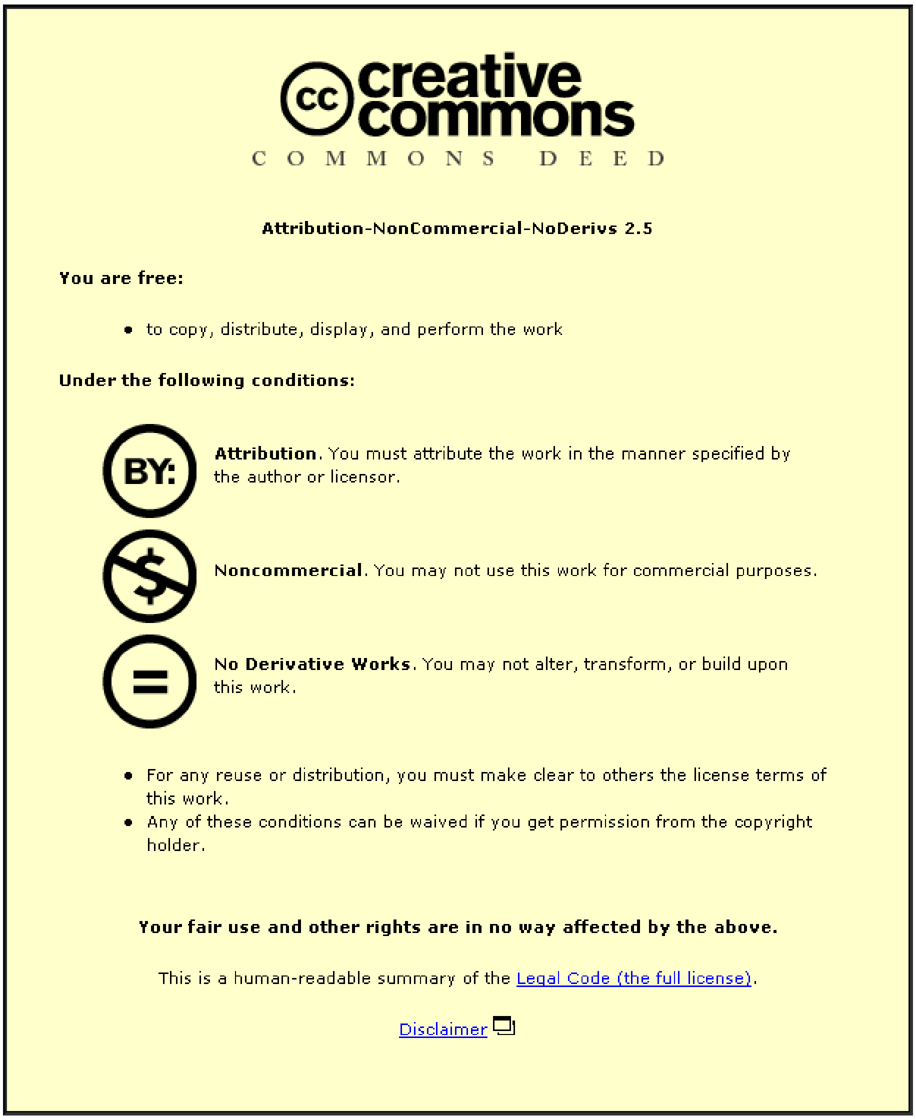
View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Loughborough University Institutional Repository

Loughborough University

This item was submitted to Loughborough's Institutional Repository (https://dspace.lboro.ac.uk/) by the author and is made available under the following Creative Commons Licence conditions.



creative commons

COMMONS DEED

Attribution-NonCommercial-NoDerivs 2.5

You are free:

- to copy, distribute, display, and perform the work

Under the following conditions:

Attribution. You must attribute the work in the manner specified by the author or licensor.

Noncommercial. You may not use this work for commercial purposes.

No Derivative Works. You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.

This is a human-readable summary of the Legal Code (the full license).

Disclaimer

For the full text of this licence, please go to: http://creativecommons.org/licenses/by-nc-nd/2.5/

Plato: A localised orbital based density functional theory code

S. D. Kenny[a], A. P. Horsfield[b]

[a]*Department of Mathematical Sciences, Loughborough University, Loughborough, Leicestershire LE11 3TU, United Kingdom*
[b]*Department of Materials, Imperial College London, South Kensington Campus, London SW7 2AZ, United Kingdom*

Abstract

The Plato package allows both orthogonal and non-orthogonal tight-binding as well as density functional theory (DFT) calculations to be performed within a single framework. The package also provides extensive tools for analysing the results of simulations as well as a number of tools for creating input files. The code is based upon the ideas first discussed in [1] with extensions to allow high-quality DFT calculations to be performed. DFT calculations can utilise either the local density approximation or the generalised gradient approximation. Basis sets from minimal basis through to ones containing multiple radial functions per angular momenta and polarisation functions can be used. Illustrations of how the package has been employed are given along with instructions for its utilisation.

