
SAX User's Guide



Version 1.0.0

AN INTEGRATED APPROACH TO
AB INITIO ELECTRONIC EXCITATIONS
WITHIN
GW APPROXIMATION

November 13, 2007

(<http://www.sax-project.org>)

This *User's Guide* covers the installation and usage of SAX (Self-energies and eXcitations), version 1.0.0. It includes a description of the capabilities of the package, how to use these capabilities, the necessary input files and formats, and how to run the programs contained in the package on both serial and parallel machines.

ABOUT SAX is an open-source GNU General Public License package for the calculation of electronic and optical properties, developed at the national Research Center on nanoStructures and bioSystems at Surfaces (S3) of the Italian INFN-CNR (<http://www.s3.infn.it>) under the coordination of Layla Martin-Samos and Giovanni Bussi. SAX works in the framework of many-body perturbation theory in the GW approximation. This is the state-of-the-art method to describe electronic band structures and excitonic effects beyond the density-functional theory (DFT), and in the last years it has proved to be a successful approach for a wide range of materials. SAX implements the GW approximation and the Bethe-Salpeter Equation using periodic cells, plane-waves and pseudopotentials, and the various matrix elements are calculated on real-space grids through fast-Fourier transforms. Depending on the available computational resources, SAX can be used to study systems with large cells (up to 1000 Å³) and large number of atoms (up to 100).

The SAX package is written in Fortran90 and designed in a modern way, taking advantage of an object-oriented structure. The low-level operations are hidden inside object-libraries which define plane-waves grids, wavefunctions, etc. The modularity of the code simplifies greatly the higher level implementation. For this reason, SaX is a perfect tool for methodological research. On the other side, the efficiency of the code is tight to the efficiency of the low-level routines, which can be optimized (and are already fairly optimized) independently.

The SAX package operates, in principles, as a post-processing of any DFT electronic structure code. In the present version, the user will find a wrapper to run SAX from the results of a DFT calculation done using PWSCF code (<http://www.pwscf.org>) included in the quantum-espresso distribution (<http://www.quantum-espresso.org>).

SaX calculations will provide the user with:

- Full dynamical structure of the random-phase-approximation polarizability.
- Quasiparticle energies and wavefunctions.
- Different implementations of GW scheme.
- Different degrees of self-consistency in the GW scheme.
- Excitonic effects with the Bethe-Salpeter equation.

Current limitations (and planned developments):

- Metallic systems can not be treated properly.
- Symmetries are not taken into account.

CONTACTS The web site for SaX is:

<http://www.sax-project.org/> <http://www.sax-project.org/>

Releases and patches of SaX can be downloaded from this site or following the links contained on it.

If you specifically need to contact the developers of SaX (installation, usage, bug's report), write to saxteam@unimore.it <mailto:saxteam@unimore.it>.

Other pointers:

S3: <http://www.s3.infn.it/> <http://www.s3.infn.it/>

INFN: <http://www.infn.it/> <http://www.infn.it/>

CINECA: <http://www.cineca.it/> <http://www.cineca.it/>

Quantum-Espresso: <http://www.quantum-espresso.org/> <http://www.quantum-espresso.org/>

TERMS OF USE SaX is free software, released under the GNU General Public License (the file `License` in the distribution). Although users are not under any obligation in the spirit of the GNU General Public License, the SaX team would appreciate the acknowledgement of the effort to produce such codes in the form of the following reference:

in the text: "The results of this work have been obtained using the SaX package [...]"

in the references: "[...] L. Martin-Samos, G. Bussi *et al* <http://www.sax-project.org/>."

All trademarks mentioned in this guide belong to their respective owners.

DISCLAIMER While the developers of SaX make every effort to deliver a high quality scientific software, we do not guarantee that our codes are free from defects. Our software is provided "as is". Users are solely responsible for determining the appropriateness of using this package and assume all risks associated with the use of it, including but not limited to the risks of program errors, damage to or loss of data, programs or equipment, and unavailability or interruption of operations. Due to the limited human resources involved in the development of this software package, only a limited support will be given to individual users for either installation or execution of the codes. Finally, in the spirit of every open source project, any contribution from external users is welcome, encouraged and, if appropriate, it will be included in future releases.

LICENCE All the material included in this distribution is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. These programs are distributed in the hope that they will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

CREDITS SAX is a project of the National Research Center on nanoStructures and bioSystems at Surfaces (S3) of the Italian INFN-CNR (<http://www.s3.infn.it/>), based in Modena, Italy.

The computer time and specific technical support is partly provided by CINECA through a S3-CINECA agreement.

The scientific board of the SAX project is composed by Layla Martin-Samos, Giovanni Bussi, Alice Ruini, Marilia J. Caldas and Elisa Molinari.

The present release of SAX has been realized by Layla Martin-Samos and Giovanni Bussi. A list of contributors include Chiara Campani (implementation of `memory.x`).

For a full list of developers and contributors see the file `sax/CREDITS`

ACKNOWLEDGEMENTS The SAX team want to thank Carlo Cavazzoni (CINECA) for his deep and continuative consulence about numerical techniques; Andrea Benassi for the SaX

web page and its suggestions for code improvements; Andrea Ferretti, Eleonora Luppi and Daniele Varsano for useful discussions; People from PWSCF for helping us to keep our project compatible with PWSCF; People from WanT for helping us to keep our project compatible with WanT.

Contents

1	Theoretical background	6
1.1	The GW method	6
1.2	Inclusion of excitonic effects, the Bethe-Salpeter equation	8
1.3	In practice	12
1.3.1	The plasmon-pole model	14
1.3.2	Divergencies in the Coulomb potential	15
1.3.3	Analytic treatment of the dielectric matrix	17
1.3.4	The Bethe-Salpeter Equation	18
2	Getting Started with SAX	19
2.1	Description of the package	19
2.2	Installation	20
2.3	Configure	21
2.4	Compile	22
2.5	DFT preliminary step	23
2.6	Run examples	23
3	Using SaX	26
3.1	Input data	26
3.2	gw.x	27
3.2.1	Calculation options	27
3.2.2	Typical cases	28
3.3	bse.x	30
3.3.1	Calculation options	30
3.3.2	Typical cases	31
3.4	spectra.x	31
3.4.1	Calculation options	31
3.4.2	Typical cases	32
3.5	pptools.x	33
3.5.1	Calculation options	33
3.6	memory.x	34
3.6.1	Additional Calculation options	34

1 Theoretical background

The GW approximation for the electronic self-energy has its roots in the work by Hedin [1, 2]. In its first formulation, an approximated form for the electron self-energy is obtained through an expansion in terms of the screened Coulomb potential. This procedure can be interpreted as a generalization of the Hartree-Fock method, where the expansion is based on the bare Coulomb potential. The effect of the screening is particularly important in semiconductors and in metals, where the Hartree-Fock approach is completely unsuitable. In more recent times, see among others Refs. [3, 4, 5, 6, 7, 8, 9, 10], the GW method has been used on top of density-functional theory (DFT) calculations [11, 12], to improve the agreement of electronic band structures with experimental results. This combined approach is presently the state-of-the-art technique for the *ab initio* prediction of electronic properties. Moreover, the excitonic effects are naturally included using the so-called Bethe-Salpeter equation (BSE), which is a two-particle equation for an excited electron-hole pair interacting through a screened Coulomb potential treated within the Random Phase Approximation (RPA). The effect of excitons is particularly important in insulators and semi-conductors where, due to the small screening, the electron-hole interaction is strong. For a review on the applications of these methods the reader is invited to refer to Refs. [13] and [14].

1.1 The GW method

In the general framework of many-body perturbation theory the one particle Green's function is defined as:

$$G(xtx't') = -i\langle N|T[\Psi(xt)\Psi^\dagger(x't')]|N\rangle, \quad (1)$$

where $\langle N|$ is the many-body ground state of the system with N electrons, $\Psi^\dagger(xt)$ and $\Psi(x't')$ are the Heisenberg operators for creation and annihilation, respectively, and T indicates the time-ordered product [15]. When a GW calculation is performed on top of a DFT calculation, a reasonable initial guess for the frequency representation of the single particle Green's function can be written as

$$G^{KS}(xx';\omega) = \sum_{nk} \frac{\psi_{nk}(x)\psi_{nk}^*(x')}{\omega - \epsilon_{nk} - i\eta\text{sign}(\mu - \epsilon_{nk})} \quad (2)$$

where $\psi_{nk}(x)$ is the Kohn-Sham orbital corresponding to the band n and the k crystal momentum, ϵ_{nk} its Kohn-Sham energy, and μ the Fermi energy. η here is a positive infinitesimal number, necessary to obtain the correct time-ordering in the Green's function, and the sign function returns +1 (-1) when the argument is positive (negative). We consider here a system of spin-compensated fermions, completely neglecting the spin dependence of the operators. The irreducible polarizability is the direct product of two Green's functions in real-space and real-time, $-iG(xt, x't')G(x't', xt)$. This direct product in time is written as a convolution when expressed in frequency domain. Using equation (2), the polarizability takes the following form:

$$P(xx';\omega) = 2 \sum_{nkn'k'} \psi_{nk}(x)\psi_{nk}^*(x')\psi_{n'k'}(x')\psi_{n'k'}^*(x) \times \left[\frac{\theta(\mu - \epsilon_{nk})\theta(\epsilon_{n'k'} - \mu)}{(\omega - (\epsilon_{nk} - \epsilon_{n'k'}) - i\eta)} - \frac{\theta(\epsilon_{nk} - \mu)\theta(\mu - \epsilon_{n'k'})}{(\omega - (\epsilon_{nk} - \epsilon_{n'k'}) + i\eta)} \right] \quad (3)$$

where the factor 2 comes from the spin degeneracy. The θ function returns 1 for a positive argument and 0 for a negative argument. The random-phase approximation for the screened potential is

obtained by solving the Dyson-like equation:

$$W(xx'; \omega) = v_C(xx') + \int dx'' dx''' dx'''' v_C(xx'') P(x''' x''''; \omega) W(x'''' x'; \omega) \quad (4)$$

where v_C is the Coulomb potential. We split the screened potential as a bare potential (v_C) plus a polarization potential (W^P), defined as $W^P = W - v_C$. In the GW approximation the self-energy is a simple product of the Green's function and the screened potential:

$$\Sigma = iW(xt^+ x't')G(xt, x't'). \quad (5)$$

The self-energy is naturally split in an exchange part (Σ_x) and a correlation part (Σ_c). The exchange is static and is calculated in the same way as in the Hartree-Fock method:

$$\Sigma_x(xx') = - \sum_{vk} \theta(\mu - \epsilon_{vk}) \psi_{vk}(x) \psi_{vk}^*(x') v_C(xx') \quad (6)$$

The correlation part is a convolution in frequency domain and takes, in the DFT-GW scheme, the following form:

$$\Sigma_c(xx'; \omega) = - \sum_{nk} \psi_{nk}(x) \psi_{nk}^*(x') \frac{1}{2\pi i} \int d\omega' \frac{W^P(xx'; \omega')}{\omega + \omega' - \epsilon_{nk} - i\eta \text{sign}(\mu - \epsilon_{nk})} \quad (7)$$

This self-energy operator can be used to calculate the energy of the quasi-particle states. In the quasi-particle picture, an effective quasi-particle Hamiltonian can be defined as a sum of the single-electron Hamiltonian, the Hartree potential, the exchange self-energy and the correlation self-energy:

$$H_{QP}(xx'; \omega) = h(xx') + V_H(x)\delta(xx') + \Sigma_x(xx') + \Sigma_c(xx'; \omega) \quad (8)$$

This Hamiltonian is energy-dependent, and, in general, non Hermitian.

The evaluation of the integral in Eq. (7) is still nowadays numerically unaffordable for systems with more than very few electrons. As a first crude approximation $W^P(xx'; \omega)$ might be set to zero. In this case, just the exchange part of the self-energy contributes, leading to the so called Hartree-Fock self-energy. This scheme fails when the polarization effects are not negligible, i.e. for metals and semiconductors. A better choice is to assume a static effective screening, i.e. $W^P(xx'; \omega) = W^P(xx')$. Different models for the evaluation of this effective screening have been devised. The easiest is the statically screened Hartree-Fock (SSHf)[6], where the zero-frequency screen is used, $W^P(xx'; \omega) = W^P(xx'; \omega = 0)$. Thus, Σ_c takes the following form:

$$\Sigma_c(xx') = - \sum_{vk} \theta(\mu - \epsilon_{vk}) \psi_{vk}(x) \psi_{vk}^*(x') W^P(xx'; 0), \quad (9)$$

where only valence bands contribute to the electron self-energy. Another possibility is the Coulomb-hole plus screened exchange (COH-SEX) approximation [2, 6], where

$$\Sigma_c(xx') = -\frac{1}{2} \sum_{nk} [\theta(\mu - \epsilon_{nk}) - \theta(\epsilon_{nk} - \mu)] \psi_{nk}(x) \psi_{nk}^*(x') W^P(xx'; 0). \quad (10)$$

With respect to SSHf, the self-energy in the COH-SEX approximation has an additional factor ($-\frac{1}{2}W^P(xx'; 0)$) which represent the so-called Coulomb-hole correction, for more details see Refs. [2,

16]. However, when the dynamical behavior of $W^p(xx'; \omega)$ is non-trivial, these approximations are going to fail.

The use of an explicitly dynamical screening is non-trivial. An intermediate possibility is to assume a dynamical screening but a static self-energy. Since the quasiparticle concept is meaningful only for the states close to the Fermi energy, the dynamic self-energy operator can be approximated with a static operator calculated for a frequency equal to the Fermi energy, i.e. assuming $H_{QP}(\omega) = H_{QP}(\mu)$. We recall that, at the Fermi energy, the self-energy operator is Hermitian. Thus, within this approximation the quasiparticle Hamiltonian is static and Hermitian. It is possible to show [5] that the integral in (7) can be deformed as an integral on the imaginary axis plus a somme on pole contributions arising from the poles of the Green function near the real axis. But, if the self-energy is evaluated at the Fermi energy there is no pole contributions and the integral in (7) becomes:

$$\Sigma_c(xx'; \mu) = - \sum_{nk} \psi_{nk}(x) \psi_{nk}^*(x') \frac{1}{2\pi} \int_0^{+\infty} ds W^p(xx'; is) \frac{2(\mu - \epsilon_{nk})}{(\mu - \epsilon_{nk})^2 + s^2}. \quad (11)$$

This contour deformation is possible as long as $\Re\omega \in [\epsilon_H - \epsilon_g, \epsilon_L + \epsilon_g]$, where ϵ_H is the energy of the highest valence state, ϵ_L is the energy of the lowest conduction state, and $\epsilon_g = \epsilon_L - \epsilon_H$ is the electronic gap. The strong advantage of this procedure is that the behavior of W on the imaginary ω axis is much smoother than on the real axis.

We also notice that W for imaginary frequency is obtained from the analytic continuation of P , simply using an imaginary ω in Eq. 3 and taking the limit $\eta \rightarrow 0$. With this procedure, the η parameter is completely eliminated from the actual calculation. This approach is equivalent to [6] as a zero order development. Our test calculations on this method show that it seems to work only in very particular cases. Looking at equation (11), it is easy to show that one of the requirement is that the self-energy should be a constant around the fermi energy. The main contribution to self-energy expectation values differences should come from the differences in the overlap integrals between bands. For all the details concerning analytic continuations of time-ordered functions, the reader can refer to Ref. [16]. The integral in (7) can be performed numerically, i.e. computing W on a grid of imaginary frequencies, or using some model for its frequency-dependency. A common choice is the use of plasmon pole models [17, 18, 19], see section 1.3.1.

1.2 Inclusion of excitonic effects, the Bethe-Salpeter equation

The key words for the inclusion of excitonic effects within the GW-Bethe-Salpeter scheme implemented in SAX are the two-particle Green's function (G_2), the four point reducible polarizability (Π), the resonant approximation for the two particle hamiltonian and the static approximation for the screened Coulomb potential (W). In theory, BSE should be used on top of a GW calculation.

Let first assume that the excited state can be obtained by a coherent linear superposition of vertical single-pair excitations plus corrections of higher order:

$$|E\rangle = \sum_{nn'k} A_{nn'k} \hat{a}_{nk}^\dagger \hat{a}_{n'k} |0\rangle + |C\rangle, \quad (12)$$

where $|C\rangle$ represents higher order corrections.

$|E\rangle$ is an eigenstate of an "unknown" excitonic Hamiltonian and the excitonic energy Ω can therefore be calculated as:

$$\hat{H}|E\rangle = \Omega|E\rangle. \quad (13)$$

A possibility for calculating the excitonic energy could be to develop an equation in the coefficient A_{cvk} and energy Ω .

Considering only the linear part of $|E\rangle$ in the single-particle transition, it is also possible to write $|E\rangle$ on the basis of single-particle orbitals:

$$\Psi(xx') = \sum_{nn'k} \psi_{nk}^{qp}(x) \psi_{n'k}^{qp*}(x'). \quad (14)$$

The excitation energies are the solutions of an eigenvalue problem and it is necessary to find a form of the effective two-particle hamiltonian of (13). In the context of neutral excitations, this effective two-particle Hamiltonian is called the Bethe-Salpeter Equation, where these excitations are described in terms of a bound electron-hole pair.

The information on the two-particle excitations of a system is contained in the two-particle Green's function [15]:

$$G_2(x_1t_1, x_2t_2, x_3t_3, x_4t_4) = (-i)^2 \langle N|T [\Psi(x_1t_1)\Psi(x_2t_2)\Psi^\dagger(x_3t_3)\Psi^\dagger(x_4t_4)] |N\rangle, \quad (15)$$

By contraction of indexes the two-particle Green's function is written as:

$$\begin{aligned} G_2(xt, x't', xt^+, x't'^+) &= -\langle N|T [\Psi(xt)\Psi(x't')\Psi^\dagger(xt^+)\Psi^\dagger(x't'^+)] |N\rangle = \\ &\left\{ \begin{array}{l} \langle N| [\Psi^\dagger(xt^+)\Psi(xt)\Psi^\dagger(x't'^+)\Psi(x't')] |N\rangle \quad t > t' \\ \langle N| [\Psi^\dagger(x't'^+)\Psi(x't')\Psi^\dagger(xt^+)\Psi(xt)] |N\rangle \quad t' > t \end{array} \right\} = \\ &\langle N|T [\Psi^\dagger(xt^+)\Psi(xt)\Psi^\dagger(x't'^+)\Psi(x't')] |N\rangle. \end{aligned}$$

G_2 describes the propagation of an electron-hole pair created at some point x and t (x' and t') and $|N\rangle$ is the ground state of the N particle system.

