Documentation for the RTA code in 3D

F. Coppens[a], M. Vincendon[a], P.-G. Reinhard[c], E. Suraud[a,b]

[a] *Université de Toulouse; UPS; Laboratoire de Physique Théorique, IRSAMC; F-31062 Toulouse Cedex, France*

[b] *Laboratoire de Physique Théorique, Université Paul Sabatier, CNRS, F-31062 Toulouse Cédex, France*

[c] *Institut für Theoretische Physik, Universität Erlangen, D-91058 Erlangen, Germany*

Abstract

This is a first draft for the CPC presenting the TDLDA+RTA code to the public. The layout of presentation has yet to be discussed. Presently it mixes theory and algorithm with code. We may also consider collecting all theory & numerics together then followed by a huge block detailing the code.

Keywords: electronic dissipation, time-dependent density functional theory, ...

1. Introduction

PGR2all: ... yet to be written ...

PGR2all: Two general points yet to be clarified:

The code is still capable of coupling to a dielectric environment. We should skip that for the CPC publication.

Another question to be discussed is whether we maintain the option to run finite differences instead of FFT. If we do so, then we have to test that branch carefully.

2. Formal background: electronic DFT coupled to ionic motion

2.1. Brief introduction to TDLDA-MD

PGR2all: The explanation of formal background is extremely compact and reduced to the basic formula. Here we should provide a breif overview of the history of DFT and of the physics behind.

*Corresponding author: coppens@irsamc.ups-tlse.fr

2.2. The total energy of the model

The basic dynamical variables are the set of single-particle (s.p.) wavefunctions with their occupation probabilities at teh side of electrons and classical coordniates and momenta for the ions, together

$$\begin{array}{lll} \text{s.p. wavefunctions:} & \varphi_\alpha\ ,\ \alpha=1...\Omega & , \\ \text{s.p. occupation probabilities:} & w_\alpha\ ,\ \alpha=1...\Omega & , \\ \text{ionic coordinates:} & \mathbf{R}_I\ ,\ I=1,N_{\rm ion} & , \\ \text{ionic momenta:} & \mathbf{P}_I\ ,\ I=1,N_{\rm ion} & . \end{array} \tag{1}$$

A key quantity in connection with energy-density functionals is the electronic local density

$$\varrho_\uparrow\mathbf{r}) = \sum_{\alpha\in\uparrow} w_\alpha\varphi_\alpha^+(\mathbf{r})\varphi_\alpha(\mathbf{r}) \quad , \quad \varrho_\downarrow\mathbf{r}) = \sum_{\alpha\in\downarrow} w_\alpha\varphi_\alpha^+(\mathbf{r})\varphi_\alpha(\mathbf{r}) \quad , \tag{2}$$

which covers, in fact, two separate densities for spin-up and spin-down. To simplify the presentation of the formalism we ignore this distinction in the following and use simply one density $\varrho(\mathbf{r})$.

Starting point is an expression for the total energy of the coupled electronic and ionic system:

$$\begin{aligned}
E_{\rm total} &= E_{\rm kin} + E_{\rm C} + E_{\rm xc} + E_{\rm ext} + E_{\rm el,ion} + E_{\rm kin.ion} + E_{\rm pot,ion}\ , &\text{(3a)}\\
E_{\rm kin}[\{\varphi_\alpha\}] &= \int d\mathbf{r}\sum_\alpha w_\alpha\varphi_\alpha^+\frac{\hat{p}^2}{2m}\varphi_\alpha \quad , &\text{(3b)}\\
E_{\rm C}[\varrho] &= \frac{e^2}{2}\int d^3r d^3r'\,\frac{\varrho(\mathbf{r})\varrho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} &\text{(3c)}\\
E_{\rm xc}[\varrho] &= \int d\mathbf{r}\,\varrho(\mathbf{r})\epsilon_{\rm xc}\left(\varrho(\mathbf{r})\right) \quad , &\text{(3d)}\\
E_{\rm ext}[\varrho,t] &= \int d\mathbf{r}\,\varrho(\mathbf{r})U_{\rm ext}(\mathbf{r},t) \quad , &\text{(3e)}\\
E_{\rm el,ion}[\{\varphi_\alpha\},\{\mathbf{R}_I\}] &= \sum_I\int d\mathbf{r}\sum_\alpha w_\alpha\varphi_\alpha^+\hat{V}_{\rm PsP}(\mathbf{r}-\mathbf{R}_I)\varphi_\alpha \quad , &\text{(3f)}\\
E_{\rm kin,ion}(\mathbf{P}_I) &= \sum_I\frac{\mathbf{P}_I^2}{2M_I} \quad , &\text{(3g)}\\
E_{\rm pot,ion}(\mathbf{R}_I) &= \frac{1}{2}\sum_{J\neq I}\frac{e^2}{|\mathbf{R}_I-\mathbf{R}_J|} + U_{\rm ext,ion}(\mathbf{R}_I,t) \quad , &\text{(3h)}
\end{aligned}$$

where functionals of density are wavefunctions are indicated by square brackts and functions of coordinates by round brackets. The $E_{\rm C}$ is the direct part of the electronic Coulomb energy. The $E_{\rm xc}$ is the energy-density functional for electronic exchange and correlations for which we use in the code two options: the functional of [1] or the older from from [2]. The $E_{\rm ext}$ stands for the excitation mechanisms by external sources, either from a photon pulse of from the Coulomb field of a fast bypassing ion, see section 2.2.5. This part is, of course, absent in static calculations of the ground state. The $E_{\rm el,ion}$ carries the interaction with the ions which is usually described by a pseudo-potential $\hat{V}_{\rm PsP}$, see section 2.2.2, or may be simplified in terms of the jellium model, see section 2.2.3. Ionic kinetic and potential energy are described by the obvioous classical expressions. The $U_{\rm ext,ion}$ in the ionic potengtial energy describes the action of an external field on the ions (which is usually negligible as compared to the effect on the electrons).

2.2.1. Self-interaction correction (SIC)

PGR2all: We have yet to decide which levels of SIC we keep in the code and then explain them here properly.

2.2.2. The pseudo-potentials

There is a great variety of pseudo-potentials available. The code employs two variants. Particularly efficient and simple to use are local pseudo-potentials integrated from a pseudo-density which is represented as sum of Gaussians. The corresponding potential is then a sum of error functions

$$V_{\rm PsP}(\mathbf{r}) = \sum_{i=1}^{2} c_i \frac{\mathrm{erf}(\mathbf{r}/\sigma_i)}{|(\mathbf{r})|} \quad , \tag{4a}$$

$$\mathrm{erf}(x) = \sqrt{\frac{2}{\pi}} \int_0^x dy\, e^{-y^2} \quad , \tag{4b}$$

where the σ_i are widths and the strength parameters c_i have to line up to the total charge of the ionic core $c_1 + c_2 = Z_{\rm ion}$. This pseudo-potential is well suited for alkaline atoms for which it was originally developed [3].

More involved, but also more versatile in the applicability, is the pseudo-potential from [4] which is composed from a local and a non-local part as

$$\hat{V}_{\rm PsP} = V_{\rm loc}(\mathbf{r}-\mathbf{R}) + \hat{V}_{\rm nloc} \quad , \tag{5a}$$

$$V_{\rm loc}(\mathbf{r}-\mathbf{R}) = -\frac{Z_{\rm ion}}{x}\mathrm{erf}\,(x) + e^{-x^2}\sum_{n=0}^{3} C_{n+1}2^n x^{2n} \quad , \quad x = \frac{|\mathbf{r}-\mathbf{R}|}{r_{\rm loc}} \quad , \tag{5b}$$

$$\hat{V}_{\rm nloc}(\mathbf{r},\mathbf{r}') = \sum_{jl\mu} \mathcal{G}(\mathbf{r}-\mathbf{R})\mathcal{H}_{jl}\mathcal{G}(\mathbf{r}'-\mathbf{R}) \quad , \tag{5c}$$

where $\mathbf{R}$ is the position of the ionic core with respect to which the pseudo-potential is defined. The $\mathcal{G}$ in the non-local part (5c serve to project out the electronic states which are occupied in the ionic core. These projector functions for the first few quantum numbers read

$$\mathcal{G}_{000}(\mathbf{x}) = r_{\rm nloc}^{-3}\pi^{-3/2}\exp\left(-\frac{x^2}{2r_{\rm nloc}^2}\right) \quad , \tag{5d}$$

$$\mathcal{G}_{01\mu}(\mathbf{x}) = \sqrt{\frac{2}{3}} r_{\rm nloc}\nabla_\mu \mathcal{G}_{000}(\mathbf{x}) \quad , \tag{5e}$$

$$\mathcal{G}_{100}(\mathbf{x}) = \frac{r_{\rm nloc}^2}{\sqrt{6}}\left(2\Delta - \frac{3}{r_{\rm nloc}^2}\right)\mathcal{G}_{000}(\mathbf{x}) \quad . \tag{5f}$$

The projection looks expensive at first glance. However, one can exploit the fact that the Gaussians cover only a small region space of order of a few multiples of the non-local radius $r_{\rm nloc}$. Thus one needs to evaluate the projector only on a small sub-grid which reduces expense dramatically.