PACS: 71.15.-m, 71.15.Ap, 71.15.Mb

Key words: Density functional theory, Localised orbitals

PROGRAM SUMMARY

Manuscript Title: Plato: A localised orbital based density functional theory code
Authors: S.D. Kenny and A.P. Horsfield
Program Title: Plato
Journal Reference:
Catalogue identifier:
Licensing provisions: None
Programming language: C and PERL
Computer: Apple Macintosh, PC, Unix machines
Operating system: Unix, Linux and Mac OS X
RAM: Up to 2 Gbytes per processor

Email addresses: `S.D.Kenny@lboro.ac.uk` (S. D. Kenny), `a.horsfield@imperial.ac.uk` (A. P. Horsfield)

Number of processors used: Up to 256
Keywords: Density functional theory, Localised orbitals
PACS: 71.15.-m, 71.15.Ap, 71.15.Mb
Classification: 7.3 Electronic Structure
External routines/libraries: LAPACK, BLAS and optionally ScaLAPACK, BLACS, PBLAS, FFTW

Nature of problem: Density functional theory study of electronic structure and total energies of molecules, crystals and surfaces.

Solution method: Localised orbital based density functional theory.

Restrictions: Tight-binding and density functional theory only, no exact exchange.

Unusual features: Both atom centred and uniform meshes available. Can deal with arbitrary angular momenta for orbitals, whilst still retaining Slater-Koster tables for accuracy.

Additional comments:

Running time: Test cases will run in a few minutes, large calculations may run for several days.

LONG WRITE-UP

1. Overview of Plato

Accurate calculations of the energies and structures of molecules and solids are widespread, and often form an essential part of current research activity. The popularity of Density Functional Theory (DFT) calculations is at least in part a result of the availability of powerful software packages that are straightforward to use. There is a broad spectrum of methods in use, with the main difference between programs being the decisions made about how to represent the wavefunctions. The two most popular basis sets used to expand the wavefunctions are planewaves and gaussians. Plato makes another, increasingly popular, choice that of numeric atom-centred orbitals.

Plato is a general-purpose total energy electronic structure code that represents single particle electronic wave functions as an expansion in terms of atomic type orbitals. The orbitals are not expanded in analytic functions, but rather are finite in range and represented by a radial function that is specified at points on a mesh multiplied by spherical harmonics. Plato has been designed for a pragmatic approach to performing atomistic calculations: the computationally lightest method that is sufficiently accurate should be used. The choice of atomic orbitals makes available a range of approximate techniques (see below for the available methods).

The choice of a numerical representation for the orbitals means that they can rigorously go to zero at a finite radius, which eliminates the need to evaluate a large number of very small integrals leading to a marked improved in efficiency. However, the price that has to be paid is that integrals must be evaluated numerically rather than analytically. To improve efficiency those integrals that can be tabulated are evaluated prior to a simulation, and the required values obtained by interpolation. However, tabulation does not extend to two-electron integrals, therefore exact exchange is expensive to compute, and rendering this method unsuitable for wave function based methods (such as Møller-Plesset perturbation theory or Configuration Interaction). The Hartree potential is affordable because many integrals can be summed together, and then evaluated numerically in one batch.

Two techniques are implemented for performing those integrals that cannot be tabulated. The first is to use the standard uniform regular mesh on which the integral weight at every mesh point is equal. This mesh also permits Poisson's equation to be solved by means of Fast Fourier Transforms. The second uses a sum of atom centred meshes. This is particularly suitable for all-electron calculations as the mesh concentrates points near the nucleus where both the orbitals and the potential vary most rapidly. Electrostatics become more complex on this mesh (especially for periodic systems), but Poisson's equation can be solved efficiently by making an atom centred expansion of the charge density [2].

The code has been built both for three-dimensional periodic systems and for molecular or cluster calculations (no periodic boundaries). There are three methods provided for determining electronic structure: orthogonal tight binding, non-orthogonal tight binding [3] and LCAO Density Functional theory. Charge self-consistency and collinear spin are available for all models. Necessarily simplified schemes (a monopole on each atomic site) are, however, used for tight binding. The program allows for static relaxation (including optimisation of the cell coordinates), the search for minimum energy pathways (both generalized drag method and nudged elastic band [4, 5]) and molecular dynamics. Experimental versions of Ehrenfest dynamics and open boundary calculations have been implemented [6], but are not ready for general release.

Vibrational frequencies are computed by means of numerical differentiation of the forces (performed by means of an external script). Additional utility programs allow for analysis of the charge density (including the determination of STM images) and electronic states. The main program has been parallelized. Furthermore, computational steering has been introduced to permit the adjustment of input parameters while the program is running. This is controlled by a graphical user interface.

Plato has been strongly influenced by a number of other codes, but most especially the following three. The original code was an implementation of a modified version of method of Sankey and Niklewski [1], whose FIREBALL code was groundbreaking, demonstrating the usefulness of numerical orbitals that are truncated exactly, and the efficiency possible through the use of integral tables. The extension of these ideas in the original SIESTA [7] code to include a fully self consistent implementation of Density Functional Theory demonstrated the possibility of using the Fast Fourier Transform mesh to perform numerical integrals with atomic orbitals, as well as the possibility of achieving linear scaling. An important commercial code is DMOL [2] (now DMOL^3). It also made use of numeric atomic orbitals for its basis set, but uses an atom centred mesh to perform the integrals as this accommodates the cusps in the electron-nuclear interaction near the nucleus. It showed that not only could the integrals be performed efficiently on this mesh without resorting to pseudopotentials, but also that the Hartree potential can be evaluated efficiently as well.

Plato differs from the SIESTA code [8] due to the fact that all integrals in SIESTA are now performed on a mesh (normally uniform although there is a atom-centred version), whilst Plato uses a combination of a mesh (either uniform or atom-centred) and integral tables. Similarly the DMOL [2] and FHI-AIMS [9] codes perform all of their integrals on an integration mesh, in these cases an atom-centred one. The use of integral tables adds complexity to Plato and requires the generation of the tables but means that these integrals can be performed very accurately relatively cheaply. Until recently it was very difficult to deal with f-orbitals and beyond using an integral table approach but this has been addressed by the emergence of recursive techniques for calculating Slater-Koster contributions [10, 11].

