

CPMD

Car-Parrinello Molecular Dynamics

An *ab initio* Electronic Structure and
Molecular Dynamics Program

Max-Planck-Institut für Festkörperforschung, Stuttgart
IBM Research Division, Zürich Research Laboratory
Atomistic Simulation Group, Queen's University of Belfast
Nanostructures et Magnétisme, DRFMC/SP2M, Grenoble

May 22, 2001

Send comments and bug reports to
cpmd@pr.mpi-stuttgart.mpg.de
or
Thierry.Deutsch@cea.fr

This manual is for CPMD version 3.4.3

Contents

1	Introduction	1
1.1	Version 1.0 - 3.0	1
1.2	Version 3.3	1
2	Citation	2
3	Installation	2
4	Running CPMD	3
4.1	Starting a run	3
4.2	Stopping a run	3
4.3	Files	4
4.4	Memory	5
5	Input Description	6
5.1	Basic rules	6
5.2	General description	7
5.3	Section &CPMD ...&END	8
5.3.1	Main keywords	10
5.3.2	Optimization keywords	12
5.3.3	Molecular dynamics keywords	14
5.3.4	Diagonalisation keywords	17
5.3.5	Free energy keywords	18
5.3.6	Mixing keywords	18
5.3.7	Restart and Save keywords	19
5.3.8	Further keywords	22
5.4	Section &SYSTEM ...&END	26
5.4.1	Main keywords	27
5.4.2	k-point keywords	28
5.4.3	Further keywords	29
5.5	Section &PIMD ...&END	33
5.6	Section &ATOMS ...&END	36
5.7	Section &DFT ...&END	41
5.8	Section &PROP ...&END	44
5.9	Section &BASIS ...&END	46
6	Pseudopotentials	47
6.1	Norm-conserving Pseudopotentials: Atomic Code for PP generation	48
6.2	Vanderbilt ultra-soft Pseudopotentials: Atomic Code	49
7	Input Examples	52

8	Visualization	63
8.1	Molecular Dynamics	63
8.2	Contour and 3-D Plots	63
9	Hints and Tricks	64
9.1	Wavefunction Initialization	64
9.2	Geometry Optimization	64
9.3	Molecular Dynamics	65
9.3.1	Choosing the Nosé-Hoover chain thermostat parameters	65
9.4	Hints of using FEMD	66
9.4.1	Lanczos Parameters	66
9.4.2	Other important FEMD parameters	67

1 Introduction

1.1 Version 1.0 - 3.0

In summer 1993 we started a project to combine the two different ab initio molecular dynamics codes that were used in the group for computational physics of the IBM Research Laboratory in Rüschlikon. There was the IBM AIX version of the original Car-Parrinello code ported by J. Kohanoff and F. Buda and the version of the code by A. Pasquarello, K. Laasonen and F. Buda that could handle pseudopotentials of the Vanderbilt type. Further goals were to provide a common platform for future developments, as new integration techniques or parallelization. New features of the code include a keyword driven input, an initial guess from atomic pseudo-wavefunctions, a module for geometry optimization, several new types of molecular dynamics, Nosé thermostats, a diagonalisation routine to get Kohn-Sham energies and many more. The following people contributed in many different ways to the code and its pseudopotential library:

Wanda Andreoni, Peter Blöchl, Jürg Hutter, Dominik Marx, Mark Tuckerman, Paolo Focher, Paolo Giannozzi, Ettore Fois, Alessandro Curioni, Michele Parrinello, Ursula Röthlisberger.

Since January 1995 development of the code proceeds at the Max-Planck-Institut für Festkörperforschung in Stuttgart. New features of this version are constant pressure molecular dynamics using the Parrinello-Rahman Lagrangian, the possibility for symmetry constraints, and Stefan Goedecker's dual space pseudopotentials. The library concept for the pseudopotentials has been changed. The code is now much cleaner and should cause no major problems to port to new platforms. Currently IBM Risc and Cray PVP multi processor versions, and distributed memory parallel versions for the IBM SP2 and Cray T3D are available. Special thanks to Pietro Ballone, Marco Bernasconi, Mark Tuckerman, and Stefan Goedecker as well as Rüdiger Wolff from Cray.

1.2 Version 3.3

Since 1996, Thierry Deutsch and Ali Alavi, working in Queen's University Belfast, have implemented a k-point code based on the free energy density functional (FEMD) [42]. This method has now been integrated into V3.3. The k-point code has been developed primarily to do electronic structure calculations and ab initio MD for metallic systems. This code is based on a diagonalisation scheme together with density mixing to achieve self-consistency.

2 Citation

Publications of results obtained with CPMD should acknowledge its use by an appropriate citation of the following kind.

CPMD, J. Hutter, A. Alavi, T. Deutsch, M. Bernasconi, St. Goedecker, D. Marx, M. Tuckerman, M. Parrinello, MPI für Festkörperforschung and IBM Zurich Research Laboratory 1995-1999.

3 Installation

This version of CPMD is equipped with a shell script to create automatically an appropriate Makefile for the current platform. If you run the shell script `Configure` without any options it will tell you which platforms are available. Choose the label for the target platform and run the script again:

```
Configure platform > Makefile
```

The executable can then be installed using the `make` command. The `Configure` script will also set the architecture-dependent parameter `IRAT` according to the target platform. Use `Configure -help` to have all options.

4 Running CPMD

4.1 Starting a run

The CPMD program is started with the following command:

```
cpmd.x file.in [PP_path] > file.out
```

Running `cpmd.x` requires:

an **input file** `file.in` (see 5 and 7)

and

pseudopotential files for all atomic species specified in `file.in` (see 5.6).

The path to the pseudopotential library can be given in different ways:

- One possibility is to specify the path in an environment variable **PP_LIBRARY_PATH**.
- If the variable **PP_LIBRARY_PATH** is not set the program checks if a second command line argument `[PP_path]` is given.
- If neither the environment variable nor the second command line argument are set, the program assumes that the pseudopotential files are in the current directory.

During the run `cpmd.x` creates different outputs:

Various information to control the correct operation of the program is printed to standard output (in our case redirected to the file `file.out`).

Detailed data is written into different files (depending on the keywords specified in the input `file.in`). An overview on them is given in Section 4.3.

4.2 Stopping a run

Jobs can be stopped at the next breakpoint by creating a file:

```
EXIT
```

in the run directory, e.g.

```
touch EXIT
```

The file is removed automatically after it has been detected.

In IBM, DEC, SGI, and SUN machines there is the possibility to use the command:

```
kill -30 pid
```

where `pid` is the process ID.

4.3 Files

This is an incomplete list of the files used or created by CPMD:

<i>file.in</i>	Input file
RESTART	Original restart file (read if no LATEST-file exists)
RESTART.x	Old/New restart files
LATEST	Information file on the last restart file
GEOMETRY	Current ionic positions and velocities
GSHELL	\mathbf{G}^2 NOT normalized, G-shells $ \mathbf{G} $ in a.u. and related shell index
TRAJECTORY	All ionic positions and velocities (trajectory)
ENERGIES	All energies along the trajectory
MOVIE	Atomic coordinates in Movie format
STRESS	The "trajectory" of stress tensors
CELL	The "trajectory" of the unit cell
CONSTRAINT	The "trajectory" of constraint forces
DIPOLE	The "trajectory" of dipole moments
DENSITY.x	Charge density in Fourier space
SPINDEN.x	Spin density in Fourier space
ELF	Electron localization function in Fourier space
HESSIAN	Approximate hessian
FINDIF	Positions and gradients for finite difference calculations
MOLVIB	The matrix of second derivatives, as used by the program MOLVIB
ENERGYBANDS	Eigenvalues for each k points
KPTS_GENERATION	Output of k points generation
WANNIER_CENTER	Centers of the Wannier functions
IONS+CENTERS	Trajectory of ions and WFCs in a.u.
WANNIER_DOS	Projection of the Wannier functions onto the Kohn-Sham states

In case of path integral runs, every replica $s = \{1, \dots, P\}$ gets its own RESTART_ s , RESTART_ s .x, DENSITY_ s .x, ELF_ s .x, and GEOMETRY_ s file.

In general, existing files are **overwritten!**

Exceptions are “trajectory” type files (TRAJECTORY, ENERGIES, MOVIE, STRESS, ...), they are **appended**.

4.4 Memory

Running cpmd.x involves certain requirements on the size of the memory. This section shall help to estimate the size of a given calculation.

The following quantities specify the size of a calculation:

N_{SP}	Number of atomic species
N_{AT}	Total number of atoms
N_{ST}	Number of states
N_{GW}	Number of plane waves (wavefunction cut-off)
N_{KP}	Number of k points

All these parameters are determined by the system under study and the chosen computational set-up. The number of plane waves N_{GW} depends on the wavefunction cut-off E_{cut} and can be approximated as

$$N_{\text{GW}} \approx \frac{\Omega}{6\pi^2} E_{\text{cut}}^{3/2},$$

where Ω is the volume of the cell and E_{cut} the wavefunction cut-off.

- The **basic memory** needed is $\approx (80 + 24N_{\text{SP}} + 4N_{\text{AT}})N_{\text{GW}}$.
- The calculation of ionic forces needs $\approx 80N_{\text{GW}}$.
- Wavefunctions and velocities need $6N_{\text{ST}}N_{\text{GW}}$, for Vanderbilt pseudo-potentials you have to add another $2N_{\text{ST}}N_{\text{GW}}$.
- Wavefunction initialization with atomic pseudo-wavefunctions needs $2N_{\text{AT}}N_{\text{AST}}N_{\text{GW}}$ memory, where N_{AST} is the number of atomic states per atom.
- The memory needed for DIIS is $(2 + 1.5N_{\text{DIIS}})N_{\text{ST}}N_{\text{GW}}$, where N_{DIIS} is the number of DIIS vectors.
- The memory needed for Molecular Dynamics is $3N_{\text{ST}}N_{\text{GW}}$, and $4N_{\text{ST}}N_{\text{GW}}$ when Vanderbilt pseudo-potentials or nonorthogonal orbitals are used.
- Dipole Dynamics runs need additional memory. Especially for parallel runs, the memory per processor has to be at least $8N_{\text{GW}}$.

5 Input Description

5.1 Basic rules

- **Warning:** Do not expect the input to be logic. The programmer's logic may be different from yours.
- **Warning:** This input description may not refer to the actual version of the program you are using. Therefore the ultimate and authoritative input guide is the source code.
Most of the input can be found in the files `control.F`, `sysin.F`, `pi_cntl.F`, `ratom.F`, `recpnew.F`, `dftin.F`, `proppt.F`, and `setbasis.F`.
- The input is in free format unless especially stated.
- Lines that do not match a keyword are treated as comments.
- The order of the keywords is arbitrary except where especially stated.
- Only keywords with capital letters match.
- Lists inclosed in { } imply that you can choose **exactly one** of the items.
- Lists inclosed in [] imply that you can choose **any number** of items on the same line.
- Keywords typed in slanted style are not fully tested and their use is on your own risk.
- The full keyword has to be within columns 1 to 80.

5.2 General description

The input file is composed of different parts

&CPMD ...	&END	↔	Control parameters for calculation
&SYSTEM ...	&END	↔	Simulation cell
&PIMD ...	&END	↔	Path Integral Molecular Dynamics (PIMD)
&ATOMS ...	&END	↔	Atoms and pseudopotentials
&DFT ...	&END	↔	Exchange and correlation functional
&PROP ...	&END	↔	Calculation of properties
&BASIS ...	&END	↔	Basis sets for properties or initial guess

A detailed discussion of the different keywords will be given in the following chapters.

CMASS
QUENCH [IONS, ELECTRONS, CELL, BO]
TEMPERATURE [ELECTRON]
TEMPCONTROL [IONS, ELECTRONS, CELL]
NOSE [IONS, ELECTRONS, CELL] [ULTRA,MASSIVE]
NOSE PARAMETERS
ANNEALING [IONS,ELECTRONS,CELL]
RESCALE OLD VELOCITIES
DIPOLE DYNAMICS [SAMPLE]

Diagonalisation keywords:

LANCZOS DIAGONALISATION [ALL]
LANCZOS PARAMETER [ALL]
DAVIDSON DIAGONALISATION
DAVIDSON PARAMETER

Free energy keywords:

TROTTER FACTORISATION OFF
TROTTER FACTOR
BOGOLIUBOV CORRECTION {OFF}

Mixing keywords:

ANDERSON MIXING
DIIS MIXING
ALEXANDER MIXING
MOVERHO
BROYDEN MIXING

Restart and Save keywords:

RESTART [WAVEFUNCTION, COORDINATES, ..., LATEST]
PRINT {ON,OFF} [INFO, EIGENVALUES, COORDINATES,
FORCES, WANNIER]
PRINT ENERGY {ON,OFF} [EKIN, ELECTROSTATIC, ESR, ESELF,
EBAND, EPSEU, EHEP, EHEE, EHII, ENL,
EXC, VXC, EGC, EBOGO]

STORE [{OFF} WAVEFUNCTIONS, DENSITY]
MAXCPU
STRUCTURE [BONDS, ANGLES, DIHEDRALS]
RHOOUT {BANDS}
ELF [PARAMETER]
ELECTROSTATIC POTENTIAL
EXTERNAL POTENTIAL
RESTFILE
TRAJECTORY [SAMPLE] {OFF}
MOVIE [SAMPLE] {OFF}
COMPRESS WRITE_{enn}
ENERGYBANDS

Further keywords:

LSD or LOCAL SPIN DENSITY
NONORTHOGONAL ORBITALS {OFF}
HARMONIC REFERENCE SYSTEM {OFF}
SCALED MASSES {OFF}
INITIALIZE WAVEFUNCTIONS {RANDOM, ATOMS}
HAMILTONIAN CUTOFF
ORTHOGONALIZATION {LOWDIN, GRAM-SCHMIDT}
RATTLE
RANDOMIZE [WAVEFUNCTION, COORDINATES, CELL, DENSITY]
FINITE DIFFERENCES
PROJECT {NONE, DIAGONAL, FULL}
MEMORY {SMALL, BIG}
ISOLATED MOLECULE
CENTER MOLECULE [OFF]
CLASSTRESS
SPLINE [POINTS QFUNCTION INIT RANGE]
REAL SPACE FNL [PARAMETER]
FILEPATH
TASKGROUPS {MAXIMUM, MINIMUM}
CHECK MEMORY
DISTRIBUTE FNL

5.3.1 Main keywords

It is compulsory to specify one of the following keywords:

MOLECULAR DYNAMICS
OPTIMIZE WAVEFUNCTION
OPTIMIZE GEOMETRY
KOHN-SHAM ENERGIES
VIBRATIONAL ANALYSIS
PROPERTIES
INTERFACE

They are mutually exclusive.

