

Parsec 1.1 User's Guide ¹²

Prof. J. R. Chelikowsky's Research Group³

September 29, 2005

¹<http://www.ices.utexas.edu/parsec/>

²this text written by M. Tiago, mtiago@ices.utexas.edu

³<http://www.ices.utexas.edu/ccm>

Contents

1	What it is	5
2	Getting the code	7
3	Compiling Parsec	9
3.1	Porting to New Platforms	9
4	Running Parsec	11
4.1	Basic input	11
4.2	The parsec.in file	11
4.3	Basic output	12
5	Input options	13
5.1	The real-space grid	13
5.2	Diagonalization parameters	14
5.3	Self-consistency and mixing parameters	15
5.4	Global atom and pseudopotential parameters	15
5.5	Parameters specific to each chemical element	16
5.6	Electronic parameters	17
5.7	Structural relaxation	19
5.8	Molecular dynamics	20
5.9	Polarizability	21
5.10	Flags for additional input/output	21
6	Post-processing tools	23
6.1	PVOX	23
6.2	Other tools	23
7	Problems, unfinished work, etc.	25
7.1	Troubleshooting	25
7.2	Future implementations	25
	Index	25

Chapter 1

What it is

Parsec is a computer code that solves the Kohn-Sham equations by expressing electron wave-functions directly in real space, without use of explicit basis set. The name stands for "Pseudopotential Algorithm for Real-Space Electronic Calculations". It handles periodic boundary conditions (on all three directions) and confined-system boundary conditions. In the latter, the charge density and wave-functions are required to vanish outside a spherical shell enclosing the electronic system. A finite-difference approach is used for the calculation of spatial derivatives. Norm-conserving pseudopotentials are used to describe the interaction between valence electrons and ionic cores (core electrons+nuclei). Owing to the sparsity of the Hamiltonian matrix, the Kohn-Sham equations are solved by direct diagonalization, with the use of extremely efficient sparse-matrix eigensolvers. Besides standard DFT calculations, parsec is able to perform structural relaxation, simulated annealing, Langevin molecular dynamics, polarizability calculations (confined-system boundary conditions only). It can also be used to provide input for TDLDA (time-dependent density functional theory, in adiabatic local-density approximation), GW approximation, and solution of the Bethe-Salpeter equation for optical excitations.

The development of parsec started in the early 1990's with the work of J.R. Chelikowsky, N. Troullier, and Y. Saad (Phys. Rev. Lett **72**, 1240 ; Phys. Rev. B **50**, 11355). Later, many people have contributed with additional features and performance improvement:

- A. Stathopolous : mathematical subroutines;
- H. Kim : core correction, multipole expansion;
- I. Vasiliev : Polarizability, asymptotically-corrected functionals;
- M. Jain and L. Kronik : GGA, generalized Broyden mixer;
- K. Wu : improvements in iterative diagonalizer and Hartree solver;
- R. Burdick : major rewrite and improvement of multipole expansion;

- M. Alemany and M. Jain : periodic boundary conditions, parallel implementation;
- M. Alemany and E. Lorin : ARPACK solver;
- A. Makmal: dynamical memory allocation;
- M. Tiago : parallelization, symmetry operations, interface with GW and BSE, this user's guide.

Chapter 2

Getting the code

Public versions of the code can be downloaded from Prof. Chelikowsky's group page¹. There is also a development version, with implementations that are being tested for stability and performance. If you are interested in contributing to development, contact the developers.

Parsec is distributed under the General Public License (GPL). In a nutshell, these are the conditions implied in using it:

- It should be used for non-commercial purposes only.
- The authors give no warranty regarding its reliability or usefulness to your particular application.
- The authors offer no technical support.

¹<http://www.ices.utexas.edu/parsec/>

Chapter 3

Compiling Parsec

At the beginning of the `Makefile` file, edit the include line with the correct machine-dependent compilation file. Existing compilation files are inside subdirectory `config`. Some of the existing files are:

- `make.ibmsp3` `make.ibmsp3_mpi` : sp.msi.umn.edu, with/without MPI support.
- `make.intel` : plain i686 machine running linux.

This code has been ported and tested on a variety of different platforms: IBM SP3 and SP4 (AIX compiler), SGI (intel and f90 compilers), Intel/Linux (g95, intel compilers).

Usually, the executable is named `parsec.ser` (no MPI support) or `parsec.mpi` (with MPI support).

