CPMD

Car-Parrinello Molecular Dynamics

An *ab initio* Electronic Structure and
Molecular Dynamics Program

The CPMD consortium
WWW: http://www.cpmd.org/
Mailing list: cpmd-list@cpmd.org
E-mail: cpmd@cpmd.org

March 6, 2024

Send comments and bug reports to
cpmd@cpmd.org

This manual is for CPMD version 4.3.0

Contents

Part I
Overview

1 About this manual

Many members of the CPMD consortium (http://www.cpmd.org/) contributed to this manual. This version of the manual is based on a compilation by Barbara Kirchner, Ari P. Seitsonen and Jürg Hutter working at the Physical Chemistry Institute of the University of Zurich. Recent updates by Mauro Boero, Alessandro Curioni, Jürg Hutter, Axel Kohlmeyer, Nisanth Nair and Wolfram Quester.
If you want to contribute or have constructive criticism please contact cpmd@cpmd.org.

2 Citation

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

CPMD, http://www.cpmd.org/,
Copyright IBM Corp 1990-2019,
Copyright MPI für Festkörperforschung Stuttgart 1997-2001.

3 Important constants and conversion factors

Input and output are in Hartree atomic units (a.u.), unless otherwise explicitly mentioned.

IMPORTANT NOTICE:

As of CPMD version 3.15.1 all constants and conversion factors have been consolidated and updated to the CODATA 2006 data set[1]. For details see the file `cnst.inc` and http://physics.nist.gov/constants.

quantity	*conversion factor*
time step	1 a.u. = 0.02418884326505 fs
coordinates	1 Bohr = 1 a_0 = 0.52917720859 Å
velocity	1 a.u. = 1 Bohr / 1 a.t.u. = 2187691.2541 m/s
energy	1 Ha = 27.21138386 eV = 627.5094706 kcal/mol = 2625.4996251 kJ/mol
plane wave cutoff	1 Ry = 1/2 Ha = 13.60569193 eV
dipole moment	1 a.u. = 2.5417462289 Debye
atomic mass	1 a.u. = 0.00054857990943 a.m.u

4 Recommendations for further reading

- **General Introduction to Theory and Methods**
  Jorge Kohanoff, "Electronic Structure Calculation for Solids and Molecules",
  Cambridge University Press, 2006, ISBN-13 978-0-521-81591-8
  http://www.cambridge.org/9780521815918

- **General introduction to Car-Parrinello simulation**
  D. Marx and J. Hutter, "Ab Initio Molecular Dynamics - Basic Theory and Advanced Methods",
  Cambridge University Press, 2009
  D. Marx and J. Hutter, "Modern Methods and Algorithms of Quantum Chemistry",
  Forschungszentrum Jülich, NIC Series, Vol. 1 (2000), 301-449
  W. Andreoni and A. Curioni,
  "New Advances in Chemistry and Material Science with CPMD and Parallel Computing",
  Parallel Computing, 26 (2000) 819.

- **Electronic Structure Theory**
  Richard M. Martin, "Electronic Structure: Basic Theory and Practical Methods'",
  Cambridge University Press, 2004, ISBN-13: 978-0-521-78285-2
  http://electronicstructure.org

- **General overview about quantum simulation techniques**
  J. Grotendorst, D. Marx, and A. Muramatsu, *Quantum Simulations of Complex Many–Body Systems: From Theory to Algorithms*, (John von Neumann Institute for Computing, Forschungszentrum Jülich 2002);
  Printed Version: ISBN 3–00–009057–6
  Electronic Version: http://www.fz-juelich.de/nic-series/volume10/
  Audio–Visual Version: http://www.fz-juelich.de/video/wsqs/

- **Molecular dynamics simulation**
  M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids* (Clarendon Press, Oxford, 1987; reprinted 1990).
  D. Frenkel and B. Smit, *Understanding Molecular Simulation – From Algorithms to Applications* (Academic Press, San Diego, 1996).
  M. E. Tuckerman and G. J. Martyna, J. Phys. Chem. B **104** 159 (2000).

- **Pseudopotentials**
  http://www.cpmd.org/documentation/useful-links
  http://www.cpmd.org/cpmd_download.html
  http://cvs.berlios.de/cgi-bin/viewcvs.cgi/cp2k/potentials/Goedecker/cpmd/
  http://www.fhi-berlin.mpg.de/th/fhi98md/fhi98PP/
  http://www.physics.rutgers.edu/~dhv/uspp/
  http://www.pwscf.org/pseudo.htm

- **Parallelization & Performance**
  J. Hutter and A. Curioni, Parallel Computing **31**, 1 (2005).
  J. Hutter and A. Curioni, ChemPhysChem **6**, 1788-1793 (2005).
  C. Bekas and A.Curioni, Comp. Phys. Comm.**181**, 1057 (2010).

5 History

5.1 CPMD Version 1

In summer 1993 a project was started to combine the two different ab initio molecular dynamics codes [2] that were used in the group for computational physics of the IBM Research Laboratory in Rüschlikon. There was the IBM-AIX version (ported by J. Kohanoff and F. Buda) of the IBM-VM version (by W. Andreoni and P. Ballone) of the original Car-Parrinello [3] code and a version of the code by K. Laasonen and F. Buda that could handle ultra-soft pseudopotentials [4]. Further goals were to provide a common platform for future developments, as new integration techniques or parallelization. The original Car–Parrinello code [3, 5] was about 8000 lines of Fortran. A first parallel version using the IBM MPL library was finished in 1993. Many people contributed to this effort in different ways: M. Parrinello, J. Hutter, W. Andreoni, A. Curioni, P. Giannozzi, E. Fois, D. Marx and M. Tuckerman.

5.2 CPMD Version 2

5.2.1 Version 2.0

The first major update of the code was finished in summer 1993. New features of the code included a keyword driven input, an initial guess from atomic pseudo-wavefunctions, a module for geometry optimization, several new types of molecular dynamics, Nosé–Hoover [6, 7] thermostats and a diagonalization routine to get Kohn-Sham energies [8]. This code had 17'000 lines.

5.2.2 Version 2.5

In 1994 many additions were made to the code. The communication was improved and a library interface for MPI was introduced. The code reached its most stable version at the end of the year with version number 2.5. At this stage a working version of *ab initio* path integrals [9, 10] based on a one level parallelization was implemented in a separate branch of the code by Dominik Marx.

5.3 CPMD Version 3

5.3.1 Version 3.0

This major update included changes to improve the portability of the code to other platforms. Most notable was the shmem interface for optimal parallel performance on Cray computers. New features of this version were constant pressure molecular dynamics using the Parrinello-Rahman Lagrangian [11, 12], the possibility for symmetry constraints and Stefan Goedecker's dual space pseudopotentials [13]. The library concept for the pseudopotentials had been changed. The code had grown to 55'000 lines.

5.3.2 Version 3.1

Only minor updates were made for this version. However, it served as a starting point for two major new developments. The free energy functional [14] code with **k** points was developed by Ali Alavi and Thierry Deutsch in Belfast. An efficient path integral version using two level parallelism was put together by Mark Tuckerman, Dominik Marx, and Jürg Hutter [15] .

5.3.3 Version 3.2

This version included several new algorithms. Some of these were lost in the transfer to the next version.

5.3.4 Version 3.3

This version was developed using the free energy functional version (based on 3.1) as a basis. The path integral version was fully included but only part of the changes from the "main" version 3.2 were taken over. The QM/MM interface to the EGO code was included [16]. Development of the linear response [17] parts of the code started. Maximally localized Wannier functions [18] were implemented. This version was finished in 1998, the code was about 115'000 lines long.

5.3.5 Version 3.4

The most notable change to this version was the inclusion of the QM/MM interface developed by Alessandro Laio, Joost VandeVondele and Ursula Röthlisberger [19, 20, 21]. Besides that only minor changes to the functionality of the code were done. This version included mostly bug fixes and was finished in 2000.

5.3.6 Version 3.5

This was the first version made generally available at www.cpmd.org in early 2002. Many bugs were fixed, most notably the code for the ultra-soft pseudopotentials was working again. The new size of the code was 136'000 lines.

5.3.7 Version 3.6

This developers version included the final versions of the linear response methods for the calculation of the polarizability and the chemical NMR shifts developed by Anna Putrino and Daniel Sebastiani [22, 23, 24]. Marcella Iannuzzi contributed a $\mathbf{k} \cdot \mathbf{p}$ module [25]. Time-dependent density functional response theory was implemented and forces for excited state energies programmed. Salomon Billeter, Alessandro Curioni and Wanda Andreoni implemented new linear scaling geometry optimizers that allow to locate geometrical transition states in a clean way [26]. Fine grained parallelism with OpenMP was added (by Alessandro Curioni and Jürg Hutter) and can be used together with the distributed memory MPI version.

5.3.8 Version 3.7

The stable version of the developers code was made publicly available in early 2003. The code has 150'000 lines.

5.3.9 Version 3.8

Developer's version.

5.3.10 Version 3.9

Many new developments, improvements, cleanups, and bug fixes have been added since the last public version of the code. Most notably, the methodology for reactive Car-Parrinello metadynamics [27, 28] is made available in this version.
Other new functionality includes G-space localization of wavefunctions, Hockney-type Poisson Solver [29] for slabs with influence function in G-Space, code to determine molecular KS states from Wannier functions, code for trajectory analysis, calculation of dipole moments using the Berry phase and in real space, transition matrix elements between orbitals, growth function for constraints and restraints, new code for applying static electrical fields, periodic or final diagonalization of WF, van der Waals force field according to Elstner's formula [30] and dumping files for PDOS.
Improvements of the code include performance and OpenMP improvements, improved code for keeping wavefunction in real space, updated TDDFT, SAOP TDDFT functional, a much improved configure script, bug fixes for HF exchange, screened exchange, cleanup of memory management, more checks on unsupported options, fixed constraints in geometry optimization.

Modified ROKS [31], Ports to MacOS-X/PPC, Cray X1, and Intel EM64T, k-points with swapfiles are working again on many platforms, detection of incompatible Vanderbilt pseudopotentials.

5.3.11 Version 3.10

Developer's version.

5.3.12 Version 3.11

Many improvements, cleanups, bug fixes and some new features have been added since the last public version of the code. New functionalities include calculation of the electric field gradient tensor along MD trajectory, EPR calculations, efficient wavefunction extrapolation for BOMD, distance screening for HFX calculation and hybrid functional with PBC, interaction perturbation method, molecular states in TDDFT calculations, analytic second derivatives of gradient corrected functionals [32], Born charge tensor during finite difference vibrational analysis, Gromacs QM/MM interface [33], and distributed linear algebra support.
New supported platforms include, IBM Blue Gene/L [34], Cray XT3, NEC-SX6 Earth Simulator (Vector-Parallel) and Windows NT/XT using GNU Gfortran. Performance tunings for existing platforms include FFTW interface, 16 Byte memory, alignment for Blue Gene/L, extension of the taskgroup implementation to cartesian taskgroups (Blue Gene/L), parallel distributed linear algebra, alltoall communication in either single (to reduce communication bandwidth) or double precision, special parallel OPEIGR, improved OpenMP support [35], and improved metadynamics.

5.3.13 Version 3.12

Developer's version.

5.3.14 Version 3.13

Several improvements, cleanups, bug fixes and a few new features have been added since the last public version of the code. New functionalities include additional distributed linear algebra code for initialization, final wavefunction projection and Friesner diagonalization, mean free energy path search method, multiscale shock method [36], Langevin integrator for metadynamics with extended Lagrangian, calculation of non-adiabatic couplings, Landau-Zener Surface hopping, ROKS-based Slater transition-state density, linear-response DFPT with a ROKS-based reference state [37], simplified van der Waals correction according to Grimme [38], simplified ROKS input options with hard-wired variants of modified Goedecker algorithms for ROKS, PBEsol functional, ports to IBM Blue Gene/P, MacOS-X/x86 and PACS-CS / T2K, support for fftw-3, improved ultrasoft pseudopotential parallel code (VDB) (MPI and OpenMP), optimizations for scalar CPUs, new collective variables for metadynamics, variable cell support in DCD output, isotropic and zflexible cell for Parrinello-Rahman dynamics, damped dynamics and Berendsen thermostats for electrons, ions and cell, path-integral support for BO-MD, support for completely reproducible outputs for CPMD TestSuite, consistent and updated unit conversions throughout the code, spin-density Mulliken analysis, aClimax format output of vibrational frequencies, optimization scheme for Goedecker pseudopotential parameters for use as link atoms in QM/MM applications, support for QUENCH BO with PCG MINIMIZE when using VDB potentials, corrections for a number of serious bugs in the Gromos QM/MM code, use of PDB format coordinate files for Amber2Gromos, Taskgroup support for Gromos QM/MM with SPLIT option, BO-MD with EXTRAPOLATE WFN fully restartable, access to QM and MM energy in QM/MM calculations, and improvements of the manual.

5.3.15 Version 3.14

Developer's version.

5.3.16 Version 3.15

New features, performance improvement and bug fixes have been added since the latest version of the code. These include a new efficient orthogonalization scheme [40], Constrained Density Functional Theory [41], force matching in QM/MM runs, the new generalized Langevin thermostat [42], the screened hybrid functional HSE06 from the group of Scuseria [44, 45], multiple walkers in metadynamics, surface hopping dynamics with non-adiabatic coupling vectors in TDDFT [46], extensions of Grimme vdW corrections and initial support for the Ehrenfest Molecular Dynamics [47], and Kerker mixing for handling metallic systems [48]. The new version includes also ports to IBM POWER7, Fujitsu-Primergy BX900, several Linux updates and an updated pseudopotential library.

5.3.17 Version 3.17

New features, scalability improvements and bug fixes have been added since the latest version of the code. Among the several novel implementations we number a new highly parallel scheme for the evaluation of the Hartree–Fock exact exchange with an efficient thresholding, support for parallel I/O in writing and reading the RESTART file, the introduction of a second parallelization layer over the molecular states which replaces the original TASKGROUP parallelization, increasing further the scalability together with the already available parallelization over plane-waves. Finally, improved OpenMP 3.0 support has been introduced both in the main code and in the QM/MM interface. This allows for a substantial speedup of the code in a hybrid OMP/MPI distribution (see note below). Between the novel methodologies we mention the availability of the *ab-initio* vdW corrections, of a fully functional Ehrenfest Dynamics module and of several improvements to the FO-DFT scheme. Additional developments include novel schemes for treating QM/MM link-atoms and the possibility of post-processing Nosé-Hoover trajectories and energies. The new version includes also porting and optimization to the IBM BlueGene/Q and different updates to the Linux architecture files.
Note on OMP 3.0: The *collapse(n)* clause, although powerful in well nested loops, may not work in old compilers and is known to have troubles e.g. in old Intel Fortran Compilers (versions 11 and former ones). Please, check carefully your OS and your compiler berfore compiling blindly and refer to the discussion in http://openmp.org/forum/viewtopic.php?f=3&t=1023&start=10.

5.4 CPMD Version 4

5.4.1 Version 4.0

A new development version started in 2012 (and developed concurrently with version 3.17.1) with the purpose to refactor CPMD.

5.4.2 Version 4.1

Release 4.1 is the new version made available after almost two years from the last public release. It sets a turning point to CPMD and its new position as a modern software (object oriented) while still retaining all the features and functionalities from the previous versions.
Among the major technical breakthroughs: all memory allocations are converted to dynamical allocations using standard fortran; extensive usage of modules so to promote a code re-usage policy; fixed several instabilities due to arithmetical exceptions; all floats operations are now converted to arbitrary precision; support of few more compilers; all COMMONs have been converted to TYPES; extensive usage of IMPLICIT NONE; removed a massive amounts of unused variables and procedures. Finally, we have build-up a regression tester that covers more than half of the functionalities (700 tests and keeps increasing day by day), providing a quality control for developers and users.
Beyond cleanup and bug fixes improving the stability of all functionalities, we also report the implementation of new features such as: Spin-Orbit-Coupling, local control, inter-system-crossing SH dynamics together, Adiabatic Bohmian dynamics; coupling with IPHIGENIE to perform QM/MM calculations with polarizable schemes and a wider usage of CP_GROUPS to improve scalability of

different other functionalities. The new version includes also porting, optimisation and different updates for various architectures. The work initiated with version 4.0 will continue tirelessly on the forthcoming version.

5.4.3 Version 4.3

With the version 4.3, we have ported some of the core-procedures to gpu. The XC procedures have been refactored and it was added the possibility to link to libxc. Coulomb-attenuated functionals (CAM) and the Minnesota functionals (M05, M06, M08 and M11 families) are now available as internal xc procedures. A new development for Eherenfest dynamics with ionized states is deployed. Few novel implementations are available: a multiple time step scheme for MD, accelerated exact exchange for isolated systems, the stress tensor, LSD and UltraSoft Vanderbilt pseudopotentials and NPT. Also a new bicanonical ensemble method is made available with version 4.3. Extensive cleanup and bug fixes and strenghtening of the regression test suite.

5.4.4 Version 4.5

The Minnesota non-separable exchange functional families (MN12, MN15) as well as the revM06-L functional are now available.

6 Installation

This version of CPMD is equipped with a shell script to create a Makefile for a number of given platforms. If you run the shell script `configure.sh` (NOTE: this script was previously named `config.sh`) without any options it will tell you what platforms are available. Choose the label for a target platform close to what your machine is and run the script again.

```
# ./configure.sh PLATFORM
```

NOTE: Due to filesystem implementation limitations, compilation under MacOS X, and Windows NT/XP/Vista requires the compilation outside of the **src** directory. See below.
Most likely the generated makefile with **not** match the setup on your machine and you have to adapt the various definitions of compilers, optimization flags, library locations and so on. To display additional information about a configuration type:

```
# ./configure.sh -i PLATFORM
```

The executable can then be compiled using the `make` command. To see all possible options use

```
# ./configure.sh -help
```

A common problem is that the default names of the libraries and the path to the libraries are not correct in the `Makefile`. In this case you have to change the corresponding entries in the `Makefile` manually. If you are changing the preprocessor flags (the CPPFLAGS entry), e.g. going from a serial to a parallel compilation, you have to delete all preprocessed, module and object files first, preferably by executing:

```
# make clean
```

Alternatively you can compile CPMD outside the source directory. This is highly recommended, if you need to compile several executables concurrently, e.g. if you are doing development on several platforms. This is done by creating a directory for each platform (e.g. by 'mkdir ./bin/cpmd-pc-pgi; mkdir ./bin/cpmd-pc-pgi-mpi') and then create a makefile for each of those directories and pointing to the original source directory with with `SRC` and `DEST` flags. For the above examples this would be:

```
# ./configure.sh -m -SRC=$PWD -DEST=./bin/cpmd-pc-pgi PC-PGI

# ./configure.sh -m -SRC=$PWD -DEST=./bin/cpmd-pc-pgi-mpi PC-PGI-MPI
```

Now you can do development in the original source directory and only need to recompile the altered modules by typing 'make' in the respective subdirectories.
NOTE: For compilation under Mac OS-X this procedure is currently **required**.

Compiling CPMD on Linux platforms can be particularly tricky, since there are several Fortran compilers available and there are no standard locations for supporting libraries (which on top of that usually have to be compiled with the same or a compatible compiler). If you run into problems, you may want to check out the CPMD Mailing list archives at
http://www.cpmd.org/pipermail/cpmd-list/
to see, whether your specific problem has already been dealt with.

Please also note that only recent versions of the GNU gfortran compiler (4.1 and later) are sufficient to compile CPMD. The now obsolete GNU Fortran 77 compiler, `g77`, and the G95 Fortran compiler, `g95`, are *not* able to compile CPMD.

7 Running CPMD

The CPMD program is started with the following command:

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

Running `cpmd.x` requires the following files:

- an **input file** *file.in* (see section 9)
- **pseudopotential files** for all atomic species specified in *file.in* (see section 9.5.1).

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

- The second command line argument [*PP_path*] is set.
- If the second command line argument [*PP_path*] is not given, the program checks the environment variables
CPMD_PP_LIBRARY_PATH and PP_LIBRARY_PATH.
- If neither the environment variables 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 status messages to monitor 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 8. Large files are written either to the current directory, the directory specified by the environment variable **CPMD_FILEPATH**, or the directory specified in the input file using the keyword **FILEPATH**.

In case CPMD quits with a non-zero exit status, an error message is written to a file called *LocalError-X-X-X.log* by every task, indicating the procedure in which the error occured and the call stack. In certain cases (mostly when processing the input), more ample error information is written to the output unit as well.

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

EXIT

in the run-directory.

8 Files

Incomplete list of the files used or created by CPMD:

File	*Contents*
RESTART	Restart file
RESTART.x	Old/New restart files
LATEST	Info file on the last restart file
GEOMETRY	Current ionic positions and velocities
GEOMETRY.xyz	Current ionic positions and velocities in Å
GEOMETRY.scale	Current unit cell vectors in Åand atomic units and ionic positions in scaled coordinates
GSHELL	$\mathbf{G}^2$ (NOT normalized), G-shells $\lvert\mathbf{G}\rvert$ in a.u. and related shell index
TRAJECTORY	All ionic positions and velocities along the trajectory
TRAJEC.xyz	All ionic positions along the trajectory in xyz-format
TRAJEC.dcd	All ionic positions along the trajectory in dcd-format
GEO_OPT.xyz	All ionic positions along the geometry optimization in xyz-format
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
NOSE_ENERGY	The kinetic, potential and total energies of the Nose-Hoover thermostat(s)
NOSE_TRAJEC	The "trajectory" of the Nose-Hoover variables and velocities
CONSTRAINT	The "trajectory" of constraint/restraint forces
METRIC	The "trajectory" of collective variable metric (restraints only)
DIPOLE	The "trajectory" of dipole moments
dipole.dat	The dipole file produced in the spectra calculation using the propagation TDDFT scheme
DENSITY.x	Charge density in Fourier space
SPINDEN.x	Spin density in Fourier space
ELF	Electron localization function in Fourier space
LSD_ELF	Spin polarized electron localisation function
ELF_ALPHA	Electron localisation function of the alpha density
ELF_BETA	Electron localisation function of the beta density
WAVEFUNCTION	Wavefunction instead of density is stored
HESSIAN	Approximate Hessian used in geometry optimization
FINDIF	Positions and gradients for finite difference calculations
VIBEIGVEC	Eigenvectors of Hessian
MOLVIB	The matrix of second derivatives, as used by the program MOLVIB
VIB1.log	Contains the modes $4-3$N in a style similar to the gaussian output for visualization with MOLDEN, MOLEKEL,...
VIB2.log	Contains the modes $1-3$N-3
ENERGYBANDS	Eigenvalues for each k points
KPTS_GENERATION	Output of k points generation
WANNIER_CENTER	Centers of the Wannier functions

WC_SPREAD	Spread of the Wannier functions
WC_QUAD	Second moments of the Wannier functions as : ISTEP QXX QXY QXZ QYY QYZ QZZ
IONS+CENTERS.xyz	Trajectory of ionic positions and WFCs in Å
WANNIER_DOS	Projection of the Wannier functions onto the Kohn-Sham states
WANNIER_HAM	KS Hamiltonian in the Wannier states representation
WANNIER_1.x	Wannier orbitals or orbital densities
wavefunctions	Containes the complex wavefunctions for Ehrenfest dynamics
HARDNESS	Orbital hardness matrix
RESTART.NMR	Files needed to restart a NMR/EPR calculation
RESTART.EPR	
RESTART.L_x	
RESTART.L_y	
RESTART.L_z	
RESTART.p_x	
RESTART.p_y	
RESTART.p_z	
jα_Bβ.cube	Files in .CUBE format that contain the
Bα_Bβ.cube	induced current densities and magnetic fields
	in an NMR calculation, respectively (α,β=x,y,z)
PSI_A.i.cube	Files in .CUBE format that contain (spinpolarized)
PSI_B.i.cube	orbitals and densities.
RHO_TOT.cube	
RHO_SPIN.cube	
QMMM_ORDER	Relation between the various internal atom order lists in QM/MM runs.
QM_TEMP	"local" temperatures of the QM/MM subsystems.
CRD_INI.grm	Positions of all atoms of first step in Gromos format (QM/MM only).
CRD_FIN.grm	Positions of all atoms of last step in Gromos format (QM/MM only).
MM_TOPOLOGY	New Gromos format topology file (QM/MM only).
ESP	Contains the ESP charges of the QM atoms (QM/MM only).
EL_ENERGY	Contains the electrostatic interaction energy (QM/MM only).
MULTIPOLE	Contains the dipole- and the quadrupole-moment of the quantum system.
MM_CELL_TRANS	Contains the trajectory of the offset of the QM cell (QM/MM only).
OCCMAT	The "trajectory" of the Occupation Matrix (DFT+U only)

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

In contrast, in case of mean free energy path searches, each replica uses its own directory for nearly all files. Exception are the file `LATEST` and the geometry files named as for path integrals: `GEOMETRY_`s. The directories are built from the **FILEPATH** path (or else from the environment variable **CPMD_FILEPATH**) with the suffixes _s, $s = \{1, \ldots, P\}$.

In case of bicanonical Ensemble MD

File	*Contents*
OUTPUT_CNF2	standard output of the smaller canonical system
RESTART_CNF1	Restart file of the larger canonical system
RESTART_CNF2	Restart file of the smaller canonical system
RESTART_CNF1.x	Old/New restart files

RESTART_CNF2.x	Old/New restart files
LATEST_CNF1	Info file on the last restart file
LATEST_CNF2	Info file on the last restart file
ENERGIES	Chemical potential, bicanonical weight, Kohn-Sham energy difference along the trajectory
ENERGIES_CNF1	All energies along the trajectory larger canonical system
ENERGIES_CNF2	All energies along the trajectory smaller canonical system
GEOMETRY_CNF1	Current ionic positions and velocities larger canonical system
GEOMETRY_CNF2	Current ionic positions and velocities smaller canonical system
GEOMETRY_CNF1.xyz	Current ionic positions and velocities in Ålarger canonical system
GEOMETRY_CNF2.xyz	Current ionic positions and velocities in Åsmaller canonical system
TRAJECTORY_CNF1	All ionic positions and velocities along the trajectory larger canonical system
TRAJECTORY_CNF2	All ionic positions and velocities along the trajectory smaller canonical system
TRAJEC_CNF1.xyz	All ionic positions along the trajectory in xyz-format larger canonical system
TRAJEC_CNF2.xyz	All ionic positions along the trajectory in xyz-format smaller canonical system
TRAJEC_CNF1.dcd	All ionic positions along the trajectory in dcd-format larger canonical system
TRAJEC_CNF2.dcd	All ionic positions along the trajectory in dcd-format smaller canonical system

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

Exceptions are “trajectory” type files (`TRAJECTORY`, `ENERGIES`, `MOVIE`, `STRESS`, ...), in them data are **appended**.

Part II
Reference Manual

9 Input File Reference

The following sections **try** to explain the various keywords and syntax of a CPMD input file. It is not meant to teach how to create good CPMD input files, but as a reference manual.

9.1 Basic rules

- **Warning:** Do not expect the input to be logical. The programmers 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 file is read via the code in the files
`control_utils.mod.F90`, `sysin_utils.mod.F90`, `dftin_utils.mod.F90`, `ratom_utils.mod.F90`, `recpnew_utils.mod.F90`, `detsp_utils.mod.F90`, `proppt_utils.mod.F90`, `setbasis_utils.mod.F90`, `vdwin_utils.mod.F90`, `respin_p_utils.mod.F90`, `lr_in_utils.mod.F90`, `orbhard_utils.mod.F90`, `egointer_utils.mod.F90`, `pi_cntl_utils.mod.F90`, `cl_init_utils.mod.F90`, `cplngs_utils.mod.F90`, `mts_utils.mod.F90`
- The input is free format except when especially stated
- In most cases, only the first 80 characters of a line are read (exceptions are lists, that have to be on one line).
- Lines that do not match a keyword are treated as comments and thus ignored.
Warning: For most sections there will be a report of unrecognized keywords. For the `&ATOMS` this is not possible, so please check the atom coordinates in the output with particular care.
- Take warnings seriously. There are a few warnings, that can be safely ignored under specific circumstances, but usually warnings are added to a program for a good reason.
- The order of the keywords is arbitrary unless it is explicitly stated. For keywords that select one of many alternatives (e.g. the algorithm for wavefunction optimization), the last one 'wins'.
- Only keywords with capital letters match
- Lists inclosed in **{ }** imply that you have to choose **exactly one** of the items
- Lists inclosed in [ ] imply that you can choose **any number** of items on the same line
- Arguments to a keyword are given on the following line(s) if not explicitly stated otherwise
- The full keyword/input line has to be within columns 1 to 80
- There are exceptions to those rules.

9.2 Input Sections

The input file is composed of different sections. Each section is started by `&SECTIONNAME` and ended by `&END`. All input outside the sections is ignored.

&INFO ... &END ↔ A place to put comments about the job.
The contents of this section will be copied to the output file at the beginning of the calculation.

&CPMD ... &END ↔ General control parameters for calculation (**required**).

&SYSTEM ... &END ↔ Simulation cell and plane wave parameters (**required**).

&PIMD ... &END ↔ Path integral molecular dynamics (PIMD)
This section is only evaluated if the **PATH INTEGRAL** keyword is given in the &CPMD section.

&PATH ... &END ↔ Mean free energy path calculation (MFEP)
This section is only evaluated if the **PATH MINIMIZATION** keyword is given in the &CPMD section.

&ATOMS ... &END ↔ Atoms and pseudopotentials and related parameters (**required**).
Section 9.5.1 explains the usage of pseudopotentials in more detail.

&DFT ... &END ↔ Exchange and correlation functional and related parameters.

&PROP ... &END ↔ Calculation of properties
This section is only fully evaluated if the **PROPERTIES** keyword is given in the &CPMD section.

&BASIS ... &END ↔ Atomic basis sets for properties or initial guess

&RESP ... &END ↔ Response calculations
This section is always evaluated, even if it is not used.

&PTDDFT ... &END ↔ Propagation TDDFT for Ehrenfest dynamics and spectra

&LINRES ... &END ↔ General input for HARDNESS and TDDFT calculations

&HARDNESS ... &END ↔ Input for HARDNESS calculations
This section is only evaluated if the **ORBITAL HARDNESS LR**
keyword is given in the &CPMD section.

&TDDFT ... &END ↔ Input for TDDFT calculations

&QMMM ... &END ↔ Input for Gromos QM/MM interface (see section 11.16).
Required if the **QMMM** keyword is given in the &CPMD section

&CLAS ... &END ↔ Simple classical code interface

&EXTE ... &END ↔ External field definition for EGO QM/MM interface

&VDW ... &END ↔ Settings associated to van der Waals-correction schemes.
This section is only evaluated if either the keyword **DCACP**, **VDW CORRECTION**, or **VDW WANNIER** is given in the &CPMD section.

&MTS ... &END ↔ Parameters for the Multiple Time-Step MD scheme.

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

9.3 List of Keywords by Sections

9.3.1 &CPMD ... &END

ALEXANDER MIXING
ALLTOALL {SINGLE,DOUBLE}
ANDERSON MIXING
ANNEALING {IONS,ELECTRONS,CELL}
BENCHMARK
BERENDSEN {IONS,ELECTRONS,CELL}
BFGS
BICANONICAL ENSEMBLE {CHEMICALPOTENTIAL, XWEIGHT} [INFO]
BLOCKSIZE STATES
BOGOLIUBOV CORRECTION [OFF]
BOX WALLS
BROYDEN MIXING
CDFT [NEWTON, DEKKER],[SPIN, ALL, PCGFI, RESWF, NOCCOR, HDA[AUTO, PHIOUT, PROJECT]]
CENTER MOLECULE [OFF]
CHECK MEMORY
CLASSTRESS
CMASS
COMBINE SYSTEMS [REFLECT,NONORTH,SAB]
COMPRESS {WRITEnn}
CONJUGATE GRADIENTS {ELECTRONS, IONS}
CONVERGENCE [ORBITALS, GEOMETRY, CELL]
CONVERGENCE [ADAPT, ENERGY, CALFOR, RELAX, INITIAL]
CONVERGENCE [CONSTRAINT]
CP_GROUPS
CZONES [SET]
DAMPING {IONS,ELECTRONS,CELL}
DAVIDSON DIAGONALIZATION
DAVIDSON PARAMETER
DCACP
DEBUG FILEOPEN
DEBUG FORCES
DEBUG MEMORY
DEBUG NOACC
DIIS MIXING
DIPOLE DYNAMICS {SAMPLE,WANNIER}
DISTRIBUTED LINALG {ON,OFF}
DISTRIBUTE FNL
ELECTRONIC SPECTRA
ELECTROSTATIC POTENTIAL [SAMPLE=nrhoout]
ELF [PARAMETER]

EMASS
ENERGYBANDS
EXTERNAL POTENTIAL {ADD}
EXTRAPOLATE WFN {STORE}
EXTRAPOLATE CONSTRAINT
FILE FUSION
FILEPATH
FINITE DIFFERENCES
FIXRHO UPWFN
FREE ENERGY FUNCTIONAL
GDIIS
GSHELL
HAMILTONIAN CUTOFF
HARMONIC REFERENCE SYSTEM [OFF]
HESSCORE
HESSIAN [DISCO,SCHLEGEL,UNIT,PARTIAL]
IMPLICIT NEWTON RAPHSON options
INITIALIZE WAVEFUNCTION {RANDOM, ATOMS} [PRIMITIVE]
INTERFACE {EGO,GMX} {[MULLIKEN, LOWDIN, ESP, HIRSHFELD],PCGFIRST}
INTFILE[READ,WRITE,FILENAME]
ISOLATED MOLECULE
KOHN-SHAM ENERGIES [OFF,NOWAVEFUNCTION]
KSHAM [MATRIX,ROUT,STATE]
LANCZOS DIAGONALIZATION {ALL}
LANCZOS DIAGONALIZATION {OPT,RESET=n}
LANCZOS PARAMETER [N=n] [ALL]
LANGEVIN {WHITE, CPMD, OPTIMAL, SMART, CUSTOM} [MOVECOM]
LBFGS [NREM, NTRUST, NRESTT, TRUSTR]
LINEAR RESPONSE
LSD
LOCAL SPIN DENSITY
MAXRUNTIME
MAXITER
MAXSTEP
MEMORY [SMALL, BIG]
MIRROR
MIXDIIS
MIXSD
MODIFIED GOEDECKER [PARAMETERS]
MOLECULAR DYNAMICS {CP, BO, EH, PT, CLASSICAL, FILE [XYZ, NSKIP=N, NSAMPLE=M]}
MOVERHO
MOVIE [OFF, SAMPLE]

NOGEOCHECK
NONORTHOGONAL ORBITALS [OFF]
NOSE {IONS, ELECTRONS, CELL} [ULTRA,MASSIVE,CAFES]
NOSE PARAMETERS
ODIIS [NOPRECONDITIONING,NO_RESET=nreset]
OPTIMIZE GEOMETRY [XYZ, SAMPLE]
OPTIMIZE WAVEFUNCTION
ORBITAL HARDNESS {LR,FD}
ORTHOGONALIZATION [LOWDIN, GRAM-SCHMIDT] *[MATRIX]*
PARA_BUFF_SIZE
PARA_STACK_BUFF_SIZE
PARA_USE_MPI_IN_PLACE
PARRINELLO-RAHMAN {NPT,SHOCK}
PATH INTEGRAL
PATH MINIMIZATION
PATH SAMPLING
PCG [MINIMIZE,NOPRECONDITIONING]
PRFO [MODE, MDLOCK, TRUSTP, OMIN, PRJHES, DISPLACEMENT, HESSTYPE]
PRFO [NVAR, CORE, TOLENV, NSMAXP]
PRFO NSVIB
PRINT {ON,OFF} options
PRINT ENERGY {ON, OFF} options
PRNGSEED
PROJECT {NONE, DIAGONAL, FULL}
PROPAGATION SPECTRA
PROPERTIES
QUENCH [IONS, ELECTRONS, CELL, BO]
RANDOMIZE [COORDINATES, WAVEFUNCTION], [DENSITY, CELL]
RATTLE
REAL SPACE WFN KEEP [SIZE]
RESCALE OLD VELOCITIES
RESTART *[options]*
RESTFILE
REVERSE VELOCITIES
RFO ORDER=nsorder
RHOOUT [BANDS,SAMPLE=nrhoout]
ROKS {SINGLET, TRIPLET},{DELOCALIZED, LOCALIZED, GOEDECKER}
SCALED MASSES [OFF]
SHIFT POTENTIAL
SOC
SPLINE [POINTS, QFUNCTION, INIT, RANGE]
SSIC
STEEPEST DESCENT [ELECTRONS, IONS, CELL, NOPRECONDITIONING,

LINE]
STORE {OFF} [WAVEFUNCTIONS, DENSITY, POTENTIAL]
STRESS TENSOR
STRUCTURE [BONDS, ANGLES, DIHEDRALS, SELECT]
SUBTRACT [COMVEL, ROTVEL]
SURFACE HOPPING
TDDFT
TEMPCONTROL IONS, ELECTRONS, CELL
TEMPERATURE [RAMP]
TEMPERATURE ELECTRON
TIMESTEP
TIMESTEP ELECTRONS
TIMESTEP IONS
TRACE [ALL,MASTER]
TRACE_PROCEDURE
TRACE_MAX_DEPTH
TRACE_MAX_CALLS
TRAJECTORY [OFF, XYZ, DCD, SAMPLE, BINARY, RANGE, FORCES]
TROTTER FACTOR
TROTTER FACTORIZATION OFF
USE_IN_STREAM
USE_OUT_STREAM
USE_MPI_IO
USE_MTS
QMMM [QMMMEASY]
FORCEMATCH
VGFACTOR
VIBRATIONAL ANALYSIS [FD, LR, IN], [GAUSS, SAMPLE, ACLIMAX]
VMIRROR
VDW CORRECTION [ON, OFF]
VDW DCACP
VDW WANNIER
WANNIER DOS
WANNIER MOLECULAR
WANNIER NPROC
WANNIER OPTIMIZATION {SD,JACOBI,SVD}
WANNIER PARAMETER
WANNIER REFERENCE
WANNIER RELOCALIZE_EVERY
WANNIER RELOCALIZE_IN_SCF
WANNIER SERIAL
WANNIER TYPE {VANDERBILT,RESTA}
WANNIER WFNOUT [ALL,PARTIAL,LIST,DENSITY]
WOUT [FULL]

9.3.2 &SYSTEM . . . &END

ACCEPTOR [HDASINGLE,WMULT]
ANGSTROM
CELL [ABSOLUTE, DEGREE, VECTORS]
CHARGE
CHECK SYMMETRY [OFF]
CLASSICAL CELL [ABSOLUTE, DEGREE]
CLUSTER
CONSTANT CUTOFF
COUPLINGS {FD,PROD} [NAT]
COUPLINGS LINRES {BRUTE FORCE,NVECT} [THR,TOL]
COUPLINGS NSURF
CUTOFF [SPHERICAL,NOSPHERICAL]
DENSITY CUTOFF [NUMBER]
DONOR
DUAL
ENERGY PROFILE
EXTERNAL FIELD
HFX CUTOFF
ISOTROPIC CELL
KPOINTS *options*
LOW SPIN EXCITATION
LOW SPIN EXCITATION LSETS
LSE PARAMETERS
MESH
MULTIPLICITY
OCCUPATION [FIXED]
NSUP
POINT GROUP [MOLECULE], [AUTO], [DELTA=delta]
POISSON SOLVER {HOCKNEY, TUCKERMAN, MORTENSEN} [PARAMETER]
POLYMER
PRESSURE
REFERENCE CELL [ABSOLUTE, DEGREE, VECTORS]
SCALE [CARTESIAN] *[S=sascale] [SX=sxscale] [SY=syscale] [SZ=szscale]*
STATES
STRESS TENSOR
SURFACE
SYMMETRIZE COORDINATES
SYMMETRY
TESR
WGAUSS *NWG*
WCUT *CUT*
ZFLEXIBLE CELL

9.3.3 &PIMD ... &END

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

9.3.4 &PATH ... &END

This section contains specific information for Mean Free Energy Path searches[49]. However, the space of collective variables in which the search will be performed has to be defined using **restraints** in the &ATOMS...&END section (see 9.5.2). The initial string in this collective variable space is read in an external file *STRING.0*. This file contains one line per replica, each giving first the replica index and then the list of collective variables values. The basename for directories where each replica is ran should also be specified using the **FILEPATH**, or else the environment variable **CPMD_FILEPATH**. The code writes the files CONSTRAINT.x and METRIC.x during the execution, where x is the string number.

REPLICA NUMBER
NLOOP
NEQUI
NPREVIOUS
FACTOR
ALPHA
OUTPUT [ALL, GROUPS, PARENT]
PRINT LEVEL
PROCESSOR GROUPS

9.3.5 &PTDDFT ... &END

This section contains specific information for Ehrenfest dynamics and the spectra calculation computed using the propagation of the perturbed KS orbitals (Fourier transform of the induced dipole fluctuation). searches[50, 47].

ACCURACY
PROP_TSTEP
EXT_PULSE

EXT_POTENTIAL
N_CYCLES
PERT_TYPE
PERT_AMPLI
PERT_DIRECTION
RESTART
TD_POTENTIAL
PIPULSE

9.3.6 &ATOMS ... &END

This section also contains the nuclear coordinates and information on the pseudopotentials to be used. See section 9.5.1 for more details on this.

ATOMIC CHARGES
CHANGE BONDS
CONFINEMENT POTENTIAL
CONSTRAINTS ... END CONSTRAINTS
METADYNAMICS ... END METADYNAMICS
DUMMY ATOMS
GENERATE COORDINATES
ISOTOPE
MOVIE TYPE
VELOCITIES ... END VELOCITIES

9.3.7 &DFT ... &END

ACM0
ACM1
ACM3
ANALYTICAL_DIV
BECKE BETA
COULOMB ATTENUATION
CP_LIBRARY_ONLY
EXCHANGE CORRELATION TABLE [NO]
FUNCTIONAL *functional(s)*
HUBBARD [NORM,ORTHO,NUATM=*nuatm*,OCCMAT=*printfreq*,VERB]
HARTREE
HARTREE-FOCK
HFX_BLOCK_SIZE
HFX_DISTRIBUTION [BLOCK_CYCLIC,DYNAMIC]
HFX SCREENING {WFC,DIAG,EPS_INT,RECOMPUTE_TWO_INT_LIST_EVERY}

SCEX

SCALED EXCHANGE
GC-CUTOFF
GRADIENT CORRECTION *functionals*
LIBRARY *library_for_functional_1, library_for_functional_2, ...*
KERNEL_LIBRARY *library_for_kernel_1, library_for_kernel_2, ...*
LIBXC_ONLY
LDA CORRELATION *functional*
LR KERNEL *functionals*
NEWCODE
NUMERICAL_DIV
OLDCODE
OLD_DEFINITIONS
PBE_FLEX_KAPPA
PBE_FLEX_GAMMA
PBE_FLEX_MU
PBE_FLEX_BETA
PBE_FLEX_UEG_CORRELATION *functional*
SCALES *scaling_for_functional_1, scaling_for_functional_2, ...*
HFX_SCALE *scaling_for_hfx*
KERNEL_SCALES *scaling_for_kernel_1, scaling_for_kernel_2, ...*
KERNEL_HFX_SCALE *scaling_for_hfx_in_kernel*
SCREENED EXCHANGE **{ASHCROFT,CAM,ERFC,EXP}**
SLATER [**NO**]
SMOOTH
REFUNCT *functionals*
XC_DRIVER

XC_KERNEL *functionals*
MTS_LOW_FUNC *functionals*
MTS_HIGH_FUNC *functionals*

9.3.8 &PROP ... &END

The keyword **PROPERTIES** has to be present in the &CPMD-section of the input-file if this section shall be evaluated.

CHARGES
CONDUCTIVITY
CORE SPECTRA
CUBECENTER
CUBEFILE **{ORBITALS,DENSITY}** [**HALFMESH**]
DIPOLE MOMENT [**BERRY,RS**]
EXCITED DIPOLE
LDOS

LOCALIZE
OPTIMIZE SLATER EXPONENTS
LOCAL DIPOLE
NOPRINT ORBITALS
POLARIZABILITY
POPULATION ANALYSIS [MULLIKEN, DAVIDSON, n-CENTER]
PROJECT WAVEFUNCTION
TRANSITION MOMENT
n-CENTER CUTOFF
AVERAGED POTENTIAL

9.3.9 &RESP . . . &END

CG-ANALYTIC
CG-FACTOR
CONVERGENCE
DISCARD [OFF, PARTIAL, TOTAL, LINEAR]
EIGENSYSTEM
EPR *options*
FUKUI [N=nf, COEFFICIENTS]
HAMILTONIAN CUTOFF
HARDNESS
INTERACTION
KEEPREALSPACE
KPERT *options*
LANCZOS [CONTINUE,DETAILS]
NMR *options*
NOOPT
OACP [DENSITY, REF_DENSITY, FORCE]
PHONON
POLAK
RAMAN
TIGHTPREC

9.3.10 &LINRES . . . &END

CONVERGENCE
DIFF FORMULA
HTHRS
MAXSTEP
OPTIMIZER [SD,DIIS,PCG,AUTO]
QS_LIMIT

STEPLENGTH
THAUTO
XC_ANALYTIC
XC_DD_ANALYTIC
XC_EPS
ZDIIS
GAUGE **{PARA,GEN,ALL}**

9.3.11 &TDDFT . . . &END

DAVIDSON PARAMETER
DAVIDSON RDIIS
DIAGONALIZER **{DAVIDSON,NONHERMIT,PCG} [MINIMIZE]**
FORCE STATE
LOCALIZATION
MOLECULAR STATES
LZ-SHTDDFT
LR-TDDFT
PCG PARAMETER
PROPERTY **{ STATE }**
RANDOMIZE
REORDER
REORDER LOCAL
ROTATION PARAMETER
STATES **[MIXED,SINGLET,TRIPLET]**
T-SHTDDFT
TAMM-DANCOFF **[SUBSPACE,OPTIMIZE]**
TD_METHOD_A **[*functionals*]**

9.3.12 &HARDNESS . . . &END

DIAGONAL **[OFF]**
LOCALIZE
ORBITALS
REFATOM

9.3.13 &CLASSIC . . . &END

FORCE FIELD ... END FORCE FIELD
FREEZE QUANTUM
FULL TRAJECTORY

PRINT COORDINATES
PRINT FF

9.3.14 &VDW ... &END

DCACP Z=z
NO_CONTRIBUTION
INCLUDE_METALS
VDW PARAMETERS
VDW-CUTOFF
VDW-CELL
VDW WANNIER

9.3.15 &QMMM ... &END

COORDINATES
INPUT
TOPOLOGY
ADD_HYDROGEN
AMBER
ARRAYSIZES ... END ARRAYSIZES
BOX TOLERANCE
BOX WALLS
CAPPING
CAP_HYDROGEN
ELECTROSTATIC COUPLING [LONG RANGE]
ESPWEIGHT
EXCLUSION {GROMOS,LIST{NORESP}}
FLEXIBLE WATER [ALL,BONDTYPE]
FORCEMATCH ... END FORCEMATCH
GROMOS
HIRSHFELD [ON,OFF]
MAXNN
NOSPLIT
RCUT_NN
RCUT_MIX
RCUT_ESP
RESTART TRAJECTORY [FRAME {num},FILE '{fname}',REVERSE]
SAMPLE INTERACTING [OFF,DCD]
SPLIT
TIMINGS
UPDATE LIST

VERBOSE
WRITE LOCALTEMP [**STEP** {**nfi_lt**}]

9.3.16 &MTS . . . &END

TIMESTEP_FACTOR
PRINT_FORCES
LOW_LEVEL_FORCES {**DFT, EXTERNAL**}
HIGH_LEVEL_FORCES {**DFT, EXTERNAL**}

9.4 Alphabetic List of Keywords

Note 1: Additional components of CPMD input files that do not fit into the following list are explained in the succeeding section 9.5.
Note 2: Keywords for the &QMMM section of the CPMD/Gromos QM/MM-Interface code are not listed here but in section 11.16.2.

ACM0
Section: &DFT

Add exact exchange to the specified **FUNCTIONAL** according to the adiabatic connection method 0. [51, 52]

ACM1
Section: &DFT

Add exact exchange to the specified **FUNCTIONAL** according to the adiabatic connection method 1. [52, 53] The parameter is read from the next line.

ACM3
Section: &DFT

Add exact exchange to the specified **FUNCTIONAL** according to the adiabatic connection method 3. [52, 54] The three needed parameters are read from the next line.

ACCEPTOR [HDASINGLE,WMULT]
Section: &SYSTEM

Set the **CDFT** acceptor atoms. Parameter NACCR must be specified next to the keyword. NACCR $\in [1, 2, ..., N]$ is the number of acceptor Atoms (N being the total number of atoms). The indices of NACCR atoms separated by whitespaces are read from the next line.
HDASINGLE off if set together with CDFT HDA, CPMD performs a constrained HDA calculation with only an ACCEPTOR group weight but different constraint values N_c. **WMULT off** if set together with CDFT HDA, CPMD performs a constrained HDA calculation with two different an ACCEPTOR group weights for the two states.
HDASINGLE and **WMULT** are mutually exclusive.

ACCURACY
Section: &PTDDFT

Specifies the accuracy to be reached in the Cayley propagation scheme used in Ehrenfest type of dynamics and spectra calculation.

ALEXANDER MIXING
Section: &CPMD

Mixing used during optimization of geometry or molecular dynamics. Parameter read in the next line.
Default value is **0.9**

ALPHA
Section: &PATH

Smoothing parameter for iterating the string (see [49]).
Default value is **0.2**

ALLTOALL {SINGLE,DOUBLE}
Section: &CPMD

Perform the matrix transpose (AllToAll communication) in the 3D FFT using single/double precision numbers. Default is to use double precision numbers.

ANALYTICAL_DIV
Section: &DFT

Calculate the $\mathbf{G} = 0$ term for the Coulomb-attenuated exact exchange using an analytical description of the integral term. This is the default in combination with all internal hybrid xc functionals.

ANDERSON MIXING $N = n$
Section: &CPMD

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 is **0.2** .
With the additional option $N = n$ a 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.

ANGSTROM
Section: &SYSTEM

The atomic coordinates and the supercell parameters and several other parameters are read in Ångstroms.
Default is **atomic units** which are always used internally. Not supported for **QMMM** calculations.

ANNEALING {IONS,ELECTRONS,CELL}
Section: &CPMD

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

ATOMIC CHARGES
Section: &ATOMS

Changes the default charge (0) of the atoms for the initial guess to the values read from the next line. One value per atomic species has to be given.

AVERAGED POTENTIAL
Section: &PROP

Calculate averaged electrostatic potential in spheres of radius Rcut around the atomic positions.
Parameter Rcut is read in from next line.

BECKE BETA
Section: &DFT

Change the β parameter in Becke's exchange functional [55] to the value given on the next line.

BENCHMARK
Section: &CPMD

This keyword is used to control some special features related to benchmarks. If you want to know more, have a look in the source code.

BERENDSEN {IONS,ELECTRONS,CELL}
Section: &CPMD

Use a simple Berendsen-type thermostat[56] to control the respective temperature of ions, electrons, or cell. The target temperature and time constant τ (in a.u.) are read from the next line.
These thermostats are a gentler alternative to the **TEMPCONTROL** mechanism to thermalize a system. For production runs, please use the corresponding **NOSE** or **LANGEVIN** thermostats, as the Berendsen scheme does not represent any defined statistical mechanical ensemble.

