.3×10^{-8} E_h from the full RI-*J* calculation. Surprisingly, the low-precision calculation yields a difference of only 2.2×10^{-8} E_h. This is a similar result as for the zeolite fragments and shows that the low-precision MARI-*J* calculations are accurate enough for most applications. Only for dense three-dimensional systems, or systems with very diffuse basis sets one should consider using the high-precision parameter set. We conclude that the errors introduced by the multipole expansions are negligible compared to the errors of the RI-*J* approximation itself, incomplete basis sets and numerical integrations.

4.1 Demonstrative tests

This section describes the application of the MARI-*J* method to some model systems: graphitic sheets, zeolite fragments and insulin molecule (Figure 1). We believe that this choice of the systems corresponds more to the problems that DFT methods are typically applied to than the usual one and two dimensional model systems used to test the $O(N)$ algorithms.

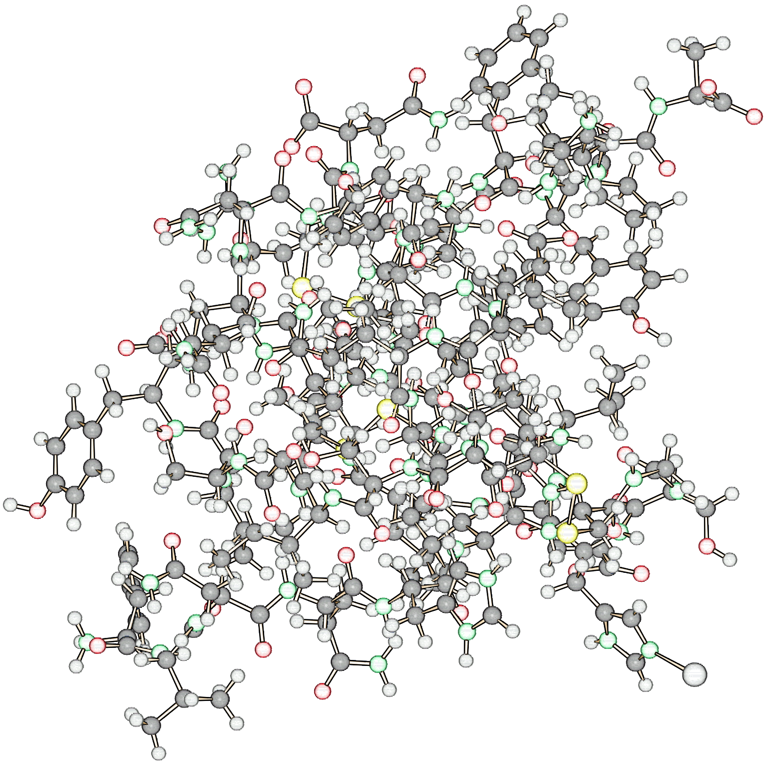


Figure 1: Schematic draw of the insulin molecule used for our calculations.

All DFT calculations employ the Becke-Perdew (BP86) exchange-correlation functional [26, 27, 28]. Unless specified otherwise we use split-valence basis sets with polarization functions on all non-hydrogen atoms, denoted SV(P) [29] and corresponding auxiliary bases [15, 19]. To determine the differences of total energies between the MARI-J method and full RI-J treatment the DFT energies are converged better than 1×10^{-10} E_h and the numerical integrations use grid 3 (see Ref. [7] for details). For the timing runs the energy convergence criterion is set to 1×10^{-6} E_h and numerical integrations use grids m3 and m4[7, 19]. Whereas grid m3 is recommended for smaller molecules grid m4 should be used for larger ones. All calculations are performed on an HP J6700 workstation with a PA RISC HP785 processor (750 MHz) and 6 GB main memory.

4.1.1 Model systems

The 2-D series of graphitic sheets, $C_{6n^2}H_{6n}$, $n = 2, \ldots, 12$, all in D_{6h} symmetry, have C-C and C-H bond lengths set to 1.42 and 1.0 Å, respectively. These are similar models as used by Strain et al.[30] and Pérez-Jordá and Yang[31] to asses performance of their multipole-based methods. The largest sheet used in this study, $C_{864}H_{72}$, contains 12240 basis and 32328 auxiliary basis functions.

Table 1: Selected results for the largest model systems studied. Number of atoms ($N_{\rm at}$), number of basis functions and auxiliary basis functions ($N_{\rm bf}$), CPU times (min) per iteration for the NF ($t_{\rm NF}$) and FF ($t_{\rm FF}$) portions of the MARI-*J* calculations and the full RI-*J* treatment ($t_{\text{RI-}J}$), and absolute errors in the total energies ($E_{\rm h}$) compared to the full RI-*J* calculations ($\Delta E_{\rm MA}$). Results for high- (hp) and low-precision (lp) MARI-*J* calculations (see text for details). For comparison CPU timings for grid construction (grid m4) are given ($t_{\rm grid}$).

	Graphitic sheets	Zeolite fragments	Insulin molecule
Composition	$C_{864}H_{72}$	$Si_{96}O_{216}H_{48}$	$C_{256}H_{383}O_{76}N_{65}S_6Zn$
Symmetry	D_{6h}	C_1	C_1
$N_{\rm at}$	936	360	787
$N_{\rm bf}$	12240	4848	6456
$N_{\rm bf}$ (aux)	32328	12072	16912
$t_{\rm NF}$ (hp)	4.9	11.5	35.1
$t_{\rm FF}$ (hp)	2.3	1.26	3.4
$t_{\rm NF}$ (lp)	4.1	8.1	25.3
$t_{\rm FF}$ (lp)	1.6	0.9	2.3
$t_{\text{RI-}J}$	33.7	58.6	147.8
$\Delta E_{\rm MA}$ (hp)	1.6×10^{-8}	6.3×10^{-8}	1.3×10^{-8}
$\Delta E_{\rm MA}$ (lp)	6.1×10^{-5}	2.6×10^{-7}	2.2×10^{-8}
$t_{\rm grid}$ (m4)	18.1	30.3	47.0[a]

The fragments of pure-silica zeolite chabazite are constructed from the experimental crystal structure [32]. We take a unit cell consisting of a double six-membered silica ring unit and create zeolite fragments containing between one and eight such units, all in C_1 symmetry. The dangling Si-O bonds are saturated with hydrogen atoms. The coordinates of the insulin molecule (Figure 1), [33] in C_1 symmetry, are taken form the PDB database [34]. It comprises 787 atoms, 6456 basis and 16912 auxiliary basis functions. Table 1 summarizes the largest molecules calculated in this study. The coordinates of all structures are available in internet under `ftp://ftp.chemie.uni-karlsruhe.de/pub/marij`.

4.1.2 Timings and scaling

First, we would like to comment on the often cited $O(N^3)$ computational effort of the RI-*J* method due to the density fitting step, i.e. solution of Eq. (16). The TURBOMOLE implementation of the RI-*J* method is based on a very fast Cholesky decomposition of the positive definite matrix $<P|Q>$. For symmetric molecules the times needed to calculate the fully symmetric part of two-center repulsion integrals $<P|Q>$ and following Cholesky decomposition are negligible. For the insulin molecule with C_1 symmetry, 787 atoms and 16912 auxiliary basis functions this step takes approximately 20 min, and is done only once at the beginning of the SCF procedure for both RI-*J* and MARI-*J* calculations. For most of the practical calculations the cost and scaling behavior of the RI-*J* method is determined by the calculation of three-center Coulomb integrals. For very large systems methods can be implemented which reduce the computational cost of the density fitting step even below $O(N^2)$ [35, 36].

Figures 2 and 3 show CPU times per SCF iteration for the system series studied using the RI-*J* and MARI-*J* methods and Table 1 summarizes results for the largest molecules. For comparison, the times needed for evaluation of exchange-correlation energies with grids m3 and m4 are also shown. These timings do not include the costs of the grid formation, which is done only once at the beginning of the SCF procedure. Table 1 shows timings of this step for the largest molecules. In most cases the application of the MARI-*J* method allows one to reduce the computational effort for the Coulomb term to a level comparable to the calculation of the exchange-correlation energy. The MARI-*J* method shows the best performance for two-dimensional graphitic sheets and zeolite fragments. For the largest graphitic sheet the CPU times are reduced 4.7 and 5.9 times for high and low-precision parameters sets, respectively, as compared to the full RI-*J* calculation. A similar reduction of the CPU times (factors 4.6 and 6.5) is observed for the largest zeolite fragment. For the insulin molecule we obtain 3.8 and 5.3-fold speedups.

For all systems studied the “crossover point” with full RI-*J* treatment is reached already for the smallest systems. For graphitic sheets and zeolite fragments the MARI-*J* calculations are already faster at about 250-350 basis functions, depending on the accuracy. A few test calculations on even smaller systems show that the MARI-*J* does not introduce any significant overhead compared to the full RI-*J* treatment.

The influence of the required precision on the CPU timings for the MARI-*J* method depends on the system studied. For graphitic sheets, zeolite clusters and diamond pieces the difference between CPU times for high and low-precision MARI-*J* calculations is about 30%.

Table 1 also shows a comparison of the CPU times for NF and FF portions of the Coulomb calculations for the largest systems in each series. Although only a few percent of the three-center ERIs are evaluated analytically the NF part still dominates the calculations. For the molecules with C_1 symmetry the FF part of the MARI-*J* calculations takes 10% or less of the CPU time for the Coulomb part. For symmetric molecules the CPU times for the FF part increase to 20-30 %. The current implementation of the MARI-*J* method does not fully take advantage of symmetry in the calculations of the FF part. Symmetry implementation in all parts of the MARI-*J* algorithm should reduce these times but would only slightly influence the total calculation times.

We note, that all calculations reported here employ the standard SCF procedure, and the diagonalization of the Fock matrix is not a dominant step. For the insulin molecule the average CPU times per SCF iteration are 42, 17, and 39 or 28 CPU minutes for the diagonalization, exchange-correlation, and high- or low-precision MARI-*J* steps, respectively.

Calculation		Graphitic sheets	Zeolite fragments
Full RI-J		2.00	2.11
MARI-J	Total	1.44	1.54
high precision	NF	1.41	1.54
	FF	1.49	1.52
MARI-J	Total	1.47	1.56
low precision	NF	1.45	1.57
	FF	1.51	1.45
XC (m4)		1.23	1.33
XC (m3)		1.23	1.34
N_{dist}		1.09	1.20

Table 2: Scaling exponents of different steps for the computation of the Coulomb and exchange-correlation (grids m3 and m4) terms. For comparison the scaling exponents of significant shell-pairs of basis functions (N_{dist}) are also shown.

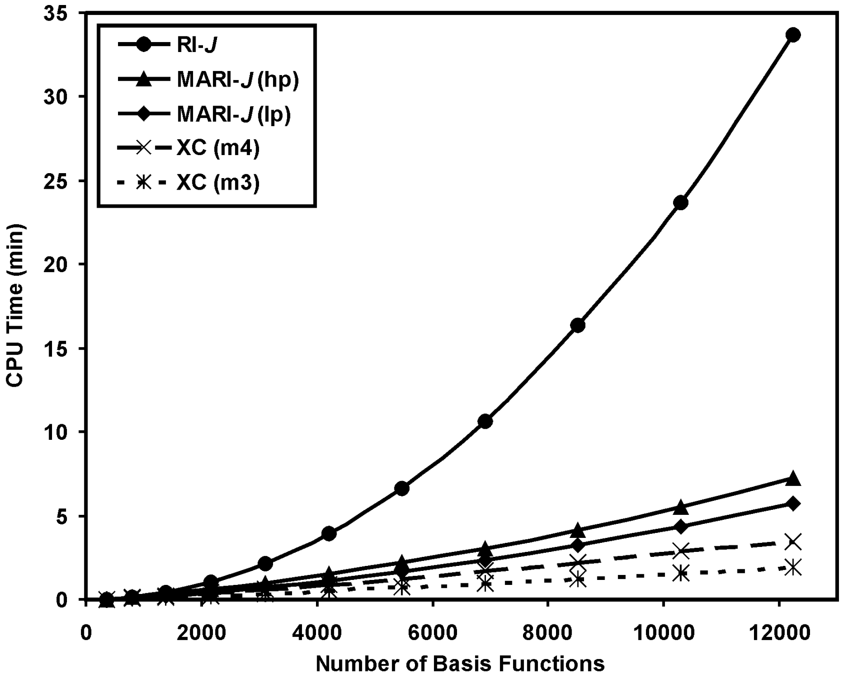


Figure 2: CPU time per SCF iteration for calculation of the Coulomb term versus the number of basis functions in a series of graphitic sheets, $C_{6n^2}H_{6n}$, $n = 2, \ldots, 12$. Results for full RI-J calculations and MARI-J with high- (hp) and low-precision (lp) parameters sets. For comparison CPU times needed for evaluation of exchange-correlation energy (XC) with grids m3 and m4 are included.

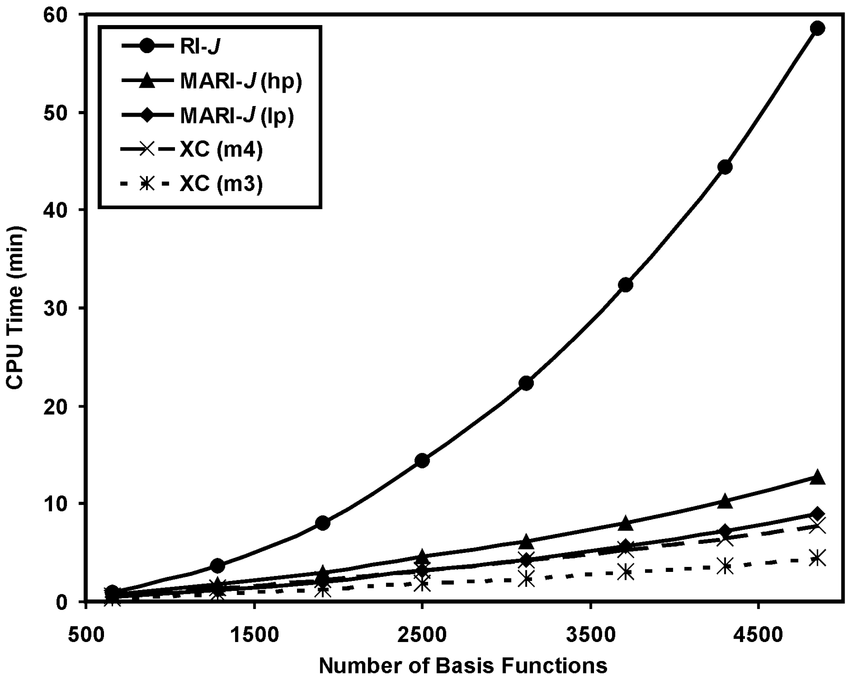


Figure 3: CPU time per SCF iteration for calculation of the Coulomb term versus the number of basis functions in a series of zeolite fragments. Results for full RI-J calculations and MARI-J with high- (hp) and low-precision (lp) parameters sets. For comparison CPU times needed for evaluation of exchange-correlation energy (XC) with grids m3 and m4 are included.

Table 2 shows scaling exponents of different calculation steps for the systems studied. They are obtained by a logarithmic fit using results for the largest molecules in each series. As expected the exponents for the full RI-J calculations are close to 2.0 and are larger for dense three-dimensional systems than for graphitic sheets and zeolite fragments. The scaling exponent is reduced to about 1.5 for MARI-J, and that for E_{XC} is about 1.3, i.e. we have nearly linear scaling. As expected, the scaling exponents of the RI-J calculations are closely related to the scaling exponents of numbers of significant products of basis functions comprising the electron density as shown in Table 2. The thresholding procedure applied to multipole moments, as described in [25], significantly reduces the formal $O(N^2)$ scaling behavior of the FF part of the MARI-J calculations. The scaling exponents are lowered by more than 0.5. For zeolite fragments and diamond pieces changing from high- to low-precision parameter set lowers the scaling exponents for the FF part. For graphitic sheets the scaling exponent increases slightly when going from high- to low-precision parameters. It is difficult to say *a priori* whether increasing the precision of the MARI-J calculations causes also increases in the scaling exponent of the FF part.

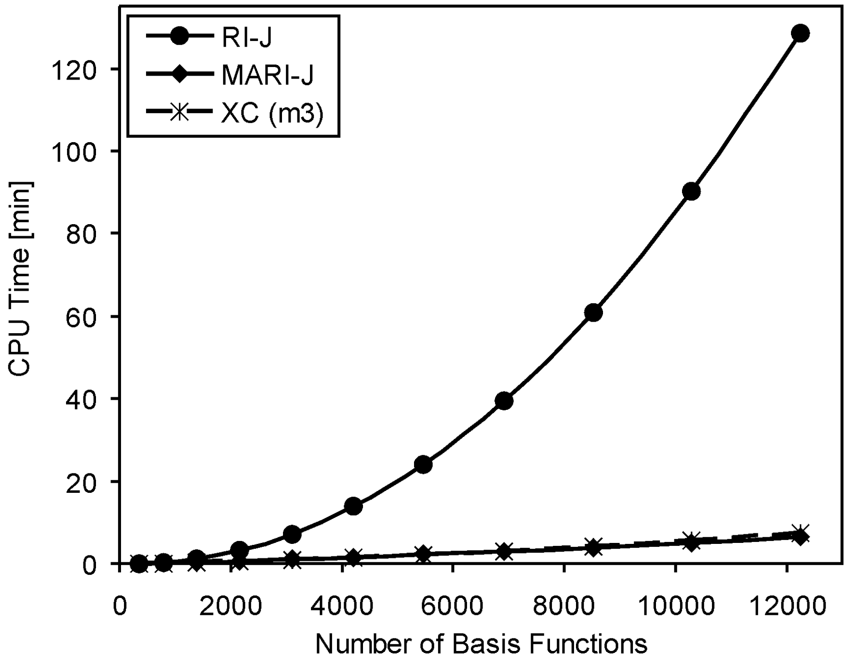


Figure 4: Comparison of timing for various parts of gradient calculations for graphitic sheets

4.2 MARI-J Gradient evaluation

Geometry optimization require a succession of energy and gradient calculations, and it is highly desirable to take advantage of the multipole expansion in both steps. The implementation of the MARI-J gradient is a demanding technical task. We will thus not go into the details [37] and will merely report the results of our efforts. In Figure 4 we show a comparison of CPU times for various parts of the gradient calculations for the case of graphitic sheets. The state of affairs is even better than for the energy calculation: timings for the Coulomb term J are reduce by a factor of 15, they are now comparable to the XC term.

5 DFT second analytical derivatives

We have already pointed out that first order derivatives E^{ξ} can be computed faster than the energy in HF or DFT, and that this is vital for daily routine in theoretical treatments of molecules. Even more useful would be the knowledge of second derivatives, the so called Hessian.

$$H_{\xi\eta} = \frac{d^2E}{d\xi d\eta}. \tag{31}$$

The Hessian determines if a given stationary point, Eq. (20), is a local minimum specifying an isomer, or a saddle point which characterizes a transition state of a reaction, provided only one eigenvalue of $\mathbf{H}$ is negative. From the Hessian one also gets quite directly the frequencies of infra-red (IR) and Raman spectra within the harmonic approximation.

The explicit expression for $H_{\xi\eta}$ is best obtained by differentiating $\frac{dE}{d\xi}$, Eq. (21) to Eq. (23), once more with respect to a second coordinate η

$$H_{\xi\eta} = \frac{d}{d\eta}\left(\frac{\partial E}{\partial \xi} - tr\mathbf{W}\mathbf{S}^{\xi}\right). \tag{32}$$

The detailed formulae, which are quite lengthy, need not concern us here; the most important aspect of (32) is that one now has to compute the perturbed MOs, i.e. $\frac{d}{d\xi}C_{\nu i} = C^{\xi}_{\nu i}$. This leads to the so called coupled perturbed HF or KS equations (CPHF or CPKS) which are typically solved in the following way. One expresses the perturbed MO in terms of the unperturbed ones with the help of a transformation matrix $\mathbf{U}^{\xi}$,

$$C^{\xi}_{\nu i} = \sum_{q} C_{\mu q} U^{\xi}_{qi} \tag{33}$$

which is determined by HF-type equations

$$\begin{aligned}
(\epsilon_i - \epsilon_a) U^{\xi}_{ai} &- 4G_{ai}\left[\mathbf{U}^{\xi}\right] = RHS^{\xi}_{ai} && (34)\\
G_{p,q}\left[\mathbf{M}^{\chi}\right] &= \sum_{rs\nu\mu\kappa\lambda} C_{\nu p} C_{\mu q} C_{\kappa r} C_{\lambda s} \cdot \{(\nu\mu|\kappa\lambda) + f_{XC\nu\mu\kappa\lambda}\} M^{\chi}_{rs} && (35)\\
f_{XC\nu\mu\kappa\lambda} &= \frac{1}{2}\int \frac{\partial^2}{\partial\rho^2}\phi_\nu\phi_\mu\phi_\kappa\phi_\lambda d\tau && (36)
\end{aligned}$$

The technical details are complicated but the essential point is that one necessarily has to solve a CPHF or CPKS equation for every degree of freedom f, i.e. the evaluation of the Hessian is at least f times as expensive as a single point or a gradient calculation. Since f increases linearly with molecular size, $f = O$(N), the computation of the Hessian is O(N^3). This is O(N) more than for the gradient E^{ξ} but one also gets O(N) more information. We have demonstrated above that energy and gradient calculations can be done with about O(N) effort - but second derivatives can presently only be treated for molecular sizes for which the reduced scaling does not apply.

The challenge in the development of DFT or HF implementations is not only computational demands, it is also algorithmic complexity, since one really has to cope with lengthy expressions. Our plan was originally to start from the second derivatives HF code of TURBOMOLE and to add DFT. It was decided, however, to restructure the existing code which would have lead to efficiency problems for systems with 50 or more atoms.

Our implementation of DFT second derivatives has been described in a publication together with some applications to demonstrate efficiency and accuracy [38]. The code shares some general features with other programs, e.g. GAUSSIAN, such as integral direct and multigrid techniques for the CPKS equations, and inclusion of weight derivatives in the quadrature. Other features to increase efficiency appear to be unique for TURBOMOLE.

The iterative solution of CPKS is based on a preconditioned conjugate gradient method with subspace acceleration, i.e. all solution vectors are developed in a single subspace which is enlarged in each iteration. This guarantees good convergence: typically four to six iterations suffice to push the residual norm below the target of 10^{-5}.

We decompose the space of f internal coordinates into irreducible subspaces (of the molecular symmetry group). This reduces memory and disc storage requirements since occurring matrices are symmetry blocked, which also facilitates treatment of CPKS and enhances efficiency. It is a further advantage that the evaluation of the Hessian can be limited to certain irreps, e.g. those corresponding to IR- or Raman active modes.

All following results were obtained by force constant calculations using the new TURBOMOLE module **AOFORCE** with the BP86 DFT-method. We employed an SV(P) (split valence plus polarization except at hydrogen) [29] basis set. To give an impression of computational cost for systems of different size and symmetry, we display in Table 3 total CPU times (wall times do not differ by more than 2 %) and their most important components in benchmark calculations on several hydrocarbons. The molecules treated are n-alkanes (of formula C_5H_{12} to $C_{37}H_{76}$), planar graphitic sheets (which results in compositions from C_6H_6 to $C_{48}H_{18}$), and diamond like sp^3 carbon clusters (starting with adamantane, $C_{10}H_{16}$, and creating further clusters by adding larger sheets from one side up to $C_{51}H_{52}$). The alkanes were treated in their C_{2v} structure, the aromatic sheets in D_{2h}, and the diamond clusters in T_d. The CPKS equation solver needed four iterations for each alkane and diamond cluster and five iterations for each aromatic sheet, thus matrices $\mathbf{G}[M^{\chi}_{pq}]$ had to be formed six times for each of the first and seven times for the latter compounds. As can be seen, the total CPU time increases approximately as $O(N^3)$ for larger systems, which is due to the evaluation of the Coulomb part of $\mathbf{G}[M^{\chi}_{pq}]$ discussed above Eq. (35). The effort for first weight derivatives needed in both $\mathbf{F}^{(\chi)}$ and $E^{(\chi)(\xi)}$ [1] is negligible. For smaller molecules, the DFT quadrature in $\mathbf{F}^{(\chi)}$ is clearly more expensive than the differentiated four-center integrals - in larger systems like $C_{48}H_{18}$ these two contributions exhibit similar timings.

As a demonstration for the 'IR-only' and 'Raman-only' option we have treated fullerene C_{60}, again on AMD-Athlon, 1.2 GHz:

C_{60} (I_h), 900 BF	all irreps	t=3.40 h
	T_{1u} (IR)	t=1.21 h
	A_g and H_g (Raman)	t=1.61 h

[1] $E^{(\chi)(\xi)} = \frac{\partial^2 E}{\partial\lambda\partial\xi}$, $F^{(\chi)} = \frac{\partial F}{\partial\chi}$ which is a contribution to the RHS^{χ}_{ai} in Eq. (34)

Table 3: CPU times and its most important constituents (in hours) on AMD-Athlon (1.2 GHz, 768 MB RAM) for BP86/SV(P) force constant calculations of various classes of hydrocarbons. f denotes the number of degrees of freedom and N_{BF} the number of basis functions. For the definition of the particular CPU time contributions see text.

molecule	f	N_{BF}	$\mathbf{G}[M_{pq}^{\chi}]$	$\mathbf{F}^{(\chi)}$	$E^{(\chi)(\xi)}$	total
linear alkanes (C_{2v})						
C_5H_{12}	45	99	0.04	0.02	0.03	0.09
$C_{13}H_{28}$	117	251	0.54	0.16	0.24	0.95
$C_{21}H_{44}$	189	403	2.27	0.41	0.68	3.51
$C_{29}H_{60}$	261	555	5.60	0.82	1.40	8.32
$C_{37}H_{76}$	333	707	10.37	1.46	2.37	15.47
aromatic sheets (D_{2h})						
C_6H_6	30	102	0.02	0.01	0.02	0.04
$C_{16}H_{10}$	72	260	0.36	0.13	0.23	0.73
$C_{30}H_{14}$	126	478	2.76	0.58	1.11	4.58
$C_{48}H_{18}$	192	756	12.21	1.90	3.40	18.24
diamond clusters (T_d)						
$C_{10}H_{16}$	72	182	0.07	0.04	0.03	0.15
$C_{26}H_{32}$	168	454	1.63	0.38	0.45	2.64
$C_{51}H_{52}$	303	869	13.48	2.52	2.47	20.48

5.1 Implementation of RI-J for second derivatives

We have so far considered the computation of analytical second derivatives in general and have pointed out that the most demanding task is the solution of CPKS equations. In each CPKS iteration one has to evaluate a Coulomb term J (and for hybrid functionals an additional exchange term K), which dominates CPU times. The computational effort can be reduced if J is treated by the RI-J technique for non-hybrid functionals. This concerns actually only the first term in Eq. (35), which includes a Coulomb matrix $\mathbf{J}[\tilde{\mathbf{M}}]$

$$J_{\nu\mu}[\tilde{\mathbf{M}}] = \sum_{\kappa\lambda}(\nu\mu|\kappa\lambda)\tilde{M}_{\kappa\lambda} \qquad (37)$$

$$\tilde{\mathbf{M}} = \mathbf{CMC}^{\dagger} \qquad (38)$$

where $\mathbf{C}$ is the MO coefficient matrix from Eq. (6). With RI-J we get

$$J_{\nu\mu}[\tilde{\mathbf{M}}] \approx \tilde{J}_{\nu\mu}[\tilde{\mathbf{M}}] = \sum_{PQ\kappa\lambda}(\nu\mu|P)(P|Q)^{-1}(Q|\kappa\lambda)\tilde{M}_{\kappa\lambda} \qquad (39)$$

The replacement of $\mathbf{J}$ by $\tilde{\mathbf{J}}$ requires 'only' to import the RI-J machinery into the **AOFORCE** module.

Our implementation of RI-J for second analytical derivatives is described in a publication [39], which documents reliability and efficiency of RI-J for this purpose. CPU times for the evaluation of J are reduced to about 10%, which is not as spectacular as for energy or gradient calculations. The reason for this is simple: our CPKS solver treats a set of CPKS equations simultaneously and two-electron integrals (the most expensive part) are evaluated only once for the entire set. Total timings are typically reduced by a factor 2.5, the bottleneck is now in the treatment of E_{XC}, i.e. in the second term in Eq. (35), which is believed to be efficiently implemented.

5.2 Demonstrative tests

5.2.1 Indinavir

For this more realistic system ($C_{36}H_{47}N_5O_4$, f=270, C_1 symmetry) we carried out BP86 partial RI-J DFT second nuclear derivative calculations. On Intel Xeon (2.4 GHz) computers, we obtained the following timings:

SV(P), 769 basis functions	$J_{aux}[M^X]$: 3.2 h	tot. CPU$_{aux}$: 25.7 h
TZVP [40], 1182 basis functions	$J_{aux}[M^X]$: 8.8 h	tot. CPU$_{aux}$: 74.2 h

Indinavir has some floppy modes with frequencies below 10 cm $^{-1}$. This requires a careful structure optimization since otherwise the computed frequencies can be imaginary, i.e. $\approx i5$ cm^{-1}. We recommend to include derivatives of quadrature weights in the structure optimization to make sure the energy minimum has been accurately located and to avoid spurious imaginary frequencies.

5.2.2 Cyanocobalamin

As a last example we report timings for the computation of the second derivatives of cyanocobalamin (vitamin B12, $C_{63}H_{88}N_{14}O_{14}PCo$, f=537, C_1 symmetry). Using again an SV(P) basis, 1492 basis functions, and grid m4 the calculation took 18 days and 22 hours. In this case it was decided to use grid m4 (instead of the coarser grid m3), since for systems exceeding about 50 atoms we generally recommend to use a finer grid. The RI-J part required just 13% of the total time. Matrix algebra, e.g. Eq. (38) accounts only 3% of the CPU time.

In Figure 5 we present a comparison of the experimental solid state infrared absorption spectrum [41] with the one computed in this work. For this purpose we broadened the computed lines by 30 cm^{-1} and scaled globally the intensities to roughly match experiment reported in arbitrary units. The line at 2152 cm^{-1} corresponds to the CN stretch of the central CoCN group in good agreement with experiment. At 2731 cm^{-1} we find an intramolecular O$\cdot\cdot\cdot$H-O mode, around 3170 cm^{-1} are various NH stretches, which are all localized at the molecular surface. These modes are affected by intermolecular interactions causing frequency shifts and broadening, as shown by experiment. Even the peak

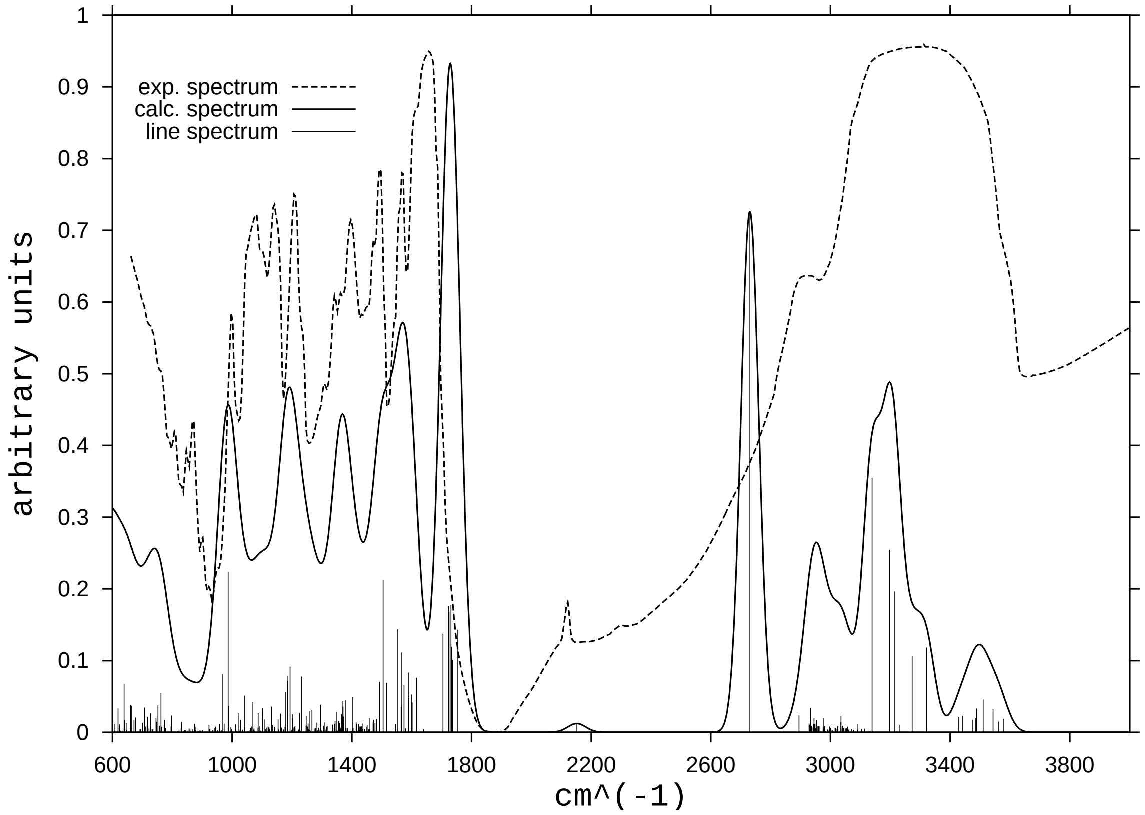


Figure 5: Comparison of the experimental solid state absorption infrared spectrum of cyanocobalamin (solid line) [41] with the one computed in this work (dashed line).

at 1754 cm^{-1}, again at the surface of B12, should be affected by packing effects. The detailed assignment of vibrations resulting from the theoretical treatment thus shows that information provided by solid state spectra is limited. This conclusion also holds for the IR spectra reported in polar solvents D_2O, ethanol and 75% glycerol [42]. There are three peaks denoted B, C and D between 1530 and 1680 cm^{-1}. In this range we find numerous modes including surface modes affected by solvation.

6 Summary

We have developed and implemented the MARI-J technique in the TURBOMOLE modules **RIDFT** and **RDGRAD**, which serve to optimize wavefunctions and to compute forces on the nuclei within DFT. This has considerably reduced the effort to deal with the interelectronic Coulomb repulsion J, which was the dominating step before. Since larger molecules - with more than 100 atoms - are mainly treated by DFT, and since most CPU time is typically spent in the modules **RIDFT** and **RDGRAD**, which are required in the iterative procedure to determine molecular structures, we have greatly increased the efficiency of TURBOMOLE. The gains are especially pronounced for larger systems with more than 300 atoms, and up to 1000 atoms are now feasible, see Figures 2-4.

The other important project concerned the extension of functionality. The module **AOFORCE** can now deal with second analytical derivatives within DFT. This was not implemented before. **AOFORCE** was completely redesigned, it is now efficient for closed and open shell states treated by HF and DFT. As demonstrated above, see e.g. Figure 5, one can now even compute IR- and RAMAN-frequencies for molecules with more than 100 atoms.

All timings reported in this work and the corresponding publications are conservative in the following sense. Shortly after the HPC-Chem project ended but in direct connection with it, the present authors have redesigned the integral routines computing $< \rho|P >$ in RI-J energy and gradient calculations, which are also called in all other modules employing the RI technique. This has increased efficiency, CPU times for the NF part of MARI-J are reduced by 30 % compared to the timings reported in Table 1.

Bibliography

[1] J. P. Perdew, *Phys. Rev. B* **33**, 8822 (1986).

[2] J. P. Perdew, *Phys. Rev. B* **34**, 7046 (1986).

[3] A. D. Becke, *Phys. Rev. A* **38**, 3098 (1988).

[4] J. Almlöf, K. Faegri, and K. Korsell, *J. Comput. Chem.* **3**, 385 (1982).

[5] M. Häser and R. Ahlrichs, *J. Comput. Chem.* **10**, 104 (1989).

[6] R. Ahlrichs, M. Bär, M. Häser, H. Horn, and C. Kölmel, *Chem. Phys. Lett.* **162**, 165 (1989).

[7] O. Treutler and R. Ahlrichs, *J. Chem. Phys.* **102**, 346 (1995).

[8] F. Haase and R. Ahlrichs, *J. Comp. Chem.* **14**, 907 (1993).

[9] F. Weigend and Häser, *Theor. Chem. Acc.* **97**, 331 (1997).

[10] C. Hättig and F. Weigend, *J. Chem. Phys.* **113**, 5154 (2000).

[11] F. Furche, *J. Chem. Phys.* **114**, 5982 (2001).

[12] R. Bauernschmitt and R. Ahlrichs, *J. Chem. Phys.* **104**, 9047 (1996).

[13] R. Bauernschmitt and R. Ahlrichs, *Chem. Phys. Lett.* **256**, 454 (1996).

[14] O. Christiansen, H. Koch, and P. Jørgensen, *Chem. Phys. Lett.* **243**, 409 (1995).

[15] K. Eichkorn, H. Treutler, O. Öhm, M. Häser, and R. Ahlrichs, *Chem. Phys. Lett.* **242**, 652 (1995).

[16] B. Dunlap, J. Conolly, and J. Sabin, *J. Chem. Phys.* **71**, 3396 (1979).

[17] J. Mintmire and B. Dunlap, *Phys. Rev. A* **25**, 88 (1982).

[18] O. Vahtras, J. Almlöf, and M. Feyereisen, *Chem. Phys. Lett.* **213**, 514 (1993).

[19] K. Eichkorn, F. Weigend, O. Treutler, and R. Ahlrichs, *Theor. Chim. Acta* **97**, 119 (1997).

[20] F. Weigend, *Phys. Chem. Chem. Phys.* **4**, 4285 (2002).

[21] C. Hättig and A. Köhn, *J. Chem. Phys.* **117**, 6939 (2002).

[22] F. Weigend, M. Häser, H. Patzelt, and R. Ahlrichs, *Chem. Phys. Letters* **294**, 143 (1998).

[23] F. Weigend, A. Köhn, and C. Hättig, *J. Chem. Phys.* **116**, 3175 (2002).

[24] P. Pulay, G. Fogarasi, F. Pang, and J. E. Boggs, *J. Am. Chem. Soc.* **101**, 2550 (1979).

[25] M. Sierka, A. Hogekamp, and R. Ahlrichs, *J. Chem. Phys.* **118**, 9136 (2003).

[26] A. D. Becke, *Phys. Rev. A* **38**, 3098 (1988).

[27] J. P. Perdew, *Phys. Rev. B* **33**, 8822 (1986).

[28] S. Vosko, L. Wilk, and M. Nussair, *Can. J. Phys.* **58**, 1200 (1980).

[29] A. Schäfer, H. Horn, and R. Ahlrichs, *J. Chem. Phys.* **97**, 2571 (1992).

[30] M. Strain, G. Scuseria, and M. Frisch, *Science* **271**, 51 (1996).

[31] J. Pérez-Jordá and W. Yang, *J. Chem. Phys.* **107**, 1218 (1997).

[32] C. Baerlocher, W. Meier, and D. Olson, *Atlas of Zeolite Framework Types*, Elsevier Science, Amsterdam, 2001.

[33] A. Wlodawer, H. Savage, and G. Dosdon, *Acta Crystallogr. B* **45**, 99 (1989).

[34] H. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. Bhat, H. Weissig, I. Shindyalov, and P. Bourne, *Nucleic Acids Res.* **28**, 235 (2000).

[35] C. Fonseca-Guerra, J. Snijders, G. Te Velde, and E. Baerends, *Theor. Chim. Acta* **99**, 391 (1998).

[36] A. St-Amant and R. Gallant, *Chem. Phys. Lett.* **256**, 569 (1996).

[37] M. Sierka, *in preparation* .

[38] P. Deglmann, F. Furche, and R. Ahlrichs, *Chem. Phys. Lett.* **362**, 511 (2002).

[39] P. Deglmann, K. May, F. Furche, and R. Ahlrichs, *Chem. Phys. Lett.* **384**, 103 (2004).

[40] A. Schäfer, C. Huber, and R. Ahlrichs, *J. Chem. Phys.* **100**, 5829 (1994).

[41] Coblentz Society, Inc., 'Evaluated Infrared Reference Spectra' in NIST Chemistry WebBook, NIST Standard Reference Database Number 69, Eds. P.J. Linstrom and W.G. Mallard, March 2003, National Institute of Standards and Technology, Gaithersburg MD, 20899 (`http://webbook.nist.gov`).

[42] K. Taraszka, C. Eefei, T. Metzger, and M. Chance, *Biochemistry* **30**, 1222 (1991).

QUICKSTEP: Make the Atoms Dance

Matthias Krack and Michele Parrinello

Computational Science
Department of Chemistry and Applied Biosciences
ETH Zürich
USI-Campus, via Giuseppe Buffi 13
6900 Lugano, Switzerland
E-mail: krack@phys.chem.ethz.ch

1 Introduction

During the last decade density functional theory (DFT) has become a very efficient tool for electronic structure calculations. DFT methods were successfully applied to many problems in different fields ranging from material science to chemistry and bio-chemistry. Most of the applied methods use either plane waves or Gaussian-type functions for the expansion of the Kohn-Sham orbitals. Both types of basis functions have their own merits. Plane waves provide from the beginning an orthogonal basis set and are independent of the atomic positions which makes the force calculation very simple. Moreover, the calculation of the Hartree (Coulomb) potential can be efficiently performed by fast Fourier transformation (FFT). Unfortunately, there are also some disadvantages. The strong variations of the wave function close to the nuclei require a huge number of plane waves. Atomic pseudo potentials are usually employed to alleviate this problem, but for many applications the number of plane waves is still large. Furthermore, the whole space is equally filled with plane waves and therefore each point in space is described with the same accuracy, but this feature of the plane waves turns out to be rather inefficient for systems of low density like biological systems where the homogeneous description of empty and atom-filled regions results in a waste of basis functions and thus computer time. By contrast, Gaussian-type functions localized at the atomic centers are much more efficient in this respect, since they provide a more compact description of the atomic charge densities and basically there is no need to employ atomic pseudo potentials. Nevertheless, Gaussian-type functions have also some disadvantages. The force calculation is more complicated, the Hartree term usually requires the computation of a large number of multi-center integrals, and possibly basis set superposition errors have to be considered. The Gaussian plane waves (GPW) method [1] tries to combine the merits of Gaussian-type functions and plane waves. In that way it becomes feasible to build the Kohn-Sham operator matrix with a computational cost scaling linearly for a growing system size.

A new implementation of the GPW method, called QUICKSTEP, was performed within the framework of the HPC-Chem project [2]. The goal of the project was to implement the GPW method in a fully modular and efficiently parallelized manner. QUICKSTEP is part of the open source project CP2K [3, 4] which ensures the continuation of the development even after the end of the HPC-Chem project. The next section will provide a short outline of the GPW method followed by a description of the pseudo potentials and the Gaussian basis sets employed by QUICKSTEP. Finally, the accuracy and the efficiency of the new parallelized QUICKSTEP implementation will be shown.

2 Gaussian and plane waves method

The energy functional for a molecular or crystalline system in the framework of the Gaussian plane waves (GPW) method [1] using the Kohn-Sham formulation of density functional theory (DFT) [5, 6] is defined as

$$\begin{aligned}
E[n] =& E^{\mathrm{T}}[n] + E^{\mathrm{V}}[n] + E^{\mathrm{H}}[n] + E^{\mathrm{XC}}[n] + E^{\mathrm{II}} \qquad (1)\\
=& \sum_{\mu\nu} P_{\mu\nu}\, \langle \varphi_\mu(\boldsymbol{r})| -\frac{1}{2}\nabla^2 |\varphi_\nu(\boldsymbol{r})\rangle \\
&+ \sum_{\mu\nu} P_{\mu\nu}\, \langle \varphi_\mu(\boldsymbol{r})| V_{\mathrm{loc}}^{\mathrm{PP}}(r) |\varphi_\nu(\boldsymbol{r})\rangle \\
&+ \sum_{\mu\nu} P_{\mu\nu}\, \langle \varphi_\mu(\boldsymbol{r})| V_{\mathrm{nl}}^{\mathrm{PP}}(\boldsymbol{r},\boldsymbol{r}') |\varphi_\nu(\boldsymbol{r}')\rangle \\
&+ 2\pi\,\Omega \sum_{\boldsymbol{G}} \frac{\tilde{n}^*(\boldsymbol{G})\,\tilde{n}(\boldsymbol{G})}{\boldsymbol{G}^2} \\
&+ \int \bar{n}(\boldsymbol{r})\,\varepsilon_{\mathrm{XC}}[\bar{n}]\,d\boldsymbol{r} \\
&+ \frac{1}{2}\sum_{I\neq J} \frac{Z_I Z_J}{|\boldsymbol{R}_I - \boldsymbol{R}_J|}
\end{aligned}$$

where $E^{\mathrm{T}}[n]$ is the kinetic energy, $E^{\mathrm{V}}[n]$ is the electronic interaction with the ionic cores, $E^{\mathrm{H}}[n]$ is the electronic Hartree (Coulomb) energy and $E^{\mathrm{XC}}[n]$ is the exchange–correlation energy. The interaction energies of the ionic cores with charges Z_I and positions $\boldsymbol{R}_I$ is denoted by E^{II}. The electronic interaction with the ionic cores is described by norm-conserving pseudo potentials with a potential split in a local part $V_{\mathrm{loc}}^{\mathrm{PP}}(r)$ and a fully non-local part $V_{\mathrm{nl}}^{\mathrm{PP}}(\boldsymbol{r}, \boldsymbol{r}')$ (see section 3).