3.1 Porting to New Platforms

Besides creating a new `make.*` file for the machine you want to use, these are the files you may have to modify in order to port it to other platforms: `myflush.F`, `timing.F`, `cftw.F`

Parsec uses the following external libraries:

- ARPACK¹, diagonalization of sparse matrices.
- FFTW², versions 2.x or 3.x, used for 3-dimensional FFT of real functions.
- LAPACK/BLAS³, linear algebra. Alternatively, support for libraries ESSL (IBM) and MKL⁴ (Intel Math Kernel Library) is also available.
- MPI : message passing interface, for parallel machines.

¹<http://www.caam.rice.edu/software/ARPACK/>

²<http://www.fftw.org>

³<http://www.netlib.org/lapack/>

⁴<http://www.intel.com/cd/software/products/asm-na/eng/index.htm>

Chapter 4

Running Parsec

4.1 Basic input

- `parsec.in` : input file with user-provided parameters.
- `*_POTRE.DAT` : pseudopotential files, in format consistent with Prof. J.L. Martins' pseudopotential generator¹. The distributors of Parsec have a version of Prof. Martins' code with minor but important modifications, so that files in the proper format are created. For the moment, only norm-conserving pseudopotentials are implemented. For back-compatibility, files in "old format" are also accepted, see below `Old_Pseudopotential_Format` flag. Pseudopotential files used in Parsec are compatible with the "Siesta"² DFT program, but the opposite is not true: pseudo-wavefunctions are appended to `*_POTRE.DAT` files, but the same does not necessarily happen to formatted pseudopotential files used in Siesta.

4.2 The `parsec.in` file

The user-provided input uses the electronic structure data format (ESDF), implemented as a module written by Chris J. Pickard. Comments can be added on any line, and are preceded by any of the symbols "# ; !". Keywords and blockwords are case insensitive. Characters ". : =" and multiple spaces are ignored. Most parameters have default values. All parameters with true/false value have default value false. Parameter values can be written anywhere in the file, with the exception that information pertinent to the same chemical element should be given in consecutive lines (sometimes, data from different elements can be mixed, but this should be avoided for the sake of clarity). Keywords are always followed by their corresponding value on the same line. Input values

¹<http://bohr.inesc-mn.pt/~jlm/pseudo.html>

²<http://www.uam.es/siesta/>

with intrinsic physical unit can be given in different units, with name following the value. Default unit is used if none is specified. For example, the line `grid_spacing 0.5 ang` sets a grid spacing of 0.5 angstroms. Blocks have the syntax:

```
begin Blockword
.
.
.
end Blockword
```

WARNING: parsec does not watch out for misprints in `parsec.in`. Lines with unrecognized keywords or blockwords are usually ignored.

4.3 Basic output

- `parsec.out` : unified output file.
- `out.*` : standard output. It contains some warnings and error messages.
- `wfn.dat` : contains eigenvalues, density function, effective potential, wavefunctions and other output data, in binary (unformatted) form. See `Olddat_Style`.

Chapter 5

Input options

5.1 The real-space grid

The following parameters define the region of space where the Kohn-Sham equations will be solved, and the appropriate boundary conditions. In parallel environment, grid points are distributed among processors. Symmetry operations are used to reduce the sampled region to an irreducible wedge.

Periodic_System (*boolean*), default value = **false**

This flag enables the use of periodic boundary conditions on all three directions. If set true, the input grid spacing may be increased so as to make it commensurable with the given size of periodic cell. Also, atomic coordinates are used as given, with no initial repositioning around the origin of real-space grid. If atoms are found to be outside the cell, they are replaced by their periodic image inside the cell.

Cell_Shape (*block*), default value = **no default**

Length of periodic cell along the x, y, z coordinates. Relevant only in periodic systems.

Lattice_Vector_Scale (*real*), default value = **1** , default unit = *bohr* (Bohr radii)

Constant that multiplies the length of all three unit lattice vectors. This is useful for volumetric expansions or contractions in periodic systems. Relevant only in periodic systems.

Boundary_Sphere_Radius (*real*), default value = **no default** , default unit = *bohr* (Bohr radii)

Radius of enclosing spherical shell. Boundary conditions in non-periodic systems are such that all wave-functions vanish beyond a spherical shell centered in the origin and with given radius. The Hartree potential is also set to vanish on the shell (internally, this is done by solving Poisson's

equation with a compensating charge outside the shell). Calculation stops if any atom is found outside the shell. The amount of extra space that should be given between the shell and any inner atom is strongly system-dependent and should be carefully tested for convergence.