MOLECULAR DYNAMICS [CP, BO]

Perform a **molecular dynamics (MD)** run. **CP** stands for a Car-Parrinello type MD. With the option **BO** a Born-Oppenheimer MD is performed, i.e. the wavefunctions are reconverged after each MD-step. **Default** is **CP**.

OPTIMIZE [GEOMETRY, WAVEFUNCTION, COMBINED]

GEOMETRY causes the program to optimize the geometry and the wavefunction of the system.

Using **WAVEFUNCTION** only the wavefunction is optimized.

COMBINED stands for a molecular dynamics based geometry optimization method. This method is not fully operational in the current version.

INTERFACE {EGO}

[MULLIKEN, LOWDIN, ESP, HIRSHFELD] Use CPMD together with a classical molecular dynamics code. This needs special software.

KOHN-SHAM ENERGIES [OFF, NOWAVEFUNCTION]

Calculation of the Kohn-Sham energies and the corresponding orbitals. The number of empty states that have to be calculated in addition to the occupied states is read from the next line.

The Kohn-Sham orbitals are stored on the file **RESTART.x** except if the keyword **NOWAVEFUNCTION** is used. In this case, the program does not allocate memory for wavefunctions for all k points. It computes eigenvalues k point per k point losing information about wavefunctions. This keyword is used for band structure calculation to compute the eigenvalues for many k points.

Default is not to calculate Kohn-Sham energies (**OFF**).

Warning: The usage of this keyword needs special care (especially during restarts).

VIBRATIONAL ANALYSIS

Calculate harmonic frequencies by finite differences of first derivatives. See also keyword **Finite Differences**.

PROPERTIES

Calculate some properties.

This keyword requires further input in the section &PROP ... &END.

PATH SAMPLING

Use CPMD together with a reaction path sampling program. This needs special software.

FREE ENERGY FUNCTIONAL

Calculates the electronic free energy using **free energy density functional** [42] from DFT at finite temperature.

This option needs additional keywords (free energy keywords).

By **default** we use **Lanczos diagonalisation** with **Trotter factorization** and **Bogoliubov correction**. If the number of states is not specified, use $N_{electrons}/2 + 4$.

PATH INTEGRAL

Perform a **path integral molecular dynamics** calculation [43, 44].

This keyword requires further input in the section &PIMD ... &END.

5.3.2 Optimization keywords**MAXSTEP**

The **maximum number of steps** for wavefunction or geometry optimization or molecular dynamics to be performed. The value is read from the next line.

Default is 10000 steps.

CONVERGENCE [ORBITALS, GEOMETRY, CELL]

The convergence criteria for optimization runs are specified.

The maximum value for the biggest element of the gradient of the wavefunction (**ORBITALS**), of the ions (**GEOMETRY**), or the cell (**CELL**) is read from the next line.

Default values are 10^{-5} for the wavefunction, $5 \cdot 10^{-4}$ for the ions, and 1. for the cell.

For diagonalisation scheme, the first value is the biggest variation of a density component (defaults are 10^{-3} and 10^{-3}).

STEEPEST DESCENT [ELECTRONS, IONS, CELL] [LINE]

[NOPRECONDITIONING] (only electrons)

[LINE] (only ions)

Use the method of **steepest descent** for the **optimization** of wavefunction and/or atomic positions and/or cell.

If both options are specified in a geometry optimization run, a simultaneous optimization is performed.

Preconditioning of electron masses (scaled masses) is used by default. The preconditioning is controlled by the keyword **HAMILTONIAN CUTOFF**. Optionally preconditioning can be disabled.

For the ionic optimization the steplength is controlled by the keywords **TIMESTEP** and **EMASS**.

[TSDE, TSDP, TSDC]

[NOPRECONDITIONING](only electrons.)

Short forms for the different **STEEPEST DESCENT** options.

PCG [MINIMIZE,NOPRECONDITIONING]

Use the method of **preconditioned conjugate gradients** for **optimization** of the **wavefunction**.

The fixed steplength is controlled by the keywords **TIMESTEP** and **EMASS**.

If option **MINIMIZE** is chosen then line searches are performed.

The preconditioning is controlled by the keyword **HAMILTONIAN CUTOFF**. Optionally preconditioning can be disabled.

CONJUGATE GRADIENTS [ELECTRONS, IONS]

[NOPRECONDITIONING] (only electrons)

For the electrons, the keyword is equivalent to **PCG**. For the ions, use the conjugate gradients scheme for optimisation of ionic positions.

ODIIS [NOPRECONDITIONING]

Use the method of **direct inversion** in the iterative subspace for the **optimization** of the **wave functions** [6].

The number of DIIS vectors is read from the next line.

ODIIS with **10 vectors** is the **default** method in optimization runs.

The preconditioning is controlled by the keyword **HAMILTONIAN CUTOFF**. Optionally preconditioning can be disabled.

GDIIS

Use the method of **direct inversion** in the iterative subspace combined with a **quasi-Newton method** (using BFGS) for **optimization** of the **ionic positions** [13]. The number of DIIS vectors is read from the next line.

GDIIS with **5 vectors** is the **default** method in optimization runs.

BFGS

Use a **quasi-Newton** method for **optimization** of the **ionic positions**. The approximated Hessian is updated using the Broyden-Fletcher-Goldfarb-Shano procedure [14].

RFO ORDER=nsorder

Rational function approximation combined with a quasi-Newton method (using BFGS) for **optimization** of the **ionic positions** is used [7]. A saddle point of order nsorder is searched for.

HESSIAN [DISCO,SCHLEGEL,UNIT]

The initial approximate **Hessian** for a **geometry optimization** is constructed using empirical rules with the DISCO [12] or Schlegel's [10] parametrization, or simply a unit matrix is used.

5.3.3 Molecular dynamics keywords**PARRINELLO-RAHMAN [NPT]**

To be used together with **MOLECULAR DYNAMICS**.

A **variable cell MD** with the **Parrinello-Rahman Lagrangian** is performed. With the additional keyword a **constant NPT MD** using the method of Martyna, Tobias, and Klein [39].

If this keyword is used together with other run options like **OPTIMIZE WAVEFUNCTIONS**, calculations with different reference cells can be performed.

TIMESTEP

The **time step** in **atomic units** is read from the next line.

Default is a timestep of **5 a.u.** ($1 a.u. = 0.0241888428$ fs).

EMASS

The **fictitious electron mass** in **atomic units** is read from the next line.

Default is **400 a.u.**

CMASS

The **fictitious cell mass** in **atomic units** is read from the next line.

Default is **200 a.u.**

QUENCH [IONS, ELECTRONS, CELL, BO]

The **velocities** of the **ions**, **wavefunctions** or the **cell** are set to zero at the beginning of a run.

With the option **BO** the wavefunctions are converged at the beginning of the MD run.

TEMPERATURE

The **initial temperature** in Kelvin of the **ions** is read from the next line.

TEMPCONTROL [IONS, ELECTRONS, CELL]

The **temperature** of the **ions** in Kelvin, the **fictitious kinetic energy** of the **electrons** in atomic units or the **kinetic energy** of the **cell** in atomic units (?) is controlled by scaling.

The **target** temperature and the **tolerance** for the ions or the target kinetic energy and the tolerance for the electrons or the cell are read from the next line.

NOSE [IONS, ELECTRONS, CELL] [ULTRA,MASSIVE]

Nosé-Hoover chains [8, 9] for the **ions**, **electrons** or **cell parameters** are used.

The **target temperature** in Kelvin and the **thermostat frequency** in cm^{-1} , respectively the **fictitious kinetic energy** in atomic units and the **thermostat frequency** in cm^{-1} are read from the next line.

For the ionic case the additional keyword **ULTRA** selects a thermostat for each species, and the keyword **MASSIVE** selects a thermostat for each degree of freedom.

NOSE PARAMETERS

The **parameters** controlling the **Nosé thermostats** are read in the following order from the next line:

- The **length** of the Nosé-Hoover chain for the **ions**,
- the **length** of the Nosé-Hoover chain for the **electrons**,
- the **length** of the Nosé-Hoover chain for the **cell parameters**.

(The respective **default** values are **4**.)

- The **multiplication factor** (NEDOF0, a real number) for the number of **electronic** degrees of freedom. The used degrees of freedom (NEDOF) are defined as $NEDOF = NEDOF0 * X$. If NEDOF0 is a negative number X is the true number of DOFs, if it's a positive number, X is the number of electronic states (**default** for NEDOF0 is **6**).
- The order of the **Suzuki/Yoshida integrator** (**default** is **7**, choices are 3, 5, 7, 9, 15, 25, 125, and 625),
- and the **decomposition ratio** of the time step (**default** is **1**).

If this keyword is omitted, the defaults are used.

If the keyword is used all parameters have to be specified.

ANNEALING [IONS,ELECTRONS,CELL]

Scale the **ionic**, **electronic**, or **cell** velocities every time step.
The scaling factor is read from the next line.

RESCALE OLD VELOCITIES

Rescale ionic velocities after **restart** to the temperature specified by either TEMPERATURE, TEMPCONTROL IONS, or NOSE IONS (useful if the type of ionic thermostating is changed, no RESTART NOSEP in this case).

Note only for path integral runs: the scaling is only applied to the first (centroid) replica.

DIPOLE DYNAMICS [SAMPLE,WANNIER]

Calculate the **dipole moment** every *NSTEP* iteration in MD.

NSTEP is read from the next line if the keyword SAMPLE is present.

Default is **every** time step.

The keyword **Wannier** allows the calculation of optimally localized Wannier functions. The localization procedure is controlled by the following keywords.

WANNIER OPTIMIZATION [SD,JACOBI]

Use steepest descent or Jacobi rotation method for the orbital localization.

Default is **Jacobi** rotations.

WANNIER PARAMETER

W_STEP, *W_EPS*, *W_RAN*, *W_MAXS* are read from the next line. *W_STEP* is the step size of the steepest descent algorithm used in the optimization procedure (default value 0.1). *W_EPS* the convergence criteria for the gradient (default value $1.e - 7$). *W_RAN* the amplitude for the initial random rotation of the states (default value 0.0). *W_MAXS* is the maximum steps allowed in the optimization (default value 200).

WANNIER REFERENCE

The vector *W_REF* is read from the next line, which consists of 3 coordinates *x*, *y*, *z*. These are assumed as the origin for the WFCs positions and related ionic coordinates (i.e. $\mathbf{R}_I \rightarrow \mathbf{R}_I - (x, y, z)$).

WANNIER TYPE [VANDERBILT,RESTA]

Indicates the type of Wannier functions. Vanderbilt type is the default.

WANNIER WFNOUT [ALL,PARTIAL,LIST,DENSITY]

Controls the printing of Wannier functions. Either all or only some

of the functions can be printed. This will be done at the end of each calculation of Wannier functions.

5.3.4 Diagonalisation keywords

By **default** the program uses **conjugate-gradient** or **Car-Parrinello** schemes.

You need to specify LANCZOS or DAVIDSON DIAGONALISATION in order to use these diagonalisation schemes at $T_{el} = 0K$.

LANCZOS DIAGONALISATION [ALL]

Use **Lanczos diagonalisation** scheme.
Default with free energy functional.

LANCZOS PARAMETER [N=n] [ALL]

Four parameters for Lanczos diagonalisation are read from the next line:

- Maximal number of Lanczos iterations (50 is enough),
- Maximal number for the Krylov sub-space (8 best value),
- Blocking dimension (\leq NSTATE, best in range 20-100) If you provide a negative number or zero, this parameter is fixed by the program in function of the number of states ($((n+1)/(int(n/100+1)))$).
- Tolerance for the accuracy of wavefunctions (10^{-8} otherwise 10^{-12} with Trotter approximation)

If n is specified, read $n-1$ lines after the first one, containing a threshold density and a tolerance. See section on hints for more information.

DAVIDSON DIAGONALISATION

Use **Davidson diagonalisation** scheme.

DAVIDSON PARAMETER

This keyword controls the Davidson diagonalisation routine used to determine the **Kohn-Sham energies**.

The maximum number of additional vectors to construct the Davidson matrix, the convergence criterion and the maximum number of steps are read from the next line.

Defaults are 10^{-5} and the same number as states to be optimized. If the system has 20 occupied states and you ask for 5 unoccupied states, the default number of additional vectors is 25. By using less than 25 some memory can be saved but convergence might be somewhat slower.

5.3.5 Free energy keywords

Specific keywords for the FREE ENERGY FUNCTIONAL scheme.

TEMPERATURE ELECTRON

The **electronic temperature** is read from the next line.

Default is 1000K.

TROTTER FACTOR

Solve $e^{-H/k_B T}$ directly using **Trotter approximation**
 $(e^{-pH} \simeq e^{-pK/2} e^{-pV} e^{-pK/2})$.

The Trotter approximation is twice as fast.

The Trotter factor is read from the next line (typically 0.001 is very accurate).

TROTTER FACTORIZATION OFF

Do not use Trotter factorization to calculate free energy functional.

BOGOLIUBOV CORRECTION [OFF]

Computes the Bogoliubov correction for the energy of the Trotter approximation or not.

Default is no Bogoliubov correction.

5.3.6 Mixing keywords

ANDERSON MIXING

Anderson mixing for the electronic density during self-consistent iterations.

In the next line the parameter (between 0 and 1) for the Anderson mixing is read (**default 0.2**).

ANDERSON MIXING $N = n$

Like ANDERSON MIXING, but mixing parameter can be specified for different threshold densities.

n different thresholds can be set. The program reads n lines, each with a threshold density and an Anderson mixing parameter.

DIIS MIXING

Use the **Direct Inversion Iterative Scheme** to mix density.

Read in the next line the number of previous densities (NRDIIS) for the mixing (not useful).

DIIS MIXING [$N = n$]

Like DIIS MIXING, but number of previous densities for the mixing can be specified as a function of the density.

n different thresholds for the density can be set. The program reads n lines with a threshold density and a NRDIIS number (number of previous densities for the mixing). Numbers NRDIIS have to increase. If the NRDIIS is equal to 0, Anderson mixing is used. Very efficient is to use Anderson mixing and afterwards DIIS mixing.