The electronic density

$$n(\boldsymbol{r}) = \sum_{\mu\nu} P_{\mu\nu}\varphi_\mu(\boldsymbol{r})\varphi_\nu(\boldsymbol{r}) \tag{2}$$

is expanded in a set of contracted Gaussian functions

$$\varphi_\mu(\boldsymbol{r}) = \sum_i d_{i\mu} g_i(\boldsymbol{r}) \tag{3}$$

where $P_{\mu\nu}$ is a density matrix element, $g_i(\boldsymbol{r})$ is a primitive Gaussian function, and $d_{i\mu}$ is the corresponding contraction coefficient. The density matrix $\boldsymbol{P}$ fulfills normalization and idempotency conditions

$$\mathrm{Tr}(\boldsymbol{PS}) = N \tag{4}$$

$$\boldsymbol{PS} = (\boldsymbol{PS})(\boldsymbol{PS}) \tag{5}$$

where $\boldsymbol{S}$ is the overlap matrix of the Gaussian basis functions

$$S_{\mu\nu} = \langle\varphi_\mu(\boldsymbol{r})||\varphi_\nu(\boldsymbol{r})\rangle \tag{6}$$

and N is the number of electrons.

In the original work by Lippert et al. [1] the same auxiliary basis approximation was used for the Hartree and exchange-correlation energy. It was useful to relax this constraint and use two independent approximations to the density, denoted $\tilde{n}(\boldsymbol{G})$ for the Hartree energy and $\bar{n}(\boldsymbol{r})$ for the exchange-correlation energy. Both approximate electronic charge densities are functions of the density matrix $\boldsymbol{P}$.

3 Pseudo potentials

The GPW method works like plane waves methods with atomic pseudo potentials, since an expansion of Gaussian functions with large exponents is numerically not efficient or even not feasible.

The current implementation of the GPW method uses only the pseudo potentials of Goedecker, Teter, and Hutter (GTH) [7, 8]. The separable dual-space GTH pseudo potentials consist of a local part

$$V_{\text{loc}}^{\text{PP}}(r) = -\frac{Z_{\text{ion}}}{r}\,\text{erf}\left(\alpha^{\text{PP}} r\right) + \sum_{i=1}^{4} C_i^{\text{PP}} \left(\sqrt{2}\alpha^{\text{PP}} r\right)^{2i-2} \exp\left[-\left(\alpha^{\text{PP}} r\right)^2\right] \tag{7}$$

with

$$\alpha^{\text{PP}} = \frac{1}{\sqrt{2}r_{\text{loc}}^{\text{PP}}}$$

and a non-local part

$$V_{\text{nl}}^{\text{PP}}(\boldsymbol{r},\boldsymbol{r}') = \sum_{lm}\sum_{ij}\langle\, \boldsymbol{r} \mid p_i^{lm}\,\rangle\, h_{ij}^{l}\, \langle\, p_j^{lm} \mid \boldsymbol{r}'\,\rangle \tag{8}$$

with the Gaussian-type projectors

$$\langle \boldsymbol{r} \mid p_i^{lm}\rangle = N_i^l Y^{lm}(\hat{r}) r^{l+2i-2} \exp\left[-\frac{1}{2}\left(\frac{r}{r_l}\right)^2\right]$$

as shown in Eq. 1 resulting in a fully analytical formulation which requires only the definition of a small parameter set for each element. Moreover, the GTH pseudo potentials are transferable and norm-conserving. Nevertheless, plane waves methods employ this pseudo potential type only for reference calculations or if no other reliable pseudo potentials are available, since this type requires relative high cut-off values, i.e. more plane waves. However, in the framework of the GPW method there are no such limitations, since all contributions are integrals over Gaussian functions which can be calculated analytically. Therefore the GTH pseudo potentials are particularly suited for the use with QUICKSTEP and that is why QUICKSTEP only supports GTH pseudo potentials, currently. The GTH pseudo potential parameters were optimized with respect to atomic all-electron wavefunctions obtained from fully relativistic density functional calculations using a numerical atom code. The optimized pseudo potentials include all scalar relativistic corrections via an averaged potential [8], because the consideration of relativistic effects is indispensable for applications involving heavier elements. A database with many GTH pseudo potential parameter sets optimized for different exchange-correlation potentials is already available [3]. It provides all parameter sets formatted for a direct usage with QUICKSTEP and it contains parameter sets for almost the whole periodic table based on the local density approximation (LDA). Moreover, there are also many optimized parameter sets for the exchange-correlation potentials based on the generalized gradient approximation (GGA) of Becke, Lee, Yang, and Parr (BLYP) [9, 10, 11], Becke and Perdew (BP) [9, 12], Hamprecht, Cohen, Tozer and Handy (HCTH/120, HCTH/407) [13] and Perdew, Burke and Ernzerhof (PBE) [14]. The following GTH pseudo potentials are currently available:

LDA:
H(1), He(2), Li(1), Li(3), Be(2), Be(4), B(3), C(4), N(5), O(6), F(7), Ne(8), Na(1), Na(9), Mg(10), Mg(2), Al(3), Si(4), P(5), S(6), Cl(7), Ar(8), K(1), K(9), Ca(10), Ca(2), Sc(11), Sc(3), Ti(12), Ti(4), V(13), V(5), Cr(14), Cr(6), Mn(15), Mn(7), Fe(16), Fe(8), Co(17), Co(9), Ni(10), Ni(18), Cu(1), Cu(11), Zn(12), Zn(2), Zn(20) Ga(13), Ga(3), Ge(4), As(5), Se(6), Br(7), Kr(8), Rb(1), Rb(9), Sr(10), Sr(2), Y(11), Y(3), Zr(12), Zr(4), Nb(13), Nb(5), Mo(14), Mo(6), Tc(15), Tc(7), Ru(16), Ru(8), Rh(17), Rh(9), Pd(10), Pd(18), Ag(1), Ag(11), Cd(12), Cd(2), In(13), In(3), Sn(4), Sb(5), Te(6), I(7), Xe(8), Cs(1), Cs(9), Ba(10), Ba(2), La(11), Ce(12), Pr(13), Nd(14), Pm(15), Sm(16), Eu(17), Gd(18), Tb(19), Dy(20), Ho(21), Er(22), Tm(23), Yb(24), Lu(25), Hf(12), Ta(13), Ta(5), W(14), W(6), Re(15), Re(7), Os(16), Os(8), Ir(17), Ir(9), Pt(10), Pt(18), Au(1), Au(11), Hg(12), Hg(2), Tl(13), Tl(3) Pb(4), Bi(5), Po(6), At(7), Rn(8)

BLYP:
H(1), He(2), Li(3), Be(4), B(3), C(4), N(5), O(6), F(7), Ne(8), Na(9), Mg(10), Al(3) Si(4), P(5), S(6), Cl(7), Ar(8), Ca(10), Ti(12), V(13), Cr(14), Mn(15), Fe(16), Co(17), Ni(18), Cu(11), Zn(12), Ge(4), Br(7), Zr(12), I(7), Ba(10), Ba(2), W(14)

BP:
H(1), He(2), Li(3), Be(4), B(3), C(4), N(5), O(6), F(7), Ne(8), Na(1), Na(9), Mg(10), Al(3), Si(4), P(5), S(6), Cl(7), Ar(8), Ca(10), Sc(11), Ti(12), V(13), Cr(14) Mn(15), Fe(16), Co(17), Ni(18), Cu(11), Zn(12), Zr(12), Cs(1), Cs(9)

HCTH/120:
H(1), O(6), Ar(8)

HCTH/407:
H(1), O(6)

PBE:
H(1), He(2), Li(3), Be(4), B(3), C(4), N(5), O(6), F(7), Ne(8), Na(9), Mg(10), Mg(2), Al(3), Si(4), P(5), S(6), Cl(7), Ar(8), Ca(10), Ti(12), Zr(12)

The numbers in brackets denote the number of valence electrons employed by the respective pseudo potential, i.e. the effective core charge. The pseudo potential data base is maintained within the CP2K project [3] and thus all the listed GTH pseudo potential data sets are available online.

4 Basis sets

The Kohn-Sham orbitals are expanded in Gaussian orbital functions in the framework of the GPW method as described in section 2. Therefore an appropriate set of Gaussian functions has to be defined as a basis set for each QUICKSTEP calculation. There is a plenty of Gaussian basis sets available in the literature. However, proper basis sets have to be optimized for the usage with the GTH pseudo potentials from the previous section. Therefore, the exponents of a set of primitive Gaussian functions were optimized for all first- and second-row elements with an atomic DFT code applying the appropriate GTH potential parameters for each element. The same set of exponents was employed for each angular momentum quantum number of the occupied valence states of the actual element which are only s and p orbitals for the elements from H to Ar. The optimization was performed for a growing numbers of primitive Gaussian functions in the set in order to obtain basis sets of increasing quality. The atomic DFT code allows for the calculation of first analytic derivatives of the total atomic energy with respect to the Gaussian orbital exponents. The second derivatives were calculated by an updated Hessian procedure (BFGS). Finally, the primitive Gaussian functions were contracted using the coefficients of the respective atomic wavefunctions. These basis sets were augmented by polarization functions which were taken from the all-electron basis sets cc-pVXZ (X = D, T, Q) of Dunning [15, 16]. In that way a new sequence of basis sets was created with an increasing number of primitive Gaussian functions and polarization functions for each first- and second-row element. The basis sets were labelled DZVP, TZVP, TZV2P, QZV2P, and QZV3P due to the applied splitting of the valence basis where DZ, TZ, and QZ denote double-, triple- , and quadruple-zeta, respectively, and the increasing number of polarization functions. The quality of the basis sets should improve systematically within this sequence. These basis sets can be further augmented by diffuse functions, if required, analogous to the aug-cc-pVXZ basis sets resulting in a sequence aug-DZVP, aug-TZVP, aug-TZV2P, aug-QZV2P, and aug-QZV3P analogous to the aug-cc-pVXZ basis sets. The inclusion of diffuse functions may improve the accuracy of certain molecular properties, however, they are prohibitive for condensed phase calculations, since they introduce linear dependencies into the basis set. The basis sets for H to Ar are collected in a basis set file which is included into the CP2K program package.

5 Wavefunction optimization

The total ground state energy (see Eq. 1) of a system for a given atomic configuration is minimized by an iterative self-consistent field (SCF) procedure. Three methods are currently available in QUICKSTEP to perform an SCF iteration procedure: a traditional diagonalization (TD) scheme, a pseudo diagonalization scheme [17], and an orbital transformation (OT) method [19]. For the sake of simplicity, we will restrict our description of these methods in the following to closed-shell systems, however, the generalization to open-shell (spin-polarized) systems is straightforward and QUICKSTEP can deal with both types of systems using each of these methods.

5.1 Traditional diagonalization (TD)

The traditional diagonalization scheme uses an eigensolver from a standard parallel program library called ScaLAPACK to solve the general eigenvalue problem

$$\boldsymbol{K}\,\boldsymbol{c} = \boldsymbol{S}\,\boldsymbol{c}\,\epsilon \tag{9}$$

where $\boldsymbol{K}$ is the Kohn-Sham matrix and $\boldsymbol{S}$ is the overlap matrix of the system. The eigenvectors $\boldsymbol{c}$ represent the orbital coefficients, and the ϵ are the corresponding eigenvalues. Unfortunately, the overlap matrix $\boldsymbol{S}$ is not the unit matrix, since QUICKSTEP employs an atom-centered basis set of non-orthogonal Gaussian-type orbital functions. Thus we have to transform the eigenvalue problem to its special form

$$\boldsymbol{K}\,\mathrm{c} = \boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}\,\boldsymbol{c}\,\epsilon \tag{10}$$

$$(\boldsymbol{U}^{\mathrm{T}})^{-1}\,\boldsymbol{K}\,\boldsymbol{U}^{-1}\,\boldsymbol{c}' = \boldsymbol{c}'\,\epsilon \qquad (\texttt{pdsygst}) \tag{11}$$

$$\boldsymbol{K}'\,\boldsymbol{c}' = \boldsymbol{c}'\,\epsilon \qquad (\texttt{pdsyevx}\ \text{or}\ \texttt{pdsyevd}) \tag{12}$$

using a Cholesky decomposition of the overlap matrix

$$\boldsymbol{S} = \boldsymbol{U}^{\mathrm{T}}\boldsymbol{U} \qquad (\texttt{pdpotrf}) \tag{13}$$

as the default method for that purpose. Now, Eq. 12 can simply be solved by a diagonalization of $\mathbf{K}'$. The orbital coefficients $\boldsymbol{c}$ in the non-orthogonal basis are finally obtained by the back-transformation

$$\boldsymbol{c}' = \boldsymbol{U}\,\boldsymbol{c} \qquad (\texttt{pdtrsm}). \tag{14}$$

The names in brackets denote the ScaLAPACK routines employed for the respective operation by QUICKSTEP.

Alternatively, a symmetric orthogonalization instead of a Cholesky decomposition can be applied by using

$$\boldsymbol{U} = \boldsymbol{S}^{1/2}. \tag{15}$$

However, the calculation of $\boldsymbol{S}^{1/2}$ involves a diagonalization of $\boldsymbol{S}$ which is computationally more expensive than a Cholesky decomposition. On the other hand, linear dependencies in the basis set introduced by small Gaussian function exponents can be detected when $\boldsymbol{S}$ is diagonalized. Eigenvalues of $\boldsymbol{S}$ smaller than 10^{-5} usually indicate significant linear dependencies in the basis set and a filtering of the corresponding eigenvectors might help to ameliorate numerical difficulties during the SCF iteration procedure. Both orthogonalization schemes are implemented in QUICKSTEP. For small systems the choice of the orthogonalization has no crucial impact on the performance, since it has to be performed only once for each configuration during the initialization of the SCF run. By contrast, the eigenvectors and eigenvalues of the full Kohn-Sham matrix $\boldsymbol{K}'$ have to be calculated in each iteration step as indicated by Eq. 12 using a divide-and-conquer (`pdsyevd`) scheme or an expert driver (`pdsyevx`) which allows to request only the build of an eigenvector sub-set. The divide-and-conquer scheme is faster than the expert driver, if all eigenvectors have to be computed. However, for the construction of the new density matrix

$$\boldsymbol{P} = 2\,\boldsymbol{c}_{\mathrm{occ}}\boldsymbol{c}_{\mathrm{occ}}^{\mathrm{T}} \tag{16}$$

only the occupied orbitals are needed. In that case the expert driver is superior, since for standard basis sets only 10–20% of the orbitals are occupied and the orthonormalization of the requested eigenvectors is a time-consuming step, especially on parallel computers where it requires heavy communication between the processes.

The TD scheme is usually combined with methods to improve the convergence of the SCF iteration procedure. The most efficient SCF convergence acceleration is achieved by the direct inversion in the iterative sub-space (DIIS) [17, 20] exploiting the commutator relation

$$\boldsymbol{e} = \boldsymbol{K}\,\boldsymbol{P}\,\boldsymbol{S} - \boldsymbol{S}\,\boldsymbol{P}\,\boldsymbol{K} \tag{17}$$

between the Kohn-Sham and the density matrix where the error matrix $\boldsymbol{e}$ is zero for the converged density. The TD/DIIS scheme is an established method for electronic structure calculations. The DIIS method can be very efficient in the number of iterations required to reach convergence starting from a sufficiently pre-converged density which is significant, if the Kohn-Sham matrix construction is much more time consuming than the diagonalization. Nevertheless, the cost for the TD/DIIS scales as $O(M^3)$, where M is the size of the basis set. This implies that, even at fixed system size, increasing the number of basis functions results in a cubic growth of the computational cost. A further disadvantage of the DIIS is that the method might fail to converge or that a sufficiently pre-converged density cannot be obtained. This happens more frequently for electronically difficult systems. For instance spin-polarized systems or systems with a small energy gap between the highest occupied (HOMO) and the lowest unoccupied orbital (LUMO) like semiconductors or metals belong often to this kind of systems.

5.2 Pseudo diagonalization (PD)

Alternatively to the TD scheme, a pseudo diagonalization [17, 18] can be applied as soon as a sufficiently pre-converged wavefunction is obtained. The Kohn-Sham matrix $\boldsymbol{K}^{\mathrm{AO}}$ in the atomic orbital (AO) basis is transformed into the molecular orbital (MO) basis in each

SCF step

$$\boldsymbol{K}^{\mathrm{MO}} = \boldsymbol{c}^{\mathrm{T}}\boldsymbol{K}^{\mathrm{AO}}\boldsymbol{c} \qquad \text{(PDSYMM and PDGEMM)} \tag{18}$$

using the MO coefficients $\boldsymbol{c}$ from the preceding SCF step. The converged $\boldsymbol{K}^{\mathrm{MO}}$ matrix using TD is a diagonal matrix and the eigenvalues are its diagonal elements. Already after a few SCF iteration steps the $\boldsymbol{K}^{\mathrm{MO}}$ matrix becomes diagonal dominant. Moreover, the $\boldsymbol{K}^{\mathrm{MO}}$ matrix shows the following natural blocking

$$\left(\begin{array}{c|c} oo & ou \\ \hline uo & uu \end{array}\right) \tag{19}$$

due to the two MO sub-sets of $\boldsymbol{c}$ namely the occupied (o) and the unoccupied (u) MOs. The eigenvectors are used during the SCF iteration to calculate the new density matrix (see Eq. 16), whereas the eigenvalues are not needed. The total energy only depends on the occupied MOs and thus a block diagonalization which decouples the occupied and unoccupied MOs allows to converge the wavefunctions, i.e. only all elements of the block ou or uo have to become zero, since $\boldsymbol{K}^{\mathrm{MO}}$ is a symmetric matrix. Hence the transformation into the MO basis

$$\boldsymbol{K}^{\mathrm{MO}}_{ou} = \boldsymbol{c}^{\mathrm{T}}_{o}\boldsymbol{K}^{\mathrm{AO}}\boldsymbol{c}_{u} \qquad \text{(PDSYMM and PDGEMM)} \tag{20}$$

has only to be performed for that matrix block. Then the decoupling can be achieved iteratively by consecutive 2×2 Jacobi rotations

$$\theta = \frac{\epsilon_q - \epsilon_p}{2\,K^{\mathrm{MO}}_{pq}} \tag{21}$$

$$t = \frac{\mathrm{sgn}(\theta)}{|\theta| + \sqrt{1+\theta^2}} \tag{22}$$

$$c = \frac{1}{\sqrt{t^2+1}} \tag{23}$$

$$s = tc \tag{24}$$

$$\tilde{\boldsymbol{C}}_p = c\,\boldsymbol{C}_p - s\,\boldsymbol{C}_q \qquad \text{(DSCAL and DAXPY)} \tag{25}$$

$$\tilde{\boldsymbol{C}}_q = s\,\boldsymbol{C}_p + c\,\boldsymbol{C}_q \qquad \text{(DSCAL and DAXPY)} \tag{26}$$

where the angle of rotation θ is determined by the difference of the eigenvalues of the MOs p and q divided by the corresponding matrix element K^{MO}_{pq} in the ou or uo block. The

Jacobi rotations can be performed with the BLAS level 1 routines DSCAL and DAXPY. The *oo* block is significantly smaller than the uu block, since only 10–20% of the MOs are occupied using a standard basis set. Consequently, the ou or uo block also includes only 10–20% of the $\boldsymbol{K}^{\mathrm{MO}}$ matrix. Furthermore, an expensive re-orthonormalization of the MO eigenvectors $\boldsymbol{c}$ is not needed, since the Jacobi rotations preserve their orthonormality.

5.3 Orbital transformations (OT)

Finally, an orbital transformation method [19] is implemented in QUICKSTEP which performs a direct minimization of the wavefunctions. The OT method is guaranteed to converge and it scales, depending on the preconditioner, as $O(MN^2)$, where M is the total number of MOs or basis functions and N is the number of occupied MOs. A detailed description of the OT method is given in Ref. [19]. In the framework of the OT method the electronic energy $E(\boldsymbol{c})$ is minimized using the constraint

$$\boldsymbol{c}^{\mathrm{T}}\boldsymbol{S}\,\boldsymbol{c} = \boldsymbol{I} \tag{27}$$

where $\boldsymbol{c}$, $\boldsymbol{S}$, and $\boldsymbol{I}$ are the matrix of the orbital coefficients, the overlap matrix, and the identity matrix, respectively. Given the constant start vectors $\boldsymbol{c}_0$ which fulfill the condition

$$\boldsymbol{c}_0^{\mathrm{T}}\boldsymbol{S}\,\boldsymbol{c}_0 = \boldsymbol{I} \tag{28}$$

a new set of vectors $\boldsymbol{c}(\boldsymbol{x})$ is obtained by

$$\boldsymbol{c}(\boldsymbol{x}) = \boldsymbol{c}_0\cos(\boldsymbol{U}) + \boldsymbol{x}\,\boldsymbol{U}^{-1}\sin(\boldsymbol{U}) \tag{29}$$

with

$$\boldsymbol{U} = (\boldsymbol{x}^{\mathrm{T}}\boldsymbol{S}\,\boldsymbol{x})^{1/2} \quad \text{and} \quad \boldsymbol{x}^{\mathrm{T}}\boldsymbol{S}\,\mathbf{c}_0 = \mathbf{0} \tag{30}$$

This implies

$$\boldsymbol{c}^{\mathrm{T}}(\boldsymbol{x})\,\boldsymbol{S}\,\boldsymbol{c}(\boldsymbol{x}) = \boldsymbol{I} \quad \forall\ \boldsymbol{x} \tag{31}$$

$\boldsymbol{x}$ can be used to optimize the energy $E(\boldsymbol{c}(\boldsymbol{x}))$ with standard methods like conjugate gradient in combination with line search, since the allowed $\boldsymbol{x}$ span a linear space. In that way, the OT method as a direct minimization method addresses both deficiencies of the TD or PD scheme, as the method is guaranteed to converge, and scales, depending on the preconditioner, as $O(MN^2)$. In more detail, the following scalings can be observed for the OT method:

- matrix product sparse-full like $\boldsymbol{S}\,\boldsymbol{X}$: $O(M^2N) \rightarrow O(MN)$
- matrix products full-full like $(\boldsymbol{K}\,\boldsymbol{C})^{\mathrm{T}}\boldsymbol{X}$: $O(MN^2)$
- diagonalization of the $N \times N$ matrix $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{S}\,\boldsymbol{X}$: $< O(N^3)$

The computational cost of the OT method is normally dominated by the computation of the $O(MN)$ terms $\boldsymbol{Hc}$ and $\boldsymbol{Sx}$, but is in principle $O(MN^2)$ with a sparse preconditioner, and $O(M^2N)$, if a non-sparse preconditioner is used. The relative efficiency of TD/DIIS and OT depends on many factors such as system size, basis set size, and network latency and bandwidth.

6 Accuracy

As a first accuracy test for QUICKSTEP, we employed the new basis sets described in section 4 for the geometry optimization of small molecules using the local density approximation (LDA). The CP2K geometry optimizer works with first analytic derivatives whereas the second derivatives are obtained via an updated Hessian method. In that way each molecule of the following test set of 39 small molecules:

> H_2, Li_2, LiH, BH_3, CH_4, C_2H_2, C_2H_4, C_2H_6, N_2, NH_3, HCN, H_2O, H_2O_2, CO,
> CO_2, CH_3OH, N_2O, F_2, HF, LiF, CH_3F, OF_2, AlH, SiH_4, SiO, P_2, PH_3, HCP,
> PN, S_3, H_2S, CS, CS_2, SO_2, COS, SF_6, HCl, CH_3Cl, LiCl

consisting of first- and second-row elements was optimized using Cartesian coordinates. Figure 1 compares the optimized bond distances obtained with QUICKSTEP using different basis sets with the NUMOL results of Dickson and Becke [21]. NUMOL is a purely numerical DFT code and thus considered to be free of basis set effects. The smallest basis set DZVP gives on average slightly too long bond distances, but already the TZVP basis set works fine for most of the molecules. Finally, the TZV2P, QZV2P, and QZV3P show an excellent agreement for all bond distances. Figure 2 shows the results for the optimized bond and dihedral angles. The agreement for the small DZVP and the TZVP basis set is already excellent. Only one data point is off which corresponds to the dihedral angle of H_2O_2. This angle is known to be very sensitive to the number of employed polarization functions. Thus one set of polarization functions is insufficient as shown by the results for the DZVP and TZVP basis set. However, for the TZV2P basis set the dihedral angle is already very close to the reference value and for the QZV3P basis set shows more or less a converged result. A comprehensive view of the numerical results of the geometry optimizations is provided by Table 1 which shows the maximum and the root mean square deviation of all bond distances and angle compared to the NUMOL results based on a statistics including 52 bond distances and 18 angles and dihedral angles. The errors become smaller for a growing basis set size as expected. The TZV2P basis set gives already an excellent overall agreement and for the QZV3P most distances coincide within the expected errors. Note, that a fully agreement with the NUMOL values is not possible, since NUMOL uses a slightly different LDA implementation and it employs a frozen core approximation for the elements beyond Beryllium that differs from the GTH pseudo potentials used by QUICKSTEP. These difference may cause a change of the bond distances of about 0.001 Å. This small error also shows that the effect of the pseudo potential is negligible compared to basis

Table 1: Maximum ($\Delta_{\max}$) and root mean square deviation (σ) of bond distances (Å), bond angles, and dihedral angles (°) compared to the NUMOL results for different basis sets.

basis set	distances [Å]		angles [°]	
	$\Delta_{\max}$	σ	$\Delta_{\max}$	σ
DZVP	0.048	0.018	6.4	1.6
TZVP	0.040	0.013	8.5	2.1
TZV2P	0.015	0.006	1.7	0.6
QZV2P	0.012	0.005	2.1	0.6
QZV3P	0.011	0.004	0.7	0.3

set effects concerning structural properties. Thus a basis set can be chosen tuned due to the accuracy requirements of the actual application, but finally the accuracy of QUICKSTEP is determined by the error of the employed exchange-correlation potential.

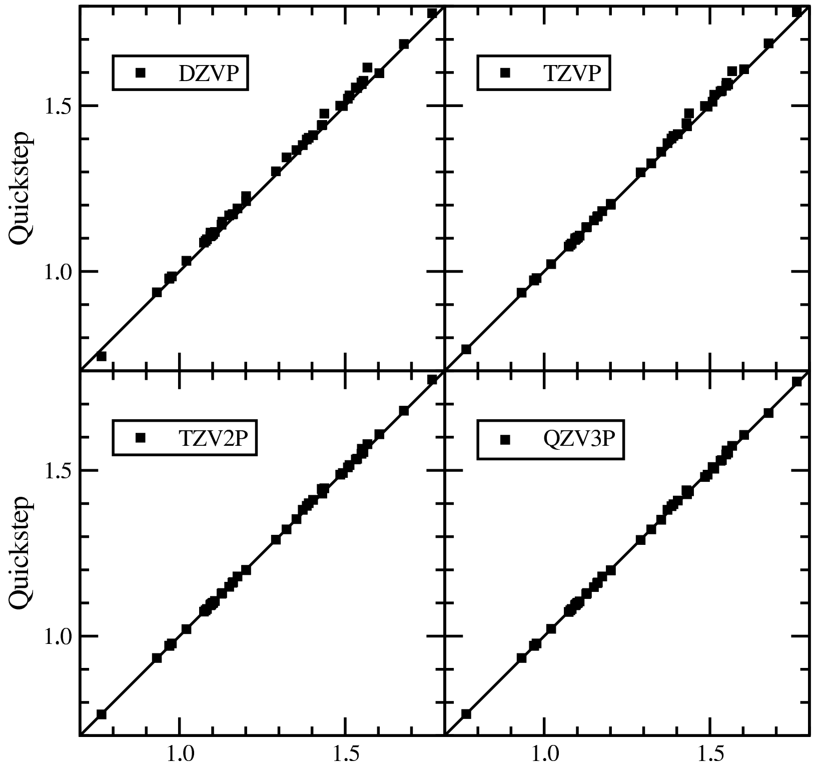


Figure 1: The optimized bond distances for 39 small molecules calculated with QUICKSTEP using different basis sets are compared to the NUMOL results of Dickson and Becke [21].

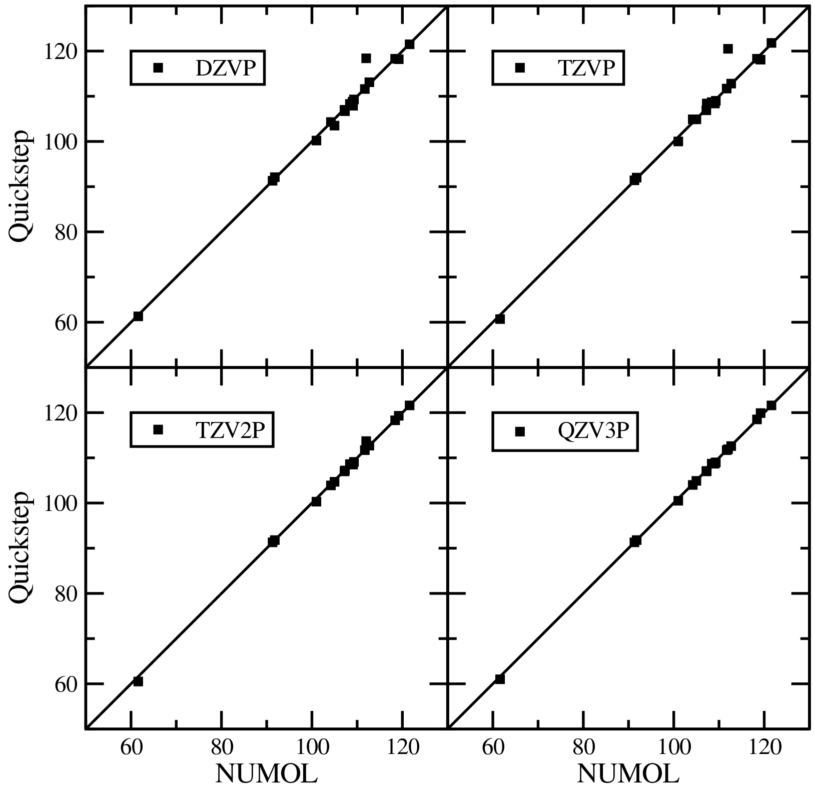


Figure 2: The optimized bond angles and dihedral angles for 39 small molecules calculated with QUICKSTEP using different basis sets are compared to the NUMOL results of Dickson and Becke [21].

7 Benchmarks

After proving the accuracy of QUICKSTEP in the previous section, it will be shown in this section that QUICKSTEP can achieve that accuracy with high computational efficiency. For that purpose, we firstly selected liquid water at ambient conditions as a benchmark system to show both the serial performance of QUICKSTEP and its scalability on a parallel computer. Moreover, we will report the performance results of geometry optimizations for some molecular and a crystalline system.

7.1 Liquid water

Liquid water is often used as a benchmark system, since it can easily be scaled by simply doubling the number of water molecules in the unit cell which is equivalent to a doubling of the unit cell at the same time. For instance, liquid water is employed as a standard benchmark system for the CPMD code [22] to check its performance and scalability on various parallel computers. Furthermore, water is an important ingredient of many bio-chemical applications involving water as the natural solvent and molecular dynamics (MD) simulations are performed to study the properties and the behavior of such systems. Therefore,

Table 2: Detailed characteristics of the employed benchmark systems for liquid water at ambient conditions (300 K, 1 bar). The edge length of the cubic simulation cell, the number of atoms, electrons, Gaussian-type orbitals (M), occupied orbitals (N), and plane waves, i.e. grid points, is listed.

system	cell [Å]	atoms	electrons	M	N	grid points ($\times 10^6$)
32 H_2O	9.9	96	256	1280	128	1.3
64 H_2O	12.4	192	512	2560	256	2.0
128 H_2O	15.6	384	1024	5120	512	4.1
256 H_2O	19.7	768	2048	10240	1024	9.3
512 H_2O	24.9	1536	4096	20480	2048	16.0
1024 H_2O	31.3	3072	8192	40960	4096	32.8

MD runs for pure liquid water at ambient conditions (300 K, 1 bar) were conducted for benchmarking using realistic input parameters as they would also be chosen for production runs. A GTH pseudo potential and a TZV2P basis set for hydrogen and oxygen were employed in all benchmark runs including 40 contracted spherical Gaussian-type orbital functions per water molecule. The high accuracy of the TZV2P basis set was already shown in section 6. Table 2 lists the detailed characteristics of the employed benchmark systems ranging from 32 water molecules in a cubic unit cell of edge length 9.9 Å up to 1024 water molecules in a cubic unit cell of 31.3 Å edge length. These unit cell sizes required up to $32.8 \cdot 10^6$ plane waves, i.e. grid points, as an auxiliary basis set given a density cut-off of 280 Ry for the expansion of the electronic density. This density cut-off was used for all the benchmark calculations of liquid water. Equally, the orbital basis set is linearly growing from 1280 to 40960 Gaussian-type orbital functions. However, the involved matrices like the overlap or Kohn-Sham matrix are growing quadratically for this entity. Thus the Kohn-Sham matrix calculation for 1024 H_2O requires to deal with matrices of the size 40960×40960 and it is therefore indispensable to take advantage of the localized character of the atomic interactions as efficient as possible. Table 3 shows the occupation of the overlap matrix for each benchmark system using a TZV2P basis set and a numerical threshold value of 10^{-12} a.u. for the overlap integral between two primitive Gaussian functions. For the systems with 32 and 64 H_2O each water molecule interacts with each other in the unit cell. Starting from roughly 200 water molecules, the interaction sphere of a water molecule is completely confined in the unit cell, i.e. for larger systems more and more water molecules inside the unit cell do not interact any longer with each other. This can be retrieved from the overlap matrix occupations starting with 256 H_2O, since the occupation is halved for each doubling of the simulation cell. Thus beginning with 256 H_2O in the unit cell the number of interactions grows linearly and similarly the sparsity of the matrices increases continuously. QUICKSTEP takes efficiently advantage of the matrix sparsity, however, this becomes only effective for more than 200 water molecules in the simulation cell. It is further important to recognize that the number of occupied orbitals

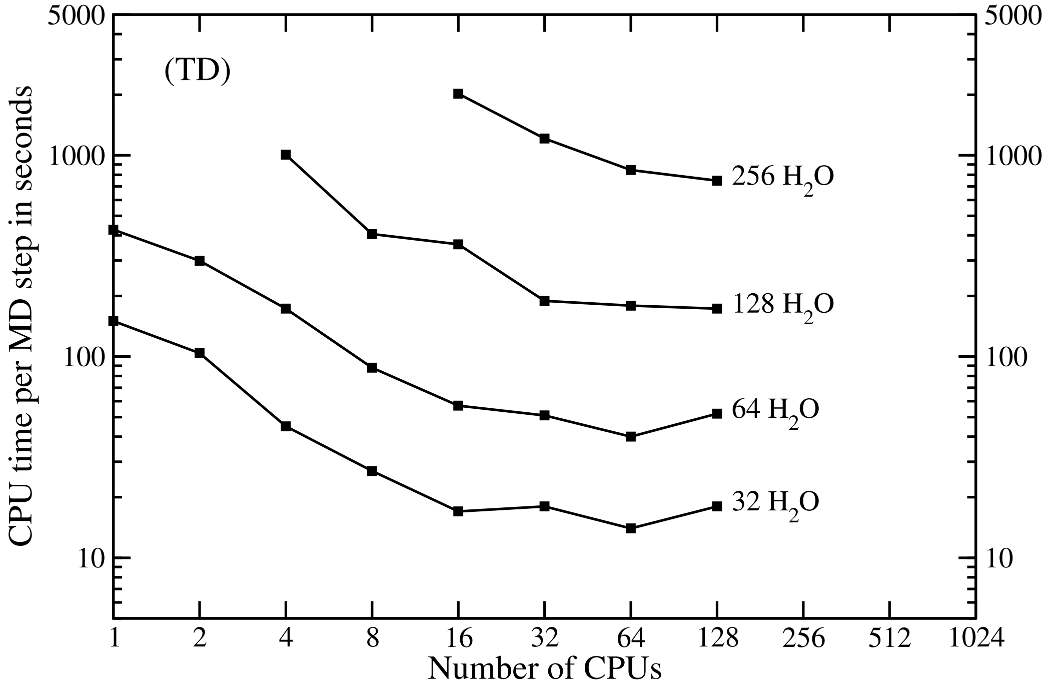
Table 3: Occupation of the overlap matrix applying a numerical threshold value of 10^{-12} for the overlap contribution of two primitive Gaussian orbital functions.

system	occupation
32 H_2O	100.0 %
64 H_2O	99.6 %
128 H_2O	85.1 %
256 H_2O	51.3 %
512 H_2O	25.8 %
1024 H_2O	12.9 %

N is significantly smaller than the total number of orbitals M (see Table 2). In this benchmark using the TZV2P basis set only 10 % of the orbitals are occupied. Thus any operation only dealing with the occupied orbitals (MN) is favorable compared to (M^2) for the full matrix. This is a crucial performance issue when comparing the eigensolvers implemented in QUICKSTEP. Figure 3 shows the timings for the benchmark systems of Table 2 using the IBM Regatta p690+ system at the Research Centre Jülich, called Jump. The Jump system consists of 39 compute nodes. Each node provides 32 Power4+ (1.7 GHz) processors. The processors are interconnected by an IBM High Performance Switch[1] (HPS). The results are given using a double logarithmic scale to show the scaling of the TD and the PD scheme. Each MD step included a full wavefunction optimization followed by a calculation of the forces on each atom. The total energy of the system was converged to 10^{-7} a.u. and the deviation of the electron count for the converged density was less than 10^{-5}. Ten MD steps were performed for each benchmark system (except 1024 H_2O) using a time step of 0.5 fs. The CPU timings of the last 5 MD steps were averaged. Figure 3 displays the obtained CPU timings per MD step for various CPU numbers and system sizes using the TD and the PD scheme. The missing data points are due to the limited memory per CPU which did not allow to run larger systems using only a small number of CPUs. The small systems with 32 and 64 H_2O can efficiently be run on a small number of CPUs. 64 H_2O need roughly one CPU minute per MD step, i.e. 2 CPU minutes per fs simulation time, when using 16 CPUs. The larger systems with 128 and 256 H_2O run efficiently on 32 and 64 CPUs, respectively. However, 14 minutes per MD step for 256 H_2O does not allow to obtain appropriate MD trajectories in reasonable time. It was not possible to run 512 H_2O, even if using 256 CPUs, since the TD scheme which is based on ScaLAPACK/BLACS requires to deal with a distribution of several full matrices during the SCF procedure exceeding the available memory.

A direct comparison of the two panels of Figure 3 shows that the PD scheme scales slightly

[1] This benchmark was run on the Jump system before the major software update (PTF7) in July 2004 which improved the MPI communication performance significantly.



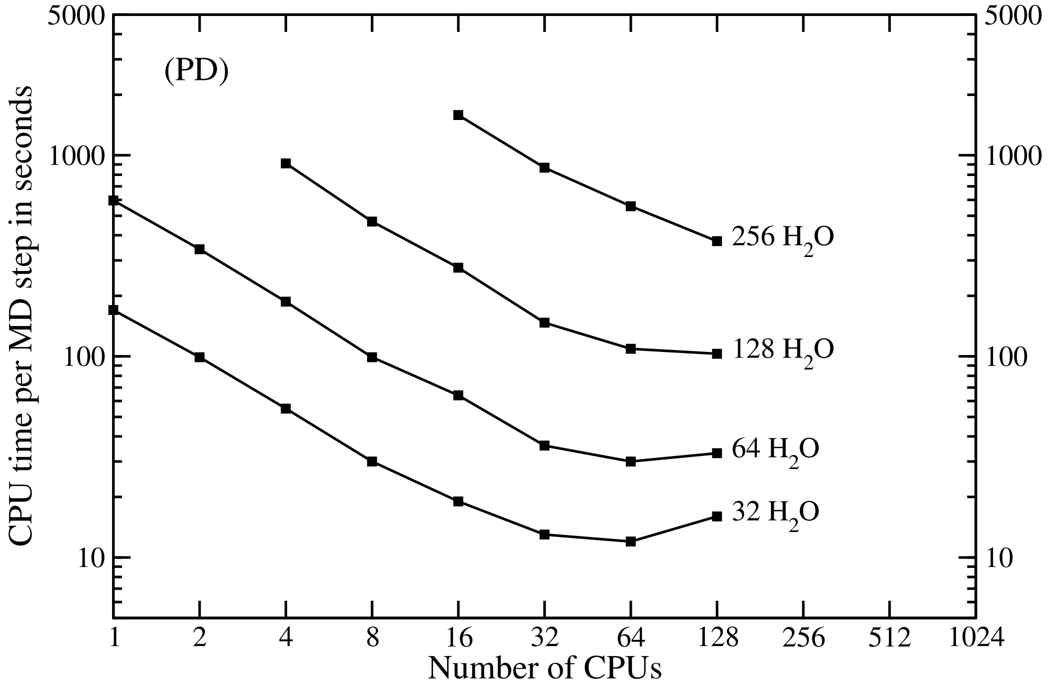


Figure 3: Scaling of the CPU time per MD step using the traditional diagonalization (TD) scheme and the pseudo diagonalization (PD) scheme for the benchmarks systems of Table 2. The calculations were performed on an IBM Regatta p690+ system with 32 Power4+ (1.7 GHz) per node interconnected by an IBM High Performance Switch (HPS).

better than the TD scheme. The small systems with 32 and 64 H_2O scale up to 32 CPUs and the largest system with 256 H_2O scales even up to 128 CPUs using the PD scheme. However, the absolute CPU times per MD step for the TD and the PD scheme are very close, even if the PD scheme requires less communication than the TD scheme. The PD scheme shows only for 256 H_2O on average significant shorter CPU times per MD step compared to the TD scheme. As described in section 5, the PD scheme can only be applied to sufficiently pre-converged wavefunctions. The TD scheme is employed until this convergence is achieved and thus no speed-up with respect to the TD scheme is obtained for the first SCF iteration steps. Furthermore, the initialization of the PD scheme requires at least once a diagonalization of the Kohn-Sham matrix including the calculation of *all* eigenvectors. This step turns out to be rather expensive. It is known that the orthonormalization of a large eigenvector set is computationally expensive step that involves a lot of communication. In fact, this SCF step may consume two or three times more CPU time than a normal TD SCF step and turns out to be a bottleneck for the larger systems. However, once the PD scheme is set up, the following iteration steps are less expensive than a TD step. Moreover, the PD steps are becoming cheaper and cheaper, since the number of matrix elements which have to be processed by the Jacobi rotations decrease continuously. However, a typical MD step only involves approximately 8 SCF iteration steps and at least two or three of these are normal TD steps followed by an expensive TD step providing the full eigenvector set. Thus there are only four or five SCF steps left for the faster PD scheme and finally nothing is gained compared to the pure TD scheme for most of the test systems.

By contrast, the OT method shows a much better performance as shown in Figure 4. The OT method needs less memory than the TD and the PD scheme, because it does not deal with full matrices during the SCF iteration and therefore it allows to run larger benchmark systems with up to 1024 water molecules in the unit cell. Also the scaling behavior of the OT method is much better. The small systems with 32 and 64 H_2O scale nicely up to 32 CPUs. A scaling beyond 32 CPUs cannot be expected, since the data blocks per CPU become too small to keep an SP4+ processor efficiently busy and the calculation will be completely dominated by the communication between the processes. At least one or two H_2O molecules per CPU are needed, formally. Also the larger benchmark systems show a better scaling with OT as indicated by the slope. The 512 H_2O system shows a continuous scaling up to 128 CPUs including 4 compute nodes of the Jump system. This shows that the scaling behavior of QUICKSTEP is also preserved when the processors of more than one compute node are employed.

