

## Feature article

# NWChem: A comprehensive and scalable open-source solution for large scale molecular simulations <sup>☆</sup>

M. Valiev <sup>a,\*</sup>, E.J. Bylaska <sup>a</sup>, N. Govind <sup>a</sup>, K. Kowalski <sup>a</sup>, T.P. Straatsma <sup>a</sup>, H.J.J. Van Dam <sup>a</sup>, D. Wang <sup>a</sup>, J. Nieplocha <sup>a</sup>, E. Apra <sup>b</sup>, T.L. Windus <sup>c</sup>, W.A. de Jong <sup>a</sup>

<sup>a</sup> William R. Wiley Environmental Molecular Sciences Laboratory, Pacific Northwest National Laboratory, P.O. Box 999, Richland, WA 99352, United States

<sup>b</sup> Oak Ridge National Laboratory, Computational Sciences and Math Division, Oak Ridge, TN 37831, USA

<sup>c</sup> Iowa State University, Department of Chemistry, Ames, IA 50011, USA

## ARTICLE INFO

## Article history:

Received 20 May 2009

Received in revised form 31 March 2010

Accepted 28 April 2010

Available online 13 May 2010

## Keywords:

NWChem

DFT

Coupled cluster

QMMM

Plane wave methods

## ABSTRACT

The latest release of NWChem delivers an open-source computational chemistry package with extensive capabilities for large scale simulations of chemical and biological systems. Utilizing a common computational framework, diverse theoretical descriptions can be used to provide the best solution for a given scientific problem. Scalable parallel implementations and modular software design enable efficient utilization of current computational architectures. This paper provides an overview of NWChem focusing primarily on the core theoretical modules provided by the code and their parallel performance.

## Program summary

*Program title:* NWChem

*Catalogue identifier:* AEGI\_v1\_0

*Program summary URL:* [http://cpc.cs.qub.ac.uk/summaries/AEGI\\_v1\\_0.html](http://cpc.cs.qub.ac.uk/summaries/AEGI_v1_0.html)

*Program obtainable from:* CPC Program Library, Queen's University, Belfast, N. Ireland

*Licensing provisions:* Open Source Educational Community License

*No. of lines in distributed program, including test data, etc.:* 11 709 543

*No. of bytes in distributed program, including test data, etc.:* 680 696 106

*Distribution format:* tar.gz

*Programming language:* Fortran 77, C

*Computer:* all Linux based workstations and parallel supercomputers, Windows and Apple machines

*Operating system:* Linux, OS X, Windows

*Has the code been vectorised or parallelized?:* Code is parallelized

*Classification:* 2.1, 2.2, 3, 7.3, 7.7, 16.1, 16.2, 16.3, 16.10, 16.13

*Nature of problem:* Large-scale atomistic simulations of chemical and biological systems require efficient and reliable methods for ground and excited solutions of many-electron Hamiltonian, analysis of the potential energy surface, and dynamics.

*Solution method:* Ground and excited solutions of many-electron Hamiltonian are obtained utilizing density-functional theory, many-body perturbation approach, and coupled cluster expansion. These solutions or a combination thereof with classical descriptions are then used to analyze potential energy surface and perform dynamical simulations.

*Additional comments:* Full documentation is provided in the distribution file. This includes an *INSTALL* file giving details of how to build the package. A set of test runs is provided in the *examples* directory.

The distribution file for this program is over 90 Mbytes and therefore is not delivered directly when download or Email is requested. Instead a html file giving details of how the program can be obtained is sent.

*Running time:* Running time depends on the size of the chemical system, complexity of the method, number of cpu's and the computational task. It ranges from several seconds for serial DFT energy calcu-

<sup>☆</sup> This paper and its associated computer program are available via the Computer Physics Communications homepage on ScienceDirect (<http://www.sciencedirect.com/science/journal/00104655>).

\* Corresponding author.

E-mail address: [marat.valiev@pnl.gov](mailto:marat.valiev@pnl.gov) (M. Valiev).

lations on a few atoms to several hours for parallel coupled cluster energy calculations on tens of atoms or *ab-initio* molecular dynamics simulation on hundreds of atoms.

© 2010 Elsevier B.V. All rights reserved.

## 1. Introduction

Computational modeling has become an integral part of many research efforts in key application areas in chemical, physical, and biological sciences. The ability to provide a fine level of detail through the use of electronic structure and the freedom to utilize diverse external constraints and conditions makes computational modeling an extremely versatile tool to probe various physical observables of the system. In many cases the information gained from computational studies is unique and would be difficult to obtain from experiment.

To fully realize the potential of computational modeling in answering key scientific questions requires the availability of high quality computational codes that offer wide range of capabilities and are able to take advantage of massively parallel computer architectures. One particular example of such a code is the NorthWest computational Chemistry (NWChem) software package developed in the W.R. Wiley Environmental Molecular Sciences Laboratory (EMSL) at the Pacific Northwest National Laboratory (PNNL). The NWChem software package is currently distributed to over 2800 sites world wide and has played a key role in solving a wide range of complex scientific problems. The success of NWChem stems from the fact that it offers a broad array of molecular modeling capabilities that can be deployed on all the major supercomputing platforms and that the software is freely available to the general scientific community. The latest release, version 6.0, marks a new phase in the NWChem development efforts – a transition to the open source software model under the Educational Community License.

This work presents a high level overview of NWChem focusing on the core computational modules provided by the code.

## 2. Parallel infrastructure

The parallel infrastructure of NWChem is built upon the Global Array (GA) toolkit [1,2] that has been co-developed with NWChem to meet the requirements of distributed data algorithms in the computational chemistry area. The GA toolkit combines the best features of both the shared and distributed memory programming models [1,2]. It implements a shared-memory programming model in which data locality can be managed explicitly by the programmer. This management is achieved by explicit calls to functions that transfer data between a global address space (a distributed array) and local storage. In this respect, the GA model has similarities to distributed shared-memory (DSM) models that provide an explicit acquire/release protocol. However, the GA model acknowledges that remote data is slower to access than local data and allows data locality to be explicitly specified and hence managed. Another advantage is that GA, by optimizing and moving only the data requested by the user, avoids issues such as false sharing or redundant data transfers present in some DSM solutions. The GA model exposes to the programmer the hierarchical memory of modern high-performance computer systems, and by recognizing the communication overhead for remote data transfer, it promotes data reuse and locality. The GA programming model includes message passing as a subset, and in particular the programmer can use the full MPI functionality both on GA and non-GA data. The GA library can be used in C, C++, Fortran 77, Fortran 90 and Python programs. The GA toolkit offers support for both task and data parallelism. The task parallelism is supported

through one-sided (noncollective) copy operations that transfer data between global memory (distributed/shared array) and local memory. Each process is able to directly access data held in a section of a global array that is logically assigned to that process. Atomic operations are provided that can be used to implement synchronization and assure correctness of an accumulate operation (floating-point sum reduction that combines local and remote data) executed concurrently by multiple processes and targeting overlapping array sections. The accumulate operation is a fundamental operation needed by distributed-data implementations of several electronic structure algorithms in computational chemistry. A typical example of an algorithm that uses this common computational kernel is the evaluation of the Fock matrix elements in a Hartree–Fock (HF) calculation [3] using a local basis set (also known as a “Fock build”); a very similar algorithm is the evaluation of the Kohn–Sham matrix elements in density functional (DFT) calculations. The data parallel computing model is supported through the set of collectively called functions that operate on either entire arrays or sections of global arrays. The set includes BLAS-like operations (copy, additions, transpose, dot products, and matrix multiplication). Some of them are defined and supported for all array dimensions (e.g., addition). Other operations, such as matrix multiplication, are limited to two-dimensional arrays (however, multiplication is also offered on two-dimensional subsections of higher dimensional arrays). GA extends its capabilities in the area of linear algebra by offering interfaces to third party libraries e.g., standard and generalized real symmetric eigensolvers (PcIGS), and linear equation solvers (ScaLAPACK) [4].

## 3. Input file structure

In order to run NWChem calculations an input file has to be prepared, which is a free-format text file that contains start-up directives, definition of the chemical system, specification of various parameters for the calculations, and task directives. The actual processing of the input file is performed in several phases using the input parsing module. The first phase consists of processing the start-up directives that define the general features of the calculation, including available memory, database name, file locations, and other global options. First the input parsing module tries to locate the memory directive, which specifies the amount of memory requested for the NWChem job. The need for this directive arises partly from the use of the Fortran 77 language, which is used in the majority of NWChem code. If this directive is not specified, memory is allocated according to installation-dependent defaults. A typical strategy for the memory specification is to set it in the range of the total free CPU memory leaving sufficient room for standard system processes. In the next step the input module looks for the directive that specifies the database file. This binary file (characterized by .db extension) plays an important role in NWChem calculations. It serves as a repository for nearly all the input and calculation parameters and facilitates restart or continuation of a previous run. It should be noted that large datasets (such as the wavefunction) are stored outside the database file with a reference to the filename stored in the database file. The database file can be defined using the start or restart directives, which would typically be among the first lines in the input file. In the example given in [Input 1](#) the `start h2o` directive would cause the creation of a `h2o.db` database file which will then accumulate the parameters pertinent to this calculation. Alternatively the

**Input 1.** Example NWChem input file for H<sub>2</sub>O energy and optimization.

```

echo
start h2o
memory total 200 mb
permanent_dir ./perm
scratch_dir    /tmp

geometry units au
  O 0      0      0
  H 0      1.430  -1.107
  H 0      -1.430 -1.107
end

basis
  * library 6-31g*
end
dft
  xc b3lyp
end

driver
  maxiter 20
end

task dft energy
task dft optimize

```

use of a `restart h2o` directive would imply that the existing `h2o.db` database file from some prior calculation should be used. This will trigger the reuse of the information contained in this database file, which can be adjusted through the input file. Should the `start/restart` directives be absent, the name of the database is taken from the name of the input file (without the extension). If such a database file already exists then the restart behavior will be assumed. While parsing the input file for the database file definition, the input module will also look for the `permanent_dir`, `scratch_dir`, and `echo` directives. The `permanent_dir` and `scratch_dir` define the location of directories that contain scratch and permanent files created by NWChem. Permanent (or persistent) files are those files that might be needed for the continuation/restart of the calculation. In addition to the database file itself, other permanent files may include the wavefunction file and the hessian file. The actual read and write operation on the permanent files is performed by a single master node which then sends the data to other processes as needed. Scratch files are the temporary files generated during the calculation by any given parallel process. These scratch files are not meant to be reused beyond the scope of the current calculation, and thus can be safely deleted at the end of the calculation. The scratch directory is typically set to an appropriate temporary disk partition for a given node. The default option for the permanent and scratch directory is the top level calculation directory (i.e. the directory where the calculation is launched). This choice, however, is not advisable, and a better option would be to place these files to separate locations, such as `./perm` and `/tmp` as in the example in [Input 1](#). A convenient option is to designate a system-wide definition for the scratch directory by defining the environmental variable `SCRATCH_DIR`. The optional `echo` start-up directive provides a convenient way to include a listing of the entire input file in the output of a calculation.