ALEXANDER MIXING

Mixing used during optimization of geometry or molecular dynamics. Parameter read in the next line (**default 0.9**).

BROYDEN MIXING

Mixing used during optimization of geometry or molecular dynamics. Parameters read in from next line.

BROYMIX, *ECUTBROY*, *W02BROY*, *NFRBROY*, *IBRESET*

These mean:

BROYMIX: Initial mixing (e.g. 0.1, default value 0.5).

ECUTBROY: Cutoff for Broyden mixing (DUAL*ECUT is the best choice and the default).

W02BROY: w_0^2 parameter of Johnson [52] (default 0.01).

NFRBROY: Number of Anderson mixing steps done before Broyden mixing (default 0).

IBRESET: Number of Broyden vectors (5 is usually a good value and the default).

You can also specify some parameters with the following syntax:

[**BROYMIX**=*BROYMIX*] [**ECUTBROY**=*ECUTBROY*]

[**W02BROY**=*W02BROY*] [**NFRBROY**=*NFRBROY*]

[**IBRESET**=*IBRESET*]

Finally, you can use the keyword **DEFAULT** to use the default values.

MOVERHO

Use atomic or pseudowavefunctions to project wavefunctions in order to calculate the new ones with movement of atoms Read in the next line the parameter (typically 0.2).

5.3.7 Restart and Save keywords**RESTART** [*OPTIONS*]

This keyword controls what data is read (at the beginning) from the file RESTART.x.

Warning: You can only read data that has been previously written into the RESTART-file.

A list of different *OPTIONS* can be specified. List of valid options:

WAVEFUNCTION Read old **wavefunction** from restart file.

OCCUPATION Read old **occupation numbers** (useful for free energy functional).

COORDINATES Read old **coordinates** from restart file.

VELOCITIES Read old **ionic, wavefunction and cell velocities** from restart file.

CELL Read old **cell parameters** from restart file.

GEOFILE Read old **ionic positions and velocities** from file **GEOMETRY**. This file is updated every timestep during the run.

ACCUMULATORS Read old **accumulator values** from restart file.

HESSIAN Read old **approximate Hessian** from file *HESSIAN*.

NOSEE Restart **Nosé thermostats for electrons** with values stored on restart file.

NOSEP Restart **Nosé thermostats for ions** with values stored on restart file.

NOSEC Restart **Nosé thermostats for cell parameters** with values stored on restart file.

LATEST Restart from the **latest restart** file as indicated in file *LATEST*.

VIBANALYSIS Use the information on finite differences stored in the file *FINDIF*. This option requires a valid restart file for the wavefunctions, even when wavefunctions and coordinates are recalculated or read from the input file.

POTENTIAL Read an old potential from the restart file. This applies to restarts for Kohn-Sham energy calculations.

KPOINTS Restart with kpoints.

DENSITY Restart with electronic density.

ALL Restart with all fields of RESTART file

PRINT {ON,OFF} [INFO, EIGENVALUES, COORDINATES, FORCES, WANNIER]

A **detailed output** is printed every *IPRINT* iterations. Either only

different contribution to the energy or in addition the atomic coordinates and the forces are printed. *IPRINT* is read from the next line if the keywords **ON** or **OFF** are not specified.

Default is **only energies** after the first step and at the end of the run. **OFF** switches the output off.

PRINT ENERGY {ON, OFF} [EKIN, ELECTROSTATIC, ESR, ESELF, EBAND, EPSEU, EHEP, EHEE, EHII, ENL, EXC, VXC, EGC, EBOGO]

Display or not information about energies.

STORE {OFF} [WAVEFUNCTIONS, DENSITY, POTENTIAL]

The **RESTART** file is **updated** every *ISTORE* steps. *ISTORE* is read from the next line. **Default** is at the **end of the run**.

Moreover, in the same line of the number *ISTORE*, you can specify the number of self-consistent iterations (with *SC=number*) between two updates of **RESTART** file. If **OFF** is specified, do not store wavefunctions and/or density (*ISTORE* is not necessary).

MAXCPU

The **maximum CPU TIME** of **steps** is read from the next line.

Default is no limit.

STRUCTURE [BONDS, ANGLES, DIHEDRALS]

Print **structure information** at the end of the run.

Bonds, angles and dihedral angles can be printed. Dihedral angles are defined between 0 and 180 degrees. This might change in the future.

RHOOUT [BANDS]

Store the density at the end of the run on file *DENSITY*.

If the keyword **BANDS** is defined then on the following lines the number of bands to be plotted and their position have to be given. Each band is stored on its own file *DENSITY.num*. For spin polarized calculations besides the total density also the spin density is stored on the file *SPINDEN*.

ELF [PARAMETER]

Store the **total density** and the **electron localization function** ELF [36] on file.

ELF parameters can be changed optionally with the **PARAMETER** keyword. The two parameters *elfcut* and *elfeps* are read from the next line.

ELECTROSTATIC POTENTIAL

Store the **electrostatic potential** on file.

EXTERNAL POTENTIAL

Read an external potential from file.

RESTFILE

The number of distinct RESTART-files is read from the next line.

The restart files are written in turn. **Default is 1**. If you specify e.g. 3, then the files RESTART.1, RESTART.2, RESTART.3 are used.

TRAJECTORY [OFF, SAMPLE]

Store the **atomic positions and velocities** at every *NTRAJ* time step on file *TRAJECTORY*. This is the **default for MD runs**. This can be prevented by specifying the keyword OFF. If the keyword SAMPLE is given then *NTRAJ* is read from the next line, otherwise the default value for *NTRAJ* is 1.

MOVIE [OFF, SAMPLE]

Write the atomic coordinates without applying periodic boundary conditions in MOVIE [15] format every *IMOVIE* time steps on file *MOVIE*. This can be prevented by specifying the keyword OFF. **Default is not** to write a movie file. If the keyword SAMPLE is given *IMOVIE* is read from the next line, otherwise the default value for *IMOVIE* is 1.

COMPRESS [WRITE_{nn}]

Write the wavefunctions with *nn* bytes precision to the restart file.

WRITE32, **WRITE16**, **WRITE8**, and **WRITEAO** are possible choices.

WRITE32 corresponds to the compress option in older versions.

WRITEAO stores the wavefunction as a projection on atomic basis sets. The atomic basis set can be specified in the section &BASIS ... &END. If this input section is missing a default basis from Slater type orbitals is constructed.

ENERGYBANDS

Write the band energies (eigenvalues) for *k* points in the file ENERGY-BANDS.

5.3.8 Further keywords**LSD or LOCAL SPIN DENSITY**

Use the **local spin density** approximation.

Warning: Not all functionals are implemented for this option.

NONORTHOGONAL ORBITALS [OFF]

Use the norm constraint method [5] for molecular dynamics or non-orthogonal orbitals in an optimization run.

On the next line the limit of the off diagonal elements of the overlap matrix is defined. **Warning:** Adding or deleting this option during a MD run needs special care.

HARMONIC REFERENCE SYSTEM [OFF]

Switches harmonic reference system integration on/off.

The number of shells included in the analytic integration is controlled with the keyword **HAMILTONIAN CUTOFF**.

By **default** this option is switched **off**.

SCALED MASSES [OFF]

Switches the usage of g-vector dependent masses on/off.

The number of shells included in the analytic integration is controlled with the keyword **HAMILTONIAN CUTOFF**.

By **default** this option is switched **off**.

INITIALIZE WAVEFUNCTION [RANDOM, ATOMS]

The **initial guess for wavefunction optimization** are either random functions or functions derived from the atomic pseudo-wavefunctions.

Default is to use the **atomic pseudo-wavefunctions**.

HAMILTONIAN CUTOFF

The lower cutoff for the diagonal approximation to the Kohn-Sham matrix is read from the next line.

Default is **0.5** atomic units.

For variable cell dynamics only the kinetic energy as calculated for the reference cell is used.

ORTHOGONALIZATION [LOWDIN, GRAM-SCHMIDT]

Orthogonalization in optimization runs is done either by a Lowdin (symmetric) or Gram-Schmidt procedure.

Default is Gram-Schmidt except for parallel runs where Lowdin orthogonalization is used with the conjugate-gradient scheme.

RATTLE

This option can be used to set the maximum number of iterations and the tolerance for the **iterative orthogonalization**. These two numbers are read from the next line.

Defaults are 30 and 10^{-6} .

**RANDOMIZE [COORDINATES, WAVEFUNCTION]
[DENSITY, CELL]**

The **ionic positions** or the **wavefunction** or the **cell parameters** are **randomly displaced** at the beginning of a run.

(DENSITY ??) The maximal amplitude of the displacement is read from the next line.

CLASSTRESS

Not documented.

FINITE DIFFERENCES

The step length in a finite difference run for vibrational frequencies (VIBRATIONAL ANALYSIS keywords) is read from the next line.

PROJECT [NONE, DIAGONAL, FULL]

This keyword is controlling the calculation of the constraint force in optimization runs.

MEMORY [SMALL, BIG]

Using **BIG**, the structure factors for the density cutoff are only calculated once and stored for reuse.

This option allows for considerable time savings in connection with Vanderbilt pseudopotentials.

Default is (**SMALL**) to **recalculate** them whenever needed.

ISOLATED MOLECULE

Calculate the ionic temperature assuming that the system consists of an isolated molecule or cluster.

Note: This keyword affects exclusively the determination of the number of dynamical degrees of freedom. This keyword does not activate the 'cluster option' SYMMETRY 0, but it is activated if SYMMETRY 0 is used. It allows studying an isolated molecule or cluster within periodic boundary conditions.

CENTER MOLECULE [OFF]

The center of mass is moved/not moved to the center of the computational box in a calculation with the cluster option.

This is only done when the coordinates are read from the input file.

SPLINE [POINTS, QFUNCTION]

This option controls the generation of the pseudopotential functions in g-space.

All pseudopotential functions are first initialized on a evenly spaced grid in g-space and then calculated at the needed positions with a

spline interpolation. The number of spline points is read from the next line when **POINTS** is specified.

(The **default** number is **501**.)

In addition it is possible to keep the Q-functions of the Vanderbilt pseudopotentials on the spline grid during the whole calculation and do the interpolation whenever needed. This option may be useful to save time during the initialization phase and memory in the case of Vanderbilt pseudopotentials when the number of shells is not much smaller than the total number of planewaves, i.e. for all cell symmetries except simple cubic and fcc.

FILEPATH

The path to the files written by CPMD (RESTART.x, MOVIE, ENERGIES, DENSITY.x etc.) is read from the next line. **Default** is the **current directory**.

TASKGROUPS [MINIMAL,MAXIMAL]

The number of taskgroups is read from the next line. The number of taskgroups has to be a divisor of the number of nodes in a parallel run.

CHECK MEMORY

Check sanity of dynamically allocated arrays whenever a change in the allocation is done. By default memory is checked only at break points.

DISTRIBUTE FNL

The array FNL is distributed in parallel runs.

5.4 Section &SYSTEM ... &END

In this section information on the supercell and the basis have to be given.
(This section is read in the subroutine sysin.F.)

Possible keywords:

Main keywords:

SYMMETRY
CELL [ABSOLUTE] [DEGREE]
CUTOFF [SPHERICAL,NOSPHERICAL]
STATES
OCCUPATION

K-point keywords:

KPOINT [SCALED] [BLOCK=mkpnt] [ALL]
KPOINT MONKHORST-PACK [SYMMETRIZE] [FULL]
KPOINT BANDS [SCALED]

Further keywords:

ANGSTROM
REFERENCE CELL [ABSOLUTE] [DEGREE]
ISOTROPIC CELL
CHARGE
CONSTANT CUTOFF
DENSITY CUTOFF NUMBER
PRESSURE
STRESS TENSOR
TESR
MESH
SCALE [S=sascale] [SX=sxscale] [SY=syscale] [SZ=szscale]
MULTIPLICITY
POINT GROUP [MOLECULE] [AUTO] [DELTA=delta]
SYMMETRIZE COORDINATES
POISSON SOLVER [HOCKNEY,TUCKERMAN,MORTENSEN]
SURFACE
POLYMER
DUAL
LOW SPIN EXCITATION
LSE PARAMETERS

5.4.1 Main keywords

The following keywords are compulsory:
SYMMETRY, **CELL**, **CUTOFF**.

SYMMETRY

The **supercell symmetry type** is read from the next line.
 You can put a number or a keyword.

- 0 **ISOLATED** system in a cubic/orthorhombic box [33] with ISOLATED MOLECULE option activated.
- 1 Simple **CUBIC**
- 2 **FACE CENTERED CUBIC (FCC)**
- 3 **BODY CENTERED CUBIC (BCC)**
- 4 **HEXAGONAL**
- 5 **TRIGONAL** or **RHOMBOHEDRAL**
- 6 **TETRAGONAL**
- 7 **BODY CENTRED TETRAGONAL (BCT)**
- 8 **ORTHORHOMBIC**
- 12 **MONOCLINIC**
- 14 **TRICLINIC**

CELL [ABSOLUTE, DEGREE]

The parameters specifying the **super cell** are read from the next line.
 Six numbers in the following order have to be provided: a , b/a , c/a , $\cos \alpha$, $\cos \beta$, $\cos \gamma$. For cubic phases, a is the lattice parameter. Redundant variables can be set to arbitrary values. With the keyword **ABSOLUTE**, you give a , b , and c . With the keyword **DEGREE**, you provide α , β , and γ in degrees instead of their cosine.

CUTOFF [SPHERICAL,NOSPHERICAL]

The **cutoff** for the plane wave basis in **Rydberg** is read from the next line. The keyword **SPHERICAL** is used with k points in order to have $|g + k|^2 < E_{cut}$ instead of $|g|^2 < E_{cut}$. This is the default.

STATES

The number of states used in the calculation is read from the next line.
 This keyword has to precede the keyword **OCCUPATION**.

OCCUPATION

The occupation numbers are read from the next line.
 This keyword must be preceded by **STATES**.

5.4.2 k-point keywords

KPOINTS [OPTIONS]

With no option, read in the next line the number of k-points and for each k-point, read the components in the cartesian coordinates (units $1/a$) and the weight.