7.2 Molecular and crystalline systems

As a final performance test for QUICKSTEP, geometry optimizations for a couple of molecular and crystalline systems were performed. The detailed characteristics of the employed molecular and crystalline benchmark systems is listed in Table 4. The DZVP basis set described in section 4 was used for all elements including hydrogen, even if the p-type

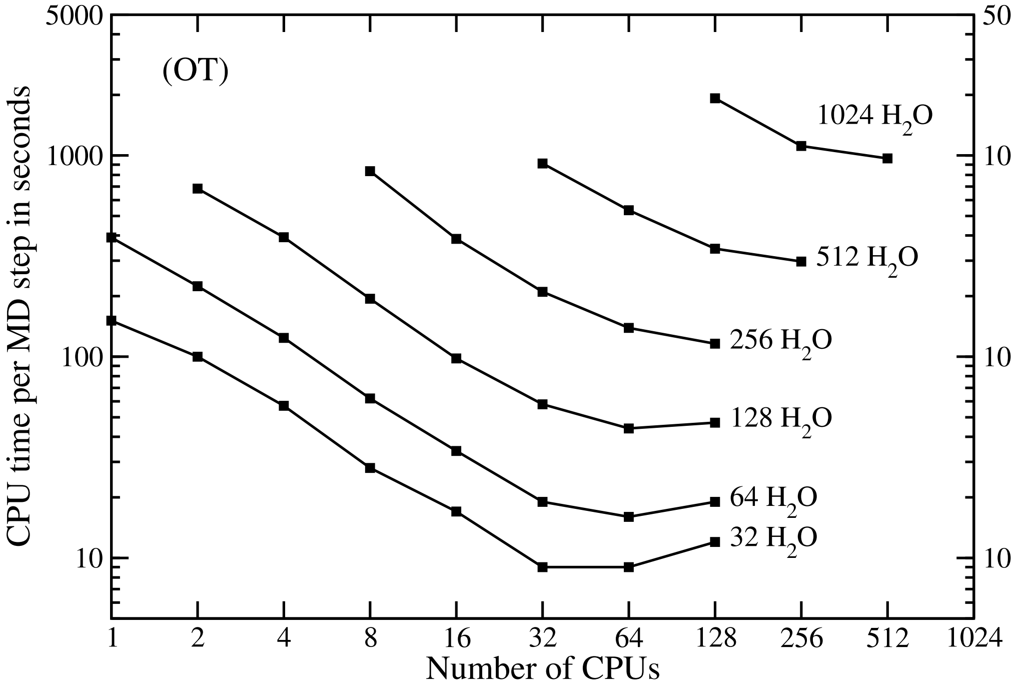


Figure 4: Scaling of the CPU time per MD step using the orbital transformation (OT) scheme for the benchmarks systems of Table 2. The calculations were performed on an IBM Regatta p690+ system with 32 Power4+ (1.7 GHz) per node interconnected by an IBM High Performance Switch (HPS).

polarization functions for hydrogen are not needed in most cases. It turned out that the quality of the DZVP basis set is sufficient for most of the applications. Optionally, a refinement of the structure can be obtained with the TZV2P basis set based on the structure pre-optimized with the DZVP basis set. It was shown in section 6 that the TZV2P basis set provides structures of high accuracy within the actual density functional approximation. The density cut-off for the plane waves expansion of the electronic density was chosen sufficiently large, i.e. in correspondence with the largest Gaussian function exponent of the

Table 4: Detailed characteristics of the employed molecular and crystalline benchmark systems. The number of atoms, electrons, Gaussian-type orbitals (M), and occupied orbitals (N) is listed. The employed exchange-correlation functional is given in brackets.

system	atoms	electrons	M	N	Cut-off [Ry]
C_{60} fullerene (LDA)	60	240	780	120	240
C_{180} fullerene (LDA)	180	720	2340	360	240
Grubbs catalysator (BP)	120	284	774	142	280
Taxol (BLYP)	113	328	908	164	280
[2]Catenan (BLYP)	164	460	1524	230	280
RNA duplex (BLYP)	368	1192	3444	596	320

Table 5: CPU time per geometry optimization step for the molecular and crystalline benchmark systems as described in Table 4. The calculations were performed on an IBM Regatta p690+ system with 32 Power4+ (1.7 GHz) per node interconnected via an IBM High Performance Switch (HPS).

system	4 CPUs	8 CPUs	16 CPUs
C_{60} fullerene	30	11	8
C_{180} fullerene	115	69	36
Grubbs catalysator	178	108	63
Taxol	208	118	74
[2]Catenan	246	138	92
RNA duplex	432	186	128

orbital basis set based on the accuracy of the computed electron count. The OT method was employed in all optimization runs.

C_{60} is the well-known hollow, soccer ball shaped molecule called buckminsterfullerene or simply bucky ball. The C_{180} fullerene is a bigger variety of the C_{60} which is also a hollow ball structure. Figure 5 shows the ruthenium based olefin metathesis catalysts also called after its inventor Grubbs catalysator. Taxol (see Figure 6) is a compound which is used as an anti-cancer drug. The [2]Catenan [23] is an electronically reconfigurable molecular switch which consists of two interlocked rings: (1) a tetracationic cyclophane that incorporates two bipyridinium units and (2) a crown ether containing a tetrathiafulvalene unit and a 1,5-dioxynaphthalene ring system located on opposite sides of the crown ether (see Figure 7). Finally, Figure 8 shows the unit cell of a fully hydrated RNA duplex crystal structure [24]. For all the molecular systems a sufficiently large unit cells were chosen to eliminate the interaction with the images. GTH pseudo potentials were employed for all structure optimization runs. For ruthenium and sodium the semi-core pseudo potential versions were used including 16 and 9 valence electrons, respectively.

The CPU times per geometry optimization step are listed in Table 5 using 4, 8, and 16 CPUs of one compute node of the Jump system at the Research Centre Jülich. Each geometry optimization step includes like an MD step a full wavefunction optimization followed by a calculation of the forces on all atoms. The timings depend not only on the size of the orbital basis set, but also on the selected exchange-correlation functional and the density cut-off. For instance, the C_{180} fullerene has a large orbital basis, however, the pseudo potential of carbon is softer than the pseudo potential of oxygen and thus it requires only the relatively small density cut-off of 240 Rydberg. Moreover, the gradient of the electronic density has not to be calculated in the framework of the local density approximation (LDA), whereas this is needed for the exchange-correlation functionals BLYP [9, 10, 11] and BP [9, 12] (see section 3) based on the generalized gradient approximation (GGA). A couple of geometry optimization steps can be performed for all the presented systems within the limits of an

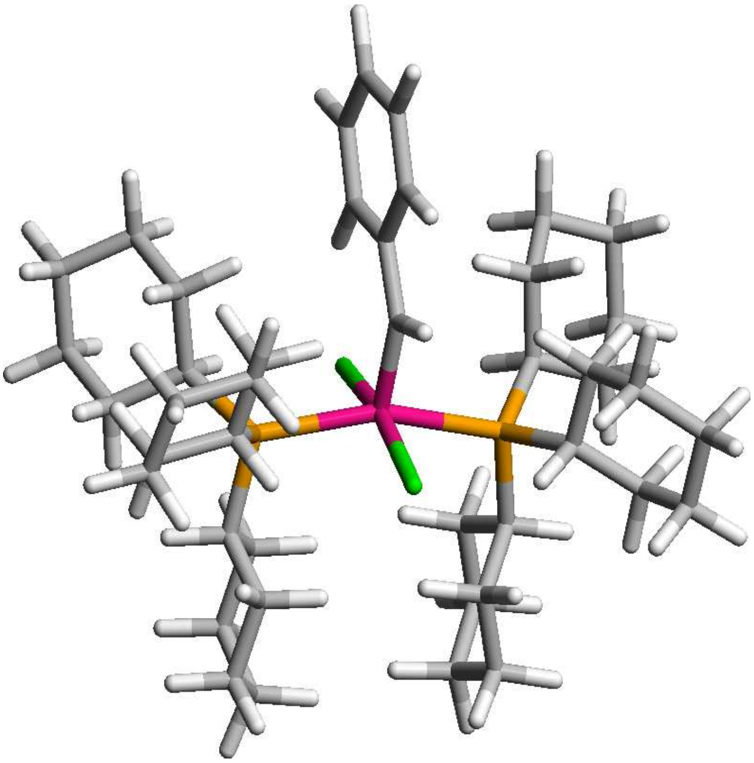


Figure 5: Grubbs catalysator ($RuC_{43}H_{72}P_2Cl_2$)

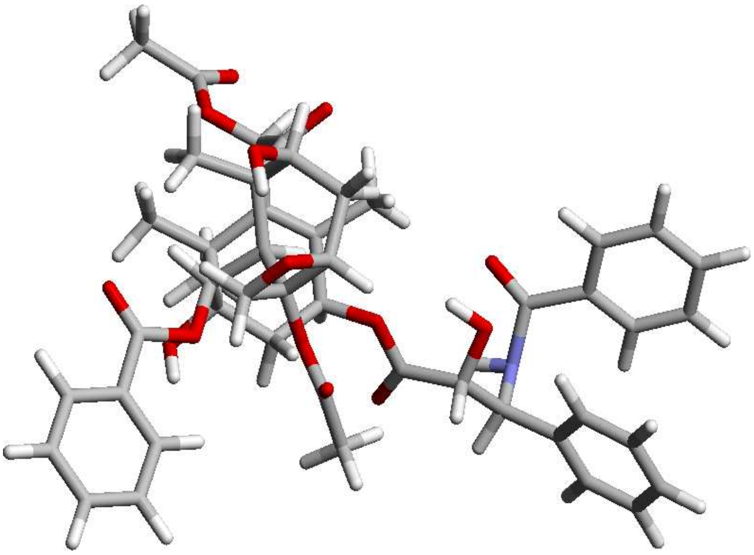


Figure 6: Taxol ($C_{47}H_{51}O_{14}N$)

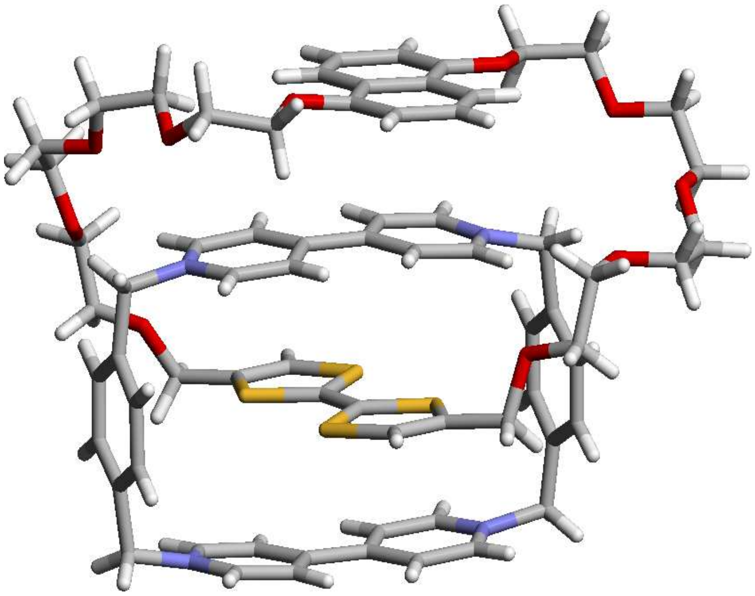


Figure 7: [2]Catenan ($C_{70}H_{76}O_{10}N_4S_4$)

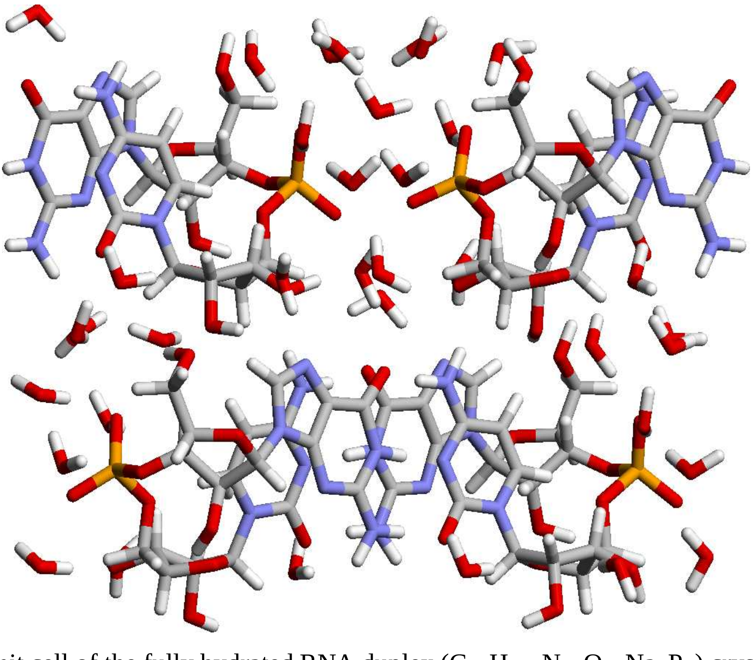


Figure 8: Unit cell of the fully hydrated RNA duplex ($C_{76}H_{168}N_{32}O_{84}Na_4P_4$) crystal structure

interactive job on the Jump system which provides up to 16 CPUs for 30 minutes. In that way, QUICKSTEP allows to optimize efficiently the structure of small and medium-sized molecular or crystalline systems.

8 Summary and outlook

It was shown that QUICKSTEP allows for fast and accurate density functional calculations of molecules and condensed phase systems. It provides the basic functionality needed to perform structure optimizations and to run Born-Oppenheimer molecular dynamics simulations. The nice scaling behavior of QUICKSTEP was proved using the new IBM parallel computer system Jump at the Forschungszentrum Jülich. The efficient parallelization of QUICKSTEP allows to obtain results in shorter time or to investigate larger systems.

Nevertheless, there are many possibilities to improve further the efficiency and functionality of QUICKSTEP. The extension of the GPW method to the Gaussian augmented plane waves (GAPW) method [25] will significantly speedup the calculations. Moreover, the GAPW method will also allow to perform all-electron density functional calculations [26].

Acknowledgment

The current state of the QUICKSTEP project is the result of the contributions from the whole CP2K developers team and in particular the QUICKSTEP developers, namely J. VandeVondele (University of Cambridge), T. Chassaing and J. Hutter (University of Zürich), and F. Mohamed and M. Krack (ETH Zürich).

Bibliography

[1] G. Lippert, J. Hutter, and M. Parrinello, *Mol. Phys.* **92**, 477 (1997).

[2] HPC-Chem (High Performance Computing in der Chemie), BMBF-Verbundprojekt, `http://www.fz-juelich.de/zam/hpc-chem`.

[3] The CP2K developers group, `http://cp2k.berlios.de`, 2004.

[4] J. VandeVondele, M. Krack, F. Mohamed, M. Parrinello, T. Chassaing, and J. Hutter, *Comput. Phys. Comm.*, submitted (2004).

[5] P. Hohenberg and W. Kohn, *Phys. Rev. B* **136**, B864 (1964).

[6] W. Kohn and L. J. Sham, *Phys. Rev.* **140**, A1133 (1965).

[7] S. Goedecker, M. Teter, and J. Hutter, *Phys. Rev. B* **54**, 1703 (1996).

[8] C. Hartwigsen, S. Goedecker, and J. Hutter, *Phys. Rev. B* **58**, 3641 (1998).

[9] A. D. Becke, *Phys. Rev. A* **38**, 3098 (1988).

[10] C. T. Lee, W. T. Yang, and R. G. Parr, *Phys. Rev. B* **37**, 785 (1988).

[11] B. Miehlich, A. Savin, H. Stoll, and H. Preuss, *Chem. Phys. Lett.* **157**, 200 (1989).

[12] J. P. Perdew, *Phys. Rev. B* **33**, 8822 (1986).

[13] F. A. Hamprecht, A. J. Cohen, D. J. Tozer, and N. C. Handy, *J. Chem. Phys.* **109**, 6264 (1998).

[14] J. P. Perdew, K. Burke, and M. Ernzerhof, *Phys. Rev. Lett.* **77**, 3865 (1996).

[15] T. H. Dunning, *J. Chem. Phys.* **90**, 1007 (1989).

[16] D. E. Woon and T. H. Dunning, *J. Chem. Phys.* **98**, 1358 (1993).

[17] P. Pulay, *J. Comput. Chem.* **3**, 556 (1982).

[18] M. J. Fengler, in *Beiträge zum Wissenschaftlichen Rechnen – Ergebnisse des Gaststudentenprogramms 2002 des John von Neumann-Instituts für Computing*, Technical Report IB-2002-12, edited by R. Esser, 119 (2002).
`http://www.fz-juelich.de/zam/gaststudenten/ib-2002-12.pdf`

[19] J. VandeVondele and J. Hutter, *J. Chem. Phys.* **118**, 4365 (2003).

[20] P. Pulay, *Chem. Phys. Lett.* **73**, 393 (1980).

[21] R. M. Dickson and A. D. Becke, *J. Chem. Phys.* **99**, 3898 (1993).

[22] CPMD, Version 3.7, copyright IBM Corp. 1990–2003, copyright MPI für Festkörperforschung Stuttgart 1997-2001; `http://www.cpmd.org`.

[23] C. P. Collier, G. Mattersteig, E. W. Wong, Y. Luo, K. Beverly, J. Sampaio, F. M. Raymo, J. F. Stoddart, and J. R. Heath, *Science* **289**, 1172 (2000).

[24] J. Hutter, P. Carloni, and M. Parrinello, *J. Am. Chem. Soc.* **118**, 8710 (1996).

[25] G. Lippert, J. Hutter, and M. Parrinello, *Theor. Chem. Acc.* **103**, 124 (1999).

[26] M. Krack and M. Parrinello, *Phys. Chem. Chem. Phys.* **2**, 2105 (2000).

Local Electron Correlation Methods with Density Fitting in MOLPRO

Hans-Joachim Werner, Martin Schütz[1] and Andreas Nicklaß

Institute for Theoretical Chemistry
University of Stuttgart
Pfaffenwaldring 55, 70569 Stuttgart, Germany
E-mail: werner@theochem.uni-stuttgart.de

1 Introduction

Recent advances in computer hardware and its exploitation through the techniques of high-performance computing, as well as advances in the development of approximate theories of quantum chemistry and quantum molecular dynamics, have together brought us to the position where theory can provide reliable answers to chemical questions that are of relevance not only to furthering fundamental understanding, but also to real-life industrial and environmental problems. This emergence of the relevance of quantum chemistry has been recognized through the award of the 1998 Nobel prize for chemistry, and through the wholesale adoption of quantum chemical methods by the chemical community, both academic and industrial.

The breakthrough in the ability to treat realistic chemical systems has to a large extent been due to improvements of density functional theory (DFT). The development of accurate density functionals and linear scaling techniques have made it possible to predict energies and molecular structures for molecules with 1000 or even more atoms. These techniques are the subject of other articles in this volume, and will not be further discussed here. The problem of DFT is that there is no way to systematically assess or to improve the

[1] present address: Institute for Physical and Theoretical Chemistry, University of Regensburg, Universitätsstraße 31, 93040 Regensburg

accuracy of a calculation, since the exact functional is unknown. Thus, the results depend on the chosen functional, and experience is needed to select a suitable functional for a given problem. The best currently available density functionals contain some parameters, which have been fitted to obtain best agreement with experiment, and therefore DFT can be viewed as a semi-empirical method. Even for such functionals it is often difficult or impossible to estimate the accuracy of a computed result, unless calculations are performed for similar molecules as contained in the training set.

On the other hand, there is a well established hierarchy of wave-function based *ab initio* methods, which allow to approach the exact solution of the electronic Schrödinger equation systematically. In most cases, such methods are based on the Hartree-Fock method as zeroth-order approximation. In Hartree-Fock, each electron moves independently in an average field caused by the the other electrons. To go beyond this approximation it is necessary to treat the electron correlation, i.e., the direct dynamical interaction of all electrons. Typically, the correlation energy (the difference between the exact energy and the Hartree-Fock energy) amounts only to 1% of the total energy. However, this is of the same order of magnitude as the energy differences which are relevant in chemistry, and since the correlation energy may change significantly from reactants to products, a high-level electron correlation treatment is mandatory for an accurate prediction of energy differences and molecular properties.

Unfortunately, the computational cost of wave-function based methods is much higher than for DFT, and for the best methods the increase of computer time with molecular size is extremely steep. Even for the simplest method to treat the electron correlation, second-order Møller-Plesset theory (MP2), the computer time formally scales as $\mathcal{O}(\mathcal{N}^5)$, where $\mathcal{N}$ is a measure of the molecular size. This means that doubling the size increases the computer time by a factor of 32 (this factor can be somewhat reduced by screening techniques [1]). For the more accurate fourth-order perturbation theory (MP4) or the coupled cluster method with single and double excitations (CCSD) and perturbative triple excitations [CCSD(T)] the scaling is even $\mathcal{O}(\mathcal{N}^7)$, i.e., the CPU time increases by a factor of 128 if the number of electrons is doubled. Therefore, despite the increase of computer speed by 3-4 orders of magnitude during the last decade, still only relatively small molecules (10-20 atoms) can be be treated by such accurate methods. Even the use of the largest supercomputers cannot significantly extend the applicability of conventional electronic structure methods.

The steep scaling of the computational cost with molecular size is mainly caused by the delocalized character of the canonical Hartree-Fock orbitals, which are traditionally used as a basis. However, (in insulators) electron correlation is a short range effect which decreases approximately as r^{-6}, where r is the distance between two local charge distributions. This can be exploited by localising the molecular orbitals and neglecting distant interactions. Based on a local ansatz originally proposed by Pulay [2, 3, 4, 5], our group has recently been able to develop local MP2 and CCSD(T) methods with linear [$\mathcal{O}(\mathcal{N})$] scaling of the computational cost [6, 7, 8, 9, 10, 11, 12, 13]. This has dramatically extended the range of applicability of such high-level methods, and energies for molecules with 100 atoms or more can now be computed with good basis sets.

The HPC-Chem project has significantly contributed to further develop and improve the efficiency of these unique new methods, which have been implemented in the MOLPRO package of *ab initio* programs [14]. Work has been done in four areas: firstly, the new local correlation methods have been parallelized. The use of parallel computer hardware is particularly useful for linear scaling methods, since then the size of the molecules which can be treated in given time increases linearly with the number of processors. Secondly, the pre-factor of the cost function has been reduced by the implementation of so called density fitting approximations for computing the integrals. Third, the method has been extended to open-shell cases. And finally, the slow convergence of the electron correlation energy with basis set size has recently been improved by the implementation of efficient local r_{12}-methods. Before describing these methods and new developments in more detail, we will give a short description of the MOLPRO package. This highlights the long history of method development which is typical for many quantum chemistry codes. The enormous amount of code (about one million lines) and the ”grown” structure makes it rather difficult to maintain, modularize, and parallelize the program. In this respect, the man-power provided by the HPC-Chem project has been extremely helpful.

2 About MOLPRO

The development of the MOLPRO program was started by Wilfried Meyer and Peter Pulay in 1969. At a very early stage they implemented a general Hartree-Fock program, including spin restricted (RHF) and unrestricted open-shell (UHF) treatments. Based on this, Pulay wrote the first analytical gradient program, which is one of the key developments in quantum chemistry and forms the basis for molecular geometry optimization. At the same time, Meyer developed his famous pseudo natural orbital configuration interaction method (PNO-CI) and the coupled electron pair approximation (CEPA) [15, 16]. These methods made it possible to obtain for the first time 80-90% of the electron correlation energy in small molecules like H_2O [15] and CH_4 [16]. A second generation of electron correlation methods was implemented into MOLPRO by H.-J. Werner and E. A. Reinsch in 1978 and the following years. These methods were based on Meyer’s theory of self-consistent electron pairs (SCEP) [17]. This is a particularly efficient direct CI method in which any complicated logic for computing the Hamiltonian matrix elements was eliminated by a suitable renormalization of the configurations. This leads to efficient matrix algebra, which allows to use modern hardware to the best possible extent. Additionally, the theory was formulated in a basis of non-orthogonal atomic orbitals (AOs). Only very much later it has turned out that the use of such a non-orthogonal basis is the key to linear scaling in local electron correlation methods. In the early 80ths, MOLPRO was extended by multiconfiguration-self-consistent field (MCSCF) [18, 19] and internally contracted multireference configuration interaction (MRCI) methods [20]. In 1984, one of the present authors (HJW) started to collaborate with Peter Knowles, and more efficient MCSCF and MRCI programs were written [21, 22, 23, 24, 25, 26], in which new techniques for computing Hamiltonian matrix elements and high-order density matrices [22, 24] were applied. Later on, efficient closed and open-shell coupled cluster methods [27, 28, 29, 30], multireference perturbation theory (MRPT2, MRPT3, CASPT2) [31, 32], and DFT methods were developed. Furthermore, analytic energy gradients for DFT, RHF, UHF, MCSCF [33], MP2 [34, 35], CASPT2 [36], and QCISD(T) [37], as well as many other utilities were implemented (for more details see `www.molpro.net`). By now, `MOLPRO` has the reputation to be one of the most efficient and general programs for highly accurate electronic structure calculations. It is world-wide used by hundreds of research groups.

The development of local electron correlation methods, which will be described in the following sections, was started in the group of H.-J. Werner in 1996 [6], and linear scaling was achieved for LMP2 for the first time in 1999 [7]. Quite recently, density fitting was introduced into `MOLPRO` by Manby and Knowles, and their integral program forms the basis for the local density fitting methods described in later sections.

3 Local correlation methods

As already mentioned in the introduction, the steep scaling of conventional electron correlation methods mainly originates from the delocalized character of the canonical molecular orbitals, which are traditionally used as a basis. This leads to a quadratic scaling of the number of electron pairs to be correlated, and in turn the correlation space for each pair also grows quadratically with molecular size, leading overall to an $\mathcal{O}(\mathcal{N}^4)$ increase of the number of double excitations and corresponding coefficients (amplitudes) with the number of electrons. However, dynamic electron correlation in non-metallic systems is a short-range effect with an asymptotic distance dependence of $\propto r^{-6}$ (dispersion energy), and therefore the high-order dependence of the computational cost with the number of electrons of the system is not physically imposed.

In order to avoid these problems, local correlation methods have been proposed by very many authors (see citations in our original work [6, 7, 11, 38]). Our method is based on the local correlation method of Pulay [2, 3, 4, 5]. As in most local correlation methods, localized occupied orbitals (LMOs)

$$\phi_i^{\text{loc}} = \sum_\mu \chi_\mu L_{\mu i} \tag{1}$$

are used to represent the Hartree-Fock reference wavefunction. The rectangular coefficient matrix $\mathbf{L}$ represents the LMOs in the AO basis $\{\chi_\mu\}$. The particular feature of the Pulay ansatz is to use non-orthogonal projected atomic orbitals (PAOs) to span the virtual space

$$\phi_r^{\mathrm{pao}} = \sum_\mu \chi_\mu P_{\mu r} \; . \tag{2}$$

The coefficient matrix (often called projector) is defined as

$$\mathbf{P} = \mathbf{1} - \mathbf{L}\mathbf{L}^\dagger \mathbf{S}^{\mathrm{AO}} \; , \tag{3}$$

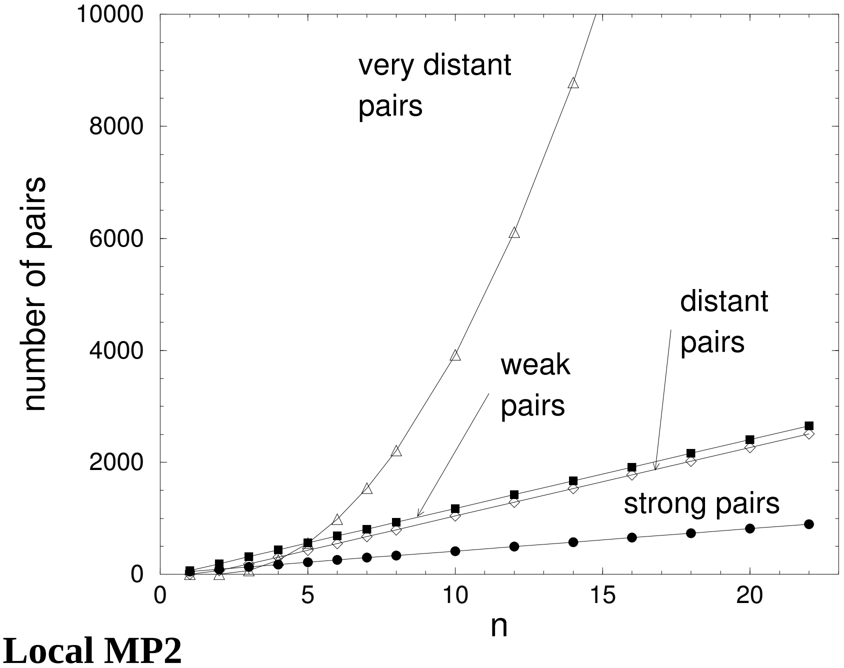
where $\mathbf{S}^{\mathrm{AO}}$ the overlap matrix of the AOs. Due to this definition, the PAOs are orthogonal to the occupied space

$$< \phi_r^{\mathrm{pao}} | \phi_i^{\mathrm{loc}} > = \left[\mathbf{P}^\dagger \mathbf{S}^{\mathrm{AO}} \mathbf{L} \right]_{ri} = 0 \tag{4}$$

but non-orthogonal among themselves. They are inherently local, and it is therefore possible to assign to each localized orbital an individual subset (*orbital domain*) of PAOs, which are spatially close to the region where the LMO is large. Similarly, for each orbital pair one can form *pair domains*, which are the union of the two orbital domains involved. Single excitations are made into the orbital domains, double excitations into the pair domains, and so on. For a given electron pair, the number of functions in a pair domain is independent of the molecular size, which reduces the scaling of the number of configuration state functions (CSFs) and corresponding amplitudes from $\mathcal{O}(\mathcal{N}^4)$ to $\mathcal{O}(\mathcal{N}^2)$. Furthermore, a hierarchical treatment of different electron pairs depending on the distance of the two correlated localized occupied molecular orbitals (LMOs) can be devised. *Strong* pairs, where the two electrons are close together, are treated at highest level, e.g. LCCSD, while *weak* and *distant* pairs can be treated at lower level, e.g. LMP2. For distant pairs it is possible to approximate the relevant two-electron integrals by multipole expansions [39]. *Very distant* pairs, which contribute to the correlation energy only a few micro-hartree or less, are neglected. An important advantage of the local treatment is that the number of strong, weak, and distant pairs scales linearly with molecular size, independently of the distance criteria used for their classification (cf. Figure 1). Only the number of the neglected very distant pairs scales quadratically. The number of amplitudes in each class scales linearly as well, since the number of amplitudes per pair is independent of the molecular size. This forms the basis for achieving linear scaling of the computational cost.

The errors introduced by these local approximations are normally very small. Typically they amount to only 1% of the valence-shell correlation energy for a triple zeta (cc-pVTZ) basis set if the domains are chosen as originally proposed by Boughton and Pulay [40]. The errors can be reduced and controlled by extending the domain sizes. For instance, about 99.8% of the canonical correlation energy for a given basis set are recovered if the standard domains are augmented by the PAOs at the first shell of neighboring atoms.

Figure 1: Number of pairs as a function of chain length for glycine polypeptides (gly)$_n$



3.1 Local MP2

In the local basis the first-order wavefunction takes the form

$$|\Psi^{(1)}\rangle \;=\; \frac{1}{2}\sum_{ij\in P}\sum_{rs\in[ij]} T^{ij}_{rs}|\Phi^{rs}_{ij}\rangle \qquad \text{with } T^{ij}_{rs}=T^{ji}_{sr}\;, \tag{5}$$

where P represents the orbital pair list and $[ij]$ denotes a pair domain of PAOs, which is defined in advance (for details, see Refs. [6, 7]). Here and in the following, indices i, j, k, l denote occupied orbitals (LMOs) and r, s, t, u virtual orbitals (PAOs). Note that the number of PAOs $r, s \in [ij]$ for a given pair (ij) is rather small and *independent* of the molecular size. Therefore, the individual amplitude matrices T^{ij}_{rs} are very compact and their sizes are independent of the molecular size. The total number of amplitudes T^{ij}_{rs} depends linearly on the molecular size and it is assumed that they can be stored in high-speed memory.

Since the local orbital basis does not diagonalize the zeroth order Hamiltonian, an iterative procedure is required to determine the amplitude matrices $(\mathbf{T}^{ij})_{rs} \equiv T^{ij}_{rs}$. The optimization is based on the minimization of the MP2 Hylleraas functional [41]

$$E_2 = \sum_{ij\in P}\sum_{rs\in[ij]}(2\mathbf{T}^{ij}-\mathbf{T}^{ji})_{rs})(\mathbf{K}^{ij}+\mathbf{R}^{ij})_{rs} \tag{6}$$

with respect to the amplitudes $(\mathbf{T}^{ij})_{rs}$, where

$$\mathbf{R}^{ij} \;=\; \mathbf{K}^{ij}+\mathbf{F}\mathbf{T}^{ij}\mathbf{S}+\mathbf{S}\mathbf{T}^{ij}\mathbf{F}-\sum_k \mathbf{S}\left[F_{ik}\mathbf{T}^{kj}+F_{kj}\mathbf{T}^{ik}\right]\mathbf{S} \tag{7}$$

are the so called residual matrices. The quantities $\mathbf{S}$ and $\mathbf{F}$ are the overlap and Fock matrices in the projected basis, respectively, and the exchange matrices $(\mathbf{K}^{ij})_{rs} = (ri|sj)$ represent a small subset of the transformed two-electron repulsion integrals (ERIs). Due to the absence of any coupling of amplitudes with ERIs in eq. (7) there is a one-to-one mapping between amplitude and exchange matrices. Hence, the number of required transformed ERIs is identical to the number of relevant amplitudes and therefore obviously of $\mathcal{O}(\mathcal{N})$. Note that this is a particular feature of the algebraic structure of the LMP2 equations, and no longer holds for LCCSD, which will be discussed in the next section.

At the minimum of E_2, the $(\mathbf{R}^{ij})_{rs}$ must vanish for $r, s \in [ij]$, and then E_2 corresponds to the second-order energy $E^{(2)}$. Thus, one has to solve the system of linear equations $(\mathbf{R}^{ij})_{rs} = 0$. The iterative method to solve these equations is described in detail in Ref. [6].

For a given pair (ij), only the local blocks $(\mathbf{K}^{ij})_{rs}$, $\mathbf{F}_{rs}$, and $\mathbf{S}_{rs}$ for $r, s \in [ij]$ are needed in the first three terms, while for the overlap matrices in the sum only the blocks connecting the domain $[ij]$ with $[ik]$ or $[jk]$ are required. The sizes of all these matrix blocks are independent of the molecular size. Taking further into account that for a given pair (ij) the number of terms k in the summation becomes asymptotically independent of the molecular size if very distant pairs are neglected, it follows that the computational effort scales linearly with molecular size.

The exchange matrices $\mathbf{K}^{ij}$ are conventionally obtained from the two-electron repulsion integrals in AO basis (ERIs) by a four-index transformation, i.e.

$$\begin{aligned} K^{ij}_{rs} = (ri|sj) &= \sum_{\mu} P_{\mu r} \sum_{\nu} P_{\nu s} \sum_{\rho} L_{\rho i} \sum_{\sigma} L_{\sigma j} (\mu\rho|\nu\sigma) \ , \\ (\mu\rho|\nu\sigma) &= \int d\mathbf{r}_1 \int d\mathbf{r}_2 \chi_\mu(\mathbf{r}_1)\chi_\rho(\mathbf{r}_1) r_{12}^{-1} \chi_\nu(\mathbf{r}_2)\chi_\sigma(\mathbf{r}_2) \ , \end{aligned} \tag{8}$$

where the coefficient matrices $\mathbf{L}$ and $\mathbf{P}$ represent the LMOs and PAOs, respectively, in the atomic orbital basis. The ERIs $(\mu\rho|\nu\sigma)$ in AO basis are computed on the fly and not stored on disk. In order to keep the computational effort for the transformation in eq. (8) as low as possible, the four indices are transformed one after the other. By defining suitable test densities and screening procedures it is possible to reduce the formal $\mathcal{O}(\mathcal{N}^5)$ scaling to $\mathcal{O}(\mathcal{N})$ [7]

For distant pairs the expensive integral transformation can be avoided by performing a multipole expansion [39]

$$K^{ij}_{rs} = \sum_{mn} Q^{ri}_m U^{ij}_{mn} Q^{sj}_n \ , \tag{9}$$

where $\mathbf{Q}^{ri}$ is a vector containing the multipole moments (up to octopole) of the overlap distribution ri and $\mathbf{U}^{ij}$ is an interaction matrix depending only on the centres of i and j. In this way, the $\mathbf{K}^{ij}$ for distant pairs can be evaluated in negligible time, and this leads to significant savings in large molecules.

Figure 2: CPU times (in seconds on P4/2 GHz) of LMP2/cc-pVDZ calculations as a function of chain length for glycine polypeptides (gly)$_n$.

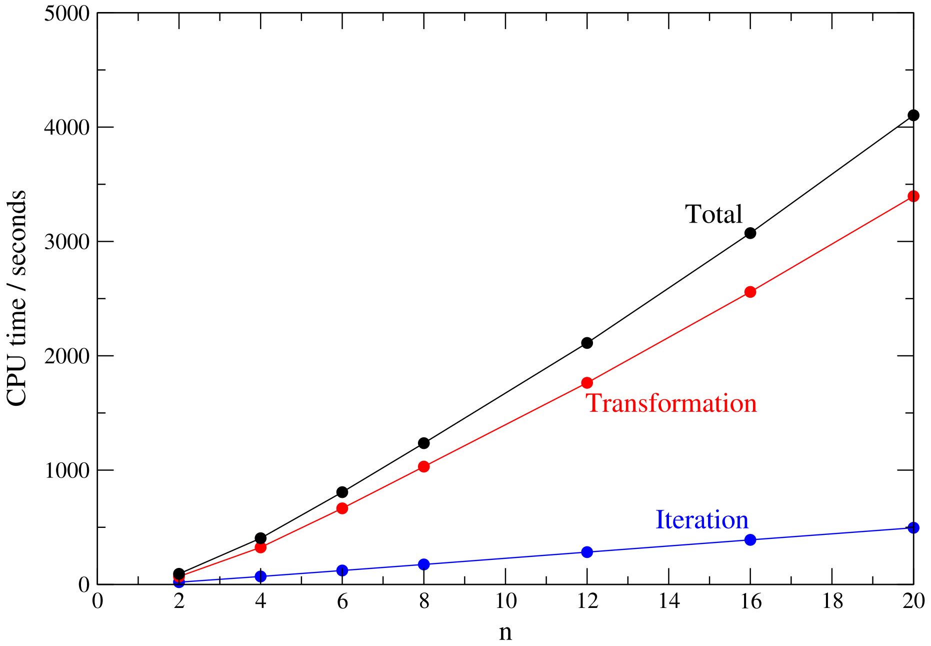


Figure 2 demonstrates the linear scaling behavior of LMP2 calculations for a linear glycine polypeptide chain, both for the transformation and the iteration steps. Naturally, the timings depend sensitively on the efficiency of integral screening. Therefore, the very extended model system used in Figure 2 represents an optimum case. The screening becomes less efficient for molecules with a more compact two- or three-dimensional structure or if larger basis sets are used. Some timings for more realistic molecules will be presented in sections 4.1 and 4.2. MP2 calculations for molecules of this size were previously not possible.

It should be pointed out that the absolute cost (i.e., the pre-factor) depends strongly on the basis set size per atom. If the double zeta (cc-pVDZ) basis set is replaced by a triple zeta (cc-pVTZ) set, as is required in order to obtain reasonably accurate results, the basis set size per atom increases by about a factor of 2. Since the transformation depends on the fourth power of the number of basis functions per atom, the corresponding increase of CPU time is a factor of 16 (in practice, a factor of 10-12 is found, since due to the larger matrix sizes some efficiency is gained). This means that LMP2 calculations for large molecules are still computationally very demanding, despite the linear scaling (which does not affect the dependence of the cost on the basis set quality). In the course of the HPC-Chem project this problem was attacked in three possible ways:

(i) parallelization: this reduces the elapsed time, but of course not the total computational cost. Some aspects will be discussed in section 5.

(ii) development of local density fitting methods. This approach, which will be discussed in section 4, strongly reduces the pre-factor. Most importantly, in this method the CPU time depends only cubically on the number of basis functions per atom, and therefore dramatic savings can be achieved for large basis sets.

(iii) development of a local MP2-R12 method, using density fitting techniques. This method improves the convergence of the correlation energy with basis set size. A description of this method is beyond the scope of the present article and will be presented elsewhere [42].

3.2 Local CCSD(T)

In coupled cluster theory, the wavefunctions is expanded as

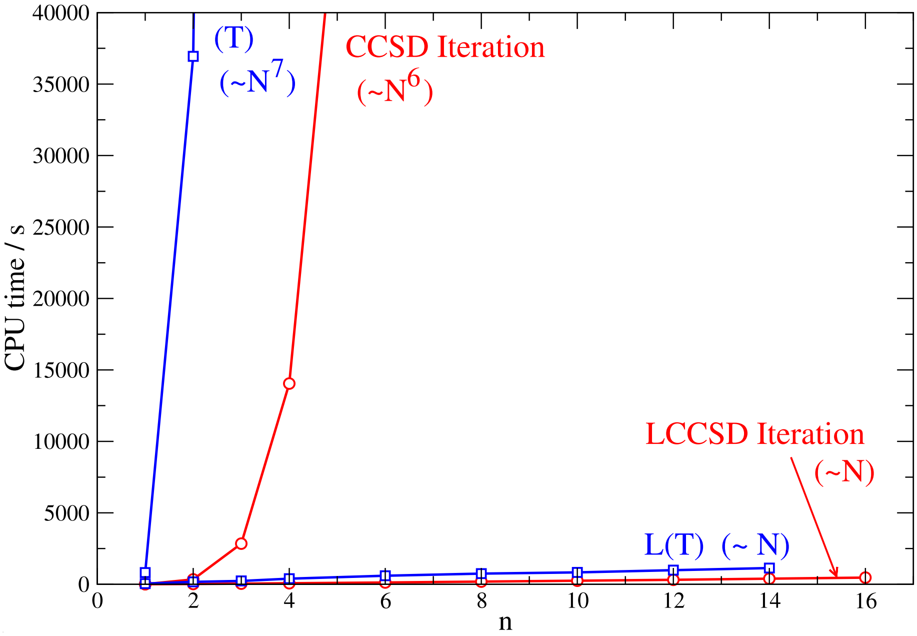
$$|\Psi^{\mathrm{CC}}\rangle \;=\; \exp(\hat{T})|\Psi^{\mathrm{HF}}\rangle \tag{10}$$

where $\hat{T}$ is a generalised excitation operator. In local coupled cluster theory with single and double excitations (LCCSD), this operator is approximated as

$$\hat{T} \;=\; \sum_{i}\sum_{r\in[i]} t_r^i \hat{E}_{ri} + \frac{1}{2}\sum_{ij\in P_s}\sum_{rs\in[ij]} T_{rs}^{ij}\hat{E}_{ri}\hat{E}_{sj} \tag{11}$$

where $\hat{E}_{ri}$ are spin-summed excitation operators, which excite an electron from LMO ϕ_i^{loc} to PAO ϕ_r^{pao}. The single and double excitations are restricted to orbital domains $[i]$ and pair domains $[ij]$, respectively. In most cases, it is sufficient to include only excitations for the *strong pairs* in the expansion, and to compute the energy contribution of the weak and distant pairs only at the LMP2 level. The list of strong pairs is denoted P_s. It is obvious that the number of single and double excitation amplitudes, t_r^i and T_{rs}^{ij}, respectively, scales only linearly with molecular size. Therefore, similar linear scaling techniques as for LMP2 can be devised for local CCSD [6, 11, 13], even though the algorithms are much more complicated. In contrast to LMP2, where one needs only exchange integrals $(ri|sj)$ over two LMOs and two PAOs, in the LCCSD case all other types of transformed integrals are required as well, in particular also those involving three and four PAOs $(ri|st)$, $(rs|tu)$. This requires additional integral-direct transformations. Furthermore, in contrast to the LMP2 case the LCCSD residual equations do contain products of amplitudes and ERIs. Nevertheless, it is still possible to restrict the LMO and PAO ranges in the related supermatrices to certain lists (*operator lists*) and domains (*operator domains*), which are larger that the amplitude pair lists and domains, but still independent of molecular size [11]. Nevertheless, in spite of these additional complications, dramatic computational savings were achieved, in particular for the calculation of the LCCSD residual matrices, which are evaluated in each iteration. Consequently, in strong contrast to conventional CCSD calculations, the computational effort in LCCSD calculations is often dominated by the time spent for the integral transformations. This problem is particularly severe for large basis sets. Fortunately, as

Figure 3: CPU times (in sec on Athlon 1.2 GHz) of LCCSD(T) calculations as a function of chain length for glycine polypeptides (gly)$_n$.



will be demonstrated in section 4, density fitting approximations can be used to overcome this bottleneck.

Figure 3 shows the scaling of the computational cost as a function of the chain length for the polyglycine model system. Perfect linear scaling is observed, and the savings are dramatic. For the largest calculation it can be estimated that the corresponding conventional CCSD(T) would take 56 years, while the local (T) calculation can be done in less than an hour [9, 10] (this does not include the time for the integral transformation, which in the present case dominates the computational effort.) Timings for some other more realistic applications will be presented in section 4.3.

4 Density fitting approximations

The idea of simplifying electron repulsion integrals by fitting products of orbitals in an auxiliary basis goes back at least as far as 1959, when Boys and Shavitt used the technique to compute the intractable 3-centre Slater integrals in calculations on the H_3 molecule [43]. The method saw relatively little use in *ab initio* theory for a number of decades, but proved invaluable in DFT [44], where the emphasis was on fitting the entire density in an auxiliary basis for the efficient solution of the Coulomb problem [45, 46, 47, 48]. The accuracy of the method has been carefully investigated, and it has been shown that with suitable fitting basis sets the errors are much smaller than other typical errors in the calculations, such as for instance basis set errors [49]. Optimized fitting basis sets are available for Coulomb [50] and exchange [51] fitting, as well as for MP2 [52, 49].

Some authors, including those of TURBOMOLE, denote the density fitting approximation as ”resolution of the identity” (RI). We prefer the name density fitting (DF) for two reasons: first, it is strictly not a resolution of the identity, since a Coulomb metric is used. Secondly, a resolution of the identity is used in the MP2-R12 method in a different context, and in our implementation of MP2-R12 both RI and DF approximations with different auxiliary basis sets are involved.

