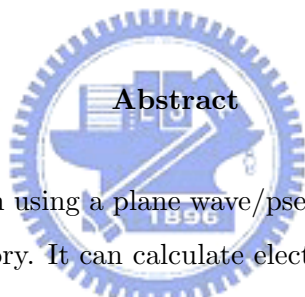


Computation of the Ground-State Energy by CPMD

Student: Bi-Wei Huang **Advisor:** Dr.Li-Ming Yeh

Department of Applied Mathematics
National Chiao Tung University
Hsinchu, Taiwan, R.O.C.



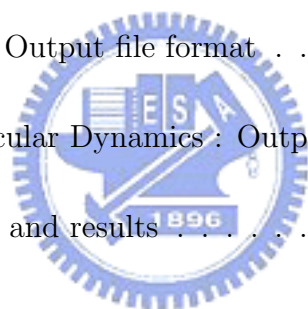
Abstract

CPMD is the program using a plane wave/pseudopotential implementation of density functional theory. It can calculate electronic properties. The aim of this paper is to calculate the ground state energy of isolated H_2 , H_2O molecule, and 32 water molecules($H_{64}O_{32}$) by using the three methods in CPMD: optimize wavefunction, optimize geometry and Car-Parrinello molecular dynamics and then compare their computation process and results.

Contents

1	Introduction	1
1.1	Density functional theory	1
1.2	The Kohn-Sham equations	2
1.3	Local-Density approximation	3
1.4	Bloch's theorem	3
1.5	Pseudopotential approximation	4
1.6	Computational procedure with conventional matrix diagonalization	5
2	CPMD program	7
2.1	Optimize Wavefunction by DIIS	7
2.1.1	DIIS Method	7
2.1.2	Wavefunction Optimization: Input File Format	9
2.2	Optimize Geometry by GDIIS	11
2.2.1	Geometry Optimization: Input File Format	12
2.3	Car-Parrinello Molecular Dynamics	13
2.3.1	Molecular-dynamics Lagrangian	15
2.3.2	Molecular-dynamics equations of motion	16
2.3.3	Numerical Integration	16
2.3.4	Integration of equations of motion	18
2.3.5	Car-Parrinello Molecular Dynamics: Input File Format	21

3	Procedure	23
3.1	Optimize Geometry: Procedure	24
3.1.1	GEOPT.F: Initialization	24
3.1.2	GEOPT.F: The main loop for approximation	25
3.2	Car-Parrinello Molecular Dynamics: Procedure	27
3.2.1	MDMAIN.F: Initialization	27
3.2.2	MDMAIN.F: The main loop for approximation	29
4	Discussion and Output File Format	31
4.1	Optimize wavefunction: Output file format	31
4.2	Optimize geometry: Output file format	39
4.3	Car-Parrinello Molecular Dynamics : Output file format	43
4.4	Computational tests and results	46
A	Appendix	51
A.1	Optimize Geometry Procedure: Initialization	51
A.2	Optimize Geometry Procedure: Main Loop	52
A.3	Molecular Dynamics CP Procedure: Initialization	55
A.4	Molecular Dynamics CP Procedure: Main Loop	56



1 Introduction

The aim of an electronic structure calculation is to calculate properties of a system from only the knowledge of the component atoms. When done without explicit recourse to experimental data (with the exception of the use of fundamental constants and comparison with experimental results), these can be termed *ab initio* calculations. Electronic structure calculations have been proved useful in many areas of condensed matter physics and chemistry, especially with the rapid growth in computer power. An alternative to wavefunction based methods, Density Functional Theory (DFT) has become widely used. DFT provides an (in principle) exact method for calculating the energy of a system of interacting electrons in terms of a set of single electron equations.

1.1 Density functional theory

Density functional theory (DFT) is developed by Hohenberg and Kohn (1964) and Kohn and Sham (1965). Hohenberg and Kohn proved that the total energy, including exchange and correlation, of an electron gas (even in the presence of a static external potential) is a unique functional of the electron density. The minimum value of the total-energy functional is the ground-state energy of the system, and the density that yields this minimum value is the exact single-particle ground-state density. Kohn and Sham then showed how it is possible, formally, to replace the many-electron problem by an exactly equivalent set of self-consistent one-electron equations.

The Kohn-Sham total energy functional for a set of single occupied electronic states ψ_i can be written

$$E[\{\psi_i\}] = \sum_i \int \psi_i \left[-\frac{\hbar^2}{2m} \nabla^2 \psi_i \right] d^3r + \int V_{ion}(r) n(r) d^3r + \frac{e^2}{2} \int \frac{n(r)n(r')}{|r-r'|} d^3r d^3r' + E_{XC}[n(r)] + E_{ion}(\{R_I\}), \quad (1.1)$$

where E_{ion} is the Coulomb energy associated with interactions among the nuclei (or ions) at positions $\{R_I\}$, V_{ion} is the static total electron-ion potential, $n(r)$ is the

electronic density in term of occupied single-particle orthonormal given by

$$n(r) = \sum_i |\psi_i|^2, \quad (1.2)$$

and $E_{xc}[n(r)]$ is the exchange-correlation functional.

Only the minimum value of the Kohn-Sham energy functional has physical meaning. At the minimum, the Kohn-Sham energy functional is equal to the ground-state energy of the system of electrons with the ions in positions $\{R_I\}$.

1.2 The Kohn-Sham equations

It is necessary to determine the set of wave functions ψ_i that minimize the Kohn-Sham energy functional. These are given by the self-consistent solutions to the Kohn-Sham equations (Kohn and Sham, 1965):

$$\left[\frac{-\hbar^2}{2m} \nabla^2 + V_{ion}(r) + V_H(r) + V_{XC}(r) \right] \psi_i(r) = \varepsilon_i \psi_i(r), \quad (1.3)$$

where ψ_i is the wave function of electronic state i , ε_i is the Kohn-Sham eigenvalue, V_H is the Hartree potential of the electrons given by

$$V_H = e^2 \int \frac{n(r')}{|r - r'|}. \quad (1.4)$$

The exchange-correlation potential, V_{XC} , is given formally by the functional derivative

$$V_{XC}(r) = \frac{\delta E_{XC}[n(r)]}{\delta n(r)}, \quad (1.5)$$

The Kohn-Sham equations represent a mapping of the interacting many-electron system onto a system of noninteracting electrons moving in an effective potential due to all the other electrons. If the exchange-correlation energy functional were known exactly, then taking the functional derivative with respect to the density would produce an exchange-correlation potential that included the effects of exchange and correlation exactly.

The Kohn-Sham equations are a set of eigenequations, and the terms within the bracket in Eq.(1.3) can be regarded as a Hamiltonian. The bulk of the work involved

in a total-energy pseudopotential calculation is the solution of this eigenvalue problem once an approximate expression for the exchange-correlation energy is given.

1.3 Local-Density approximation

The Hohenberg-Kohn theorem provides some motivation for using approximate methods to describe the exchange-correlation energy as a function of the electron density. The simplest method of describing the exchange-correlation energy of an electronic system is to use the *Local-density approximation*(LDA). In the local-density approximation the exchange-correlation energy of an electronic system is constructed by assuming that the exchange-correlation energy per electron at a point \mathbf{r} , in the electron gas, $\varepsilon_{XC}(r)$, is equal to the exchange-correlation energy per electron in a homogeneous electron gas that has the same density as the electron gas at point \mathbf{r} . Thus

$$E_{XC}[n(r)] = \int \varepsilon_{XC}(r)n(r)d^3r \quad (1.6)$$

and

$$\frac{\delta E_{XC}[n(r)]}{\delta n(r)} = \frac{\partial [n(r)\varepsilon_{XC}(r)]}{\partial n(r)}$$

with

$$\varepsilon_{XC}(r) = \varepsilon_{XC}^{hom}[n(r)]$$

where $\varepsilon_{XC}(r)$ is the exchange-correlation energy density(single variable) function of uniform electron gas. The local-density approximation assumes that the exchange-correlation energy functional is purely local. Several parameterizations exist for the exchange-correlation energy of a homogeneous electron gas, all of which lead to total-energy results that are very similar.

1.4 Bloch's theorem

Bloch's Theorem states that in a periodic solid each electronic wave function can be written as the product of a cell-periodic part and a wavelike part,

$$\psi_i(\mathbf{r}) = \exp[i\mathbf{k} \cdot \mathbf{r}]f_i(\mathbf{r}) \quad (1.7)$$

The cell-periodic part of the wave function can be expanded using a basis set consisting of a discrete set of plane waves whose wave vectors are reciprocal lattice vectors of the crystal,

$$f_i(\mathbf{r}) = \sum_G c_{i,G} \exp[i\mathbf{G} \cdot \mathbf{r}] \quad (1.8)$$

where the reciprocal lattice vectors \mathbf{G} are defined by $\mathbf{G} \cdot \mathbf{l} = 2\pi m$ for all \mathbf{l} where \mathbf{l} is a lattice vector of the crystal and m is an integer. Therefore each electronic wave function can be written as a sum of plane waves,

$$\psi_i(\mathbf{r}) = \sum_G c_{i,k+G} \exp[i(\mathbf{k} + \mathbf{G}) \cdot \mathbf{r}] \quad (1.9)$$

When plane waves are used as a basis set for the electronic wave functions, the Kohn-Sham equations assume a particularly simple form. Substitution of Eq.(1.9) into (1.3) and integration over \mathbf{r} gives the secular equation

$$\sum_{G'} \left[\frac{\hbar^2}{2m} |k + G|^2 \delta_{GG'} + V_{ion}(G - G') + V_H(G - G') + V_{XC}(G - G') \right] c_{i,k+G'} = \varepsilon_i c_{i,k+G}. \quad (1.10)$$

In this form, the kinetic energy is diagonal, and the various potentials are described in terms of their Fourier transforms. Solutions of Eq.(1.10) proceeds by diagonalization of a Hamiltonian matrix whose matrix elements $H_{k+G,k+G'}$ are given by terms in the brackets above. The size of the matrix is determined by the choice of cutoff energy $(\hbar^2/2m)|k + G_c|^2$, and will be intractable large for systems that contain both valence and core electrons. This is a severe problem, but it can be overcome by the use of the pseudopotential approximation, as will be discussed in next section.

1.5 Pseudopotential approximation

Although Bloch's theorem states that the electronic wave functions can be expanded using a discrete set of plane wave, a plane-wave basis set is usually very poorly suited to expanding electronic wave functions because a very large number of plane wave needed to expand the tightly bound core orbitals and to follow the rapid oscillations of the wave functions of the valence electrons in the core region. An extremely

large plane-wave basis set would be required to perform an all-electron calculation, and a vast amount of computational time would be required to calculate the electronic wave functions. The pseudopotential approximation allows the electronic wave functions to be expanded using a much smaller number of plane-wave basis states.

It is well known that most physical properties of solids are dependent on the valence electrons to a much greater extent than on the core electrons. The pseudopotential approximation exploits this by removing the core electrons and by replacing them and the strong ionic potential by a weaker pseudopotential that acts on a set of pseudo wave functions rather than the true valence wave functions. An ionic potential, valence wave function and the corresponding pseudopotential and pseudo wave function are illustrated schematically in Fig.1.

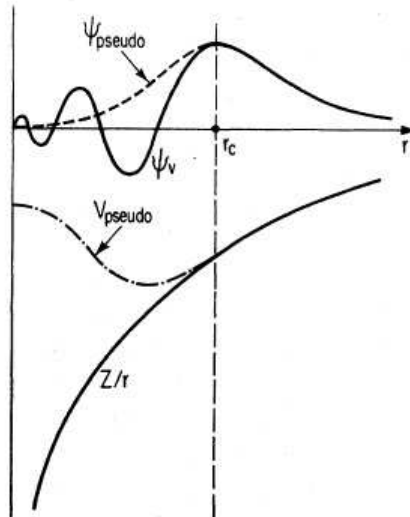


FIG.1. Schematic illustration of all-electron (solid lines) and pseudoelectron (dashed lines) potentials and their corresponding wave function. The radius at which all-electron and pseudoelectron values match is designated r_c .

1.6 Computational procedure with conventional matrix diagonalization

The sequence of steps required to carry out a total-energy pseudopotential calculation with conventional matrix diagonalization techniques is shown in the flow

diagram in Fig.2. The procedure requires an initial guess for the electronic charge density, from which the Hartree potential and the exchange-correlation potential can be calculated. The Hamiltonian matrices for each of the \mathbf{k} points included in the calculation must be constructed, as in Eq.(1.10), and diagonalized to obtain the Kohn-Sham eigenstates. These eigenstates will normally generate a different charge density from the one originally used to construct the electronic potential, and hence a new set of Hamiltonian matrices must be constructed using the new electronic potential. The eigenstates of the new Hamiltonians are obtained, and the process is repeated until the solutions are self-consistent. In practice the new electronic potentials generated by the old and the new eigenstates, since this speeds the convergence to self-consistency. To complete the total-energy calculation, tests should be performed to ensure that the total energy is converged both as a function of the number of \mathbf{k} points and as a function of the cutoff energy for the plane-wave basis set.

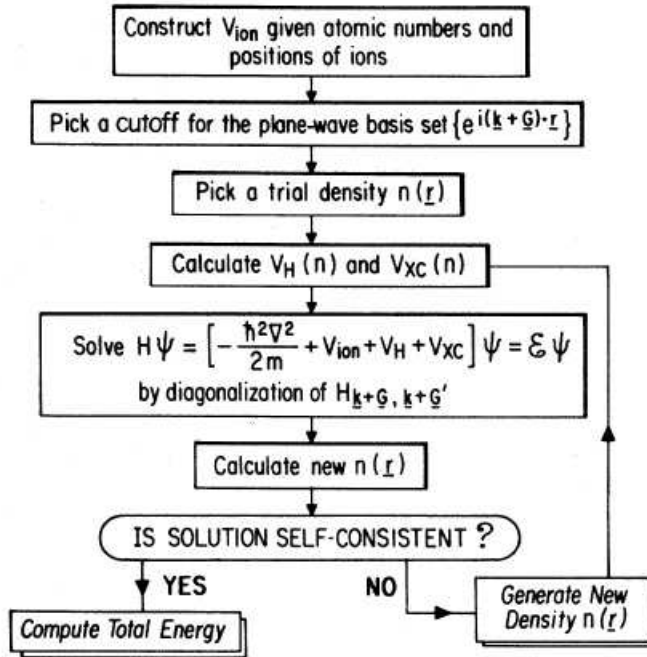


FIG.2. Flow chart describing the computational procedure for the calculation of the total energy of a solid, using conventional matrix diagonalization.

2 CPMD program

CPMD is an ab initio electronic structure and molecular dynamics (MD) program using a plane wave/pseudopotential implementation of density functional theory. It is mainly targeted at Car-Parrinello MD simulations, but also supports wavefunction optimizations, geometry optimizations, Born-Oppenheimer MD, path integral MD, response functions, excited states and calculation of some electronic properties. In this paper we use CPMD version 3.9.2.

2.1 Optimize Wavefunction by DIIS

There are three methods available for optimizing wavefunction : preconditioned conjugate gradient(PCG), steepest decent and direct inversion in the iterative subspace(DIIS). The default method (DIIS) is the most powerful method but sometimes it needs some assistance from outside. We have implemented some empirical rules for restarting the DIIS procedure in case that the optimization gets stuck. If the DIIS converged gets stuck, stop the run and restart using preconditioned conjugate gradient with line search(PCG MINIMIZE). 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. The aim of this method is to find a better set of initial wavefunction.