It is convenient to introduce the polarization propagator Π for the electron-hole pair defined as:

$$i\Pi(xt, x't') = \langle N|T [\Psi^\dagger(xt^+)\Psi(xt)\Psi^\dagger(x't'^+)\Psi(x't')] |N\rangle. \quad (16)$$

The polarization propagator in its Lehman representation takes the following form:

$$\Pi_{\lambda\mu, \alpha\beta}(\omega) = \sum_n \left(\frac{\langle N|c_\mu^\dagger c_\lambda|\Psi_n\rangle \langle \Psi_n|c_\alpha^\dagger c_\beta|N\rangle}{\omega - (E_n - E_0) + i\eta} - \frac{\langle N|c_\alpha^\dagger c_\beta|\Psi_n\rangle \langle \Psi_n|c_\mu^\dagger c_\lambda|N\rangle}{\omega + (E_n - E_0) - i\eta} \right) \quad (17)$$

where $|\Psi_n\rangle$ is an intermediate state. The poles of the polarization propagator determine the excitations (resonant term) and de-excitations energies (antiresonant term), similar to the polarizability in (3).

The polarization propagator satisfies a Dyson-like equation:

$$\Pi_{\lambda\mu, \alpha\beta}(\omega) = \Pi_{\lambda\mu, \alpha\beta}^0(\omega) + \sum_{\eta\nu, \rho\sigma} \Pi_{\lambda\mu, \rho\nu}^0(\omega) K_{\rho\nu, \eta\sigma}(\omega) \Pi_{\eta\sigma, \alpha\beta}(\omega). \quad (18)$$

This form of the equation for the two-particle polarizability suggests an inversion procedure to

be the natural algorithm, but this requires to perform an explicit inversion for each value of the frequency ω .

An alternative formulation of the propagation of an electron-hole pair can be found starting from Hedin's equations and defining four-point quantities. In this formulation the four-point Coulomb interaction takes the following form:

$$v(1, 1', 2, 2') = v(1, 2)\delta(1, 1')\delta(2, 2'), \quad (19)$$

similar to the four-point screened interaction:

$$W(1, 1'; 2, 2') = W(1, 1')\delta(1, 2)\delta(1', 2'), \quad (20)$$

and the four-point independent particle polarizability:

$$P^0(1, 1'; 2, 2') = -iG(1', 2')G(2, 1), \quad (21)$$

where we have used an abbreviated notation (1 stays for $(x_1 t_1)$ etc) It is possible, then, to obtain a generalized four-point irreducible polarizability:

$$P(1, 1', 2, 2') = P_0(1, 1', 2, 2') + - \int P_0(1, 1', 3, 3')W(3, 3', 4, 4')P(4, 4', 2, 2') d3 d3' d4 d4', \quad (22)$$

where only the attractive screened interaction W appears.

With the same considerations, we can obtain a generalized four-point form for the modified response function and the modified polarizability:

$$\chi(1, 1', 2, 2') = P_0(1, 1', 2, 2') + \int P_0(1, 1', 3, 3') \times [v(3, 3', 4, 4' - W(3, 3', 4, 4'))] P(4, 4', 2, 2') d3 d3' d4 d4', \quad (23)$$

$$\bar{P}(1, 1', 2, 2') = P_0(1, 1', 2, 2') + \int P_0(1, 1', 3, 3') \times [\bar{v}(3, 3', 4, 4' - W(3, 3', 4, 4'))] \bar{P}(4, 4', 2, 2') d3 d3' d4 d4'. \quad (24)$$

The only difference between (23) and (24) is the long range tail of the Coulomb potential which is responsible for the local-field effects [14]. Equation (24) for the modified polarizability has a Dyson-like form, in analogy with (18) for the polarizability propagator, just considering the lowest order of the interaction kernel.

Taking into account the translational invariance in time, it is possible to Fourier transform into the frequency space:

$$\begin{aligned} \chi(1, 1', 2, 2') &= \chi(x_1 t_1, x_1' t_1^+, x_2 t_2, x_2' t_2^+) = \chi(x_1, x_1', x_2, x_2', t_2 - t_1) \\ &\rightarrow \chi(x_1, x_1', x_2, x_2', \omega) \delta(x_1 - x_1') \delta(x_2 - x_2'). \end{aligned} \quad (25)$$

An approximation commonly used and currently implemented in SAX , concerns the energy dependence of the screened potential: this is neglected and one assumes $W(\omega) = W(\omega = 0)$ [20].

This equation for the four-point polarizability can be solved in principle inverting the matrix for

each possible frequency ω . However, for practical purpose, it is computationally convenient to reformulate the problem as an effective eigenvalue problem, as also suggested by the initial physical picture of interacting electron-hole pairs. The first step is to change the basis set to describe the quantity we are dealing with, considering as a new basis single-particle eigenfunctions $\psi_n(x)$ (the subindex n has to be read as $n = (n, k)$, that it contains the information of the band and the k point).

In this basis any four point quantity is:

$$S(x_1, x_{1'}, x_2, x_{2'}) = \sum_{(n_1, n_{1'})(n_2, n_{2'})} \psi_{n_1}^*(x_1) \psi_{n_{1'}}(x_{1'}) \psi_{n_2}(x_2) \phi_{n_{2'}}^*(x_{2'}) S_{(n_1, n_{1'})(n_2, n_{2'})}. \quad (26)$$

For instance, the modified polarizability now becomes:

$$\bar{P}_{(n_1, n_{1'})(n_2, n_{2'})} = \bar{P}_{(n_1, n_{2'})(n_2, n_{2'})}^0 + \bar{P}_{(n_1, n_{1'})(n_3, n_{3'})}^0 \Xi_{(n_3, n_{3'})(n_4, n_{4'})} \bar{P}_{(n_4, n_{4'})(n_2, n_{2'})}. \quad (27)$$

Here the interaction kernel is defined as:

$$\begin{aligned} \Xi_{(n_1, n_{1'})(n_2, n_{2'})} = & \\ & - \int dx_1 dx_{1'} \psi_{n_1}(x_1) \psi_{n_{1'}}^*(x_{1'}) W(x_1, x_{1'}) \psi_{n_2}^*(x_1) \psi_{n_{2'}}(x_{1'}) + \\ & + \int dx_1 dx_{1'} \psi_{n_1}(x_1) \psi_{n_{1'}}^*(x_1) v(x_1, x_{1'}) \psi_{n_2}^*(x_{1'}) \psi_{n_{2'}}(x_{1'}), \end{aligned} \quad (28)$$

where the static approximation for the screening has been assumed.

After some manipulations (see reference [14]), the equivalent eigenvalue problem becomes:

$$\sum_{(n_1, n_{1'})(n_2, n_{2'})} H_{(n_1, n_{1'})(n_2, n_{2'})}^{2p} A_{(n_2, n_{2'})}^\mu = E^\mu A_{(n_1, n_{1'})}^\mu, \quad (29)$$

where $H_{(n_1, n_{1'})(n_2, n_{2'})}^{exc}$ is the two particle hamiltonian defined as:

$$H_{(n_1, n_{1'})(n_2, n_{2'})}^{2p} = (\epsilon_{n_{1'}} - \epsilon_{n_1}) \delta_{(n_1, n_2)(n_{1'}, n_{2'})} + (f_{n_1} - f_{n_{1'}}) \Xi_{(n_1, n_{1'})(n_2, n_{2'})}, \quad (30)$$

where $\epsilon_{n_{1'}}$ and ϵ_{n_1} are single-particle energy eigen-values and $\Xi_{(n_1, n_{1'})(n_2, n_{2'})}$ is the interaction kernel. In general, the eigen-states A^μ are not orthogonal.

In the particular case of semi-conductors (or insulators), this expression can be simplified analyzing the form of the effective two particle Hamiltonian $H_{(n_1, n_{1'})(n_2, n_{2'})}^{2p}$. In the definition of $H_{(n_1, n_{1'})(n_2, n_{2'})}^{2p}$ there are $\delta_{(n_1, n_2)(n_{1'}, n_{2'})}$, that represent the identity in the transition basis, and $(f_{n_1} - f_{n_{1'}})$, that is different from zero only for those transitions for which $(n_1, n_{1'}) = (\text{occupied}, \text{unoccupied})$ or $(n_1, n_{1'}) = (\text{unoccupied}, \text{occupied})$.