BFGS
Section: &CPMD

Use a quasi-Newton method for optimization of the ionic positions. The approximated Hessian is updated using the Broyden-Fletcher-Goldfarb-Shano procedure [57].

BICANONICAL ENSEMBLE {CHEMICALPOTENTIAL, XWEIGHT} [INFO]
Section: &CPMD

Use the bicanonical ensemble AIMD for simulation of open systems [58].
Note, requesting this runtype cpmd must be executed on more then one MPI thread. Within the bicanonocal ensemble one out of two sampling modes must be specified in the input:

CHEMICALPOTENTIAL requests to sample the bicanonical ensemble at constant chemical potential ($\mu_1^\dagger$) or

XWEIGHT requests to sample the bicanonical ensemble at constant bicanonical weight (x_1).

Default is to use no INFO printing at each AIMD step.

Two parameters read in from next line:

[**chemicalPotential** | **xWeight**] [**temperature**]

chemicalPotential ($\mu_1^\dagger$) if CHEMICALPOTENTIAL run type was requested.

A reasonable value can be obtained taking the ensemble average of $\langle\mu_1^\dagger\rangle_{x_1=0.5}$ of a converged bicanonical AIMD simulation using the XWEIGHT run type (see Ref. [58]).

xWeight (x_1) if XWEIGHT run type was requested.

The value of **xWeight** must be within the interval $[0, 1]$. Setting the value of **xWeight** to 0 and 1 corresponds to the canonical limits of canonical system 2 and system 1, *i.e.* small and the large system, respectively.

temperature specifies the value of the temperature used to calculate the γ_1 prefactor (see Ref. [58]).

Preparation of the INPUT (see also CPMD regtests BICAN):

The simulation setup requires two input files with the second one having the same name plus the extension "_2". The two files are for the large and the small canonical system, system 1 and system 2, respectively. The two input files must differ in one atom. This atom must be declared as the _last_ atomic species in the input of system 1. Thus the INPUT for system 2 is created from 1 by removing that species and then apply optional changes, *e.g.* modify the number of electrons in that system via the CHARGE keyword. Note, positions and velocities of the atomic nuclei must be the same in both systems. A restart from preoptimized electronic wavefunctions of the individual canonical systems is recommended (see Sec. 8 and CPMD regtests BICAN/mdX0_50, BICAN/c-cnf1, BICAN/c-cnf2). The output file of system 2 is written on OUTPUT_CNF2 while that of system 1 is stdout. The suffixes "_CNF1" and "_CNF2" are used to specify the canonical systems for the other cpmd files. For example the content of the file ENERGIES of a canoncal cpmd run is written to the files ENERGIES_CNF1 and ENERGIES_CNF2. In the bicanonical AIMD the ENERGIES then stores: nfi, chemical potential, xweight, ΔU, and $\mu_{x_1=0.5}$. Where ΔU is the difference of the total energy of system 1 and system 2 at timestep nfi.
Note: For the moment bicanonical ensemble is only implemented for MOLECULAR DYNAMICS. Other MD types BO, EH, PT, CLASSICAL or FILE as well as runtypes QMMM, path integral or surface hopping are not supported.

BLOCKSIZE STATES

Section: &CPMD

Parameter read in from next line.
NSTBLK
Defines the minimal number of states used per processor in the distributed linear algebra calculations.
Default is to equally distribute states over all processors.

BOGOLIUBOV CORRECTION [OFF]

Section: &CPMD

Computes the Bogoliubov correction for the energy of the Trotter approximation or not.
Default is **no Bogoliubov correction**.
The keyword has to appear after **FREE ENERGY FUNCTIONAL**.

BOX WALLS

Section: &CPMD

The thickness parameter for soft, reflecting QM-box walls is read from the next line. This keyword allows to reverse the momentum of the particles ($\mathbf{p}_I \rightarrow -\mathbf{p}_I$) when they reach the walls of the simulation supercell in the case in which no periodic boundary conditions are applied. Specifically, in the unidimensional surface-like case, molecules departing from the surface are reflected back along the direction orthogonal to the surface, whereas in the bidimensional polymer-like case, they are reflected back in the two dimensions orthogonal to the "polymer" axis. Warning: This procedure, although keeping your particles inside the cell, affect the momentum conservation.
This feature is **disabled by default**

BROYDEN MIXING

Section: &CPMD

Parameters read in from next line.
BROYMIX, ECUTBROY, W02BROY, NFRBROY, IBRESET, KERMIX
These mean:

BROYMIX:	Initial mixing, e.g. 0.1; **default** value is **0.5**
ECUTBROY:	Cutoff for Broyden mixing. **DUAL*ECUT** is the best choice and the **default**
W02BROY:	w_0^2 parameter of Johnson [59]. **Default 0.01**
NFRBROY:	Number of Anderson mixing steps done before Broyden mixing. **Default** is **0**
IBRESET:	Number of Broyden vectors. 5 is usually a good value and the default.
KERMIX:	Kerker mixing according to the original definition of Ref. [48]. By default the mixing parameter is set to 0.

You can also specify some parameters with the following syntax:
[**BROYMIX=***BROYMIX*] [**ECUTBROY=***ECUTBROY*]
[**W02BROY=***W02BROY*] [**NFRBROY=***NFRBROY*]
[**IBRESET=***IBRESET*]
[**KERMIX=***KERMIX*]
Finally, you can use the keyword **DEFAULT** to use the default values.

CAYLEY

Section: &CPMD

Used to propagate the Kohn-Sham orbitals in **MOLECULAR DYNAMICS** EH and **PROPAGATION SPECTRA**. At present is the only propagation scheme available.

CDFT [NEWTON, DEKKER],[SPIN, ALL, PCGFI, RESWF, NOCCOR, HDA[AUTO, PHIOUT, PROJECT]]

Section: &CPMD

The main switch for constrained DFT. Parameters N_c, V_{init}, and MAXSTEP are read from the next line.
NEWTON, **DEKKER** (**off**) are switches to enable either the Newton or the Dekker optimisation scheme for the constraint. If neither of those are set a simple gradient scheme is used.
SPIN (**off**) if activated the constraint will act on the spin density instead of the charge density. This may help against excessive spin contamination.
ALL (**off**) activates dual spin and charge constraint, all inputs for N_c and V_{init} have to be given twice (first for charge then for spin)
PCGFI (**off**) instructs CPMD to do PCG for the first V optimisation cycle regardless of the choice of optimiser.
RESWF (**off**) if activated this switch re-initialises the wavefunction after each V optimisation step. This is useful if the wavefunction convergence between the optimisation steps is slow. Usage in conjunction with **INITIALIZE WAVEFUNCTION** RANDOM may help.
NOCCOR (**off**) if activated this switch turns off cutoff correction for the forces.
HDA (**off**) if activated this switch turns on the calculation of the transition matrix element between the constrained states given by N_c and $\hat{N}_c$ which is then read from the second line. For this keyword to take effect the **OPTIMIZE WAVEFUNCTION** option has to be activated.
Sub-options of **HDA**
AUTO (**off**) if activated this switch lets CPMD choose the constraint values for the transition matrix calculation. N_c is chosen from the initial charge distribution and $\hat{N}_c = -N_c$. It might be a good idea to use **INITIALIZE WAVEFUNCTION** ATOMS and **ATOMIC CHARGES** (&ATOM section) so that CPMD initialises the wavefunction with the desired pseudo wavefunction.
PHIOUT (**off**) if activated this switch tells CPMD to write out the overlap matrices $\Phi_{AA}, \Phi_{BB}, \Phi_{AB}$, and Φ_{BA} to the file PHI_MAT.
PROJECT (**off**) if activated this switch lets CPMD read in two reference states from RESTART.REF1 and RESTART.REF2 after the actual HDA calculation in order to project the two constrained states on them and thus calculate the diabatic transition matrix element in an orthogonalised "dressed" basis.
If CDFT is activated the program writes the current V value to CDFT_RESTART every time the RESTART file is written.

CELL {ABSOLUTE, DEGREE, VECTORS}

Section: &SYSTEM

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. CPMD will check those values, unless you turn off the test via **CHECK SYMMETRY**. 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. With the keyword **VECTORS**, the lattice vectors $a1$, $a2$, $a3$ are read from the next line instead of the 6 numbers. In this case the **SYMMETRY** keyword is not used.

CENTER MOLECULE [OFF]

Section: &CPMD

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.

CENTROID DYNAMICS

Section: &PIMD

Adiabatic centroid molecular dynamics, see Ref. [60, 61, 62] for theory and details of our implementation, which yields quasiclassical dynamics of the nuclear centroids at a specified temperature of the non–centroid modes. This keyword makes only sense if used in conjunction with the normal mode propagator via the keyword NORMAL MODES *and* FACSTAGE > 1.0 *and* WMASS $= 1.0$. The centroid adiabaticity control parameter FACSTAGE, which makes the non-centroid modes artificially fast in order to sample adiabatically the quantum fluctuations, has to be chosen carefully; note that FACSTAGE $= 1/\gamma$ as introduced in Ref. [62] in eq. (2.51).

CG-ANALYTIC

Section: &RESP

The number of steps for which the step length in the conjugate gradient optimization is calculated assuming a quadratic functional E(2) (quadratic in the linear response vectors). No accuracy impact, pure convergence speed tuning.
Default value is **3** for NMR and **99** otherwise.

CG-FACTOR

Section: &RESP

The analytic length calculation of the conjugate-gradient step lengths yields in general a result that is slightly too large. This factor is used to correct for that deficiency. No accuracy impact, pure convergence speed tuning.
Default is **0.8** .

CHANGE BONDS
Section: &ATOMS

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
```

CHARGES
Section: &PROP

Calculate atomic charges. Charges are calculated according to the method of Hirshfeld [63] and charges derived from the electrostatic potential [64].

CHARGE
Section: &SYSTEM

The total charge of the system is read from the next line.
Default is **0** .

CHECK MEMORY
Section: &CPMD

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

CHECK SYMMETRY [OFF]
Section: &SYSTEM

The precision with which the conformance of the **CELL** parameters are checked against the (supercell) **SYMMETRY** is read from the next line. With older versions of CPMD, redundant variables could be set to arbitrary values; now **all** values have to conform. If you want the old behavior back, you can turn the check off by adding the keyword **OFF** or by providing a negative precision. **Default** value is: **1.0e-4**

CLASSICAL CELL [ABSOLUTE, DEGREE]

Section: &SYSTEM

Not documented.

CLASSICAL TEST

Section: &PIMD

Test option to reduce the path integral branch to the classical code for the special case $P = 1$ in order to allow for a one-to-one comparison to a run using the standard branch of CPMD. It works only with primitive propagator, i.e. not together with NORMAL MODES, STAGING and/or **DEBROGLIE** CENTROID.

CLASSTRESS

Section: &CPMD

Not documented.

CLUSTER

Section: &SYSTEM

Isolated system such as a molecule or a cluster. Same effect as **SYMMETRY** 0, but allows a non-orthorhombic cell. Only rarely useful.

CMASS

Section: &CPMD

The fictitious mass of the cell in atomic units is read from the next line.
Default value is **200**

COMBINE SYSTEMS [REFLECT,NONORTH,SAB]

Section: &CPMD

Read in two wavefunctions from RESTART.R1 and RESTART.R2 and combine them into RESTART.1 which can then be used in an FODFT calculations. The option NONORTH disables orthogonalisation of the combined WF's. Parameters NTOT1, NTOT2, NSUP1, NSUP2 are read from the next line.
NTOT1/NTOT2 total number of electrons in state 1/2 (mandatory).
NSUP1/NSUP2 number of alpha electrons in state 1/2 (only LSD).
If the option REFLECT is given a fifth parameter (CM_DIR) is read and the WF given in RESTART.R2 will be either mirrored through the centre of the box (CM_DIR=0), mirrored through the central yz-plane of the box (CM_DIR=1) or if CM_DIR=4 mirrored through the central yz-plane and translated in x direction by CM_DR (sixth parameter to be read).
If the option SAB is set, write out the overlap matrix element between orbitals K and L. Parameters K and L are read from the next line.
After combining the wavefunctions CPMD will exit. For this option to work the RESTART option and **OPTIMIZE WAVEFUNCTION** have to be activated.

COMPRESS [WRITEnn]

Section: &CPMD

Write the wavefunctions with nn bytes precision to the restart file.
Possible choices are `WRITE32`, `WRITE16`, `WRITE8` and `WRITEAO`.
`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. See section 9.5.3 for more details.

CONDUCTIVITY

Section: &PROP

Computes the optical conductivity according to the Kubo-Greenwod formula

$$\sigma(\omega) = \frac{2\pi e^2}{3m^2 V_{\text{cell}}} \frac{1}{\omega} \sum_{i,j} (f_i - f_j) |\langle \psi_i | \hat{\mathbf{p}} | \psi_j \rangle|^2 \delta(\epsilon_i - \epsilon_j - \hbar\omega)$$

where ψ_i are the Kohn-Sham eigenstates, ϵ_i their corresponding eigenvalues, f_i the occupation number and the difference $f_i - f_j$ takes care of the fermionic occupancy. This calculation is executed when the keyword PROPERTIES is used in the section &CPMD ... &END. In the section &PROP ... &END the keyword CONDUCTIVITY must be present and the interval interval $\Delta\omega$ for the calculation of the spectrum is read from the next line. Note that, since this is a "PROPERTIES" calculation, *you must have previously computed the electronic structure of your system and have a consistent **RESTART** file ready to use.* Further keyword: `STEP=0.14`, where (e.g.) 0.14 is the bin width in eV of the $\sigma(\omega)$ histogram if you want it to be different from $\Delta\omega$. A file MATRIX.DAT is written in your working directory, where all the non-zero transition amplitudes and related informations are reported (see the header of MATRIX.DAT). An example of application is given in Refs. [65, 66].

CONFINEMENT POTENTIAL
Section: &ATOMS

The use of this label activates a spherical Gaussian confinement potential in the calculation of the form factor of pseudopotentials. In the next line(s) two parameters for each atomic species must be supplied: the amplitude α and the cut off radius r_c. The Gaussian spherical amplitude is computed as $A = \pi^{3/2} r_c^3 \cdot \alpha$ and the Gaussian confinement potential reads

$$V(\mathbf{G}) = \sum_{\mathbf{G}} A \cdot |\mathbf{G}| \cdot e^{-G^2 r_c^2/4}$$

being $\mathbf{G}$ the G-vectors, although in practice the loop runs only on the G-shells $G = |\mathbf{G}|$.

CONJUGATE GRADIENTS [ELECTRONS, IONS, NOPRECONDITIONING]
Section: &CPMD

For the electrons, the keyword is equivalent to **PCG**. The `NOPRECONDITIONING` parameter only applies for electrons. For the ions the conjugate gradients scheme is used to relax the atomic positions.

CONSTANT CUTOFF
Section: &SYSTEM

Apply a cutoff function to the kinetic energy term [67] 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 Rydberg).

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

CONSTRAINTS ... END CONSTRAINTS
Section: &ATOMS

With this option you can specify several constraints and restraints on the atoms. (see section 9.5.2 for more information on the available options and the input format).

CONVERGENCE [ADAPT, ENERGY, CALFOR, RELAX, INITIAL]
Section: &CPMD

The adaptive convergence criteria for the wavefunction during a geometry optimization are specified. For more informations, see [26]. The ratio *TOLAD* between the smallest maximum component of the nuclear gradient reached so far and the maximum allowed component of the electronic **gradient** is specified with **CONVERGENCE ADAPT**. This criterion is switched off once the value *TOLOG* given with **CONVERGENCE ORBITALS** is reached. By default, the adaptive gradient criterion is not active. A reasonable value for the parameter *TOLAD* is 0.02.
If the parameter *TOLENE* is given with **CONVERGENCE ENERGY**, in addition to the gradient criterion for the wavefunction, the energy change between two wavefunction optimization cycles must be smaller than the energy change of the last accepted geometry change multiplied by *TOLENE* for the wavefunction to be considered converged. By default, the adaptive energy criterion is not active. It is particularly useful for **transition state search** with P-RFO, where the trust radius is based on the quality of energy prediction. A reasonable value for *TOLENE* is 0.05.
To save CPU time, the gradient on the ions is only calculated if the wavefunction is almost converged. The parameter *TOLFOR* given with **CONVERGENCE CALFOR** is the ratio between the convergence criteria for the wavefunction and the criteria whether the gradient on the ions is to be calculated. **Default** value for *TOLFOR* is **3.0** .
If the wavefunction is very slowly converging during a geometry optimization, a small nuclear displacement can help. The parameter *NSTCNV* is given with **CONVERGENCE RELAX**. Every *NSTCNV* wavefunction optimization cycles, the convergence criteria for the wavefunction are relaxed by a factor of two. A geometry optimization step resets the criteria to the unrelaxed values. By default, the criteria for wavefunction convergence are never relaxed.
When starting a geometry optimization from an unconverged wavefunction, the nuclear gradient and therefore the adaptive tolerance of the electronic gradient is not known. To avoid the **full convergence** criterion to be applied at the beginning, a convergence criterion for the wavefunction of the initial geometry can be supplied with **CONVERGENCE INITIAL**. By default, the initial convergence criterion is equal to the full convergence criterion.

CONVERGENCE [ORBITALS, GEOMETRY, CELL]
Section: &CPMD

The convergence criteria for optimization runs is 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 $\mathbf{10^{-5}}$ for the wavefunction, $\mathbf{5{\times}10^{-4}}$ for the ions and **1.0** for the cell. For diagonalization schemes the first value is the biggest variation of a density component. **Defaults** are $\mathbf{10^{-3}}$ and $\mathbf{10^{-3}}$.

CONVERGENCE CONSTRAINT
Section: &CPMD

Set constraint convergence parameters. Parameters VCCON and VCCONU are read from the next line:
VCCON $\in \mathbb{R}_+$ is the maximally allowed total deviation of the constraint from the desired value N_c.
VCCONU $\in \mathbb{R}_+$ is the upper bound for the deviation in MD runs, excess of which triggers a new optimisation of V.
Defaults are $\mathbf{10^{-5}}$ and **VCCON** .

CONVERGENCE
Section: &LINRES

Convergence criterion for linear response calculations.
Default value is $\mathbf{10^{-5}}$.

CONVERGENCE
Section: &RESP

Convergence criterion on the gradient $\delta E/\delta\psi^*$ **Default** value is $\mathbf{10^{-5}}$.

CORE SPECTRA
Section: &PROP

Computes the X-ray adsorption spectrum and related transition matrix elements according to Ref. [68]. This calculation is executed when the keyword PROPERTIES is used in the section &CPMD ... &END. In the section &PROP ... &END the keyword CORE SPECTRA must be present and the core atom number (e.g. 10 if it is the $10th$ atom in your list) and core level energy (in au) are read from the next line, while in the following line the n and l quantum numbers of the selected core level, along with the exponential factor a of the STO orbital for the core level must be provided. In the case of $1s$ states, the core orbital is reconstructed as

$$\psi_{1s}(r) = 2a^{\frac{3}{2}} r \cdot \exp(-a \cdot r)$$

and it is this a value in au that must be supplied in input. As a general rule, first-row elements in the neutral case have the following a values: B (4.64), C (5.63), N (6.62), O (7.62). For an excited atom these values would be of course a bit larger; e.g. for O it is 7.74453, i.e. 1.6 % larger. Since this is a "PROPERTIES" calculation, *you must have previously computed the electronic structure of your system and have a consistent* ***RESTART*** *file ready to use.* A file XRAYSPEC.DAT is written in your working directory, containing all the square transition amplitudes and related informations, part of which are also written in the standard output. Waring: in order to use this keyword you need special pseudopotentials. These are provided, at least for some elements, in the PP library of CPMD and are named as *_HOLE.psp

COULOMB ATTENUATION
Section: &DFT

Activates the long-range correction/Coulomb attenuation method (**LC, CAM**). With this keyword, all selected *internal* LDA or GGA exchange functionals will be Coulomb-attenuated, and the corresponding exact exchange contribution is added by default. The three CAM parameters α, β and μ are read from the next line. Setting $\alpha = 0$ and $\beta = 1.0$ is equivalent to using the long-range correction (LC) method. This option is only available in combination with the new **XC_DRIVER**. The keyword is aliased and may be abbreviated by **CAM**.

COUPLINGS {FD=ϵ,PROD=ϵ} [NAT]
Section: &SYSTEM

Calculate non-adiabatic couplings [69] using finite differences (FD and PROD are two different finite-difference approximations). The displacement ϵ is expected in atomic units. If NAT=n is given, the coupling vector acting on only a subset of n atoms is calculated. In this case, a line containing n atom sequence numbers is expected. See **COUPLINGS NSURF**.

COUPLINGS LINRES {BRUTE FORCE,NVECT=n} [THR,TOL]
Section: &SYSTEM

Calculate non-adiabatic couplings [69] using linear-response theory. With BRUTE FORCE, the linear response to the nuclear displacements along all Cartesian coordinates is calculated. With NVECT=n, at most n cycles of the iterative scheme in [69] are performed. However, the iterative calculation is also stopped earlier if its contribution to the non-adiabatic coupling vector is smaller a given tolerance (TOL=C_{tol}). In the case of the iterative scheme, also the option THR can be given, followed by three lines each containing a pair of a threshold contribution to the non-adiabatic coupling vector and a tolerance for the linear-response wavefunction (see [69]). Do not forget to include a &LINRES section in the input, even if the defaults are used. See **COUPLINGS NSURF**.

COUPLINGS NSURF
Section: &SYSTEM

Required for non-adiabatic couplings: the Kohn-Sham states involved in the transition. For the moment, only one pair of states makes sense, NSURF=1. On the following line, the orbital numbers of the two Kohn-Sham states and a weight of 1.0 are expected. For singlet-singlet transitions, the ROKS-based Slater transition-state density (**LOW SPIN EXCITATION LSETS**) should be used. For doublet-doublet transitions, the local spin-density approximation (**LSD**) with the occupation numbers (**OCCUPATION**, **NSUP**, **STATES**) of the corresponding Slater transition-state density should be used.

CP_GROUPS
Section: &CPMD

Set the number of groups to be used in the calculation. Default is 1 group. The number of groups is read from the next line and shall be a divisor of the number of nodes in a parallel run.

CP_LIBRARY_ONLY
Section: &DFT

Use the new xc driver, but use only functionals that are available internally (this excludes libxc). *Cf.* **XC_DRIVER**.

CUBECENTER
Section: &PROP

Sets the center of the cubefiles produced by the **CUBEFILE** flag. The next line has to contain the coordinates of the center in Bohr or Angstrom, depending on whether the **ANGSTROM** keyword was given. **Default** is the geometric center of the system.

CUBEFILE ORBITALS,DENSITY HALFMESH
Section: &PROP

Plots the requested objects in .CUBE file format. If ORBITALS are demanded, the total number as well as the indices have to be given on the next and second next line. HALFMESH reduces the number of grid points per direction by 2, thus reducing the file size by a factor of 8.

CUTOFF [SPHERICAL,NOSPHERICAL]
Section: &SYSTEM

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.

CZONES [SET]
Section: &CPMD

Activates convergence zones for the wavefunction during the **CDFT** constraint minimisation. If SET is set the parameters CZONE1, CZONE2, and CZONE3 are read from the next line and CZLIMIT1 and CZLIMIT2 from the line after.
CZONE1 10^{-3} ,CZONE2 10^{-4} ,CZONE3 10^{-5} $\in \mathbb{R}_+$ are the orbital convergences in zones 1-3, respectively.
CZLIMIT1 **0.3** , CZLIMIT2 **0.1** $\in \mathbb{R}_+$ define the boundaries between zone 1-2 and 2-3, respectively.

DAMPING {IONS,ELECTRONS,CELL}

Section: &CPMD

Add a damping factor $f_{damp}(x) = -\gamma \cdot v(x)$ to the ionic, electronic, or cell forces in every time step. The scaling factor γ is read from the next line. Useful values depend on the employed masses are generally in the range $5.0 \rightarrow 50.0$.
Damping can be used as a more efficient alternative to **ANNEALING** for wavefunction, geometry or cell optimization (and particularly combinations thereof) of systems where the faster methods (e.g. **ODIIS**, **PCG**, **LBFGS**, **GDIIS**) fail to converge or may converge to the wrong state.

DAVIDSON DIAGONALIZATION

Section: &CPMD

Use Davidson diagonalization scheme.[70]

DAVIDSON PARAMETER

Section: &CPMD

This keyword controls the Davidson diagonalization 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 $\mathbf{10^{-5}}$ 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.

DAVIDSON PARAMETER

Section: &TDDFT

The maximum number of Davidson iterations, the convergence criteria for the eigenvectors and the maximal size of the Davidson subspace are set. The three parameters *ndavmax*, *epstdav*, *ndavspace* are read from the next line.
Default values are **100** , $\mathbf{10^{-10}}$ and **10** .

DAVIDSON RDIIS

Section: &TDDFT

This keyword controls the residual DIIS method for TDDFT diagonalization. This method is used at the end of a DAVIDSON diagonalization for roots that are not yet converged. The first number gives the maximum iterations, the second the maximum allowed restarts, and the third the maximum residual allowed when the method is invoked.
Default values are **20** , **3** and 10^{-3} .

DEBROGLIE [CENTROID]

Section: &PIMD

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. [71]. Using DEBROGLIE CENTROID each nuclear position obtained from the &ATOMS ... &END section serves as the centroid (center of geometry) for obtaining the centroid (center of geometry) for obtaining the P normal modes in three dimensions, see e.g. Ref. [15]. 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.

DCACP

Section: &CPMD

Activates the DCACP scheme [176] (dispersion-corrected atom-centred potentials) in order to account for van der Waals-interactions. Advanced settings can be specified in the &VDW section.

DCACP Z=z

Section: &VDW

Set custom DCACP parameters for the element with core charge Z =z. The values of the DCACP projectors σ_1 and σ_2 are read from the next line.
The keyword can be repeated multiple times, once for every element. Useful if the desired parameters are not available in the internally coded library, *e.g.* for an exotic xc functional.

DEBUG CODE

Section: &CPMD

Very verbose output concerning subroutine calls for debugging purpose.

DEBUG FILEOPEN
Section: &CPMD

Very verbose output concerning opening files for debugging purpose.

DEBUG FORCES
Section: &CPMD

Very verbose output concerning the calculation of each contribution to the forces for debugging purpose.

DEBUG MEMORY
Section: &CPMD

Very verbose output concerning memory for debugging purpose.

DEBUG NOACC
Section: &CPMD

Do not read/write accumulator information from/to the **RESTART** file. This avoids putting timing information to the restart and makes restart files identical for otherwise identical runs.

DENSITY CUTOFF [NUMBER]
Section: &SYSTEM

Set the plane wave energy cutoff for the density. The value is read from the next line. The density cutoff is usally automatically determined from the wavefunction **CUTOFF** via the **DUAL** factor.
With the additional flag **NUMBER** the number of plane waves can be specified directly. This is 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.

DIAGONALIZER {DAVIDSON,NONHERMIT,PCG} [MINIMIZE]
Section: &TDDFT

Specify the iterative diagonalizer to be used.
Defaults are *DAVIDSON* for the Tamm–Dancoff method, *NONHERMIT* (a non-hermitian Davidson method) for TDDFT LR and *PCG* (Conjugate gradients) for the optimized subspace method. The additional keyword *MINIMIZE* applies to the PCG method only. It forces a line minimization with quadratic search.
Default is **not to use line minimization** .

DIAGONAL [OFF]
Section: &HARDNESS

Not documented

DIFF FORMULA
Section: &LINRES

Number of points used in finite difference formula for second derivatives of exchange–correlation functionals. Default is two point central differences.

DIIS MIXING
Section: &CPMD

Use the direct inversion iterative scheme to mix density.
Read in the next line the number of previous densities (NRDIIS) for the mixing (however not useful).

DIIS MIXING [$N = n$]
Section: &CPMD

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.

DIPOLE DYNAMICS {SAMPLE,WANNIER}
Section: &CPMD

Calculate the dipole moment [72, 73] 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[18, 74, 75]. The procedure used is equivalent (for single k-point) to Boys localization.
The produced output is IONS+CENTERS.xyz, IONS+CENTERS, DIPOLE, WANNIER_CENTER and WANNIER_DOS. The localization procedure is controlled by the following keywords.

DIPOLE MOMENT [BERRY, RS]
Section: &PROP

Calculate the dipole moment.
Without the additional keywords **BERRY** or **RS** this is only implemented for simple cubic and fcc supercells. The keyword **RS** requests the use of the real-space algorithm. The keyword **BERRY** requests the use of the Berry phase algorithm.
Default is to use the real-space algorithm.

DISCARD [OFF, PARTIAL, TOTAL, LINEAR]
Section: &RESP

Request to discard trivial modes in vibrational analysis from linear response (both **PHONON** and **LANCZOS**).

OFF = argument for performing no projection.
PARTIAL = argument for projecting out only translations (this is the default).
TOTAL = argument for projecting both rotations and translations.
LINEAR = argument for projecting rotations around the $C - \infty$ axis in a linear molecule (not implemented yet).

DISTRIBUTED LINALG {ON, OFF}
Section: &CPMD

Perform linear algebra calculations using distributed memory algorithms. Setting this option ON will also enable (distributed) initialization from atomic wavefunctions using a parallel Lanczos algorithm [76]. Note that distributed initialization is not available with **KPOINTS** calculations. In this case, initialization from atomic wavefunctions will involve replicated calculations.
When setting **LINALG ON** the keyword **BLOCKSIZE STATES** becomes relevant (see entry). The number of **BLOCKSIZE STATES** must be an **exact divisor** of the number of **STATES**.

DISTRIBUTE FNL
Section: &CPMD

The array `FNL` is distributed in parallel runs.

DONOR
Section: &SYSTEM

Set the **CDFT** donor atoms. Parameter NACCR must be specified next to the keyword. NDON $\in \mathbb{R}_+$ is the number of Donor Atoms (N being the total number of atoms). If NDON> 0 the indices of NDON atoms separated by whitespace are read from the next line else only use an Acceptor group in the CDFT weight.

DUAL

Section: &SYSTEM

The ratio between the wavefunction energy **CUTOFF** and the **DENSITY CUTOFF** is read from the next line.
Default is **4**.
There is little need to change this parameter, except when using ultra-soft pseudopotentials, where the wavefunction cutoff is very low and the corresponding density cutoff is too low to represent the augmentation charges accurately. In order to maintain good energy conservation and have good convergence of wavefunctions and related parameters, **DUAL** needs to be increased to values of 6–10.
Warning: You can have some trouble if you use the **DUAL** option with the symmetrization of the electronic density.

DUMMY ATOMS

Section: &ATOMS

The definition of dummy atoms follows this keyword.
Three different kinds of dummy atoms are implemented. Type 1 is fixed in space, type 2 lies at the arithmetic mean, type 3 at the center of mass 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, TYPE3, TYPE4**.
For type 1 dummy atoms the label is followed by the Cartesian coordinates.
For type 2 and type 3 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. A negative number of atoms stands for all atoms. For type 4, the dummy atom is defined as a weighed average of coordinates of real atoms with user-supplied weights. This feature is useful e. g. in constrained dynamics, because allows to modify positions and weights of dummy atoms according to some relevant quantity such as forces on selected atoms. Example:

```
DUMMY ATOMS
3
TYPE1  0.0  0.0  0.0
TYPE2  2    1    4
TYPE3  -1
```

Note: Indices of dummy atoms always start with total-number-of-atoms plus 1. In the case of a Gromos-QM/MM interface simulations with dummy hydrogen atoms for capping, it is total-number-of-atoms plus number-of-dummy-hydrogens plus 1

EIGENSYSTEM

Section: &RESP

Not documented.

ELECTRONIC SPECTRA

Section: &CPMD

Perform a TDDFT calculation [77, 78] to determine the electronic spectra. See below under Electronic Spectra and under the other keywords for the input sections &LINRES and &TDDFT for further options.

ELECTROSTATIC POTENTIAL [SAMPLE=nrhoout]

Section: &CPMD

Store the electrostatic potential on file. The resulting file is written in platform specific binary format. You can use the cpmd2cube tool to convert it into a Gaussian cube file for visualization. Note that this flag automatically activates the **RHOOUT** flag as well.
With the optional keyword **SAMPLE** the file will be written every *nrhoout* steps during an MD trajectory. The corresponding time step number will be appended to the filename.

ELF [PARAMETER]

Section: &CPMD

Store the total valence density and the valence electron localization function ELF [79, 80, 81, 82] on files. The default smoothing parameters for ELF can be changed optionally when specifying in addition the PARAMETER keyword. Then the two parameters “elfcut” and “elfeps” are read from the next line. The particular form of ELF that is implemented is defined in the header of the subroutine elf_utils.mod.F90.
Note 1: it is a *very* good idea to increase the planewave cutoff and then specify “elfcut” = 0.0 and “elfeps” = 0.0 if you want to obtain a smooth ELF for a given nuclear configuration. In the case of a spin–polarized (i.e. spin unrestricted) DFT calculation (see keyword **LSD**) in addition the spin–polarized average of ELF as well as the separate α– and β–orbital parts are written to the files LSD_ELF, ELF_ALPHA and ELF_BETA, respectively; see Ref. [83] for definitions and further infos.
Note 2: ELF does not make much sense when using Vanderbilt’s ultra-soft pseudopotentials!

EMASS

Section: &CPMD

The fictitious electron mass in atomic units is read from the next line.
Default is **400 a.u.**.

ENERGY PROFILE

Section: &SYSTEM

Perform an energy profile calculation at the end of a wavefunction optimization using the ROKS or ROSS methods.

ENERGYBANDS

Section: &CPMD

Write the band energies (eigenvalues) for k points in the file ENERGYBANDS.

EPR *options*

Section: &RESP

Calculate the EPR g tensor for the system. This routine accepts most, if not all, of the options available in the NMR routine (RESTART, NOSMOOTH, NOVIRTUAL, PSI0, RHO0, OVERLAP and FULL). Most important new options are:
FULL SMART: does a calculation with improved accuracy. A threshold value (between 0 and 1) must be present on the next line. The higher the threshold value, the lower the computational cost, but this will also reduce the accuracy (a bit). Typically, a value of 0.05 should be fine.
OWNOPT: for the calculation of the g tensor, an effective potential is needed. By default, the EPR routine uses the local potential ($V_{LOC} = V_{PP,LOC}+V_{HARTREE}+V_{XC}$). This works well with Goedecker pseudopotentials, but rather poor with Troullier-Martins pseudopotentials. When using this option, the following potential is used instead:

$$V_{EFF} = -\frac{Z}{r}\text{erf}(r/r_c) + V_{HARTREE} + V_{XC}$$

and r_c (greater than 0) is read on the next line.
HYP: calculates the hyperfine tensors. See epr_hyp_utils.mod.F90 for details.

Contact Reinout.Declerck@UGent.be should you require further information.

EXCHANGE CORRELATION TABLE [NO]

Section: &DFT

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.
Note that this keyword is only relevant when using **OLDCODE** and even then it is set to **NO** be default. Previous default values were 30000 and 2.0.

EXCITED DIPOLE

Section: &PROP

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.

EXTERNAL POTENTIAL {ADD}

Section: &CPMD

Read an external potential from file. With ADD specified, its effects is added to the forces acting on the ions.

EXT_PULSE

Section: &PTDDFT

A Dirac type pulse is applied to the KS orbitals before propagation. This keyword is used with the principal keyword **PROPAGATION SPECTRA**. The intensity of the perturbing field is read from the next line.

EXT_POTENTIAL

Section: &PTDDFT

An external potential is applied to the KS orbitals before propagation. This keyword is used with the principal keyword **MOLECULAR DYNAMICS EH** or **PROPAGATION SPECTRA**. The type of the perturbation is specified with the keyword **PERT_TYPE** (in &PTDDFT) and its intensity with **PERT_AMPLI** (in &PTDDFT).

EXTERNAL FIELD

Section: &SYSTEM

Applies an external electric field to the system using the Berry phase. The electric field vector in AU is read from the next line.

EXTPOT

Section: &TDDFT

Non adiabatic (nonadiabatic, non-adiabatic) Tully's trajectory surface hopping dynamics using TDDFT energies and forces, coupled with an external field [84]. To be used together with the keywords **MOLECULAR DYNAMICS** BO, **TDDFT** in the &CPMD section, and **T-SHTDDFT** in the &TDDFT section. Do NOT use the keyword **T-SHTDDFT** together with the keyword **SURFACE HOPPING** in &CPMD, which invokes the SH scheme based on **ROKS** [85] (see **SURFACE HOPPING**).
This keyword follow the same principle as described for the keyword **T-SHTDDFT**, except that, in the present dynamics, the trajectory starts on the ground state and is coupled with an external field through the equations of motion for the amplitudes of Tully's trajectory surface hopping. According to the evolution of the amplitudes of the different excited states, the running trajectory can jump on an excited state. From there, deactivation through nonradiative processes is possible, within the normal trajectory surface hopping scheme.
Parameter *aampl*, *adir*, *afreq*, and *apara1* are read from the next line. The amplitude of the vector potential is provided in *aampl* and its polarization is given in *adir* (1 = x-polarized, 2 = y-polarized, 3 = z-polarized, 4 = all components). The keyword *afreq* gives the frequency of the field and *apara1* is a free parameter for a specific user-specified pulse.
Important points: the applied electromagnetic field needs to be hard coded in the subroutine sh_tddft_utils.mod.F90, in the subroutine SH_EXTPOT. The vector potential is used for the coupling with the amplitudes equations. Be careful to use a time step small enough for a correct description of the pulse. The pulse is printed in the file SH_EXTPT.dat (step, A(t), E(t)).

EXTRAPOLATE WFN {STORE}

Section: &CPMD

Read the number of wavefunctions to retain from the next line.
These wavefunctions are used to extrapolate the initial guess wavefunction in Born-Oppenheimer MD. This can help to speed up BO-MD runs significantly by reducing the number of wavefunction optimization steps needed through two effects: the initial guess wavefunction is much improved and also you do not need to converge as tightly to conserve energy. BO-MD without needs CONVERGENCE ORBITALS to be set to 1.0e-7 or smaller to maintain good energy conservation.
With the additional keyword **STORE** the wavefunction history is also written to restart files. See **RESTART** for how to read it back.

EXTRAPOLATE CONSTRAINT

Section: &CPMD

In a CDFT MD run extrapolate the next value for V using a Lagrange polynomial. The order k of the polynomial is read from the next line. **Default** is **k=5** , but it pays off to use the orderfinder.py python script on the ENERGIES file of a former run to estimate the optimal extrapolation order k_{opt}.

FACMASS

Section: &PIMD

Obtain the fictitious nuclear masses M_I' within path integral molecular dynamics from the real physical atomic masses M_I (as tabulated in the DATA ATWT / . . . / statement in atoms_utils.mod.F90) by *multiplying* them with the dimensionless factor WMASS that is read from the following line; if the NORMAL MODES or STAGING propagator is used obtain $M_I'^{(s)}$ = WMASS $\cdot M_I^{(s)}$ for *all* replicas $s = 1, \ldots, P$; see e.g. Ref. [62] eq. (2.37) for nomenclature.
Default value of WMASS is **1.0**

FACTOR

Section: &PATH

Step for propagating string (see [49]).
Default value is **1.0**

FILE FUSION

Section: &CPMD

Reads in two separate **RESTART** files for ground state and **ROKS** excited state and writes them into a single restart file.
Required to start **SURFACE HOPPING** calculations.

FILEPATH

Section: &CPMD

The path to the files written by CPMD (RESTART.x, MOVIE, ENERGIES, DENSITY.x etc.) is read from the next line. This overwrites the value given in the environment variable **CPMD_FILEPATH**. **Default** is the **current directory**.

FINITE DIFFERENCES

Section: &CPMD

The step length in a finite difference run for vibrational frequencies (VIBRATIONAL ANALYSIS keywords) is read from the next line.
With the keywords **COORD=***coord_fdiff(1..3)* and **RADIUS=***radius* put in the same line as the step length, you can specify a sphere in order to calculate the finite differences only for the atoms inside it. The sphere is centered on the position *coord_fdiff(1..3)* with a radius *radius* (useful for a point defect).

NOTE: The the step length for the finite difference is **always** in Bohr (default is 1.0d-2 a.u.). The (optional) coordinates of the center and the radius are read in either Angstrom or Bohr, depending on whether the **ANGSTROM** keyword is specified or not.

FIXRHO UPWFN [VECT LOOP WFTOL]

Section: &CPMD

Wavefunctions optimization with the method of direct inversion of the iterative subspace (DIIS), performed without updating the charge density at each step. When the orbital energy gradients are below the given tolerance or when the maximum number of iterations is reached, the KS energies and the occupation numbers are calculated, the density is updated, and a new wavefunction optimization is started. The variations of the density coefficients are used as convergence criterion. The default electron temperature is 1000 K and 4 unoccupied states are added. Implemented also for k-points. Only one sub-option is allowed per line and the respective parameter is read from the next line. The parameters mean:

VECT: The number of DIIS vectors is read from the next line. (ODIIS with 4 vectors is the default).

LOOP: the minimum and maximum number of DIIS iterations per each wfn optimization is read from the following line. Default values are 4 and 20.

WFTOL: The convergence tolerance for the wfn optimization during the ODIIS is read from the following line. The default value is 10^{-7}. The program adjusts this criterion automatically, depending on the convergence status of the density. As the density improves (when the density updates become smaller), also the wavefunction convergence criterion is set to its final value.

FORCE FIELD ... END FORCE FIELD

Section: &CLASSIC

FORCEMATCH

Section: &CPMD

Activates the QM/MM force matching procedure. This keywords requires the presence of a &QMMM ... &END section with a corresponding **FORCEMATCH ... END FORCEMATCH** block. See sections 11.16 and 11.16.11 for more details.

FORCE STATE

Section: &TDDFT

The state for which the forces are calculated is read from the next line. Default is for state 1.

FREE ENERGY FUNCTIONAL

Section: &CPMD

Calculates the electronic free energy using free energy density functional [14, 86, 87] from DFT at finite temperature.
This option needs additional keywords (free energy keywords).
By **default** we use **Lanczos diagonalization** with **Trotter factorization** and **Bogoliubov correction**. If the number of states is not specified, use $N_{electrons}/2+4$.

FREEZE QUANTUM

Section: &CLASSIC

Freeze the quantum atoms and performs a classical MD on the others (in QMMM mode only !).

FULL TRAJECTORY

Section: &CLASSIC

Not documented

FUNCTIONAL *functional_1, functional_2, . . .*
Section: &DFT

The xc functional(s) are specified on the same line, separated by a space.

XC_DRIVER:
Both functionals from the internal CP library and those linked from libxc are available within the same run and can be freely mixed when using the **XC_DRIVER**. Please note that the internal CP functionals offer a considerable performance advantage over libxc. While the code will default to the internal implementations, use of libxc for single functionals can be requested by the keyword **LIBRARY**. The functional code names for the functionals available internally follow the nomenclature introduced by libxc: The functional name must be preceeded by a tag specifying its level (hybrid, MGGA, GGA...). B3LYP would therefore be specified as HYB_GGA_XC_B3LYP.
Fixed (hybrid) exchange-correlation functionals available in the internal CP library include:

GGA_XC _BLYP, _BP86, _HCTH_93, _HCTH_120, _HCTH_147, _HCTH_407, _OLYP, _OPBE, _PBE, _PBE_SOL, _REVPBE
MGGA_XC _TPSS, _M06_L, _REVM06_L, _M11_L, _MN12_L, _MN15_L
HYB_GGA_XC _B3LYP, _CAM_B3LYP, _O3LYP, _PBE0, _PBE0_SOL, _REVPBE0
HYB_MGGA_XC _M05, _M05_2X, _M06, _M06_2X, _M06_HF, _M08_SO, _M08_HX, _M11, _MN12_SX

Important note on Minnesota functionals:
At the usual plane wave cutoff values, the Minnesota functionals require unusually dense integration grids. In order to avoid introducing noise, this should be achieved by increasing the value of the **DUAL**. For usage recommendations and more details: Bircher, Lopez-Tarifa and Rothlisberger[188], DOI: 10.1021/acs.jctc.8b00897.

The following exchange and correlation functionals are available as separate keywords within the internal CP library, both in spin-restricted and spin-unrestricted form, and can be freely combined:

LDA_X
LDA_C _PZ, _PW, _OB_PW, _VWN
GGA_X _B88, _OPTX, _PBE, _PBE_SOL, _REVPBE, _PBE_FLEX
GGA_C _LYP, _P86, _PBE, _PBE_SOL, _PBE_FLEX
MGGA_X _TPSS
MGGA_C _TPSS

The parameters for VWN used within the CP library correspond to what is sometimes denoted ‘VWN5’. The version of HCTH available in prior releases corresponds to GGA_XC_HCTH_120. The definitions for the LDA part of the P86, PBE and TPSS correlation have been updated to comply with the standard definitions of most codes, for certain functionals, the definitions used in CPMD versions prior to 4.3 may be accessed by using **OLD_DEFINITIONS**. A custom value of the parameter β used in GGA_X_B88 can be input using the keyword **BECKE BETA**. A completely customised parameterisation of PBE can be set up by using the _PBE_FLEX routines in combination with the PBE_FLEX_. . . keywords in the &DFT section. Since many different but highly specific parametrisations of PBE exist, this allows access to virtually all declinations based on the PBE xc equations.

NEWCODE and **OLDCODE**:
Single keyword for setting up XC-functionals. Only *one single* functional keyword can be specified in when using **OLDCODE** or **NEWCODE**.
Available functionals are NONE, SONLY, LDA (in PADE form),
BONLY, BP, BLYP, XLYP, GGA (=PW91), PBE, PBES, REVPBE,
HCTH, OPTX, OLYP, TPSS, PBE0, B1LYP, B3LYP, X3LYP,PBES,
HSE06

FUKUI [N=nf]

Section: &RESP

Calculates the response to a change of occupation number of chosen orbitals. The indices of these orbitals are read from the following nf lines (**default nf=1**). The orbitals themselves are not read from any **RESTART** file but from WAVEFUNCTION.* files generated with **RHOOUT** in the &CPMD section; to recall this the orbital numbers have to be negative, just like for the **RHOOUT** keyword.

A weight can be associated with each orbital if given just after the orbital number, on the same line. It corresponds to saying how many electrons are put in or taken from the orbital. For example;