In the following we assume that the basis functions (AOs) $\{\chi_\mu\}$ and orbitals $\{\phi_i^{\rm loc}, \phi_r^{\rm pao}\}$ are real. The two-electron integrals $(\mu\nu|\rho\sigma)$ in the AO basis can be written as

$$(\mu\nu|\rho\sigma) \;\;=\;\; \int d\mathbf{r}_1 \int d\mathbf{r}_2 \frac{\rho_{\mu\nu}(\mathbf{r}_1)\rho_{\rho\sigma}(\mathbf{r}_2)}{r_{12}} \; . \tag{12}$$

In the density fitting methods the one-electron product densities $\rho_{\mu\nu}(\mathbf{r}_1) = \chi_\mu(\mathbf{r}_1)\chi_\nu(\mathbf{r}_1)$ are approximated by linear expansions

$$\rho_{\mu\nu}(\mathbf{r}) \;\approx\; \tilde{\rho}_{\mu\nu}(\mathbf{r}) \;\;=\;\; \sum_A D_A^{\mu\nu}\, \chi_A(\mathbf{r}) \; , \tag{13}$$

where $\chi_A(\mathbf{r})$ are fitting basis functions (e.g., atom-centred Gaussian-type orbitals, GTOs). The expansion coefficients $D_A^{\mu\nu}$ are obtained by minimizing the positive definite functional [45, 46, 47]

$$\Delta_{\mu\nu} \;\;=\;\; \int d\mathbf{r}_1 \int d\mathbf{r}_2 \, \frac{[\rho_{\mu\nu}(\mathbf{r}_1) - \tilde{\rho}_{\mu\nu}(\mathbf{r}_1)]\,[\rho_{\mu\nu}(\mathbf{r}_2) - \tilde{\rho}_{\mu\nu}(\mathbf{r}_2)]}{r_{12}} . \tag{14}$$

This leads to the linear equations

$$\sum_A D_A^{\mu\nu} J_{AB} \;\;=\;\; R_B^{\mu\nu} \; , \tag{15}$$

where

$$J_{AB} \;\equiv\; (A|B) \;\;=\;\; \int d\mathbf{r}_1 \int d\mathbf{r}_2 \, \frac{\chi_A(\mathbf{r}_1)\chi_B(\mathbf{r}_2)}{r_{12}} \; , \tag{16}$$

$$R_A^{\mu\nu} \;\equiv\; (\mu\nu|A) \;\;=\;\; \int d\mathbf{r}_1 \int d\mathbf{r}_2 \, \frac{\chi_\mu(\mathbf{r}_1)\chi_\nu(\mathbf{r}_1)\chi_A(\mathbf{r}_2)}{r_{12}} \; . \tag{17}$$

The 4-index integrals are then obtained by simple matrix multiplications

$$(\mu\nu|\rho\sigma) \;\approx\; \sum_A D_A^{\mu\nu} R_A^{\rho\sigma} \; . \tag{18}$$

In the LCAO approximation the molecular orbitals $\{\phi_r\}$ are linear expansions of the basis functions (AOs) $\{\chi_\mu\}$. Therefore, exactly the same fitting procedure as outlined above can be used to approximate integrals over molecular orbitals

$$(rs|tu) \;\approx\; \sum_A D_A^{rs} R_A^{tu} \;, \tag{19}$$

where $R_A^{rs} \equiv (rs|A)$ are 3-index integrals in the MO basis, and D_A^{rs} the corresponding fitting coefficients. Thus, in principle all types of integrals needed in any *ab initio* method can be approximated in this way.

We have implemented density fitting in Hartree-Fock (HF), density functional theory (DFT), second-order Møller-Plesset theory (MP2), and coupled cluster theory with single and double excitations (CCSD). If used with conventional methods and canonical orbitals, the advantage of density fitting is mainly a faster computation of the 2-electron integrals and a simplification of the integral transformation. Since the contributions of the 4-external integrals can be conveniently evaluated in the AO basis, significant savings result only for the integrals involving at least one occupied orbital. Furthermore, in HF, DFT, and MP2 the scaling of the CPU time with the number of basis functions per atom is reduced from $(N_{\mathrm{AO}}/N_{\mathrm{atoms}})^4$ to $(N_{\mathrm{AO}}/N_{\mathrm{atoms}})^3$. This means that the savings increase with increasing basis set. However, there is no change in the scaling with molecular size. Furthermore, the $(N_{\mathrm{AO}}/N_{\mathrm{atoms}})^4$ dependence cannot be removed in CCSD. In the following sections it will be shown that these disadvantages can be eliminated when the density fitting approximation is combined with local correlation methods.

4.1 DF-HF and DF-DFT

Density fitting is most easily applied to the Coulomb part of the Fock matrix, which is needed both in HF and DFT. This has sometimes been denoted RI-HF and RI-DFT, respectively, but for the reasons explained above we call the methods DF-HF and DF-DFT. The Coulomb contribution formally scales as $\mathcal{O}(\mathcal{N}^2)$, but close to linear scaling can be achieved using Poisson fitting basis sets [53, 54, 55, 56, 57]. The evaluation of the exchange contribution to the Fock matrix is more involved. It can be written as

$$k_{\mu\nu} \;\approx\; \tilde{k}_{\mu\nu} \;=\; \sum_i \sum_{A\in[i]_{\mathrm{fit}}} D_A^{\mu i} R_A^{\nu i} \;. \tag{20}$$

If canonical orbitals are used, all fitting functions A, B must be included in the summation, and the computational effort scales as $\mathcal{O}(\mathcal{N}^4)$ [51]. In the work carried out as part of the HPC-Chem project we have shown that this bottleneck can be avoided by using localized orbitals [38, 57]. The use of localized orbitals offers two advantages: First, the integrals $R_A^{\mu i} \equiv (\mu i|A)$ become negligible unless the basis functions χ_μ are close to the localized

Table 1: CPU times^a for Fock-matrix construction using conventional or Poisson auxiliary basis functions. The full cc-pVQZ basis set has been used.

		CPU times/s					
	N_{AO}	integrals	transf.	solve	assembly	Coulomb	total
Indinavir:							
GTO	3885	3577	1585	480	855	2572	9112
Poisson	3885	1529	837	615	890	663	4578
Pregnanediol:							
GTO	2345	1307	580	187	196	953	3237
Poisson	2345	771	450	243	227	302	2009

^{a} in seconds on AMD Opteron 2 GHz processor.

orbital ϕ_i^{loc}. Therefore, the number of non-negligible integrals $(\mu i|A)$ scales asymptotically as $\mathcal{O}(\mathcal{N}^2)$. Secondly, since the charge densities $\rho_{\mu i}(\mathbf{r}) = \chi_\mu(\mathbf{r})\phi_i^{\mathrm{loc}}(\mathbf{r})$ are local, a fitting basis located in the same region of space as $\rho_{\mu i}$ is sufficient for a given orbital i. The subset of fitting basis functions belonging to orbital i is called the fitting domain related to i and is denoted $[i]_{\mathrm{fit}}$. For large molecules, the number of functions χ_A in each fitting domain becomes independent of the molecular size. Then, the remaining number of required integrals $(\mu i|A)$ scales only linearly with molecular size. Furthermore, the fitting and assembly steps for a fixed i become independent of the molecular size, leading overall to linear scaling for the evaluation of the exchange contribution. The price one has to pay is that for each fitting domain a set of linear equations has to be solved. Since this scales with the third power of the number of coupled equations, the cost (i.e., the pre-factor of the linear scaling algorithm) will depend sensitively on the sizes of the fitting domains. Fortunately, relatively small fitting domains are sufficient to optimize the orbitals. Furthermore, the inverse of the Coulomb matrix **J** (or the corresponding LU-decomposition) can be reused in subsequent iterations, provided the fitting domains remain unchanged. In order to minimize the errors, the final energy can be computed accurately with the full fitting basis without explicit construction of the Fock matrix. For details of our implementation and an analysis of the errors caused by local fitting we refer to Ref. [57]. We found that the errors on relative energies, equilibrium distances, and harmonic vibrational frequencies are negligible.

The local fitting procedure leads to significant savings in the Fock matrix evaluation, in particular for large basis sets. Table 1 shows some timings for pregnanediol and indinavir molecules, using the cc-pVQZ basis sets. The calculation for indinavir includes almost 4000 basis functions. This calculation has not been possible using our conventional direct HF program. From the timings it can be seen that if a GTO fitting basis is used about 70% of the time is spent in the evaluation of the 3-index integrals (this corresponds to the sum of columns *integrals and Coulomb*). It should be noted that our current integral program is still far from being optimum. A new implementation, using similar techniques as used in

TURBOMOLE, is currently in progress, and this should lead to a significant reduction of the integration times.

A further speedup can be achieved by using the Poisson equation [53, 54, 55, 56]

$$\hat{P}v[\rho] = \rho, \tag{21}$$

(where $\hat{P} = -(4\pi)^{-1}\nabla^2$), which relates the Coulomb potential $v[\rho]$ to the density ρ that gave rise to it. For a given basis function $|\chi_A >\equiv |A>$ one can define new functions $\hat{P}|\chi_A >\equiv |\tilde{A}>$, formed by the application of the Poisson operator $\hat{P}$. Functions of this type are called Poisson functions to distinguish them from standard Gaussian basis functions. The integrals $J_{\tilde{A}\tilde{B}}$ and $(\tilde{A}|\mu\nu)$ then simplify to integrals over the Laplacian

$$J_{\tilde{A}\tilde{B}} \;=\; \int d\mathbf{r}\, \chi_A(\mathbf{r})\hat{P}\chi_B(\mathbf{r}) =< A|\hat{P}|B > \tag{22}$$

and to 3-index overlap integrals

$$(\tilde{A}|\mu\nu) \;=\; \int d\mathbf{r}\, \chi_A(\mathbf{r})\chi_\mu(\mathbf{r})\chi_\nu(\mathbf{r}) =< A|\mu\nu > , \tag{23}$$

respectively. These 3-dimensional integrals are much faster to evaluate then the 6-dimensional Coulomb integrals. Furthermore, the 3-index overlap integrals decay fast with the distance between A and $\mu\nu$, and therefore the number of integrals scales linearly with molecular size. Unfortunately functions like $\hat{P}\chi$ carry no total charge [55, 56]

$$q = \int d\mathbf{r}\, \hat{P}\chi(\mathbf{r}) = 0, \tag{24}$$

nor indeed any net multipole of any order, because

$$q_{\ell m} = \int d\mathbf{r}\, r^\ell Y_{\ell m}(\mathbf{r}/r)\hat{P}\chi(\mathbf{r}) = 0. \tag{25}$$

One must therefore augment the Poisson basis set with a few standard basis functions. For the exact fitting of arbitrary densities it is sufficient to have only a single, standard basis function of each angular momentum. We have optimized Poisson fitting basis sets for the cc-pVTZ and cc-pVQZ basis sets. The errors of the final HF energies are of similar size (or even smaller) as with the corresponding optimized GTO basis set of Weigend [51]. Using the Poisson basis sets, the CPU time for Fock matrix evaluation is typically reduced by a further factor of 2 (see Table 1).

Figures 4 and 5 show the scaling of the CPU time as a function of molecular size for a linear chain of glycine polypeptides and polyalanine helices, respectively. The latter systems have a 3-dimensional structure and are much more compact than the linear glycine chains. The scaling obtained with the Poisson basis is better than quadratic. By comparing the timings for the linear glycine chain and the alanine helices it is found that for molecules of comparable size the time is approximately a factor of 2 larger in the latter case. This is caused by less efficient screening in the 3-dimensional case. However, the effect of screening is much less pronounced than in conventional direct Fock matrix evaluation.

Figure 4: CPU times for DF-HF Fock-matrix evaluation for glycine polypeptides (gly)$_n$ as a function of the chain length n

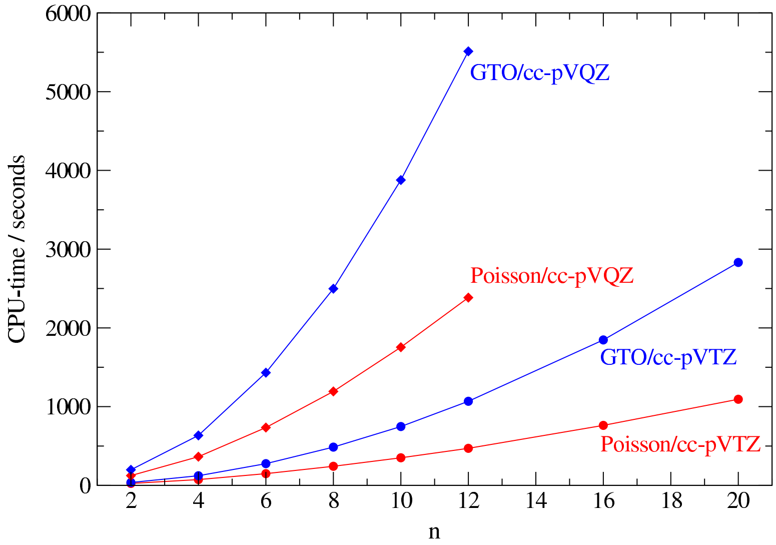
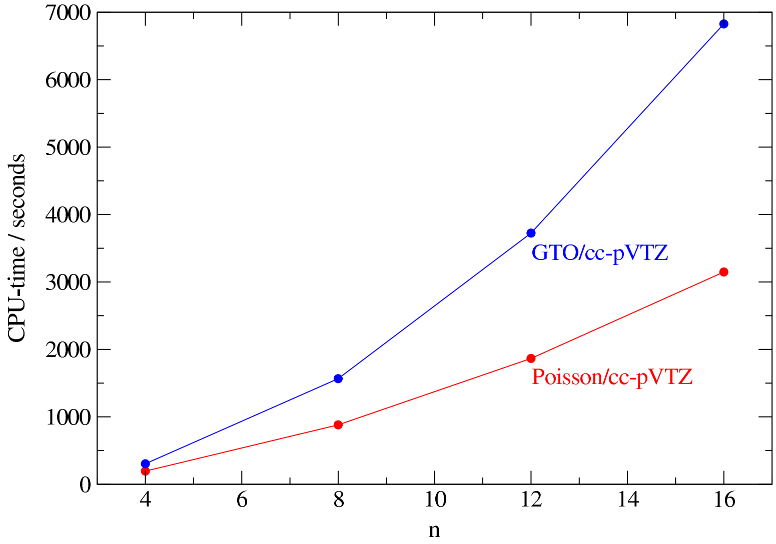


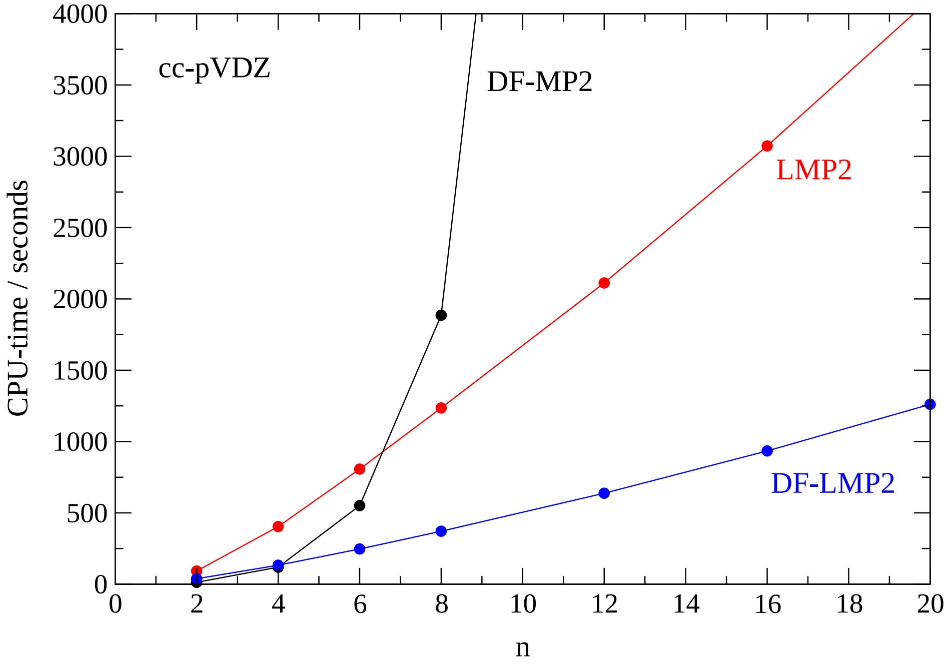
Figure 5: CPU times for DF-HF Fock-matrix evaluation for alanine helices (ala)$_n$ as a function of the chain length n



4.2 DF-LMP2 and DF-LMP2 gradients

The first implementation of density fitting in MP2 theory was described by Feyereisen, Fitzgerald and Komornicki [58]. This reduced the cost for evaluating the transformed integrals, but the scaling with molecular size was still $\mathcal{O}(\mathcal{N}^5)$. This bottleneck can be eliminated if the DF method is combined with the local correlation methods. In LMP2 theory, one needs 2-electron integrals $K^{ij}_{rs} = (ri|sj)$ over two occupied orbitals $\phi^{\rm loc}_i$, $\phi^{\rm loc}_j$

Figure 6: CPU times (in seconds on P4/2 GHz) for DF-MP2 and DF-LMP2 calculations for glycine polypeptides (gly)$_n$ as a function of the chain length n



and two PAOs ϕ_r^{pao}, ϕ_s^{pao}. In the DF approximation these integrals can be written as

$$K_{rs}^{ij} = \sum_{A\in[ij]_{\text{fit}}} D_A^{ri} R_A^{sj} \,, \tag{26}$$

Since the PAOs r, s are close to the localized orbitals i, j, the charge densities ρ_{ri}, ρ_{sj} are local, and therefore for a given pair (ij) the fitting functions can be restricted to a pair fitting domain $[ij]_{\text{fit}}$. This means that the fitting coefficients D_{ri}^A are the solution of the linear equations in the subspace of the fitting domain $[ij]_{\text{fit}}$. This is very similar to the local fitting of the exchange matrix in DF-HF, but in this case a system of linear equations has to be solved for each pair (ij). Alternatively, one can use orbital dependent fitting domains $[i]_{\text{fit}}$ for all pairs (ij) involving a particular orbital i. These orbital fitting domains are larger than the pair fitting domains, but one has to solve the linear equations only once for each correlated orbital. A possible disadvantage of using orbital-dependent fitting domains is that this is not symmetric with respect to exchange of i and j. We therefore denote this as ”asymmetric fitting” procedure. However, this can be cured by using the *robust* fitting formula [59]

$$K_{rs}^{ij} = \sum_{A\in[i]_{\text{fit}}} D_A^{ri} R_A^{sj} + \sum_{B\in[j]_{\text{fit}}} R_B^{ri} D_B^{sj} - \sum_{A\in[i]_{\text{fit}}} \sum_{B\in[j]_{\text{fit}}} D_A^{ri} J_{AB} D_B^{sj} \,, \tag{27}$$

Table 2: Analysis of CPU times^a for indinavir (cc-pVTZ, 2008 BF).

	LMP2	DF-MP2	DF-LMP2
Integrals	25540	2992	2816
Transformation	56620	4795	970
Fitting	0	3364	362
Assembly	0	82663	38
Total $(ri\|sj)$	82160	93900	4208
Iteration	3772	0	3775
Total MP2	86177	93914	8247

a) In seconds for HP ZX6000 Itanium2/900 MHz.

which can be rewritten as

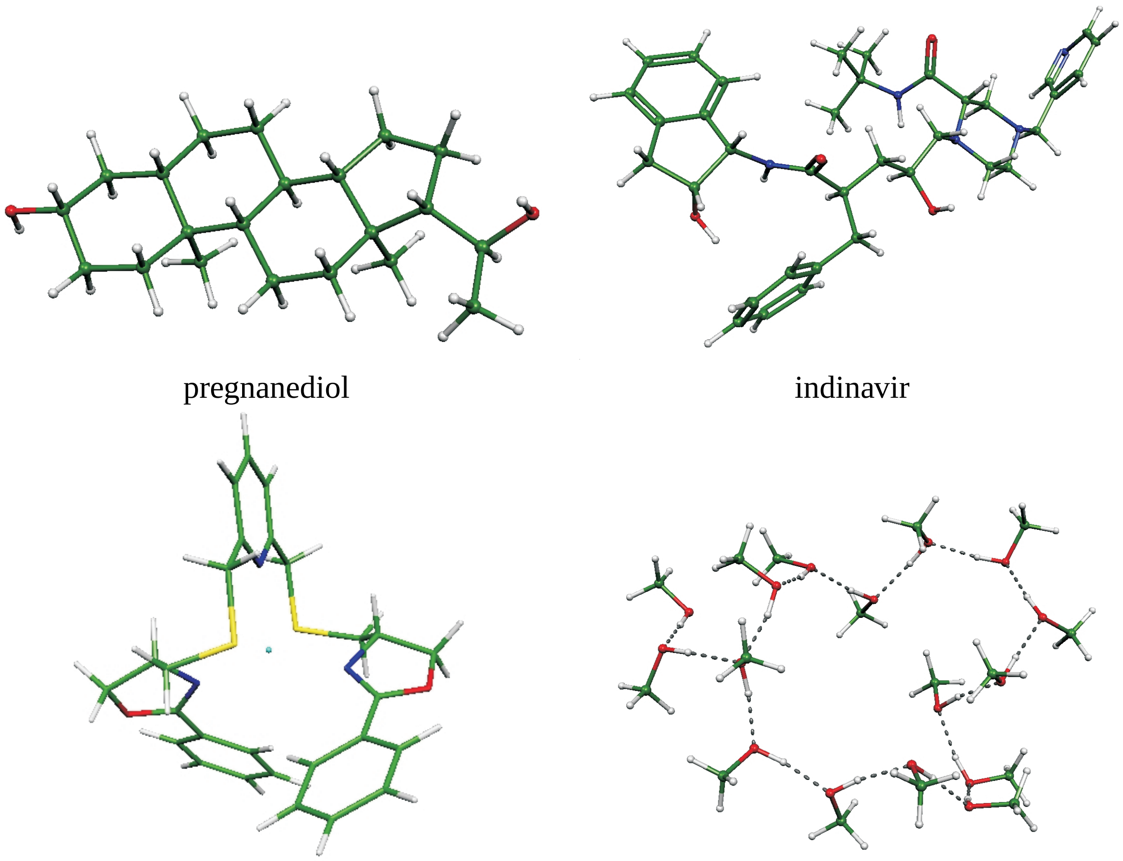
$$K_{rs}^{ij} = \sum_{A\in[i]_{\rm fit}} D_A^{ri} R_A^{sj} + \sum_{B\in[j]_{\rm fit}} \tilde{R}_B^{ri} D_B^{sj} \tag{28}$$

with

$$\tilde{R}_B^{ri} = R_B^{ri} - \sum_{A\in[i]_{\rm fit}} D_A^{ri} J_{AB} \; . \tag{29}$$

Thus, an additional matrix multiplication is required to evaluate $\tilde{R}_B^{ri}$, and the computational effort in the robust assembly step [eq. (28)] is doubled as compared to the non-symmetric approximation, in which the second term of eq. (28) is neglected (note that $\tilde{R}_B^{ri}$ vanishes if the full fitting basis is used). In our original work [38] we used the asymmetric procedure with orbital fitting domains $[i]_{\rm fit}$ which were the union of all orbital fitting domains $[ij]_{\rm fit}$ for a fixed i. Here $[i]_{\rm fit}$ includes all auxiliary basis functions centered at the atoms belonging to the orbital domain $[i]$. It was found that with this choice the errors due to local fitting are negligible, and robust fitting was not needed. More recently, we found that much smaller fitting domains are sufficient if robust fitting is performed. Some results will be presented in section 4.3. Independent of the choice of the local fitting domains, their sizes are independent of the molecular size, and – provided that distant pairs are neglected – linear scaling of the computational effort can be achieved. Table 2 shows some timings for indinavir. The savings by the local fitting approximations is most dramatic for the assembly step [eq. (28)]. This step takes 82663 seconds in canonical DF-MP2, but only 36 seconds in DF-LMP2. This is a reduction by a factor of 2175! The saving of DF-LMP2 vs. LMP2 is about a factor of 20 for the evaluation of the integrals $(ri|sj)$ and a factor of 10 overall. The latter factor is smaller, since the time to solve the LMP2 equations is the same with or without density fitting. While this time is only a small fraction of the total time in LMP2, it amounts to almost 50% in DF-LMP2. The scaling of the CPU time as function of molecular size is shown in Figure 6. It can be seen that the scaling is very close to linear, both for LMP2 and DF-LMP2. This has made it possible to perform LMP2 calculations for much

Figure 7: Some plots of molecular systems for which geometry optimizations with the analytic DF-LMP2 gradient program have been performed. In all cases a basis set of triple-zeta size has been used.



pregnanediol

indinavir

Zn(II) Δ_2 complex

$(CH_3OH)_{16}$

larger molecules than previously possible. For instance, we have been able to perform DF-LMP2 calculations for indinavir with the full cc-pVQZ basis (3885 basis functions). Without local density fitting, not even the Hartree-Fock calculation had been possible. Similar techniques have been used to implement analytic energy gradients for DF-HF/DF-LMP2. The theory is quite involved and we refer to our original work [35] for details. Most importantly, all 4-index objects except for the LMP2 amplitudes are entirely avoided in this method. In particular, there are no 4-index derivative integrals to compute, in contrast to the DF-MP2 gradient method of Weigend and Häser [60]. The LMP2 amplitudes, which by virtue of the local approximation are very compact anyway, are contracted with 3-index integrals to a 3-index object immediately after the DF-LMP2 energy calculation. Local fitting is used both in the evaluation of the direct gradient contributions as well as in the coupled-perturbed Hartree-Fock equations. Again, this leads to significant savings, and much larger molecules can be treated than before. The additional errors in the geometries,

Table 3: Timings (in minutes) of the individual steps of a DF-LMP2 gradient calculation for some exemplary test molecules. The calculations were performed on an AMD Opteron 2.0 GHz processor machine.

molecule	Zn(II)Δ_2 complex	(MeOH)$_{16}$	pregnanediol	indinavir
basis	TZVP	AVDZ	VTZ(f/P)	VTZ(f/P)
N_{AO}	1114	1312	1014	1773
N_{AUX}(MP2)	2349	3776	2943	5055
N_{AUX}(JK)	3599	4448	2897	4965
DF-HF	461	251	109	375
DF-LMP2	155	75	57	376
LMP2 iter.	144	32	41	285
DF-LMP2 GRD	286	341	155	526
Z-CPHF	175	177	77	231
DF-HF GRD	134	91	54	163
TOTAL	1036	758	375	1440

inflicted by density fitting are clearly negligible, as was demonstrated in Ref. [35]. Some examples of molecular systems for which geometry optimizations have been performed, are shown in Figure 7. Table 3 compiles the corresponding timing results. Evidently, the correlation-specific parts of the gradient do not dominate the overall cost of the calculation. The Hartree-Fock-specific parts turn out to be roughly as expensive. This implies that for a given AO basis set size a DFT gradient based on a hybrid functional is not much faster than the DF-LMP2 gradient, even when employing density fitting as well. This is particularly interesting for applications in the field of intermolecular complexes and clusters, where DFT has severe shortcomings due to its inability to describe dispersive forces. The new DF-LMP2 gradient has recently been used in a combined experimental and theoretical study on predetermined helical chirality in pentacoordinate Zinc(II) complexes [61]. One of these complexes is shown in Figure 7. The five coordinating atoms in the ligand are represented by one pyridine nitrogen atom $\mathrm{N}_{\mathrm{pyridine}}$, two oxazoline nitrogen atoms $\mathrm{N}_{\mathrm{oxazoline}}$, and two further atoms, denoted by X. Experiment and theory agree that, depending on the choice of X, the Zn complex has either a Λ_2 (X=O) or a Δ_2 (X=S) conformation. Furthermore, there is agreement that the Δ_2 conformer has perfect C_2 symmetry, whereas the symmetry of the Λ_2 conformer is distorted. This is also evident from Table 4, which compares experimental and theoretical values of the most relevant geometrical parameters of these two conformers. As can be seen, there is good agreement between the X-ray and the theoretically predicted structures. It is unlikely that calculations at that level would have been possible with any other program currently available.

Table 4: Comparison of selected bond lengths and angles of the Λ_2 (X=O) and Δ_2 (X=S) conformers of the pentacoordinate Zinc(II) complex studied in Ref. [61].The experimental values were determined by X-ray structure analysis. The theoretical values were obtained by performing geometry optimizations using the analytic DF-LMP2 gradient program. The TZVP basis [62] with the related fitting basis sets [63] was used. For Zinc a quasi relativistic energy-adjusted pseudopotential based on the Ne-like Zn^{20+} core together with the related 6s5p3d1f AO basis [64] was employed. All values are given in Å and degrees.

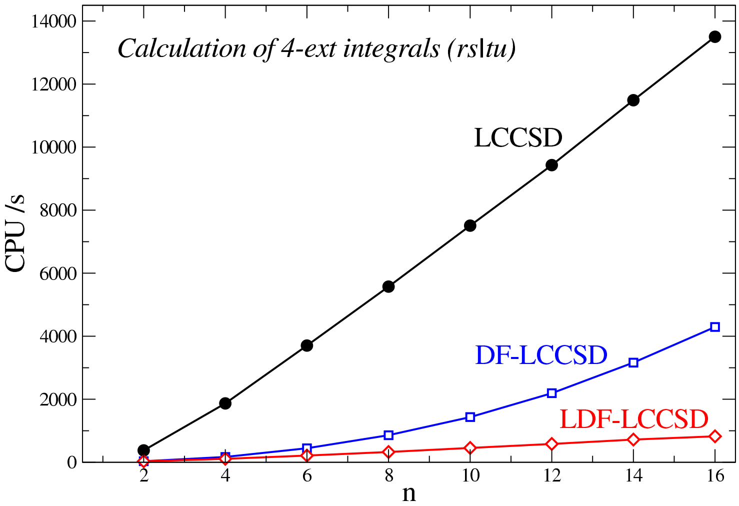
	Λ_2 (X=O)		Δ_2 (X=S)	
	X-ray	DF-LMP2	X-ray	DF-LMP2
$N_{pyridine}$–Zn	2.03	2.05	2.10	2.12
$N_{oxazoline1}$–Zn	1.96	1.97	1.98	2.00
$N_{oxazoline2}$–Zn	1.95	1.96	1.98	2.00
X_1–Zn	2.22	2.23	2.53	2.55
X_2–Zn	2.28	2.25	2.53	2.55
$\angle(N_{pyridine},Zn,N_{oxazoline1})$	114	115	110	110
$\angle(N_{pyridine},Zn,N_{oxazoline2})$	116	116	110	110
$\angle(N_{pyridine},Zn,X_1)$	77	76	84	83
$\angle(N_{pyridine},Zn,X_2)$	76	75	84	83
$\angle(X_1,Zn,X_2)$	153	150	169	167

4.3 DF-LCCSD

As already pointed out in section 3.2, the integral transformations constitute the most severe bottleneck in local coupled-cluster calculations, despite linear scaling algorithm [11] Density fitting approximations are therefore particularly useful in LCCSD. In a first step, such methods been implemented for the most expensive integral class, namely those integrals involving four PAOs (4-external integrals) [65]. Speedups by up to two orders of magnitude were achieved in this step. Similar to the case of DF-LMP2, local fitting domains can be introduced to restore the $\mathcal{O}(\mathcal{N})$ scaling behavior of the parental LCCSD method, as is shown in Figure 8. Furthermore, even though the scaling with respect to the number of basis function per atom, i.e., the basis set size, cannot be reduced from quartic to cubic as in the case of DF-HF and DF-LMP2, the computational speedups due to DF increase substantially when larger basis sets are used.

Very recently, our program has been extended such that density fitting is employed for all types of integrals needed in coupled cluster theory [66]. Table 5 shows some preliminary timings for the $(gly)_4$ test case. The times for the integral transformations are reduced by a factor of 10-20 if full fitting domains are used, and up by a factor of 34 (for the 0-2 external integrals) with local fitting domains. In the LCCSD without density fitting, the transformed 3- and 4-external integrals are stored on disk. Due to the use of domains, the number of these integrals scales linearly with molecular size, and the necessary contractions with the amplitude vectors and matrices in each iteration are very fast. However, in each iteration

Figure 8: CPU times (in seconds on Athlon/1.2 GHz) for the calculation of the 4-external integrals as a function of the chain length n for poly-glycine peptides (Gly)$_n$, $n = 1 \ldots 16$, The cc-pVDZ orbital basis set together with the corresponding MP2 fitting basis of Weigend *et al.* [49] was employed. In DF-LCCSD the full fitting basis and in LDF-LCCSD local fitting domains were used.



an additional Fock-like operator $\mathbf{G}(\mathbf{E})$ must be computed in the full PAO basis

$$[\mathbf{G}(\mathbf{E})]_{rs} = \sum_i \sum_u [2(rs|ui) - (ru|si)]\, t_u^i \, . \tag{30}$$

Due to the long-range nature of the Coulomb operator, domains cannot be used for the indices r, s in the first term without introducing significant errors [11]. This operator is therefore computed in the AO basis, and the time is the same as for evaluating a Fock matrix. In the density fitted case, this operator can be computed using the same techniques as described in section 4.1, and this removes the main bottleneck in the iteration time.

In the density fitting case one has furthermore the option of either store the transformed 3- and 4-external 4-index integrals as in the standard case, or to store the smaller sets of 3-index integrals $(rs|A)$, $(ri|A)$ (and/or the corresponding fitting coefficients) and assemble the 4-index integrals on the fly in each iteration. The latter case is faster in the transformation step but requires significantly more time per iteration. The two different cases are shown in Table 5 as well. Clearly, overall, storing the 4-index quantities is advantageous, provided there is enough disk space available.

The errors introduced by the density fitting approximation are demonstrated in Table 6. It is found that the errors for DF-LCCSD are even smaller than for DF-LMP2, despite the fact that optimized MP2 fitting basis sets of Weigend et al. [52] have been used. This is due to a fortuitous error cancellation: While the errors at the LMP2 level are positive (relative to the result without density fitting), the errors caused by fitting of the 4-external integrals are negative. In order to keep the latter error small, we found it necessary to use a larger fitting basis than for the 0-3 external integrals. This is due to the fact that in the 4-external

Table 5: CPU times for LCCSD calculations for $(\text{gly})_4$, cc-pVTZ basis[a], 706 basis functions, 96 correlated electrons

Step	LCCSD	DF-LCCSD[b,c]	DF-LCCSD[b,d]
Integral evaluation and transformation:			
0-2 external integrals	11280	496	328
3 external integrals	12370	838	1718
4 external integrals	33257	1420	1628
Total transformation	56907	2754	3674
Times per iteration:			
Operator **G**(**E**)	1570	140	100
Contractions with 3-external integrals	30	531	30
Contractions with 4-external integrals	40	1233	40
Residual	52	52	52
Total time per iteration	1692	1956	221
Total time (12 iter.)	76002	26433	6567

a) CPU-times in seconds on AMD Opteron 2.0 GHZ.

b) cc-pVTZ/MP2 fitting basis [52] for the 0-3 external integrals;
cc-pVQZ/MP2 fitting basis [52] for the 4-external integrals.

c) Using the full fitting basis for the 0-2 external integrals.
The 3-index integrals or fitting coefficients are stored on disk,
and the 3,4-external 4-index integrals are assembled in each iteration.

d) Using local fitting domains for 0-2 and 4 external integrals and **G**(**E**).
The time for computing the 4-external integrals without local fitting domains is 2615 seconds.
Robust fitting with domains extended by one shell of neighboring atoms (see text)
is used for the 0-2 external exchange integrals.
All 4-index integrals are precomputed and stored on disk.

case only products of two PAOs are fitted, and this requires fitting functions with higher angular momenta than for the other integral types, in which at least one occupied orbital is involved.

In LCCSD larger domains are needed for the transformed integrals than in DF-LMP2, and therefore also larger fitting domains are required if local fitting is performed. It turns out, however, that the fitting domain sizes can be much reduced if robust fitting is performed (cf. section 4.2). Table 6 shows the errors of LMP2 and LCCSD correlation energies caused by local fitting of the 0-2 external integrals, as compared to a calculation with the full fitting basis. The local fitting domains for each orbital i include all fitting functions at the atoms belonging to the standard orbital domain $[i]$; in addition, this domain was extended by the functions at 1 or 2 shells of neighboring atoms (denoted ”Ext.” in the Table). It can be seen that with an average number of only 334 fitting functions per orbital the error amounts only to 0.06 mH with robust fitting. Using the asymmetric fitting procedure without robust

Table 6: Effect of robust local fitting on LMP2 and LCCSD correlation energies for (gly)$_4$, cc-pVTZ basis^a

Fitting	Ext.^{b}	$N_{\mathrm{fit}}^{\mathrm{av}}$	E_{corr}^{LMP2}	$\Delta E_{corr}^{\mathrm{LMP2}}$	$E_{corr}^{\mathrm{LCCSD}}$	$\Delta E_{corr}^{\mathrm{LCCSD}}$
none			-3.219946	-0.000224	-3.298234	0.000067
non-local		1797	-3.219723	0.0	-3.298301	0.0
asymmetric	1	334	-3.216307	0.003416	-3.295158	0.003143
asymmetric	2	565	-3.219358	0.000364	-3.297971	0.000329
robust	1	334	-3.219655	0.000068	-3.298251	0.000050
robust	2	565	-3.219707	0.000016	-3.298297	0.000004

a) cc-pVTZ/MP2 fitting basis [52] for the 0-3 external integrals; cc-pVQZ/MP2 fitting cc-pVQZ/MP2 fitting basis [52] for the 4-external integrals.
ΔE_{corr} is the energy difference to the density fitted result with the full fitting basis.
b) domain extension for fitting basis, see text.

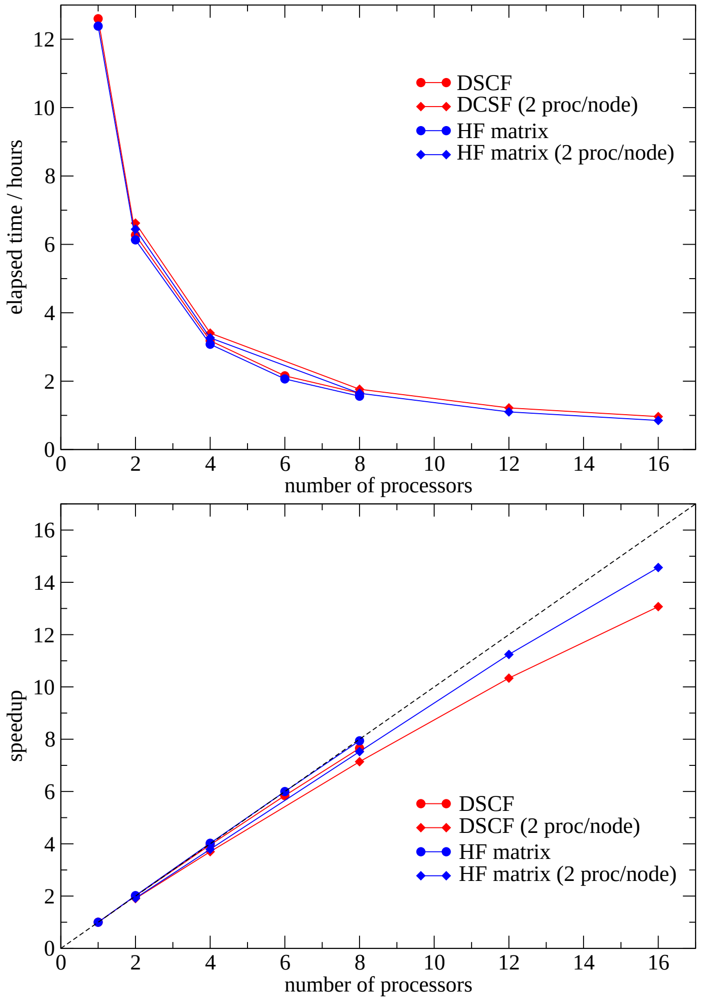
fitting as outlined in section 4.2, the error is more than 50 times larger. It can also be seen that the errors are of very similar size for LMP2 and LCCSD; thus, despite the fact that robust fitting is not used for the 2-external Coulomb integrals $(rs|ij)$, it appears that the local fitting does not introduce extra errors in the LCCSD. The extra effort for the robust fitting is by far overcompensated by the reduction of the fitting domains. More details of our method will be presented elsewhere [66].

5 Parallelization

As part of the HPC-Chem project a number of programs in MOLPRO were parallelized or the previously existing parallelization was improved. Newly parallelized were the integral-direct local transformations, the LMP2 and CCSD programs, and the density fitting Hartree-Fock program. Additional work was done on the MCSCF and MRCI programs. The infrastructure was extended and generalised to support different communication libraries (TCGMSG, MPI) and network protocols (Myrinet, TCP-IP).

The parallelization in MOLPRO is based on the Global Array (GA) Software developed at Pacific Northwest Laboratories (see www.emsl.pnl.gov/docs/global/). The GA library provides facilities to create, write, and read GAs, which are distributed over the compute nodes and can be used as a shared memory device. Access is one-sided, i.e., each processor can asynchronously read or write data from/to parts of the GAs which are located on remote nodes. The GA software allows to use distributed and shared memory machines in the same way, but of course it has to be taken into account in the algorithms that accessing data via the network is slower than from local or shared memory. In order to minimize communication, a symmetric coarse grain-parallelization model is used, and

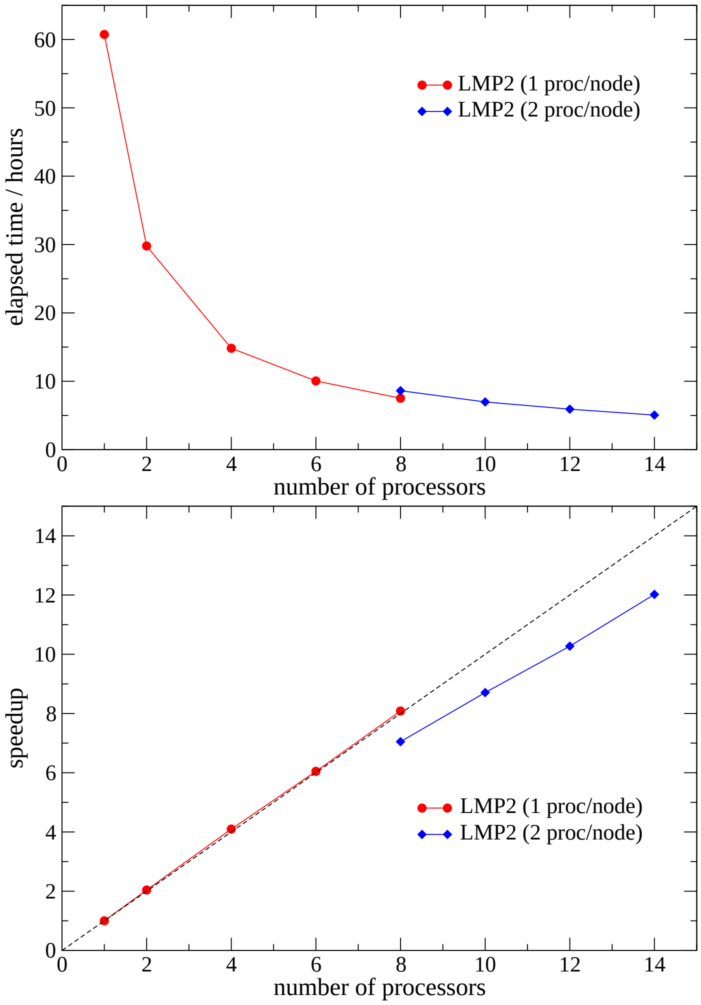
Figure 9: Elapsed times (upper pannel) and speedups (lower panel) of direct Hartree-Fock calculations for progesterol using the cc-pVTZ basis set on a PC cluster (PIII/933 MHz, Myrinet)



data are replicated on all nodes if sufficient memory is available. The tasks are allocated to processors dynamically using a shared counter.

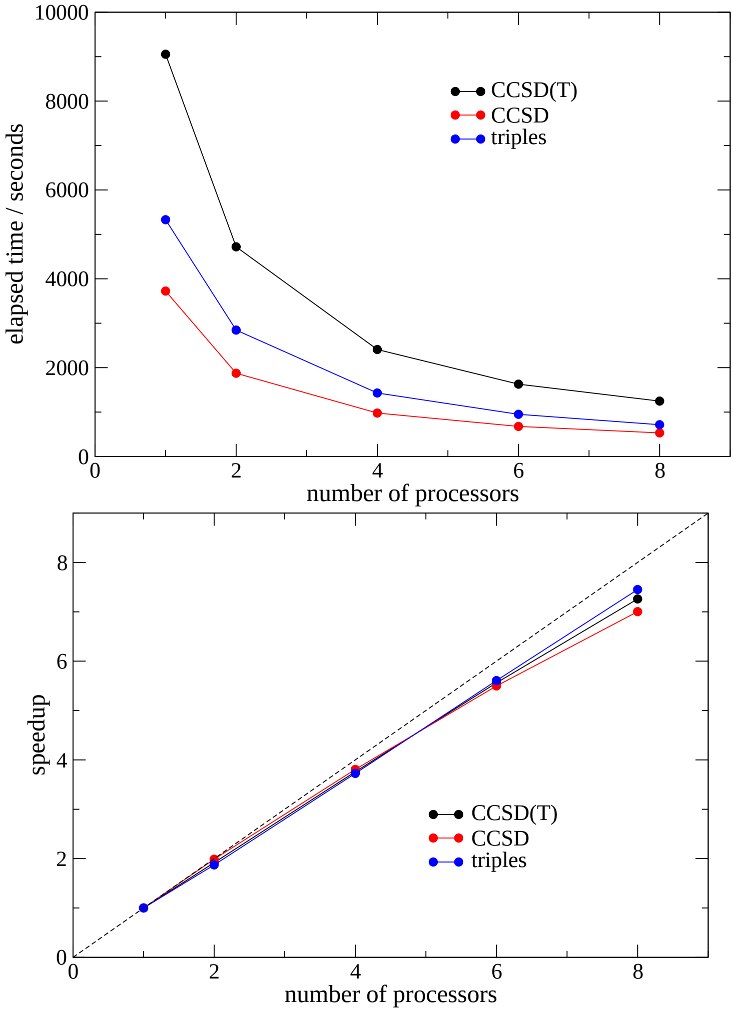
Some programs, as direct Fock-matrix evaluation, are rather easy to parallelize, since the amount of data to be communicated is minimal. However, this is not the case in the integral transformations or in the CCSD program. The transformations involve sorting steps, and in the CCSD the amplitudes are combined with all transformed integrals in a non-sequential way. Therefore, the communication requirements in these programs are much larger than in HF or DFT. Furthermore, many data must be stored on disk and nevertheless be accessible from all CPUs. In order to minimize the communication and I/O overheads, MOLPRO uses various different file types: (i) exclusive access files (EAF), which are local to a particular processor and can only be read from the processor to which they belong; (ii) shared files (SF), which can be read from all processors. However, all processors must be synchronized

Figure 10: Elapsed times (upper panel) and speedups (lower pannel) of LMP2 calculations for progesterol using the cc-pVTZ basis set on a PC cluster (PIII/933 MHz, Myrinet)



when I/O takes place, and thus the processors cannot perform the I/O independently. In order to eliminate the latter restriction, global files (GF) were implemented by the Jülich group as part of the HPC-Chem project. Global files behave like GAs but reside on disk. One-sided access from any processor is possible. It is also possible to distribute the file with a predefined fragmentation over the processors, which can be used in the algorithms to further reduce the communication. Finally, files can also be mapped to GAs so that all data reside in the distributed memory. This can be useful if a large amount of memory is available and I/O is a bottleneck. A typical usage of EAF files is the storage of the 2-electron integrals in conventional calculations. The integrals are evenly distributed over all nodes, and each node processes its own subset. On the other hand, shared or global files are used to store the amplitudes and transformed integrals, since these must often be accessed in a random manner.