We now discuss briefly the methods implemented within Plato, and some of the results have been obtained with it.

2. Models Available

As stated above, there are three electronic structure models implemented in Plato: tight binding (both orthogonal and non-orthogonal) and Density Functional Theory (both self-consistent and non-self-consistent). We now present some details of the implementations.

2.1. Tight Binding

Tight binding can produce quantitatively useful results [12]. It is implemented in Plato both as an orthogonal model and with the overlap matrix included. The non-orthogonal version is strongly influenced by the model of Thomas Frauenheim and his colleagues [13]. The two-centre integrals (hopping integrals and overlap) are stored in tables just as for density functional calculations.

Monopole charge self consistency has been included in a manner very similar to that of Elstner *et al.* [14], though the charge was represented by a gaussian rather than an exponential as this makes the algebra much simpler. Spin has also been included using a simplified version of the model of Frauenheim *et al.* [3]; separate on-site energies are introduced for up and down spin and they are shifted relative to each other by an amount proportional to the net spin on the site.

2.2. Density Functional Theory

This is the highest level of theory implemented in Plato. The local density approximation and the generalised gradient approximation of Perdew, Burke and Ernzerhof [15] have been implemented. These are available for collinear spin polarised systems. Charge mixing follows the scheme used in Vasp [16]. The full functionality is currently only available for the uniform integral mesh.

The calculation of the total energy of the system can be rewritten as a generalised eigenvalue problem, due to the use of non-orthogonal basis functions. The calculation of the matrix elements for the generalised eigenvalue problem requires numerous integrals to be performed. Due to the use of finite ranged orbitals employed in this methodology the integrals that need evaluating can be written in a form similar to a tight-binding formualism. We want to evaluate the total energy which can be written in the following form

$$E_{tot} = \sum_i f_i \epsilon_i - \frac{1}{2}\int V_{Ha}[\rho;\mathbf{r}]\rho(\mathbf{r})d\mathbf{r} + E_{xc}[\rho] - \int V_{xc}[\rho;\mathbf{r}]\rho(\mathbf{r})d\mathbf{r} + \frac{1}{2}\sum_{I\neq J}\frac{Z_I Z_J}{|\mathbf{R}_I - \mathbf{R}_J|}.$$

where Z_I is the charge on atom I and the ϵ_i's are eigenvalues of the Hamiltonian

$$\hat{H} = \hat{T} + V_{e-Ion}(\mathbf{r},\mathbf{r}') + V_{Ha}[\rho;\mathbf{r}] + V_{xc}[\rho;\mathbf{r}]$$

and f_i is the occupancy of the state i. The Hartree and exchange and correlation potentials are given by:

$$
\begin{aligned}
V_{Ha}[\rho;\mathbf{r}] &= \int \frac{\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}d\mathbf{r}' \\
V_{xc}[\rho;\mathbf{r}] &= \frac{\delta E_{xc}[\rho]}{\delta\rho(\mathbf{r})}.
\end{aligned}
$$

The eigenstates, $|\psi_i\rangle$, are found by expanding in terms of atomic like basis functions, $|\phi_{I\alpha}\rangle$, centred on the atoms giving

$$|\psi_i\rangle = \sum_{I\alpha} C_{I\alpha}^{(i)}|\phi_{I\alpha}\rangle$$

where the I represents the atom number and α labels the orbital on an atom. This allows us to write

$$\hat{H}|\psi_i\rangle = \epsilon_i|\psi_i\rangle$$

as

$$\sum_{I\alpha} C_{I\alpha}^{(i)}\hat{H}|\phi_{I\alpha}\rangle = \epsilon_i \sum_{I\alpha} C_{I\alpha}^{(i)}|\phi_{I\alpha}\rangle$$

If we then define the Hamiltonian and overlap matrix elements as

$$
\begin{aligned}
H_{I\alpha,J\beta} &= \int \phi_{I\alpha}(\mathbf{r})\hat{T}\phi_{J\beta}(\mathbf{r})d\mathbf{r} + \int\int \phi_{I\alpha}(\mathbf{r})V_{e-Ion}(\mathbf{r},\mathbf{r}')\phi_{J\beta}(\mathbf{r}')d\mathbf{r}d\mathbf{r}' \\
&+ \int \phi_{I\alpha}(\mathbf{r})V_{Ha}[\rho;\mathbf{r}]\phi_{J\beta}(\mathbf{r})d\mathbf{r} + \int \phi_{I\alpha}(\mathbf{r})V_{xc}[\rho;\mathbf{r}]\phi_{J\beta}(\mathbf{r})d\mathbf{r} \\
S_{I\alpha,J\beta} &= \int \phi_{I\alpha}(\mathbf{r})\phi_{J\beta}(\mathbf{r})d\mathbf{r}
\end{aligned}
$$

then the problem of finding the eigenstates reduces to solving the generalised eigenvalue problem:

$$\sum_{J\beta} H_{I\alpha,J\beta}C_{J\beta}^{(i)} = \epsilon_i \sum_{J\beta} S_{I\alpha,J\beta}C_{J\beta}^{(i)}.$$

The eigenvalues are found by direct diagonalisation using LAPACK, ARPACK [17] or ScaLAPACK [18, 19] routines.