Grid_Spacing (*real*), default value = `no default` , default unit = *bohr* (Bohr radii)

Distance between neighbor points in the real-space grid. This is a critical parameter in the calculation and it should be carefully tested for convergence. The grid is assumed to be regular and orthogonal.

Expansion_Order (*integer*), default value = 12

Order used in the finite-difference expansion of gradients and Laplacians. It corresponds to twice the number of neighbor points used on each side and for each direction. It must be an even number.

Olddat_Style (*boolean*), default value = `false`

Compatibility flag. Removes the grid shift and ignores symmetry operations, so that the resulting grid is equal to the one used in older versions of this code. Output wave-functions and density are written to file `old.dat` instead of `wfn.dat`. Notice that these two files have different format!

5.2 Diagonalization parameters

Many DFT codes solve the Kohn-Sham equations by minimizing the total energy functional. Parsec uses a different approach: it performs direct diagonalization of the Kohn-Sham equations. Expressed in real space, the Hamiltonian is a sparse matrix, and efficient algorithms can be used to obtain the low-energy eigenstates with minimal memory usage and parallel communication.

Eigensolver (*word*), default value = `arpack`

Implemented eigensolvers are: `arpack` (implicitly restarted Lanczos/Arnoldi) and `diagla` (block preconditioned Lanczos). The last solver uses significantly less memory, but it occasionally fails to converge in periodic systems.

Diag_Tolerance (*real*), default value = `1.d-4`

Tolerance in the residual norm of eigen-pairs.

Initial_Diag_Tolerance (*real*), default value = `100.0`

Tolerance in the residual norm used for the trial diagonalization. This can be a fairly large number, since the trial diagonalization does not need to be very accurate.

Subspace_Buffer_Size (*integer*), default value = 1

Number of additional eigenvalues to be computed for each representation. Must be some positive value, so that the full set of eigenvalues can be reconstructed from the sum of representations without missing levels.

5.3 Self-consistency and mixing parameters

Max_Iter (*integer*), default value = `integer`

Maximum number of iterations in the self-consistent loop.

Convergence_Criterion (*real*), default value = `2.d-4` , default unit = *Ry* (rydbergs)

The density function is considered converged when the self-consistent residual error (SRE) falls below this value. The SRE is an integral of the square of the difference between the last two self-consistent potentials, weighted by electron density and taken squared root. This parameter should not be chosen less than the typical diagonalization accuracy.

Mixing_Method (*word*), default value = `Anderson`

Defines the scheme used in the mixing of old and new potentials. Implemented schemes are `Broyden` and `Anderson` (which were proved to be equivalent by V. Eyert, 1996). In practice, there is no significant difference between these two schemes. Broyden mixing uses disk for input/output of internal data.

Mixing_Param (*real*), default value = `0.30`

Choice for the diagonal Jacobian used in the mixing of potentials. Small values reduce the amount of mixing and result in slower but more controlled convergence. For improved performance, the user may try reducing this parameter in cases where the energy landscape is too corrugated (e.g.: magnetic systems, or systems with many degrees of freedom).

Memory_Param (*integer*), default value = `2`

Number of previous SCF steps used in mixing. Ideal values range from 4 to 6. Larger values may actually worsen convergence because it includes steps far from the real solution.

5.4 Global atom and pseudopotential parameters

Atom_Types_Num (*integer*), default value = `no default`

Number of different chemical elements present in the system.

Coordinate_Unit (*word*), default value = **Cartesian_Bohr**

By default, positions of atoms are given in cartesian coordinates and units of Bohr radius. Other implemented options are:

- **Cartesian_Ang** atomic coordinates given in angstroms, cartesian.
- **Lattice_Vectors** atomic coordinates given in units of lattice vectors. Available only with periodic boundary conditions.

Old_Pseudopotential_Format (*boolean*), default value = **false**

By default, pseudopotential files have atomic parameters (cut-off radii, etc.) included. For back-compatibility, this flag enables the reading of old-format files, which do not have atomic parameters included. Notice that the naming convention of old-format files is ***_POTR.DAT** instead of ***_POTRE.DAT**. The use of mixed formats is not allowed: parsec reads all files in new format or all files in old format.

Old_Interpolation_Format (*boolean*), default value = **false**

This is a compatibility flag. Normally, the local component of pseudopotentials V at some arbitrary point r is evaluated by linear interpolation of the function r^*V . Due to the Coulomb-like behavior of V far away from the atom, this interpolation is exact beyond the cut-off radius. This flag reverts to the old method: linear interpolation of V itself. It should be used for debugging purposes only.