2.1.1 DIIS Method

We start with the Kohn-Sham eigenvalue problem

$$H\psi_j = \varepsilon_j\psi_j, \tag{2.1}$$

writing Eq.(2.1) for iteration i as

$$H^i\{\psi_j\}^i = \{\varepsilon_j\}^i\{\psi_j\}^i. \tag{2.2}$$

Suppose that we have a set of trial vectors $\phi^i = \{\psi_j\}^i$ of the iteration as a sum of the converged solution ϕ^F plus an error vector e^i :

$$\phi^i = \phi^F + e^i. \quad (2.3)$$

The DIIS method assumes that a good approximation to the final solution ϕ^F in subspace of ϕ^i can be obtained in a least squares sense by writing

$$\phi^{m+1} = \sum_{i=1}^m d_i \phi^i, \quad (2.4)$$

where The coefficients d_i are subject to the restriction

$$\sum_{i=1}^m d_i = 1, \quad (2.5)$$

and therefore

$$e^{m+1} = \sum_{i=1}^m d_i e^i. \quad (2.6)$$

When $e^{m+1} = 0$, then $\phi^{m+1} = \phi^F$ and convergence is achieved. In real-life applications, the number of degree of freedom is much larger than m . Therefore, in finite number of iterations, the condition $e^{m+1} = 0$ can be achieved only in a mean squares sense. This leads to a set of $m+1$ linear equations from which the d_i can be calculated:

$$\begin{bmatrix} b_{11} & b_{12} & \dots & b_{1m} & 1 \\ b_{21} & b_{22} & \dots & b_{2m} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ b_{m1} & b_{m2} & \dots & b_{mm} & 1 \\ 1 & 1 & \dots & 1 & 0 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \\ -\lambda \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix},$$

where the $b_{i,j} = e_i^T e_j$ and λ is the Lagrangian multiplier arising from the previously mentioned constraint. The error vectors e^i are not known, but within a quadratic approximation they are given by

$$e^i = K^{-1} g^i, \quad (2.7)$$

where g and K are first and second derivatives of the energy density functional with respect to the coefficients ϕ . In the same approximation, K is constant and the new

trial vector becomes

$$\phi = \phi^{m+1} - K^{-1}g^{m+1}, \quad (2.8)$$

where the first derivative of the energy density functional is estimated to be

$$g^{m+1} = \sum_{i=1}^m d_i g^i. \quad (2.9)$$

In most applications, the matrix K is too large to be stored and inverted. Therefore approximations are needed. One could choose K to be a constant diagonal matrix. Another popular choice is to take K as the diagonal part of the Kohn-Sham Hamiltonian in the plane-wave representation.

2.1.2 Wavefunction Optimization: Input File Format

For calculation we will need the input file *1-h2o-wave.inp* and the pseudo-potential file *H-MT-BLYP.psp*, *O-MT-BLYP.psp*. Now let's have a look at the input file. The input is organized in sections which start with **&NAME** and end with **&END**. Everything outside those sections is ignored. Also all keywords have to be in upper case or else they will be ignored. The sequence of the sections does not matter, nor does the order of keywords, except where noted in the manual. A minimal input file must have a **&CPMD**, **&SYSTEM** and an **&ATOMS** section.

```
&INFO
  Isolate water molecule.
  Single point calculation.
&END
```

The input file starts with an (optional) **&INFO** section. This section allows you to put comments about the calculation into the input file and they will be repeated in the output file. This can be very useful to match input and output files.

```

&CPMD
OPTIMIZE WAVEFUNCTION
CONVERGENCE ORBITALS
  1.0d-7
STRUCTURE BONDS ANGLES
&END

```

This first part of **&CPMD** section instructs the program to do a wavefunction optimization (i.e. a single point calculation) with a convergence criterion 1.0d-7 (the default is 1.0d-5 for the wavefunction(**ORBITALS**)) and print structure information(**BONDS,ANGLES**) at the end of the run.

```

&SYSTEM
SYMMETRY
  1
CELL
  20.0  1.0  1.0  0.0  0.0  0.0
CUTOFF
  70.0
&END

```

The **&SYSTEM** section contains various parameters related to the simulations cell and the representation of the electronic structure. The keyword **SYMMETRY** defines the supercell symmetry type. You can put a number or a keyword in the next line (i.e. **1**: Simple **CUBIC**, **2**: FACE CENTERED CUBIC (**BCC**), **3**: BODY CENTERED CUBIC (**BCC**) and for more details on the input syntax, please refer to the CPMD manual). The keyword **CELL** specifies the size of the supercell. Six numbers in the following order have to be provided: $a, \frac{b}{a}, \frac{c}{a}, \cos\alpha, \cos\beta, \cos\gamma$. For cubic phases, a is the lattice parameter which describes the distance between the adjacent corners of the cubic. It does not give the distance from an atom to its nearest neighbor. Furthermore for cubic lattice, lattice constant a will be equal to b and c , $\alpha = \beta = \gamma = 90^\circ$. The last keyword **CUTOFF** defines the size of the basis set as well as the plane wave cutoff.

```

&DFT
FUNCTIONAL BLYP
GC-CUTOFF
  1.0d-06
&END

```

The **&DFT** section is used to select the density functional and related parameters. In this case we go with the Becke-Lee-Yang-Parr(BLYP) functionals (default is the local density approximation(LDA)). The keyword **GC-CUTOFF** specifies the density cutoff for calculation of the gradient correction. The default value is 10^{-8} . Experience showed that for a small **CUTOFF** value a bigger values have to be used.

```

&ATOMS
*O_MT_BLYP.psp KLEINMAN-BYLANDER
LMAX=P
  1
  10.0  10.0  10.0
*H_MT_BLYP.psp
LMAX=S
  2
  8.5   9.0   10.0
 11.5  9.0   10.0
&END

```

Finally the **&ATOMS** section is needed to specify the number of atom and atom coordinates and the pseudopotentials, that are used to represent them. In this case we use the Becke-Lee-Yang-Parr(BLYP) pseudopotential basis. The detailed syntax of the pseudopotential specification is a bit complicated. If you want to know more, please refer to the Further Details of the Input section of the CPMD manual.

2.2 Optimize Geometry by GDIIS

Optimize Geometry causes the program to optimize the geometry and the wavefunction of the system. The geometry optimization using direct inversion in the itera-

tive subspace(GDIIS) is not much else than repeated single point calculations, where the positions of the atoms are updated according to the forces acting on them. Any combination of methods for geometry optimization and wavefunction optimization is allowed. Possible options for geometry optimization are GDIIS, RFO, BFGS, and steepest descent. If you choose steepest descent for both, geometry variables and the wavefunction, a combined method is used. For all other combinations a full wavefunction optimization is performed between changes of the ionic coordinates. The default options are GDIIS and ODIIS. The quasi-Newton methods (GDIIS, RFO, and BFGS) are using the BFGS method to update an approximate Hessian. At the beginning of a run, the Hessian can either be initialized as a unit matrix HESSIAN UNIT or with an empirical force field by default.

Geometry Optimization and Wavefunction Optimization use the same method, DIIS, to iterate vectors. The main difference of them is that Geometry Optimization updates the positions of the atoms and uses BFGS method to update the Hessian matrix each time as wavefunction converge.

2.2.1 Geometry Optimization: Input File Format

The input file of the wavefunction optimization and geometry optimization are similar. The required changes in the input file are rather small (*1-h2o-geoopt.inp*):

```
&CPMD
OPTIMIZE GEOMETRY XYZ
HESSIAN UNITY
CONVERGENCE ORBITALS
  1.0d-7
CONVERGENCE GEOMETRY
  3.0d-4
STRUCTURE BONDS ANGLES
&END
```

We have replaced **WAVEFUNCTION** with **GEOMETRY** and added the sub-option **XYZ** to have CPMD write a 'trajectory' of the optimization in a file name *GEO_OPT.xyz* (so it can be visualized later). The keyword **HESSIAN UNIT** tells

the CPMD program to set the initial approximate Hessian for a geometry optimization constructed by simply a unit. Also, we specify the convergence parameter for the ions(**GEOMETRY**) with 3×10^{-4} . Default value is 5×10^{-4} .

2.3 Car-Parrinello Molecular Dynamics

The basic idea of the Car-Parrinello approach can be viewed to exploit the quantum mechanical adiabatic time-scale separation of fast electronic and slow nuclear motion by transforming that into classical-mechanical adiabatic energy scale separation in the framework of dynamical systems theory. In order to achieve this goal the two-component quantum/classical problem is mapped onto a two-component purely classical problem with two separate energy scales at the expense of losing the explicit time-dependence of the quantum subsystem dynamics. Furthermore, the central quantity, $\langle \Psi_0 | H | \Psi_0 \rangle$, evaluated with some wavefunction Ψ_0 , is certainly a function of the nuclear positions $\{R_I\}$. But at the same time it can be considered to be a functional of the wavefunction Ψ_0 and thus of a set of one-particle orbitals $\{\psi_i\}$ used to build up this wavefunction (being for instance a Slater determinant $\Psi_0 = \det\{\psi_i\}$ or a combination thereof). Now, in classical mechanics the force on the nuclei is obtained from the derivative of a Lagrangian with respect to the nuclear positions. This suggests that a functional derivative with respect to the orbitals, which are interpreted as classical fields, might yield the force on the orbitals, given a suitable Lagrangian. In addition, possible constraints within the set of orbitals have to be imposed, such as e.g. orthonormality (or generalized orthonormality conditions that include an overlap matrix).

Car and Parrinello postulated the following class of Lagrangians

$$L = \underbrace{\frac{1}{2} \sum_i \mu \langle \dot{\psi}_i | \dot{\psi}_i \rangle}_{\text{kinetic energy}} + \frac{1}{2} \sum_I M_I \dot{R}_I^2 - \underbrace{\langle \Psi_0 | H | \Psi_0 \rangle}_{\text{potential energy}} + \underbrace{\text{constraints}}_{\text{orthonormality}}, \quad (2.10)$$

to serve this purpose. The corresponding Newtonian equations of motion are obtained from the associated Euler-Lagrange equations

$$\frac{d}{dt} \left[\frac{\partial L}{\partial \dot{R}_I} \right] = \frac{\partial L}{\partial R_I} \quad (2.11)$$

$$\frac{d}{dt} \left[\frac{\partial L}{\partial \dot{\psi}_i^*} \right] = \frac{\partial L}{\partial \psi_i^*}, \quad (2.12)$$

like in classical mechanics, but here for both the nuclear positions and the orbitals and that the constraints are holonomic. Following this route of ideas, generic Car-Parrinello equations of motion are found to be of the form

$$M_I \ddot{R}_I = -\frac{\partial}{\partial R_I} \langle \Psi_0 | H | \Psi_0 \rangle + \frac{\partial}{\partial R_I} \{constraints\} \quad (2.13)$$

$$\mu \ddot{\psi}_i = -\frac{\partial}{\partial \psi_i^*} \langle \Psi_0 | H | \Psi_0 \rangle + \frac{\partial}{\partial \psi_i^*} \{constraints\}, \quad (2.14)$$

where μ are the fictitious masses or inertia parameters assigned to the orbital degrees of freedom; the units of the mass parameter μ are energy times a squared time for reasons of dimensionality. Note that the constraints within the total wavefunction lead to constraint forces in the equations of motion. Note also that these constraints

$$constraints = constraints(\{\psi_i\}, \{R_I\}), \quad (2.15)$$

might be a function of both the set of orbitals $\{\psi_i\}$ and the nuclear positions $\{R_I\}$. These dependencies have to be taken into account properly in deriving the Car-Parrinello equations following from Eq.(2.10) using Eq.(2.11)-(2.12). The corresponding CP total energy (Hamiltonian) as constant of motion is

$$E_{HAM} = \frac{1}{2} \sum_i \mu \langle \dot{\psi}_i | \dot{\psi}_i \rangle + \frac{1}{2} \sum_I M_I \dot{R}_I^2 + E[\{\psi_i\}, \{R_I\}]. \quad (2.16)$$

According to the Car-Parrinello equations of motion, the nuclei evolve in time at a certain (instantaneous) physical temperature $\propto \sum_I M_I \dot{R}_I^2$, whereas a "fictitious temperature" $\propto \sum_i \mu \langle \dot{\psi}_i | \dot{\psi}_i \rangle$ is associated to the electronic degrees of freedom. In this terminology, "low electronic temperature" or "cold electrons" means that the electronic subsystem is close to its instantaneous minimum energy $min_{\{\psi_i\}} \langle \Psi_0 | H | \Psi_0 \rangle$, i.e. close to the exact Born-Oppenheimer surface. Thus, a ground-state wavefunction optimized for the initial configuration of the nuclei will stay close to its ground state also during time evolution if it is kept at a sufficiently low temperature.

The remaining task is to separate in practice nuclear and electronic motion such that the fast electronic subsystem stays cold also for long times but still follows the slow nuclear motion adiabatically (or instantaneously). Simultaneously, the nuclei are

nevertheless kept at a much higher temperature. This can be achieved in nonlinear classical dynamics via decoupling of the two subsystems and (quasi-) adiabatic time evolution. This is possible if the power spectra stemming from both dynamics do not have substantial overlap in the frequency domain so that energy transfer from the "hot nuclei" to the "cold electrons" becomes practically impossible on the relevant time scales. This amounts in other words to imposing and maintaining a metastability condition in a complex dynamical system for sufficiently long times.

2.3.1 Molecular-dynamics Lagrangian

Car and Parrinello formulated their method in the language of molecular dynamics. Their essential step was to treat the electronic wavefunctions as dynamical variables. Here Eq.(2.10), (2.13), and (2.14) are specialized to the case of a plane wave basis within Kohn-Sham density functional theory. Specifically the functions ψ_i are replaced by the expansion coefficients $c_i(G)$ and the orthonormality constraint only depends on the wavefunctions, not the nuclear positions. The equations of motion for the Car-Parrinello method are derived from this specific extended Lagrangian

$$L = \mu \sum_i \sum_G |\dot{c}_i(G)|^2 + \frac{1}{2} \sum_I M_I \dot{R}_I^2 - E[\{G\}, \{R_I\}], \quad (2.17)$$

where μ is a electron mass, M_I are the masses of the nuclei, E is the Kohn-Sham energy functional, R_I is the position of ion I . The Kohn-Sham energy functional takes the place of the potential energy in a conventional Lagrangian formulation. The electronic wave functions are subject to the constraints of orthonormality,

$$\int \psi_i^*(r) \psi_j(r) d^3r = \delta_{ij}. \quad (2.18)$$

These constraints are incorporated in the molecular-dynamics Lagrangian by using the method of Lagrange multipliers. The molecular-dynamics becomes

$$L = \mu \sum_i \sum_G |\dot{c}_i(G)|^2 + \frac{1}{2} \sum_I M_I \dot{R}_I^2 - E[\{G\}, \{R_I\}] + \sum_{i,j} \Lambda_{ij} \left(\sum_G c_i^*(G) c_j(G) - \delta_{ij} \right). \quad (2.19)$$

The Lagrange multipliers Λ_{jj} ensure that wavefunctions remain normalized, while the Lagrange multipliers $\Lambda_{ij}(i \neq j)$ ensure that the wavefunctions remain orthogonal.

The Lagrange multiplier may be thought of as providing additional forces acting on the wavefunctions, which ensure that the wavefunctions remain orthonormal.

2.3.2 Molecular-dynamics equations of motion

The Lagrangean in eq.(2.18) generates a dynamics for the parameters $\{c_i\}$'s and $\{R_I\}$'s through the equations of motion:

$$\mu\ddot{c}_i(G) = -\frac{\partial E}{\partial c_i^*(G)} + \sum_j \Lambda_{ij}c_j(G) \quad (2.20)$$

$$M_I\ddot{R}_I = -\frac{\partial E}{\partial R_I}. \quad (2.21)$$

The forces needed in a CPMD calculation are the partial derivative of the Kohn-Sham energy functional with respect to the independent variables, i.e. the wavefunction ψ_i and the nuclear positions R_I . The forces are calculated as the action of the Kohn-Sham Hamiltonian on the wavefunction $-\partial E/\partial c_i^*(G)$. The forces with respect to the nuclear positions are $-\partial E/\partial R_I$. The Lagrange multipliers add forces $\Lambda_{ij}c_j(G)$ to the force $-\partial E/\partial c_i^*(G)$. These forces ensure that the electronic wave functions remain orthonormal as they propagate along their molecular-dynamics trajectories. Because of the expansion of the electronic wavefunction in plane waves, the orthonormality constraint does not depend on the nuclear positions. For basis sets that depend on the atomic positions (e.g. atomic orbital basis sets) or methods that introduce an atomic position dependent metric the integration methods have to be adapted.

2.3.3 Numerical Integration

In a computer experiment we will not be able to generate the true trajectory of a system with a given set of initial positions and velocities. For all potentials used in real applications only numerical integration techniques can be applied. These techniques are based on a discretization of time and a repeated calculation of the forces on the particles. However, what we are looking for is a method with special properties: long time energy conservation and short time reversibility. Long time

energy conservation ensures that we stay on (in fact close) to the constant energy hypersurface and the short time reversibility means that the discretize equations still exhibit the time reversible symmetry of the original differential equations. Using these methods, the numerical trajectory will immediately diverge from the true trajectory (the divergence is exponential), but as they stay on the correct hypersurface, they still sample the same microcanonical ensemble. On the other hand, a short time accurate method will manage to stay close to the true trajectory for a longer time and ultimately will also exponentially diverge but will not stay close to the correct energy hypersurface and therefore will not give the correct ensemble averages.