2.2.3. The soft jellium model for the ionic background

The electronic wavefunctions of metals and metal clusters are spread softly over the whole system and hardly resolve the spatial structure the ionic cores. This allow to replace the detailed ionic background by a smooth, positive background density. Such jellium approximation is a standard in the theory of bulk metals [5] and the adaptation to a finite cluster is straightforward. One carves from bulk jellium a finite element of constant positive charge corresponding to the average bulk density. The volume is chosen such that its total charge coincides with the given ionic charge. For finite clusters, it is advantageous to use jellium with a soft surface profile. This is more suited for numerical handling and it improves the quakity of the model, e.g., by producing a correct peak energy for the optical response of metal clusters [6]. Versatile and easy to handle in this respect is a Woods-Saxon profile for the jellium density

$$\varrho_{\rm jel}(\mathbf{r}) \quad = \quad \frac{3}{4\pi r_s^3}\left[1+\exp\left(\frac{|\mathbf{r}|-R(\vartheta,\phi)}{\sigma_{\rm jel}}\right)\right]^{-1} \quad , \tag{6a}$$

$$\text{with} \qquad R(\vartheta,\phi) \quad = \quad R_{\rm jel}\left(1+\sum_{lm}\alpha_{lm}Y_{lm}(\vartheta,\phi)\right) \quad , \tag{6b}$$

$$\int d\mathbf{r}\,\varrho_{\rm jel} \quad = \quad N_{\rm ion} \quad . \tag{6c}$$

The jellium radius $R_{\rm jel}$ is determined by eq. (6c), the normalization to desired number of ions. The central density is determined by the bulk density $\rho_0 = 3/(4\pi r_s^3)$ and the Wigner-Seitz radius r_s is a genuine material parameter [5]. The $\sigma_{\rm jel}$ parametrizes the surface width and the transition from 90% to 10% bulk density is achieved within $4\sigma_{\rm jel}$. The model does also allow to describe deformed clusters by angular dependence $R(\vartheta,\phi)$ of the extension. This is achieved through the deformation coefficients α_{lm} weighting the impact of the spherical harmonics. For example, axially symmetric ellipsoids are tuned by α_{20}, positive values producing prolate and negative values oblate shapes. The α_{30} generate octupole (pear-like) shapes which have considerable influence in metal cluster spectroscopy and α_{40} generates hexadecapole which play a role for fine-tuning the shape [7]. Triaxial shapes are produced by moments with $m \neq 0$. The cluster radius $R_{\rm jel}$ is fixed by the others parameters through eq. (6c) which is to be solved numerically by a root finding procedure. After all, the leading parameters of the soft jellium model Eq. (6) are the Wigner-Seitz radius r_s and the surface thickness $\sigma_{\rm jel}$. They are universal for a given material. Typical values are $r_s \sim 4\,{\rm a}_0$ with $\sigma_{\rm jel} \sim 0.9\,{\rm a}_0$ for Na clusters, $r_s \sim 2.66\,{\rm a}_0$ and $\sigma_{\rm jel} \sim 0.76\,{\rm a}_0$ for Mg clusters, or $r_s \sim 3\,{\rm a}_0$ and $\sigma_{\rm jel} \sim 0.78\,{\rm a}_0$ for Ag clusters. The deformation parameters α_{lm} depend on the actual cluster and strongly vary with system and size.

The jellium approximation then consists in discarding the ionic contribution to the total energy (3), i.e. the terms $E_{\rm kin,ion}$ and $E_{\rm pot,ion}$, and to replace the pseudo-potential background in $E_{\rm el,ion}$ by the Coulomb potential of the jellium density (6). There is no dynamics associated with the jellium. The model thus applies to situations where ionic motion can be ignored.

2.2.4. Phenomenological electronic shell models

Experience shows that KS potentials for metal clusters have a typical profile characterized by a flat bottom and a smooth transition to zero. For neutral clusters, one can approximate that very well by a Woods-Saxon profile

$$U_{\rm WS}(\mathbf{r}) = -U_0 \left[1 + \exp\left(\frac{|\mathbf{r}| - R(\vartheta,\phi)}{\sigma_{\rm WS}}\right)\right]^{-1} \quad , \tag{7}$$

where, again, a possible deformation can be parametrized by an angular dependence of the radius $R(\vartheta,\phi) = R_0\,(1+\sum_{lm}\alpha_{lm}Y_{lm}(\vartheta,\phi))$, similar to the jellium model of Eq. (6) for the density. In Eq. (7) the potential depth U_0 is taken as the average binding potential in bulk matter. Computations in a fixed potential are somewhat simpler than with the self-consistent scheme. The Woods-Saxon model has thus been very often used for describing the electronic shell structure of clusters, particularly in early studies, for a review see [8]. It is meanwhile somewhat out of fashion. But we provide it in the code for quick tests and comparison.

A substantial further simplification is achieved by realizing that shell effects are determined by the states near the Fermi surface and that these states practically see a harmonic potential. This suggests to use for first estimates a simple harmonic oscillator shell model. To mimic surface profile of the Woods-Saxon potential and so reproduce the sequence of shell closures in metal clusters a phenomenologically tuned $\hat{l}^2$ term is added to the oscillator model. This yields the Clemenger-Nilsson model for the mean field potential

$$\hat{U}_{\rm CN} = \frac{m}{2}\left(\omega_x^2x^2 + \omega_y^2y^2 + \omega_z^2z^2\right) - U_{l2}\hbar\bar{\omega}\left(\hat{l}^2 - n(n+3)/6\right) \tag{8}$$

where n is the global shell number ($n = n_x + n_y + n_z$). The three separate curvatures ω_i allow one to accommodate deformed situations including triaxiality. Volume conservation restricts their choice to $\omega_x\omega_y\omega_z = \omega_0^3$ =constant. The separate values can also be expressed in terms of the quadrupole deformation α_{lm} as introduced above and in Eq. (6), e.g. for axially symmetric shapes by $\omega_x = \omega_y = \omega_0\exp{(\alpha_{20}\sqrt{5/4\pi})}, \omega_z = \omega_0\exp{(-2\alpha_{20}\sqrt{5/4\pi})}$. The parameter U_{l2} serves to tune the downshift of high angular momentum orbits and thus of the shell sequence. The Clemenger-Nilsson model was introduced into clusters physics in [9] taking up a much similar nuclear oscillator model [10]. An extensive review of early applications is given in [11]. The oscillator model is kept in the code mainly for testing because one has analytical the solutions for $U_{l2} = 0$.

Both shell models, Woods-Saxon or harmonic oscillator, overrule the total energy (3). What remains is only the kinetic energy (3b) together with the given model potentials. Also this approach is, as the jellium model, applicable only in circumstances where ionic motion plays little role.

2.2.5. External excitation fields

Lasers are the most important and very flexible means for a dedicated, well tuned excitation of electronic systems. They produce a strong coherent electromagnetic field which can be well approximated by a classical time-dependent electromagnetic field.Typical wavelengths are in the range of several hundredths of nm. This is a huge distance as compared to the spatial extension of atoms, molecules, and (most) clusters. One can thus treat the laser field in the limit of long wavelengths ($k \rightarrow 0$) which means to deal with a spatially homogeneous electrical field $\mathbf{E}$ at the cluster site and we can neglect the effect of the magnetic field. The coupling Hamiltonian leaves of gauge transformation [12]. The external-field operator in velocity gauge reads[1]

$$U_{\rm ext} = e\mathbf{E}_0 F(t)\hat{\mathbf{p}} \quad , \quad F(t) = \int dt'\, f(t') \exp\left(-{\rm i}\omega_{\rm las}t'\right) \quad , \tag{9}$$

where $f(t')$ is the envelope of the photon pulse. The same pulse in space gauge becomes

$$U_{\rm ext} = e\mathbf{E}_0 f(t)\cdot\hat{\mathbf{r}} \exp\left(-{\rm i}\omega_{\rm las}t\right) \tag{10}$$

which is simpler to handle because the laser field acts here simply as a time-dependent local operator. The code thus uses the external field in space gauge. Transformation to velocity gauge, if needed, can be performed a posteriori by standard rules of gauge transformation [13][2]. The photon pulse is characterized by frequency $\omega_{\rm las}$, peak field strength $E_0 = |\mathbf{E}_0|$, polarization $\mathbf{E}_0/E_0$, and time profile $f(t)$. The field strength is usually parametrzed in terms of the laser intensity I as

$$E_0 = c_{EI} I^{1/2} \quad , \quad c_{EI} = 1.07 * 10^{-8} \frac{\rm eV}{\rm a_0} \left(\frac{\rm W}{\rm cm^2}\right)^{-1/2} \quad . \tag{11}$$

The profile is a matter of debate. The experimental pulse profiles are not precisely known either. It is assumed to be a well peaked function with a certain full-width at half maximum (FWHM), in practice well approximated by a Gaussian. In order to have a finite support for the pulse, we use the $\sin^2$ pulse

$$f(t) = \begin{cases} \sin^2\left(\pi \frac{t}{2T_{\rm pulse}}\right) & \text{for} \quad t \in \{0, 2T_{\rm pulse}\} \\ 0 & \text{else} \end{cases} \quad . \tag{12}$$

It is close to a Gaussian pulse in the vicinity of peak field strength and combines high spectral selectivity with finite bounds. Note that the form (12) is scaled such that the pulse parameter $T_{\rm pulse}$ is identical with the full-width at half-maximum (FWHM).

Probing electronic systems by beams of charged particles is a standard tool in atomic and molecular physics [14] and is also used in cluster physics. Highly charged ions, protons and to some extend also electrons can be considered as being structureless. What counts is only their Coulomb field. Charged ions are heavy and can be treated with classical trajectories $\mathrm{R}_{\rm ext}(t)$. For sifficvioently heavy and fast projectiles, one can approximate these trajectories by straight lines and that is what is done in the code. In any case, the effect of the ion (charge $Z_{\rm ext}$) on the system can again be described by a time dependent external field

$$U_{\rm ext}(\mathbf{r},t) = \frac{Z_{\rm ext}e^2}{|\mathbf{r} - \mathbf{R}_{\rm ext}(t)|} \quad . \tag{13}$$

Magnetic effects are neglected. They may play a role only for extremely fast ions in the relativistic domain. The ionic trajectories are characterized by the ion velocity $v_{\rm ion}$ and

[1] I am not sure that we shoulod show velocity gauge. It may be useful whne explaining the computation of PES later on.

[2] Cross ref to phase correction in PES in section 2.4.

the impact parameter $b_{\rm ion}$ which is the distance of closest approach. The velocity is more or less well defined by the experimental setup. But the impinging ion beam will cover a broad range of impact parameters. From the theoretical side, one has then to run several calculations with systematically varied impact parameters. Reaction cross sections are then computed by integration of the reaction probability over impact parameters, for examples see [15].