```
FUKUI N=2
-i 1.0
-j -1.0
```

corresponds to the response to taking one electron from orbital i and put it in orbital j.

GAUGE {PARA,GEN,ALL}

Section: &LINRES

Gauge of the linear-response wavefunctions. Default is the parallel-transport gauge (PARA) for closed-shell calculations and a sensible combination of the parallel-transport gauge and the full-rotation gauge (GEN) for all other cases. The full-rotation gauge can be enforced for all states by selecting ALL. See [88].

GC-CUTOFF

Section: &DFT

On the next line the density cutoff for the calculation of the gradient correction has to be specified. The default value is 10^{-8}. Experience showed that for a small **CUTOFF** value (e.g. when using Vanderbilt pseudopotentials) a bigger values have to be used. A reasonable value for a 25 ryd cutoff calculation is $5 \cdot 10^{-6}$.
Warning: for the HCTH functional, since it includes both the xc part and the gradient correction in a unique functional, a GC-CUTOFF too high (e.g. $\geq 5 \cdot 10^{-5}$) could result in not including any xc part with uncontrolled related consequences.

GDIIS

Section: &CPMD

Use the method of direct inversion in the iterative subspace combined with a quasi-Newton method (using BFGS) for optimization of the ionic positions [89].The number of DIIS vectors is read from the next line.
GDIIS with **5 vectors** is the **default** method in optimization runs.

GENERATE COORDINATES

Section: &ATOMS

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.

GENERATE REPLICAS

Section: &PIMD

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.

GLE_LAMBDA

Section: &PIMD

Set the scaling factor λ of the generalized Langevin thermostat read from the next line for removing the resonances between the vibrations of the system and the harmonic potential representing the quantum kinetic energy term in the description of the path integral. The default value is 0.5 as suggested in Ref. [90].

GRADIENT CORRECTION *[functionals]*

Section: &DFT

Individual components of gradient corrected functionals can be selected. Rarely needed anymore, use the **FUNCTIONAL** keyword instead. Only available for **OLDCODE** and **NEWCODE**; but the new **XC_DRIVER** offers similar flexibility in combination with the **FUNCTIONAL** keyword.

Functionals implemented are for the exchange energy:
BECKE88 [55], **GGAX** [91] **PBEX** [92], **REVPBEX** [93],
HCTH [94], **OPTX** [95],**PBESX** [96]
and for the correlation part:
PERDEW86 [97], **LYP** [98], **GGAC** [91], **PBEC** [92], **REVPBEC** [93], **HCTH** [94] **OLYP** [95],**PBESC** [96].
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.

GSHELL

Section: &CPMD

Write a file **GSHELL** with the information on the plane waves for further use in S(q) calculations.

HAMILTONIAN CUTOFF
Section: &CPMD

The lower cutoff for the diagonal approximation to the Kohn-Sham matrix [99] 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.

HAMILTONIAN CUTOFF
Section: &RESP

The value where the preconditioner (the approximate Greens function $(V_{kin} + V_{coul} - \epsilon_{KS})^{-1}$) is truncated. HTHRS can also be used. Default value is 0.5.

HARDNESS
Section: &RESP

Not documented.

HARMONIC REFERENCE SYSTEM [OFF]
Section: &CPMD

Switches harmonic reference system integration [99] 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**.

HARTREE-FOCK
Section: &DFT

Do a Hartree–Fock calculation. This only works correctly for isolated systems. It should be used with care, it needs enormous amounts of CPU time.

HARTREE
Section: &DFT

Do a Hartree calculation. Only of use for testing purposes.

HESSCORE
Section: &CPMD

Calculates the partial Hessian after relaxation of the environment, equivalent to *NSMAXP=0* (**PRFO NSMAXP**).

HESSIAN [DISCO,SCHLEGEL,UNIT,PARTIAL]

Section: &CPMD

The initial approximate **Hessian** for a **geometry optimization** is constructed using empirical rules with the DISCO [100] or Schlegel's [101] parametrization or simply a unit matrix is used.
If the option **PARTIAL** is used the initial approximate Hessian for a geometry optimization is constructed from a block matrix formed of the parametrized Hessian and the partial Hessian (of the reaction core). If the reaction core spans the entire system, its Hessian is simply copied. The keywords **RESTART** PHESS are required.

HFX_BLOCK_SIZE

Section: &DFT

Set the block size for the HFX calculations. This keyword is active only when no thresholding of the integrals is used. Default is 2. The block size is read from the next line.

HFX CUTOFF

Section: &SYSTEM

Set an additional cutoff for wavefunction and density to be used in the calculation of exact exchange. Cutoffs for wavefunctions and densities are read from the next line in Rydberg units. Defaults are the same cutoffs as for the normal calculation. Only lower cutoffs than the defaults can be specified.

HFX_DISTRIBUTION

Section: &DFT

Set the block distribution for the parallel HFX calculations. This keyword is active only when no thresholding of the integrals is used. Default is BLOCK_CYCLIC. The distribution is read from the next line.

HFX SCREENING {WFC,DIAG,EPS_INT,RECOMPUTE_TWO_INT_LIST_EVERY}

Section: &DFT

Read value from the next line.
Perform the calculation of exact exchange using Wannier functions. Orbital pairs are screened according to the distance of the Wannier centers *WFC*, the value of the integrals *EPS_INT*, or only the diagonal terms are included (*DIAG*). *RECOMPUTE_TWO_INT_LIST_EVERY* allows to set how often the integral list is recomputed.

HIGH_LEVEL_FORCES {DFT, EXTERNAL}
Section: &MTS

Set the computational model for the calculation of high level forces in the MTS scheme. For now two options exists: (i) The forces are obtained from the DFT code internal to CPMD. In that case, the high level functional has to be set in the &DFT section with the **MTS_HIGH_FUNC** keyword. (ii) The forces are calculated by an external program which called via a script which should be named `EXT_HIGH_FORCES`. When the forces are needed, CPMD will write a file `geometry.xyz` containing the current geometry in xyz format (cartesian coordinates in Angstroms). Then the script will be called and CPMD will wait until a file `forces.xyz` becomes available with the new forces in xyz format (atomic units are expected). See the `get_external_forces` routine in the `interface_utils.mod.F90` file.
The keyword **LOW_LEVEL_FORCES** can be used in exactly the same way for the selection of the computational model used to calculate the low level forces in the MTS scheme. Similarly, if the DFT code internal to CPMD is chosen as low level, the keyword **MTS_LOW_FUNC** (&DFT section), can be used to select the low level density functional.

Default: Forces are calculated with the DFT code internal to CPMD.

HTHRS
Section: &LINRES

Threshold for Hessian in preconditioner for linear response optimizations. Default is 0.5.

HUBBARD [NORM,ORTHO,NUATM=*nuatm*,OCCMAT=*printfreq*,VERB]
Section: &DFT

Use the **HUBBARD-U** correction to partly correct the self interaction error of DFT. Specifying NORM and/or ORTHO causes the orbitals for the projections to be normalized and/or orthogonalized, respectively. VERB causes the occupation matrix to be printed in the output, otherwise it is printed to the file OCCMAT. The keyword OCCMAT=*printfreq* specifies the frequency in which the OCCMAT is written. Default is 1. *nuatm* is the number of atoms which are treated with the Hubbard U correction (U-atom). Default is 1.
For each U-atom the atom number, the U parameter [eV], the linear response parameter α [eV] and the number of different angular momenta for projectors is read from a new line. For each of those, the shell number of the projector and angular momentum l is read from the next lines.

IMPLICIT NEWTON RAPHSON {PREC, CONTINUE, VERBOSE, ALTERNATIVE, STEP} [N=nreg]
Section: &CPMD

Not documented.

INCLUDE_METALS
Section: &VDW

When using the **DCACP** dispersion-correction scheme, include contributions from metal centres, too (where available). This should only rarely be needed. In the vast majority of cases (metal complexes, metallo-organic compounds), contributions from the metal centres are vanishingly small.

INITIALIZATION
Section: &PIMD

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.

INITIALIZE WAVEFUNCTION **[RANDOM, ATOMS]** *[PRIMITIVE]*
Section: &CPMD

The initial guess for wavefunction optimization are either random functions or functions derived from the atomic pseudo-wavefunctions.
For INITIALIZE WAVEFUNCTION ATOMS PRIMITIVE, CPMD will use the occupation information given in the &BASIS section in order to construct a minimum spin multiplicity (i.e. doublet or singlet) initial wavefunction from the pseudo atomic orbitals. This option may be helpful to avoid excessive spin contamination in CDFT calculations (together with an already good initial guess for V) as it allows a strict initial localisation of excess spins on any atom species.
Default is to use the **atomic pseudo-wavefunctions**.

INTERACTION
Section: &RESP

Not documented.

INTERFACE **{EGO,GMX} {[MULLIKEN, LOWDIN, ESP, HIRSHFELD],PCGFIRST}**
Section: &CPMD

Use CPMD together with a classical molecular dynamics code. CPMD and the classical MD code are run simultaneously and communicate via a file based protocol. See the file egointer_utils.mod.F90 for more details. This needs a specially adapted version of the respective classical MD code. So far, there is an interface[16, 33] to the MD programs ego[102, 103] and Gromacs[104].
When using the suboption PCGFIRST the code will use **PCG** MINIMIZE on the very first wavefunction optimization and then switch back to DIIS.

INTFILE [READ,WRITE,FILENAME]
Section: &CPMD

This keyword means *Interface File* and allows to select a special file name in the reading and writing stages. The file name (max 40 characters) must be supplied in the next line.

ISOLATED MOLECULE
Section: &CPMD

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 **unless** the keyword **QMMM** is set as well. It allows studying an isolated molecule or cluster within periodic boundary conditions.

ISOTOPE
Section: &ATOMS

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 (in a.m.u.) of the respective species has to be specified (in order of their definition).

ISOTROPIC CELL
Section: &SYSTEM

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

KEEPREALSPACE
Section: &RESP

Like the standard CPMD option, this keeps the C0 ground state wavefunctions in the direct space representation during the calculation. Can save a lot of time, but is incredibly memory intensive.

KOHN-SHAM ENERGIES [OFF,NOWAVEFUNCTION]
Section: &CPMD

Calculation of the Kohn-Sham energies and the corresponding orbitals [8].
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 restarts).

KSHAM [MATRIX,ROUT,STATE]
Section: &CPMD

Write out the Kohn-Sham Hamiltonian Matrix in the orbital basis given in the RESTART file to KS_HAM. For this option to work the **RESTART** option and **OPTIMIZE WAVEFUNCTION** have to be activated. This option is useful for fragment orbital DFT (FODFT) calculations. Orbitals for the output of the FO-DFT matrix element can be given with the option **STATE**, then indics of the two orbitals are read from the next line. **ROUT** controls printing of involved orbitals.
MATRIX instructs CPMD to read a transformation matrix from the file LOWDIN_A to transform the KS-Hamiltonian to the non-orthogonal orbital basis
.

KPERT [MONKHORSTPACK,SCALE]
Section: &RESP

Calculation of total energy and electronic density of states with an arbitrary number of k-points (at almost no additional computational effort). The method is based on a $\mathbf{k}\cdot\mathbf{p}-$like approximation developed in the framework of the density functional perturbation theory [25]. For a sampling of the BZ determined by the Monkhorst-Pack algorithm, the option **MONKHORSTPACK** has to be specified, followed by the dimension of the mesh along the 3 reciprocal space axis (NK_1, NK_2, NK_3). If omitted, the individual absolute coordinates of the k-points have to be given one by one in the following lines. The **SCALE** option allows to specify them in units of the reciprocal cell vectors.
The line after **KPERT** has to contain the the total number of k-points ($NKPTS$), which have then to be given by their coordinates and the associated weights (RK, WK) in the format:
$NKPTS$
RK_{x1} RK_{y1} RK_{z1} WK_1
...
RK_{xNKPTS} RK_{yNKPTS} RK_{zNKPTS} WK_{NKPTS}.
Three response wavefunctions are calculated, corresponding to the three independent orientations of the k basis vectors in reciprocal space. Therefore, 3 independent optimization loops are started (x, y and z), and the 3 sets of wfns are stored (you need 4 times the memory required for a standard wavefunction optimization). The second order correction to the Γ-point total energy is calculated for the requested k-point mesh.
Further options are (each in a new line of the input file):

WRITE_C1 the 3 sets of response wfns are stored in three separate restart files.

HAMILTONIAN the k-dependent Hamiltonian is constructed via the second order perturbation theory approximation, and the corresponding KS energies are calculated. Due to technical reasons, for each k-point $2 * NSTATE$ KS energies are calculated, however only those corresponding to occupied orbitals are reliable.

READ_C1 the response wfns are read from RESTART.P_{xyz}.

BUILD_C00 the set of k-dependent wfns (first order correction) is calculated from the unperturbed Γ-point wfns together with the response orbitals. They are then written in a standard **RESTART** file. From this restart file one can perform a calculation of the Hamiltonian matrix for each kpoint and calculate the KS energies (use **LANCZOS DIAGO** in &CPMD and the **KPOINT** option **ONLYDIAG** in &SYSTEM. The k-point mesh must be the same used in the linear response calculation. set also **NOSPHERICAL CUTOFF** in &SYSTEM).

NORESTART no RESTART file is written.

KPOINTS *options*
Section: &SYSTEM

With no option, read in the next line with the number of k-points and for each k-point, read the components in the Cartesian coordinates (units $2\pi/a$) 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. Useful for molecular dynamics simulation The keywords **SYMMETRIZED FULL** preserves all symmetry of Bravais lattice so there is no need to symmetrize density and forces.

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

BANDS This option is to calculate the band structure.
For each line you have to specify the number of k-points for the band, the initial and the final k-point. To finish the input, put:
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.

LANCZOS DIAGONALIZATION {ALL}

Section: &CPMD

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

LANCZOS DIAGONALIZATION {OPT,RESET=n}

Section: &CPMD

Use **Lanczos diagonalization** scheme after (OPT) or periodically during (RESET=n) direct wavefunction optimization using **ODIIS**. The number n specifies the number of DIIS resets (ODIIS NO_RESET=nreset) due to poor progress until the wavefunction is diagonalized. This can be helpful if the wavefunction is converging very slowly.

LANCZOS PARAMETER [N=n] [ALL]

Section: &CPMD

Give four parameters for Lanczos diagonalization in 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 put a negative or zero number, 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 the hints section 11.12.1 for more information.

LANCZOS [CONTINUE,DETAILS]

Section: &RESP

lanczos_dim iterations conv_threshold lanczos_dim= dimension of the vibrational d.o.f. iterations = no. of iterations desired for this run conv_threshold = threshold for convergence on eigenvectors CONTINUE = argument for continuing Lanczos diagonalization from a previous run (reads file LANCZOS_CONTINUE) DETAILS = argument for verbosity. prints a lot of stuff

LANGEVIN {WHITE, CPMD, OPTIMAL, SMART, CUSTOM, CENTROID-OFF} [MOVECM] *[W0, NS]*

Section: &CPMD

Use a (generalized) Langevin equation to thermostat the simulation[43]. By default, the component of the noise parallel to the center of mass velocity is removed at each step of the thermostat. Removal can be disabled by the option *MOVECM*.

CUSTOM: The **number of additional momenta** of the generalized Langevin equation *NS* is read from the next line. The drift matrix (dimension $(NS+1) \times (NS+1)$) is read from the file `GLE-A`, which must be in the same directory in which the program is run. Optionally, the static covariance for the GLE dynamics can be provided in the file `GLE-C`, so as to generate **non-canonical sampling**. A library of GLE parameters can be downloaded from http://gle4md.berlios.de/

A few **presets** are provided, and are activated by the keywords:

WHITE: A simple **white-noise** Langevin dynamics is used. The optimally-sampled frequency *W0* (in cm^{-1}) is read from the next line. Note that use of *LANGEVIN WHITE* in conjunction with *MOLECULAR DYNAMICS CPMD* will most likely cause a large drift of the electronic temperature.

OPTIMAL: An **optimal-sampling** generalized Langevin dynamics is used. The frequencies in the range from 10^{-4}*W0* up to *W0* will be sampled efficiently. Note that use of *LANGEVIN OPTIMAL* in conjunction with *MOLECULAR DYNAMICS CPMD* will cause a large drift of the electronic temperature. This option is suggested for use in Born-Oppenheimer MD.

CPMD: A generalized Langevin dynamics is used which is designed to work in conjunction with Car-Parrinello MD. The highest ionic frequency *W0* (in cm^{-1}) is read from the next line. Ionic frequencies down to 10^{-4}*W0* will be sampled efficiently, but not as much as for the *OPTIMAL* keyword.

SMART: A generalized Langevin dynamics that aims to be as efficient as possible on the slowest time scale accessible to a typical ab initio simulation. In practice, vibrations with a time scale which is about 10000 time steps will be sampled optimally, and faster modes will be sampled as efficiently as possible without disturbing slower modes. The highest ionic frequency *W0* (in cm^{-1}) is read from the next line. Will be about 50% more efficient than *OPTIMAL* for slow modes, but less efficient for fast vibrations. Use only with Born-Oppenheimer dynamics.

CENTROIDOFF: For centroid and ring-polymer dynamics the generalized Langevin thermostat for the centroids can be switched off by using the optional keyword CENTROIDOFF otherwise the thermostat with the frequency read from the next line is attached to the centroids. For each non-centroid mode the frequency of the thermostat is automatically determined depending on the frequency of its associated harmonic potential representing the quantum kinetic energy term.

LBFGS [NREM, NTRUST, NRESTT, TRUSTR]

Section: &CPMD

Use the limited-memory BFGS method (L-BFGS) for linear scaling **optimization** of the **ionic positions**. For more informations, see [26]. The information about the Hessian for the quasi-Newton method employed is derived from the history of the optimization [26, 105].
Only one sub-option is allowed per line and the respective parameter is read from the next line. The parameters mean:

NREM:	**Number** of **ionic gradients** and **displacements remembered** to approximate the Hessian. The default is either 40 or the number of ionic degrees of freedom, whichever is smaller. Values greater the number of degrees of freedom are not advisable.
NTRUST:	*NTRUST=1* switches from a trust radius algorithm to a **line search** algorithm. The default value of 0 (**trust radius**) is recommended.
NRESTT:	*NRESTT>0* demands a **periodic reset** of the optimizer every *NRESTT* steps. Default is 0 (no periodic reset). This option makes only sense if the ionic gradient is not accurate.
TRUSTR:	Maximum and initial **trust radius**. Default is 0.5 atomic units.

It can be useful to combine these keywords with the keywords **PRFO**, **CONVERGENCE** ADAPT, **RESTART** LSSTAT, **PRINT** LSCAL ON and others.

LDA CORRELATION *[functional]*

Section: &DFT

The LDA correlation functional is specified. Only available for **OLDCODE** and **NEWCODE**.
Possible functionals are **NO** (no correlation functional), **PZ** [106], **VWN** [107], **LYP** [98] and **PW** [108].
Default is the **PZ**, the Perdew and Zunger fit to the data of Ceperley and Alder [109].

LDOS

Section: &PROP

Calculate the layer projected density of states. The number of layers is read from the next line.

To use the LDOS keyword, the user must first have performed a wavefunction optimization and then restart with with the **PROPERTIES** and **LANCZOS DIAGONALIZATION** keywords in the &CPMD section (and LDOS in the &PROP section).

Warning: If you use special k-points for a special structure you need to symmetrize charge density for which you must specify the **POINT GROUP**.

LIBRARY *library_for_functional_1, library_for_functional_2, ...*

Section: &DFT

In combination with **XC_DRIVER**, both functionals from a new, internal CP library and from libxc are available. On the same line, for every functional specified in **FUNCTIONAL**, the library has to be indicated by **INTERNAL** or **CP** for the internal CP library, or by **LIBXC** for libxc. If the keyword is omitted, the functionals will default to the internal CP library.
The use of LIBXC is usually associated to a considerable overhead with respect to the internal implementation. It is therefore discouraged for functionals that are also available in the internal library, *cf.* **FUNCTIONAL**.

KERNEL_LIBRARY *library_for_kernel_1, library_for_kernel_2, ...*

Section: &DFT

Same functionalty as **LIBRARY**, but applies to the **FUNCTIONAL** rather than the **LR KERNEL**.

LIBXC_ONLY

Section: &DFT

Use the new xc driver, but use only functionals from libxc (this excludes all internal functionals). *Cf.* **XC_DRIVER**.

LINEAR RESPONSE

Section: &CPMD

A perturbation theory calculation is done, according to the (required) further input in the &RESP section. In the latter, one of the possible perturbation types (PHONONS, LANCZOS, RAMAN, FUKUI, KPERT, NMR, EPR, see section 11.9.2) can be chosen, accompanied by further options.

LOCAL DIPOLE

Section: &PROP

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

LOCALIZATION

Section: &TDDFT

Use localized orbitals in the TDDFT calculation. Default is to use canonical orbitals.

LOCALIZE
Section: &HARDNESS

Use localized orbitals in an orbital hardness calculation

LOCALIZE
Section: &PROP

Localize the molecular orbitals [110] 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 section.

LR KERNEL *functional*
Section: &DFT

Use another functional for the linear response kernel. To be used like **FUNCTIONAL**.

XC_KERNEL *functional*
Section: &DFT

Alias for **LR KERNEL**.

LR-TDDFT
Section: &TDDFT

Use full linear response version of TDDFT. Default is to use **TAMM-DANCOFF** approximation.

LSD
Section: &CPMD

Use the local spin density approximation.
Warning: Not all functionals in **OLDCODE** or **NEWCODE** are implemented for this option. Only in the **XC_DRIVER**, all functionals are available both for spin-restricted and -unrestricted calculations.

LOCAL SPIN DENSITY
Section: &CPMD

Alias for **LSD**.

LOW_LEVEL_FORCES {DFT, EXTERNAL}
Section: &MTS

Analogue to **HIGH_LEVEL_FORCES**.

LOW SPIN EXCITATION [ROKS,ROSS,ROOTHAAN,CAS22]
Section: &SYSTEM

Use the low spin excited state functional [31]. For ROKS calculations, see also the **ROKS** keyword in the &CPMD-section.

LOW SPIN EXCITATION LSETS
Section: &SYSTEM

Slater transition-state density with restricted open-shell Kohn-Sham (low spin excited state). Currently works only with ROKS but not with ROSS, ROOTHAAN, or CAS22. See Ref. [88].

LSE PARAMETERS
Section: &SYSTEM

Determines the energy expression used in LSE calculations. The two parameters LSEA and LSEB are read from the next line.

$$E = \text{LSEA} \cdot E(Mixed) + \text{LSEB} \cdot E(Triplet)$$

The default (LSEA = 2 and LSEB = 1) corresponds to singlet symmetry. For the lowest triplet state, the **LSE PARAMETERS** must be set to 0 and 1 (zero times mixed state plus triplet). See ref [31] for a description of the method.

LZ-SHTDDFT
Section: &TDDFT

Non adiabatic (nonadiabatic, non-adiabatic) trajectory surface hopping scheme based on Landau-Zener transition probabilities. Excited state dynamics is performed on TDDFT potential energy surfaces. To be used together with the keywords **MOLECULAR DYNAMICS** BO and **TDDFT** in the &CPMD section (see section 11.8.2). Do NOT use this keyword together with **SURFACE HOPPING** in &CPMD, which invokes the surface hopping scheme based on **ROKS** (see **SURFACE HOPPING**). See also the related approach based on Tully's hopping probabilities **T-SHTDDFT**.

MAXRUNTIME
Section: &CPMD

The maximum RUN TIME (ELAPSED TIME) in seconds to be used is read from the next line. The calculation will stop after the given amount of time.
Default is no limit.

MAXITER
Section: &CPMD

The maximum number of iteration steps for the self-consistency of wavefunctions. Recommended use instead of **MAXSTEP** for pure wavefunction optimisation. The value is read from the next line.
Default is **10000** steps.

MAXSTEP
Section: &CPMD

The maximum number of steps for geometry optimization or molecular dynamics to be performed. In the case of pure wavefunction optimisation, this keyword may be used instead of **MAXITER**. The value is read from the next line.
Default is **10000** steps.

MAXSTEP
Section: &LINRES

Maximum number of optimization steps for linear response optimizations. Default is 1000.

MEMORY {SMALL, BIG}
Section: &CPMD

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.

MESH
Section: &SYSTEM

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.

METADYNAMICS ... END METADYNAMICS

Section: &ATOMS

Initiate Metadynamics (see section 11.10 for more information on the available options and the input format).

MIRROR

Section: &CPMD

Write the input file to the output.

MIXDIIS

Section: &CPMD

Not documented

MIXSD

Section: &CPMD

Not documented

MODIFIED GOEDECKER [PARAMETERS]

Section: &CPMD

To be used in combination with **LOW SPIN EXCITATION ROKS**.
Calculation of the off-diagonal Kohn-Sham matrix elements F_{AB} and F_{BA} (with A, B: ROKS-SOMOs) is performed according to a modified Goedecker-Umrigar scheme ($F_{AB} := (1-\lambda_{AB})F_{AB} + \lambda_{AB}F_{BA}$ and $F_{BA} := (1-\lambda_{BA})F_{BA} + \lambda_{BA}F_{AB}$). Default values are $\lambda_{AB} = -0.5$ and $\lambda_{BA} = 0.5$. see Ref. [37].

With the optional keyword **PARAMETERS**: λ_{AB} and λ_{BA} are read from the next line. Can be used to avoid unphysical rotation of the SOMOs. Always check the orbitals!

See also 11.11.

MOLECULAR DYNAMICS [CP, BO, ET, PT, CLASSICAL, FILE [XYZ, NSKIP=N, NSAMPLE=M]]

Section: &CPMD

Perform a molecular dynamics (MD) run. **CP** stands for a Car-Parrinello type MD. With the option **BO** a Born-Oppenheimer MD is performed where the wavefunction is re-converged after each MD-step. **EH** specifies Ehrenfest type dynamics according to which the Kohn-Sham orbitals are propagated in time (real electronic dynamics coupled to the nuclear dynamics). In this case the time step has to be decreased accordingly due to the small mass of the electrons (typical values between 0.01 and 0.1 au). If you use EH dynamics an additional input section &PTDDFT has to be specified. You need to start the dynamics with well converged KS orbitals from the RESTART file (before starting the EH dynamics do an optimization of the wavefunction with a convergence of 1.D-8 or 1.D-9, if possible. An additional file called "wavefunctions" is produced, which contains the complex KS orbitals needed for the restart of the EH dynamics (see restart options in &PTDDFT). Typical (minimal) input &CPMD and &PTDDFT sections to be used with EH dynmiacs
&CPMD
MOLECULAR DYNAMICS EH
RESTART WAVEFUNCTION COORDINATES LATEST
CAYLEY
RUNGE-KUTTA
TIMESTEP
0.01
MAXSTEP
10000
&END
&PTDDFT
ACCURACY
1.0D-8
RESTART
2
&END
The keywords CAYLEY and RUNGE-KUTTA specifies the algorithms used for the propagation of the KS orbitals (are the default and recommended options).
CLASSICAL means that a MD that includes classical atoms is performed.
BD stands for Bohminan Dynamics. In the next line the number of trajectories (N_{traj}) used to describe the nuclear wavepacket are specified. The initial spread of the Gaussians associated to each element is guessed from the De Broglie wavelength. A special tuning of this value for the different elements may be necessary (not yet automatized). During the dynamics the N_{traj} trajectories (each corresponding to a configuration space point or fluid element) are interacting within each other through the action of the quantum potential. For the more details about the implementation see Ref. [111].
If **FILE** is set, then the trajectory is reread from a file instead of being calculated. This is useful for performing analysis on a previous trajectory. Can be used in conjunction with the standard MD options like DIPOLE DYNAMICS and WANNIER; some other features like LINEAR RESPONSE are also enabled. The trajectory is read from a file named TRAJSAVED (usually a copy of a previous TRAJECTORY file), or TRAJSAVED.xyz if **XYZ** is set. **NSKIP** and **NSAMPLE** control the selection of frames read: the frame read at step ISTEP is NSKIP+ISTEP*NSAMPLE.
Default is **CP**.

MOLECULAR STATES
Section: &TDDFT

Calculate and group Kohn–Sham orbitals into molecular states for a TDDFT calculation.

MOVERHO
Section: &CPMD

Mixing used during optimization of geometry or molecular dynamics. 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).

MOVIE TYPE
Section: &ATOMS

Assign special movie atom types to the species.
The types are read from the next line. Values from 0 to 5 were allowed in the original MOVIE format.

MOVIE [OFF, SAMPLE]
Section: &CPMD

Write the atomic coordinates without applying periodic boundary conditions in MOVIE format every *IMOVIE* time steps on file *MOVIE*. *IMOVIE* is read from the next line.
Default is **not** to write a movie file.

MULTIPLICITY
Section: &SYSTEM

This keyword only applies to LSD calculations.
The multiplicity ($2S$+1) is read from the next line.
Default is the **smallest possible** multiplicity.

MTS_HIGH_FUNC *functionals*
Section: &DFT

Alias for **FUNCTIONAL** that makes sense when used with the MTS scheme since it corresponds to the high level functional in that context (see &MTS section). **Note:** The functionals in combination with the MTS scheme have to be set with **XC_DRIVER**.

MTS_LOW_FUNC *functionals*
Section: &DFT

Select the low level functional in the MTS scheme. To be used like **FUNCTIONAL**. **Note:** The functionals in combination with the MTS scheme have to be set with **XC_DRIVER**.

N_CYCLES
Section: &PTDDFT

Defines the number of cycles (time steps) used in the propagation of the perturbed KS orbitals in a **PROPAGATION SPECTRA** calculation. The number of cycles is read from the next line.

NEQUI
Section: &PATH

Number of equilibration steps discarded to calculate the mean force.

NEWCODE
Section: &DFT

This keyword will be deprecated in a future release. Both **NEWCODE** and **OLDCODE** have been replaced by the new **XC_DRIVER**.

The following description applies up to CPMD version prior to 4.1:
NEWCODE is a switch to select one out of two versions of code to calculate exchange-correlation functionals.
NEWCODE is the default, but not all functionals are available with NEWCODE, if you select one of these, **OLDCODE** is used automatically. NEWCODE is highly recommended for all new projects and especially for vector computers, also some of the newer functionality is untested or non-functional with OLDCODE.

NLOOP
Section: &PATH

Maximum number of string searches for Mean Free Energy Path searches.

NMR *options*
Section: &RESP

Calculate the NMR chemical shielding tensors for the system. Most important option: FULL, does a calculation with improved accuracy for periodic systems but takes a lot of time. Isolated systems: Use OVERLAP and 0.1 (on next line) for the same effect. *Be careful for non-hydrogen nuclei.* The shielding is calculated without contribution from the core electrons. Contact sebastia@mpip-mainz.mpg.de for further details.

NO_CONTRIBUTION
Section: &VDW

If the **DCACP** van der Waals-correction scheme is adpoted, the indeces of atomic species (in the same order as specified in the &ATOMS section) that shall not contribute to the dispersion forces (capping atoms, metals...) are read from the next line.

NOGEOCHECK
Section: &CPMD

Default is to check all atomic distances and stop the program if the smallest disctance is below 0.5 Bohr. This keyword requests not to perform the check.

NONORTHOGONAL ORBITALS [OFF]
Section: &CPMD

Use the norm constraint method [112] for molecular dynamics or nonorthogonal 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.

NOOPT
Section: &RESP

Do not perform a ground state wfn optimization. Be sure the restarted wfn is at the BO-surface.

NOPRINT ORBITALS
Section: &PROP

Do not print the wavefunctions in the atomic basis set.

NORMAL MODES
Section: &PIMD

Use the normal mode representation [15] of the path integral propagator. It is possible to impose a mass disparity between centroid and non–centroid coordinates by dividing the fictitious masses of only the *non*–centroid $s = 2, \ldots, P$ replicas by the adiabaticity control factor FACSTAGE. This dimensionless factor *must always* be specified in the following line. Note: the eigen–*frequencies* of the $s > 1$ replicas are changed by only $\sqrt{\text{FACSTAGE}}$, see Ref. [61](b). Using FACSTAGE $\neq 1.0$ makes only sense in conjunction with CENTROID DYNAMICS where WMASS=1.0 has to be used as well.

NOSE PARAMETERS

Section: &CPMD

The **parameters** controlling the **Nosé thermostats** [6, 7] 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 <u>all</u> parameters have to be specified.

NOSE {IONS, ELECTRONS, CELL} [ULTRA,MASSIVE,CAFES,LOCAL,CENTROIDOFF]

[T0]
Section: &CPMD

Nosé-Hoover chains [6, 7] 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. Two files NOSE_ENERGY and NOSE_TRAJEC are written at each step containing the Nosé-Hoover kinetic, potential and total energies along the dynamics (NOSE_ENERGY) and the Nosé-Hoover variables and their velocities (NOSE_TRAJEC); these are useful in a wealth of post-processing calculations such as, e. g. heat transfer problems[183, 184]. For the ionic case the additional keyword **ULTRA** selects a thermostat for each species, the keyword **MASSIVE** selects a thermostat for each degree of freedom, and the keyword **CAFES** can be used to give different temperatures to different groups of atoms[113]. The syntax in the **CAFES** case is:

`NOSE IONS CAFES`
ncafesgrp
cpnumber_a_1 *cpnumber_a_2* Temperature Frequency
. . .
cpnumber_n_1 *cpnumber_n_2* Temperature Frequency

There are *ncafesgrp* groups, specified by giving their first CPMD atom number (*cpnumber_X_1*) and last CPMD atom number (*cpnumber_X_2*). In the case of hybrid QM/MM simulations, you have to consult the QMMM_ORDER file to find those numbers. The temperature and frequency can be different for each group. All atoms of the system have to be in a CAFES group. A new file, `CAFES` is created containing the temperature of each group (cols. 2 . . . *ncafesgrp+1*) and the energy of the Nose-Hoover chains of that group (last columns).
Using CAFES with different temperatures only makes sense if the different groups are decoupled from each other by increasing the masses of the involved atoms. The mass can be specified in the topology / or with the **ISOTOPE** keyword. However, you can only change the mass of a complete CPMD species at a time. Hence, the topology and/or the input should be such that atoms of different CAFES group are in different species.
NOTE: CAFES is currently not restartable.

The keyword **LOCAL** collects groups of atoms to separate thermostats, each having its own Nosé-Hoover chain. Specify the local thermostats as follows:

`NOSE IONS LOCAL`
n_l *(number of local thermostats)*
temperature 1 *frequency 1*
⋮
temperature n_l *frequency* n_l

n_r *(number of atom ranges)*
thermostat number *start atom* *end atom*
⋮ (n_r *entries)*

The parser for the atom ranges uses either the CPMD ordering or the GROMOS ordering in case of classical or QM/MM runs. Multiple ranges may be specified for the same thermostat. Atoms belonging to the same CPMD constraint or the same solvent molecule in QM/MM runs must belong to the same local thermostat.
If **T0** option is present, the initial temperature for the Nosé-Hoover chains are read soon after the thermostat frequencies in the same line (also for the LOCAL thermostat). By default it is same as the target temperature of the thermostat. Note: This is not implemented for the CAFES thermostat.

The keyword **CENTROIDOFF** is intended to be used in connection with the centroid and ring-polymer dynamics. In this type of path integral MD, you can request not to attach the Nosé-Hoover chains for ions to the centroids by explicit use of the keyword CENTROIDOFF, otherwise the thermostat with the frequency read from the next line is attached to the centroids. For each non-centroid mode the frequency with the Nosé-Hoover chains is automatically determined depending on the frequency of its associated harmonic potential representing the *quantum* kinetic energy term.

NPREVIOUS

Section: &PATH

String index to restart from. Note that this is just for numbering files, the initial path in collective variables for the search is always *string.inp*.

NUMERICAL_DIV

Section: &DFT

Calculate the $\mathbf{G} = 0$ term for the Coulomb-attenuated exact exchange using a numerical description of the integral term. This is the default only when CAM-B3LYP is used with LIBXC. In all other situations, **ANALYTICAL_DIV** is used.

OACP [DENSITY, REF_DENSITY, FORCE]

Section: &RESP

Not documented.

OCCUPATION [FIXED]

Section: &SYSTEM

The occupation numbers are read from the next line.
This keyword must be preceded by **STATES**. The FIXED option fixes the occupation numbers for the diagonalization scheme, otherwise this option is meaningless.

ODIIS [NOPRECONDITIONING,NO_RESET=nreset]

Section: &CPMD

Use the method of **direct inversion** in the iterative subspace for **optimization** of the **wavefunction** [114].
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.
By default, the number of wavefunction optimization cycles until DIIS is **reset** on poor progress, is the number of DIIS vectors. With **ODIIS NO_RESET**, this number can be changed, or DIIS resets can be **disabled** altogether with a value of -1.

OLD_DEFINITIONS
Section: &DFT

For certain functionals, older versions of CPMD did not use standard definitions. When using the new **XC_DRIVER**, those definitions can be invoked by using **OLD_DEFINITIONS**. The option is available in combination with any derivative of PBE and TPSS correlation.

OLDCODE
Section: &DFT

This keyword will be deprecated in a future release. Please use the new **XC_DRIVER**. *Cf.* **NEWCODE**.

OPTIMIZE GEOMETRY [XYZ, SAMPLE]
Section: &CPMD

This option causes the program to optimize the geometry of the system through a sequence of wavefunction optimizations and position updates. The additional keyword XYZ requests writing the "trajectory" of the geometry additionally in xmol/xyz-format in a file *GEO_OPT.xyz*. If the keyword SAMPLE is given, *NGXYZ* is read from the next line, and then only every *NGXTZ* step is written to the xmol/xyz file. The **default** is to write every step ($NGXYZ = 1$).
By default the a BFGS/DIIS algorithm is used (see **GDIIS**) to updated the ionic positions. Other options are: **LBFGS**, **PRFO**, and **STEEPEST DESCENT** IONS. See **OPTIMIZE WAVEFUNCTION** for details on the corresponding options for wavefunction optimizations.

OPTIMIZE SLATER EXPONENTS
Section: &PROP

Not documented

OPTIMIZE WAVEFUNCTION
Section: &CPMD

Request a single point energy calculation through a wavefunction optimization. The resulting total energy is printed (for more output options see, e.g.,: **PRINT**, **RHOOUT**, **ELF**) and a **RESTART** file is written. This restart file is a prerequisite for many other subsequent calculation types in CPMD, e.g. **MOLECULAR DYNAMICS** CP or **PROPERTIES**. By default a DIIS optimizer is used (see **ODIIS**), but other options are: **PCG** (optionally with MINIMIZE), **LANCZOS DIAGONALIZATION**, **DAVIDSON DIAGONALIZATION**, and **STEEPEST DESCENT** ELECTRONS.

OPTIMIZER [SD,DIIS,PCG,AUTO]
Section: &LINRES

Optimizer to be used for linear response equations. Default is “AUTO” which will first use PCG, then switch to DIIS and finally switch to DIIS with full storage and state dependent preconditioner. **THAUTO** sets the two tolerances for when to do the switch.

ORBITAL HARDNESS [LR,FD]
Section: &CPMD

Perform an orbital hardness calculation. See section &Hardness for further input options.

ORBITALS
Section: &HARDNESS

Specify the number of orbitals to be used in a hardness calculation on the next line.

ORTHOGONALIZATION {LOWDIN, GRAM-SCHMIDT} [MATRIX]
Section: &CPMD

Orthogonalization in optimization runs is done either by a Löwdin (symmetric) or Gram-Schmidt procedure.
Default is Gram-Schmidt except for parallel runs where Löwdin orthogonalization is used with the conjugate-gradient scheme.
With the additional keyword **MATRIX** the Löwdin transformation matrix is written to a file named LOWDIN_A.

OUTPUT [ALL, GROUPS, PARENT]
Section: &PIMD

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.

OUTPUT [ALL, GROUPS, PARENT]
Section: &PATH

Idem as above, here for Mean Free Energy Path runs.

PARA_BUFF_SIZE
Section: &CPMD

Set the buffer size for parallel operation (sum, ...). Default is 2**16. The size is read from the next line.

PARA_STACK_BUFF_SIZE
Section: &CPMD

Set the stack buffer size for parallel operation (sum, ...). Default is 2**8. The size is read from the next line.

PARA_USE_MPI_IN_PLACE
Section: &CPMD

Use MPI_IN_PLACE for parallel operation (sum, ...). Default is FALSE.

PARRINELLO-RAHMAN {NPT,SHOCK}
Section: &CPMD

To be used together with **MOLECULAR DYNAMICS**.
A **variable cell MD** with the **Parrinello-Rahman Lagrangian** is performed [11, 12]. With the additional keyword a **constant NPT MD** using the method of Martyna, Tobias, and Klein [115].
If this keyword is used together with other run options like OPTIMIZE WAVEFUNCTIONS, calculations with different reference cells can be performed.
With the additional keywork **SHOCK**
a MD simulation using the multiscale shock method [36] is performed.

PATH INTEGRAL
Section: &CPMD

Perform a **path integral molecular dynamics** calculation [9, 10].
This keyword requires further input in the section &PIMD ... &END.

PATH MINIMIZATION
Section: &CPMD

Perform a **mean free energy path** search [49].
This keyword requires further input in the section &PATH ... &END.

PATH SAMPLING
Section: &CPMD

Use CPMD together with a reaction path sampling [116] program. This needs special software. Note: this keyword has *nothing* to do with path integral MD as activated by the keyword PATH INTEGRAL and as specified in the section &PIMD ... &END.

PBE_FLEX_BETA
Section: &DFT

In combination with the PBE_FLEX correlation functional, the parameter β can be specified on the next line.

PBE_FLEX_GAMMA
Section: &DFT

In combination with the PBE_FLEX correlation functional, the parameter γ can be specified on the next line.

PBE_FLEX_KAPPA
Section: &DFT

In combination with the PBE_FLEX exchange or correlation functionals, the parameter κ can be specified on the next line.

PBE_FLEX_MU
Section: &DFT

In combination with the PBE_FLEX exchange or correlation functionals, the parameter μ can be specified on the next line.

PBE_FLEX_UEG_CORRELATION *functional*
Section: &DFT

In combination with the PBE_FLEX correlation functional, the LDA correlation functional can be specified on the same line. Available options are LDA_C_PZ, LDA_C_PW (the default), LDA_C_OBPW and LDA_C_PZ.

PCG PARAMETER
Section: &TDDFT

The parameters for the PCG diagonalization are read from the next line. If *MINIMIZE* was used in the **DIAGONALIZER** then the total number of steps (default 100) and the convergence criteria (default 10^{-8}) are read from the next line. Without minimization in addition the step length (default 0.5) has also to be given.

PCG [MINIMIZE,NOPRECONDITIONING]

Section: &CPMD

Use the method of **preconditioned conjugate gradients** for **optimization** of the **wavefunction**.
The fixed step length is controlled by the keywords **TIMESTEP ELECTRONS** and **EMASS**.
If the additional option **MINIMIZE** is chosen, then additionally line searches are performed to improve the preconditioning.
The preconditioning is controlled by the keyword **HAMILTONIAN CUTOFF**. Optionally preconditioning can be disabled.

PERT_TYPE

Section: &PTDDFT

The type of the static perturbation used with **MOLECULAR DYNAMICS** EH or **PROPAGATION SPECTRA** is read from the next line. 1: Dirac pulse, 2: Heaviside step function (constant value after $t = 0$).

PERT_AMPLI

Section: &PTDDFT

The amplitude of the perturbation used with **MOLECULAR DYNAMICS** EH or **PROPAGATION SPECTRA** is read from the next line.

PERT_DIRECTION

Section: &PTDDFT

Use with MOLECULAR DYNAMICS EH or PROPAGATION SPECTRA to specify the direction (x=1,y=2,z=3) of the perturbing field. The direction code (1,2,3) is read from the next line.

PHONON

Section: &RESP

Calculate the harmonic frequencies from perturbation theory.

PIPULSE

Section: &PTDDFT

Specifies a time dependent pi-pulse to be used with MOLECULAR DYNAMICS EH. Use PIPULSE together with TD_POTENTIAL. The pulse strength is read from the next line (see subroutine gaugepot_laser in td_utils.mod.F90 for further details).

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

Section: &SYSTEM

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 by its name using the keyword *NAME=* followed 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. DNH). 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.

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

Section: &SYSTEM

This keyword determines the method for the solution of the Poisson equation for isolated systems. Either Hockney's method [29] or Martyna and Tuckerman's method [117] is used. The smoothing parameter (for Hockney's method) or $L \times \alpha$ for Tuckerman's method can be read from the next line using the **PARAMETER** keyword.
For more information about the usage of this parameter see also section 11.4.

POLAK
Section: &RESP

Uses the Polak-Ribiere formula for the conjugate gradient algorithm. Can be safer in the convergence.

POLARIZABILITY
Section: &PROP

Computes the polarisability of a system, intended as dipole moment per unit volume.

POLYMER
Section: &SYSTEM

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

POPULATION ANALYSIS [MULLIKEN, DAVIDSON],[n-CENTER]
Section: &PROP

The type of population analysis that is performed with the projected wavefunctions. Löwdin charges are given with both options. For the Davidson analysis [118] 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 in a separate line. An input example for a water molecule is given in the hints section 11.13.

PRESSURE
Section: &SYSTEM

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

PRFO [MODE, MDLOCK, TRUSTP, OMIN, PRJHES, DISPLACEMENT, HESSTYPE]
Section: &CPMD

Use the partitioned rational function optimizer (P-RFO) with a quasi-Newton method for **optimization** of the **ionic positions**. For more informations, see [26]. The approximated Hessian is updated using the Powell method [119]. This method is used to find **transition states** by **following eigenmodes** of the approximated Hessian [120, 26].

Only one suboption is allowed per line and the respective parameter is read from the next line. The suboption **PRJHES** does not take any parameter. If it is present, the translational and rotational modes are removed from the Hessian. This is only meaningful for conventional (not microiterative) transition state search. The parameters mean:

MODE: Number of the initial Hessian **eigenmode** to be followed. Default is 1 (lowest eigenvalue).

MDLOCK: *MDLOCK=1* switches from a mode following algorithm to a **fixed eigenvector** to be maximized. The default value of 0 (**mode following**) is recommended.

TRUSTP: Maximum and initial **trust radius**. Default is 0.2 atomic units.

OMIN: This parameter is the minimum **overlap** between the maximized mode of the previous step and the most overlapping eigenvector of the current Hessian. The trust radius is reduced until this requirement is fulfilled. The default is 0.5.

DISPLACEMENT: Finite-difference **displacement** for initial partial Hessian. The default is 0.02.

HESSTYPE: **Type** of initial partial Hessian. 0: Finite-difference. 1: Taken from the full Hessian assuming a block-diagonal form. See keyword **HESSIAN**. The default is 0.

It can be useful to combine these keywords with the keywords **CONVERGENCE** ENERGY, **RESTART** LSSTAT, **RESTART** PHESS, **PRFO** NSVIB, **PRINT** LSCAL ON and others.

PRFO [NVAR, CORE, TOLENV, NSMAXP]

Section: &CPMD

If any of these suboptions is present, the **microiterative transition state search** scheme for **optimization** of the **ionic positions** is used. For more informations, see [26]. A combination of the **L-BFGS** and **P-RFO** methods is employed for linear scaling search for transition states [26, 121]. Before each P-RFO step in the reaction core towards the transition state, the **environment** is fully **relaxed** using L-BFGS. Only one suboption is allowed per line. The **reaction core** can be selected using the **NVAR** or **CORE=ncore** suboptions. The value in the line after **PRFO NVAR** sets the number of ionic **degrees of freedom** in the reaction core. The *ncore* values following the line **PRFO CORE=ncore** select the **member atoms** of the reaction core. If unspecified, the *NVAR/3* first atoms form the reaction core. The parameters read with the two remaining suboptions are:

TOLENV: **Convergence criterion** for the maximum component of the gradient acting on the ions of the **environment** until a P-RFO step within the reaction core is performed. Default is one third of the convergence criterion for the gradient of the ions (**CONVERGENCE GEOMETRY**).

NSMAXP: Maximum number of P-RFO **steps** to be performed in the reaction core. The keyword **HESSCORE** corresponds to **PRFO NSMAXP** with *NSMAXP=0*.

It can be useful to combine these keywords with the keywords **LBFGS**, **CONVERGENCE** ADAPT, **CONVERGENCE** ENERGY, **RESTART** LSSTAT, **RESTART** PHESS, **PRFO NSVIB**, **PRINT** LSCAL ON, the other suboptions of PRFO, and others.

PRFO NSVIB
Section: &CPMD

Perform a **vibrational analysis** every NSVIB P-RFO steps **on the fly**. This option only works with the P-RFO and microiterative transition state search algorithms. In case of microiterative TS search, only the reaction core is analyzed.

PRINT COORDINATES
Section: &CLASSIC

Not documented

PRINT ENERGY {ON, OFF} [EKIN, ELECTROSTATIC, ESR, ESELF, EFREE, EBAND, ENTROPY, EPSEU, EHEP, EHEE, EHII, ENL, EXC, VXC, EGC, EBOGO]
Section: &CPMD

Display or not information about energies.

PRINT FF
Section: &CLASSIC

Not documented

PRINT_FORCES [OFF]

Section: &MTS

Turn on/off the printing of the high and low level forces along the MTS-MD trajectory. The forces are printed to two distinct trajectory files in the `.xyz` format, `MTS_LOW_FORCES.xyz` and `MTS_HIGH_FORCES.xyz`.

Default: OFF.

PRINT LEVEL

Section: &PIMD

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.

PRINT LEVEL

Section: &PATH

Idem as above, here for Mean Free Energy Path searches.

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

Section: &CPMD

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.

PRNGSEED

Section: &CPMD

The seed for the random number generator is read as an integer number from the next line.

PROCESSOR GROUPS

Section: &PIMD

This is only needed for *fine*–tuning load balancing in case of path integral runs *iff* two level parallelization is used. The default optimizes the combined load balancing of the parallelization over replicas and g–vectors. The default load distribution is usually optimal. Separate the total number of processors into a certain number of processor groups that is read from the following line; only $2^N = 2, 4, 8, 16, \ldots$ 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. 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.

PROCESSOR GROUPS
Section: &PATH

Idem as above, here for mean free energy path search.

PROJECT WAVEFUNCTION
Section: &PROP

The wavefunctions are projected on atomic orbitals.
The projected wavefunctions are then used to calculate atomic populations and bond orders. The atomic orbitals to project on are taken from the &BASIS section. If there is no &BASIS section in the input a minimal Slater basis is used. See section 9.5.3 for more details.

PROJECT [NONE, DIAGONAL, FULL]
Section: &CPMD

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

PROP_TSTEP
Section: &PTDDFT

Propagation timestep used in Ehrenfest dynamics. It is used in the spectra calculation (**PROPAGATION SPECTRA**) to specify the time step for the propagation of the KS orbitals.

PROPAGATION SPECTRA
Section: &CPMD

Calculates the electronic absorption spectra using the TDDFT propagation of the Kohn-Sham orbitals. Use the section &PTDDFT to define the parameters. Use this principal keyword always with CAYLEY (in &CPMD). The program produces a file "dipole.dat" with the time series of the variation of the dipole in x, y, and z directions. After Fourier transform of this file one gets the desired absorption spectra.
Typical (minimal) input file (for the sections &CPMD and &PTDDFT) &CPMD PROPAGATION SPECTRA RESTART WAVEFUNCTION COORDINATES LATEST CAYLEY &END
&PTDDFT
ACCURACY
1.0D-8
N_CYCLES
100000
PROP_TSTEP
0.01
EXT_PULSE
1.D-5
PERT_DIRECTION
1
RESTART
2
&END
The time step is specified by setting **PROP-TSTEP**. The total number of iteration is controlled by **N-CYCLES**.

PROPERTIES

Section: &CPMD

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

PROPERTY { STATE }

Section: &TDDFT

Calculate properties of excited states at the end of an **ELECTRONIC SPECTRA** calculations. default is to calculate properties for all states. Adding the keyword **STATE** allows to restrict the calculation to only one state. The number of the state is read from the next line.

RING-POLYMER DYNAMICS

Section: &PIMD

Perform ring-polymer molecular dynamics (RPMD) [122]. In the present implementation the fictitious mass of the nuclei for all the replicas is set to the real physical mass divided by the number of replicas (P) in the primitive representation and the physical mass in the normal mode representation for compatibility with the formulation of the conventional path integral. According to our choice of the fictitious masses, the target temperature you should set is not TP in the original formulation of Ref. [122] but just T as in the standard path integral and centroid MD. Note that RPMD in the staging mode representation is not supported and the generalized Langevin thermostat is available in the normal mode representation only.

QMMM [QMMMEASY]

Section: &CPMD

Activate the hybrid QM/MM code. This keyword requires further input in the section &QMMM . . . &END.
The QM driver is the standard CPMD. An interface program (**MM_Interface**) and a classic force field (Gromos[123]/Amber[124]-like) are needed to run the code in hybrid mode[19, 20, 21, 125, 126]. This code requires a *special licence* and is **not** included in the standard CPMD code. (see section 11.16 for more information on the available options and the input format).

QS_LIMIT

Section: &LINRES

Tolerance above which we use quadratic search algorithm in linear response calculations.

QUENCH [IONS, ELECTRONS, CELL, BO]

Section: &CPMD

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.

RAMAN

Section: &RESP

Calculate the polarizability (also in periodic systems) as well as Born-charges and dipole moment.

RANDOMIZE [COORDINATES, WAVEFUNCTION, DENSITY, CELL]

Section: &CPMD

The **ionic positions** or the **wavefunction** or the **cell parameters** are **randomly displaced** at the beginning of a run.
The maximal amplitude of the displacement is read from the next line.

RANDOMIZE
Section: &TDDFT

Randomize the initial vectors for the diagonalization in a TDDFT calculation. The amplitude is read from the next line. Default is not to randomize the vectors.

RATTLE
Section: &CPMD

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}.

READ REPLICAS
Section: &PIMD

Read all P replicas from a file with a name to be specified in the following line, for the input format see subroutine rreadf_utils.mod.F90.

REAL SPACE WFN KEEP [SIZE]
Section: &CPMD

The real space wavefunctions are kept in memory for later reuse. This minimizes the number of Fourier transforms and can result in a significant speedup at the expense of a larger memory use. With the option **SIZE** the maximum available memory for the storage of wavefunctions is read from the next line (in MBytes). The program stores as many wavefunctions as possible within the given memory allocation.

REFATOM
Section: &HARDNESS

Specify the reference atom to be used in a hardness calculation on the next line. This option is to be used together with the **ORBITALS** and **LOCALIZE**.

REFERENCE CELL [ABSOLUTE, DEGREE, VECTORS]
Section: &SYSTEM

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.

REFUNCT *functionals*
Section: &DFT

Use a special reference functional in a calculation. This option is not active.

REORDER LOCAL
Section: &TDDFT

Reorder the localized states according to a distance criteria. The number of reference atoms is read from the next line. On the following line the position of the reference atoms within the set of all atoms has to be given. The keyword **LOCALIZE** is automatically set. The minimum distance of the center of charge of each state to the reference atoms is calculated and the states are ordered with respect to decreasing distance. Together with the *SUBSPACE* option in a **TAMM-DANCOFF** calculation this can be used to select specific states for a calculation.

REORDER
Section: &TDDFT

Reorder the canonical Kohn–Sham orbitals prior to a TDDFT calculation. The number of states to be reordered is read from the next line. On the following line the final rank of each states has to be given. The first number given corresponds to the HOMO, the next to the HOMO - 1 and so on. All states down to the last one changed have to be specified, no holes are allowed. This keyword can be used together with the *SUBSPACE* option in a **TAMM-DANCOFF** calculation to select arbitrary states. Default is to use the ordering of states according to the Kohn–Sham eigenvalues.

REPLICA NUMBER
Section: &PATH

Number of replicas along the string.

RESCALE OLD VELOCITIES
Section: &CPMD

Rescale **ionic** velocities after **RESTART** to the temperature specified by either **TEMPERATURE**, **TEMPCONTROL IONS**, or **NOSE IONS**. Useful if the type of ionic thermostatting is changed, (do not use RESTART NOSEP in this case). Note only for path integral runs: the scaling is only applied to the first (centroid) replica.

RESTART *[OPTIONS]*

Section: &CPMD

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 time step. It has higher priority than the COORDINATES option.

ACCUMULATORS Read old **accumulator values**, for example the time step number, 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*.

PHESS Read partial Hessian (Hessian of the reaction core) for **transition state search** or **vibrational analysis** from restart file. Useful with the keywords **PRFO** or **HESSIAN** [DISCO,SCHLEGEL,UNIT] PARTIAL.

LSSTAT Read all **status information** of the **linear scaling optimizers** (L-BFGS and P-RFO) including L-BFGS history but excluding partial Hessian for P-RFO from restart file. The **partial Hessian** is read separately using **RESTART PHESS**. Useful with the keywords **LBFGS** and/or **PRFO**.

ADPTTL Read **wavefunction convergence criteria** at the current point of geometry optimization from restart file. Useful with the keywords **CONVERGENCE** [ADAPT, ENERGY, CALFOR].

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 k points.

DENSITY Restart with electronic density.

GLE Restart the extended variables of the GLE dynamics

PRNG Restart the internal state of the Marsaglia random number generator

CONSTRAINTS Restart with old values for constraints. This option is mainly for restraints with GROWTH option.

EXTRAP Restart from a previously saved wavefunction history. See **EXTRAPOLATE WFN** for details.

CVDFT Read CDFT multiplier V from `CDFT_RESTART`, for HDA run read V and wavefunction of the appropriate state.

ALL Restart with all fields of RESTART file

RESTFILE

Section: &CPMD

The number of distinct **RESTART** files generated during CPMD runs 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 in rotation.

RESTFILE

Section: &PTDDFT

Defines a restart code for the restart of the Ehrenfest dynamics (**MOLECULAR DYNAMICS** EH) and the propagation spectra (**PROPAGATION SPECTRA**). The restart option is read from the next line: 0(=default) restart from the (complex)wavefunctions in the file wavefunctions. This option is used in case of a continuation run; 1. restart from the orbital files WAVEFUNCTION.n, where n is the index of the KS orbital and runs from 1 to the number of states (This states a prepare in a previous run using the KOHN-SHAM ENERGIES principal keyword), 2; restart from the orbitals stored in RESTART (obtained from a optimization run with tight convergence (at least 1.D-7)).

REVERSE VELOCITIES

Section: &CPMD

Reverse the ionic and electronic (if applicable) velocities after the initial setup of an MD run. This way one can, e.g., go “backwards” from a given **RESTART** to improve sampling of a given MD “path”.

RFO ORDER=nsorder
Section: &CPMD

Rational function approximation combined with a quasi-Newton method (using BFGS) for **optimization** of the **ionic positions** is used [120]. A saddle point of order nsorder is searched for.

RHOOUT [BANDS,SAMPLE=nrhoout]
Section: &CPMD

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 (or orbitals) to be plotted and their index (starting from 1) have to be given. If the position specification is a negative number, then the wavefunction instead of the density is written. 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*. The following example will request output of the orbitals or bands number 5, 7, and 8 as wavefunctions:

```
RHOOUT BANDS
  3
  -5 -7 -8
```

With the optional keyword **SAMPLE** the requested file(s) will be written every *nrhoout* steps during an MD trajectory. The corresponding time step number will be appended to the filename.

ROKS {SINGLET, TRIPLET},{DELOCALIZED, LOCALIZED, GOEDECKER}
Section: &CPMD

Calculates the first excited state using Restricted Open-shell Kohn-Sham theory [31]. By default, the singlet state is calculated using the delocalized variant of the modified Goedecker-Umrigar scheme, which is supposed to work in most cases. That is, for doing a ROKS simulation, it is usually sufficient to just include this keyword in the CPMD section (instead of using the **LSE** input). See 11.11 for further information.

ROTATION PARAMETER
Section: &TDDFT

The parameters for the orbital rotations in an optimized subspace calculation (see **TAMM-DANCOFF**) are read from the next line. The total number of iterations (default 50), the convergence criteria (default 10^{-6}) and the step size (default 0.5) have to be given.

RUNGE_KUTTA
Section: &CPMD

Defines the integration schemes used in the Ehrenfest **MOLECULAR DYNAMICS**. Always used this option.

SCALED MASSES [OFF]

Section: &CPMD

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**.

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

Section: &SYSTEM

Scale atomic coordinates of the system with the lattice constants (see **CELL**). You can indicate an additional scale for each axis with the options **SX**, **SY** and **SZ**. For instance, if you indicate SX=sxscale, you give your x-coordinates between 0. and sxscale (by default 1.). This is useful when you use many primitive cells. With the keyword **CARTESIAN**, you specify that the given coordinates are in Cartesian basis, otherwise the default with the **SCALE** option is in direct lattice basis. In all cases, the coordinates are multiplied by the lattice constants. If this keyword is present an output file GEOMETRY.scale is written. This file contains the lattice vectors in Åand atomic units together with the atomic coordinates in the direct lattice basis.

SCALES *scaling_for_functional_1, scaling_for_functional_2, ...*

Section: &DFT

Only available in combination with **XC_DRIVER**. On the same line, rescaling values ($\in [0,1]$) for the different exchange-correlation functionals specified in **FUNCTIONAL** are read in the same order in which the functionals were specified.

HFX_SCALE *scaling_for_exact_exchange*

Section: &DFT

Only available in combination with **XC_DRIVER**. On the same line, a scaling value ($\in [0,1]$) for the exact exchange contribution to the overall xc functional is specified. If used with a built-in hybrid functional, the default value is changed; if used with any other functional, the keyword activates the inclusion of exact exchange. Note that its value should always be set to one if combined with a Coulomb-attenuated functional (in which case the global scaling can be modified by changing the CAM parameters α and β, instead).

KERNEL_SCALES *scaling_for_kernel_1, scaling_for_kernel_2, ...*

Section: &DFT

Identical functionality as described in **SCALES**, but applies to the **LR KERNEL** rather than the **FUNCTIONAL**.

KERNEL_HFX_SCALE *scaling_for_exact_exchange_in_kernel*
Section: &DFT

Identical functionality as described in **HFX_SCALE**, but applies to the **LR KERNEL** rather than the **FUNCTIONAL**.

SCEX
Section: &DFT

Activate the use of the coordinate-scaled exact exchange scheme by Bircher and Rothlisberger[187] for *isolated systems*. Speedups of up to one order of magnitude can be achieved. Please note that the simulation supercell must imperatively span twice the size of the charge density, and that the molecule has to remain centred in the box.

SCALED EXCHANGE
Section: &DFT

Alias for **SCEX**.

SCREENED EXCHANGE {ASHCROFT,CAM,ERFC,EXP}
Section: &DFT

Screening / range separation applied to the Hartree-Fock exchange term *only*. Possible options are:

ASHCROFT
Ashcroft exchange. r_{cut} is read from the next line.

CAM
Coulomb-attenuation method. α, β and μ are read from the next line. Please note that in case attenuation of the (semi-)local part is desired, too, the keyword **COULOMB ATTENUATION** has to be used.

ERFC
Screened exchange using the complementary error function. The switching parameter γ is read from the next line.

EXP
Screened exchange using an exponential function. The switching parameter γ is read from the next line.

SHIFT POTENTIAL
Section: &CPMD

After this keyword, useful in Hamiltonian diagonalization, the shift value $V_{\rm shift}$ must be provided in the next line. This option is used in the Davidson diagonalization subroutine and shifts rigidly the total electronic potential as $V_{\rm pot}(\mathbf{r}) \to V_{\rm pot}(\mathbf{r}) + V_{\rm shift}$ then it is subtracted again at the end of the main loop, restoring back the original $V_{\rm pot}(\mathbf{r})$ that remains basically unaffected once that the calculation is completed.

SLATER [NO]
Section: &DFT

The α value for the Slater exchange functional [127] 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.

SMOOTH
Section: &DFT

A smoothening function is applied to the density [128].
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.

SOC
Section: &CPMD

Compute the spin-orbit coupling (SOC) elements between the two states specified in the next line. To be used together with the principal keyword SPECTRA. The calculation is based on the Auxiliary Many-Electron Wavefunction Approach for TDDFT developed in Ref. [46]. For more information see Ref. [129].

SPLINE [POINTS, QFUNCTION, INIT, RANGE]
Section: &CPMD

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 **5000**.) For calculations with the small cutoffs typically used together with Vanderbilt PP a much smaller value, like 1500 or 2000, is sufficient.
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 plane waves, i.e. for all cell symmetries except simple cubic and fcc.

SSIC

Section: &CPMD

Apply an *ad hoc* Self Interaction Correction (SIC) to the ordinary DFT calculation expressed in terms of total energy as

$$E^{\text{tot}} - a \cdot E_H[m] - b \cdot E_{xc}[m, 0]$$

where $m(\mathbf{x}) = \rho_\alpha(\mathbf{x}) - \rho_\beta(\mathbf{x})$. The value of a must be supplied in the next line, while in the present implementation b is not required, being the optimal values $a = 0.2$ and $b = 0.0$ according to Ref. [130]. These are assumed as default values although it is not always the case [131]. Note that if you select negative $\{a, b\}$ parameters, the signs in the equation above will be reversed. The Hartree electronic potential is changed accordingly as $V_H[\rho] \rightarrow V_H[\rho] \pm a \cdot V_{\text{SIC}}[m]$, being

$$V_{\text{SIC}}[m] = \frac{\delta E_H[m]}{\delta m(\mathbf{x})}$$

where the sign is $+$ for α spin and $-$ for β spin components, respectively. Be aware that this keyword should be used together with LSD (set by default).

STAGING

Section: &PIMD

Use the staging representation [15] of the path integral propagator. It is possible to impose a mass disparity between centroid and non–centroid coordinates by dividing the fictitious masses of only the *non*–centroid $s = 2, \ldots, P$ replicas by the adiabaticity control factor FACSTAGE. This dimensionless factor *must always* be specified in the following line. Note: the eigen–*frequencies* of the $s > 1$ replicas are changed by only $\sqrt{\text{FACSTAGE}}$, see Ref. [61](b). Note: using FACSTAGE $\neq 1.0$ essentially makes no sense within the STAGING scheme, but see its use within CENTROID DYNAMICS and NORMAL MODES.

STATES

Section: &SYSTEM

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

NSUP
Section: &SYSTEM

The number of states of the same spin as the first state is read from the next line.
This keyword makes only sense in spin-polarized calculations (keyword **LSD**).

STATES {MIXED,SINGLET,TRIPLET}
Section: &TDDFT

The number of states to be calculated is read from the next line. The type of state *SINGLET*, *TRIPLET* can be given for non-spinpolarized calculations. Default is to calculate one singlet state for LDA and 1 mixed state for LSD calculations.

STEEPEST DESCENT [ELECTRONS, IONS, CELL],[NOPRECONDITIONING],[LINE]
Section: &CPMD

NOPRECONDITIONING works only for electrons and LINE only for 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 ions optimization, the step length is controlled by the keywords **TIMESTEP** and **EMASS**.

STEPLENGTH
Section: &LINRES

Step length for steepest descent and preconditioned conjugate gradient methods used in linear response calculations. Default is 0.1.

STORE {OFF} [WAVEFUNCTIONS, DENSITY, POTENTIAL]
Section: &CPMD

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).

STRESS TENSOR
Section: &CPMD

Calculate the **stress tensor** every *NSTEP* iteration in a constant volume MD. *NSTEP* is read from the next line. Works also for wavefunction or geometry optimisation. In this case NSTEP is meaningless.

STRESS TENSOR
Section: &SYSTEM

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:

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

STRUCTURE [BONDS, ANGLES, DIHEDRALS, SELECT]
Section: &CPMD

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.
If the option **SELECT** is used the output is restricted to a set of atoms. The number of atoms and a list of the selected atoms has to be given on the next lines.

SUBTRACT [COMVEL, ROTVEL]
Section: &CPMD

If COMVEL is selected, the total momentum of the system is removed, if ROTVEL is selected the global angular momentum of the system is removed. Both options can be used separately and simultaneously. The subtraction is done each **ncomv** or **nrotv** steps, where the value is read in the next line.

If this key is activated but no number provided, the **default** is 10000 steps.

Note: The use of these keywords is strongly recommended for long runs (e.g. $t > 10$ ps) and/or low density systems (e.g. isolated molecules, gas phase & Co.). Otherwise the whole system will start to translate and/or rotate toward a (random) direction with increasing speed and spinning. The "relative" translation within the system slows down correspondingly and thus the system effectively cools down. As a consequence dynamic properties, like self-diffusion coefficients will be wrong.

This option should not be used for systems, where some atoms are kept at fixed positions, e.g. slab configurations. Here the center of mass may (or should) move. Due to the interactions with the fixed atoms, a drift of the whole system should be much reduced, anyways.

Note: since the subtracted kinetic energy is put back into the system by simple rescaling of the ionic velocities, these options is not fully compatible with **NOSE** thermostats.

SURFACE HOPPING

Section: &CPMD

Nonadiabatic dynamics involving the ground state and a **ROKS** excited state[85]. Do NOT use this keyword together with **T-SHTDDFT**, which invokes the surface hopping MD scheme based on TDDFT [132] (see **T-SHTDDFT**).

SURFACE [XY, YZ, ZX]

Section: &SYSTEM

By default, if nothing is specified, assume **periodic boundary** condition in x**- and** y**-direction**. With the extra keywords *XY*, *YZ* or *ZX*, the periodicity of the systems is assumed to be along (x, y), (y, z) or (z, x), respectively.

SYMMETRIZE COORDINATES

Section: &SYSTEM

Input coordinates are **symmetrized** according to the **point group** specified. This only makes sense when the structure already is close to the symmetric one.

SYMMETRY

Section: &SYSTEM

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 [29, 133] with ISOLATED MOLECULE option activated. By default the Hockney method (see **POISSON SOLVER**) is used for solving the Poisson equations. You can use this option in combination with **POLYMER** or **SURFACE** for systems that are periodic in only 1 or 2 dimensions. The default Poisson solver is MORTENSEN in this case. See the Hints and Tricks section for some additional requirements when calculating isolated system.

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**

Warning: This keyword should not be used with the keyword **CELL VECTORS**.

T-SHTDDFT

Section: &TDDFT

Non adiabatic (nonadiabatic, non-adiabatic) Tully's trajectory surface hopping dynamics using TDDFT energies and forces. To be used together with the keywords **MOLECULAR DYNAMICS** BO and **TDDFT** in the &CPMD section (see section 11.8.2). Do NOT use the keyword **T-SHTDDFT** together with the keyword **SURFACE HOPPING** in &CPMD, which invokes the SH scheme based on **ROKS** [85] (see **SURFACE HOPPING**).
For a given initial configuration, the run produces a trajectory that undergoes surface hopping according to the algorithm by Tully adapted to TDDFT [132]. The forces on the excited state surfaces are computed using TDDFT as for the adiabatic case. A sufficiently large number of excited states must be declared using the keyword **STATES** in the section &TDDFT. The initial running surface is specified with the keyword **FORCE STATE** in the section &TDDFT. This can change during the dynamics when a surface hop occurs. After a restart the value of the running state is taken from the file SH_STATE.dat (see below). The run produces a series of additional files: SH_COEFA.dat (absolute value of the state amplitudes), SH_COEFC.dat (their complex values), SH_COUPL.dat (the coupling strength per state), SH_ENERG.dat (the energy of the different states: setp number, ground state energy, first excited state energy, . . . , highest excited state energy, energy of the running state), SH_PROBS.dat (transition probabilities between running state and all other states), SH_STATE.dat (the running state at each step). All these files (in addition to SH_WAVEFUNCTIONS and SH_LRWAVEFUNCTIONS) are needed to restart the SH dynamics. Note that each run produces a single SH trajectory. Several subsequent runs starting from different initial coordinates and velocities are required to collect statistics.

TAMM-DANCOFF [SUBSPACE,OPTIMIZE]
Section: &TDDFT

Use the Tamm–Dancoff approximation. This is the default for TDDFT calculations. Optionally, only a *SUBSPACE* of the occupied orbitals can be included in the calculation. The subspace can be optimized at each step (not recommended). Default is to use all states.

TD_METHOD_A [*functionals*]
Section: &TDDFT

Use a different potential for the eigenvalue differnce part of the response equations than was used to generate the ground state orbitals. The potential generating functional has to be given after the keyword. For possible functionals see the code. Most likely you want to use the **SAOP** functional.
This functional does not affect the choice of functional used in the TDDFT kernel. The kernel functional is set in the &DFT section. It is either the standard functional or the functional defined by the keyword **LR KERNEL**.

TDDFT
Section: &CPMD

Calculate the energy according to TDDFT. This keyword can be used together with **OPTIMIZE GEOMETRY** or **MOLECULAR DYNAMICS** BO. Use the &TDDFT section to set parameters for the calculation. This keyword requires **RESTART** LINRES.

TD_POTENTIAL
Section: &PTDDFT

Defines a time dependent external potential to be used in Ehrenfest dynamics (**MOLECULAR DYNAMICS** EH). Can be used with the keyword **PIPULSE**. The frequency of the external field is read from the next line (in atomic units). For more information see subroutine gaugepot_laser in td_utils.mod.F90 .

TEMPCONTROL [IONS, ELECTRONS, CELL]
Section: &CPMD

The **temperature** of the **ions** in Kelvin or 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.
As a gentler alternative you may want to try the **BERENDSEN** scheme instead.

TEMPERATURE ELECTRON

Section: &CPMD

The **electronic temperature** is read from the next line.
Default is 1000K.

TEMPERATURE [RAMP]

Section: &CPMD

The **initial temperature** in Kelvin of the **system** is read from the next line. With the additional keyword **RAMP** the temperature can be linearly ramped to a target value and two more numbers are read, the ramping target temperature in Kelvin and the ramping speed in Kelvin per atomic time unit (to get the change per timestep you have to multiply it with the value of **TIMESTEP**). Note that this ramping affects the target temperatures for **TEMPCONTROL**, **BERENDSEN** and the global **NOSE** thermostats.

TESR

Section: &SYSTEM

The number of additional supercells included in the real space sum for the Ewald term is read from the next line. Default is 0, for small unit cells larger values (up to 8) have to be used.

THAUTO

Section: &LINRES

The two values read from the next line control the switch to different optimizers for an automatic selection of optimizers during a linear response calculation. This also applies to the Z-vector optimization for TDDFT forces. The first value is the threshold for switching from conjugate gradients to DIIS (with compressed storage and averged preconditioner, subspace size defined with **ODIIS**). The second value is the threshold for switching to DIIS with full storage and state dependent preconditioner. See also **ZDIIS** for specification of the subspace size.

TIGHTPREC

Section: &RESP

Uses a harder preconditioner. For experts: The Hamiltonian is approximated by the kinetic energy, the G-diagonal Coulomb potential and the KS-energies. The number obtained this way must not be close to zero. This is achieved by smoothing it with This is achieved by smoothing it with

$$x \to f(x) = \sqrt{x^2 + \epsilon^2} \text{ [default]}$$

or

$$x \to f(x) = (x^2 + \epsilon^2)/x \text{ [this option]}$$

The HARD option conserves the sign of the approximate Hamiltonian whereas the default formula does never diverge.

TIMESTEP ELECTRONS

Section: &CPMD

The time step for electron dynamics in atomic units is read from the next line. This is can be used to tweak the convergence behavior of the wavefunction optimization in Born-Oppenheimer dynamics, where the default time step may be too large. see, e.g. **PCG**

TIMESTEP IONS

Section: &CPMD

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

TIMESTEP

Section: &CPMD

The time step in atomic units is read from the next line.
Default is a time step of **5 a.u.** ($1\,a.u. = 0.0241888428$ fs).

TIMESTEP_FACTOR

Section: &MTS

The time step factor (n) used in the MTS scheme is read from the next line.

For instance, if **TIMESTEP IONS** is set to **15 a.u.**, and ($n = 4$) then the high level correction to the forces in the MTS scheme will be calculated every 4 steps, i.e. every **60 a.u.** **Default** is a factor of 1, i.e. the MD will be identical to a Velocity-Verlet MD with the high level forces.

TRACE [ALL,MASTER]

Section: &CPMD

Activate the tracing of the procedures. *ALL* specifies that all the mpi tasks are traced. *ALL* specifies that only the master is traced.

TRACE_PROCEDURE
Section: &CPMD

Select a procedure to be traced. The procedure is read from the next line.

TRACE_MAX_DEPTH
Section: &CPMD

Set the maximal depth for tracing. The depth is read from the next line.

TRACE_MAX_CALLS
Section: &CPMD

Set the maximal number of calls for tracing. The number is read from the next line.

TRAJECTORY [OFF, XYZ, DCD, SAMPLE, BINARY, RANGE, FORCES]
Section: &CPMD

Store the atomic positions, velocities and optionally forces at every *NTRAJ* time step on file *TRAJECTORY*. This is the **default for MD runs**. With the additional keyword XYZ the trajectory is also writthen in xyz-format on the file *TRAJEC.xyz*, similarly with the additional keyword DCD a trajectory in dcd-format (binary and single precision, as used by CHARMM, X-PLOR and other programs) is written on the file TRAJEC.dcd. If the keyword SAMPLE is given *NTRAJ* is read from the next line, otherwise the default value for *NTRAJ* is 1. A negative value of *NTRAJ* will disable output of the *TRAJECTORY* file, but e.g. TRAJEC.xyz will still be written every -*NTRAJ* steps. A value of 0 for *NTRAJ* will disable writing of the trajectory files altogether.
The TRAJECTORY file is written in binary format if the keyword BINARY is present. If FORCES is specified also the forces are written together with the positions and velocities into the file FTRAJECTORY. It is possible to store the data of a subset of atoms by specifying the suboption RANGE, the smallest and largest index of atoms is read from the next line. If both, SAMPLE and RANGE are given, the RANGE parameters have to come before the SAMPLE parameter.

TRANSITION MOMENT
Section: &PROP

Calculate the dipole transition matrix element.
On the following lines, the number of transitions and the involved orbitals are given.
Example:

TRANSITION MOMENT
```
2
6     7
6     8
```
This calculates the dipole transition matrix elements between KS states 6 and 7, and between 6 and 8.

TROTTER DIMENSION

Section: &PIMD

The Trotter number P, i.e. the number of "replicas", "beads", or "imaginary time slices" which are used in order to discretize the Feynman–Kac path integral of the nuclei, is read from the next line. If NORMAL MODES or STAGING is not activated the path integral is discretized in cartesian coordinates in real space (so–called "primitive coordinates"). A discussion about controlling discretization errors and on estimating P in advance is given in Ref. [134].

TROTTER FACTORIZATION OFF

Section: &CPMD

Do not use Trotter factorization to calculate free energy functional.
Remark: Place this keywords only after FREE ENERGY FUNCTIONAL; before it has no effect. Note: this keyword has *nothing* to do with path integral MD as activated by the keyword PATH INTEGRAL and as specified in the section &PIMD ... &END.

TROTTER FACTOR

Section: &CPMD

Solve $e^{-H/k_B T}$ directly using **Trotter approximation**
$\left(e^{-pH} \simeq e^{-pK/2} e^{-pV} e^{-pK/2}\right)$.
The Trotter approximation is twice as fast.
The Trotter factor is read from the next line (typically 0.001 is very accurate).

USE_IN_STREAM

Section: &CPMD

Specify that the RESTART file shall be read in the stream mode. This allows for parallel restarting (to be used with *USE_MPI_IO*).

USE_OUT_STREAM

Section: &CPMD

Specify that the RESTART file shall be written in the stream mode. This allows for parallel restarting (to be used with *USE_MPI_IO*).

USE_MPI_IO
Section: &CPMD

Specify that MPI shall be used for parallel reading/writing of the RESTART file. This shall be used with *USE_IN_STREAM* and /or *USE_OUT_STREAM*.

USE_MTS
Section: &CPMD

Switch on the Multiple Time-Step scheme for molecular dynamics (see &MTS section).[135]

VDW CORRECTION [ON,OFF]
Section: &CPMD

An empirical van der Waals correction scheme is applied to pairs of atom types specified with this keyword. This activates reading the corresponding parameters from the &VDW ... & END in which you have to specify all the VDW parameters between the opening and closing section keywords EMPIRICAL CORRECTION and END EMPIRICAL CORRECTION. Note that the two possible vdW options, EMPIRICAL CORRECTION and WANNIER CORRECTION are mutually exclusive. See **VDW PARAMETERS** for more details.

VDW DCACP
Section: &VDW

Synonymous to **DCACP**.

VDW PARAMETERS
Section: &VDW

Parameters for empirical van der Waals correction schemes are set with the keyword. This requires the **VDW CORRECTION** keyword to be set in the &CPMD section. For Grimme's **DFT-D2** and **DFT-D3** type (see below) an automatic assignment of the parameters can be requested by putting **ALL DFT-D2** or **ALL DFT-D3** on the next line. Otherwise the number of pairs *NVDW* is read from the next line and followed by *NVDW* lines of parameters: *TYPE*, α, β, $C_6^{\alpha\beta}$, $R_0^{\alpha\beta}$, and d for each pair of atom types α and β, where α and β are the indexes of pseudopotentials (and their associated groups of atoms) in the order they are listed in the &ATOMS section. For type **DFT-D2** and **DFT-D3** only α and β are required. If the other parameters are omitted the internal table of parameters is used.
A presently implemented damped dispersion model, described by M. Elstner *et al.*[30], having the same form as that constructed by Mooij *et al.*[136], is activated by specifying **C6** as *TYPE*. This model is expressed as

$$E_{vdW} = \sum_{ij} \frac{C_6^{\alpha\beta}}{R_{ij}^{\alpha\beta\,6}} \left(1 - \exp\left[-d\left(\frac{R_{ij}^{\alpha\beta}}{R_0^{\alpha\beta}}\right)^7\right]\right)^4 . \tag{1}$$

A table of parameters appropriate for this particular model, using the PBE and BLYP functionals, is available [137].
Alternatively Van der Waals correction according to Grimme can be used [38] by selecting *TYPE* **DFT-D2** and its D3 version [39] with **DFT-D3**.

$$E_{disp} = -s_6 \sum_{i=1}^{N_{at}-1} \sum_{j=i+1}^{N_{at}} \frac{C_6^{ij}}{R_{ij}^6} f_{dmp}(R_{ij}) \tag{2}$$

The values of C_6 and R_0 are not specific that are used by this method are taken from [38] and stored internally (see above for details). Namely, all elements from H ($Z = 1$) to Rn ($Z = 86$) are available, whereas elements beyond Rn give by default a zero contribution. Note that the parameter s_6 depends on the functional used and has to be provided consistently with the DFT one chosen for the calculation. The following line has to be added S6GRIMME and the type of functional is read from the next line. One of the following labels has to be provided: BP86, BLYP, B3LYP, PBE, TPSS, REVPBE, PBE0. Note that Grimme vdW does not support other functionals. In the **DFT-D3** version [39] there is no need to specify manually s_6

VDW-CUTOFF

Section: &VDW

On the next line the short range cutoff of van der Waals correction has to be specified. The default value is 10^{-2}.

VDW-CELL

Section: &VDW

The number of additional supercells to be included in the sum of van der Waals correction.

VDW WANNIER
Section: &CPMD

A first-principle van der Waals correction scheme [179, 180, 181] is applied to selected groups of atoms on which maximally localized Wannier functions (WF) and centers (WFC) have been previously computed. The file WANNIER-CENTER generated upon WFC calculation must be present. This activates the reading procedure of the corresponding parameters from the &VDW ... &END section.

WANNIER CORRECTION ... END WANNIER CORRECTION
Section: &VDW

Between these opening and ending keywords, the partitioning of the system and the calculation procedure must be selected. Three implementations are available for partitioning the system: (1) choosing a *zlevel*, namely a z coordinate separating the first fragment form the second (this is appropriate for cases where there are only two fragments such as, for instance two graphene layers or adsorption of molecules on surfaces); in this case the keyword FRAGMENT ZLEVEL must be used. (2) give reference ion and a cut-off radius around which WFCs are supposed to belong to the given atom or fragment; in this case the keyword FRAGMENT RADIUS must be used. (3) the system is subdivided into fragments automatically detected by using predefined covalent bond radii. in this case the keyword FRAGMENT BOND must be used. This is also the default in case no specification is done.
The syntax for the different options is:
VERSION
iswitchvdw (method 1 [179] or 2 [180])
FRAGMENT ZLEVEL
zlevel (in a.u.)
FRAGMENT RADIUS
multifrag
i radius(i)
...
FRAGMENT BOND
tollength
DAMPING [DIPOLE]
a6
RESTART WANNIER
ENERGY MONOMER
enmonomer
TOLERANCE WANNIER
tolwann
TOLERANCE REFERENCE
tolref
CHANGE BONDS
nboadwf
i j ± 1
CELL
nxvdw nyvdw nzvdw
PRINT [INFO,FRAGMENT,C6,FORCES]
Note that the total number of WFCs in your system depends on the spin description you use (1 for LSD, 2 for LDA). The coefficient a6 is the smoothing parameter and the reference total energy intended as a sum of all the total energies of your fragments (e.g. the ETOT you get by a standard calculation not including vdW corrections). For a6 the suggested parameter is 20.0 [182]. An alternative method avoiding the use of the empirical parameter a6 is the one exploiting the replacement of the short-range damping function by an estimation of the Pauli exchange repulsion given in terms of solely Wannier function centers and spreads [181]. To activate this option the keyword DAMPING has to be given along with DIPOLE on the same line. The following line (a6) has to be obviously removed, since anyhow is neither read nor used in this (DAMPING DIPOLE) case. Note that the two possible vdW options, EMPIRICAL CORRECTION and WANNIER CORRECTION are mutually exclusive.

VELOCITIES ... END VELOCITIES

Section: &ATOMS

Sets an **initial velocity** for specified atoms.
The first line contains first the total number of specified atomic velocities followed **on the same line** by the list of atomic numbers for which the velocities will be read. On each of the following lines the x, y and z coordinates of the velocities of an atom have to be specified. These values will ignored in case of starting with **RESTART** VELOCITIES..
NOTE: these velocities are rescaled to produce the initial temperature as specified by **TEMPERATURE**. The default temperature, however, is 0K, so you **have** to set the matching temperature or your initial velocities will be useless.

VGFACTOR

Section: &CPMD

For **CDFT** runs read the inverse of the gradient optimiser step size ($1/dx$) from the next line. The standard value of **10.0** should be fine in most situations.

VIBRATIONAL ANALYSIS [FD, LR, IN], [GAUSS, SAMPLE, ACLIMAX]

Section: &CPMD

Calculate harmonic frequencies by finite differences of first derivatives **(FD)** (see also keyword **FINITE DIFFERENCES**), by **linear response** to ionic displacements **(LR)** or from a **pre-calculated** Hessian **(IN)**. K-point sampling is currently possible using finite differences. If the option GAUSS is specified, additional output is written on the file *VIB1.log* which contains the modes in a style similar to GAUSSIAN 98 output. This file can be read in and visualized with programs like MOLDEN or MOLEKEL. The option SAMPLE reads an integer from the next line. If this number is 2 an additional file *VIB2.log* containing the lowest modes is written. The **default** value is 1. If the option ACLIMAX is specified, additional output is written on the file VIB.aclimax which contains the modes in a style readable by aClimax (http://www.isis.rl.ac.uk/molecularspectroscopy/aclimax/). If a section **&PROP** is present with the keyword **DIPOLE MOMENT**[BERRY] or **DIPOLE MOMENT**[RS], the Born charge tensor is calculated on the fly. See also the block &LINRES ... &END and the keywords **RESTART** PHESS and **HESSIAN** {DISCO,SCHLEGEL,UNIT} PARTIAL.

VMIRROR

Section: &CPMD

For **CDFT** HDA runs initialise V for the second state as the negative final V value of the first state. Useful in symmetric systems.

WANNIER DOS

Section: &CPMD

Outputs the projected density of states of the Wannier orbitals (file WANNIER_DOS) and the KS Hamiltonian in the Wannier states representation (file WANNIER_HAM). When running **MOLECULAR DYNAMICS** CP the files WANNIER_DOS and WANNIER_HAM solely written at the last step.

WANNIER MOLECULAR

Section: &CPMD

Generates effective molecular orbitals from the Wannier representation. It first attributes Wannier orbitals to molecules and then diagonalizes by molecular blocks the KS Hamiltonian.
Does not work with **MOLECULAR DYNAMICS** CP.

WANNIER NPROC

Section: &CPMD

Set the number of mpi tasks to be used for localization. Default is to use all the tasks available. The number of tasks is read from the next line and shall be a divisor of the number of tasks in a parallel run.

WANNIER OPTIMIZATION {SD,JACOBI,SVD}

Section: &CPMD

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

WANNIER PARAMETER

Section: &CPMD

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

Section: &CPMD

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. $\mathbf{R}_I \to \mathbf{R}_I - (x, y, z)$). The default value is the center of the supercell, if **CENTER MOLECULE** keyword is active (Note, that this is implicitly turned on, for calculations with **SYMMETRY** 0). Otherwise it is set to (0,0,0), which is usually not the center of the box. In order to get the best results displaying the IONS+CENTERS.xyz file this parameter should be set explicitly.

WANNIER RELOCALIZE_IN_SCF
Section: &CPMD

If present, relocalize/project the wavefunction at every SCF step.

WANNIER RELOCALIZE_EVERY
Section: &CPMD

If present, relocalize the wavefunction at every SCF step with the Jacobi method. The stride is read from the next line.

WANNIER SERIAL
Section: &CPMD

Requests that the calculation of Wannier functions is performed using the serial code, even in parallel runs.

WANNIER TYPE {VANDERBILT,RESTA}
Section: &CPMD

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

WANNIER WFNOUT [ALL,PARTIAL,LIST,DENSITY]
Section: &CPMD

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. For **PARTIAL** output you have to give the indices of the first and the last Wannier function to print; the *LIST* directive follows the syntax of **RHOOUT** *BANDS*.

```
WANNIER WFNOUT PARTIAL
   5  8
```

WCUT *CUT*
Section: &SYSTEM

Set the radial **CDFT** weight cutoff for all atom species to CUT, which is specified next to the keyword. Default is a species specific cutoff at the distance where the magnitude of the respective promolecular density is smaller than 10^{-6}.

WGAUSS *NWG*
Section: &SYSTEM

Use Gaussian weight functions instead of Hirshfeld promolecular orbitals in the **CDFT** weight. Parameter NWG is specified next to the keyword and has to be equal to the number of different atom species in the calculation. The Gaussian widths σ_i of the species i are read from subsequent lines.

WOUT [FULL]
Section: &CPMD

Controls the printing of the CDFT weight(s). If the keyword FULL is set the full weight is written out in the form of a density to WEIGHT-(suff), where (suff) is defined by the kind of the CDFT job. (suff)=WFOPT for single point calculations, while for geometry optimisations and MD two weights are written, (suff)=INIT at the beginning and (suff)=FINAL for the last step. If FULL is not set write out a slice of the weight in gnuplot readable form to WEIGHT-(suff).dat. Parameters WSLICE and WSTEP are read from the next line.
WSLICE **0.5** is if larger than zero the z coordinate of the x-y weight plane to write out divided by the total box height. If WSLICE< 0 the weight at the z coordinate of the first acceptor atom will be used.
WSTEP **1** is the grid point step size for the output.

XC_ANALYTIC
Section: &LINRES

Use analytic second derivatives of the XC functional (only available for some LDA functionals)

XC_EPS
Section: &LINRES

Finite difference parameter for XC derivative. Default is $5 \cdot 10^{-4}$.

XC_DD_ANALYTIC
Section: &LINRES

Use analytic second derivatives of the XC functional, see Ref. [32] (only available for some LDA and gradient-corrected functionals). For the analytic third derivatives of some LDA XC functionals, **XC_ANALYTIC** can be combined with this keyword

XC_DRIVER
Section: &DFT

Use the new CP xc driver. This keyword replaces both **OLDCODE** and **NEWCODE** and offers more ample customisation options (*cf.* **COULOMB ATTENUATION**, **LIBRARY** and **SCALES**). All functionals are now consistently available for both spin-restricted and open-shell systems. Additionally, functionals from libxc can be used and freely mixed with the internal functionals, *cf.* **LIBRARY**. Please note that the new **FUNCTIONAL** abbreviations differ from **OLDCODE** and **NEWCODE**, *cf.* the corresponding entry for a list of available functionals. The new driver uses the standard definitions for all functionals; where those definitions differ from older versions of CPMD, the old definitions (where implemented) can be accessed by using the keyword **OLD_DEFINITIONS**.

ZDIIS

Section: &LINRES

The subspace size for the optimizer is read from the next line.

ZFLEXIBLE CELL

Section: &SYSTEM

Specifies a constraint on the super cell in constant pressure dynamics or geometry optimizations. The supercell may only shrink or grow in z-direction. Should be very useful for "dense slab" configurations, e.g. a water layer between solid slabs.
Please note: this is by no means intended to give a statistically meaningful ensemble, but merely to provide a tool for efficient equilibration of a specific class of system.

[TSDE, TSDP, TSDC] [NOPRECONDITIONING] *NOPRECONDITIONING only electrons*

Section: &CPMD

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

n-CENTER CUTOFF

Section: &PROP

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

9.5 Further details of the input

9.5.1 Pseudopotentials

The general format for entering the pseudo potentials in the input file 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-HERMITE, KLEINMAN-BYLANDER]
      specifies the method to be used for the calculation of the **nonlocal parts** of the **pseudopotential** [138]. It can be omitted for Vanderbilt pseudopotentials and Stefan Goedecker's pseudopotentials. For semi-local pseudopotentials the default is Gauss-Hermite integration with 20 special points. The number of integration points can be changed using **GAUSS-HERMITE=xx**.
    - . It is further possible to specify **nonlinear core correction** [**139**] [**NLCC**] and the width of the **ionic charge distribution** [**RAGGIO**]. (**Default** is **no NLCC** and the default value for **RAGGIO** is **1.2**.)
    - . The label **UPF** indicates that the pseudopotential was stored using the Universal Pseudopotential Format.
    - . For **Vanderbilt ultrasoft pseudopotentials** one of the following options has to be specified: **BINARY** indicates the binary version of the output file from Vanderbilt's atomic code.
      FORMATTED indicates the formatted version of the Vanderbilt pseudopotential files after a conversion with the program 'reform.f' from the Vanderbilt atomic code package (see section 4)
      For Vanderbilt pseudopotentials the option NLCC is ignored. The nonlinear core correction will always be used if the pseudopotential was generated with a partial core.
      It is strongly recommended to use only Vanderbilt pseudopotentials that were generated with a new version of Vanderbilt's atomic code (version 7.3.0 or newer).
    - . The label **CLASSIC** indicates that the following atoms are to be treated with classical force fields only. See section &CLASSIC for more information.
    - . The label **_EAM_** indicates that the following atoms are treated using the EAM approach.
    - . The label **FRAC** indicates that the core charge of a pseudopotential should not be rounded for the calculation of the number of electrons (for pseudopotentials with fractional core charge).
    - . The label **ADD_H** indicates that the potential should used to saturate dangling bonds or "hydrogenize" united atom potentials in a CPMD/Gromos-QM/MM calculation (see section 11.16 for more details).

- The next line contains information on the **nonlocality** of the **pseudopotential** (*LMAX, LOC, SKIP*) [140, 141, 142].

- 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). For CPMD/Gromos-QM/MM calculation, however, the Gromos atom numbers have to be given instead of coordinates (see section 11.16 for more details).

The information on the **nonlocal part** of the pseudopotential [140, 141, 142] 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 [143] this line has to be in a **valid format**, but the actual values are **not** used.

9.5.2 Constraints and Restraints

CONSTRAINTS ... END CONSTRAINTS

Within this input block you can specify several **constraints** and **restraints** on the atoms. Please note, that for calculations using the Gromos QM/MM-interface (see section 11.16) the atom indices refer to the ordering of the atoms as it appears in the respective Gromos coordinate file. In all cases the indices of dummy atoms start sequentially from total-number-of-atoms plus one. The following suboptions are possible:

FIX ALL
All coordinates of **all atoms** are kept fixed.
For wavefunction optimization via simulated annealing.

FIX QM
All coordinates of **all QM atoms** are kept fixed.
This is the same as above unless you are running a QM/MM calculation with the Gromos interface code.

FIX MM
All coordinates of **all MM atoms** are kept fixed.
This is ignored unless you are running a QM/MM calculation with the Gromos interface code.

FIX SOLUTE
All coordinates of **all solute atoms** are kept fixed.
This is ignored unless you are running a QM/MM calculation with the Gromos interface code. The definition of what is a solute is taken from the respective GROMOS topology file.

FIX SEQUENCE
All coordinates of a series of atoms are kept fixed.
This keyword is followed by the index numbers of the first and the last atoms to be fixed in the next line. Example:

FIX SEQUENCE

```
5 25
```
all coordinates of atoms no. 5 to 25 are kept fixed

FIX ELEMENT [SEQUENCE]

All coordinates of all atoms belonging to the same **element** are kept fixed. This works across pseudopotential types or QM and MM atoms in case of a QM/MM calculation. The keyword is followed by the core charge of the respective element. With the optional SEQUENCE modifier two more numbers are read in, specifying the first and the last index of a sequence of atoms to which this keyword will be applied. Example:

FIX ELEMENT

`8` *all coordinates of oxygen atoms are kept fixed*

FIX PPTYPE [SEQUENCE]

All coordinates of all atoms belonging to the same potential type are kept fixed. The keyword is followed by the atom type index number on the next line, corresponds to the sequence of how the atom types are specified in the &ATOMS section of the CPMD input. In case of a QM/MM calculation this is expanded to respective classical atom types. In this case the QM atom types come first followed by the GROMOS atom types. With the optional SEQUENCE modifier two more numbers are read in, specifying the first and the last index of a sequence of atoms to which this keyword will be applied. Example:

FIX PPTYPE SEQUENCE

`2 5 25` *atoms corresponding to the second atom type with an index between 5 and 25 are kept fixed*

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 in the input file (NOTE: in the file GEOMETRY.xyz the atoms have the same ordering). Example:

FIX ATOMS

`5 2 5 20 21 23` *all coordinates of atoms 2, 5, 20, 21, and 23 are kept fixed*

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.
NOTE: This currently works only for **OPTIMIZE GEOMETRY** and not for the **LBFGS** optimizer.

FIX STRUCTURE [SHOVE]

This keyword starts a group of individual constraints where whole **structural units** can be fixed. The keyword is followed by the number of individual constraints on the next line.

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 $n2$.

DIFFER $n1$ $n2$ $n3$ R

Fixes $R_{12} - R_{23}$ defined by the atoms $n1$, $n2$, and $n3$, where R_{ab} is the distance between atoms a and b.

DISAXIS $n1$ $n2$ k R_0 [+1,-1,0] GROWTH aa DEST Rd
Distance between two atoms $n1$ and $n2$ along x or y or z direction. $n1$ $n2$ k and the value R_0 are read next on the same line. Here $k = 1$ means x, $k = 2$ means y and $k = 3$ means z coordinate.

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

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

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

RIGID nr $n1$ $n2$ ... nx
Keeps the structure formed by the nr atoms $n1$, $n2$, ...
You can put your atom index in several lines. The number of constraints **nfix** is equal to $3nr - 6$ for $nr > 2$ ($nfix = 1$ for $nr = 2$).

COORD $n1$ κ Rc d^0
Constraint on the coordination (number of atoms around a selected one within a specific spherical range of radius $\sim Rc$) for atom $n1$. The parameters κ and Rc for the Fermi function are given in Bohr (Rc) and 1/Bohr (κ), or in Å(Rc) and 1/Å(κ), if the keyword **ANGSTROM** is set. See Ref. [144].

COORSP $n1$ jsp κ Rc d^0
Fixes the coordination number (CN) of one selected atom i with respect to only one selected species jsp. The CN is defined by a Fermi-like function as for *COORD*, but in this case j runs only over the atoms belonging to the selected species jsp.

COOR_RF $n1$ jsp p q Rc d^0
CN of one selected atom i with respect to one selected species, jsp. The CN value is calculated as the sum of rational functions

$$CN_i = \sum_{j \neq i}^{n_{list}} \frac{1 - \left(\frac{d_{ij}}{d^0}\right)^p}{1 - \left(\frac{d_{ij}}{d^0}\right)^{p+q}}, \tag{3}$$

where j runs over the indexes of the atoms belonging to jsp or over the indexes given in the list $j1 \cdots jn_{list}$.

BNSWT $n1$ $n2$ p q Rc d^0
Reciprocal CN between 2 selected atoms, defined with the same functional form as the one described for *COOR_RF*. This coordinate states the presence of the bond between the two atoms i and j.

TOT_COOR isp jsp p q Rc d^0
Average CN of the atoms belonging to a selected species isp with respect to a second selected species, jsp, or with respect to a given list of atoms, $j1 \cdots jn_{list}$. The same functional forms and input options are used, as those described for *COOR_RF*, but the index of one selected species isp is read in place of the index of one atom.

$n1$, ... 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 Lagrangian according to the *Blue Moon* ensemble prescription[145]. The values of the Lagrange multipliers and of the actual constraint are printed in the file CONSTRAINT.
The options **DIST, STRETCH, BEND, TORSION, OUTP, DIFFER, COORD, COORSP, COOR_RF, TOT_COOR** can have an optional additional keyword at the end of the line of the form

```
DIST 1 2 -999.  GROWTH 0.001
```

The keyword **GROWTH** indicates that the constraint value should be changed at each

time step. The rate of change is given after the keyword in units per atomic time unit, i.e. **independent** from the current length of a time step.
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).

RESTRAINTS [HYPERPLANE [K=scal]]
Defines restraints.

nres
Number of restraints.

DIST $n1$ $n2$ R $kval$
Restrains the distance R between the atoms $n1$ and $n2$ by a harmonic potential having spring constant $kval$.

STRETCH $n1$ $n2$ R $kval$
Restrains R^2 defined by the atoms $n1$ and $n2$ by a harmonic potential having spring constant $kval$.

DIFFER $n1$ $n2$ $n3$ R $kval$
Restrains $R_{12} - R_{23}$ defined by the atoms $n1$, $n2$, and $n3$, where R_{ab} is the distance between atoms a and b by a harmonic potential having spring constant $kval$.

DISAXIS $n1$ $n2$ k R_0 $kval$
Restraints the distance between two atoms $n1$ and $n2$ along x or y or z direction. $n1$ $n2$ k and the value R_0 are read next on the same line. Here $k = 1$ means x, $k = 2$ means y and $k = 3$ means z coordinate.

BEND $n1$ $n2$ $n3$ θ $kval$
Restrains the bending angle defined by the atoms $n1$, $n2$ and $n3$ by a harmonic potential having spring constant $kval$.

TORSION $n1$ $n2$ $n3$ $n4$ Θ $kval$
Restrains the torsion angle defined by the atoms $n1$, $n2$, $n3$ and $n4$ by a harmonic potential having spring constant $kval$.

OUTP $n1$ $n2$ $n3$ $n4$ Θ $kval$
"Out of Plane"; Angle between plane ($n1$, $n2$, $n3$) and atom $n4$ is restrained by a harmonic potential having spring constant $kval$.

COORD $n1$ K RC $C0$ $kval$
Coordination restraint for atom $n1$. The parameters K and RC for the Fermi-like function are given in Bohr (RC) and 1/Bohr (K), or in Å(RC) and 1/Å(K), if the keyword **ANGSTROM** is set. See Ref. [144]. The harmonic potential spring constant is $kval$.

COORSP $n1$ is K RC $C0$ $kval$
Retraint on the coordination number (CN) of one selected atom $n1$ with respect to a single selected species is. The CN is defined by a Fermi-like function as for *COORD*. As in all the above cases, the harmonic potential spring constant is $kval$.

COOR_RF $n1$ is N M RC $C0$ $kval$
Restraint on CN of one selected atom $n1$ with respect to one selected species, is. The CN value is calculated as the sum of rational functions

$$CN_i = \sum_{j \neq i}^{n_{list}} \frac{1 - \left(\frac{d_{ij}}{d^0}\right)^p}{1 - \left(\frac{d_{ij}}{d^0}\right)^{p+q}}, \qquad (4)$$

where j runs over the indexes of the atoms belonging to is or over the indexes given in the list $j1 \cdots jn_{list}$. As in all the above cases, the harmonic potential spring constant is $kval$.

BNSWT $n1$ $n2$ N M RC $C0$ $kval$

Reciprocal CN between 2 selected atoms, defined with the same functional form as the one described for $COOR\_RF$. This coordinate states the presence of the bond between the two atoms $n1$ and $n2$. As in all the above cases, the harmonic potential spring constant is $kval$.

TOT_COOR $is1$ $is2$ N M RC $C0$ $kval$

Average CN of the atoms belonging to a selected species $is1$ with respect to a second selected species, $is2$, or with respect to a given list of atoms, $j1 \cdots jn_{list}$. The same functional forms and input options are used, as those described for $COOR\_RF$, but the index of one selected species isp is read in place of the index of one atom. As in all the above cases, the harmonic potential spring constant is $kval$.

RESPOS $n1$, x_0, y_0, z_0 d_0 $kval$

Restrains the position $\mathbf{R} = (x, y, z)$ of atom $n1$ to oscillate around $\mathbf{R}_0 = (x_0, y_0, z_0)$ with a constraint harmonic potential $V_c = (kval/2)(|\mathbf{R} - \mathbf{R}_0| - d_0)^2$ [146]. The limits $kval = 0$ and $kval \to \infty$ correspond to free and fixed atomic positions, respectively. The keyword GROWTH is not supposed to be used for this restraint. For the sake of clarity and consistency with the atomic units used through the code, coordinates and distances are expected to be in atomic units (not Å).

$n1$, ... are the atom numbers, R distances and Θ angles. A function value of -999. for R or Θ refers to the current value. The restraining potential is harmonic with the force constant $kval$. The options can have an optional additional keyword at the end of the line of the form

```
DIST 1 2 -999.  0.1 GROWTH 0.001
```

The keyword **GROWTH** indicates that the constraint value should be changed at each time step. The rate of change is given after the keyword in units per atomic time unit.

If the keyword **HYPEPLANE** is set, the system is not restrained around a point in the collective variable space but in an hyperplane. This hyperplane is defined as going through a point in the collective variable space, defined from the R and Θ above, and by a vector defined from the $kval$ values. **K=scal** applies a scaling to the vector defining the hyperplane so as to modulate the strength of the restraint.

The energy formula for an hyperplane restraint is then:

$E_r = \frac{1}{2}\left((\vec{c} - \vec{c}_0) \cdot \vec{n}\right)^2$,

where the vectors are vectors in the collective variable space.

If a file **RESVAL** is found after parsing the input, the current restraint target values will be replaced by the values found in this file.

PENALTY

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

9.5.3 Atomic Basis Set

The &BASIS section is used in CPMD only to provide an atomic basis set for generating the initial guess and for analyzing orbitals. If the input file contains no &BASIS section, a minimal Slater basis is used.

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 AO 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
```

9.5.4 Van der Waals potential

This section (&VDW . . . &END) contains information about the van der Waals correction scheme. Currently, three schemes are available:

- A Wannier-center based approach (**VDW WANNIER**)
- A non-local potential-based approach (**DCACP** [176])
- A parametrized force-field-like potential (**VDW CORRECTION**)

The first two approaches both depend on the actual electron density of the system, the third is a post-SCF correction that is solely dependent on nuclear coordinates.