Our method of choice is the velocity Verlet algorithm. It has the advantage that it uses as basic variables positions and velocities at the same time instant t . The velocity Verlet algorithm looks like a Taylor expansion for the coordinates:

$$R(t + \delta t) = R(t) + V(t)\delta t + \frac{F(t)}{2M}(\delta t)^2. \quad (2.22)$$

This equation is combined with the update for the velocities

$$V(t + \delta t) = V(t) + \frac{F(t + \delta t) + F(t)}{2M}\delta t \quad (2.23)$$

where δt is the length of the time step, $R(t)$ is the value at the present time step, and $R(t + \delta t)$ is the value at the next time step.

The velocity Verlet algorithm can easily be cast into a symmetric update procedure that looks in pseudo code

$V(:) := V(:) + (\delta t/2M(:)) * F(:)$

$R(:) := R(:) + \delta t * V(:)$

Calculate new forces $F(:)$

$V(:) := V(:) + (\delta t/2M(:)) * F(:)$

To perform a computer experiment the initial values for positions and velocities have to be chosen together with an appropriate time step (discretization length) δt .

2.3.4 Integration of equations of motion

The integrator used in the CPMD code is based on the velocity Verlet/rattle algorithm. The velocity Verlet algorithm requires more operations and more storage than the Verlet algorithm. However, it is much easier to incorporate temperature control via velocity scaling into the velocity Verlet algorithm. In addition, velocity Verlet allows to change the time step trivially and is conceptually easier to handle. The equations of propagation may be integrated using the velocity Verlet algorithm that takes the following form:

$$\begin{aligned}
 c_i(t + \delta t) &= c_i(t) + \delta t \dot{c}_i(t) + \frac{\delta t^2}{2\mu} \left[f_i(t) + \sum_j \Lambda_{ij} c_j(t) \right] \\
 \dot{c}_i(t + \delta t) &= \dot{c}_i(t) + \frac{\delta t}{2\mu} \left[[f_i(t) + f_i(t + \delta t)] + \sum_j \Lambda_{ij} c_j(t) \right] \\
 R_I(t + \delta t) &= R_I(t) + \delta t \dot{R}_I(t) + \frac{\delta t^2}{2M_I} F_I(t) \\
 \dot{R}_I(t + \delta t) &= \dot{R}_I(t) + \frac{\delta t}{2M_I} [F_I(t) + F_I(t + \delta t)].
 \end{aligned}$$

It is defined by the following equations

$$\dot{\tilde{R}}_I(t + \delta t) = \dot{R}_I(t) + \frac{\delta t}{2M_I} F_I(t) \tag{2.24}$$

$$R_I(t + \delta t) = R_I(t) + \delta t \dot{\tilde{R}}_I(t + \delta t)$$

$$\dot{\tilde{c}}_i(t + \delta t) = \dot{c}_i(t) + \frac{\delta t}{2\mu} f_i(t)$$

$$\tilde{c}_i(t + \delta t) = c_i(t) + \delta t \dot{\tilde{c}}_i(t + \delta t)$$

$$c_i(t + \delta t) = \tilde{c}_i(t + \delta t) + \sum_j X_{ij} c_j(t)$$

calculate $F_I(t + \delta t) : -\partial E / \partial R_I$

calculate $f_i(t + \delta t) : -\partial E / \partial c_i^*(G)$

$$\dot{R}'_I(t + \delta t) = \dot{\tilde{R}}_I(t + \delta t) + \frac{\delta t}{2M_I} F_I(t + \delta t)$$

$$\dot{c}'_i(t + \delta t) = \dot{\tilde{c}}_i(t + \delta t) + \frac{\delta t}{2\mu} f_i(t + \delta t)$$

$$\dot{c}_i(t + \delta t) = \dot{c}'_i(t + \delta t) + \sum_j Y_{ij} c_j(t + \delta t),$$

where $R_I(t)$ and $c_i(t)$ are the atomic positions of particle I and the Kohn-Sham orbital i at time t respectively. Here, F_I are the forces on atom I , and f_i are the forces on Kohn-Sham orbital i . The matrices \mathbf{X} and \mathbf{Y} are directly related to the Lagrange multipliers by

$$X_{ij} = \frac{\delta t^2}{2\mu} \Lambda_{ij}^p \quad (2.25)$$

$$Y_{ij} = \frac{\delta t}{2\mu} \Lambda_{ij}^v. \quad (2.26)$$

Notice that in the rattle algorithm the Lagrange multipliers to enforce the orthonormality for the positions Λ_{ij}^p and velocities Λ_{ij}^v are treated as independent variables. Denoting with \mathbf{C} the matrix of wavefunction coefficients $c_i(G)$, the orthonormality constraint can be written as

$$C^\dagger(t + \delta t)C(t + \delta t) - I = 0 \quad (2.27)$$

$$[\tilde{C} + XC]^\dagger [\tilde{C} + XC] - I = 0 \quad (2.28)$$

$$\tilde{C}^\dagger \tilde{C} + X\tilde{C}^\dagger C + C^\dagger \tilde{C}X^\dagger + XX^\dagger - I = 0 \quad (2.29)$$

$$XX^\dagger + XB + B^\dagger X^\dagger = I - A, \quad (2.30)$$

where the new matrices $A_{ij} = \tilde{c}_i^\dagger(t + \delta t)\tilde{c}_j(t + \delta t)$ and $B_{ij} = c_i^\dagger(t)\tilde{c}_j(t + \delta t)$ have been introduced in eq.(2.29). The unit matrix is denoted by the symbol I . By noting that $A = I + O(\delta t^2)$ and $B = I + O(\delta t)$, Eq.(2.29) can be solved iteratively using

$$X^{(n+1)} = \frac{1}{2} \left[I - A + X^{(n)}(I - B) + (I - B)X^{(n)} - (X^{(n)})^2 \right], \quad (2.31)$$

and starting from the initial guess

$$X^{(0)} = \frac{1}{2}(I - A). \quad (2.32)$$

In Eq.(2.30) it has been made use of the fact that the matrices X and B are real and symmetric, which follows directly from their definitions. Eq.(2.30) can usually be iterated to a tolerance of 10^{-6} within a few iterations.

The rotation matrix Y is calculated from the orthogonality condition on the orbital velocities

$$\dot{c}_i^\dagger(t + \delta t)c_j(t + \delta t) + c_i^\dagger(t + \delta t)\dot{c}_j(t + \delta t) = 0. \quad (2.33)$$

Applying Eq.(2.32) to the trial states $\dot{C}' + YC$ yields a simple equation for Y

$$Y = \frac{1}{2}(Q + Q^\dagger), \quad (2.34)$$

where $Q_{ij} = c_i^\dagger(t + \delta t)c_j^\dagger(t + \delta t)$. The fact that Y can be obtained without iteration means that the velocity constraint condition Eq.(2.32) is satisfied exactly at each time step.

The procedure for performing a total-energy pseudopotential calculation using the molecular dynamics technique is shown in the flow diagram of Fig.3 The procedure requires an initial set of trial wavefunctions from which the Hartree potential and the exchange-correlation potential can be calculated. The Hamiltonian matrices are constructed, and from these the accelerations of the wavefunctions are calculated. The equations of the motion for the electronic states are integrated, and the wave functions are orthogonalized and normalized. The charge density generated by the new set of wavefunctions is then calculated. This charge density used to construct a new set of Hamiltonian matrices, and a further set of wavefunctions is obtained by integration of the equations of motion and orthonormalization of the resultant wavefunctions. These iterations are repeated until the wavefunctions are stationary. The wavefunction are then linear combinations of the Kohn-Sham eigenstates. The Kohn-Sham energy functional is minimized and its value gives the total energy of the system. The solution is identical to the solution that would be obtained by using

matrix diagonalization techniques with the same basis states.

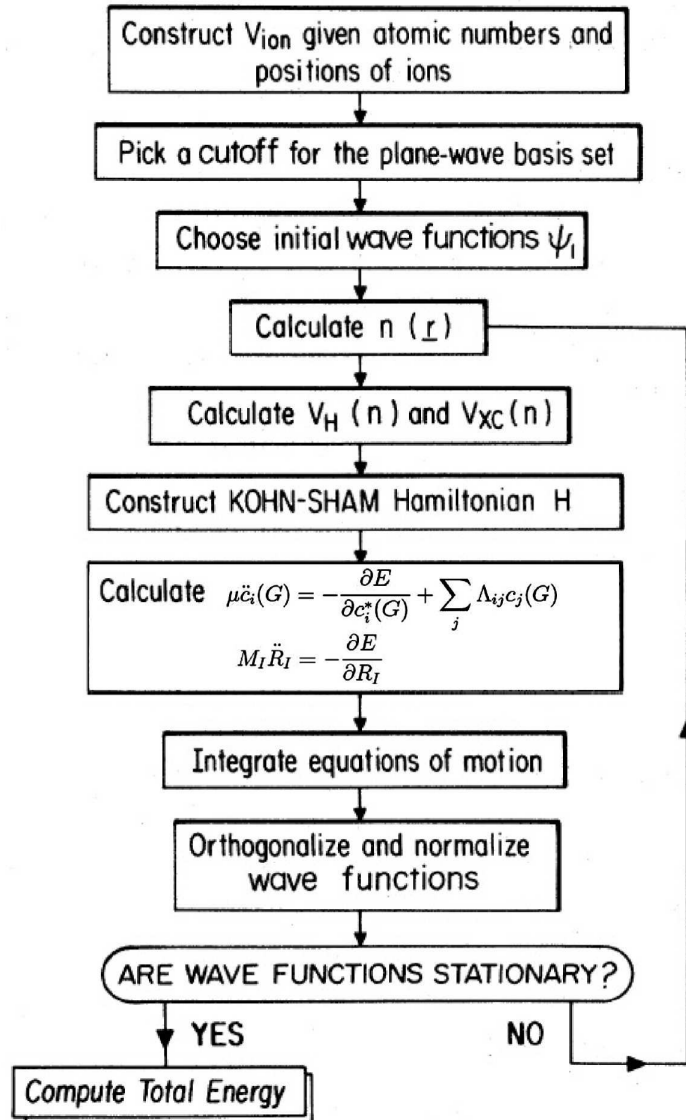


FIG.3. Flow chart describing the computational procedure for the calculation of the total energy of a solid with molecular dynamic.

2.3.5 Car-Parrinello Molecular Dynamics: Input File Format

For the CP-MD job you need a new input file, *1-h2o-cpmd.inp*, which should be copied into the same directory, where you started the wavefunction optimization run. If you compare it to the previous input files, you will find, that the only changes

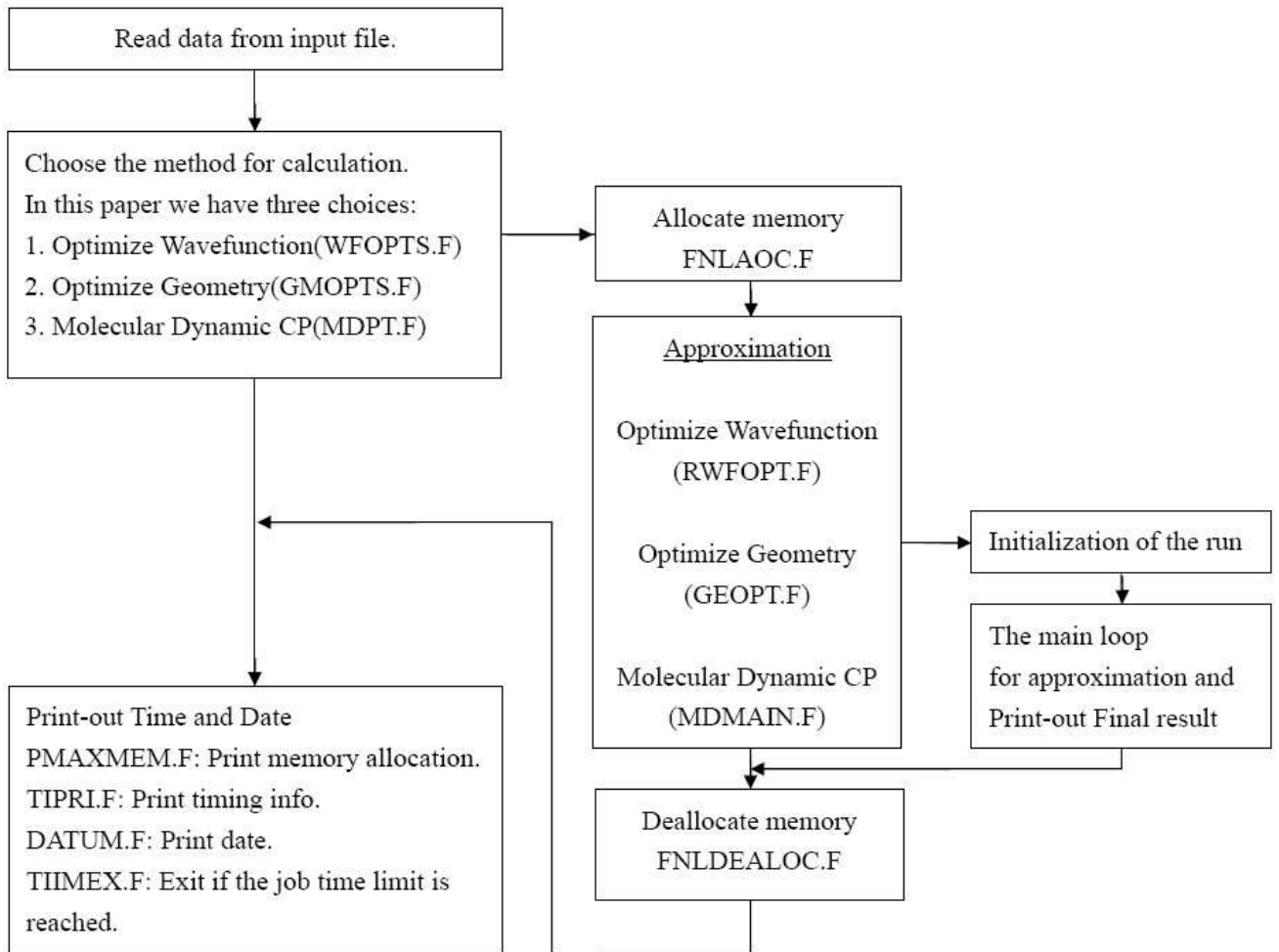
are again only in the **&CPMD** section of the input file.

```
&CPMD
MOLECULAR DYNAMICS CP
RESTART WAVEFUNCTION COORDINATES LATEST
MAXSTEP
  500
TIMESTEP
  2.0
STRUCTURE BONDS ANGLES
&END
```

The keyword **MOLECULAR DYNAMICS CP** defines the job type. **RESTART** [*OPTION*] this keyword controls what data is read(at the beginning) from the latest restart file (which is named RESTART.1 by default). A list of different *OPTIONS* can be specified (i.e. **WAVEFUNCTION**: Read old wavefunction from restart file, **COORDINATES**: Read old coordinates from restart file, **VELOCITIES**: Read old ionic,wavefunction and (cell)velocities from restart file, **LATEST**: Restart from the latest restart file as indicated in file LATEST, **ALL**: Restart with all fields of RESTART file, etc.). **MAXSTEP** limits the MD to 500 steps(default is 10000 steps) and the equations of motion will be solved for a time step of 2 atomic units (1 *a.u.* = 0.0241888428 *femtoseconds*).

