A (not so) random walk into CPMD and its input/ouput files

Mauro Boero

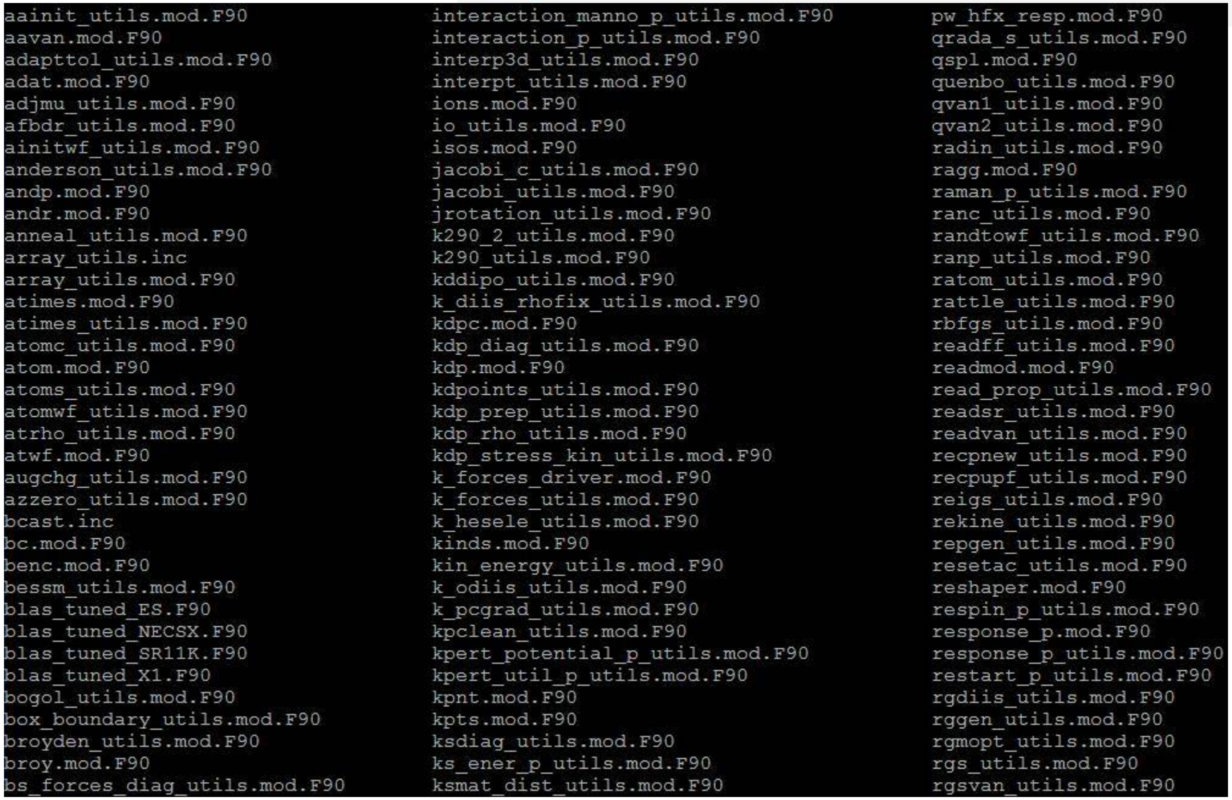
ICube, University of Strasbourg - CNRS, F-67034 Strasbourg, France

and

Institute of Materials and Systems for Sustainability,
Nagoya University – Oshiyama-Shiraishi Group, Nagoya Japan

practical example of input & output files

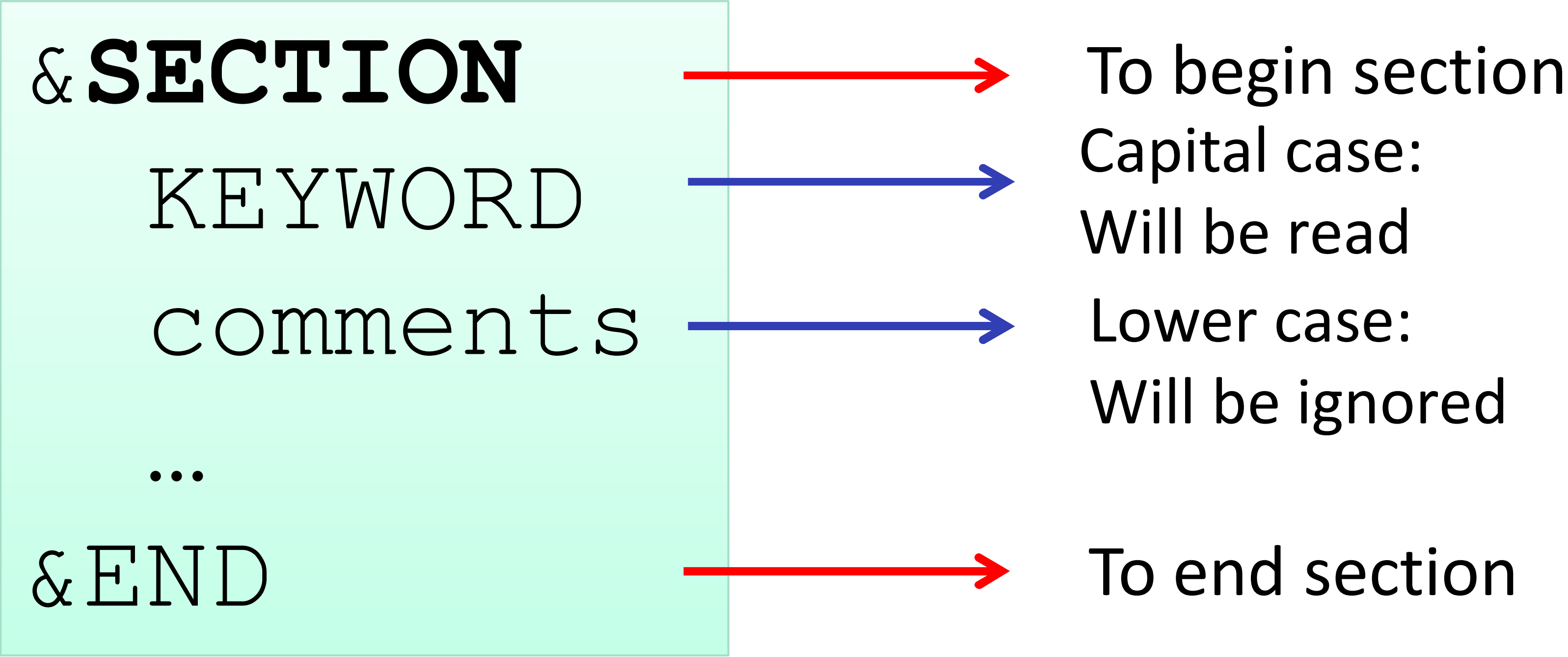
```
&CPMD
  MOLECULAR DYNAMICS
  RESTART WAVEFUNCTIONS COORDINATES VELOCITIES ACCUMULATORS nosee nosep
  restart velocities accumulators nosep nosee
  RESTART LATEST
  TIMESTEP
    9.0
  MAXSTEP
   45000
  EMASS
    1000
  NOSE IONS
   300.0  200.0
  NOSE ELECTRONS
  0.04  600.0
  TRAJECTORY SAMPLE  XYZ
    10
  STORE
   200
  SUBTRACT COMVEL
   1
  SUBTRACT ROTVEL
   1
  RESTFILES
   2
  MEMORY BIg
  DISTRIBUTE FNL ON
&END

&DFT
  NEWCODE
  FUNCTIONAL BLYP
  GC-CUTOFF
  1.0E-07
&END

&SYSTEM
  SYMMETRY
  1
  CELL
  29.17271 1.0  1.0  0  0  0
  CUTOFF
  20.0
&END

&ATOMS
*GE-MT-BLYP.pps  KLEINMAN-BYLANDER
   LMAX=P
  40
     9.69109513160687        8.21319888830087       12.53612134909310
    -2.42721637605981      -10.09425423621903      -10.28590734290584
```



```
aainit_utils.mod.F90                   interaction_manno_p_utils.mod.F90         pw_hfx_resp.mod.F90
aavan.mod.F90                          interaction_p_utils.mod.F90               qrada_s_utils.mod.F90
adapttol_utils.mod.F90                 interp3d_utils.mod.F90                    qspl.mod.F90
adat.mod.F90                           interpt_utils.mod.F90                     quenbo_utils.mod.F90
adjmu_utils.mod.F90                    ions.mod.F90                              qvan1_utils.mod.F90
afbdr_utils.mod.F90                    io_utils.mod.F90                          qvan2_utils.mod.F90
ainitwf_utils.mod.F90                  isos.mod.F90                              radin_utils.mod.F90
anderson_utils.mod.F90                 jacobi_c_utils.mod.F90                    ragg.mod.F90
andp.mod.F90                           jacobi_utils.mod.F90                      raman_p_utils.mod.F90
andr.mod.F90                           jrotation_utils.mod.F90                   ranc_utils.mod.F90
anneal_utils.mod.F90                   k290_2_utils.mod.F90                      randtowf_utils.mod.F90
array_utils.inc                        k290_utils.mod.F90                        ranp_utils.mod.F90
array_utils.mod.F90                    kddipo_utils.mod.F90                      ratom_utils.mod.F90
atimes.mod.F90                         k_diis_rhofix_utils.mod.F90               rattle_utils.mod.F90
atimes_utils.mod.F90                   kdpc.mod.F90                              rbfgs_utils.mod.F90
atomc_utils.mod.F90                    kdp_diag_utils.mod.F90                    readff_utils.mod.F90
atom.mod.F90                           kdp.mod.F90                               readmod.mod.F90
atoms_utils.mod.F90                    kdpoints_utils.mod.F90                    read_prop_utils.mod.F90
atomwf_utils.mod.F90                   kdp_prep_utils.mod.F90                    readsr_utils.mod.F90
atrho_utils.mod.F90                    kdp_rho_utils.mod.F90                     readvan_utils.mod.F90
atwf.mod.F90                           kdp_stress_kin_utils.mod.F90              recpnew_utils.mod.F90
augchg_utils.mod.F90                   k_forces_driver.mod.F90                   recpupf_utils.mod.F90
azzero_utils.mod.F90                   k_forces_utils.mod.F90                    reigs_utils.mod.F90
bcast.inc                              k_hesele_utils.mod.F90                    rekine_utils.mod.F90
bc.mod.F90                             kinds.mod.F90                             repgen_utils.mod.F90
benc.mod.F90                           kin_energy_utils.mod.F90                  resetac_utils.mod.F90
bessm_utils.mod.F90                    k_odiis_utils.mod.F90                     reshaper.mod.F90
blas_tuned_ES.F90                      k_pcgrad_utils.mod.F90                    respin_p_utils.mod.F90
blas_tuned_NECSX.F90                   kpclean_utils.mod.F90                     response_p.mod.F90
blas_tuned_SR11K.F90                   kpert_potential_p_utils.mod.F90           response_p_utils.mod.F90
blas_tuned_X1.F90                      kpert_util_p_utils.mod.F90                restart_p_utils.mod.F90
bogol_utils.mod.F90                    kpnt.mod.F90                              rgdiis_utils.mod.F90
box_boundary_utils.mod.F90             kpts.mod.F90                              rggen_utils.mod.F90
broyden_utils.mod.F90                  ksdiag_utils.mod.F90                      rgmopt_utils.mod.F90
broy.mod.F90                           ks_ener_p_utils.mod.F90                   rgs_utils.mod.F90
bs_forces_diag_utils.mod.F90           ksmat_dist_utils.mod.F90                  rgsvan_utils.mod.F90
```