DCACP: When using DCACP, the electron density is updated self-consistently according to the vdW potential that the system experiences. See the description of the keywords **DCACP Z=z**, **NO_CONTRIBUTION** and **INCLUDE_METALS** for customisation options.

VDW CORRECTION: Two major types of correction types are implemented in connection with the **VDW CORRECTION** keyword: The one described by M. Elstner *et al.*[30] requires parameter sets designed to use only in conjunction with a specific corresponding density functional, the alternate parametrization by S. Grimme[38] is less specific and therefore easier to use and independent of the chose functional. Parameters for elements up to Xe have been directly coded into CPMD. See the description of the keyword **VDW PARAMETERS** for more details regarding custom input.

Part III
Miscellaneous

10 Postprocessing

The given output from a calculation with the CPMD code can be used for postprocessing. There are several types of output. The most typical types of output are density-like files, trajectories and/or xyz-files. These can be visualized or analyzed with a number of different programs. Some of them are (in no specific order):
Molden: (homepage: http://www.cmbi.kun.nl/ schaft/molden/molden.html
gOpenMol: (homepage: http://www.csc.fi/gopenmol/)
Molekel: (homepage: http://www.cscs.ch/molekel/)
VMD: (homepage: http://www.ks.uiuc.edu/Research/vmd/)
Starting with version 1.8.2 VMD does fully support the CPMD trajectory format, xyz-movie format and Gaussian Cube files. Since version 1.8.3 VMD supports periodic display of non-orthogonal supercells. A tutorial on Visualization and Analysis of CPMD data with VMD can be found at http://www.theochem.ruhr-uni-bochum.de/go/cpmd-vmd.html.

10.1 Density files

10.1.1 List

DENSITY.x, ELF, LSD_ELF, SPINDEN.x, WANNIER_1.x ...

10.1.2 Postprocessing

These files are created in a binary format, they have to be transformed to a Gaussian cube-File format to be readable by visualization programs. The `cpmd2cube.x` to convert the output can be download at www.cpmd.org and is used in the following way:

```
cpmd2cube: Convert CPMD's Wannier-function files to cube
usage is: cpmd2cube [options] Wannier_file [Wannier_file...]
   If you specify more than one Wannier file, they MUST have the
   same g-vectors and (for the moment) atom positions
   The program will create one cube file for each Wannier file
   and one pdb file with the atom positions
Example:
   cpmd2cube.x WANNIER_1.*
possible options are
   -v <verbosity>:
      <verbosity> is 0-2 (default is 1)
   -double:
      Read the density in double precision (default is single)
   -halfmesh:
      leave out half the grid points in each direction.
      Reduces the file size by 1/8th (on by default).
   -fullmesh:
      use the full real space grid.
   -n <n1> <n2> <n3>:
      change the REAL-space mesh. Default is to take the same mesh as CPMD
   -o <prefix>:
      specify the prefix of the name used for the cube and pdb-files
   -rep <n1> <n2> <n3>:
      replicate the cell n<j> times along the <j>-th direction by periodicity
```

```
-shift <r1> <r2> <r3>:
   shift cube density by r1*a1+r2*a2+r3*a3
-centre:
-center:
   centre density around centre of mass of system.
-inbox:
   put atoms inside unit cell centred around origin
-rho:
-dens:
   store the density instead of the wavefunction into the cube file.
-psi:
-wave:
   store the wavefunction instead of the density into the cube file.
--:
   last option. Useful if you have a file with the same name as an option
-h  or  -?  or  -help  or  --help  or no files:
   write this help
```

10.2 xyz-files

10.2.1 List

GEOMETRY.xyz, ION+CENTERS.xyz

10.2.2 Postprocessing

These files can be directly read by a visualisation program. Note, that at the current status it is useful to include a reference point (WANNIER REFERENCE) for the ION+CENTERS.xyz which has to be put at the middle of the box.

10.3 TRAJECTORY-File

10.3.1 List

TRAJECTORY

10.3.2 Postprocessing

1. Looking at a movie
The TRAJECTORY files contains the coordinates and the velocities. To create a movie file in the xyz-format, you have to transfer the coordinates from Bohr to Å and you have to add the symbols of the atoms in the first position. Two lines have to be at the beginning of each time step, from which the first line gives the number of the total atoms. An .xyz file can also be recorded directly during the simulation Using the **TRAJECTORY** keyword with the option **XYZ**.
Please note, that CPMD does not apply the minimum image convention to these trajectory files, i.e. atoms are **not** replaced by their images if they leave the supercell.

2. Calculating radial pair distribution functions
The simplest analysis of the structure is given by the radial pair distribution function g(r). This quantity is a to unity normalized function and describes the probability of finding two atoms separated by a distance r relative to the probability expected for a completely random distribution

at the same density. It is formally defined as:

$$\begin{aligned} g(r) &= \rho^{-2} \left\langle \sum_i \sum_{i \neq j} \delta(\mathbf{r}_i)\delta(\mathbf{r}_j - \mathbf{r}) \right\rangle \\ &= \frac{V}{N^2} \left\langle \sum_i \sum_{i \neq j} \delta(\mathbf{r} - \mathbf{r}_{ij}) \right\rangle \end{aligned}$$

with r being the atomic separation, ρ the number density, N the number of atoms, V the volume, $\mathbf{r}$ the atomic position, and $\mathbf{r}_{ij}$ the position of the atom i relative to the atom j. The average <> is taken over particles and time. Examples of code can be found in the following references [2, 147].

10.4 The MOVIE format

Besides the TRAJECTORY file CPMD also produces a specially formatted trajectory in the MOVIE file. This file contains the position of all atoms of the system in Angstrom units at a rather low precision (10^{-4}). The sampling of the positions can be done independently from the trajectory file. The sets of coordinates are following each other contiguously. The format of each line is

x-coordinate, y-coordinate, z-coordinate, atomic number, type .

11 Hints and Tricks

11.1 Pseudopotentials and Plane Wave Cutoff

The selection of proper pseudopotentials and the corresponding plane wave cutoff are crucial for obtaining good results with a CPMD calculation. The cutoff required is mainly determined by the type and the "softness" of a pseudopotential.
Ideally a pseudopotential for a specific atom type should be usable for all kinds of calculations (Transferability), but in practice one frequently has to make compromises between accuracy and impact on the computational effort when creating a pseudopotential. Therefore one always has to test pseudopotentials before using them on new systems. There are quite a large number of CPMD calculations published (see http://www.cpmd.org/cpmd_publications.html) which can serve as a guideline.

Since CPMD uses a plane wave basis, a concept of several different, formalized basis sets of different quality, like with gaussian basis set based quantum chemical software, does not apply here. Since plane waves are 'delocalized' in space, they provide the same 'quality' everywhere in space and one can increase the basis set almost arbitrarily by increasing the number of plane waves via the **CUTOFF** keyword. The cutoff has to be chosen in such a way, that all required quantities are reasonably converged with respect to the plane wave cutoff. For a molecular dynamics run this refers primarily to the atomic forces. For the calculation of other properties like the stress tensor a different, usually a much higher cutoff is required. It's always a good idea to make checks at some critical points of the calculations by increasing the cutoff.

Typical cutoff values range from 20–40 ry for Vanderbilt ultra-soft pseudopotentials, 60–100 ry for Troullier-Martins norm-conserving pseudopotentials to 80–200 ry for Goedecker pseudopotentials. Pseudopotentials of different types can be freely mixed, but the required plane wave cutoff is determined by the "hardest" pseudopotential. Support for Vanderbilt ultra-soft pseudopotentials is (mostly) limited to the basic functionality like molecular dynamics and geometry optimization.

11.2 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 from random 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.

11.2.1 Using Vanderbilt Ultrasoft Pseudopotentials

When using Vanderbilt ultrasoft pseudopotentials (USPPs) [4] and starting from atomic pseudo-wavefunctions, the calculations often do not converge or converge to a wrong state, especially if 3d-elements are involved. Convergence is generally much better when assigning (partial) charges via the **ATOMIC CHARGES** keyword in the &SYSTEM ...&END section. Values from a classical MD forcefield or an NBO calculation are usually good values. Alternatively a random initialization of the wave functions (via **INITIALIZE WAVEFUNCTION** RANDOM) can be used.
Also, due to the comparatively small plane wave cutoffs, you will have small but significant modulations of the density in especially in regions with little electron density. These lead to "strange" effects with gradient corrected functionals, causing the optimization to fail. To avoid this, you can skip the calculation of the gradient correction for low electron density areas using **GC-CUTOFF** with a value between 1.D-6 and 1.D-5 in the &DFT section.

In case of geometry optimizations, also the accurate calculation of the forces due to the augmentation charges may need a higher density cutoff and/or a tighter real space grid. This can be achieved by either using a higher plane wave cutoff or via increasing **DUAL** to 5.0 or even 6.0 and/or setting the real space grid explicitly via the **MESH** keyword in the &SYSTEM section. For the same reason, these options may be needed to increase energy conservation during molecular dynamics runs. Use these options with care, as they will increase the cpu time and memory requirements significantly and thus can easily take away one of the major advantages of ultra-soft pseudopotentials.

11.3 Wavefunction Convergence

Some general comments on wavefunction optimizations:

Any optimization that takes more than 100 steps should be considered slow.

Optimizations using ODIIS that have repeated resets (more than a few) will probably never converge. Convergence for LSD is normally slower and more difficult than for unpolarized cases.

If the ODIIS convergence gets stuck (more than one reset) stop the run and restart using

```
PCG MINIMIZE
TIMESTEP
 20
```

The conjugate gradient minimizer with line search is much more robust. For LSD and larger systems it should be used from the start.

A typical behavior will be that after the restart the energy goes down and the gradient increases. This means that we are in a region where there are negative curvatures. In such regions the DIIS minimizer moves in the wrong direction. After some iterations we will be back to normal behavior, energy and gradient get smaller. At this point it may be save to switch back to ODIIS.

Sometimes, it can also be helpful to wait longer for a DIIS reset and to diagonalize after repeated resets to get out of this region. This can be accomplished using

```
ODIIS NO_RESET=20
  5
LANCZOS DIAGONALIZATION RESET=2
```

Starting a Car-Parrinello MD from a random wavefunction with all atom positions fixed, a comparatively high electron mass and using **ANNEALING** ELECTRONS is another alternative to get to a reasonably converged wavefunction. Due to the exponential convergence of the annealing procedure, one should switch to a different optimizer as soon as the fictitious kinetic energy of the electrons drops below the typical range for a (normal) MD run.

Wavefunction optimizations for geometries that are far from equilibrium are often difficult. If you are not really interested in this geometry (e.g. at the beginning of a geometry optimization or this is just the start of a MD) you can relax the convergence criteria to 10^{-3} or 10^{-4} and do some geometry steps. After that optimization will be easier.

Some general remarks on comparing the final energies:

Converge the wavefunction very well, i.e. set **CONVERGENCE** ORBITALS to 10^{-6} or better.

Make sure that all parameters are the same:
- same geometry,
- same functional,
- same number of grid points (this may differ if you use different FFT libraries)
- same number of spline points for the PP
(IMPORTANT: the default for **SPLINE POINTS** has changed between different CPMD versions,

500 → 3000 → 5000). A very good test is to start always from the same RESTART file and only do one single step. This way ALL energies have to be exactly the same and no problems with different convergence rates occur.

11.4 Cell Size for Calculations with SYMMETRY 0

Calculations of isolated systems (i.e. decoupling of the electrostatic images in the Poisson solver) are initialized with:

```
SYMMETRY
   0
```

The box is assumed to be orthorhombic. With the additional options **SURFACE** or **POLYMER** periodic boundary conditions in two or one dimensions, respectively, are assumed.
Three different kinds of **POISSON SOLVER** {HOCKNEY,TUCKERMAN,MORTENSEN} are available. All methods require that the charge density is zero at the border of the box. For normal systems and the Hockney solver, this means that about 3 Angstrom space between the outermost atoms and the box should be enough. For large molecules, Tuckerman will require a considerably higher margin, see below. However, for some systems and for high accuracy these rules of thumb may not be enough. Some methods have additional requirements (see below).
Note that the **ISOLATED MOLECULE** keyword has only an effect on the calculation of the degrees of freedom (3N-6 vs. 3N-3 for periodic systems). The main purpose of **CENTER MOLECULE** ON/OFF is to center the molecule (center of mass) in the box. This is needed for the HOCKNEY Poisson solver. This solver gives wrong results if the charge density is not centered in the computational box. All other solvers behave like the periodic counterpart, i.e. the relative position of the charge density and the box are not important.
Further requirements on the Poisson solvers:

HOCKNEY Method:

- molecule has to be in the center of the box
- box size molecule + 3 Å border
- expensive for very small systems
- not available for some response calculations
- **POLYMER** is available but gives (currently, Version 3.9) wrong results.
- **SURFACE** is available and works.

TUCKERMAN-MARTYNA Method:

- box size : molecule + 3 Å border **and** - imperatively! - *twice* the size of the electron charge distribution (check the density after a single-point wavefunction optimization).
- expensive for large systems, smaller boxes might be used without loosing too much accuracy
- **SURFACE** or **POLYMER** are not available

MORTENSEN Method:

- same as TUCKERMAN, but using analytic formula made possible by using special boundary conditions (sphere, rod)
- **SURFACE** and **POLYMER** are available and should be safe to use (MORTENSEN is default for SURFACE and POLYMER)
- If you do an isolated system calculation, your cell has to be cubic, if you use **POLYMER** cell dimensions b and c have to be equal.

Finally, for many systems using a large enough cell and periodic boundary conditions is also an option. In general, the computed properties of molecules should be independent of the scheme used (either pbc or isolated box) except in difficult cases such as charged molecules, where the calculation in an isolated box is recommended. The PBC calculation is always cheaper for a box of the same size, so for a neutral molecule such as water molecule you would save time and memory by not using **SYMMETRY** 0.

11.5 Geometry Optimization

Any combination of methods for geometry optimization and wavefunction optimization is allowed. Possible options for geometry optimization are GDIIS, LBFGS, PRFO, 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 convergence criteria for the wavefunction optimization can be adapted to the requirements of the geometry optimization (CONVERGENCE ADAPT and CONVERGENCE ENERGY). The default options are GDIIS and ODIIS. Some 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**.
The linear-scaling geometry optimizers (options LBFGS and PRFO) do not require an approximate Hessian. To achieve linear scaling with the system size, the L-BFGS optimizer starts from a unit Hessian and applies the BFGS update on the fly using the history of the optimization. The P-RFO method can find transition states by following eigenmodes of the Hessian. The mode to be followed does not necessarily have to be the lowest eigenvalue initially (**PRFO MODE**). For larger systems, only the reaction core should be handled by the P-RFO optimizer (**PRFO NVAR** and **PRFO CORE**), and the environment is variationally decoupled using the L-BFGS optimizer. The normal LBFGS options can be used for the environment. The Hessian used for transition-state search therefore spans only a subset of all degrees of freedom, is separate from the Hessian for the other optimizers and the vibrational analysis, but it can be transferred into the appropriate degrees of freedom of the regular Hessian (**HESSIAN PARTIAL**). In order to allow negative eigenvalues, the Powell update instead of BFGS is used for transition-state search.
Although tailored for large systems, the linear-scaling geometry optimizers are suitable for smaller systems as well.

11.6 Molecular Dynamics

11.6.1 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.[99]). 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 that 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 Δt 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.

11.7 Restarts

11.7.1 General information

All restart information for CPMD simulations are stored within one binary file. There are very few exceptions we will discuss later. The name of the restart files is `RESTART` or `RESTART.n`, where n stands for an integer number. If the keyword **RESTART** is found the program processes the file with the name `RESTART`. Using suboptions to the **RESTART** option, the information retained from the file can be specified. For example the suboptions `COORDINATES WAVEFUNCTION` will force the program to use the geometry and orbitals from the `RESTART` file.
At the end of a simulation or at regular intervals (using the keyword **STORE**) a restart file with the default name `RESTART.1` is written. If this happens more than once (e.g. during a molecular dynamics run) the restart file is being overwritten. Using the keyword **RESTFILE** it can be specified that more than one restart file should be used for writing. If the **RESTFILE** parameter was set to 4, then 4 restart files with the names `RESTART.1`, ..., `RESTART.4` will be written. If more than 4 restart files are needed the first one will be overwritten. This option is useful if there is the possibility that restart files get corrupted (e.g. on instable systems), or if simulations are performed that might lead to unphysical results. In this case it might be possible to go back to a restart file which contains still intact information.
The name of the last restart file written is stored in the file `LATEST`. Using the suboption LATEST to the keyword RESTART changes the default name of the file to be read from RESTART to the name found in the file `LATEST`. The danger of using this option is that the file from which the simulation is started gets overwritten during the simulation. Using the default (starting from RESTART) ensures that the original file stays intact. However, it requires the renaming of the final file of a simulation from `RESTART.n` to `RESTART`.

11.7.2 Typical restart scenarios

Wavefunction optimizations The restart options used in wavefunction optimizations are **RESTART WAVEFUNCTION COORDINATES**. The suboption COORDINATES is not really necessary but it is advised to use it anyway, as in this way the correspondence of wavefunction and ionic geometry is assured.

Geometry optimizations Typical suboptions used in a geometry optimizations are **RESTART WAVEFUNCTION COORDINATES HESSIAN**. With the suboption HESSIAN the information from previous runs stored in the updated approximate HESSIAN can be reused.

Molecular dynamics Molecular dynamics simulations use restart options of the kind **RESTART WAVEFUNCTION COORDINATES VELOCITIES**. These are the minimal options needed for a smooth continuation of a Car–Parrinello molecular dynamics simulation. Use of the suboption ACCUMULATORS ensures that the calculated means (e.g. temperature) are correct for the whole simulation, not just the current run. If Nosé thermostats are used it is important also the restart the thermostat variables. This is achieved by adding the corresponding keywords to the RESTART (NOSEE, NOSEP, NOSEC).

Kohn–Sham energies The calculation of canonical Kohn–Sham orbitals requires a restart. In general, this will be a restart from converged orbitals from a wavefunction optimization. There is no way that the program can check this. However, if the same convergence criteria are used, the number of occupied states orbitals should converge in the first iteration of the diagonalization.

11.7.3 Some special cases

The suboption VELOCITIES will result in a restart from both, ionic and wavefunction velocities. In special cases, this is not the desired behavior. Using the additional keyword **QUENCH** the read velocities can be set back to zero. This will be most likely used for wavefunctions with QUENCH ELECTRONS. Another possibility is to reoptimize the wavefunction at the start of a molecular dynamics simulation. This is achieved with the keywords QUENCH BO.
For performance reasons the writing of the restart file should be done only occasionally. This might cause problems if the simulation was terminated incorrectly. Several hundreds or thousands of simulation steps might be lost. For this reason CPMD writes a special output file `GEOMETRY` after each molecular dynamics step. Together with a normal restart file this allows to start the simulation form the last ionic configuration and velocities. To achieve this another suboption GEOFILE has to be added to the RESTART keyword. After reading the positions and velocities of the ions from the restart file, they are also read from the GEOMETRY file and overwritten.
Special restarts to be used with the keywords **TDDFT** and **VIBRATIONAL ANALYSIS** are discussed in the sections covering that type of simulations.

11.8 TDDFT

The TDDFT part of CPMD is rather new. Therefore it hasn't yet reached the stability of other parts of the code. It has to be used with special care.
There are four different type of calculations that can be performed using the TDDFT module; calculation of the electronic spectra, geometry optimization and vibrational analysis, and molecular dynamics in excited states.
All options (spectra and forces, spin polarized and unpolarized) are implemented for the Tamm–Dancoff approximation to TDDFT. Only part of these options are available for the full TDDFT response calculation.

11.8.1 Electronic spectra

Electronic excitation energies can be calculated using the keyword **ELECTRONIC SPECTRA** in the &CPMD section. This calculation is performed in three parts. First, the ground state wavefunctions are optimized, then a limited set of unoccupied orbitals is determined and finally the TDDFT response equations are solved. A typical input for such a calculation would look like

```
&CPMD
  ELECTRONIC SPECTRA
  DIAGONALIZATION LANCZOS
  COMPRESS WRITE32
&END

&TDDFT
  STATES SINGLET
    5
  TAMM-DANCOFF
  DAVIDSON PARAMETER
  150 1.D-7 50
&END
```

For this calculation of the electronic spectra defaults are used for the ground state optimization (ODIIS and 10^{-5} convergence). The calculation of the empty states is performed using the Lanczos diagonalizer with default settings. The final wavefunction will be stored in the restart file using 32 bit precision.
Five single states with the Tamm–Dancoff approximation have to be calculated. The parameters for the Davidson diagonalization have been changed to 50 for the Davidson subspace and a convergence criteria of 10^{-7} is used.
Restarting this type of calculation has to be done with care. At the end of each phase of the calculation a new restart file is written. If the defaults are used, each time the file `RESTART.1` is overwritten. For a restart from converged ground state wavefunctions and canonical Kohn–Sham orbitals a restart with
`RESTART WAVEFUNCTION COORDINATES`
will be used. A restart also including the linear response orbitals will use
`RESTART WAVEFUNCTION COORDINATES LINRES`.
In this case only restarts from the file `RESTART` are possible as after phase one and two the file `RESTART.1` would be overwritten and the information on the linear response orbitals, read only in phase three, would be lost.

11.8.2 Geometry optimizations and molecular dynamics

Geometry optimizations and molecular dynamics simulations can only be performed after an electronic spectra calculation. A typical input file would contain the sections

```
 &CPMD
   OPTIMIZE GEOMETRY
   TDDFT
   RESTART WAVEFUNCTION COORDINATES LINRES
 &END

 &TDDFT
  STATES SINGLET
    1
  TAMM-DANCOFF
  DAVIDSON PARAMETER
  150 1.D-7 50
```

```
  FORCE STATE
    1
&END
```

The keywords in section &CPMD are all mandatory. The section &TDDFT specifies that the optimization should be performed for the first excited singlet state. Replacing **OPTIMIZE GEOMETRY** by **MOLECULAR DYNAMICS** BO would result in a molecular dynamics simulation. In this case further input specifying the time step, maximal number of steps, thermostats, etc. would also be supplied.

11.9 Perturbation Theory / Linear Response

11.9.1 General

Ref: [22].
Perturbation theory describes the reaction of a system onto an external perturbation. At the time when the perturbation occurs, the system is in its ground state (or unperturbed state). The perturbation then changes slightly the potential energy surface and therefore also the state where the system's energy is minimum. As a consequence, the system tries to move towards that state of minimum energy. This movement or the new state often have properties which can be accessed experimentally. Example: An external electric field will slightly deform the electronic cloud, creating a dipole. That dipole can then be measured.
Assume that the magnitude of the perturbation is small compared to the strength of the forces acting in the unperturbed system. Then, the change in the minimum energy state will be small as well and perturbation theory can be applied to compute how the system reacts onto the perturbation. Generally, the Schrödinger equation is expanded in powers of the perturbation parameter (ex: the strength of the electric field), and the equations obtained for those powers are solved individually. At power zero, one finds the equation of the unperturbed system:

$$(H^{(0)} - \varepsilon_k)|\varphi_k^{(0)}\rangle \quad = \quad 0. \tag{5}$$

For the part which is linear in the perturbation, the general format of the resulting equation is

$$(H^{(0)} - \varepsilon_k)|\varphi_k^{(1)}\rangle \quad = \quad -H^{(1)}|\varphi_k^{(0)}\rangle. \tag{6}$$

Grosso modo, this equation is solved during a linear response calculation through a wavefunction optimization process for $|\varphi_k^{(1)}\rangle$.
The presence of a first order perturbation correction for the wavefunctions, $|\varphi_k^{(\mathsf{tot})}\rangle = |\varphi_k^{(0)}\rangle + |\varphi_k^{(1)}\rangle$ implies that the total density of the perturbed system is no longer equal to the unperturbed one, $n^{(0)}$, but also contains a first order perturbation correction, $n^{(1)}$. That density is given by

$$n^{(1)}(\mathbf{r}) \quad = \quad \sum_k \langle\varphi_k^{(1)}|\mathbf{r}\rangle\ \langle\mathbf{r}|\varphi_k^{(0)}\rangle + \text{c.c.} \tag{7}$$

The Hamiltonian depends on the electronic density. Therefore, the first order density correction implies automatically an additional indirect perturbation Hamiltonian coming from the expansion of the unperturbed Hamiltonian in the density. It has to be added to the explicit perturbation Hamiltonian determined by the type of the (external) perturbation. The contribution is

$$H^{(1)}_{\mathsf{indirect}}(\mathbf{r}) \quad = \quad \int d^3r'\ \frac{\partial H^{(0)}(\mathbf{r})}{\partial n^{(0)}(\mathbf{r}')}\ n^{(1)}(\mathbf{r}') \tag{8}$$

The calculation of this indirect Hamiltonian represents almost 50% of the computational cost of the response calculation, especially in connection with xc-functionals. After several unsuccessful trials with analytic expressions for the derivative of the xc-potential with respect to the density, this is done numerically. That means that at each step, the xc-potential is calculated for the density $n^{(0)} + \epsilon n^{(1)}$ and for $n^{(0)} - \epsilon n^{(1)}$ (with an ϵ empirically set to 0.005), and the derivative needed in (8) is calculated as

$$\int d^3r' \, n^{(1)}(\mathbf{r}') \, \frac{\partial v_{\mathrm{xc}}}{\partial n^{(0)}(\mathbf{r}')} \quad = \quad \frac{v_{\mathrm{xc}}[n^{(0)} + \epsilon n^{(1)}] - v_{\mathrm{xc}}[n^{(0)} - \epsilon n^{(1)}]}{2\epsilon}. \tag{9}$$

In the case of the local density approximation, the derivative can be done analytically, in which case it only needs to be done once. This improves the performance of the optimization.

11.9.2 &RESP section input

Generally, the keyword **LINEAR RESPONSE** in the &CPMD input section initiates the calculation. In the section &RESP, the type of the perturbation needs to be specified. Either one of the following keywords must appear:

PHONON
LANCZOS
RAMAN
FUKUI
KPERT
NMR
EPR
HARDNESS
EIGENSYSTEM
INTERACTION
OACP

The first six types are discussed in detail in the following. An overview is also contained in the file respin_p_utils.mod.F90. In addition to the specific keywords of every option, there are several keywords which are common to all perturbation types. They determine fine-tuning parameters of the wavefunction optimization process, and usually you do not need to change them. A # indicates that a command takes an argument which is read from the next line. All other keywords toggle between two states and do not require any argument. Those keywords can be put together, and they can also figure in the main keyword line (example: **NMR NOOPT FAST WANNIERCENTERS**)
NB: The linear response code works with all cell symmetries[1], but it is ***not implemented*** for k-points.

\# **CG-ANALYTIC:** The wavefunction optimization uses a preconditioned conjugate gradient technique. The optimum length of the "time step" can be calculated analytically assuming a purely linear equation, according to the Numerical Recipes Eq. 10.6.4. However, this is somewhat expensive, and experience shows that the time step is almost constant except at the very beginning. Therefore, it is only calculated a few times, and later on, the last calculated value is used. This option controls the number of times the step length is calculated analytically. Default is 3 for NMR and 99 for all other perturbations.

\# **CG-FACTOR:** The analytic formula for the time step assumes that the equation to be solved is purely linear. However, this is not the case, since the right hand side can still depend on the first order wavefunctions through the dependence of the perturbation Hamiltonian $H^{(1)}$ on the perturbation density $n^{(1)}$. Therefore, the analytic formula has a tendency to overshoot. This is corrected by an empirical prefactor which is controlled by this option. Default is 0.7.

[1] except for isolated systems, **SYMMETRY=0**, where only the NMR part is adapted to.

CONVERGENCE: The criterion which determines when convergence is reached is that the maximum element of the gradient of the energy with respect to the wavefunction coefficients be below a certain threshold. This value is read from the next line. Default is 0.00001. Experience shows that often, it is more crucial to use a strict convergence criterion on the ground state wavefunctions than for the response. A rule of thumb is that good results are obtained with a 10 times stricter convergence on the ground state orbitals compared to that of the response orbitals.

HTHRS or HAMILTONIAN CUTOFF: The preconditioning calculates the diagonal $(\mathbf{G},\mathbf{G})$ matrix elements of $\eta = H^{(0)} - \frac{1}{N}\sum_k \varepsilon_k$ to do an approximate inversion of Eq. (6). However, these diagonal values can become very small, yielding numerical instabilities. Therefore, a smoothing is applied instead of simply taking the reciprocal values:

$$\eta^{-1} \mapsto \left(\eta^2 + \delta^2\right)^{-1/2} \tag{10}$$

$$\eta^{-1} \mapsto \frac{\eta}{\eta^2 + \delta^2} \tag{11}$$

The value of the parameter δ in a.u. is read from the line after **HTHRS**, default is 0.5. By default, Eq. (10) is used. **TIGHTPREC** switches to Eq. (11).

- **NOOPT:** In order for the wavefunction optimization to work properly, the ground state wavefunction must be converged. For this reason, a ground state optimization is performed by default prior to computing the response. When restarting from an already converged wavefunction, this step can be skipped through this keyword and the computer time for initializing the ground state optimization routine is saved. However, the use of this option is strongly discouraged.

- **POLAK:** There are several variants of the conjugate gradient algorithm. This keyword switches to the Polak-Ribiere formulation (see the Numerical Recipes, Eq. 10.6.7) which is usually significantly slower but safer in the convergence. By default, the Fletcher-Reeves formula is used.

- **TIGHTPREC:** Switches to another preconditioning formula. See **HTHRS**.

11.9.3 Response output

While the calculations are being done, the program prints the progress of the optimization process:

- The "scaling input" prints the number by which the right-hand-side of Eq. (6) is multiplied in order to deal with reasonable numbers during the optimization. The number is determined through the condition

$$||\ H^{(1)}|\varphi_k^{(0)}\rangle\ ||_2 \quad = \quad 1. \tag{12}$$

  When leaving the optimization routine, the inverse scaling is applied to the $|\varphi_k^{(1)}\rangle$, of course.

- The standard output shows (A) the maximum gradient of the second order energy with respect to the first order perturbation wavefunction, (B) the norm of that gradient, (C) the second order energy, (D) its difference with respect to the last value, and finally (E) the CPU time needed for one step. The last value decreases by somewhat after the number of steps determined by the **CG-ANALYTIC** keyword, because the analytic line search is no longer performed, as discussed above.

- A line full of tildes (˜) indicates that the energy has increased. In that case, the conjugate direction is erased and the conjugate gradient routine is restarted. Also the time step is calculated again using the analytic quadratic line search formula.

- The warning "line search instable" indicates that the length which has been calculated using the analytic quadratic line search approximation (Numerical Recipes Eq. 10.6.4) has given a numerical values larger than 3. This does not happen in normal cases, and it can yield to program crashes due to floating point overflows ($\mapsto$ `NaN`, not a number). Thus, a safe value, 1, is used instead.
- The warning "gradient norm increased by more than 200%" indicates that the quadratic approximation is not valid in the momentary position of the optimization. If it was, the gradient (of the energy with respect to the first order perturbation wavefunctions) would only decrease and finally reach zero. If this situation occurs, the conjugate direction is erased and the conjugate gradient algorithm is restarted.

11.9.4 Phonons

Theory

A phonon corresponds to small displacements of the ionic positions with respect to their equilibrium positions. The electrons principally follow them, in order to minimize again the energy of the system.
The expansion of the Hamiltonian in powers of the displacement u_α^R of the ion (labeled by its position R) in the Cartesian direction $\alpha = 1, 2, 3$ consists of two parts[2]:

$$H^{(1)} = H_C^{(1)} + H_{PP}^{(1)} \tag{13}$$

The contribution $H_C^{(1)}$ comes from the Coulomb term, the electrostatic potential:

$$H_C^{(1)} = u_\alpha^R \frac{\partial}{\partial R_\alpha} \frac{Z_R}{|\mathbf{r} - \mathbf{R}|}. \tag{14}$$

The second is due to the pseudopotential which is rigidly attached to the ions and which must be moved simultaneously. In particular, the nonlocal pseudopotential projectors must be taken into account as well:

$$H_{PP}^{(1)} = u_\alpha^R \frac{\partial}{\partial R_\alpha} \left[\sum_i |\mathtt{P}_i^R\rangle\langle\mathtt{P}_i^R| \right] \tag{15}$$

$$= u_\alpha^R \sum_i \left[\left[\frac{\partial}{\partial R_\alpha} |\mathtt{P}_i^R\rangle \right] \langle\mathtt{P}_i^R| + |\mathtt{P}_i^R\rangle \left[\frac{\partial}{\partial R_\alpha} \langle\mathtt{P}_i^R| \right] \right] \tag{16}$$

where $|\mathtt{P}_i^R\rangle$ designates the projectors, whatever type they are. The index i comprises the l and m quantum numbers, for example. The superscript R just says that of course only the projectors of the pseudopotential of the displaced atom at R are considered in this equation.
In CPMD 4.3.0, these projectors are stored in G-space, and only one copy is stored (that is the one for a fictitious ion at the origin, $R = 0$). The projectors for an ion at its true coordinates is then obtained as

$$\langle\mathbf{G}|\mathtt{P}_i^R\rangle = \mathrm{e}^{i\mathbf{G}\cdot\mathbf{R}} \langle\mathbf{G}|\mathtt{P}_i^{R=0}\rangle. \tag{17}$$

This makes the derivative $\frac{\partial}{\partial R_\alpha}$ particularly simple, as only the $i\mathbf{G}$ comes down, and the translation formula (17) remains valid for $|\mathtt{P}_i^R\rangle$. Thus, there is only an additional nonlocal term appearing which can be treated almost in the same way as the unperturbed pseudopotential projectors.

[2]The reader might wonder whether Einstein summations over repeated indices like α and R are done or not. The answer is that *it depends*. If only one atom is displaced along one of the axes, there is no summation needed. The generalization to a simultaneous displacement of all atoms in arbitrary directions is trivially done by assuming the summation over α and R.

A perturbative displacement in Cartesian coordinates can have components of trivial eigenmodes, that is translations and rotations. They can be written *a priori* (in mass weighted coordinates in this case) and thus projected out from the Hessian matrix;

$$\mathbf{t}_j = \begin{pmatrix} \sqrt{m_1}(\mathbf{e}_j) \\ \sqrt{m_2}(\mathbf{e}_j) \\ \vdots \\ \sqrt{m_N}(\mathbf{e}_j) \end{pmatrix} \qquad \mathbf{s}_j = \begin{pmatrix} \sqrt{m_1}(\mathbf{e}_j \times \mathbf{R}_1) \\ \sqrt{m_2}(\mathbf{e}_j \times \mathbf{R}_2) \\ \vdots \\ \sqrt{m_N}(\mathbf{e}_j \times \mathbf{R}_N) \end{pmatrix} \tag{18}$$

where j=x,y,z, $\mathbf{e}_j$ = unit vectors in the Cartesian directions and $\mathbf{R}_k$ = positions of the atoms (referred to the COM). The rotations $\mathbf{s}_j$ constructed in this way are not orthogonal; before being used they are orthogonalized with respect to the translations and with each other.
The projection on the internal mode subspace is done this way: first one constructs a projector of the form,

$$\mathbf{P} = \sum_j (\mathbf{t}_j \cdot \mathbf{t}_j^T + \mathbf{s}_j \cdot \mathbf{s}_j^T) \tag{19}$$

and then applies the projector to the Hessian matrix,

$$\mathbf{A} = (\mathbf{I} - \mathbf{P}) \cdot \mathbf{A} \cdot (\mathbf{I} - \mathbf{P}) \tag{20}$$

with **I** being the unit matrix. The projection is controlled by the keyword **DISCARD**, *vide infra.*
Phonon input
The input for the phonon section is particularly simple due to the absence of any special keywords. Only the word **PHONON** should appear in the &RESP section.
Phonon output
In total analogy to CPMD 4.3.0's **VIBRATIONAL ANALYSIS**, the displacement of all atoms in all Cartesian directions are performed successively. The difference is, of course, that there is no real displacement but a differential one, calculated in perturbation theory. Thus, only one run is necessary per ion/direction[3].
At the end, the harmonic frequencies are printed like in the **VIBRATIONAL ANALYSIS**. They should coincide to a few percent.

11.9.5 Lanczos

Theory
Ref: [148]
A different way of diagonalizing the Hessian matrix comes from the Lanczos procedure. It is easy to generalize eq.16 to a collective displacement of atoms,

$$H^{(1)}_{PP} = \sum_{R,\alpha} u^R_\alpha \frac{\partial}{\partial R_\alpha} \left[\sum_i |\mathrm{P}^R_i\rangle\langle \mathrm{P}^R_i| \right] \tag{21}$$

$$= \sum_{R,\alpha} u^R_\alpha \sum_i \left[\left[\frac{\partial}{\partial R_\alpha} |\mathrm{P}^R_i\rangle \right] \langle \mathrm{P}^R_i| + |\mathrm{P}^R_i\rangle \left[\frac{\partial}{\partial R_\alpha} \langle \mathrm{P}^R_i| \right] \right] \tag{22}$$

In this way the information contained in the Hessian matrix, **A**, can be used to compute terms of the type

$$\mathbf{A} \cdot \mathbf{w} = \begin{pmatrix} \frac{\partial^2 E}{\partial R_{1x} \partial \mathbf{w}} \\ \frac{\partial^2 E}{\partial R_{1y} \partial \mathbf{w}} \\ \vdots \\ \frac{\partial^2 E}{\partial R_{Nz} \partial \mathbf{w}} \end{pmatrix} \tag{23}$$

[3] In contrast to this, the **VIBRATIONAL ANALYSIS** displaces each atom in each direction first by +0.01 and then by -0.01.

where $\mathbf{w}$ is the collective displacement. This is the building block for Lanczos diagonalization, that is performed by iterative application of the symmetric matrix to be diagonalized, over a set of vectors, known as Lanczos vectors, according to the scheme

- $\mathbf{r}_0 = \mathbf{q}_1; \beta_0 = 1; \mathbf{q}_0 = 0; k = 0$

WHILE $\beta_k \neq 0$

- $k = k + 1$
- $\alpha_k = \mathbf{q}_k^T \mathbf{A} \mathbf{q}_k$
- $\mathbf{r}_k = \mathbf{A}\mathbf{q}_k - \alpha_k \mathbf{q}_k - \beta_{k-1}\mathbf{q}_{k-1}$
- $\beta_k = \parallel \mathbf{r}_k \parallel_2$
- $\mathbf{q}_{k+1} = \mathbf{r}_k / \beta_k$
- Orthogonalization, $\mathbf{q}_{k+1} \perp \{\mathbf{q}_1, \cdots, \mathbf{q}_k\}$
- Diagonalization of $\mathbf{T}_k$

END WHILE

The matrix $\mathbf{A}$ is thus projected onto a k×k subspace, in a tridiagonal form, $\mathbf{T}_k$. $\mathbf{T}$'s eigenvalues are approximations to $\mathbf{A}$'s, while the eigenvectors are brought in the n×n space by means of the orthogonal matrix $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_k]$. The advantage with respect to the usual diagonalization schemes is that the highest and the lowest end of the spectrum tend to converge before the ionic degrees of freedom are fully explored.
The projection of the trivial eigenmodes is performed by eliminating their contribution from every Lanczos vector at the orthogonalization step reported in the algorithm above,

$$\mathbf{q}_k = \mathbf{q}_k - \sum_j \left((\mathbf{q}_k^T \cdot \mathbf{t}_j)\mathbf{t}_j + (\mathbf{q}_k^T \cdot \mathbf{s}_j)\mathbf{s}_j \right) \tag{24}$$

This procedure seems slightly different from that of the **PHONON** case, but they are perfectly equivalent. It is controlled by the same keyword **DISCARD** with the same arguments.
Lanczos input
The input for the Lanczos routine is given by the keyword **LANCZOS** plus some optional arguments and a mandatory following line with numerical data. Please note that this keyword is analogous to that for the Lanczos diagonalization of the electronic degrees of freedom. Given its presence in the &RESP section there should be no ambiguity.

- **LANCZOS [CONTINUE,DETAILS];** the keyword **LANCZOS** simply activates the diagonalization procedure. At every cycle the program writes a file, `LANCZOS_CONTINUE.1` which contains the information about the dimension of the calculation, the number of iteration, the elements of the Lanczos vectors and the diagonal and subdiagonal elements of the **T** matrix.
  - With the option **CONTINUE** one restart a previous job, asking the program to read the needed information from the file `LANCZOS_CONTINUE`; if the program does not find this file, it stops. This option is to be used when the calculation that one wants to perform does not fit in the time of a queue slot.
  - The argument **DETAILS** prints a lot of information about the procedure at the end of the output. It is intended for debugging purposes.

The subsequent line is mandatory, and contains three numerical parameters;

- `Lanczos_dimension`; it is the dimension of the vibrational degrees of freedom to be explored. Normally it is $3\times N_{at}$, -3 if you are eliminating the translations or -6 if you are eliminating also the rotations(*vide infra*). It is possible to use lower values. Higher are non sense.
- `no. of iterations`; it is the number of cycles that are to be performed during the actual calculation. It must be $\leq$ to `Lanczos_dimension`; the program checks and stops if it is higher. In case of a continued job, the program checks if the given number of iterations + the index of the iteration from which it restarts is within the limit of `Lanczos_dimension`; if not it resets the number to fit the dimension, printing a warning on std. output.
- `conv_threshold`; it is the threshold under which an eigenvector is to be considered as converged. It is the component of the eigenvector over the last Lanczos vector. The lower it is, the lower is the information about this mode which has still to be explored.

- **DISCARD {PARTIAL,TOTAL,OFF,LINEAR};** this keyword controls the elimination of the trivial eigenmodes (*i.e.* translations and rotations) from the eigenvector calculations. It works both for **PHONON** and **LANCZOS**. Omitting it is equivalent to **DISCARD PARTIAL**. When using it, the choice of one of the arguments is mandatory; the program stops otherwise. They are;
  - **PARTIAL**; only the translational degrees of freedom are eliminated (useful for crystals). This is the default.
  - **TOTAL**; both translational and rotational degrees of freedom are eliminated.
  - **OFF**; the option is disactivated and no projection is performed.

Lanczos output In the output, all the informations about the perturbed wavefunction optimization are reported, just like in the **PHONON** case. The differences are in the form of the perturbation and in the eigenmodes information coming from the diagonalization of $\mathbf{T}_k$.
The report of the perturbation for a water dimer reads like;

```
**********************   perturbations    *********************

**** atom=    1     O    .04171821      .07086763      .07833475
**** atom=    2     O    .03615521      .08499767      .07082773
**** atom=    3     H    .29222111      .13357862      .09032954
**** atom=    4     H    .33764012      .15750912      .25491923
**** atom=    5     H    .06426954      .26020430      .01822161
**** atom=    6     H    .15765937      .27370013      .29183999
cpu time for wavefunction initialization:          89.22 seconds
```

where at every atom corresponds the x,y,z displacements applied to calculate the perturbed wavefunctions.
At every iteration information about the elements of the $\mathbf{T}_k$ matrix, alpha's and beta's, are reported. Here we are at the end of the calculation. Note the numerical zero in the final +1 beta value. Then the spectrum in a.u. is reported, together with the convergence information. At the last iteration there are vectors which are not "converged". But this comes only from the definition, since some of the eigenvectors *must* have a component over the last Lanczos vectors. Following there are the familiar eigenvalues in cm^{-1}.

```
****************************************************************
 *+* L2 norm[n[i+1]]       =  0.100262974102936016E-02
 *=*   overlap: alpha[ 12 ] =  0.982356847531824437E-03
 *=*   off-diag: beta[ 13 ] =  0.300011777832112837E-19
 *=*   norm:               =  0.300011777832112837E-19
****************************************************************
*** SPECTRUM, run 12 :
*** eigenvalue  12 =     .4855525       (converged: .000000).
*** eigenvalue  11 =     .4785309       (converged: .000000).
*** eigenvalue  10 =     .4605248       (converged: .000000).
*** eigenvalue   9 =     .4303958       (converged: .000000).
```

```
*** eigenvalue   8 =     .0973733       (converged: .000000).
*** eigenvalue   7 =     .0943757       (converged: .000000).
*** eigenvalue   6 =     .0141633       (converged: .000004).
*** eigenvalue   5 =     .0046807   (NOT converged: .004079).
*** eigenvalue   4 =     .0020023   (NOT converged: .010100).
*** eigenvalue   3 =     .0010310   (NOT converged: .313417).
*** eigenvalue   2 =     .0009886   (NOT converged: .933851).
*** eigenvalue   1 =     .0006303   (NOT converged: .171971).

*****************************************************************
 harmonic frequencies [cm**-1]:

       129.0591         161.6295         165.0552         230.0241
       351.6915         611.7668        1579.1898        1604.0732
      3372.3938        3488.4362        3555.9794        3581.9733

*****************************************************************
```

11.9.6 Raman

Ref: [23]
Upon the specification of the RAMAN keyword, the polarizabilities of the system are calculated by evaluating the response to an applied electrical field. The calculation is done by means of the Berry phase approach, which is also suited for periodic systems.

11.9.7 Nuclear Magnetic Resonance

Theory
Ref: [24]
A magnetic field **B** is applied to the system, which reacts by induced electronic ring currents. These currents produce an additional magnetic field by themselves, which is not homogeneous in space. Therefore, the actual magnetic field at the ionic positions is different for all atoms in the cell. This field determines the resonance frequency of the nuclear spin, and this resonance can be measured with a very high accuracy.
The perturbation Hamiltonian is given by

$$H^{(1)} = \frac{1}{2}\frac{e}{m}\mathbf{p} \times \mathbf{r} \cdot \mathbf{B}. \tag{25}$$

The difficult part of this Hamiltonian lies in the position operator which is ill defined in a periodic system. To get around this, the wavefunctions are localized and for each localized orbital, Eq. (25) is applied individually assuming the orbital being isolated in space. Around each orbital, a *virtual cell* is placed such that the wavefunction vanishes at the borders of that virtual cell.
The perturbation and therefore also the response are purely imaginary, so that there is no first order response density. This simplifies the equations and speeds up convergence.
NMR input
The options which control the specific NMR features are discussed below. None of them requires an argument, they can all be put in the same line.

- **RESTART:** The run is restarted from a previous stop. The user has to take care that the required files RESTART.*xxx* (where *xxx* are NMR, p_x, p_y, p_z) exist.
- **CURRENT:** Three density files containing current density values are written to disk. Further, the nucleus-independent chemical shift fields are written to disk. All in cube format.
- **NOSMOOTH:** At the border of each virtual cell, the position operator is smoothened through an $\exp -r^2$. This option turns smoothing off. Can be useful if your cell is too small so that this smoothing would already occur in a region where the orbital density has not yet vanished.

- **NOVIRTUAL:** If this keyword is specified, no individual virtual cells are used at all. All orbitals will have the same virtual cell, equal to the standard unit cell.
- **PSI0:** With this option, the Wannier orbitals are plotted as cube files.
- **RHO0:** With this option, the orbital densities are plotted as cube files.

\# **OVERLAP:** Overlapping orbitals (overlap criterion read from next line) are assigned the same virtual cells. Useful for isolated systems.

- **FULL** A full calculation of the Δj term in the NMR scheme is done. Relatively expensive, but quite important for accurate results in periodic systems.

NMR output
At the end of the six perturbation runs, the results are printed. This includes the magnetic susceptibility and the chemical shieldings. For the chemical shieldings, there are two values: the raw and the net ones. The raw shieldings correspond to a molecule in vacuum, without the susceptibility correction, whereas the net shieldings contains that correction for a spherical sample. It consists of an additive constant to all eigenvalues, which is printed out at the end of the net shieldings.
In more detail, the results are:

- The magnetic susceptibility tensor in both SI and cgs units. This quantity is an extensive quantity, i.e. two molecules in the simulation box will give twice the susceptibility of one.
- The raw shielding matrices of all atoms and its eigenvalues. Generally, the shielding tensor is not symmetric. To obtain unique eigenvalues, it is symmetrized ($A_{ij} \mapsto (A_{ij} + A_{ji})/2$) before the diagonalization.

  All values are given in ppm, parts per million. Chemical shieldings are dimensionless quantities.
- The principal values (eigenvalues) of the raw and net shielding tensors. As mentioned above, they only differ by an additive constant, the susceptibility correction for a spherical sample, which is printed out at the end of the list. The numbers are the shielding eigenvalues from most negative to most positive, the isotropic shielding (the average of the eigenvalues), and the anisotropy (the difference between the most positive and the average of the other two).

What to look at? If you search for the values which are peaked in a spectrum, you have to take the isotropic shieldings (the **iso** column of the output). If the system is a gas, take the raw shielding, if it is condensed matter, take the net shielding. If your system is a molecule in vacuo, but the experimentalists measure it in a solvent, add the susceptibility correction to the raw shieldings by yourself.
Why are my numbers so strange in absolute value? One more point shall be mentioned: For all nuclei except hydrogen, pseudopotentials screen the core electrons. The chemical shielding, however, is very sensitive to core and semi-core electrons. This can be corrected through a semi-empirical additive constant in many cases. This constant still needs to be added to the values given from the program. It depends on the nucleus, on the pseudopotential, and on the xc-functional.
In other cases, the calculated chemical shieldings are completely meaningless due to this deficiency. Then, you have to use a pseudopotential which leaves out the semicore states such that they are correctly taken into account. Example: carbon shieldings can be corrected very well through a constant number, silicon shieldings cannot. For Si, you have to take the $n = 3$ shell completely into the valence band, requiring a cutoff larger than 300Ry.
How to compare to experiment? Usually, experimentalists measure the *difference* between the resonance frequency of the desired system and that of a reference system, and they call it δ (the **shift**) instead of σ (the **shielding**). To make life more complicated, they usually define the shift of nucleus A of molecule X with respect to reference molecule ref as $\delta^{\mathsf{A}}_{\mathsf{ref}}(\mathsf{X}) = \sigma^{\mathsf{A}}(\mathsf{ref}) - \sigma^{\mathsf{A}}(\mathsf{X})$. Example: To calculate $\delta^{\mathsf{H}}_{\mathsf{TMS}}(\mathsf{CH_4})$, where TMS=tetramethylsilane, the standard reference molecule

for H-shifts, one would have to calculate the H-shielding of TMS and of CH_4 and subtract them. Unfortunately, TMS is nontrivial to calculate, because it is a large molecule and the geometry is complicated (and the shieldings probably must be calculated taking into account vibrational and rotational averaging). Thus, in most cases it is better to take for instance the CH_4 shielding as a (computational) reference, and transform the shieldings relative to CH_4 to those relative to TMS through the experimental shielding of CH_4 with respect to TMS.
While doing so, you should not forget that the shielding is a property which is usually not yet fully converged when energies and bonding are. Therefore, the reference molecule should be calculated with the same computational parameters as the desired system (to reproduce the same convergence error). In particular, computational parameters include the type of the pseudopotential and its projectors, the xc-functional and the cutoff.
What accuracy can I expect? This is a difficult question, and there is no overall answer. First, one has to consider that on the DFT-pseudopotential level of theory, one will never reach the accuracy of the quantum chemistry community. However, for "normal" systems, the absolute accuracy is typically 0.5-1 ppm for hydrogen and about 5-20ppm for Carbon. The spread between extreme regions of the carbon spectrum is not reached: instead of 200ppm, one only reaches 150ppm between aliphatic and aromatic atoms, for instance. The anisotropy and the principal values of the shielding tensor can be expected to be about 10-20% too small. For hydrogen shieldings, these values are usually better, the error remains in the region of a single ppm.