3 Procedure

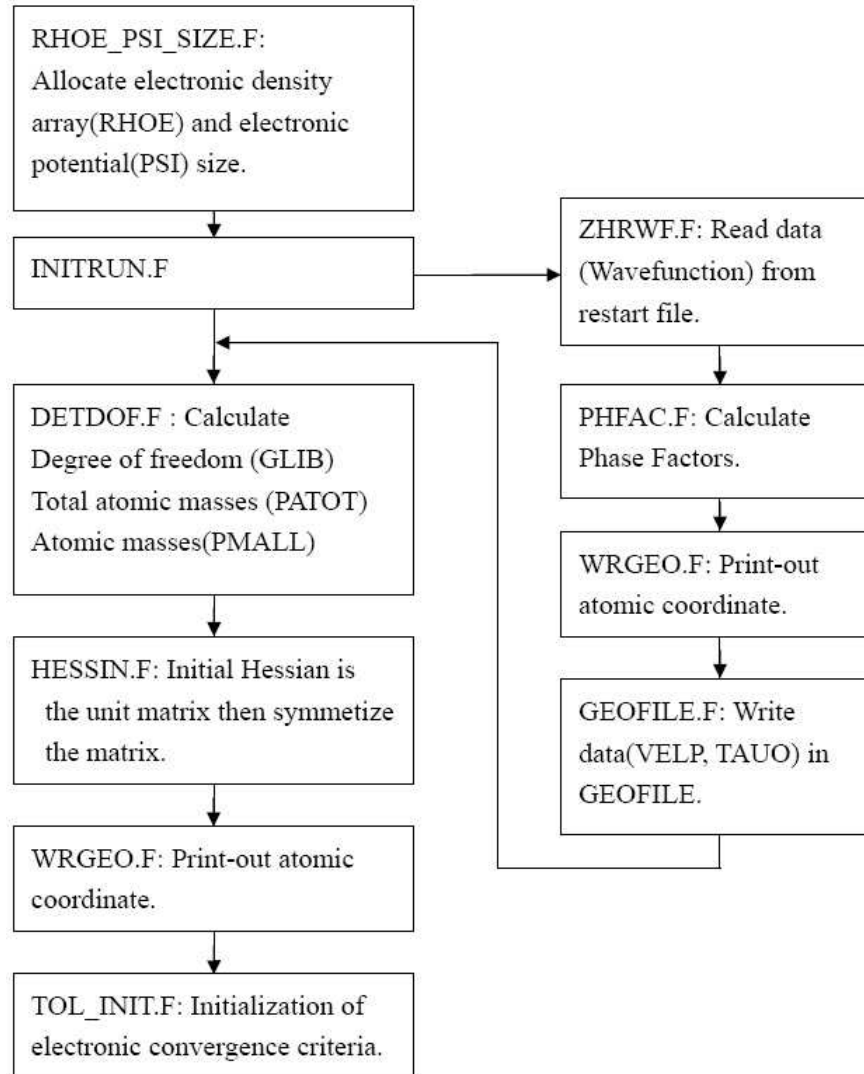
An outline of the three procedures for the CPMD program is given below:



The CPMD program starts with reading the input files, setting up the factors, which are applied to the computation, and the environment, e.g. the number of plane waves for wavefunction cutoff, super cell etc. First, decide which method, Optimize Wavefunction, Optimize Geometry, or Molecular Dynamic CP, will be used to compute the ground state energy. After that, allocate the memory for approximation, and give the initial guess for the electron structure. When the approximation completes, deallocate the memory and print out the total running time.

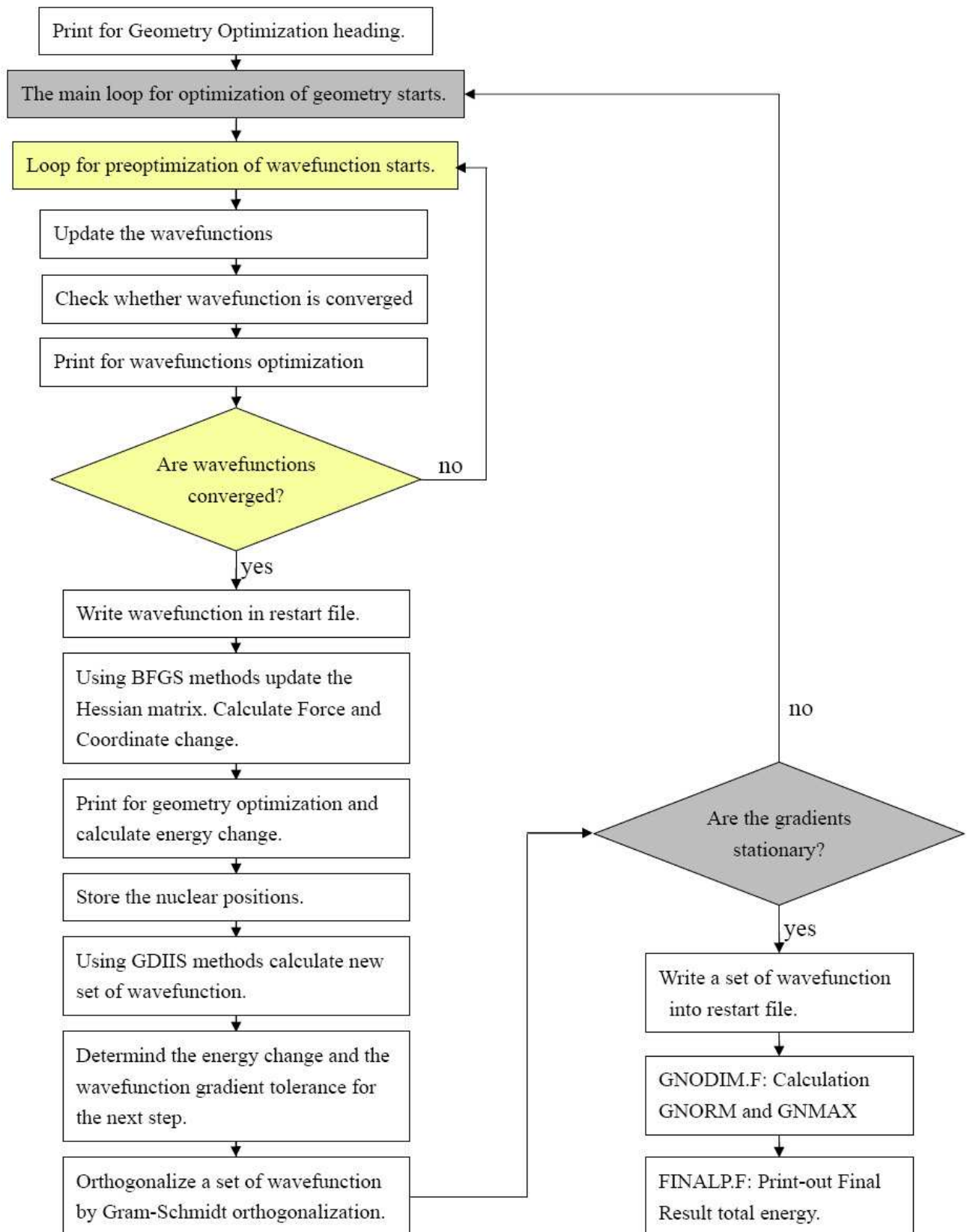
3.1 Optimize Geometry: Procedure

3.1.1 GEOPT.F: Initialization



Initialization starts with allocating electronic density array and electronic potential size. Second, check whether the data comes from the restart file. Then, find phase factors, atomic masses, and degree of freedom etc.. Subsequently, set either unit matrix as the initial Hessian matrix or an empirical force field by default, and then symmetrize the matrix. Finally, specify the electronic convergence criteria.

3.1.2 GEOPT.F: The main loop for approximation



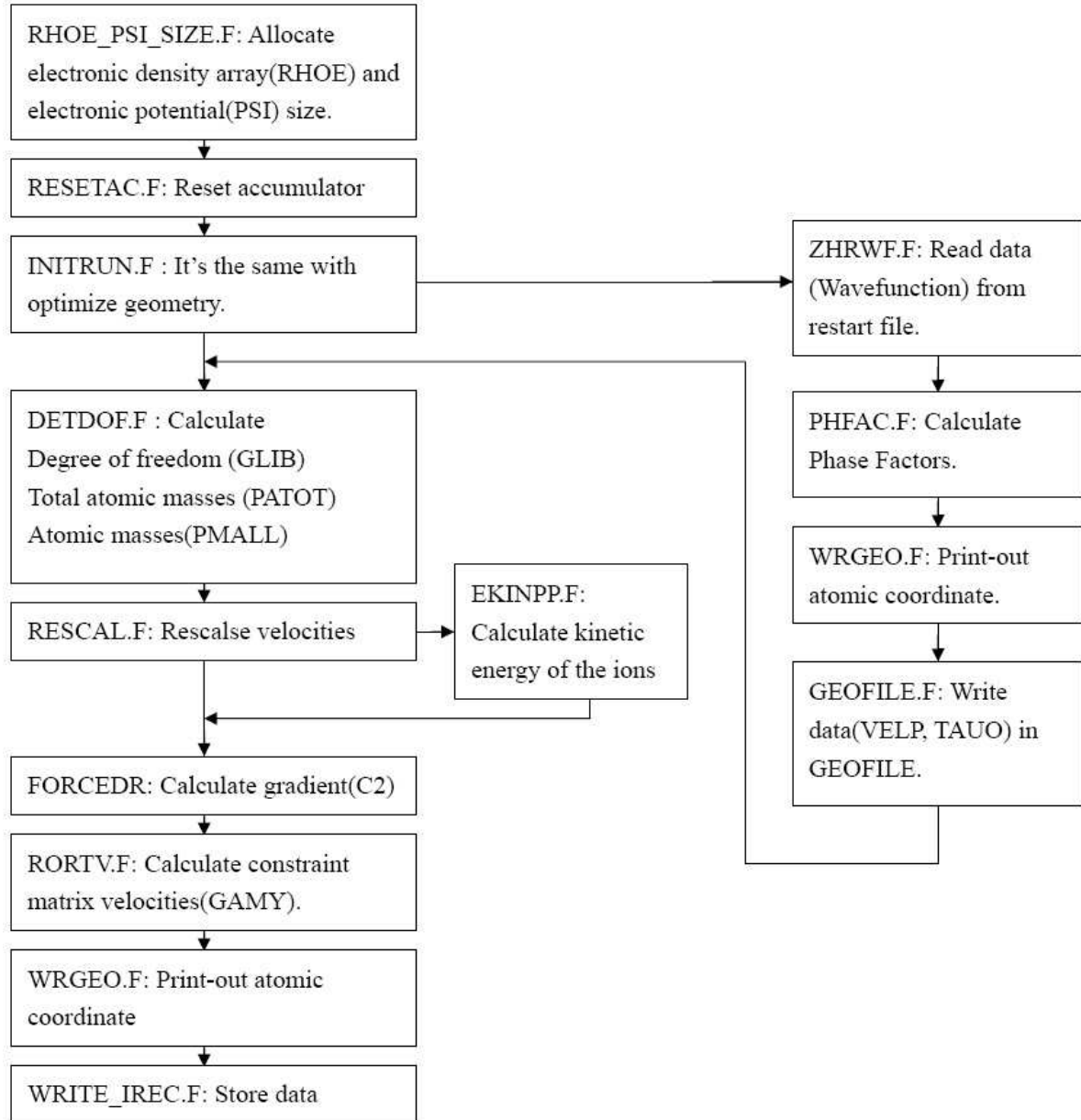
After the initialization, calculate a new set of wavefunctions by using ODIIS subroutine repeatedly until the wavefunctions are stationary. After that, start the geometry optimization by using BFGS method to update the Hessian matrix and calculate new forces and coordinate changes. A set of wavefunctions are calculated again by DIIS method and check the convergence of gradient and orthogonalize a set of wavefunctions by Gram-Schmidt method. These routines repeat until both the set of wavefunction and the gradient are stationary. The evolution of computation will be printed out in each iteration.

Print-out the final result: store data into the restart file and GEOMETRY file, and calculate the values, GNORM(Norm of Gradient) and GNMAX(maximum component). Then, print-out final result, the total energy and computational information, in the output file.



3.2 Car-Parrinello Molecular Dynamics: Procedure

3.2.1 MDMAIN.F: Initialization

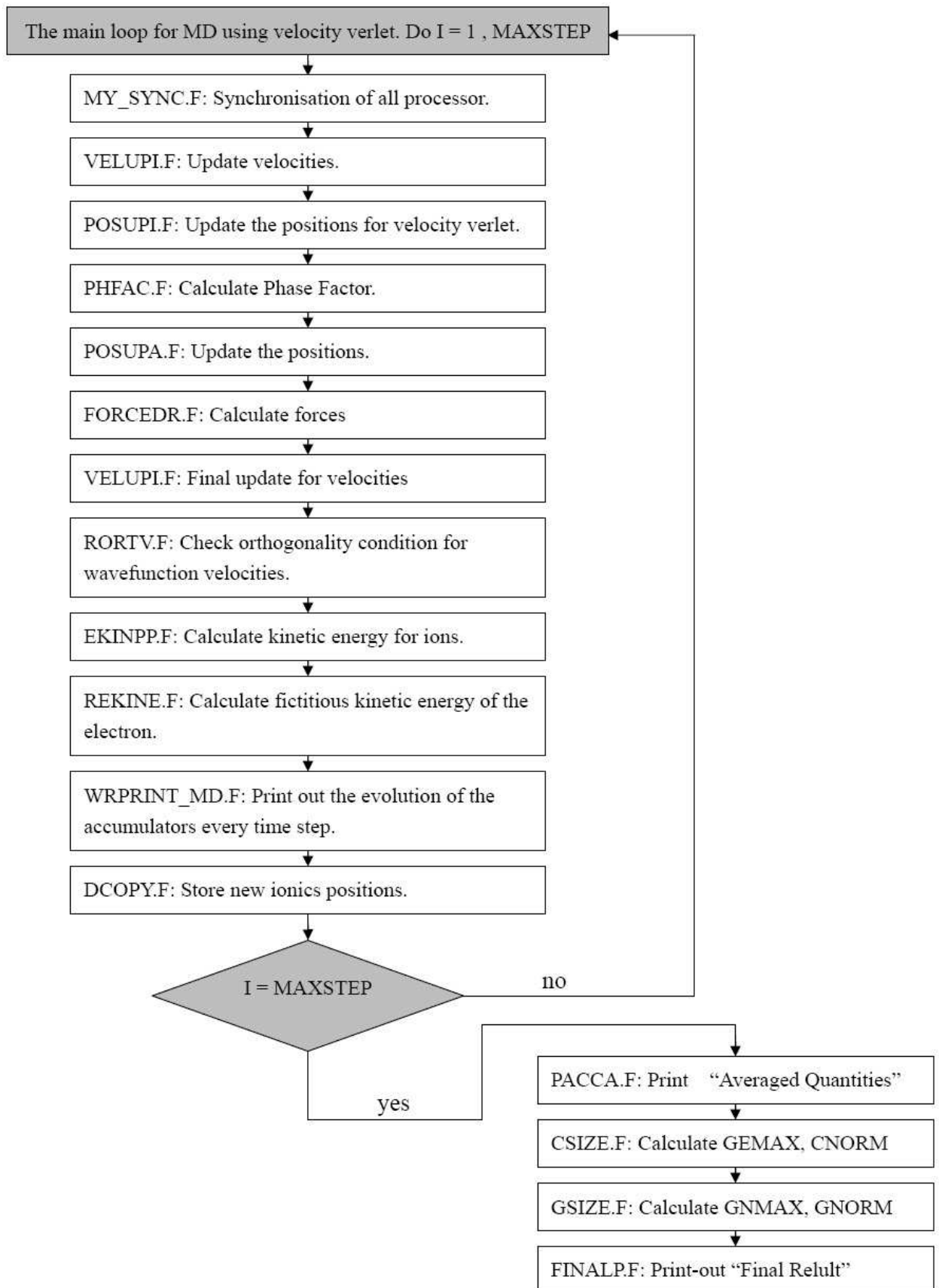


The program starts with allocating electronic density array and electronic potential size. After that, reset accumulator, save initial positions and reset the time counter, and then give the starting configuration i.e.read data from the restart file, calculate phase factor etc, which are the same as the previous method. Subsequently,

give the kinetic energy of the ions. Then, calculate the potential and the force on the ions as well as the electronic force and electronic density. Finally, calculate the constraint matrix velocities.



3.2.2 MDMAIN.F: The main loop for approximation



The procedure starts with synchronizing all processors. From the initialization, we have the initial guess for the electron structure. The structure of the algorithm for the wavefunctions is given below

1. Update velocities $VELP(:) = VELP(:) + (\delta t/2m) * FION(:)$
2. Update positions $TAUP(:) = TAUP(:) + \delta t * VELP(:)$
3. Calculate Lagrange multiplier L
4. Position constraints $TAUO(:) = TAUP(:) + L * TAUP(:)$
5. Calculate forces $FION(:) = H * TAUO(:)$
6. Final update for velocities $VELP(:) = VELP(:) + (\delta t/2m) * FION(:)$
7. Velocities constraints $VELP(:) = VELP(:) + L * TAUO(:)$

Now we have a new set of wavefunctions to calculate kinetic energy for ions, mean squared displacement of the atoms from the initial coordinates, and fictitious kinetic energy of the electrons. At each time step, print out the evolution of the accumulators and store the new ionic positions. Perform the procedure stated above repeatedly until the approximation reaches the maximum time step. Then, print out the averaged quantities values in the output file and calculate GEMAX(maximum component of electronic gradient), CNORM(norm of electronic gradient), GNMAX(maximum component) and GNORM(Norm of Gradient). Finally, print the final total energy and computational information in the output file.

