

The GAMESS-UK electronic structure package: algorithms, developments and applications

MARTYN F. GUEST^{†*}, IAN J. BUSH[†], HUUB J. J. VAN DAM[†], PAUL SHERWOOD[†], JENS M. H. THOMAS[†], JOOP H. VAN LENTHE[‡], REMCO W. A. HAVENITH[‡] and JOHN KENDRICK[§]

[†]Computational Science and Engineering Department, CLRC Daresbury Laboratory, Daresbury, Warrington, Cheshire WA4 4AD, UK

[‡]Theoretical Chemistry Group, Debye Institute, Utrecht University, Utrecht, The Netherlands
[§]QMolecular Ltd, 13 Castle Close, Middleton St George, Darlington DL2 1DE, UK

A description of the *ab initio* quantum chemistry package GAMESS-UK is presented. The package offers a wide range of quantum mechanical wavefunctions, capable of treating systems ranging from closed-shell molecules through to the species involved in complex reaction mechanisms. The availability of a wide variety of correlation methods provides the necessary functionality to tackle a number of chemically important tasks, ranging from geometry optimization and transition-state location to the treatment of solvation effects and the prediction of excited state spectra. With the availability of relativistic ECPs and the development of ZORA, such calculations may be performed on the entire Periodic Table, including the lanthanides. Emphasis is given to the DFT module, which has been extensively developed in recent years, and a number of other, novel features of the program. The parallelization strategy used in the program is outlined, and detailed speedup results are given. Applications of the code in the areas of enzyme and zeolite catalysis and in spectroscopy are described.

1. Introduction

The UK version of GAMESS (Generalised Atomic and Molecular Electronic Structure System) [1] has been under development for nearly a quarter of a century and this volume, in celebration of the contribution of Nicholas Handy, presents an ideal opportunity to describe the current status of the program. The code represents the principal piece of software developed and maintained under the auspices of Collaborative Computational Project No. 1 (CCP1). Nicholas Handy has, over this period, played a leading role in many aspects of this project, and we, the GAMESS-UK developers, owe a sincere debt of thanks to Nicholas for the support he has given to the concept of a freely available, centrally supported UK-based quantum chemistry code. Many parts of the program, specifically in the areas of perturbation theory for electron correlation and analytic derivatives, can be traced to developments in the Cambridge group. More recently, Nicholas has been the leading UK proponent of Density Functional Theory (DFT) in the field of molecular

quantum chemistry, as well as key contributor to the improvement of functionals for chemical problems and the techniques for its efficient implementation. His influence was key to the decision to set up a CCP1 flagship project in this area, and the results of this project are now maintained and distributed as part of GAMESS-UK, as well as being available for use in other projects. The DFT module in GAMESS-UK has been available since version 6.0 of the code, and is described in some detail in section 3 of this paper.

The article is organized as follows. An overview of the program is given in section 2, tracing the development of the code and outlining the possible wavefunctions that GAMESS-UK can calculate and the types of runs that can be performed. Section 3 describes the GAMESS-UK implementation of DFT. Section 4 focuses on the implementation aspects for a number of the more popular, and the more novel, features available within the code. These include the modules for (i) Geometry Optimization, (ii) the treatment of Relativistic Effects and ZORA, (iii) Valence Bond, (iv) MRDCI, (v) RPA and MCLR treatments of Excited States, (vi) QM/MM and the treatment of Large Systems and, finally, our recent work on graphical interfaces, also performed within the CCP1 programme. Details of the parallel

* Corresponding author. e-mail: m.f.guest@dl.ac.uk

implementation of GAMESS-UK and the associated performance attributes are given in section 5, while section 6 considers the applications of the code in a number of areas, including enzyme and zeolite catalysis and spectroscopy.

2. Overview of the program

The program is derived from the original GAMESS code, obtained from Michel Dupuis in 1981 (then at the National Resource for Computational Chemistry, NRCC), and has been extensively modified and enhanced over the past two decades. The key driver in our adopting the code lay in the availability of gradient capabilities, absent in the ATMOL suite of programs [2] that had been the main electronic structure code supported under CCP1. Note that these developments have been conducted independently of the impressive programme of extensions to the GAMESS code itself, conducted under the leadership of Mark Gordon and Mike Schmidt at Iowa State University [3].

When first acquired, the code was essentially limited to HF/gradient functionality allowing for basis sets involving s, p and d Cartesian Gaussian orbitals, with open- and closed-shell SCF treatments available within both the RHF and UHF framework. Generalized valence bond [4] treatments were also supported. The program utilized Rotation [5] techniques to evaluate repulsion integrals over s and p Gaussians, and the Rys Polynomial [6] for integrals involving d Gaussians. SCF convergence controls were provided through a hybrid scheme of level shifters and damping factors. The analytic energy gradient was available for the above wavefunctions, with gradients for s and p Gaussians evaluated using the algorithm due to Schlegel [7], while gradients involving d Gaussians utilized the Rys Polynomial Method [8]. Force constants were evaluated by numerical differentiation. *Ab initio* core potentials were provided in a semi-local [9–11] formalism for performing valence-only molecular orbital treatments.

Many of the initial developments around the code focused on enriching the range of available post-Hartree–Fock capabilities, with functionality originally developed within the ATMOL suite of programmes integrated into the code. Thus conventional CI treatments using the table-driven selection algorithms within the framework of MR-DCI calculations [12–14] allowed for the treatment of electronic spectra and related phenomena, while large-scale CI calculations of both ground and first few excited states were provided by the Direct-CI [15] module.

These initial correlation treatments have since been extended through the incorporation of Full-CI [16], and

both CCSD [17, 18] and CCSD(T) [19, 20] coupled-cluster calculations, although the latter remain limited to closed-shell systems [21]. A size-consistent variant of multi-reference MP2 theory, popularized in its CASPT2 form by Roos *et al.* [22], is also available [23]. With no restriction to CAS wavefunctions, the module also provides MR-MP3 capabilities. One of the more recent developments includes an implementation of the semi-direct table-driven MRDCI module [24, 25], providing for more extensive capabilities in the treatment of electronic spectra and related phenomena.

The treatment of both excited and ionized states has long been a major requirement from users of the code. In addition to the MRD-CI treatments above, calculations of electronic transition energies and corresponding oscillator strengths may be performed using either the Random Phase Approximation (RPA) method or the Multiconfigurational Linear Response (MCLR) procedure [26]. The RPA calculations may be performed either within the conventional approach where the two-electron integrals are transformed or with a ‘direct’ implementation. The direct calculation of molecular valence ionization energies may be performed through Green’s function techniques, using either the outer-valence Green’s function (OVGF) [27] or the two-particle-hole Tamm–Dancoff method (2ph-TDA) [28].

While employing effectively the same integral and gradient technology, the program has been extended to evaluate repulsion and the associated gradient integrals over f and g Gaussians. The original limitation to cartesian basis sets is lifted through the provision of spherical harmonic basis sets for all options within the programme. SCF controls now use a hybrid scheme of level shifters and the DIIS method [29, 30]. In addition, complete active space SCF [22, 31, 32] and more general MCSCF [33], and Møller Plesset (MP2 and MP3) [34, 35] calculations may now be performed. Geometry optimization is conducted using a quasi-Newton rank-two update method, while transition state location is available through either a synchronous transit [36], trust region [37] or ‘hill-walking’ [38, 39] method. Force constants may be evaluated analytically [40, 41], while coupled Hartree–Fock (CHF) calculations provide for a range of molecular properties, including polarizabilities and molecular hyperpolarizabilities [42] and, through the calculation of dipole moment and polarizability derivatives, the computation of infra-red and Raman intensities [43].

Many new *ab initio* core potentials have been incorporated into the code which now includes both semi-local and non-local [44] formalisms for valence-only molecular orbital treatments.

A wide variety of wavefunction analysis methods are available. These include population analysis, Natural

Bond Orbital (NBO) [45–47] and distributed multipole analysis [48], localized orbitals, graphical analysis and calculation of one-electron properties. An interface to the AIMPAC code of Bader is provided [49].

The treatment of solvation effects is based on the DRF (Direct Reaction Field) model [50–52], an embedding technique enabling the computation of the interaction between a quantum-mechanically described molecule and its classically described surroundings. The classical surroundings may be modelled in a number of ways, (i) by point charges to model the electrostatic field due to the surroundings, (ii) by polarizabilities to model the (electronic) response of the surroundings, (iii) by an enveloping dielectric to model bulk response (both static and electronic) of the surroundings, and (iv) by an enveloping ionic solution, characterized by its Debye screening length.

The treatment of relativistic effects is provided within the ZORA scheme (Zeroth Order Regular Approximation [53]), a two-component alternative to the full four-component Dirac equation. While much cheaper than the latter, ZORA recovers a large part of the relativistic effects. The scalar (one-component) form is now available, with the full two-component implementation (including spin–orbit coupling) in progress. The current implementation will allow all usual *ab initio* (and DFT) methods to be performed, incorporating the major relativistic effects. More details are presented in section 4.2.

The program structure is effectively open-ended in direct-SCF, -DFT and direct-MP2 mode, so that direct-SCF calculations of up to 10 000 basis functions have been performed. Such calculations make use of the extensive parallel capabilities, which are described in detail in section 5. In addition to the functionality outlined above, the program provides for Hybrid QM/MM calculations, through an interface to the ChemShell package and the CHARMM QM/MM code (see section 4.7).

Arguably the most important of the recent additions to the code, available from Version 6.0 onwards, is a full-featured Density Functional Theory (DFT) module, with access to a wide variety of functionals (S-VWN, B-LYP, B-P86, B3-LYP, HCTH, B97, B97-1, B97-2, PBE, EDF1, FT97, etc.). This module is subject to on-going developments under the auspices of CCPI, and will be described in some detail in the sections below.

As with any *ab initio* electronic structure code, the areas of application of GAMESS-UK are numerous. The code is designed to be a full-featured computational chemistry package serving a broad user community. As such, we have developed, or will be developing, a wide range of application modules for the package.

In the following sections we highlight a number of the implementation aspects of the more popular, and the more novel, features available within the code. Particular attention is focused on the developing DFT capabilities (section 3).

3. The GAMESS-UK implementation of Density Functional Theory

3.1. Background

Since Hohenberg and Kohn [54] laid down the foundations for a rigorous electronic theory requiring only the electron density, density functional theory (DFT) has enjoyed a steady increase in popularity. This popularity increased throughout the 1990s to such an extent that DFT has become a mainstream method for studying electronic structure problems in chemistry. The GAMESS-UK implementation originated with the CCP1 flagship project (1994–1997) which undertook to construct a DFT module to be built into a Hartree–Fock host code. This initial work, by Philip Young, has been extended continuously since the project ended.

3.2. Functionality

Currently the module supports the evaluation of the Kohn–Sham energy and matrix elements, plus the evaluation of the gradient of the energy with respect to the nuclear coordinates. All terms can be evaluated for GGA functionals. In all cases extensive use is made of screening and pruning techniques to exploit the local character of the density functionals. Finally, the evaluation of all terms is driven by the same integrator routine to ensure consistency of the quadrature throughout.

The code supports Dunlap’s approximation to the Coulomb repulsion energy in which the charge density can be fitted to an auxiliary basis [55, 56]:

$$\rho(r) = \sum_{pq} D_{pq} |pq\rangle \approx \sum_u \left(\sum_{pq} D_{pq} C_u^{pq} \right) |u\rangle = \sum_u d_u |u\rangle, \quad (1)$$

where the fitting coefficients C can be obtained from

$$C^{pq} = V^{-1} b^{pq}, \quad (2)$$

In this equation V_{uv} are the electron repulsion integrals in the charge density basis and b_i^{pq} are the three-centre electron repulsion integrals between the wavefunction basis set and the charge density basis. This way the evaluation of four-centre two-electron integrals can be circumvented using at most three-centre

two-electron integrals, clearly reducing the formal scaling of the computational cost from fourth order to third order. We can reduce the computational cost further by implementing a mechanism to store the three-centre two-electron integrals in main memory in a distributed fashion on parallel computers.

Second derivatives of the energy are computed [57] according to the formulation given by Johnson and Frisch [58], and the coupled perturbed Kohn–Sham (CPKS) equations can be solved for both external field as well as geometrical perturbations. Terms in the CPKS equations can be evaluated either in the AO or MO basis dependent on what is most efficient for the size of system under consideration, and the quadrature weights are differentiated to ensure they are true derivatives of the energy expression.

As GAMESS-UK is often used by non-experts some considerable effort has been invested in defining the predefined DFT quadratures and these are described in the next sections.

3.3. DFT quadrature

Developing a quadrature for DFT calculations is challenging because of three particular features of molecular electronic densities. First of all there is a cusp at the position of every atom, second the density decays exponentially with increasing distance to the atoms, and, third, in the inter-atomic region the density is smooth but can have a number of extrema due to contributions from various atoms. An adequate quadrature is one that can simultaneously address all three features in reasonable fashion.

Because quadrature development is so challenging it has long remained an active area of research and various general approaches have been proposed. There are two main strategies to meet the challenge. The first is to devise a scheme to partition space into distinct volumes and integrate each volume with a particular grid. Space can either be partitioned into spheres with some special partitioning for the space remaining between the spheres [59, 60], or into cubes which are recursively partitioned in further cubes until a specific accuracy is achieved [61]. The advantage of these approaches is that each volume element is integrated only once, leading to favourable scaling behaviour. The disadvantage is that either complex partitioning schemes or a large number of volume elements are needed to achieve high accuracy.

In contrast, Becke [62] proposed to partition the integrand into a number of atom-like terms. Each term can then be integrated with an appropriately selected grid in polar coordinates. The advantages of this scheme

are that it is simple and can achieve high accuracy. The downside is that the atom-like partitions will cover overlapping volumes of space, so that each volume of space will be integrated multiple times by the atomic grid centred on different atoms, leading to higher computational costs. Indeed, a Becke-type approach scales formally with at least the square of the number of atoms, although this can be addressed by appropriate approximations while maintaining its advantages.

For reference we briefly reiterate the important components of the Becke quadrature. The starting point is the nuclear-weight function which has to satisfy two main criteria. First, given a three-dimensional molecular integral, multiplying the integrand with the nuclear-weight function of a particular atom has to result in a suitably atom-like function called an atomic partition. Secondly, the nuclear-weight functions of all atoms have to add up to one for all points in space, so that the sum of all atomic partitions is equal to the full molecular integrand. Each atomic partition can be integrated using straightforward polar coordinate grids which are simple Cartesian products of suitably chosen angular and radial coordinate grids. The nuclear-weight function is obtained simply by assigning a cell-function to every atom and calculating the fraction its cell-function contributes to the total sum of cell-functions. Within these general principles a quadrature can be modified by making different choices for the cell-functions, the angular grids and the radial grids.

To apply the above scheme to a particular molecule, a set of grid points needs to be generated. Ideally this generation would be adaptive so that any accuracy can be obtained on demand. However the nuclear-weight function makes the on-the-fly error analysis needed to attribute a fraction of the overall error to specific grid parameters very hard. Alternatively, trying various grid parameters for each atomic partition until a satisfactory result is obtained is very costly. Therefore, ‘predefined grids’ which have been optimized to achieve certain accuracies on a chosen set of molecules remain the most practical way forward.

Before describing the way we have chosen to optimize the predefined grids for the DFT module, it is revealing to look briefly at a simple model analysis that illustrates where the errors in the integration originate. This highlights those aspects one needs to be wary of.

Consider an H_2^+ -like molecule, i.e. two atoms A and B, each of which is described exactly by a single exponential function and the molecular wavefunction is simply the sum of the two atomic wavefunctions. In the case under consideration here, we allow the two

exponents to differ as this highlights additional sources of error that would arise in real molecules. Furthermore, we limit our analysis to the line that connects the two atoms and which is assumed to be the z axis. Within this approximation the atomic wavefunctions are chosen to be

$$\varphi_A(z) = \frac{3}{2} \exp\left(-\frac{3}{4}|Z_A - z|\right), \quad (3)$$

$$\varphi_B(z) = \exp(-|Z_B - z|), \quad (4)$$

$$\varphi_{AB}(z) = \varphi_A(z) + \varphi_B(z), \quad (5)$$

where $Z_A = -1$ and $Z_B = 1$.

The integration of the ‘atomic density’ now boils down to a one-dimensional integration of an exponential function for which a grid can be defined which integrates this exactly. Each grid point accounts for a particular section of the z axis. To obtain reference results each section can be integrated with multiple points as well. Having established these grids, we consider integrating the molecular density along the line using Becke-type quadratures.