Once the start-up directives are processed, the input module initiates sequential parsing of the input file for the parameters that define the actual calculation. It should be noted that only a minimal amount of computation happens at this point, mostly consisting of migrating relevant input parameters into the database file. This input file parsing is facilitated by grouping common sets of parameters into blocks. For example, all the input parameters specifying the geometry of the system are collected into the geometry block. In its simplest form this amounts to specifying the atom name and Cartesian coordinates on separate lines delimited

by the geometry and the end directives. A number of other options and settings for the geometry block, as well as other input parameters, are discussed in the user manual. Parsing of the input file continues until a task directive is encountered. Once the input parser recognizes a task directive, further parsing of the input file is halted and control is passed to the appropriate module, which extracts relevant data from the database and initiates the appropriate computational task. Upon completion of the task, the significant results will be again stored in the database and control will be passed back to the input parsing module. This procedure will be repeated until the end of the input file is encountered. This model of input data processing allows multiple computational task directives and provides facilities for building complex task flows.

The task directive in its simplest form is given by `task <theory> <operation>`. The keyword `<theory>` specifies the theoretical description (e.g. `scf`, `dft` or `mp2`) and the keyword `<operation>` denotes particular operations (e.g. `energy`, `gradient` or `optimize`). A number of theories are supported by NWChem, the most widely used are reviewed in the remainder of the paper. In addition to the capabilities described in the next couple of sections, NWChem also includes parallel implementations of SCF, MPn theories, Resolution of the Identity MP2, ONIOM, vibrational SCF, Dynamical Nucleation Theory Monte Carlo, Python interfaces, Venus and Polyrate interfaces for dynamics and rate determination, NBO interface and COSMO for continuum solvation effects. The reader is referred to the user manual or NWChem webpage for a complete listing of all capabilities.

#### 4. Density-functional theory calculations

Quantum-mechanical calculations of chemical systems using density-functional theory (DFT) [5,6] is one the most broadly used capabilities of NWChem. DFT provides a good mix of efficiency and accuracy and is applicable over a wide range of chemical and material systems containing up to a thousand atoms. The efficiency of the DFT approach stems from the fact that, similar to Hartree-Fock methods, the many-electron problem is cast into the solution of single-particle equations, [5–7] also known as Kohn–Sham equations:

$$\left(-\frac{1}{2}\nabla^2 + v^C(\mathbf{r})\right)\psi_i(\mathbf{r}) + \int v^{xc}(\mathbf{r}, \mathbf{r}')\psi_i(\mathbf{r}')d\mathbf{r}' = \epsilon_i\psi_i(\mathbf{r}) \quad (1)$$

Here the  $v^C(\mathbf{r})$  represents the classical Coulomb potential consisting of nuclei and Hartree contributions:

$$v^C(\mathbf{r}) = -\sum_a \frac{Z_a}{|\mathbf{R}_a - \mathbf{r}|} + \int \frac{\rho(\mathbf{r}, \mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r}' \quad (2)$$

where  $Z_a$  and  $\mathbf{R}_a$  represent nuclei charges and positions, and  $\rho(\mathbf{r}, \mathbf{r}')$  is the electron density which is a diagonal of the electron density matrix:

$$\rho(\mathbf{r}, \mathbf{r}') = \sum_\alpha \psi_\alpha^*(\mathbf{r})\psi_\alpha(\mathbf{r}') \quad (3)$$

The complexities of many-electron interactions are hidden in the last term  $v^{xc}$ , the so-called exchange-correlation potential. Other than formal diagrammatic expansions [8,9], the closed form expression for  $v^{xc}$  is not available and in practice approximate expressions have to be utilized. From a computational point of view, these can be broadly divided into two categories – local and non-local. In the local form  $v^{xc}$  acts as a local multiplicative potential

$$v^{xc}(\mathbf{r}, \mathbf{r}') = v_{loc}^{xc}(\mathbf{r})\delta(\mathbf{r} - \mathbf{r}') \quad (4)$$

whereas in the nonlocal form it is truly an integral operator as shown in (1). Most of the early developed approximations for  $v^{xc}$

are of the local form. These include original local density approximation (LDA) [7] where  $v^{xc}$  is actually a function of the electron density, and generalized gradient approximations [10–13] where dependence on the density gradient is also added. Local forms of  $v^{xc}$  significantly simplify solution of single-particle equations (1) and used predominantly in plane wave based implementations of DFT. It is a generally accepted fact however, that the local form of  $v^{xc}$  does not provide an adequate description of exchange effects, and all the modern approximations treat those effects explicitly in terms of a nonlocal component similar in structure to Hartree–Fock exchange operator:

$$v^x(\mathbf{r}, \mathbf{r}') = -\frac{\rho(\mathbf{r}, \mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} \quad (5)$$

Among the nonlocal approximations for  $v^{xc}$  are the original hybrid functionals which include part of  $v^x$  in conjunction with local forms of  $v^{xc}$ , [14,15] metafunctionals which also include a part of the kinetic energy [16], and the recently developed range separated forms [17–20]. Detailed discussion of different classes of exchange–correlation functionals is beyond the scope of this paper, and we refer the interested reader to more comprehensive references on this subject [6,7,16,21–25].

In practical applications the solution of (1) is facilitated by expanding single-particle orbitals  $\psi_i(\mathbf{r})$  in terms of a finite basis set. Two popular choices are atom-centered local basis sets and plane waves. Both are implemented in NWChem and discussed in the following sections.

#### 4.1. Local basis implementation

The local basis implementation of DFT in NWChem is aimed at the description of isolated molecular systems and utilizes atom-centered Gaussian type orbitals (GTO) [26],

$$\phi_\mu(\mathbf{r}) = x^l y^m z^n e^{-ar^2} \quad (6)$$

or linear combinations of thereof (contracted GTO's). The single-particle orbitals, represented in this case as real functions, are expanded as

$$\psi_i(\mathbf{r}) = \sum_{\mu} c_{i\mu} \phi_\mu(\mathbf{r} - \mathbf{R}_{a\mu}) \quad (7)$$

The actual specification of the local basis, which is in general atom specific, is handled by the `basis` set directive. The user can specify contracted Cartesian Gaussian functions up to a maximum angular momentum of five (h functions), and also SP (or L) functions. Both spherical and Cartesian angular functions are supported. A wide range of basis set libraries are available as part of NWChem distribution. Alternatively the EMSL basis set exchange (<https://bse.pnl.gov/bse/portal>) can be used to provide explicit basis set specification [27]. Further details about basis set specification can be found in the user manual.

With the help of the local basis set expansion (7) Eq. (1) can be transformed into the matrix form:

$$\mathbf{FC} = \mathbf{SCE} \quad (8)$$

where  $\mathbf{C} = \{c_{i\mu}\}$  is the matrix of expansion coefficients (see Eq. (7)),  $\mathbf{E} = \{\epsilon_i\}$  is the eigenvalue matrix, and  $\mathbf{S}$  is the overlap matrix:

$$\mathbf{S}_{\mu\nu} = \int \phi_\mu(\mathbf{r}) \phi_\nu(\mathbf{r}) d\mathbf{r} \quad (9)$$

The Fock matrix,  $\mathbf{F}$ , consists of the kinetic, Coulomb and exchange–correlation matrix elements

$$\mathbf{F}_{\mu\nu} = t_{\mu\nu} + v_{\mu\nu}^c + v_{\mu\nu}^{xc} \quad (10)$$

where

$$t_{\mu\nu} = -\frac{1}{2} \int \phi_\mu(\mathbf{r}) \nabla^2 \phi_\nu(\mathbf{r}) d\mathbf{r} \quad (11)$$

$$v_{\mu\nu}^c = \int \phi_\mu(\mathbf{r}) v^c(\mathbf{r}) \phi_\nu(\mathbf{r}) d\mathbf{r} \quad (12)$$

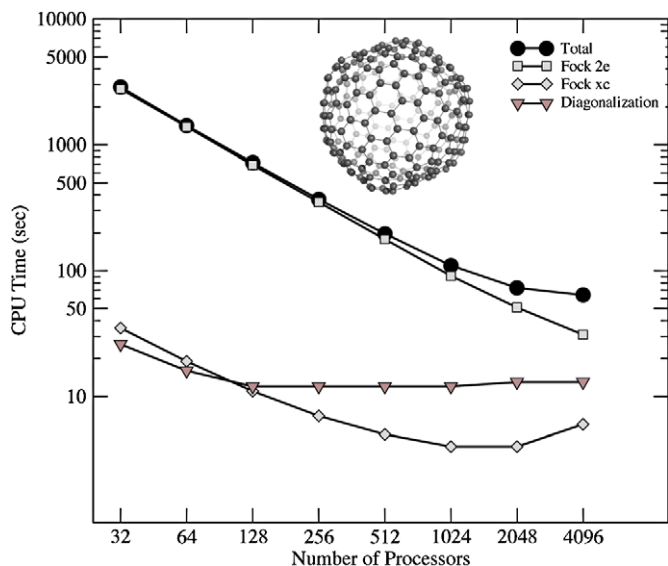
$$v_{\mu\nu}^{xc} = \iint \phi_\mu(\mathbf{r}) v^{xc}(\mathbf{r}, \mathbf{r}') \phi_\nu(\mathbf{r}') d\mathbf{r} d\mathbf{r}' \quad (13)$$

The generalized nonlinear eigenvalue equation (8) is solved using a self-consistent field (SCF) procedure akin to the solution of the Roothaan–Hall (and Pople–Nesbet) equations in the Hartree–Fock (HF) approximation [28–30]. The two main computational steps involve construction of the Fock matrix and its subsequent diagonalization. Given the dependence of the Fock matrix (through Coulomb and exchange–correlation contributions) on the single-particle orbitals, the procedure is repeated until self-consistence is reached.

In the GTO representation, the kinetic, Coulomb contributions to Fock matrix can be evaluated analytically and scale as  $O(N^2)$  and  $O(N^4)$  respectively, with system size. The  $O(N^4)$  scaling of the Coulomb component can be reduced to  $O(N^3)$ , by utilizing charge density (CD) fitting or resolution of identity (RI) approaches. In NWChem the CD fitting is accomplished using the Dunlap method [31] implemented along the lines proposed by von Arnim and Ahlrichs [32]. The in-core strategy for storing all the 3-center, 2-electron integrals in memory allows for a much faster evaluation of the Coulomb contribution compared with disk-based or direct approaches. This approach also lends itself to efficient parallelization in the sense that the available memory is the aggregate of the memory of all processors involved in the computation. It is also worth noting that all the coefficients for CD fitting are computed in parallel.

In the evaluation of the exchange–correlation matrix elements one distinguishes between local and nonlocal components (Eq. (5)) of the exchange–correlation potential. The nonlocal component has a structure similar to that of the Coulomb matrix and is again evaluated analytically, scaling as  $O(N^4)$ . This scaling can be reduced to  $O(N^3)$  following the strategy of CD or RI approaches mentioned above. The local part of the exchange–correlation potential has to be evaluated numerically on the radial grids scaling as  $O(N^3)$ . The calculation is performed in parallel involving spatial decomposition of the atom-centered grids [33,34] and the use of mirrored arrays to reduce communication costs.