4 Discussion and Output File Format

4.1 Optimize wavefunction: Output file format

With the input file *1-h2o-wave.inp*,

```
&INFO
  Isolate water molecule.
  Single point calculation.
&END
&CPMD
  OPTIMIZE WAVEFUNCTION
  CONVERGENCE ORBITALS
    1.0d-7
  STRUCTURE BONDS ANGLES
&END
&DFT
  FUNCTIONAL BLYP
  GC-CUTOFF
    1.0d-06
&END
&SYSTEM
  SYMMETRY
    1
  CELL
    20.0  1.0  1.0  0.0  0.0  0.0
  CUTOFF
    70.0
&END
&ATOMS
*O_MT_BLYP.psp KLEINMAN-BYLANDER
  LMAX=P
    1
    10.0   10.0   10.0
*H_MT_BLYP.psp
  LMAX=S
    2
    8.5    9.0    10.0
    11.5   9.0    10.0
&END
```

we get the output file *1-h2o-wave.out*. Let's have a closer look at the contents of this file.

```

PROGRAM CPMD STARTED AT: Sat Jun 23 17:01:30 2007

*****
*****
*****
*****
*****
*****
*****
*****
*****

VERSION 3.9.2

COPYRIGHT
IBM RESEARCH DIVISION
MPI FESTKOERPERFORSCHUNG STUTTGART

The CPMD consortium
WWW: http://www.cpmc.org
Mailinglist: cpmc-list@cpmc.org
E-mail: cpmc@cpmc.org

*** Jun 9 2007 -- 15:13:18 ***

THE INPUT FILE IS:                1-h2o-wave.inp
THIS JOB RUNS ON:                  yui.am.nctu.edu.tw
THE CURRENT DIRECTORY IS:         /root/CPMD/SOURCE
THE TEMPORARY DIRECTORY IS:      /root/CPMD/SOURCE
THE PROCESS ID IS:                3650
THE JOB WAS SUBMITTED BY:        root

```

We start with the header, where you can see, when the run was started, what version on CPMD you were using, and when it was compiled. Here we have some technical information about the environment, where this job was running.

```

*****
* INFO - INFO - INFO - INFO - INFO - INFO - INFO - INFO - INFO - INFO - INFO *
*****
* Isolate water molecule. *
* Single point calculation. *
*****

```

Here we see the contents of the **&INFO** section copied to the output.

```
SINGLE POINT DENSITY OPTIMIZATION

PATH TO THE RESTART FILES:                ./
GRAM-SCHMIDT ORTHOGONALIZATION
MAXIMUM NUMBER OF STEPS:                  10000 STEPS
PRINT INTERMEDIATE RESULTS EVERY         10001 STEPS
STORE INTERMEDIATE RESULTS EVERY         10001 STEPS
NUMBER OF DISTINCT RESTART FILES:         1
TEMPERATURE IS CALCULATED ASSUMING EXTENDED BULK BEHAVIOR
FICTITIOUS ELECTRON MASS:                 400.0000
TIME STEP FOR ELECTRONS:                  5.0000
TIME STEP FOR IONS:                      5.0000
CONVERGENCE CRITERIA FOR WAVEFUNCTION OPTIMIZATION: 1.0000E-07
WAVEFUNCTION OPTIMIZATION BY PRECONDITIONED DIIS
THRESHOLD FOR THE WF-HESSIAN IS          0.5000
MAXIMUM NUMBER OF VECTORS RETAINED FOR DIIS: 10
STEPS UNTIL DIIS RESET ON POOR PROGRESS: 10
FULL ELECTRONIC GRADIENT IS USED
SPLINE INTERPOLATION IN G-SPACE FOR PSEUDOPOTENTIAL FUNCTIONS
NUMBER OF SPLINE POINTS:                  5000
```

This section now gives you a summary of the parameters read in from the **&CPMD** section, or their respective default settings.



```

EXCHANGE CORRELATION FUNCTIONALS
LDA EXCHANGE:                SLATER (ALPHA = 0.66667)
LDA CORRELATION:             LEE, YANG & PARR
                             [C.L. LEE, W. YANG, AND R.G. PARR, PRB 37 785 (1988)]
GRADIENT CORRECTED FUNCTIONAL
DENSITY THRESHOLD:          1.00000E-06
EXCHANGE ENERGY
                             [A.D. BECKE, PHYS. REV. A 38, 3098 (1988)]
PARAMETER BETA:              0.004200
CORRELATION ENERGY
                             [LYP: C.L. LEE ET AL. PHYS. REV. B 37, 785 (1988)]

***   DETSP1 THE NEW SIZE OF THE PROGRAM IS   1636/ 42832 kBYTES ***

***** ATOMS *****
NR  TYPE    X(bohr)    Y(bohr)    Z(bohr)    MBL
  1   O     10.00000    10.00000    10.00000    3
  2   H      8.50000     9.00000     10.00000    3
  3   H     11.50000     9.00000     10.00000    3
*****

NUMBER OF STATES:              4
NUMBER OF ELECTRONS:           8.00000
CHARGE:                         0.00000
ELECTRON TEMPERATURE(KELVIN):  0.00000
OCCUPATION
2.0 2.0 2.0 2.0

=====
| Pseudopotential Report   Thu Nov 30 13:19:26 1995 |
-----
| Atomic Symbol           :  O           |
| Atomic Number           :  8           |
| Number of core states   :  1           |
| Number of valence states :  2           |
[... ]
| Number of Mesh Points   :  631         |
| Pseudoatom Total Energy : -15.775323   |
=====

```

This part of the output tells you which and how many atoms and electrons are used, what functional and what pseudopotentials were used, and what the values of some related parameters are.

```

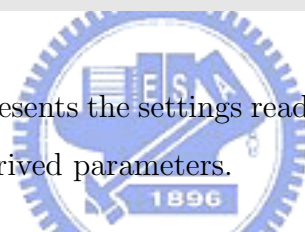
***** SUPERCELL *****
SYMMETRY:                               SIMPLE CUBIC
LATTICE CONSTANT(a.u.):                  20.00000
CELL DIMENSION: 20.0000 1.0000 1.0000 0.0000 0.0000 0.0000
VOLUME(OMEGA IN BOHR^3):                  8000.00000
LATTICE VECTOR A1(BOHR):                  20.0000 0.0000 0.0000
LATTICE VECTOR A2(BOHR):                  0.0000 20.0000 0.0000
LATTICE VECTOR A3(BOHR):                  0.0000 0.0000 20.0000
RECIP. LAT. VEC. B1(2Pi/BOHR):           0.0500 0.0000 0.0000
RECIP. LAT. VEC. B2(2Pi/BOHR):           0.0000 0.0500 0.0000
RECIP. LAT. VEC. B3(2Pi/BOHR):           0.0000 0.0000 0.0500
REAL SPACE MESH:                          108      108      108
WAVEFUNCTION CUTOFF(RYDBERG):              70.00000
DENSITY CUTOFF(RYDBERG):                  (DUAL= 4.00) 280.00000
NUMBER OF PLANE WAVES FOR WAVEFUNCTION CUTOFF: 39559
NUMBER OF PLANE WAVES FOR DENSITY CUTOFF:  316426
*****

*** RINFORCE! THE NEW SIZE OF THE PROGRAM IS 33968/ 74424 kBYTES ***
*** FFTPRP! THE NEW SIZE OF THE PROGRAM IS 73584/ 112216 kBYTES ***

GENERATE ATOMIC BASIS SET
O      SLATER ORBITALS
  2S    ALPHA= 2.2458      OCCUPATION= 2.00
  2P    ALPHA= 2.2266      OCCUPATION= 4.00
H      SLATER ORBITALS
  1S    ALPHA= 1.0000      OCCUPATION= 1.00

```

This part of the output presents the settings read in from the **&SYSTEM** section of the input file and some derived parameters.



```

(K+E1+L+N+X)      TOTAL ENERGY = -16.76192374 A.U.
(K)                KINETIC ENERGY = 13.19562960 A.U.
(E1=A-S+R)        ELECTROSTATIC ENERGY = -11.08250703 A.U.
(S)                ESELF = 12.63317221 A.U.
(R)                ESR = 0.88954969 A.U.
(L) LOCAL PSEUDOPOTENTIAL ENERGY = -16.49021572 A.U.
(N) N-L PSEUDOPOTENTIAL ENERGY = 2.02457900 A.U.
(X) EXCHANGE-CORRELATION ENERGY = -4.40940959 A.U.
GRADIENT CORRECTION ENERGY = -0.25322421 A.U.

```

After some output to report the setup of the initial guess for the electron structure, we now see a summary of the various energy contribution to the total energy of the system, based on the initial guess. Now the program is ready to start the wavefunction optimization.

Starting from the initial guess based on atomic wavefunctions the wavefunction for the total system is now calculated with an optimization procedure. You can follow

the progress of the optimization in the output file.

NFI	GEMAX	CNORM	ETOT	DETOT	TCPU
1	4.077E-02	4.719E-03	-16.761924	0.000E+00	3.05
2	2.362E-02	1.922E-03	-17.060918	-2.990E-01	3.05
3	1.189E-02	8.838E-04	-17.125082	-6.416E-02	3.05
4	5.113E-03	4.252E-04	-17.138541	-1.346E-02	3.10
5	3.353E-03	1.652E-04	-17.140948	-2.407E-03	3.10
[...]					
18	5.360E-07	2.894E-08	-17.141714	-7.373E-11	3.17
19	2.004E-07	1.353E-08	-17.141714	-1.238E-11	3.17
20	1.102E-07	6.425E-09	-17.141714	-3.077E-12	3.15
21	4.590E-08	3.263E-09	-17.141714	1.954E-13	3.14

The columns have the following meaning:

NFI : Step number (number of finite iterations)

GEMAX : largest off-diagonal component

CNORM : average of the off-diagonal components

ETOT : total energy

DETOT : change in total energy to the previous step

TCPU : (CPU) time for this step

and you can see that the calculation stops after the convergence criterion of $1.0d-7$ has been reached for the GEMAX value

```

*****
*
*              FINAL RESULTS
*
*
*****

*****
*              ATOMIC COORDINATES
*
*****
   1      O           10.000000    10.000000    10.000000
   2      H           8.500000     9.000000    10.000000
   3      H          11.500000     9.000000    10.000000
*****

ELECTRONIC GRADIENT:
  MAX. COMPONENT =    4.59033E-08      NORM =    3.26286E-09

TOTAL INTEGRATED ELECTRONIC DENSITY
  IN G-SPACE =                            8.000000
  IN R-SPACE =                            8.000000

(K+E1+L+N+X)      TOTAL ENERGY =      -17.14171423 A.U.
(K)                KINETIC ENERGY =     12.93978690 A.U.
(E1=A-S+R)        ELECTROSTATIC ENERGY = -11.34014526 A.U.
(S)                ESELF =                12.63317221 A.U.
(R)                ESR =                  0.88954969 A.U.
(L) LOCAL PSEUDOPOTENTIAL ENERGY =     -16.23946339 A.U.
(N) N-L PSEUDOPOTENTIAL ENERGY =        1.70818774 A.U.
(X) EXCHANGE-CORRELATION ENERGY =      -4.21008022 A.U.
    GRADIENT CORRECTION ENERGY =      -0.25011122 A.U.

```

Here we have the final summary of the results from our single point calculation.

```

          <<<<< ASSUMED BONDS >>>>>
      2 <--> 1   3 <--> 1
TOTAL NUMBER OF MOLECULAR STRUCTURES: 1
          BONDS
ATOM 1  ATOM 2  TYPE 1  TYPE 2  DISTANCE (BOHR, ANGSTROM)
    2     1     H     O        1.80278   0.95399
    3     1     H     O        1.80278   0.95399
          ANGLES
ATOM NUMBERS      ATOM TYPES      BOND ANGLES(DEGREES)
    2  1  3       H   O   H          112.6199
*****

```

Since we have requested the output print structure information at the end of run you can see them in title ASSUMED BONDS.

BIG MEMORY ALLOCATIONS			
SCR	7108197	PSI	2590058
YF	2590058	XF	2590058
PME	1582440	RHOE	1295029
GK	949278	RHOPS	632852
SCG	632852	VPS	632852

[PEAK NUMBER	78]	PEAK MEMORY	25048782 = 200.4 MBytes

```

*****
*                                     *
*                                     *
*                                     *
*                                     *
*                                     *
*****

```

TIMING			
SUBROUTINE	CALLS	CPU TIME	ELAPSED TIME
FWFFT	110	14.41	14.50
GCENER	22	14.14	14.19
INVFFT	89	11.47	11.60
S_INVFFT	88	6.40	6.52
XCENER	22	5.41	5.40
S_FWFFT	46	3.82	3.93
FFT-G/S	268	3.11	3.00
GRADEN	22	2.54	2.62
ODIIS	21	2.25	2.27
PHASE	199	1.96	2.04
VPSI	24	1.28	1.28
RHOFR	21	1.10	1.13
ATRHO	1	1.04	1.08
VOFRHOA	22	1.04	0.99
VOFRHOB	22	0.91	0.92
EICALC	22	0.68	0.69
NLIN	2	0.39	0.40
FORMFN	2	0.26	0.26
RGGEN	1	0.20	0.24
NUMPW	1	0.18	0.19

TOTAL TIME		72.59	73.25

```

CPU TIME : 0 HOURS 1 MINUTES 13.20 SECONDS
ELAPSED TIME : 0 HOURS 1 MINUTES 13.94 SECONDS

PROGRAM CPMD ENDED AT: Sat Jun 23 17:02:44 2007

```

In the final part of the output, we see some statistics regarding memory and CPU time usage. This is mainly of interest for CPMD developers, but it does not hurt to have an occasional look and see if the numbers are reasonable. Please note, that the

retrieval of this information is highly platform dependent, and that on some platforms the output may be bogus or very unreliable.