If the electron-ion interaction is represented with a seperable non-local pseudopotential (see section 2.5) then the Hamiltonian integrals can be rewritten as

$$
\begin{aligned}
H_{I\alpha,J\beta} = \int \phi_{I\alpha}(\mathbf{r}) \left(\hat{T} + \left[\sum_K \int \frac{Q_K(\mathbf{r})}{|\mathbf{r}-\mathbf{r}'|}d\mathbf{r}' \right] + \int \frac{\Delta\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}d\mathbf{r}' + V_{xc} \right) \phi_{J\beta}(\mathbf{r})d\mathbf{r} \\
+ \int \phi_{I\alpha}(\mathbf{r}) \sum_{Klm} v_{Klm}(\mathbf{r})v_{Klm}(\mathbf{r}')\phi_{J\beta}(\mathbf{r}')d\mathbf{r}d\mathbf{r}'.
\end{aligned}
$$

where

$$Q_I = \rho_I^0(\mathbf{r}) - \rho_{I,loc}(\mathbf{r})$$

with $\rho_{I,loc}(\mathbf{r})$ being an effective charge density that generates the local part of the pseudopotential, $V_{I,loc}$, on atom I and ρ_I^0 is the spherical electron density that comes from solving for atom I in a confining potential that forces the wavefunction to zero at distance r_c. Thus Q_i combines the local part of the pseudopotential and a compensating spherical electron distribution. This produces a zero electric field outside the range of the electron density, as long as r_c is larger than the cutoff distance for the non-local pseudopotential. The resulting potential, $\int \frac{Q_I(\mathbf{r})}{|\mathbf{r}-\mathbf{r}'|} d\mathbf{r}'$, is labelled as the neutral atom potential, $V_{I,NA}(\mathbf{r})$, after Sankey [1] and is short-ranged, having the same range as the orbital cutoff. The Hartree potential term can be split up into two parts, the spherical compressed atomic density and the change in density on bonding, due to the fact that the potential is linear in the charge density.

Due to the use of finite range basis functions the following integrals can all be tabulated and the Slater-Koster rules [20] can be used to evaluate arbitrary integrals. The overlap integrals

$$\int \phi_{I\alpha}(\mathbf{r})\phi_{J\beta}(\mathbf{r})d\mathbf{r},$$

the kinetic energy integrals

$$\int \phi_{I\alpha}(\mathbf{r})\hat{T}\phi_{J\beta}(\mathbf{r})d\mathbf{r},$$

the one-centre neutral atom integrals

$$\int \phi_{I\alpha}(\mathbf{r})V_{I,NA}(\mathbf{r})\phi_{I\alpha}(\mathbf{r})d\mathbf{r},$$

the two-centre neutral atom integrals

$$\int \phi_{I\alpha}(\mathbf{r})(V_{I,NA}(\mathbf{r}) + V_{J,NA}(\mathbf{r}))\phi_{J\beta}(\mathbf{r})d\mathbf{r},$$

the non-local pseudopotential terms

$$\int \phi_{I\alpha}(\mathbf{r})\sum_{Klm} v_{Klm}(\mathbf{r})v_{Klm}(\mathbf{r}')\phi_{J\beta}(\mathbf{r}')d\mathbf{r}d\mathbf{r}'$$

which can be split and evaluated as the multiple of two two-centre terms

$$\int \phi_{I\alpha}(\mathbf{r})v_{Klm}(\mathbf{r})d\mathbf{r}.$$

This leaves us with three terms that cannot be evaluated by tabulating them and using the Slater-Koster rules. These are the three-centre neutral atom terms

$$\int \phi_{I\alpha}(\mathbf{r})V_{K,NA}(\mathbf{r})\phi_{J\beta}(\mathbf{r})d\mathbf{r},$$

the excess Hartree term

$$\int\int \phi_{I\alpha}(\mathbf{r})\frac{\Delta\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}\phi_{J\beta}(\mathbf{r})d\mathbf{r}d\mathbf{r}',$$

and the exhange-correlation terms

$$\int \phi_{I\alpha}(\mathbf{r})V_{xc}[\rho]\phi_{J\beta}(\mathbf{r})d\mathbf{r}.$$

These terms are all evaluated on an integration mesh. Two different integration meshes are available which are described in more detail in section 2.6.

There is also an electrostatic contribution to the double counting, part of which includes the atomic charge densities. The part from the atomic density can be combined with the electrostatic core-core interaction to produce a short ranged pair potential, which is also tabulated. The remaining terms are evaluated by performing the integrals on the mesh.

2.3. Integral Tables

The use of integral tables for one and two centre integrals achieves two things: it allows improved efficiency for Density Functional Theory calculations; it allows tight binding and DFT to be treated in a unified way, with DFT being treated as a set of corrections to tight binding. As previously stated the two-site integrals have the rotational properties of the Slater-Koster table. For systems described by s, p and d orbitals only the σ, π and δ integrals need be evaluated. The tables for these integrals are widely available in the literature, see for example Sutton [21]. These rules and the corresponding rules for the derivatives of these terms are explicitly coded, which makes the evaluation of these terms very fast.

Tables of rules for integrals involving f-orbitals and above grow increasingly complex, with many of the published tables containing errors due to this. The direct implementation of such tables and particularly the derivatives, which are required for the evaluation of the forces, become incredibly difficult to implement. Recently work has been performed on formalisms for the automatic generation of Slater-Koster tables [10, 11]. We have implemented these approaches in Plato and can use them to apply the Slater-Koster rules to arbitrary angular momenta.