MONKHORST-PACK Read in the next line three numbers for the Monkhorst-Pack mesh. The program calculates then the special k-points. With the keyword **SHIFT=kx ky kz** in the same line, you can precise the constant vector shift.

SYMMETRIZED Symmetrized special k-points mesh (useful if you use a constant vector shift).

FULL Construct full Monkhorst-Pack mesh with only inversion symmetry.

SCALED You can give k-points in reciprocal space coordinates.

BLOCK=n The block option, specifies the number of k-points in the memory. The program uses a swap file to store the wavefunctions.

BLOCK=n ALL Three swap files are used to store wavefunctions and others arrays related to k-points. Swap files are in the current directory or the temporary directory given by environment variable TMPDIR.

BANDS This options is to calculate the band structure.

For each line you have to specify the number of k-points for the band and the initial and the final k-point. To finish the input specify:

0 0. 0. 0. 0. 0. 0.

BLOCK=n [OPTIONS] The block option, specifies the number of k-points in the memory. The program uses a swap file to store the wavefunctions only by default. With the following options, you can change this behaviour:

ALL Three swap files are used to store wavefunctions and others arrays related to k-points. Swap files are in the current directory or the temporary directory given by environment variable TMPDIR. The use of memory is smaller than with the above option.

CALCULATED One swap file is used to store only wavefunctions. The other arrays related to k-points are calculated each time if needed.

NOSWAP The wavefunctions are not swapped. This is useful to calculate eigenvalues for each k point with few memory used.

Warning: The wavefunctions calculated are irrelevant. You have to specify explicitly some other options to use it:
 MAXSTEP 1 and
 STORE OFF WAVEFUNCTIONS DENSITY POTENTIAL.

5.4.3 Further keywords

ANGSTROM

The atomic coordinates and the supercell parameters are read in Ångströms.

Default is **atomic units**.

REFERENCE CELL [ABSOLUTE, DEGREE]

This cell is used to calculate the Miller indices in a constant pressure simulation. This keyword is only active together with the option **PARRINELLO-RAHMAN**.

The parameters specifying the reference (super) cell are read from the next line.

Six numbers in the following order have to be provided: a , b/a , c/a , $\cos \alpha$, $\cos \beta$, $\cos \gamma$.

The keywords **ABSOLUTE** and **DEGREE** are described in **CELL** option.

ISOTROPIC CELL

Specifies a constraint on the super cell in **constant pressure dynamics**. The shape of the cell is held fixed, only the volume changes.

CHARGE

The charge of the system is read from the next line.

Default is 0.

CONSTANT CUTOFF

Apply a **cutoff function to the kinetic energy** term in order to simulate **constant cutoff dynamics**.

The parameters A , σ , and E_o are read from the next line (all quantities have to be given in **Rydbergs**).

$$G^2 \rightarrow G^2 + A \left[1 + \operatorname{erf} \left(\frac{\frac{1}{2}G^2 - E_o}{\sigma} \right) \right]$$

DENSITY CUTOFF NUMBER

Read the **number of plane waves for density**.

Useful to calculate bulk modulus or properties depending on the volume. The given energy cutoff has to be bigger than the one to have the required plane wave density number.

PRESSURE

The **external pressure** on the system is read from the next line (in kbar).

STRESS TENSOR

In extension to the keyword PRESSURE the complete **stress tensor** in kbar can be specified. The **stress** on the system is read in the form:

$$\begin{array}{ccc} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{array}$$

TESR

The number of **additional supercells** included in the real space sum for the **Ewald term** is read from the next line. **For periodic systems, this keyword is unavoidable** (8 is enough).

MESH

The number of **real space mesh** points in x -, y - and z -direction is read from the next line.

If the values provided by the user are not compatible with the plane-wave cutoff or the requirements of the FFT routines the program chooses the next bigger valid numbers.

Default are the **minimal values** compatible with the energy cutoff and the **FFT** requirements.

SCALE [S=sascale] [SX=sxscale] [SY=syscale] [SZ=szscale]

Scale atomic coordinates of the system with the lattice constants (see **CELL**). You can indicate the scale for each axis. For instance, if you indicate SX=sxscale, you give your x-coordinates between 0. and sxscale (by default 1.).

MULTIPLICITY

This keyword only applies to **LSD calculations**.

The **multiplicity** ($2S+1$) is read from the next line.

Default is the **smallest possible** multiplicity.

POINT GROUP [MOLECULE] [AUTO] [DELTA=delta]

The **point group symmetry** of the system can be specified in the next line.

With the keyword **AUTO** in the next line, the space group is **determined automatically**.

This affects the calculation of nuclear forces and ionic positions. The electronic density and nuclear forces are symmetrized in function of point group symmetry. The group number is read from the next line.

Crystal symmetry groups:

1	1 (c1)	9	3m (c3v)	17	4/mmm (d4h)	25	222 (d2)
2	<1>(ci)	10	<3>m(d3d)	18	6 (c6)	26	mm2 (c2v)
3	2 (c2)	11	4 (c4)	19	<6> (c3h)	27	mmm (d2h)
4	m (c1h)	12	<4> (s4)	20	6/m (c6h)	28	23 (t)
5	2/m(c2h)	13	4/m (c4h)	21	622 (d6)	29	m3 (th)
6	3 (c3)	14	422 (d4)	22	6mm (c6v)	30	432 (o)
7	<3>(c3i)	15	4mm (c4v)	23	<6>m2(d3h)	31	<4>3m(td)
8	32 (d3)	16	<4>2m(d2d)	24	6/mmm(d6h)	32	m3m (oh)

You can specify the point group with their names using the keyword **NAME=**following by the name of the point group (one of both notations).

For molecular point groups the additional keyword **MOLECULE** has to be specified. The Schönflies symbol of the group is read in the following format from the next line:

Group symbol; order of principle axis

Possible group symbols are any Schönflies symbol with the axis number replaced by n (e.g. D n H). For molecular point groups a special orientation is assumed. The principle axis is along z and vertical symmetry planes are orthogonal to x .

DELTA= specifies the required accuracy (default= 10^{-6}).

With the keyword **AUTO**, the point group is **determined automatically**.

SYMMETRIZE COORDINATES

Input coordinates are **symmetrized** according to the **point group** specified.

This only makes sense when the structure already is close to the symmetric one.

POISSON SOLVER {HOCKNEY, TUCKERMAN, MORTENSEN} [PARAMETER]

Keyword determines the method for the solution of the Poisson equation for isolated systems. Either Hockney's method [33] or Martyna and Tuckerman's method [49] is used. The smooting parameter (for Hockney's method) or $L \times \alpha$ for Tuckerman's method can be read from the next line using the **PARAMETER** keyword.

SURFACE

Assume **periodic boundary** condition in *x*- and *y*-direction.

POLYMER

Assume **periodic boundary** condition in *x*-direction.

DUAL

The **ratio** between the **plane wave cutoff** for the **density** and the **wavefunction** is read from the next line.

Default is 4 (3 is often enough). Warning: You can have some trouble if you use DUAL option with the symmetrization of the electronic density.

LSE

Use the low spin excited state functional [50].

LSE PARAMETER

Only for LSE experts, see code.

5.5 Section &PIMD ...&END

Control section for the *ab initio* path integral molecular dynamics method (PIMD). (This section is read in the subroutine pi_cntl.F.)

See Refs. [43, 44, 46] for the implemented *ab initio* path integral molecular dynamics method.

Warning: Many options do not work in conjunction with quantum nuclei in the path integral representation.

Possible keywords:

TROTTER DIMENSION
 CENTROID DYNAMICS
 CLASSICAL TEST
 FACMASS
 INITIALIZATION
 GENERATE REPLICAS
 DEBROGLIE [CENTROID]
 READ REPLICAS
 STAGING
 NORMAL MODES
 PROCESSOR GROUPS
 OUTPUT [ALL, GROUPS, PARENT]
 PRINT LEVEL

TROTTER DIMENSION

The Trotter number P , i.e. the number of “replicas”, “beads”, or “time slices”, is read from the next line.

CENTROID DYNAMICS

Adiabatic centroid dynamics yielding quasiclassical canonical Boltzmann and Wigner density distributions in conjunction with NORMAL MODES and STAGING, respectively, see Refs. [47, 48]. FACMASS has to be unity in this case and the mass disparity FACSTAGE has to be chosen carefully.

CLASSICAL TEST

Test option to reduce the path integral code to the classical code for the special case $P = 1$. It works only with primitive propagator, i.e., not together with NORMAL MODES, STAGING, and / or DEBROGLIE CENTROID.

FACMASS

Obtain the fictitious nuclear masses for the path integral molecular dynamics from the real physical atomic masses (as tabulated in the DATA ATWT / ... / statement in atoms.F) by multiplying them with a factor that is read from the following line.

INITIALIZATION

Provide an initial configuration for all replicas as specified either by GENERATE REPLICAS or by READ REPLICAS. This option is automatically activated if RESTART COORDINATES is not specified. It is defaulted to GENERATE REPLICAS together with DEBROGLIE CENTROID and a temperature of 500 Kelvin.

GENERATE REPLICAS

Generate quantum free particle replicas from scratch given a classical input configuration according to the keyword DEBROGLIE specification. This is the default if INITIALIZATION is active.

DEBROGLIE [CENTROID]

An initial configuration assuming quantum free particle behavior is generated for each individual atom according to its physical mass at the temperature given in Kelvin on the following input line. Using DEBROGLIE each nuclear position obtained from the &ATOMS ... &END section serves as the starting point for a Gaussian Lévy walk of length P in three dimensions, see e.g. Ref. [45]. Using DEBROGLIE CENTROID each nuclear position obtained from the &ATOMS ... &END section serves as the centroid (center of geometry) for obtaining the P normal modes in three dimensions, see e.g. Ref. [46]. This option does only specify the generation of the initial configuration if INITIALIZATION and GENERATE REPLICAS are active. Default is DEBROGLIE CENTROID and 500 Kelvin.

READ REPLICAS

Read all P replicas from a file with a name to be specified in the following line, for the input format see subroutine rreadf.F.

STAGING

Use the staging representation [46] of the path integral and obtain a mass disparity by dividing the effective masses of the transformed $P > 1$ replicas by the adiabaticity control factor FACSTAGE that has to be read from the following line. Consequently the eigen-frequencies of the $P > 1$ replicas are changed by only $\sqrt{\text{FACSTAGE}}$, see Ref. [48](b).

NORMAL MODES

Use the normal mode representation [46] of the path integral and obtain a mass disparity by dividing the effective masses of the transformed $P > 1$ replicas by the adiabaticity control factor FACSTAGE that has to be read from the following line. Consequently the eigen-frequencies of the $P > 1$ replicas are changed by only $\sqrt{\text{FACSTAGE}}$, see Ref. [48](b).

PROCESSOR GROUPS

Separate the total number of processors into a certain number of processor groups that is read from the following line; only $2^N = 2, 4, 8, 16, \dots$ groups are allowed and the maximum number of groups is the number of replicas. Every processor group is headed by one PARENT and has several CHILDREN that work together on a single replica at one time; the processor groups work sequentially on replicas if there is more than one replica assigned to one processor group. The default optimizes the combined load balancing of the parallelization over replicas and g-vectors. Note: if the resulting number of processor groups is much smaller than the number of replicas (which occurs in “odd” cases) specifying the number of processor groups to be equal to the number of replicas might be more efficient. This keyword is only active in parallel mode.

OUTPUT [ALL, GROUPS, PARENT]

Output files for each processor, processor group, or only grandparent. Default is PARENT to standard output file (Note: some information such as messages for correct reading / writing of restart files is lost); GROUPS and ALL write to the files OUTPUT_n where n is the group and bead number, respectively.

PRINT LEVEL

The detail of printing information is read as an integer number from the next line. Currently there is only minimal output for < 5 and maximal output for ≥ 5 .

5.6 Section &ATOMS ... &END

In this section the positions of the atoms and the pseudopotentials are specified. (The different parts are read in the files `ratom.F` and `recpnew.f.`)

Structure and list of possible keywords :

<i>*ECPNAME</i> [<i>labels</i>]	<i>Pseudopotential file and options</i>
{ LMAX = <i>l</i> [LOC = <i>l</i> , SKIP = <i>l</i>]}	<i>Nonlocality options</i>
<i>NATOMS</i>	<i>Number of atoms of this species</i>
<i>X</i> <i>Y</i> <i>Z</i>	<i>Coordinates of atoms</i>
... 	
<i>X</i> <i>Y</i> <i>Z</i>	

ATOMIC CHARGES

ISOTOPE

MOVIE TYPE

GENERATE COORDINATES

CHANGE BONDS

DUMMY ATOMS

CONSTRAINTSEND CONSTRAINTS

VELOCITIES END VELOCITIES

The general format is:

- The input for a **new atom type** is started with a **"*"** in the **first column**. This line further contains:
 - the **file name** (*ECPNAME*) where to find the **pseudopotential** information starting in column 2
 - and several *labels*:
 - . The **first label** [**GAUSS-HERMIT, KLEINMAN-BYLANDER**] specifies the method to be used for the calculation of the **nonlocal parts** of the **pseudopotential**. It can be omitted for Vanderbilt pseudopotentials and Stefan Goedecker's pseudopotentials.
 - . It is further possible to specify **nonlinear core correction** [**NLCC**] and the width of the **ionic charge distribution** [**RAGGIO**]. (Default is **no NLCC** and the default value for **RAGGIO** is **1.2**.)

- . For **Vanderbilt pseudopotentials** three further options can be specified:

BINARY indicates the binary version of the output file from the atomic code.

NEWF is used in connection with binary and indicates that the pseudopotential file was generated with the new version of the atomic code.

TPSEU asks for an external pseudization of the localized charge. This needs some care and additional input files.

- The next line contains information on the **nonlocality** of the **pseudopotential** (*LMAX*, *LOC*, *SKIP*).
- On the following lines the **coordinates** for this **atomic species** have to be given.
The first line gives the number of atoms (*NATOMS*) of the current type.
Afterwards the coordinates of the atoms are listed (in cartesian coordinates by default).

The information on the **nonlocal part** of the pseudopotential can be given in two different styles:

- You can specify the maximum *l* - quantum number with "**LMAX=*l***" where *l* is S, P or D.
If this is the only input, the program assumes that LMAX is the local potential (LOC).
You can use another local function by specifying "**LOC=*l***".
In addition it is possible to assign the local potential to a further potential with "**SKIP=*l***".
- Alternatively you can specify these three angular quantum numbers by their numerical values (S=0, P=1, D=2) in the order "**LMAX LOC SKIP**".
If values for LOC and SKIP are provided outside the range 0 - LMAX the program uses the default.