The efficiency of the local basis DFT implementation in NWChem is illustrated in Fig. 1 which shows the wall clock time per iteration for different numbers of processors using C<sub>240</sub> cluster as an example system. The calculations utilize the PBE0 [15] nonlocal exchange–correlation functional and the 6-31G\* basis set [35,36] with a total of 3600 basis functions for the whole system. All the integral evaluations were performed using the direct approach and Fock matrix replication. At the lower range of processor count, the  $O(N^4)$  scaling associated with construction of Coulomb and nonlocal exchange–correlation to the Fock matrix dominates the computational time. This part of the code however lends itself to efficient parallelization with almost perfect scaling through 4096 CPU's. The local exchange–correlation component of the Fock matrix is also exhibits good scaling, however increase in communication costs limits the efficiency to a 1000 CPU's. As the cost of Fock matrix construction drops for large processor counts ( $\geq 1000$ ) the performance is impacted by the diagonalization step which is not as easily parallelizable. One potential solution to this problem consists in utilizing iterative approaches to the eigenvalue problem which presents an alternative to a traditional diagonalization scheme currently implemented in NWChem. Such efforts are underway and will be a part of a future release.



**Fig. 1.** Scaling of local basis DFT calculation of  $C_{240}$  using the hybrid PBE0 exchange-correlation functional and the 6-31G\* basis with a total of 3600 basis functions for the whole system. Fock 2e denotes computational costs associated with construction of Coulomb and nonlocal exchange-correlation to the Fock matrix. Fock xc denotes computational costs associated with construction of local exchange-correlation part of the Fock matrix.

In NWChem the DFT parameters are set within the `dft` input block (see [Input 1](#)) and described in detail in the user manual. We provide a brief description of some of the important options below.

The initial guess can be supplied as follows:

```
dft
...
vectors input <option>
...
end
task dft energy
```

Among the available options are `atomic` (default) which uses a superposition of atomic densities, `hcore` which uses the one-electron Hamiltonian, `project` which uses a set of orbitals generated using a smaller basis set where convergence is typically easier to project onto a larger basis set, and `fragment` which combines orbitals generated from fragments of the full molecule.

A number of standard convergence options for the solution of eigenvalue equations are provided such as DIIS [37,38], damping, level shifting [39] and smearing [40]. The latter option is especially useful in cases where there are near-degeneracies in the valence space of orbitals.

NWChem also provides an extensive and flexible framework for the specification of exchange-correlation functionals. This is specified using `xc` keyword. For example, `xc hfexch` will perform a pure Hartree-Fock calculation, B3LYP [14] can be specified as `xc b3lyp`, PBE0 [15] as `xc pbe0`, PBE96 [10] as `xc xpbe96 cpbe96`, etc. One of the appealing features of NWChem is that one can build a custom functional by suitably mixing available forms. This gives the user an added level of control. For example, the B3LYP functional may also be specified in terms of its hybrid components as:

```
dft
...
xc vwn_1_rpa 0.19 lyp 0.81 HFexch 0.20 \
  slater 0.80 becke88 nonlocal 0.72
...
```

```
end
task dft energy
```

In a similar way, the PBE0 functional may also be specified as:

```
dft
...
xc xpbe96 0.75 cpbe96 1.0 hfexch 0.25
...
end
task dft energy
```

Many other choices of exchange-correlation functionals are also available. We refer the reader to the NWChem user manual for details. All the available exchange-correlation functionals have associated analytic first derivatives. However, analytic second derivatives are not available for all functionals. Long-range dispersion corrections have been recently implemented. These corrections can be added empirically to the DFT energies (DFT-D) [41–44] as well as when computing gradients and Hessians. These are useful when treating weakly interacting systems such as van der Waals complexes. Range-separated [45,46] and double-hybrid functionals [47] have also been implemented in the latest version of NWChem.

Relativistic effects can also be included in DFT/HF calculations via the all-electron spin-free and spin-orbit one-electron Douglas-Kroll-Hess (DKH) method [48–50] and zeroth-order relativistic approximation (ZORA) [51–54] as well as through effective core (ECP) and spin-orbit (SO) potentials.

Excited state calculations can also be performed using a variety of single excitation methods (or linear response approaches) such as time-dependent Hartree-Fock (TDHF) also known as the random-phase approximation (RPA), time-dependent density functional theory (TDDFT) [55] and the Tamm-Dancoff approximation (TDA) to TDDFT [56]. These methods are available within the TDDFT module in NWChem in both the spin-restricted and unrestricted formalisms. To improve the description of Rydberg excited states, two asymptotic correction methods [57] are also available. These have been shown to markedly improve the TDDFT excitation energies. The range-separated functionals are also available within the TDDFT module. These functionals have shown a lot of promise in the description of charge-transfer (CT) excited states [58,19,59, 20].

#### 4.2. Plane wave implementation

The plane wave DFT module in NWChem is a collection of three programs.

- PSPW – a pseudopotential plane wave  $\Gamma$ -point code for calculating molecules, liquids, crystals, and surfaces.
- Band – a band structure code for calculating crystals and surfaces with small band gaps (e.g. semi-conductors and metals).
- PAW – a ( $\gamma$ -point) projector augmented plane-wave code for calculating molecules, crystals, and surfaces

All three programs make use of a common infrastructure composed of generic operations related to plane wave basis sets. The use of the PSPW, BAND, or PAW approach in the calculations is accomplished by using `pspw`, `band`, or `paw` keyword in the task directive. In addition to the standard directives, e.g. `energy`, `optimize`, there are additional directives specific to the plane wave module that are used to run Car-Parrinello molecular dynamics simulations (i.e. `task pspw car-parrinello` or `task paw car-parrinello`), generate band structure plots (i.e. `task band structure`), and compute the density of states (i.e. `task band dos`). As shown in the example [Input 2](#) the actual parameters

**Input 2.** Example NWChem input file for H<sub>2</sub>O energy and optimization using plane wave DFT implementation.

```
echo
start h2o
memory total 200 mb
permanent_dir ./perm
scratch_dir      /tmp

geometry units au
  O  0      0      0
  H  0      1.430  -1.107
  H  0     -1.430  -1.107
end

nwpw
  simulation_cell
    SC 26.0
  end
  cutoff 50.0
end

driver
  maxiter 20
end

task pspw energy
task pspw optimize
```

defining the various plane wave calculations are entered in the `nwpw` block. A list of the numerous options available for the various types of calculations can be found in the NWChem user manual.

The plane wave DFT module in NWChem can be used to describe both extended periodic system and isolated molecular systems with free space boundary conditions. In either case the system is assumed to be enclosed in a suitably defined unit cell characterized by the unit cell vectors  $\{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3\}$ . In conjunction with periodic boundary conditions this allows to express single-particle orbitals in terms of the following plane wave expansion

$$\psi_{\alpha\mathbf{k}}(\mathbf{r}) = e^{i\mathbf{k}\cdot\mathbf{r}} \sum_{\mathbf{G}} \tilde{\psi}_{\alpha\mathbf{k}}(\mathbf{G}) e^{i\mathbf{G}\cdot\mathbf{r}} \quad (14)$$

where  $\mathbf{k}$  is a vector in the first Brillouin zone. The reciprocal lattice vectors  $\mathbf{G}$  form a discrete set given by

$$\mathbf{G}_{i_1 i_2 i_3} = i_1 \mathbf{b}_1 + i_2 \mathbf{b}_2 + i_3 \mathbf{b}_3 \quad (15)$$

Here reciprocal unit lattice vectors are given by

$$\mathbf{b}_i = 2\pi \epsilon_{ijk} \frac{\mathbf{a}_j \times \mathbf{a}_k}{\Omega} \quad (16)$$

where  $\epsilon_{ijk}$  is Levi-Civita symbol and  $\Omega = \mathbf{a}_1 \cdot (\mathbf{a}_2 \times \mathbf{a}_3)$  is the volume of the unit cell. In the calculations of isolated molecular systems, where the single-particle orbitals are expected to be localized within the unit cell, the sampling over the Brillouin zone is necessarily restricted to a  $\Gamma$ -point ( $\mathbf{k} = 0$ ) giving rise to the so-called  $\Gamma$ -point representation as

$$\psi_{\alpha}(\mathbf{r}) = \sum_{\mathbf{G}} \tilde{\psi}_{\alpha}(\mathbf{G}) e^{i\mathbf{G}\cdot\mathbf{r}} \quad (17)$$

The reciprocal space representation,  $\psi_{\alpha\mathbf{k}}(\mathbf{G})$ , as defined by Eqs. (14) and (17) forms discrete but infinite set (see Eq. (15)). For practical calculations, it is truncated by considering  $\psi_{\alpha\mathbf{k}}(\mathbf{r})$  only on a subset of points forming a uniform grid within the unit cell:

$$\mathbf{r}_{i_1 i_2 i_3} = \left(\frac{i_1}{N_1} - \frac{1}{2}\right) \mathbf{a}_1 + \left(\frac{i_2}{N_2} - \frac{1}{2}\right) \mathbf{a}_2 + \left(\frac{i_3}{N_3} - \frac{1}{2}\right) \mathbf{a}_3 \quad (18)$$

where  $i_m \in [1, N_m]$ . The corresponding reciprocal grid is then reduced to

$$\mathbf{G}_{i_1 i_2 i_3} = \left(i_1 - \frac{N_1}{2}\right) \mathbf{b}_1 + \left(i_2 - \frac{N_2}{2}\right) \mathbf{b}_2 + \left(i_3 - \frac{N_3}{2}\right) \mathbf{b}_3 \quad (19)$$

Further reduction can be achieved by eliminating plane components beyond predefined maximum kinetic energy cutoff:

$$\frac{1}{2} |\mathbf{G}|^2 < E_{cut} \quad (20)$$

There are several advantages associated with using a plane waves basis set. The plane wave expansion coefficients,  $\tilde{\psi}_{\alpha\mathbf{k}}(\mathbf{G})$ , as well as other quantities can be efficiently computed using Fast Fourier transform techniques [60]. Compared to a local basis set, plane wave basis set is also independent of nuclei positions which ensures equal treatment of all the regions in the system and avoids any over-completeness or superposition errors. The main shortcoming of a plane basis set is the inefficient description of rapid oscillations of electronic wavefunction in the vicinity of the atomic nucleus. Large numbers of plane waves are required for an accurate representation of such behavior, leading, in most cases, to an intractably large computational problem. One potential solution is offered by the pseudopotential planewave (PSPW) method [61–64] where the fast varying core regions of the atomic potentials and the core electrons are removed and replaced by smoothly varying pseudopotentials. The pseudopotentials are chosen such that the scattering properties of the resulting pseudoatoms are the same as the original atoms [65,66]. Detailed description of PSPW approach can be found elsewhere. The important point for the purpose of this paper is that PSPW approximation results in the appearance of the extra pseudopotential term in the original single-particle equations

$$\left(-\frac{1}{2}\nabla^2 + v^c(\mathbf{r})\right) \psi_{\alpha}(\mathbf{r}) + \int v^{ps}(\mathbf{r}, \mathbf{r}') \psi_{\alpha}(\mathbf{r}') d\mathbf{r}' + \int v^{xc}(\mathbf{r}, \mathbf{r}') \psi_{\alpha}(\mathbf{r}') d\mathbf{r}' = \epsilon_{\alpha} \psi_{\alpha}(\mathbf{r}) \quad (21)$$

The nonlocal pseudopotential term is atom-centered:

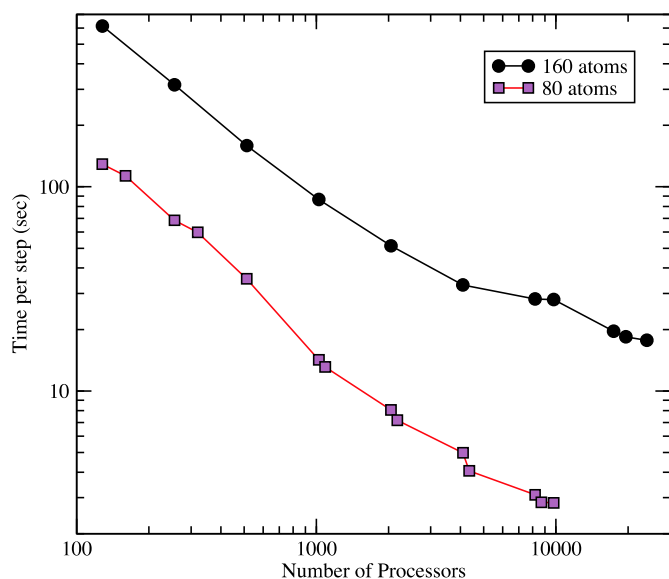
$$v_a^{ps}(\mathbf{r}, \mathbf{r}') = \sum_a v_a^{ps}(\mathbf{r} - \mathbf{R}_a, \mathbf{r}' - \mathbf{R}_a) \quad (22)$$

and, in general, nonseparable. It can, however, be approximated in a separable form [67] as:

$$v_a^{ps}(\mathbf{r}, \mathbf{r}') = v_{a,0}^{ps}(\mathbf{r}) \delta(\mathbf{r} - \mathbf{r}') + \sum_{ij} p_i^*(\mathbf{r}) v_{a,ij}^{ps} p_j^*(\mathbf{r}') \quad (23)$$

where  $p_i(\mathbf{r})$  are projector functions and  $v_{a,0}^{ps}, v_{a,ij}^{ps}$  are expansion coefficients that depend only on the nature of chemical species. An alternative to the PSPW method can be found in the projected augmented plane wave method (PAW) [68–72]. Instead of discarding the rapidly varying parts of the electronic orbitals in the core region, in PAW these are treated by projecting them onto a local basis set (e.g. a basis of atomic functions). By carefully choosing the local basis the convergence of the plane wave problem can be systematically improved. The PAW method is an all electron method, and no part of the electron density is removed from the problem. This can be a significant advantage over the PSPW program for some problems, particularly for magnetic systems. Similar to PSPW approach, PAW methodology also results in the appearance of a nonlocal potential term (see Eq. (23)), but the expansion coefficients  $v_{a,0}^{ps}, v_{a,ij}^{ps}$  now also acquire an orbital dependence. There are other subtle differences between PSPW and PAW methods, which are discussed in detail elsewhere [68–72].

In general, the size of the basis sets in plane wave methods are very large compared to those encountered in local basis methods. As a result, a different strategy is utilized for the solution of the



**Fig. 2.** The overall fastest timings taken for an 80 and 160 atom  $\text{Fe}_2\text{O}_3$  hybrid plane wave DFT energy calculations. Calculations were performed on the Franklin Cray-XT4 computer system at NERSC.

Kohn–Sham single-particle equations (21). Instead of explicit construction and direct diagonalization of the corresponding “Fock” matrix, plane-wave methods evaluate of the wavefunction gradient  $\mathbf{S}$ , represented by the left hand side of (21)

$$\begin{aligned} S(\mathbf{r}) = & \left( -\frac{1}{2}\nabla^2 + v^c(\mathbf{r}) \right) \psi_\alpha(\mathbf{r}) + \int v^{ps}(\mathbf{r}, \mathbf{r}') \psi_\alpha(\mathbf{r}') d\mathbf{r}' \\ & + \int v^{xc}(\mathbf{r}, \mathbf{r}') \psi_\alpha(\mathbf{r}') d\mathbf{r}' \end{aligned} \quad (24)$$

for use in a conjugate gradient algorithm to solve the Kohn–Sham equations or, Car–Parrinello dynamics. It should be noted that both real and reciprocal space calculations are involved in the evaluation of Eq. (24). Typically, kinetic and nonlocal pseudopotential terms are calculated in reciprocal space, whereas local pseudopotential, Coulomb, and exchange–correlation contribution are performed in real space. The most expensive parts of the algorithm involve evaluation of exact exchange (if present), nonlocal pseudopotential, and wavefunction orthogonalization.

Plane wave DFT implementation can greatly benefit from parallelization, and several strategies have been developed [73–76,72]. There are several options for parallel data layout, involving distribution over the  $\mathbf{k}$ -space (Brillouin zone), distribution over the orbital index  $\alpha$ , and spatial distribution in real/reciprocal space. Distribution over the Brillouin zone [73] ( $\mathbf{k}$  in Eq. (14)) is potentially beneficial for many solid-state applications, but the scalability of this approach is limited by the size of the Brillouin zone, and the method is not applicable to  $\Gamma$ -point calculations. Distribution over the orbital index  $\alpha$  has been shown to scale quite well [75], but requires storage of the entire one-electron orbital on a single node which may not be practical for large molecular systems that require large number of plane wave expansion coefficients. In the spatial distribution approach sections of three-dimensional real or reciprocal space coefficients (typically along a single dimension) are stored on different processors [74,76,72]. Given the large size of real/reciprocal space ( $10^4$ – $10^6$ ) grids, this model would seem to makes a lot of sense in terms of scalability and memory management and is commonly available in many of the plane wave codes including NWChem. The tradeoff in this type of approach is the increased communication costs incurred by parallel three-dimensional FFT transformations. The scalability of the latter levels

off beyond  $\sqrt[3]{N_1 N_2 N_3}$  number of processors, which for a typical applications limits the number of processors to 1000 or less. This limitation can be avoided by distributing both the spatial and orbital degrees of freedom on a two-dimensional processor grid [77]. This approach is also available in NWChem plane wave DFT implementation.

A limitation of many implementations of plane wave DFT has been the lack of support for hybrid forms of exchange–correlation potentials which contain a small amount of exact exchange. Hybrid functionals are important for correct estimates of band gaps, spin localized states, and reaction barriers. NWChem no longer suffers from this restriction and offers efficient and highly scalable treatment of exact exchange term. The drawback of including an exact exchange term is the expense; plane-wave methods require the computation of  $O(N_e^2)$  3D-FFTs, where  $N_e$  is the number of valence electrons. We have developed a highly scalable algorithm for exact exchange and incorporated it into a plane-wave DFT application. Using this approach, we have been able to overcome high resource demands, scaling to over 20 000 CPUs for even modest size problems. This is illustrated in Fig. 2 that shows parallel scaling of hybrid plane wave DFT calculation for  $\text{Fe}_2\text{O}_3$  with the unit cell containing 80 and 160 atoms.

## 5. Coupled cluster calculations

The coupled cluster theory [78–81] (CC) is considered by many to be a gold standard for accurate quantum-mechanical description of ground and excited states of chemical systems. This accuracy, however, comes at a significant computational cost which limits CC calculations to systems containing under hundred atoms with a total time to solution for single point calculation approaching several hours. The computational expense of CC calculations comes from the fact that it is a true many-body approach that seeks solution of full Schrödinger equation in terms of the exponential ansatz:

$$|\Psi\rangle = e^T |\Phi\rangle \quad (25)$$

Here the  $T$  operator represents the cluster operator

$$T = T_1 + T_2 + T_3 + \dots + T_N \quad (26)$$

$$T_n = (n!)^{-2} \sum_{\substack{i_1, i_2, \dots \\ a_1, a_2, \dots}} t_{i_1 \dots i_n}^{a_1 \dots a_n} \hat{c}_{a_1}^\dagger \dots \hat{c}_{a_n}^\dagger \hat{c}_{i_1} \dots \hat{c}_{i_n} \quad (27)$$

where  $\hat{c}^\dagger$ ,  $\hat{c}$  denote creation and annihilation operators, and indices  $a_k$ ,  $i_k$  denote occupied and unoccupied states respectively. The single determinant reference wavefunction  $|\Phi\rangle$  is assumed to have nonzero overlap with a true ground state wave-function, its choice is otherwise arbitrary obtained typically from Hartree–Fock or DFT calculations. As it stands, the exponential ansatz (25) does not introduce any approximations, but allows to reformulate the solution of Schrödinger equation

$$H|\Psi\rangle = E|\Psi\rangle \quad (28)$$

in terms of cluster amplitudes  $t_{i_1 \dots i_n}^{a_1 \dots a_n}$ . Projection of Eq. (28) onto a reference state  $|\Phi\rangle$  provides an expression for ground state energy:

$$\begin{aligned} E &= \langle \Phi | H e^T | \Phi \rangle \\ E &= \langle \Phi | H \left( 1 + T_1 + T_2 + \frac{1}{2} T_1^2 \right) | \Phi \rangle \end{aligned} \quad (29)$$

While the ground state energy is fully determined in terms of  $T_1$  and  $T_2$  amplitudes, the knowledge of the latter requires solution of set of coupled equations involving all the amplitudes. These equations are obtained by projecting Eq. (28) onto various reference excited states  $|\Phi_{i_1 \dots i_n}^{a_1 \dots a_n}\rangle = \hat{c}_{a_1}^\dagger \dots \hat{c}_{a_n}^\dagger \hat{c}_{i_1} \dots \hat{c}_{i_n} |\Phi\rangle$

$$\langle \Phi_{i_1 \dots i_n}^{a_1 \dots a_n} | (He^T)_C | \Phi \rangle = 0 \quad (30)$$

where  $C$  signifies that only connected terms should be considered. A similar strategy is used for the determination of excited states in the Equation of Motion CC (EOMCC) method [82–84]. In this case in addition to cluster operator  $T$ , the excitation operator  $R$  for  $K$ th excited states is defined:

$$R_K = R_{K,0} + R_{K,1} + R_{K,2} + \dots \\ R_{K,n} = (n!)^{-2} \sum_{\substack{i_1, i_2, \dots \\ a_1, a_2, \dots}} r_{K, i_1 \dots i_n}^{a_1 \dots a_n} \hat{c}_{a_1}^\dagger \dots \hat{c}_{a_n}^\dagger \hat{c}_{i_1} \dots \hat{c}_{i_n} \quad (31)$$

which expresses the excited state  $|\Psi_K\rangle$  in terms of ground state wavefunction:

$$|\Psi_K\rangle = R_K e^T |\Phi\rangle \quad (32)$$

The coefficients determining excitation operator  $R_K$  and corresponding excited state energy  $E_K$  are again obtained by a projection techniques applied to Schrödinger equation.

For practical applications of CC methodology the chain of Eq. (30) that defines cluster amplitudes is disentangled by truncating the expansion of cluster operator to a certain order. For example, in the well-known CCSD (CC with singles and doubles) [85] and CCSDT [86,87] (CC with singles, doubles and triples) approaches the cluster operators are approximated as follows:

$$T \approx T_1 + T_2 \text{ (CCSD)} \quad (33)$$

$$T \approx T_1 + T_2 + T_3 \text{ (CCSDT)} \quad (34)$$

Likewise truncation of the excitation operator  $R_K$  at double and triple excitations leads to EOMCCSD [82–84] and EOMCCSDT [88] methods for excited states.