Figure 11: Elapsed times (upper panel) and speedups (lower panel) of CCSD calculations for 1-butene (C_1 symmetry, 168 basis functions) on a PC cluster (PIII/933 MHz, Myrinet)



Figures 9 and 10 shows timings of direct HF and LMP2 calculations, respectively, on the PC cluster funded by the HPC-Chem project. This cluster contains 8 dual processor nodes. If only one processor per node is used, the speedup is almost linear. Some degradation is observed if 2 processors per node are used, since the memory bandwidth is insufficient for two processors. Furthermore, some global array operations, like global summations or broadcasting, are performed via one CPU (usually CPU 0), and this leads to a communication bottleneck in this node if many CPUs are used.

Figure 11 shows timings for conventional CCSD(T) calculations as function of the number of processors. In this case significantly more communication is needed than in the direct LMP2 calculations. Nevertheless, the speedup is very satisfactory as long as only one processor per node is used. The performance gets much worse if two processors per node are used (not shown). With 16 processors, the speedup is only about 10. To a large extent this is due to poor memory bandwidth of this machine. This can also be seen from the facts that (i) running two similar jobs on one node increases the CPU time of each calculation by a factor of 1.5-1.8, and (ii), running a parallel job using 2 processors on a single node leads hardly to any speedup, despite the fact that shared memory is used and no inter-node communication is required. Thus, for memory intensive applications like CCSD(T) it is not advantageous to use dual CPU compute nodes. It should be noted, however, that with more recent hardware (e.g. AMD Opteron) this memory bottleneck is not observed any more, since in these machines each CPU has independent memory access. Unfortunately, at the time of writing this report, no detailed benchmark results for such machines are available.

6 Conclusions

The work reviewed in this article has shown that local correlation methods combined with density fitting approximations have extended the applicability of high-level *ab initio* methods to much larger molecular systems than could previously be treated. Local approximations lead to linear scaling of the computational cost as a function of the molecular size for all popular single-reference electron correlation methods like MP2-MP4, QCISD(T), and CCSD(T). While the absolute cost of the original linear scaling methods was still relatively large if good basis sets were used, density fitting approximations have made it possible to reduce the computer times by additional 1-2 orders of magnitude. This applies in particular to the integral transformations, which constituted the main bottleneck. The speedup of the correlation treatments has lead to the situation that LMP2 calculations with density fitting took only a small fraction of the time needed for the preceding direct Hartree-Fock calculation. Therefore, additional work has been devoted to speed up the Fock matrix construction. It has been shown that by localising the orbitals in each iteration and applying local density fitting approximations a speedup of 1-2 orders of magnitude (depending on the basis set quality) can be achieved. The new methods have also been parallelized, which further reduces the elapsed times. All methods described in this work have been implemented in the MOLPRO package of *ab initio* programs and will be made available to the users of this software in the near future.

Acknowledgements

The authors like to thank Fred Manby and Peter Knowles for their collaboration and support of this work. In particular, the density fitting methods described in this article are based on their integral programs.

Bibliography

[1] M. Schütz, R. Lindh, and H.-J. Werner, *Mol. Phys.* **96**, 719 (1999).

[2] P. Pulay, *Chem. Phys. Lett.* **100**, 151 (1983).

[3] S. Saebø and P. Pulay, *Chem. Phys. Lett.* **113**, 13 (1985).

[4] P. Pulay and S. Saebø, *Theor. Chim. Acta* **69**, 357 (1986).

[5] S. Saebø and P. Pulay, *J. Chem. Phys.* **115**, 3975 (2001).

[6] C. Hampel and H.-J. Werner, *J. Chem. Phys.* **104**, 6286 (1996).

[7] M. Schütz, G. Hetzer, and H.-J. Werner, *J. Chem. Phys.* **111**, 5691 (1999).

[8] G. Hetzer, M. Schütz, H. Stoll, and H.-J. Werner, *J. Chem. Phys.* **113**, 9443 (2000).

[9] M. Schütz and H.-J. Werner, *Chem. Phys. Lett.* **318**, 370 (2000).

[10] M. Schütz, *J. Chem. Phys.* **113**, 9986 (2000).

[11] M. Schütz and H.-J. Werner, *J. Chem. Phys.* **114**, 661 (2001).

[12] M. Schütz, *J. Chem. Phys.* **113**, 8772 (2002).

[13] M. Schütz, *Phys. Chem. Chem. Phys.* **4**, 3941 (2002).

[14] H.-J. Werner, P. J. Knowles, R. Lindh, M. Schütz, P. Celani, T. Korona, F. R. Manby, G. Rauhut, R. D. Amos, A. Bernhardsson, A. Berning, D. L. Cooper, M. J. O. Deegan, A. J. Dobbyn, F. Eckert, C. Hampel, G. Hetzer, A. W. Lloyd, S. J. McNicholas, W. Meyer, M. E. Mura, A. Nicklass, P. Palmieri, R. Pitzer, U. Schumann, H. Stoll, A. J. Stone, R. Tarroni, and T. Thorsteinsson, MOLPRO, version 2002.6, a package of *ab initio* programs, 2003, see `http://www.molpro.net`.

[15] W. Meyer, *Int. J. Quantum Chem. Symp.* **5**, 341 (1971).

[16] W. Meyer, *J. Chem. Phys.* **58**, 1017 (1973).

[17] W. Meyer, *J. Chem. Phys.* **64**, 2901 (1976).

[18] H.-J. Werner and W. Meyer, *J. Chem. Phys.* **73**, 2342 (1980).

[19] H.-J. Werner and W. Meyer, *J. Chem. Phys.* **74**, 5794 (1981).

[20] H.-J. Werner and E. A. Reinsch, *J. Chem. Phys.* **76**, 3144 (1982).

[21] H.-J. Werner and P. J. Knowles, *J. Chem. Phys.* **82**, 5053 (1985).

[22] P. J. Knowles and H.-J. Werner, *Chem. Phys. Lett.* **115**, 259 (1985).

[23] H.-J. Werner and P. J. Knowles, *J. Chem. Phys.* **89**, 5803 (1988).

[24] P. J. Knowles and H.-J. Werner, *Chem. Phys. Lett.* **145**, 514 (1988).

[25] H.-J. Werner and P. J. Knowles, *Theor. Chim. Acta* **78**, 175 (1990).

[26] P. J. Knowles and H.-J. Werner, *Theor. Chim. Acta* **84**, 95 (1992).

[27] C. Hampel, K. A. Peterson, and H.-J. Werner, *Chem. Phys. Lett.* **190**, 1 (1992).

[28] P. J. Knowles, C. Hampel, and H.-J. Werner, *J. Chem. Phys.* **99**, 5219 (1993).

[29] P. J. Knowles, C. Hampel, and H.-J. Werner, *J. Chem. Phys.* **112**, 3106 (2000).

[30] M. J. O. Deegan and P. J. Knowles, *Chem. Phys. Lett.* **227**, 321 (1994).

[31] H.-J. Werner, *Mol. Phys.* **89**, 645 (1996).

[32] P. Celani and H.-J. Werner, *J. Chem. Phys.* **112**, 5546 (2000).

[33] T. Busch, A. D. Esposti, and H.-J. Werner, *J. Chem. Phys.* **94**, 6708 (1991).

[34] A. ElAzhary, G. Rauhut, P. Pulay, and H.-J. Werner, *J. Chem. Phys.* **108**, 5185 (1998).

[35] M. Schütz, H.-J. Werner, R. Lindh, and F. R. Manby, *J. Chem. Phys.* **121**, 737 (2004).

[36] P. Celani and H.-J. Werner, *J. Chem. Phys.* **119**, 5044 (2003).

[37] G. Rauhut and H.-J. Werner, *Phys. Chem. Chem. Phys.* **3**, 4853 (2001).

[38] H.-J. Werner, F. R. Manby, and P. Knowles, *J. Chem. Phys.* **118**, 8149 (2003).

[39] G. Hetzer, P. Pulay, and H.-J. Werner, *Chem. Phys. Lett.* **290**, 143 (1998).

[40] J. W. Boughton and P. Pulay, *J. Comput. Chem.* **14**, 736 (1993).

[41] P. Pulay and S. Saebø, *Theor. Chim. Acta* **69**, 357 (1986).

[42] H.-J. Werner and F. R. Manby, *to be published* (2004).

[43] S. F. Boys and I. Shavitt, *University of Wisconsin, Report WIS-AF-13* (1959).

[44] E. J. Baerends, D. E. Ellis, and P. Ros, *Chem. Phys.* **2**, 41 (1973).

[45] J. L. Whitten, *J. Chem. Phys.* **58**, 4496 (1973).

[46] B. I. Dunlap, J. W. D. Connolly, and J. R. Sabin, *J. Chem. Phys.* **71**, 3396 (1979).

[47] B. I. Dunlap, J. W. D. Connolly, and J. R. Sabin, *J. Chem. Phys.* **71**, 4993 (1979).

[48] O. Vahtras, J. Almlöf, and M. W. Feyereisen, *Chem. Phys. Lett.* **213**, 514 (1993).

[49] F. Weigend, A. Köhn, and C. Hättig, *J. Chem. Phys.* **116**, 3175 (2002).

[50] K. Eichkorn, F. Häser, O. Treutler, and R. Ahlrichs, *Theor. Chim. Acta* **97**, 119 (1997).

[51] F. Weigend, *Phys. Chem. Chem. Phys.* **4**, 4285 (2002).

[52] F. Weigend, M. Häser, H. Patzelt, and R. Ahlrichs, *Chem. Phys. Lett.* **294**, 143 (1998).

[53] J. W. Mintmire and B. I. Dunlap, *Chem. Phys. Lett.* **25**, 88 (1982).

[54] B. I. Dunlap, *J. Chem. Phys.* **78**, 3140 (1983).

[55] F. R. Manby and P. J. Knowles, *Phys. Rev. Lett.* **87**, 163001 (2001).

[56] F. R. Manby, P. J. Knowles, and A. W. Lloyd, *J. Chem. Phys.* **115**, 9144 (2001).

[57] H.-J. W. R. Polly, F. R. Manby, and P. Knowles, *Mol. Phys.*, (2004), in press.

[58] M. W. Feyereisen, G. Fitzgerald, and A. Komornicki, *Chem. Phys. Lett.* **208**, 359 (1993).

[59] B. I. Dunlap, *Phys. Chem. Chem. Phys.* **2**, 2113 (2000).

[60] F. Weigend and M. Häser, *Theor. Chim. Acta* **97**, 331 (1997).

[61] M. Seitz, S. Stempfhuber, M. Zabel, M. Schütz, and O. Reiser, *Angew. Chem.*, (2004), in press.

[62] A. Schäfer, C. Huber, and R. Ahlrichs, *J. Chem. Phys.* **100**, 5829 (1994).

[63] F. Weigend, M. Häser, H. Patzelt, and R. Ahlrichs, *Chem. Phys. Lett.* **294**, 14 (1998).

[64] M. Dolg, U. Wedig, H. Stoll, and H. Preuss, *J. Chem. Phys.* **86**, 866 (1987).

[65] M. Schütz and F. R. Manby, *Phys. Chem. Chem. Phys.* **5**, 3349 (2003).

[66] H.-J. Werner and M. Schütz, *to be published* (2004).

Parallel DFT in TURBOMOLE Linear Algebra

Thomas Müller

John von Neumann Institute for Computing
Central Institute for Applied Mathematics
Research Centre Jülich
52425 Jülich, Germany
E-mail: th.mueller@fz-juelich.de

1 Introduction

The TURBOMOLE quantum chemistry program package in its current version allows for the treatment of electronic ground states at the Hartree-Fock (HF), Density Functional Theory (DFT), second-order Møller-Plesset perturbation theory (MP2) and Coupled Cluster Theory (CC2) level of theory including ground state properties (structure constants, vibrational frequencies, NMR chemical shifts). Further included are methods for the treatment of electronic excited states by linear response techniques at DFT, HF and CC2 level of theory. Specific to TURBOMOLE is the emphasis on integral-direct implementations of the available methods, combined with fast integral evaluation (RI-J and MARI-J techniques) and the exploitation of the full finite point group symmetry.

This part of the HPC-Chem project aims at further extending its applicability to very large systems by means of parallelization. In view of the fast methodological development - as exemplified by the other contributions to this report - parallelization efforts should separate the infrastructure required for parallel optimization from the actual code of the quantum chemical (QC) methods and supply only a (limited) set of library routines supporting maintenance, parallelization or re-parallelization of existing code with little effort. Discarding the master-slave concept greatly simplifies parallelization while minimizing the differences between serial and parallel QC codes. Finally, machine-independence is advantageous in view of the short life cycles of current hardware. Anticipating the more detailed discussion of efficiency, functionality and performance of the serial code in chapter I it is evident that (easily parallelizable) integral evaluation is of diminishing importance. This does not exactly facilitate parallelization. Moreover, parallel code directly derived from a serial implementation usually does not simply scale to arbitrary problem sizes: memory requirements per processor might be excessive and switching to distributed data might not be trivial and collide with parallel efficiency.

Linear algebra routines have to be replaced in many cases by parallel versions because either the size of the matrices enforces switching to distributed data or cubic scaling requires parallelization. Specific cases may force the replacement by alternative algorithms with improved performance either due to better parallel scalability or more favorable cache optimization.

For parallel computer systems I/O poses a potentially serious problem and should - whenever possible - be completely avoided. As (distributed) memory availability scales linearly with the number of processors, shortages in distributed memory are likely to be alleviatable. This does not appear to be the case now or in future for secondary storage media. In addition storage requirements can be reduced by data compression.

The exploitation of symmetry largely reduces the computational cost (integral evaluation is sped up by approximately the order of the point group n_g, some linear algebra tasks by n_g^2 and memory demand is down by a factor of n_g) at the expense of somewhat more complicated load-balancing and data access. A key ingredient for good performance is systematic, dense data access which can be taken advantage of by the communication routines. TURBOMOLE very efficiently implements symmetry for integral evaluation and data storage. An exception comprises the symmetry redundant storage of the Fock and density matrices in CAO basis.

In the subsequent sections, these aspects are discussed along with reasonable solutions in some detail. For a brief introduction to the RI-J and MARI-J methods refer to sections I.3 and I.4, respectively.

2 General considerations

This article primarily deals with the parallelization of the RIDFT and RDGRAD modules required for structure optimization at DFT level with non-hybrid functionals using the RI method (cf. section I.3). With few exceptions all details also apply to the DSCF and GRAD modules which furnish the basis for geometry optimization at DFT and HF level of theory and share most computational steps with their RI counterparts. The latter handle additionally the auxiliary basis set, the associated integrals, and compute Coulomb contributions to the Kohn-Sham matrix differently.

TURBOMOLE is a collection of independent specialized modules which communicate via files. The input is set up interactively by the module DEFINE and the calculation is carried out by means of the JOBEX script, which executes the individual modules in the correct order (Figure 1).

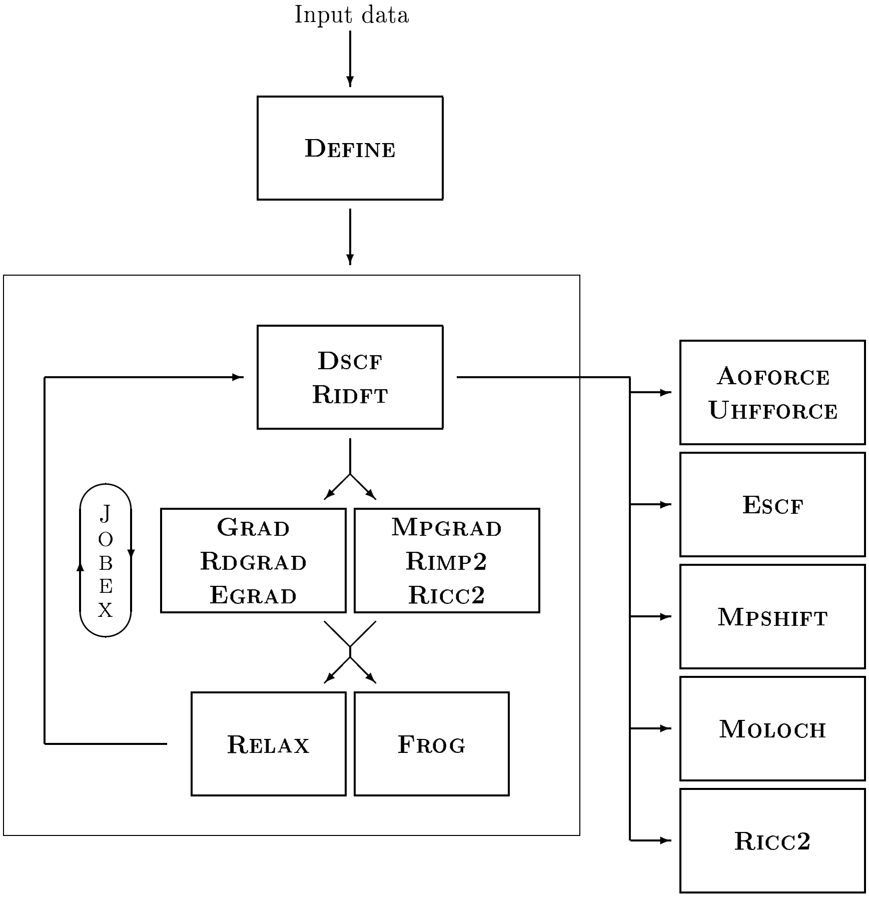


Figure 1: Schematic flow chart of the TURBOMOLE package (for details see text)

The master-slave concept is discarded and dynamic (bag of tasks) and static load-balancing based on the data distribution is implemented. Dynamic load-balancing is mainly used for integral evaluation since the computational costs associated with the integral batches cannot be accurately estimated. Dynamic load-balancing is most efficient in combination with the replicated data approach, where the communication load is independent of the number of tasks. However, memory limitations force dynamic load-balancing with globally distributed data imposing additional constraints to reduce the communication overhead. Either fully encapsulated separate parallelized tasks (such as linear algebra) are used or parallelization is restricted to the highest possible level in order to avoid instabilities due to excessive code changes and to retain a maximum overlap between serial and parallel codes. Despite its computational simplicity the RI based DFT does not have one dominantly CPU time consuming subtask, so that parallelization requires more effort in order to achieve scalability. All timings refer to the Jülich multiprocessor system, a cluster of IBM eServer p690 with 32 Power4 CPUs (1.7 GHz) per SMP node connected with a high-speed network. Reproducibility of performance data is limited on SMP systems to some ±5-10% even for serial codes such as matrix multiply. Since computations have not been carried out with exclusive access, performance depends on other jobs running simultaneously.

3 Communication libraries

The parallel implementation of TURBOMOLE primarily utilizes several public domain communication and mathematical libraries complemented by a set of special-purpose routines.

The **Global Array toolkit** [1] provides distributed multi-dimensional arrays along with *one-sided* transparent access to the distributed data, i.e. there is no need for cooperative communication calls between the individual processes (pair-wise send and receive calls). This toolkit is of particular use for dynamic load-balancing avoiding the master-slave concept. This feature is not yet available with the current MPI-1 standard, while vector-specific implementations may provide some features of the future MPI-2 standard. Other features are easy control of the data distribution over the processes, the ease of data access and the provision for taking advantage of data locality by the user's code. The toolkit's communication library is interfaced to MPI, specific network protocols (quadrinet, myrinet) as well as to the mixed usage of shared memory and MPI (similar to TURBOMPI [2]) and runs on a variety of machines.

BLACS [3] is the basic communication library usually implemented on top of MPI used by the parallel linear algebra package ScaLAPACK. It is not very useful with quantum chemical program packages as the usage is tedious and does not offer much advantage over the direct use of MPI, here.

Compared to the GA toolkit the widely used **MPI-1** standard lacks the one-sided access to the distributed data forcing the master-slave concept or static load-balancing. The lack of more complex data structures and the tedious implementation of the basic library utilities makes the ready-to-use infrastructure available with the GA toolkit preferable. However, the possibility to create process subgroups and carry out *several parallel tasks simultaneously* makes a limited use of MPI based communication valuable. Additionally, certain data redistributions are very efficient with MPI.

The **ScaLAPACK** [5] and **PBLAS** [4] implementations of parallel linear algebra are based on the BLACS communication library. The block-cyclic distribution of one- and two-dimensional matrices is - however - not only extremely inconvenient but also incompatible with the algorithms used in quantum chemistry. Since they appear to be very favorable in the realm of linear algebra and regarding the large number of parallel routines available with ScaLAPACK and PBLAS, a set of MPI based routines interface the BLACS based block-cyclic distribution of matrices and the corresponding dense GA data structures.

The **interface library** comprises a set of special routines and utilities for a variety of tasks occurring in the context of parallelization. It is also designed to be implemented in a serial and parallel version so that serial and parallel code are as close as possible (somewhat in the spirit of OpenMP, though not restricted to SMP architectures) and simplify code maintenance and development. It also includes extensions for simple data-parallel algorithms and for shared memory usage on clusters of SMP nodes.

4 Data structures

As in other codes, TURBOMOLE stores two-dimensional square matrices as vectors with column-major labeling (FORTRAN notation, Figure 2a). For symmetric matrices only the upper half matrix is stored (Figure 2b).

1	9	17	25	33	41	49	57
2	10	18	26	34	42	50	58
3	11	19	27	35	43	51	59
4	12	20	28	36	44	52	60
5	13	21	29	37	45	53	61
6	14	22	30	38	46	54	62
7	15	23	31	39	47	55	63
8	16	24	32	40	48	56	64

Figure 2a

1	2	4	7	11	16	22	29
	3	5	8	12	17	23	30
		6	9	13	18	24	31
			10	14	19	25	32
				15	20	26	33
					21	27	34
						28	35
							36

Figure 2b

Figure 2: Storage of non-symmetric (Figure 2a) and symmetric (Figure 2b) two-dimensional square matrices as vectors. The numbers are the vector element indices.

In the presence of non-trivial symmetry, block-diagonal matrices occur with elements in the off-diagonal blocks vanishing. Each diagonal block is stored as a vector and the vectors of subsequent diagonal blocks are concatenated (Figure 3).

These arrays are stored in distributed manner as vectors or as full two-dimensional matrices depending on the access pattern of the algorithm. The two-dimensional matrix may be distributed over the processors by rows, columns or blocks (Figure 4).

The global data structures are accessed (i) by retrieving a copy of any part of the distributed data and (ii) by direct pointer-access to the local portion of the distributed data that a given process owns.

1	6	11	16	21			
2	7	12	17	22			
3	8	13	18	23			
4	9	14	19	24			
5	10	15	20	25			
					26	29	32
					27	30	33
					28	31	34

Figure 3a

1	2	4	7	11			
	3	5	8	12			
		6	9	13			
			10	14			
				15			
					16	17	19
						18	20
							21

Figure 3b

Figure 3: Storage of two-dimensional block-diagonal non-symmetric (Figure 3a) and symmetric (Figure 3b) matrices as one-dimensional vectors.

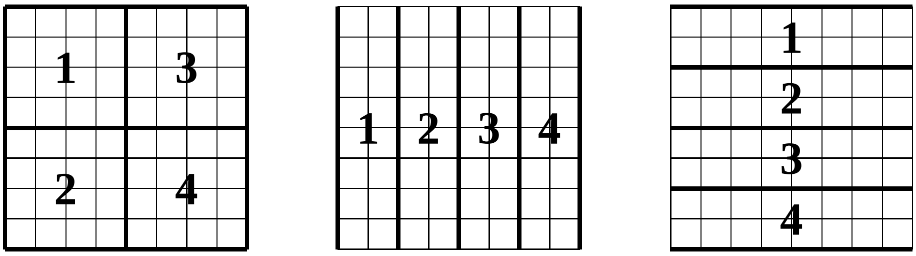


Figure 4a Figure 4b Figure 4c

Figure 4: Distribution of a two-dimensional array over 4 processes by blocks (Figure 4a), by columns (Figure 4b) and by rows (Figure 4c).

ScaLAPACK relies on block-cyclic (BC) data distributions of two-dimensional matrices. A process grid is constructed such that the product of process rows and process columns equals the total number of processes and the grid is as close to square shape as possible (Figure 5a). The elements of the initial matrix are grouped into subblocks (Figure 5b) with a typical size of 50. The optimum value depends upon the specific task. These subblocks are distributed in cyclic manner over process rows and process columns (Figure 5c). The resulting distribution guarantees that no process owns only a continuous part of the initial matrix thereby optimizing static load-balancing. The MPI based routines for conversion from/to BC data distributions introduce a negligible overhead compared to the time spent in the linear algebra routines.

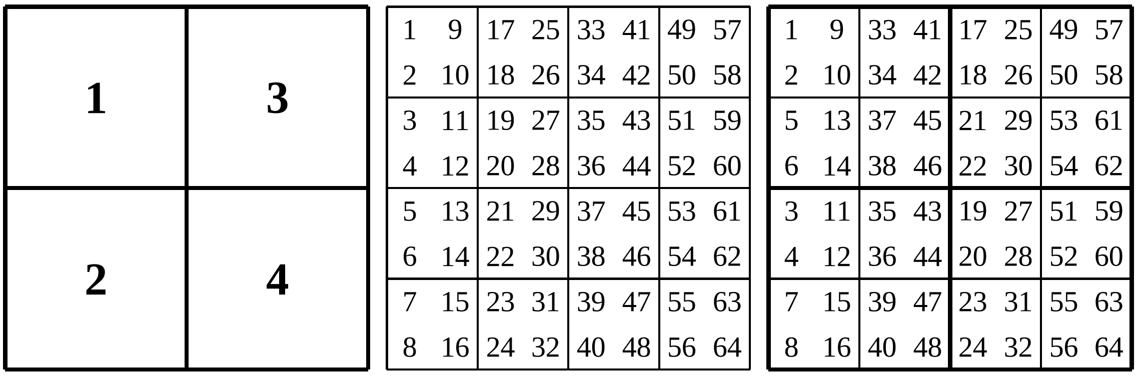


Figure 5a

1	9	17	25	33	41	49	57
2	10	18	26	34	42	50	58
3	11	19	27	35	43	51	59
4	12	20	28	36	44	52	60
5	13	21	29	37	45	53	61
6	14	22	30	38	46	54	62
7	15	23	31	39	47	55	63
8	16	24	32	40	48	56	64

Figure 5b

1	9	33	41	17	25	49	57
2	10	34	42	18	26	50	58
5	13	37	45	21	29	53	61
6	14	38	46	22	30	54	62
3	11	35	43	19	27	51	59
4	12	36	44	20	28	52	60
7	15	39	47	23	31	55	63
8	16	40	48	24	32	56	64

Figure 5c

Figure 5: Size of the local matrix associated with each process (Figure 5a), subblock formation (Figure 5b) and block-cyclic distribution of the subblocks (Figure 5c). The numbers indicate the global consecutive numbering of the matrix as in Figure 2.

5 Parallel linear algebra

A number of simple algebraic operations on distributed data (e.g. scalar products of vectors, traces of matrices or matrix-matrix products, etc.) are embarrassingly parallel, scale ideally to an arbitrary large number of processes and require little or no interprocess communication. Point group symmetry does not impose any restrictions.

For other important operations (e.g. similarity transforms, standard eigenproblem solver, Cholesky decomposition) use is made of the ScaLAPACK parallel linear algebra package.

Point group symmetry gives rise to block-diagonal matrices (Figure 3) so that the individual blocks can be treated independently. The resulting saving factor amounts to approximately the order of the point group squared, for the serial code. Parallelization schemes include (i) executing each block in parallel on all processes, (ii) executing all blocks simultaneously serially, and (iii) executing all blocks simultaneously and in parallel. The speedup in scheme (i) is limited by the block dimension and degrades with increasing symmetry. Scheme (ii) favors high symmetry cases and is memory intensive. The implemented scheme (iii) uses multi-level parallelism by dividing the total number of processes into a number of subgroups and each of these subgroups is assigned one and possibly more blocks to operate on.

The three most important elementary linear algebra operations are Cholesky decomposition, similarity transform and the symmetric eigenvalue problem, each with cubic scaling. Figure 6 illustrates the approximate scaling that can be achieved for large problem sizes with no symmetry. Whereas Cholesky decomposition and similarity transform come close to ideal scaling this does not apply to the eigensolver.

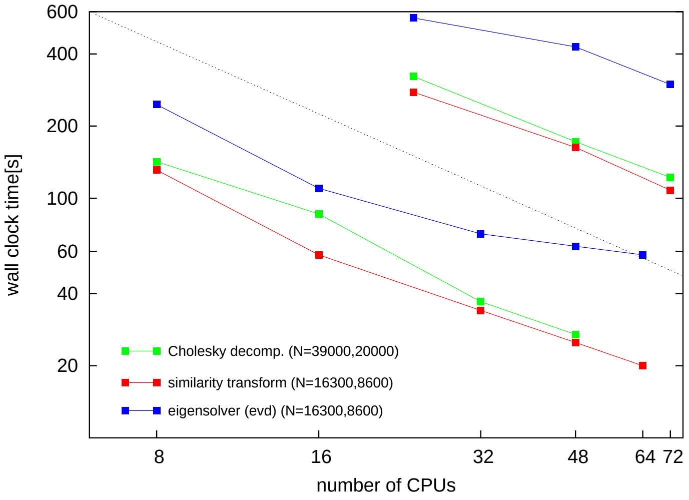


Figure 6: Parallel performance of important linear algebra operations (dotted line illustrates ideal scaling).

6 Utilities: parallel I/O and data compression

The Global Array toolkit has been extended by a Global I/O (GIO) library, which supports one-sided transparent access to data which are distributed over the local hard disks of the individual nodes of a PC cluster. This was primarily intended for supporting the MOLPRO package, where the large amount of data which are generated during calculations with parallel electron correlation methods cannot be kept in memory. Data access and control over the data distribution mimic the global arrays in memory: blocked distribution, program control over the data distribution, record based and non-continuous data access.

Each GIO file consists of several records. The size and the distribution of data in a record over the nodes can be fixed individually for each record. The access to data on remote hard disks is one-sided and requires no details about the distribution.

This library extension has been used for the parallelization of MOLPRO. The developers group of the Global Array toolkit has obtained the GIO extensions and may choose to incorporate it into the official release.

The data compression scheme [6] relies on the assumption of a constant absolute error so that numbers are stored only with that many bits that are necessary not to exceed the error bound. This results in a substantially better control over the errors compared to storing with a fixed reduced mantissa length. The compression factor increases with a decreasing numerical range of the input data. The initial idea to use compression for integral storage in memory, however, was abandoned because the integrals yield only a low compression factor of two to three. Moreover, the recent advances in the electronic structure codes make integral evaluation less and less critical, so that compression schemes may be rather applied to more suitable quantities such as the DIIS error matrices. In context of the DSCF code, the compression of the difference density matrices and difference Fock matrices might be a useful target.

7 The modules RIDFT and RDGRAD

In Figure 7 the flow charts for the RIDFT and RDGRAD modules are displayed. Whereas RIDFT involves the (iterative) wavefunction optimization step, RDGRAD computes the gradient for a previously optimized wavefunction. The steps indicated in the flow charts may become time-consuming and enforce parallelization subject to point group symmetry and size of the molecule. Additional minor steps are also parallelized as they operate on distributed data. Parallelization is driven by the demand for reduced wall clock timings and the accommodation of memory requirements. Subsequently, the individual steps are described in more detail as far as conceptional changes and parallelization are concerned.

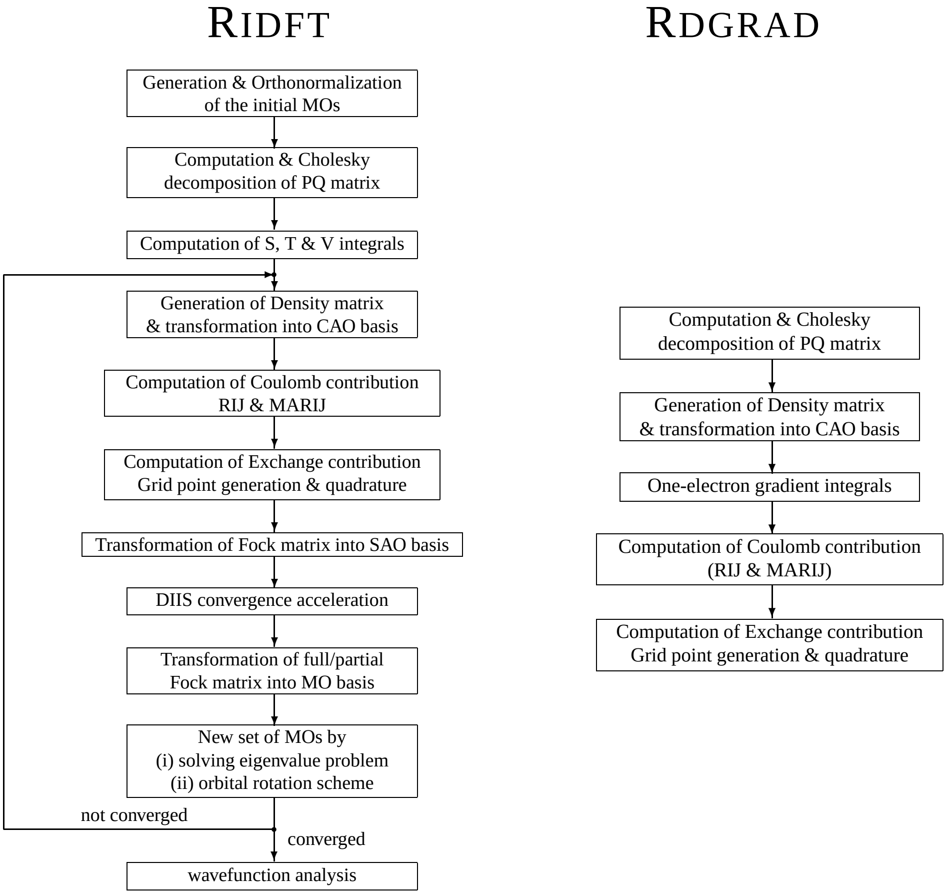
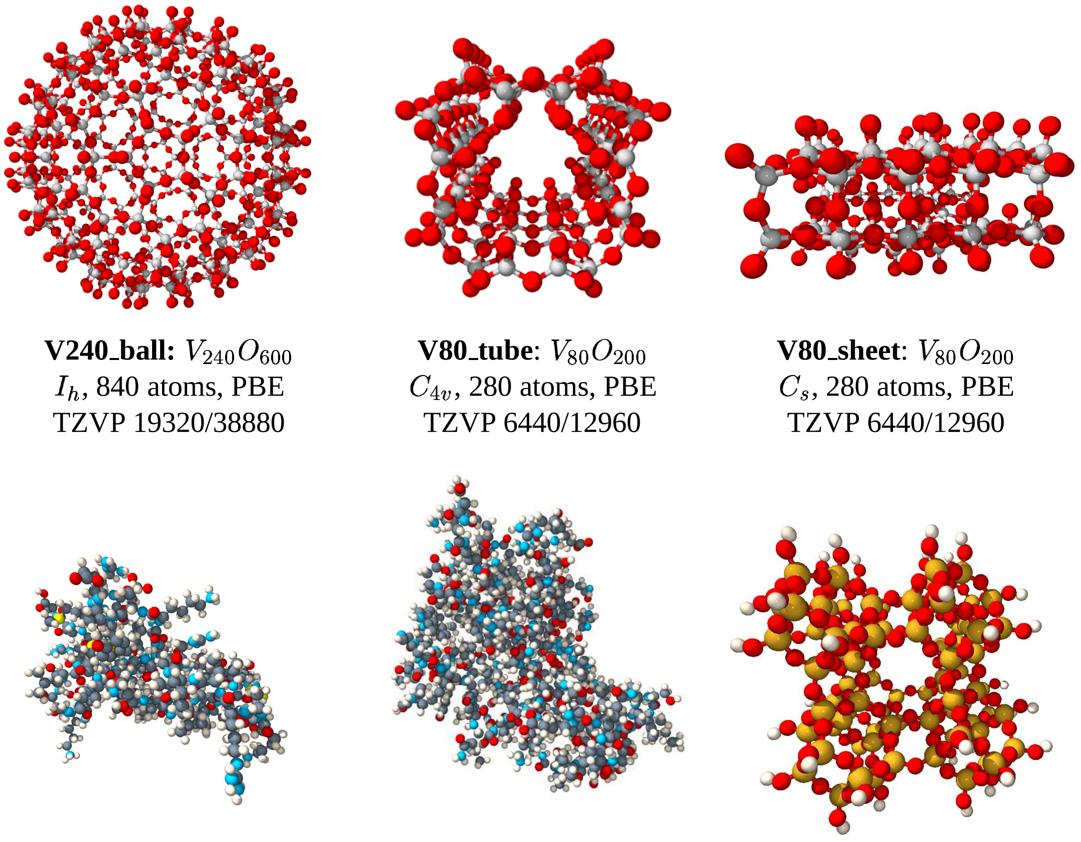

RIDFT
Generation & Orthonormalization of the initial MOs
Computation & Cholesky decomposition of PQ matrix
Computation of S, T & V integrals
Generation of Density matrix & transformation into CAO basis
Computation of Coulomb contribution RIJ & MARIJ
Computation of Exchange contribution Grid point generation & quadrature
Transformation of Fock matrix into SAO basis
DIIS convergence acceleration
Transformation of full/partial Fock matrix into MO basis
New set of MOs by (i) solving eigenvalue problem (ii) orbital rotation scheme
not converged
converged
wavefunction analysis
RDGRAD
Computation & Cholesky decomposition of PQ matrix
Generation of Density matrix & transformation into CAO basis
One-electron gradient integrals
Computation of Coulomb contribution (RIJ & MARIJ)
Computation of Exchange contribution Grid point generation & quadrature

Figure 7: Schematic flow chart of the RIDFT and RDGRAD modules.

The clusters and molecules chosen to demonstrate the current performance are collected in Figure 8: the vanadiumoxide clusters (V_{240_ball}, V_{80_sheet}, V_{80_tube}) illustrate the impact of symmetry; zeolites represent unsymmetric cage structures; small enzymes (BPTI, Barnase) cover aspects from biochemical applications.



V240_ball: $V_{240}O_{600}$
I_h, 840 atoms, PBE
TZVP 19320/38880

V80_tube: $V_{80}O_{200}$
C_{4v}, 280 atoms, PBE
TZVP 6440/12960

V80_sheet: $V_{80}O_{200}$
C_s, 280 atoms, PBE
TZVP 6440/12960

BPTI: $\mathrm{C_{284}H_{438}N_{84}O_{79}S_7}$
$\mathrm{C_1}$, 892 atoms, B-P86
SVP 8574/20323

Barnase: $\mathrm{C_{550}H_{891}N_{151}O_{168}}$
$\mathrm{C_1}$, 1710 atoms, B-P86
SVP 16371/38881

Zeolite: $Si_{96}O_{216}H_{48}$
C_1, 360 atoms, B-P86
SVP 4992/12216

Figure 8: Molecules and clusters used for evaluation purposes: brutto formula, point group symmetry, number of atoms, exchange-correlation functional, basis set, basis set size/auxiliary basis set size.

7.1 Generation and orthonormalization of the molecular orbitals

The iterative wavefunction optimization process begins by constructing a density matrix **D** from an initial set of occupied molecular orbitals (MOs) **C**:

$$\mathbf{D}_{\mu\nu} = \sum_{i=1}^{occ} 2\mathbf{C}_{\mu i}\mathbf{C}_{\nu i} \tag{1}$$

The Kohn-Sham (KS) matrix is formed by contracting the density with the integrals and adding the exchange-correlation contribution which is also a function of the density (cf. I.3). After application of convergence acceleration schemes (direct inversion of iterative subspace (DIIS), level shifting, damping) the eigenvalues and eigenvectors of the KS matrix are computed, which in turn serve to compute an improved density matrix and the optimization cycle is repeated until energy and/or density are converged. The initial MOs are either obtained from a previous calculation of the same system at a nearby geometry or by projection from a more approximate, faster Extended Hückel Theory (EHT) calculation. The subsequent formalism requires orthonormal MOs and it is advantageous for convergence control to have them spanning the full function space of the basis. In geometry optimizations the start MOs taken from a previous calculation at a nearby geometry must be re-orthonormalized. Traditionally this proceeds by Schmidt orthonormalization which does not parallelize and is rather slow.

The alternative is a Cholesky decomposition based scheme which is faster, scalable and shares the advantages of the traditional approach: (i) transform the overlap matrix **S** from atomic orbital (AO) into the MO basis using the current approximate MOs $\hat{\mathbf{C}}$, (ii) compute the Cholesky decomposition thereof, and (iii) multiply the approximate set of MOs by the inverse of **U** to the right. The orthonormality condition is expressed as

$$\mathbf{C}^T\mathbf{S}C = \mathbf{I} \tag{2}$$

where **I** denotes the unit matrix.

$$\begin{aligned} \hat{\mathbf{C}}^T\mathbf{S}\hat{\mathbf{C}} &= \mathbf{U}^T\mathbf{U} &(3)\\ (\hat{\mathbf{C}}\mathbf{U}^{-1})^T\mathbf{S}(\hat{\mathbf{C}}\mathbf{U}^{-1}) &= \mathbf{I} &(4) \end{aligned}$$

All steps are available with ScaLAPACK/PBLAS. This procedure also serves for intermediate re-orthonormalization of the MOs in order to reduce the accumulation of round-off errors. In fact, this scheme is quite similar to the re-orthonormalization routine already available in TURBOMOLE, which relies on perturbation theory and tolerates small deviations from orthonormality only. On the other hand, starting with the projected occupied orbitals from an EHT calculation a full orthonormal basis is desired without corrupting the EHT orbital guess. Supplementing the missing virtual MOs by random numbers, which serve as a non-linear dependent virtual orbital guess, the same procedure is applicable as well. Performance data are given in Table 1.

Strictly, the SCF scheme does not necessarily require to construct a full set of orthonormal MOs which resemble the canonical KS orbitals: the standard procedure for the general symmetric eigenvalue problem constructs an orthonormal basis by Cholesky decomposition of the overlap matrix in AO basis (cf. section II.5, [13]). However, experience indicates a less favorable convergence behavior.

cluster	point	MOs	wall clock time[s]						
	group[b]	(total/occc)	4_1^a	8_1^a	16_1^a	32_1^a	48_2^a	64_2^a	72_3^a
V240_ball	$I_h(120)$	19320/5160	-	-	0.6	0.4	0.2	0.1	-
V80_tube	$C_{4h}(8)$	6440/1720	6.4	2.1	1.2	0.7	-	-	-
V80_sheet	$C_s(2)$	6440/1720	49	19.0	11.3	6.2	5.1	-	-
Zeolite	$C_1(1)$	4992/1560	60	30	17.2	9.5	-	-	-
BPTI	$C_1(1)$	8574/1734	-	185	84	45	36	28.4	-
Barnase	$C_1(1)$	16371/3271	-	-	-	-	233	-	159

[a] N_n: N CPUs distributed symmetrically over n SMP nodes. [b] Order of point group in parentheses.

Table 1: Performance data for the Cholesky based orthonormalization procedure.

7.2 Computation and Cholesky decomposition of the PQ matrix

In the RI-J method the electron density is expanded in terms of an atom-centered auxiliary basis set. Its size is roughly twice the size of the basis set used for expansion of the electronic wave function from the outset and can reach dimensions beyond 40000. The elements of the PQ matrix contain the scalar products $< P|Q >$ defined as (cf. section I.3.3.2)

$$< P|Q > \; = \; \int P(r_1)Q(r_2)|r_1 - r_2|^{-1} d\tau \tag{5}$$

As only the totally symmetric component of the PQ matrix is required, symmetry reduces memory demands by $\approx n_g^2$ and computational effort by $\approx n_g^3$. Especially for low-symmetry cases it is important that the PQ matrix remains distributed throughout so that its size is no longer the limiting factor. The $O(N^2/n_g^2)$ serial evaluation of the PQ matrix elements is except for large processor numbers faster than the Cholesky decomposition, which is carried out with ScaLAPACK (for performance see Figure 6).

7.3 Evaluation of one-electron integrals

The one-electron integrals (overlap, kinetic energy, electron-nuclear attraction, and effective core potential; COSMO solvent integrals are excluded here as they have to be recalculated in each SCF iteration) are computed very much in line with the old master-slave implementation but using distributed memory, instead. The computational effort for evaluating these integrals ($O(N^2/n_g)$) is negligible compared to the previous setup step for the RI-J method.