5.5 Parameters specific to each chemical element

These parameters should be given in sequence in `parsec.in`, and they refer to a particular chemical element in the system.

Atom_Type (*word*), default value = **no default**

Chemical symbol of the current element. THIS IS CASE SENSITIVE!

Potential_Num (*integer*), default value = **no default**

Number of available angular components. Relevant only if old-format pseudopotentials are used.

Core_Cutoff_Radius (*real*), default value = **no default** , default unit = *bohr* (Bohr radii)

Maximum cut-off radius used in the generation of pseudopotential. Relevant only if old-format pseudopotentials are used.

Electron_Per_Orbital (*block*), default value = **no default**

Number of electrons per atomic orbital used in the generation of pseudopotential. Relevant only if old-format pseudopotentials are used.

Local_Component (*integer*), default value = **no default**

Choice of angular component to be considered local in the Kleinman-Bylander transformation. No ghost-state check is done in parsec. Therefore, check if the chosen component does not have ghost states.

Move_Flag (*integer*), default value = 0

Type of movement to be applied to atoms. Relevant only if structural relaxation is performed. Value 0 makes all atoms movable in all directions. Value n positive makes the first n atoms movable. A negative value makes selected atoms movable (the selection is made inside block **Atom_Coord**).

Atom_Coord (*block*), default value = **no default**

Coordinates of atoms belonging to the current chemical species. See **Coordinate_Unit**. If structural relaxation is done with selected atoms (**Move_Flag** \neq 0), each triplet of coordinates must be followed by an integer: 0 if this atom is fixed, 1 if it is movable.

Alpha_Filter (*real*), default value = 0.0

Cut-off parameter in Fourier filtering of pseudopotentials, in reciprocal space. If value is non-positive, pseudopotentials are not filtered. Fourier filtering is implemented according to Briggs *et al.* Phys. Rev. B **54**, 14362 (1996), but no second filtering in real space has been implemented so far.

Beta1_Filter (*real*), default value = 0.0

Cut-off parameter in Fourier filtering of pseudopotentials, in reciprocal space. Referenced only if **alpha_filter** is positive.

Core_Filter (*real*), default value = equal to **Alpha_Filter**

Cut-off parameter in Fourier filtering of non-linear core density. If non-positive, then core density is not filtered.

5.6 Electronic parameters

Restart_Run (*boolean*), default value = **false**

By default, the initial charge density in a parsec run is built from the superposition of atomic charge densities. If you want instead to start from a previous run, choose value true. For confined systems, having **restart_run = true** prevents the initial centering of atoms. If this flag is used but an adequate file **wfn.dat** is not found, the calculation is interrupted.

States_Num (*integer*), default value = **no default**

Number of energy eigenvalues to be computed. In spin-polarized calculations, this is the number of eigenvalues per spin orientation.

Net_Charges (*real*), default value = 0.0 , default unit = e (electron charge)

Net electric charge in the system, in units of electron charge. It can also be used in periodic systems. In that case, a background charge is used and the divergent component of the Hartree potential is ignored.

Fermi_Temp (*real*), default value = 80.0 , default unit = K (kelvins)

Fermi temperature used to smear out the Fermi energy. If a negative value is given, then the occupancy of Kohn-Sham orbitals is read from file `occup.in` and kept fixed throughout the calculation.

Correlation_Type (*word*), default value = CA

Choice of exchange-Correlation functional to be used. Available choices and their variants are:

- CA, PZ : Ceperley-Alder (LDA), with spin-polarized option. D.M. Ceperley and B.J. Alder, Phys. Rev. Lett. **45**, 566 (1980); J.P. Perdew and A. Zunger, Phys. Rev. B **23**, 5048 (1981).
- XA: Slater's x-alpha (LDA). J.C. Slater, Phys. Rev. **81** 385 (1951).
- WI: Wigner's interpolation (LDA). E. Wigner, Phys. Rev. **46**, 1002 (1934).
- HL: Hedin-Lundqvist (LDA). L. Hedin and B.I. Lundqvist, J. Phys. C **4**, 2064 (1971).
- LB: Ceperley-Alder with Leeuwen-Baerends correction (asymptotically corrected). R. van Leeuwen and E. J. Baerends, Phys. Rev. A **49**, 2421 (1994).
- CS: Ceperley-Alder with Casida-Salahub correction (asymptotically corrected). M.E. Casida and D.R. Salahub, J. Chem. Phys. **113**, 8918 (2000).
- PW, PW91: Perdew-Wang (GGA), with spin-polarized option. J.P. Perdew and Y. Wang, Phys. Rev. B **45**, 13244 (1992).
- PB, PBE: Perdew-Burke-Ernzerhof (GGA), with spin-polarized option. J.P. Perdew, K. Burke, and M. Ernzerhof, Phys. Rev. Lett. **77**, 3865 (1996).