The Input File

- The input file is divided in ‘Sections’



```
&SECTION
   KEYWORD
   comments
   …
&END
```

To begin section

Capital case:
Will be read

Lower case:
Will be ignored

To end section

- 4 mandatory sections:

CPMD **DFT** **SYSTEM** **ATOMS**

The Input File

- 4 mandatory sections

```
&CPMD
   …
&END

&DFT
   …
&END

&SYSTEM
   …
&END

&ATOMS
   …
&END
```

More elaborated

```
&INFO
   …
&END

&CPMD
   …
&END

&DFT
   …
&END

&VDW
   …
&END

&PROP
   …
&END

&SYSTEM
   …
&END

&ATOMS
   …
&END
```

The Input File

- **INFO** section … your own reminder for your task(s)
- Written in the output as a reminder but NOT used

```
&INFO
  SiO2, 480 atoms
  DFT-GGA-BLYP, CPMD …
  or whatever you want to write
  to describe what you are doing
&END
```

The Input File

- &**CPMD** section: Keywords to select the type of calculation

```
&CPMD
  WAVEFUNCTION OPTIMIZATION
  molecular dynamics cp bo
  INITIALIZE WAVEFUNCTION ATOMS
  TIMESTEP
    3.0
  EMASS
    340.0
  MAXSTEP (or MAXRUNTIME)
    50000
  PCG
  CONVERGENCE ORBITALS
    1.0E-05
&END
```

The calculation:

- OPTIMIZE WAVEFUNCTION
- GEOMETRY OPTIMIZATION
- MOLECULAR DYNAMICS
- KOHN-SHAM ENERGIES
- PROPERTIES
- LINEAR RESPONSE
- EHERENFEST DYNAMICS
- TDDFT
- FREE ENERGY MD
- ...

Possible to restart:

```
RESTART WAVEFUNCTION LATEST
RESTART COORDINATES LATEST
RESTART VELOCITIES LATEST
RESTART ACCUMULATORS LATEST
RESTART NOSEE NOSEP LATEST
```

Or on a single line:

```
RESTART WAVEFUNCTION COORDINATES VELOCITIES NOSEE NOSEP LATEST
```

The Input File

- The **DFT** section to describe the DFT parameters

```
&DFT
   FUNCTIONAL PBE
   GC-CUTOFF
   0.1E-06
&END
```

Eliminate useless numerical noise below DFT accuracy

DFT XC functionals: **LDA, PW, BP, BLYP, PBE, revPBE, HCTH, OLYP,…**
*Hybrid XC functionals :***B3LYP, PBE0, HSE06,…**

The Input File

- In the **DFT** section for hybrid functionals a speed-up can be obtained with the new HF exchange scheme

```
&DFT
   USE_NEW_HFX
   HFX_BLOCK_SIZE
     4
   HFX_DISTRIBUTION DYNAMIC
   FUNCTIONAL PBE0
   GC-CUTOFF
   0.1E-06
&END
```

The Input File

- The VDW section to include van der Waals interactions (Grimme D2, D3 or ab-initio Wannier-based vdW)

```
&CPMD
  …
  VDW CORRECTION
  …
&END
```

```
&VDW
 EMPIRICAL CORRECTION
 VDW PARAMETERS
  ALL DFT-D2
 S6GRIMME
  PBE
 END EMPIRICAL CORRECTION
&END
```

```
&CPMD
  …
  VDW WANNIER
  …
&END
```

```
&VDW
  WANNIER CORRECTION
   VERSION
     2
   FRAGMENT BOND
     1.38
   TOLERANCE WANNIER
     5.0
   TOLERANCE REFERENCE
     1.0
   PRINT INFO FRAGMENT
 END WANNIER CORRECTION
&END
```

The Input File

- The **PROP** section to compute several properties… ***if you have optimized wavefunctions***

in the **CPMD** section use the keyword PROPERTIES:

```
&CPMD
  PROPERTIES
  …
&END
```

```
&PROP
  PROJECT WAVEFUNCTION
  POPULATION ANALYSIS MULLIKEN
&END
```

The Input File

- The **SYSTEM** section to describe the geometry

```
&SYSTEM
   ANGSTROM
   SYMMETRY
      1
   CELL ABSOLUTE DEGREE
      10.00  10.00  10.00  90.00  90.00  90.00
   CUTOFF
      100.0
    KPOINTS MONKHORST-PACK
      2   2   1
&END
```

Cell parameters and atomic coordinates to be read in Å, default is atomic units.