11.9.8 FUKUI

Compute, within the linear response perturbation theory, the nuclear Fukui function ϕ_I [149] formally identified as a reactivity index of the density functional theory [150, 151], according to the postulated criterion for $\delta\mu$. The quantity $\delta\mu$ is the change in the electronic chemical potential μ and is given by

$$\delta\mu = \int f(\mathbf{r})\delta V^{ext}(\mathbf{r})d^3r = -\sum_I \phi_I \delta\mathbf{R}_I \tag{26}$$

where $\phi_I = (\partial\mathbf{F}_I/\partial N)_{V^{ext}} = -(\partial\mu/\partial R_I)_N$, N is the number of electrons, $f(\mathbf{r})$ the electronic Fukui function [149, 150], $V^{ext}(\mathbf{r})$ the external potential at $\mathbf{r}$, $\mathbf{R}_I$ the Cartesian coordinate of the I^{th} nucleus and $\mathbf{F}_I$ the force on the I^{th} nucleus.

11.9.9 KPERT: kdp k-point calculations

Described in section 9.4, page68. Ref: [25]

11.10 Metadynamics

These are some notes about the use of the metadynamics (MTD) machinery within CPMD. It is just a first version of a manual that I hope will be improved by the comment and possibly the contributions of the users of this method.
The metadynamics can run in a standard NVE/NVT MD run or in a NPE/NPT run (variable cell). In order to apply the MTD algorithms in CPMD some (not few) lines have to be added in the input file. These lines are to be in the &ATOMS .. &END section and they provide information about the kind of MTD to be performed, the choice of collective variables (CV), some parameters required to determine the time dependent potential and some other options. All the MTD input must be between an initial and a final line which are:

```
METADYNAMICS
...
END METADYNAMICS
```

If the initial line contains also the keyword *COLLECTIVE VARIABLES*, the standard MTD, with one single set of CV, is initialized. If, instead, the keyword *MULTI* is found, more than one MTD are performed simultaneously on the same system; therefore, the whole set of CV is constituted by *NSUBSYS* subsets, which are independent one from each other. The number of subsystems is given on the same line by writing $NS =$ followed by an integer number (default: 1). Alternatively, if $MULTIPLE\ WALKERS$ is present in the same line multiple walker metadynamics is preformed using the extended Lagrangian metadynamics; number of walkers is read in the same line immediately after $NW =$ (see 11.10.7). Instead, if $CELLFULL$ is the keyword, the CV are the 6 cell parameters (3 side lengths and 3 angles), and the MTD is performed without extended Lagrangian, i.e. the contribution coming from $V(t)$ is directly added into the stress tensor (see below in **MTD Algorithm**).

For almost all the input parameters there is a reasonable default value, but, since the range of applications of MTD is quite wide, it is likely that the default values do not fit your problem. Therefore some effort is required to choose the optimal conditions for your run. Of course, it is important to know something about MTD before using it. There are some references about the method [27, 28, 152, 153, 154], and about some successful applications, as e.g. [155, 156, 157, 158, 159, 160, 161, 162, 163]. It can be of great help to read about the choices and results obtained by other users. But I remark that there are very few general rules that can be held valid for different problems and systems.

The method is based on the definition of a manifold of CV as functions of the degrees of freedom characterizing your system, $\mathbf{S} = \{S_\alpha(\mathbf{R}, \phi, \mathbf{h})\}$, where $\mathbf{R}$ are the ionic degrees of freedom, ϕ are the electronic wavefunctions, and $\mathbf{h}$ defines the cell box. The CV which are implemented in the code, have been chosen according to the needs of those who used the method up to now. Of course they do not exhaust all the problems, and many more CV might be needed in the future. To implement them, once the analytical formula and its derivatives are available, is not complicated at all. In principle, the implementation should be easy for anybody who knows a bit the CPMD code.

11.10.1 MTD Algorithm

Once the CV have bee chosen, the MTD method can be applied in two different fashions.

Direct MTD: The simplest approach is to define the time dependent potential as function of $\mathbf{S}$, $V(t, \mathbf{S})$, and apply it directly onto the involved degrees of freedom. In this case, the equations of motion of the dynamic variables of the system, $\mathbf{R}, \phi, \mathbf{h}$, will include an additional term in the total forces, due to the contribution of $V(t, \mathbf{S})$. The disadvantage of this simplified version is that there is scarce control on the dynamics in the space defined by the CV (CV-space), which is a projection of the space of all the possible configurations. In general, we would like to span thoroughly the CV-space, and to acquire information about the underlying potential. Often, this means that we need a slow dynamics in this space, where, for each set of values of the CV, we allow the system to equilibrate and to choose the configuration with the highest probability. Only in this way we will be able to construct a reasonable probability distribution in the configurational space that has been explored and consequently we will be able to reproduce the Free Energy surface.

Lagrangian MTD: This formulation is based on the method of the extended Lagrangian. In addition to the dynamic variables that characterize your system, a new set of variables $\mathbf{s} = \{s_\alpha\}$ is introduced. Each s_α is associated to one of the selected S_α, it has a fictitious mass M_α and velocity $\dot{s}_\alpha$. The equations of motion for the s_α variables are derived by a properly extended Lagrangian, where we add the fictitious kinetic energy and the potential energy as a function of $\mathbf{s}$. Therefore the total potential energy includes two new terms, a sum of harmonic potentials, which couple the s_α to the respective $S_\alpha(\mathbf{R}, \phi, \mathbf{h})$, $\sum_\alpha k_\alpha(S_\alpha(\cdots) - s_\alpha)^2$, and the time dependent potential, which now is a function of $\mathbf{s}$, $V(t, \mathbf{s})$. The coupling constants $\{k_\alpha\}$ and the fictitious masses $\{M_\alpha\}$ are the parameters that determine the dynamics of the $\{s_\alpha\}$ in the CV-space. Please notice that the units of k are Hartree divided by the square power of u.s., the characteristic units of the specific CV (if CV is a distance it will be $a.u.^2$, if an angle $radiants^2$, etc.). In analogy, the units of the fictitious mass are $Hartree((t)/(u.s.))^2$, where t indicates the unit of time. Some guide lines on the choice of these parameters will be given in the following paragraphs. By choosing the temperature $T_\mathbf{s}$, the velocities of the components of $\mathbf{s}$ can be initialized giving via a Boltzmann distribution. Moreover, the velocities can be kept in a desired range by the activation of a temperature control algorithm (at the moment only the rescaling of velocity is implemented).

11.10.2 The Shape of $V(t)$

Several shapes have been tested (and more might be proposed in the future). The default choice is the construction of $V(t)$ by the accumulation of Gaussian-like hills, i.e. (within the Lagrangian formulation, but the expressions are the same for the direct MTD approach, providing to exchange $\mathbf{s}$ with $\mathbf{S}(\cdots)$)

$$V(t,\mathbf{s}) = \sum_{t_i<t}\left[W_i \exp\left\{-\frac{(\mathbf{s}-\mathbf{s}^i)^2}{2(\Delta s^{\perp})^2}\right\} \exp\left\{-\frac{\left((\mathbf{s}^{i+1}-\mathbf{s}^i)\cdot(\mathbf{s}-\mathbf{s}^i)\right)^2}{2(\Delta s_i^{||})^4}\right\}\right], \quad (27)$$

Here, t indicates the actual simulation time, i counts the metadynamics steps, the first exponential gives the hill's shape in the direction perpendicular to the trajectory, whereas the second exponential tunes the shape along the trajectory. In this form, the width of the hill along the trajectory is determined by the displacement in the CV-space, walked between two consecutive metadynamics steps, $\Delta s_i^{||} = f_b\sqrt{\left[\sum_\alpha (s_\alpha^{i+1} - (s_\alpha^i)^2\right]}$. f_b is a factor, which is read in input and can be used to change the size of the hills along the trajectory, by default it is 1. The height W and the width $\Delta s^{\perp}$ are input parameters that can also be tuned during the MTD, in order to better fit the hill shape to the curvature of the underlying energy surface (in the CV-space). As a rule of thumb, $\Delta s^{\perp}$ should have roughly the size of the fluctuations of CV at equilibrium (half the amplitude of the well) and W should not exceed few percents of the barrier's height. These information can be obtained by some short MD runs at equilibrium (without MTD) and from some insight in the chemical/physical problem at hand. Since, in general, different CV fluctuate in wells of different size, it is important to define one scaling factor scf_α for each component s_α, so that $\langle \delta s_\alpha \rangle / scf_\alpha = \Delta s^{\perp}\ \forall \alpha$.

Other implemented shapes of $V(t)$ are:

Shift: the tails of the Gaussians are cutoff, by setting to zero the Gaussian at a distance $R_{cutoff}\Delta s^{\perp}$ from its center. In this way the problem of the overlap of the tails in regions far from the actual trajectory is reduced.

Rational: instead of Gaussian-like hills, some kind of rational functions are used,

$$V(t,\mathbf{s}) = \sum_{t_i<t}\left[W_i \frac{1-\left(\frac{\sqrt{(\mathbf{s}-\mathbf{s}^i)^2}}{\Delta s^{\perp}}\right)^n}{1-\left(\frac{\sqrt{(\mathbf{s}-\mathbf{s}^i)^2}}{\Delta s^{\perp}}\right)^m} \exp\left\{-\frac{\left((\mathbf{s}^{i+1}-\mathbf{s}^i)\cdot(\mathbf{s}-\mathbf{s}^i)\right)^2}{2(\Delta s_i^{||})^4}\right\}\right], \quad (28)$$

where the exponents n and m determine the decay.

Lorentzian: Lorentzian functions are used in place of Gaussians.

In all the cases, a new hill is added at each step of MTD, where $\Delta t_{meta} = t_{i+1} - t_i$ is usually chosen equal to $10 \div 500$ steps of CP-MD (it depends on the relaxation time of the system and the size of the hills). The center of the new hill at time t_{i+1} is positioned along the vector $\mathbf{s}-\mathbf{s^i}$.

11.10.3 Metadynamics Keywords

Now let's start with the explanation of the keywords. First, the definition of the CV is required. The selected CV are read from the input subsection enclosed between the initial and final lines:

DEFINE VARIABLES

...

END DEFINE

Between these two lines the first input line states how many CV are used, *NCOLVAR*. In the following, each variable is described by one or more lines, according to its type. In general, each

line must start with the name of the CV, $type - name$, followed by some indexes or parameters that are needed to specify its analytical function and the kind of atoms or species that are involved. At the end, always on the same line of the $type - name$, the scaling factor scf and, if the extended Lagrangian is used, k and M can be given. If not specified scf, k and M take some default values.
scf: by default, $scf = 1$ and it is fixed during the whole run. Otherwise, you can write **SCF** followed by the value, or **SCA** followed by the value, a lower bound and an upper bound. In the latter case, the scf is tuned along the MTD run. In practice, the average amplitude of the CV fluctuation is checked every time to time, and, if $scf_\alpha \cdot \delta s_\alpha$ is far from $\Delta s^\perp$, the scf_α is changed accordingly.
M: it determines how fast the s variable spans the entire well. Given the width of the well $scf \cdot \Delta s^\perp$ and the temperature T_s, it is possible to choose M by stating the number of complete fluctuations per ps. The default value is taken for 10 fluctuation per ps. Otherwise, you can write **MCV** followed by the desired value for M in $Hartree((t)/(u.s.))^2/1822 = a.m.u.(a.u./a.s.)^2$.
k: it determines the dynamics of s with respect to the dynamics of the physical CV. If $S(\cdots)$ is dominated by fast modes, it is recommended that s be slower and its fluctuations span the entire well. Given the characteristic frequencies of the normal modes ω_0, k can be chosen such that $\sqrt{k/M} < \omega_0$. On the other hand, we want k big enough, so that s and S stay close, and S fluctuates many times at each position in the configuration space. By satisfying the latter condition, the average forces due to the underlying potential can be accurately estimated, and the trajectory lays on the minimum energy path. Therefore, also for k, the default value is chosen in terms of T_s and $scf_\alpha \cdot \delta s_\alpha$ Otherwise, you can write **KCV** followed by the desired value.

On the same line, by writing **WALL+** or **WALL-**, some fixed upper and lower boundaries for the CV can be determined. After the keyword the position of the boundary and the value of the constant repulsive force have to be specified.
Warning: if even only one k or one M is read from input, the Lagrangian formulation of the MTD is initialized.

11.10.4 The Implemented Types of CV

Please note, that for calculations using the Gromos QM/MM-interface (see section 11.16) the atom indices refer to the ordering of the atoms as it appears in the respective GROMOS coordinate file.

- STRETCH: Bond stretch: give the indexes of the 2 atoms $i1$ $i2$, $s = (d_{i1,i2})^2$
- BEND: Bond angle: give the indexes of the 3 atoms defining the angle, $i1$ $i2$ $i3$.
- TORSION: Torsion angle: give the indexes of the 4 atoms defining the torsion angle, $i1$ $i2$ $i3$ $i4$.
- DIST: Distance between two atoms: give the indexes of the 2 atoms $i1$ $i2$, $s = d_{i1,i2}$.
- DISAXIS: Distance between two atoms $i1$ and $i2$ along x or y or z direction. $i1$ $i2$ n are read next on the same line. Here $n = 1$ means x, $n = 2$ means y and $n = 3$ means z coordinate.
- OUTP: Angle out of plane: give the indexes of the 3 atoms defining the plane and a fourth index of the atom for which the angle out of plane is computed, $i1$ $i2$ $i3$ $i4$.
- COORD: Coordination number (CN) of one atom with respect to all the other atoms in the system. The CN is defined by a Fermi-like function

$$CN_i = \sum_{j \neq i}^{NATOM} \frac{1}{1 + e^{k(d_{ij} - d^0)}} \tag{29}$$

  where i is the index of the selected atom, j runs over all the other atoms in the system, k is the parameter which determines the steepness of the decay and d^0 is the reference distance. After the type-name, in the same line, give i k d^0.

- DIFFER: Difference between two distances, give the indexes of the 3 atoms defining the 2 vectors, $i1$ $i2$ $i3$, $s = d_{i1i2} - d_{i2i3}$.

- COORSP: CN of one selected atom i with respect to only one selected species jsp. The CN is defined by a Fermi like function as for $COORD$, but in this case j runs only over the atoms belonging to the selected species jsp. After the type-name, in the same line, give i jsp k d^0.

- COORGROUP: Sum of the CN of a group of atoms A with respect to individual group of atoms (B). CN is estimated using the Fermi function. Different cutoff distances are allowed for each type of A atoms.
$$CN = \sum_{i}^{N_A} \sum_{j}^{N_B(i)} \frac{1}{1 + e^{k[d_{ij} - d^0(i)]}} \tag{30}$$
After the keyword `COORGROUP`, N_A and k should be specified. In the next lines should be:
i, $d^0(i)$, $N_B(i)$
$j(1) \cdots j(N_B(i))$
This has to be done for all i in list of A type atoms.

- COOR_RF: CN of one selected atom i with respect to one selected species, jsp, or a list of atoms, $j1 \cdots jn_{list}$. The CN value is calculated as the sum of rational functions
$$CN_i = \sum_{j \neq i}^{n_{list}} \frac{1 - \left(\frac{d_{ij}}{d^0}\right)^p}{1 - \left(\frac{d_{ij}}{d^0}\right)^{p+q}}, \tag{31}$$
where j runs over the indexes of the atoms belonging to jsp or over the indexes given in the list $j1 \cdots jn_{list}$. For the option of the species, you should provide, after the type-name, the indexes i and jsp, the exponents p and q, and the reference distance d^0 are read. If, instead, the list option is your choice, write immediately after the type-name the keyword $INDAT$, and next the values of i, n_{list}, p, q, and d^0. The indexes of the atoms belonging to the list are read from the next line.
If the keyword $2SHELL$ is found, in the same line as $COOR\_RF$, the first real number after this keyword is a second reference distance d_{2sh}. In this case, the functional form of CN is modified, in order to take into account only the neighbors belonging to one farther shell, and d_{2sh} is the average distance of these atoms from i:
$$CN_i^{2sh} = \sum_{j \neq i}^{n_{list}} \frac{1 - \left(\frac{(d_{ij} - d_{2s})}{d^0}\right)^p}{1 - \left(\frac{(d_{ij} - d_{2s})}{d^0}\right)^{p+q}}. \tag{32}$$
For the modified CN the exponents must be even.

- BNSWT: Reciprocal CN between 2 selected atoms, defined with the same functional form as the one described for $COOR\_RF$. This coordinate states the presence of the bond between the two atoms i and j. After the type-name, give i, j, p, q, and d^0.

- TOT_COOR: Average CN of the atoms belonging to a selected species isp with respect to a second selected species, jsp, or with respect to a given list of atoms, $j1 \cdots jn_{list}$. The same functional forms and input options are used, as those described for $COOR\_RF$, but the index of one selected species isp is read in place of the index of one atom.

- DISPL1: Average displacement of one group of species with respect to a second group of species, computed along one specified direction in space (lattice axis in crystals). This CV is useful to study diffusion processes in condensed matter. If the keyword MIL is found, the 3 Miller indexes, which define the direction in space, are read immediately after (default: $\mathbf{v} = (hkl) = (100)$).

- COORDIS:
- PLNANG: Angle between two planes. Each plane is defined by the coordinates of 3 atoms; after the type-name, give the indexes of the 3 atoms defining the first plane, $i1$ $i2$ $i3$, and the indexes of the atoms defining the second plane, $j1$ $j2$ $j3$.
- HBONDCH:
- DIFCOOR: Difference between the CN of two different atoms, $i1$ and $i2$, with respect to the same species jsp, or the same list of atoms,$j1 \cdots jn_{list}$. The same functional forms and input options are used, as those described for $COOR\_RF$, but the index of two selected atoms are read, $i1$ and $i2$, rather than one.
- RMSD_AB: Given two atomic configurations A and B, the root mean square displacements (RMSD) of the actual configuration from A, $rmsdA$, and from B, $rmsdB$, are calculated (global translation and rotation are subtracted by the method of quaternions). The RMSD can be calculated on selected group of species: after the type name give the number of species ($NUMSPEC$) and the indexes of the selected species ($IS_1 \cdots IS_{NUMSPEC}$). If $NUMSPEC = 0$ all the species are included. If in the same line the keyword **FILEAB** is found, next the file name is read, where the atomic positions of the configurations A and B are given. Otherwise the file name is by default **STRUCTURE_AB**. File format: 2 consecutive blocks of $1 + NATOM$ lines. In each block, the first line is a title (Character) and it is followed by the list of atomic coordinates in a.u. ($element\_name\, x\, y\, z$).
- COOR_CHAIN: Conditioned CN. Given three species $isp1$, $isp2$, and $isp3$, the following average CN is calculated

$$CN = \frac{1}{N_{sp1}} \sum_{i1=1}^{N_{sp1}} \left[\sum_{i2=1}^{N_{sp2}} \left(\frac{1 - \left(\frac{d_{i1i2}}{d_{12}^0}\right)^p}{1 - \left(\frac{d_{i1i2}}{d_{12}^0}\right)^{p+q}} \times \sum_{i3=1}^{N_{sp3}} \frac{1 - \left(\frac{d_{i2i3}}{d_{23}^0}\right)^p}{1 - \left(\frac{d_{i2i3}}{d_{23}^0}\right)^{p+q}} \right) \right]. \tag{33}$$

  After the type-name, the parameters $isp1$, $isp2$, $isp3$, p, q, d_{12}^0, and d_{23}^0 are read.
- HYDRONIUM:
- DIS_HYD:
- SPIN: Distance between a selected atom and the center of the spin polarization ($\rho_\uparrow - \rho_\downarrow$), where ρ indicate the polarized density. The center is located where the difference is maximum, and this kind of variable is useful only when some spin polarization is present. The position of the center in systems with PBC can be calculated by the definition proposed by Resta [73, 75]. Obviously, this CV can be used only together with LSD. After the type-name, give the index of the selected atom.
- VOLVAR: Volume of the cell. It can be used only with NPE/NPT MD.
- CELLSIDE: Length of one cell's side: give the cell-side's index i ($i_a = 1$,$i_b = 2$,$i_c = 3$).It can be used only with NPE/NPT MD.
- CELLANGLE: Cell-angle: give the cell-angle's index i ($i_\alpha = 1$,$i_\beta = 2$,$i_\gamma = 3$)). It can be used only with NPE/NPT MD.
- VCOORS This CV represents the coordination of one point (V) with respect to a selected species of atoms $jspec$ in the system:

$$CN_V^{jspec} = \sum_{i \in jspec}^{N_{jspec}} \frac{1}{1 + e^{k(d_{iV} - d_0)}} \tag{34}$$

  After the keyword the parameters $jspec, k, d_0$ are read. In the next line the coordinates of the point V are read in a.u.

- DIPOLE The dipole of the atoms $i_1, ..., i_N$ with respect to the atom j is defined as:

$$\vec{D}_j = \frac{1}{Q} \sum_{i=i_1,..,i_N} q_i(\vec{r}_i - \vec{r}_j); \quad Q = \sum_{i=i_1,..,i_N} q_i \tag{35}$$

  The three spherical coordinates of $\vec{D}_j$, that is $(\rho_j, \theta_j, \phi_j)$, can be used independently as CV. The keywords are **DIPOLERHO**, **DIPOLETHA**, **DIPOLEPHI**. In the same line after the keywords are read the index of the atom j and the number N of atoms which constitute the dipole. In the next two lines are read the indexes of the N atoms and the corresponding charge q_i.

If $CELL\ FULL$ is defined in the first line of the MTD input, none of the CV defined above is used. The CV are the 6 cell parameters. In the section $DEFINE\ VARIABLE$, the number of CV is 6 and in the following 6 lines the scaling factors are given: for each line write the index i of the corresponding CV ($i_a = 1$,$i_b = 2$,$i_c = 3$,$i_\alpha = 4$,$i_\beta = 5$,$i_\gamma = 6$) followed by SCF or SCA and the desired values (see the description at the beginning of this subsection).

11.10.5 Other Keywords

- ANALYSIS: A standard MD run is performed, where the equations of motions are not affected by the hills-potential or the coupling potential. The selected CV are monitored and the values are reported in the output file, after every 10 MD steps. This option is useful in order to observe the behavior of the selected CV in equilibrium conditions. With this option only two output files are written: *istvar_mtd*, and *enevar_mtd* (see section 11.10.6). The former file contains the values of the S_α and their averages in time.

- METASTEPNUM: The maximum number of MTD steps is read from the next line, I_META_MAX (default: 100).

- META_RESTART: To restart a metadynamic's run from where the former run has stopprd, one can use this keyword, and write in the following line, the number of meta-steps completed already I_META_RES (default: 0). Beware that for restarting the MTD in this way, the output files of the previous run are to be available in the run's directory and must contain a number of lines at least equal to the I_META_RES. From this files the previous history of the MTD is read and the MTD is initialized accordingly.
  Otherwise, it is possible to restart from the restart file of the MTD, $MTD\_RESTART$. This is an unformatted file, which is written whenever the standard CPMD restart file, $RESTART$.1, is also written. It contains the number of meta-steps already performed, the number of CV used in the previous run and the information about the position and the size of the hills which have been already located. To restart the MTD from this file, the same keyword is used, and the keyword $RFILE$ is added in the same line, providing that the unformatted restart file is available in the run's directory. In this second case the number of performed meta-step is not read from the input file but from the restart file
  . Obviously, when a run is restarted, the same number and the same kinds of CV must be used. However masses, force constants, scaling factors, and the width and height of the hills can be changed.

- MINSTEPNUM INTERMETA: The minimum number of MD steps between two MTD steps (in general, the MTD step is characterized by the positioning of a new hill in the CV-space) is read from the next line, $INTER\_HILL$ (default: 100). This is a lower bound because, before the construction of a new hill, the displacement in the CV-space is checked, and the new step is accepted only if the calculated displacement is above a given tolerance.

- MOVEMENT CHECK: The tolerance for the acceptance of a new MTD step is read from the next line (default: 0.001D0)

- CHECK DELAY: The number of MD steps to be run, before a new check of the displacement is done, is read from the following line (default: 20).

- MAXSTEPNUM INTERMETA: The maximum number of MD steps that can be run, before a new MTD step is accepted anyway, is read from the following line (default: 300).

- METASTORE [NO TRAJECTORY]: In the following line, three integer numbers are given, which indicate respectively how often (in terms of MTD steps) the $RESTART$.1 and the $MTD\_RESTART$ files are over-written (default: 50), how often the trajectory files are appended (default: 1), and how often a quench of the electronic wavefunctions onto the BO surface is performed (default: 10000). With the additional flag NO TRAJECTORY, the trajectories are still written according to the settings as indicated by the **TRAJECTORY** keyword in the &CPMD section. The selection of the files (e.g. turning on TRAJEC.xyz via the XYZ flag and turning TRAJECOTORY off via a negative value of NTRAJ in combination with the SAMPLE flag) is always honored.

- MAXKINEN: From the following line, the maximum electronic kinetic energy is given, above which a quench of the electronic wavefunctions onto the BO surface is performed anyway (by default no quench is done whatever is the kinetic energy).

- LAGRANGE TEMPERATURE: The temperature T_s used to initialize the velocities of the **s** CV is read from the next line. By default T_s is chosen equal to the temperature of the ions, the units are Kelvin. Notice that this keyword causes the initialization of the Lagrangian formulation of MTD.

- LAGRANGE TEMPCONTROL: The control of the T_s is activated, and the rescaling velocities algorithm is used. The average temperature and the permitted range of variation are read from the next line. By default T_s is not controlled. Notice that this keyword causes the initialization of the extended degrees of freedom of the Lagrangian formulation of MTD.

- LAGRANGE LANGEVIN: Performs Langevin dynamics for the Lagrangian formulation of MTD. In the next line the Temperature (Kelvin) and the friction γ (a.u.) are read. The Langevin equation in its standard form writes (z is a CV):

$$M\ddot{z} = F(z) - \gamma M\dot{z} + \sqrt{2k_BT\gamma M}\eta(t) \tag{36}$$

where M is the CV mass, F a generic force field, γ the friction coefficient, T the temperature and η is a white noise. The integration algorithm is a Velocity-Verlet which can be written as:

$$\begin{aligned}
\dot{z}_{n+1/2} &= \dot{z}_n + \frac{1}{2}dt\Big[\frac{F(z_n)}{M} - \gamma\dot{z}_n\Big] + \frac{1}{2}\sqrt{\frac{2k_BT\gamma dt}{M}}\xi_n \\
z_{n+1} &= z_n + \dot{z}_{n+1}dt \\
\dot{z}_{n+1} &= \dot{z}_{n+1/2} + \frac{1}{2}dt\Big[\frac{F(z_{n+1})}{M} - \gamma\dot{z}_{n+1/2}\Big] + \frac{1}{2}\sqrt{\frac{2k_BT\gamma dt}{M}}\xi_n
\end{aligned} \tag{37}$$

the ξ_n are independent Gaussian random numbers with mean zero and variance one.

- HILLS: With this keyword it is defined the shape of $V(t)$. If OFF is read in the same line, no hill potential is used. The default hills' shape is the Gaussian-like one described above. If $SPHERE$ is read from the same line, the second exponential term in equation 27 is not applied. i.e., a normal Gaussian function rather than a Gaussian tube formalism is used. If, instead, $LORENZIAN$ is read in the same line as $HILLS$, Lorentzian functions are used in place of Gaussians. If, instead, $RATIONAL$ is read in the same line as $HILLS$, the rational function described in the previous section is used; in this case, if $POWER$ is read, the exponents n and m, and the boosting factor f_b are also read immediately after (defaults: $n = 2$, $m = 16$, $f_b = 1$). If, instead, $SHIFT$ is read on the same line as $HILLS$, the shifted Gaussians are used, where the tails after a given cutoff are set equal to zero; in this case, if $RCUT$ is read, the cutoff $rshift$ and the boosting factor f_b are also read immediately after (defaults: $rshift = 2$,$f_b = 1$).

In all this cases, if the symbol = is read in the same line as $HILLS$, the perpendicular width $\Delta s^{\perp}$ and the height W are read immediately after (defaults: $\Delta s^{\perp} = 0.1\,u.s.$, $W = 0.001\,Hartree$).

- TUNING HHEIGHT: With this keyword the height of the hills is tuned according to the curvature of the underlying potential. If the symbol = is read in the same line as $TUNING$ $HHEIGHT$, the lower and upper bounds of W are read immediately after (defaults: $W_{min} = 0.0001\,Hartree$, $W_{max} = 0.016\,Hartree$).

- HILLVOLUME: With this keyword the volume of the hills, $\sim W \cdot (\Delta s^{\perp})^{NCOLVAR}$, is kept constant during the MTD run, i.e. when the height changes due to the tuning (see the previous keyword), the width is changed accordingly. This option is can be used only if the tuning of the hills' height is active.

- CVSPACE BOUNDARIES: With this keyword the confinement of the CV-space is required, in the direction of a selected group of CV. The number of dimensions of the CV-space, in which the confinement is applied, is read from the next line, $NUMB$ (default: 0). The following $NUMB$ lines describe the type of confinement. For each line the following parameters are required: index of the CV (as from the list given in the definition of the CV), the strength of the confinement potential V_0 in Hartree, two real numbers, $C1$ and $C2$, that determine from which value of the CV the confining potential is active. Finally, if the keyword EPS is read in the same line, the real number, which is read immediately after, determines how smoothly the confining potential is switched on (default $\varepsilon = 0.01$). The confining potential can be used with the following CV:
  DIST: $V_{conf} = V_0 \left(\frac{s_\alpha}{C1}\right)^4$ and it becomes active only if $s_\alpha > C2$.
  DIFFER: $V_{conf} = V_0 \left(\frac{|s_\alpha|}{C1}\right)^4$ and it becomes active only if $s_\alpha > C2$ or $s_\alpha < -C2$.
  Coordination numbers: if $(CN - \varepsilon) < C1$, $V_{conf} = V_0(\frac{C1}{CN})^{10}$, if $(CN + \varepsilon) > C2$, $V_{conf} = V_0(\frac{CN}{C2})^{10}$.

- RESTRAIN VOLUME: With this keyword a confining potential is applied to the volume variations. The option can be used only in combination with the NPE/NPT MD. From the next line, the following parameters are required: f_{min}, f_{max}, and V_0. The factors f_{min} and f_{max} multiplied by the initial volume of the cell, give, respectively, the lower and upper bounds for the volume, whereas V_0 gives the strength of the confining potential.

- MULTI NUM: his keyword should be used when a multiple set of MTD runs are performed simultaneously on the same system. Here the number of separated sets CV for each subsystem has to be given in the following line, $NCVSYS_1 \cdots NCVSTS_{NSUBSYS}$. This means that the first $NCVSYS_1$ CV, in the list of those defined in the $DEFINE\ VARIABLES$ section, will belong to the first subset, the next $NCVSYS_2$ to the second, and so on. This option is implemented only together with the extended Lagrangian formulation.

- MONITOR: This keyword requires that an additional monitoring of the values of the CV is performed along the MD trajectory. This means that the values are written on an output file every *WCV_FREQ* MD steps, even if no hill is added at that step. The frequency for updating the file is read from the following line, the file name is *cvmdck_mtd*, and it is not created if this option is not activated.

- RANDWALK: In the case of multiple walker metadynamics, collective variables of all walkers are initialized with different random velocities.

11.10.6 Output files

During a run of MTD, several output file are updated at each MTD step, which are characterized by the extension *_mtd*. These files contain the history of the MTD, the parameters giving the additional potential, and other information that can be useful during the analysis of results. The

first column for all these files is the CPMD step number (NFI) that corresponds to this MTD step. In the case of $MULTI$ MTD some of the files have a further extension $\_s\#$, which indicates the related subsystem.
Extended Lagrangian MTD:
istvar_mtd: the first $NCOLVAR$ columns are the $S_\alpha(\cdots)$, the next $NCOLVAR$ columns are differences $S_\alpha(\cdots) - s_\alpha$.
colvar_mtd: the first $NCOLVAR$ columns are the s_α, the next $NCOLVAR$ columns are the corresponding scaling factors scf_α.
parvar_mtd: norm of the total displacement in the CV-space, $\Delta s^{||}$, hill's width, $\Delta s^{\perp}$, hill's height, W.
disvar_mtd: the first $NCOLVAR$ columns are the displacements of the s_α, the next $NCOLVAR$ columns are their diffusivities, the final $NCOLVAR$ columns are the coupling constants, k_α.
velvar_mtd: the velocities of the s_α.
forfac_mtd: the first $NCOLVAR$ columns are the forces coming from the coupling potential (sum of harmonic terms, V_{harm}), the next $NCOLVAR$ columns are the forces coming from $V(t)$, the last $NCOLVAR$ are the forces coming from the confining potential.
enevar_mtd: ions temperature, electrons kinetic energy, **s** CV temperature $2K_{\mathbf{s}}/(NCOLVARk_B)$, V_{harm}, $V(t)$ at the actual position in CV-space, KS energy E_{KS}, $E_{tot} + K_s + V_{harm}$, $E_{tot} + K_s + V_{harm} + V(t)$. *cvmdck_mtd*: monitoring o the CV along the MD trajectory. This file is updated every it WCV_FREQ MD steps (see previous section $MONITOR$)
Direct MTD:
colvar_mtd: the first $NCOLVAR$ columns are the $s(\cdots)_\alpha$, the next $NCOLVAR$ columns are the corresponding scaling factors scf_α.
sclvar_mtd: the first $NCOLVAR$ columns are the scaled $s(\cdots)_\alpha$, the next $NCOLVAR$ columns are the corresponding scaled diffusivities.
parvar_mtd: norm of the total displacement in the CV-space, $\Delta s^{||}$, hill's width, $\Delta s^{\perp}$, hill's height, W.
disvar_mtd: the first $NCOLVAR$ columns are the displacements of the s_α, the next $NCOLVAR$ columns are their diffusivities, the final $NCOLVAR$ columns are the coupling constants, k_α.
forfac_mtd: the first $NCOLVAR$ columns are the forces coming from $V(t)$, the last $NCOLVAR$ are the forces coming from the confining potential.
enevar_mtd: ion temperature, electrons kinetic energy, $V(t)$ at the actual position in CV-space, KS energy E_{KS}, $E_{tot} + V(t)$.

11.10.7 Using multiple walker metadynamics

Multiple walker metadynamics is activated using the $MULTIPLE\ WALKER$ keyword in the initial input line of metadynamics (i. e., after $METADYNAMICS$ keyword). Multiple walker using the extended Lagrangian metadynamics in combination with the Car–Parrinello type of dynamics is only implemented at the moment. From the same line of $MULTIPLE\ WALKER$ keyword, the number of walkers (NWALK) is read as $NW = NWALK$ (without any space in between).
$NWALK$ replicas are created, and for each DFT forces and energy calculations are done independently. However, all the replicas fill the same free energy surface encompassed by the set of reaction coordinates specified [164]. Implementation is done in such a way that each replica belongs to a different processor group,and each processor group is able to perform independent DFT calculations in parallel; if $NPROC$ number of processors are used, each replica is using $NPROC/NWALK$ number of processors [165] for computations. See the output file for the details on the division of processors in to corresponding processor groups. Note that output of all the walkers are currently dumped in to the (same) standard output file. Trajectory, geometry and energy files of each walkers are separately written out in files having their usual names augmented with $\_IWALK$, where $IWALK$ is the walker ID.
If a multiple walker run has to be started from a single restart file, copy or link it $NWALK$ times as RESTART_1, $\cdots$ RESTART_$NWALK$ (similarly the MTD_RESTART file, if is also restarted). In the procedure of creating new walkers, like above from one restart or increasing

walker numbers during the run, it is advised to initially run with zero hill height, still keeping all the biasing potentials accumulated upto then (i. e. restarting from the MTD_RESTART file if any previous metadynamics runs have been done), and use *RANDWALK* keyword until the (all) walkers are far apart (at least 1.5 x hill width) from each other. In the following (restart) run, use the required hill height and remove the *RANDWALK* keyword. Note that the frequency of adding hills will be nearly *NMTD/NWALK* if *NMTD* is the number of MD steps required to add a hill using 1 walker. Thus consider decreasing the *MINSTEPNUM INTERMETA* appropriately. However, it is highly recommended to use the adaptive metadynamics time step using *MOVEMENT CHECK* keyword, and the tolerance is typically 1.5 times the hill width parameter [162]; better set the *CHECK DELAY* to 1. Displacement tolerance are also forced to satisfy between the walkers when *MOVEMENT CHECK* is used. Note that it is possible to decrease the number of walkers, in a straight forward manner during a restart run.

11.10.8 Shooting from a Saddle

Once one reactive trajectory has been found, one may want to determine more precisely the position of the transition state region. A standard way to do this is to select some points along the trajectory, and, by shooting with random velocities a new MD from this point, measure the probability to reach the surrounding basins of attraction [166]. The different basins of attraction can be identified by different values of a selected set of CV. One can say that the trajectory has fallen in one of the known basins when all the actual CV values satisfy the values characterizing that basin, within a certain tolerance. Given a set of coordinates, one can start a CPMD run where this check is iterated as many times as you like, in order to establish the commitor distribution.The search for the saddle point region is initialized when in the section &ATOMS &END of the input file, the keyword *SADDLE POINT* is found. In what follows, a subsection for the description of the selected CV is required. It has the same format as the one used for the MTD run.

11.10.9 Keywords

The list of keyword regarding the shooting needs to be ended by the line
END SADDLE
Other keywords are:

- KNOWN_MINIMA The values of the CV, which characterize the known basin of attraction, are read from the following lines. The first line after the keyword contains the number of the known minima *NCVMIN*. The next *NCVMIN* lines contain the set of values for each of these minima. The list of the values must keep the same order used in the definition of the CV. If on the same line as *KNOWN_MINIMA*, the keyword *EXTENDED* is also found, each line contains *NCOLVAR* more entries, which are the tolerances for the acceptance of the corresponding minimum configuration (the order of the tolerances must be the same as the one for the CV values). By using the *EXTENDED* keyword, each minimum configuration can be accepted with different tolerances.

- SADDLE TOLERANCES If the *EXTENDED* keyword is not used, one single set of tolerances (one for each CV) can be given by using this keyword. The tolerances are read from the next line in the same order used for the CV definition. Otherwise, default values are assigned at each tolerance.

- MAXSEARCH The maximum number of trials, where a new MD trajectory is generated, is read from the next line. At each trial, the MD starts from the same initial coordinates, whereas the initial velocities are randomly generated at every new restart. During one trials, every *NSTEP* the actual values of the CV are checked and compared to the values given for the known minima. If all the values of one of these minima are satisfied within the given tolerances, the MD is stopped and restarted for the next trial.

- STEPCHECK The number of MD steps between two consecutive checks is read from the next.

- MAXCHECKS The maximum number of checks for each trial is read from the next line.

11.11 Restricted Open-Shell Calculations

Molecular dynamics simulations in the first excited state can be performed using Restricted Open-Shell Kohn-Sham (ROKS) theory [31]. The keyword **ROKS** in the &CPMD section defaults to the first excited singlet state. Solving open-shell equations is not simple unless

1. a high-spin state is computed.
2. the two singly occupied molecular orbitals (SOMOs) have different spatial symmetry.

In these two cases the Goedecker-Umrigar-Algorithm (GOEDECKER) may be used which shows the best convergence properties and is applicable in connection with Car-Parrinello molecular dynamics. Otherwise it is necessary to use a modified variant of the Goedecker-Umrigar-Algorithm and to do Born-Oppenheimer molecular dynamics (unless you know what you are doing). In almost all cases, the default algorithm (DELOCALIZED) is applicable, whereas for example some dissociation reactions require the localized variant to enable localization of the orbitals on the fragments.

```
ROKS LOCALIZED
```

In order to make sure that the chosen algorithm works for a certain system, the conservation of energy during a molecular dynamics simulation and the shape of the orbitals should always be checked. One of the SOMOs should have the same nodal structure as the HOMO obtained by a ground state calculation. If using the unmodified Goedecker-Umrigar scheme (GOEDECKER), the energy of the singlet may collapse to approximately the triplet energy if the two SOMOs do not have different symmetries. The triplet energy can be calculated by specifiying

```
ROKS TRIPLET
```

or also

```
ROKS TRIPLET GOEDECKER
```

See the description of the keywords **LOW SPIN EXCITATION**, **LSE PARAMETERS** and **MODIFIED GOEDECKER** for a description of how to do ROKS calculations using the older input LOW SPIN EXCITATION ROKS. ROKS GOEDECKER corresponds to LOW SPIN EXCITATION ROKS whereas ROKS DELOCALIZED corresponds to LOW SPIN EXCITATION ROKS with MODIFIED GOEDECKER. Do not use LOW SPIN EXCITATION in the &SYSTEM section and ROKS in the &CPMD section at the same time.

ROKS is not implemented with Vanderbilt pseudopotentials.

A Slater transition-state density between a singlet ground state and the first excited singlet state (or any pair of states described with ROKS) can be useful whenever one set of Kohn-Sham states is required which is equally well suited for each of the states involved in a transition, e.g., to calculate the couplings between the electronic transition and an external influence. This method is analogous to state-averaged multiconfigurational SCF methods and shares many of their benefits with them. In CPMD, it can be used to calculate non-adiabatic couplings between singlet states [69, 88], see options **COUPLINGS**.

11.12 Hints on 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.

11.12.1 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 $H[\psi_1, ..., \psi_{nblock}]$. $Nblock$ 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

$$|H\psi - \epsilon\psi|^2 < tolerance \tag{38}$$

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.

11.12.2 Other important FEMD parameters

The keyword STATES defines the dimension of the subspace used in the diagonalization. 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.

11.13 The Davidson analysis and the shared electron number

The calculation of the shared electron number can have the following input section:

```
&PROPERTIES
  PROJECT WAVEFUNCTION
  POPULATION ANALYSIS MULLIKEN DAVIDSON 2-CENT 3-CENT
    4
    1
  WAVEFUNCTION LATEST
&END
```

Note, that for the hydrogen it is enough to specify one atomic orbital to project on, for the elements Li to Ne it is sufficient to specify 4 atomic orbitals.

11.14 CDFT Theory

We implemented the constrained DFT (CDFT) method as developed by Wu and van Voorhis [167, 168, 169, 170, 171].

11.14.1 The density constraint

Here, one imposes a scleronomic constraint on the electron density to reproduce a certain charge distribution on the atoms,

$$\int w(\mathbf{r})\rho(\mathbf{r})\, d\mathbf{r} - N_c = 0, \tag{39}$$

with $\rho(r)$ being the electron density $w(r)$ the weight and $\mathrm{N_c}$ the constraint value. The new energy functional:

$$W[\rho, V] = E[\rho] + V(\int w(\mathbf{r})\rho(\mathbf{r})\, d\mathbf{r} - N_c), \tag{40}$$

with $E[\rho]$ being the normal Kohn-Sham energy functional.

11.14.2 The weight

We chose the Hirshfeld partitioning scheme [63] in order to define charges. The weight for imposing a charge difference between a donor D and an acceptor A is then given by:

$$w(\mathbf{r}) = \frac{\sum_{i\in D} \rho_i(\mathbf{r} - \mathbf{R}_i) - \sum_{i\in A} \rho_i(\mathbf{r} - \mathbf{R}_i)}{\sum_{i=1}^{N} \rho_i(\mathbf{r} - \mathbf{R}_i)}, \tag{41}$$

where the sums in the numerator range over all donor and acceptor atoms, respectively, whereas the sum in the denominator ranges over all N atoms. $\rho_i(r)$ is the unperturbed electron density of atom i which is given by

$$\rho_i(r) = \sum_j n_j \frac{|\psi_i^j(r)|^2}{r^2}, \tag{42}$$

where the sum ranges over all orbitals and $\psi_i^j(r)$ and n_j denote the reference orbitals and n_j their occupation number, respectively.

11.14.3 Constraint forces

Using the Hellmann-Feynman theorem we can express forces on atom k at position R_k

$$\frac{\partial W}{\partial \mathbf{R}_k} = \langle\psi|\frac{\partial \hat{H}}{\partial \mathbf{R}_k}|\psi\rangle. \tag{43}$$

Applying this relation we can calculate the additional forces due to the bias potential:

$$\mathbf{F}_{\mathrm{Bias},k} = -V\int \rho(\mathbf{r})\frac{\partial w(\mathbf{r} - \mathbf{R}_k)}{\partial \mathbf{R}_k}\, d\mathbf{r}, \tag{44}$$

with

$$\frac{\partial w(\mathbf{r} - \mathbf{R}_k)}{\partial \mathbf{R}_k} = -\frac{\rho_k'(|\mathbf{r} - \mathbf{R}_k|)}{\sum \rho_i(|\mathbf{r} - \mathbf{R}_k|)} G_k(\mathbf{r} - \mathbf{R}_k) \tag{45}$$

$$G_k(\mathbf{r} - \mathbf{R}_k) = \begin{cases} w(\mathbf{r} - \mathbf{R}_k) - 1 & k \in D \\ w(\mathbf{r} - \mathbf{R}_k) + 1 & k \in A \\ w(\mathbf{r} - \mathbf{R}_k) & k \notin D \cup A \end{cases} \tag{46}$$

Finally the derivative of ρ_i is given by

$$\begin{aligned}\rho'_k(|\mathbf{r}-\mathbf{R}_k|) &= \frac{\partial\rho_k(|\mathbf{r}-\mathbf{R}_k|)}{\partial\mathbf{R}_k} = \frac{\partial\rho_k(|\mathbf{r}-\mathbf{R}_k|)}{\partial|\mathbf{r}-\mathbf{R}_k|}\frac{\partial|\mathbf{r}-\mathbf{R}_k|}{\partial\mathbf{R}_k}, \\ &= \frac{\partial\rho_k(|\mathbf{r}-\mathbf{R}_k|)}{\partial|\mathbf{r}-\mathbf{R}_k|}\frac{\mathbf{r}-\mathbf{R}_k}{|\mathbf{r}-\mathbf{R}_k|}. \end{aligned} \tag{47}$$

The radial partial derivative ρ_i of is finally calculated numerically using splines.
For a more thorough treatment of the topic of constrained DFT MD please consult reference [172].

Cutoff correction In order to avoid integrating over every real space gridpoint we introduced a cutoff R_c in the generation of the weights. R_c is chosen for each atom species such that the total reference density is smaller than 10^{-6}.
The action of the cutoff can be described by a Heaviside function θ

$$\rho_k(|\mathbf{r}-\mathbf{R}_k|) \to \rho_k(|\mathbf{r}-\mathbf{R}_k|)\theta(R_c - |\mathbf{r}-\mathbf{R}_k|). \tag{48}$$

Therefore the derivative of the reference density becomes

$$\rho'_k(|\mathbf{r}-\mathbf{R}_k|) \to \rho'_k(|\mathbf{r}-\mathbf{R}_k|)\theta(R_c - |\mathbf{r}-\mathbf{R}_k|) + \rho_k(|\mathbf{r}-\mathbf{R}_k|)\delta(R_c - |\mathbf{r}-\mathbf{R}_k|), \tag{49}$$

with the Dirac δ function. Thus the full force splits up into

$$\mathbf{F}_{\text{Full},k} = \mathbf{F}_{\text{Bias},k} + \mathbf{F}_{\text{Bound},k}. \tag{50}$$

Here $\mathbf{F}_{\text{Bound},k}$ denotes forces due to the finite cutoff. They can be expressed by the following surface integral in spherical coordinates

$$\mathbf{F}_{\text{Bound},k} = -V_c\rho_k(R_c)R_c \int \frac{\rho(R_c,\vartheta,\varphi)G(R_c,\vartheta,\varphi)}{\sum\rho_i(R_c,\vartheta,\varphi)} R_c \begin{pmatrix}\sin\vartheta\cos\varphi \\ \sin\vartheta\sin\varphi \\ \cos\vartheta\end{pmatrix} \sin\vartheta \; d\vartheta d\varphi \tag{51}$$

As we do have a Cartesian grid we need to perform the integral as an integral over a thin shell in real space. R_c times the vector is just the position vector of a point on the surface (x, y, z) and the surface element can be expressed in Cartesian coordinates as

$$\sin\vartheta \; d\vartheta d\varphi = \text{sgn}(z)\frac{y\; dxdz - x\; dydz}{R_c(R_c^2 - z^2)} \tag{52}$$

11.14.4 Transition Matrix Element calculation

The following technical description of CDFT matrix element calculations can also be found, together with some test calculation and an investigation of the influence of the choice of the weight function on the results, in ref. [173].
In order to calculate the electronic transition matrix element we first need to calculate two constrained wave functions ϕ_A and ϕ_B and write down the Hamiltonian in the non-orthogonal constrained basis spanned by these two states.

Hamiltonian in the non-orthogonal constrained basis

$$\mathbb{H}_{\text{no}} = \begin{pmatrix} H_{BB} & H_{BA} \\ H_{AB} & H_{AA}\end{pmatrix}, \tag{53}$$

where A and B denote the two states, respectively. Here the diagonal elements are just given by the respective energies of donor and acceptor due to:

$$H_{BB} = \langle\phi_B|H|\phi_B\rangle = E_B \tag{54}$$

$$H_{AA} = \langle\phi_A|H|\phi_A\rangle = E_A, \tag{55}$$

while the off-diagonal elements are determined by:

$$\begin{aligned} H_{BA} &= F_B\langle\phi_B|\phi_A\rangle - V_c^B\langle\phi_B|W|\phi_A\rangle && (56)\\ &= F_B S_{BA} - V_c^B W_{BA} && (57)\\ H_{AB} &= F_B S_{AB} - V_c^A W_{AB}. && (58)\end{aligned}$$

Here we also introduced the overlap matrix element S_{AB} and the weight matrix element W_{AB}. From now on we will only consider one of the off-diagonal elements as the calculation of the other one is analogous. Approximating the wave functions of the two states by slater determinants built from Kohn-Sham orbitals we can write in a plane wave basis:

$$S_{AB} = \langle\phi_B|\phi_A\rangle = \det\left[\sum_G (c_i^A(G))^\star c_j^B(G)\right] = \det\Phi_{ij}, \tag{59}$$

where the $c_{i,j}^{A,B}(G)$ denote the plane wave coefficients of the respective wave function for their respective electronic states i and j and Φ denotes the full overlap matrix. Note that the diagonal elements of $\mathbb{S}$ are equal to one due to the normalisation of the two states. The off-diagonal $\mathbb{W}$ matrix element is given by:

$$W_{AB} = \langle\phi_B|W|\phi_A\rangle = N\sum_{i=1}^{N}\sum_{j=1}^{N}\langle\phi_B^i|W|\phi_A^j\rangle(-1)^{i+j}\det\Phi_{(i,j)}, \tag{60}$$

with $\Phi_{(i,j)}$ being the i,jth minor of Φ and the integral given by

$$\langle\phi_B^i|W|\phi_A^j\rangle = \sum_{G,G'}(c_i^A(G))^\star c_j^B(G)\tilde{W}(G-G'). \tag{61}$$

Note that Equ. 61 actually constitutes a convolution in G-space. Therefore the most efficient way to calculate the elements $\langle\phi_B^i|W|\phi_A^j\rangle$ is by transforming Equ. 61 back into real-space where the convolution becomes a simple multiplication. Additionally CPMD already stores the wavefunctions in real-space (REAL SPACE WFN KEEP which is activated automatically by the CDFT HDA calculation) and thus saves us the necessary FFTs. Calculating every minor determinant of the overlap matrix in Equ. (60) would be rather costly, therefore we note that the second part of Equ. (60) is actually the cofactor matrix C of Φ.

$$(-1)^{i+j}\det\Phi_{(i,j)} = C_{ij}, \tag{62}$$

which can be calculated using the Laplace Expansion of the inverse of Φ:

$$C^T = \Phi^{-1}.[\det(\Phi)I] \tag{63}$$

Thus we only have to calculate the inverse and the determinant of Φ instead of all the minors. The diagonal elements of $\mathbb{W}$ are just the constraint values N_c for the two states. Having performed this calculation we symmetrise $\mathbb{H}_{\text{no}}$ in order to correct for inaccuracies of the approximations we made.