4.2 Optimize geometry: Output file format

With the input file *1-h2o-geoopt.inp*,

```
&CPMD
  OPTIMIZE GEOMETRY XYZ
  HESSIAN UNITY
  CONVERGENCE ORBITALS
    1.0d-7
  CONVERGENCE GEOMETRY
    3.0d-4
  STRUCTURE BONDS ANGLES
&END
&DFT
  FUNCTIONAL BLYP
  GC-CUTOFF
    1.0d-06
&END
&SYSTEM
  SYMMETRY
    1
  CELL
    20.0  1.0  1.0  0.0  0.0  0.0
  CUTOFF
    70.0
&END
&ATOMS
  *O_MT_BLYP.psp KLEINMAN-BYLANDER
    LMAX=P
    1
    10.0   10.0   10.0
  *H_MT_BLYP.psp
    LMAX=S
    2
    8.5    9.0    10.0
    11.5   9.0    10.0
&END
```

we get the output file *1-h2o-geopt.out*.

```
OPTIMIZATION OF IONIC POSITIONS

PATH TO THE RESTART FILES:                ./
GRAM-SCHMIDT ORTHOGONALIZATION
MAXIMUM NUMBER OF STEPS:                  10000 STEPS
PRINT INTERMEDIATE RESULTS EVERY         10001 STEPS
STORE INTERMEDIATE RESULTS EVERY         10001 STEPS
STORE INTERMEDIATE RESULTS EVERY         10001 SELF-CONSISTENT STEPS
NUMBER OF DISTINCT RESTART FILES:        1
TEMPERATURE IS CALCULATED ASSUMING EXTENDED BULK BEHAVIOR
FICTITIOUS ELECTRON MASS:                 400.0000
TIME STEP FOR ELECTRONS:                  5.0000
TIME STEP FOR IONS:                      5.0000
CONVERGENCE CRITERIA FOR WAVEFUNCTION OPTIMIZATION: 1.00000E-07
WAVEFUNCTION OPTIMIZATION BY PRECONDITIONED DIIS
THRESHOLD FOR THE WF-HESSIAN IS          0.5000
MAXIMUM NUMBER OF VECTORS RETAINED FOR DIIS: 10
STEPS UNTIL DIIS RESET ON POOR PROGRESS: 10
FULL ELECTRONIC GRADIENT IS USED
CONVERGENCE CRITERIA FOR GEOMETRY OPTIMIZATION: 3.000000E-04
GEOMETRY OPTIMIZATION BY GDIIS/BFGS
  SIZE OF GDIIS MATRIX:                   5
GEOMETRY OPTIMIZATION IS SAVED ON FILE GEO_OPT.xyz
INITIAL HESSIAN IS UNIT MATRIX
SPLINE INTERPOLATION IN G-SPACE FOR PSEUDOPOTENTIAL FUNCTIONS
  NUMBER OF SPLINE POINTS:                5000
```



As you can see from the first part of the output file (*1-h2o-geopt.out*), CPMD has recognized the job type, our convergence parameter and the request to write a *GEO_OPT.xyz* file.

```

=====
                                GEOMETRY OPTIMIZATION
=====
NFI      GEMAX      CNORM      ETOT      DETOT      TCPU
EWALDI  SUM IN REAL SPACE OVER      1* 1* 1 CELLS
 1  4.082E-02  4.702E-03  -16.768409  -1.677E+01  3.02
 2  2.360E-02  1.912E-03  -17.066259  -2.978E-01  3.04
 3  1.190E-02  8.789E-04  -17.130679  -6.442E-02  3.06
 4  5.077E-03  4.267E-04  -17.144081  -1.340E-02  3.07
 5  3.466E-03  1.677E-04  -17.146522  -2.441E-03  3.06
[...]
```

NFI	GEMAX	CNORM	ETOT	DETOT	TCPU
1	4.082E-02	4.702E-03	-16.768409	-1.677E+01	3.02
2	2.360E-02	1.912E-03	-17.066259	-2.978E-01	3.04
3	1.190E-02	8.789E-04	-17.130679	-6.442E-02	3.06
4	5.077E-03	4.267E-04	-17.144081	-1.340E-02	3.07
5	3.466E-03	1.677E-04	-17.146522	-2.441E-03	3.06
17	1.236E-06	5.355E-08	-17.147317	-2.701E-10	3.17
18	5.729E-07	2.745E-08	-17.147317	-6.422E-11	3.14
19	1.931E-07	1.291E-08	-17.147317	-1.323E-11	3.14
20	1.106E-07	6.200E-09	-17.147317	5.862E-13	3.22
21	3.672E-08	2.932E-09	-17.147317	-2.931E-12	3.22

```

RESTART INFORMATION WRITTEN ON FILE      ./RESTART.1

  ATOM      COORDINATES      GRADIENTS (-FORCES)
  1  O 10.0000 10.0000 10.0000  1.219E-13 -3.294E-02  8.541E-13
  2  H  8.5000  9.0000 10.0000  6.981E-03  1.771E-02  1.728E-15
  3  H 11.5000  9.0000 10.0000 -6.981E-03  1.771E-02  2.505E-15
FILE ./GEO_OPT.xyz EXISTS, NEW DATA WILL BE APPENDED
*****
*** TOTAL STEP NR.    21      GEOMETRY STEP NR.    1 ***
*** GNMAX=  3.293683E-02      ETOT=  -17.147317 ***
*** GNORM=  1.417885E-02      DETOT=  0.000E+00 ***
*** CNSTR=  0.000000E+00      TCPU=    65.82 ***
*****
[...]
```

TOTAL STEP NR.	GEOMETRY STEP NR.	GNMAX	GNORM	CNSTR	ETOT	DETOT	TCPU
21	1	3.293683E-02	1.417885E-02	0.000000E+00	-17.147317	0.000E+00	65.82
130	8	2.112504E-04 [1.34E-03]	1.153189E-04	0.000000E+00	-17.149541	-7.227E-07	37.40

```

*****
*** TOTAL STEP NR.    130      GEOMETRY STEP NR.    8 ***
*** GNMAX=  2.112504E-04 [1.34E-03]  ETOT=  -17.149541 ***
*** GNORM=  1.153189E-04      DETOT=  -7.227E-07 ***
*** CNSTR=  0.000000E+00      TCPU=    37.40 ***
*****
=====
                                END OF GEOMETRY OPTIMIZATION
=====
```

In the following output you can see, that an almost identical wavefunction optimization takes place. After printing the positions and forces of the atoms, however, you see a small report block and then another wavefunction optimization starts. The numbers for GNMAX, GNORM, and CNSTR stand for the largest absolute component of the force on any atom, average force on the atoms, and the largest absolute component of a constraint force on the atoms respectively.

```

*****
*
*                               *
*                               *
*                               *
*                               *
*****

ATOM          COORDINATES          GRADIENTS (-FORCES)
1  O 10.0000 10.0950 10.0000 -7.705E-11 2.839E-05 -2.877E-09
2  H  8.5477  8.9580 10.0000 -1.217E-04 -2.113E-04 2.569E-10
3  H 11.4523  8.9580 10.0000  1.217E-04 -2.112E-04 -1.342E-09
*****

ELECTRONIC GRADIENT:
  MAX. COMPONENT = 6.08859E-08      NORM = 5.42925E-09
NUCLEAR GRADIENT:
  MAX. COMPONENT = 2.11250E-04      NORM = 1.15319E-04

TOTAL INTEGRATED ELECTRONIC DENSITY
  IN G-SPACE = 8.000000
  IN R-SPACE = 8.000000

(K+E1+L+N+X)      TOTAL ENERGY = -17.14954112 A.U.
(K)                KINETIC ENERGY = 12.82168498 A.U.
(E1=A-S+R)        ELECTROSTATIC ENERGY = -11.42593635 A.U.
(S)                ESELF = 12.63317221 A.U.
(R)                ESR = 0.81399784 A.U.
(L) LOCAL PSEUDOPOTENTIAL ENERGY = -16.08651050 A.U.
(N) N-L PSEUDOPOTENTIAL ENERGY = 1.73103901 A.U.
(X) EXCHANGE-CORRELATION ENERGY = -4.18981825 A.U.
      GRADIENT CORRECTION ENERGY = -0.24918638 A.U.

```

At the end of the geometry optimization, you can see that the forces and the total energy have significantly decreased from their start values as it is to be expected.

4.3 Car-Parrinello Molecular Dynamics : Output file format

With the input file *1-h2o-cpmd.inp*,


```
&CPMD
  MOLECULAR DYNAMICS CP
  RESTART WAVEFUNCTION COORDINATES LATEST
  MAXSTEP
    500
  TIMESTEP
    2.0
  STRUCTURE BONDS ANGLES
&END
&DFT
  FUNCTIONAL BLYP
  GC-CUTOFF
    1.0d-06
&END
&SYSTEM
  SYMMETRY
    1
  CELL
    20.0  1.0  1.0  0.0  0.0  0.0
  CUTOFF
    70.0
&END
&ATOMS
*O_MT_BLYP.psp KLEINMAN-BYLANDER
  LMAX=P
    1
    10.0   10.0   10.0
*H_MT_BLYP.psp
  LMAX=S
    2
    8.5    9.0    10.0
    11.5   9.0    10.0
&END
```

we get the output file *1-h2o-cpmd.out*.

```
CAR-PARRINELLO MOLECULAR DYNAMICS

PATH TO THE RESTART FILES:                ./
RESTART WITH OLD ORBITALS
RESTART WITH OLD ION POSITIONS
RESTART WITH LATEST RESTART FILE
ITERATIVE ORTHOGONALIZATION
  MAXIT:                                   30
  EPS:                                     1.00E-06
MAXIMUM NUMBER OF STEPS:                   500 STEPS
PRINT INTERMEDIATE RESULTS EVERY          10001 STEPS
STORE INTERMEDIATE RESULTS EVERY          10001 STEPS
STORE INTERMEDIATE RESULTS EVERY          501 SELF-CONSISTENT STEPS
NUMBER OF DISTINCT RESTART FILES:         1
TEMPERATURE IS CALCULATED ASSUMING EXTENDED BULK BEHAVIOR
```

The header is unchanged up to the point where the settings from the `&CPMD` section are printed. As you can see, the program has recognized the `RESTART` and the `MAXSTEP` keywords.



```
TIME STEP FOR ELECTRONS:                   2.0000
TIME STEP FOR IONS:                       2.0000
TRAJECTORIES ARE SAVED ON FILE
ELECTRON DYNAMICS: THE TEMPERATURE IS NOT CONTROLLED
ION DYNAMICS:      THE TEMPERATURE IS NOT CONTROLLED
SPLINE INTERPOLATION IN G-SPACE FOR PSEUDOPOTENTIAL FUNCTIONS
  NUMBER OF SPLINE POINTS:                 5000
```

This part of the output tells us, that the `TIMESTEP 2.0` keyword was recognized (the default is 5.0 a.u., cf. the wavefunction output file), and that there will be no temperature control, i.e. we will do a microcanonical (NVE-ensemble) simulation.

```
RV30I WARNING! NO WAVEFUNCTION VELOCITIES

RESTART INFORMATION READ ON FILE          ./RESTART.1
```

Here we get notified, that the program has read the requested data from the restart file. The warning about the missing wavefunction velocities is to be expected, since they will only be available when the restart was written by a previous Car-Parrinello

MD run.

NFI	EKINC	TEMPP	EKS	ECLASSIC	EHAM	DIS	TCPU
1	0.00000	0.1	-17.14172	-17.14171	-17.14171	0.456E-09	3.18
FILE ./TRAJECTORY EXISTS, NEW DATA WILL BE APPENDED							
2	0.00000	0.6	-17.14172	-17.14171	-17.14171	0.727E-08	3.13
3	0.00000	1.3	-17.14173	-17.14171	-17.14171	0.367E-07	3.12
4	0.00000	2.2	-17.14174	-17.14171	-17.14171	0.115E-06	3.13
5	0.00000	3.4	-17.14175	-17.14171	-17.14171	0.280E-06	3.11
6	0.00000	4.9	-17.14176	-17.14171	-17.14171	0.578E-06	3.14
7	0.00000	6.6	-17.14178	-17.14172	-17.14171	0.106E-05	3.13
8	0.00000	8.5	-17.14180	-17.14172	-17.14171	0.180E-05	3.14
9	0.00000	10.6	-17.14182	-17.14172	-17.14171	0.286E-05	3.12
10	0.00000	12.9	-17.14184	-17.14172	-17.14171	0.432E-05	3.11
11	0.00000	15.4	-17.14186	-17.14172	-17.14171	0.628E-05	3.14
12	0.00001	18.0	-17.14189	-17.14172	-17.14171	0.881E-05	3.17
[...]							
491	0.00005	85.3	-17.14257	-17.14176	-17.14171	0.474E-02	3.13
492	0.00005	85.3	-17.14257	-17.14176	-17.14171	0.487E-02	3.14
493	0.00005	85.4	-17.14257	-17.14176	-17.14171	0.499E-02	3.16
494	0.00004	85.4	-17.14257	-17.14176	-17.14171	0.511E-02	3.13
495	0.00004	85.6	-17.14257	-17.14176	-17.14171	0.524E-02	3.13
496	0.00004	85.8	-17.14257	-17.14176	-17.14171	0.536E-02	3.14
497	0.00004	86.1	-17.14257	-17.14176	-17.14171	0.548E-02	3.13
498	0.00004	86.5	-17.14258	-17.14176	-17.14171	0.560E-02	3.13
499	0.00004	86.9	-17.14258	-17.14176	-17.14171	0.573E-02	3.13
500	0.00004	87.5	-17.14259	-17.14175	-17.14171	0.585E-02	3.15

After some more output, we already discussed for the wavefunction optimization, this is now part of the energy summary for a Car-Parrinello-MD run. The individual columns have the following meaning:

- NFI : Step number (number of finite iterations)
- EKINC : (fictitious) kinetic energy of the electronic (sub-)system (K_{elc})
- TEMPP : Temperature (= kinetic energy (K_{ions}) / degrees of freedom) for atoms(ions)
- EKS : Kohn-Sham Energy (E_{KS}), equivalent to the potential energy in classical MD
- ECLASSIC : Equivalent to the total energy in a classical MD
($ECLASSIC = K_{ions} + E_{KS}$)
- EHAM : total energy, should be conserved ($E_{HAM} = K_{elc} + K_{ions} + E_{KS}$)
- DIS : mean squared displacement of the atoms from the initial coordinates.
- TCPU : (CPU) time needed for this step.


```

*****
*                AVERAGED QUANTITIES                *
*****
MEAN VALUE      +/- RMS DEVIATION
                   <x>      [ <x^2> - <x>^2 ] ** (1/2)
ELECTRON KINETIC ENERGY  0.648097E-04      0.320661E-04
IONIC TEMPERATURE          126.01              67.44
DENSITY FUNCTIONAL ENERGY -17.142976          0.670291E-03
CLASSICAL ENERGY          -17.141779          0.321361E-04
CONSERVED ENERGY         -17.141714          0.337175E-06
NOSE ENERGY ELECTRONS    0.000000          0.00000
NOSE ENERGY IONS        0.000000          0.00000
CONSTRAINTS ENERGY      0.000000          0.00000
ION DISPLACEMENT         0.171616E-01          0.163121E-01
CPU TIME                  3.1304

```

Finally we get a summary of some averages and root mean squared deviations for some of the monitored quantities. This is quite useful to detect unwanted energy drifts or too large fluctuations in the simulation.

4.4 Computational tests and results

Comparison of the results taken by Optimize Wavefunction, Optimize Geometry and Molecular Dynamic CP algorithms. All energies are in Hartree atomic units (a.u.).

Isolate hydrogen molecule (H ₂)	Number of Iterations	Total Energy (-E)	Convergence Criterion	CPU Time (Seconds)	CPU Memory (MBytes)	Atom	Atomic Coordinates (BONDS)		
							X	Y	Z
Optimize Wavefunction	10	1.13245953	1.87E-08	7.08	68.1	H	8.25999	7.558904	7.558904
						H	6.85782	7.558904	7.558904
Optimize Geometry	26	1.13245953	4.61E-08	15.83	67.8	H	8.28560	7.558904	7.558904
						H	6.83220	7.558904	7.558904
Optimize Geometry with RESTART	18	1.132896	4.45E-08	9.55	67.6	H	8.28560	7.558904	7.558904
						H	6.83220	7.558904	7.558904
Molecular Dynamic CP with Times Step 2 a.u.	300	1.1328668	5.15E-04	157.75	66.0	H	8.29188	7.558904	7.558904
						H	6.82593	7.558904	7.558904
Molecular Dynamic CP with Times Step 2 a.u.	400	1.1326409	7.80E-04	157.63	66.0	H	8.30519	7.558904	7.558904
						H	6.81261	7.558904	7.558904

Table 1: Isolated hydrogen(H₂) molecule

Isolate water molecule (H ₂ O)	Number of Iterations	Total Energy (-E)	Convergence Criterion	CPU Time (Seconds)	CPU Memory (MBytes)	Atom	Atomic Coordinates (BONDS)		
							X	Y	Z
Optimize Wavefunction	21	17.1417142	4.59E-08	73.25	200.4	O	10.00	10.00	10.00
						H	8.50	9.00	10.00
						H	11.50	9.00	10.00
Optimize Geometry	130	17.1495411	6.09E-08	415.39	197.9	O	10.0000	10.0950	10.00
						H	8.5477	8.9580	10.00
						H	11.4523	8.9580	10.00
Optimize Geometry with RESTART	100	17.1444542	6.13E-08	311.94	192.8	O	10.0000	10.1001	10.00
						H	8.5394	8.9535	10.00
						H	11.4606	8.9535	10.00
Molecular Dynamic CP with Times Step 2 a.u.	300	17.142586	4.66E-04	1562.36	179.2	O	10.0000	9.9923	10.00
						H	8.4455	8.9241	10.00
						H	11.5545	8.9241	10.00