2.4. Basis sets

Strictly localised atomic like basis functions are used in the density functional theory code. These are produced by solving for the atom in a potential that goes to infinity for $r > r_c$, where r_c is a user defined cutoff radius. This produces basis functions that go strictly to zero at the cutoff radius, but with a non-zero first and second derivative. In order that the first two derivatives vanish at r_c the functions can be multiplied by $1 - \exp(-[r - r_c]^2/(2\sigma^2))$, where σ is chosen such that the derivatives vanish in a smooth fashion.

Basis sets are constructed by using functions from both the atom and positively charged ions solved in this fashion [2, 22]. Polarisation functions are also included in the basis set and these play an important role in giving the basis set the required variational freedom. Basis functions that have the same angular form are orthogonalised, as this is found to improve the stability of the calculations.

Basis sets consisting of two sets of the occupied functions and a set of polarisation functions, referred to as double numeric plus polarisation (DNP), are the most commonly used. So for example for silicon this would consist of two sets of s and two sets of p functions plus a set of d functions. Triple numeric plus double polarisation (TNDP) are used in cases where high levels of accuracy are required. For silicon this would consist of three sets of s and three sets of p functions plus two sets of d functions. Optimised basis functions can be produced using the technique of Kenny *et al.* [22].

2.5. Pseudopotentials

Plato is capable of dealing with either local pseudopotentials or separable non-local ones. The local part of the pseudopotential is dealt with by adding it to the Hartree potential due to the electrons for the neutral atom, which produces a short-ranged potential. Because the non-local pseudopotentials are separable all the associated integrals can be evaluated using one and two centre tables. This is due to the fact that the three centre terms can be rewritten as the product of two two-centre terms:

$$V_{NL,I\alpha J\beta} = \sum_{Klm} V_{NL,Klm} \int \phi_{I\alpha}(\mathbf{r}) P_{NL,Klm}(\mathbf{r})\, d\mathbf{r} \int P^{*}_{NL,Klm}(\mathbf{r}) \phi_{J\beta}(\mathbf{r})\, d\mathbf{r}.$$

This also simplifies the evaluation of the derivatives. The integrals are evaluated prior to the simulation and are tabulated, with the Slater-Koster rules being used to evaluate them in the code. This makes the implementation of the pseudopotentials very efficient and also means that it scales linearly with the number of atoms.

Three different separable non-local pseudo potential schemes have been implemented in Plato. These are the pseudopotentials of Goedecker, Teter and Hutter [23], the pseudopotentials of Hartwigsen, Goedecker and Hutter [24] and those of Troullier and Martin [25]. For the Troullier and Martin pseudopotentials the input format is the identical to that used by the Abinit code [26]. Pseudopotentials of this form can be obtained using the OPIUM [27] and FHI98pp [28, 29] codes. The code also contains an implementation of the nonlinear core corrections [30] of Louie, Froyen and Cohen. This is implemented for both integral meshes and includes the corrections for the force and stress terms.

2.6. Integral Meshes

Having both uniform and the atom centred meshes available is an advantage because for some problems the uniform mesh is far more efficient (for example, periodic systems that can be described by soft pseudo potentials, such as silicon), whereas for other problems (such as organic molecules described using an all electron approach) the atom centred mesh is the only option.

The use of uniform meshes for performing integrals and solving Poisson's equation using fast Fourier transforms is well-documented (see for example Ordejon *et al.* [7]), and so will not be discussed further here except for one point. Plato has a very simple way of handling the electrostatics for a cluster (no periodic boundary conditions) using Fast Fourier Transforms. The problem is that the coulomb potential has a long tail that extends into neighbouring cells. This can be removed simply by truncating the tail, but making sure that it is long enough to reach all parts of the atoms within the cell. This means in practice cutting off the potential at one half of the linear cell dimension, and making sure that the maximum dimension of the molecule or cluster is only one half that of the cell. The Hartree potential can then be obtained from the density in the usual manner, except that the Fourier transform of the coulomb interaction is replaced by that for the truncated interaction: that is $1/q^2$ is replaced by $(1-\cos(qr_c))/q^2$ where r_c is the radius at which the coulomb potential is set to zero. Note that this removes the divergence at $q=0$ so that there is no need to introduce a neutralising background for charged systems.

While atom centred meshes are widely used in quantum chemistry calculations for performing exchange-correlation integrals, they are less well-known in the context of condensed matter calculations. In Plato they perform two functions: they provide an efficient way of performing integrals; they provide a way of solving Poisson's equation. Integrals are performed in the following way. For the sake of illustration we will consider a two centre integral involving two atomic orbitals and one nuclear charge:

$$V_{I\alpha J\beta} = \int \frac{\phi_{I\alpha}(\mathbf{r})\phi_{J\beta}(\mathbf{r})}{|\mathbf{r}-\mathbf{R}_I|}\,d\mathbf{r},$$

where $\mathbf{R}_I$ is the position of atom I. The first step is to introduce a partition function defined by

$$P_I(\mathbf{r}) = \frac{W_I(|\mathbf{r}-\mathbf{R}_I|)}{\sum_J W_J(|\mathbf{r}-\mathbf{R}_J|)}$$

and satisfies $\sum_I P_I(\mathbf{r}) = 1$, where $W_I(r)$ is a spherical weighting function that is strongly peaked at the origin. A functional form that leads to reliable integrals for both energies and forces is $W_I(r) = V_{NA,I}(r)^2$. The integral can now be written as