In what follows we will use this notation:

- $(n_1, n_{1'})$:
 - unoccupied state: $v_1, v_{1'}$
 - occupied state: $c_1, c_{1'}$

- $(n_2, n_{2'})$:
 - unoccupied state: $v_2, v_{2'}$
 - occupied state : $c_2, c_{2'}$

The v for unoccupied states, stays for the hole created in the valence band when the electron has been promoted to the conduction band c . The matrix representation of the two-body Hamiltonian $H_{(n_1, n_{1'})(n_2, n_{2'})}^{2p}$ is:

$$\left(\begin{array}{c|cccc} & v_2, c_2 & c_2, v_2 & v_2, v_{2'} & c_2, c_{2'} \\ \hline v_1, c_1 & H_{(v_1 c_1)(v_2 c_2)}^{2p} & \Xi_{(v_1 c_1)(c_2 v_2)} & \Xi_{(v_1 c_1)(v_2 v_{2'})} & \Xi_{(v_1 c_1)(c_2 c_{2'})} \\ c_1, v_1 & -\Xi_{(c_1 v_1)(v_2 c_2)} & -H_{(c_1 v_1)(c_2 v_2)}^{2p} & -\Xi_{(c_1 v_1)(v_2 v_{2'})} & -\Xi_{(c_1 v_1)(c_2 c_{2'})} \\ v_1, v_{1'} & 0 & 0 & (E_{v_{1'}} - E_{v_1})\delta_{(v_1, v_2)(v_{1'}, v_{2'})} & 0 \\ c_1, c_{1'} & 0 & 0 & 0 & (E_{c_{1'}} - E_{c_1})\delta_{(c_1, c_2)(c_{1'}, c_{2'})} \end{array} \right).$$

In a "block" representation one obtain the following matrix for H^{2p} :

$$\left(\begin{array}{cc} H_{(v_1 c_1)(v_2 c_2)}^{2p \text{ resonant}} & \Xi_{(v_1 c_1)(c_2 v_2)}^{coupling} \\ -[\Xi_{(v_1 c_1)(c_2 v_2)}^{coupling}]^* & -[H_{(v_1 c_1)(v_2 c_2)}^{2p \text{ resonant}}]^* \end{array} \right)$$

$H^{2p \text{ resonant}}$ is Hermitian and corresponds to positive absorption energy, while $-[H^{2p \text{ resonant}}]^*$ is the anti-resonant part, and gives the de-excitations energies [20]. In the calculations of optical spectra the coupling part of the matrix is often neglected (Tamm-Dancoff approximation). In fact, this is what is presently implemented in SAX .

In this approximation the eigenvalue problem can be rewritten as:

$$\sum_{(v_2 c_2)} H_{(v_1 c_1)(v_2 c_2)}^{exc, 2p \text{ resonant}} A_{(v_2 c_2)}^\mu = E_\mu^{exc} A_{(v_1 c_1)}^\mu, \quad (31)$$

and the macroscopic dielectric tensor takes the following form:

$$\epsilon_M = 1 - \lim_{q \rightarrow 0} v_0(q) \sum_\mu \sum_{(v_1 c_1)} \frac{|\langle v_1 | e^{-iqr} | c_1 \rangle A_\mu^{(v_1 c_1)}|^2}{E_\mu^{exc} - \omega - i\eta} \quad (32)$$

For a more detailed derivation of the two-particle Hamiltonian see Refs. [14, 21] .

1.3 In practice

All the previous relations are written in frequency domain, since in our implementation we never use the time-dependent operators. It is also possible to implement the expressions directly in time domain [9]. Moreover, while every operation is here written in real space, we use everywhere a

plane-waves basis set, and we switch to the real space representation with fast Fourier transforms whenever convenient.

We consider a periodic system with unit cell Ω , and we set to N the number of cells, working in the limit $N \rightarrow \infty$. Every function depending on a single spatial coordinate (such as the Kohn-Sham orbitals) is represented as a sum of normalized plane waves.

$$\psi_{nk}(x) = \frac{1}{\sqrt{N\Omega}} \sum_G \psi_{nk}(G) e^{i(k+G)\cdot x} , \quad (33)$$

where G runs over the reciprocal lattice. We introduce the product between two Kohn-Sham states in real space as

$$M_{nk,n'k'}(x) = \sqrt{N\Omega} \psi_{nk}^*(x) \psi_{n'k'}(x) . \quad (34)$$

The convention that we adopt for the Fourier representation of two-points functions such as the polarizability, P , is:

$$P(xx') = \frac{1}{N\Omega} \sum_{kGG'} e^{i(k+G)\cdot x} P_k(G, G') e^{-i(k+G')\cdot x'} . \quad (35)$$

Here the sum over k vectors runs over a uniform N -points mesh in the first Brillouin zone. We follow the same convention also for the Coulomb potential, which results to be a diagonal matrix.

After some manipulation, P can be expressed in reciprocal space in terms of the products $M_{nk,n'k'}$

$$P_q(GG'; is) = -\frac{4}{N\Omega} \sum_{cvq} \theta(\epsilon_{ck} - \mu) \theta(\mu - \epsilon_{vk+q}) \times \\ M_{ck,vk+q}^*(G) M_{ck,vk+q}(G') \frac{2(\epsilon_{ck} - \epsilon_{vk+q})}{(\epsilon_{ck} - \epsilon_{vk+q})^2 + s^2} \quad (36)$$

In principle, the sums over k and q vectors should be performed using an infinitely dense mesh covering the first Brillouin zone. In practice, we use the special points technique [22] for evaluating these sums. This technique can be interpreted as a Fourier quadrature of the integrand [23]. Thus, in the practical usage, N is finite and equal to the number of k points employed, and the summation is assumed to be on a proper mesh, generated following Ref. [23].

The screened Coulomb potential is then obtained as

$$W(is) = (1 - v_C P(is))^{-1} v_C \quad (37)$$

where matrix operations are implicit. We notice that for imaginary frequencies the P matrix is Hermitian, thus also W is Hermitian.

For the building of the quasi-particle hamiltonian we use self-energies projected on states. These states can be Kohn-Sham states or states coming from a previous GW calculation (GW scf scheme, see below). The projected self-energies (exchange and correlation) take the following form:

$$\langle nk | \Sigma_x | n'k \rangle = -\frac{1}{N\Omega} \sum_{vq} \sum_G \theta(\mu - \epsilon_{vk-q}) M_{vk-q,nk}^*(G) v_C(q+G) M_{vk-q,n'k}(G) \quad (38)$$

$$\langle nk|\Sigma_c(\omega)|n'k\rangle = -\frac{1}{N\Omega} \sum_{n''q} \sum_{GG'} M_{n''k-q,nk}^*(G) M_{n''k-q,n'k}(G') \times \frac{-1}{2\pi i} \int d\omega' \frac{W_{GG'}^p(q;\omega') e^{-i\delta\omega'}}{\omega + \omega' - \epsilon_{n''k-q} - i\eta \text{sign}(\mu - \epsilon_{n''k-q})}. \quad (39)$$

In the case of equation (11), the calculation of the last integral should be performed using the lowest possible number the integrand evaluations. Inside SAX this improper integral has been mapped into a proper one through a change of variable $s = \omega_0 \tan(\phi)$, with $\phi \in [0, \frac{\pi}{2}]$; now it can be evaluated by means of the Gauss-Legendre formula [24]. The parameter ω_0 and the number of points in the integral N_ω are used to define a list of frequencies and weighs. Those frequencies are used since the beginning to evaluate the polarizability on the imaginary axis.

It is important to notice that, when representing Σ on a plane waves basis set, the energy cutoff necessary to represent correctly Σ_x can be different, and in general is much higher, than the one necessary for Σ_c .

As the self-energy is in general energy-dependent and non-Hermitian, it is not trivial to perform GW calculation within a scf scheme. To find the new states the self-energy has to be calculated at a given energy and the quasi-particle hamiltonian has to be diagonalized. In a non-scf calculation the quasi-particle hamiltonian is:

$$\langle nk|H_{QP}|nk\rangle = [\langle nk|h|nk\rangle + \langle nk|V_H|nk\rangle + \langle nk|\Sigma_x|nk\rangle + \langle nk|\Sigma_c(\epsilon_{nk})|nk\rangle] \left[\frac{1}{1 - \frac{\partial \Sigma_c}{\partial \omega} |_{\epsilon_{nk}}} \right], \quad (40)$$

where the energy dependence of the quasi-particle Hamiltonian has been developed to first order. This expression is equivalent to those implemented in other codes (as abinit [25]), and we just took away the explicit dependence with the DFT exchange-correlation potential. For practical purpose, expression (40) saves the implementation of different exchange-correlation potentials within the GW code, as ϵ_{nk} comes from the DFT code or from a previous GW calculation. Contrary to other implementations (see abinit [25] or self [26], for instance), the self-energy derivative ($\frac{\partial \Sigma_c}{\partial \omega}$) is taken here analytically.

In the case of GW scf scheme, it has been proposed in Ref.[27] to impose an hermitean and static self-energy by using a “ponderated mean”, or in other words a modified self-energy:

$$\langle nk|\Sigma|n'k\rangle = \frac{1}{4} (\langle nk|\Sigma(\epsilon_{n'k})|n'k\rangle + \langle n'k|\Sigma(\epsilon_{n'k})|nk\rangle^* + \langle nk|\Sigma(\epsilon_{nk})|n'k\rangle + \langle n'k|\Sigma(\epsilon_{nk})|nk\rangle^*) \quad (41)$$

In our particular implementation, this ”ponderated mean” is needed just for the plasmon-pole self-energy (see 1.3.1), for the other cases (see equations (9), (10) and (11)) the self-energy is energy independent and Hermitian.