Table 2: Isolated water(H_2O) molecule

Isolate water molecule (H ₂ O)	Number of Iterations	Total Energy (-E)	Convergence Criterion	CPU Time (Seconds)	CPU Memory (MBytes)	Atom	Atomic Coordinates (BONDS)		
							X	Y	Z
Optimize Wavefunction	23	16.2078185	9.91E-08	81.34	200.4	O	10.00	10.00	10.00
						H	9.50	9.00	10.00
						H	10.50	9.00	10.00
Optimize Geometry	340	17.1444535	5.74E-08	1071.45	197.9	O	10.0000	10.0985	10.00
						H	8.5403	8.9508	10.00
						H	11.4597	8.9508	10.00
Optimize Geometry with RESTART	319	17.1444535	5.21E-08	997	192.8	O	10.0000	10.0985	10.00
						H	8.5403	8.9508	10.00
						H	11.4597	8.9508	10.00
Molecular Dynamic CP with Times Step 2 a.u.	500	16.59683	2.90E-02	1542.86	179.2	O	10.0000	11.1299	10.00
						H	2.6120	-0.0023	10.00
						H	17.3880	-0.0023	10.00

Table 3: Isolated water(H_2O) molecule when the position of the atom is not set properly at the beginning

32 water molecules (H ₆₄ O ₃₂)	Number of Iterations	Total Energy (-E)	Convergence Criterion	CPU Time (Seconds)	CPU Memory (MBytes)
Optimize Wavefunction	19	551.05080	6.56E-06	1103.89	1409.4
Optimize Geometry	1676	551.25513	6.06E-06	91131.41	1555.3
Optimize Geometry with RESTART	1571	551.25444	6.85E-06	84432.37	1274.3
Molecular Dynamic CP with Times Step 2 a.u.	2000	551.15608	5.62E-04	124388.7	686.8

Table 4: 32 water molecule(H₆₄O₃₂)

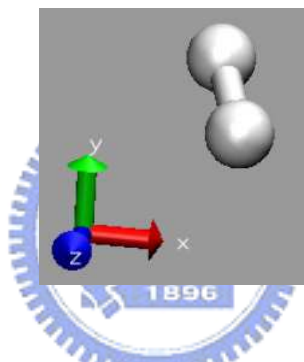


FIG.4. Isolated hydrogen(H₂) molecule

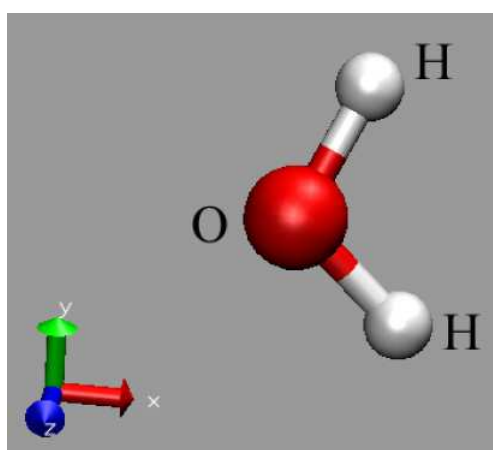


FIG.5. Isolated water(H₂O) molecule

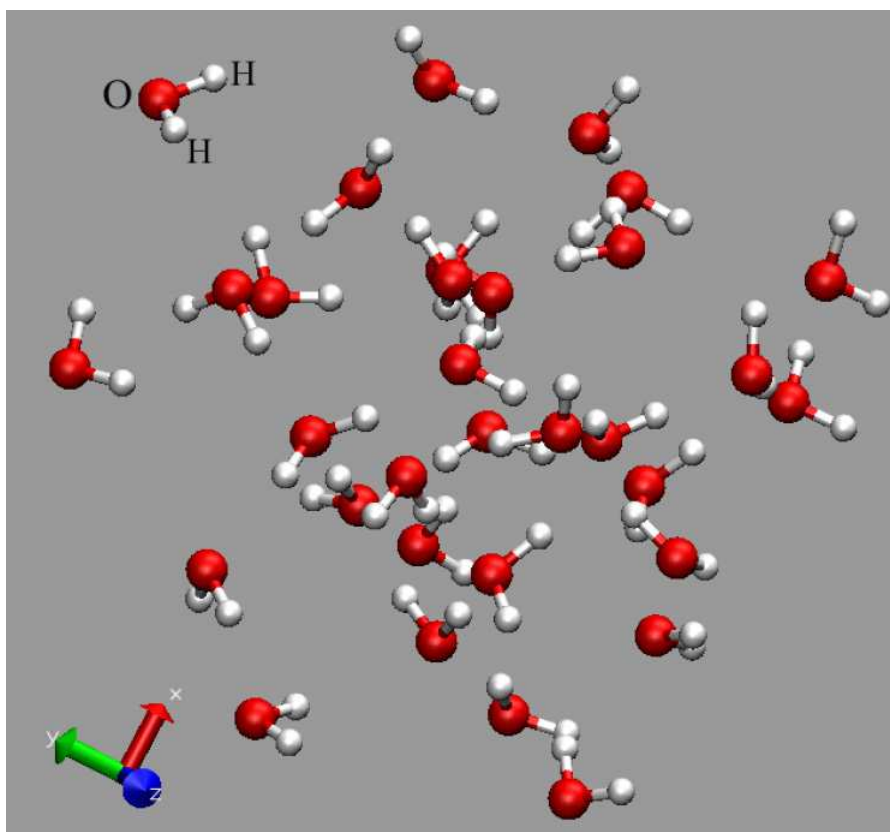


FIG.6. 32 water molecule($H_{64}O_{32}$)

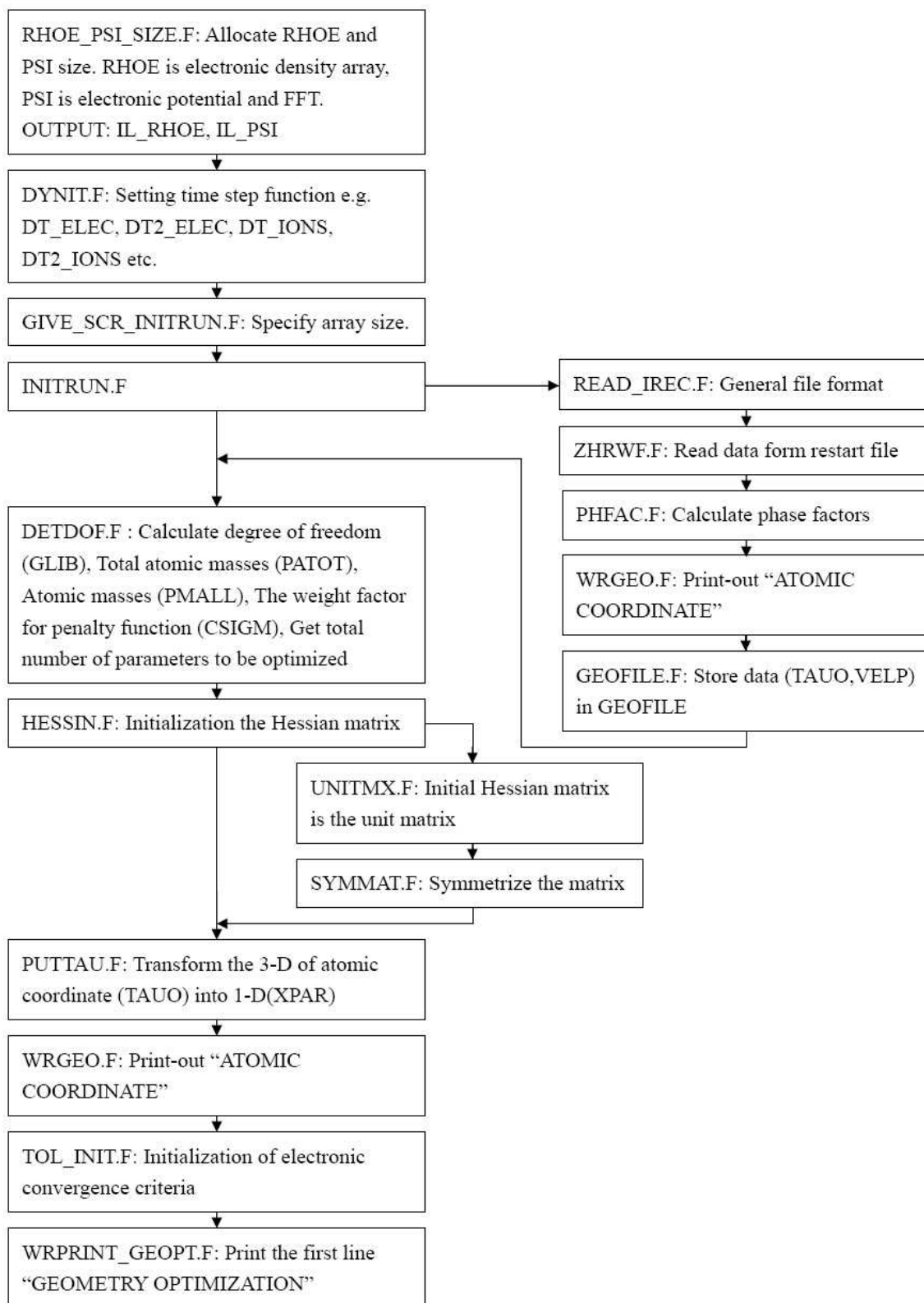
Wavefunction optimization gives the fastest result for single molecules and small(bulk) systems calculation, in comparison to calculations for large molecules (Table 4). Nonetheless, the ground-state energy calculated is not the best lowest figure because the method does not measure and change atomic position. Unlike the first method, Geometry optimization and Molecular dynamic CP measure and change atomic position in all iterations until the best optimal position is found (Table 3). As a result, the ground-state energy calculated by Wavefunction optimization will not be the optimal value if the position of the atom in an input file is not set properly at the beginning (Table 3). However, the method is suitable for finding the initial set of wavefunction which then will be applied to Geometry optimization and Molecular dynamic CP methods. The two steps process of calculation is more efficient and faster than any single method alone. For example, Geometry optimization that read the wavefunction value from a restart file required fewer iterations than the one that did not read the value.

Molecular dynamic CP is suitable for large molecules calculation since it requires much smaller CPU memory than Geometry optimization. Molecular dynamic CP takes up ≈ 600 Mbytes CPU memory while Geometry optimization takes up twice the CPU memory, ≈ 1200 , for the same water 32 molecules($H_{64}O_{32}$) ground energy calculation. In addition, Molecular dynamic CP will not check convergence value but will calculate accordingly to steps set by the user in an input file. The method focuses more on changes in energy values in the system through time.

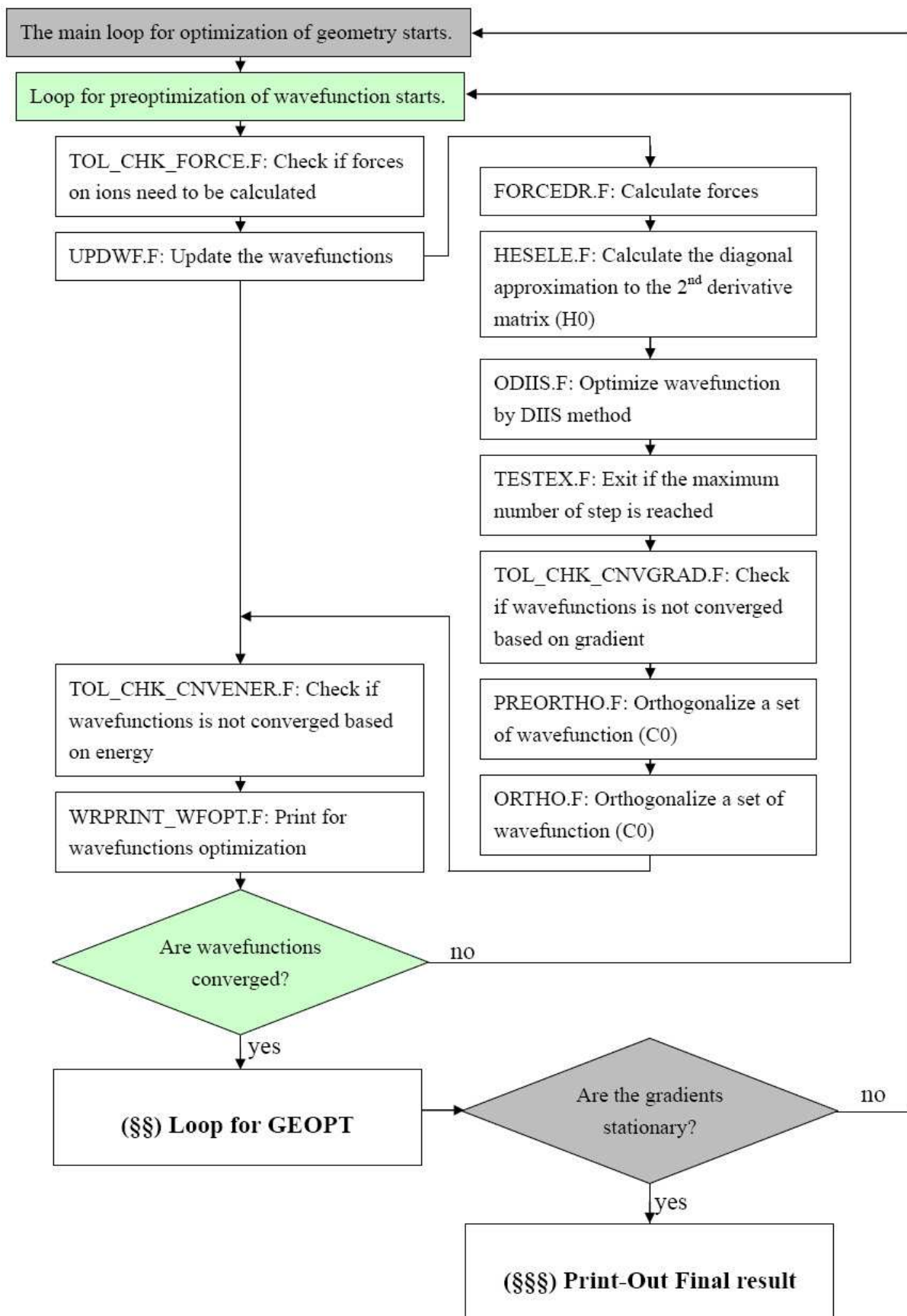


A Appendix

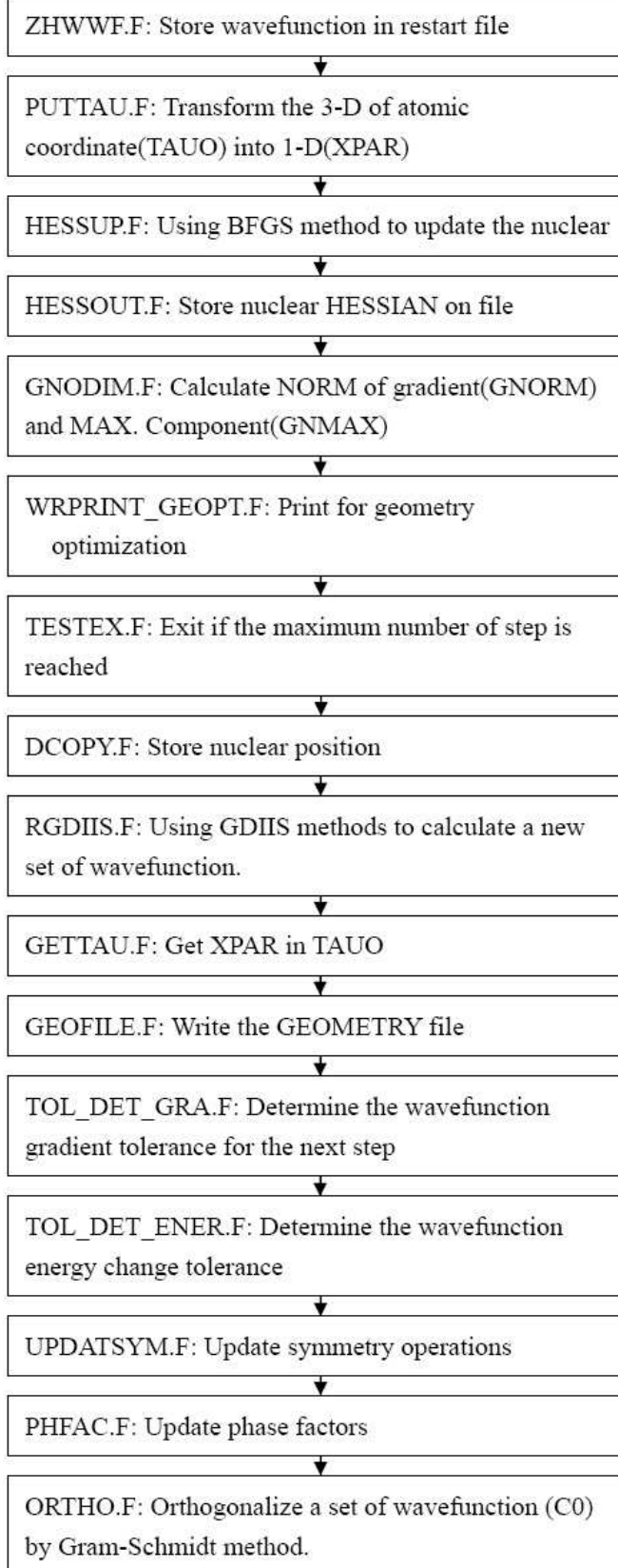
A.1 Optimize Geometry Procedure: Initialization