First we consider the Becke nuclear-weight function, which is defined as

$$\mu_{AB}(z) = \frac{|Z_A - z| - |Z_B - z|}{|Z_A - Z_B|}, \quad (6)$$

$$p(\mu) = \frac{3}{2}\mu - \frac{1}{2}\mu^3, \quad (7)$$

$$w_A(z) = \frac{1}{2}[1 - p(p(\mu_{AB}(z)))]. \quad (8)$$

Because the nuclear-weight function should partition the integrand into atom-like partitions, the Hirschfeld partitioning offers itself as an obvious contender to the Becke nuclear-weight function. Becke did mention exponential nuclear-weight functions in his original paper but did not elaborate on his reasons for rejecting them. The Hirschfeld nuclear-weight function [63] is defined as

$$w_A(z) = \frac{\rho_A(z)}{\rho_A(z) + \rho_B(z)} \quad (9)$$

which simply gives the fraction of the total density that a particular atom contributes assuming that the total density is a simple sum of atomic densities. Clearly if the molecular density is little different from the sum of atomic densities this nuclear-weight function would be expected to give very atom-like partitions

which would be ideally suited to polar coordinate quadratures.

In figure 1(a) we have plotted the atomic partitions of the molecular density for atom A obtained with the Becke nuclear-weight function and the Hirschfeld nuclear-weight function. The density of atom A and the molecular density are plotted for comparison. At first sight the Hirschfeld partition looks more like the exponentially decaying function expected for an atom-like function than the Becke partition, which goes to zero at the position of atom B. However, careful inspection reveals that there is a sudden change in the Hirschfeld partition at the position of atom B. In figure 1(b), for each grid point of atom A, we have plotted the error this grid point makes in integrating the atomic partition.

Once again this grid was chosen so as to integrate the density of atom A exactly. In this figure there are four distinctive regions. First in the region for z smaller than Z_A we find that the error is small because the exponential decay differs slightly from that of atom A due to the presence of atom B. Secondly, for z between Z_A and Z_B there are significant errors due to the nuclear-weight function not being able to recover a truly atom-like partition. Interestingly, the errors with the Becke nuclear-weight function are actually larger than those from the Hirschfeld nuclear-weight function in this region. However, in the third region close to atom B the Becke partition has decayed to zero and so has the error associated with it. The Hirschfeld partition, however, still has a remainder of the cusp of atom B which causes large errors in this region. In the fourth and final region where z is larger than Z_B the Becke partition and its associated error remain zero. The Hirschfeld partition, however, has acquired a significant atom B character leading to a noticeable systematic error for every grid point in this region. The overall result is that using the Hirschfeld nuclear-weight function leads to a 50% larger error for this example than using the Becke nuclear-weight function despite having smaller errors per grid point in the inter-atomic region. This example clearly underlines the importance of the nuclear-weight function for a given atom being zero at and behind all other atoms in the molecule. It shows also that when using a Becke-type partitioning the largest errors tend to arise in the inter-atomic region. Furthermore, using the cell-function as proposed by Becke, the error per grid point changes sign where the curvature of the function changes sign leading to a partial cancellation of errors.

In developing a set of predefined quadratures for the DFT module we first need to pick the basic

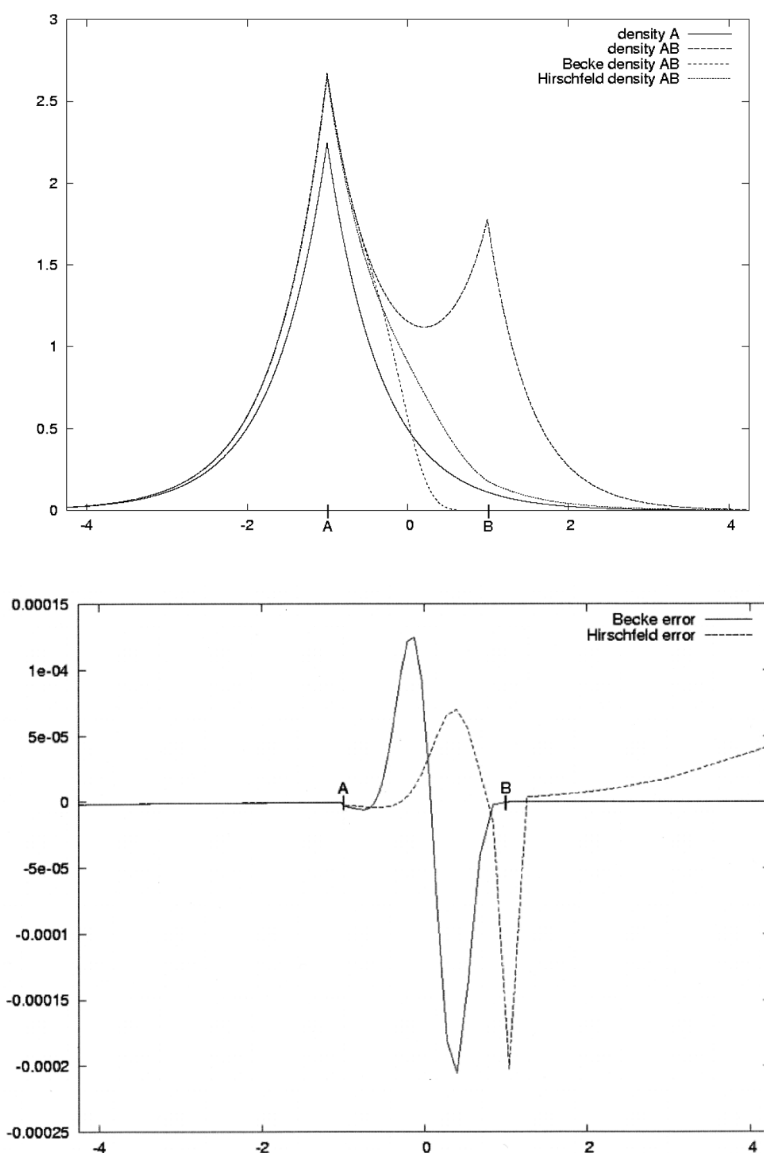


Figure 1. (a) Atomic partitions for atom A obtained from the Becke (Becke density AB) and Hirschfeld (Hirschfeld density AB) nuclear-weight functions compared to the total molecular density (density AB) and the density of atom A (density A). (b) The errors the grid that integrates atom A exactly contributes per point to the total error when attempting to integrate the Becke and Hirschfeld partitions.

components. Keeping these fixed, the grid sizes can be optimized to meet precision tolerance targets. Initially we set out to optimize four predefined grids named after the relative precision they are supposed to achieve. However, we found that the tolerance target of the ‘very high’ grid, being a relative error of less than $1.0\text{e-}10$ for the total energy and the electron count, is not generally achievable. In particular, molecules involving fluorine and row 5 or row 6 elements posed numerical problems we could not resolve. Therefore, we limited ourselves to optimizing the grids for targets listed in table 1.

Table 1. Grid names and their tolerances.

Grid	Target tolerated relative error
Low	$1.0\text{e-}4$
Medium	$1.0\text{e-}6$
High	$1.0\text{e-}8$

3.3.1. *The nuclear-weight functions implemented in GAMESS-UK.* In choosing the nuclear-weight function we limited ourselves to Becke-like options as they are expected to give most accurate results as concluded from

the above error analysis. In particular, we implemented the Becke, SSF [64] and MHL [65] nuclear weights. Testing these nuclear-weight functions demonstrated that although the SSF function had a low computational cost it also gave the least accurate results, due to the nuclear-weight function of an atom being forced to zero before another atom is reached. This has the advantage that there are easily identifiable regions where the nuclear-weight function equals one and regions where it equals zero. This can be exploited to save work but it leads to larger errors near the point where the partition reaches zero. The MHL function was found to give the most accurate result but it is also relatively expensive as it involves an 11-term polynomial. In a bid to combine the lower cost of the SSF function with the high accuracy of the MHL function two new functions were designed as follows.

In the SSF scheme the nuclear-weight function is defined as

$$\begin{aligned} w_A(z; a) &= -1, & \mu_{AB}(z) &\leq -a, \\ &= p_{\text{SSF}}(\mu_{AB}(z); a), & -a < \mu_{AB}(z) < a, \\ &= 1, & \mu_{AB}(z) &\geq a, \end{aligned} \quad (10)$$

$$p_{\text{SSF}}(\mu; a) = \left(35\left(\frac{\mu}{a}\right) - 35\left(\frac{\mu}{a}\right)^3 + 21\left(\frac{\mu}{a}\right)^5 - 5\left(\frac{\mu}{a}\right)^7 \right) / 16 \quad (11)$$

Obviously different nuclear-weight functions can be generated by choosing a different expression to replace p_{SSF} with. In particular, MHL-type expressions can be chosen which are defined by

$$\begin{aligned} \frac{dp_{\text{MHL}}(\mu)}{d\mu} &= A_{m_\mu}(1 - \mu^2)^{m_\mu}, \\ p_{\text{MHL}}(-1) &= 1, \\ p_{\text{MHL}}(1) &= 0, \end{aligned} \quad (12)$$

where MHL recommend $m_\mu = 10$. To use an MHL-type expression in an SSF scheme it is important not to distort the shape of the cell function too much. To this effect m_μ was lowered and the corresponding factor a optimized to satisfy the condition

$$\left. \frac{dp_{\text{MHLmSSF}}(\mu/a)}{d\mu} \right|_{\substack{\mu=0 \\ m_\mu=m}} = \left. \frac{dp_{\text{MHL}}(\mu)}{d\mu} \right|_{\substack{\mu=0 \\ m_\mu=10}}. \quad (13)$$

A number of functions of the above kind were tested and two were found to be particularly useful. The MHL8SSF cell-function where $m_\mu = 8$ and $a = 0.902256$ was found to give results of similar accuracy to the original MHL function but is noticeably cheaper to use. The MHL4SSF cell-function where

$m_\mu = 4$ and $a = 0.6651$ gives noticeably less accurate results but is much cheaper to use than either MHL8SSF or the original MHL function. However, MHL4SSF still gives more accurate results than the original SSF function. Thus we recommend using MHL8SSF over MHL for accurate calculations because of its lower cost, and we recommend using MHL4SSF for low accuracy calculations.

3.3.2. Selection of quadrature grids. We have implemented the Euler–MacLaurin [65] and the Mura–Knowles [66] radial grids. The Mura–Knowles transformation tends to have a higher density of grid points close to the nucleus, resulting in a much larger fraction of the grid points being positioned in locations where they make significant contributions. Thus for a given grid size the Mura–Knowles gives more accurate results than the Euler–MacLaurin grid [66], and we have chosen this transformation as the default. We also chose to use the scaling parameters α recommended by Mura *et al.* However, we extrapolated the table to cover the whole periodic system of the elements, using 5.0 for all elements except Groups I and II, for which a scale factor of 7.0 was used.

We have implemented both the Gauss–Legendre and Lebedev–Laikov [67, 68] angular grids. The Lebedev–Laikov grids have been specifically optimized for integration on a sphere. Furthermore, although the original Lebedev grids were only available with limited accuracy, the recently optimized Lebedev–Laikov grids are available in 15 digits precision and are capable of integrating 131st-order spherical harmonics exactly. This makes these grids the preferred ones.

Having chosen the basic form of the quadrature we set out to optimize the number of angular and radial grid points. We choose to allow for different numbers of grid points per row of the Periodic Table. First we created diatomic molecules for every pair of rows of the Periodic Table by combining group one and group seventeen elements. Where possible both all-electron and ECP calculations were used.

To optimize the radial grids the angular grid was kept fixed at 5810 points, and the radial grids were varied from 20 to 600 grid points. For each predefined grid and for each row the minimum radial grid size required to meet the accuracy target was recorded. The error in the energy was computed relative to the 600 radial grid points data.

A similar approach was followed for the angular grids. In this case, the radial grid size was fixed at 300 points as the result from the previous experiment showed that this was accurate enough. The angular grids were varied across all available Lebedev–Laikov

Table 2. The grid sizes for the low, medium and high accuracy grids. The ‘reference’ grid was used to obtain the reference energies against which the other grids were compared.

Grid	Row 1	Row 2	Row 3	Row 4	Row 5	Row 6	Row 7
Radial grid size							
Low	20	40	35	40	40	40	35
Medium	60	60	70	70	70	80	70
High	180	130	160	180	220	180	110
Reference	300	300	300	300	300	300	300
Angular grid size							
Low	194	266	302	302	302	302	266
Medium	266	590	590	590	770	590	434
High	770	1202	1730	1454	1730	1454	1454

grids, ranging from 194 to 1454 grid points. The data obtained in radial grid optimization with 300 radial grid points and 5810 angular grid points were used as references for the energy.

The resulting grid parameters were tested using molecules from the G2 test set [69] as well as a set of transition metal complexes including both all-electron and ECP calculations, and a few actinide systems. Initially, some two-dozen of these calculations failed to achieve the target accuracy. However, as these failures were relatively minor (the errors were typically about a factor of 2 too large) they could be addressed through experimentation. The final grid parameters obtained in this fashion passed all accuracy tests. Table 2 lists the corresponding numbers of grid points.

3.4. A DFT functional repository

Although the DFT method has become very popular, it is well known that its accuracy depends crucially on the density functional used. In practice, however, the exact functional is unknown and one has to choose an approximation that balances computational cost and accuracy. It comes as no surprise that the rise in interest in the theory has been accompanied by the development of a steady stream of new functionals.

One problem given the number of functionals available is their dissemination. Ideally, new functionals should appear quickly in various programs to allow scientists to reap the benefits they offer and to uncover any weaknesses. In practice, building new functionals into a code can be quite tedious. Often the interface between the host program and the functional is problematic because the functional expects quantities that the program does not provide, or the functional calculates quantities that the program does not expect. An example of the latter is that some functional routines compute not only the energy expression, but also the potential, whilst others compute partial derivatives.

As a consequence, re-implementation of the functional is often required.

In light of these problems, it seems useful to have a freely available collection of routines that implement the various functionals; these subroutines should ideally be certified by the original authors as correctly representing the functional they proposed. In any case it should be clear whether such certification has been obtained or not. This process also requires a set of reference data. This data should validate the results of porting a given functional routine to the host program. In those cases where a functional is re-implemented, perhaps because of efficiency reasons or to address interfacing issues, the reference data should also act to validate the correctness of the implementation.

With the above goals in mind, a density functional repository [70] has been designed. The functionals supported are listed in table 3. With respect to the reference data, we chose to provide two different data sets. The first consists of the total energies for 10 spherically symmetric atoms, chosen to cover a wide range of density and density gradient values. In addition, the spherical symmetry allows for numerical integration with high accuracy, thus minimizing uncertainties with respect to technicalities of the host program implementation. The disadvantage of these reference calculations is that the results depend on the Gaussian basis set used. This makes the data difficult to compare with results from programs in the physics arena which employ plane waves as basis functions. Therefore, a second set of data is provided which lists the results of a functional for a given set of data points. The values for the input data have been obtained from the atomic calculations, but were truncated to two significant digits to facilitate copying. More details about the design considerations are stored with the repository.

Although the current collection of functionals is still limited, we believe that the current implementation provides a framework for the dissemination of new

Table 3. Density functionals supported by the repository.

Exchange functionals	Correlation functionals	Exchange+correlation functionals
Slater	Perdew–Zunger ‘81	B3LYP
Becke ‘88	Perdew ‘86	Filatov–Thiel ‘97
Becke-3	Vosko–Wilk–Nusair 5	Hamprecht–Cohen–Tozer–Handy (HCTH) including HCTC-120, -147 and -407
PBE	Vosko–Wilk–Nusair 5 RPA	Becke ‘97
Filatov–Thiel ‘97	Lee–Yang–Parr	B97-1 and B97-2
	PBE	PBE
	Filatov–Thiel ‘97	EDF1

functionals, and hope to collect newly implemented functionals as they become available.

4. Some of the application modules of GAMESS-UK

In this section we highlight a number of the implementation aspects of the more popular, and the more novel, features available within the code. The Geometry optimization modules are overviewed in section 4.1. We then focus on the treatment of relativistic effects through the ZORA Hamiltonian, (section 4.2), and on the capability for performing valence bond calculations (section 4.3). The various treatments available for excited states, including MRDCI, MCLR, and the associated treatment of the geometric derivatives of excited state energies, are summarized in sections 4.4 to 4.6, respectively. The variety of available hybrid QM/MM schemes central to the treatment of large systems is considered in section 4.7. Finally, we consider the variety of model building tools and visualization techniques available to GAMESS-UK within the CCP1-developed GUI (section 0).