Followed by a number [0 – 14]
0 = Isolated (molecule)
1 = cubic , 8= orthorombic, etc.

Cell sizes a, b and c and angles α, β, γ to be read on the next line.
Default:
a, b/a, c/a, cos α, cos β and cos γ

K-points (if needed) in Monkhorst-Pack representation. Here **k** = 2x2x1

The Input File

- The **ATOMS** section to give the atomic positions

Pseudopotential: The core-valence part

```
&ATOMS
*PP_FILE_NAME.psp  PP_GENERATION_METHOD
  LMAX=P  LOC=P
  N_ATOMS
  x(1) y(1) z(1)
  x(2) y(2) z(2)
  …
  x(N_ATOMS) y(N_ATOMS) z(N_ATOMS)

&END
```

As many times as chemical species (or more)

You need to sort atomic coordinates by chemical species

The Input File

- The **ATOMS** section to give the atomic positions

Example:

```
&ATOMS
  *GE-MT-PBE.pps   KLEINMAN-BYLANDER
   LMAX=P   LOC=P
   1
   4.7367371096   6.6107492731   4.6811927823

  *SE-MT-PBE.pps   KLEINMAN-BYLANDER
   LMAX=P   LOC=P
   4
   -5.9669982281    -8.7467531007   11.2938278730
    1.8904372056   -10.8283093176    4.1633837562
    5.4988512174     4.9899517884   10.6568153345
    0.0059955793    -0.1450182640    4.3887246936
&END
```

The Input File

- The **ATOMS** section: how to generate dummy atoms (e.g. the center of mass of your system)

```
&ATOMS

  *PP_FILE_NAME.psp  PP_GENERATION_METHOD
   LMAX=P  LOC=P
   N_ATOMS
    x(1) y(1) z(1)
    …
   DUMMY ATOMS
    1
    TYPE2  2   1  2
&END
```

Just one dummy atom

TYPE2 = center of mass
Two (2) atoms number 1 and 2 are used. Negative numbers = ALL ATOMS
TYPE1 to TYPE4 are available (see the manual)

The Input File

- The **ATOMS** section to impose constraints to atoms (Blue Moon or fix atom positions)

```
&ATOMS

  *PP_FILE_NAME.psp  PP_GENERATION_METHOD
   LMAX=P  LOC=P
   N_ATOMS
    x(1) y(1) z(1)
    …
   CONSTRAINT
   FIX STRUCTURE
     1
    DIST  1  2  7.20000
   FIX ATOMS
    2
    10 11
   END CONSTRAINT
&END
```

Fix the distance |**R**(1)-**R**(2)| to 7.2 au

Fix the two atoms number 10 and 11, i.e. no force acting on these atoms

The Input File

- The **ATOMS** section for metadynamics

```
&ATOMS
   *PP_FILE_NAME.psp  PP_GENERATION_METHOD
   LMAX=P  LOC=P
   N_ATOMS
    x(1) y(1) z(1)
    …
   META DYNAMICS COLLECTIVE VARIABLE
   DEFINE VARIABLES
    1
    DIST  1  2  SCF 1.00  KCV 1.0  MCV  5.0
   END DEFINE
   HILLS  SHIFT  RCUT  2.0  1.0  = 0.1  0.0002
   TUNING HHEIGHT  = 0.0002  0.0010
   MINSTEPNUM INTERMETA
    100
   MAXSTEPNUM INTERMETA
    100
   CVSPACE BOUNDARIES
    1
    1 0.0002  6.5  10.5
  END METADYNAMICS
&END
```

Distance |**R**(1)-**R**(2)|
Scaling factor, k_α, M_α

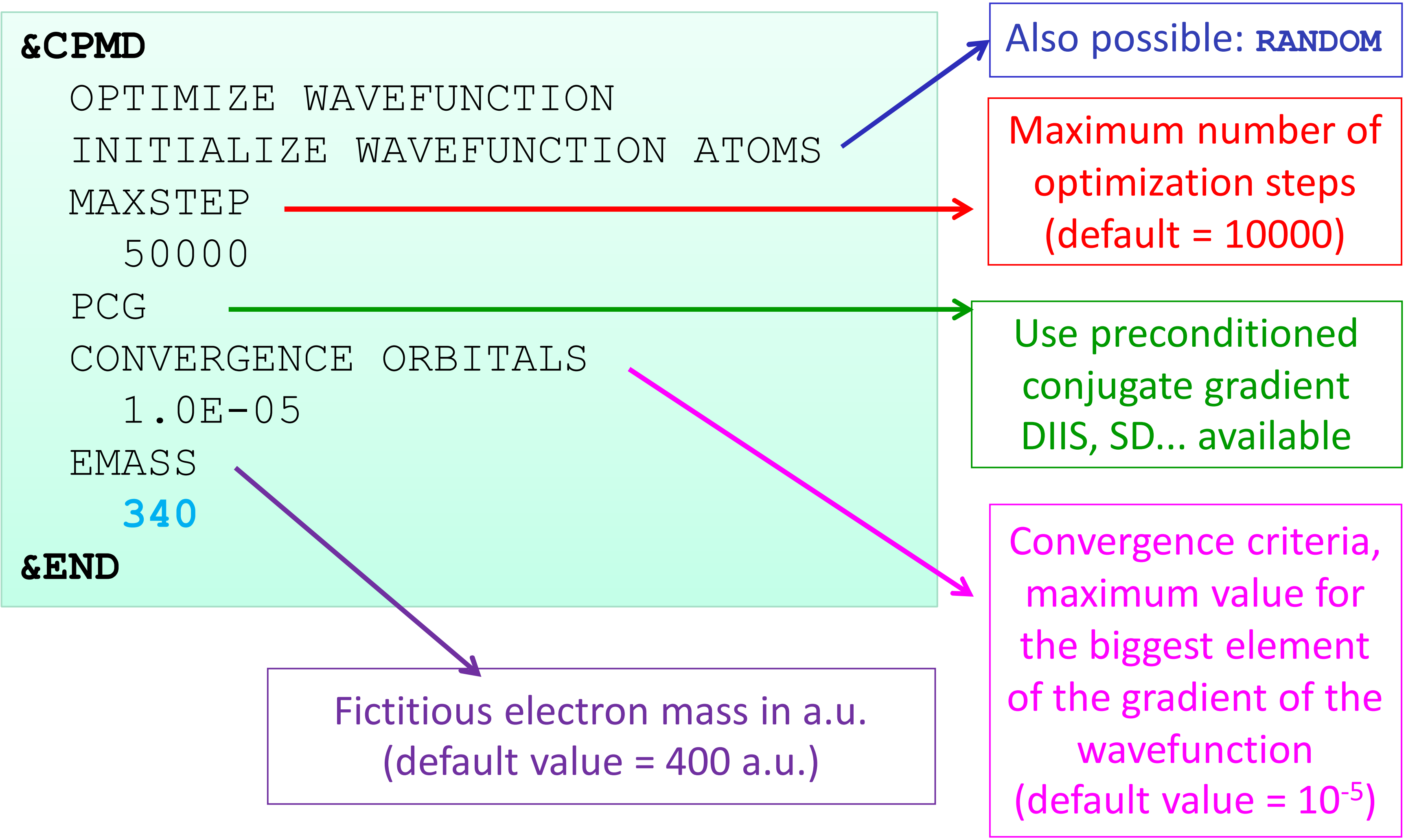
Gaussian penalty function spread

…and amplitude

Let’s do 100 steps between one Gaussian and the next one

Add boundaries to the CVs (if needed)