2.3. Equations of motion

2.3.1. Electronic Kohn-Sham equations

Static and dynamical equations determining structure and dynamics of the electron cloud are determined by variation of the total energy with respect to the s.p. wavefunctions φ_α. This yields what is called the Kohn-Sham (KS) equations [16]. We use energy-density functionals [1, 2] i.e. functionals in local density approximation (LDA). The dynamical application is called time-dependent LDA (TDLDA) and we use this acronym also in cases where we add SIC (see section 2.2.1) to the treatment.

The time-dependent KS equations determining the propagarion of electronic ground wavefunctions are

$$\begin{aligned}
\hat{h}_{\rm KS}\varphi_\alpha &= {\rm i}\partial_t\varphi_\alpha \quad \text{for} \quad \alpha\in\{1,...,N_{\rm el}\} \quad , && \text{(14a)}\\
\hat{h}_{\rm KS} &= \frac{\hat{p}^2}{2m} + \underbrace{U_{\rm C}(\mathbf{r}) + \hat{U}_{\rm xc} + \hat{U}_{\rm back} + U_{\rm ext}(\mathbf{r},t)}_{U_{\rm KS}} \quad , && \text{(14b)}\\
U_{\rm C}(\mathbf{r}) &= e^2\int d^3r'\,\frac{\varrho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} \quad , && \text{(14c)}\\
\hat{U}_{\rm xc}(\mathbf{r}) &= \epsilon\left(\varrho(\mathbf{r})\right) + \varrho(\mathbf{r})\frac{\partial\epsilon_{\rm xc}\left(\varrho(\mathbf{r})\right)}{\partial\varrho} \quad . && \text{(14d)}
\end{aligned}$$

The exchange-correlation potential $\hat{U}_{\rm xc}$ involves a construction $\partial\epsilon_{\rm xc}/\partial\varrho$ which is the formal derivative of the exchange correlation energy per particle $\epsilon_{\rm xc}$ with respect to the density ϱ considered as an independent variable. The time-dependent KS equations analogously read

$$\hat{h}_{\rm KS}\varphi_\alpha = \varepsilon_\alpha\varphi_\alpha \quad \text{for} \quad \alpha\in\{1,...,N_{\rm el}\} \quad , \tag{14e}$$

where $\hat{h}_{\rm KS}$ is composed in the same manner as in the time-dependent case (instantaneous approximation for $U_{\rm xc}$). The Kohn-Sham equations (14e) pose a stationary eigenvalue problem. They provide the electronic ground state of a system. The time-dependent Kohn-Sham equations (14a) pose an initial value problem. The natural starting point is the ground state as obtained from the stationary Kohn-Sham equation. The numerical solution of the KS equations is explained in section 2.5. The time evolution delivered by the dynamical KS Eqs. (14a) can be expressed formally by the unitary one-body time-evolution operator

$$\hat{U}(t,t') = \hat{\mathcal{T}}\exp\left(-i\int_t^{t'}\hat{h}(t'')dt''\right) \tag{15a}$$

where $\hat{\mathcal{T}}$ is the time-ordering operator. This yields a closed expression for the time-evolution of s.p. states

$$|\phi_\alpha(t)\rangle = \hat{U}(t,t')|\phi_\alpha(t')\rangle. \tag{15b}$$

This compact form will be used later on in connection with the extension of the propagation by dissipation, see section 4.

2.3.2. LDA and TDLDA with mixed states

So far, we dealt with the equations for pure states. In case of stationary states at finite temperature and in case of dissipative we encounter mixed states. This is described by associating an occupation weight w_α with each s.p. state φ_α as provided already in the initial setup in section 2.2. The weights w_α becomes crucial in the dissipative dynamics, see section 4, they are kept frozen during TDLDA propagation, and they matter again for stationary states at finite temperature in which case they are determined by

$$w_\alpha = \frac{1}{1+\exp((\varepsilon_\alpha - \mu)/T)} \tag{16}$$

where the ε_α are the s.p. energies from teh stationary KS eqs. (14e) and μ is the chemical potential tuned such that the total electron number is reproduced by $\sum_\alpha w_\alpha = N$.

In case of mixed states, it is advantageous to express the entity $\{\varphi_\alpha, w_\alpha\}$ in compact manner by the one-body density operator

$$\hat{\rho} = \sum_{\alpha=1}^{\infty} |\phi_\alpha\rangle w_\alpha \langle\phi_\alpha| \quad . \tag{17}$$

The solution of the thermal KS eqs., i.e. eqs. (14e) together with (16), provides the density operator immediately in this form which is called natural orbital representation (diagonal in the s.p. states). In general, the one-body density operator can be non-diagonal with respect to given s.p. states. Tis will play a role later on.

As said above, the occupation weights w_α are kept frozen during TDLDA. The KS equations (14a) then can be written compactly as[3]

$$\mathrm{i}\partial_t\hat{\rho} = \left[\hat{h}[\varrho], \hat{\rho}\right] \tag{18}$$

where $\hat{h}[\varrho]$ is the KS Hamiltonian as above. This pure mean-field propagation (18) leaves the occupation weights W_α unchanged and propagates only the s.p. states. The mean-field propagation of an initial state (17) then reads

$$\hat{\rho}(t) \quad = \quad \sum_{\alpha=1}^{\infty} |\phi_\alpha(t)\rangle w_\alpha \langle\phi_\alpha(t)| = \hat{U}(t,0)\hat{\rho}(0)\hat{U}^{-1}(t,0) \tag{19}$$

where $\hat{U}$ is the mean-field evolution operator (15a).

2.3.3. Ionic dynamics

We finally add the ionic dynamics to complete the picture. Ions are described by classical Molecular Dynamics (MD), i.e. classical equations of motion, under the influence of their mutual Coulomb force, the forces experienced from the electrons, and possibly

[3] I wonder whethr we should shift this introduction of the one-body density matrix to the RTA section.

external forces. We start from the total energy (3) which serves as classical Hamiltonian in terms of $\mathbf{R}_I$ and $\mathbf{P}_I$. Standard variation yields the classical equations-of-motion for the ions of positions $\mathbf{R}_I$ and momenta $\mathbf{P}_I$

$$\partial_t \mathbf{P}_I = -\nabla_{\mathbf{R}_I} \left[E_{\text{pot,ion}}(\mathrm{R}_I) + \sum_{\alpha=1}^{N_{\text{el}}} (\varphi_\alpha | V_{\text{PsP}}(\mathbf{r}-\mathbf{R}_I) | \varphi_\alpha) \right] \quad , \tag{20a}$$

$$\partial_t \mathbf{R}_I = \frac{\mathbf{P}_I}{M_I} \quad . \tag{20b}$$

They are to be propagated simultaneously with TDLDA for the electrons, here represented by wavefunctions φ_α. The ionic motion is coined molecular dynamics (MD). The simultaneous propagation scheme is called TDLDA-MD. It applies to all dynamical situations including those that are far from the adiabatic limit and embraces truly diabatic scenarios. The practical realization adds the (simple) classical propagation to the evolution of the electronic states. For solution schemes, see section 2.5.[4]

2.4. Observables

PGR2all: ... yet to be written ...

2.5. Numerical schemes

PGR2all: ... yet to be written ...

3. The structure of the TDLDA package

3.1. The TDLDA calling tree

PGR2all: Here comes a calling tree as for RTA in section 5.2. This tree is much larger and may have to be split into static and dynamic part.

3.2. The TDLDA subroutines in detail

PGR2all: Here is to come a detailed description of all subroutines similar as for RTA in 5.3, but here for the LDA part. One example is given as appetizer.

`SUBROUTINE coul_mfield(rho)`

`rho(1:2*kdfull2)` in/out
density for which Coulomb field is computed

Computes Coulomb potential for given density by invoking Poisson solver. Th emerging coulomb potential is communicated as `chpcoul` via module `params`. In case of dielectric external media, adds pseudo-density for image charge.
PGR2all: Why is `rho` *also* `INTENT OUT`*? Is all* `1:2*kdfull2` *used or only the first block* `1:kdfull2`*?*

[4] Should we add a word on the limitations of TDLDA-MD?

4. Relaxation-time approximation (RTA)

4.1. The formal background of RTA

The quantum Boltzmann equation is the quantum mechanical counterpart of the semi-classical Vlasov-Uehling-Uhlenbeck equation [17, 18]. It complements the self-consistent TDLDA propagation of the one-body density matrix $\hat{\rho}$ by dynamical correlations through a collision term. It reads in general [19, 20] $\mathrm{i}\partial_t\hat{\rho} - \big[\hat{h},\hat{\rho}\big] = \hat{I}[\hat{\rho}]$ where the left hand side contains the mean-field propagation. the $\hat{I}$ at the right-hand side consists stands for the quantum-mechanical collision term which, however, is extremely hard to handle for finite Fermion systems. A great simplification can be achieved by the relaxation-time approximation (RTA) which was used successfully in a wide variety of homogeneous systems [21, 5]. The RTA equations for the present case of finite Fermion systems read [22]

$$\partial_t\hat{\rho} = -\mathrm{i}\big[\hat{h},\hat{\rho}\big] - \frac{1}{\tau_{\mathrm{relax}}}\left(\hat{\rho} - \hat{\rho}_{\mathrm{eq}}[\varrho,\mathbf{j},E_{\mathrm{sp}}]\right) \ , \tag{21a}$$

$$\varrho(\mathbf{r},t) = \sum_\alpha |\phi_\alpha(\mathbf{r},t)|^2\, W_\alpha \ , \tag{21b}$$

$$\mathbf{j}(\mathbf{r},t) = \sum_\alpha W_\alpha \phi^*_\alpha(\mathbf{r},t)\frac{\overrightarrow{\nabla} - \overleftarrow{\nabla}}{2\mathrm{i}}\phi_\alpha(\mathbf{r}) \ , \tag{21c}$$

$$E_{\mathrm{sp}} = \sum_\alpha W_\alpha \varepsilon_\alpha \ , \tag{21d}$$

$$\frac{\hbar}{\tau_{\mathrm{relax}}} = 0.40\frac{\sigma_{ee}}{r_s^2}\frac{E^*_{\mathrm{intr}}}{N} \ , \ r_s = \left(\frac{3\bar{\varrho}}{4\pi}\right)^{-1/3} \ , \ \sigma_{ee} = \sigma_{ee}(\bar{\varrho}) \ , \tag{21e}$$