4.1. Geometry specification, optimization and transition-state location

The atomic coordinates of a molecule can be provided to the program as Cartesian coordinates or as a Z-matrix. Optimization of the Cartesian coordinates is enabled through a Broyden–Fletcher–Goldfarb–Shanno (BFGS) rank-two update algorithm [71] with the option of maintaining some of the atoms fixed in space, or of performing some partial optimizations. The most general geometry and optimization combination is provided by the Z-matrix geometry specification which, as well as allowing the geometry to be specified in the natural bond length, bond angle and torsion angle format, also allows the user to specify the position of atoms in Cartesian coordinates. This very flexible geometry specification allows for dummy atoms, point charges and ghost atoms, the latter being useful for

the calculation of basis set superposition errors. Each numerical variable can be assigned a name which may then be included in the geometry optimization, or which may be held fixed throughout the calculation. There are also options for the calculation or estimation of the variable’s second derivative, as well as options for the specification of values of the variable corresponding to minima. This latter option is useful when performing transition state calculations.

Three geometry optimization algorithms are available to the user employing Z-matrix input. The default algorithm begins with a calculation of the energy and its gradient, followed by a line search using energy calculations only. At the minimum energy along the line of search an additional calculation of the gradients is performed and the current estimate of the inverse Hessian is updated using a rank-two BFGS update. A new search direction is calculated from the current gradient, position and Hessian and the process is iterated until convergence. Convergence is dictated by a series of four thresholds, all of which must be met for the optimization to complete. The four criteria are: the largest predicted step in any coordinate, the average predicted step, the largest gradient term and the average gradient values. Other optimization methods include an optimization in Cartesian space and a method based on the approach of Simons *et al.* [38].

Transition state searches can be performed by several methods. The default algorithm is based on the work of Cerjan and Miller [37]. The algorithm expects that some estimation of the Hessian has been performed, either numerically using the options on the geometric variable specification, or analytically by calculating the full second derivative matrix. Using the available Hessian information, an estimate is made of the energy profile on a sphere (in the space of the variables) defined by the trust region radius, around the current point. If the transition state is predicted to be outside the trust region on the basis of a straightforward application of a Newton step, then the appropriate minimum on the trust region is taken for the next point, where an energy and a gradient calculation are performed. Various

algorithms are incorporated to ensure that the trust region radius expands and contracts in line with the quality of the predictions of the algorithm. The inverse Hessian is updated with a rank-one Murtagh-Sargent update [71], although other options are available, including a symmetric Powell update.

For those cases where it is difficult to provide sufficient Hessian information an alternative method based on the Quadratic Synchronous Transit (QST) algorithm of Bell and Crighton [36] is available. A quadratic search path joining two minima through the current point is defined. This line is first searched for a maximum. The Hessian is updated based on the change in the gradients and a set of directions conjugate to the maximum search direction is defined, in which the energy is minimized. If the direction of the maximum search direction changes during the minimization step, the algorithm recalculates a maximum along the QST. Although fairly robust, this algorithm can find itself in difficulties if the directions defined by the QST and the eigenvector corresponding to the negative eigenvalue of the Hessian do not coincide. To address this problem the method implemented has been modified to allow a more general functional form for the quadratic curve. A quartic polynomial is used to interpolate between the two minima, the current point and the additional flexibility of the quartic is used to encapsulate the Hessian information which is available, so that at the current point the curve aligns itself locally with the direction defined by the eigenvector of the lowest eigenvalue of the Hessian.

4.2. The treatment of relativistic effects and ZORA

For an adequate description of molecules containing heavier elements, the relativistic behaviour of the inner electrons cannot be ignored. The standard Schrödinger equation is inadequate for these compounds, which include the very important class of catalytic agents. The relativistic electron density near the nucleus may be simulated using a relativistic effective core potential [9]. Other approaches make approximations to the Dirac-Fock operator, as in the Douglas-Kroll-Hess method [72,73].

The CPD or ZORA method [74–76] is a two-component approximation to the Dirac-Fock equation, which is derived by using an expansion in $E/2c^2 - V$ and only keeping the zeroth term. Currently, the scalar ZORA [77] one-component approximation is operational within GAMESS-UK. Here the kinetic energy operator is replaced by a potential-dependent term:

$$\hat{T} = \mathbf{p} \cdot \frac{c^2}{2c^2 - V_c} \cdot \mathbf{p} \quad (14)$$

For practical implementation, resolutions of the identity are inserted and ensuring that the correct non-relativistic limit is retained, the elements of the ZORA kinetic energy operator are written as

$$T_{\mu\nu} = \frac{1}{2} \langle \phi_\mu | p^2 | \phi_\nu \rangle - \frac{1}{2} \sum_\lambda \langle \phi_\mu | \mathbf{p} | \phi_\lambda \rangle S_\phi^{-1} \langle \phi_\lambda | \mathbf{p} | \phi_\nu \rangle \\ + \frac{1}{2} \sum_{\lambda,\kappa} \langle \phi_\mu | \mathbf{p} | \phi_\lambda \rangle S_\phi^{-1} \langle \phi_\lambda | 1 - (V_c/2c^2) | \phi_\kappa \rangle^{-1} S_\phi^{-1} \langle \phi_\kappa | \mathbf{p} | \phi_\nu \rangle. \quad (15)$$

The internal basis $\{\phi_\lambda\}$ is generated from the molecular basis set by approximate application of the \hat{p} operator [77]. The coulomb operator $V_C = V_{\text{nuc}} + J_{\psi\psi}$ is generated using a standard direct Fock-matrix operator. Subsequently, the $1 - (V_C/2c^2)$ matrix is inverted using standard matrix inversion and transformed to the molecular basis. All basic integrals required are available in GAMESS-UK. The result is an effective one-electron operator which, for example, allows for the description of core ionization. All post-Hartree-Fock techniques may be used without change [78].

The ZORA method features a coulomb operator in the denominator, which causes the loss of gauge invariance. The scaled ZORA approach [76] reduces the error by orders of magnitude. The problem can be fixed exactly by using a strictly atomic scheme, where the ZORA corrections are calculated separately for each atom [79]. This approach has the advantage that the ZORA corrections only have to be determined once and that all gradients calculated are exact, whereas the approximation is generally negligible. Work is also underway to incorporate the two-component ZORA, which includes spin-orbit coupling [80], at the SCF level. This involves some complex arithmetic.

4.3. Valence bond

Valence Bond theory has always struck a sympathetic chord in chemists' minds, because it can be linked so closely to the familiar Lewis structure. A bond is immediately translated in the wavefunction by two non-orthogonal orbitals on neighbouring atoms that are singlet coupled. An ionic structure may contain an atomic orbital that is occupied twice, or alternatively two orbitals on the same atom. So it is simply possible to translate the assumed concept of bonding in a molecule into a (small) set of structures. Alternatively, if the nature of the bonding is in question, the relative importance of the different structures may provide insight. The benzene molecule is naturally described as two resonating structures, each describing local bonding. Vital for an unbiased wavefunction is the ability to optimize the wavefunction, both its orbitals and its structure coefficients [81].

The TURTLE program [82] has been integrated into GAMESS-UK taking advantage of, for example, the gradient capabilities and the possibility of generating atomic wavefunctions. Thanks to this integration all options available to, for example CASSCF, such as effective core potentials or scalar (atomic) relativistic effects [79] are now also available in VB (SCF). A review of the algorithms used in evaluating matrix elements and in optimizing the orbitals has recently been published [83].

Due to the non-orthogonality of the orbitals, and therefore of the structures, a VB (SCF) calculation is intrinsically more time consuming than an orthogonal calculation, with a Hamiltonian matrix element between two structures taking for N electrons in the order of N^4 operations [84]. Nevertheless, basis sets with over 160 AOs have been used for small molecules and the interpretative power of Valence Bond has already been extended to metal–organic chemistry [85]. The calculation of the matrix elements, by far the most time-consuming part, has been parallelized (see section 5.5). In summary the key features of the code are:

- using a VBSCF procedure, it is possible to optimize wavefunctions of both Heitler–London and Coulson–Fischer type: indeed, it is even possible to use different orbitals for different structures (e.g., covalent or ionic);
- automatic orbital optimization using Super CI and more approximate methods [86] offers the possibility to keep orbitals strictly local to atoms, ‘optimal hybrids’;
- all gradient-based methods that GAMESS-UK has to offer can be used with these wavefunctions; associated input routines are provided that offer extended possibilities to specify the starting orbitals.

In collaboration with Inorganic Chemistry in Nijmegen (P.H.M. Budzelaar) we are attempting to apply the Valence Bond method in an area where it is always assumed to be most useful, the elucidation of chemical bonding. In contrast to transition metals, main-group metals show a bewildering variety of bonding arrangements to cyclo-pentadienyl (Cp) groups, including electron-precise, electron-deficient and electron-excess structures. Valence Bond calculations (VB) have been performed on cyclo-pentadienyl silicon hydride (CpSiH), cyclo-pentadienyl silicon trihydride (CpSiH₃) and cyclo-pentadienyl aluminium dihydride (CpAlH₂) to gain more insight into the number of bonds (hapticity) of the metal atom with the ring [85]. VB makes use of non-orthogonal orbitals, which means orbitals can overlap; the overlap between two orbitals can be used as a measure of the bond strength.

VB wavefunctions are constructed from structures, to which one may attribute chemical meaning. One such structure for CpSiH is Cp–/SiH+. Another structure is Cp–SiH in which the bond between the Cp-ring and the SiH-group is covalent. When the total wavefunction is variationally optimized weights can be calculated from the coefficients of the structures. These weights show the relative importance of all structures to the total wavefunction and thus the importance of each bonding type.

4.4. MRDCI

The MRDCI module as implemented in GAMESS-UK is now the (semi-)direct version of Buenker, Engels, and Peyerimhoff [24, 25], which is based on the original table-driven implementation [12–14, 25]. It is especially suited for the calculation of excited states, as a multi-root Davidson diagonalization routine is implemented which can calculate multiple solutions of the same symmetry simultaneously. CI expansions of a few million Configuration State Functions (CSF) can be treated using this code.

The procedure starts like any other CI program with an initial chosen set of reference configurations $\{\psi_j^{(1)}\}$ which is used for the generation of the second set $\{\psi_k^{(2)}\}$. In the MRDCI formulation, this second set is generated by performing all single and double excitations from the parent set $\{\psi_j^{(1)}\}$ subject to a threshold criterion:

$$\left| \frac{|\psi_k^{(2)}|H|\psi_j^{(1)}|^2}{\langle \psi_k^{(2)}|H|\psi_k^{(2)} \rangle - \langle \psi_j^{(1)}|H|\psi_j^{(1)} \rangle} \right| > \delta \text{ for at least one } j \text{ [12].} \quad (16)$$

Here, δ is a preset energy criterion, typically in the range 10^{-3} – $10^{-6} E_h$ to keep the calculation feasible. The CSFs corresponding to all spin paths of a particular configuration are selected if at least one of them satisfies the threshold criterion. This is required to set up calculations that are unitarily invariant for orbital rotations that leave the energy of the foregoing (MC)SCF unchanged.

The program avoids the calculation of the complete \mathbf{H} matrix in a Direct-CI like fashion [15]. It arranges the configurations in so-called supercategories according to their open-shell structure, thus treating the spin coupling schemes for a whole set of configurations at the same time.

The simultaneous diagonalization of the equisymmetric states of interest ensures the orthogonality requirements, enabling easy calculation of oscillator strengths. Furthermore, the code will be extended with MRACPF [87] and MRAQCC [88] variants.

GAMESS-UK also provides an interface to the DIESEL-CI program [89], which is an efficient (and parallel) direct implementation of the selected CI procedure.

4.5. Multi-configurational linear response

Response methods are a class of approaches in which properties are calculated by considering how a molecule responds to an external, most often time-dependent, perturbation. In general, these methods can be used to compute a wealth of properties depending on the time-dependent perturbation, e.g. an oscillating electric or magnetic field, and the observable for which the response is calculated. Examples are magnetic susceptibility, NMR chemical shifts, infrared spectra, optical rotation strengths, and UV/Vis spectra. The latter is the property of interest here. In this case the response of a molecule to a time-dependent electric field is considered and the transition energies and oscillator strengths calculated. Thus response methods provide an alternative to CI methods. Within the context of molecules described by a multi-configurational wavefunction, the Multi-Configurational Linear Response (MCLR) method was first proposed by Yeager and Jørgensen [90] and later reformulated by Fuchs [26] to allow for expansions to arbitrary high orders of perturbation theory. The latter formulation forms the basis for the implementation discussed here.

The general approach behind the MCLR method is to derive an equation of motion for the response function from Ehrenfest's theorem, employing an exponential parameterization, and to linearize this to obtain an eigenvalue problem. The time-dependent wavefunction itself is actually only used for the purpose of derivation. The equation of motion for the response function is Fourier transformed to the frequency domain to obtain a matrix equation suitable for numerical computation. The time-independent reference wavefunction is chosen to be an optimized MCSCF wavefunction which therefore satisfies the generalized Brillouin theorem. Based on the parameterization of the MCSCF wavefunction, it is natural to parameterize the time-dependent wavefunction as

$$|0(t)\rangle = e^{K(t)}e^{A(t)}|0\rangle. \quad (17)$$

As a result of the time dependence this wavefunction has to be complex valued and K and A are defined as

$$K(t) = \sum_{r>s} [\kappa_{rs}(t)E_s^r + \kappa_{rs}^*(t)E_r^s], \quad (18)$$

$$A(t) = \sum_{m>0} [\lambda_m(t)|m\rangle\langle 0| + \lambda_m^*(t)|0\rangle\langle m|], \quad (19)$$

where r and s refer to orbitals, κ describes the time dependence of the orbitals, m refers to states in the orthogonal complement of the MCSCF wavefunction, and λ describes the time dependence of the N electron states. After the derivation sketched above and described in detail by Fuchs [26] the following eigenvalue problem is obtained:

$$(\omega_i \mathbf{S} - \mathbf{E})\mathbf{x}_i = 0, \quad (20)$$

where

$$\mathbf{S} = \begin{pmatrix} \Sigma & \Delta \\ -\Delta^* & -\Sigma^* \end{pmatrix}, \quad (21)$$

$$\mathbf{E} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^* & \mathbf{A}^* \end{pmatrix}, \quad (22)$$

The submatrices are given by

$$\Sigma = \begin{pmatrix} \langle 0|[E_q^p, E_r^s]|0\rangle & \langle 0|[E_q^p, A_n]|0\rangle \\ \langle 0|[A_m^+, E_r^s]|0\rangle & \langle 0|[A_m^+, A_n]|0\rangle \end{pmatrix}, \quad (23)$$

$$\Delta = \begin{pmatrix} \langle 0|[E_q^p, E_r^s]|0\rangle & \langle 0|[E_q^p, A_n^+]|0\rangle \\ \langle 0|[A_m^+, E_r^s]|0\rangle & \langle 0|[A_m^+, A_n^+]|0\rangle \end{pmatrix}, \quad (24)$$

$$\mathbf{A} = \begin{pmatrix} \langle 0|[[E_q^p, H_0], E_r^s]|0\rangle & \langle 0|[[E_q^p, H_0], A_n]|0\rangle \\ \langle 0|[[A_m^+, H_0], E_r^s]|0\rangle & \langle 0|[[A_m^+, H_0], A_n]|0\rangle \end{pmatrix}, \quad (25)$$

$$\mathbf{B} = \begin{pmatrix} \langle 0|[[E_q^p, H_0], E_r^s]|0\rangle & \langle 0|[[E_q^p, H_0], A_n^+]|0\rangle \\ \langle 0|[[A_m^+, H_0], E_r^s]|0\rangle & \langle 0|[[A_m^+, H_0], A_n^+]|0\rangle \end{pmatrix}, \quad (26)$$

where H_0 is the MCSCF Hamiltonian. The straightforward way to solve these eigenvalue equations is to assume that there is a set of vectors x that are eigenvectors of the matrices \mathbf{S} and \mathbf{E} simultaneously. In that case direct methods based on the Davidson diagonalization method can be applied. However, the existence of such a set of vectors can only be guaranteed if the reference wavefunction corresponds to a ground-state MCSCF wavefunction.