Simple Example: Optimize Wavefunctions



```
&CPMD
  OPTIMIZE WAVEFUNCTION
  INITIALIZE WAVEFUNCTION ATOMS
  MAXSTEP
    50000
  PCG
  CONVERGENCE ORBITALS
    1.0E-05
  EMASS
    340
&END
```

Also possible: **`RANDOM`**

Maximum number of optimization steps (default = 10000)

Use preconditioned conjugate gradient DIIS, SD... available

Convergence criteria, maximum value for the biggest element of the gradient of the wavefunction (default value = 10^{-5})

Fictitious electron mass in a.u. (default value = 400 a.u.)

…and now Optimize the Structure

```
&CPMD
   OPTIMIZE GEOMETRY
   RESTART WAVEFUNCTION LATEST
   HESSIAN UNIT
   MAXSTEP
      10000
   CONVERGENCE GEOMETRY
      1.0E-04
   EMASS
      600
&END
```

The initial approximate Hessian for geometry optimization

Maximum number of optimization steps (default value = 10000)

Convergence criteria on the maximum force component.
Default = $5.0\mathrm{x}10^{-4}$ a.u.

Fictitious electron mass in a.u.
(default value = 400 a.u.)

The Car-Parrinello dynamics (eventually)

```
&CPMD
  MOLECULAR DYNAMICS CP
  RESTART WAVEFUNCTION COORDINATES LATEST
  MAXSTEP
    100000
  TIMESTEP
    4.0
  EMASS
    340
&END
```

Also possible:
BO
EH
PI
PR
etc.

Maximum number of MD steps
(default value = 10000)

MD time step
default = 5.0 a.u.

Fictitious electron mass in a.u.
(default value = 400 a.u.)

The Car-Parrinello type Quenching

```
&CPMD
  MOLECULAR DYNAMICS CP
  RESTART WAVEFUNCTION COORDINATES LATEST
  ANNEALING IONS
   0.9995
  MAXSTEP
    100000
  TIMESTEP
    4.0
  EMASS
    340
&END
```

Also possible:
BO
EH
PI
PR
etc.

Damping factor for the quench
(Langevin-type damped dynamics)
Warning: Quench your atoms is
conflicting with any thermostat (heating)

Maximum number of
MD steps
(default value = 10000)

MD time step
default = 5.0 a.u.

Fictitious electron mass in a.u.
(default value = 400 a.u.)

The Car-Parrinello with Nosé T-control

```
&CPMD
  MOLECULAR DYNAMICS CP
  RESTART WAVEFUNCTION COORDINATES LATEST
  NOSE IONS
   300  1000
  MAXSTEP
    100000
  TIMESTEP
    4.0
  EMASS
    340
&END
```

Also possible:
BO
EH
PI
PR
etc.

Nosé-Hoover thermostat with a target temperature of 300 K and an oscillation frequency of 1000 cm^{-1}

Maximum number of MD steps
(default value = 10000)

MD time step
default = 5.0 a.u.

Fictitious electron mass in a.u.
(default value = 400 a.u.)

The dipole dynamics
(useful for computing IR spectra)

```
&CPMD
   MOLECULAR DYNAMICS CP
   RESTART WAVEFUNCTION COORDINATES LATEST
   MAXSTEP
      100000
   TIMESTEP
      4.0
   EMASS
      340
   DIPOLE DYNAMICS SAMPLE
     1
&END
```

Also possible:
BO
PR
etc....

Compute the total dipole moment **P**(*t*) of the system at each step

The IR absorption coefficient will then be

$$\alpha(\omega) = [4\pi\ \omega \tanh(\beta\hbar\omega)] \int e^{-i\omega t} < \mathbf{P}(t)\mathbf{P}(0) > dt \cdot 1/3\hbar\ n(\omega) c V$$

Wannier states & dipole dynamics

```
&CPMD
  MOLECULAR DYNAMICS CP
  ...
  DIPOLE DYNAMICS WANNIER
  WANNIER TYPE VANDERBILT
  WANNIER OPTIMIZATION JACOBI
  WANNIER PARAMETERS
   0.2  1.E-05  0.01  200
  WANNIER REFERENCE
   0.000  0.000  0.000
&END
```

Also possible:
BO
PR
etc....

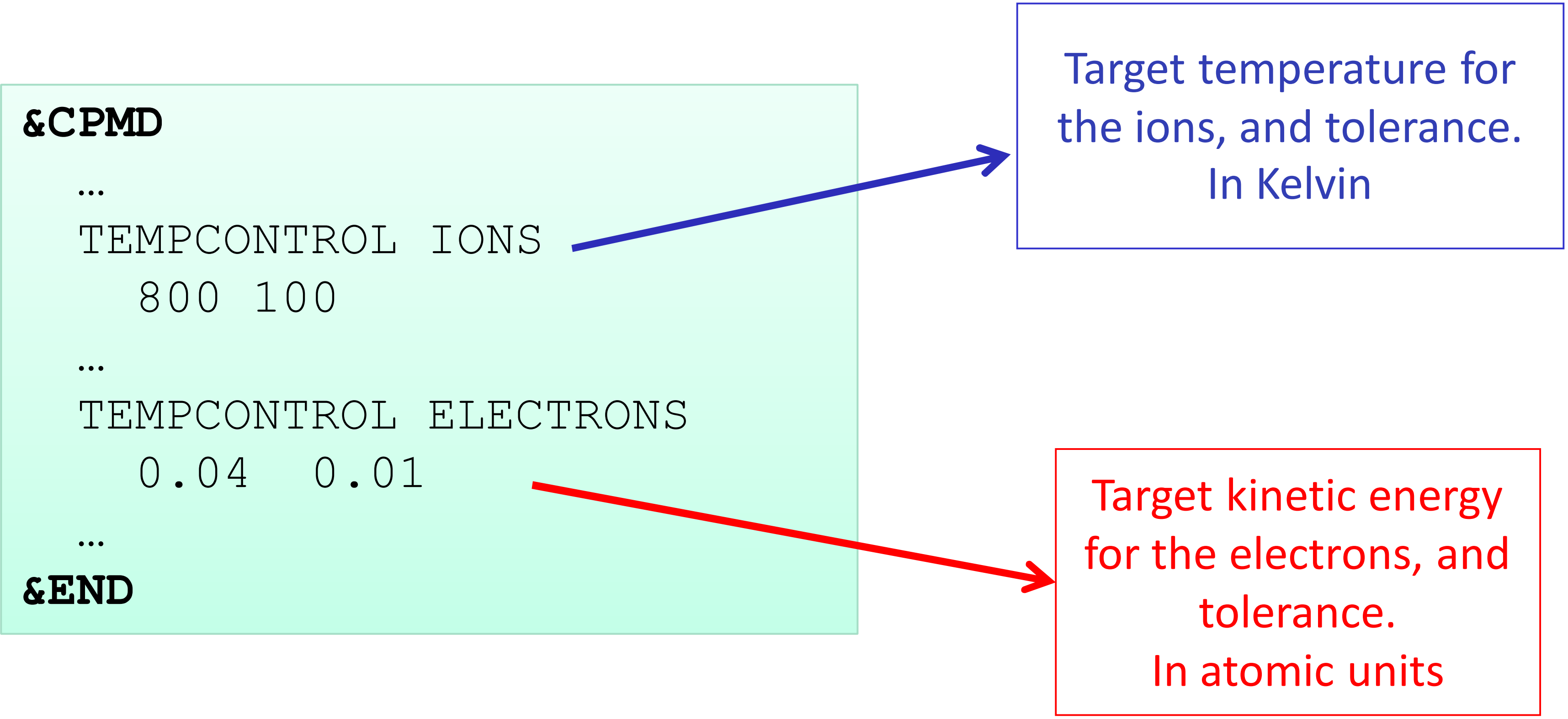
Compute the dipole moment **P**(*t*) from maximally localized Wannier functions and centers