where $\hat{\rho}_{\mathrm{eq}}$ is the density operator of the thermal equilibrium for local density $\varrho(\mathbf{r},t)$, current distribution $\mathbf{j}(\mathbf{r},t)$. The reference energy should be, in fact, the total energy $E(t)$ and computed from the actual state $\hat{\rho}(t)$. We replace that by the simpler total s.p. energy $E_{\mathrm{sp}}(t)$ which is legitimate because the difference to E is a functional of $\varrho(\mathbf{r})$ only, a quantity which is kept frozen by construction. The forms (21b,21c) hold for the diagonal representation. A crucial parameter is the relaxation time τ_{relax} which is taken over from semi-classical Fermi liquid theory, for details see [22]. Key entries are: the intrinsic (thermal) energy of the system E^*_{intr} (see appendix AppendixA), the actual number of particles N, the in-medium electron-electron cross section σ_{ee}, the effective Wigner-Seitz radius r_s of the electron cloud, and the average electron density $\bar{\varrho}$. Note that r_s and σ_{ee} depend on an average density $\bar{\varrho}$ because a spatially varying τ_{relax} would be very cumbersome to implement in a quantum mechanical expression. The average density is deduced from the r.m.s. radius r of the actual electron cloud as $\bar{\varrho} = (3N/(4\pi r^3))$[5].

This RTA equation (21) is rather involved because its entries depend in various ways on the actual state $\hat{\rho}(t)$. The most expensive piece is the instantaneous equilibrium density operator

$$\hat{\rho}_{\mathrm{eq}}[\varrho,\mathbf{j},E] = |\phi^{(\mathrm{eq})}_\alpha\rangle W^{(\mathrm{eq})}_\alpha \langle\phi^{(\mathrm{eq})}_\alpha| \tag{22a}$$

[5] Should we add that as displayed equation?

which minimizes LDA energy with constraint on the actual $\varrho(\mathbf{r})$, $\mathbf{j}(\mathbf{r})$ and energy $E_{\rm sp}$. It is determined by the density-constrained mean-field (DCMF) equation

$$\begin{aligned}
\hat{h}_{\rm DCMF}[\varrho,\mathbf{j},E]\phi_\alpha^{\rm (eq)} &= \varepsilon_\alpha^{\rm (eq)}\phi_\alpha^{\rm (eq)} && \text{(22b)}\\
\hat{h}_{\rm DCMF}[\varrho,\mathbf{j}] &= \hat{h} - \int d^3r\lambda(\mathbf{r})\hat{\varrho}(\mathbf{r}) - \int d^3r\,\boldsymbol{\lambda_j}(\mathbf{r})\hat{\mathbf{j}}(\mathbf{r}) \\
&\quad -\mu\int d^3r(\hat{\varrho}(\mathbf{r})-\varrho(\mathbf{r},t))^2 - \mu_j\int d^3r\,(\hat{\mathbf{j}}(\mathbf{r})-\mathbf{j}(\mathbf{r},t))^2 && \text{(22c)}
\end{aligned}$$

in combination with adjustment of particle number N and s.p. energy $E_{\rm sp}(t)$ through a Fermi distribution

$$\begin{aligned}
W_\alpha^{\rm (eq)} &= \frac{1}{1+\exp((\langle\phi_\alpha^{\rm (eq)}|\hat{h}|\phi_\alpha^{\rm (eq)}\rangle-\mu^{\rm (eq)})/T^{\rm (eq)})} && \text{(22d)}\\
\mu^{\rm (eq)} &\leftrightarrow \sum_\alpha W_\alpha^{\rm (eq)} = N(t)\ ,\ T^{\rm (eq)} \leftrightarrow \sum_\alpha W_\alpha^{\rm (eq)}\langle\phi_\alpha^{\rm (eq)}|\hat{h}|\phi_\alpha^{\rm (eq)}\rangle = E_{\rm sp}(t)\ . && \text{(22e)}
\end{aligned}$$

Although cumbersome to evaluate, it is important to use exactly this local, instantaneous equilibrium in the relaxation term. This guarantees that the dissipative step conserves local density, current, and energy as it is mandatory for a good collision term [23].

4.2. Observables specific to relaxation

Most of the observables computed with RTA are exactly the same as for TDLDA, e.g., energy, density, excitation spectra, ionization, PES, or PAD. New are observables related to the mixed character of the one-body operator which is characterized by the occupation numbers W_α. A specific quantity in that respect is the entropy which is computed in diagonal representation (17) by the standard expression [24]

$$S = -\sum_\alpha \left[W_\alpha \log W_\alpha + (1-W_\alpha)\log(1-W_\alpha)\right] \tag{23}$$

in units of Boltzmann constant. It serves as a direct indicator of thermalization and allows to read off the typical time scale of relaxation processes.

4.3. Summary of the RTA procedure

The solution of the RTA equations (21) with (22) is rather involved. We briefly summarize the solution scheme for one step from $t\equiv t_0$ to $t{+}\Delta t\equiv t_1$, for more details see [22][6]. The TDLDA propagation runs at a much faster pace than relaxation. We resolve it by standard techniques [25, 26] on a time step δt which is much smaller (factor 10–100) than the RTA step Δt. We summarize this TDLDA propagation in the evolution operator $\hat{U}$ from Eq. (15a) and discuss only one RTA step:

1. We first propagate $\hat{\rho}$ by pure TDLDA. The s.p. states in diagonal representation (17) evolve as $|\phi_\alpha(t)\rangle \rightarrow |\phi_\alpha^{\rm (mf)}\rangle = \hat{U}(t_1,t_0)|\phi_\alpha(t)\rangle$, while the occupation weights $W_\alpha(t_1) = W_\alpha(t_0)$ are kept frozen (pure mean-field propagation).

[6] PGR2all: can we outsource all details to reference [22]?