Finally, in order to understand the electronic structure of the system under consideration it is crucial to know the dominant configurations in the wavefunction of the examined m th excited state. Although this is a simple task in CI calculations, in the MCLR formalism the question to answer is in which sense do the eigenvectors of the response matrices correspond to the wavefunctions of the excited states. It can be shown [26] that

$$|m_{\text{MCLR}}\rangle = \sum_j (\mathbf{x}_m)_j (K_j, A_j, K_j^+, A_j^+)|0\rangle \quad (27)$$

plays a role analogous to the m th excited state and this formula can be used to analyse the contributions of the excitations from the reference wavefunction.

4.6. Random phase approximation and geometric derivatives of the excited state

The Random Phase Approximation (RPA) is a relatively simple response method. It can be obtained from the same formalism as the MCLR method but applied to a single determinant wavefunction. In particular, for the closed-shell case the RPA method can be obtained by choosing the set of active orbitals to be empty. This leads to an ordinary eigenvalue problem for a non-Hermitian matrix:

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B} & \mathbf{A} \end{pmatrix} \begin{pmatrix} Z \\ Y \end{pmatrix} = E \begin{pmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & -\mathbf{1} \end{pmatrix} \begin{pmatrix} Z \\ Y \end{pmatrix}, \quad (28)$$

$$1 = Z^T Z - Y^T Y, \quad (29)$$

$$A_{ai,bj} = (\varepsilon_a - \varepsilon_i) \delta_{ab} \delta_{ij} + 2(ai|bj) - (ab|ij), \quad (30)$$

$$B_{ai,bj} = -2(ai|bj) - (ai|bj), \quad (31)$$

where the eigenvalues E represent the excitation energies of the molecule. Here, $(\varepsilon_a - \varepsilon_i)$ is an orbital energy difference from Hartree–Fock theory, and $(ai|bj)$ denotes the Coulomb matrix element coupling occupied orbitals (ij) and orbitals in the virtual manifold (ab) . A similar formulation can be derived from the Kohn–Sham equations of density functional theory, leading to a time-dependent DFT approach.

Neglecting the matrix \mathbf{B} , the equations decouple to two identical expressions which are equivalent to the Single CI or Tamm–Dancoff (TDA) equations. Therefore, it may be said that RPA is somewhat corrected for the effect of double excitation which appears in matrix \mathbf{B} . However, not all double excitations are present nor are these excitations coupled to the single excitations. Therefore, if the double excitations really cannot be neglected, the RPA can only be expected to give qualitative results.

An important implementation aspect of the RPA equations is that the expressions are simple enough to be readily coded in a direct fashion. This allows the integral transformation to be avoided and the two-electron integrals are computed in the AO-basis whenever needed [26]. This aids in removing storage bottlenecks and in writing a replicated data parallel implementation.

The chemical behaviour of excited states within the RPA formalism can be studied by considering the Newtonian dynamics of these excited state potential energy surfaces, for which an accurate energy gradient,

or force, is required. These gradients were first published by Ortiz [91]. More recently, a derivation based on a slightly different formulation of the RPA equations was suggested by van Caillie *et al.* [92]. Following the equations by Ortiz, we found an error probably due to a misinterpretation of the Coupled Perturbed Hartree–Fock (CPHF) equations. Traditionally, the CPHF equations are written as

$$U_{ij}^\alpha = \frac{Q^\alpha + \sum_{dl} U_{dl}^\alpha \{4(ij|dl) - (id|jl) - (il|dj)\}}{\varepsilon_j - \varepsilon_i}. \quad (32)$$

Clearly, this equation is singular if $i = j$, but this does not mean that U_{ii}^α is undefined. Differentiating the normalization condition of the orbitals, one obtains

$$U_{pq}^\alpha + U_{ap}^\alpha + S_{pq}^\alpha = 0, \quad (33)$$

from which it follows that

$$U_{ii}^\alpha = -\frac{1}{2} S_{ii}^\alpha. \quad (34)$$

This result was incorporated into the original derivation and the implementation within GAMESS-UK. The implementation was tested in two ways. First the gradients were compared against numerical gradients of the energy. Second, we used the fact that the total energy should be conserved while the nuclei move. Any systematic error in the gradients would result in inconsistent changes of the kinetic energy of the nuclei and the potential energy. Following the molecule through about 1000 geometry changes, any error would be strongly amplified making this test extremely sensitive. The implementation was found to yield gradients indistinguishable from the numerical results. Also the conservation of the energy was obeyed to within 5μ Hartrees after 1000 steps. Given the accuracy of the various parts of a typical quantum chemistry code, one cannot expect to improve beyond this.

Increasingly, these excited state dynamical processes, which occur on femtosecond timescales, are becoming the subject of experimental study, as exemplified by the Nobel prize-winning work of Ahmed Zewail [93]. However, interpreting the measured signals in terms of underlying chemical processes is not easy without simulations. In particular, in pump–probe experiments the calculation of trajectories of molecules in the excited state is important. The above approach can be used in this context [94] to produce valuable data, although the costs are quite high because of the large number of time steps per trajectory and the large number of trajectories needed to converge the statistics of the thermodynamics.

4.7. QM/MM and the treatment of large systems

While developments in computer performance and QM algorithms are bringing increasingly complex systems within the scope of quantum mechanical calculations, many important chemical systems remain too large for pure quantum simulation. This is especially true if energies for many configurations are required, as in molecular dynamics studies. Over the last ten years or so we have sought to extend the capabilities of the GAMESS-UK code by including environmental effects via interfaces to other programme packages. Two approaches have been pursued in parallel: on one hand, the ChemShell package [95] provides for a generalized coupling of QM and MM packages, while the GAMESS-UK/CHARMM interface specifically addresses the requirement of providing a QM/MM capability for users of the CHARMM package.

4.7.1. *ChemShell*. ChemShell [95, 96] is a scripting language, based on Tcl, which provides a number of QM/MM schemes together with a number of utility functions for dealing with the combined QM/MM potential energy surface (for example, MD and geometry optimization drivers). ChemShell is not specific to GAMESS-UK, interfaces exist to MOLPRO, CADPAC, Turbomole and Gaussian, however a number of modifications to GAMESS-UK to support interfacing to the environment make it the most useful code in this context. This interface is suitable for a variety of application areas [95], but to date most examples that have employed GAMESS-UK have focussed on solid-state embedding problems such as zeolite and metal oxide catalysis [97, 98].

4.7.2. *The CHARMM/GAMESS-UK Interface*. A second approach, particularly appropriate for biomolecular systems, is exemplified by the direct coupling of GAMESS-UK with the CHARMM [99] macromolecular modelling package. In this case all the tools for system setup and exploration of conformational space are those of CHARMM, with the QM contribution simply a part of the energy expression. CHARMM is one of the most widely used packages for the study of macromolecules such as proteins, nucleic acids and lipids. It supports energy minimization and molecular dynamics approaches using a classical parameterized force-field. In order to permit studies of reacting species it is useful to be able to incorporate the quantum mechanical energy of a part of the system into the force-field, and over recent years a number of interfaces to quantum mechanical programs have been developed. Initially, these were based on semi-empirical wavefunctions. More recently, computational and hardware

developments, such as those described in this paper, have led to increased interest in *ab-initio* QM/MM schemes and interfaces to the GAMESS(US) [100] and CADPAC packages [101] have been implemented. The coupling between CHARMM and GAMESS-UK follows a similar approach to these.

In the CHARMM QM/MM model the standard CHARMM force-field is used for the classical partition and the QM/MM van der Waals interactions. The QM/MM electrostatics are handled by including, in the Hamiltonian, point charges at the MM positions. The energy and forces from the QM calculation, including electrostatic forces acting on the classical centres, are added to those computed by CHARMM. The QM/MM approach involves introducing additional hydrogen (link) atoms to the edges of the QM cluster to terminate the quantum mechanical calculation. The forces on the link atoms can be handled by CHARMM using the same methods developed for treating explicit models of lone pairs.

The DFT module within GAMESS-UK employs an auxiliary basis fit of the charge density to provide an approximation to the Coulomb energy (see above). We have used these elements of GAMESS-UK to implement an alternative model in which the charge density of the classical system is included in the QM Hamiltonian not as a set of point charges but as a continuous charge distribution represented as a sum of Gaussian terms [102]. This allows greater overlap between the QM and MM charge distributions without the introduction of major artefacts and thereby permits the exploration of a number of QM/MM schemes. The CHARMM/GAMESS-UK interface supports a QM/MM implementation of the replica path algorithm, which enables simultaneous optimization of a complete reaction path using an algorithm particularly well suited to parallel computing [103].

4.8. Visualization and the CCP1 GUI

Visualization and input preparation for GAMESS-UK is supported through the CCP1 GUI project, which has arisen as a result of demand within the UK academic community (principally through CCP1) for a free, extensible Graphical User Interface (GUI) for community codes, particularly for teaching purposes.

The GUI has been built around the Python open-source programming language and the VTK visualization toolkit, both of which have been ported to all the major operating system platforms. The GUI is therefore capable of running on all of these systems. Distributions of the GUI have already been successfully tested on Windows, and Suse and Redhat Linux distributions, and packages for these distributions are freely available

from our web site [104]. The design of the GUI makes the most of Python's high degree of object-orientation, including advanced features such as multiple inheritance that are not available in other object-oriented languages such as C++ and Java. The strong reliance on object orientation means that it is quick and easy to create interfaces to new computational chemistry codes as the need arises. The GUI already has a highly featured interface for the GAMESS-UK program, and there are working interfaces for the ChemShell QM/MM package and MOPAC.

4.8.1. CCPI GUI functionality. The GUI supports GAMESS-UK SCF, DFT or post-Hartree-Fock calculations, and geometry optimizations, including transition state searches. As well as options to configure everything from the convergence criteria of SCF calculations and geometry optimizations, to functionals and grid settings for DFT calculations, the interface includes a tool to configure the basis sets on individual atoms. The interface also provides access to the various analytical options offered by GAMESS-UK and the results of these calculations are automatically imported into the GUI for display with the visualization tools.

Supported file formats. From its inception, the GUI was intended to work with a number of different codes, so a variety of file formats are supported both for reading in molecular structures and outputting data. As well as conventional formats such as Z-matrix and PDB, there are programme-specific formats for CHARMM, ChemShell, XMol and Gaussian, as well as support for XML.

Editing capabilities. The GUI has a suite of easy-to-use editing tools that allows complex molecules to be created using simple point-and-click operations. There is also a fully functional Z-matrix editor that can be used to edit the atomic coordinates of a molecule in internal or Cartesian form. Symbolic variables (and constants) can be defined, and the editor can automatically generate a Z-matrix (including the requisite reordering of the atoms) which can subsequently be customized. Used in conjunction with the graphical tools this creates a powerful and flexible environment for building molecular structures.

Visualization tools. The GUI has powerful and highly customizable visualization capabilities. Scalar data can be represented through volume rendering, coloured 3D-isosurfaces, 2D-slices or as a raw grid of points. Vector data can be represented as hedgehog plots, glyphs or streamlines, and the colours and opacity of any of the vector or scalar representations can be easily adapted to

a given schema. Multiple representations of the various molecular properties for the same or different molecules can be created and overlaid to extract the maximum amount of information from the results of a calculation. The GUI can also be used to view the dynamic properties of a calculation, by animating molecular vibrations or creating a movie from the different steps in a geometry optimization.

e-Science developments. Work is already underway [105] to enable the GUI to exploit the latest developments in the field of e-Science. Initially these will allow the GUI to download molecular structures from databases around the world for viewing, or to serve as the inputs for calculations. As the e-Science Data Model matures, however, it will become possible to not only download and view the results of calculations carried out with different codes, but to use the GUI to import the inputs to serve as a basis for new calculations.

5. Parallel implementation and performance of GAMESS-UK

5.1. The replicated data MPI implementation

Historically the development of the parallel version of the code resulted in both SCF and DFT modules being parallelized in replicated data fashion, with each node maintaining a copy of all data structures present in the serial version. While this structure limits the treatment of molecular systems beyond a certain size, experience suggests that it is possible on the current generation of machines to handle systems of up to 5000 basis functions. The main source of parallelism in the SCF module is the computation of the one- and two-electron integrals and their summation into the Fock matrix, with the more costly two-electron quantities allocated dynamically using a shared global counter. The result of parallelism implemented at this level is a code scalable to a modest number of processors (around 32), at which point the cost of other components of the SCF procedure start to become significant. This mode of parallelization is available in the current distribution and is referred to historically as the MPI implementation.

Improving the scalability of the code beyond this has been addressed in part by adopting a number of the tools developed by the High Performance Computational Chemistry Group (HPCCG) from the Environmental Molecular Sciences Laboratory at PNNL. The HPCCG has developed molecular modelling software to take full advantage of the parallel computing power of MPPs. These efforts have culminated in the NWChem package that includes a broad

range of electronic structure and molecular dynamics functionality [106–108]. We briefly outline in the next section the tools that have been adopted in collaboration with the HPCCG.

5.2. Global array tools and PeIGS

The first of these is the Global Array (GA) toolkit [109–111], which provides an efficient and portable ‘shared-memory’ programming interface for distributed-memory computers. The toolkit enables each process in a MIMD parallel program to asynchronously access logical blocks of physically distributed matrices, without need for explicit co-operation by other processes. Unlike other shared-memory environments, the GA model exposes the programmer to the non-uniform memory access (NUMA) timing characteristics of the parallel computers and acknowledges that access to remote data is slower than to local data. From the user perspective, a global array can be used as if it were stored in shared memory, except that explicit library calls are required to access it. The information on the actual data distribution can be obtained and exploited whenever data locality is important. Each process is assumed to have fast access to some ‘local’ portion of each distributed matrix, and slower access to the remaining ‘remote’ portion. Remote data can be accessed through operations like ‘get’, ‘put’ or ‘accumulate’ (floating point summation) that involve copying the globally accessible data to/from process-private buffer space. A number of BLAS-like data-parallel operations have been developed on top of these primitives.

The second tool is the scalable, fully parallel eigensolver, PeIGS, whose numerical properties satisfy the needs of the chemistry applications [112]. PeIGS solves dense real symmetric standard ($Ax = \lambda x$) and generalized ($Ax = \lambda Bx$) eigenproblems. The numerical method used is multisection for eigenvalues and repeated inverse iteration and orthogonalization for eigenvectors [112]. Accuracy and orthogonality are similar to LAPACK’s DSPGV and DSPEV [113]. Unlike other parallel inverse iteration eigensolvers, PeIGS guarantees orthogonality of eigenvectors even for arbitrarily large clusters that span processors. Internally, PeIGS uses a conventional message passing programming model and column-distributed matrices. However, it is more commonly accessed through an interface provided by the GA toolkit, with the necessary data reorganization handled by the interface.

Once the capability for GA is added to GAMESS-UK, and the PeIGS-based diagonalization module introduced, parallelization of the linear algebra becomes straightforward by forming a distributed copy of the relevant matrices and calling the library routine.

Depending on the subsequent operation, it may then be necessary to re-replicate the array. As an example, the SCF convergence acceleration algorithm (DIIS—direct inversion in the iterative subspace) uses GA storage for all matrices, and parallel matrix multiply and dot-product functions. This not only reduces the time to perform the step, but the use of distributed memory storage (instead of disk) reduces the need for I/O. We have also used the GA tools to map disk files into distributed memory, an approach which proves more efficient than keeping all files on node 0 and distributing to all nodes using broadcast operations.

5.3. Moving to distributed data

With the increasing availability of large parallel architectures it is becoming more and more important to exploit such systems effectively. Methodologies that work well on small to medium size machines may not scale well to the top end, and so it is becoming necessary to investigate alternative strategies that make better use of the resources available.

The replication of all the data implies that the largest problem that can be solved is governed by the amount of memory available to one processor, not that available on the whole machine. So, for example, if a machine consists of 1024 processors each with 1 GByte per processor, the limit is not the terabyte of memory available on the whole system, but the GByte available locally. More generally, the communications overhead can become very large because once a parallel operation has been performed it will be necessary to re-replicate all the data across the whole machine so that each processor has a current version. This requires a global communication, a potentially costly operation for which the time taken grows with the number of processors. While this may not be so important for such very expensive operations as the Fock build, for fast, highly efficient operations which are performed repeatedly, such as matrix multiplies, the communication overhead can become large relative to the time taken in the computation, thus inhibiting scaling. Finally, replicated data simply does not fit well with many algorithms. For computational chemistry the most important of these is diagonalization, which becomes an increasingly important stage of the calculation as the system size increases. The net result of these drawbacks is that a replicated data strategy is not an efficient methodology for exploiting large parallel computers due to both system size and parallel scalability limitations.