Ion_Energy_Diff (*real*), default value = 0.0, default unit = Ry (rydbergs)

Energy difference used in the construction of the Casida-Salahub functional. Relevant only if that functional is used.

Spin_Polarization: (*boolean*), default value = false

Enables spin-polarized densities and functional.

5.7 Structural relaxation

Minimization (*word*), default value = **none**

Choice of method for structural relaxation. Available options are: **none** (no relaxation), **simple** (steepest-descent algorithm), **BFGS** (Broyden-Fletcher-Goldfarb-Shanno, the best), and **manual** (manual movement). Steepest-descent should be used as last resort. With manual movement, the sequence of atomic coordinates must be provided by the user in file **manual.dat** and it is used "as is". The last choice is useful for debugging. Symmetry operations are ignored if manual movement is performed

Movement_Num (*integer*), default value = 500

Maximum number of iterations during structural relaxation.

Force_Min (*real*), default value = 1.5d-3 , default unit = *Ry/bohr* (rydbergs/Bohr radii)

Maximum magnitude of the force on any atom in the relaxed structure. This should not be set less than about 0.01 *Ry/a.u.*, which is the usual accuracy in the calculation of forces.

Max_Step (*real*), default value = 0.05 (steepest descent), -1 (BFGS), default unit = *bohr* (Bohr radii)

If steepest-descent minimization is performed, this parameter sets the maximum displacement of atoms between two steps in relaxation. With BFGS, this defines the maximum displacement of atoms along each Cartesian component from their original position (if negative, no constraint in the movement of atoms is imposed). Ignored in all other circumstances.

Min_Step (*real*), default value = 1.d-4 , default unit = *bohr* (Bohr radii)

Minimum displacement of atoms between two steps in steepest-descent relaxation. Relevant only if steepest descent is performed.

BFGS_Number_Corr (*integer*), default value = 7

Number of variable metric corrections used to defined the limited-memory matrix. Relevant only if BFGS relaxation is performed.

Relax_Restart (*boolean*), default value = **false**

Flag for the restart of a relaxation run. If true, the user must provide information about previous run in file **relax_restart.dat**. No need to input file **wfn.dat**. If this option is used, the atomic coordinates in **parsec.in** are replaced with the ones in **relax_restart.dat**. Appropriate warnings are written to output. If the restart file is not found, this flag is ignored and no information from previous relaxations is used.

5.8 Molecular dynamics

Molecular_Dynamics (*boolean*), default value = **false**

Enables molecular dynamics. This feature can be used both for simulated annealing and for direct simulation of hot systems. Both canonical and micro-canonical ensembles are implemented. Symmetry operations are ignored if molecular dynamics is being performed.

Cooling_Method (*word*), default value = **stair**

Type of cooling (i.e.: change of ensemble temperature) used during molecular dynamics. Possible choices are: **stair** (stair-case), **log** (logarithmic cooling), and **linear** (linear cooling). As the molecular dynamics simulation proceeds, the ensemble temperature changes according to the cooling method and to the choices of initial and final temperatures. At each step, the actual temperature is calculated from equipartition theorem and the kinetic energy (difference between the total energy and potential energy). In the first steps, the actual and ensemble temperatures may be very different, particularly if there are few atoms, when thermal fluctuations are enormous.

T_init (*real*), default value = **500.0** , default unit = *K* (kelvins)

Initial ensemble temperature in the system.

T_final (*real*), default value = **500.0** , default unit = *K* (kelvins)

Final ensemble temperature.

T_Step (*real*), default value = **150.0** , default unit = *K* (kelvins)

Step temperature used in the cooling.

Time_Step (*real*), default value = **150.0** , default unit = *ps* (10^{-12} seconds)

Time interval between two steps in molecular dynamics. The dynamical equations of motion are numerically integrated using a modified Verlet algorithm.