$$\begin{aligned} V_{I\alpha J\beta} &= \sum_K \int \phi_{I\alpha}(\mathbf{r})\frac{P_K(\mathbf{r})}{|\mathbf{r}-\mathbf{R}_I|}\phi_{J\beta}(\mathbf{r})\,d\mathbf{r} \\ &= \sum_K \int_0^{2\pi} \mathrm{d}\phi \int_0^{\pi} \sin\theta \mathrm{d}\theta \int_0^{\infty} \phi_{I\alpha}(\mathbf{r})\frac{r^2 P_K(\mathbf{r})}{|\mathbf{r}-\mathbf{R}_I|}\phi_{J\beta}(\mathbf{r})\,dr \end{aligned}$$

where we have made explicit use of spherical coordinates in the final integral. This version handles the singularity in the coulomb potential near the nucleus. If $K = I$ then near $r = 0$ the weighting factor r^2 suppresses the $1/r$ divergences. When $I \neq K$ the partition function P_K does the suppressing. To solve Poisson's equation we do something similar. We have to perform the following integral $V_{Ha}(\mathbf{r}) = \int \rho(\mathbf{r}')/|\mathbf{r}-\mathbf{r}'|\, d\mathbf{r}'$. We introduce another partition function. The idea is the same as for the integrals except that the weighting function is now the atomic charge density, $W_I(r) = \rho_I(r)$. This allows us to write the integral as

$$V_{Ha}(\mathbf{r}) = \sum_I \int \frac{P_I(\mathbf{r})\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}\, d\mathbf{r}' \tag{1}$$

We now use the identity

$$\frac{1}{|\mathbf{r}-\mathbf{r}'|} = \sum_{lm} \frac{4\pi}{(2l+1)} \frac{r_<^l}{r_>^{l+1}} \mathrm{Y}_{lm}(\hat{r})\mathrm{Y}^*_{lm}(\hat{r}') \tag{2}$$

where $\hat{r}$ represent the angular part of $\mathbf{r}$ and $r_{<,>}$ are the lesser (greater) of r and r'. Substituting Eq. 2 into Eq. 1 we get

$$V_{Ha}(\mathbf{r}) = \sum_{Ilm} \mathrm{Y}_{lm}(\widehat{r-R_I}) \frac{4\pi}{(2l+1)} \int_0^\infty \rho_{Ilm}(x) \frac{x_<^l}{x_>^{l+1}} x^2\, \mathrm{d}x \tag{3}$$

where $x_{<,>}$ are the lesser (greater) of x and $|\vec{r}-\vec{R}_I|$ and $\rho_{Ilm}(x) = \int \mathrm{Y}^*_{lm}(\hat{x})P_I(\vec{x})\rho(\vec{x}+\vec{R}_I)\, \mathrm{d}\hat{x}$. So we project the total density into atomic-like densities, carry out angular integrals to produce the radial functions ρ_{Ilm}, and then perform further radial integrals to produce the multipole fields given by Eq. 3. For periodic systems, we subtract gaussians from $\rho_{Ilm}$to make the multipole fields short ranged (and so treatable in real space), and then add back the long ranged fields by treating the gaussians in reciprocal space.

3. Functionality

Plato has many of the standard features that appear in other total energy electronic structure codes. Atomic forces and the stress tensor are computed (one exception is that the stress tensor has not yet been fully implemented for atom centred integral meshes). Methods for computing these quantities have been described in detail elsewhere [31, 32, 33]. The relaxation of atoms can be performed using the usual algorithms (steep descent, conjugate gradient, variable metric). When both the cell coordinates and atomic coordinates are relaxed simultaneously, the atomic coordinates are relaxed first, then the cell is allowed to relax, followed by the atoms again *etc.* This nested approach was found to greatly improve the stability of the relaxation. Minimum energy pathways can be found using either nudged elastic band or a generalized drag method.

The vast majority of the effort involved in the evaluation of the matrix elements and the calculation of the energy, the forces and the stress is parallelised.

In addition to this two different approaches are employed in the parallelisation of the calculation of the eigenvalues and eigenvectors. The first approach is for cases where a number of k-points are employed, in this case the work is split by asking each processor to calculate the eigenvalues and eigenvectors corresponding to one or more of the k-points. This methodology is very efficient as the diagonalisations that need to be performed on each processor are independent and the required communication between the processors is minimal. This leads to a code that scales very well with processor number but is inherently limited by the number of k-points used in the simulation. The second methodology uses the ScaLAPACK parallel diagonalisers [18, 19] so that the evaluation of the eigenvalues and eigenvectors corresponding to a single k-point is split across the processors. This is less efficient, as communication takes place during the diagonalisation, but gives the ability to scale to much larger processor numbers.

One of the suite of analysis programs within the Plato package allows for the production of simulated STM images using the methodology of Tersoff and Hamann [34]. This can be used in conjuction with a purpose built 3D viewer to study the STM images whilst simultaneously visualising the atoms in the system. This allows for the electronic structure effects to be visualised and separated from the pure topology picture.

One further analysis tool that is available enables the charge density to be analysed using the ideas from Bader's atoms in molecules [35]. The tool for use with Plato uses the algorithm described in the work by Sanville *et al.* [36]. In order to analyse the data core charge density files need to be created if a pseudopotential is being employed. The resulting analysis is capable of giving charges and spins on atoms as well as analysing the bond critical points to identify the nature of the bonds between atoms.