An alternative method is distributed data. Here each processor holds only part of each of the (major) data structures. This overcomes many of the drawbacks of replicated data; the limit on system size is now that given

by all the memory on the computer, global communications are largely avoided as re-replication is not required, and many of the algorithms that cannot be effectively addressed by a replicated data strategy fit more naturally within a distributed data one. Due to these attractive features this method is being increasingly used both in quantum chemistry codes, and more generally in other areas. One example in quantum chemistry is NWChem [106–108], which uses this strategy throughout.

As noted above the standard parallel implementation of GAMESS-UK uses an approach between these two extremes. Most of the data is replicated, but when a parallel linear algebra operation is to be performed the data is copied into a global array; GA tools are used to perform the operation, and then the data is copied back into a replicated object. Whilst this solves some of the problems associated with a replicated data strategy the majority still remain, especially the large memory requirements and the re-replication overheads. This would suggest that a move to a truly distributed data model would be desirable. However, distributed data is not a universal panacea, for as with replicated data it is difficult to implement certain algorithms efficiently in parallel using it. One is the build of the Fock matrix; consider the form of the coulomb and exchange contributions to element $\mu\nu$ of the Fock matrix:

$$F_{\mu\nu} = H_{\mu\nu}^0 + \sum_{\lambda\sigma} P_{\lambda\sigma}[(\mu\nu|\sigma\lambda)] - \frac{1}{2}(\mu\lambda|\sigma\nu). \quad (35)$$

In principle, each element of the density matrix contributes to every element of the Fock matrix. Within a distributed data parallelization strategy each processor would hold a portion of both the Fock and density matrices. Therefore, one way to implement this is for each processor to calculate all the integrals appropriate for the $\mu\nu\lambda\sigma$ it currently holds, form the contribution to the Fock matrix and then pass the part of the density matrix that it currently holds onto the next processor, in a systolic-loop-type algorithm. This is repeated until the Fock matrix is fully formed. However, this algorithm has a number of drawbacks, the main one being that of load balancing. Each communication of the density matrix will require synchronization between a pair of processors (assuming a standard two-sided communication protocol), and, further, since the virtual processor topology used in the above algorithm is a ring this implies a loose synchronization between all processors. Therefore, to avoid load imbalance at each stage all the processors must perform the same amount of work, and this is not straightforward to achieve. So while good scalability in the Fock build can be achieved readily using a replicated data strategy, it is much more

difficult using distributed data. Though these problems can be addressed, as they are in the NWChem package by use of one-sided communications, we have taken a more pragmatic approach by implementing a partially replicated version of the SCF and DFT modules, which we describe in section 5.7.

5.4. The GA-based DFT and MP2 modules

The Density Functional Theory (DFT) module within GAMESS-UK is a MPP implementation of the Hohenberg–Kohn–Sham formalism [54, 114] of DFT. The Gaussian basis DFT method breaks the Hamiltonian down into the same basic one- and two-electron components as traditional Hartree–Fock (HF) methods, with the latter component further reduced to a Coulomb and an exchange-correlation term. The treatment of the former can be accomplished in identical fashion to that used in traditional self-consistent field (SCF) methods or from the commonly used Dunlap charge density fit [55, 56].

The performance of both DFT and DFT second derivative modules on a variety of high-end parallel hardware is shown in table 4. This includes both the IBM systems at HPCx (<http://www.hpcx.ac.uk>)—the Phase1 p690 system with 1.3 GHz power4 CPUs and the Phase2 p690+ system, with 1.7 GHz power4+ CPUs—and the SGI Altix 3700 systems, with 1.3 GHz Itanium2 CPUs (‘newton’) at CSAR (<http://www.csar.cfs.ac.uk>) and 1.5 GHz CPUs (‘ram’) at ORNL (<http://www.ccs.ornl.gov>). Older systems also shown include ‘teras’, the SGI Origin 3800/R14k-500 at SARA (<http://www.sara.nl>), and the HP/Compaq AlphaServer SC ES45/1000 system, ‘TCS1’, at the Pittsburgh Supercomputing Centre (<http://www.psc.edu>). Unless stated otherwise, the DFT calculations did not exploit CD fitting, but evaluated the coulomb matrix explicitly.

Note that all reported timings for the IBM p690+ system refer to the High Performance Switch (HPS) microcode contained in Service Pack 7 (SP7), unless stated otherwise. Considering the DFT results, modest speedups ranging from 55 (IBM p690) to 78 (SGI Origin 3800) are obtained on 128 processors for the larger cyclosporin calculation. Somewhat better scalability is found in both valinomycin DFT calculations where a greater proportion of time is spent in integral evaluation arising from the more extended basis sets [115]. Speedups of 87 and 102 are obtained on 128 processors of the IBM p690 and p690+, the latter figure comparable to that found on the AlphaServer (97), the SGI Origin 3800 (102) and Altix 3700/1300 (104) in the 1620 GTO calculation. The performance of both GAMESS-UK and NWChem using releases of the Global Array Tools up to and including V3.3 showed a

Table 4. Time in wall clock seconds for GAMESS-UK benchmark calculations on the Compaq AlphaServer SC ES45/1000, SGI Origin 3800/R14k-500, IBM p690 and p690+, and SGI Altix 3700/1300 and 3700/1500.

CPU's	SGI Origin 3800/R14k-500	Compaq Alpha ES45/1000	IBM p690	IBM p690+(SP7)	IBM p690+(SP9)	SGI Altix 3700/1300	SGI Altix 3700/1500
Cyclosporin (1000 GTOs) DFT/B3LYP 6-31G							
32	1049	580	556	385	392	420	379
64	587	355	369	254	231	249	221
128		269			160	173	
Cyclosporin (1855 GTOs) DFT/B3LYP 6-31G**							
32	3846	2084	2047	1366	1319	1522	1383
64	2271	1276	1429	876	804	900	808
128	1585	896	1193	739	555	612	615
Valinomycin (882 GTOs) DFT/HCTH							
32	1947	1024	1099	802	816	893	780
64	1065	577	635	463	441	487	420
128	623	369	489	309	260	390	263
Valinomycin (882 GTOs) DFT/HCTH (<i>J</i> -fit)							
32	649	429	473	269	279	258	237
64	425	283	370	243	193	169	
128	352		355				
Valinomycin (1620 GTOs) DFT/HCTH							
32	9636	4738	4723	3513	3502	4147	3595
64	5110	2556	2690	1966	1886	2195	1901
128	3024	1557	1777	1259	1104	1277	1190
256	2026	1013	1343	985	731		
$(C_6H_4(CF_3))_2$ second derivatives DFT/HCTH							
32	1772	922	842	457	471	567	497
64	1081	557	471	362	295	325	338
128	826	337	294	240	180	238	326

marked deterioration on SGI Altix systems that involve partitioning across two or more linux64 partitions. This deficiency has been remedied in V3.4 of the GA library, so that the timings on the multi-partition CSAR Altix system are now consistent (allowing for the clock speed ratio of 1.15) with those on the 256-way single system image characterizing the 'ram' system at ORNL (see table 4). With the release of SP9, we see that the IBM p690+ and Altix systems exhibit comparable performance in the 128 CPU DFT calculations. The enhanced performance of the DFT 2nd Derivative module on the IBM p690 and p690+ arises from the decreased dependency on latency exhibited by the current implementation compared to the DFT energy module.

The less than impressive scalability of both GAMESS-UK and NWChem on the IBM p690 and p690+ systems (up to and including SP7) arises to some extent from the dependency of both codes on a Global Array implementation that is dependent on IBM's LAPI communication library [117]. The implementations of LAPI on POWER4-based architectures has proved

far from optimal, with the measured latencies and bandwidths significantly inferior to those measured on corresponding POWER3-based systems. This provided one of the main motivations behind a re-implementation of the SCF and DFT modules using only tools based on MPI (see section 5.7). The recent release of Service Pack 9 (SP9) on the p690+ has done much to address these shortcomings, as is evident from the timings on higher processor counts in figure 4 (vide infra).

5.4.1. *The fitted Coulomb DFT module.* Following on from the discussion of section 3.2 we note that the parallelization of the Fitted Coulomb module is performed trivially by distributing the evaluation of integrals with the same set of wavefunction basis functions (pq) over the processors. Because the integrals are stored and read when needed a static load balancing scheme has to be applied to avoid communication. If the amount of memory available is not sufficient to store all of the integrals then the ones that were not stored can be recalculated. This combined in-core/direct approach can

be optimized further by storing the largest integrals and recalculating the small ones. This way, as with direct-SCF, the Schwarz inequality can be tightened by including the density factors, reducing the number of integrals further. The inversion of the matrix \mathbf{V} was implemented using the PeIGS diagonalization package, with the matrix distributed column-wise, to maximize data locality in the evaluation of the fitting coefficients.

Timings for a number of DFT calculations on the valinomycin molecule, conducted on the variety of high-end hardware under consideration, are shown in table 4. Calculations used a DZV_A2 DGauss basis of 882 functions and the HCTH functional, with timings reported for calculations in which the Coulomb matrix was evaluated explicitly (J -explicit) and for those that used CD fitting (J -fit). The latter employed an A1_DFT fitting basis (3012 functions). It can be seen that the current implementation of the fitted Coulomb module provides significant benefit in terms of time to solution, with all systems showing a significant reduction in time when using the fitted Coulomb matrix compared to explicit treatment of the Coulomb term. These reduction factors on 64 CPUs range from 2.51 (SGI Origin 3800/R14k) to 1.72 (IBM p690) across the variety of systems given in table 4. A comparison with the explicit- J timings shows, not surprisingly, poorer scalability of the fitted approach given the greater dependency of this approach on interconnect.

5.4.2. The parallel MP2 module. Substantial modifications were required to enable the second-order Möller Plesset (MP2) gradient to be computed in parallel [116]. The conventional integral transformation step is now omitted, with the SCF step performed in direct fashion and the MO integrals, generated by re-computation of the atomic orbital (AO) integrals, stored in the global memory of the parallel machine. The storage and subsequent access is managed by the GA tools. The basic principle by which the subsequent steps are parallelized involves each node computing a contribution to the current term from molecular orbital (MO) integrals resident on that node. For some steps, however, more substantial changes to the algorithms are required. For the MP2 gradient, the construction of the Lagrangian (the right-hand side of the coupled perturbed Hartree-Fock (CPHF) equations) requires MO integrals with three virtual orbital indices. Given the size of this class of integrals, they are not stored, the required terms of the Lagrangian being constructed directly from AO integrals. A second departure from the serial algorithm concerns the MP2 two-particle density matrix. This quantity, which is required in the AO basis, is of a similar size to the two-electron integrals and is stored on disk in the conventional algorithm, but

generated as required during the two-electron derivative integral generation from intermediates stored in the GAs.

5.5. The parallel valence bond module

The VB program has now been parallelized. Extending earlier implementations [117] of the separate program using MPI, the integrated version now employs the Global Array tools. In addition to the major time-consuming part of the code, the calculation of the matrix elements, the four-index transformation has also been adapted. Although this is not so efficient by itself, it does enhance the overall efficiency of the code. The parallel efficiency is quite adequate, as is shown in figure 2.

Using the parallel code we have been able to perform calculations on the aromaticity of benzyne [118] and benzene which, for the first time, employ a completely balanced description of the systems to be compared when trying to estimate the resonance energy. This is a breathing-orbital-type approach [119], where both structures in a two-structure calculation are described with their own set of orbitals.

5.6. Parallel Direct-CI

In the Direct-CI method [15, 120–122] the explicit calculation of the CI Hamiltonian matrix is avoided, but instead the matrix-vector product

$$\mathbf{Z} = \mathbf{H}\mathbf{C} \quad (36)$$

is calculated directly from the molecular integrals. Here \mathbf{C} is a trial vector of CI expansion coefficients and \mathbf{Z} is the resulting matrix-vector product. The Hamiltonian matrix elements between two CSFs μ and ν ($H_{\mu\nu}$) can be written in terms of coupling coefficients and molecular integrals

$$H_{\mu\nu} = \sum_{pq} A_{pq}^{\mu\nu} [pq] + \sum_{pqrs} A_{pqrs}^{\mu\nu} [pq|rs]. \quad (37)$$

In the parallel implementation, the original structure of the Direct-CI program as described in detail in [15] is maintained. The generation of the \mathbf{H} matrix and \mathbf{Z} vector is divided in model contributions, where each model depends on the integral indices (number of indices in internal and external space). For each model, a set of coupling coefficients and a sorted list of integrals are stored on disk. Parallelization of the generation of the \mathbf{Z} -vector is achieved by dividing the coupling coefficients or the integrals (the latter only the $(ab|cd)$ and $(ia|bc)$ integrals) among the nodes.

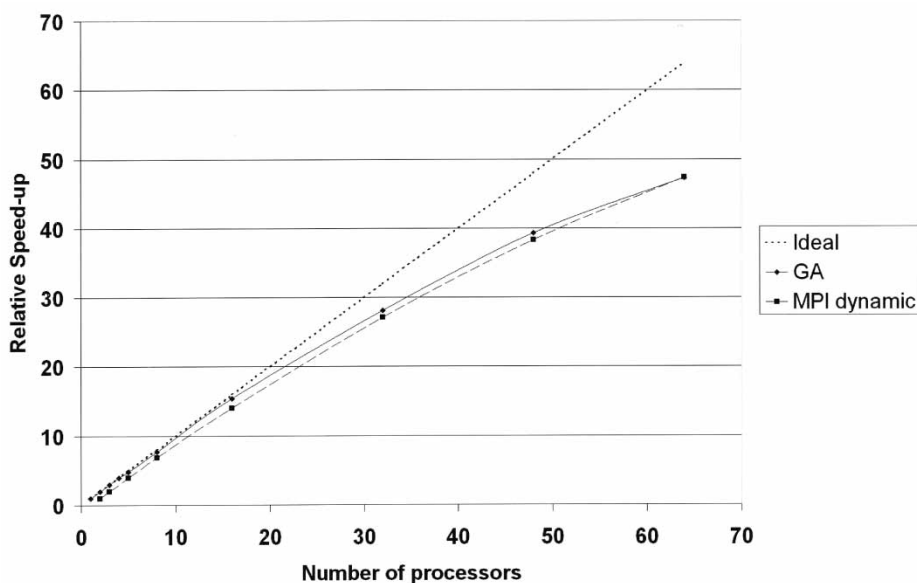


Figure 2. Scaling of the GAMESS-UK VB Module for MPI with dynamic load balancing and GA with static distribution of work units. In the first case, one processor is used for distributing work units. Speed-ups are relative to the GA single CPU result.

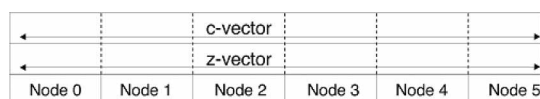


Figure 3. The distribution of the global array, which holds the C- and Z-vectors, among six nodes.

An efficient parallelization is obtained by the implementation of the Global Array tools [109] for storing the most current C- and Z-vectors. Each node generates a part of the total Z-vector, which is added to the Global Array with a distributed update. For this, a global (two-dimensional) array is created that can hold one C-vector and one Z-vector, and it is distributed among the nodes as figure 3 shows. This arrangement of the C- and Z-vectors in memory has the advantage that each node has a part of the C-vector and its matching part of the Z-vector in core for the calculation of a new C-vector.

The Davidson sub-space manipulations are parallelized automatically by the layout of the global array: each node operates on its own partial C- and Z-vectors to generate the partial dot-products, and the total is obtained by a global sum.

A CI calculation on cyclopentapyrene was used as a test case. The CI expansion had a length of 106 353 143 CSFs. The program is found to scale reasonably well, up till 32 processors, showing a speedup of 20 at this processor count. It is expected that, for larger CI jobs, the scaling should improve further. A comparison of our speedups and that with other CI programs like the selected DIESEL-CI program [89] (which is found to

scale to 32 processors [123]) and for the COLUMBUS system [124] shows that our program behaves similarly. The present parallel implementation of the Direct-CI program enables large-scale CI jobs, and it benefits especially from the large amount of shared memory that can be allocated when multiple processors are used.

5.7. An MPI/ScaLAPACK implementation of the SCF module

As discussed in the overview of GAMESS-UK parallelization (section 5.1) the program is still predominantly a replicated data one. However, the replicated data SCF and DFT modules were showing limited scaling, in particular those based on SP7 on the HPCx system (see figure 4). There were problems related to the largest chemical system which would fit in the available memory, and problems with the scaling of the GA-based linear algebra when implemented over LAPI (*vide infra*). In the light of these problems we investigated an alternative approach in which:

- (i) MPI-based tools (such as ScaLAPACK) were used in place of GA and LAPI, and
- (ii) all data structures except those required for the Fock matrix build were fully distributed.