1.3.1 The plasmon-pole model

In the plasmon-pole model the dynamical behavior of the dielectric matrix is given by a parametrized model of the dielectric function, in reciprocal space. The dielectric function is, then, given by:

$$\epsilon_{GG'}^{-1} = 1 + \frac{\Omega_{GG'}^2(q)}{\omega^2 - (\omega_{\tilde{G}G'}(q) - i\eta)^2}, \quad (42)$$

where $\Omega_{GG'}(q)$ (effective bare plasma frequency) and $\omega_{GG'}$ are the two parameters of the model. The small $i\eta$ is just to ensure the right time-ordering. There exist different ways to fit these parameters. To this purpose, two constraints are needed. In SAX we choose two frequencies, where ϵ^{-1} is actually computed: $\omega = 0$ and $\omega = i\omega_p$ (ω_p is the plasmon frequency). In systems where plasmons are not well defined this model should, in theory, not work.

Once the screening is known, the correlation part of the self-energy (integral in Eq. (7)) can be evaluated analytically. By closing the integration path through an arc in the lower half plane ($e^{-i\delta\omega'}$) and applying the residue theorem, one gets for the integral:

$$\begin{aligned} \frac{-1}{2\pi i} \int d\omega' \frac{W_{GG'}^p(q; \omega') e^{-i\delta\omega'}}{\omega + \omega' - \epsilon_{nk} - i\eta \text{sign}(\mu - \epsilon_{nk})} = \\ \frac{-1}{2\pi i} \int d\omega' \frac{\epsilon_{GG'}^{-1}(q) v_C(G' + q) e^{-i\delta\omega'}}{\omega + \omega' - \epsilon_{nk} - i\eta \text{sign}(\mu - \epsilon_{nk})} = \\ -\frac{\Omega_{GG'}^2 v_C(G' + q)}{2\omega_{GG'}(q)} \left[\frac{\theta(\mu - \epsilon_{nk})}{\omega + \tilde{\omega}_{GG'}(q) - \epsilon_{nk} - i\eta} + \frac{\theta(\epsilon_{nk} - \mu)}{\omega + \tilde{\omega}_{GG'}(q) - \epsilon_{nk} + i\eta} \right]. \quad (43) \end{aligned}$$

For the self-energy projected on states, the implemented formula is:

$$\begin{aligned} \langle nk | \Sigma_c(\omega) | n'k \rangle = \\ \frac{1}{N\Omega} \sum_{n''q} \sum_{GG'} M_{n''q, nk}^*(G) M_{n''q, n'k}(G') v_C(G' + q) \times \\ - \frac{\Omega_{GG'}^2}{2\omega_{GG'}(q)} \left[\frac{\theta(\mu - \epsilon_{n''k-q})}{\omega + \tilde{\omega}_{GG'}(q) - \epsilon_{n''k-q} - i\eta} + \frac{\theta(\epsilon_{n''k-q} - \mu)}{\omega + \tilde{\omega}_{GG'}(q) - \epsilon_{n''k-q} + i\eta} \right]. \quad (44) \end{aligned}$$

The derivative of the self-energy, calculated at a given ϵ_{nk} is:

$$\begin{aligned} \langle nk | \frac{\partial \Sigma_c}{\partial \omega} | nk \rangle |_{\epsilon_{nk}} = \\ \frac{1}{N\Omega} \sum_{n''q} \sum_{GG'} M_{n''q, nk}^*(G) M_{n''q, n'k}(G') v_C(G' + q) \times \\ \frac{\Omega_{GG'}^2}{2\omega_{GG'}(q)} \left[\frac{\theta(\mu - \epsilon_{n''k-q})}{(\epsilon_{nk} + \tilde{\omega}_{GG'}(q) - \epsilon_{n''k-q} - i\eta)^2} + \frac{\theta(\epsilon_{n''k-q} - \mu)}{(\epsilon_{nk} + \tilde{\omega}_{GG'}(q) - \epsilon_{n''k-q} + i\eta)^2} \right]. \quad (45) \end{aligned}$$

1.3.2 Divergencies in the Coulomb potential

One common problem that is encountered when implementing the GW equations on a plane-waves basis set is that the Coulomb potential is singular in both the real and the reciprocal space. The explicit expression of the Coulomb potential, in Rydberg units, is

$$v_C(x) = \frac{2}{|x|} ; v_C(q) = \frac{8\pi}{q^2}, \quad (46)$$

where the Fourier transform is defined so that $v_C(x) = \frac{1}{N\Omega} \sum_q v_C(q) e^{iq \cdot x}$. An analytical treatment of the Coulomb potential is necessary when one deals with integrals on the Brillouin zone. For instance, this is required when calculating the Hartree-Fock exchange operator in periodic systems.

In this case, a special-point summation is not reliable and the discontinuity should be handled in a special manner. In general, the following integral form is needed:

$$I = \frac{1}{(2\pi)^3} \int dk v_C(k) f(k) , \quad (47)$$

where $f(k)$ is a function smooth enough to be integrated with the special-point technique. First, we split the Coulomb potential in a long-range part v_L and a short-range part v_S so that $v_C(x) = v_L(x) + v_S(x)$. Then, we add and subtract the long-range part of the Coulomb potential times the function calculated in the critical point, that is

$$I = \frac{1}{(2\pi)^3} \int dk (v_C(k) f(k) - v_L(k) f(k=0)) + \frac{1}{(2\pi)^3} \int dk v_L(k) f(k=0) . \quad (48)$$

The reason for adding and subtracting the long-range part only is to have a finite value for the second integral in the last expression. We define v_L and v_S as

$$v_L(q) = \frac{8\pi}{q^2} e^{-\frac{q^2}{2\sigma^2}} ; v_S(q) = \frac{8\pi}{q^2} \left(1 - e^{-\frac{q^2}{2\sigma^2}} \right) . \quad (49)$$

Since the long-range part has to be as close as possible to the bare potential, whose divergency has to be cancelled, we chose σ to be significantly larger than the spacing of the special-point mesh. The first integral in Eq. (48) can be assumed to be suitable for the special-point technique (i.e. $\frac{1}{(2\pi)^3} \int dk = \frac{1}{N\Omega} \sum_k$) while the second one leads exactly to $v_L(r=0) f(k=0)$. Now, isolating the $k=0$ term in the summation, and remembering $v_C = v_S + v_L$, one has

$$I = \frac{1}{N\Omega} \sum'_k v_C(k) f(k) + \left(v_L(x=0) + \frac{1}{N\Omega} v_S(k=0) - \frac{1}{N\Omega} \sum'_k v_L(k) \right) f(k=0) \quad (50)$$

This equation can be interpreted as a substitution of the divergent v_C with a modified \bar{v}_C so that

$$I = \frac{1}{N\Omega} \sum_k \bar{v}_C(k) f(k) \quad (51)$$

$$\bar{v}_C(k) = \begin{cases} N\Omega v_L(r=0) + v_S(k=0) - \sum'_{k'} v_L(k) & \text{if } k=0, \\ v_C(k) & \text{otherwise.} \end{cases} \quad (52)$$

where $v_L(x=0) = \frac{2\sqrt{2}\sigma}{\sqrt{\pi}}$ and $v_S(k=0) = \frac{4\pi}{\sigma^2}$. The sum over k' can be evaluated using a finite number of k vectors with the same spacing as the grid used for special-point integration. The necessary number of k vectors depends on the cutoff σ . This procedure is quite easy to implement, since the smoothed potential \bar{v}_C is calculated once and then used everywhere as in the standard special-point method.

A similar approach can be used for the screened potential. Here, for small q , the screened Coulomb potential is approximately

$$v_{SC}(q) = \frac{8\pi}{q\epsilon q} \quad (53)$$

where ϵ is in general a complex non-Hermitian tensor. We introduce a screened long-range potential as

$$v_{SL} = v_{SC}(q) e^{-\frac{q^2}{2\sigma}} \quad (54)$$

The real space representation of this potential is not analytical; however quantities such as $v_{SL}(x=0) = \frac{2\sqrt{2}\sigma}{\sqrt{\pi}} \langle \frac{q^2}{q\epsilon q} \rangle$ and $v_{SL}(q=0) = \frac{4\pi}{\sigma^2} \langle \frac{q^2}{q\epsilon q} \rangle$, can be obtained, where $\langle \frac{q^2}{q\epsilon q} \rangle$ means a spherical average which can be easily evaluated numerically.