7.4 Transformation of operators between different representations

Operators are represented by matrices which are frequently transformed between different basis sets (CAO, SAO, MO) in order to use the most favorable representation for a given task. Whereas symmetry for operators in SAO and MO representations gives rise to dense block-diagonal matrices, in CAO representation the matrix can be decomposed into groups of symmetry related scattered submatrices. Although it would be sufficient to store the symmetry independent submatrices, only, TURBOMOLE chooses to keep the full matrix and to compute the symmetry-nonredundant contributions. Symmetrization thereafter leads to the symmetry-correct representation:

$$\mathbf{O}' \;=\; \mathbf{C}^T\mathbf{O}\mathbf{C} \tag{6}$$

As the transformation and operator matrices (**C** and **O**, respectively) are dense block-diagonal for the SAO and MO representations the similarity transformation is of order $O(N^3/n_g^2)$ carried out by standard PBLAS routines. In fact the similarity transform is becoming more expensive than integral evaluation for Barnase ($N = 16371$), but scalibility and GFLOP rate is good (cf. Figure 6).

The transformation between SAO and CAO representation, however, involves very sparse transformation matrices **C** which contain at most n_g non-zero elements per row and column, respectively, disregarding the transformation between cartesian and real spherical harmonics basis functions. The sparse matrix-multiply has been adapted for the use of distributed data and optimized for better cache usage. The scalability is limited as the static load-balancing tied to the data structure is rather poor. For very large high symmetry cases (V240_ball) these operations are particularly problematic.

7.5 The Coulomb contribution to the Kohn-Sham matrix

The construction of the Coulomb part of the Kohn-Sham matrix follows the formulae given in section I.3 which is evaluated by parts: beginning with the contraction of the three-index integrals $I_{\mu\nu,\alpha}$ and the density matrix in CAO basis,

$$\gamma_\alpha^{CAO} \;=\; \sum_{\mu\nu} I_{\mu\nu,\alpha} D_{\mu\nu} \tag{7}$$

the resulting total symmetric component of the γ_α^{CAO} vector is transformed to SAO basis.

$$\gamma'^{SAO} \;=\; (PQ)^{-1}\gamma^{SAO} \tag{8}$$

The multiplication by the inverse PQ matrix follows the standard scheme avoiding its explict formation [13]: starting with the Cholesky decomposition of the PQ matrix **U**, two sets of linear equations are solved. This step is implemented serially on a distributed upper-triangular packed matrix **U** in order to minimize memory consumption. After backtransformation of γ'^{SAO} into CAO basis and contraction with the integrals the Coulomb contribution to the KS matrix is obtained.

$$J_{\mu\nu}^{CAO} \;=\; \sum_{\alpha} I_{\mu\nu,\alpha}\gamma'^{CAO} \tag{9}$$

For the multipole accelerated RI-J technique (MARI-J) integral evaluation is split into a near-field part using standard methods to evaluate $I_{\mu\nu,\alpha}$ and a far-field part which uses the multipole approximation to evaluate the integrals and contract them with $D_{\mu\nu}$ and γ'^{CAO}, respectively.

KS and density matrices are kept distributed by columns and rows, respectively. Provided tasks are defined over shell pairs (μ, ν) such, that large, non-overlapping, densely stored stripes of these matrices are accessed, only, the total communication load is almost independent of the number of processors and does not limit the scalability. This strategy implies minimizing the number of independent tasks while maintaining an acceptable load balance. This is achieved in a two-step procedure based on assembling tasks according to rough cost estimates for integral evaluation and obtaining more accurate timings during the first SCF iteration cycle which is used for re-optimization of the task definitions. Due to the constraint of minimizing the communication load perfect load-balancing is not possible here.

cluster	point	MOs			wall clock time[s]				
	group^b	(total)	4_1^a	8_1^a	16_1^a	32_1^a	48_2^a	64_2^a	72_3^a
V240_ball	I$_h$(120)	19320	-	-	6.9	6.1	6.0	6.2	-
V80_tube	C$_{4h}$(8)	6440	40	19.1	9.8	5.4	-	-	-
V80_sheet	C$_s$(2)	6440	172	81	44	20	16	-	-
Zeolite	C$_1$(1)	4992	107	48	27	13	-	-	-
BPTI	C$_1$(1)	8574	-	199	103	52	44	31	-
Barnase	C$_1$(1)	16371	-	-	-	-	126	-	98

[a] N_n: N CPUs distributed symmetrically over n SMP nodes. [b] Order of point group in parentheses.

Table 2: Construction of the Coulomb part of the KS matrix

Large high-symmetry cases (V240_ball) scale to a very limited number of processors, only, which is in view of the negligible computational effort of little importance. All of the examples yield a reasonable parallel scaling (Table 2).

7.6 The exchange contribution to the Kohn-Sham matrix

The evaluation of the exchange-correlation functional is carried out by numerical quadrature. The symmetry-nonredundant grid point generation and sorting of the grid points as to allow for fast evaluation and efficient contraction with (a small number of) density matrix elements contributes to the startup time. Both grid point generation and sorting are executed in parallel.

Although the evaluation of the exchange-correlation functional is approximately linear scaling, it suffers from two serious drawbacks affecting parallelization: (i) the reordering of patches of density and KS matrices permits fast quadrature but amounts essentially to random access to the original matrix elements rendering it incompatible with the notion of distributed data and - for large matrix dimensions - produces large numbers of cache misses. (ii) The computational effort per batch of grid points is - due to the data access - not reproducible, which rules out dynamic load-balancing with a small number of tasks as required for distributed data usage.

Since for large cases it is not possible to keep both density and KS matrices replicated, as reasonable compromise the density matrix is kept replicated once per SMP node reducing memory consumption while maintaining direct fast simultaneous access by all processes. The KS matrix is kept distributed with a local buffering mechanism for adding individual contributions. For efficient buffering, tasks are constituted by a large number of spatially close grid points. As communication overhead still may amount to 50% of the total wall clock time, load-balancing is far from optimum (Table 3). Some improvement can be expected from the direct use of MPI in the buffering mechanism, as it is better suited for the kind of data distributions occurring there. Still better, though dependent upon the topology of the molecule, is a reordering of the initial density and Fock matrices such that small dense patches of the matrices are accessed only with little redundancy.

cluster	point group[c]	MOs (total)	grid[b]	wall clock time[s] 4_1^a	8_1^a	16_1^a	32_1^a	48_2^a	64_2^a	72_3^a
			grid construction							
V240_ball	$I_h(120)$	19320	2	-	-	2.7	2.8	2.8	3.0	-
V80_tube	$C_{4h}(8)$	6440	2	4.5	2.6	1.6	1.3	-	-	-
V80_sheet	$C_s(2)$	6440	2	21	13	5.3	3.5	3.0	-	-
BPTI	$C_1(1)$	8574	2	-	151	65	38	31	24	-
BPTI	$C_1(1)$	8574	4	-	-	277	155	121	88	-
Barnase	$C_1(1)$	16371	2	-	-	-	-	119	-	85
Barnase	$C_1(1)$	16371	4	-	-	-	-	468	-	324
			quadrature							
V240_ball	$I_h(120)$	19320	2	-	-	12.0	9.6	9.8	9.8	-
V80_tube	$C_{4h}(8)$	6440	2	22	12	8	4.5	-	-	-
V80_sheet	$C_s(2)$	6440	2	82	42	20	12	10	-	-
BPTI	$C_1(1)$	8574	2	-	98	40	23	18	14	-
Barnase	$C_1(1)$	16371	2	-	-	-	-	43	-	31

[a] N_n: N CPUs distributed symmetrically over n SMP nodes. [b] Larger numbers indicate finer grid.

[c] Order of point group in parentheses.

Table 3: Construction of the exchange part of the KS matrix

7.7 DIIS convergence acceleration

The DIIS (direct inversion of iterative subspace) technique by Pulay [7, 8] is an efficient extrapolation procedure for the convergence acceleration of the SCF scheme. The most

CPU time intensive step involves the formation of the so-called error matrices **e**:

$$\mathbf{e} \;=\; \mathbf{FDS} - \mathbf{SDF} \tag{10}$$

The error and KS matrices of previous iterations up to the maximum subspace dimension (usually 4) are stored in distributed memory. Storage requirements can be reduced by compressing the error vectors, since with forthcoming convergence the entries are becoming vanishingly small. Specifically on PC clusters storing and retrieving data from disk may be worthwhile and the GIO extensions might prove useful. The computational effort is somewhat higher than for the similarity transform with quite similar scaling behavior (cf. Figure 6). For symmetric problems execution time rapidly vanishes due to the $O(N^3/n_g^2)$ dependence.

7.8 Wavefunction optimization

7.8.1 Standard SCF procedure

The standard SCF procedure in TURBOMOLE starts by generating an initial density matrix, computing the KS matrix thereof, invoking DIIS extrapolation, and transforming the resulting KS matrix into an orthonormal basis using the MOs of the previous iteration. After applying level shift (i.e. increasing the HOMO-LUMO gap) and damping (i.e. scaling the diagonal KS matrix elements), the eigenvalue problem is solved and the eigenvectors are used to compute the new density matrix and the cycle starts again (cf. Figure 7).

In section II.5 a slightly different scheme has been discussed. As the density matrix depends only on the occupied MOs (typically 10 to 30% of the total number of MOs) only as many eigenvectors with the lowest eigenvalues must be computed as there are occupied MOs. The intermediate orthonormal basis is obtained by Cholesky decomposition of the overlap matrix. The DIIS convergence acceleration remains unaffected. However, level shifting and damping the *KS* matrix is impossible. Instead, some additional control over the convergence is exercised by averaging new and old density matrices. The standard SCF procedure in TURBOMOLE seems to be more robust, so that it is the preferred scheme, as long as CPU time consumption for the eigensolver does not become prohibitive.

7.8.2 Rotation based SCF procedures

Alternatives to the standard SCF procedure are motivated by the unfavorable cubic scaling of the eigensolver and the insufficient stability of the iterative scheme. The cubic scaling renders attempts for linear scaling of DFT or HF impossible. It can be improved only by taking advantage of the specific structure and properties of the KS matrix. Hence, standard eigensolvers aiming at the solution of the general problem are probably not the ultimate solution to the problem. The KS matrix is sparse in terms of a large number of small though non-vanishing entries whereas sparse linear algebra relies on matrices with a small number of non-zero elements and is thus not readily applicable. Approximate diagonalization may be achieved by deleting elements below a given threshold and transforming the matrix to block-tridiagonal form which is passed to a special block-tridiagonal divide-and-conquer eigensolver [20]. So far it is not yet clear whether this approach can compete in terms of efficiency and economy with standard parallel eigensolvers. Moreover, it does not address the stability problem of the standard SCF procedure.

An alternative to the numerically not well understood standard SCF procedure is a direct minimization of the total electronic energy with respect to the MO coefficients subject to the orthonormalization constraint. The number of non-redundant coefficients is the number of unoccupied times occupied MOs. Thus, the number of independent parameters to be optimized for large systems reaches easily 10^7! The fact that the standard procedure works for such a large number of parameters at all indicates that the optimization problem is simpler than the sheer number of parameters suggests. Rotation based SCF procedures incorporate the orthonormalization constraint mostly by using an exponential parametrization

$$\mathbf{C}_{new} = \mathbf{C}_{old}\mathbf{U} = \mathbf{C}_{old}\exp(\mathbf{A}), \qquad \mathbf{A} = \left(\begin{array}{c|c} 0 & X \\ \hline -X^T & 0 \end{array}\right) \tag{11}$$

where **C** is the MO coefficient matrix and the antisymmetric matrix **A** collects the non-redundant parameters in the off-diagonal blocks. The matrix elements of **A** are in general a function of the matrix **C**. The various number of schemes that have been suggested over about 25 years, differ essentially in (i) the computation of the first and second derivatives of the energy with respect to the matrix elements of **A**, (ii) the evaluation of the matrix exponential, and (iii) the optimization scheme (conjugate gradients, Newton-Raphson etc.).

In the orbital transformation method by Hutter et al. [9, 10] the exponential is evaluated exactly. Thus, the analytic gradient and the orthonormality constraint is obeyed exactly for an arbitrary choice of the reference point. Moreover, this method requires matrix operations over the occupied MOs, only. A minor disadvantage of this procedure is the rather complicated expression for the energy gradient and the lack of the second derivative. More important, the convergence rate depends on the preconditioner of the conjugate gradient optimization with no recipe for its improvement. For dense matrices, the scaling is still cubic with a reduced prefactor. Further reductions are feasible, if the involved matrices are sparse enough for efficient use of sparse matrix multiply.

In standard second-order SCF methods [11] gradient and Hessian are evaluated by Taylor expansion about **A**=0, which yields simple expressions in terms of Fock matrix entries and two-electron integrals in the MO representation for gradient **g** and Hessian **H** [12]. The new parameter set **A** is computed and the matrix exponential is approximated in a linear expansion.

$$\mathbf{A} = -\mathbf{H}^{-1}\mathbf{g} \tag{12}$$

$$\exp(\mathbf{A}) \approx \mathbf{I} + \mathbf{A} \tag{13}$$

Consequently the orthonormality of **C** is not preserved and requires orthonormalization ($O(N^3)$). The initial Hessian is updated during the subsequent iteration cycles by BFGS [13]. Since only Hessian gradient products are required the explicit construction of the Hessian can be avoided [14]. These second order schemes require a good set of start orbitals usually obtained from a few iterations with the standard SCF procedure which is not guaranteed to succeed.

In summary, a scheme is required, that is simple enough to be readily parallelized, has modest (distributed) memory requirements, exploits the potential sparsity combined with well-controlled convergence behavior, and overcomes the problem of ”good” start orbitals.

Expanding the orbital exponential approximately in terms of products of orthonormality preserving Givens rotation matrices $\mathbf{G}^{ov}$ which depend on the off-diagonal matrix element $\mathbf{A}_{ov}$ only, yields

$$\exp(\mathbf{A}) \quad \approx \quad \prod_{ov} \mathbf{G}^{ov}. \tag{14}$$

This bears close relation to the Jacobi procedure for matrix diagonalization [13], Pulay’s pseudo block-diagonalization scheme [15], and the parametrization of the energy by means of Givens rotation matrices [16]. The quality of this approximation goes beyond the linear approximation in Eq. 13 above. The individual matrix elements $\mathbf{A}_{ov}$ are given by

$$\mathbf{A}_{ov} \quad = \quad -\frac{\mathbf{F}_{ov}}{\mathbf{F}_{vv} - \mathbf{F}_{oo} - (oo|vv) - 3(ov|ov)} \approx -\frac{\mathbf{F}_{ov}}{\mathbf{F}_{vv} - \mathbf{F}_{oo} + \lambda} \tag{15}$$

where o and v refer to the index of an occupied and unoccupied (virtual) orbital, respectively. The approximative term to the right is an approximation to the diagonal Hessian supplemented by a level shift parameter λ. This approximation holds for a sufficiently large HOMO-LUMO gap and assumes the analytic Hessian to be diagonally dominant except for a small region with non-negligible off-diagonal elements. The level shift parameter serves to keep the Hessian positive definit and to restrict the step-length $||\mathbf{A}||$. For $\lambda = 0$ this expression is similar to the pseudo block-diagonalization [15] which is numerically less stable than the standard SCF procedure, as presented in more detail in section II.5.2.

The Givens rotations are trivial to implement in parallel given row-wise distribution of the MO coefficient matrix over the processes. The maximum number of operations is $4n_{occ}(N - n_{occ})N \leq N^3$. Since only rotations above a certain threshold are actually considered, the number of operations drops rapidly (case dependent) under 10% of the maximum value with forthcoming convergence. This procedure is in fact no more expensive than a dense matrix multiply for the full matrix. Additionally, it is sufficient to transform the off-diagonal block plus the diagonal matrix elements saving some additional 40% for the similarity transform. As a reasonably diagonal dominant KS matrix is required, starting from scratch one iteration with the standard SCF procedure must be carried out. In geometry optimizations the orthonormalized MOs of a nearby geometry may be directly used. As in any other rotation based scheme, the optimized MOs are not in the canonical form, i.e. they are not eigenvectors of the KS matrix. Depending upon the later use of the MO coefficients a final full or partial diagonalization is necessary.

The crucial ingredient for stable SCF convergence is the (dynamic) computation of λ. The Hessian matrix is decomposed into a (large) diagonally dominant part, which is assumed to be a reasonably accurate approximation to the analytic Hessian, and the remaining critical part that is of limited use only. The level shift λ serves the purpose to maintain the curvature almost everywhere and to increase the curvature for the critical part. There is a subtle difference to the trust region minimization method [17]: in this technique the level shift parameter λ is adjusted such as to remain within the trust region of convergence. It is increased or decreased depending upon the ratio of actual and predicted energy changes. Here, the Hessian approximation may be that poor, that the feedback mechanism fails. Instead λ is determined by bisection such that the step-length $||\mathbf{A}||$ remains below a given value and the diagonal Hessian remains positive definite. The maximum step-length is dynamically adjusted via a feedback mechanism coupled to the readily available quantities: energy changes, norm of the DIIS error matrices (indicating the vicinity to a minimum), and the gradient. The procedure readily applies to closed shell and unrestricted KS and requires some modifications for open shell HF as outlined by [9]. Regarding UHF/UKS the Hessian has large off-diagonal elements connecting α and β terms [18]. Hence, we may expect this procedure based on diagonal Hessian approximation to work less well for UHF/UKS.

It is important to stress, that this scheme crucially relies on DIIS extrapolation. M. Krack pointed out, that DIIS is not reliably converging with small HOMO-LUMO gaps or bad initial orbitals. The answer to this apparent contradiction is that DIIS is an extrapolation procedure which depends on the input data and the standard SCF procedure tends to somewhat uncontrolled strong changes in the resulting KS matrix that DIIS cannot cope with: poor input, poor extrapolation. Hence, the sole task for the eigensolver or any substitute thereof is to provide adequate input for the extrapolation procedure.

Tests on a variety of systems reveal three remarkable properties: (i) On systems which exhibit no problem to the standard procedure, the suggested procedure works as well. (ii) For difficult systems (small HOMO-LUMO gap, bad starting orbitals, root flipping) the scheme does not suffer from wild oscillations or poor convergence but instead shows a smooth robust convergence. (iii) Close to the minimum the convergence rate slows down. Thus, the startup problems of most second-order SCF methods are nicely overcome, but problems arise where they are normally expected to succeed. While for second-order SCF procedures relying on BFGS updates the Hessian ideally converges to the analytic Hessian close to convergence, this does not apply to procedures relying on (modified) diagonal approximations to the Hessian, which may produce a few completely wrong entries for small HOMO-LUMO gaps. Hence, the step vector will point into the wrong direction and - with no adequate input - DIIS cannot overcome this deficiency. A remedy to this problem is to incorporate corrections for the missing two-electron MO integrals (cf. Eq. 15).

cluster	point	MOs	wall clock time[s]						
	group	(total)	4_1^a	8_1^a	16_1^a	32_1^a	48_2^a	64_2^a	72_3^a
divide-and-conquer eigensolver & full similarity transform									
Zeolite	C_1	4992	120	64	40	26	-	-	-
BPTI	C_1	8574	-	367	168	105	89	78	-
Barnase	C_1	16371	-	-	-	-	592	-	406
orbital rotation & partial similarity transform									
Zeolite	C_1	4992	50	27	17	11	-	-	-
BPTI	C_1	8574	-	198	74	51	35	29	-
Barnase	C_1	16371	-	-	-	-	124	-	96

[a] N_n: N CPUs distributed symmetrically over n SMP nodes.

Table 4: Eigenvalue problem including transformation into the orthonormal basis.

Symmetric clusters have been excluded from Table 4 as the computational effort scales with $O(N^3/n_g^2)$ and thus, are in most cases not very relevant. For symmetric systems, the parallel scalability is not improved uniformly for a smaller number of processors as a consequence of the multilevel parallelism used. The timings for the rotation based procedure are a linear function of the number of Givens rotations actually carried out, which depends on the molecule and the convergence characteristic. Since the rotations are BLAS level 1 routines, they achieve less than 1 Gflops, compared to 1 to 2 Gflops for the eigensolver. Also note that for the orbital rotation based scheme more than half of the time is spent on the partial similarity transform. Comparing the eigensolver and the orbital rotations, only, the latter is typically faster by a factor of 3 to 20.

7.9 Gradients

The evaluation of the one-electron and two-electron integral contributions to the gradient of the energy with respect to displacement of the nuclear coordinates closely follows the scheme outlined for the RIDFT module with regard to parallelization. Some additional routines such as the calculation of the integral derivative estimator have been parallelized as well. Scalability and memory requirements thus closely mimic those of the RIDFT module (Table 5).

Highly symmetric compounds (V240_ball) display short execution times at poor parallel scaling: 80% of the one-electron contribution goes into transformation of density and energy-weighted density matrices from SAO into CAO representation. The remaining overhead primarily arises from the serial preparation of symmetry tables and transformation coefficients. The other extreme are large unsymmetric compounds (Barnase) which scale reasonably with the primary contributions by one-electron and exchange contributions. The latter are dominated by grid construction (80%). The general overhead contributing to the total execution time stems almost solely from the computation of the PQ matrix and its Cholesky decomposition.

cluster	point group[b]	MOs (total)	grid	wall clock time[s]						
				4_1^a	8_1^a	16_1^a	32_1^a	48_2^a	64_2^a	72_3^a
one-electron contributions										
V240_ball	I_h(120)	19320	4	-	124	89	60	41	40	-
V80_tube	C_{4h}(8)	6440	4	69	37	21	11	-	-	-
V80_sheet	C_s(2)	6440	4	311	159	93	47	-	-	-
BPTI	C_1(1)	8574	4	-	-	474	263	211	143	-
Barnase	C_1(1)	16371	4	-	-	-	-	874	-	593
Coulomb contribution										
V240_ball	I_h(120)	19320	4	-	52	40	31	34	34	-
V80_tube	C_{4h}(8)	6440	4	79	38	21	13	-	-	-
V80_sheet	C_s(2)	6440	4	317	159	82	40	-	-	-
BPTI	C_1(1)	8574	4	-	-	194	91	74	53	-
Barnase	C_1(1)	16371	4	-	-	-	-	201	-	151
exchange contribution										
V240_ball	I_h(120)	19320	4	-	21	17	18	17	18	-
V80_tube	C_{4h}(8)	6440	4	73	39	20	14	-	-	-
V80_sheet	C_s(2)	6440	4	277	123	72	41	-	-	-
BPTI	C_1(1)	8574	4	-	-	405	236	175	131	-
Barnase	C_1(1)	16371	4	-	-	-	-	589	-	407
total timings										
V240_ball	I_h(120)	19320	4	-	293	248	213	201	207	-
V80_tube	C_{4h}(8)	6440	4	229	121	71	50	-	-	-
V80_sheet	C_s(2)	6440	4	943	472	279	155	-	-	-
BPTI	C_1(1)	8574	4	-	-	1235	701	574	434	-
Barnase	C_1(1)	16371	4	-	-	-	-	2117	-	1549

[a] N_n: N CPUs distributed symmetrically over n SMP nodes. [b] Order of point group in parentheses.

Table 5: Construction of the exchange part of the KS matrix

7.10 Total performance

The effective speedup that can be achieved is considerably case dependent and closely related to the memory access pattern. As the IBM SMP cluster at the Research Centre Jülich is very sensitive to cache misses applications involving large matrices display variations in execution times. Hence, absolute speedup values are of little value, especially as they are only qualitatively transferable among different parallel computer systems. Thus, aspects of practical relevance and better transferability are focussed on. Figure 9 summarizes the parallel scalability of the RIDFT (wall clock time per SCF iteration) and the RDGRAD module. With increasing order of the point group scalability typically decreases as symmetry related overhead increases.

In Table 6 the total timings of RIDFT and RDGRAD are decomposed into the major contributions: overhead due to symmetry treatment (preparation of transformation coefficients, CAO-SAO transformation), grid construction, and linear algebra. The effect of symmetry

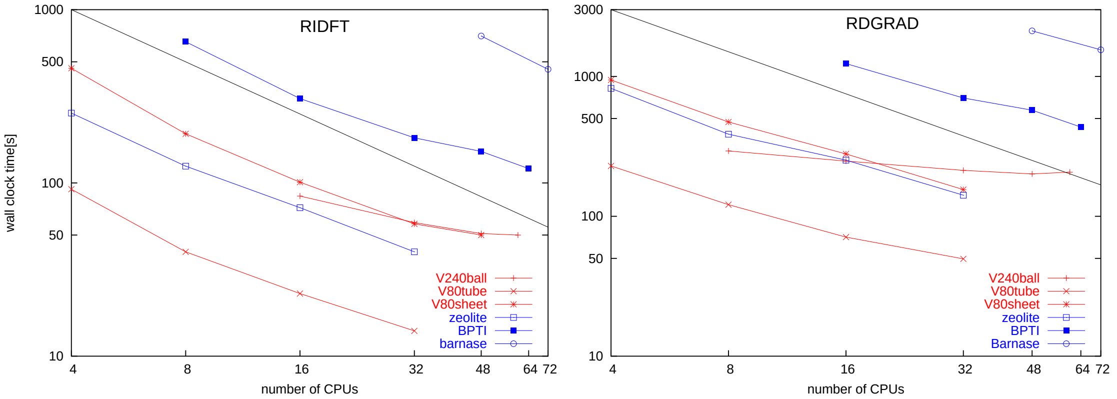


Figure 9: Parallel performance the RIDFT (wall clock time per SCF iteration) and the RDGRAD module.

is immediately apparent: high symmetry results in negligible effort for integral evaluation, quadrature and linear algebra contrasted by substantial (either serial or not well load balanced) contributions of symmetry-related operations and transformations. Large unsymmetric clusters spend more than 75% of the execution per SCF iteration in linear algebra using the standard scheme, which can be approximately halfed using the sketched orbital rotation based scheme, the remainder being almost exclusively due to matrix multiply.

Since matrix multiply is an operation of order $O(N^3/n_g^2)$ whereas integral evaluation and quadrature scale at most with order $O(N^{1.5})$ it is immediately apparent that unsymmetric clusters with basis set sizes of 40000 will spend almost all the execution time in matrix multiply. The use of symmetry greatly reduces this computational burden: even with point group C_s of order 2 linear algebra execution times go down by a factor of 4. Hence, the possibility exploiting point group symmetry whenever possible must not be lightheartedly discarded.

cluster	point	MOs	N_n^a	wall clock time [s]	
	group[b]	(total)			
	RIDFT			startup[c]	SCF iteration[d]
V240_ball	I_h(120)	19320	32_1	177 (43%, 2%, <1%)	58 (1%, 29%)
V80_tube	C_{4h}(8)	6440	16_1	13 (7%, 12%, 10%)	23 (18%, 2%)
BPTI	C_1(1)	8574	32_1	214 (<1%, 18%, 37%)	235 (58%, <1%)
Barnase	C_1(1)	16371	48_2	857 (<1%, 14%, 47%)	1095 (75%, <1%)
	RDGRAD			gradient[e]	
V240_ball	I_h(120)	19320	32_1	213 (57%, 4%, <1%)	
Barnase	C_1(1)	16371	48_2	2117 (<1%, 22%, 8%)	

[a] N_n: N CPUs distributed symmetrically over n SMP nodes. [b] Order of point group in parentheses.

[c] Time spent on symmetry treatment, grid construction (grid 2) and linear algebra in parentheses.

[d] Time spent on linear algebra and symmetry treatment in parentheses (standard scheme).

[e] Time spent on symmetry treatment, grid construction (grid 4) and linear algebra in parentheses.

Table 6: Percentage of wall clock time spent in linear algebra, grid construction and symmetry treatment, respectively.

The evaluation of the gradient requires larger grids as compared to the SCF iterations so that grid construction is computationally more intensive. Overhead due to symmetry treatment is somewhat higher as the CAO-SAO transformation of the density and the energy-weighted density are necessary. Linear algebra is of little importance for the gradient code.

8 The modules DSCF and GRAD

In fact, almost everything can be taken over from RIDFT and RDGRAD to these modules. Density and possibly KS matrix are replicated once per SMP node during the evaluation of Coulomb and exchange contribution. On systems with a large amount of memory, this is still the most economic solution. Additionally, difference density and KS matrices are stored in distributed memory to avoid I/O. These quantities should be attractive targets for compression. If memory becomes scarce, switching to distributed data storage (quadrature is treated identical to RIDFT) and separate calculation of Coulomb and exchange contributions offers a simple road to reduce communication at the expense of at most doubling the integral evaluation costs. For the use of a very large number of processors, switching from dynamic to static load-balancing is presumably the only way to keep communication demand within bounds. Overall scaling is much better than for RIDFT with the MARI-J method since Coulomb and exact HF exchange evaluation may take orders of magnitude longer.

9 Summary and outlook

The modules RIDFT, RDGRAD, DSCF AND GRAD have been parallelized with no restriction of the symmetry treatment. Tests on molecules with up to 1710 atoms and up to $\approx$ 20000 basis functions ($\approx$ 39000 auxiliar basis functions) have been carried out. Geometry optimizations applied to research problems in the field of nano-structured compounds [21] are being carried out. A simple parallelizable orbital rotation scheme has been suggested, which overcomes convergence with the standard SCF scheme while being substantially faster than the conventional procedure, although still of cubic scaling. Linear algebra operations and in particular matrix multiply are dominating the execution time in RIDFT for the largest unsymmetric molecule. Although exploitation of symmetry greatly reduces the associated computational effort, the cubic scaling of matrix multiply will render calculations for much larger problem sizes computationally very expensive. Most linear algebra is closely related to the wave function optimization step, so that future efforts in quantum chemistry will involve exploring efficient schemes minimizing the number of general matrix multiplications.

Acknowledgments

This work has not been possible without the support by the Theoretical Chemistry group of the University of Karlsruhe headed by Prof. Reinhart Ahlrichs.

Marek Sierka (group of Prof. Joachim Sauer, TU Berlin) provided many inputs for large clusters and carried out realistic large calculations by the new experimental code.

The MPI based conversion routines from distributed block-diagonal matrices into their block-cyclic equivalents and vice versa have been implemented by Inge Gutheil (ZAM).

Matthias Bolten (guest student at ZAM) implemented initially Global Array based conversion routines and interfaced them to the ScaLAPACK package.

Martin Fengler (guest student at ZAM) implemented and experimented with a parallel version of Pulay's pseudo block-diagonalization scheme.

The Global I/O extension to the Global Array toolkit originated from Karsten Scholtyssik (formerly ZAM).

Bibliography

[1] J. Nieplocha, J. Ju, M.K. Krishnan, B. Palmer, and V. Tipparaju `http://www.emsl.pnl.gov/docs/global/ga.html`

[2] `http://www.research.ibm.com/actc`

[3] `http://www.netlib.org/blacs`

[4] `http://www.netlib.org/scalapack/pblas_qref.html`

[5] `http://www.netlib.org/scalapack`

[6] H. Dachsel and H. Lischka, *Theor. Chim. Acta* **92**, 339 (1995).

[7] P. Pulay, *Chem. Phys. Lett.* **73**, 393 (1980).

[8] P. Pulay, *J. Comput. Chem.* **3**, 556 (1982).

[9] J. Hutter and M. Parrinello, *J. Chem. Phys.* **101**, 3862 (1994).

[10] J. VandeVondele and J. Hutter, *J. Chem. Phys.* **118**, 4365 (2003).

[11] G. Chaban, M. W. Schmidt, and M. S. Gordon, *Theor. Chim. Acta* **97**, 88 (1997).

[12] J. Douady, Y. Ellinger, R. Subra, and B. Levy, *J. Chem. Phys.* **72**, 1452 (1980).

[13] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes*, 2nd Ed. Cambridge, University Press (1992).

[14] T. H. Fischer and J. E. Almlöf, *J. Phys. Chem.* **96**, 9768 (1992).

[15] P. Császár and P. Pulay, *J. Comp. Chem.* **3**, 227 (1982).

[16] M. Head-Gordon and J. A. Pople, *J. Phys. Chem* **92**, 3063 (1988).

[17] T. Helgaker in *Lecture Notes in Quantum Chemistry*, Ed. B. O. Roos, Springer Verlag, Heidelberg, pp. 313 (1992).

[18] F. Neese, *Chem. Phys. Lett.* **325** 93 (2000).

[19] Y. Bai, W. N. Gansterer, and R. C. Ward, *ACM Trans. Math. Softw.* **30**, 326 (2004).

[20] W. N. Gansterer, R. C. Ward, and R. P. Muller, *ACM Trans. Math. Softw.* **28**, 45 (2002).

[21] Th. Müller, J. Sauer, M. Sierka, and J. Döbler, *VSR project: Structure and properties of nanostructured clusters of transition metal oxides* (2004).

Continuous Fast Multipole Method

Holger Dachsel

John von Neumann Institute for Computing
Central Institute for Applied Mathematics
Research Centre Jülich
52425 Jülich, Germany
E-mail: h.dachsel@fz-juelich.de

1 Introduction

In several scientific applications such as molecular dynamics [1] and plasma physics [2] the evaluation of a pairwise potential is required. Very often this is the most time-consuming step in a calculation. The direct method to evaluate these potentials scales quadratically with the number of particles N which places a severe restraint on the size of systems which can be treated. Many methods have been proposed to avoid the quadratic scaling [3]. Unfortunately, all these methods lead to unpredictable errors because they rely upon not generally applicable approximations [4]. In particular cut-off approaches show errors which often can not be accepted due to the significance of the long range charge-charge interaction. It is highly desired to avoid the order N^2 scaling. One of the methods to achieve linear scaling is Greengard's [5] Fast Multipole Method (FMM). The purpose of the FMM is to group together remote charges such that a collection of distant charges can be treated as one single charge. The Fast Multipole Method expands local charges in multipole expansions. The multipole expansions of several particles about a common origin can be summed to represent a collection of point charges by just one multipole expansion. The collections of point charges are grouped in boxes which form the FMM tree. The FMM is a computational scheme how to manipulate these expansions to achieve linear scaling. The Fast Multipole Method can be applied to the evaluation of $r^{-n}(n > 0)$ pairwise interactions. Unfortunately, the FMM is not free of parameters. The computation time and the accuracy depend on three parameters, the length of the multipole expansions, the depth of the FMM tree, and finally the separation criteria - the number of boxes between two boxes which can interact via multipoles. It is very inconvenient to set the parameters by an user-request. In addition, the three parameters are not independent among each other. One can define a function $f(L, D, ws) = 0$, where L is the length of the multipole expansions, D is the depth of the FMM tree, and ws is the separation criteria. The computation time t depends not only on $L, D,$ and $ws.$ The requested threshold and the kind of distribution, homogeneous or heterogeneous distributed particles have also an impact on the computation time. In our FMM implementation we minimize the computation time $t = t(L, D, ws, kind\,of\,distribution, threshold)$. $L, D,$ and ws are the variables, the $kind\,of\,distribution$ and the $threshold$ are the constants. With this approach we have found a reasonable solution of the problem on separating the particles in near and far field.

Within the framework of the HPC-Chem project [6] our implementation of the FMM to treat point charges in a very efficient way is the first step towards the CFMM (Continuous Fast Multipole Method) to calculate charge distributions arising in Density Functional and Hartree Fock calculations. The ideas of FMM can be applied to the evaluation of Electron Repulsion Integrals (ERI's). The computation of the ERI's is in general a step which requires $O(n^4)$ work regarding the number of basis functions n. By several computational techniques [7] the scaling could be improved significantly to $O(n^2)$. The use of CFMM gives the possibility to make a further improvement in scaling, from $O(n^2)$ to $O(n)$. The Coulomb interaction of two charge distributions decreases exponentially with increasing separation, and the two distributions then interact as classical point charges.

2 Theory

The basics of our FMM implementation are described by C. A. White and M. Head-Gordon [8, 9]. In addition, a new scheme of estimating the FMM errors and an approach to evaluate the Wigner rotation matrices [9, 10] more stable for higher multipole moments have been implemented.

A. Factorization of inverse distance

The inverse distance between two point charges located at $\mathbf{a} = (a, \alpha, \beta)$ and $\mathbf{r} = (r, \theta, \phi)$ can be written as an expansion of the associated Legendre polynomials.

$$\frac{1}{|\mathbf{r} - \mathbf{a}|} = \sum_{l=0}^{\infty} P_l(cos(\gamma)) \frac{a^l}{r^{l+1}} \tag{1}$$

$$\frac{1}{|\mathbf{r} - \mathbf{a}|} = \sum_{l=0}^{\infty} \sum_{m=-l}^{m=l} \frac{(l - |m|)}{(l + |m|)} \frac{a^l}{r^{l+1}} P_{lm}(cos(\alpha)) P_{lm}(cos(\theta)) cos(m(\beta - \phi)) \tag{2}$$

$$\frac{1}{|\mathbf{r}-\mathbf{a}|} = \sum_{l=0}^{\infty} \sum_{m=-l}^{m=l} \frac{(l-|m|)}{(l+|m|)} \frac{a^l}{r^{l+1}} P_{lm}(cos(\alpha)) P_{lm}(cos(\theta)) e^{-im(\beta-\phi)} \tag{3}$$

The expansion converges under the condition $a < r$. γ is the angle between the two vectors $\mathbf{a}$ and $\mathbf{r}$. Eq. (2) and Eq. (3) represent a complete factorization of an interaction of two unit charges. On the basis of Eq. (3) one can define moments of a multipole expansion. q is the particle charge.

$$\omega_{lm} = qO_{lm} = qa^l \frac{1}{(l+|m|)} P_{lm}(cos(\alpha)) e^{-im\beta} \tag{4}$$

Based on Eq. (3) one can also define the coefficients of a Taylor expansion.

$$\mu_{lm} = qM_{lm} = q\frac{1}{r^{l+1}}(l-|m|) P_{lm}(cos(\theta)) e^{im\phi} \tag{5}$$

Combining Eqs. (3), (4), and (5) together a factorization of the inverse distance can be written in a compact form.

$$\frac{1}{|\mathbf{r}-\mathbf{a}|} = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \omega_{lm}\mu_{lm} \tag{6}$$

The moments of a multipole expansion and the coefficients of a Taylor expansion about a common origin can of course be summed.

B. Translation operators

Essential to the FMM are the three operators to translate multipole expansions and Taylor expansions in space. The first operator, A, is used to shift a multipole expansion from $\mathbf{a}$ to $\mathbf{a}+\mathbf{b}$.

$$\omega_{lm}(\mathbf{a}+\mathbf{b}) = \sum_{j=0}^{l} \sum_{k=-j}^{j} A_{jk}^{lm}(\mathbf{b})\omega_{jk}(\mathbf{a}) \tag{7}$$

The operator A_{jk}^{lm} is given by

$$A_{jk}^{lm} = O_{l-j,m-k} \tag{8}$$

The second operator, B, transforms a multipole expansion into a Taylor expansion.

$$\mu_{lm} = \sum_{j=0}^{\infty} \sum_{k=-j}^{j} B_{jk}^{lm}\omega_{jk} \tag{9}$$

The operator B_{jk}^{lm} is given by

$$B_{jk}^{lm} = M_{j+l,k+m} \tag{10}$$

The third operator, C, translates a Taylor expansion of a point $\mathbf{r}$ about the origin to a Taylor expansion of $\mathbf{r}$ about a point $\mathbf{a}$.

$$\mu_{lm}(\mathbf{r} - \mathbf{a}) = \sum_{j=0}^{\infty} \sum_{k=-j}^{j} C_{jk}^{lm}(\mathbf{a}) \mu_{jk}(\mathbf{r}) \tag{11}$$

The operator C_{jk}^{lm} is given by

$$C_{jk}^{lm} = O_{j-l,k-m} \tag{12}$$

3 The Fast Multipole Method

The FMM consists of several parts. First all particles are enclosed by a box with coordinate ranges [0,1]x[0,1]x[0,1]. The parent box which contains all the particles is divided in half along each Cartesian axis to yield a set of 8 smaller child boxes. The child boxes are subdivided again (Figure 1). The depth of the tree is determined so that the computation time becomes a minimum by achieving an error in the energy which is less or equal to a requested threshold. The particles are sorted by box numbers using the radix sort algorithm [11] which scales linearly. In addition to scaling and sorting the FMM consists of four passes schematically shown in Figure 2. In Pass 1 the charges contained within each lowest level box are expanded in multipoles about the center of the box. The multipole expansions are translated to the center of the parent boxes (Figure 3). In Pass 2 the multipole expansions are transformed into Taylor expansions. The two boxes must be separated by at least one box on the current tree level, but only provided that parents of the two boxes

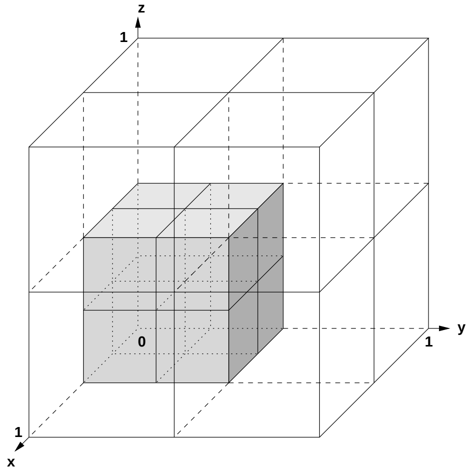


Figure 1: The particle space is divided in child boxes along the Cartesian axes

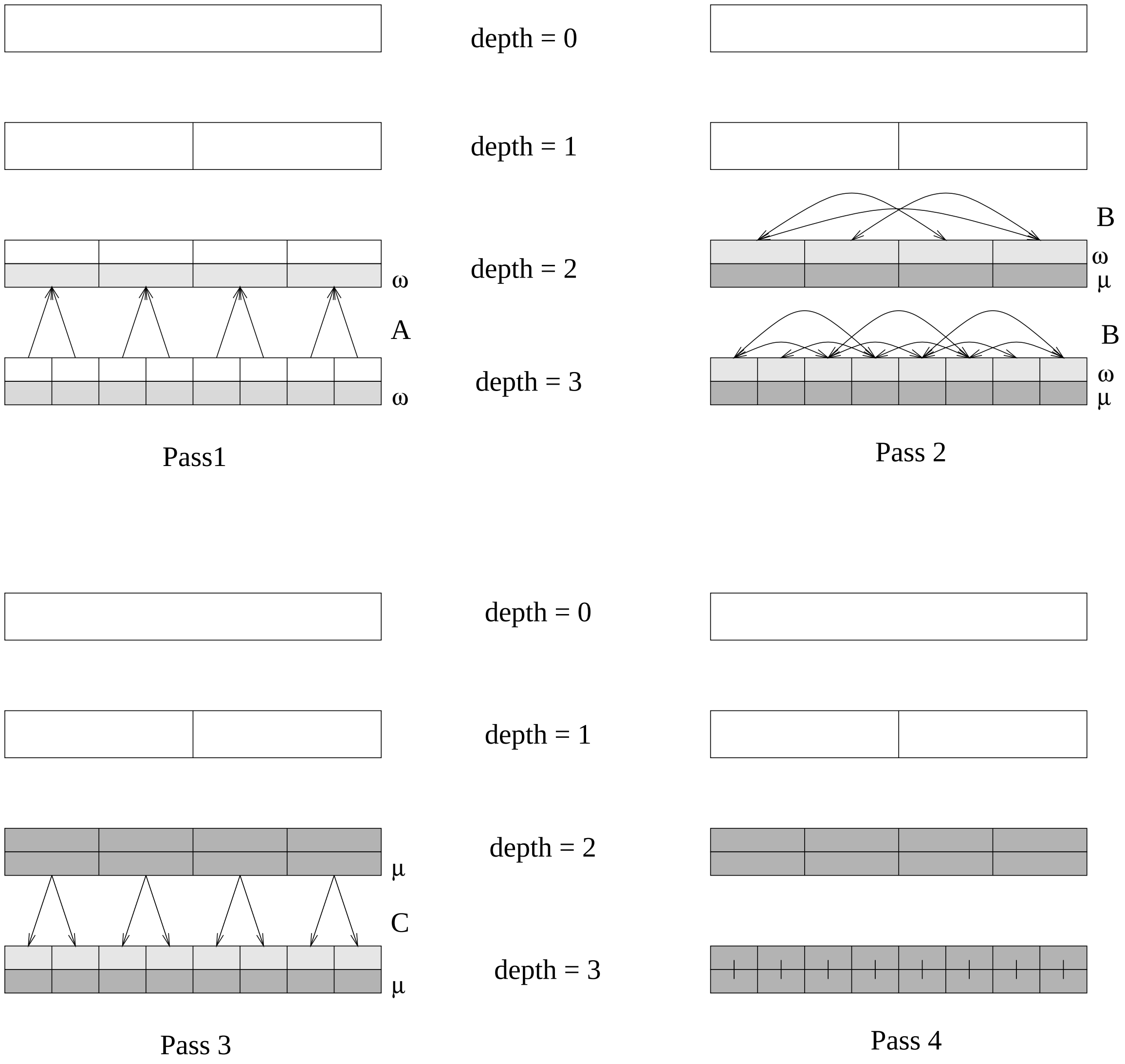


Figure 2: Schematic view on one dimensional FMM with parameter $ws = 1$

are not separated on the next higher tree level. Pass 2 is by far the most time-consuming step of the FMM (Figure 4). In Pass 3 the parent's Taylor expansions are translated to the centers of the parent's children. At the end of Pass 3 each lowest level box contains a Taylor expansion of all far field interactions (Figure 5). In Pass 4 for each lowest level box the multipole expansion and the Taylor expansion are multiplied. The sum over all lowest level boxes gives the far field energy. Finally, in Pass 5 the remaining near field energy is computed by the direct method.