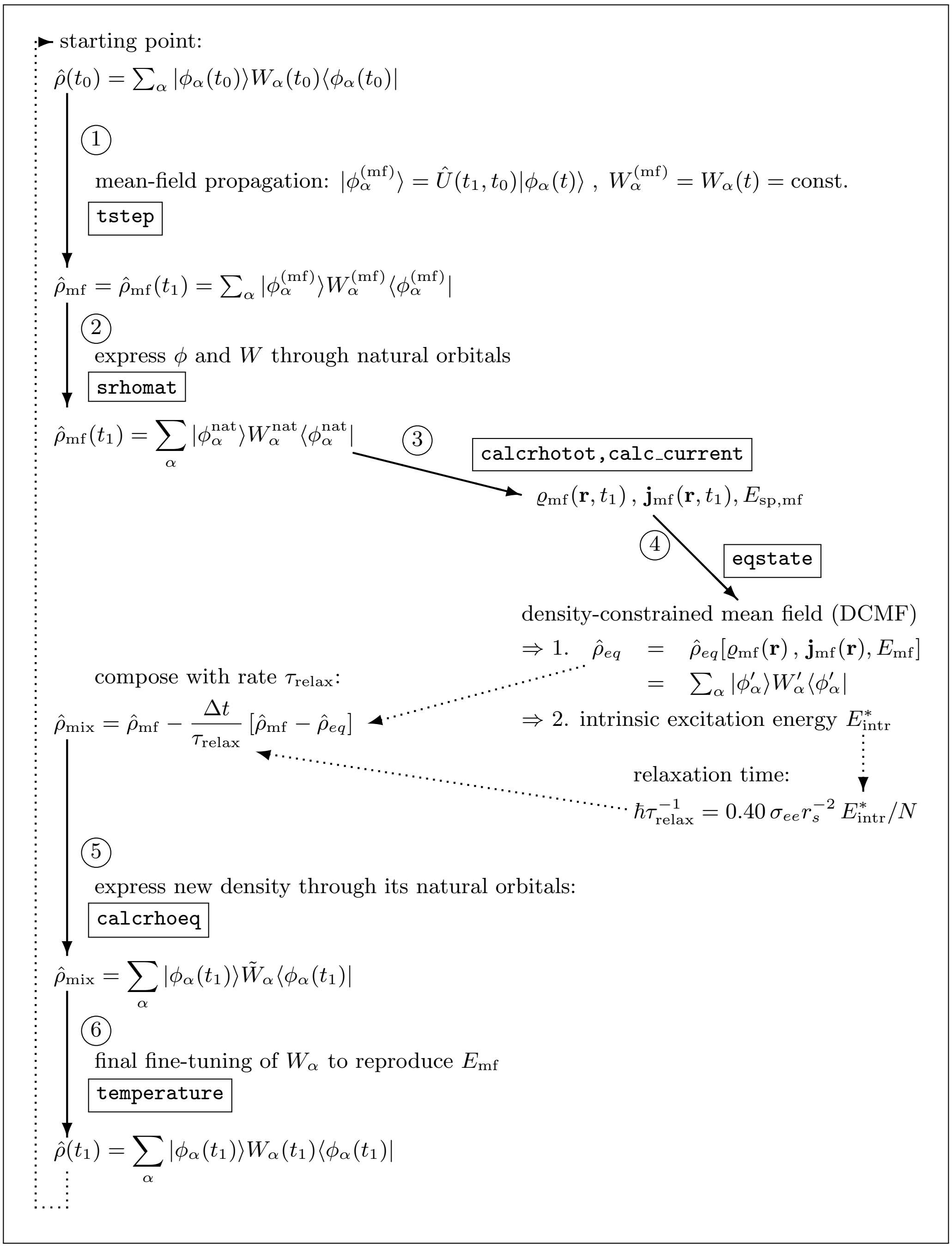

starting point:
$\hat\rho(t_0) = \sum_\alpha |\phi_\alpha(t_0)\rangle W_\alpha(t_0)\langle\phi_\alpha(t_0)|$
1
mean-field propagation: $|\phi_\alpha^{(\mathrm{mf})}\rangle = \hat U(t_1,t_0)|\phi_\alpha(t)\rangle$, $W_\alpha^{(\mathrm{mf})} = W_\alpha(t) = \mathrm{const.}$
tstep
$\hat\rho_{\mathrm{mf}} = \hat\rho_{\mathrm{mf}}(t_1) = \sum_\alpha |\phi_\alpha^{(\mathrm{mf})}\rangle W_\alpha^{(\mathrm{mf})}\langle\phi_\alpha^{(\mathrm{mf})}|$
2
express ϕ and W through natural orbitals
srhomat
$\hat\rho_{\mathrm{mf}}(t_1) = \sum_\alpha |\phi_\alpha^{\mathrm{nat}}\rangle W_\alpha^{\mathrm{nat}}\langle\phi_\alpha^{\mathrm{nat}}|$
3
calcrhotot,calc_current
$\varrho_{\mathrm{mf}}(\mathbf{r},t_1)$, $\mathbf{j}_{\mathrm{mf}}(\mathbf{r},t_1), E_{\mathrm{sp,mf}}$
4
eqstate
density-constrained mean field (DCMF)
$\Rightarrow$ 1. $\hat\rho_{eq} = \hat\rho_{eq}[\varrho_{\mathrm{mf}}(\mathbf{r}), \mathbf{j}_{\mathrm{mf}}(\mathbf{r}), E_{\mathrm{mf}}] = \sum_\alpha |\phi'_\alpha\rangle W'_\alpha\langle\phi'_\alpha|$
$\Rightarrow$ 2. intrinsic excitation energy E^*_{intr}
compose with rate τ_{relax}:
$\hat\rho_{\mathrm{mix}} = \hat\rho_{\mathrm{mf}} - \frac{\Delta t}{\tau_{\mathrm{relax}}}[\hat\rho_{\mathrm{mf}} - \hat\rho_{eq}]$
relaxation time:
$\hbar\tau_{\mathrm{relax}}^{-1} = 0.40\,\sigma_{ee} r_s^{-2}\,E^*_{\mathrm{intr}}/N$
5
express new density through its natural orbitals:
calcrhoeq
$\hat\rho_{\mathrm{mix}} = \sum_\alpha |\phi_\alpha(t_1)\rangle \tilde W_\alpha\langle\phi_\alpha(t_1)|$
6
final fine-tuning of W_α to reproduce E_{mf}
temperature
$\hat\rho(t_1) = \sum_\alpha |\phi_\alpha(t_1)\rangle W_\alpha(t_1)\langle\phi_\alpha(t_1)|$

Figure 1: Sketch of the scheme for performing one large time step $t_0 \longrightarrow t_1 = t_0 + \Delta t$ in solving the RTA equations. The numbers in open circles indicate the steps as outlined in the text. The names in `typewriter` font refer to subroutines in the code as detailed in section 5.

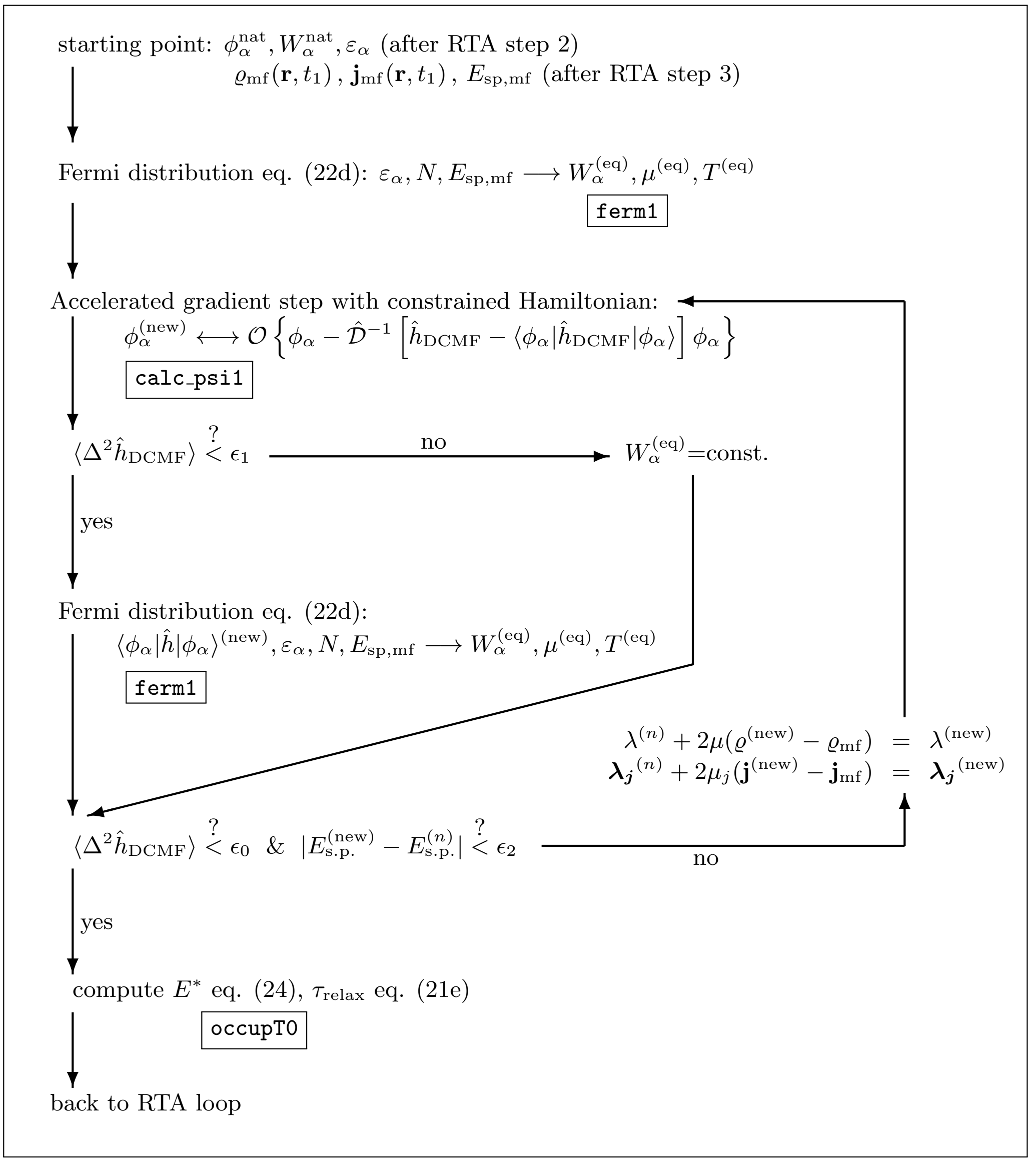

starting point: $\phi_\alpha^{\rm nat}, W_\alpha^{\rm nat}, \varepsilon_\alpha$ (after RTA step 2)
$\varrho_{\rm mf}(\mathbf{r},t_1)\,,\,\mathbf{j}_{\rm mf}(\mathbf{r},t_1)\,,\,E_{\rm sp,mf}$ (after RTA step 3)
Fermi distribution eq. (22d): $\varepsilon_\alpha, N, E_{\rm sp,mf} \longrightarrow W_\alpha^{\rm (eq)}, \mu^{\rm (eq)}, T^{\rm (eq)}$
ferm1
Accelerated gradient step with constrained Hamiltonian:
$\phi_\alpha^{\rm (new)} \longleftrightarrow \mathcal{O}\left\{\phi_\alpha - \hat{\mathcal{D}}^{-1}\left[\hat{h}_{\rm DCMF} - \langle\phi_\alpha|\hat{h}_{\rm DCMF}|\phi_\alpha\rangle\right]\phi_\alpha\right\}$
calc_psi1
$\langle\Delta^2\hat{h}_{\rm DCMF}\rangle \overset{?}{<} \epsilon_1$
no
$W_\alpha^{\rm (eq)}$=const.
yes
Fermi distribution eq. (22d):
$\langle\phi_\alpha|\hat{h}|\phi_\alpha\rangle^{\rm (new)}, \varepsilon_\alpha, N, E_{\rm sp,mf} \longrightarrow W_\alpha^{\rm (eq)}, \mu^{\rm (eq)}, T^{\rm (eq)}$
ferm1
$\lambda^{(n)} + 2\mu(\varrho^{\rm (new)} - \varrho_{\rm mf}) = \lambda^{\rm (new)}$
$\boldsymbol{\lambda_j}^{(n)} + 2\mu_j(\mathbf{j}^{\rm (new)} - \mathbf{j}_{\rm mf}) = \boldsymbol{\lambda_j}^{\rm (new)}$
$\langle\Delta^2\hat{h}_{\rm DCMF}\rangle \overset{?}{<} \epsilon_0$ & $|E_{\rm s.p.}^{\rm (new)} - E_{\rm s.p.}^{(n)}| \overset{?}{<} \epsilon_2$
no
yes
compute E^* eq. (24), $\tau_{\rm relax}$ eq. (21e)
occupT0
back to RTA loop

Figure 2: Sketch of the scheme for solving the DCMF eqs. (22). This scheme expands `SUBROUTINE eqstate` in step 4 from the RTA scheme 1. The symbol $\mathcal{O}$ stands for ortho-normalization of the new set of s.p. wavefunctions and $\mathcal{D}$ for the damping operator in the accelerated gradient step (*PGR2all: Cross references to TDLDA section yet to be defined.*).