While the inclusion of the higher order cluster excitations leads to improved accuracy, it is also accompanied by a significant increase in computational demands. The numerical cost of the CCSD method scales as  $N^6$  with respect to system size ( $N$ ), increasing to  $N^8$  for CCSDT. This makes the CCSDT calculations prohibitively expensive for large systems. Several methods have been developed that can circumvent these difficulties. Arguably the most efficient way to include the triply excited configurations in ground and excited state CC calculations can be found in the noniterative approaches [89–96]. These methods take advantage of  $T_1$ ,  $T_2$  ( $R_{K,0}$ ,  $R_{K,1}$ ,  $R_{K,2}$ ) amplitudes obtained in less expensive CCSD (EOMCCSD) calculations to estimate the  $T_3$  ( $R_{K,3}$ ) amplitudes using many-body perturbation theory. The resulting CCSD(T) approach reduces the  $n_o^3 n_u^5$  cost of the CCSDT method down to  $n_o^3 n_u^4$ , where  $n_o$  is the number of occupied spin-orbitals and  $n_u$  is the number of unoccupied spin-orbitals.

The accuracy of CCSD(T), however, may become inadequate for stretched bond regions. In that case a more sophisticated route for approximation of  $T_3$  ( $R_{K,3}$ ) can be taken using perturbative methods based on the similarity transformed Hamiltonian [92–95,97–108] or method of moments formulation of coupled cluster equations [109]. The latter includes the completely renormalized CCSD[T] and CCSD(T) approaches (CR-CCSD[T], CR-CCSD(T)) [109], locally renormalized CCSD(T) and CCSD(TQ)-1 methods (LR-CCSD(T), LR-CCSD(TQ)-1) [110], very efficient method CR-CC(2,3) and CR-CC(2,4) methods [111–113], and generating functional methods [114,115].

Active-space methods [116–118] represent another strategy for approximate inclusion of triply excited configurations. In this case the essential parts of the exact  $T_3$  operator are elegantly represented using the so-called  $t_3$  operator based on active orbitals concept

$$|\Psi^{CCSDt}\rangle = e^{T_1 + T_2 + t_3} |\Phi\rangle \quad (35)$$

In the basic version (referred here as to “version I”) this operator is defined only by  $t_{ABC}^{ijk}$  amplitudes with  $i, j, k(a, b, c)$  representing generic occupied (unoccupied) spin-orbitals and  $I, J, K(A, B, C)$  indices refer to active occupied (unoccupied) spin-orbitals. The numerical overhead of the CCSDT method in this case is equal to  $N_o N_u n_o^2 n_u^4$ , where  $N_o(N_u)$  refers to occupied (unoccupied) active spin-orbitals. This presents significant computational savings compared to the  $n_o^3 n_u^5$  cost of the full CCSDT method. Further reduction in numerical costs can be achieved by using  $t_{ABC}^{ijk}$  and  $t_{ABC}^{JK}$  amplitudes, which corresponds to Versions II and III CCSDT methods.

A computational implementation of CC description is generally based on local basis expansion and composed of several steps characterized by various degree of complexity. The first step is associated with calculating the relevant reference wavefunction  $|\Phi\rangle$  usually using Hartree–Fock or DFT methods. The second step involves transformation of one- and two-electron integrals in terms of molecular orbitals associated with the reference wavefunction. This process, characterized by the  $N^5$  complexity is referred to as the 4-index transformation. Having obtained all integrals, in the subsequent step the nonlinear CC equations are solved iteratively for cluster amplitudes. The simplest iterative scheme (Jacobi iterations) is based on the following representation of the CC equations:

$$\left( \sum_{k=1}^n \epsilon_{a_k} - \sum_{k=1}^n \epsilon_{i_k} \right) t_{i_1 \dots i_n}^{a_1 \dots a_n} + \langle \Phi_{i_1 \dots i_n}^{a_1 \dots a_n} | (V_N e^T)_C | \Phi \rangle = 0 \quad (36)$$

where  $\epsilon$ 's refer to the orbitals energies (for simplicity Fock operator is assumed to be in the diagonal form) and  $V_N$  refers to the two-body part of electronic Hamiltonian in normal product form. Using the above equations cluster amplitudes in  $(n+1)$ th iterative step can be conveniently expressed through amplitudes obtained in  $n$ -iterative step (with corresponding  $T^{(n)}$  cluster operator)

$$t_{i_1 \dots i_n}^{a_1 \dots a_n(n+1)} = \frac{1}{\sum_{k=1}^n \epsilon_{i_k} - \sum_{k=1}^n \epsilon_{a_k}} \langle \Phi_{i_1 \dots i_n}^{a_1 \dots a_n} | (V_N e^{T^{(n)}})_C | \Phi \rangle \quad (37)$$

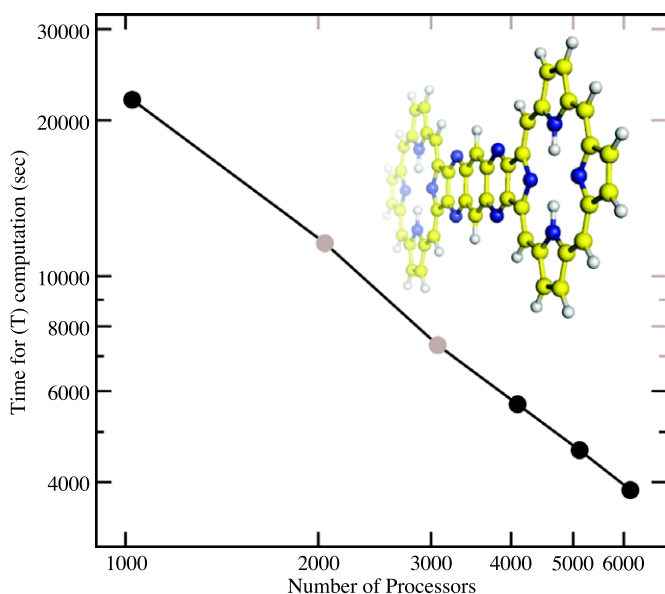
Using this procedure and employing initial guesses for cluster amplitudes equal to zero one can easily track the contributions coming from various orders of perturbative expansion. Once convergence is achieved the cluster amplitudes are used to calculate the energies. In the excited state CC calculations there is a fourth step which involves diagonalization of the similarity transformed Hamiltonian  $\hat{H}$  ( $\hat{H} = e^{-T} He^T$ ) in order to calculate excited-state energies and corresponding  $R_K$ -amplitudes.

NWChem provides a full featured of implementation of CC theory geared towards efficient execution on high performance parallel architectures.

The first implementation of the CCSD and CCSD(T) approaches [119] was developed for closed-shell systems utilizing spin-free formalism for restricted Hartree–Fock reference function. Compared to the spinorbital formalism, this rudimentary form of the spin adaptation offers a significant reduction in the number of cluster amplitudes. Recently, it was demonstrated that this highly efficient parallel implementation can scale across 240 000 cores [120].

Current development is mostly focused on CC implementation which can utilize any type of single-determinantal reference function using an improved version of the Tensor Contraction Engine (TCE) developed by Hirata [57,96,121]. It was subsequently modified to improve management of the look-up tables and increase parallel scaling of CCSD/EOMCCSD and CCSD(T)/CR-CCSD(T)/CR-EOMCCSD(T) modules. The latter included reduction of multiple uses of the largest class of two-electron integrals corresponding to the so-called 4-particles integrals (i.e., carrying four particle indices). As a by-product we substantially reduced the maximum size of the recursive intermediates down to  $n_o^2 n_u^2$ . These changes





**Fig. 3.** Parallel scaling of the most computationally expensive  $N^7$  (T) part of the CR-EOMCCSD(T) calculations across 6144 CPUs performed for oligoporphyrin dimer ( $P_2TA$  system [122]) using 578 basis set functions.

were accompanied by improved handling of 2-electron integrals for the restricted Hartree–Fock (RHF) and restricted open shell Hartree–Fock (ROHF) cases (available by using the 2eorb keyword in the tce input group). In these cases, the integrals are stored in the orbital form and inherit the tile structure used in the CC/EOMCC calculations. Fig. 3 shows parallel scaling of the most computationally expensive  $N^7$  (T) part of the CR-EOMCCSD(T) calculations across 6144 CPUs performed for oligoporphyrin dimer ( $P_2TA$  system [122]) using 578 basis set functions. The most recent scalability tests show a good scalability of the triples part of the CR-EOMCCSD(T) approach across 34 000 cores [123].

The parameters controlling coupled cluster calculations in NWChem are provided in the “tce” input block. For example, the basic input for performing ground CCSD calculations is given by

```
tce
  ccscd
end
```

The same input for excited state EOMCCSD calculations takes a slightly different form, requiring specification of the number of excited state roots to be calculated:

```
tce
  ccscd
  nroots 3
end
```

In all cases the convergence threshold can be controlled by the “thresh” keyword. For large parallel calculations it is critical to define the “tilesize” parameter, which defines the decomposition of the orbital basis set (according to spin and spatial symmetry) and the size of the building blocks of all tensors used in the CC calculations. Defining “tilesize” too big may result in problems with local memory, while defining “tilesize” too small may cause significant worsening in the performance of the CC codes due to increased communication. Typically, setting the “tilesize” value to 15–20 will provide good performance for the CCSD(T)/CR-EOMCCSD(T) [124] calculations. In the iterative CCSD/EOMCCSD calculations larger tilesizes (around 30) can be used. The newly implemented CCSDT

approaches can be invoked in NWChem by using “ccsdta” directive followed by “t3a\_lvl” keyword defining the version to be used and “active\_oa”, “active\_ob”, “active\_va”, and “active\_vb” keywords specifying the number of occupied alpha, occupied beta, unoccupied alpha, and unoccupied beta active spinorbitals. For example, invoking the following sequence in the tce group (RHF reference is assumed):

```
tce
  freeze atomic
  ccscdta
  t3a_lvl 2
  active_oa 4
  active_ob 4
  active_va 2
  active_vb 2
  nroots 2
end
```

will result in execution of version II of active-space CCSDT/EOMCCSDT approaches with four HOMO and two LUMO orbitals defined as active. Large CC calculations for the RHF or ROHF references will greatly benefit from usage of the “2eorb” directive, which allows the two-electron integrals to be stored in a more efficient orbital form. This option works only with Global Arrays as an input–output scheme (which is assumed by a default) and for the RHF or high-spin ROHF references.

## 6. Classical force field calculations

The molecular dynamics (MD) module in NWChem provides a highly scalable parallel framework for molecular simulations based on classical potentials or force fields. The default force field is based on the Amber-type [125] representation:

$$E = \sum_r k_r (r - r_0)^2 + \sum_\theta k_\theta (\theta - \theta_0)^2 + \sum_{n,\phi} \frac{V_n}{2} [1 + \cos(n\phi - \gamma)] + \sum_{i < j} \frac{q_i q_j}{r_{ij}} + \sum_{i < j} \epsilon \left[ \left( \frac{r_m}{r_{ij}} \right)^{12} - 2 \left( \frac{r_m}{r_{ij}} \right)^6 \right] \quad (38)$$

The first three terms describe bonded, angle, and dihedral interactions, and the electrostatic and van der Waals interactions are contained in the last two terms. The molecular dynamics capabilities in NWChem are contained in five related modules PREPARE, CAFE, SPACE, NWMD, and ANALYSIS.