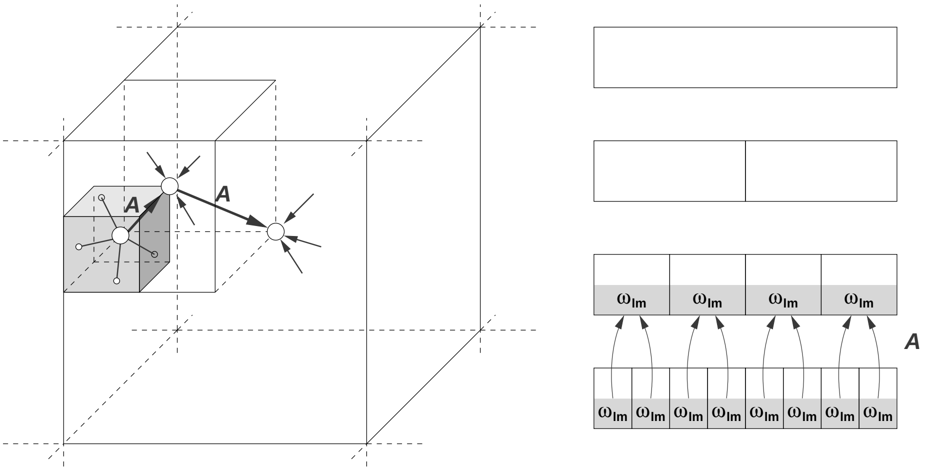


Figure 3: Calculation and shifting of multipole moments (Pass 1)

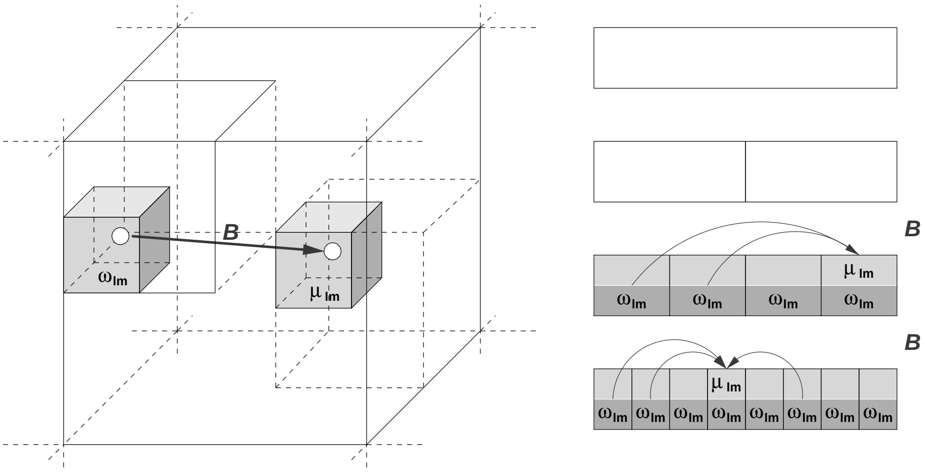


Figure 4: Transformation of multipole moments to Taylor coefficients (Pass 2)

3.1 The Wigner rotation matrices

The conventional Fast Multipole Method requires $O(L^4)$ work with regard to the length of the multipole expansions L. $O(L^3)$ scaling can be achieved by performing the translations in three steps. First the moments of a multipole expansion or the coefficients of a Taylor expansion are rotated about the z-axis and y-axis such that the phase factors in Eq. (4) and Eq. (5) vanish and the associated Legendre polynomials P_{lm} degenerate to the Legendre polynomials P_l. In the second step the translations are performed.

$$\omega_{lm} = \sum_{j=m}^{l} \frac{a^{l-j}}{(l-j)!} \omega_{jm} \tag{13}$$

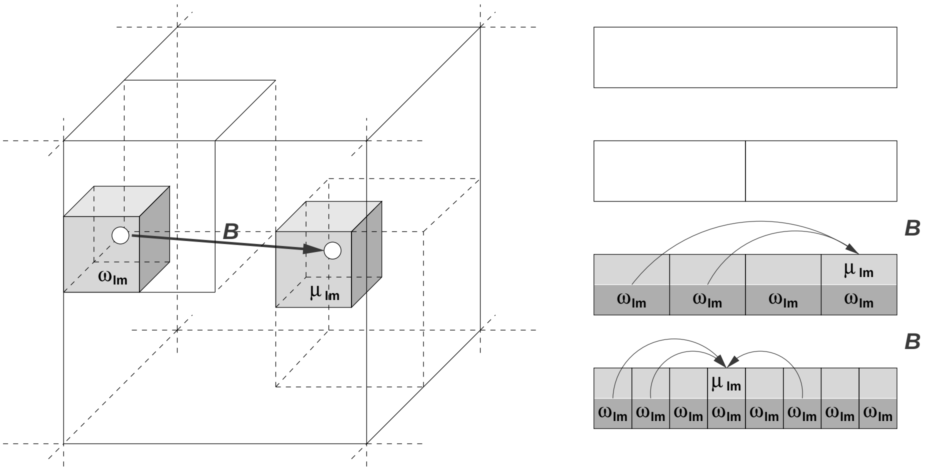


Figure 5: The Taylor expansions are shifted to the centres of the child boxes

$$\mu_{lm} = \sum_{j=m}^{\infty} \frac{(j+l)!}{r^{j+l+1}} \omega_{j,-m} \tag{14}$$

$$\mu_{lm} = \sum_{j=l}^{\infty} \frac{a^{j-l}}{(j-l)!} \mu_{jm} \tag{15}$$

Finally, the translated multipole moments and the Taylor coefficients are rotated back using the inverse rotation matrices. The rotation about the z-axis is simply a complex multiplication. The only difficult portion is the determination of the Wigner rotation matrices $d^l_{km}(\theta)$ which correspond to the rotation about the y-axis. The analytical calculation of the $d^l_{km}(\theta)$ requires $O(L^4)$ work and is numerically instable.

$$\begin{aligned} d^l_{k\,m} &= \frac{1}{2^l} \sqrt{\frac{(l-m)!\,(l+m)!}{(l-k)!\,(l+k)!}}\,(1 + sign(k)cos(\theta))^{|k|}\,(sin(\theta))^{m-|k|} \\ &\cdot \sum_{n=0}^{l-m} (-1)^{l-m-n} \binom{l-k}{n} \binom{l+k}{l-m-n} (1+cos(\theta))^n\,(1-cos(\theta))^{l-m-n} \end{aligned} \tag{16}$$

$$d^l_{m\,k} = (-1)^{k+m} d^l_{k\,m} \tag{17}$$

$$l \geq 0 \quad , \quad k = -l, ..., l \quad , \quad |k| \leq m \leq l \tag{18}$$

$$d^l_{k\,m} = (-1)^{k+m} d^l_{-k\,-m} \tag{19}$$

$$l > 0 \quad , \quad m = -l, ..., l-1 \quad , \quad k = -l, ..., -(m+1) \tag{20}$$

The essential recursion relation we will use to determine the rotation matrices is given by White [9] and Edmonds [10].

$$\begin{aligned} d^l_{k+1\,m} = {} & \frac{k+m}{\sqrt{l(l+1)-k(k+1)}} \frac{sin(\theta)}{1+cos(\theta)} d^l_{k\,m} \\ & + \sqrt{\frac{l(l+1)-m(m-1)}{l(l+1)-k(k+1)}} d^l_{k\,m-1} \end{aligned} \tag{21}$$

$$d^l_{0m} = \sqrt{\frac{(l-m)!}{(l+m)!}} P_{lm}, \; m \geq 0 \tag{22}$$

$$d^l_{0m} = (-1)^m \sqrt{\frac{(l-|m|)!}{(l+|m|)!}} P_{l|m|}, \; m < 0 \tag{23}$$

Unfortunately, Eq. (21) becomes instable in case of higher moments. We have combined Eq. (21) with a second recurrence to overcome the numerical instabilities.

$$\begin{aligned} d^l_{k\,m-1} = {} & \sqrt{\frac{l(l+1)-k(k+1)}{l(l+1)-m(m-1)}} d^l_{k+1\,m} \\ & - \frac{k+m}{\sqrt{l(l+1)-m(m-1)}} \frac{sin(\theta)}{1+cos(\theta)} d^l_{k\,m} \end{aligned} \tag{24}$$

$$d^l_{kl} = \frac{1}{2^l} \sqrt{\frac{(2l)!}{(l-k)!(l+k)!}} (sin(\theta))^{l-k} (1+cos(\theta))^k \tag{25}$$

In addition to the two recurrences the error accumulations are evaluated for both of the recurrences to decide which recursion relation is more accurate for a given component of the rotation matrix. Both of the recursion relations should be used only for $cos(\theta) \geq 0$. In case of $cos(\theta) < 0$ addition theorems can be used given by Edmonds [10]. The combination of the two recurrences show a significant improvement of accuracy. Table 1 shows the absolute errors for $\theta = \frac{\pi}{2}$.

3.2 Error estimation

The error estimation by White and Head-Gordon [8] gives an upper limit for the error but is often not practical. We have used a different approach. The FMM has two error sources, the truncation of the multipole expansions and the truncation in the transformation of the multipole moments to Taylor coefficients. The errors depend on the three parameters of

L	First recursion relation	Both recursion relations
5	$1.20 \cdot 10^{-15}$	$1.11 \cdot 10^{-16}$
10	$3.33 \cdot 10^{-14}$	$2.78 \cdot 10^{-16}$
15	$1.65 \cdot 10^{-12}$	$7.49 \cdot 10^{-16}$
20	$1.99 \cdot 10^{-10}$	$1.50 \cdot 10^{-15}$
25	$1.35 \cdot 10^{-8}$	$6.27 \cdot 10^{-15}$
30	$1.07 \cdot 10^{-6}$	$6.89 \cdot 10^{-14}$
35	$9.64 \cdot 10^{-5}$	$1.38 \cdot 10^{-13}$
40	$8.89 \cdot 10^{-3}$	$1.35 \cdot 10^{-12}$
45	$5.52 \cdot 10^{-1}$	$4.48 \cdot 10^{-12}$
50	$1.18 \cdot 10^{2}$	$3.13 \cdot 10^{-11}$
55	$9.63 \cdot 10^{3}$	$1.27 \cdot 10^{-10}$
60	$1.54 \cdot 10^{6}$	$6.04 \cdot 10^{-10}$
65	$9.79 \cdot 10^{7}$	$4.83 \cdot 10^{-9}$

Table 1: Maximum absolute errors in computation of the d^l_{km}

the FMM, the depth of the tree, the separation criteria, and the length of the multipole expansions. The distribution is defined by the charges and positions of the particles. The separation criteria should be 1 to take full advantage of the FMM approach. The remaining parameters, the depth of the tree D and the length of the multipole expansions L can be optimized such that the computation time t is minimal and the energy error ΔE is not greater than an user-requested absolute error Δ. The floating-point operations can be computed separately for the near and far field part.

$$\frac{\partial t}{\partial D} = 0 \tag{26}$$

$$\frac{\partial t}{\partial L} = 0 \tag{27}$$

$$\Delta E \left(D,\, L\right) \leq \Delta \tag{28}$$

In general the solutions of Eqs. (26), (27), and (28) are non-integers. The length of the multipole expansion L must be an integer. The next larger integer is taken. The depth of tree D need not be an integer. Table 2 shows the number of multipoles depending on requested thresholds.

Req. abs. error	Abs. error	L	Depth
10^{-2}	$0.47 \cdot 10^{-2}$	1	10.0
10^{-3}	$0.98 \cdot 10^{-3}$	3	8.2
10^{-4}	$0.99 \cdot 10^{-4}$	5	6.5
10^{-5}	$0.26 \cdot 10^{-5}$	7	5.9
10^{-6}	$0.39 \cdot 10^{-6}$	10	5.4
10^{-7}	$0.76 \cdot 10^{-7}$	13	5.1
10^{-8}	$0.21 \cdot 10^{-8}$	17	4.8
10^{-9}	$0.29 \cdot 10^{-9}$	21	4.7
10^{-10}	$0.66 \cdot 10^{-10}$	25	4.6
10^{-11}	$0.93 \cdot 10^{-11}$	30	4.4
10^{-12}	$0.38 \cdot 10^{-12}$	34	4.3
10^{-13}	$0.66 \cdot 10^{-13}$	39	4.2

Table 2: Number of multipole moments depending on requested absolute errors

3.3 Implementation issues

Our FMM implementation is designed to evaluate systems consisting of billions of point charges. Heterogeneous distributions can be treated in the same efficient way as homogeneous distributions. The parameters of the FMM are determined such that the computation time is minimal depending on a requested threshold of the energy. All empty boxes on the tree are completely neglected. Any arrays in the dimension of all boxes are avoided. The maximal depth of the trees depends only on the integer length. Logical bit operations are used for box numbering. A logical right shift by three positions of a box number results in the number of the parent box. A logical left shift by three positions of a box number gives the number range of all child boxes (Figure 6). Our FMM implementation is fully based on spherical coordinates. We have made the first approach in parallelizing our FMM implementation. An efficiency of more than 90% up to 16 CPU's have been seen. In the parallel version Pass 3 is avoided because it can not be parallelized efficiently. It is easily possible to shift the work which is done in Pass 3 to Pass 4. All the four passes of the FMM are parallelized. Our parallelization strategy is based on the replicated data model which is a severe bottleneck. We will implement a data distributed version.

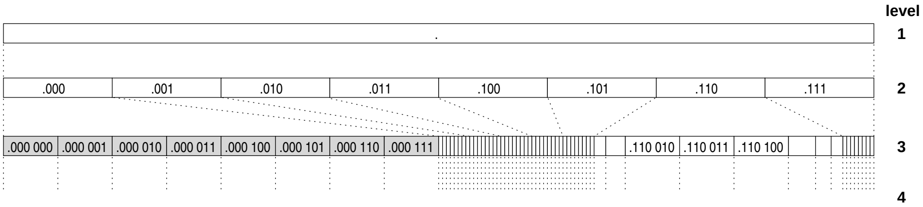


Figure 6: Boxes are numbered by logical bit shifts

Number of particles	Time [hh:mm:ss]	Scaling	Theoretical scaling
262.144	1:09		
2.097.152	10:02	8.7	8.0
16.777.216	1:23:48	8.4	8.0

Table 3: Scaling of FMM regarding the number of particles

3.4 Test calculations

We have performed several calculations on the IBM Regatta p690+ system at the Research Centre Jülich on a single CPU. We have chosen systems consisting of homogeneously distributed point charges. Table 3 shows the scaling of FMM for three systems consisting of 262.144, 2.097.152, and 16.777.216 particles. Each lowest level box contains 8 charges. We are capable of computing systems consisting of more than a billion of point charges. The energy computation of a system consisting of 1.073.741.824 particles required a computation time of only 8 hours on a single CPU. Because of memory limitations the FMM parameters were not optimized. The depth of the tree was set to 8, the length of the multipole expansion was equal to 10. Each lowest level box contained 64 charges. The relative error of the energy was less than 10^{-9}. In general, for many applications such a small relative error is not necessary. A multipole length of 5 instead of 10 would reduce the computation time by a factor of 8. The same system could be computed in 1 hour. A massively parallel version of our FMM implementation would be capable of treating such a system within seconds.

4 The Continuous Fast Multipole Method

The interest of using the FMM approach to reduce the scaling of the Coulomb problems has shifted to electronic structure calculations, particularly to density functional theory (DFT) and Hartree Fock calculations. The FMM theory is not immediately applicable to problems in which charge distributions have a non-zero extent. In a certain distance two separated charge distributions can interact as classical point charges within a given absolute error. This approach makes the FMM applicable to treat charge distributions arising in quantum chemistry.

4.1 Separation in near and far field

To obtain the error when treating charge distributions as point charges we compute the two-electron integral of four normalized s-type Gaussian functions analytically. The normalized s-type Gaussian function is given by

$$s \;=\left(\frac{2\alpha}{\pi}\right)^{\frac{3}{4}} e^{-\alpha r^2} \tag{29}$$

The product of two basis functions defines a charge distribution of non-zero extent. The product of two normalized s-type basis functions having the same exponent β represents a delta function for infinite large β which is the expression of a unit point charge. Assuming the normalized s-type basis functions of the products are located at the same positions in space the two-electron integral can be written as

$$<s_1 s_2|s_3 s_4> = \left(\frac{2\alpha_1}{\pi}\right)^{\frac{3}{4}}\left(\frac{2\alpha_2}{\pi}\right)^{\frac{3}{4}}\left(\frac{2\alpha_3}{\pi}\right)^{\frac{3}{4}}\left(\frac{2\alpha_4}{\pi}\right)^{\frac{3}{4}}\int\int\frac{e^{-\gamma r_1^2}\, e^{-\delta|\mathbf{r}_2-\mathbf{R}|^2}}{|\mathbf{r}_1-\mathbf{r}_2|}\,d\mathbf{r}_1\,d\mathbf{r}_2 \tag{30}$$

α_1 and α_2 are the exponent of the first distribution, α_3 and α_4 are the exponents of the second distribution. γ is the sum of α_1 and α_2, δ is the sum of α_3 and α_4. R is the distance between the two charge distributions. The integral can be calculated analytically.

$$<s_1 s_2|s_3 s_4> = \frac{erf(\sqrt{\frac{\gamma\cdot\delta}{\gamma+\delta}}R)}{R} \tag{31}$$

erf is the normalized Gaussian error function defined by

$$erf(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt \tag{32}$$

The factor $erf(\sqrt{\frac{\gamma\cdot\delta}{\gamma+\delta}}R)$ rapidly approaches 1 with increasing separation, and the two charge distributions then interact as classical point charges. Consequently, we define an error ϵ as

$$\epsilon\,(\gamma,\delta,R) = \frac{1}{R} - \frac{erf(\sqrt{\frac{\gamma\cdot\delta}{\gamma+\delta}}R)}{R} \tag{33}$$

$1 - erf$ can be substituted by the complementary Gaussian error function $erfc$.

$$\epsilon\,(\gamma,\delta,R) = \frac{erfc(\sqrt{\frac{\gamma\cdot\delta}{\gamma+\delta}}R)}{R} \tag{34}$$

ϵ	R	R_γ + R_δ
10^{-8}	6.46	9.24
10^{-9}	6.92	9.87
10^{-10}	7.36	10.50
10^{-11}	7.77	11.08
10^{-12}	8.17	11.64
10^{-13}	8.55	12.17
10^{-14}	8.91	12.68
10^{-15}	9.26	13.17

Table 4: Minimum distances [a.u.] for far field interaction (γ = 0.7, δ = 0.7)

Assuming one of the two distributions is a delta function Eq. (34) can be used to determine the extension for a single charge distribution.

$$\epsilon\,(\gamma, R_\gamma) = \frac{erfc(\sqrt{\gamma}R_\gamma)}{R_\gamma} \tag{35}$$

Considering two charge distributions having the exponents γ and δ the sum of R_γ and R_δ is always greater than R. Table 4 shows the accuracy of the approximations of R by the sum of R_γ and R_δ.

If the two basis functions of a charge distribution are not located at the same position in space the threshold ϵ is divided by the Gaussian pre-factor. Because the Gaussian pre-factor is always less than or equal to one the extensions of charge distributions of separated basis functions are always smaller. Charge distributions of higher angular momenta are treated as s-type distributions.

4.2 Extensions of products of contracted basis functions

A generally contracted basis function χ

$$\chi = Nx^l y^m z^n \sum_{i=1}^{j} c_i e^{-\alpha_i r^2} \tag{36}$$

with the property

$$<\chi|\chi> = 1 \tag{37}$$

is approximated by the function χ_a

$$\chi_a = Nx^l y^m z^n \left(\sum_{i=1}^{j} |c_i| \right) e^{-min(\alpha_1,..,\alpha_j)r^2} \tag{38}$$

satisfying the condition $\chi_a \geq \chi$. The extension of the distribution $\chi_a\chi_a$ is always larger compared with the extension of the distribution $\chi\chi$.

4.3 Multipole moments of charge distributions

In the FMM theory point charges are expanded in multipole moments. Now we have to expand charge distributions instead of point charges. Because a charge distribution has non-zero extent contributions of the distribution are existing everywhere in space. Compared with FMM where one sums over all point charges we must integrate over the charge distributions in the Continuous Fast Multipole Method (CFMM).

$$\text{FMM:}\;\; \omega_{lm} = \sum_j q_j\, r_j^l\, \frac{P_{lm}\left(sin\left(\theta_j\right), cos\left(\theta_j\right)\right)}{(l+m)!}\left(cos\left(m\phi_j\right) - i \cdot sin\left(m\phi_j\right)\right) \tag{39}$$

$$\begin{aligned}\text{CFMM:}\;\; \omega_{lm} = & \int_0^\infty \int_0^\pi \int_0^{2\pi} \chi_a\chi_b\, r^l\, \frac{P_{lm}\left(sin\left(\theta\right), cos\left(\theta\right)\right)}{(l+m)!} \\ & \cdot \left(cos\left(m\phi\right) - i \cdot sin\left(m\phi\right)\right)\, r^2 sin\left(\theta\right)\, d\phi\, d\theta\, dr\end{aligned} \tag{40}$$

The distribution $\chi_a\chi_b$ can easily be expressed in spherical coordinates.

$$\chi_a\chi_b = N_a\, N_b\, x^L\, y^M\, z^N\, e^{-(\alpha_a+\alpha_b)r^2} \tag{41}$$

$$\chi_a\chi_b = N_a\, N_b\, r^{L+M+N}\;\; sin^{L+M}\left(\theta\right)\;\; cos^N\left(\theta\right)\;\; cos^L\left(\phi\right)\;\; sin^M\left(\phi\right)\;\; e^{-(\alpha_a+\alpha_b)r^2} \tag{42}$$

The multipole moments are computed first at the positions of the charge distribution $\chi_a\chi_b$. Using the A operator a multipole expansion can be shifted to any position in space. The associated Legendre polynomials can be written as a sum over products of sine and cosine functions.

$$P_{lm}\left(sin\left(\theta\right), cos\left(\theta\right)\right) = \sum_i \sum_j c_{lm}^{ij}\;\; sin^i\left(\theta\right)\;\; cos^j\left(\theta\right) \tag{43}$$

The c_{lm}^{ij} are constants. $cos\left(m\phi\right)$ and $sin\left(m\phi\right)$ can be expanded in a similar way.

$$cos\,(m\phi) = \sum_i \sum_j g_m^{ij}\ sin^i\,(\theta)\ \ cos^j\,(\theta) \tag{44}$$

$$sin\,(m\phi) = \sum_i \sum_j h_m^{ij}\ sin^i\,(\theta)\ \ cos^j\,(\theta) \tag{45}$$

The integral (40) can be written as a product of three single integrals. Only three types of CFMM integrals remain which can easily be calculated by recursion relations. The CFMM integrals do not have any restrictions with regard to the angular momenta of the basis functions. The CFMM integrals are computed once at the beginning and the computational cost is negligible.

$$\int_0^\infty r^i\, e^{-r^2}\, r^2\, dr \tag{46}$$

$$\int_0^{2\pi} sin^i\,(\phi)\ cos^j\,(\phi)\ d\phi \tag{47}$$

$$\int_0^{\pi} sin^i\,(\theta)\ cos^j\,(\theta)\ sin\,(\theta)\ d\theta \tag{48}$$

The integrals (47) and (48) are calculated by numerically stable recursion relations. The shifting of a multipole expansion from the position of the charge distribution to the box center requires $\mathcal{O}(\mathrm{L}^4)$ work. L is the length of the multipole expansion. Usually, many of the CFMM integrals are zero and the multipole expansions at the positions of the charge distributions are sparse which reduces the scaling from $\mathcal{O}(\mathrm{L}^4)$ to $\mathcal{O}(\mathrm{L}^{1.5})$. Any zero-tasks in the translations of the multipole expansions to the box centers are skipped.

4.4 Structure of CFMM

At the beginning of a CFMM calculation all CFMM integrals are computed and stored in a four-dimensional array. The first dimension is used to store all the multipole moments for each combination of the angular momenta of x, y, and z. These multipole expansions can be shifted to any locations in space. In the second step all charge distributions which contribute very little to the Fock matrix are pruned from the CFMM tree. Usually, more than the half of all distributions can be skipped. In the CFMM approach the FMM is embedded in the outer loops over shells. In general, the sequence of the shell pairs is arbitrary. Each Fock matrix element can be computed independently of the others. The use of CFMM requires a certain sequence of the distributions to minimize the transformations on the CFMM tree. We have to ensure that the multipole expansion of a given box is transformed only once to a certain remote box. Assuming all boxes on each tree level contain at least one distribution the shell pairs of box 1 must be treated first. After box 1 has completed the shell pairs of box 2 are computed and so on. On the next lower level of the CFMM tree all children of box 1 on the parent level are computed in the order box 1 to box 8. After the child boxes of box 1 on the parent level have completed the 8 child boxes of box 2 on the parent level are treated and so on. The algorithm is applied for each tree level giving the sequence of the shell pairs on the lowest tree level. Each transformation of a multipole expansion to a Taylor expansion is done only once for a given pair of boxes.

Each box has an extension equal to the largest extension of its distributions. Two boxes can interact via multipoles if the distance between the boxes is less than or equal to the sum of the extensions of the two boxes. Because a box contains usually more than one distribution shell pair ij located in box A for example can interact via multipoles with shell pair kl of box B but not vice versa. Because index symmetry is used in the computation of the near field interaction incorrect results would occur. On the lowest level on the tree we have to split the distributions of a box in distributions which can interact via multipoles with a given distribution and which have to be computed in the near field part. The distributions of a box must be sorted according to their extensions to avoid additional computational effort. A logarithmic search algorithm is implemented to minimize the number of search steps. Only on the lowest tree level boxes are divided.

In case a box extension is to large to interact via multipoles with a given distribution the box is divided in its child boxes and the interaction is computed on the next lower level. If the lowest level was already reached the box is split and one part of the distributions must be calculated conventionally.

Because of the non-zero extents all possible values for the separation criteria can occur. The most time-consuming step within the CFMM is the transformation of the multipole to Taylor expansions as it is for the FMM. Unfortunately, we have to calculate more rotation matrices compared to FMM. The number of rotation matrices grows with the separation criterion. Nevertheless, like in the FMM implementation each rotation matrix is calculated only once and used for many transformations. The computational effort to compute the rotation matrices is negligible compared with the computation time for the transformations of the multipole expansions.

For each distribution ij the contribution to the Fock matrix element is computed for all tree levels separately.

$$F_{ij}^{C} = F_{ij}^{C} + \sum_{i=3}^{tree\ levels} \sum_{l=0}^{L} \sum_{m=-l}^{l} \omega_{lm}^{i}\ \mu_{lm}^{i} \tag{49}$$

The sum over the tree levels starts at 3 because level 3 is the first level having separated boxes. L is the length of the multipole expansions. After the computation of the far field interaction the near field contribution is still to evaluate. The CFMM routine returns a list of distributions which cannot interact with the distribution ij via multipole moments because the sum of the extensions is greater than the distance between the distributions. This list is passed to the routine which computes the two-electron integrals conventionally.

4.5 CFMM implementation in TURBOMOLE

In the DSCF routine shloop the loop structure has been changed. The original structure of the routine was as follows.

Scheme 1: Original structure of routine shloop

```
do $i$ = 1, $n$: First loop over shells
    do $j$ = 1, $i$: Second loop over shells
        do $k$ = 1, $i$: Third loop over shells
            if($k$.eq.$i$) then
                do $l$ = 1, $j$: Fourth loop over shells
                    Computation of two-electron integrals
                end do
            else
                do $l$ = 1, $k$: Fourth loop over shells
                    Computation of two-electron integrals
                end do
            endif
        end do
        Fock matrix update
    end do
end do
```

Index n is the number of shells. The most outer loops are replaced by a single loop over shell pairs. The loop over shell pairs is implemented as a loop over all occupied lowest level boxes and a second loop over the shell pairs in each of the lowest level boxes.

Scheme 2: Current structure of routine shloop

```
do ibox = 1, nboxes: Loop over all occupied lowest level boxes
    do ij = 1, number of shell pairs in box ibox: Loop over shell pairs
        call cfmm(fij, length of kl-list, kl-list): FMM
        if(length of kl-list .gt. 0) then
            do kl = 1, length of kl-list
                Computation of near field contribution
            end do
        endif
    end do
end do
```

$nboxes$ is the number of occupied lowest level boxes. The transformation of multipole expansions to Taylor expansions is done once for each separated pair of boxes. The evaluation of any multipole moments is only done if a multipole interaction has been recognized. No multipole expansions are computed in advance. No sorting of the list passed to the conventional integral calculation is necessary.

4.6 Accuracy of CFMM

We have compared several two-electron integrals computed conventionally and by the CFMM approach for normalized basis functions of s- and p-type. The shell pairs are separated by 8 a.u., the exponents are 0.5. Table 5 shows the absolute errors for a length of the multipole expansions of 6, Table 6 for a length of 10. An increase of the length of the multipole expansions by four moments decreases the absolute errors by approximately two magnitudes.

4.7 Test calculations

We have tested our CFMM implementation on two systems of industrial interest. The calculations were performed on the new IBM computer at the Research Centre Jülich equipped with Power4+ processors p690, 1.7 GHz. The first system is a cobalt catalyst consisting of 213 atoms (Figure 7). The second one is a rhodium complex consisting of 99 atoms (Figure 8). Tables (7) and (8) show the computation times and the absolute errors of the CFMM based DFT compared to conventional DFT for lengths of multipole expansions of 6 and 10.

Integral	$\lvert \mathrm{I_{CFMM}} - \mathrm{I_{conv.}} \rvert$
$<ss\|ss>$	$2.7 \cdot 10^{-6}$
$<ss\|sp>$	$2.9 \cdot 10^{-6}$
$<sp\|sp>$	$3.0 \cdot 10^{-6}$
$<ss\|pp>$	$3.2 \cdot 10^{-6}$
$<sp\|pp>$	$3.2 \cdot 10^{-6}$
$<pp\|pp>$	$3.4 \cdot 10^{-6}$

Table 5: Accuracy of the CFMM integral calculation for normalized basis functions (L = 6, Exponent = 0.5, Distance of shell pairs: 8 a.u.)

Integral	$\lvert \mathrm{I_{CFMM}} - \mathrm{I_{conv.}} \rvert$
$<ss\|ss>$	$1.9 \cdot 10^{-8}$
$<ss\|sp>$	$2.2 \cdot 10^{-8}$
$<sp\|sp>$	$2.4 \cdot 10^{-8}$
$<ss\|pp>$	$2.7 \cdot 10^{-8}$
$<sp\|pp>$	$3.0 \cdot 10^{-8}$
$<pp\|pp>$	$3.2 \cdot 10^{-8}$

Table 6: Accuracy of the CFMM integral calculation for normalized basis functions (L = 10, Exponent = 0.5, Distance of shell pairs: 8 a.u.)

Length of Multipole expansion	t[s]	$\mathrm{t_{CFMM}}$[s]	$\lvert \mathrm{E} - \mathrm{E_{CFMM}} \rvert$
6	5833	878	$4.9 \cdot 10^{-6}$
10	5833	951	$4.3 \cdot 10^{-8}$

Table 7: CPU time for routine shloop for conventional and CFMM based DFT (Cobalt catalyst)

Length of Multipole expansion	t[s]	$\mathrm{t_{CFMM}}$[s]	$\lvert \mathrm{E} - \mathrm{E_{CFMM}} \rvert$
6	4550	907	$6.3 \cdot 10^{-6}$
10	4550	971	$7.8 \cdot 10^{-8}$

Table 8: CPU time for routine shloop for conventional and CFMM based DFT (Rhodium complex)

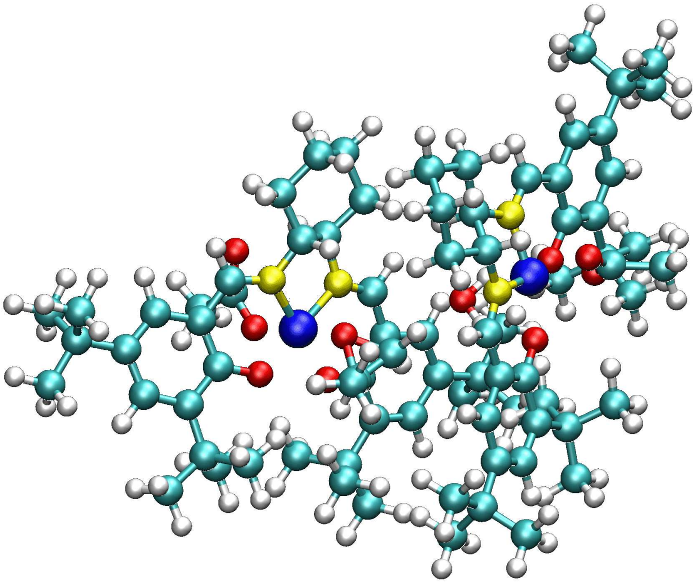


Figure 7: Cobalt catalyst: 213 atoms, 814 shells, 1683 basis functions

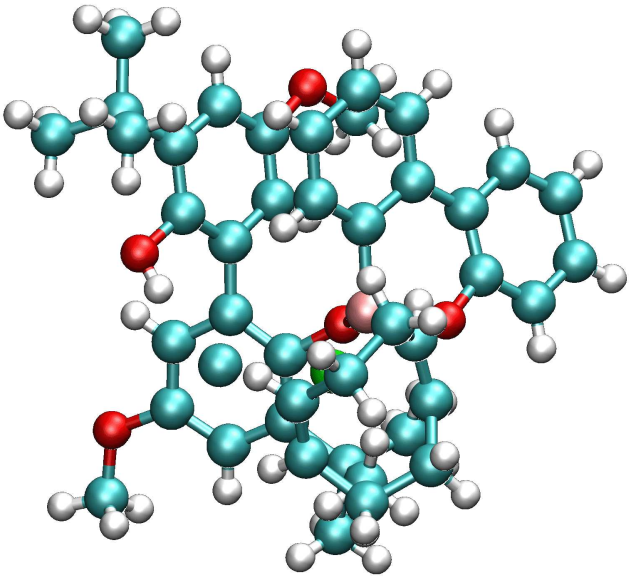


Figure 8: Rhodium complex: 99 atoms, 650 shells, 1312 basis functions

Table 8 shows timings and errors for the rhodium complex. The timings in Tables (7) and (8) are average times for one iteration. The absolute errors are the errors in the total energies after convergence has been reached.

Iteration	$\lvert\Delta\mathrm{E}\rvert$
1	$6.3 \cdot 10^{-6}$
2	$5.9 \cdot 10^{-6}$
3	$2.8 \cdot 10^{-6}$
4	$7.5 \cdot 10^{-6}$
5	$3.4 \cdot 10^{-6}$
6	$6.9 \cdot 10^{-6}$
7	$2.2 \cdot 10^{-6}$
8	$7.6 \cdot 10^{-6}$
9	$8.4 \cdot 10^{-6}$
10	$3.7 \cdot 10^{-6}$
$\vdots$	$\vdots$
30	$9.6 \cdot 10^{-6}$

Table 9: Error accumulation in the SCF iteration for an expansion length of 15 (Rhodium complex)

Iteration	$\lvert\Delta\mathrm{E}\rvert$
1	$1.2 \cdot 10^{-8}$
2	$3.7 \cdot 10^{-8}$
3	$4.9 \cdot 10^{-8}$
4	$3.6 \cdot 10^{-8}$
5	$9.7 \cdot 10^{-9}$
6	$4.5 \cdot 10^{-8}$
7	$2.8 \cdot 10^{-8}$
8	$2.2 \cdot 10^{-8}$
9	$9.6 \cdot 10^{-9}$
10	$4.3 \cdot 10^{-8}$
$\vdots$	$\vdots$
30	$3.8 \cdot 10^{-8}$

Table 10: Error accumulation in the SCF iteration for an expansion length of 25 (Rhodium complex)

The accumulation of errors in the SCF iteration has been tested for the rhodium complex. Table 9 shows the error accumulation for a multipole expansion length of 15, Table 10 for a length of 25. The energies are compared to the energies of the conventional DFT calculation. An increase of the expansion length by 10 decreases the errors by approximately three orders of magnitude.

	Its.	Energy	ΔE	CPU time per iteration [s]
CFMM-DFT	26	-2918.641427	-0.00001	923
RI-DFT	27	-2918.647531	-0.006	322

Table 11: Comparison CFMM-DFT with RI-DFT (Rhodium complex)

In comparison with RI-DFT our CFMM based DFT implementation is still a factor between two and three slower. The accuracy of the energy is about three orders of magnitude higher. Table 11 shows the timings and the energy errors for our CFMM based DFT implementation compared to RI-DFT.

5 Summary and outlook

We have described an improved implementation of the rotation based Fast Multipole Method to evaluate systems of point charges as the basis for the Continuous Fast Multipole Method to treat charge distributions. First steps in parallelizing the program have been made. Further work to improve the parallel performance is necessary.

The serial version of our FMM program is able to treat very large systems of point charges up to several billions of particles. We have proposed a new approach for the separation of near and far field within the theory of FMM to minimize the computation time depending on an user-requested threshold. Within the framework of the Blue Gene/L project our FMM program will further optimized with regard to the IBM power processor architecture.

The CFMM implementation is based on our FMM program. It is an alternative to DFT and RI-DFT. Depending on the geometry of the molecule and basis set more than 90% of all electron repulsion integrals can be computed via multipole expansions which takes approximately 15% of the total computation time. Our CFMM implementation is still at least a factor of two slower compared to RI-DFT whereas the accuracy of the total energy is about three magnitudes higher.

Acknowledgments

The author likes to thank Jürgen Wieferink and Eric Lorenz, guest students at ZAM, for their contributions to the parallelization of the FMM program.

Bibliography

[1] M. P. Allen and D. J. Tildesley, Computer Simulation of Liquids (Oxford University, Oxford, 1990).

[2] J. M. Dawson, *Rev. Mod. Phys.* **55**, 403 (1983).

[3] W. F. van Gunsteren and H. J. C. Berendsen, *Angew. Chem. Int. Ed. Engl.* **29**, 992 (1990).

[4] M. Saito, *Mol. Simul.* **8**, 321 (1992).

[5] L. Greengard and V. Rokhlin, *J. Comput. Phys.* **60**, 187 (1985).

[6] `http://www.fz-juelich.de/hpc-chem`.

[7] J. M. Ugalde and C. Sarasola, *Int. J. of Quantum* Chemistry **62**, 273 (1997).

[8] C. A. White and M. Head-Gordon, *J. Chem. Phys.* **101**, 6593 (1994).

[9] C. A. White and M. Head-Gordon, *J. Chem. Phys.* **105**, 5061 (1996).

[10] A. R. Edmonds, Angular Momentum in Quantum Mechanics (Princeton University Press, Princeton, 1957).

[11] I. J. Davis, *Computer J.* **35**, 636 (1992).

Conductor-like Screening Model COSMO

Michael Diedenhofen

COSMOLogic GmbH & Co. KG
Burscheider Str. 515, 51381 Leverkusen, Germany
E-mail: michael.diedenhofen@cosmologic.de

The treatment of solute-solvent interactions in quantum chemical calculations has become an important field, because most of the problems, which can be addressed with modern quantum chemical methods, are dealing with liquid phase chemistry. The continuum solvation models (CSMs), such as the polarizable continuum model (PCM) [1], the solvation models of Truhlar and Cramer (SMx) [2], COSMO [3], and others, have become well-established models, which take into account solvent effects on molecular energies, properties, and structures. An overview is given in the Refs. [4, 5, 6]. The following chapters will give an overview of the COSMO theory and implementations made in the HPC-Chem project.

1 Basic theory

The basic idea of the CSMs is to present the solvent by a continuum, which describes the electrostatic behavior of the solvent. The polarization of the dielectric continuum, induced by the solute, is represented by the screening charge density appearing on the boundary surface between the continuum and the solvent. Usually the exact dielectric boundary condition is used to calculate the screening charge density. The basic idea of COSMO is to replace this condition by the simpler boundary condition of the vanishing potential on the surface of a conducting medium. Using a discretization of the solute-solvent boundary surface $\mathbf{S}$ into m sufficiently small segments with center-coordinates $\mathbf{t}_s$, and the segment area S_s, this condition reads

$$\mathbf{\Phi}^{tot} = \mathbf{\Phi}^{sol} + \mathbf{\Phi}^{q} = 0. \tag{1}$$

Here the m-dimensional vector $\mathbf{\Phi}^{tot}$ denotes the total electrostatic potential on the m surface segments, which consists of the solute potential $\mathbf{\Phi}^{sol}$ (electronic and nuclear) and the potential arising from the screening charges on the segments $\mathbf{\Phi}^{q}$. The last term can be expressed by the product of the $m \times m$-dimensional Coulomb interaction matrix $\mathbf{A}$ and the m-dimensional screening charge vector $\mathbf{q}$. Then we have

$$\begin{aligned} 0 &= \mathbf{\Phi}^{sol} + \mathbf{A}\mathbf{q} && (2)\\ \mathbf{q} &= -\mathbf{A}^{-1}\mathbf{\Phi}^{sol} && (3) \end{aligned}$$

which gives an exact expression for the screening charges in a conducting continuum. The screening charges in a dielectric medium are approximated by the introduction of a scaling function that depends on the dielectric constant of the solvent:

$$\begin{aligned} \mathbf{q}^{\star} &= f(\epsilon)\mathbf{q} && (4)\\ f(\epsilon) &= \frac{\epsilon - 1}{\epsilon + \frac{1}{2}}. && (5) \end{aligned}$$

It can be shown that the relative error introduced by this approximation is very small for strong dielectrics and within 10 % for weak dielectrics and consequently within the accuracy of the dielectric continuum approach itself [3].

The interaction energy E_{int} of the solute and the continuum, i.e. the screening charges, is given by the dot product of $\mathbf{\Phi}^{sol}$ and $\mathbf{q}^{\star}$. To obtain the total dielectric energy E_{diel} one has to add the energy that is needed to create the screening charges ($\frac{1}{2}\mathbf{q}^{\star\dagger}\mathbf{\Phi}^{q}$). Using $\mathbf{\Phi}^{q} = -\mathbf{\Phi}^{sol}$ we get

$$E_{diel} = f(\epsilon)\left[\mathbf{q}^{\dagger}\mathbf{\Phi}^{sol} + \frac{1}{2}\mathbf{q}^{\dagger}\mathbf{\Phi}^{q}\right] = f(\epsilon)\left[\mathbf{q}^{\dagger}\mathbf{\Phi}^{sol} - \frac{1}{2}\mathbf{q}^{\dagger}\mathbf{\Phi}^{sol}\right] = \frac{1}{2}f(\epsilon)\mathbf{q}^{\dagger}\mathbf{\Phi}^{sol}. \tag{6}$$

As usual for linear response theory the free electrostatic energy gained by the solvation process is half of the total interaction energy. Non-electrostatic terms as for instance used in the PCM [11] or COSMO-RS [10] models, will not be discussed here.

2 Implementation in HF/KS SCF calculations

The standard implementation scheme of the COSMO model in SCF programs is given in Scheme 1. After the input parameters have been set the molecular surface $\mathbf{S}$ can be constructed, followed by the $\mathbf{A}$-matrix setup. This has to be done only once for a given molecular geometry. During the SCF cycles the current density is used for the calculation

Scheme 1: Work Schedule of a COSMO SCF Calculation

```
0)  COSMO parameter setup
1)  Cavity construction
2)  $\mathbf{A}$-matrix setup
    LOOP until SCF convergence is reached
        3)  Evaluation of the solute potential $\mathbf{\Phi}^{sol}$
        4)  Evaluation of $\mathbf{q}$ and $E_{diel}$
        5)  Computation of $E = E(\Psi^{solv}) + E_{diel}$ and insertion of the scaled screening charges $\mathbf{q}^*$into the solute Hamiltonian
    END LOOP
6)  Outlying charge correction
```

of the solute potential, which is used in step 4 to calculate the screening charges and the dielectric energy according to Eqs. (3) and (6). The scaled screening charges are introduced into the Fock or Kohn-Sham operator, respectively. The total energy is defined as the sum of the energy calculated with the solvated orbitals and the dielectric energy

$$E = E(\Psi^{solv}) + E_{diel}. \tag{7}$$

The outlying charge correction at the end of a converged SCF calculation corrects the error due to small residual solute density that reaches into the continuum.