The Plato code can be steered during simulations through an implementation of the RealityGrid computational steering library [37, 38]. This allows the user to alter parameters such as the convergence criteria, the electron temperature and the number of bands for which eigenvalues are evaluated during the course of a simulation. This is an excellent tool for the optimisation of the time taken to perform simulations. The steering interface also provides users with checkpointing facilities, these allow the code to be stopped gracefully at relatively short notice.

New functionality is now being introduced for electron transport calculations. Two approaches are being taken: a static version that is a variant of the non-equilibrium Green's functions called Hairy Probes [39]; the solution of the time-dependent problem within the Ehrenfest approximation. These are still at an experimental stage.

4. Applications

Plato has been used to study a wide variety of systems ranging from molecules through metals, semiconductors and insulators. It has been used extensively to study atoms and molecules on surfaces including the study of a low coverage of cobalt on a silicon surface [40] and the adhesion of Ag and Au atoms and small

Ag and Au clusters to graphite surfaces [41, 42, 43]. Plato was used in the first ab-initio studies of the bonding of C_{60} molecules to the Si (100) surface that allowed for atomic relaxation [44, 45]. Subsequently it has been used to study the adhesion of C_{82} molecules and endohedral C_{60} molecules and the interaction between C_{60} molecules on the Si (100) surface [46, 47, 48]. In these studies TNDP basis sets were employed, which massively reduced the basis set superposition error to the point where such corrections were unnecessary. Plato has also been used to study the adhesion of thiol molecules to Au surfaces [49] and the orientation of C_{70} molecules in nanotubes [50] both of which have important roles in the field of nanotechnology.

It has also been used to investigate defects in the bulk including work on semiconductor materials to model amorphous In-Se alloys [51] and Cu in Si [52] as well as metals in the modelling of self-interstitial atom defects in bcc transition metals [53]. Defects in ceramics have also been studied through an investigation of typical defects created by radiation damage in MgO [54, 55] to enable a comparison with empirical potential based methodologies typically used in this field. Due to the vast amount of data that it is possible to obtain through the use of the methodology it has also been employed to derive neural network based empirical potentials which have been fitted to either a tight-binding model [56] or DFT [57].

5. Installing and Testing

The program comes in a compressed tar file, upon uncompressing and de-taring all the associated files will be contained in the `Plato` directory. Inside of this you will find the following directories:

`Data`	this is where data for executing the codes is stored
`doc`	some documentation for the code including some worked examples
`Examples`	example input files
`Scripts`	scripts for running processes such as minimum energy path finding
`src`	the source files for the codes that make up the package
`System`	the files that control compiler options and libraries
`Test`	a suite of test input files and corresponding outputs.

In order to compile the main code, `tb2`, and the associated codes for building the integral tables and analysing the results the `makefile.h` and `makefile-tb2.h` files in the `System` directory must be edited. These need to indicate your choice of C compiler, the linker that you wish to use as well as the options that are to be passed to both of these. They both also need to indicate which BLAS and LAPACK libraries that you would like to link against and whether the LAPACK libraries contain the LAPACK version 3 routines. The `makefile-tb2.h` additionally needs details of whether you wish to compile the parallel version of the code and if so what MPI libraries you want to link against. It also contains options to link with the ScaLAPACK and FFTW libraries to further parallelise the code and improve the scaling. Examples of these files can be found in the `machine` directory contained in this directory, a full list of compilation options and instructions for installing and running the code can be found in the Install file in the Plato directory.

Testing of the compiled code can be performed by running the test input files, which can be found in the Test directory. All these input files can be executed and comparisons carried out with the correct outputs by executing the Scripts\TEST.pl script. Example input files for both the generation of basis sets and the execution of DFT calculations can be found in the Examples directory. Documentation on how to build basis sets and execute some example calculations can be found in the doc directory, with many of these processes illustrated with substantial examples.

All of the input files for the programs that build the integral tables and the integral tables themselves are stored in the Data directory structure. The PseudoV directory contains the data files for the specification of the pseudopotentials used in the calculations. The Fraunheim and SCF_LCAO directories are where integral tables generated for Frauheim non-orthogonal tight-binding and DFT calculations respectively are placed.

Finally the Script directory contains scripts for running more complicated calculations within the code. These include scripts for running and analysing binding energy curves and for the calculation of the matrix of second derivatives and the subsequent evaluation of the vibrational modes. More advanced relaxation techniques are also found here such as the simulated annealing technique and a script for executing the nudged elastic band method for the calculation of transition barriers.

Acknowledgments

We would like to thank B.D. Ringham, D.J. King and E. Sanville for contributions to the code.

References

[1] Sankey, O. F. and Niklewski, D. J., Phys. Rev. B **40** (1989) 3979.

[2] Delley, B., J. Chem. Phys. **92** (1990) 508.

[3] Frauenheim, T. et al., Phys. Stat. Sol. (b) **217** (2000) 41.

[4] Henkelman, G. and Jonsson, H., J Chem Phys **113** (2000) 9978.

[5] Henkelman, G., Uberuaga, B., and Jonsson, H., J Chem Phys **113** (2000) 9901.

[6] Horsfield, A. P. et al., Computational Materials Science **44** (2008) 16.

[7] Ordejon, P., Artacho, E., and Soler, J. M., Phys. Rev. B **53** (1996) R10441.

[8] Artacho, E. et al., J Phys-Condens Mat **20** (2008) 064208.

[9] Blum, V. et al., Comp. Phys. Comm. **in press** (2009).