Calculation method Jacobi rotations or steepest descent (SD)

Reference point for Wannier calculations (here is the origin of your supercell)

Optimization step (0.2), accuracy (10^{-5}) , amplitude for the initial random unitary rotation (0.01), max. n. of iterations (200)

Controlling the temperature by rescaling

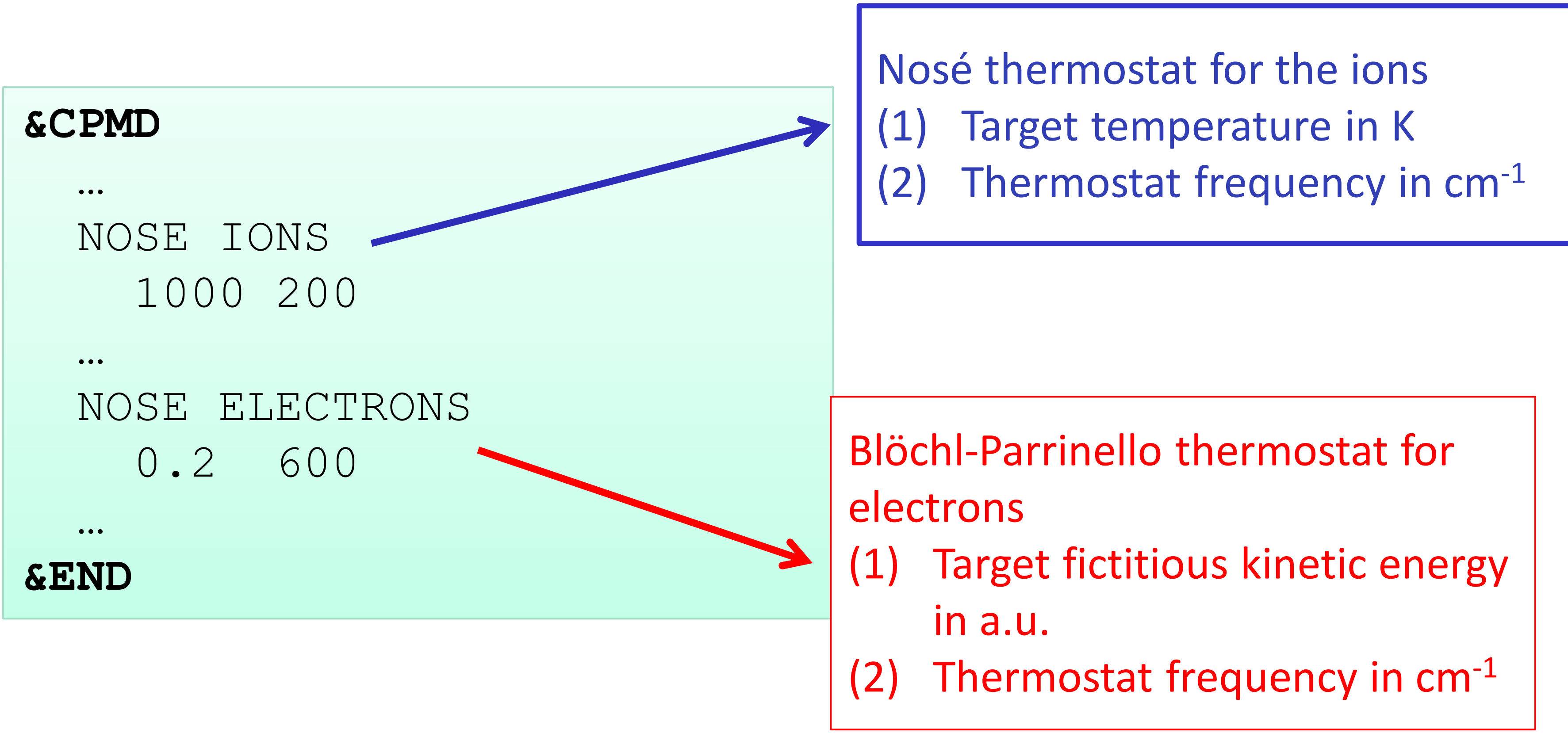


```
&CPMD
  …
  TEMPCONTROL IONS
    800 100
  …
  TEMPCONTROL ELECTRONS
    0.04  0.01
  …
&END
```

Target temperature for the ions, and tolerance.
In Kelvin

Target kinetic energy for the electrons, and tolerance.
In atomic units

Controlling the temperature by Nosé-Hoover thermostat and chains + Blöchl-Parrinello



```
&CPMD
   …
   NOSE IONS
      1000 200
   …
   NOSE ELECTRONS
      0.2   600
   …
&END
```

Nosé thermostat for the ions

(1) Target temperature in K

(2) Thermostat frequency in cm^{-1}

Blöchl-Parrinello thermostat for electrons

(1) Target fictitious kinetic energy in a.u.

(2) Thermostat frequency in cm^{-1}

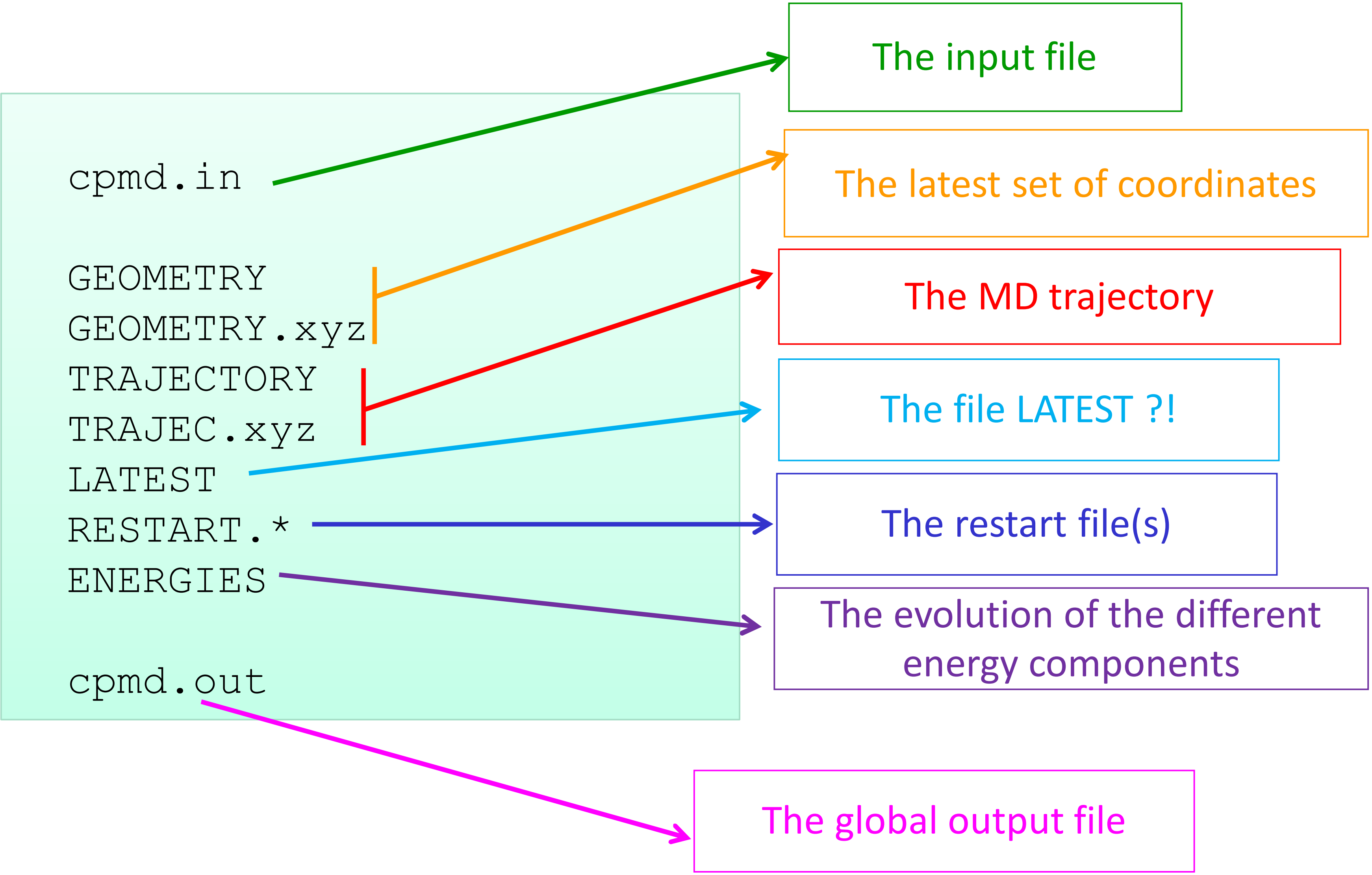
Computing Kohn-Sham eigenvalues and eigenfunctions