2. Absorbing bounds may have removed parts from the s.p. wavefunctions and so destroyed ortho-normalization. We transform the propagated density operator to a representation in terms of natural orbitals which is diagonal representation (17) with an ortho-normal set of s.p. wavefunctions together with corresponding occupation weights $\{\phi_\alpha^{(\mathrm{nat})}, W_\alpha^{(\mathrm{nat})}\}$. This step can be overridden if reflecting (or periodic) boundaries are used in which case TDLDA preserves ortho-normalization.
3. We compute density $\varrho(\mathbf{r}, t_1)$, current $\mathbf{j}(\mathbf{r}, t_1)$, and total energy E_{mf} associated to the TDLDA-propagated density matrix $\hat{\rho}_{\mathrm{mf}}$.
4. We determine the thermal mean-field equilibrium state $\hat{\rho}_{\mathrm{eq}}$ constrained to the given ϱ, $\mathbf{j}$, and E_{mf} from step 3. This is achieved by the Density-Constrained Mean eqs. (22) which is done iteratively as sketched in figure 2. The equilibrium state $\hat{\rho}_{\mathrm{eq}}$ is represented by new s.p. states $\{|\phi'_\alpha\rangle\}$ and new occupation numbers W'_α in diagonal form (17). Having these, we determine finally the excitation energy as energy relative to the temperature zero state

$$E^*_{\mathrm{intr}} = E_{\mathrm{sp}} - \sum_\alpha W_\alpha^{(T=0)} \langle \phi_\alpha^{(\mathrm{eq})} | \hat{h} | \phi_\alpha^{(\mathrm{eq})} \rangle \tag{24}$$

   where $W_\alpha^{(T=0)}$ are the occupation numbers determined with the s.p. energies $\langle \phi_\alpha^{(\mathrm{eq})} | \hat{h} | \phi_\alpha^{(\mathrm{eq})} \rangle$ at temperature $T = 0$. This E^*_{intr} thus measures the amount of thermal excitation energy in the system[7]
5. We compose the new density operator as mixture of TDLDA propagated state $\hat{\rho}_{\mathrm{mf}}$ and equilibration driving term $\hat{\rho}_{\mathrm{mf}} - \hat{\rho}_{\mathrm{eq}}$ with weight $\Delta t/\tau_{\mathrm{relax}}$ as

$$\hat{\rho}_{\mathrm{mix}} = \hat{\rho}_{\mathrm{mf}} - \frac{\Delta t}{\tau_{\mathrm{relax}}} \left[\hat{\rho}_{\mathrm{mf}} - \hat{\rho}_{eq}\right]$$

   where the relaxation time τ_{relax} requires the actual intrinsic excitation energy E^*_{intr} which is also obtained from DCMF. While evaluating the mixing, the new state is expressed in natural-orbital representation Eq. (17). This yields the final s.p. states $\{|\phi_\alpha(t_1)\rangle\}$ for this step and preliminary new occupations $\tilde{W}_\alpha$.
6. The mixing in step 5 may have slightly changed the energy such that we remain with a small energy mismatch as compared to the goal E_{mf}. We now apply a small iterative thermalization step to readjust the energy, as outlined in Appendix AppendixB. This then yields the final occupation weights $W_\alpha(t_1)$ which comply with energy conservation.

The above steps are sketched in Figure 1 whereby the step numbers here correspond to the ones in the Figure. The most involved part is step 4, the solution of the DCMF equations. It is expanded in details in figure 2. A word is in order about the termination criteria in DCMF iteration. The final check for total variance ϵ_0 is an input parameter `rtasumvar2max` to the RTA part, see section 5.1. The two other switch criteria in figure 2 are hardwired in `SUBROUTINE eqstate` in file `rta.F90` with the values $\epsilon_1 = 10^{-1}$ and $\epsilon_2 = 10^{-4}$*PGR2all: Switching DCMF iteration is more involved than that. There is also a parameter* `err`*. How is that defined?*

[7] Is this definition sufficient? Then we could skip planned appendix AppendixA.

We should turn ϵ_1 and ϵ_2 to input variables and also for `err`*.*
A similar problem exists for `ferm1`*. The strategy for the choice of the search interval* `[T0i,T1i]` *requires explanation which should come at that place here.* .

As said above, the time step δt for propagation of TDLDA is very small because it is limited from above by the maximal energy on the grid representation. The stepping Δt for the relaxation term needs only to resolve the changes in the actual mean field which allows much larger values. Typically, we can do 50–200 TDLDA steps before calling one RTA step. For detailed values see the examples delivered with the code.[8]

A word is in order about the system for which the present form of RTA can be used. The relaxation time $\tau_{\rm relax}$ is allowed to depend on time which allows to accommodate changes of the dynamical state. But $\tau_{\rm relax}$ is one global number chosen according to the average electron density $\bar{\varrho}$, see eq. (21e). This requires systems which can be characterized by such an average density, i.e., systems having only small density variations in the bulk as it holds typically for metallic bonds. The RTA rate is insensitive to many details of the microscopic collision term as energy- and angle-dependent scattering cross sections [27] or a broad spectrum of relaxation rates. However, these details are usually resolved only (if at all) for fast and energetic processes which are anyway deep in the regime of semi-classical VUU. The grossly averaged treatment of RTA is acceptable for not too fast and not too energetic processes, preferably in compact systems.

5. The structure of the RTA package in `rta.F90`

5.1. Input and output related to RTA

PGR2all: This subsection needs yet to be worked out in detail.

The `NAMELIST dynamic` contains the following variables used in the RTA procedure:[9]

`jrtaint`: Modulus for calling the RTA subroutine, i.e., nr. of TDLDA steps per one RTA step. Course time step Δt for RTA and fine time step for TDLDA `dt1` are related as $\Delta t =$`jrtaint`$*$`dt1`.

`rtamu`: Parameter μ in front of the quadratic density constraint in the DCMF Hamiltonian (22c).

`rtamuj`: Parameter μ_j in front of the quadratic current constraint in the DCMF Hamiltonian (22c).

`rtasumvar2max`: Termination criterion ϵ_0 in the RTA step as used in figure 2.

`rtaeps`: Step size δ in the damping operator *PGR2all: (cross ref to be defined)* $\mathcal{D}$ for the RTA step.

`rtae0dmp`: Energy offset $E_0 0$ in the damping operator *PGR2all: (cross ref to be defined)* $\mathcal{D}$ for the RTA step.

`rtasigee`: In medium e^--e^- cross section used for the relaxation time (21e).

[8] PGR2all: A link to be set once we have the place for the benchmarks.
[9] We should define a new `NAMELIST` and shift the RTA variables to there.

`rtars`: Effective Wigner-Seitz radius r_s used for the relaxation time (21e).

`rtatempinit`: The value `rtatempinit`/10 is used as lower value for the search of temperature in `SUBROUTINE ferm1`.

`rtaforcetemperature`: *PGR2all: Seems to be obsolete?*

Most observables were already defined at TDLDA stage and thus are returned in the standard output files as explained in the TDLDA section[10]. New output files specific to observables from RTA are:[11]

`prta`:

`peqstat`:

`pspeed`:

`prhov`:

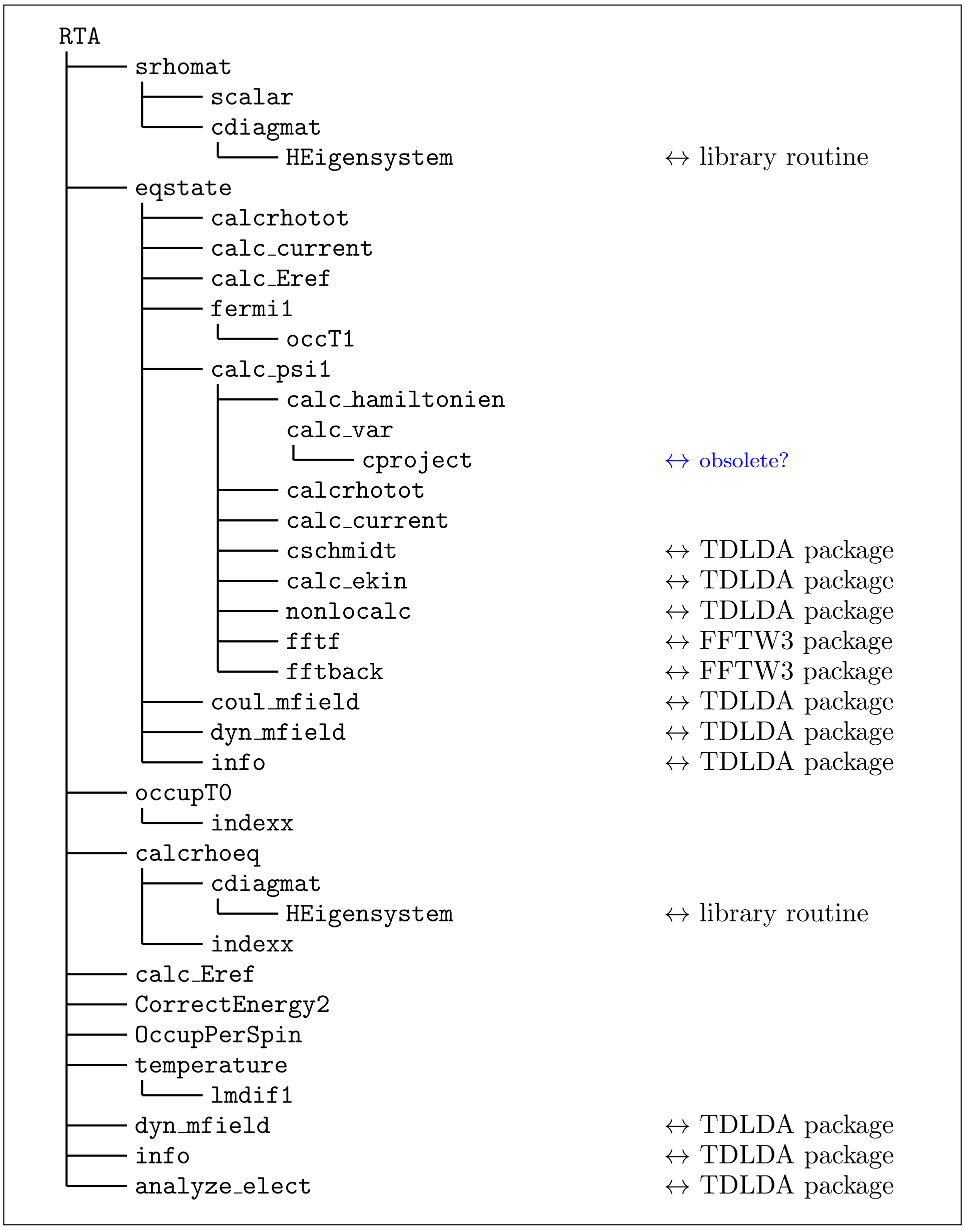
5.2. The calling tree

Here is an oversight over the tree structure of the RTA routines. Those subroutines contained in `rta.F90` are explained in detail in section 5.3. Subroutines coming from the TDLDA package or external sources are marked[12]

[10] Cross reference yet to be defined.