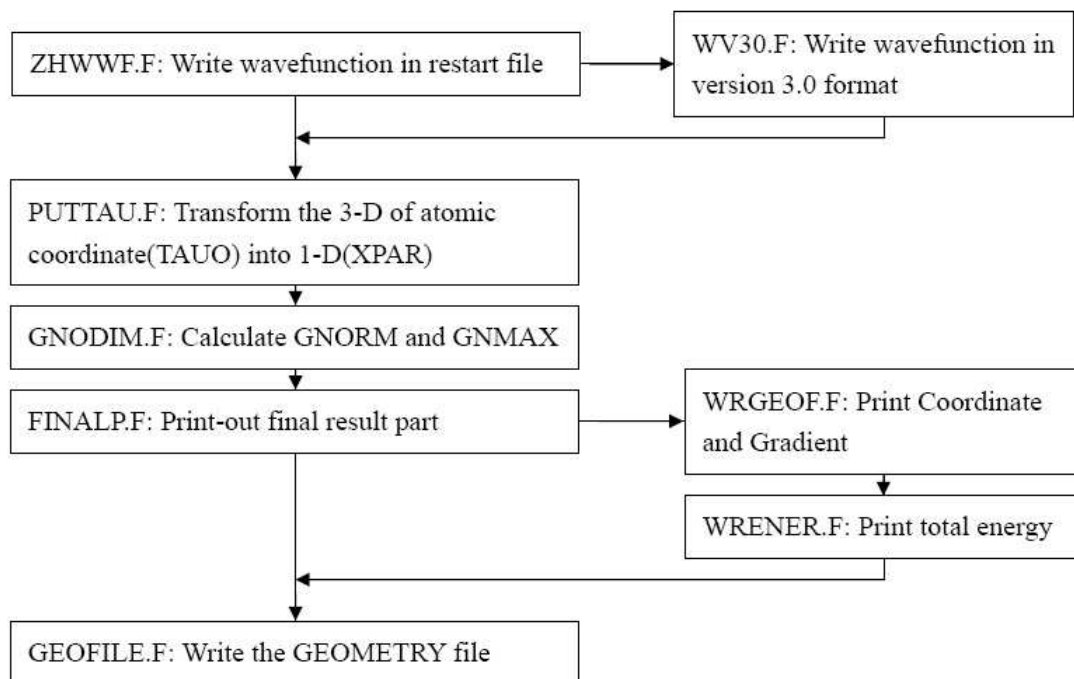
A.2 Optimize Geometry Procedure: Main Loop



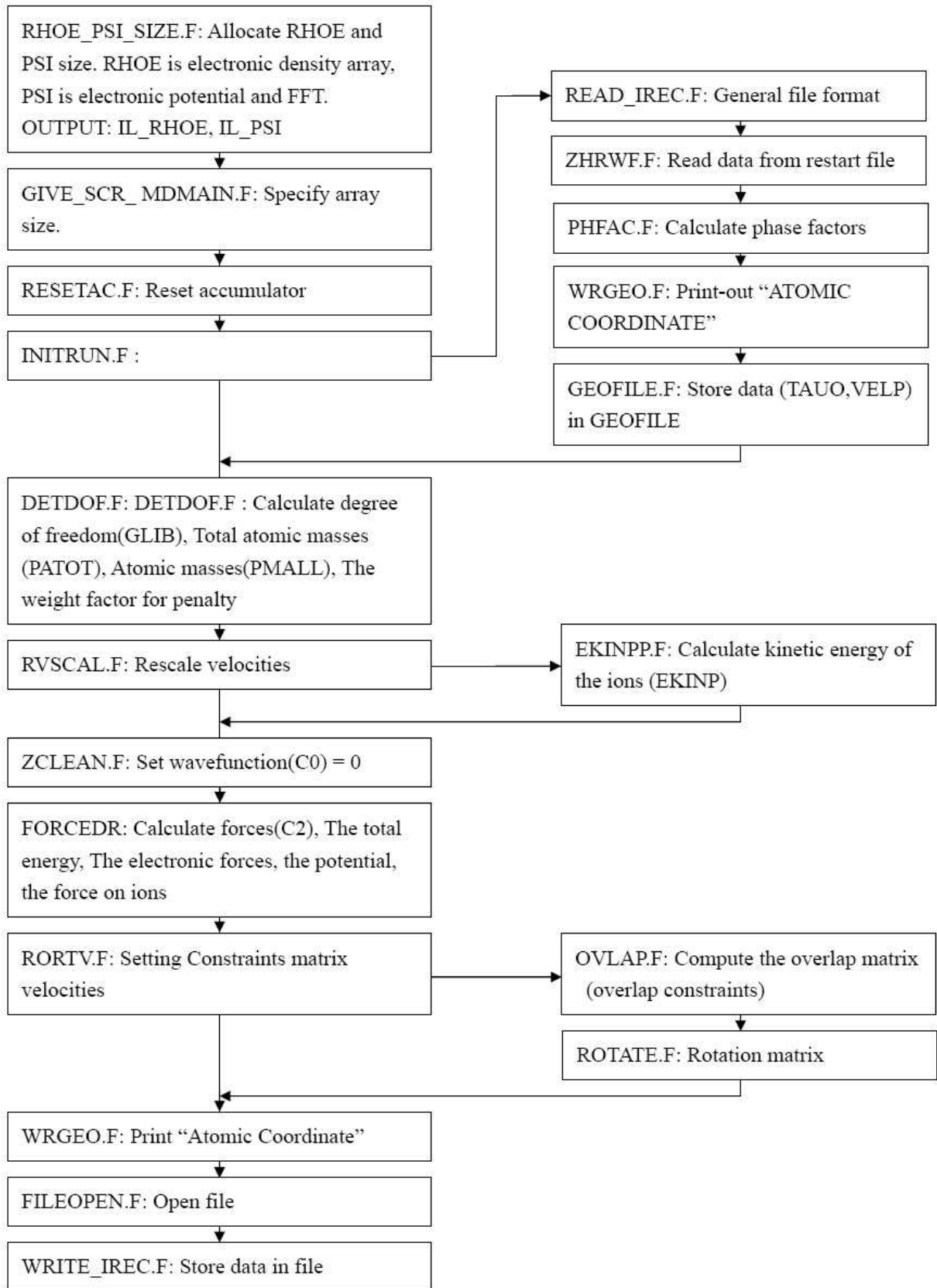
(§§) Loop for GEOPT



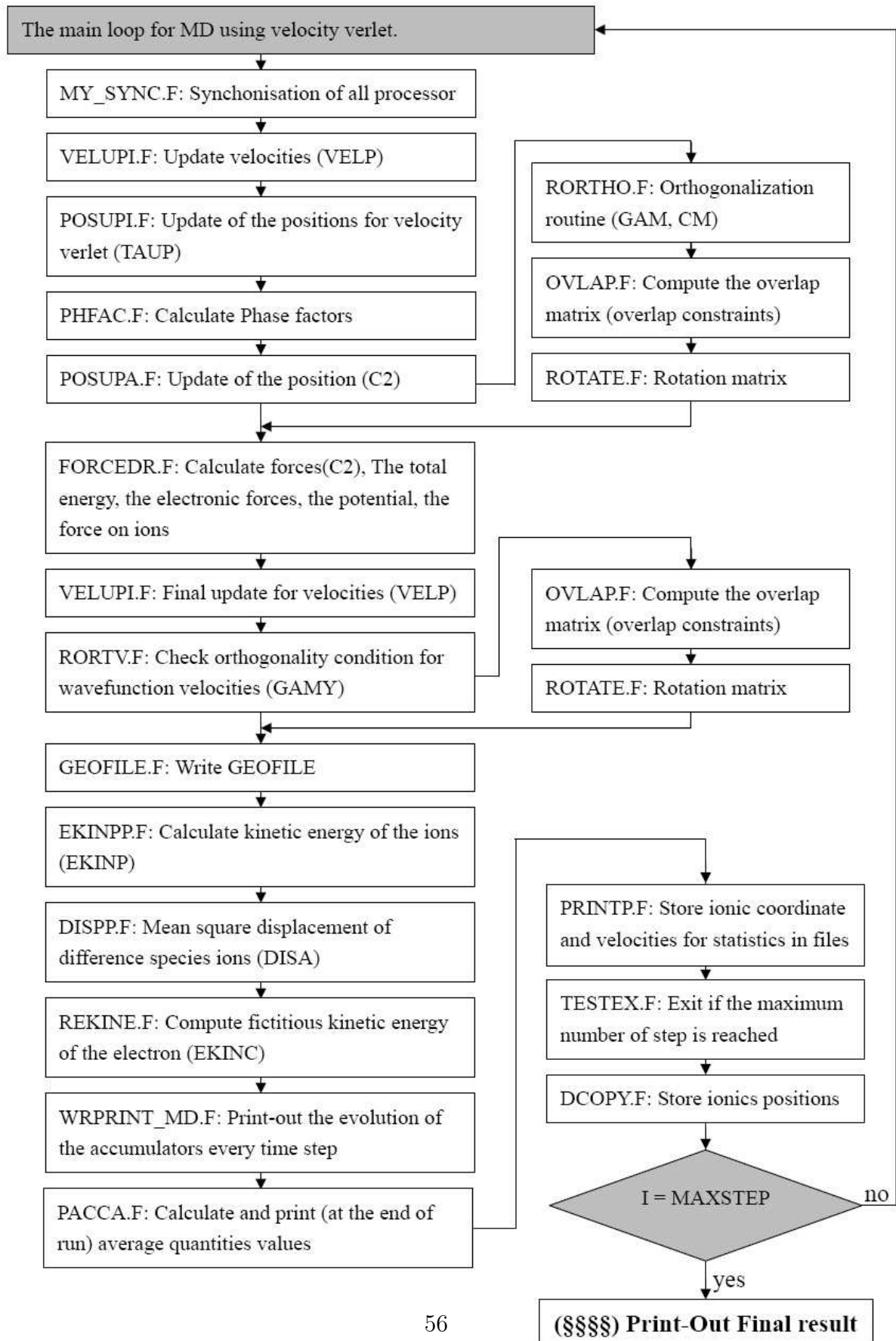
(§§§) Print-Out Final result

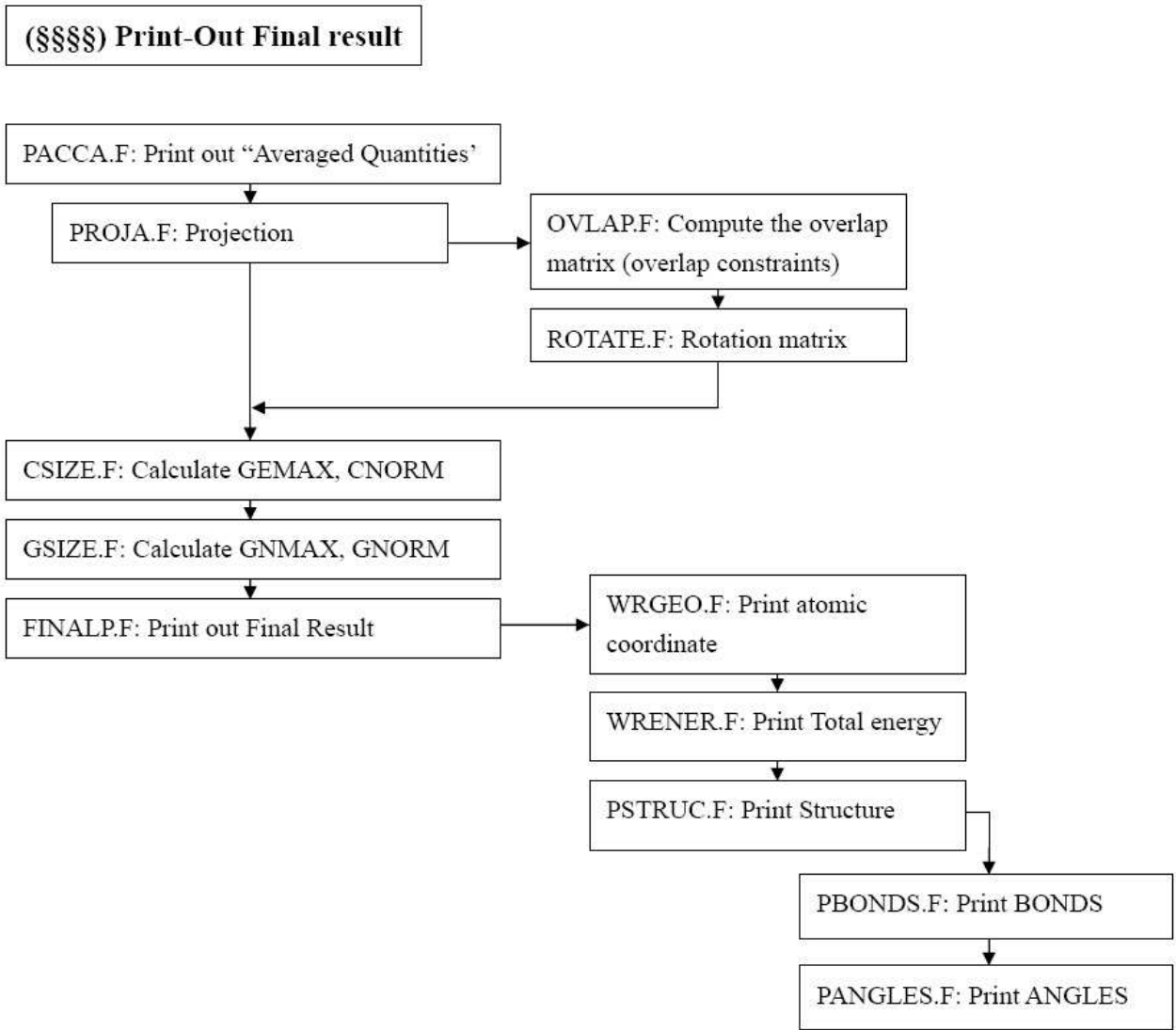


A.3 Molecular Dynamics CP Procedure: Initialization



A.4 Molecular Dynamics CP Procedure: Main Loop





References

- [1] M.C. Payne, M.P. Teter, D.C. Allan, T.A. Arias, J.D. Joannopoulos, “Iterative minimization techniques for ab initio total-energy calculations: molecular dynamics and conjugate gradients”, Reviews of Modern Physics, Vol. 64, pp. 1045-1096, October 1992.
- [2] Michael P. Marder., “Condensed matter physics”, John Wiley & Sons, New York, 2000.
- [3] Laboratoire Raynaud, Laurent Maron, Jean-Pierre Daudey, Franck Jolibois, “Reconsidering Car-Parrinello molecular dynamics using direct propagation of molecular orbitals developed upon Gaussian type atomic orbitals”, Phys. Chem. Chem. Phys., 6, pp. 4226-4232, 2004.
- [4] The CPMD consortium, “CPMD : Car-Parrinello Molecular Dynamics : An ab initio Electronic Structure and Molecular Dynamics Program”, March 29, 2006.
- [5] Jürg Hutter, “Lecture Notes : Introduction to Ab Initio Molecular Dynamics”, September 14, 2002.
- [6] Jürg Hutter, Hans Peter Lüthi, Michele Parrinello, “Electronic structure optimization in plane-wave-based density functional calculations by direct inversion in the iterative subspace”, Computational Materials Science, 2, pp. 244-248, 1994.
- [7] Ödön Farkas, H. Bernhard Schlegel, “Method for optimizing large molecules: Part 3. An improved algorithm for geometry optimization using direct inversion in the iterative subspace (GDIIS)”, Phys. Chem. Chem. Phys., 4, pp. 11-15, 2002.
- [8] R. Car, M. parrinello, “Unified Approach for Molecular Dynamics and Density-Functional Theory”, Physical Review Letters, Volume 55, Number 22, pp. 2471-2474, November 1985.