1.3.3 Analytic treatment of the dielectric matrix

In the last subsection we have shown how to treat the divergency of the Coulomb potential in the integral required for the self-energy evaluation. We have also derived the expression required for the long-range part of the screened potential in terms of the macroscopic dielectric function. Here we show how we calculate the macroscopic dielectric function. Equation (37) can be casted equivalently as:

$$W^{-1}(\omega) = v^{-1} - P(\omega) \quad (55)$$

Due to the periodicity of the external Hamiltonian, all these operators are block-diagonal and representable in the form of Eq. (35). To obtain the limit of $q \rightarrow 0$ of the screened potential we first notice that the so-called heads and wings [20] of the polarizability can be expanded around $q = 0$ as:

$$P_q(G = 0, G' = 0) \sim \sum_{\alpha\alpha'} q_\alpha S^{\alpha\alpha'} q_{\alpha'} \quad (56)$$

$$P_q(G, G' = 0) \sim \sum_{\alpha} L^\alpha(G) q_\alpha \quad (57)$$

$$P_q(G, G' = 0) \sim \sum_{\alpha} q_\alpha (R^\alpha(G'))^* \quad (58)$$

These equations define S , R , and L . We will not write their explicit expression as a function of the Kohn-Sham orbitals, which can be obtained taking the Hessian and the gradients with respect to q of the polarizability in (36). We simply recall that the calculation of the gradients $\frac{\partial M_{nk, nk+q}(G)}{\partial q}$, is not trivial and has to include a contribution from the derivative with respect to q of the non-local part of the pseudopotentials. We notice that for imaginary frequencies the polarizability operator is Hermitian, therefore $L = R$. However we write the following equations in their general form, which can be applied also to non-Hermitian polarizability. We first introduce an operator \check{P} , defined as:

$$\check{P}_q(G, G') = \begin{bmatrix} 1 & 0 \\ 0 & \frac{(q+G)^2}{8\pi} - P_q(G, G') \end{bmatrix} \quad (59)$$

In terms of the \check{P} operator, the inverse of the screened potential operator is

$$W_q^{-1}(G, G') \sim \begin{bmatrix} \sum_{\alpha\alpha'} q_\alpha (\frac{1}{8\pi} - S^{\alpha\alpha'}) q_{\alpha'} & - \sum_{\alpha} q_\alpha (R^\alpha(G'))^* \\ - \sum_{\alpha} L^\alpha(G) q_\alpha & \check{P}(G, G') \end{bmatrix} \quad (60)$$

The matrix can be inverted in blocks. The block corresponding to $G = 0, G' = 0$ is

$$W_q(G = 0, G' = 0) = \frac{1}{\sum_{\alpha\alpha'} q_\alpha (\frac{1}{8\pi} - S^{\alpha\alpha'}) - \sum_{GG'} (R^\alpha(G))^* \check{P}_q(G, G') L^\alpha(G') q_{\alpha'}} \quad (61)$$

and the block corresponding to $G \neq 0, G' \neq 0$ is:

$$W_q(G, G') = W_q(G = 0, G' = 0) \begin{bmatrix} 1 & \sum_{\alpha} q_\alpha (\rho^\alpha(G'))^* \\ \sum_{\alpha} \lambda^\alpha(G) q_\alpha & \sum_{\alpha\alpha'} q_\alpha \lambda^\alpha(G) (\rho^{\alpha'}(G'))^* q_{\alpha'} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & (\check{P}^{-1})_q(G, G') \end{bmatrix} \quad (62)$$

It can be shown that the out-of-diagonal blocks of W^p are proportional to q . Thus, when integrated in (39), their contribution will be vanish. Thus, we set them to zero from the beginning:

$$G' \neq 0 \bar{W}_q(G = 0, G') = 0 \quad (63)$$

$$G \neq 0 \bar{W}_q(G, G' = 0) = 0 \quad (64)$$

1.3.4 The Bethe-Salpeter Equation

In practice the excitonic contribution is included first diagonalizing an effective two particle Hamiltonian (the Bethe-Salpeter Equation):

$$[(\epsilon_{ck} - \epsilon_{vk})\delta_{cc'}\delta_{vv'}\delta_{kk'} + K_{cvk,c'v'k'}^x + K_{cvk,c'v'k'}^d]\Psi_{c'v'k'}^n = \Omega^n \Psi_{cvk}^n, \quad (65)$$

where ϵ_{ck} and ϵ_{vk} are single particle energies from conduction and valence respectively, K^x is the exchange contribution to the kernel (present only for singlet states), K^d is the direct contribution to the kernel, $\Psi_{c'v'k'}^n$ is the n excitonic wave function and Ω^n is the n exciton energy. In a plane-waves expansion the kernel is:

$$K_{cvk,c'v'k'}^x = \sum_G M_{ck,vk}^*(G)v_C(G+q)M_{c'k',v'k'}(G') \quad (66)$$

$$K_{cvk,c'v'k'}^d = - \sum_{GG'} M_{ck,c'k+q}^*(G)W_{GG'}(q;0)M_{vk,v'k+q}(G') \quad (67)$$

The static approximation for the screening has been used.

Once the excitonic structure is obtained, the macroscopic dielectric tensor becomes:

$$\epsilon_M = 1 - \lim_{q \rightarrow 0} v_0(q) \sum_n \sum_{(v,c)} \frac{|M_{ck-q,vk}(0)A_{(v_1 c_1)}^n|^2}{\Omega^n - \omega - i\eta}. \quad (68)$$

2 Getting Started with SaX

2.1 Description of the package

The SaX package contains the following codes for the calculation of electronic and optical properties within the GW approximation, using a Plane-Wave basis set and pseudopotentials:

- `gw.x` : program for the calculation of quasi-particle band structure,
- `bse.x` : program for the solution of the Bethe-Salpeter Equation,
- `spectra.x` : program for the calculation of the macroscopic dielectric tensor;

the following auxiliary codes:

- `pptools.x` (Post-Processing tools): program for the calculation of localization properties,
- `memory.x` : program for the estimation of the memory needed by `gw.x` and `bse.x`,

and the following libraries:

- `iotk` (Input-Output ToolKit) : xml input-output (<http://www.s3.infm.it/iotk>),
- `ptk` (Parallel ToolKit) : MPI interface,
- `numrec` : contains mathematic utilities inspired on algorithms from numerical recipes,
- `lasi` : interface to blas and lapack,
- `lapack` : internal version of lapack,
- `fftw` and `ffti` : internal version of Fast Fourier Transform and FFT interfaces.

The SaX's codes work on many different types of Unix machines, including parallel machines using Message Passing Interface (MPI).

The SaX's codes work on top of a previous Density Functional Theory calculation. At the moment, SaX is interfaced with `pwscf.x` which belongs to the Quantum-Espresso package (<http://www.quantum-espresso.org>).

Further documentation, beyond what is provided in this guide, can be found in:

- the `doc/` directory of SaX distribution
In particular the `INPUT_*` files contain the detailed listing of available input variables and xml tags.
- the various `README` files found in the distribution
- the SaX web page <http://www.sax-project.org>.

Combining the different codes in SaX, the following quantities can be calculated:

1. Macroscopic dielectric tensor as a function of energy (optical absorption, real part, eels) within the Random Phase Approximation including or not including local field effects.
2. Quasi-particle energies and wave functions within the Hartree-Fock approximation (non-explicit self-consistency).
3. Quasi-particle energies and wave functions within the GW approximation (non-explicit self-consistency).
4. Excited states in the single configuration interaction (single CI) scheme, i.e. unscreened Bethe-Salpeter Equation.
5. Exciton energies and wave functions solving the Bethe-Salpeter Equation within the GW and Tamm-Dancoff approximation.
6. Macroscopic dielectric tensor as a function of energy including excitonic effects (optical absorption, real part, eels).
7. For Γ calculations: Localization properties as Participation Ration and Self-Interaction.

All of the above works for semi-conductors, insulators and molecules, in any crystal structure, for different approximations to the screened potential, for norm-conserving pseudopotentials (Hamann-Schlüter-Chiang) and starting from different initial electronic states (previous DFT, GW, or HF). A postprocessing program is available for the calculation of localization degrees (Participation Ration or Self-Interaction).

As the package is very modular and as every physical parameter can be set by the user, there exist many other ways of reaching quasi-particle energies, excitons or spectra. The SaX team does not recommend any other choice that is not described above.

2.2 Installation

Presently, the SAX package is only distributed in source form.

Stable releases of the SaX source package (current version is 1.0.0) can be downloaded from this URL:

```
http://www.sax-project.org/download.htm
```

Uncompress and unpack the distribution using the command:

```
tar -zxvf sax-1.0.0.tar.gz
```

If your version of `tar` doesn't recognize the `z` flag, use this instead:

```
gunzip -c sax-1.0.0.tar.gz | tar -xv
```

`cd` to the directory `sax-1.0.0/` that will be created.

To install SaX from source, you need C and Fortran-95 compilers (Fortran-90 is not sufficient, but most "Fortran-90" compilers are actually Fortran-95-compliant). If you don't have a commercial Fortran-95 compiler, you may install the free `g95` compiler (<http://www.g95.org/>) or the GNU Fortran compiler `gfortran` (<http://www.gfortran.org/>). You also need a minimal Unix environment: basically, a command shell (e.g., `bash` or `tcsh`) and the utilities `make`, `awk` and `sed`. MS-Windows users need to have Cygwin (a UNIX environment which runs under Windows) installed (see <http://www.cygwin.com/>)

Instructions for the impatient:

```

./configure
make allmake
cd sax
cd src
make all

```

Executable programs (actually, symlinks to them) will be placed in the `sax/bin/` directory.

If you encounter problems or you would like to tweak the default settings, read the detailed instructions below.

2.3 Configure

`configure` is a GNU-style configuration script, automatically generated by GNU Autoconf. (If you want to play with it, its source file is `$TOPDIR/conf/configure.ac`; you may also want to edit `$TOPDIR/conf/make.sys.in`) It generates the following files:

```

$TOPDIR/make.sys      compilation settings and flags
$TOPDIR/*/make.depend  dependencies, in each source dir

```

Files `make.depend` are actually generated by the `makedeps.sh` shell script. If you modify the program sources, you might have to rerun it. Note that you must run it from the directory it is in.

To force using a particular compiler, or compilation flags, or libraries, you may set the appropriate environment variables when running the configuration script. For example:

```
./configure CC=gcc CFLAGS=-O3 LIBS="-llapack -lblas -lfftw"
```

Some of those environment variables are:

```

TOPDIR           : top directory of the SAX tree (defaults to 'pwd')
F90, F77, CC     : Fortran 90, Fortran 77, and C compilers
CPP             : source file preprocessor (defaults to "$CC -E")
LD              : linker (defaults to $F90)
CFLAGS, FFLAGS,
F90FLAGS, CPPFLAGS, LDFLAGS : compilation flags
LIBDIRS         : extra directories to search for libraries (see below)

```

In principle, you should always be able to compile the SAX suite of programs without editing any of the generated files. If you ever have to do it, that should be considered a bug in the configuration script and you are encouraged to submit a bug report.