Friction_Coefficient (*real*), default value = **0.0**

Viscosity parameter in molecular dynamics, in atomic units (inverse time). If less than $1.d-6$ *a.u.*, then a micro-canonical ensemble is used: total energy is assumed constant, and the atoms start moving with a velocity distribution consistent with the initial temperature. If more, then a canonical ensemble is used and true Langevin molecular dynamics is performed.

Step_num (*integer*), default value = **250**

Number of steps in molecular dynamics. A successful simulation ends only after reaching the requested number of steps.

Restart_mode (*boolean*), default value = **false**

Enables the restart of an aborted molecular dynamics simulation. With this feature, the last two sets of positions, velocities and accelerations stored in input file `mdinit.dat` are used. The number of steps in that file is included in the current run as well.

5.9 Polarizability

Polarizability (*boolean*), default value = **false**

Enables calculation of polarizability using a finite-field method. Available only with non-periodic boundary conditions. An uniform electric field is applied successively along the +x,-x,+y,-y,+z,-z directions, and dipole moments are calculated and collected and printed in output file `polar.dat`. From dipole moments, the diagonal part of polarizability tensor is obtained. Symmetry operations are ignored if this flag is used.

Electric_Field (*real*), default value = 0.001, *default unit = ry/bohr/e* (rydbergs/Bohr radii/electron charge)

Magnitude of external electric field to be applied.

5.10 Flags for additional input/output

Skip_force (*boolean*), default value = **false**

Suppresses calculation of forces. Since forces are easily obtained, this flag is useless.

Output_All_States (*boolean*), default value = **false**

Forces output of all wave-functions up to requested number of states to file `wfn.dat`, instead of only the occupied ones (default).

Save_Intermediate_Charge_Density (*boolean*), default value = **false**

Causes wave-functions to be written to output after each self-consistent iteration, instead of only after self-consistency is reached. Previous wave-functions are over-written. This is useful for debugging purposes or if diagonalization is too demanding.

Output_Eigenvalues (*boolean*), default value = **false**

Saves eigenvalues to file `eigen.dat` after self-consistency in density function is achieved. This is useful for debugging purposes.

Save_Intermediate_Eigenvalues (*boolean*), default value = **false**

Saves eigenvalues after each SCF iteration, instead of after the last one. Useful for debugging purposes.

`Output_Level` (*integer*), default value = 0

Output flag for additional info in `parsec.out`. With default value, only the basic output is written. Positive values cause additional information to be printed out:

- > 0 : print out local/non-local components of force
- > 1 : print out all symmetry data and additional arpack output
- > 2 : print out more BFGS output
- > 3 : print out all BFGS output
- > 4 : print out all g-stars (warning: `_lots_` of data)
- > 5 : print out local potentials (warning: `_lots_` of data)

Chapter 6

Post-processing tools

6.1 PVOX

PVOX (parsec visualization toolbox) is a set of matlab/python scripts, with graphical user interface. It was developed in 2005 by Michael Frasca, Lee Ballard, and Nick Voshell. It reads parsec output directly and extracts the density of states, makes plots of electron charge density, self-consistent potential, and selected wave-functions. It has been tested with Matlab 7.0¹ and Python 2.4.1.

6.2 Other tools

There are additional tools that read parsec output and perform specific tasks. Some of them are: `angproj`, for calculation of angle-projected density of states; `plotdx`, interface with Data Explorer² for visualization of charge density, wave-functions, etc.

¹<http://www.mathworks.com/products/matlab/>

²<http://www.opendx.org>

Chapter 7

Problems, unfinished work, etc.

7.1 Troubleshooting

1. Diagonal occasionally shows bad performance if periodic boundary conditions are used. If you get error messages related to the eigensolver, try Arpack.
2. Missing symmetry operations: only symmetry operations belonging to the cubic system are tested for existence. That is because the finite grid spacing breaks all other symmetries (for example, the five-fold rotational symmetry in icosahedral molecules). In the Hamiltonian, only some of the present operations are used, and the selection is very stringent. If you feel not all possible symmetry operations are included in the Hamiltonian, try rotating or translating the whole system.
3. Unrecognized chemical element: Parsec has an internal periodic table (file `ptable.f`), and it checks for the existence of input elements. If the calculation stops with "unknown element", edit the periodic table.
4. Current pseudopotential generator does not support asymptotically corrected functionals (LB, CS).

7.2 Future implementations

1. Add k-point sampling in periodic systems.
2. Allow for a non-orthorhombic periodic cell if periodic boundary conditions are used.