Examples: The following lines are **equivalent**

LMAX=P

LMAX=P LOC=P

1 1 2

1 2 2

Note:

Also for Vanderbilt and Goedecker pseudopotentials this line has to be in a **valid format**, but the actual values are **not** used.

Further options:

(to be specified outside the definition range of the atomic species):

ISOTOPE

Changes the default **masses** of the atoms.

This keyword has to be followed by *NSP* lines (number of atom types).

In each line the new mass of the respective species has to be specified (in order of their definition).

ATOMIC CHARGES

Changes the default charge (0) of the atoms for the initial guess to the values read from the next line.

MOVIE TYPE

Assign special movie atom types to the species.

The types are read from the next line. Values from 0 to 5 are recognized by the MOVIE program.

GENERATE COORDINATES

The number of generator atoms for each species are read from the next line.

These atoms are used together with the point group information to generate all other atomic positions. The input still has to have entries for all atoms but their coordinates are overwritten. Also the total number of atoms per species has to be correct.

CHANGE BONDS

With this keyword the buildup of the empirical Hessian can be affected. You can either add or delete bonds. The number of changed bonds is read from the next line. This line is followed by the description of the bonds. The format is { *ATOM1* *ATOM2* *FLAG* } .

ATOM1 and *ATOM2* are the numbers of the atoms involved in the bond. A *FLAG* of -1 causes a bond to be deleted and a *FLAG* of 1 a bond to be added.

Example:

CHANGE BONDS

2

1 2 +1

6 8 -1

DUMMY ATOMS

The definition of dummy atoms follows this keyword.

Two different kinds of dummy atoms are implemented. Type 1 is fixed in space and type 2 lies at the arithmetic mean of the coordinates of real atoms.

The first line contains the total number of dummy atoms. The following lines start with the type label **TYPE1**, **TYPE2**.

For type 1 dummy atoms the label is followed by the cartesian coordinates.

For type 2 dummy atoms the first number specifies the total number of atoms involved in the definition of the dummy atom. Then the number of these atoms has to be specified on the same line.

Example:

```

DUMMY ATOMS
2
TYPE1  0.0  0.0  0.0
TYPE2  2    1    4

```

CONSTRAINTS ... END CONSTRAINTS

Within this option you can specify several **constraints** on the atoms. The following suboptions are possible:

FIX ATOMS

All coordinates of certain atoms can be fixed.

This keyword is followed by the number of atoms to be fixed and a list of these atoms specifying them by the number of their position.

FIX COORDINATES

Certain coordinates of atoms are fixed.

This keyword is followed by the number of atoms with fixed coordinates and a list of these atoms together with flags indicating which coordinates are fixed. A zero indicates a fixed coordinate.

Example:

```

FIX COORDINATES
2                Two atoms have fixed coordinates
1  1  1  0      for atom #1 z is fixed
4  0  1  0      for atom #4 x and z are fixed

```

FIX COM

Fix the center of mass

FIX STRUCTURE [SHOVE]

Whole **structural units** are fixed.

nfix

Number of constraints.

DIST $n1\ n2\ R$

Fixes the distance R between the atoms $n1$ and $n2$.

STRETCH $n1\ n2\ R$ Fixes R^2 defined by the atoms $n1$ and $n2$.

BEND $n1\ n2\ n3\ \theta$

Fixes the bending angle defined by the atoms $n1$, $n2$, and $n3$.

TORSION $n1\ n2\ n3\ n4\ \Theta$

Fixes the torsion angle defined by the atoms $n1$, $n2$, $n3$ and $n4$.

OUTP $n1\ n2\ n3\ n4\ \Theta$ "Out of Plane"; Angle between plane ($n1$, $n2$, $n3$) and atom $n4$ is fixed.

COORD $n1\ \kappa\ Rc\ C0$ "Coordination constraint" for atom $n1$.

The other parameters (κ and Rc) are explained in the paper by Sprik [51]. The function is constraint to the value $C0$.

RIGID $nr\ n1\ n2\ \dots\ nx$

Keeps the structure formed by the nr atoms $n1$, $n2$, ...

$n1$, ... are the atom numbers, R distances and Θ angles. A function value of -999. for R or Θ refers to the current value to be fixed. The constraint is linearly added to the CP Lagrangean according to the *Blue Moon* ensemble prescription [41]. The values of the Lagrange multipliers and of the actual constraint are printed in the file CONSTRAINT.

Note: In MD runs **only** the actual initial value (-999.) can be fixed. The **Shove** option requires an additional entry at the end of each constraint line. This entry has to be either -1 , 0 , or 1 . The constraint is then either fixed (0) or allowed to shrink (-1) or grow (1).

PENALTY

The weight factors for the penalty function for stretches, bends, and torsions are read from the next line.

VELOCITIES ... END VELOCITIES

Sets an **initial velocity** for specified atoms.

The first line is the number of specified atomic velocities followed by the same number of lines with:

the species number, the atom number in species entry and three velocity coordinates in atomic units.

5.7 Section &DFT ... &END

This section contains information about the **density functional** to be used.
(This part of the input is read in dftin.f)

Possible keywords:

OLDCODE/NEWCODE

FUNCTIONAL {LDA, BONLY, BP, BLYP, GGA, PBE, REVPBE, HCTH}

[NO] EXCHANGE CORRELATION TABLE

SLATER

LDA CORRELATION functionals

Correlation functionals: PZ, VWN, LYP, PW, HCTH

GRADIENT CORRECTION functionals

Gradient correction functionals: BECKE88, GGAX, PERDEW86, PBEX,
REVPBEX, LYP, GGAC, PBEC, REVPBEC,
HCTH

GC-CUTOFF

SMOOTH

BECKE BETA

Warning: Not all combinations of functionals for exchange and correlation and gradient corrections make sense.

Warning: The functionals GGAC and GGAX are **not** tested.

OLDCODE, NEWCODE

Switch to select one out of two versions of code to calculate exchange-correlation functionals.

Not all functionals are available with NEWCODE, if you select one of these, the OLDCODE is used. The NEWCODE is highly recommended for all new projects and especially for vector computers.

(NEWCODE is now the default.)

FUNCTIONAL functionals

Single keyword for setting up XC-functionals.

Available functionals are LDA, BONLY, BP, BLYP, GGA, PBE, REVPBE and HCTH.

EXCHANGE CORRELATION TABLE [NO]

Specifies the range and the granularity of the lookup table for the local exchange-correlation energy and potential.

The number of table entries and the maximum density have to be given

on the next line.

The default values are 30000 and 2.0.

SLATER [NO]

The α value for the Slater exchange functional [16] is read from the next line. With NO the exchange functional is switched off.

Default is a value of 2/3.

This option together with no correlation functional, allows for $X\alpha$ theory.

LDA CORRELATION [functional]

The LDA correlation functional is specified.

Possible functionals are NO (no correlation functional), PZ [17], VWN [18], LYP [19], and PW [20].

Default is the PZ, the Perdew and Zunger fit to the data of Ceperley and Alder [21].

GRADIENT CORRECTION [functionals]

A gradient corrected functional is to be used.

Functionals implemented are for the exchange energy :

BECKE88 [22], GGAX [23] PBEX [24], REVPBEX [25], HCTH [26] and for the correlation part:

PERDEW86 [27], LYP [19], GGAC [23], PBEC [24], REVPBEC [25], HCTH [26].

Note that for HCTH, exchange and correlation are treated as a unique functional.

The keywords EXCHANGE and CORRELATION can be used for the default functionals (currently BECKE88 and PERDEW86). If no functionals are specified the default functionals for exchange and correlation are used.

GC-CUTOFF

On the next line the density cutoff has to be specified. The default value is 10^{-8} . Experience showed that for Vanderbilt pseudopotentials a large value has to be used. A reasonable value is $5 \cdot 10^{-5}$.

Warning: for the HCTH functional, since it includes both the xc part and the gradient correction in a unique functional, a too high GC-CUTOFF (e.g. $\geq 5 \cdot 10^{-5}$) could result in not including any xc part with uncontrolled related consequences.

SMOOTH

A smoothening function is applied to the density [28].

The function is of the Fermi type.

$$f(G) = \frac{1}{1 + e^{\frac{G - G_{cut}}{\Delta}}}$$

G is the wavevector, $G_{cut} = \alpha G_{max}$ and $\Delta = \beta G_{max}$. Values for α and β have to be given on the next line.

BECKE BETA

Change the β parameter in Becke's exchange functional to the value given on the next line.

5.8 Section &PROP ...&END

This section contains the information related to the calculation of properties. If the keyword PROPERTIES in the &CPMD ...&END is given this section will be read (in proppt.F).

Possible keywords:

PROJECT WAVEFUNCTIONS
POPULATION ANALYSIS [MULLIKEN, DAVIDSON] [n-CENTER]
n-CENTER CUTOFF
CHARGES
LOCALIZE
NOPRINT [ORBITALS]
DIPOLE MOMENT
LOCAL DIPOLE
EXCITED DIPOLE

PROJECT WAVEFUNCTION

The wavefunctions are projected on atomic orbitals.

The projected wavefunctions are then used to calculate atomic populations and bond orders. The atomic orbitals have to be specified in the file *ATWFN*, which is found in the pseudo-potential library directory.

POPULATION ANALYSIS [MULLIKEN, DAVIDSON] [n-CENTER]

The type of population analysis that is performed with the projected wavefunctions.

Lowdin charges are given with both options. For the Davidson analysis [34] the maximum complexity can be specified with the keyword **n-CENTER**.

Default for n is 2, terms up to 4 are programmed. For the Davidson option one has to specify the number of atomic orbitals that are used in the analysis. For each species one has to give this number on the next line.

n-CENTER CUTOFF

The cutoff for printing the n-center shared electron numbers is read from the next line. All one and two center terms are printed.

CHARGES

Calculate atomic charges.

Charges according to the method of Hirshfeld [37] and charges derived from the electrostatic potential are calculated [38].

LOCALIZE

Localize the molecular orbitals as defined through the atomic basis set. The same localization transformation is then applied also to the wavefunctions in the plane wave basis. These wavefunction can be printed with the keyword **RHOOUT** specified in the section `&CPMD ...&END`.

NOPRINT [ORBITALS]

Do not print the wavefunctions in the atomic basis set.

DIPOLE MOMENT

Calculate the dipole moment.

This is only implemented for simple cubic and fcc supercells.

LOCAL DIPOLE

Calculate *numloc* local dipole moments.

numloc is read from the next line followed by two *numloc* lines with the format:

```
xmin ymin zmin  
xmax ymax zmax
```

EXCITED DIPOLE

Calculate the difference of dipole moments between the ground state density and a density generated by differently occupied Kohn-Sham orbitals.

On the next line the number of dipole moments to calculate and the total number orbitals has to be given. On the following lines the occupation of the states for each calculation has to be given. By default the dipoles are calculated by the method used for the **DIPOLE MOMENT** option and the same restrictions apply. If the **LOCAL DIPOLE** option is specified the dipole moment differences are calculated within the same boxes.

5.9 Section &BASIS ... &END

This section specifies the atomic basis used in the properties section or as an initial guess. (For details see setbasis.f)

There have to be *number of species* different entries in this section.

The order of the basis sets has to correspond with the order of the atom types in the section **&ATOMS ... &END**.

With the keyword **SKIP** the species is skipped and the default minimal Slater function basis is used.

Basis sets are either specified as Slater functions or given on an additional input file.

The respective input formats are given below:

Slater type basis

```
SLATER  nshell [OCCUPATION]
  n1  l1  exp1
  ..  ..  ....
  nx  lx  expx
  [f1 f2 ... ]
```

Pseudo atomic orbitals

```
PSEUDO A0 nshell [OCCUPATION]
  l1  l2  ..  lx      !a function with l=-1 is skipped
  [f1 f2 ... ]
```

Numerical functions

```
*filename nshell FORMAT=n [OCCUPATION]
  l1  l2  ..  lx
  [f1 f2 ... ]
```

Gaussian basis functions

```
*filename nshell GAUSSIAN [OCCUPATION]
  l1  l2  ..  lx
  [f1 f2 ... ]
```

Skip atom type and use default minimal slater function

```
SKIP
```

`nshell` is the number L-values `l1 l2 ... lx` to be used.
`[f1 f2 ...]` is their occupation.

The format **PSEUDO AO** refers to the &WAVEFUNCTION section on the corresponding pseudopotential file.

With a L-value of -1 a specific function can be skipped.

The * for the numerical basis has to be in the first column. The default format is 1, other possible formats are 2 and 3. The numbers correspond to the format numbers in the old pseudopotential definitions for the atomic wavefunctions.

The format **GAUSSIAN** allows to use any linear combination of Gaussian functions. The format of the file is as follows:

```

Comment line
Lmax
(for each l value)
  Comment line
  # of functions; # of exponents
  exp1 exp2 ... expn
  c11 c21      cn1
  c12 c22      cn2
  ...  ...      ...
  c1m c2m      cnm

```

6 Pseudopotentials

See the directory `/boero/pseudo` for the available pseudopotentials at JRCAT and refer to the READ.ME file for details.