IMPORTANT: SAX can take advantage of several optimized numerical libraries:

- ESSL on AIX systems (shipped by IBM)
- MKL together with Intel compilers (shipped by Intel, free for non-commercial use)
- ATLAS (freely downloadable from <http://math-atlas.sourceforge.net>)
- FFTW (freely downloadable from <http://www.fftw.org>)

The configuration script attempts to find those libraries, but may fail if they have been installed in non-standard locations. You should look at the `LIBS` environment variable (either in the output

of the configuration script, or in the generated `make.sys`) to check whether all available libraries were found or not. If any libraries weren't found, you can rerun the configuration script and set manually a list of directories to search, by means of the environment variable `LIBDIRS`; directories in the list must be separated by spaces. For example:

```
./configure LIBDIRS="/opt/intel/mkl/mkl64/lib/32 /usr/local/lib/fftw-2.1.5"
```

If this still fails, you may set the environment variable `LIBS` manually and retry. For example:

```
./configure LIBS="-L/cineca/prod/intel/lib -lfftw -llapack -lblas"
```

Beware that in this case, you must specify **all** the libraries that you want to link to. The configuration script will blindly accept the specified value, and will **not** search for any extra library.

If you want to use the FFTW library, the `fftw.h` include file is also required. If the configuration script wasn't able to find it, you can specify the correct directory in the `INCLUDEFFTW` environment variable. For example:

```
./configure INCLUDEFFTW="/cineca/lib/fftw-2.1.3/fftw"
```

2.4 Compile

From the main directory of the SaX package the `make` command with no arguments yields a list of valid compilation targets:

Possible `<target>`'s are:

<code>make</code>	make IDE global Makefile
<code>allmake</code>	make IDE Makefiles in each dir
<code>clean</code>	IDE total cleanup
<code>add</code>	launch a script to add a new package
<code><dir>_all</code>	make all executable in <code><dir></code>
<code><dir>_loclib</code>	make the local library in <code><dir></code>
<code><dir>_make</code>	make the Makefile for <code><dir></code>
<code><dir>_allmake</code>	make all the Makefiles for <code><dir></code> and its libs
<code><dir>_clean</code>	cleanup in <code><dir></code>
<code><dir>_tar</code>	tar of a single dir
<code><dir>_alltar</code>	tar of a dir with its libs

Possible `<dir>` values are:

- `blas`: internal blas library
- `fftw`: internal FFT library
- `iotk`: iotk library
- `numrec`: mathematical utilities library
- `ptk`: MPI interface library
- `lapack`: internal lapack library

- `fti`: FFT interface library
- `lasi`: blas and lapack interface library
- `sax`: sax codes

2.5 DFT preliminary step

NOTE: At present the SAX suite of codes is implemented to work as post processing of DFT calculations done using PWSCF (<http://www.pwscf.org>) contained in the Quantum-Espresso distribution (<http://www.quantum-espresso.org>); we will refer to that code in the following.

IMPORTANT: For a correct result, the following steps **MUST** be done in the reported order.

1. DFT Self-consistent calculation.
For the description of the input and for further details see the PWSCF manual.
2. Bandstructure calculation.
Starting from the self-consistent charge calculated in point (1), we calculate the Bloch functions for a **REGULAR k-point grid in the COMPLETE Brillouin Zone**; Gamma point must be included. Reduction of k-points due to time-reversal symmetry is not (yet) allowed. The complete list of k-points should be specified in the K_POINTS card.
3. From PWSCF to the SAX package.
Use the post processing `pw_export.x` (distributed in the PWSCF package ¹), to extract the input data necessary for the following calculations from PWSCF output datafile. Data will be stored in the newly-created directory `$prefix.export/`.

NOTE: Steps (1-3) should be run, using the parallel version of the code, paying attention to use the same number of processors. From this point to the end, instead, the code is scalar.

2.6 Run examples

As a final check that compilation was successful, you may want to run some or all of the examples contained within the `/sax/examples` directory of the SaX distribution. Those examples try to exercise some of the standard uses of the tools implemented in the SaX package. An explanation on HOW TO run these examples is contained in the `~/examples/README` file. In each example directory, you will also find a `README` file describing the physics implied by test, the results and some suggestions for further calculations. Each example contains a `reference` directory `moooooohhh-hhh`. If you find that any relevant feature isn't being tested, please contact us (or even better, write and send us a new example yourself!).

The necessary pseudopotentials are included in the directory `/sax/examples/pseudos`.

Sax is used on top of a previous DFT calculation, before running the examples you should install a Quantum-espresso distribution (<http://www.quantum-espresso.org>).

To run the examples, you should follow this procedure:

¹The export utility is already distributed in the Quantum-Espresso v3.0, but a patch to include it also in versions v2.1.x is available at <http://www.sax-project.org> or <http://www.pwscf.org>. When using v.3.0.0 do not compile with the `-D__NEWPUNCH` pre-processor flag.

1. Go to the `/sax/examples` directory and edit the `set_env` file, setting the following variables as needed:

```
uncomment the line DFT_BIN="/my-espresso-full-path binaries"  
DFT_BIN= directory where Quantum-ESPRESSO executables reside  
source set_env
```

2. If you have compiled the parallel version of SAX (this is the default if parallel libraries are detected), you will usually have to specify a driver program (such as `poe`, `mpiexec` or `mpirun`) and the number of processors.

In order to do that, edit again the `set_env` file and set the `PARA_PREFIX` and `PARA_POSTFIX` variables as needed. Parallel executables will be run by a command like this:

```
$PARA_PREFIX executable.x $PARA_POSTFIX  
$FILE.in  
$FILE.out  
EOF
```

For example, if the command line is like this (as for an IBM clx):

```
mpirun gw.x -procs 4  
file.in  
file.out  
EOF
```

you should set `PARA_PREFIX="poe"`, `PARA_POSTFIX="-procs 4"`.

3. Non-Interactive mode: if your machine does not support interactive use, you must modify the `jrun` files on each example directory following the instructions of the batch queueing system installed on that machine. Ask your system administrator for instructions.
4. To run a single example, go to the corresponding directory (for instance, `examples/example1`) and execute:

```
./jrun
```

Some examples take only a few seconds to run, while others may require several minutes depending on your system.

To run all the examples in one shot, execute:

```
./jrun_all_examples
```

from the `/sax/examples` directory. On a single-processor machine, this typically takes from one to three hours.

5. In each example's directory, the **reference** subdirectory contains verified output files, that you can check your results against. They were generated on a Linux PC using the g95 compiler. On different architectures the precise numbers could be slightly different, in particular if different FFT mesh dimensions are automatically selected. For this reason, a plain **diff** of your results against the reference data doesn't work, or at least, it requires human inspection of the results. Usually IBM sp5 essl gives slightly different results, this discrepancy is enhanced in the case of molecules.

3 Using SaX

3.1 Input data

The input files of all the codes in SaX are built on the same general xml structure:

```
<input>
<sax_options
  ...
/>
```

Calculation options

```
<structure>
  <direct
    ...
  \>
<\structure>
```

Cell vectors should be in a.u.

```
<atoms>
  <types ntypes="number_of_atomic_species">
    ...
  <\types>

  <positions natoms="number_of_atoms" units="bohr|angstrom|alat|crystal"
    alat="value_of_cellparameter in bohr">
    atomic_symbol X Y Z
  </positions>
<\atoms>

<kmesh>
<mesh nk="nkx nky nkz" shift="shiftx shifty shiftz"/>
kme
</kmesh>
</input>
```

You may take the examples distributed with SaX as templates for writing your own input files: see section 2.6, “Run examples”. In the following, whenever we mention “Example N”, we refer to those ones. Input files are those with names ending in `.in` (they will appear after you have run the examples).

The keywords which do not fit input keywords are ignored by the codes. See file `doc/INPUT_GW`, `doc/INPUT_BSE`, `doc/INPUT_SPECTRA` or `doc/INPUT_PPTOOLS` for a detailed explanation of the meaning and format of the various fields.

The differences between input files for different codes are in the xml `sax_options` tag. The following xml tags in `<input>` must always be specified:

```

<structure>
  <direct
    ...
  \>
<\structure>

<atoms>
  <types ntypes="number_of_atomic_species">
    ...
  <\types>

  <positions natoms="number_of_atoms" units="bohr|angstrom|alat|crystal"
                                         alat="value_of_cellparameter in bohr">
    atomic_symbol X Y Z
  </positions>
<\atoms>

<kmesh>
<mesh nk="nkx nky nkz" shift="shiftx shifty shiftz"/>
kme
</kmesh>

```

Note about k points: The k-point grid is automatically generated following the convention of Monkhorst and Pack. The previous DFT calculation has to be done forcing a calculation with no symmetries (In the case of PWscf setting `ibrav=0` and `nosym=.true.`).

3.2 gw.x

gw.x performs quasi-particle electronic structure calculations within two main philosophies: GW or Hartree-Fock (the last one in the sense of Gv). G stays for the single-particle green function, W and v stay for the screen and the bare Coulomb potential respectively.