Full diabatic non-orthogonal Hamiltonian However, the matrix $\mathbb{H}_{\text{no}}$ is not the full Hamiltonian as it neglects the fact that the two diabatic wavefunctions form a non-orthogonal basis ($S_{AB} \neq 0$). Nevertheless, we can construct the full Hamiltonian using the overlap matrix element S_{AB}:

$$\mathbb{H}_{\text{full}} = \frac{1}{1-S_{AB}^2}\begin{pmatrix} H_{BB} - S_{AB}H_{AB} & H_{BA} - S_{AB}H_{AA} \\ H_{AB} - S_{AB}H_{BB} & H_{AA} - S_{AB}H_{BA}\end{pmatrix}, \tag{64}$$

Diabatic orthogonal Hamiltonian In order to compare our matrix elements with other methods an experiments we follow the procedure of Wu and Van Voorhis and first solve the two-dimensional generalised eigenvalue problem

$$\mathbb{W}\mathbb{V} = \mathbb{S}\mathbb{V}\mathbb{L}, \tag{65}$$

where $\mathbb{V}$ is the matrix of generalised eigenstates and $\mathbb{L}$ is the diagonal matrix of generalised eigenvalues. To calculate the diabatic orthogonal Hamiltonian we then perform the similarity transformation

$$\mathbb{H}_{\text{diab}} = \mathbb{V}^{-1}\mathbb{H}_{\text{full}}\mathbb{V}. \tag{66}$$

The off-diagonal elements of $\mathbb{H}_{\text{diab}}$ –or their average if the two states have been very different– are then the desired transition matrix element.

Adiabatic Hamiltonian The adiabatic energies of ground and excited state $\varepsilon_{g,e}$ and their mixing coefficients $\mathbf{x}_{g,e}$ can be calculated by simply diagonalising the diabatic Hamiltonian (in principle either one as a similarity transform does not change the eigenvalues of a matrix, here we use the full Hamiltonian)

$$\mathbb{H}_{\text{full}}\mathbf{x}_{g,e} = \varepsilon_{g,e}\mathbf{x}_{g,e} \tag{67}$$

With the matrix $\mathbb{X}$ of mixing coefficients we could then in principle also calculate the adiabatic states themselves via

$$\begin{pmatrix} \psi_a^{\text{ad}} \\ \psi_b^{\text{ad}} \end{pmatrix} = \mathbb{X} \begin{pmatrix} \psi_A \\ \psi_B \end{pmatrix} \tag{68}$$

Projection on reference states Unfortunately, CDFT only constrains a total charge density, thus sometimes one may get a spurious delocalization of the orbitals over both donor and acceptor leading to a too large overlap S_{AB} of the two diabatic states. One possible way to overcome this is to project the wavefunctions on non-interacting reference states and transform the diabatic Hamiltonian into this basis.
The reference states are constructed from four wavefunctions $\Phi_{D^-}, \Phi_{D^+}, \Phi_{A^+}, \Phi_{A^-}$ describing donor and acceptor in the charge states corresponding to the two diabatic states

$$\begin{aligned} \psi_a^0 &= \Phi_{D^-} \times \Phi_{A^+} \\ \psi_b^0 &= \Phi_{D^+} \times \Phi_{A^-}. \end{aligned} \tag{69}$$

Then we project the adiabatic states - Equ. 68 - onto the reference states

$$\begin{pmatrix} \psi'_a \\ \psi'_b \end{pmatrix} = \begin{pmatrix} \langle\psi_a^{\text{ad}}|\psi_a^0\rangle & \langle\psi_b^{\text{ad}}|\psi_a^0\rangle \\ \langle\psi_a^{\text{ad}}|\psi_b^0\rangle & \langle\psi_b^{\text{ad}}|\psi_b^0\rangle \end{pmatrix} \begin{pmatrix} \psi_a^{\text{ad}} \\ \psi_b^{\text{ad}} \end{pmatrix} = \mathbb{K} \begin{pmatrix} \psi_a^{\text{ad}} \\ \psi_b^{\text{ad}} \end{pmatrix}. \tag{70}$$

These, so called "dressed", states are not yet orthogonal which we rectify using the Löwdin scheme

$$\begin{pmatrix} \psi_1 \\ \psi_2 \end{pmatrix} = \mathbb{S}^{-1/2} \begin{pmatrix} \psi'_a \\ \psi'_b \end{pmatrix}, \tag{71}$$

with the elements of the Löwdin matrix $\mathbb{S}$ defined as $\mathbb{S}_{j,k} = \langle\psi'_j|\psi'_k\rangle$ where $j,k \in a,b$. Combining this with Equ. 70 we can then express the new projected diabatic states in terms of the adiabatic constrained states:

$$\begin{pmatrix} \psi_1 \\ \psi_2 \end{pmatrix} = \mathbb{S}^{-1/2}\mathbb{K} \begin{pmatrix} \psi_a^{\text{ad}} \\ \psi_b^{\text{ad}} \end{pmatrix}. \tag{72}$$

The similarity transformation from the adiabatic Hamiltonian matrix $\mathbb{H}_{\text{ad}}$ to the projected diabatic Hamiltonian $\mathbb{H}^{\text{p}}_{\text{diab}}$ is then simply

$$\mathbb{H}^{\text{p}}_{\text{diab}} = (\mathbb{S}^{-1/2}\mathbb{K})\mathbb{H}_{\text{ad}}(\mathbb{S}^{-1/2}\mathbb{K})^{\text{T}}. \tag{73}$$

11.15 Fragment Orbital DFT (FO-DFT)

The basic idea of FO-DFT, as formulated by Senthilkumar, Grozema, Bickelhaupt, and Siebbeles [174], is to construct the two reference states from calculations of the non-interacting isolated donor and acceptor groups, respectively. This way one avoids spurious delocalization of electrons over both groups, but also neglects polarisation effects of the donor on the acceptor and vice versa. Thus

the diabatic states - and therefore the matrix element H_{AB} - will in general be slightly different from those calculated with CDFT.
With these diabatic reference states the transition matrix element is then in principle given by:

$$H_{AB} = \left\langle \psi^{D}_{\mathrm{HOMO}} \left| \mathcal{H}_{\mathrm{KS}} \right| \psi^{A}_{\mathrm{HOMO}} \right\rangle , \qquad (74)$$

where $\mathcal{H}_{\mathrm{KS}}$ denotes the Kohn-Sham Hamiltonian and $\psi^{D,A}_{\mathrm{HOMO}}$ the highest occupied molecular orbitals (HOMO's) of donor and acceptor, respectively. Here we make the assumption that we can - to a sufficient degree of accuracy - approximate the acceptors LUMO (lowest unoccupied molecular orbital) by its HOMO in a system with one more electron. Successful applications of our implementation of FO-DFT can be found in ref.[173].

11.15.1 FODFT with CPMD

An FO-DFT calculation in CPMD consists of six different calls to CPMD:

1 and 2: the respective wavefunction optimisation of donor and acceptor, on positions as they would be if they were combined (**CENTER MOLECULE OFF**). Note that here the calculated acceptor state should be the desired acceptor state plus one electron, which is necessary because CPMD does not calculate Kohn-Sham matrix elements for unoccupied orbitals. These two calls are the only time consuming ones in the procedure.

3 and 4: the diagonalization of the two states in order to get their respective Kohn-Sham Orbitals (KOHN-SHAM ENERGIES).

5: the combination and orthogonalisation of both wavefunctions with keywords COMBINE WAVEFUNCTION and one of the orthogonalisation schemes (ORTHOGONALIZATION LOWDIN of GRAM-SCHMIDT).

6: the calculation of the Kohn-Sham matrix for the combined and orthogonalised states (keyword KSHAM).

If both donor and acceptor atoms are set the constraint value N_{c} denotes as usual the charge difference between donor and acceptor. If NDON= 0, N_{c} is the desired charge of the acceptor.

TODO's and usage WARNINGS

- WARNING: don't use diagonalization schemes (e.g. Lanczos) during the MD because it breaks the force calculation.
- WARNING: don't use non-orthogonal supercells, only SYMMETRY 0, 1 or 8 should be used.
- WARNING: CDFT and FODFT are not implemented for QM/MM calculations, yet.

11.16 CPMD/Gromos QM/MM Calculations

11.16.1 General Overview

An additional interface code (**MM_Interface** folder) and an adapted classical force field code [123] (**Gromos** folder) are needed to run CPMD in fully Hamiltonian hybrid QM/MM mode[19]. To use this code a *Gromos license* is required and therefore it is **not** included in the standard CPMD code. The interface code and the adapted classical force field code can be obtained by directly contacting the CPMD developers.
To create a makefile for compilation of a QM/MM enabled CPMD binary, you have to copy the two above folders (or create symbolic links to them) in the CPMD source directory and then to add the `-qmmm` flag when executing the `config.sh` script (see section 6). The resulting binary can be used for normal CPMD runs as well as for QM/MM simulations.

11.16.2 Input files for QM/MM CPMD

A QM/MM run requires a modified CPMD input file, some additional input files, and creates the normal CPMD output file and some new ones. The input file consists of a standard CPMD input with with the **QMMM** keyword in the &CPMD section, a modified &ATOMS section and a mandatory &QMMM section. Furthermore three files for the classical code are needed (coordinates, topology and input file). These can be taken from previous fully classical simulations and have to be in Gromos format. Topologies and coordinates files created with the Amber[124] package are also supported. A converter to Gromos format[123] is available.

11.16.3 Starting a QM/MM run

To start a QM/MM simulation, you first do a simulation of your system with a regular classical MD-code to get an equilibrated configuration. The tricky part in this is usually the treatment of (metal-)ion or special molecules, that are not parameterized withing a given force field but are in the active center of your molecule (one of the prominent reasons why you want to do a QM/MM run in the first place). It is usually easiest to keep that part rigid throughout the equilibration, until after you have defined the QM-subsystem.
Starting from the classically equilibrated structure, you have to create a topology, a coordinate and an input file in Gromos format (either by using the Gromos tools or a converter). Now you need to define your QM system by assigning pseudopotentials to selected atoms in your CPMD input file (see 11.16.5).
You can now start to continue the classical equilibration with CPMD using **MOLECULAR DYNAMICS** CLASSICAL. Please note, that there are several special constraints available to ease the transition in case of strong interactions within the QM part or between the QM and the MM part. Finally, a wavefunction optimization (either directly or via **QUENCH** BO) and a normal **MOLECULAR DYNAMICS** CP or BO can be performed.

11.16.4 Defining internal Gromos array dimensions

One rather new feature of this QM/MM interface code is the **ARRAYSIZES ... END ARRAYSIZES** block in the &QMMM section which allows to change the internal array dimensions of the Gromos part dynamically. Previously one had to change some include files and recompile everything to adapt the code for a different system.
These settings have to be consistent during a series of calculations, or else you may not be able to read your restart files correctly.

11.16.5 Defining the QM system

For a QM/MM calculation a subset of atoms are selected from the classical restart and then for this QM part an isolated system (SYMMETRY 0) calculation is performed. The supercell size has to follow the requirements of the various Poisson solvers, as listed in the hints section (11.4).
If not otherwise specified, the QM system (atoms and wavefunction) is always re-centered in the given supercell (the current offset of the QM cell is recorded in the file MM_CELL_TRANS).
The quantum atoms are specified in the &ATOMS section similar to normal CPMD calculations. Instead of explicit coordinates one has to provide the atom index as given in the Gromos topology and coordinates files.

11.16.6 List of keywords in the &QMMM section

Mandatory keywords:

COORDINATES
Section: &QMMM

On the next line the name of a Gromos96 format coordinate file has to be given. Note, that this file must match the corresponding input and topology files. Note, that in case of hydrogen capping, this file has to be modified to also contain the respective dummy hydrogen atoms.

INPUT
Section: &QMMM

On the next line the name of a Gromos input file has to be given. A short summary of the input file syntax and some keywords are in section 11.16.7. Note, that it has to be a correct input file, even though many options do not apply for QM/MM runs.

TOPOLOGY
Section: &QMMM

On the next line the name of a Gromos topology file has to be given. Regardless of the force field, this topology file has to be in Gromos format[123]. Topologies created with Amber can be converted using the respective conversion tools shipped with the interface code. A short summary of the topology file syntax and some keywords are in section 11.16.7.

Other keywords:

ADD_HYDROGEN
Section: &QMMM

This keyword is used to add hydrogens to the QM system if a united atom topology is used (like in Gromos). On the next line the number of atoms to be "hydrogenized" has to be given and in the line following that, the corresponding gromos atom numbers. A number of hydrogens consistent with the hybridization of the "hydrogenized" carbons are added.

AMBER
Section: &QMMM

An Amber functional form for the classical force field is used. In this case coordinates and topology files as obtained by Amber have to be converted in Gromos format just for input/read consistency. This is done with the tool amber2gromos availabe with the CPMD/QMMM package.
This keyword is mutually exclusive with the **GROMOS** keyword (which is used by default).

ARRAYSIZES ... END ARRAYSIZES

Section: &QMMM

Parameters for the dimensions of various internal arrays can be given in this block. The syntax is one label and the according dimension per line. The suitable parameters can be estimated using the script `estimate_gomos_size` bundled with the QM/MM-code distribution. Example:

```
ARRAYSIZES
  MAXATT 20
  MAXAA2 17
  MXEX14 373
END ARRAYSIZES
```

BOX TOLERANCE

Section: &QMMM

The value for the box tolerance is read from the next line. In a QM/MM calculation the size of the QM-box is fixed and the QM-atoms must not come to close to the walls of this box. On top of always recentering the QM-box around the center of the distribution of the atoms, CPMD prints a warning message to the output when the distribution extends too much to fit into the QM-box properly anymore. This value may need to be adjusted to the requirements of the Poisson solver used (see section 11.4).
Default value is 8 a.u.

BOX WALLS

Section: &QMMM

The thickness parameter for soft, reflecting QM-box walls is read from the next line. This keyword allows to reverse the momentum of the particles ($\mathbf{p}_I \rightarrow -\mathbf{p}_I$) when they reach the walls of the simulation supercell similar to the full quantum case, but acting along all the three directions x, y, z. In the case this keyword is used in the &QMMM section,QM particles are reflected back in the QM box. Contrary to the normal procedure of re-centering the QM-box, a soft, reflecting confinement potential is applied if atoms come too close to the border of the QM box [175]. It is highly recommended to also use **SUBTRACT** COMVEL in combination with this feature.
NOTE: to have your QM-box properly centered, it is best to run a short MD with this feature turned off and then start from the resulting restart with the soft walls turned on. Since the reflecting walls reverse the sign of the velocities, $\mathbf{p}_I \rightarrow -\mathbf{p}_I$ (I = QM atom index), be aware that this options affects the momentum conservation in your QM subsystem.
This feature is **disabled by default**

CAPPING

Section: &QMMM

Add (dummy) hydrogen atoms to the QM-system to saturate dangling bonds when cutting between MM- and QM-system. This needs a special pseudopotential entry in the &ATOMS section (see section 11.16.9 for more details).

CAP_HYDROGEN
Section: &QMMM

same as **CAPPING**.

ELECTROSTATIC COUPLING [LONG RANGE]
Section: &QMMM

The electrostatic interaction of the quantum system with the classical system is explicitly kept into account for all classical atoms at a distance $r \leq$ **RCUT_NN** from any quantum atom and for all the MM atoms at a distance of **RCUT_NN** $< r \leq$ **RCUT_MIX** and a charge larger than $0.1e_0$ (NN atoms).

MM-atoms with a charge smaller than $0.1e_0$ and a distance of **RCUT_NN** $<$ $r \leq$ **RCUT_MIX** and all MM-atoms with **RCUT_MIX** $< r \leq$ **RCUT_ESP** are coupled to the QM system by a ESP coupling Hamiltonian (EC atoms).

If the additional `LONG RANGE` keyword is specified, the interaction of the QM-system with the rest of the classical atoms is explicitly kept into account via interacting with a multipole expansion for the QM-system up to quadrupolar order. A file named `MULTIPOLE` is produced.
If `LONG RANGE` is omitted the quantum system is coupled to the classical atoms not in the NN-area and in the EC-area list via the force-field charges.

If the keyword `ELECTROSTATIC COUPLING` is omitted, all classical atoms are coupled to the quantum system by the force-field charges (mechanical coupling).

The files INTERACTING.pdb, TRAJECTORY_INTERACTING, MOVIE_INTERACTING, TRAJ_INT.dcd, and ESP (or some of them) are created. The list of NN and EC atoms is updated every 100 MD steps. This can be changed using the keyword **UPDATE LIST**.

The default values for the cut-offs are RCUT_NN=RCUT_MIX=RCUT_ESP=10 a.u.. These values can be changed by the keywords **RCUT_NN**, **RCUT_MIX**, and **RCUT_ESP** with $r_{nn} \leq r_{mix} \leq r_{esp}$.

ESPWEIGHT
Section: &QMMM

The ESP-charg fit weighting parameter is read from the next line.
Default value is $0.1e_0$.

EXCLUSION {GROMOS,LIST{NORESP}}
Section: &QMMM

Specify charge interactions that should be excluded from the QM/MM Hamiltonian. With the additional flag GROMOS, the exclusions from the Gromos topology are used. With the additional flag LIST, an explicit list is read from following lines. The format of that list has the number of exclusions in the first line and then the exclusions listed in pairs of numbers of the QM atom and the MM atom in Gromos ordering; the optional flag NORESP in this case requests usage of MM point charges for the QM atoms instead of the D-RESP charges (default).

FLEXIBLE WATER [ALL,BONDTYPE]
Section: &QMMM

Convert some solven water molecules into solute molecules and thus using a flexible potential.
With the BONDTYPE flag, the three bond potentials (OH1, OH2, and H1H2) can be given as index in the BONDTYPE section of the Gromos topology file. Note that the **non-bonded** parameters are taken from the SOLVENATOM section of the **TOPOLOGY** file. **Default** is to use the values: 35, 35, 41.
With the additional flag ALL this applies to all solvent water molecules, otherwise on the next line the number of flexible water molecules has to be given with the Gromos index numbers of their respective Oxygen atoms on the following line(s).
On successful conversion a new, adapted topology file, MM_TOPOLOGY, is written that has to be used with the **TOPOLOGY** keyword for subsequent restarts. Also the **INPUT** file has to be adapted: in the SYSTEM section the number of solvent molecules has to be reduced by the number of converted molecules, and in the SUBMOLECULES section the new solute atoms have to be added accordingly.
Example:

```
FLEXIBLE WATER BONDTYPE
 4 4 5
 26
   32   101   188   284   308   359   407   476   506   680
  764   779   926  1082  1175  1247  1337  1355  1607  1943
 1958  1985  2066  2111  2153  2273
```

FORCEMATCH ... END FORCEMATCH
Section: &QMMM

Input block for the QM/MM forcematching. A general description is given in section 11.16.11.

READ REF FORCES [FILE,COVALENT]
Flag to read the QM/MM reference forces directly from the file FM_REF_FORCES, i.e. no QM/MM SPs are computed. **Default**: false. An alternative file name can be specified on the next line with the option **FILE**. With the option **COVALENT** covalent forces are read from the file FM_REF_COVFORCES.

READ REF TRAJ [FILE]
Read reference trajectory from file TRAJECTORY_REF (or set the **FILE** option to read a non-default file name from the next line) with a given stride and compute single points on the respective frames.

RESTART SP
If in a previous force matching run not all of the SPs could be computed (e.g. limited wall time) this flag indicates cpmd to restart the SP calculations. The FM_REF* files from the previous run have to be present and they will be appended. With this option make sure that the frames contained in the already existing FM_REF* files are consistent. **Default**: false.

READ REF STRIDE
Stride to apply when reading the TRAJECTORY_REF file is read from the next line. Default=1, i.e. every frame is used for the SP calculations.

TOPOL OUT
Filename for the final topology file. **Default**: FMATCH.top.

INITWF [OFF]
Generate an initial guess for the wfkt for the SP calculations based on AOs (default). With the **OFF** option the wfkt of the previous frame is used as an intial guess.

CHARGES [ONLY,NO],[FIX]
Charge fitting is on by default and can be switched off with the **NO** option. In this case the charges from the initial topology will not be modified. **ONLY** will let the program stop after the charge fitting and the other parameters are not updated.
With the **FIX** option target values for the restraints in the charge fitting on specific atoms can be specified by the user. Usually the charges are restraint to the respective Hirshfeld values. On the next line the number of charges to be fixed has to be given and then the corresponding number of lines with: gromos index   charge.

MV
Weight on the potential in the charge fitting. **Default**=0.1.

WF
Weight on the field in the charge fitting. **Default**=0.0.

WQ INDIVIDUAL
Weights on the charge restraints can be given individually here. From the next line the total number of individual weights is read. Then the lines with: gromos index weight.

WQ GENERAL
The weight for all the charge restraints that were not specified by individual weights can be given on the next line. **Default**=0.1.

WTOT
Weight of the total charge contribution in the charge fitting. **Default**=1.0E7.

EQUIV
Specify equivalent atoms. Syntax:

```
EQUIV
n_equiv
atom1   atom4
atom1   atom3
...
atom5   atom7
```

There are ***n_equiv*** equivalencies specified (`n_equiv` lines are read from the input). For each pair of equivalencies the gromos indexes have to be specified on one separate line. The lower index has to be given first!
If an atom is equivalent to more then one other atom. E.g. atom1,atom3 and atom4 are equivalent. Then this has to be encoded by:

```
atom1   atom3
atom1   atom4
```

and not by:

```
atom1   atom4
atom3   atom4
```

where atom1 has a lower gromos index then atom3 and atom3 has a lower one then atom4. Per default no equivalencies are assumed.
OPT FC ONLY
Serves as a flag to remove the equilibrium values of the bonded interactions from the list of fitted parameters. I. e. only force constants are fitted for the bonded interactions.
NO BONDS
Do not fit bonds. **Default**=false.
NO ANGLES
Do not fit angles. **Default**=false.
NO DIHEDRALS
Do not fit dihedrals. **Default**=false.
NO IMPROPERS
Do not fit improper dihedrals. **Default**=false.
MAXITER
Give on the next line the maximal number of iterations for the non-linear fitting procedure of the bonded interactions. **Default**=500.
COMPUTE RMS [NO]
Per default the RMS on the forces is computed after the fitting has been completed. Switch it off with the **NO** option. Example:

```
FORCEMATCH
  READ REF TRAJ FILE
    TRAJECTORY_REF
  READ REF STRIDE
    10
  WV
  1.0
  WOT
  1000000.0
  WQ GENERAL
```

```
  0.1
END FORCEMATCH
```

GROMOS

Section: &QMMM

A Gromos functional form for the classical force field is used (this is the default). This keyword is mutually exclusive with the **AMBER** keyword.

HIRSHFELD [ON,OFF]

Section: &QMMM

With this option, restraints to Hirshfeld charges [63] can be turned on or off
Default value is ON.

MAXNN

Section: &QMMM

Then maximum number of NN atoms, i.e. the number of atoms coupled to the QM system via **ELECTROSTATIC COUPLING** is read from the next line. (Note: This keyword was renamed from MAXNAT in older versions of the QM/MM interface code to avoid confusion with the MAXNAT keyword in the **ARRAYSIZES ... END ARRAYSIZES** block.)
Default value is 5000.

NOSPLIT

Section: &QMMM

If the program is run on more than one node, the MM forces calculation is performed on all nodes. Since the MM part is not parallelized, this is mostly useful for systems with a small MM-part and for runs using only very few nodes. Usually the QM part of the calculation needs the bulk of the cpu-time in the QM/MM.
This setting is the default. See also under **SPLIT**.

RCUT_NN

Section: &QMMM

The cutoff distance for atoms in the nearest neighbor region from the QM-system ($r \leq r_{nn}$) is read from the next line. (see **ELECTROSTATIC COUPLING** for more details).
Default value is 10 a.u.

RCUT_MIX
Section: &QMMM

The cutoff distance for atoms in the intermediate region ($r_{nn} < r \leq r_{mix}$) is read from the next line. (see **ELECTROSTATIC COUPLING** for more details).
Default value is 10 a.u.

RCUT_ESP
Section: &QMMM

The cutoff distance for atoms in the ESP-area ($r_{mix} < r \leq r_{esp}$) is read from the next line. (see **ELECTROSTATIC COUPLING** for more details).
Default value is 10 a.u.

RESTART TRAJECTORY [FRAME {num},FILE '{fname}',REVERSE]
Section: &QMMM

Restart the MD with coordinates and velocities from a previous run. With the additional flag FRAME followed by the frame number the trajectory frame can be selected. With the flag FILE followed by the name of the trajectory file, the filename can be set (Default is TRAJECTORY). Finally the flag REVERSE will reverse the sign of the velocities, so the system will move backwards from the selected point in the trajectory.

SAMPLE INTERACTING [OFF,DCD]
Section: &QMMM

The sampling rate for writing a trajectory of the interacting subsystem is read from the next line. With the additional keyword OFF or a sampling rate of 0, those trajectories are not written. The coordinates of the atoms atoms contained in the file INTERACTING.pdb are written, in the same order, on the file TRAJECTORY_INTERACTING every. If the **MOVIE** output is turned on, a file MOVIE_INTERACTING is written as well. With the additional keyword DCD the file TRAJ_INT.dcd is also written to. if the sampling rate is negative, then **only** the TRAJ_INT.dcd is written.
Default value is 5 for MD calculations and OFF for others.

SPLIT
Section: &QMMM

If the program is run on more than one node, the MM forces calculation is performed on a separate node. This is mostly useful for systems with a large MM-part and runs with many nodes where the accumulated time used for the classical part has a larger impact on the performance than losing one node for the (in total) much more time consuming QM-part.
Default is **NOSPLIT**.

TIMINGS
Section: &QMMM

Display timing information about the various parts of the QM/MM interface code in the output file. Also a file `TIMINGS` with even more details is written. This option is off by **default**.

UPDATE LIST
Section: &QMMM

On the next line the number of MD steps between updates of the various lists of atoms for **ELECTROSTATIC COUPLING** is given. At every list update a file INTERACTING_NEW.pdb is created (and overwritten).

Default value is 100.

VERBOSE
Section: &QMMM

The progress of the QM/MM simulation is reported more verbosely in the output. This option is off by **default**.

WRITE LOCALTEMP [STEP {nfi_lt}]
Section: &QMMM

The Temperatures of the QM subsystem, the MM solute (without the QM atoms) and the solvent (if present) are calculated separately and written to the standard output and a file `QM_TEMP`. The file has 5 columns containing the QM temperature, the MM temperature, the solvent temperature (or 0.0 if the solvent is part of the solute), and the total temperature in that order. With the optional parameters STEP followed by an integer, this is done only every **`nfi_lt`** timesteps.

11.16.7 Keywords in the Gromos Input and Topology files

For a detailed description of the Gromos file formats please have a look at the Gromos documentation[123]. Note, that not all keyword are actually active in QM/MM simulations, but the files still have to be syntactically correct. Both, the input and the topology file are structured in sections starting with a keyword in the first column and ending with the keyword END. Lines starting with a pound sign '#' may contain comments and are ignored. Both files are required to have a TITLE section as the first section. The rest can be in almost any order. Here is a short list of some important flags and their meaning.

Gromos Input File:

TITLE
Text that identifies this input file. Will be copied into the CPMD output.

SYSTEM
This section contains two integer numbers. The first is the number of (identical) solute molecules (NPM) and the second the number of (identical) solvent molecules (NSM).

BOUNDARY

This section defines the classical simulation cell. It contains 6 numbers. The first (NTB) defines the type of boundary conditions (NTB < 0 means truncated octahedron boundary conditions, NTB=0 vacuum, and NTB > 0 rectangular boundary conditions). The next three numbers (BOX(1..3)) define the size of the classical cell. The fifth number (BETA) is the angle between the x- and z-axes and the last number usually determines whether the cell dimensions are taken from the input file (NRDBOX=0) or from the BOX section of the **COORDINATES** file (NRDBOX=1), but is ignored for QM/MM simulations.

Note: that even for vacuum simulations valid simulation cell sizes must be provided.

PRINT

This section determines how often some properties are monitored. Here only the first number (NTPR) matters, as it determines the number of MD steps between printing the various energies to the CPMD output.

Note many old Gromos input files created by the amber2gromos program default to NTPR=1, which makes the CPMD output huge.

SUBMOLECULES

Defines number of submolecules in the solute. The first number is the number of submolecules followed by the index number of the last atom of each submolecule. The last number must be identical to the number of atoms in the solute.

FORCE

Contains two groups of numbers, that controls the various force component and the partitioning of the resulting energies. The first group of 1/0 flags turn the various force components on or off. The second group defines energy groups (the first number is the number of groups followed by the index number of the last atom in each group). The last number must be identical to the number of all atoms.

Gromos Topology File:

TITLE

Text that identifies this topology file. Will be copied into the CPMD output.

ATOMTYPENAME

This section contains the number of classical atom types (NRATT) followed by the respective labels, one per line. Note that the **ARRAYSIZES ... END ARRAYSIZES** MAXATT must be large enough to accomodate all defined atom types.

RESNAME

This section contains the number of residues in the solute (NRAA2) followed by the respective residue names.

SOLUTEATOM

This section defines the number (NRP) and sequence of atoms in the solute, their names, residue numbers, non-nonded interaction codes, masses, charges, charge groups and their full + scaled 1-4 exclusions.

BONDTYPE

This section contains the list of parameters for bonded interactions. You have to pick the two matching entries from this list for the O-H and H-H potential, when using the **FLEXIBLE WATER** keyword to convert solvent water back into solute (e.g. to included them into the QM part).

SOLVENTATOM

This section defines the number of atoms (NRAM) in the solvent and their respective names, non-bonded interactions types, masses, and charges.

SOLVENCONTSTR

This section defines the number (NCONS) and parameters for the distance constraints, that are used to keep the solvent rigid.

11.16.8 Files generated by the interface code

- **QMMM_ORDER**
  The first line specifies the total number of atoms (NAT) and the number of quantum atoms (NATQ). The subsequent NAT lines contain, for every atom, the gromos atom number, the internal CPMD atom number, the CP species number isp and the number in the list of atoms for this species NA(isp). The quantum atoms are specified in the first NATQ lines.
- **CRD_INI.grm**
  Contains the positions of all atoms in the first frame of the simulation in Gromos extended format.
- **CRD_FIN.grm**
  Contains the positions of all atoms in the last frame of the simulation in Gromos extended format.
- **INTERACTING.pdb**
  Contains (in pdb format) all the QM atoms and all the MM atoms in the electrostatic coupling NN list. The 5-th column in this file specifies the gromos atom number as defined in the topology file and in the coordinates file. The 10-th column specifies the CPMD atom number as in the TRAJECTORY file. The quantum atoms are labelled by the residue name QUA.
- **INTERACTING_NEW.pdb**
  The same as INTERACTING.pdb, but it is created if the file INTERACTING.pdb is detected in the current working directory of the CPMD run.
- **TRAJECTORY_INTERACTING**
  Contains the coordinates and the velocities (in TRAJECTORY format) of the atoms listed in INTERACTING.pdb. The format is the same as in the files TRAJECTORY and MOVIE, hence frames belonging to different runs are separated by the line ¡¡¡¡ NEW DATA ¿¿¿¿¿.

  The atoms in this file do not necessarily coincide with the NN atoms, that are written at every update of the pair list in the file INTERACTING_NEW.pdb.
- **MOVIE_INTERACTING**
  The MOVIE-like file corresponding to TRAJECTORY_INTERACTING.
- **ESP**
  Contains the ESP charges of the QM atoms in CPMD order (the corresponding Gromos numbers can be found in QMMM_ORDER). The first column is the frame number.
- **EL_ENERGY**
  Contains the electrostatic interaction energy. First column: frame number. Second column: total electrostatic interaction energy. Other columns: interaction energy of the NN atoms with the QM system; interaction energy with the ESP coupled atoms; multipolar interaction energy; electrostatic interacion energy evaluated using the classical force field charges for the QM atoms.
- **MULTIPOLE**
  Contains, for every frame (specified in the first column), the three components of the dipol D(ix) and the five independent components of the quadrupole Q(ix,jx) of the quantum system in a.u. The order is: D(1),D(2),D(3),Q(1,1),Q(2,2),Q(1,2),Q(1,3),Q(2,3).

- **MM_CELL_TRANS**
  Contains, the Trajectory of the re-centering offset for the QM-box. The first column is the frame number (NFI) followed by the x-, y-, and z-component of the cell-shift vector.

11.16.9 Hydrogen Capping vs. Link Atoms

Whenever the QM/MM-boundary cuts through an existing bond, special care has to be taken to make sure that the electronic structure of the QM-subsystem is a good representation of an all-QM calculation and also the structure in the boundary region is preserved. So far, two methods methods are available to do this: using special link-atom pseudopotentials and hydrogen capping.

Link Atom

The simplest way is to use a link-atom pseudopotential. In the simplest case, this would be a scaled down pseudopotential with the required valence change (e.g. ZV=1 when cutting through a single carbon-carbon bond). However in this case it is required to constrain the distance between the link atom and the (full-QM) neighbor atom to the (full-QM) equilibrium distance, to preserve the electronic structure in the center of the QM subsystem. You should be aware of the fact, that this is a rather crude approximation and that the constraint will create a small imbalance in the forces between the QM and MM subsystems, that can result in a drift in the total energy, if the length of the constraint is badly chosen.
A more rigorous approach would be to use an optimized pseudopotential constructed with the method described in ref. [176], that should take care of the need for the constraint.

Hydrogen Capping

An alternative way would be to use the **CAPPING** flag in order to introduce additional (dummy) hydrogen atoms to saturate the dangling bonds. These capping hydrogen atoms have to be hidden from the MM hamiltonian so the Gromos **INPUT** and **TOPOLOGY** files have to be modified for subsequent runs, in addition to adding an explicit **EXCLUSION** LIST to the cpmd input. The whole procedure is a bit complicated so here is a short protocol of the required steps.

1a Set up a normal QM/MM run as for using link atoms with the additional keyword **CAPPING** and instead of the link-atom potential use a hydrogen potential with the additional flag `ADD_H`. Note that you have to provide the correct number of hydrogens, but no atom index number.

1b Run a short MD (a couple of steps) and use the resulting `CRD_FIN.grm` file (under a different name) in the **COORDINATES** section.

2a Modify the Gromos input file to match the new coordinate file.

  – Increase the number of atoms per solute molecule in the SUBMOLECULES section.
  – increase the total number of atoms in the FORCE section.

2b Modify the Gromos topology file to match the new coordinate file.

  – Add a DUM atom type at the end of the ATOMTYPENAME section if not already present (and increase NRATT accordingly).
  – if you have added a new atom type, you have to add the corresponding entries in the LJPARAMETERS section as well. Since the capping hydrogen atoms should be invisible from the MM Hamiltonian all Lennard-Jones parameters are set to 0.0 for those new entries. This section is a triangular matrix, so you have to add NRATT lines (and increase NRATT2 accordingly).
  – Add new residues named DUM (one for each capping hydrogen) to the RESNAME section (and increase NRAA2).
  – In the SOLUTEATOM section you have to increase NRP and add the dummy hydrogens at the end of the solute. The structure of the entry is: `<atom nr> <residue nr> <atom type name> <vdw type index> <mass> <charge> 1 0` and a single '0' on the next line. Use a mass of 1.008 and a charge of 0.000.

2c Modify the CPMD input file.

- Make sure that the **TOPOLOGY**, **INPUT**, and **COORDINATES** keywords in the &QMMM section match the newly created or modified files.
- Add the capping hydrogens to the &ATOMS section as normal QM atoms, but add the DUMMY flag to the pseudopotential line.
- Build an **EXCLUSION** LIST entry that lists for each capping atom the respective QM/MM atoms pairs that should be excluded from the electrostatic coupling (all other MM interactions are set to zero already in the topology file). For consistency only full charge groups should be excluded. In the supplementary material should be a script genexcl.tcl which can help you in building that list (it needs the modified Gromos coordinate and topology file as well as the QMMM_ORDER file as input).
- Update the **ARRAYSIZES ... END ARRAYSIZES** entry to match the new topology.

With the three modified files you should be able to run a regular QM/MM run. Note, that you may have to update the exclusion list occasionally, depending on your system and that you should pick the bond(s) to cut very carefully.

11.16.10 What type of QM/MM calculations are available?

The QM/MM interface only supports a subset of the functionality of CPMD. Please note, that although there are some tests and warnings included into the source code, not every job that runs without a warning will be automatically correct. So far, the interface code requires the use of norm-conserving pseudopotentials. Tested and supported job types are: **MOLECULAR DYNAMICS** (CLASSICAL, CP and BO), **OPTIMIZE WAVEFUNCTION**, **KOHN-SHAM ENERGIES**, and **ELECTRONIC SPECTRA**. Supported are closed shell systems as well as **LSD** and **LOW SPIN EXCITATION** calculations.
OPTIMIZE GEOMETRY is experimental and currently supports optimization of the QM atom positions only. Use of the linear scaling geometry optimizer (**LBFGS**) is highly recommended and the currently also the default.
PROPERTIES calculations with QM/MM are experimental. Most properties (WF projection, population analysis, localization) that only need the plain QM wavefunction work.
LINEAR RESPONSE calculations are currently at an twofold experimental status. Both, the isolated system setup (**SYMMETRY** 0) and the QM/MM coupling of the response calculations itself are not yet fully tested.
Options that are known to be incompatible with **QMMM** are **VIBRATIONAL ANALYSIS**, **PATH INTEGRAL**, and all calculations that require a wavefunction optimization via a diagonalization method at some point.

11.16.11 QM/MM Force Matching

This tool allows the automated (re)parametrization of classical force fields from QM/MM reference calculations via a force matching protocol as published in [177]. Thereby only MM parameters among the atoms comprised in the QM subsystem are reparametrized. In this first release VdW parameters are excluded from the optimization and kept constant. Fitting of these parameters will be a feature of a future release.
The jobs requires a QM/MM reference TRAJECTORY file and the corresponding gromos topology, input and coordinate files and the cpmd input that were used to generate the reference TRAJECTORY file. The initial topology can be a reasonable guess and will be refined during the actual force matching procedure. Currently the QM/MM reference trajectory has to be generated prior to the force matching job.
The actual forcematching job is invoked by the **FORCEMATCH** in &CPMD and the **FORCEMATCH ... END FORCEMATCH** block in &QMMM. Besides, the &CPMD and &QMMM sections should contain sensible keywords and parameters for high quality reference forces (e.g.

convergence orbitals 1.0d-7).
The parametrization protocol consists of a three-step process: First, the reference trajectory is read with a given stride. On each of the selected frames QM/MM reference forces (BO) are calculated. The forces on the atoms of the QM subsystem are stored (FM_REF_FORCES) along with the Hirshfeld charges (FM_REF_CHJ) as well as the electrostatic potential and field on the nearby MM atoms (FM_REF_PIP). Second, a set of atomic point charges that reproduce the electrostatic potential and forces that the QM system exerts on the surrounding classical atoms is derived. Third, the nonbonded contributions, computed with the charges obtained in the second step and given Lennard-Jones parameters, are subtracted from the total reference forces on the QM atoms. The remaining forces are assumed to be derived from bonded interactions. The parameters for bonded interactions (torsions, bending and bonds) are thus adjusted in order to reproduce the remaining forces. See reference [177] for details. An updated topology file FMATCH.top is written at the end of the run. To check the quality of the fitting procedure a section with the absolute and relative force RMS per atom is printed at the end to standard output.
Files generated by the force matching code (Some of the following default filenames can be changed via the respective keywords in the FORCEMATCH block):

- **FMATCH.top**
  Updated topology file at the end of the job.
- **FM_REF_CHJ**
  Hirshfeld charges on the QM atoms from the reference force calculations. Format: Two lines per frame. First line contains frame index from the original reference trajectory file. Second line gives the Hirshfeld charges on the QM atoms in cpmd ordering.
- **FM_REF_PIP**
  Electrostatic potential and field on the NN atoms from the reference force calculations.
- **FM_REF_FORCES**
  For each frame extracted from the reference TRAJECTORY file and for which QM/MM forces were calculated, the QM/MM forces on the QM atoms are dumped into this file. One line per frame with the original frame index and the number of QM atoms. Then for each QM atom in cpmd ordering:
  atom index,x,y,z,fx,fy,fz

All force matching related information written to standard output are labeled with ' fm '. After the initialization the reference trajectory file is parsed and the line 'fm extracting total number of frames for SPs' is printed. For each frame you should find the following lines: 'frame number', 'computing reference forces' and 'Total nr. of iterations:'. The beginning of the second part of the force matching protocol is marked with the line 'Reading values for charge fitting from file'. At the end the RMS deviation of the charges, electrostatic potential and field are printed. The third part starts with 'Will now loop over reference frames again' to compute the non-bonded interactions. After 'Done with classical loop' the covalent parameters are fitted and you can monitor the change of the absolute and relative RMS deviation from the reference covalent forces during the optimization. 'Optimization successful' indicates the end of the fitting of the covalent parameters, FMATCH.top is written and, finally, the total (non-bonded plus covalent) forces are calculated with the updated topology to get the RMS deviation of the total force. The force matching related output ends with a block containing 'computing RMS per atom'.

11.17 Gromacs/CPMD QM/MM Calculations

As of version 3.3 the Gromacs[104] classical MD code contains a generic API for QM/MM calculations. So far this API has been used to QM/MM interfaces for GAMESS, Gaussian, and . . . CPMD. Unlike in the Gromos/CPMD QM/MM interface code (see above) the main MD driver is in the classical code. The Gromacs/CPMD interface code is based on the EGO/CPMD interface and was developed by Pradip Kumar Biswas in the group of Valentin Gogonea at Cleveland State University.

For additional information and downloads see http://comppsi.csuohio.edu/groups/qmmm.html, and the respective publication [33].

11.17.1 Technical Introduction

The whole interface code is divided into two parts: one part is embedded in the Gromacs code and the other in CPMD. Both, the modified Gromacs and the CPMD codes are compiled independently and communicate via files in the current working directory.
Since the Gromacs code acts as the driver, you first have to set up a regular Gromacs classical MD simulation in the usual way by building/providing a .pdf/.gro file and a .top file. Before running grompp, you also need to create an index file (usually named index.ndx) that lists the atoms of the QM subsystem and provide further parameters for the CPMD calculation like the size of QM-simulation box, plane-wave cutoff for CPMD, Coulomb cutoff, if any, etc (for details, see the rgmx script in the QM/MM examples).
During mdrun, the interface is controlled by two function calls:
a) init_cpmd() prepares the ground for the QM/MM interface. It sets the flags for the QM and MM atoms finds LINK atoms from the topology and prepares temporary structures to process the QM/MM data etc.
b) call_cpmd() first creates the CPMD input file "CPMD_inp.run" using a template "CPMD_inp.tmpl" and then kickstarts the CPMD code via a "fork/exec" or "system" call. The interface is set to use "fork". If system call is preferred, you need to set the defined variable NOFORK to 1. call_cpmd() gets forces and energy from CPMD and appends them to Gromacs structures. Gromacs then moves the atoms and while evaluating the forces, calls CPMD again (this is the QM/MM loop). Thus this interface essentially performs a QM/MM Born-Oppenheimer MD simulation.

11.17.2 Compilation of Gromacs

In its present state, you need to use CFLAGS = -DGMX_QMMM_CPMD in configure to include the CPMD interface code into Gromacs. In the adapted Gromacs package, a script "build" is provided in the gromacs folder that takes care of the QM/MM configuration and compilation. Please adapt as needed.

11.17.3 Execution of QM/MM runs

It is like running Gromacs with the additional needs are given by:
a) having an index.ndx file specifying the QM atoms (see example index.ndx file). You can create a usual Gromacs index.ndx file and then append to it the QM group.
b) specifying other QM informations like planewave cutoff, qmbox size etc in the grompp setup (for the mdp file).
c) having a CPMD input file template "CPMD_inp.tmpl" where essential keywords for CPMD run need to be mentioned. "INTERFACE GMX" is essential for QMMM; it ensures a single-point calculation inside CPMD each time it is invoked. Inside the interface, all the QM & MM atoms are translated in such a way that the QM system be at the center of the QM box. Thus the keyword "MOLECULE CENTER OFF" is required to avoid any further movements of the QM atoms.
Right now the **ODIIS** minimizer and **PCG** minimizer (including PCG MINIMIZE) are allowed to be used inside CPMD. There also is a hybrid scheme where for the MD first step it will use the "PCG MINIMIZE" but for all subsequent steps it will use the faster ODIIS minimizer.). Other sections of CPMD input structures need to be kept as usual though the final values for the CELL size and CUTOFF will be those provided by you in the mdp file.

11.17.4 QM/MM Examples

Example inputs for a H_2O-dimer and an ethane molecule are bundled with the modified gromacs distribution from.

http://comppsi.csuohio.edu/groups/qmmm.html,

11.18 QM/(P)MM Interface to IPHIGENIE

The interface of CPMD to the PMM-MD program IPHIGENIE[185, 186] (https://sourceforge.net/projects/iphigenie) supersedes the CPMD interface of Eichinger et al. [16] to the MD program EGO.[102] It provides a Hamiltonian coupling to polarizable MM force fields (PMM). Here, similar to the Eichinger implementation and the derived CPMD/Gromacs coupling[33], CPMD is used only to compute energies and forces but the propagation of the equations of motion is done externally by IPHIGENIE.
The implementation is based on a single executable, which links CPMD and the interface routines as libraries to the IPHIGENIE MD executable 'iffi'. For CPMD-3.17 a patch is provided on `www.cpmd.org` with example configurations for the compilation. For CPMD-4.0 and later the interface to IPHIGENIE is provided as a module. Here, using the configure.sh script with the option `-iphigenie` generates a makefile which builds the interface library along with a regular cpmd.x executable.
For further instructions on compiling and example inputs for QM/PMM runs please consult the main iphigenie page https://sourceforge.net/projects/iphigenie and the associated Wiki pages.

11.19 CPMD on parallel computers

There are three different parallel strategies implemented in CPMD. The actual strategy has to be chosen at compile time.

- Shared memory parallelization

  This strategy uses the OpenMP library. The code is compiled *without* the **PARALLEL** preprocessor flag and compilation and linking need the corresponding OpenMP flags (dependent on compiler).
  Depending on the overhead of the OpenMP system implementation good speedups can be achieved for small numbers of processors (typically 4 to 8). The advantages of this version of the code are small additional memory usage and it can be used in non-dedicated CPU environments.

- Distributed memory parallelization

  This is the standard parallelization scheme used in CPMD based on MPI message passing library.
  The single processor version of this code typically shows an overhead of ca. 10% with respect to the optimal serial code. This overhead is due to additional copy and sort operations during the FFTs.
  All the basic system data and many matrices of the size of the number of electrons are replicated on all processors. This leads to considerable additional memory usage (calculated as the sum over the memory of all processors compared to the memory needed on a single processor). For large systems distributed over many processors the replicated data can dominate the memory usage.
  The efficiency of the parallelization depends on the calculated system (e.g. cutoff and number of electrons) and the hardware platform, mostly latency and bandwidth of the communication system. The most important bottleneck in the distributed memory parallelization of CPMD is the load-balancing problem in the FFT. The real space grids are distributed over the first dimension alone (see line `REAL SPACE MESH:` in the output. As the mesh sizes only vary between 20 (very small systems, low cutoffs) and 300 (large systems, high cutoff) we have a rather coarse grain parallelization. To avoid load unbalance the number of processors should be a divisor of the mesh size. It is therefore clear that even for large systems no speedup can be achieved beyond 300 processors. A solution to this problem is provided

with the keyword **CP GROUPS** This technique, together with optimal mapping, allow to scale from thousands to millions of cores on modern supercomputers such as IBM BG/Q (in particular when running HFX calculations). To learn more about the distributed memory parallelization of CPMD consult D. Marx and J. Hutter, "Modern Methods and Algorithms of Quantum Chemistry", Forschungszentrum Jülich, NIC Series, Vol. 1 (2000), 301-449. For recent developments and for a perspective see http://www.cpmd.org.

When selecting **NSTBLK** for **BLOCKSIZE STATES** it is important to take into account the granularity of the problem at hand. For example, in cases where the number of **STATES** is smaller than the total number of the available processors, one must choose a value for **NSTBLK** such that only a subgroup of the processors participate in the distributed linear algebra calculations. The same argument is also relevant when the number of **STATES** is only moderately larger than the number of processors.

- Mixed shared/distributed memory parallelization

  The two parallelization schemes described above are implemented in such a way that they don't interfere. Therefore it is easy to combine them if this is supported by hardware (shared/distributed memory architecture) and software (libraries). Since the MPI parallelization is very efficient for a small to medium number of nodes and all modern MPI libraries are able to take advantage from shared memory communication, using the mixed shared/distributed memory parallelization is of most use if you run a job on a large number of SMP nodes, when the distributed memory parallelization has reached its scalability limit (see above). To learn more about the mixed parallelization scheme of CPMD consult [35].

As with all general statements, these are only guidelines. The only way to get reliable information is to run benchmarks with the system you want to calculate.

11.20 Using the new xc driver and assembling functionals

The new **XC_DRIVER** offers more flexibility compared to **OLDCODE** and **NEWCODE**. The minimal setup for the use of, *e.g.*, the BLYP xc functional:

```
&DFT
 XC_DRIVER
 FUNCTIONAL GGA_XC_BLYP
&END
```

is equivalent to specifying

```
&DFT
 XC_DRIVER
 FUNCTIONAL GGA_X_BLYP GGA_C_LYP
&END
```

or, in a more verbose way,

```
&DFT
 XC_DRIVER
 FUNCTIONAL GGA_X_BLYP GGA_C_LYP
 SCALES     1.0        1.0
 LIBRARY    CP         CP
&END
```

If not explicitly stated, the xc driver will always use the internal CP functional library. The use of libxc functionals can be requested by using the keyword **LIBRARY**, and both libxc and internal CP functionals may be used simultaneously. The following example combines a particular parametrisation of VWN available in libxc with Slater exchange computed within the internal CP library.

```
&DFT
 XC_DRIVER
 FUNCTIONAL LDA_X LDA_C_VWN_3
 LIBRARY    CP    LIBXC
&END
```

If not specified, all functionals will contribute equally to the total xc energy and the potential. A scaling factor for every functional can be introduced by using **SCALES**. A manual setup equivalent to HYB_GGA_XC_CAM_B3LYP, but using different attenuation parameters $\alpha = 0.19, \beta = 0.66, \mu = 0.330$ would read:

```
&DFT
 XC_DRIVER
 FUNCTIONAL GGA_X_B88 GGA_C_LYP LDA_C_VWN
 SCALES     1.00      0.81      0.19
 COULOMB ATTENUATION
 0.190  0.660  0.330
&END
```

It is also possible to use a linear response kernel different from the one used to generate the ground-state orbitals. In this case, duplicates of the aforementionned exist with the prefix **KERNEL_**. Note that is currently not possible to combine different forms of the Coulomb operator between the kernel and the ground-state functional (*i.e.* it is not possible to use the combination of B3LYP and CAM-B3LYP; however, combinations between GGA and any (screened) hybrid are possible).

```
&DFT
 XC_DRIVER
 FUNCTIONAL       GGA_X_B88 GGA_C_LYP LDA_C_VWN_3
 LIBRARY          CP        CP        LIBXC
 SCALES           0.80      0.81      0.19
 HFX_SCALE        0.20
 LR_KERNEL        GGA_X_B88 GGA_C_LYP LDA_C_VWN
 KERNEL_LIBRARY   CP        CP        CP
 KERNEL_SCALES    0.75      0.81      0.19
 KERNEL_HFX_SCALE 0.25
&END
```

12 Questions and Answers

The following section is a slightly edited collection of questions and answers from the cpmd mailing list, cpmd-list@cpmd.org.