The main purpose of the PREPARE module is to transform the initial structural description of the molecular system in terms of Protein Data Bank (PDB) files into a format suitable for classical force field simulation. In particular this involves assignment of the atom types, connectivity, and force field interaction parameters. The conversion process is facilitated by the parameter (extension .par) and fragment (extension .frg) files. The parameter file contains all the force field constants, except partial charges. These constants are referenced with respect to atom types which are different from the atom names in the actual PDB file. The actual mapping between the atom names and atom types is handled by the fragment files, which also define connectivity, partial charges, and polarizabilities. The fragment files are defined on a per-residue basis so that there is a unique fragment file for each residue found in the PDB file. NWChem has an extensive library of fragment files that covers standard amino acid residues, nucleotides, and some organic molecules. For all other cases the user is responsible for creating a new fragment file. The successful completion

of PREPARE run will result in the creation of topology (extension .top) and restart (extension .rst) files. The topology file contains the static system definitions that do not change during a simulation. These include the bonded, third neighbor and excluded pair lists, all force field parameters and other persistent data needed. The restart file contains the dynamic information of the system that will continuously change during a simulation such as coordinates and forces. The restart file is periodically overwritten during a molecular simulation and together with topology file provides all data required to resume a simulation.

The actual classical molecular simulation is performed using the NWMD module that contains routines for energy minimization, molecular dynamics, and free energy calculations. The NWMD module serves as the top level driver, and relies on the CAFE module for the calculation of the classical atomic forces and energies and the SPACE module for all the spatial decomposition, atom distribution and communication tasks. This modular implementation allows for ease of maintenance and future implementations of alternate methods. Molecular dynamics simulations routinely result in large molecular trajectories for further analysis. NWChem includes the ANALYZE module for such analyses including root mean square deviation (RMSD), internal coordinate analyses such as Ramachandran distributions, and essential dynamics analysis. To provide the best theoretical scalability of memory use and communication costs, the parallelization strategy of the NWChem MD module is based on a domain decomposition approach [126,127]. In the domain decomposition model the physical space occupied by the molecular system is distributed, and the atoms are assigned to a given processor based on their physical location. In particular, the total molecular volume for the simulation is divided by slicing the total system in each of the three dimensions. A combination of input parameters that include the cutoff radius for the atomic interactions and the number of processors available for the simulation determines the size of the resulting rectangular boxes that are subsequently assigned to those processors. This procedure fully supports the use of periodic boundary conditions. To optimize the communication efficiency, the processors are themselves logically arranged on a three-dimensional grid as well. Each processor can be assigned one or more sub-domains. This arrangement is used to preserve as much as possible locality of sub-domains that will have atoms within the cutoff distance. The domain decomposition approach has several advantages. The replication of atomic data on each processor is avoided, reducing the memory required to hold the entire system in core. A judicious choice of the decomposition of the physical space in conjunction with distance cutoff criteria for the evaluation of atomic interactions reduces communications only to neighboring processors. This strategy has been shown to be particularly effective for large molecular systems on computing resources with large numbers of processors. However, there are some complicating factors related to the implementation of domain decomposition approach. Reassignments are periodically required to deal with atoms that move from one processor's sub-domain to another. In addition a sophisticated dynamic load balancing scheme is needed to handle heterogeneous systems. To effectively handle situations where interactions extend beyond the immediate neighboring processors, NWChem is not using the shift algorithm for the communication of atomic coordinates between processors, but takes advantage of the one-sided asynchronous operations permitted by the Global Array toolkit. This also allows for more sophisticated dynamic load balancing algorithms. By making the size of sub-domains depend on the cutoff distance, the communication of data that will not be used in the force calculations is minimized. While the code has the ability to determine the optimal number of sub-domains and their size, the user has the option to take full control over these parameters. The evaluation of forces and energies is done in succession for each sub-domain pair that

could contain atoms within the cutoff distance. For each of these pairs the required coordinate data is retrieved from the appropriate processor and an atom based pair-list is created and stored for multiple time steps in the simulation. These pair lists are optionally periodically re-evaluated, or when atom redistribution or dynamic load balancing is performed.

A particular challenge for molecular dynamics simulations using a cutoff distance for nonbonded interactions is the neglect of important contributions from parts of the system beyond this cutoff distance. This is particularly true for electrostatic contributions. In NWChem a correction for the use of cutoff distances for electrostatic interactions is implemented in the form of the smooth Particle Mesh Ewald (sPME) method [128]. This method involves setup of a charge grid used to determine contributions in reciprocal space. This involves two three-dimensional fast Fourier transforms (3D-FFT) to be carried out for each force evaluation. Because of the data access patterns in Fourier transforms, the 3D-FFT method is difficult to efficiently parallelize. In NWChem a slab based 3D-FFT is used that allows the transform in two dimensions to take place on a single processor, thereby avoiding interprocessor communication. This leaves one dimension in which communication is required. The primary difficulty in scaling this algorithm is that it only allows as many processors to be used for the 3D-FFT as there are slabs in the grid. The number of grid points is typically determined by the requirements of the simulation. In NWChem, therefore, the evaluation of forces and energies is organized such that a sub-set of the processors are involved in the 3D-FFT calculations while the remaining processors execute the code for the real space components. This leads to an effective use of the available processing resources, even though the construction and use of the slab-wise grids with a block based atomic decomposition is a significant challenge.

One of the distinguishing features of the NWChem Molecular Dynamics module is the implementation of dynamic load balancing. This is particularly important for simulations of systems with a significantly heterogeneous composition leading to a heterogeneous processor work load. It also is important for spatial decompositions that do not fit nicely onto the number of processors available for the calculation. In NWChem two ways of dynamic load balancing have been implemented and can be used in combination. The first is a global technique that reduces the physical size of the sub-domain on the busiest processor. This automatically leads to larger sub-domains on the other processors. This is a global technique that requires a redistribution of the atoms. As a result it requires significant communication. The second technique is local. The responsibility for the calculation of forces between atoms on two processors is transferred from the busiest processor to the other one. This can be used in NWChem in a cascading fashion, such that multiple processor pairs can switch this inter-processor responsibility simultaneously. Since no atoms change processor domain, this method does not require significant communication. A second advantage of this method is that the order in which atomic forces and energies are evaluated is not changed. Therefore, this load balancing method has a much smaller effect on numerical differences between runs that could lead to diverging trajectories. Together, the use of these methods leads to a significant increase in load balance.

## 7. Combined quantum-mechanical molecular mechanics calculations

The combined quantum-mechanical molecular mechanics (QM/MM) approach provides a simple and effective tool to study localized molecular transformations in large scale systems such as those encountered in solution chemistry or enzyme catalysis. In this method an accurate but computationally intensive quantum-

mechanical (QM) description is only used for the regions where electronic structure transformations are occurring (e.g. bond making and breaking). The rest of the system, whose chemical identity remains essentially the same, is treated at the approximate classical molecular mechanics (MM) level. The total energy of the system is then given by the sum of the quantum ( $E_{qm}$ ) and classical energies ( $E_{mm}$ ):

$$E = E_{qm}[\mathbf{r}, \mathbf{R}; \psi] + E_{mm}[\mathbf{r}, \mathbf{R}] \quad (39)$$

where  $\mathbf{r}$ ,  $\mathbf{R}$  represent the coordinates of QM and MM regions respectively, and  $\psi$  denotes the ground state electronic wavefunction of the QM region. The QM energy can be conveniently separated into internal and external contributions [129]:

$$E_{qm} = E_{qm}^{\text{int}}(\mathbf{r}; \psi) + E_{qm}^{\text{ext}}(\mathbf{r}, \mathbf{R}; \rho) \quad (40)$$

The internal part  $E_{qm}^{\text{int}}(\mathbf{r}; \psi)$  is the gas phase energy expression. The external part contains the electrostatic interactions of the classical charges ( $Z_i$ ) of the MM region with the electron density ( $\rho$ ):

$$E_{qm}^{\text{ext}}(\mathbf{r}, \mathbf{R}; \rho) = \sum_i \int \frac{Z_i \rho(\mathbf{r}')}{|\mathbf{R}_i - \mathbf{r}'|} d\mathbf{r}' \quad (41)$$

The last term in Eq. (39), the classical energy ( $E_{mm}$ ), contains all the classical interactions in the MM system, as well as the solute-solvent van der Waals and Coulomb nuclear interactions. Clean separation between QM and MM region is not always possible and in the case when the boundary cuts across chemical bonds link atoms or other bond capping schemes [130,131] have to be utilized.

The QM/MM module in NWChem is built as a top level interface between the classical MD module and various QM modules, managing initialization, data transfer, and various high level operations such as optimizations. The size of the system ( $10^3$ – $10^5$  atoms) and the need for classical force field parameters precludes description of the system through just the geometry input block as would be done in pure QM simulations. Instead the initial setup is performed similar to classical MD simulations through the PREPARE module. This requires a properly formatted PDB file, a parameter file containing information about the force field, and fragment files. The latter provides the mapping between entries in the PDB file and force field parameters in the parameter file. It is at this PREPARE stage where the QM region is defined on an atom-by-atom or residue basis. Further system modification can also be performed at this point, including addition of solvent and constraints. In a typical setting this preparation stage will be done in a separate calculation from the main QM/MM simulations resulting in the generation of topology and restart files. The topology file contains a list of all relevant force field interactions encountered in the system but has no information about the actual atom positions. Typically the topology file will be generated once and reused throughout the entire simulation. The actual structural information about the system is contained in the restart file, which will be changing as the system coordinates are updated during the course of the simulation.

Once restart and topology files are generated, the QM/MM simulation can be initiated by defining the specifics of the QM and MM descriptions as described in the earlier sections, and if necessary QM/MM interface parameters. Among the important QM/MM interface parameters that apply to all the tasks are the definition of the link atom scheme and classical charge interactions. The current implementation supports two different treatments of link atoms – hydrogen and pseudo-carbon capping schemes [131]. The actual choice is controlled by the keyword `link_atoms`. The default option (`link_atoms hydrogen`) is to use the hydrogen capping

scheme which places a hydrogen link atom ( $\mathbf{R}_{link}$ ) along the bond formed by the quantum ( $\mathbf{R}_{qm}$ ) and classical ( $\mathbf{R}_{mm}$ ) atoms:

$$\mathbf{R}_{link} = (1 - g)\mathbf{R}_{qm} + g\mathbf{R}_{mm} \quad (42)$$

where  $g = 0.709$ . In the case of the pseudo-carbon scheme (`link_atoms halogen`) a fluoride-like atom takes the place of the classical atom ( $\mathbf{R}_{qm}$ ). The choice of the effective core basis required for this scheme is controlled by the keyword `link_ecp` and can be set automatically (default option) or specified by the user. The classical charge interaction parameters, specified by the keywords `bqzone` and `mm_charges`, define the classical region that is allowed to interact with the QM part through long range Coulomb interactions. The typical choice involves inclusion of classical atoms within a specified distance from the QM region with various other options available.