We need to use an optimized wavefunction !

```
&CPMD
  RESTART WAVEFUNCTION COORDINATES LATEST
  KOHN-SHAM ENERGIES
    100
  …
&END
```

The number of unoccupied electronic states, depends on the system and has to be large enough to include at least a few empty electronic states to get the HOMO-LUMO gap

The Output files
(depending on the type of calculation)



```
cpmd.in

GEOMETRY
GEOMETRY.xyz
TRAJECTORY
TRAJEC.xyz
LATEST
RESTART.*
ENERGIES

cpmd.out
```

The input file

The latest set of coordinates

The MD trajectory

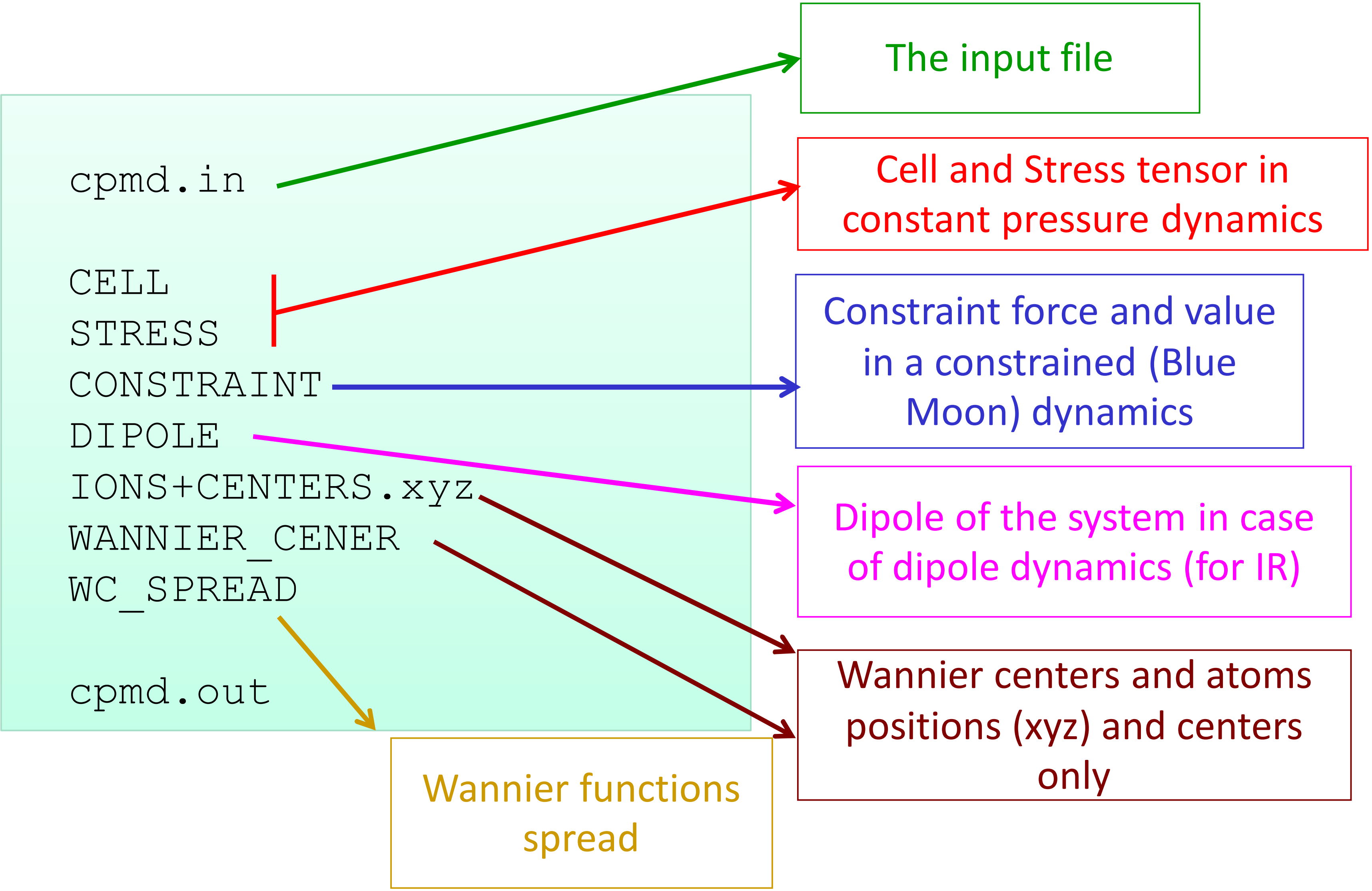
The file LATEST ?!

The restart file(s)

The evolution of the different energy components

The global output file

More Output files
(depending on the type of calculation)