6.1 Norm-conserving Pseudopotentials: Atomic Code for PP generation

An atomic program, derived from the original code of D. H. Hamann, can be found in `/boero/pseudo/ppgen/source` (Fortran source) and `/boero/pseudo/ppgen/atomc.x` (*executable*). It is able to handle the following functionals and GGA

- LDA correlation : Ceperley-Alder, Vosko-Wilk-Nusair, Lee-Yang-Parr, Perdew et al (1992), Wigner, Hedin-Lundqvist, Ortiz-Ballone, Hamprecht-Cohen-Tozer-Handy
- Exchange GC : Becke (1988), Perdew-Wang, Perdew-Burke-Ernzerhof (1996), revPBE after Zhang-Yang (1998), Hamprecht-Cohen-Tozer-Handy (1998)
- Correlation GC : Perdew (1986), Perdew-Wang, Lee-Yang-Parr, Perdew-Burke-Ernzerhof (1996), Hamprecht-Cohen-Tozer-Handy (1998)

The atomic code can generate pseudopotentials of the type

- Bachelet-Hamann-Schluter
- Hamann-Schluter-Chang
- Martins-Trouiller

with and without Non Linear Core Corrections (NLCC). It requires a simple input file in which each instruction is included between `&ATOM ... &END` like in a single section of the CPMD code. The following example refers to the case of Co and is intended only as a guide-line. Here we ask the program to generate a relativistic Martins-Trouiller PP using LDA in the Perdew-Zunger parametrization + Becke and Perdew Gradient Corrections. NLCC are included and the corresponding cut-off radius is specified in the last line as *rccore 1.20*. The 3 lines above refer instead to the cut-off radii of the pseudo-wavefunctions. Further details can be understood by looking at the example itself in `/boero/pseudo/ppgen/test` and at the source code.

```
&ATOM
atom=Co
core=5
valence=2
relativistic
lda = PZ
gcx = BECKE
gcc = PERDEW
states
1 S 2.
```

```

2 S 2.
2 P 6.
3 S 2.
3 P 6.
3 D 7.00
4 S 2.00
MARTINS
NLCC
channels 3
0 1.9d0 0.d0
1 2.0d0 0.d0
2 1.6d0 0.d0
rccore 1.20
&END

```

To use this program you have to log-in on the DEC (jrc4), since the executable version is compiled for this machine. There is no need to use VPP or SP2. However, if you want to use the code on these machines, copy the source code and the Makefile on your account and comment/uncomment the appropriate sections of the Makefile. To run *atomc.x*, simply write an input file like in the afore reported example (e.g Co.in) and then type

```
atomc.x Co.in > Co.out
```

In *Co.out* the standard output is dumped out. The pseudopotential, in the standard CPMD format will be written in a file whose name, in capital letters, corresponds to the related chemical element, i.e. *CO* in this case.

6.2 Vanderbilt ultra-soft Pseudopotentials: Atomic Code

An atomic code for the generation of ultrasoft pseudopotentials, derived from the original code of D. Vanderbilt and D. King-Smith is available in */boero/pseudo/ppvan/* under the name **atomv.x**. The source code and the relevant Makefile can be found in the subdirectory

```
/boero/pseudo/ppvan/source
```

while examples for H, O and Co are stored in the twin directory

```
/boero/pseudo/ppvan/test.
```

A general script command to run this atomic code (on JRC0) is indicated with the extension *.sh* (e.g. *H.sh*, *O.sh*, *Co.sh*) and you can refer to these examples as well as to the README file in the */source* directory. Here below the main informations are listed as indicated in the README.

- first input line -

ifae : 0 = read all-electron (ae) data file , 1 = do ae calculation from scratch and write ae data file

ifpsp : 0 stop after ae part , 1 read pseudo data file and compare to ae , 2

generate pseudopotential and compare to ae

ifprt : -3 no prints or graphs , -2 prints but no graphs , -1 more or less standard , 1 everything

ifplw : if generate data file for wf plots

ilogd : num l values for which log derivatives are calculated and output

- second input line -

rlogd : radius at which log derivatives are calculated **emin**, **emax**, **nnt** : min and max energy interval and (nnt) number of mesh points

All Electron Case:

z, **xion** : nuclear charge and net charge on atom

e1, **e2**, **e3**, **e4** : type of exchange and correlation according to the values tabulated in the README file. E.g. 1 3 1 2 refers to BLYP)

rmax , **aasf**, **bbsf** : parameters for generation of the logarithmic mesh

ncspvs,irel : number of core states + valence states and relativistic/non relativistic option. In the present version irel = 0 as no relativistic code available

nnlz(i), **wwnl(i)**, **ee(i)** : quantum numbers, occupancies and trial energy levels

Pseudopotential Case:

ncores, **nvaes**, **nang** : number of core states, number of valence states and maximum angular momentum of the beta functions

keys, **ifpcor**, **rinner** : pseudopotential specifications, i.e.

keys = 0 → standard hsc pseudopotential with exponent 4.0,

keys = 1 → standard hsc pseudopotential with exponent 3.5,

keys = 2 → vanderbilt modifications using defaults,

keys = 3 → new generalized eigenvalue pseudopotentials,

keys = 4 → frozen core all-electron case

ifpcor = 1 for Non-Linear Core Corrections (this option has never been tested on the Vanderbilt pseudopotentials !)

rinner = radius at which to cut off partial core or q_{ij}

nbeta, **rcloc** : number of beta's and core radius used to pseudize the local potential

rc(i) : cut-off radii for pseudo generation

ll, **keyee**, **eeread**, **ipread** : angular momentum l , reference energy levels, pseudisation type (ipread), i.e. 0 = polynomial 1 = exponential

lloc, **eloc** : lloc = -1 original pseudization scheme, lloc = +ve get log-derivative correct in channel lloc at the energy specified by eloc

ifqopt, **nqf**, **qtryc** : ifqopt = 0 → original polynomial pseudization, ifqopt = 1 → new scheme with taylor expansion of nqf terms optimising for cutoff wavevector qtryc (not recommended if you are not an expert)

Beside the standard output on the monitor, the program produces 3 output files: `out.van` in which all the infos relevant to the calculation are printed, `pseudo.van` containing IN THE MACHINE format (like the RESTART files of CPMD) the pseudopotential and **ppcpmd.van** the Vanderbilt pseudopotential in the correct format to be used with CPMD. This file must be renamed to avoid possible overwriting when `atomv.x` is used again. The `ppcpmd.van` file is ASCII formatted and can be transferred on each platform.

7 Input Examples

```

! Wavefunction optimization using default setting
!
&CPMD
  OPTIMIZE WAVEFUNCTION
&END
&SYSTEM
  SYMMETRY
    1
  CELL
    10.2612  1.0    1.0    0.0 0.0 0.0
  CUTOFF
    13.
&END
&ATOMS
*SI_SGS  KLEINMAN-BYLANDER
  LMAX=P
    8
          .00000      .00000      .00000      1
          .00000      5.13000      5.13000      1
          5.13000      .00000      5.13000      1
          5.13000      5.13000      .00000      1
          2.56500      2.56500      2.56500      1
          2.56500      7.69500      7.69500      1
          7.69500      2.56500      7.69500      1
          7.69500      7.69500      2.56500      1
&END

! Restart for wavefunction optimization of test_0001
! Increasing the cutoff to 18 Ry
! Storing the wavefunction in 32 bit precision
!
&CPMD
  OPTIMIZE WAVEFUNCTION
  RESTART WAVEFUNCTION LATEST
  COMPRESS WRITE32
&END

&SYSTEM
  SYMMETRY
    1
  CELL
    10.2612  1.0    1.0    0.0 0.0 0.0
  CUTOFF
    18.
&END

! Restart wavefunction optimization from test_0004
! use preconditioned conjugate gradients

```

```
! store wavefunction by projection on atomic functions
!   (use the default setting, minimal STO basis)
! randomize wavefunction to get orthogonalization invoked
!
&CPMD
  OPTIMIZE WAVEFUNCTION
  RESTART WAVEFUNCTION LATEST
  RANDOMIZE WAVEFUNCTION
    1.D-5
  PCG
  COMPRESS WRITEAO
&END

! Optimize wavefunction starting from random guess
! Store the final density
! use a reduced accuracy for the density (DUAL)
! Print additional information every 10 steps
! store the wavefunction every 20 steps
!
&CPMD
  OPTIMIZE WAVEFUNCTION
  INITIALIZE WAVEFUNCTION RANDOM
  DUAL
    3
  PRINT INFO
    10
  STORE
    20
&END

! Wavefunction optimization with LSD
! Print electron density
! Give bond angles and bond length
!
&CPMD
  OPTIMIZE WAVEFUNCTION
  LSD
  RHOOOUT
  STRUCTURE BONDS ANGLES
&END

&SYSTEM
  SYMMETRY
    1
  CELL
    18.00 1.0 1.0 0.0 0.0 0.0
  CUTOFF
    12.
  MULTIPLICITY
    3
&END
```



```
! Wavefunction optimization with LSD
! Use isolated system option
!
&CPMD
  OPTIMIZE WAVEFUNCTION
  LSD
&END

&SYSTEM
  SYMMETRY
  0
  CELL
  18.00 1.0 1.0 0.0 0.0 0.0
  CUTOFF
  12.
  MULTIPLICITY
  3
&END

! use gradient corrections (Becke-Perdew)
!
&DFT
  GRADIENT CORRECTIONS BECKE88 PERDEW86
&END

! use gradient corrections (Becke-LYP)
! set cutoff for density in GC part to 1.D-7
!
&DFT
  LDA CORRELATION LYP
  GRADIENT CORRECTIONS BECKE88 LYP
  GC-CUTOFF
  1.D-7
&END

! Wavefunction optimization
! Vanderbilt PP
! LOWDIN ORTHOGONALIZATION
!
&CPMD
  OPTIMIZE WAVEFUNCTION
  LOWDIN ORTHOGONALIZATION
&END

&DFT
  GRADIENT CORRECTIONS BECKE88 PERDEW86
  GC-CUTOFF
  5.D-5
&END
&SYSTEM
```

```
SYMMETRY
  1
CELL
  12.00  1.0  1.0    0.0 0.0 0.0
CUTOFF
  25.
&END
&ATOMS
*O_VDB_BP_140_100_135
  0 0 0
  1
  0.0  0.0  0.0
*H_VDB_BP_080_060_075
  0 0 0
  2
  1.188  0.000  1.447
  1.188  0.000 -1.447
&END

! Wavefunction optimization
! Vanderbilt PP
! Spline interpolation of Q-functions
!
&CPMD
  OPTIMIZE WAVEFUNCTION
  SPLINE QFUNCTIONS POINTS
  512
&END

! Geometry optimization
! Hessian DISCO
! GDIIS/DIIS method
! Change default convergence criteria
! Give bond angles and bond length
!
&CPMD
  OPTIMIZE GEOMETRY
  HESSIAN DISCO
  GDIIS
  5
  ODIIS
  6
  CONVERGENCE
  5.D-5 5.D-3
  STRUCTURE BONDS ANGLES DIHEDRALS
&END

! Geometry optimization
```

```
! Restart coordinates and wavefunctions
! Use old Hessian
! Use BFGS method
! Switch on gradient corrections
! Change default convergence criteria
! Give bond angles and bond length
!
&CPMD
  OPTIMIZE GEOMETRY
  RESTART WAVEFUNCTION COORDINATES HESSIAN LATEST
  BFGS
  CONVERGENCE
  1.D-5 1.D-3
  STRUCTURE BONDS ANGLES DIHEDRALS
&END

! Geometry optimization
! Restart coordinates
! Setup an unit Hessian
! Use Vanderbilt PP
! RFO method
! Give bond angles and bond length
!
&CPMD
  OPTIMIZE GEOMETRY
  RESTART COORDINATES LATEST
  HESSIAN UNIT
  RFO
  CONVERGENCE
  1.D-5 1.D-4
  STRUCTURE BONDS ANGLES DIHEDRALS
&END

! Wavefunction optimization
! Kohn-Sham Energies
!
&CPMD
  OPTIMIZE WAVEFUNCTION
  KOHN-SHAM ENERGIES
  0
&END

! Kohn-Sham Energies
! Restart wavefunction
!
&CPMD
  KOHN-SHAM ENERGIES
  4
&END

! Kohn-Sham Energies
! Restart calculation
```

```
!  
&CPMD  
  KOHN-SHAM ENERGIES  
  4  
  RESTART POTENTIAL WAVEFUNCTION COORDINATES LATEST  
&END  
  
! Wavefunction optimization  
! LSD  
! Kohn-Sham Energies  
!  
&CPMD  
  OPTIMIZE WAVEFUNCTION  
  LSD  
  KOHN-SHAM ENERGIES  
  4  
&END  
  
! LSD  
! Kohn-Sham Energies  
! Restart for higher accuracy  
!  
&CPMD  
  KOHN-SHAM ENERGIES  
  4  
  LSD  
  RESTART WAVEFUNCTION COORDINATES LATEST  
  DAVIDSON PARAMETER  
  4 5.D-6  
&END  
  
! Molecular dynamics  
! use of annealing to increase the temperature  
! initial temperature set to 30 Kelvin  
!  
&CPMD  
  MOLECULAR DYNAMICS  
  RESTART WAVEFUNCTION LATEST  
  ANNEALING  
  1.03  
  TEMPERATURE  
  30  
  MAXSTEP  
  100  
  TRAJECTORY OFF  
  EMASS  
  400  
  TIMESTEP  
  5  
&END
```

```
! Molecular dynamics
! use of Nose thermostats to equilibrate the system
!
&CPMD
  MOLECULAR DYNAMICS
  RESTART WAVEFUNCTION COORDINATES VELOCITIES LATEST
  TRAJECTORY SAMPLE
    10
  NOSE IONS
    300 2000
  NOSE ELECTRONS
    0.0001 6000
  NOSE PARAMETERS
    8 8 4 40 125 1
  MAXSTEP
    200
  EMASS
    400
  TIMESTEP
    5
&END

! Molecular dynamics
! Restart of the full calculation
! use of Nose thermostats to equilibrate the system
!
&CPMD
  MOLECULAR DYNAMICS
  RESTART WAVEFUNCTION COORDINATES VELOCITIES LATEST
  RESTART NOSEE NOSEP ACCUMULATORS
  TRAJECTORY SAMPLE
    10
  NOSE IONS
    300 2000
  NOSE ELECTRONS
    0.0001 6000
  NOSE PARAMETERS
    4 4 4 6 15 2
  MAXSTEP
    200
  EMASS
    400
  TIMESTEP
    5
&END

! Molecular dynamics
! Restart
! quench electrons
! use scaled masses and harmonic reference system
!
&CPMD
```

```
MOLECULAR DYNAMICS
RESTART WAVEFUNCTION COORDINATES VELOCITIES LATEST
TRAJECTORY SAMPLE
  10
QUENCH ELECTRONS
SCALED MASSES
HARMONIC REFERENCE SYSTEM
HAMILTONIAN CUTOFF
  2.
MAXSTEP
  100
EMASS
  500
TIMESTEP
  12
&END

! Molecular Dynamics
! Restart Wavefunctions Coordinates Velocities
! Stress Tensor calculation
! Big memory option
&CPMD
  MOLECULAR DYNAMICS
  RESTART WAVEFUNCTIONS COORDINATES VELOCITIES LATEST
  BIG MEMORY
  MAXSTEP
    100
  STRESS TENSOR
    2
  EMASS
    300
  TIMESTEP
    4
&END

! Molecular Dynamics
! Restart Wavefunctions Coordinates Velocities
! Vanderbilt PP, BLYP functional
! Use constraint nonorthogonal orbitals
! Big memory option
&CPMD
  MOLECULAR DYNAMICS
  NONORTHOGONAL ORBITALS
    0.0001
  RESTART WAVEFUNCTIONS COORDINATES VELOCITIES LATEST
  BIG MEMORY
  MAXSTEP
    100
  EMASS
    300
```

```
TIMESTEP
  4
&END

! Molecular Dynamics
! Restart Wavefunctions Coordinates
! Randomize Coordinates and Quench ions
! Reoptimize wavefunctions
! Vanderbilt PP, BLYP functional
! Write Restart file every 20 steps
! Use 2 Restart files
! Big memory option
&CPMD
  MOLECULAR DYNAMICS
  RESTART WAVEFUNCTIONS COORDINATES LATEST
  BIG MEMORY
  QUENCH IONS
  QUENCH BO
  RANDOMIZE COORDINATES
    0.1
  STORE
    20
  RESTFILE
    2
  MAXSTEP
    100
  EMASS
    300
  TIMESTEP
    4
&END

! Constraints
!
&ATOMS
CONSTRAINTS
  FIX STRUCTURE
    1
  DIST 1 2 -999.
END CONSTRAINTS
.....
&END

!
! Free energy calculation
! Monkhorst-Pack mesh 20x20x20
! per Block of 50 k points (in memory).
! Two different Anderson parameter in function of DRHOMAX
! Two different tolerance for Lanczos diagonalisation.
! Start with pseudowavefunction (very good start)
```

```
&CPMD
  OPTIMIZE WAVEFUNCTION
  FREE ENERGY FUNCTIONAL
  RESTART DENSITY WAVEFUNCTIONS
  LANCZOS DIAGONALISATION
  LANCZOS PARAMETERS N=10
    50      8      9      1.D-8
    0.05    1.D-9
    0.03    1.D-10
    0.01    1.D-11
    0.009   1.D-12
    0.008   1.D-13
    0.007   1.D-14
    0.004   1.D-15
    0.002   1.D-16
    0.0012  1.D-17
  TROTTER FACTOR
    0.001
  BOGOLIUBOV CORRECTION
  CONVERGENCE
    1.D-3   5.D-4
  TEMPERATURE ELECTRON
    300.DO
  ANDERSON MIXING N=1
    1.000 0.4
    0.020 0.6
    0.
    100
  ENERGYBANDS
&END
&SYSTEM
  SYMMETRY
    BCC
  POINT GROUP
    32
  SCALE
  TESR
    10
  ANGSTROM
  CELL
    3.30  1.0  1.0          0.0  0.0  0.0
  CUTOFF
    63.
  DENSITY CUTOFF ENERGY
    6820
  KPOINTS MONKHORST-PACK BLOCK=50
    20 20 20
  STATES
    9
&END
&ATOMS
```



```
*NB_TM.2.20 KLEINMAN-BYLANDER
  LMAX=D LOC=P
  1
  0.00D0      0.00D0      0.00D0
&END

&BASIS
  PSEUDO ATOMIC ORBITALS
    PSEUDO AO 3      OCCUPATION
    0      1      2
    1.45   0.10   3.45
&END
```