Following the general logic of the NWChem input file, the actual QM/MM calculation is initiated by the task directive:

```
task qmmm <string theory> <string operation>
```

The `theory` keyword specifies the level of QM theory, which may include HF, DFT, MP2, as well as various flavors of coupled cluster theories as provided by the TCE module. The `operation` keyword could be energy, gradient, optimize, saddle, frequencies, property, or dynamics. The energy task can be performed for both ground and excited states. The latter implementation is limited to TD-DFT and EOMCC theories provided by the TCE module. The `optimize` operation follows a micro-iteration strategy where optimization of the QM and MM regions are alternated until self-consistency is reached. To expedite the optimization of the MM region, the fixed QM region is represented by the electrostatic potential (ESP) charges. The same strategy can be employed during a saddle operation, which searches for saddle points on the potential energy surface. Additional tasks that are available within the QM/MM module are the calculations of free energies [129,132–135] and thermal averages of ground state and excited state properties [136]. The user interface for these tasks is currently being finalized and will be reported elsewhere.

## 8. Concluding remarks

New capabilities and improvements in the parallel algorithms are continually being added to NWChem to enable computational modeling of large and complex scientific problems using latest high performance hardware. Thus the development strategy for NWChem in the coming years will be focused on parallel performance, science capabilities, and open source. In the area of parallel performance the focus is on the efficient utilization of petascale and upcoming exascale architectures. This includes incorporating higher levels of task parallelism with dynamic load balancing, multiple levels of parallelism, algorithms with less disk dependence, and the use of new classes of hardware such as graphical processing units (GPUs). Many of the scientific capability developments involve new types of algorithms that can exploit petascale architectures in different ways than the traditional quantum chemistry and molecular dynamics methodologies, and can benefit from new applied math methodologies.

The expansion of the scientific capabilities NWChem continues to be focused on the development and implementation of new methods for accurate kinetics and dynamics of chemical transformation, chemistry of the interfaces and other complex environments in the condensed phase, and spanning longer time regimes with dynamics. Reliable and efficient description of the kinetics and dynamics of chemical processes, including discovery of novel reaction pathways is of crucial importance to many scientific areas related to energy production and clean fuel technology. Many

of technologically important processes occur in complex heterogeneous environments which require methods that scale with the system size, and can bridge from atomic to the macroscopic scales. In many cases it also becomes important to probe longer time domains (e.g. the dynamics of large conformational changes in biological systems) which, for the most part, are still outside the reach of current molecular simulation codes.

The latest release of NWChem brings the software into the open-source environment, with future releases, development directions, and quality control governed by a consortium of developers. NWChem will be released using the Educational Community License. Both the new scientific capability development and the need to maintain scalability of the software on next-generation exascale architectures are major efforts requiring resources, man-power, and expertise that exceed any one group. The open-source environment enables NWChem to become a real community development code with extensive participation and collaboration from external developers in the computational chemistry, computer science, and applied math arena. Increased collaborative research and development will ensure that NWChem will be able to make effective use of the next-generation hardware architectures. New and improved capabilities developed by the community can quickly be dispersed to its user community.

## Acknowledgements

The funding sources for the development of NWChem and the parallel software tools over the lifetime of the project include the division of Chemical Sciences, the Office of Basic Energy Sciences, Mathematical, Information, and Computational Sciences division of the Office of Advanced Scientific Computing Research, the Office of Naval Research, the U.S. DOE High Performance Computing and Communications Initiative, the Environmental and Molecular Sciences Laboratory, the Construction Project, D-384 and Operations, the Office of Biological and Environmental Research. A portion of the research was performed using EMSL, a national scientific user facility sponsored by the Department of Energy's Office of Biological and Environmental Research and located at Pacific Northwest National Laboratory. PNNL is operated by Battelle Memorial Institute for the U.S. Department of Energy. This research used resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. T.L.W. acknowledges funding from an NSF PetaApps grant. The submitted manuscript has been authored by a contractor (E.A.) of the U.S. Government under Contract No. DE-AC05-00OR22725.

## References

- [1] J. Nieplocha, R.J. Harrison, *Journal of Supercomputing* 11 (1997) 119–136.
- [2] J. Nieplocha, B. Palmer, V. Tipparaju, M. Krishnan, H. Trease, E. Apra, *International Journal of High Performance Computing Applications* 20 (2006) 203–231.
- [3] R.J. Harrison, M.F. Guest, R.A. Kendall, D.E. Bernholdt, A.T. Wong, M. Stave, J.L. Anchell, A.C. Hess, R.J. Littlefield, G.L. Fann, J. Nieplocha, G.S. Thomas, D. Elwood, J.L. Tilson, R.L. Shepard, A.F. Wagner, I.T. Foster, E. Lusk, R. Stevens, *J. Comp. Chem.* 17 (1996) 124–132.
- [4] L.S. Blackford, J. Choi, A. Cleary, E. D'Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, R. Whaley, *ScaLAPACK Users' Guide*, Society for Industrial and Applied Mathematics, 1997.
- [5] W. Kohn, L.J. Sham, *Phys. Rev. A* 140 (1965) 1133–1138.
- [6] R. Dreizler, E. Gross, *Density Functional Theory*, Plenum Press, 1995.
- [7] R.G. Parr, W. Yang, *Density-Functional Theory of Atoms and Molecules*, Oxford Science Publications, 1989.
- [8] L.J. Sham, *Phys. Rev. B* 32 (1985) 3876–3882.
- [9] M. Valiev, G.W. Fernando, *Physics Letters A* 227 (1997) 265–270.
- [10] J.P. Perdew, K. Burke, M. Ernzerhof, *Phys. Rev. Lett.* 77 (1996) 3865.
- [11] J.P. Perdew, J.A. Chevary, S.H. Vosko, K.A. Jackson, M.R. Pederson, D.J. Singh, C. Fiolhais, *Phys. Rev. B* 46 (1992) 6671.
- [12] A.D. Becke, *Phys. Rev. A* 88 (1988) 3098.
- [13] C. Lee, W. Yang, R.G. Parr, *Phys. Rev. B* 37 (1988) 785.
- [14] P. Stephens, F. Devlin, C. Chabalowski, M. Frisch, *J. Phys. Chem.* 98 (1994) 11623.
- [15] C. Adamo, V. Barone, *J. Chem. Phys.* 110 (1999) 6158–6170.
- [16] J. Perdew, K. Schmidt, in: V. VanDoren, C. VanAlsenoy, P. Geerlings (Eds.), *Density Functional Theory and Its Application to Materials*, International Conference on Density Functional Theory and Its Applications to Materials, Antwerp, Belgium, June 08–10, 2000, in: *AIP Conference Proceedings*, vol. 577, Amer. Inst. Physics, 2001, pp. 1–20.
- [17] Y. Tawada, T. Tsuneda, S. Yanagisawa, T. Yanai, K. Hirao, *J. Chem. Phys.* 120 (2004) 8425.
- [18] E. Livshits, R. Baer, *Phys. Chem. Chem. Phys.* 9 (2007) 2932.
- [19] N. Govind, M. Valiev, L. Jensen, K. Kowalski, *J. Phys. Chem. A* 113 (2009) 6041.
- [20] L. Jensen, N. Govind, *J. Phys. Chem. A* 113 (2009) 9761.
- [21] S. Kummel, L. Kronik, *Rev. Mod. Phys.* 80 (2008) 3–60.
- [22] M. Marques, E. Gross, in: C. Fiolhais, F. Nogueira, M. Marques (Eds.), *Primer in Density Functional Theory*, in: *Lecture Notes in Physics*, vol. 620, Univ. Coimbra, Ctr. Computat. Phys., Springer-Verlag, Berlin, 2003, pp. 144–184.
- [23] R.M. Martin, *Electronic Structure: Basic Theory and Practical Methods*, Cambridge University Press, 2004.
- [24] G.E. Scuseria, V.N. Staroverov, *Theory and Applications of Computational Chemistry: The First Forty Years*, Elsevier, Amsterdam, 2005, pp. 669–724.
- [25] J. Kohanoff, *Electronic Structure Calculations for Solids and Molecules: Theory and Computational Methods*, Cambridge University Press, 2006.
- [26] S. Boys, *Proc. R. Soc. London, Ser. A* 200 (1950) 532.
- [27] K.L. Schuchardt, B.T. Didier, T. Elsethagen, L.S. Sun, V. Gurumoorthi, J. Chase, J. Li, T.L. Windus, *Journal of Chemical Information and Modeling* 47 (2007) 1045–1052.
- [28] A. Szabo, N. Ostlund, *Modern Quantum Chemistry*, Dover Publications, Inc., 1996.
- [29] R. Harrison, M. Guest, R. Kendal, D. Bernholdt, A. Wong, M. Stave, J. Anchell, A. Hess, R. Littlefield, G. Fann, J. Nieplocha, G. Thomas, D. Elwood, J. Tilson, R. Shepard, A. Wagner, I. Foster, E. Lusk, R. Stevens, *J. Comp. Chem.* 17 (1993) 124.
- [30] F. Jensen, *Introduction to Computational Chemistry*, John Wiley & Sons, 1999.
- [31] B. Dunlap, J. Connolly, J. Sabin, *J. Chem. Phys.* 71 (1979) 3396.
- [32] M.V. Arnim, R. Ahlrichs, *J. Comp. Chem.* 19 (1998) 1746.
- [33] R. Stratmann, G. Scuseria, M. Frisch, *Chem. Phys. Lett.* 257 (1996) 213.
- [34] E. Apra, E.J. Bylaska, D.J. Dean, A. Fortunelli, F. Gao, P.S. Krstic, J.C. Wells, T.L. Windus, *Computational Materials Science* 28 (2003) 209–221.
- [35] W.J. Hehre, R. Ditchfield, J.A. Pople, *J. Chem. Phys.* 56 (1972) 2257–2261.
- [36] P.C. Hariharan, J.A. Pople, *Theoretical Chemistry Accounts: Theory, Computation, and Modeling*, *Theoretica Chimica Acta* 28 (1973) 213–222.
- [37] P. Pulay, *Chem. Phys. Lett.* 73 (1980) 393.
- [38] P. Pulay, *J. Comp. Chem.* 3 (1982) 556.
- [39] M.F. Guest, V. Saunders, *Mol. Phys.* 28 (1974) 819.
- [40] R. Warren, B. Dunlap, *Chem. Phys. Lett.* 262 (1996) 384.
- [41] Q. Wu, W. Yang, *J. Chem. Phys.* 116 (2002) 515.
- [42] U. Zimmerli, M. Parrinello, P. Koumoutsakos, *J. Chem. Phys.* 120 (2004) 2693.
- [43] S. Grimme, *J. Comp. Chem.* 25 (2004) 1463.
- [44] S. Grimme, *J. Chem. Phys.* 124 (2006) 034108.
- [45] H. Iikura, T. Tsuneda, T. Yanai, K. Hirao, *J. Chem. Phys.* 115 (2001) 3540.
- [46] T. Yanai, D.P. Tew, N.C. Handy, *Chem. Phys. Lett.* 393 (2004) 51.
- [47] S. Grimme, *J. Comp. Chem.* 27 (2006) 1787.
- [48] B. Hess, *Phys. Rev. A* 32 (1985) 756.
- [49] B. Hess, *Phys. Rev. A* 33 (1986) 3742.
- [50] W.A. de Jong, R.J. Harrison, D.A. Dixon, *J. Chem. Phys.* 114 (2001) 48.
- [51] C. Chang, M. Pelissier, M. Durand, *Physica Scripta* 34 (1986) 294.
- [52] E. van Lenthe, E. Baerends, J. Snijders, *J. Chem. Phys.* 99 (1993) 4597.
- [53] S. Faas, J. Snijders, J. van Lenthe, E. van Lenthe, E. Baerends, *Chem. Phys. Lett.* 246 (1995) 632.
- [54] P. Nichols, N. Govind, E. Bylaska, W. de Jong, *Journal of Chemical Theory and Computation* 5 (2009) 491.
- [55] M. Casida, *Recent Advances in Density Functional Methods*, vol. 1, World Scientific, Singapore, 1995, pp. 155–192.
- [56] S. Hirata, M. Head-Gordon, *Chem. Phys. Lett.* 314 (1999) 291.
- [57] S. Hirata, *J. Phys. Chem. A* 107 (2003) 9887–9897.
- [58] A.W. Lange, M.A. Rohrdanz, J.M. Herbert, *J. Phys. Chem. B* 112 (2008) 6304.
- [59] J. Andzelm, C. Rindspacher, A.M. Rawlett, J. Dougherty, R. Baer, N. Govind, *J. Chem. Theory Comput.* 5 (2009) 2835.
- [60] E.O. Brigham, *The Fast Fourier Transform*, Prentice Hall, 1974.
- [61] J.C. Phillips, *Phys. Rev.* 112 (1958) 685–695.
- [62] J.C. Phillips, L. Kleinman, *Phys. Rev.* 116 (1959) 287–294.
- [63] B.J. Austin, V. Heine, L.J. Sham, *Phys. Rev.* 127 (1962) 276–282.
- [64] M.T. Yin, M.L. Cohen, *Phys. Rev. B* 25 (1982) 7403–7412.
- [65] G.B. Bachelet, D.R. Hamann, M. Schluter, *Phys. Rev. B* 26 (1982) 4199–4228.
- [66] D.R. Hamann, *Phys. Rev. B* 40 (1989) 2980–2987.
- [67] L. Kleinman, D.M. Bylander, *Phys. Rev. Lett.* 55 (1985) 2471.
- [68] P.E. Blochl, *Phys. Rev. B* 50 (1994) 17953–17979.