[11] These files seem to carry only protocol for numerics. Where do we print energy balance ($E^*_{\rm intr}$ etc)?

[12] The `HEigensystem` seems copied from some library. This could cause copyright problems if we publish the code. Is it from BLAS/LINPACK? Then we could replace the Fortran source by a library call.



```
RTA
├──── srhomat
│     ├──── scalar
│     └──── cdiagmat
│           └──── HEigensystem            ↔ library routine
├──── eqstate
│     ├──── calcrhotot
│     ├──── calc_current
│     ├──── calc_Eref
│     ├──── fermi1
│     │     └──── occT1
│     ├──── calc_psi1
│     │     ├──── calc_hamiltonien
│     │     │     calc_var
│     │     │     └──── cproject      ↔ obsolete?
│     │     ├──── calcrhotot
│     │     ├──── calc_current
│     │     ├──── cschmidt               ↔ TDLDA package
│     │     ├──── calc_ekin              ↔ TDLDA package
│     │     ├──── nonlocalc              ↔ TDLDA package
│     │     ├──── fftf                   ↔ FFTW3 package
│     │     └──── fftback                ↔ FFTW3 package
│     ├──── coul_mfield                  ↔ TDLDA package
│     ├──── dyn_mfield                   ↔ TDLDA package
│     └──── info                         ↔ TDLDA package
├──── occupT0
│     └──── indexx
├──── calcrhoeq
│     ├──── cdiagmat
│     │     └──── HEigensystem            ↔ library routine
│     └──── indexx
├──── calc_Eref
├──── CorrectEnergy2
├──── OccupPerSpin
├──── temperature
│     └──── lmdif1
├──── dyn_mfield                         ↔ TDLDA package
├──── info                               ↔ TDLDA package
└──── analyze_elect                      ↔ TDLDA package
```

5.3. The subroutines in detail

`SUBROUTINE rta(psi,aloc,rho,iterat)`

`iterat`	in	external iteration number (TDLDA time step)
`psi(1:kdfull2,1:kstate)`	in/out	set of s.p. wavefunctions
`rho(1:2*kdfull2)`	in/out	local densities for spin up and down
`aloc(1:2*kdfull2)`	in/out	local potentials for spin up and down

Basic RTA routine performing density constrained mean-field (DCMF) iterations, energy adjustment, admixing of local equilibrium states by calls to subroutines (see calling tree).

`SUBROUTINE calcrhoeq(psiorthloc,psieqloc,psiloc,occuporthloc,occuploc,nstateloc)`

`nstateloc`	in	number of s.p. states in spin block
`psiorthloc(1:kdfull2,1:nstateloc),`	in	set of TDLDA wavefunctions (natural orbitals)
`occuporthloc(1:nstateloc)`	in	occupations of TDLDA states
`psieqloc(1:kdfull2,1:nstateloc)`	in	set of local-equilibrium wavefunctions
`psiloc(1:kdfull2,1:nstateloc)`	out	set of final mixed wavefunctions
`occuploc(1:nstateloc)`	in/out	

Encapsulated in `SUBROUTINE rta`. Performs the mixing of TDLDA states with local-equilibrium state according to relaxation rate for one spin block. The mixed densty matrix is expanded in a representation by both sets of s.p. states.

`SUBROUTINE calc_Eref(occup,ispin,Ei,Eref)`

`occup(1:nstate)`	in	occupation number for s.p. states.
`ispin(1:nstate)`	in	spin assignement for s.p. states.
`Ei(1:nstate)`	in	spin assignement for s.p. states.
`Eref(1:2)`	out	sum of s.p. energies per spin.

Computes the weighted sum of s.p. energies as reference energy for DCMF. The sum is accumulated for each spin separately.

`SUBROUTINE fermi1(ekmod,eref,occup,ispinact,T0i,T1i,T2,mu)`

`ekmod(1:kstate)`	in	given s.p. energies, spin up block first, then spin down
`eref`	in	reference energy = wanted sum of s.p. energies
`ispinact`	in	spin for which routine is run
`T0i, T1i`	in	lower and upper temperature for search
`occup(1:kstate)`	in/out	occupation numbers, spin block-wise
`T2`	out	final temperature for which Fermi distribution matches `eref`
`mu`	out	final chemical potential

Determines thermal Fermi occupation such that given sum of s.p. energies `eref` and particle number is matched. Is done for each spin separately. Solution scheme is bracketing. Refers to `SUBROUTINE OccT1` while iterating temperatur `T2`.

PGR2all: Nr. of spin-up/spin-down states comes through `m_params`. *We should protocol all such entries. First step is to augment each* `USE` *by* `ONLY` *such that the explicitely communicated variables becomes visible. Important variables may then be listed explicitely.*

PGR2all: Routine requires that arrays are sorted in continuous blocks of spin. Do we have an initial check for that? And we need to address that in the general part which explains the layout of arrays.

`SUBROUTINE OccT1(occrefloc,enerloc,Etotloc,muloc,occtotloc,n,T,occuploc)`

`enerloc(1:n)`	in	s.p. energies for actual spin
`n`	in	number of s.p. states treated here
`T`	in	temperature
`occrefloc`	in	wanted total number of particles
`occuploc(1:n)`	out	thermal occupation numbers for given `T` and s.p. energies
`muloc`	out	chemical potential (Fermi energy)
`occtotloc`	out	final total number of particles
`Etotloc`	out	sum of s.p. energies

Determines by bracketing chemical potential `muloc` for given array of s.p. energies, temperature `T`, and wanted number of particles `occrefloc` with precision `1D-12`. Delivers with it thermal occupation numbers and corresponding total particle number and sum of s.p. energies.
PGR2all: This routine is specific to /tt SUBROUTINE ferm1. Could we encapsulate it by a /tt CONTAINS?

`SUBROUTINE Calc_psi1(psi1,aloc,rhotot0,rhototloc,curr0,curr1,j,lambda,mu, lambdaj,muj,sumvar2,eal,ekmod)`
combined with encapsulated `SUBROUTINE calc_hamiltonien`.

`j`	in	number of DCMF iteration, used here for print
`lambda(1:kdfull2,1:2)`	in	Lagrange parameter for density for spin up&down
`lambdaj(1:kdfull2,1:3)`	in	Lagrange parameter for current
`mu, muj`	in	driving parameter for augmented Lagrangian
`aloc(1:2*kdfull2)`	in	local potentials for spin up and down
`rhoto0(1:kdfull2,1:2)`	in	initial density *PGR2all: not used ??*
`curr0(1:kdfull2,1:3)`	in	wanted current
`psi1(1:kdfull2,1:kstate)`	in/out	set of s.p. wavefunctions iterated
`rhototloc(1:kdfull2,1:2)`	out	actual density according to `psi1`
`curr1(1:kdfull2,1:3)`	out	actual current from `psi1`
`ekmod(1:nstate)`	out	final s.p. energies
`eal`	out	final sum of s.p. energies
`sumvar2`	out	variance of s.p. energies

Performs one damped gradient step of with density & current constrained Hamiltonian.
PGR2all: The density array distinguishes spin up/down while the current array does not. Reason?
PGRcommThe IN & OUT assignments in this subroutine have to be updated.

`SUBROUTINE eqstate(psi,aloc,rho,psi1,occuporth,iterat)`

`iterat`	in	actual iteration number (for printing)
`psi(1:kdfull2,1:kstate)`	in	initial set of s.p. wavefunctions
`psi1(1:kdfull2,1:kstate)`	out	final set of s.p. wavefunctions
`aloc(1:2*kdfull2)`	in/out	local part of potential, spin up/down stacked in blocks
`rho(1:2*kdfull2)`	in	initial density, spin up/down stacked in blocks
`occuporth(1:kstate)`	in	occupation numbers for `psi` and still the same for `psi1`.

DCMF iterations by reapeatedly calling `Calc_psi1`, updating Lagrangian parameters for density & current constraints, and occassionally tuning temperature to achieve correct

energy. The latter is done by calling `fermi1`. The local potential is kept constant during DCMF iteration and updated only at the very end.
PGR2all: Fetches nr. of spin up/down from `m_params`*.*
PGR2all: Lagrange parameters are started from scratch. May it be faster to recycle the previous Lagrange parameters?
PGR2all: Density `rho` *is entered via list and still recomputed as* `rhotot0`*. Unnecessary doubling?*

`SUBROUTINE OccupT0(occloc,esploc,Estar)`

`esploc(1:nstate)`	in	given s.p. energies
`occloc(1:nstate)`	in	given occupation numbers
`Estar`	out	excitation energy relative to T=0 distribution

Computes thermal excitation energy as difference of actual energy to the energy obtained by Fermi distribution for $T=0$. The latter distributions is computed for the given s.p. energies which are the same as used for the thermal state.

`SUBROUTINE calcrhotot(rho,q0)`

`q0(1:kdfull2,1:kstate)`	in	set of s.p. wavefunctions for which density is accumulated
`rho(kdfull2,2)`	out	resulting density

Computes local density for set of wavefunctions `q0`. Note that two crucial information is communicated via module `params`, namely `occup`, the array of occupation numbers, `ispin` the array assigning spin top each s.p. state, and `nstate`, the number of s.p. states.
PGR2all: Exploiting the sorting of spin in blocks of s.p. states, we could rewrite the code with to `SUM` *statements.*

`SUBROUTINE calc_var(hpsi,psi1,sumvar2)`

`psi1(kdfull2,kstate)`	in	set of s.p. states for which variance of s.p. energies of calculated
`hpsi(kdfull2,kstate)`	in/out	array $H\rightarrow\psi_\alpha$, on input in k-space, on output in r-space
`sumvar2`	out	summed variance of s.p. energies

Computes the sum of variances of the s.p. energies, $\langle\Delta\hat{h}^2|rangle$.
PGRcommThe routine projects from each $hath\psi_\alpha$ all s.p. states ψ_β from the pool of states. That is too much. The s.p. variance should be $\sum_\alpha\langle|\psi_\alpha|(\hat{h}-\varepsilon_\alpha)^2|\psi_\alpha\rangle$ where $\varepsilon_\alpha=\langle|\psi_\alpha|\hat{h}|\psi_\alpha\rangle$.