A partially distributed model was chosen because, in the absence of efficient one-sided communications, it is difficult to efficiently load balance a distributed Fock matrix build. The obvious drawback of this is that some large replicated data structures are required.

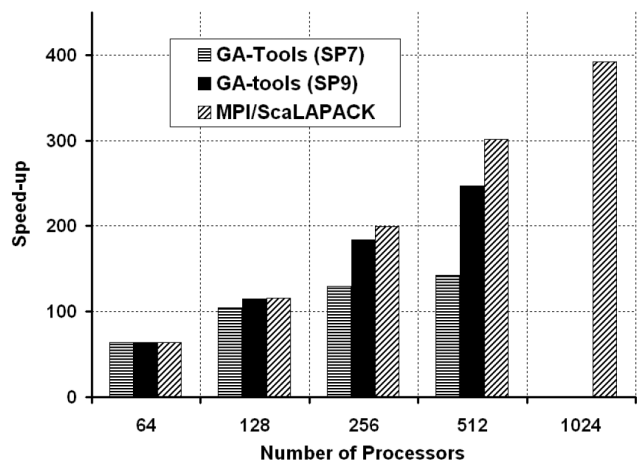


Figure 4. Scaling of the MPI-based SCF program.

However, these are kept to a minimum. For a closed-shell Hartree–Fock or density functional theory calculation only two replicated matrices are required, one Fock and one density, while for unrestricted calculations this is doubled. Further, the symmetry of these matrices is used to cut down on the required memory.

5.7.1. Implementation and current status. The MPI/ScaLAPACK SCF module is written in standard conforming Fortran 95, and uses MPI for message passing. The code is built upon a Fortran module that implements a derived matrix type and operations on such objects, the defined operations being those commonly use in quantum chemistry. The matrices can be either replicated or distributed, this being set when they are created. After that the routines that use the module need not know how a given matrix is distributed, thus simplifying the high level implementation.

For distributed matrices the matrix operations are, in general, performed using ScaLAPACK [125, 126], while LAPACK [113, 127] is used for replicated matrices. Using multiple BLACS [128] contexts allows linear algebra operations to be performed on a subset of the processors, and this is exploited by the matrix module to use the extra level of parallelism available in unrestricted calculations, i.e. that over the spins. These underlying distributed operations are transparent to the high level routines. This is achieved by overloading many of the operations on the matrix type, allowing them to act on both single matrices and arrays of them.

At present, only the MPI/ScaLAPACK SCF driver uses distributed data. As a result the memory limitations imposed by the use of replicated data in other parts of GAMESS-UK, particularly the initial guess at the wavefunction and the analysis of the solution, limit

the size of problem that can be solved. However, the implementation of a general matrix module makes it straightforward to move these to a distributed data strategy, and work is currently in progress to address this.

The performance of the MPI/ScaLAPACK SCF module has been compared with the GA implementations (Service Packs 7 and 9) of GAMESS-UK on HPCx, the large IBM p690 + cluster. The system studied was a Zeolite-Y cluster using 3752 spherical harmonic basis functions. Figure 4 shows the speed-up for the three calculations. The speed-ups have been obtained by assuming perfect scaling to 64 processors. The 1024 processor time (with a speed-up of around 400) corresponds to an elapsed time of 20 min.

It can be seen that, at the lowest number of processors considered, the run times are comparable, with the GA implementation being slightly the quicker. However, as the number of processors is increased the MPI/ScaLAPACK module displays much better scaling, and by 512 processors it is over twice as fast as that based on the GA tools (SP7). This improved scaling is due to the better performance of the linear algebra, though it can be seen that the scaling, while better, is far from perfect. This is for two reasons:

- (i) some of the linear algebra operations are not scaling well, especially the diagonalizer;
- (ii) as the MPI/ScaLAPACK module uses MPI-1 throughout, one-sided communications cannot be used, making dynamic load balancing of the integrals impossible. As a consequence, load imbalance is becoming important at higher processor counts.

Both of these effects are due to the system being somewhat too small to run efficiently on such large processor counts, but still the scalability is good up to 256 processors, and acceptable at 512.

6. Some applications of GAMESS-UK

In this section we provide some examples of recent application work undertaken with the GAMESS-UK program.

6.1. DFT calculations on biomolecules: exploring the charge distribution of isocitrate lyase

As a Dutch Challenge project, calculations on the enzyme isocitrate lyase were performed on two 64 node subsystems of the Dutch national supercomputer Aster, an SGI Altix 3700 system with 1.3 GHz Itanium2 processors and 2 GByte memory per node.

The aim of this project is to develop a model suited for the design of a new class of inhibitors of

Mycobacterium tuberculosis (TBC). In the case of chronic tuberculosis caused by persistent mycobacteria, TBC uses an alternative pathway to sustain its citric acid cycle, where it relies on the isocitrate lyase enzyme for its metabolism. In this state, the enzymes targeted by extant drugs are not used by the bacterium, rendering these drugs ineffective. However, non-drug-like chemical compounds that deactivate isocitrate lyase have been shown to kill persistent mycobacteria *in vitro*, indicating that this enzyme is a potential target for treatment of chronic tuberculosis.

In order to design inhibitors of this enzyme, we need a reliable charge distribution of the active site of the enzyme and its surroundings. Furthermore, we will try to use the computed electron density as a geometric description of the protein binding site, analogous to the way steric grids are used in traditional mechanistic approximations. We will subsequently use the charge distribution and the geometry of the binding site to design compounds that inhibit isocitrate lyase and that have the potential to become a drug.

Density Functional (B3LYP) and *ab initio* Hartree–Fock calculations have been performed on segments of the enzyme in spheres of up to 16 Å around the active site, both with and without an embedded ligand. This involved up to 1800 atoms and 10 000 atomic orbitals using 62 processors in parallel. The geometry used was derived from X-ray crystallography [129].

To assign basis functions, shells were assigned based on the distance to the centre of the enzyme, as shown in figure 5. The basis sets were designed to give the best description near the centre and slowly deteriorate going outwards. Thus two basis sets were chosen. *TZVP* consist of the shells TZVP, DZP, TZ, 6-31G, and STO6G. *SHELL4* consists of successive DZVP, 6-31G, STO6G, and STO3G basis sets. *TZVP* could only be used for a 14 Å radius; the biggest calculation with this basis employed 9556 orbitals for 1257 atoms. With *SHELL4*, systems with a 16 Å radius could be handled. The biggest calculation with this basis involved 8315 orbitals and 1981 atoms.

The results are analysed by computing point charges using DMA [48], natural population analysis [130] and by calculating molecular electrostatic potentials and electron densities on a $12 \times 12 \times 12$ Å grid around the active site, totalling 22 692 points. The densities will be used to assess the sterically accessible volume of the enzyme.

The timing analysis shows that the HF and DFT calculations scale reasonably well up to 40–50 processors on this machine. The scaling of the parallel electrostatic potential calculation is excellent and would actually allow even denser grids. The timings

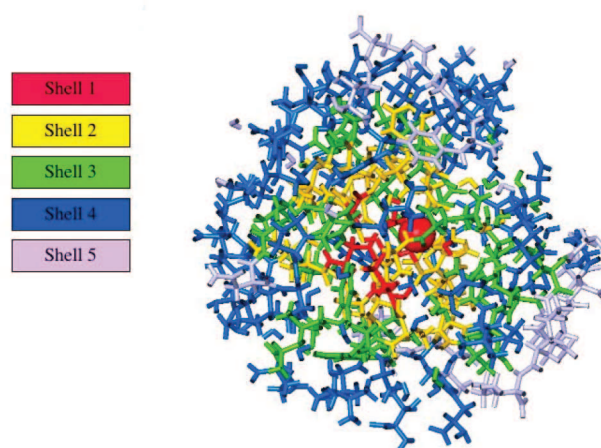


Figure 5. The isocitrate lyase subsystem, showing the shells used for the computational study.

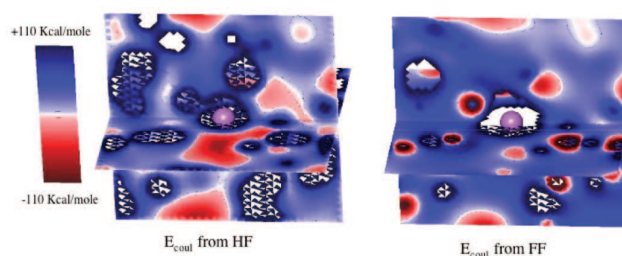


Figure 6. Slices through the electrostatic potential at the active site of the enzyme.

suggest that these calculations are quite feasible even on now quite affordable 64 node Itanium clusters. Some weeks of calculation to establish a reliable charge distribution and molecular potential is not excessive, as this provides the stage on which many docking experiments may be performed.

Preliminary results in figure 6 show the molecular potential from the *ab initio* Hartree–Fock calculations (left) next to the molecular potential calculated from force-field charges (right). These results suggest that the now commonly used charge models may not be adequate. We now have to establish if this has serious consequences for the docking of inhibitors and thus for computer-aided drug design.

Thus we will repeat the calculations for a well-documented reference protein to allow comparison with published docking work. To be able to make progress on the possible new target system of the tuberculosis bacterium [129], isocitrate lyase, we will use computational drug design methods [131, 132] to design actual drugs using the molecular potential calculated for an empty enzyme. Given that supercomputers remain only a few years ahead of the mainstream, commodity

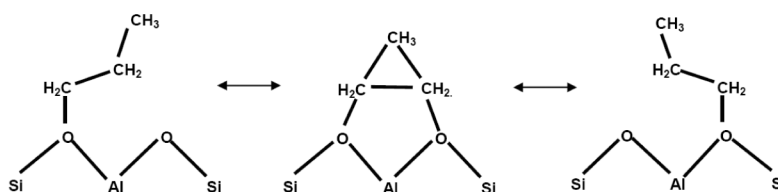


Figure 7. The methyl shift reaction for the propenium species at a Bronsted acid site.

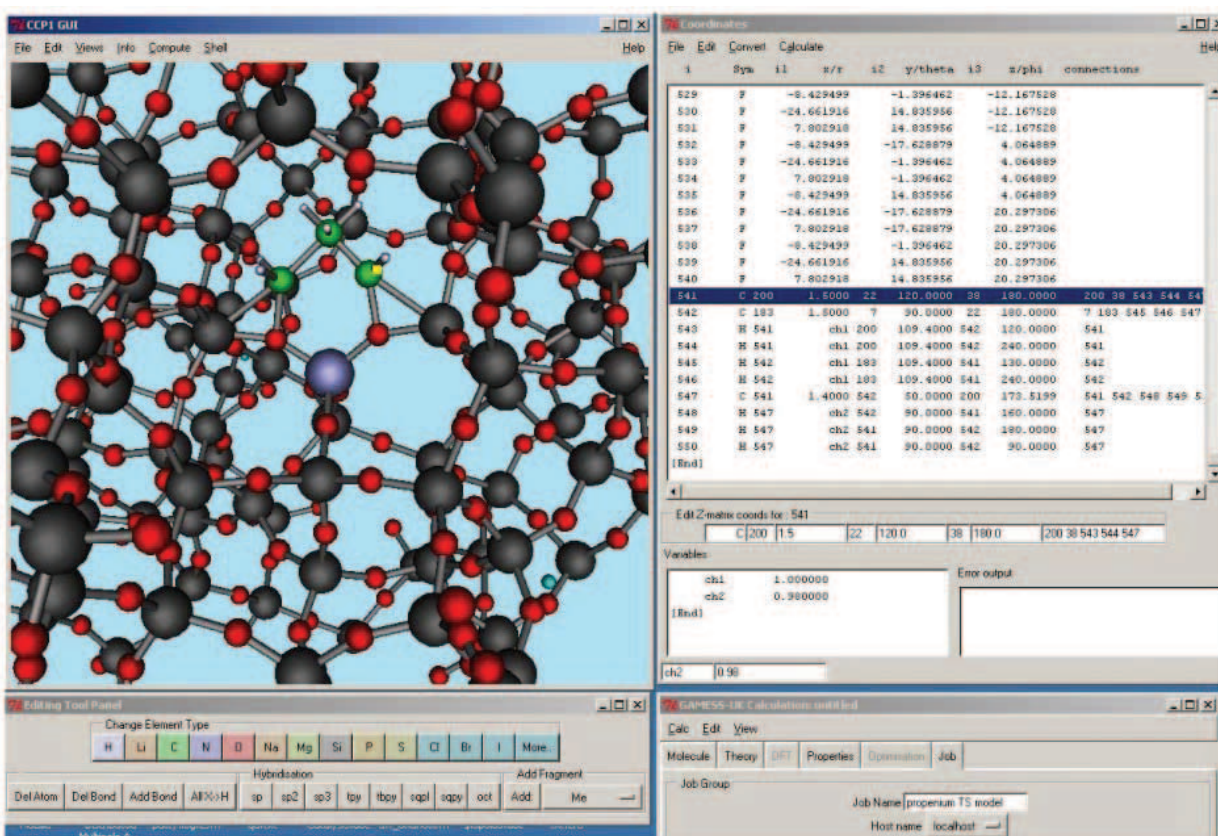


Figure 8. The methyl shift transition state under construction in the CCPI GUI.

marketplace, the techniques explored may soon find acceptance within the pharmaceutical industry.

6.2. Zeolite catalysis

We have used the QM/MM capabilities of ChemShell to couple GAMESS-UK HF and DFT functionality to the CFF zeolite force-field as implemented in the DL_POLY [133, 134] code. This approach has been applied to a number of zeolite reactions [95, 97, 135], one of which, the methyl shift of the adsorbed propenium ion, is shown here [97] (see figure 7).

Figure 8 shows how the model for this reaction appears while being constructed, using a mixture of Cartesian coordinates for the zeolite fragment and

internal (*Z*-matrix coordinates) for the propenium group.

Using a gas-phase model, the reaction is symmetrical as the two propenium end-points are equivalent. In the embedded case, the end-points and therefore the forward and reverse barriers are no longer equivalent, and we find that the barriers are significantly lower in the embedded case. We have analysed the QM/MM energies in table 5. The energy in the mechanical embedding case is easy to decompose into QM and MM terms, and this is performed in the first section of table 5.

In the electrostatic embedding case, the classical atoms contribute to the Hamiltonian as point charges. For the purpose of the analysis the QM–MM

Table 5. The decomposition of QM/MM energies for the propenium system. Z-(C,H) nb is the zeolite...hydrocarbon non-bonded energy.

Model	Energy	Propoxide 1	TS	Propoxide 2	Barrier	Barrier I
Gas phase	Total	0	316	0	316	316
Mechanical	Total	0	247	55	247	192
	QM	0	261	38	261	223
	MM	0	-6	9	-6	-15
	Z-(C,H) nb	-12	-20	-4	-8	-16
Electrostatic	Total	0	253	68	253	185
	QM-MMElec	-93	-103	-100	10	-3
Polarized	QM-MM Pol	-30	-45	-33	-15	-12

electrostatic interaction is estimated by calculating the interaction of a classical representation of the QM region (Dipole Preserving Charges, DPC) with the MM point charges. The role of polarization of the classical partition is estimated using single-point calculation of the interaction of the DPC representation of the QM region with local polarizabilities at Si and O sites, implemented using the DRF module in GAMESS-UK (last row of table 5). These analyses provide some insight into the way the zeolite lowers the barrier for the reaction. Mechanical (steric and strain) factors and electrostatic factors are both found to play an important role in this process. The mechanical effect arises because the transition state fits better into the zeolite framework structure than the propoxide minima do, lowering the barrier. The electrostatic embedding model reveals that all the QM clusters are stabilized by electrostatic interactions, but this has smaller effects on relative energies (in fact, the barrier in one direction increases, and the other one decreases). MM polarization, however, is found to contribute significantly to the lowering of the barrier this is because the transition state has a larger dipole moment than the minima (11 vs 6 Debye). Polarization of the zeolite framework stabilizes the charge separation, lowering both barriers [97].

6.3. Computational studies of electronic spectra

Multi-reference, multi-root CI studies of low-lying electronically excited states provide a stringent test of the ability of CI methods to interpret experimental absorption spectra. GAMESS-UK has been extensively used in a number of combined experimental and theoretical studies, with the excited state calculations using the selection-based, multi-reference multi-root MRD-CI module due originally to Buenker *et al.* These studies have been greatly assisted by access to the Synchrotron Source at Daresbury, with the Vacuum ultraviolet (VUV) spectra recorded using the Daresbury Molecular Sciences Absorption Apparatus coupled to

beamline 3.1. We here summarize the approach and findings in a number of such studies.