cpmd.in

CELL
STRESS
CONSTRAINT
DIPOLE
IONS+CENTERS.xyz
WANNIER_CENER
WC_SPREAD

cpmd.out

The input file

Cell and Stress tensor in constant pressure dynamics

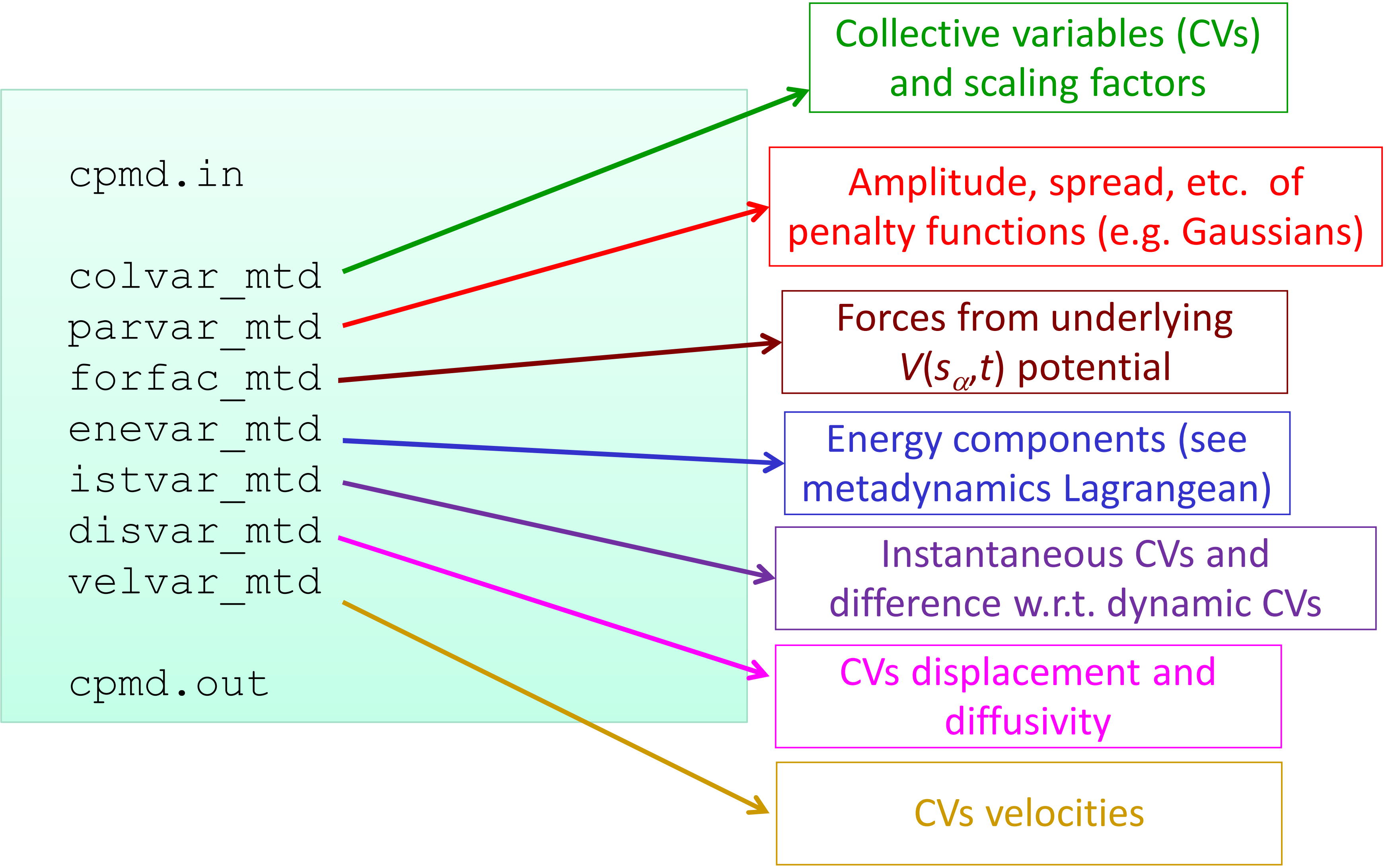
Constraint force and value in a constrained (Blue Moon) dynamics

Dipole of the system in case of dipole dynamics (for IR)

Wannier centers and atoms positions (xyz) and centers only

Wannier functions spread

And even more output files (for metadynamics)



```
cpmd.in

colvar_mtd
parvar_mtd
forfac_mtd
enevar_mtd
istvar_mtd
disvar_mtd
velvar_mtd

cpmd.out
```

Collective variables (CVs) and scaling factors

Amplitude, spread, etc. of penalty functions (e.g. Gaussians)

Forces from underlying $V(s_{\alpha},t)$ potential

Energy components (see metadynamics Lagrangean)

Instantaneous CVs and difference w.r.t. dynamic CVs

CVs displacement and diffusivity

CVs velocities

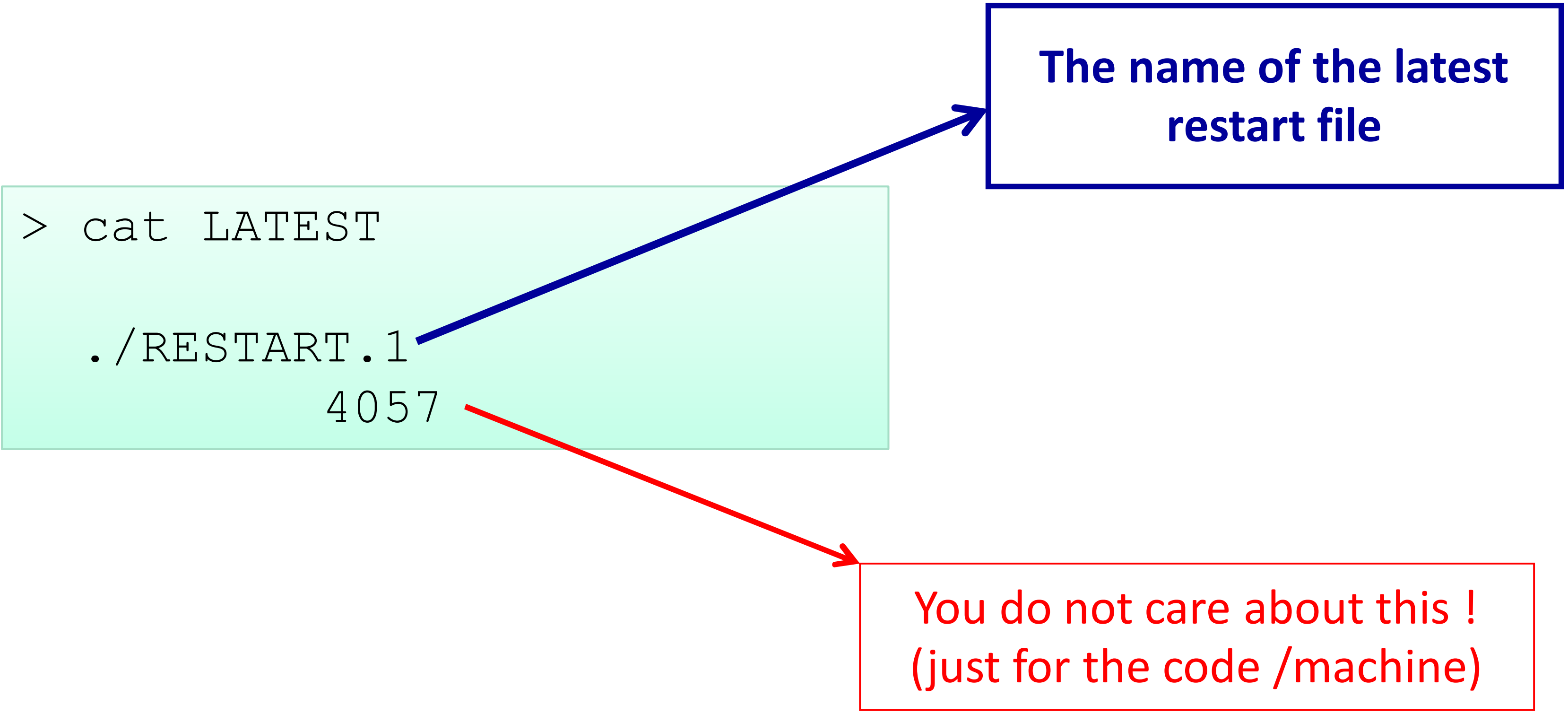
The Output files
(tricky stuff to write in parallel the RESTART)

```
&CPMD
  RESTART WAVEFUNCTION COORDINATES LATEST
  USE_MPI_IO
  USE_IN_STREAM
  USE_OUT_STREAM
…
&END
```

MPI are used to write in streaming the RESTART.* files. Particularly useful when these binary I/O objects are huge

WARNING: Requires the (pre) compilation option -D__HAS_BF_STREAM_IO

The LATEST file ?