- [69] N.A.W. Holzwarth, G.E. Mathews, A.R. Tackett, R.B. Dunning, *Phys. Rev. B* 57 (1997) 11827.
- [70] M. Valiev, J.H. Weare, *J. Phys. Chem. B* 103 (1999) 10588–10601.
- [71] G. Kresse, D. Joubert, *Phys. Rev. B* 59 (1999) 1758–1775.
- [72] E.J. Bylaska, M. Valiev, R. Kawai, J.H. Weare, *Comput. Phys. Comm.* 143 (2002) 11–28.
- [73] L.J. Clarke, I. Stich, M. Payne, *Comput. Phys. Comm.* 72 (1992) 14–28.
- [74] J.S. Nelson, S.J. Plimpton, M.P. Sears, *Phys. Rev. B* 47 (1993) 1765.
- [75] J. Wiggins, H. Jonsson, *Comp. Phys. Comm.* 81 (1994) 1–18.
- [76] D. Marx, J. Hutter, *Ab Initio Molecular Dynamics: Theory and Implementation*, NIC Series, vol. 1, Forschungszentrum, 2000, pp. 301–449.
- [77] F. Gygi, *IBM J. Res. Dev.* 52 (2008) 137–144.
- [78] J. Cizek, *J. Chem. Phys.* 45 (1966) 4256.
- [79] J. Paldus, I. Shavitt, J. Cizek, *Phys. Rev. A* 5 (1972) 50.
- [80] J. Paldus, X.Z. Li, *Critical Assessment of Coupled Cluster Method in Quantum Chemistry*, *Advances in Chemical Physics*, vol. 110, John Wiley & Sons, Inc., 1999, pp. 1–175.
- [81] R.J. Bartlett, M. Musial, *Rev. Mod. Phys.* 79 (2007) 291–352.
- [82] J. Geertsen, M. Rittby, R.J. Bartlett, *Chem. Phys. Lett.* 164 (1989) 57–62.
- [83] D.C. Comeau, R.J. Bartlett, *Chem. Phys. Lett.* 207 (1993) 414–423.
- [84] J.F. Stanton, R.J. Bartlett, *J. Chem. Phys.* 98 (1993) 7029–7039.
- [85] G.D. Purvis, R.J. Bartlett, *J. Chem. Phys.* 76 (1982) 1910–1918.
- [86] J. Noga, R.J. Bartlett, *J. Chem. Phys.* 86 (1987) 7041–7050.
- [87] G.E. Scuseria, H.F. Schaefer, *Chem. Phys. Lett.* 152 (1988) 382–386.
- [88] K. Kowalski, P. Piecuch, *J. Chem. Phys.* 115 (2001) 643–651.
- [89] K. Raghavachari, G.W. Trucks, J.A. Pople, M. Head-Gordon, *Chem. Phys. Lett.* 157 (1989) 479–483.
- [90] J.F. Stanton, J. Gauss, *J. Chem. Phys.* 103 (1995) 1064–1076.
- [91] J.F. Stanton, *Chem. Phys. Lett.* 281 (1997) 130–134.
- [92] T. Crawford, J. Stanton, *Int. J. Quantum Chem.* 70 (1998) 601–611.
- [93] S. Hirata, M. Nooijen, I. Grabowski, R.J. Bartlett, *J. Chem. Phys.* 114 (2001) 3919–3928.
- [94] S.R. Gwaltney, M. Head-Gordon, *Chem. Phys. Lett.* 323 (2000) 21–28.
- [95] S.R. Gwaltney, M. Head-Gordon, *J. Chem. Phys.* 115 (2001) 2014–2021.
- [96] S. Hirata, *J. Chem. Phys.* 121 (2004) 51–59.
- [97] J. Stanton, *Chem. Phys. Lett.* 281 (1997) 130.
- [98] J. Stanton, J. Gauss, *J. Chem. Phys.* 103 (1995) 1064.
- [99] J. Stanton, J. Gauss, *Theor. Chim. Acta* 93 (1996) 303.
- [100] S. Kucharski, R. Bartlett, *J. Chem. Phys.* 108 (1998) 5243.
- [101] S. Kucharski, R. Bartlett, *J. Chem. Phys.* 108 (1998) 5255.
- [102] S. Gwaltney, C. Sherrill, M. Head-Gordon, A. Krylov, *J. Chem. Phys.* 113 (2000) 3548.
- [103] S. Gwaltney, E. Byrd, T.V. Voorhis, M. Head-Gordon, *Chem. Phys. Lett.* 353 (2002) 359.
- [104] S. Hirata, M. Nooijen, I. Grabowski, R.J. Bartlett, *J. Chem. Phys.* 115 (2001) 3967.
- [105] Y. Bomble, J. Stanton, M. Kállay, J. Gauss, *J. Chem. Phys.* 123 (2005) 054101.
- [106] M. Kállay, J. Gauss, *J. Chem. Phys.* 123 (2005) 214105.
- [107] A.G. Taube, R.J. Bartlett, *J. Chem. Phys.* 128 (2008) 044110.
- [108] A. Taube, R.J. Bartlett, *J. Chem. Phys.* 128 (2008) 044111.
- [109] K. Kowalski, P. Piecuch, *J. Chem. Phys.* 113 (2000) 18–35.
- [110] K. Kowalski, *J. Chem. Phys.* 123 (2005) 9.
- [111] P. Piecuch, M. Wloch, *J. Chem. Phys.* 123 (2005) 224105.
- [112] P. Piecuch, M. Wloch, J. Gour, A. Kinal, *Chem. Phys. Lett.* 418 (2006) 467.
- [113] X. Li, J. Gour, J. Paldus, P. Piecuch, *Chem. Phys. Lett.* 461 (2008) 321.
- [114] K. Kowalski, P.D. Fan, *J. Chem. Phys.* 130 (2009) 084112.
- [115] K. Kowalski, M. Valiev, *J. Chem. Phys.* 121 (2009) 234107.
- [116] N. Oliphant, L. Adamowicz, *Int. Rev. Phys. Chem.* 12 (1993) 339–362.
- [117] P. Piecuch, N. Oliphant, L. Adamowicz, *J. Chem. Phys.* 99 (1993) 1875–1900.
- [118] K. Kowalski, P. Piecuch, *J. Chem. Phys.* 115 (2001) 643–651.
- [119] R. Kobayashi, A. Rendell, *Chem. Phys. Lett.* 265 (1996) 1.
- [120] E. Apra, R. Harrison, W. de Jong, A. Rendell, V. Tipparaju, S. Xantheas, R. Olsen, in: *ACM/IEEE Supercomputing 2009 Conference*.
- [121] S. Hirata, *Theoretical Chemistry Accounts* 116 (2006) 2–17.
- [122] K. Sendt, L.A. Johnston, W.A. Hough, M.J. Crossley, N.S. Hush, J.R. Reimers, *J. Am. Chem. Soc.* 124 (2002) 9299–9309.
- [123] K. Kowalski, S. Krishnamoorthy, O. Villa, J. Hammond, N. Govind, *J. Chem. Phys.* 132 (2010) 154103.
- [124] K. Kowalski, P. Piecuch, *J. Chem. Phys.* 120 (2004) 1715–1738.
- [125] W.D. Cornell, P. Cieplak, C.I. Bayly, I.R. Gould, K.M. Merz, D.M. Ferguson, D.C. Spellmeyer, T. Fox, J.W. Caldwell, P.A. Kollman, *J. Am. Chem. Soc.* 118 (1996) 2309.
- [126] T.P. Straatsma, J.A. McCammon, *IBM Systems Journal* 40 (2001) 328–341.
- [127] T.P. Straatsma, M. Philippopoulos, J.A. McCammon, *Comp. Phys. Comm.* 128 (2000) 377–385.
- [128] U. Essmann, L. Perera, M.L. Berkowitz, T. Darden, H. Lee, L.G. Pedersen, *J. Chem. Phys.* 103 (1995) 8577–8593.
- [129] M. Valiev, B.C. Garrett, M.-K. Tsai, K. Kowalski, S.M. Kathmann, G.K. Schenter, M. Dupuis, *J. Chem. Phys.* 127 (2007) 51102.
- [130] J.L. Gao, D.G. Truhlar, *Ann. Rev. Phys. Chem.* 53 (2002) 467–505.
- [131] Y.K. Zhang, H.Y. Liu, W.T. Yang, *J. Chem. Phys.* 112 (2000) 3483–3492.
- [132] M. Valiev, E.J. Bylaska, M. Dupuis, P.G. Tratnyek, *J. Phys. Chem. A* 112 (2008) 2713–2720.
- [133] B. Elsaesser, M. Valiev, J.H. Weare, *J. Am. Chem. Soc.* 131 (2009) 3869.
- [134] M. Valiev, R. D'Auria, D.J. Tobias, B.C. Garrett, *J. Phys. Chem. A* 113 (2009) 8823–8825.
- [135] M. Valiev, J. Yang, J.A. Adams, S.S. Taylor, J.H. Weare, *J. Phys. Chem. B* 111 (2007) 13455–13464.
- [136] M. Valiev, K. Kowalski, *J. Chem. Phys.* 125 (2006).