Early calculations [136, 137] involved somewhat modest basis sets, with two different basis sets typically employed in these studies. A DZP basis set augmented by centre-of-mass-based Rydberg (3s,2p,2d) functions (DZPR) was used for the Rydberg state studies. This basis set permits a wide study of both valence and Rydberg states, enabling a focus on the lowest two Rydberg series from each of the six lowest IPs with some confidence. A more extended triple-zeta valence basis with polarization functions (TZVP) was usually used for the ground state molecular electronic properties, typically evaluated at the MP2 level. While these early calculations were limited to ca. 10^4 selected CSFs, the recent incorporation of the semi-direct Table-driven CI module has significantly increased these limits (to ca. 5×10^5 CSFs), enabling the use of far more extensive basis sets.

6.3.1. *The electronic states and molecular properties of the azines.* A series of studies have considered all the known azines to date, including $C_{6-n}H_{6-n}N_n$ with $n = 0$ (benzene) [138], $n = 1$ (pyridine) [139], each of the diazines ($n = 2$) [140, 141], and the 1,3,5-isomers [142] and 1,2,4-isomers [143] of the triazines ($n = 3$), and 1,2,4,5-tetrazine [144]. The final species studied in this series was 1,2,3-triazine [145]. In each case, a combination of optical and low-energy electron energy-loss spectroscopy with CI calculations led to the assignment of the singlet and triplet states in an energy range up to ca. 10 eV, and a reconsideration of the cationic states produced by photoelectron spectroscopy.

Considering the valence states of 1,2,3-triazine [146], the VUV spectrum shows an intense absorption band around 7.4 eV, involving excitation of $^1\pi\pi^*$ type states related to the $^1E_{1u}$ state of benzene. A broad band maximum around 4.7 eV and absorption intensity between 5 and 6 eV are attributed to excitations of the low-lying $\nu\pi^*$ states. In the electron impact spectrum,

the most intense energy-loss band is around 3.6 eV, which must relate to the excitation of νp^* states that are not developed in optical absorption and hence is likely to include a triplet component. The two lowest Rydberg states, $11a_1 3s$ and $7b_2 3s$ are assigned to weak maxima in the spectrum around 6.4 and 6.7 eV; suggested energies for the p- and d-type Rydberg states are given in [145].

For valence states of type πp^* , theory suggests that the two $^1\pi p^*$ states which relate to the (doubly degenerate) benzene $^1E_{1u}$ state lie within the 7.4 eV band, although as for other compounds of this type, the computed energies are too high by about 1 eV. The assumption, from comparison of the experimental spectrum with that of benzene, that the two lower lying $^1\pi p^*$ states contribute to absorption intensity around 6.5 and 5.0 eV, is supported by both computed energies (6.83 and 5.5 eV, respectively) and intensities. In comparing theory with experiment for νp^* states, three optically allowed νp^* states in the low-energy region are computed to lie between 4 and 5.6 eV, the first two at 4.1 and 5.1 eV being the most intense. The second of these is related to the 4.7 eV absorption and the former to the leading edge of the UV band (which shows the hint of a shoulder at ~ 3.8 eV). The lowest-lying excited state detected in this series of experiments is seen in electron energy-loss at 3.6 eV, which is the computed energy of the lowest triplet state, 3B_1 (νp^*). Given this accord, we are confident that the $^1\nu p^*$ states are well represented in the theory and hence that another is excited around 5.6 eV in the VUV spectrum. Satisfactory agreement is found between the experimental assignments for Rydberg states (made using the Rydberg formula) and the computed energies.

6.3.2. Assignment of the electronic states of pyrazole and isoxazole. The gas-phase VUV absorption spectrum of pyrazole [146] and isoxazole [147] have been assigned in the light of multi-reference multi-root CI calculations using basis sets up to quadruple zeta quality and containing both valence and Rydberg-type functions. A very intense band centred near 7.8 eV in the spectrum of pyrazole appears to arise from the summation of three calculated bands of $\pi\pi^*$ character, of which the first and third are the most intense. The window resonance near the band maximum is ascribed to mutual annihilation of a Rydberg state and valence state. The electron energy loss (EEL) spectrum obtained previously showed low-lying triplet states at about 3.9 and 5.1 eV, respectively; the computations suggest that two triplet ($^3\pi\pi^*$) states lie within the 3.9 eV band. The UV-photoelectron spectrum confirms the identity of the first three IPs as $\pi_3 < \pi_2 < LP_N$ (lone pair); as a consequence, most of the excited singlet and triplet

states below 10 eV are likely to be of $\pi\pi^*$ and $\pi\sigma^*$ character with both valence and Rydberg types. However, a number of $\sigma\sigma^*$ and $\sigma\pi^*$ states including those derived from the lone pair electrons were obtained from the CI study, which give predictive values for the onset of such excitations. The highest pair of occupied orbitals (π_3 and π_2) both interact strongly with the lowest pair of valence VMOs (π_4^* and π_5^*), leading to a wide variety of valence states, and as such the molecule behaves much more like a system with degenerate pairs of π -MOs.

The VUV absorption spectrum of isoxazole (5–10.8 eV, 250–115 nm) has been recorded for the first time [147]. The molecule has also been probed using electron impact with electrons of different incident energies and the He(I) photoelectron spectrum has been re-measured. Calculated energies for Rydberg states are close to those expected. More than 30 valence excited states having finite oscillator strengths are computed to lie between 6 and 12 eV, but most of the intensity in the VUV absorption spectrum is from excitation of states of $^1\pi\pi^*$ character. The calculations suggest that the first two states lie at about 6 and 7 eV, respectively, and are separated by a state of type $^1\sigma\pi^*$ where σ is nitrogen lone pair. The lowest-lying triplet states, located by experiment at about 4.1 eV (290 nm) and 5.3 eV (225 nm), are $^3\pi\pi^*$. Short-lived anionic states (electron–molecule resonances) have been detected in both inelastic scattering and dissociative electron attachment channels.

7. Conclusions and outlook

The GAMESS-UK package offers a wide range of quantum mechanical wavefunctions, capable of treating systems ranging from closed-shell molecules through to the species involved in complex reaction mechanisms. The availability of a wide variety of correlation methods provides the necessary functionality to tackle a number of chemically important tasks, ranging from geometry optimization and transition-state location to the treatment of solvation effects and the prediction of excited state spectrum. With the availability of relativistic ECPs and the development of ZORA, such calculations may be performed on the entire Periodic Table, including the lanthanides. Future developments look in particular to enhance the existing DFT functionality, with both the treatment of excited states through time-dependent methods and the computation of a broader variety of molecular properties, e.g. NMR chemical shifts.

Energy, gradient and frequency calculations have been shown to run effectively on systems with many processors, with the MPI and Global Array-based

parallel implementation providing an excellent tool for the treatment of molecular systems with up to 10 000 basis functions, while exploiting hardware with up to 1000 processors. We plan to continue the development of GAMESS-UK on parallel systems, providing both additional parallel functionality (e.g., CCSD(T)) and allowing for the treatment of a broader range of molecular systems. The treatment of larger systems on a greater number of processors is currently in progress through the development of a MPI/ScaLAPACK-based SCF/DFT module. An effective task-farming strategy for dealing with a large number of smaller molecular systems is also under development, targeting the efficient treatment of problems in computational screening.

Acknowledgements

The GAMESS-UK program has resulted from the efforts of many researchers: we gratefully acknowledge the collaborative efforts of the following: R.D. Amos, V. Bonacic-Koutecky, R.J. Buenker, C.P. Byrman, L.C.H. van Corler, F. Dijkstra, M. Dupuis, J.J. Engelberts, S. Faas, N.C. Handy, R.J. Harrison, I.H. Hillier, P.J. Knowles, J.H. Langenberg, A.J.H.M. Meyer, T. van Mourik, W. von Niessen, A.P. Rendell, P.J.A. Ruttink, V.R. Saunders, J.G. Snijders, A.J. Stone, P.D. Todorov, D.J. Tozer, J. Verbeek, H.M. Vinkers, A.H. de Vries, P.E. Young and R. Zwaans. We are grateful to those who have supported the developments financially. Much of this work was performed under the auspices of the CCP1 support programme at the CCLRC Daresbury Laboratory with funding from the EPSRC. Work at the University of Utrecht was sponsored by the Stichting Nationale Computerfaciliteiten (National Computing Facilities Foundation, NCF) for the use of supercomputer facilities, with financial support from the Nederlandse Organisatie voor Wetenschappelijk Onderzoek (Netherlands Organization for Scientific Research, NWO). R.W.A.H. acknowledges financial support from NWO grant 700.53.401. We thank Drs. Jarek Nieplocha and Edoardo Aprà (PNNL) for their efforts in porting the GA tools and PeIGS to enable GAMESS-UK to be supported on a wide range of parallel platforms. The QM/MM interfaces have been developed in collaborative projects; CHARMM/GAMESS-UK with the group of B.R. Brooks, while ChemShell development was performed in collaboration with the groups of C.R.A. Catlow and W. Thiel. We thank our collaborators in the applications projects described in section 6, in particular A.H. de Vries for the zeolite calculations and M.H. Palmer for the electronic spectra studies. The isocitrate project was conducted in collaboration with M.R. de Jonge and the Center for

Molecular Design in Vosselaar, Belgium, a division of Janssen Pharmaceutica (a Johnson & Johnson Company). The computer time was supplied by the NWO, through the NCF.

References

- [1] GAMESS-UK is a package of *ab initio* programs written by M.F. Guest, J.H. van Lenthe, J. Kendrick, K. Schöffel, P. Sherwood, with contributions from R.D. Amos, R.J. Buenker, H.J.J. van Dam, M. Dupuis, N.C. Handy, I.H. Hillier, P.J. Knowles, V. Bonacic-Koutecky, W. von Niessen, R.J. Harrison, A.P. Rendell, V.R. Saunders, A.J. Stone, D.J. Tozer, A.H. de Vries. The package is derived from the original GAMESS code due to M. Dupuis, D. Spangler, J. Wendoloski, NRCC Software Catalog, Vol. 1, Program No. QG01 (GAMESS) (1980) (see <http://www.cfs.dl.ac.uk>).
- [2] M.F. Guest, V.R. Saunders, *Molec. Phys.*, **28**, 819 (1974).
- [3] M.W. Schmidt, K.K. Baldrige, J.A. Boatz, S.T. Elbert, M.S. Gordon, J.H. Jensen, S. Koseki, N. Matsunaga, K.A. Nguyen, S.J. Su, T.L. Windus, M. Dupuis, J.A. Montgomery, *J. Comput. Chem.*, **14**, 1347 (1993).
- [4] F.W. Brobrowicz, W.A. Goddard, In *Modern Theoretical Chemistry*, p. 79, Plenum, New York (1977).
- [5] J.A. Pople, W.J. Hehre, *J. Comput. Phys.*, **27**, 161 (1978).
- [6] M. Dupuis, J. Rys, H.F. King, *J. Chem. Phys.*, **65**, 144 (1976).
- [7] H.B. Schlegel, *J. Chem. Phys.*, **77**, 3676 (1982).
- [8] M. Dupuis, H.F. King, *J. Chem. Phys.*, **68**, 3998 (1978).
- [9] P.J. Hay, W.R. Wadt, *J. Chem. Phys.*, **82**, 270 (1985).
- [10] W.R. Wadt, P.J. Hay, *J. Chem. Phys.*, **82**, 284 (1985).
- [11] P.J. Hay, W.R. Wadt, *J. Chem. Phys.*, **82**, 299 (1985).
- [12] S.D. Peyerimhoff, R.J. Buenker, *Chem. Phys. Lett.*, **16**, 235 (1972).
- [13] R.J. Buenker, In *Stud. Phys. Theor. Chem.*, p. 17, Elsevier (1982).
- [14] R.J. Buenker, Paper presented at the Workshop on Quantum Chemistry and Molecular Physics, Wollongong, Australia (1980).
- [15] V.R. Saunders, J.H. van Lenthe, *Molec. Phys.*, **48**, 923 (1983).
- [16] R.J. Harrison, S. Zarrabian, *Chem. Phys. Lett.*, **158**, 393 (1989).
- [17] T.J. Lee, J.E. Rice, *Chem. Phys. Lett.*, **150**, 406 (1988).
- [18] G.E. Scuseria, A.C. Scheiner, T.J. Lee, J.E. Rice, H.F. Schaefer, *J. Chem. Phys.*, **86**, 2881 (1987).
- [19] A.P. Rendell, T.J. Lee, A. Komornicki, *Chem. Phys. Lett.*, **178**, 462 (1991).
- [20] T.J. Lee, A.P. Rendell, P.R. Taylor, *J. Phys. Chem.*, **94**, 5463 (1990).
- [21] T.J. Lee, J.E. Rice, A.P. Rendell, The TITAN set of electronic structure programs (1991).
- [22] B.O. Roos, P. Linse, P.E.M. Siegbahn, M.R.A. Blomberg, *Chem. Phys.*, **66**, 197 (1982).
- [23] H.J.J. van Dam, J.H. van Lenthe, P.J.A. Ruttink, *Int. J. Quant. Chem.*, **72**, 539 (1999).
- [24] B. Engels, V. Pleß, H.-U. Suter, Direct MRD-CI, University of Bonn, Bonn, Germany (1993).
- [25] S. Krebs, R.J. Buenker, *J. Chem. Phys.*, **103**, 5613 (1995).
- [26] C. Fuchs, V. Bonacic-Koutecky, J. Koutecky, *J. Chem. Phys.*, **98**, 3121 (1993).