8 Visualization

8.1 Molecular Dynamics

The trajectory of a molecular dynamics run can be saved in a special format by adding the keyword **MOVIE** in the `&CPMD` section of the input file. After adding a special header to the file, the trajectory can be visualized with the MOVIE program [15].

8.2 Contour and 3-D Plots

The program *grid.x* reads the files *DENSITY.xx*, *ELF*, *ELPOT* and produces outputfiles of different formats. The program is called with the command

```
grid.x DENSITY.x1 DENSITY.x2 .. < input
```

The format of the input file is

```
&GRID options &END
```

VERSION The version of the CPMD code you are using is read from the next line. The current version is 3.0f, for earlier versions try 3.0 first. If this does not work try 3.0d or 3.0e. For versions before 3.0 you should use the old grid programs.

PLANES,MOVIE,GAUSSIAN One of these keywords is compulsory.

PLANES asks for contour plots (old 2DGRID program), *MOVIE* generates 3-D plot information for the movie program and *GAUSSIAN* generates for the same information .pdb and Gaussian .cube files that can be processed by many graphics programs.

CONTOURS (for PLANES option) the number of contours is read from the next line.

NSPLINE (for PLANES option) the number of interpolation points in x and y direction is read from the next line.

NPLANES (for PLANES option) the number of planes for contour plots is read from the next line. For each plane you have to specify then coordinates:

1. x,y,z coordinates of the lower left corner,
2. x,y,z coordinates of the vector connecting the lower left with the lower right corner and
3. x,y,z coordinates of the vector connecting the lower left with the upper left corner.

After each (except the last) set of coordinates there should be a separator line (write only one string).

GRID (for PLANES option) The number of grid points in x and y direction is read from the next line. For best results the number of grid points should be compatible with the number of points used in the Fourier grid.

- PCOLOR** (for PLANES option) To get color output
- PATOMS** (for PLANES option) To get atoms plotted
- PBONDS** (for PLANES option) To get bonds plotted
- PLOG** (for PLANES option) To get a logarithmic plot
- GNUPLOT** (for PLANES option) you get for each plane an additional file with suffix '.a' (, '.b' ...) in the grid format. You get with gnuplot a 3-dim plot plus the contour by typing
- ```
set contour
splot 'densityfilename'
```
- BOX** (for MOVIE/GAUSSIAN option) followed by the coordinates of the box on the next four lines:
1. x,y,z coordinates of the lower left corner of the box,
  2. x,y,z coordinates of the vector connecting the lower left with the lower right corner (defining x),
  3. x,y,z coordinates of the vector connecting the lower left with the lower left corner (defining y) and
  4. x,y,z coordinates of the vector connecting the lower left with the upper left corner (defining z).
- GRID** (for MOVIE/GAUSSIAN option) The number of grid points in x, y, and z direction is read from the next line. For best results the number of grid points should be compatible with the number of points used in the Fourier grid.

## 9 Hints and Tricks

### 9.1 Wavefunction Initialization

The default initial guess for the wavefunctions is calculated from the atomic pseudo-wavefunctions and gives usually excellent results. Good results can also be obtained by using wavefunctions from other calculations with a different cutoff or slightly different geometry. The other initialization available starts by randomising the coefficients and should only be used as a last resort. Cases where the default method does not work are when the molecule has less occupied states than one of the atoms (in this case add some empty states for the molecule) or when the additional memory required for the atomic calculation is not available.

### 9.2 Geometry Optimization

Any combination of methods for geometry optimization and wavefunction optimization is allowed. Possible options for geometry optimization are GDIIS,

RFO, BFGS, and steepest descent. If you choose steepest descent for both, geometry variables and the wavefunction, a combined method is used. For all other combinations a full wavefunction optimization is performed between changes of the ionic coordinates. The default options are GDIIS and ODIIS. The quasi-Newton methods (GDIIS, RFO, and BFGS) are using the BFGS method to update an approximate Hessian. At the beginning of a run the Hessian can either be initialized as a unit matrix **HESSIAN UNIT** or with an empirical force field. Two force fields are implemented: The **DISCO** and the **SCHLEGEL** force field. The algorithm for the empirical force fields has to identify bonds in the system. For unusual geometries this may fail and the Hessian becomes singular. To prevent this you can add or delete bonds with the keyword **CHANGE BONDS**.

## 9.3 Molecular Dynamics

### 9.3.1 Choosing the Nosé-Hoover chain thermostat parameters

The Nosé-Hoover chain thermostat is defined by specifying three parameters: A target kinetic energy, a frequency, and a chain length. For the ions, given the target temperature  $T_W$ , the target kinetic energy is just  $gkT_W$ , where  $g$  is the number of degrees of freedom involved in a common thermostat. For example, if there is one thermostat on the entire ionic system, then  $g = 3N_{AT} - N_{const}$ , where  $N_{const}$  is the number of constraints to which the atoms are subject. The frequency for the ionic thermostat should be chosen to be some characteristic frequency of the ionic system for which one wishes to insure equilibration. In water, for example, one could choose the O-H bond vibrational frequency. (Having a precise value for this frequency is not important, as one only wishes to insure that the thermostat will couple to the mode of interest.) The choice of chain length is not terribly important as it only determines how many extra thermostats there will be to absorb energy from the system. Usually a chain length of 4 is sufficient to insure effective equilibration. Longer chains may be used in situations where heating or cooling effects are more dramatic.

For the electrons, the target kinetic energy is not usually known *a priori* as it is for the ions. However, by performing a short run without thermostats, one can determine a value about which the electron kinetic energy ‘naturally’ fluctuates and take this as the target value. While the precise value is not important, a little experience goes a long way, as a choice that is either too small or too large can cause spurious damping of the ions or departures from the Born-Oppenheimer surface, respectively. A good choice for the frequency of the electron thermostat can be made based on  $\Omega_I^{\max}$ , the maximum frequency in the phonon spectrum. The frequency of the electron thermostat should be at least 2-3 times this value to avoid coupling between the ions and

the electron thermostats. As an example, for silicon, the highest frequency in the phonon spectrum is 0.003 a.u., so a good choice for the electron thermostat frequency is 0.01 a.u. The chain length of the electron thermostat can be chosen in the same way as for the ions. 4 is usually sufficient, however longer chains may be used if serious heating is expected. In addition, the electron thermostats have an extra parameter that scales the number of dynamical degrees of freedom for the electrons. ( $1/\beta_e = 2E_e/N_e$ , where  $E_e$  is the desired electron kinetic energy and  $N_e$  is the number of dynamical degrees of freedom for the electrons – see Eq. (3.4) in Ref. [3]). The default value is the true number of dynamical degrees of freedom  $N_e = (2 * N_{GW} - 1) * N_{ST} - N_{ST}^p$ , where  $p = 2$  for orthonormality constraints and  $p = 1$  for norm constraints. When this number is very large, it may not be possible to integrate the electron chain thermostats stably using a frequency above that top of the phonon spectrum. Should this be the case in your problem, then the number of dynamical degrees of freedom should be scaled to some smaller number such that the system can once again be integrated stably. This parameter has no other effect than to change the relative time scales between the first element of the electron thermostat chain and the other elements of the chain.

In addition to the basic parameters defining the chains themselves, one needs to specify two more parameters related to the integration of the thermostated equations of motion. The first is the order  $M_{SUZ}$  of the Suzuki integrator. Experience shows that the choice  $M_{SUZ} = 3$  is sufficient for most applications. Finally, one must specify the number of times the Suzuki integrator will be applied in a given update. This is the parameter  $N_{SUZ}$  which determines the basic Suzuki time step  $\delta t = \Delta t/N_{SUZ}$ , where  $\Delta t$  is the time step being used in the MD run.  $N_{SUZ} = 2$  or  $3$  is usually large enough to give stable integration. If more stable integration is required, try  $M_{SUZ} = 4$  or make  $N_{SUZ}$  larger.

## 9.4 Hints of using FEMD

There are several parameters which crucially affect the speed, accuracy and robustness of the FEMD method. These are related to: LANCZOS PARAMETERS, STATES and ANDERSON MIXING. Less crucially, the ELECTRON TEMPERATURE.

### 9.4.1 Lanczos Parameters

Several parameters related to the Lanczos ( Friesner-Pollard) method are given. Generically:

```
LANCZOS PARAMETER [N=n]
 ncycle nkrylov nblock tolerance
 drhomax(2) tolerance(n)
```

```


 drhomax(n) tolerance(n)

```

*Ncycle* can always be safely set to 50. Similarly, *Nkrylov* = 8 is almost always a good choice. Exceptionally, for certain d-metallic systems, increasing *nkrylov* = 16 may be more efficient. *Nblock* is the dimension of the blocking in the evaluation of  $H[\psi_1, \dots, \psi_{nblock}]$ . *Nblock* should be a divisor of *NSTATE*, and recommended values lie in the range of 20-100. The *tolerance* specifies the accuracy to be achieved in the Lanczos method. States are considered converged if

$$|H\psi - \epsilon\psi|^2 < tolerance \quad (1)$$

For efficient calculations, the tolerance should vary according to closeness to self-consistency ( as measured by DRHOMAX). During initial stages of the SC cycle, the tolerance can be loose, gradually tightening until close to SC it is high. An example of this might be:

```

LANCZOS PARAMETER N=5
 50 8 20 1.D-9
 0.05 1.D-11
 0.01 1.D-13
 0.0025 1.D-16
 0.001 1.D-18

```

For accurate forces, a final tolerance of at least 1.D-16 is recommended, although accurate energies can be got using a lower tolerance. It is worth experimenting how best to tighten the tolerance - it could be system dependent.

#### 9.4.2 Other important FEMD parameters

The keyword STATES defines the dimension of the subspace used in the diagonalisation. STATES must be greater than or equal to  $N_{el}/2$ , but it is generally good to allow for a number of more or less empty bands (usually 10% or so). Finally, ANDERSON MIXING determines the rate of convergence to self-consistency. Properly chosen the convergence can be very fast. Typically for bulk systems we use values between 0.2-0.5, smaller values being necessary for large systems. For metallic surfaces, small values are necessary ( typically 0.03-0.05).

If using k-points, then it is usually a good idea ( and this is done by default if using MONKHORST PACK k-points) to exploit symmetries. In this case, however, beware of including the POINT GROUP keyword to symmetrise properly the density. Finally, if starting from a high-symmetry structure,

you may nevertheless want to use the full k-point mesh ( apart from time-inversion symmetry related k-points), and in this case specify the keyword FULL.