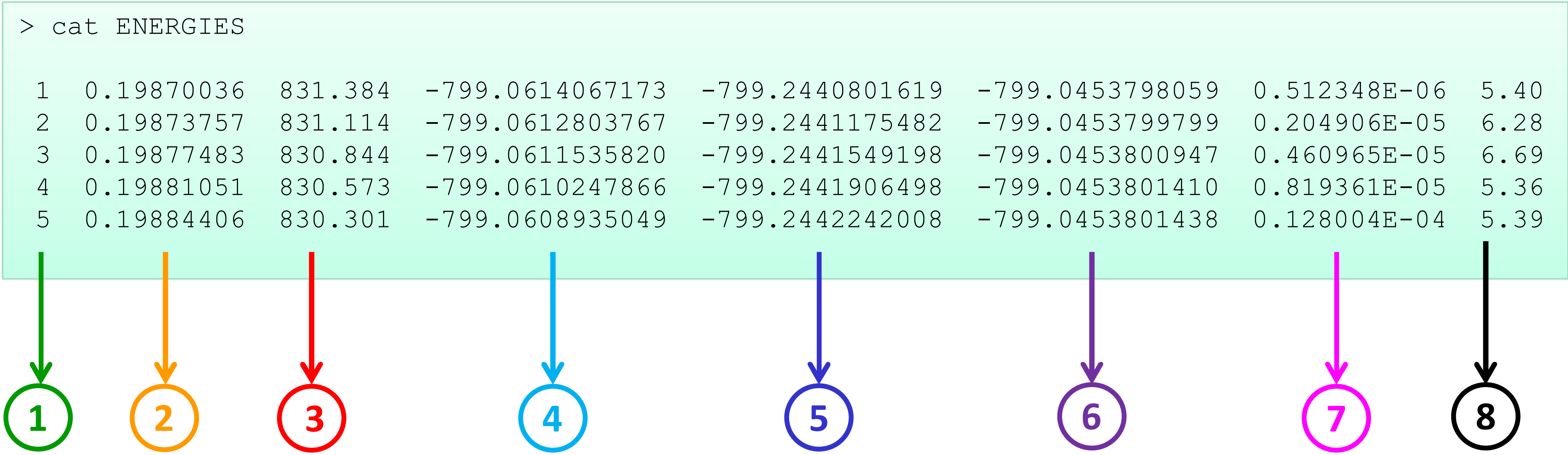


```
> cat LATEST

   ./RESTART.1
          4057
```

The name of the latest restart file

You do not care about this !
(just for the code /machine)

The ENERGIES file



```
> cat ENERGIES

 1  0.19870036  831.384  -799.0614067173  -799.2440801619  -799.0453798059  0.512348E-06  5.40
 2  0.19873757  831.114  -799.0612803767  -799.2441175482  -799.0453799799  0.204906E-05  6.28
 3  0.19877483  830.844  -799.0611535820  -799.2441549198  -799.0453800947  0.460965E-05  6.69
 4  0.19881051  830.573  -799.0610247866  -799.2441906498  -799.0453801410  0.819361E-05  5.36
 5  0.19884406  830.301  -799.0608935049  -799.2442242008  -799.0453801438  0.128004E-04  5.39
```

1 MD step number

2 Fictitious kinetic energy for e-

3 Temperature (K)

4 E_{KS} = Kohn-Sham energy (a.u.)

5 $E_{classic} = E_{KS}$ + Ionic Kin.E (a.u.)

6 $E_{HAM} = E_{KinElec} + E_{classic}$ (a.u.) This is the conserved parameter in CPMD

7 MSD (a.u.^{2})

8 CPU time for one step

And all the rest ?

Use:

- The manual (manual.tex provided with the code)
- The source code (always better)
- The regtests
- ***Ask anybody who knows the code and you can reach***

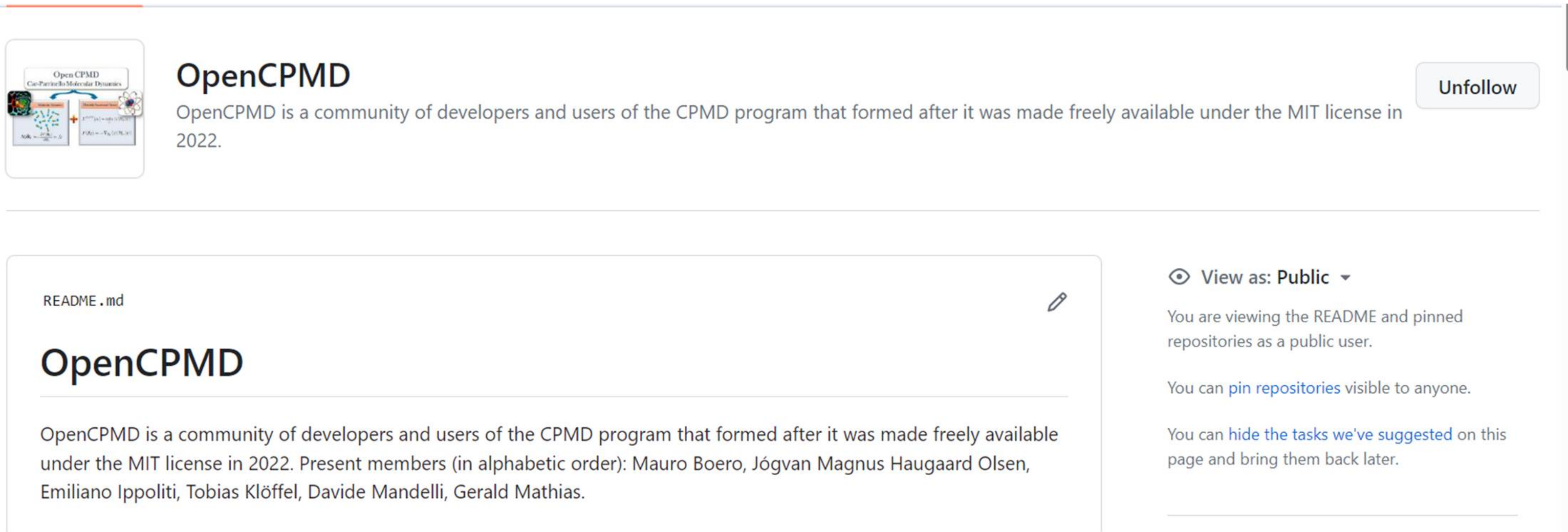
…about CPMD:

- Versions 3.* no longer developed/supported since November 2013
- **Last release: 4.3**
- Fortran 2003/8/18 (c/c++ @ sysdepend.c)
- Organized in modules & structure of the code slightly changed (more rational):

```
-|---  CPMD ---|---- src
       |       |---- configure
       |       |---- doc
       |       |---- scripts
       |       |---- modules -|-MM_Interface
       |                      |-Gromos/Amber
       |---- addons--|cpmd2cube
       |             | plumed_for_CPMD
       |---- regtests
```

- Code compilable via a (linked) configure.sh script in the CPMD directory.

See you at https://github.com/OpenCPMD



OpenCPMD

OpenCPMD is a community of developers and users of the CPMD program that formed after it was made freely available under the MIT license in 2022.

OpenCPMD

OpenCPMD is a community of developers and users of the CPMD program that formed after it was made freely available under the MIT license in 2022. Present members (in alphabetic order): Mauro Boero, Jógvan Magnus Haugaard Olsen, Emiliano Ippoliti, Tobias Klöffel, Davide Mandelli, Gerald Mathias.

View as: Public

You are viewing the README and pinned repositories as a public user.

You can pin repositories visible to anyone.

You can hide the tasks we've suggested on this page and bring them back later.