- [27] L.S. Cederbaum, W. Domcke, *Adv. Chem. Phys.*, **36**, 205 (1977).
- [28] J. Schirmer, L.S. Cederbaum, *J. Phys. B*, **11**, 1889 (1978).
- [29] P. Pulay, *Chem. Phys. Lett.*, **73**, 393 (1980).
- [30] P. Pulay, *J. Comput. Chem.*, **3**, 556 (1982).
- [31] B. Jonsson, B.O. Roos, P.R. Taylor, P.E.M. Siegbahn, *J. Chem. Phys.*, **74**, 4566 (1981).
- [32] P.J. Knowles, G.J. Sexton, N.C. Handy, *Chem. Phys.*, **72**, 337 (1982).
- [33] P.J. Knowles, H.J. Werner, *Chem. Phys. Lett.*, **115**, 259 (1985).
- [34] N.C. Handy, H.F. Schaefer, *J. Chem. Phys.*, **81**, 5031 (1984).
- [35] J.E. Rice, R.D. Amos, N.C. Handy, T.J. Lee, H.F. Schaefer, *J. Chem. Phys.*, **85**, 963 (1986).
- [36] S. Bell, J.S. Crighton, *J. Chem. Phys.*, **80**, 2464 (1984).
- [37] C.J. Cerjan, W.H. Miller, *J. Chem. Phys.*, **75**, 2800 (1981).
- [38] J. Simons, P. Jorgensen, H. Taylor, J. Ozmment, *J. Phys. Chem.*, **87**, 2745 (1983).
- [39] J. Baker, *J. Comput. Chem.*, **7**, 385 (1986).
- [40] J.A. Pople, R. Krishnan, H.B. Schlegel, J.S. Binkley, *Int. J. Quant. Chem. Symp.*, **13**, 225 (1979).
- [41] Y. Osamura, Y. Yamaguchi, P. Saxe, M.A. Vincent, J.F. Gaw, H.F. Schaefer, *Chem. Phys.*, **72**, 131 (1982).
- [42] R.M. Stevens, R.M. Pitzer, W.N. Lipscomb, *J. Chem. Phys.*, **38**, 550 (1963).
- [43] R.D. Amos, In *Adv. Chem. Phys.*, p. 99, John Wiley & Sons (1987).
- [44] P. Durand, J.C. Berthelat, *Theor. Chim. Acta*, **38**, 283 (1975).
- [45] F. Weinhold, J.E. Carpenter, In *The Structure of Small Molecules and Ions*, R. Naaman, Z. Vager (Eds), p. 227, Plenum, New York (1988).
- [46] A.E. Reed, L.A. Curtiss, F. Weinhold, *Chem. Rev.*, **88**, 899 (1988).
- [47] Version 3.0 of the NBO program from Quantum Chemistry Program Exchange (No. 408) (1980).
- [48] A.J. Stone, *Chem. Phys. Lett.*, **83**, 233 (1981).
- [49] F.W. Biegler-Konig, R.F.W. Bader, T.H. Tang, *J. Comput. Chem.*, **3**, 317 (1982).
- [50] A.H. de Vries, P.T. van Duijnen, A.H. Juffer, J.A.C. Rullmann, J.P. Dijkman, H. Merenga, B.T. Thole, *J. Comput. Chem.*, **16**, 37 (1995).
- [51] A.H. de Vries, P.T. van Duijnen, A.H. Juffer, J.A.C. Rullmann, J.P. Dijkman, H. Merenga, B.T. Thole, *J. Comput. Chem.*, **16**, 1445 (1995).
- [52] P.T. van Duijnen, A.H. de Vries, *Int. J. Quant. Chem.*, **60**, 1111 (1996).
- [53] S. Faas, J.G. Snijders, J.H. van Lenthe, E. van Lenthe, E.J. Baerends, *Chem. Phys. Lett.*, **246**, 632 (1995).
- [54] P. Hohenberg, W. Kohn, *Phys. Rev. B*, **136**, 864 (1964).
- [55] B.I. Dunlap, J.W.D. Connolly, J.R. Sabin, *J. Chem. Phys.*, **71**, 4993 (1979).
- [56] B.I. Dunlap, J.W.D. Connolly, J.R. Sabin, *J. Chem. Phys.*, **71**, 3396 (1979).
- [57] H.J.J. van Dam, 2nd Derivatives of the Electronic Energy in Density Functional Theory, CLRC Daresbury Laboratory, Warrington, DL-TR-01-002 (2001).
- [58] B.G. Johnson, M.J. Fisch, *J. Chem. Phys.*, **100**, 7429 (1994).
- [59] P.M. Boerrigter, G. te Velde, E.J. Baerends, *Int. J. Quant. Chem.*, **33**, 87 (1988).
- [60] G. te Velde, E.J. Baerends, *J. Comput. Phys.*, **99**, 84 (1992).
- [61] M. Challacombe, *J. Chem. Phys.*, **113**, 10037 (2000).
- [62] A.D. Becke, *J. Chem. Phys.*, **88**, 2547 (1988).
- [63] F.L. Hirschfeld, *Theor. Chim. Acta (Berl.)*, **44**, 129 (1977).
- [64] R.E. Stratmann, G.E. Scuseria, M.J. Frisch, *Chem. Phys. Lett.*, **257**, 213 (1996).
- [65] C.W. Murray, N.C. Handy, G.J. Laming, *Molec. Phys.*, **78**, 997 (1993).
- [66] M.E. Mura, P.J. Knowles, *J. Chem. Phys.*, **104**, 9848 (1996).
- [67] V.I. Lebedev, D.N. Laikov, *Dokl. Math.*, **59**, 477 (1999).
- [68] <http://www.ccl.net/cca/software/SOURCES/FORTRAN/Lebedev-Laikov-Grids/>
- [69] L.A. Curtiss, K. Raghavachari, G.W. Trucks, J.A. Pople, *J. Chem. Phys.*, **94**, 7221 (1991).
- [70] Density Functional Repository (<http://www.cse.clrc.ac.uk/qcg/dft/>).
- [71] R. Fletcher, *Practical Methods of Optimization*, Wiley, Chichester (1981).
- [72] M. Douglas, N. Kroll, *Ann. Phys.*, **82**, 89 (1974).
- [73] B.A. Hess, *Phys. Rev. A*, **33**, 3742 (1986).
- [74] C. Chang, M. Pelissier, P. Durand, *Physica Scripta*, **34**, 394 (1986).
- [75] J.L. Heully, I. Lindgren, E. Lindroth, S. Lundqvist, A.M. Martensson-Pendrill, *J. Phys. B*, **19**, 2799 (1986).
- [76] E. van Lenthe, E.J. Baerends, J.G. Snijders, *J. Chem. Phys.*, **99**, 4597 (1993).
- [77] S. Faas, J.G. Snijders, J.H. Van Lenthe, *Prog. Theor. Chem. Phys.*, **2**, 251 (2000).
- [78] S. Faas, J.H. Van Lenthe, J.G. Snijders, *Molec. Phys.*, **98**, 1467 (2000).
- [79] J.H. van Lenthe, S. Faas, J.G. Snijders, *Chem. Phys. Lett.*, **328**, 107 (2000).
- [80] S. Faas, J.H. van Lenthe, A.C. Hennem, J.G. Snijders, *J. Chem. Phys.*, **113**, 4052 (2000).
- [81] J.H. van Lenthe, G.G. Balint-Kurti, *J. Chem. Phys.*, **78**, 5699 (1983).
- [82] J. Verbeek, J.H. Langenberg, C.P. Byrman, J.J. Engelberts, J.H. Van Lenthe, TURTLE an *Ab Initio* VB/VBSCF program (1988–2004).
- [83] J.H. van Lenthe, F. Dijkstra, R.W.A. Havenith, In *Valence Bond Theory*, D.L. Cooper (Ed.), pp. 79–116, Elsevier, Amsterdam (2002).
- [84] J. Verbeek, J.H. van Lenthe, *J. molec. Struct. (Theochem)*, **229**, 115 (1991).
- [85] P.H.M. Budzelaar, J.J. Engelberts, J.H. van Lenthe, *Organometallics*, **22**, 1562 (2003).
- [86] J.H. van Lenthe, J. Verbeek, P. Pulay, *Molec. Phys.*, **73**, 1159 (1991).
- [87] R. Gdanitz, R. Ahlrichs, *Chem. Phys. Lett.*, **143**, 413 (1988).
- [88] P.G. Szalay, R.J. Bartlett, *Chem. Phys. Lett.*, **214**, 481 (1993).
- [89] M. Hanrath, B. Engels, *Chem. Phys.*, **225**, 197 (1997).
- [90] D.L. Yeager, P. Jorgensen, *Chem. Phys. Lett.*, **65**, 77 (1979).
- [91] J.V. Ortiz, *J. Chem. Phys.*, **101**, 6743 (1994).
- [92] C. van Caillie, R.D. Amos, *Chem. Phys. Lett.*, **308**, 249 (1999).
- [93] A. Zewail, *Femtochemistry, Ultrafast Dynamics of the Chemical Bond*, vols. I and II, World Scientific, Singapore (1974).
- [94] M. Hartmann, J. Pittner, H.J.J. van Dam, V. Bonacic-Koutecky, *Eur. Phys. J. D*, **9**, 393 (1999).

- [95] P. Sherwood, A.H. de Vries, M.F. Guest, G. Schreckenbach, C.R.A. Catlow, S.A. French, A.A. Sokol, S.T. Bromley, W. Thiel, A.J. Turner, S. Billeter, F. Terstegen, S. Thiel, J. Kendrick, S.C. Rogers, J. Casci, M. Watson, F. King, E. Karlsen, M. Sjøvoll, A. Fahmi, A. Schäfer, C. Lennartz, *J. molec. Struct. (Theochem)*, **632**, 1 (2003).
- [96] ChemShell (<http://www.cse.clrc.ac.uk/qcg/chemshell>).
- [97] P. Sherwood, A.H. de Vries, S.J. Collins, S.P. Greatbanks, N.A. Burton, M.A. Vincent, I.H. Hillier, *Faraday Discuss.*, **79** (1997).
- [98] S.A. French, A.A. Sokol, S.T. Bromley, C.R.A. Catlow, P. Sherwood, *Top. Catal.*, **24**, 161 (2003).
- [99] B.R. Brooks, R.E. Bruccoleri, B.D. Olafson, D.J. States, S. Swaminathan, M. Karplus, *J. Comput. Chem.*, **4**, 187 (1983).
- [100] K.P. Eurenium, D.C. Chatfield, B.R. Brooks, M. Hodosek, *Int. J. Quant. Chem.*, **60**, 1189 (1996).
- [101] P.D. Lyne, M. Hodosek, M. Karplus, *J. Phys. Chem. A*, **103**, 3462 (1999).
- [102] D. Das, K.P. Eurenium, E.M. Billings, P. Sherwood, D.C. Chatfield, M. Hodosek, B.R. Brooks, *J. Chem. Phys.*, **117**, 10534 (2002).
- [103] H.L. Woodcock, M. Hodosek, P. Sherwood, Y.S. Lee, H.F. Schaefer, B.R. Brooks, *Theor. Chem. Acc.*, **109**, 140 (2003).
- [104] The CCP1 GUI (<ftp://ftp.dl.ac.uk/qcg/ccp1gui>).
- [105] See the e-CCP1 Wiki Site (<http://grids.ac.uk/twiki/bin/view/ECCP/ECCP1Activities>).
- [106] T.P. Straatsma, E. Aprà, T.L. Windus, E.J. Bylaska, W. de Jong, S. Hirata, M. Valiev, M.T. Hackler, L. Pollack, R.J. Harrison, M. Dupuis, D.M.A. Smith, J.V.T. Nieplocha, M. Krishnan, A.A. Auer, E. Brown, G. Cisneros, G.I. Fann, H. Fruchtl, J. Garza, K. Hirao, R. Kendall, J. Nichols, T. Tsemekhman, K. Wolinski, J. Anchell, D. Bernholdt, P. Borowski, T. Clark, D. Clerc, H. Dachsel, M. Deegan, K. Dyall, D. Elwood, E. Glendening, M. Gutowski, A. Hess, J. Jaffe, B. Johnson, J. Ju, R. Kobayashi, R. Kutteh, Z. Lin, R. Littlefield, X. Long, B. Meng, T. Nakajima, S. Niu, M. Rosing, G. Sandrone, M. Stave, H. Taylor, G. Thomas, J. van Lenthe, A. Wong, Z. Zhang, NWChem, A Computational Chemistry Package for Parallel Computers, Version 4.6, Pacific Northwest National Laboratory, Richland, WA, USA (2004).
- [107] R.A. Kendall, E. Apra, D.E. Bernholdt, E.J. Bylaska, M. Dupuis, G.I. Fann, R.J. Harrison, J.L. Ju, J.A. Nichols, J. Nieplocha, T.P. Straatsma, T.L. Windus, A.T. Wong, *Comput. Phys. Commun.*, **128**, 260 (2000).
- [108] D.E. Bernholdt, E. Apra, H.A. Fruchtl, M.F. Guest, R.J. Harrison, R.A. Kendall, R.A. Kutteh, X. Long, J.B. Nicholas, J.A. Nichols, H.L. Taylor, A.T. Wong, *Int. J. Quant. Chem.*, 475 (1995).
- [109] J. Nieplocha, R.J. Harrison, R.J. Littlefield. Global Arrays; a portable shared memory programming model for distributed memory computers, paper presented at Supercomputing'94, pp. 340–349 (1994).
- [110] J. Nieplocha, R.J. Harrison, R.J. Littlefield, *J. Supercomput.*, **10**, 169 (1996).
- [111] J. Nieplocha, R.J. Harrison, *J. Supercomput.*, **11**, 119 (1997).
- [112] G.I. Fann, R.J. Littlefield. Parallel inverse iteration with reorthogonalisation, paper presented at the 6th SIAM Conference on Parallel Processing for Scientific Computing, SIAM, pp. 409–413.
- [113] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, D. Sorensen. *LAPACK's User's Guide*, Society for Industrial and Applied Mathematics (1992).
- [114] W. Kohn, L.J. Sham, *Phys. Rev. A*, **140**, 1133 (1965).
- [115] N. Godbout, D.R. Salahub, J. Andzelm, E. Wimmer, *Can. J. Chem.*, **70**, 560 (1992).
- [116] G.D. Fletcher, A.P. Rendell, P. Sherwood, *Molec. Phys.*, **91**, 431 (1997).
- [117] F. Dijkstra, J.H. van Lenthe, *J. Comput. Chem.*, **22**, 665 (2001).
- [118] F. De Proft, P. Von Rague Schleyer, J.H. Van Lenthe, F. Stahl, P. Geerlings, *Chemistry—A European Journal*, **8**, 3402 (2002).
- [119] P.C. Hiberty, S. Humbel, C.P. Byrman, J.H. van Lenthe, *J. Chem. Phys.*, **101**, 5969 (1994).
- [120] B.O. Roos, *Chem. Phys. Lett.*, **15**, 153 (1972).
- [121] P.E.M. Siegbahn, *J. Chem. Phys.*, **72**, 1647 (1980).
- [122] P.E.M. Siegbahn, *Int. J. Quant. Chem.*, **18**, 1229 (1980).
- [123] R.W.A. Havenith. PhD thesis, Utrecht University (2000).
- [124] H. Lischka, H. Dachsel, R. Shepard, R.J. Harrison, *ACS Symp. Ser.*, **592**, 75 (1995).
- [125] ScaLAPACK (<http://www.netlib.org/scalapack/>).
- [126] J. Choi, J. Demmel, I. Dhillon, J. Dongarra, S. Ostrouchov, A. Petitet, K. Staney, D. Walker, R.C. Whaley. LAPACK Working Note 95: ScaLAPACK: A portable linear algebra library for distributed memory computers—design issues and performance, University of Tennessee (1995).
- [127] LAPACK (<http://www.netlib.org/lapack/>).
- [128] BLACS (<http://www.netlib.org/blacs/>).
- [129] V. Sharma, S. Sharma, K.H.Z. Bentrup, J.D. McKinney, D.G. Russell, W.R. Jacobs, J.C. Sacchettini, *Nature Struct. Biol.*, **7**, 663 (2000).
- [130] A.E. Reed, R.B. Weinstock, F. Weinhold, *J. Chem. Phys.*, **83**, 735 (1985).
- [131] H.M. Vinkers, M.R. De Jonge, F.F.D. Daeyaert, J. Heeres, L.M.H. Koymans, J.H. van Lenthe, P.J. Lewi, H. Timmerman, K. van Aken, P.A. Janssen, *J. Med. Chem.*, **46**, 2765 (2003).
- [132] H.M. Vinkers, M.R. de Jonge, F.D. Daeyaert, J. Heeres, L.M.H. Koymans, J.H. van Lenthe, P.J. Lewi, H. Timmerman, P.A.J. Janssen, *J. Comput. Aided Molec. Des.*, **17**, 567 (2003).
- [133] W. Smith, T. Forester, *J. Molec. Graph.*, **14**, 136 (1996).
- [134] W. Smith, C.W. Yong, P.M. Rodger, *Molec. Sim.*, **28**, 385 (2002).
- [135] A.H. de Vries, P. Sherwood, S.J. Collins, A.M. Rigby, M. Rigutto, G.J. Kramer, *J. Phys. Chem. B*, **103**, 6133 (1999).
- [136] M.H. Palmer, I.C. Walker, C.C. Ballard, M.F. Guest, *Chem. Phys.*, **192**, 111 (1995).
- [137] M.H. Palmer, I.C. Walker, M.F. Guest, *Chem. Phys.*, **238**, 179 (1998).
- [138] M.H. Palmer, I.C. Walker, *Chem. Phys.*, **133**, 113 (1989).
- [139] I.C. Walker, M.H. Palmer, A. Hopkirk, *Chem. Phys.*, **141**, 365 (1990).
- [140] M.H. Palmer, I.C. Walker, M.F. Guest, A. Hopkirk, *Chem. Phys.*, **147**, 19 (1990).
- [141] I.C. Walker, M.H. Palmer, *Chem. Phys.*, **153**, 169 (1991).

- [142] I.C. Walker, M.H. Palmer, C.C. Ballard, *Chem. Phys.*, **167**, 61 (1992).
- [143] M.H. Palmer, I.C. Walker, M.F. Guest, M.R.F. Siggel, *Chem. Phys.*, **201**, 381 (1995).
- [144] M.H. Palmer, H. McNab, D. Reed, A. Pollacchi, I.C. Walker, M.F. Guest, M.R.F. Siggel, *Chem. Phys.*, **214**, 191 (1997).
- [145] M.H. Palmer, H. McNab, I.C. Walker, M.F. Guest, M. MacDonald, M.R.F. Siggel, *Chem. Phys.*, **228**, 39 (1998).
- [146] M.H. Palmer, M.F. Guest, *Chem. Phys.*, **291**, 287 (2003).
- [147] I.C. Walker, M.H. Palmer, J. Delwiche, S.V. Hoffmann, P.L. Vieora, N.J. Mason, M.F. Guest, M.J. Hubin-Franskin, J. Heinesch, A. Giuliani, *Chem. Phys.*, **297**, 289 (2004).