3 Technical details

Cavity Construction: For molecular shaped cavities the efficient and sufficiently accurate segmentation of the surface is an important aspect, because it has strong influence on the accuracy and the speed of the calculation. All the cavity construction techniques define the interior of the molecule as the union of atom centered spheres (see Figure 1). The radii of this spheres can be assigned element specific, as atomic charge or electronic density depended radii [12], or by using the information about the chemical surrounding, i.e. by using atom-types also known from molecular mechanic calculations. The later definitions obviously introduce more flexibility with the potential of a more accurate reproduction, but also with the danger of a loss of predictive power of the model. Therefore, we use element specific radii. Presently optimized radii, which are adjusted to thermodynamic properties, are available for H, C, O, F, Cl, Br, I, N, and S. For other elements scaled v.d.W. radii are used. The scaling factor 1.17 is in the range of findings of other groups [1]. A second important aspect of the cavity construction is the treatment of the intersection seams of the atomic spheres. These surface areas exhibit sharp cusps that lead to unreasonable high electrostatic fields, and therefore to a physically unreasonable description and mathematical instabilities. Thus, any sphere-based construction requires a smoothing algorithm for these areas. The cavity construction in the COSMO implementations starts with a union

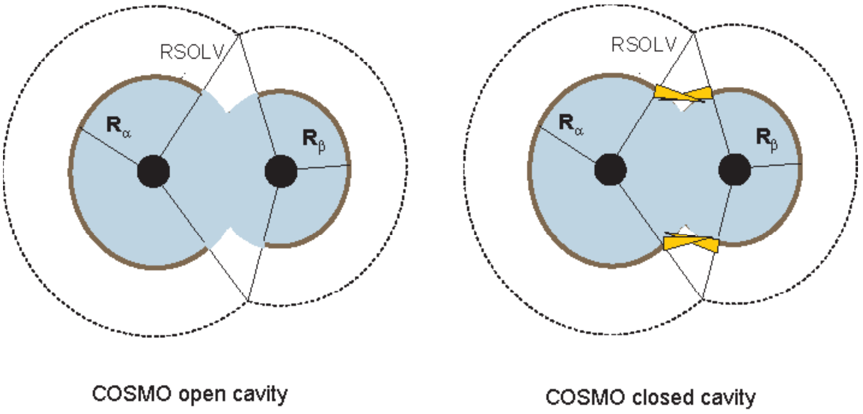


Figure 1: Illustration of the intersection smoothing methods

of spheres of radii $R_\alpha + RSOLV$ for all atoms α. The default for the auxiliary radius $RSOLV$ is the optimized H radius. The segmentation of the atomic spheres starts from a regular icosahedron with 20 triangles. A refinement of the segmentation is reached in two steps. First the triangle edges midpoints are used as new vertices and second the triangle centers are used as new vertices. The first step increases the number of triangles by a factor 4, while the subsequent step increases the number by a factor 3. In general the triangle edges can be subdivided by any integer n, leading to an increase of triangles by a factor n^2. Thus triangulations with $k = 20 \times 3^i \times n^2 (i = 0, 1)$ triangles can be generated. Eventually we do not use the triangles as segments but the corresponding hexagons and 12 pentagons. Therefore, we consider each vertex of a triangle as a center and connect the midpoints of the six or five neighbor triangles. Because the number of pentagons and hexagons is $k' = k/2 + 2$, we can construct surfaces with $k' = 10 \times 3^i \times n^2 + 2 = 12, 32, 42, 92...$ segments. This procedure has two advantages: first it reduces the number of segments and second the center-center approximation used in the $\mathbf{A}$-matrix setup is better justified for pentagons and hexagons than for triangles. In order to achieve a proper $\mathbf{A}$-matrix with a tolerable number of segments, we use a two-grid procedure. Initially a basis grid with NPPA (default: 1082) segments per non-hydrogen atom is projected onto the atomic spheres of radii $R_\alpha + RSOLV$. All the points, which are not in the interior of another sphere, are defined as remaining and projected downwards onto the radius R_α. This construction prohibits the generation of points in the problematic intersections. In the next step a segment grid of NSPH segments per H atom and NSPA segments for the other atoms is projected onto the spheres defined by R_α. NSPA (default: 92) and NSPH (default: 32) are out of the k' set. Now the remaining basis grid points are associated to the nearest segment grid centers. Segments without basis grid points are discarded. The remaining segments coordinates are redefined as the center of area of their associated basis grid points, while the segment area is the sum of the basis grid areas. In order to ensure nearest neighbor association for the new centers, this procedure is repeated once. Now the spherical part of the surface is ready and the intersection seams of the spheres have to be closed. Therefore, a ring is generated for each pair of intersecting spheres of radii $R_\alpha + RSOLV$. The parts of these rings, which do not penetrate other spheres, are projected onto the surface defined by R_α towards each of the two atom centers. The resulting two opposing rings are filled with triangles, each having two corners on one ring and one on the other. The sole corner of the triangles moves a bit towards the center of the opposing ring resulting in an inclination. The tilt angle is a function of $RSOLV$, the two atomic radii R_α and R_β and the atomic distance. At the end of the surface construction the triangular regions which arise from the intersections of three spheres, the so-called triple points, are paved with additional triangles. The ring and triple points segments are individual segments, they do not hold associated basis grid points.

A-Matrix Setup: The Coulomb interaction matrix elements A_{ij} are calculated as the sum of the contributions of the associated basis grid points of the segments i and j if their distance is below a certain threshold, the centers of the segments are used otherwise. For all segments that do not have associated basis grid points, i.e. ring and triple point segments, the segment centers are used. The diagonal elements A_{ii} that represent the self-energy of the segment are calculated via the basis grid points contributions, or by using the segment area $A_{ii} \approx 3.8\sqrt{S_i}$, if no associated basis grid points exist. Numerical instabilities can arise due to the lack of positive definiteness of the **A**-matrix, which is very often caused by matrix elements between two very close segments. In such cases the Coulomb interaction leads to an unphysical description due to a disadvantageous cavity. To avoid this problem, one has to provide proper cavities for all possible molecular structures. Because this problem is hardly solvable, we introduced the following interaction term:

$$A'_{ij} = a^s_{ij} + \frac{r_{ij}}{\bar{r}_{ij}} \left(\frac{1}{\bar{r}_{ij}} - a^s_{ij} \right) \quad \forall r_{ij} \leq \bar{r}_{ij}. \tag{8}$$

The term $a^s_{ij} \approx 2.1/\bar{r}_{ij}$ is the self-interaction of a segment with the radius $\bar{r}_{ij} = \left(r^s_i + r^s_j \right)/2$, which is the average segment radius of the two segments under consideration. If the distance of the two segments r_{ij} is less than or equal to the average segment radius $\bar{r}_{ij}$, the interaction is scaled between the self-interaction a^s_{ij} and the Coulomb interaction $1/r_{ij}$, dependent on the ratio $r_{ij}/\bar{r}_{ij}$. This procedure can also be applied to the basic grid interactions. Iterative biconjugate gradient techniques have the advantage that a positive definite matrix is not required [9] but they do not handle the physical origin of the lack of positive definiteness.

Potential Calculation and Operator Update: The full solute potential on the segments consists of an electronic and a nuclear part.

$$\mathbf{\Phi}^{sol} = \mathbf{\Phi}^{el} + \mathbf{\Phi}^{N} \tag{9}$$

In the LCAO-formalism the electronic part can be expressed in terms of potential integrals over basis functions.

$$\begin{aligned}\Phi_s^{el} &= \int \frac{\rho(\mathbf{r})}{|\mathbf{r}-\mathbf{t}_s|}d\mathbf{r} = -\sum_{\mu\nu} P_{\mu\nu}V_{\mu\nu}^s, \quad V_{\mu\nu}^s = \left\langle \mu \left| \frac{1}{|\mathbf{r}-\mathbf{t}_s|} \right| \nu \right\rangle & (10)\\ \Phi_s^N &= \sum_\alpha \frac{Z_\alpha}{|\mathbf{R}_\alpha - \mathbf{t}_s|} & (11)\end{aligned}$$

Where s is the segment index and $\mathbf{t_s}$ are center coordinates of the segment. The nuclear coordinates and charges are denoted with $\mathbf{R}_\alpha$ and Z_α, respectively. For DFT calculations it can be more efficient to calculate the electronic potential via a numerical integration of the electronic density. The COSMO update of the operator matrix can be calculated from the screening charges on the segments q_s and the potential integrals:

$$V_{\mu\nu}^{COS} = -f(\epsilon)\sum_s q_s V_{\mu\nu}^s. \quad (12)$$

If the energy is calculated using the updated operator matrix, one has to subtract the expectation value of the COSMO operator and add the dielectric energy E_{diel} in order achieve consistency with the definition in Eq. (7).

Outlying Charge Correction: The use of a cavity close to the v.d.W. surface, like in COSMO and other CSMs, implies that a significant portion of the solute electron density reaches into the continuum. This part of the charge distribution produces screening charges in the volume of the dielectric continuum and thus leads to artificial screening effects. One advantage of the COSMO approach is that it exhibits a smaller outlying charge error than models that use the electric field and the exact dielectric boundary condition. Nevertheless, the error should be corrected. Therefore, we use the double cavity approach introduced in Ref. [7]. This procedure uses a second cavity, which is constructed by an outward projection of the spherical part of the surface onto the radius $R_\alpha + ROUTF * RSOLV$ (default: $ROUTF = 0.85$). The corrected values can be calculated as follows:

$$\begin{aligned}\mathbf{\Phi}^{ot} &= \mathbf{A}^{io}\mathbf{q} + \mathbf{\Phi}^{o} & (13)\\ \mathbf{q}^{ot} &= -\mathbf{A}^{o^{-1}}\mathbf{\Phi}^{ot} & (14)\\ \mathbf{\Phi}^{c} &= -\mathbf{A}\left(\mathbf{q} \oplus \mathbf{q}^{ot}\right) & (15)\\ E^{corr} &= f(\epsilon)\frac{1}{2}\mathbf{q}^{c\dagger}\mathbf{\Phi}^{c} - E_{diel}. & (16)\end{aligned}$$

Here the $\mathbf{A}^{io}$ denotes the Coulomb interaction matrix between the charges on the inner and the charges on the outer surface, $\mathbf{\Phi}^{o}$ is the solute potential on the outer surface, and $\mathbf{A}^{o}$ is the Coulomb interaction matrix of the charges on the outer surface. The full potential on the outer surface $\mathbf{\Phi}^{ot}$ results from the outlying solute density only and is used to calculate the corresponding charge correction $\mathbf{q}^{ot}$. The fully corrected screening charge on the inner surface ($\mathbf{q} \oplus \mathbf{q}^{ot}$) can be used to calculate the corrected potential on the inner surface $\mathbf{\Phi}^{c}$. The symbol $\oplus$ denotes that a smaller array is added to the corresponding segments of $\mathbf{q}$. The energy correction E^{corr} for the dielectric energy, and thus also for the total energy, is defined as the difference between the corrected and the uncorrected dielectric energy.

Gradients: From the definition of the total energy in HF/KS-SCF calculations in Eq. (7) it is clear that the gradient consists of the SCF gradient, calculated from the solvated wave function, and the derivative of the dielectric energy.

$$\begin{aligned} E^{\xi} &= E(\Psi^{solv})^{\xi} + E^{\xi}_{diel} && (17) \\ E^{\xi}_{diel} &= f(\epsilon)\left[\frac{1}{2}\mathbf{q}^{\dagger}\mathbf{A}^{\xi}\mathbf{q} + \mathbf{q}^{\dagger}\mathbf{\Phi}^{sol\xi}\right] && (18) \\ \mathbf{\Phi}^{sol\xi} &= \mathbf{\Phi}^{el\xi} + \mathbf{\Phi}^{N\xi} && (19) \end{aligned}$$

The first term in the derivative of the dielectric energy can be calculated easily using the already known screening charges and the derivative of the $\mathbf{A}$-matrix. The derivative $\mathbf{A}^{\xi}$ includes an estimate for the surface derivative, which has to be taken into account for the diagonal elements. The solute potential derivative splits into the nuclear $\mathbf{\Phi}^{N\xi}$ and the electronic part $\mathbf{\Phi}^{el\xi}$. The first term can be computed by COSMO routines, whereas the integral derivatives needed for the second term have to be provided by the quantum chemical code.

$$\Phi^{el\xi}_{s} = -\sum_{\mu\nu} P_{\mu\nu} \left\langle \mu \left| \frac{1}{|\mathbf{r} - \mathbf{t}_s|} \right| \nu \right\rangle^{\xi} \qquad (20)$$

Like the potential itself, the derivative is known in common quantum chemical codes, because it is similar to the nuclear-electron attraction integral derivative. To ensure a fast gradient calculation the segment center approximation is used during the whole gradient calculation. It should be noted that numerical derivatives of the energy should not be calculated with the COSMO model, because due to the cavity construction mechanism the energy is not continuous.

4 Frequency calculation

The calculation of harmonic frequencies raises the problem of non-equilibrium solvation in the CSM framework. In the case of long-living states of the solute, the solvent is able to respond with its full re-orientational and electronic polarization. But processes that are on time scales that do not allow a re-orientation of the solvent molecules, such as electronic excitations or molecular vibrations for instance, have to be treated as non-equilibrium processes. Therefore, the total response of the continuum is split into a fast contribution, described by the electronic polarization, and a slow term related to the orientational relaxation. The partition depends on the susceptibility of the solvent, which can be written as the sum of the electronic and the orientational part

$$\chi_{tot} = \chi_{el} + \chi_{or}; \quad \chi_{el} = n^2 - 1; \quad \chi_{or} = \epsilon - n^2 \tag{21}$$

where n is the refractive index of the solvent. For the initial state, which is characterized by the density $\mathbf{P}^0$ and the dielectric constant ϵ, the response of the solvent, i.e. the screening charges, split into an orientational part and an electronic part:

$$\mathbf{q}^{\star,or}(\mathbf{P}^0) = \frac{\chi_{or}}{\chi_{tot}} f(\epsilon)\mathbf{q}(\mathbf{P}^0); \quad \mathbf{q}^{\star,el}(\mathbf{P}^0) = \frac{\chi_{el}}{\chi_{tot}} f(\epsilon)\mathbf{q}(\mathbf{P}^0). \tag{22}$$

During fast processes the orientational part is kept fixed while the electronic part is allowed to respond instantaneously to the disturbance. For an arbitrary disturbed state with the density $\mathbf{P} = \mathbf{P}^0 + \mathbf{P}^\Delta$ the total potential reads:

$$\mathbf{\Phi}' = \mathbf{\Phi}(\mathbf{P}) + \mathbf{A}\mathbf{q}^{\star,or}. \tag{23}$$

Where $\mathbf{A}\mathbf{q}^{\star,or}$ is the negative potential arising from the frozen initial state screening charges $\mathbf{q}^{\star,or}$. The full potential is screened by the electronic polarizability only and thus the dielectric constant in Eq. (5) has to be replaced by the square of the refractive index n^2. The electronic response contribution to the screening charges of the disturbed state can be obtained from:

$$\mathbf{q}'^{\star} = -f(n^2)\mathbf{A}^{-1}\mathbf{\Phi}' \tag{24}$$

After adding the frozen charges $\mathbf{q}^{\star,or}(\mathbf{P}^0)$ and some re-arrangements one obtains a simple expression for the total scaled screening charge of the disturbed state.

$$\mathbf{q}^{d,\star} = f(n^2)\mathbf{q}(\mathbf{P}^\Delta) + f(\epsilon)\mathbf{q}(\mathbf{P}^0) \tag{25}$$

As can be shown [8] the dielectric energy for the disturbed state can be written as follows:

$$E^d_{diel} = \frac{1}{2} f(\epsilon)\mathbf{q}(\mathbf{P}^0)\mathbf{\Phi}(\mathbf{P}^0) + \frac{1}{2} f(n^2)\mathbf{q}(\mathbf{P}^\Delta)\mathbf{\Phi}(\mathbf{P}^\Delta) + f(\epsilon)\mathbf{q}(\mathbf{P}^0)\mathbf{\Phi}(\mathbf{P}^\Delta). \tag{26}$$

The interaction is composed of three contributions: the initial state dielectric energy, the interaction of the potential difference with the initial state charges, and the the electronic screening energy that results from the density difference.

Using this theory we developed an implementation scheme for the calculation of numerical frequencies by numerical differentiation of the analytical gradients, which is given in Scheme 2. In opposition to excited states, which can be treated with one cavity and the corresponding $\mathbf{A}$-matrix, the distortions of the numerical procedure change the COSMO cavity at every step. In an early implementation of numerical PCM frequencies [18] a fixed

Scheme 2: Work Schedule of a Numerical Frequency Calculation with COSMO

```
0)  Do a standard COSMO calculation and save the screening charges and potentials as $\mathbf{q}(\mathbf{P}^0)$ and $\mathbf{\Phi}(\mathbf{P}^0)$
    LOOP over distorted structures
        1)  Set up the cavity and the $\mathbf{A}$-matrix for the distorted geometry
        2)  Map the frozen potential on the new cavity $\mathbf{\Phi}(\mathbf{P}^0) \rightarrow \mathbf{\Phi}^m(\mathbf{P}^0)$
            and recalculate the screening charges $\mathbf{q}^m(\mathbf{P}^0) = -\mathbf{A}^{-1}\mathbf{\Phi}^m(\mathbf{P}^0)$
        LOOP until SCF convergence is reached
            3)  Calculate the current $\mathbf{\Phi}(\mathbf{P})$ and $\mathbf{q}(\mathbf{P})$ and build $\mathbf{\Phi}(\mathbf{P}^\Delta) = \mathbf{\Phi}(\mathbf{P}) - \mathbf{\Phi}^m(\mathbf{P}^0)$ and $\mathbf{q}(\mathbf{P}^\Delta) = \mathbf{q}(\mathbf{P}) - \mathbf{q}^m(\mathbf{P}^0)$
            4)  Calculate the dielectric energy according to Eq. (26) using the mapped values for the initial state charges and potentials
            5)  Calculate $\mathbf{q}^{d\star}$ from Eq. (25) using the mapped initial state potentials and insert the charges into the Hamiltonian
            6)  Calculate the new density and the corresponding energy: $E = E(\Psi^{solv}) + E^d_{diel}$
        END LOOP
        7)  Calculate the gradient according to Eq. (27)
    END LOOP
8)  Calculate the numerical derivate of the gradient
```

cavity approach and a density adjusted cavity were examined. It turned out that the fixed cavity gave reasonable results for diatomic HF molecule, but cannot be expected to be applicable to polyatomic molecules. To solve the cavity problem we map the initial state potential on the cavity of the disturbed state and recalculate the screening charges from the new potential. The mapped potential of a segment of the new cavity is calculated from the distance-weighted potentials of all segments of the old cavity that fulfill a certain distance criterion. This procedure should be more stable than a direct mapping of the screening charges. The gradient of the distorted states can be derived from Eq. (26) and the fact that the gradients of the frozen initial state values $\mathbf{q}(\mathbf{P}^0)$ and $\mathbf{\Phi}(\mathbf{P}^0)$ vanish.

$$E^{d,\xi}_{diel} = f(n^2)\left[\frac{1}{2}\left(\mathbf{q}^\dagger(\mathbf{P}^\Delta)\mathbf{A}^\xi\mathbf{q}(\mathbf{P}^\Delta)\right) + \left(\mathbf{q}(\mathbf{P}^\Delta) + \frac{f(\epsilon)}{f(n^2)}\mathbf{q}(\mathbf{P}^0)\right)^\dagger \mathbf{\Phi}^\xi(\mathbf{P})\right] \qquad (27)$$

Numerical results for the ν(C=O) frequency of 4-(Dimethylamino)-benzaldehyde in six different solvents are given in Table 1. The deviations from the experimental data show that the implementation of Scheme 2 leads to a small improvement of the calculated frequencies compared to the fully relaxed COSMO calculations (V5.5u). Nevertheless, the fully relaxed COSMO frequencies exhibit the same trends and seem to be a good approximation.

Table 1: Solvent Effects on the ν(C=O) Frequencies[d] [cm^{-1}] of 4-(Dimethylamino)-benzaldehyde on the RI-DFT BP/TZVP Level.

Solvent	ϵ	η_D^{20}	Exp.[a]	V5.5u[b]	V5.5u-Exp.	new[c]	new-Exp.
CCl_4	2.23	1.46	1683.0	1662.3	-20.7	1664.7	-18.3
Benzene	2.25	1.501	1677.8	1661.6	-16.2	1664.0	-13.8
Chloroform	4.9	1.446	1662.2	1644.4	-17.8	1652.7	-9.5
Ethanol	24.6	1.361	1658.2	1628.7	-29.5	1644.0	-14.3
Methanol	32.6	1.323	1657.4	1628.1	-29.3	1643.5	-13.9
Acetonitrile	36.6	1.344	1673.6	1628.0	-45.6	1643.3	-30.3
Vacuum				1685.6			

[a] from Ref. [19]. [b] TURBOMOLE version 5.5 using uncorrected screening charges for the gradient calculation. [c] Implementation of Scheme 2 in TURBOMOLE version 5.6. [d] The calculated values are numerical harmonic, un-scaled frequencies (SCF conv. 8, step size 0.02 a.u.). The molecular structures have been optimized for the given dielectric constant.

5 COSMO at the MP2 level

For ab initio MP2 calculations within the CSM framework three alternatives, originally introduced by Olivares et al. [15], can be found in the literature. The first approach, often referred to as PTE, performs a normal MP2 energy calculation on the solvated HF wave function. The response of the solvent, also called reaction field, is still on the HF level. In the so-called PTD approach the vacuum MP2 density is used to calculate the reaction field. The third approach, often called PTED, is iterative so that the reaction field reflects the density of the first-order wave function. In contrast to the PTE approach the reaction field, i.e. the screening charges, change during the iterations until self consistency is reached. This is important if the screening charges are used afterwards e.g. as input for a COSMO-RS [10] calculation, which allows the prediction of thermodynamic properties of solutions. The PTE algorithm is less cumbersome than the PTED and suited for the analytical gradient calculations. Furthermore, it was shown by Ángyán that PET is formally consistent in the sense of second-order perturbation theory [13, 14]. In this project we implemented the PTED method given in Scheme 3. The MP2 density in step 1 of Scheme 3 is the relaxed density, which can be obtained from a coupled perturbed Hartree-Fock (CPHF) procedure for solvated systems. Such a procedure has been proposed for the PCM model by Cammi et al. [17]. The authors gave a full implementation of analytical gradients for a PTE like MP2 scheme. To adopt this procedure for the COSMO model, the screening charges have to be divided with respect to the two potential components they originate from. Thus, we obtain charges, which arise from the electronic potential and the charges that originate from the nuclear potential denoted by the superscript el and N, respectively.

$$q_s^x = -\sum_t A_{st}^{-1}\Phi_t^x, \quad x = N, el. \tag{28}$$

Scheme 3: PTED Implementation Scheme

```
0)  Do a standard COSMO HF calculation
    LOOP until convergence is reached for $\mathbf{q}^{MP2}$ and $E^{MP2}$
        1)  Calculate the MP2 energy and density
        2)  Calculate new screening charges and dielectric energy from
            the MP2 density according to Eq. (3)
        3)  Calculate $E^{MP2}_{solv}$ as defined in Eq. (34)
        4)  Perform a HF COSMO calculations with frozen $\mathbf{q}^{MP2}$
    END LOOP
5)  Perform the outlying charge correction for the MP2 density
```

Here the indices s, t refer to the segments of the COSMO cavity. Introducing this definition in Eq. (12) the COSMO part of the Fock matrix reads:

$$V^{COS}_{\mu\nu} = V^{COS,N}_{\mu\nu} + V^{COS,el}_{\mu\nu} \tag{29}$$

$$V^{COS,x}_{\mu\nu} = f(\epsilon)\sum_{s}\sum_{t} A^{-1}_{st}\Phi^{x}_{t}V^{s}_{\mu\nu}. \tag{30}$$

Using the expression for the electronic potential from Eq. (11) $V^{COS,el}_{\mu\nu}$ can be written as follows:

$$V^{COS,el}_{\mu\nu} = f(\epsilon)\sum_{\lambda\sigma} P_{\lambda\sigma}\sum_{s} q^{s}_{\lambda\sigma}V^{s}_{\mu\nu} = \sum_{\lambda\sigma} P_{\lambda\sigma}\tilde{q}_{\mu\nu,\lambda\sigma} \tag{31}$$

$$q^{s}_{\lambda\sigma} = -\sum_{t} A^{-1}_{st}V^{t}_{\lambda\sigma}. \tag{32}$$

In this definition $\tilde{q}_{\mu\nu,\lambda\sigma}$ is the contribution due to the screening charge portion arising from the charge distribution of the AO pair $\chi^{\star}_{\lambda}\chi_{\sigma}$. The two COSMO parts derived above can be added to the one-electron and the two-electron part of the Fock matrix, respectively. The solvated Fock matrix elements read:

$$F^{solv}_{\mu\nu} = \left(h_{\mu\nu} + V^{COS,N}_{\mu\nu}\right) + \sum_{\lambda\sigma} P_{\lambda\sigma}\left(\langle\mu\nu\,||\,\lambda\sigma\rangle + \tilde{q}_{\mu\nu,\lambda\sigma}\right) \tag{33}$$

Using this operator in the CPHF procedure one obtains a Z-vector equation where both sides include $\tilde{q}_{\mu\nu,\lambda\sigma}$ contracted with the current density, which take into account the response of the continuum due to the perturbed density [17]. In a non-iterative PTE calculation the relaxed MP2 density can be used to compute any one-electron property. As can be seen from Scheme 3 this is simply the first step of the PTED implementation. The last steps of the PTED scheme are straightforward. The MP2 screening charges are calculated from the full MP2 potential arising from the MP2 density and the nuclei. These charges are used to calculate the MP2 dielectric energy that is part of the full MP2 energy expression of the solvated system

$$E^{MP2}_{solv} = E^{HF} + E^{(2)} - E^{HF}_{diel} + E^{MP2}_{diel} \tag{34}$$

Table 2: MP2 Solvation Energies [kcal/mol] in a Conductor ($f(\epsilon)$ =1) at the RI-MP2/TZVP//RI-BP/TZVP[a] Level

	iter.[d]	PTED-PTE	PTED-PTED0[b]	$\Delta E^{solv}(MP2)^c$	$\Delta E^{solv}(HF)^c$
C_2H_5OH	4	0.51	-0.04	-5.49	-6.55
CF_3COO^-	4	1.41	-0.32	-59.83	-63.43
CH_3OH	3	0.48	-0.03	-5.77	-6.75
$CHCl_3$	4	0.75	-0.12	-2.20	-3.86
Chlorobenzene	4	0.87	-0.14	-2.39	-4.33
Cl_2	2	0.06	0.00	-0.90	-1.02
ClO_4^-	4	0.51	-0.22	-58.73	-60.28
CO_2	4	1.18	-0.39	-0.66	-3.89
Cyclopropane	2	-0.04	0.00	-1.56	-1.48
Diethylether	4	0.47	-0.05	-3.31	-4.29
DMSO	6	3.33	-0.35	-9.00	-16.16
Ethylamine	3	0.16	-0.01	-5.13	-5.46
Ethylene	3	0.34	-0.05	-1.13	-1.89
HCOOH	6	1.99	-0.57	-5.16	-10.20
Furane	7	3.00	-1.26	-2.35	-10.50
H_2O	3	0.45	-0.02	-8.00	-8.92
H_3O^+	3	0.05	-0.01	-93.96	-94.08
I_2	2	0.23	-0.02	-1.80	-2.30
Methylamine	3	0.09	-0.01	-5.18	-5.39
NH_3	3	0.13	-0.01	-6.18	-6.46
Nitrobenzene	6	2.53	-0.90	-2.93	-9.70
NO_3^-	3	0.47	-0.13	-66.45	-67.71
OH^-	3	0.92	-0.01	-95.85	-97.70
PF_6^-	3	-0.03	-0.04	-55.53	-55.58
Phenol	5	0.98	-0.14	-6.20	-8.39
SO_4^{2-}	4	0.44	-0.24	-238.25	-239.80

[a] COSMO optimizations. [b] PTED0 is the first cycle of the PTED scheme including the outlying charge correction. [c] Solvation energy $\Delta E^{solv}(X) = E^{COSMO}(X) - E^{gas}(X), \quad X = HF, PTED.$ [d] Convergence criteria: energy 10^{-6}; screening charges 10^{-5} (maximum) and 10^{-6} (rms).

where $E^{(2)}$ denotes the second order perturbation energy of the solvated HF wave function. If the energy and the screening charges are not converged, a HF calculation in the presence of the fixed MP2 screening charges provides the wave function for the next cycle. Some results obtained with the PTED approach are given in Table 2. The results are similar to that reported from an earlier PTED implementation in the program package GAMESS [16]. As can bee seen from the energy difference between the first cycle (PTED0) and the converged PTED, the relative small improvement of the PTED0 energies does not justify the iterative procedure. Only four compounds exhibit absolute improvements that exceed 0.5 kcal/mol. The largest effects have been found for Furane and Nitrobenzene. We used

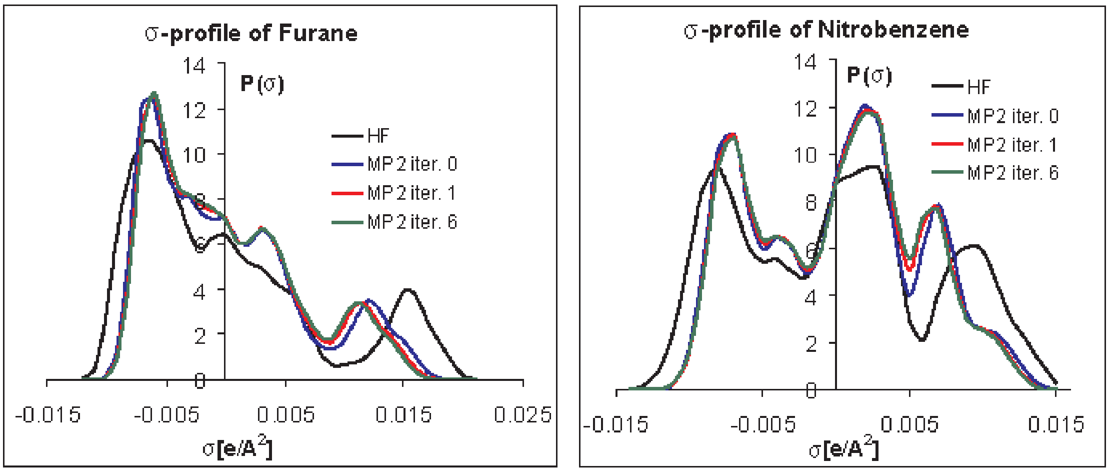


Figure 2: σ-profiles of Furane and Nitrobenzene.

this compounds to check the influence of the iterative procedure on the screening charges. The plots in Figure 2 show the distributions of the screening charge densities, the so-called σ-profiles, for the different iterations. The graphs show substantial differences between the HF profiles and the profiles of the first PTED iteration, whereas no significant changes can be observed among the MP2 profiles.

6 Implementation in TURBOMOLE

The starting point for the implementations of this project was the already existing COSMO SCF and gradient implementation in the programs `dscf` and `grad` and the corresponding RI-DFT routines `ridft` and `rdgrad` [20]. This implementation was done by the BASF and used extensively in industries and academics. The higher-level parametrizations of the COSMO-RS model, which became an important tool for the prediction of thermodynamic data of solutions, have been done with these programs [10]. Although this first implementation turned out to be very stable, the lack of positive definiteness of the **A**-matrix, occurring in the Cholesky factorization, has been reported by some users. Thus, we started this project with the analysis of the **A**-matrix problem. As discussed in section 3, it turned out, that in most of the cases the problem was caused by surface segments located in the intersection seams of the atomic spheres. In some cases the seam filling procedure leads to small segment-segment distances and thus to huge Coulomb potentials. To overcome this problem, we introduced the modified interaction term given in Eq. (8) in the **A**-matrix setup routine. This modification leads to a more stable procedure. A second modification of the basic implementation was a small change in the gradient routines. Since the outlying charge correction is not included in the gradient, the uncorrected screening charges will be used in the gradient calculation in future versions.

MP2 Cosmo Calculations: The PTED scheme (cf. section 5) has been implemented for the conventional MP2 and the RI-MP2 method. Because the TURBOMOLE package consists of stand-alone programs, we used a transfer file that contains the needed COSMO information e.g. coordinates of the surface segments and screening charges of a certain level of theory. Keywords in the control file are used to activate the features in the HF program and the MP2 modules needed for the iterative procedure. A shell script is used to manage the keyword settings and program calls needed for the PTED calculation. The user can choose the convergence criteria for the energy and the screening charges and the maximum number of iterations. A restart of the iterative procedure is possible. The COSMO related potential has been added in the CPHF procedure, i.e., to the Z-vector equation. In the case of the non-iterative PTE procedure, which is simply the first step of the PTED approach, this leads to a consistent MP2 density that can be used for the calculation of one-electron properties. The density has been checked using dipole moments calculated by numerical derivatives. Some results obtained with the described implementation have been given and discussed in section 5. PTED0 results on the RI-MP2/TZVP//RI-BP/TZVP level of 331 molecules have been used to optimize and validate the parameter set of the COSMO-RS model. It turned out that the new MP2 parametrization is inferior to the DFT parametrization BP_TZVP_C12_0104 [21]. The free energy root mean square deviation over all properties used in the optimization is 0.56 kcal/mol in the MP2 case and 0.39 kcal/mol for the DFT parametrization. Nevertheless, the results may be improved by the use of the fully optimized PTED results, a better basis set, or MP2 optimized structures. Further work has to be done to check these opportunities.

Cosmo Frequency Calculations: The implementation scheme for the calculation of numerical frequencies including the COSMO model, given in section 4, has been implemented for the RI-DFT method. The programs `ridft` and `rdgrad` that perform the SCF and gradient calculations, respectively, have been modified in order to enable the treatment of the different charge and potential contributions and to read/write the COSMO data file that is needed for the data transfer between the stand-alone modules. The shell script `NumForce` that controls the numerical frequency calculation has been extended to handle new COSMO keywords. First results have been given in section 4.

7 Implementation in MOLPRO

The present implementation of COSMO in MOLPRO was the first realization of a CSM in the MOLPRO code. Therefore, we started with the implementation of COSMO in the SCF routines of the program following Scheme 1. This has been done for HF/UHF and KS/UKS calculations using the modified **A**-matrix setup discussed in the sections 6 and 3. During this implementation we tried to introduce symmetry into the COSMO routines. Since the cavity construction described in section 3 cannot be modified for the use of symmetry in a straightforward way, we tried to symmetrize the C_1 cavity. The segments which belong to the irreducible representation and their associated basis grid points are replicated when the symmetry equivalent segments or basis grid points are required for the calculation of the **A**-matrix elements. After the **A**-matrix is set up for the irreducible representation, at the beginning of the calculation, only segments that belong to the irreducible representation have to be taken into account in the next steps. The drawback of this procedure is the symmetry dependence of the energy. If a high symmetry is used for a small molecule the deviation from the C_1 energy can be substantial. Furthermore, the segment distribution on the symmetrized cavities is worse than in the C_1 case. Because the edge segments of the cut out irreducible representation do not lie on the mirror planes, the replication can generate very small segment distances or holes. After some tests it turned out that the simpler approach of using the full C_1 cavity also for higher symmetries leads to a more stable procedure with negligible energy deviations between C_1 and higher symmetries. In the next step the COSMO gradient given in Eq. (18) has been build in using the uncorrected screening charges. The MP2 COSMO calculations have been implemented similar to the TURBOMOLE package as described in section 5. Because MOLPRO is able to handle control structures and variables in the input, the first version of the iterative PTED procedure has been implemented using a loop in the input file. The results match the data obtained with the conventional TURBOMOLE MP2 (`mpgrad`) and they are close to the RI-MP2 results presented in Table 2.

8 Implementation in QUICKSTEP

The COSMO implementation in Gaussian plane wave (GPW) codes like QUICKSTEP follows the procedure given in Scheme 1. The substantial differences to the implementation in pure LCAO codes, which has been discussed in section 3, are the interfaces to the quantum chemical program, i.e. the calculation of the solute potential on the surface segments and the COSMO contribution to the Kohn-Sham (KS) matrix.

Following the QUICKSTEP philosophy we decided to keep the changes due to the COSMO implementation, especially the COSMO data structures, as local as possible. The COSMO routines have been combined in a Fortran 90 module definition. All the relevant COSMO data are kept inside the module. The Fortran type `COSMO_DATA` holds all COSMO parameters like the radii and the dielectric constant. A set of get- and set-functions is used to access the data from the QM program. A second Fortran type called `COSMO_SEGMENT_DATA` is used to store the segment related data, e.g. the screening charges and the potentials on the surface. Because these data change during the SCF calculation, they are directly accessible from both sides. The needed memory is allocated internally whenever needed. A cleaning routine at the end of the calculation frees all allocated memory. The calls of the COSMO module routines from the SCF program are given in Scheme 4.

Scheme 4: Implementation of the COSMO Module in a SCF Program

```
cosmo_initialize        Initialize data of the cosmo_data type, set default parameter and radii. User defined parameters can be set using the set routines after the initialization
cosmo_check_rad         Check if all radii have been set. Only optimized radii for the most common elements are set in the initialization
cosmo_surf_amat         Calculate the surface and the A matrix
    LOOP until SCF convergence is reached
        Provide the solute potential
        cosmo_charges   Calculate the screening charges

        cosmo_ediel     Calculate the dielectric energy
        Update KS matrix and calculate the energy according to Eq. (7)
    END LOOP
Provide the solute potential on the outer surface
cosmo_oc_corr           Perform the outlying charge correction
cosmo_write             Write the corrected values to the output file
cosmo_clean             Deallocate the internally allocated memory
```

Bibliography

[1] S. Miertus, E Scrocco, and J. Tomasi, *Chem. Phys.* **55**, 117 (1981).

[2] C. J. Cramer and D. G. Truhlar, *J. Am. Chem. Soc.* **113**, 8305 (1991); **113**, 9901E (1991).

[3] a) A. Klamt and G. Schüürmann, *J. Chem. Soc. Perkin Trans.* **2**, 799 (1993).
b) A. Klamt, in The Encyclopedia of Computational Chemistry, edited by P. v. R. Schleyer, John Wiley & Sons: Chichester (1999).

[4] J. Tomasi and M. Persico, *Chem. Rev.* **94**, 2027 (1994).

[5] C. J. Cramer and D. G. Truhlar, *Chem. Rev.* **99**, 2161 (1999).

[6] F. J. Luque, J. M. López, and J. M. Orozco, *Theor. Chem. Acc.* **103**, 343 (2000).

[7] A. Klamt and V. Jonas, *J. Chem. Phys.* **105** (22), 9972 (1996).

[8] A. Klamt, *J. Chem. Phys.* **100**, 3349 (1996).

[9] C. C. Pye and T. Ziegler, *Theor. Chem. Acc.* **101**, 396 (1999).

[10] a) A. Klamt, *J. Phys. Chem.* **99**, 2224 (1995).
b) A. Klamt, V. Jonas, T. Bürger, and J. C. W. Lohrenz, *J. Phys. Chem. A* **102**, 5074 (1998).
c) F. Eckert and A. Klamt, *AICHE Journal* **48**, 369 (2002).

[11] J. Tomasi, R. Cammi, and B. Mennucci, *Int. J. Quantum Chem.* **75**, 783 (1999).

[12] J. B. Foresman, T. A. Keith, K. B. Wiberg, J. Snoonian, and M. J. Frisch, *J. Phys. Chem.* **100**, 16098 (1996).

[13] J. G. Ángyán, *Int. J. Quantum Chem.* **47**, 469 (1993).

[14] J. G. Ángyán, *Chem. Phys. Let.* **241**, 51 (1995).

[15] F. J. Olivares del Valle and J. Tomasi, *Chem. Phys.* **150**, 139 (1991).

[16] K. K. Baldridge and V. Jonas, *J. Chem. Phys.* **113**, 7511 (2000).

[17] R. Cammi, B. Mennucci, and J. Tomasi, *J. Phys. Chem. A* **103**, 9100 (1999).

[18] a) F. J. Olivares del Valle and J. Tomasi, *Chem. Phys.* **114**, 231 (1987).
b) F. J. Olivares del Valle, M. Aguilar, S. Tolosa, J. C. Contador, and J. Tomasi, *Chem. Phys.* **143**, 371 (1990).

[19] Y. M. Jung, J. S. Kang, S. H. Seo, and M. S. Lee, *Bull. Korean Chem. Soc.* **17**, 128 (1996).

[20] A. Schäfer, A. Klamt, D. Sattel, J. C. W. Lohrenz, and F. Eckert, *Phys. Chem. Chem. Phys.* **2**, 2187 (2000).

[21] `http://www.cosmologic.de`.

Already published:

Modern Methods and Algorithms of Quantum Chemistry - Proceedings
Johannes Grotendorst (Editor)
Winter School, 21 - 25 February 2000, Forschungszentrum Jülich
NIC Series Volume 1
ISBN 3-00-005618-1, February 2000, 562 pages
out of print

Modern Methods and Algorithms of Quantum Chemistry - Poster Presentations
Johannes Grotendorst (Editor)
Winter School, 21 - 25 February 2000, Forschungszentrum Jülich
NIC Series Volume 2
ISBN 3-00-005746-3, February 2000, 77 pages
out of print

Modern Methods and Algorithms of Quantum Chemistry - Proceedings, Second Edition
Johannes Grotendorst (Editor)
Winter School, 21 - 25 February 2000, Forschungszentrum Jülich
NIC Series Volume 3
ISBN 3-00-005834-6, December 2000, 638 pages

Nichtlineare Analyse raum-zeitlicher Aspekte der hirnelektrischen Aktivität von Epilepsiepatienten
Jochen Arnold
NIC Series Volume 4
ISBN 3-00-006221-1, September 2000, 120 pages

Elektron-Elektron-Wechselwirkung in Halbleitern: Von hochkorrelierten kohärenten Anfangszuständen zu inkohärentem Transport
Reinhold Lövenich
NIC Series Volume 5
ISBN 3-00-006329-3, August 2000, 146 pages

Erkennung von Nichtlinearitäten und wechselseitigen Abhängigkeiten in Zeitreihen
Andreas Schmitz
NIC Series Volume 6
ISBN 3-00-007871-1, May 2001, 142 pages

Multiparadigm Programming with Object-Oriented Languages - Proceedings
Kei Davis, Yannis Smaragdakis, Jörg Striegnitz (Editors)
Workshop MPOOL, 18 May 2001, Budapest
NIC Series Volume 7
ISBN 3-00-007968-8, June 2001, 160 pages

Europhysics Conference on Computational Physics - Book of Abstracts
Friedel Hossfeld, Kurt Binder (Editors)
Conference, 5 - 8 September 2001, Aachen
NIC Series Volume 8
ISBN 3-00-008236-0, September 2001, 500 pages

NIC Symposium 2001 - Proceedings
Horst Rollnik, Dietrich Wolf (Editors)
Symposium, 5 - 6 December 2001, Forschungszentrum Jülich
NIC Series Volume 9
ISBN 3-00-009055-X, May 2002, 514 pages

Quantum Simulations of Complex Many-Body Systems: From Theory to Algorithms - Lecture Notes
Johannes Grotendorst, Dominik Marx, Alejandro Muramatsu (Editors)
Winter School, 25 February - 1 March 2002, Rolduc Conference Centre, Kerkrade, The Netherlands
NIC Series Volume 10
ISBN 3-00-009057-6, February 2002, 548 pages

Quantum Simulations of Complex Many-Body Systems: From Theory to Algorithms- Poster Presentations
Johannes Grotendorst, Dominik Marx, Alejandro Muramatsu (Editors)
Winter School, 25 February - 1 March 2002, Rolduc Conference Centre, Kerkrade, The Netherlands
NIC Series Volume 11
ISBN 3-00-009058-4, February 2002, 194 pages

Strongly Disordered Quantum Spin Systems in Low Dimensions: Numerical Study of Spin Chains, Spin Ladders and Two-Dimensional Systems
Yu-cheng Lin
NIC Series Volume 12
ISBN 3-00-009056-8, May 2002, 146 pages

Multiparadigm Programming with Object-Oriented Languages - Proceedings
Jörg Striegnitz, Kei Davis, Yannis Smaragdakis (Editors)
Workshop MPOOL 2002, 11 June 2002, Malaga
NIC Series Volume 13
ISBN 3-00-009099-1, June 2002, 132 pages

**Quantum Simulations of Complex Many-Body Systems:
From Theory to Algorithms - Audio-Visual Lecture Notes**
Johannes Grotendorst, Dominik Marx, Alejandro Muramatsu (Editors)
Winter School, 25 February - 1 March 2002, Rolduc Conference Centre,
Kerkrade, The Netherlands
NIC Series Volume 14
ISBN 3-00-010000-8, November 2002, DVD

Numerical Methods for Limit and Shakedown Analysis
Manfred Staat, Michael Heitzer (Eds.)
NIC Series Volume 15
ISBN 3-00-010001-6, February 2003, 306 pages

**Design and Evaluation of a Bandwidth Broker that Provides
Network Quality of Service for Grid Applications**
Volker Sander
NIC Series Volume 16
ISBN 3-00-010002-4, February 2003, 208 pages

**Automatic Performance Analysis on Parallel Computers with
SMP Nodes**
Felix Wolf
NIC Series Volume 17
ISBN 3-00-010003-2, February 2003, 168 pages

**Haptisches Rendern zum Einpassen von hochaufgelösten
Molekülstrukturdaten in niedrigaufgelöste
Elektronenmikroskopie-Dichteverteilungen**
Stefan Birmanns
NIC Series Volume 18
ISBN 3-00-010004-0, September 2003, 178 pages

Auswirkungen der Virtualisierung auf den IT-Betrieb
Wolfgang Gürich (Editor)
GI Conference, 4 - 5 November 2003, Forschungszentrum Jülich
NIC Series Volume 19
ISBN 3-00-009100-9, October 2003, 126 pages

NIC Symposium 2004
Dietrich Wolf, Gernot Münster, Manfred Kremer (Editors)
Symposium, 17 - 18 February 2004, Forschungszentrum Jülich
NIC Series Volume 20
ISBN 3-00-012372-5, February 2004, 482 pages

**Measuring Synchronization in Model Systems and
Electroencephalographic Time Series from Epilepsy Patients**
Thomas Kreutz
NIC Series Volume 21
ISBN 3-00-012373-3, February 2004, 138 pages

Computational Soft Matter: From Synthetic Polymers to Proteins - Poster Abstracts
Norbert Attig, Kurt Binder, Helmut Grubmüller, Kurt Kremer (Editors)
Winter School, 29 February - 6 March 2004, Gustav-Stresemann-Institut Bonn
NIC Series Volume 22
ISBN 3-00-012374-1, February 2004, 120 pages

Computational Soft Matter: From Synthetic Polymers to Proteins - Lecture Notes
Norbert Attig, Kurt Binder, Helmut Grubmüller, Kurt Kremer (Editors)
Winter School, 29 February - 6 March 2004, Gustav-Stresemann-Institut Bonn
NIC Series Volume 23
ISBN 3-00-012641-4, February 2004, 440 pages

Synchronization and Interdependence Measures and their Applications to the Electroencephalogram of Epilepsy Patients and Clustering of Data
Alexander Kraskov
NIC Series Volume 24
ISBN 3-00-013619-3, May 2004, 106 pages

All volumes are available online at http://www.fz-juelich.de/nic-series/.