## References

- [1] R. Car and M. Parrinello, Phys. Rev. Lett. **55**, 2471 (1985)
- [2] G. Galli and M. Parrinello, in Proc. NATO ASI "Computer Simulation in Materials Science," Eds. M. Meyer and V. Pontikis, Kluwer 1991
- [3] M. E. Tuckerman and M. Parrinello, J. Chem. Phys. **101**, 1302 (1994), *ibid* **101**, 1316 (1994) **66**
- [4] J. Hutter, M. Parrinello, and S. Vogel, J. Chem. Phys. **101**, 3862 (1994), J. Hutter and M. Parrinello, to be published
- [5] J. Hutter, M. E. Tuckerman, and M. Parrinello, J. Chem. Phys., in press **23**
- [6] J. Hutter, H. P. Lüthi and M. Parrinello, Comput. Mat. Sci. **2**, 244 (1994) **13**
- [7] A. Banerjee, N. Adams, J. Simons, and R. Shepard, J. Phys. Chem. **89**, 52 (1985) **14**
- [8] S. Nosé, J. Chem. Phys. **81**, 511 (1984), Mol. Phys. **52**, 255 (1984) **15**
- [9] W. G. Hoover, Phys. Rev. A **31**, 1695 (1985) **15**
- [10] H. B. Schlegel, Theo. Chim. Acta **66**, 333 (1984)
- [11] N. Marzari and D. Vanderbilt, Phys. Rev. B **56**, 12847 (1997) **14**
- [12] T. H. Fischer and J. Almlöf, J. Phys. Chem. **96**, 9768 (1992) **14**
- [13] P. Császár and P. Pulay, J. Mol. Struct. **114** 31 (1984) **13**
- [14] R. Fletcher, Practical Methods of Optimization; Wiley: New York, 1980 Vol.1 **13**
- [15] J. Shelley, The MOVIE program **22, 63**
- [16] J. C. Slater, Phys. Rev. **81**, 385 (1951) **42**
- [17] J. P. Perdew and A. Zunger Phys. Rev. B **23**, 5048 (1981) **42**
- [18] S. H. Vosko, L. Wilk, and M. Nusair, Can. J. Phys. **58**, 1200 (1980) **42**
- [19] C. Lee, W. Yang, and R. G. Parr, Phys. Rev. B **37**, 785 (1988) **42**
- [20] J. P. Perdew and Y. Wang, Phys. Rev. B **45**, 13244 (1991) **42**
- [21] D. M. Ceperley and B. J. Alder, Phys. Rev. Lett. **45**, 566(1980) **42**



- [22] A. D. Becke, Phys. Rev. A **38**, 3098 (1988) 42
- [23] J. P. Perdew et al., Phys. Rev. B **46**, 6671 (1992) 42
- [24] John P. Perdew, Kieron Burke, and Matthias Ernzerhof, Phys. Rev. Lett. **77**, 3865-3868 (1996); *ibid.*, **78**, 1396-1396 (E) (1997); *ibid.*, **80**, 891 (R) (1998) 42
- [25] Yingkai Zhang and Weitao Yang, Phys. Rev. Lett. **80**, 890 (C) (1998) 42
- [26] Fred A. Hamprecht, Aron J. Cohen, David J. Tozer and Nicholas C. Handy, J. Chem. Phys. **109**, 6264 (1998); [HCTH/120:] A. Daniel Boese, Nikos L. Doltsinis, Nicholas C. Handy and Michiel Sprik, J. Chem. Phys. **112**, 1670-1678 (2000) 42
- [27] J. P. Perdew Phys. Rev. B **33**, 8822 (1986), Erratum Phys. Rev. B **34**, 7406 (1986) 42
- [28] K. Laasonen, M. Sprik, M. Parrinello, and R. Car, J. Chem. Phys. **99**, 9081 (1993) 42
- [29] D. Vanderbilt, Phys. Rev. B **41**, 7892 (1990)
- [30] G. B. Bachelet, D. R. Hamann, and M. Schlüter, Phys. Rev. B **26**, 4199 (1982)
- [31] X. Gonze, R. Stumpf, and M. Scheffler, Phys. Rev. B **44**, 8503 (1991), R. Stumpf and M. Scheffler, Research Report of the Fritz-Haber-Institut (1990)
- [32] L. Kleinman and D. M. Bylander, Phys. Rev. Lett. **48**, 1425 (1982)
- [33] R. W. Hockney, Methods Comput. Phys. **9**, 136 (1970) 27, 31
- [34] E. R. Davidson, J. Chem. Phys. **46**, 3320 (1967); K. R. Roby, Mol. Phys. **27**, 81 (1974); R. Heinzmann and R. Ahlrichs, Theoret. Chim. Acta (Berl.) **42**, 33 (1976); C. Ehrhardt and R. Ahlrichs, Theoret. Chim. Acta (Berl.) **68**, 231 (1985) 44
- [35] E. R. Davidson, J. Comput. Phys. **17**, 87 (1975);
- [36] A. D. Becke and K. E. Edgecombe, J. Chem. Phys. **92**, 5397 (1990); B. Silvi and A. Savin, Nature **371**, 683 (1994); 21
- [37] F. L. Hirshfeld, Theoret. Chim. Acta **44**, 129 (1977); 44
- [38] S. R. Cox and P. A. Kollman, J. Comput. Chem. **5**, 129 (1984); 44

- [39] G. J. Martyna, D. J. Tobias, and M. L. Klein, *J. Chem. Phys.* **101**, 4177 (1994) [14](#)
- [40] R. N. Barnett and U. Landman, *Phys. Rev. B* **48**, 2081 (1993)
- [41] M. Sprik and G. Ciccotti, *J. Chem. Phys.* **109**, 7737 (1998) [40](#)
- [42] A. Alavi, J. Kohanoff, M. Parrinello and D. Frenkel, *Phys. Rev. Lett.* **73**, 2599–2602, (1994) [1](#), [12](#)
- [43] D. Marx and M. Parrinello, *Z. Phys. B (Rapid Note)* **95**, 143 (1994) [12](#), [33](#)
- [44] D. Marx and M. Parrinello, *J. Chem. Phys.* **104**, 4077 (1996) [12](#), [33](#)
- [45] L. D. Fosdick and H. F. Jordan, *Phys. Rev.* **143**, 58 (1966), see Eq. (3.13) [34](#)
- [46] M. E. Tuckerman, D. Marx, M. L. Klein, and M. Parrinello, *J. Chem. Phys.* **104**, 5579 (1996). [33](#), [34](#), [35](#)
- [47] (a) J. Cao and G. A. Voth, *J. Chem. Phys.* **99**, 10070 (1993); (b) J. Cao and G. A. Voth, *J. Chem. Phys.* **100**, 5106 (1994). [33](#)
- [48] (a) G. J. Martyna, *J. Chem. Phys.* **104**, 2018 (1996); (b) J. Cao and G. J. Martyna, *J. Chem. Phys.* **104**, 2028 (1996). [33](#), [34](#), [35](#)
- [49] G.J. Martyna and M. E. Tuckerman, *J. Chem. Phys.* **110**, 2810 (1999). [31](#)
- [50] I. Frank, J. Hutter, D. Marx, and M. Parrinello, *J. Chem. Phys.* **108**, 4060 (1998). [32](#)
- [51] M. Sprik, *Faraday Discuss.*, **110**, 437 (1998). [40](#)
- [52] D. D. Johnson, *Phys. Rev. B.*, **38**, 12807, (1988) [19](#)

# Index

- &ATOMS...&END, 36–40, 52
- &BASIS...&END, 46–47, 60
- &CPMD...&END, 8–25, 52
- &DFT...&END, 41–43, 54–55
- &PIMD...&END, 33–35
- &PROP...&END, 44–45
- &SYSTEM...&END, 26–32, 52
- \*filename, 52
- \*filename, 36–37
- \*filename, 46
  
- ALEXANDER MIXING, 19
- ANDERSON MIXING, 18, 60, 67
- ANGSTROM, 29, 60
- ANNEALING, 16
- ATOMIC CHARGES, 38
  
- BECKE BETA, 43
- BFGS, 13, 55
- BOGOLIUBOV CORRECTION, 18, 60
- BROYDEN MIXING, 19
  
- CELL, 4
- CELL, 27, 52
- CENTER MOLECULE, 24
- CENTROID DYNAMICS, 33
- CHANGE BONDS, 38
- CHARGE, 29
- CHARGES, 44
- CHECK MEMORY, 25
- citing CPMD, 2
- CLASSICAL TEST, 33
- CLASSTRESS, 24
- CMASS, 14
- COMPRESS, 22, 52
- CONJUGATE GRADIENTS, 13
- CONSTANT CUTOFF, 29
- CONSTRAINT, 4
- CONSTRAINTS...&END CONSTRAINTS, 39–40, 60
- CONVERGENCE, 12, 55
- CUTOFF, 27, 52
  
- DAVIDSON DIAGONALISATION, 17
- DAVIDSON PARAMETER, 17, 57
  
- DEBROGLIE, 34
- DENSITY CUTOFF NUMBER, 29
- DENSITY.x, 4
- diagonalisation, 17
- DIIS MIXING, 18
- DIPOLE, 4
- DIPOLE DYNAMICS, 16
- DIPOLE MOMENT, 45
- DISTRIBUTE FNL, 25
- DUAL, 32, 53
- DUMMY ATOMS, 39
  
- ELECTROSTATIC POTENTIAL, 22
- ELF, 4
- ELF, 21
- EMASS, 14
- ENERGIES, 4
- ENERGYBANDS, 4
- ENERGYBANDS, 22
- examples, 52–62
- EXCHANGE CORRELATION TABLE, 41
- EXCITED DIPOLE, 45
- EXTERNAL POTENTIAL, 22
  
- FACMASS, 34
- FILEPATH, 25
- files (generated by CPMD), 4–5
- FINDIF, 4
- FINITE DIFFERENCES, 24
- FREE ENERGY FUNCTIONAL, 12, 18, 60
- FUNCTIONAL, 41
  
- GC-CUTOFF, 42
- GDIIS, 13, 55
- GENERATE COORDINATES, 38
- GENERATE REPLICAS, 34
- GEOMETRY, 4
- geometry, *see* optimization
- GRADIENT CORRECTION, 42
  
- HAMILTONIAN CUTOFF, 23
- HARMONIC REFERENCE SYSTEM, 23
- HESSIAN, 4
- HESSIAN, 14, 55

- INITIALIZATION, 34
- INITIALIZE WAVEFUNCTION, 23
- installing CPMD, 2
- INTERFACE EGO, 11
- ISOLATED MOLECULE, 24
- ISOTOPE, 38
- ISOTROPIC CELL, 29
  
- KOHN-SHAM ENERGIES, 11
- KPOINTS, 28–29, 60, 67
- KPTS\_GENERATION, 4
  
- LANCZOS DIAGONALISATION, 17, 60
- LANCZOS PARAMETER, 17, 60, 66
- LATEST, 4
- LDA CORRELATION, 42
- LOCAL DIPOLE, 45
- LOCAL SPIN DENSITY, *see* LSD
- LOCALIZE, 45
- LSD, 22, 53
- LSE, 32
  
- MAXCPUTIME, 21
- MAXSTEP, 12, 57
- MEMORY, 24, 59
- memory, 5, 24, 25
- MESH, 30
- mixing, 18–19
  - Alexander, 19
  - Anderson, 18, 60, 67
  - Broyden, 19
  - DIIS, 18
  - moverho, 19
- MOLECULAR DYNAMICS, 11, 14–17, 57
- MOLVIB, 4
- MOVERHO, 19
- MOVIE, 4
- MOVIE, 22
- MOVIE TYPE, 38
- MULTIPLICITY, 30, 53
  
- n-CENTER CUTOFF, 44
- NEWCODE, 41
- NONORTHOGONAL ORBITALS, 23
- NOPRINT, 45
- NORMAL MODES, 35
- NOSE, 15, 65
  
- OCCUPATION, 27
- ODIIS, 13
- OLDCODE, 41
- optimization, 12–14
  - geometry, 11, 55, 64
  - wavefunctions, 11, 52
- OPTIMIZE COMBINED, 11
- OPTIMIZE GEOMETRY, 11, 55
- OPTIMIZE WAVEFUNCTION, 11, 52
- ORTHOGONALIZATION, 23
- OUTPUT, 35
  
- PARRINELLO-RAHMAN, 14
- PATH INTEGRAL, 12
- PATH SAMPLING, 12
- PCG, 13, 52
- POINT GROUP, 30, 60, 67
- POISSON SOLVER, 31
- POLYMER, 32
- POPULATION ANALYSIS, 44
- PRESSURE, 30
- PRINT, 20–21, 53
- PRINT LEVEL, 35
- PROCESSOR GROUPS, 35
- PROJECT, 24
- PROJECT WAVEFUNCTION, 44
- PROPERTIES, 12, 44
- PSEUDO AO, 46
- pseudopotential, 36–38, 52
  - library, 3
  - non-local part, 37–38
  - spline, 24
- pseudopotentials, 47–51
  
- QUENCH, 14
  
- RANDOMIZE, 24, 52
- RATTLE, 23
- READ REPLICAS, 34
- REFERENCE CELL, 29
- RESCALE OLD VELOCITIES, 16
- RESTART, 4
- RESTART, 19–20, 52, 55
- RESTART.x, 4
- RESTFILE, 22
- RFO, 14
- RHOOUT, 21, 53

running CPMD, 3

SCALE, 30

SCALED MASSES, 23

SLATER, 42, 46

SMOOTH, 42

space group

    determination, 30

*SPINDEN.x*, 4

SPLINE, 24

STAGING, 34

starting a run, 3

STATES, 27, 60

STEEPEST DESCENT, 12

steepest descent, 12–13

step

    maximum number, 12, 57

    time, 14, 57

stopping a run, 3

STORE, 21

*STRESS*, 4

STRESS TENSOR, 30

STRUCTURE, 21, 53

SURFACE, 32

SYMMETRIZE COORDINATES, 31

SYMMETRY, 27, 52, 60

symmetry

    determination, 30

TASKGROUPS, 25

TEMPCONTROL, 15

TEMPERATURE, 14, 57

TEMPERATURE ELECTRON, 18, 60

TESR, 30, 60

TIMESTEP, 14, 57

*TRAJECTORY*, 4

TRAJECTORY, 22

TROTTER, 18, 60

TROTTER DIMENSION, 33

TSDC, 13

TSDE, 13

TSDP, 13

VELOCITIES...END VELOCITIES, 40

VIBRATIONAL ANALYSIS, 11

WANNIER, 16

Wannier functions, 16–17

*WANNIER\_CENTER*, 4

*WANNIER\_DOS*, 4

wavefunctions

    initialization, 23, 53, 64

    optimization, *see* optimization

    randomize, 24, 52

    restart, 20, 52