12.1 How to Report Problems

Up front a few remarks on how to report problems (and how to respond), so that the chances to solve the problem (permanently) are as high as possible.
If you have compilation problems, please always state what version of CPMD you are trying to compile and what kind of machine you are using, i.e. what operating system, what compiler (particularly important on linux machines), which compilation flags, and what libraries you are using. Best you include the first part of your makefile (up to 'End of Personal Configuration', please **don't** post the whole makefile) as this contains most of the required information. Also include the **relevant** part of the make output (again, the full output usually is very long and rarely needed).
If you have problems with a specific calculation, please include your input and the output of the run, so that others can try to reproduce the error. Again, please state the version of CPMD you are using and the platform you are running on.
A useful article about how to ask questions the smart way is at:
http://www.catb.org/~esr/faqs/smart-questions.html

12.2 Explanation of Warnings and Error Messages

If execution of CPMD aborts with an error message, please do not only consult the standard output, but also the **LocalError** files which are produced in the run directory. They may contain more valuable information of the problem at hand.

Q: My CPMD run suddenly aborted without any error message. What happened?

A: CPMD will only print error messages to the output file in some very specific cases. Usually, if an error is encountered, every task writes to a file called *LocalError-X-X-X.log* (where x denotes a task ID), where you will find an *error message*, the *procedure* and *file name* where the error occured, as well as a *call stack* (although the latter may sometimes be incomplete). In case the error message is empty, you may have to consult the source code directly.
If no *LocalError* files are produced, the processes may have exited due to an insufficient amount of RAM available for a single task.

Q: Could anybody explain to me the follwing error in the cpmd output during CP Molecular Dynamics runs with the flag **WANNIER WFNOUT** LIST DENSITY?

```
WANNIER CODE WARNING: GRADIENT FOR RESTA FUNCTIONAL IS GMAX=0.118E-02
```

Does it mean any serious error in the calculation?

A: The default spreadfunctional used in CPMD is the Vanderbilt type. At the end of the calculation the convergence with respect to the Resta type functional is also checked. For large cells both should be converged at the same time. However, for typical application this is not the case and you get the warning. This is not serious and you can ignore it.

Q: A warning message appeared in the output file:

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!WARNING! XC FUNCTIONALS INCONSISTENT FOR h.pp
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

Does it mean that my pseudopotential file is wrong? I used fhi98PP to create this pp file, and just recast it. So if it is wrong, what more should I do to the output file of fhi98PP?

A: It means that the XC functional used to generate the pseudo potential and the functional which you want to use in CPMD are not the same. This could be more or less serious depending on the specific situation: Some of the XC functionals in CPMD are just minor variations of each other, e.g. the LDA part may evaluated using Perdew-Wang, Perdew-Zunger, VWN or the Pade-interpolation formula. If this is the case, the resulting error is usually small. However, if your pseudo potential was generated *e.g.* with PBE and you try to use it with the BLYP functional, the error might become non-negligible. You may compare the two functionals (pp & CPMD-input), and judge for yourself if the difference is significant or not.
For **hybrid** and **meta** functionals, there are usually no pseudo potentials available. In these cases, it is common practice to use the pseudo potentials of related GGA xc functionals (*e.g.* PBE for TPSS, BLYP for CAM-B3LYP, etc.), or, if there is no 'close relative', LDA (this is possible *e.g.* in combination with the Minnesota xc functionals).

Q: I am trying to optimize a crystal structure (both ion positions and cell volume) using CPMD and get the warning message:
Warning! Spline region smaller than maximum G-value.
The optimization seems to converge nicely but what does this warning mean/imply?

A: This in fact touches several points. The following applies also to other variable cell calculations with CPMD (e.g. constant pressure simulations).

- Pseudopotential functions in CPMD are calculated on a radial grid in G-space and then used in spline interpolations. This speeds up variable cell calculations considerably. The maximum grid point is given by the cutoff. With the keyword

```
SPLINE RANGE
  x.xx
```

  you can enlarge the grid to x.xx times the cutoff. Also, you should make sure, that the number of spline points is large enough. Older version of CPMD defaulted to as little as 501. This is at the lower limit for accuracy with a fixed cell. Especially if you have high cutoffs it is better to increase this value, e.g.

```
SPLINE POINTS
  2500
```

  or larger. The current default (5000) should be large enough.

- In a variable cell calculation, CPMD uses a constant number of plane waves. Therefore if your cell contracts the cutoff increases, if the cell gets larger the cutoff decreases. So if you have the first case the spline interpolation needs points above the original cutoff and you get a warning. Depending on the amount of change in the cell you expect a value for the **SPLINE RANGE** of 2–5 is needed.

- Coming back to the constant number of plane waves. If your cell gets larger the effective cutoff decreases. This may have very undesirable effects and it is better to define the plane waves on a box larger than any box you anticipate the simulation will reach. In order to not have to start with an unreasonable cell you can define the plane waves not with the actual box but with a reference cell, use the keyword

```
REFERENCE CELL
  a b c xa xb xc
```

- However, you really want to do a constant cutoff calculation, not a constant number of plane waves. For technical reasons this is not possible and in principle you should do the calculation at a high enough cutoff in order that the calculation is converged all along the simulation path (with the slight changes in cutoff).

To avoid these very high cutoffs the group in Trieste came up with a method that allows to perform pseudo constant cutoff calculations. This method is implemented in CPMD (keyword **CONSTANT CUTOFF**) and explained in the paper [67]

Q: I was optimizing wavefunctions for my system. After a successful run I modified CELL VECTORS. This latter calculation crashed with the following error either when I restarted (only wavefunctions) or if I started from scratch.

```
GORDER| PROGRAMING ERROR. INFORM THE PROGRAMER
```

Is there something wrong with the code?

A: This is a known problem in CPMD. The message comes from a test, which probes whether the G vectors are in a "safe" order, namely such that after a restart with a different number of processes or on a different machine the results agree. Usually this error only occurs in large systems with a high cut-off energy and/or large unit cells, i. e. where one gets lots of close-lying G vectors. There are two possible workarounds:

1. Slightly change your computational box in one dimension e. g. from 10.000000 to 10.000001. This helps some times.

2. The check is not 100 % accurate. This means by just ignoring the message you will most likely get correct results. The error would only appear in restarts where you could see a small inconsistency in energy in the first step. Final results should not be affected (except for MD if you do restarts).

   To avoid the stop, comment out the two lines at the end of file loadpa_utils.mod.F90 of the form

   ```
   CALL STOPGM('GORDER','ERROR IN G-VEC ORDERING (NHG)')
   ```

   However, be sure to check that the results are reasonable.

12.3 Pseudopotentials

Q: I'm confused about how to select a pseudopotential type (Troullier-Martins, Goedecker, etc.). What makes one choose say a Goedecker potential instead of a Vanderbilt potential?

A: The choice of a pseudopotential in CPMD calculations depends on needs, available resources and taste.
Troullier-Martins norm-conserving pseudopotentials are probably the most-commonly used type of pseudopotentials in CPMD calculations. They work very well for non-problematic elements and they are quite easy to create (note, that it is also easy to create a bad pseudopotential). When using the Kleinman-Bylander separation, one also has to be careful to avoid so called ghost states (e.g. many transition metals need LOC=l with l being an angular momentum smaller than default value which is highest).
Goedecker pseudopotentials are stored in an analytical form, that ensures the separability, but they usually need a higher (sometimes much higher) plane wave cutoff than the corresponding Troullier-Martins counterparts. Also the creation procedure is more complicated, but there is a very large library of tested pseudopotentials (mostly LDA but also some GGA pseudopotentials).
Vanderbilt pseudopotentials have the advantage of needing a much reduced plane wave cutoff. The drawback is, that only a limited subset of the functionality in CPMD is actually implemented with uspps (MD, wavefunction/geometry optimization and related stuff and only at the gamma point and you have to make sure, that your real space grid is tight enough). Also due to sacrificing norm-conservation for softer pseudopotentials, your wavefunction has very limited meaning, so that not

all features available for norm-conserving pseudopotentials can actually be easily implemented or implemented at all.
For some elements it can be rather difficult to generate good (i.e. transferable) pseudopotentials, so you should always check out the available literature.

Q: How do I choose the correct value of LMAX?

A: If you use a Vanderbilt or Goedecker type potential only the format of the LMAX-line has to be valid. The actual value is read from the pseudopotential file and the value in the input file will be ignored. It is highly recommended to still use values that make sense, in case you want to do a quick test with a numerical (Troullier-Martins) pseudopotential.
Generally, the highest possible value of LMAX depends on the highest angular momentum for which a "channel" was created in the pseudopotential. In the pseudopotential file you can see this from the number of coloumns in the &POTENTIAL section. The first is the radius, the next ones are the orbital angular momenta (s, p, d, f,. . .). As an example you can determine a potential for carbon using f-electrons and set LMAX=F. Since the f-state is not occupied in this case there is very little advantage but it costs calculation time. In short, you can use values as high as there is data in the pseudopotential file, but you don't have to if it is not needed by the physics of the problem.
A fact that causes confusion is that Hamann's code for pseudopotential generation always produces output for the d-channel, even if you only request channels s and p. You should be cautious if r_c and the energy eigenvalue of the p- and d-channels are equal. In most of these cases LMAX=P should be used.

12.4 File Formats and Interpretation of Data

Q: Why is my total energy so much different from a Gaussian calculation?

A: With CPMD you are using **pseudopotentials** to describe the atoms. Since the total energy describes only the interactions between the pseudocores and the valence electrons (and **some** core electrons in the case of so-called semi-core pseudopotentials), you are missing the contribution of the core electrons and the full core charges of a regular all-electron calculation. **Energy differences** between two configurations, on the other hand, should be comparable, provided you use the same number of atoms, the same plane wave cutoff, the same pseudopotentials, and the same supercell geometry in the CPMD calculation.

Q: In a molecular dynamics simulation, CPMD prints out a list of energies for each integration step. Does anyone know the meaning of the individual values.

A: Some explanations to the energy terms:

EKINC fictitious kinetic energy of the electrons in a.u. this quantity should oscillate but not increase during a simulation.

TEMPP Temperature of the ions, calculated from the kinetic energy of the ions (EKIONS).

EKS Kohn-Sham energy (the equivalent of the potential energy in classical MD).

ECLASSIC = EKS + EKIONS

EHAM = ECLASSIC + EKINC. Hamiltonian energy, this is the conserved quantity, depending on the time step and the electron mass, this might oscillate but should not drift.

DIS mean square displacement of the ions with respect to the initial positions. Gives some information on the diffusion.

You can modify the list of individual energies to be displayed with the **PRINT ENERGY** keyword.

Q: What do GNMAX, GNORM and CNSTR mean in a geometry optimization?

A: These are abbreviations for the following quantities:

GNMAX $\max_{I,a}(|F_{Ia}|)$ = largest absolute component ($a = x, y, z$) of the force on any atom I.

GNORM $\langle F_I^2 \rangle_I$ = average force on the atoms I

CNSTR $\max_{I,a} F_{Ia}^{constr}$ = largest absolute component ($a = x, y, z$) of force due to constraints on any atom I.

Q: I found all the IR intensities in VIB.log file are zero when I try to calculate the IR of NH_4^+ ion by CPMD.

```
Harmonic frequencies (cm**-1), IR intensities (KM/Mole),
Raman scattering activities (A**4/AMU), Raman depolarization ratios,
reduced masses (AMU), force constants (mDyne/A) and normal coordinates:
                        1                      2                      3
                       ?A                     ?A                     ?A
Frequencies --   142.9800               188.9340               237.2614
Red. masses --     0.0000                 0.0000                 0.0000
Frc consts  --     0.0000                 0.0000                 0.0000
IR Inten    --     0.0000                 0.0000                 0.0000
Raman Activ --     0.0000                 0.0000                 0.0000
Depolar     --     0.0000                 0.0000                 0.0000
Atom AN       X      Y      Z        X      Y      Z        X      Y      Z
  1   7     0.00   0.00   0.00     0.00   0.00   0.00     0.00   0.00   0.00
  2   1     0.00  -0.35  -0.50    -0.35   0.00   0.00    -0.50   0.00   0.00
  3   1     0.00  -0.35   0.50    -0.35   0.00   0.00     0.50   0.00   0.00
  4   1     0.00   0.35   0.00     0.35   0.00   0.50     0.00  -0.50   0.00
  5   1     0.00   0.35   0.00     0.35   0.00  -0.50     0.00   0.50   0.00
                        4                      5                      6
```

A: That's not a problem of your calculation. The keyword **VIBRATIONAL ANALYSIS** does not calculate intensities. The calculation of intensities is currently not possible in CPMD. The intensities in the 'VIBx.log' files are arbitrarily set to zero. The entries have to be there so that visualisation programs, that are able to read output of the Gaussian program, can be also used to visualize the CPMD results.

Q: I am trying to simulate a bulk liquid in CPMD and supposing that periodic boundary conditions are built into the program. But after several thousand MD steps, I found some particles are far away from the central simulation box.
Why it is so if periodic boundary conditions (PBC) on particle coordinates are imposed in all three directions?

A: If you are not using the

```
SYMMETRY
  0
```

options your calculations are actually using periodic boundary conditions (PBC). PBC are imposed within CPMD for all calculations. However, the particle positions are not folded back to the original computational box. The reason for this is that most people prefer to have "smooth" trajectories without jumps of particles. This allows for easier tracking of special particles and nicer graphics. In addition it is easy (with a little script) to apply PBC afterwards yourself, if needed.

Q: I am trying to simulate a bulk sodium and I found electron energy is increasing continuously and it is in the range of 0.07 a.u. at the end of 20000 steps.

A: Sodium is a metal, and therefore missing an important feature that allows for stable CP dynamics: the band gap. Using Nosé thermostats (on electrons and ions) it might still be possible to perform meaningful CP simulations [178].
The choice of parameters for the thermostats, however, will be nontrivial, highly system dependent and require extensive testing. Without thermostats you will have strong coupling between electronic degrees of freedom and ionic degrees of freedom. Adiabaticity is not maintained and a steady increase of the fictitious kinetic energy will occur.

Q: I have computed RAMAN by LINEAR RESPONSE, and get three files: APT, POLARIZATION and POLARIZABILITY with lots of data in these files. I want to know the meaning of the data, please give me some answer in detail.

A: The POLARIZABILITY file simply contains the polarizability tensor of the whole system in atomic units. The POLARIZATION file contains the total dipole moment (electronic + ionic) of the whole system in atomic units. As for the file APT, it contains the atomic polar tensors for each atom in the system. The atomic polar tensor is the derivative of the forces on the atoms with respect to an applied external electric field. Equivalently it is, from a Maxwell relation, the derivative of the total dipole of the system with respect to the nuclei positions. It is thus an important ingredient of the calculation of infrared spectra intensities, for example used in an harmonic approximation. The trace of this tensor is the so-called Born charge of the considered atom. The data is arranged in the following order (still in a.u.): the APT tensor is $\frac{\mathrm{d}F_{I,i}}{\mathrm{d}E_j}$ where $F_{I,i}$ is the force on atom I along $i = x, y, z$ and E_j is the electric field along $j = x, y, z$. (I, i) are the indices of the $3N$ atoms lines in the APT file, one atom after the other, and j is the column index in the APT file.

Q: I was wondering what columns 2 to 7 in the DIPOLE file correspond to? When I run CPMD v.3.5.1, columns 2 to 4 come out identical to columns 5 to 7 respectively. When I run with CPMD v.3.4.1, the columns come out different. Is there an explanation for this?

A: Columns 2 to 4 in the DIPOLE file are the electronic contribution to the dipole moment, columns 5 to 7 are the total (electronic + ionic) dipole moment. All dipole moments are divided by the volume of the box.
In CPMD version 3.5.1 we have changed the reference point of the calculation. Now the reference point is chosen such that the ionic contribution is zero and the electronic contribution minimal (=total dipole). This avoids a problem that occasionally was seen in older versions. The electronic dipole is calculated modulo(2π/L). Now if the electronic dipole became too large, because the ionic contribution was large (bad choice of reference point) the total dipole made jumps of 2π.

Q: As you know, the cpmd RESTART file is saved as binary. But I want to change it to ASCII and vice versa, because I use several machines of different architecture, for example COMPAQ, IBM, and LINUX machine. Please help me with any comments.

A: The code to read and write the RESTART file is in the files rv30_utils.mod.F90 and wv30_utils.mod.F90. Feel free to implement an ASCII version of the restart, but be aware that the file will be **huge**.
But you may not need to do that. Let's say you decide to use big-endian binary encoding (this is what e.g. IBM, Sun and SGI machines do natively).
With Compaq machines there is a compiler flag, -convert, which you could set to big_endian (we only have here linuxalpha, but the compaq compiler should be essentially the same).
On a Linux PC you can use the use the -Mbyteswapio or the -byteswapio flag, if you have the PGI compiler.
For the Intel compiler (ifc/ifort/efc) you simply set the environment variable F_UFMTENDIAN to big (i.e.
'export F_UFMTENDIAN=big'
if you are in a bourne/korn shell and
'setenv F_UFMTENDIAN big'
if you are in a (t)csh).

Now even your cpmd executables will read and write big-endian restart files.
Check your compiler documentation for more details (search for endian).

12.5 Input Parameter Values

Q: If I set the keyword RESTART WAVEFUNCTION COORDINATES, would I have to write the &SYSTEM and &ATOM section again?

A: Yes, you have to include the &SYSTEM and &ATOM sections even if you are restarting. If you write RESTART COORDINATES, the coordinates in the RESTART file override the ones in the input. RESTART WAVEFUNCTION alone does not select the coordinates in the RESTART file, but does use those in the &ATOMS section.

Q: Could anybody tell me how to choose the energy cutoff in &SYSTEM section?

A: The best way to choose the cutoff for CPMD calculations is by running first a series of tests. Select a test system and a representative quantity (bond length, reaction energy, etc.), perform a series of calculations with increasing cutoff, pick the lowest cutoff with satisfactory results.
It's always a good idea to make checks at some critical points of the calculations by increasing the cutoff. See also section 11.1.

Q: I have a problem with visualising unoccupied orbitals. When I use **RHOOUT** BANDS or **CUBEFILE** ORBITALS after the wavefunction optimization I get only occupied orbitals. If I add one empty state when optimizing wavefunction the program never reaches convergence.

A: The most efficient way to calculate unoccupied orbitals is to first optimize the occupied orbitals and then restart the calculation using the run option

```
KOHN-SHAM ENERGIES
 n
```

where n ist the number of unoccupied orbitals. This will diagonalize the Kohn-Sham Potential (defined by the occupied orbitals alone).
To test if everything goes fine, you can check the total energy printed at the beginning of this job, it should be exactly the one at the end of the optimization. In addition, if you don't change the default convergence criteria, the number of converged Kohn-Sham states should be equal to the number of occupied states in the first step.

Q: Is there any way to force CPMD to dump DENSITY files every N steps of molecular dynamics run instead (or except) of the end of the job?

A: Short of modifying the source code, you could set the parameter **RESTFILE** to a large number and than have CPMD write a restart file every N steps via the **STORE** keyword. Now you rename each restart in turn from RESTART.# to RESTART and do a single step calculation using the **RESTART** keyword without the LATEST modifier which will write the DENSITY file (or run a **PROPERTIES** job using **CUBEFILE** DENSITY to get the cube file directly).

Q: How do I calculate a Band structure with CPMD? To calculate a band structure with CPMD, You first calculate the correct density for your system with a Monkhorst-Pack Mesh.

A: Then you use: OPTIMIZE WAVEFUNCTIONS with MAXSTEP 1 (no self-consistency) and RESTART DENSITY.
In the section KPOINTS, you should use for instance a bcc:

```
KPOINTS BANDS
     51   0   0   0      0   0   1        Gamma to H
     51   0   0   1      0  .5  .5        H to N
     51   0  .5  .5     .5  .5  .5        N to P
     51  .5  .5  .5      0   0   0        P to Gamma
```

```
51   0   0   0     .5  .5   0             Gamma to N
51   0   0   1     .5  .5  .5             H to P
0    0 0 0   0 0 0
```

You say that you want 51 points from (0,0,0) and (0,0,1) and so on. The last line with many zeros is to stop.
If the memory of your computer is not enough, you can add in the line KPOINTS the option BLOCK=50 that means you want to have only 50 kpoints in memory. This options worked some time ago.

Q: I've been recently trying to use the VELOCITIES keyword in a molecular dynamics run. I want to collide fast atoms against surfaces. Despite the code seems to read the input velocities properly, when the run starts the initial velocities are always the same (apparently coming from a thermal distribution), no matter what is the velocity you specify for the incoming atom. I'm not using QUENCH IONS, so I don't understand why the input initial velocities are not considered in the calculation.

A: There is no straightforward way in CPMD to achieve what you want. I suggest to follow this procedure:

1) Run a single step of MD with the following set up

```
MOLECULAR DYNAMICS
MAXSTEP
 1
RESTART WAVEFUNCTION COORDINATES
TEMPERATURE
  300   <- or whatever your surface should be
```

This generates RESTART and GEOMETRY files. Now edit the GEOMETRY file to change the velocities of the particles according to your experiment. Now restart the MD with the options

```
MOLECULAR DYNAMICS
MAXSTEP
 1000
RESTART WAVEFUNCTION COORDINATES VELOCITIES GEOFILE
QUENCH ELECTRONS
```

The effect of this is: IONIC coordinates and velocities are read from GEOMETRY, ELECTRON wavefunctions and velocities are read from RESTART, ELECTRON velocities are set to zero.

Q: I want to to run CPMD with basis sets equivalent to Gaussian 6-31+G(d) and 6-311+G(2d,p). How do I set up the &BASIS section?

A: You should be able to construct inputs from the description in this manual (see section 9.5.3). Please note, that the basis set generated from the &BASIS section is used in CPMD for two purposes:

1. Analyzing orbitals:
   We usually use the atomic pseudo-wavefunctions to analyze the orbitals from CPMD. The 6-31G type Gaussian basis sets are for all electron calculations. Don't expect very good results when analyzing wavefunctions from a pseudopotential calculation.

2. Generating orbitals for an initial guess:
   By default we use a Slater minimal basis. In most cases the effort to produce a better initial guess using "better" wavefunctions does not pay off.

Q: How do I add support for a new functional?

A: The exchange-correlation functionals and scratch variables are set and handled in the module cp_xc_utils.mod.F90. The individual functionals are grouped in modules (cp_gga_exchange_utils.mod.F90 etc.) which are in turn used in cp_xc_utils.mod.F90. If you wish to implement higher-order analytical derivatives for LR-TDDFT, cp_dxc_utils.mod.F90, and the cp_dgga…routines may be of interest.
The inclusion of hybrids will require additional modifications to dftin_utils.mod.F90 and cpfunc_func_init (in cp_xc_utils.mod.F90). The variables for range separation are currently still found in func.mod.F90 and the range separated exact exchange is handled in initclust_utils.mod.F90.

Q: I am trying to use the Minnesota functionals (M06, M11, …), but the results I get are most erratic, or they require very high values for the **CUTOFF**. What is wrong?

A: The Minnesota functionals require unusually dense integration grids. Simply increasing the cutoff will introduce numerical noise and make the wavefunction optimisation unstable. This issue can be overcome by increasing the **DUAL** up to 8-12. A detailled assessment of these effects along with recommendations for appropriate values can be found in the following reference: M.P. Bircher, P. Lopez-Tarifa and U. Rothlisberger: *J. Chem. Theory Comput.*, **15** (1), 557 (2019). DOI: 10.1021/acs.jctc.8b00897

Q: Results obtained from the Minnesota functionals (M05, M06, M08, M11) seem well converged, but exhibit significant differences with respect to Gaussian bases. Is there something wrong?

A: Due to their highly flexible functional form, the Minnesota functionals are extremely sensitive to the choice and flexibility of the underlying basis set. Since Gaussian functions always impose a certain rigidity, results obtained in Gaussian basis sets may occasionally differ by more than just chemical accuracy from the plane wave results. This effect is not limited to plane waves, it can also be observed - to a lesser extent - when comparing plane waves and Slater functions, and can be rather pronounced when comparing Gaussian and Slater functions. A detailled study of basis set effects is provided in the following article: M.P. Bircher, P. Lopez-Tarifa and U. Rothlisberger: *J. Chem. Theory Comput.*, **15** (1), 557 (2019). DOI: 10.1021/acs.jctc.8b00897

Q: I am running a calculation with exact exchange, but it is exceedingly slow compared to a GGA or MGGA run. Can this issue be resolved?

A: The implementation of exact (or Hartree-Fock) exchange in CPMD is highly efficient and parallel. If the number of available nodes is limited, increasing the value in **HFX_BLOCK_SIZE** can substantially speed up the calculation. In the presence of ample computational resources, the calculation can be sped up by using **CP_GROUPS**.
In isolated systems, the calculation can be sped up by up to an additional order of magnitude by using the coordinate-scaling scheme by Bircher and Rothlisberger. For details, see **SCEX**.

References

[1] P.J. Mohr, B.N. Taylor, and D.B. Newell, *The 2006 CODATA Recommended Values of the Fundamental Physical Constants*, Web Version 5.1, 2007.

[2] M.P. Allen and D.J. Tildesley, *Computer Simulations of Liquids*, Clarendon Press, Oxford, 1987.

[3] R. Car and M. Parrinello, Phys. Rev. Lett. **55**, 2471 (1985).

[4] D. Vanderbilt, Phys. Rev. B **41**, 7892 (1990).

[5] G. Galli and M. Parrinello, *Computer Simulation in Materials Science*, in *Proc. NATO ASI*, Eds. M. Meyer and V. Pontikis, Kluwer, 1991.

[6] S. Nosé, J. Chem. Phys. **81**, 511 (1984), Mol. Phys. **52**, 255 (1984).

[7] W. G. Hoover, Phys. Rev. A **31**, 1695 (1985).

[8] W. Kohn and L. J. Sham, Phys. Rev. **140**, A1133, (1965).

[9] D. Marx and M. Parrinello, Z. Phys. B (Rapid Note) **95**, 143 (1994).

[10] D. Marx and M. Parrinello, J. Chem. Phys. **104**, 4077 (1996).

[11] M. Parrinello and A. Rahman, J. Appl. Phys. **52**, 7182 (1981).

[12] M. Parrinello and A. Rahman, Phys. Rev. Lett. **45**, 1196 (1980).

[13] S. Goedecker, M. Teter, and J. Hutter, Phys. Rev. B **54**, 1703 (1996).
C. Hartwigsen, S. Goedecker, and J. Hutter, Phys. Rev. B **58**, 3641 (1998).

[14] A. Alavi, J. Kohanoff, M. Parrinello, and D. Frenkel, Phys. Rev. Lett. **73**, 2599 (1994).

[15] M. E. Tuckerman, D. Marx, M. L. Klein, and M. Parrinello, J. Chem. Phys. **104**, 5579 (1996).

[16] M. Eichinger, P. Tavan, J. Hutter, and M. Parrinello, J. Chem. Phys. **110**, 10452 (1999).

[17] H. B. Callen and R. F. Greene, Phys. Rev. **86**, 702 (1952).
R. Kubo, J. Phys. Soc. Japan **12**, 570 (1957).

[18] N. Marzari and D. Vanderbilt, Phys. Rev. B **56**, 12847 (1997).

[19] A. Laio, J. VandeVondele, and U. Röthlisberger J. Chem. Phys. **116**, 6941 (2002).

[20] A. Laio, J. VandeVondele, and U. Röthlisberger J. Phys. Chem. B **106**, 7300 (2002).

[21] M. Boero, *Reactive Simulations for Biochemical Processes* in *Atomic-Scale Modeling of Nanosystems and Nanostructured materials* p.$\tilde{8}$1-98, Springer, Berlin Heidelberg, 2010. ISBN 978-3-642-04650-6

[22] A. Putrino, D. Sebastiani, and M. Parrinello, J. Chem. Phys. **113**, 7102 (2000).

[23] A. Putrino and M. Parrinello, Phys. Rev. Lett. **88**, 176401 (2002).

[24] D. Sebastiani and M. Parrinello, J. Phys. Chem. A **105**, 1951 (2001).

[25] M. Iannuzzi and M. Parrinello, Phys. Rev. B **64**, 233104 (2002).

[26] S. R. Billeter, A. Curioni, and W. Andreoni, Comput. Mat. Sci. **27**, 437 (2003).

[27] A. Laio and M. Parrinello, Proc. Natl Acad. Sci. USA **20**, 12562 (2002).

[28] M. Iannuzzi, A. Laio, and M. Parrinello, Phys. Rev. Lett. **90**, 238302 (2003).

[29] R. W. Hockney, Methods Comput. Phys. **9**, 136 (1970).

[30] M. Elstner, P. Hobza, T. Frauneheim, S. Suhai, and E. Kaxiras, J. Chem. Phys. **114**, 5149 (2001).

[31] I. Frank, J. Hutter, D. Marx, and M. Parrinello, J. Chem. Phys. **108**, 4060 (1998).

[32] D. Egli and S. R. Billeter, Phys. Rev. B **69**, 115106 (2004).

[33] P. K. Biswas, V. Gogonea, J. Chem. Phys. **123**,164114 (2005). The corresponding modifications to the Gromacs code are available at http://comppsi.csuohio.edu/groups/

[34] J. Hutter and A. Curioni, ChemPhysChem **6**, 1788-1793 (2005).

[35] J. Hutter and A. Curioni, Parallel Computing **31**, 1 (2005).

[36] E.J. Reed, L.E. Fried and J.D. Joannopoulos, Phys. Rev. Lett. **90**, 235503 (2003).

[37] S. Grimm, C. Nonnenberg, and I. Frank, J. Chem. Phys. **119**, 11574 (2003).

[38] S. Grimme, J. Comp. Chem. **27**, 1787 (2006).

[39] S. Grimme, J. Antony, S. Ehrlich, and S. Krieg, J. Chem. Phys. **132**, 154104 (2010).

[40] C. Bekas and A. Curioni Comp. Phys. Comm.**181**, 1057 (2010)

[41] H Oberhofer and J. Blumberger, J. Chem. Phys. **131**, 064101 (2009)

[42] M. Ceriotti, G. Bussi and M. Parrinello, Phys. Rev. Lett.**102** , 02061 (2009)

[43] Ceriotti, M. and Bussi, G. and Parrinello, M., J. Chem. Th. Comput. **6**, 1170 (2010).

[44] J. Heyd, G. E. Scuseria, and M. Ernzerhof, J. Chem. Phys. **118**, 8207 (2003); J. Chem. Phys. **124**, 21990624(E) (2006).

[45] A. V. Krukau, O. A. Vydrov, A. F. Izmaylov, and G. E. Scuseria, J. Chem. Phys. **125**, 224106 (2006).

[46] I. Tavernelli, B. Curchod and U. Röthlisberger J. Chem. Phys. **131**, 196101 (2009).

[47] I. Tavernelli Phys. Rev. B **73**, 094204,(2006).

[48] G. P. Kerker, Phys. Rev. B **23**, 3082 (1981).

[49] L. Maragliano, A. Fischer, E. Vanden-Eijnden, and G. Ciccotti, J. Chem. Phys. **125**, 024106 (2006).

[50] I. Tavernelli, U.F. Rohrig, U. Röthlisberger Mol. Phys. **103**, 963 (2005).

[51] M. Ernzerhof, *Density Functionals: Theory and Applications*, in *Lecture Notes in Physics*, vol. 500, Eds. D. P. Joubert, Springer–Verlag, Berlin, 1998.

[52] C. Adamo, A. di Matteo, and V. Barone, *From Classical Density Functionals to Adiabatic Connection Methods. The State of the Art.* in *Advances in Quantum Chemistry*, Vol. 36, Academic Press (2000).

[53] A. D. Becke, J. Chem. Phys. **104** 1040 (1996).

[54] A. D. Becke, J. Chem. Phys. **98** 5648 (1993).

[55] A. D. Becke, Phys. Rev. A **38**, 3098 (1988).

[56] H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, A. DiNola, and J. R. Haak, J. Chem. Phys. **81**, 3684 (1984).

[57] R. Fletcher, *Practical Methods of Optimizations*, vol. 1, Wiley,: New York, 1980.

[58] J. Frenzel, B. Meyer, D. Marx, *Bicanonical Ab Initio Molecular Dynamics for Open Systems* J. Chem. Theory Comput., Just Accepted Manuscript (2017), DOI: 10.1021/acs.jctc.7b00263 .

[59] D. D. Johnson, Phys. Rev. B **38**, 12807 (1988).

[60] (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).

[61] (a) G. J. Martyna, J. Chem. Phys. **104**, 2018 (1996); (b) J. Cao and G. J. Martyna, J. Chem. Phys. **104**, 2028 (1996).

[62] D. Marx, M. E. Tuckerman, and G. J. Martyna, Comput. Phys. Commun. **118**, 166 (1999).

[63] F. L. Hirshfeld, Theoret. Chim. Acta **44**, 129 (1977).

[64] S. R. Cox and P. A. Kollman, J. Comput. Chem. **5**, 129 (1984).

[65] M. Boero, M. Parrinello, K. Terakura, T. Ikeshoji, and C. C. Liew, Phys. Rev. Lett. **90**, 226403 (2003).

[66] M. Boero, J. Phys. Chem. A **111**, 12248 (2007).

[67] M. Bernasconi, G. L. Chiarotti, P. Focher, S. Scandalo, E. Tosatti, and M. Parrinello, J. Phys. Chem. Solids, **56** 501 (1995).

[68] M. Cavalleri, M. Odelius, A. Nilsson and L. G. Pettersson, J. Chem. Phys. **121**, 10065 (2004).

[69] S. R. Billeter and A. Curioni, J. Chem. Phys. **122**, 034105 (2005).

[70] E. R. Davidson, J. Comput. Phys. **17**, 87 (1975).

[71] L. D. Fosdick and H. F. Jordan, Phys. Rev. **143**, 58 (1966).

[72] R. Resta, Ferroelectrics **136**, 51 (1992).
R. D. King–Smith and D. Vanderbilt, Phys. Rev. B **47**, 1651 (1993).
R. Resta, Europhys. Lett. **22**, 133 (1993).

[73] R. Resta, Rev. Mod. Phys. **66**, 899 (1994).
R. Resta, Phys. Rev. Lett. **80**, 1800 (1998).
R. Resta, Phys. Rev. Lett. **82**, 370 (1999).

[74] P. L. Silvestrelli, Phys. Rev. B **59**, 9703 (1999).

[75] G. Berghold, C. J. Mundy, A. H. Romero, J. Hutter, and M. Parrinello, Phys. Rev. B **61**, 10040 (2000).

[76] C. Bekas, A. Curioni and W. Andreoni, Parallel Computing **34**, 441 (2008).

[77] E. Runge and E. K. U. Gross, Phys. Rev. Lett. **52**, 997 (1984).
M. E. Casida, in *Recent Advances in Density Functional Methods*, Vol. 1, edited by D. P. Chong (World Scientific, Singapore, 1995).
M. E. Casida, in *Recent Developments and Applications of Modern Density Functional Theory*, Theoretical and Computational Chemistry, Vol. 4, edited by J. M. Seminario (Elsevier, Amsterdam, 1996).

C. Jamorski, M. E. Casida, and D. R. Salahub, J. Chem. Phys. **104**, 5134 (1996).
S. J. A. van Gisbergen, J. G. Snijders, and E. J. Baerends, J. Chem. Phys. **103**, 9347 (1996).
R. Bauernschmitt and R. Ahlrichs, Chem. Phys. Lett. **256**, 454 (1996).
K. B. Wiberg, R. E. Stratmann, and M. J. Frisch, Chem. Phys. Lett. **297**, 60 (1998).
R. E. Stratmann, G. E. Scuseria, and M. J. Frisch, J. Chem. Phys. **109**, 8218 (1998).
A. Görling, H. H. Heinze, S. Ph. Ruzankin, M. Staufer, and N. Rösch, J. Chem. Phys. **110**, 2785 (1999).
R. Bauernschmitt, M. Häser, O. Treutler, and R. Ahlrichs, Chem. Phys. Lett. **264**, 573 (1997).

[78] J. Hutter, J. Chem. Phys. **118**, 3928 (2003).

[79] A. D. Becke and K. E. Edgecombe, J. Chem. Phys. **92**, 5397 (1990).

[80] B. Silvi and A. Savin, Nature **371**, 683 (1994).

[81] D. Marx and A. Savin, Angew. Chem. Int. Ed. Engl. **36**, 2077 (1997).

[82] Check the ELF homepage http://www.cpfs.mpg.de/ELF/ for lots of useful information in particular on how ELF should be interpreted.

[83] M. Kohut and A. Savin, Int. J. Quant. Chem. **60**, 875–882 (1996)

[84] I. Tavernelli, B. F. E. Curchod, and U. Rothlisberger, Phys. Rev. A. **81**, 052508 (2010).

[85] N. L. Doltsinis, D. Marx, Phys. Rev. Lett. **88**, 166402 (2002).

[86] P. L. Silvestrelli, A. Alavi, M. Parrinello, and D. Frenkel, Phys. Rev. Lett. **77**, 3149 (1996).

[87] M. Boero, A. Oshiyama, P. L. Silvestrelli, and K. Murakami, Appl. Phys. Lett. **86**, 201910 (2005).

[88] S. R. Billeter and D. Egli, J. Chem. Phys. **125**, 224103 (2006).

[89] P. Császár and P. Pulay, J. Mol. Struct. **114** 31 (1984).

[90] M. Rossi, M. Ceriotti, and D. E. Manolopoulos, J. Chem. Phys. **140**, 234116 (2014).

[91] J. P. Perdew, J. A. Chevary, S. H. Vosko, K. A. Jackson, M. R. Pederson, D. J. Singh, and C. Fiolhais, Phys. Rev. B **46**, 6671 (1992); *Erratum* Phys. Rev. B **48**, 4978 (1993).

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

[93] Y. Zhang and W. Yang, Phys. Rev. Lett. **80**, 890 (1998).

[94] A.D. Boese, N.L. Doltsinis, N.C. Handy, and M. Sprik, J. Chem. Phys. **112**, 1670 (2000).

[95] N. C. Handy and A. J. Cohen, J. Chem. Phys. **116**, 5411 (2002).

[96] J. P. Perdew, A. Ruzsinszky, G. I. Csonka, O. A. Vydrov, G. E. Scuseria, L. A. Constantin, X. Zhou, and K. Burke, Phys. Rev. Lett. **100**, 136406 (2008).

[97] J. P. Perdew, Phys. Rev. B **33**, 8822 (1986); *Erratum* Phys. Rev. B **34**, 7406 (1986).

[98] C. Lee, W. Yang and R. G. Parr, Phys. Rev. B **37**, 785 (1988).

[99] M. E. Tuckerman and M. Parrinello, J. Chem. Phys. **101**, 1302 (1994); *ibid.* **101**, 1316 (1994).

[100] T. H. Fischer and J. Almlöf, J. Phys. Chem. **96**, 9768 (1992).

[101] H. B. Schlegel, Theo. Chim. Acta **66**, 333 (1984).

[102] M. Eichinger, H. Heller, and H. Grubmüller, in *Workshop on Molecular Dynamics on Parallel Computers*, p. 154-174, Singapore 912805, World Scientific, (2000).

[103] M. Eichinger, H. Grubmüller, H. Heller, and P. Tavan, J. Comp. Chem. **18**, 1729 (1997).

[104] E. Lindahl, B. Hess, and D. van der Spoel, J. Mol. Mod. **7**, 306 (2001).

[105] D. C. Liu and J. Nocedal, Math. Prog. **45**, 503 (1989).

[106] J. P. Perdew and A. Zunger, Phys. Rev. B **23**, 5048 (1981).

[107] S. H. Vosko, L. Wilk and M. Nusair, Can. J. Phys. **58**, 1200 (1980).

[108] J. P. Perdew and Y. Wang, Phys. Rev. B **45**, 13244 (1991).

[109] D. M. Ceperley and B. J. Alder, Phys. Rev. Lett.**45**, 566 (1980).

[110] J. Hutter, M. Parrinello, and S. Vogel, J. Chem. Phys. **101**, 3862 (1994).

[111] I. Tavernelli, Phys. Rev. A **87**, 042501 (2013).

[112] J. Hutter, M. E. Tuckerman, and M. Parrinello. J. Chem. Phys. **102**, 859 (1995).

[113] J. VandeVondele, and U. Röthlisberger J. Phys. Chem. B **106**, 203 (2002).

[114] J. Hutter, H. P. Lüthi, and M. Parrinello, Comput. Mat. Sci. **2**, 244 (1994).

[115] G. J. Martyna, D. J. Tobias, and M. L. Klein, J. Chem. Phys. **101**, 4177 (1994).

[116] C. Dellago, P. G. Bolhuis, F. S. Csajka, D. J. Chandler, J. Chem. Phys. **108**, 1964 (1998). P. G. Bolhuis, C. Dellago, D. J. Chandler, Faraday Discuss. **110**, 42 (1998).

[117] G.J. Martyna and M. E. Tuckerman, J. Chem. Phys. **110**, 2810 (1999).

[118] 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).

[119] M. J. D. Powell, Math. Prog. **1**, 26 (1971).

[120] A. Banerjee, N. Adams, J. Simons, and R. Shepard, J. Phys. Chem. **89**, 52 (1985).

[121] A. J. Turner, V. Moliner and I. H. Williams, Phys. Chem. Chem. Phys. **1**, 1323 (1999).

[122] R. Craig and D. E. Manolopoulos, J. Chem. Phys. **121**, 3368 (2004).

[123] W. F. van Gunsteren, S. R. Billeter, A. A. Eising, P. H. Hünenberger, P. Krüger, A. E. Mark, W. R. P. Scott, I. G. Tironi, Biomolecular Simulation: The GROMOS96 Manual and User Guide; Vdf Hochschulverlag AG an der ETH Zürich: Zürich, 1996.

[124] D. A. Case, D. A. Pearlman, J. W. Caldwell, T. E. Cheatham III, J. Wang, W. S. Ross, C. L. Simmerling, T. A. Darden, K. M. Merz, R. V. Stanton, A. L. Cheng, J. J. Vincent, M. Crowley, V. Tsui, H. Gohlke, R. J. Radmer, Y. Duan, J. Pitera, I. Massova, G. L. Seibel, U. C. Singh, P. K. Weiner, and P. A. Kollman, AMBER 7 (2002), University of California, San Francisco.

[125] F. J. Momay, J. Phys. Chem. **82**, 592 (1978).

[126] C. I. Bayly, P. Cieplak, W. D. Cornell and P. A. Kollman, J. Phys. Chem. **97**, 10269 (1993).

[127] J. C. Slater, Phys. Rev. **81**, 385 (1951).

[128] K. Laasonen, M. Sprik, M. Parrinello and R. Car, J. Chem. Phys. **99**, 9081 (1993).

[129] F. Franco de Carvalho, B.F.E. Curchod, T. Penfold, I. Tavernelli, J. Chem. Phys. **140**, 144103 (2014).

[130] J. VandeVondele and M. Sprik, Phys. Chem. Chem. Phys. **7**, 1363 (2005).

[131] F. L. Gervasio, M. Boero, and M. Parrinello, Angew. Chem. Int. Ed. **45**, 5606 (2006).

[132] E. Tapavicza, I. Tavernelli, and U. Rothlisberger, Phys. Rev. Lett. **98**, 023001 (2007). I. Tavernelli, E. Tapavicza, and U. Rothlisberger, J. Mol. Struct. : THEOCHEM **914**, 22 (2009).

[133] R. N. Barnett and U. Landman, Phys. Rev. B **48**, 2081 (1993).

[134] L. Knoll and D. Marx, Eur. Phys. J. D **10**, 353 (2000).

[135] E. Liberatore, R. Meli, and U. Rothlisberger, J. Chem. Theory Comput., **14**, 2834 (2018).

[136] W. T. M. Mooij, F. B. van Duijneveldt,J. G. C. M. van Duijneveldt-van de Rijdt, and B. P. van Eijck, J. Phys. Chem A **103**, 9872 (1999).

[137] R. W. Williams and D. Malhotra, Chem. Phys. **327**, 54 (2006).

[138] L. Kleinman and D. M. Bylander, Phys. Rev. Lett. **48**, 1425 (1982).

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

[140] G. B. Bachelet, D. R. Hamann and M.Schlüter, Phys. Rev. B **26**, 4199 (1982).

[141] 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).

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

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

[144] M. Sprik, Faraday Discuss. **110**, 437 (1998).

[145] M. Sprik and G. Ciccotti, J. Chem. Phys. **109**, 7737 (1998).

[146] K. Kamiya, M. Boero, M. Tateno, K. Shiraishi, and A. Oshiyama, J. Am. Chem. Soc. **129**, 9663 (2007).

[147] D. Frenkel and B. Smit, *Understanding Molecular Simulation*, Academic Press, San Diego, 2002.

[148] F. Filippone and M. Parrinello, Chem. Phys. Lett. **345**, 179 (2001).

[149] K. Fukui, Science **217**, 747 (1982).

[150] W. Yang and R. G. Parr, Proc. Natl. Acad. Sci. USA **82**, 6723 (1985).

[151] E. Chamorro, F. De Proft and P. Geerlings, J. Chem. Phys. **123**, 084104 (2005).

[152] C. Micheletti, A. Laio, and M. Parrinello, Phys. Rev. Lett. **92**, 170601 (2004).

[153] A. Laio and F. L. Gervasio, Rep. Prog. Phys. **71**, 126601 (2008).

[154] A. Barducci, M. Bonomi, and M. Parrinello, Wiley Interdisciplinary Reviews - Computational Molecular Science **1**, 826 (2011).

[155] A. Stirling, M. Iannuzzi, A. Laio, and M. Parrinello, ChemPhysChem **5**, 1558 (2004).

[156] A. Laio, A. Rodriguez-Fortea, F. L. Gervasio, M. Ceccarelli, and M. Parrinello, J. Phys. Chem. B **109**, 6714 (2005).

[157] M. Iannuzzi and M. Parrinello, Phys. Rev. Lett. **93**, 025901 (2004).

[158] S. Churakov, M. Iannuzzi, and M. Parrinello, J. Phys. Chem. B **108** 11567 (2004).

[159] M. Boero, T. Ikeshoji, C. C. Liew, K. Terakura and M. Parrinello, J. Am. Chem. Soc. **126**, 6280 (2004).

[160] T. Ikeda, M. Hirata and T. Kimura, J. Chem. Phys. **122**, 244507 (2005).

[161] M. Boero, M. Tateno, K. Terakura, and A. Oshiyama, J. Chem. Theor. Comput. **1**, 925 (2005).

[162] N. N. Nair, E. Schreiner, and D. Marx, J. Am. Chem. Soc. **130**, 14148 (2008).

[163] M. Boero, J. Phys. Chem. B **115**, 12276 (2011).

[164] P. Raiteri, A. Laio, F. L. Gervasio, C. Micheletti and M. Parrinello, J. Phys. Chem. B **110**, 3533 (2005).

[165] N. N. Nair, E. Schreiner and D. Marx inSiDE **6**, 30 (2008). http://inside.hlrs.de/htm/Edition_02_08/article_09.html

[166] C. Dellago, P. G. Bolhuis, F. S.Csajka, and D. Chandler, J. Chem. Phys. **108** 1964 (1998).

[167] Q. Wu and T. Van Voorhis, Phys. Rev. A **72**, 024502 (2005).

[168] Q. Wu and T. Van Voorhis, J. Chem. Phys. **125**, 164105 (2006).

[169] Q. Wu and T. Van Voorhis, J. Chem. Theory. Comput. **2**, 765 (2006).

[170] Q. Wu and T. Van Voorhis, J. Phys. Chem. A **110**, 9212 (2006).

[171] Q. Wu and T. Van Voorhis, J. Chem. Phys. **127**, 164119 (2007).

[172] H. Oberhofer and J. Blumberger, J. Chem. Phys. **131**, 64101 (2009).

[173] H. Oberhofer and J. Blumberger, Angew. Chem. Int. Ed. **49**, 3631 (2010).

[174] Senthilkumar, K. and Grozema, F. C. and Bickelhaupt, F. M. and Siebbeles, L. D. A., J. Chem. Phys. **119**, 9809 (2003).

[175] M. Boero, T. Ikeda, E. Ito and K. Terakura, J. Am. Chem. Soc. **128**, 16798 (2006)

[176] O. A. v. Lilienfeld, D. Sebastiani, I. Tavernelli, and U. Rothlisberger, Phys. Rev. Lett. **93**, 153004 (2004).

[177] P. Maurer, A. Laio, H. W. Hugosson, M. C. Colombo, and U. Röthlisberger, J. Chem. Theor. Comput. **3**, 628 (2007).

[178] P. Blöchl and M. Parrinello, Phys. Rev. B **45**, 9413 (1992).

[179] P. L. Silvestrelli, Phys. Rev. Lett. **100**, 053002 (2008); J. Phys. Chem. A **113**, 5224 (2009).

[180] A. Ambrosetti and P. L. Silvestrelli, Phys. Rev. B **85**, 073101 (2012).

[181] P. L. Silvestrelli and A. Ambrosetti, J. Chem. Phys.**150**, 164109 (2019).

[182] L. V. Slipchenko and M. S. Gordon, Mol. Phys. **107**, 999 (2009).

[183] T. Luo and J. R. Lloyd, J. Heat Transfer **130**, 122403 (2008).

[184] G. Wu and B. Li, Phys. Rev. B **76**, 085424 (2007).

[185] M. Schwörer, B. Breitenfeld, P. Tröster, S. Bauer, K. Lorenzen, P. Tavan, and G. Mathias, J. Chem. Phys., **138**, 244103 (2013).

[186] M. Schwörer, K. Lorenzen, G. Mathias, and P. Tavan, J. Chem. Phys., **142** (2015).

[187] M.P. Bircher and U. Rothlisberger, J. Phys. Chem. Lett., **9** (14), 3886 (2018).

[188] M.P. Bircher, P. Lopez-Tarifa and U. Rothlisberger, J. Chem. Theory Comput., **15** (1), 557 (2019).

Index