`SUBROUTINE forceTemp(amoy,occup,n,temp,mu)`

`amoy(1:n)`	in	given s.p. energies
`occup(1:n)`	in	given thermal occupation
`n`	in	number of s.p. states
`temp`	in	temperature
`mu`	out	emerging chemical potential

Determines chemical potential for given s.p. energies and temperature by call to `OccT1`.
PGR2all: Obsolete and never used.

`SUBROUTINE fermi_init(ekmod,T,occup,ispinact)`

`ekmod(1:nstate)`	in	given s.p. energies
`T`	in	given temperature
`ispinact`	in	actual spin
`occup(1:kstate)`	in/out	initial occupation and resulting Fermi distribution for `T`.

Determines Fermi distribution for given s.p. energies and temperature. Searches appropriate chemical potential `mu` by bracketing. Use for repreated calls to `FUNCTION occ`.
PGR2all: This routine `fermi_init` and the related `FUNCTION occ` are never used, thus obsolete. May be removed.

`SUBROUTINE srhomat(psi,aloc,psiorth,occuporth)`

`psi(1:kdfull2,1:kstate)`	in	set of s.p. wavefunctions, not orth-normalized
`psiorth(1:kdfull2,1:kstate)`	out	ortho-normalized natural orbitals
`aloc(1:2*kdfull2)`	in	actual local potential
`occuporth(1:kstate)`	out	occupation numbers for ortho-normalized states

Computes the density matrix of initial state goiven by set of wavefunctions `psi` together with their occupations `occup`, the latter communicated through module `params`. Then diagonalizes the density matrix and computes on `psiorth` the new wavefunctions associated with diagonal representation of the density matrix.
Finally updates running transformation matrix `psitophi` which is communicated and stored through module `params`.
PGR2all: Usage and propagation of `psitophi` is somewhat hidden because it is handled through a module. Needs to be explained somwhere.

`SUBROUTINE scalar(tab1,tab2,scal,ispin, mess)`

`tab1(1:kdfull2,1:kstate)`	in	1. set of s.p. wavefunctions
`tab2(1:kdfull2,1:kstate)`	in	2. set of s.p. wavefunctions
`ispin(1:nstate)`	in	spin of s.p. states
`mess`	in	message for print inside routine
`scal(nstate,nstate)`	out	matrix of wavefunction overlaps

`SUBROUTINE cdiagspin(mat, eigen, vect, N)`

`mat(N,N)`	in	complex Hermitean matrix to be diagonalized
`N`	in	dimension of matrix
`eigen(N)`	out	resulting eigenbvalues
`Vect(N,N)`	out	resulting eigenstates

Driver routine for diagonalization of a complex Hermitean matrix of dimension `N` which consists in a two blocks for separate spin. Refers for each single block to routine `cdiag` and subsequent library routines contained therein.

`SUBROUTINE indexx (n,arrin,indx)`

`n`	in	length of array
`arrin(1:n)`	in	array to be sorted
`indx(1:n)`	out	pointer array

Evaluates sorting of an array in ascending order.

`SUBROUTINE occupPerSpin(mess,Occ)`

`mess`	in	character variable with comment printed inside routine
`Occ(1:2)`	out	total number of particles in each spin

Computes number of particles in each sin block. Uses `nstate` and occupations `occup` from module `params`.

`CorrectEnergy2(Wref,Eref,w,E,Wout,nloc)`

`W(1:nloc)`	in	initial occupations numbers
`E(1:nloc)`	in	given s.p. energies
`Wref`	in	reference particle number to be reached
`Eref`	in	reference sum of s.p. energies to be reached
`nloc`	in	actual number of states
`Wout(nloc)`	out	readjusted occupation numbers

Final energy correction by one step along Fermi distribution (using Taylor expansion about actual distribution), see appendix AppendixB).

`SUBROUTINE ordo_per_spin(psi)`

`psi(1:kdfull2,1:kstate)`	in/out	s.p. wavefunctions before and after reordering

Reorder states in two blocks of spin up and down. Applies that reshuffling to all relevant field of states, s.p. wavefunctions `psi`, spin per state `ispin`, and occupations `occup`. *PGR2all: Routine has been rendered obsolete by new initialization of states which produces immediately the correct sorting. But routine should be kept for possible later use (e.g., mixing states from different sources.*

`SUBROUTINE temperature(mu,T)`

`mu`	out	resulting chemical potential
`T`	out	resulting temperature

Takes s.p. energies `amoy` and occupations `occup` from module `params` and fits a Fermi distribution to it. Temperature and chemical potentials of the fitted distribution are returned via list. Calls a fitting routine `lmdif1` using subroutine `ff` as argument.

`SUBROUTINE ff(m,n,X,FVEC,IFLAG)`

`X(1:n)`	in	array handling chemical potential and temperature
`Fvec(1:m)`	out	array of mismatches of distributions
`n`	in	number of parameters of model, actually 2
`m`	in	number of entries in array
`iflag`	in	flaf possibly written (actually not used)

Mismatch of `occup` (via modules `params`) from Fermi distribution to given chemical potential and temperature. To be used in fitting routine `lmdef1`.

`SUBROUTINE cproject(qin,qout,ispact,q0)`

`qin(1:kdfull2)`	in	s.p. wvaefunction to be projected
`q0(1:kdfull2,1:kstate)`	in	set of s.p. wavefunctions which is projected out from `qin`
`ispact`	in	spin associated with `qin`
`qout(1:kdfull2)`	out	projected s.p. wavefunction

Projects away from `qin` all contributions of the set `q0`.
PGR2all: This routine may become obsolete if we recode the the variance in routine `calc_var` *to meet the standard definition.*

AppendixA. The intrinsic excitation energy

PGR2all: Intrinsic excitation energy yet to be explained. May also become part of DCMF explanation.

AppendixB. Iterative correction of total energy

PGR2all: Yet to be imported.

[1] J. P. Perdew, Y. Wang, Phys. Rev. B 45 (1992) 13244.

[2] O. Gunnarsson, B. I. Lundqvist, Exchange and correlation in atoms, molecules and solids by the spin density functional formalism, Phys. Rev. B 13 (1976) 4274.

[3] S. Kümmel, M. Brack, P.-G. Reinhard, Eur. Phys. J. D 9 (1999) 149.

[4] S. Goedecker, M. Teter, J. Hutter, Separable dual-space gaussian pseudopotentials, Phys. Rev. B 54 (1996) 1703.

[5] N. W. Ashcroft, N. D. Mermin, Solid State Physics, Saunders College, Philadelphia, 1976.

[6] A. Rubio, L. C. Balbas, J. A. Alonso, Z. Phys. D 19 (1991) 93.

[7] B. Montag, T. Hirschmann, J. Meyer, P.-G. Reinhard, M. Brack, Shape isomerism in sodium clusters with $10 \leq z \leq 44$: Jellium model with quadrupole, octupole and hexadecapole deformations, Phys. Rev. B 52 (1995) 4775.

[8] M. Brack, Rev. Mod. Phys. 65 (1993) 677.

[9] K. Clemenger, Ellipsoidal shell structure in free metal clusters, Phys. Rev. B 32 (1985) 1359.

[10] S. G. Nilsson, K. Dan. Vidensk. Selsk. Mat. Fys. Medd. 29 (1955) No. 16.

[11] W. A. de Heer, The physics of simple metal clusters: experimental aspects and simple models, Rev. Mod. Phys. 65 (1993) 611.

[12] J. D. Jackson, Classical Electrodynamics, Wiley, New York, 1962.

[13] A. Messiah, Quantum Mechanics, Dover, New York, 1995.

[14] B. H. Bransden, M. R. C. McDowell, Charge Exchange and Theory of Ion-Atom Collisions, Clarendon, Oxford, 1992.

[15] P.-G. Reinhard, E. Suraud, C. A. Ullrich, Eur. Phys. J D 1 (1998) 303.

[16] W. Kohn, L. J. Sham, Phys. Rev. 140 (1965) 1133.

[17] G. F. Bertsch, S. Das Gupta, Phys. Rep. 160 (1988) 190.

[18] Y. Abe, S. Ayik, P.-G. Reinhard, E. Suraud, Phys. Rep. 275 (1996) 49.

[19] H. Reinhardt, P.-G. Reinhard, K. Goeke, Phys. Lett. B 151 (1985) 177.

[20] K. Goeke, P.-G. Reinhard, H. Reinhardt, Ann. Phys. (N.Y.) 166 (1986) 257.

[21] D. Pines, P. Nozières, The Theory of Quantum Liquids, W A Benjamin, New York, 1966.

[22] P.-G. Reinhard, E. Suraud, A quantum relaxation-time approximation for finite fermion systems, Ann. Phys. (N.Y.) 354 (2015) 183.
URL `http://dx.doi.org/10.1016/j.aop.2014.12.01`

[23] K. Gütter, P.-G. Reinhard, C. Toepffer, Phys. Rev. A 38 (1988) 1641.

[24] L. E. Reichl, A Modern Course in Statistical Physics, Wiley, New York, 1998.

[25] F. Calvayrac, P.-G. Reinhard, E. Suraud, C. A. Ullrich, Phys. Rep. 337 (2000) 493.

[26] P.-G. Reinhard, E. Suraud, Introduction to Cluster Dynamics, Wiley, New York, 2004.

[27] E. Giglio, P.-G. Reinhard, E. Suraud, Phys. Rev. A 67 (2003) 043202.