[10] Podolskiy, A. and Vogl, P., Phys. Rev. B **69** (2004) 233101.

[11] Elena, A. and Meister, M., Phys. Rev. B **72** (2005) 165107.

[12] Goringe, C. M., Bowler, D. R., and Hernández, E., Rep. Prog. Phys. **60** (1997) 1447.

[13] Porezag, D., Frauenheim, T., Köhler, T., Seifert, G., and Kaschner, R., Phys. Rev. B **51** (1995) 12947.

[14] Elstner, M. et al., Phys. Rev. B **58** (1998) 7260.

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

[16] Kresse, G. and Furthmiiller, J., Computational Materials Science **6** (1996) 15.

[17] Arpack home page, http://www.caam.rice.edu/software/ARPACK/index.html.

[18] Scalapack home page, http://www.netlib.org/scalapack/scalapack_home.html.

[19] Bush, I., Sunderland, A., and Pringle, G. J., Scalable eigensolvers on hpcx: Case studies, Technical Report 510, HPCx consortium, 2005, http://www.hpcx.ac.uk/research/publications/HPCxTR510.pdf.

[20] Slater, J. and Koster, G., Phys Rev **94** (1954) 1498.

[21] Sutton, A. P., *Electronic Structure of Materials*, OUP, 1993.

[22] Kenny, S. D., Horsfield, A. P., and Fujitani, H., Phys. Rev. B **62** (2000) 4899.

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

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

[25] Troullier, N. and Martins, J. L., Phys. Rev. B **43** (1991) 1993.

[26] Gonze, X. et al., Comp. Mat. Sci. **25** (2002) 478.

[27] http://opium.sourceforge.net/, 2008.

[28] Fuchs, M. and Scheffler, M., Comput Phys Commun **119** (1999) 67.

[29] Fuchs, M. and Scheffler, M., http://www.fhi-berlin.mpg.de/th/fhi98md/fhi98PP/, 2003.

[30] Louie, S. G., Froyen, S., and Cohen, M. L., Phys. Rev. B **26** (1982) 1738.

[31] Fiebelman, P. J., Phys. Rev. B **44** (1991) 3916.

[32] Balbas, L. C., Martins, J. L., and Soler, J. M., Phys. Rev. B **64** (2001) 165110.

[33] Soler, J. M. et al., J. Phys.: Condens. Matter **14** (2002) 2745.

[34] Tersoff, J. and Hamann, D. R., Phys. Rev. B **31** (1985) 805.

[35] Bader, R., *Atoms in Molecules: A Quantum Theory*, Oxford University Press, 1990.

[36] Sanville, E., Kenny, S. D., Smith, R., and Henkelman, G., J Comput Chem **28** (2007) 899.

[37] Pickles, S. M. et al., The RealityGrid Computational Steering API, http://www.sve.man.ac.uk/Research/AtoZ/RealityGrid/Steering/ReG_steering_api.pdf, 2004.

[38] Pickles, S. M., Haines, R., Pinning, R. L., and Porter, A. R., Practical tools for computational steering, in *Proceedings of the UK e-Science All Hands Meeting*, 2004.

[39] McEniry, E. J. et al., J. Phys.: Condensed Matter **19** (2007) 196201.

[40] Horsfield, A. P., Kenny, S. D., and Fujitani, H., Phys. Rev. B **64** (2001) 245332.

[41] Wang, G. M., BelBruno, J. J., Kenny, S. D., and Smith, R., Surf. Sci. **541** (2003) 91.

[42] Wang, G. M., BelBruno, J. J., Kenny, S. D., and Smith, R., Phys. Rev. B **69** (2004) 195412.

[43] Wang, G. M., BelBruno, J. J., Kenny, S. D., and Smith, R., Surf. Sci. **576** (2005) 107.

[44] Godwin, P. D., Kenny, S. D., and Smith, R., Surf. Sci. **490** (2001) 409.

[45] Godwin, P. D., Kenny, S. D., and Smith, R., Surf. Sci. **529** (2003) 237.

[46] Frangou, P. C., Kenny, S. D., and Sanville, E., Surface Science **602** (2008) 1532.

[47] King, D. J., Kenny, S. D., and Sanville, E., Surface Science **603** (2009) 178.

[48] King, D. J., Frangou, P. C., and Kenny, S. D., Surface Science **603** (2009) 676.

[49] Wang, G. M., Sandberg, W. C., and Kenny, S. D., Nanotechnology **17** (2006) 4819.

[50] Khlobystov, A. N. et al., Appl. Phys. Letts. **84** (2004) 792.

[51] Kohary, K., M., B. V., Pettifor, D. G., and Nguyen-Manh, D., Phys. Rev. B **71** (2005) 184203.

[52] Ehlers, F. J. H., Horsfield, A. P., and Bowler, D. R., Phys. Rev. B **73** (2006) 165207.

[53] Nguyen-Manh, D., Horsfield, A. P., and Dudarev, S. L., Phys. Rev. B **73** (2006) 020101(R).

[54] Gilbert, C. A., Smith, R., and Kenny, S. D., Nucl Instrum Meth B **255** (2007) 166.

[55] Gilbert, C. A., Kenny, S. D., Smith, R., and Sanville, E., Phys. Rev. B **76** (2007) 184103.

[56] Bholoa, A., Kenny, S. D., and Smith, R., Nucl Instrum Meth B **255** (2007) 1.

[57] Sanville, E., Bholoa, A., Smith, R., and Kenny, S. D., J Phys-Condens Mat **20** (2008) 285219.