Attention: the number of bands has to be a multiple of the number of processors.

3.2.1 Calculation options

The calculations options are set in the xml tag `sax_options`. Most of the options introduced by keywords have self-explanatory names, as for instance:

```

calc_polarizability
cutoff_polarizability
convert_from_pwscf
nelec

```

The following variables in `<input>` must always be specified:

```

calculation_kind
nbandmin
nbandmax
convert_from_pwscf

```

If `convert_from_pwscf=".true."` is true then the following variables have to be setted:

```
convert_from_pwscf_file
nelec
```

Most variables of the xml tag `sax_options` have default values which may or may not fit your needs:

```
system_kind = "3D"
start_from = "DFT"

gw_integration_method = "plasmon_pole"

cutoff_density = "4.0"
cutoff_vxc      = "4.0"
cutoff_vloc     = "4.0"

cutoff_fock     = "6.0"
cutoff_polarizability = "6.0"

diagonal = ".true."
sigma_first_order = ".false."
emax_polarizability = "1000.0"

plasmon_energy = "1.3"
lorentzian_broadening = "0.01"

sax_to_want = ".false."
sax_to_want_output = "sax_to_want.xml"
```

All the calculation variables are set to `.false.` by default. Explanations for the meaning of variables are in file `INPUT_GW`. Please read them carefully.

3.2.2 Typical cases

We may distinguish the following typical cases for `gw.x`:

Quasi-particle band structure within Hartree-Fock (or G_v). Set `calculation_kind="HF"`.

The HF Hamiltonian is built on single-particle Hamiltonian plus HF self-energy. The single-particle Hamiltonian is calculated setting `calc_sp_hmatrix=".true."`. Specify the range, in terms of bands, on which the HF self-energy has to be calculated setting `sigma_nbmin` and `sigma_nbmax`. DFT wave functions are not eigenstates of HF Hamiltonian. If you want to calculate the new HF quasi-particle wave-functions, then `diagonal` has to be set to `.false.`

If you want to run after `gw.x` other calculations with SaX codes but starting from this HF band structure you need to set `calc_QP_states = ".true."`. See Example 2.

The SCF is not explicitly implemented. SCF may be reached by performing further HF calculations specifying `start_from="HF"`.

Quasi-particle band structure within GW. Set `calculation_kind="GW"`. Specify which approximation to GW you want to use by setting `gw_integration_method=` to one of the following "SSHf" (Statically Screened HF), "cohsex" (COlomb Hole and Screened eXchange) or `plasmon_pole` (plasmon pole approximation). The calculation of the screening potential (W) is performed in two steps: first calculation of the polarizability (set `calc_polarizability=".true."`), second calculation of W from the polarizability (set `calc_w=".true."`). The quasi-particle Hamiltonian is made of three parts: single-particle Hamiltonian plus self-energy exchange part plus correlation part. The single-particle Hamiltonian is calculated setting `calc_sp_hmatrix=".true."`. The exchange and correlation part of the self-energy is calculated by setting `calc_sigma_x=".true."` and `calc_sigma_c=".true."`, respectively. The Quasi-particle Hamiltonian is built within the code by setting `calc_energies=".true."`. Specify the range, in terms of bands, on which the GW self-energy has to be calculated setting `sigma_nbmin` and `sigma_nbmax`. DFT wave functions are not eigenstates of quasi-particle Hamiltonian. If you want to calculate the new GW quasi-particle wave-functions, then `diagonal` has to be set to `.false..`

If you want to run after `gw.x` other calculations with SaX codes but starting from this GW band structure, you need to set `calc_QP_states = ".true."`. See Example 2.

The SCF is not explicitly implemented. SCF may be reached by performing further GW calculations specifying `start_from="GW"`.

The output data files are produced in the directory where the code execution is done. In the following, a description of the output data files is given:

HF_QP : contains the eigenvalues and eigenvectors (in the basis of previous states) of HF Hamiltonian, in textual xml format.

GW_QP : contains the eigenvalues and eigenvectors (in the basis of previous states) of GW quasi-particle Hamiltonian, in textual xml format.

polar : contains the polarizability, in binary xml format.

w.smooth : contains the screened Coulomb potential, in binary xml format.

epsilon.static : contains the static macroscopic dielectric tensor, in textual xml format.

pp-parameters : contains the calculated parameters of the plasmon pole model, in binary xml format.

states : contains DFT energies and wave functions, in binary xml format.

HF_states : contains HF energies and wave functions, in binary xml format.

GW_states : contains GW energies and wave functions, in binary xml format.

sigma_x.ik : contains the expectation values of exchange part of the self-energy for each kpoint indexed as ik, in textual xml format.

sigma_c.ik : contains the expectation values of correlation part of the self-energy for each kpoint indexed as ik, in textual xml format.

dsigma_c.ik : contains the expectation values of energy derivatives of the self-energy correlation part for each kpoint indexed as ik, in textual xml format.

exp_sp.ik : contains the expectation values of single-particle Hamiltonian (kinetic + ion-electron + Hartree) for each kpoint indexed as ik, in textual xml format.

3.3 bse.x

bse.x solves the Bethe-Salpeter equation, giving the excitonic structure. The direct electron-hole interaction can be calculated using a full screened Coulomb potential obtained from a previous GW calculation or using a static dielectric tensor.

3.3.1 Calculation options

As in the gw.x case, the calculations options are set in the xml tag **sax_options**. Most of the options introduced by keywords have self-explanatory names, as for instance:

```
cutoff_polarizability
convert_from_pwscf
bse_spin
nelec
```

The following variables in `<input>` must always be specified:

```
calculation_kind
nbandmin
nbandmax
convert_from_pwscf
```

If `convert_from_pwscf=".true."` then the following variables have to be set:

```
convert_from_pwscf_file
nelec
```

Most variables of the xml tag **sax_options** have default values which may or may not fit your needs:

```
cutoff_polarizability="6.0"
cutoff_fock="6.0"

bse_emin="0.0"
bse_emax="1000.0"
bse_spin="0"

system_kind="3D"
energy_shift="0.0"
start_from="DFT"
```

Explanations for the meaning of variables are in file `INPUT_BSE`. Please read them carefully.

3.3.2 Typical cases

We may distinguish two typical cases for `bse.x`:

Excitonic structure with macroscopic screening. Set `calculation_kind="MACRO"`. The direct electron-hole interaction is calculated using a macroscopic screening (the code reads the file `epsilon.static`). The format of `epsilon.static` is explained in `example2`.

Excitonic structure with the full screened Coulomb potential. Set `calculation_kind="FULL"`. The direct electron-hole interaction is calculated using the full screened Coulomb potential as obtained by a previous GW calculation (the code reads `w.smooth`).

The variable `energy_shift` is used to shift single particle energies rigidly.

The output data files are written in the directory where the execution is done. `bse.x` has just one output file: `bse`, which contains exciton energies and eigenvectors in the basis of single particle states, in textual xml format.

3.4 `spectra.x`

`spectra.x` computes the macroscopic dielectric tensor within the Random Phase Approximation with or without excitonic effects as a function of energy. The dielectric tensor is projected on an arbitrary q direction and is separated in different contributions: real part, Optical Absorption Spectrum (`oas`) and Electron Energy Loss spectrum (`eels`).

3.4.1 Calculation options

As in the previous cases, the calculations options are setted in the xml tag `sax_options`. Most of the options introduced by keywords have self-explanatory names, as for instance:

```
cutoff_polarizability
convert_from_pwscf
bse_spin
nelec
```

The following variables in `<input>` must always be specified:

```
calculation_kind
nbandmin
nbandmax
convert_from_pwscf
```

If `convert_from_pwscf=".true."` then the following variables have to be set:

```
convert_from_pwscf_file
nelec
```

Most variables of the xml tag `sax_options` have default values which may or may not fit your needs:

```

cutoff_polarizability="6.0"
cutoff_fock="6.0"
local_field_effects=".false."

bse_emin="0.0"
bse_emax="1000.0"
bse_spin="0"

system_kind="3D"
energy_shift="0.0"
start_from="DFT"

imaginary_axis=".false."
lorentzian_broadening="0.01"

omegamax="0.0"
nomega="0"

q="1.0 0.0 0.0"
just_project_on_q_direction=".false."

output_format="txt"

```

Explanations for the meaning of variables are in file `INPUT_SPECTRA`. Please read them carefully.

3.4.2 Typical cases

We can distinguish two typical cases:

Macroscopic dielectric tensor. Set `calculation_kind="RPA"`. The macroscopic dielectric tensor is calculated in RPA taking into account only interband transitions, see a more detailed explanation in `example1`.

Macroscopic dielectric tensor with excitonic effects. Set `calculation_kind="RPAEXC"`. The macroscopic dielectric tensor is calculated in RPA including excitonic effects. The code need a previous bse calculation (needs file `bse`).

The variable `energy_shift` is used to shift single particle energies rigidly. When used with `calculation_kind="RPAEXC"` should be coherent with the shift used in the previous bse calculation.

The output data files are written in the directory where the execution is done. In the following a description of the output data files is given:

epsilon.nlf.RPA : contains the dielectric macroscopic tensor as a function of energy calculated within RPA, in textual xml format.

epsilon.lf.RPA : contains the dielectric macroscopic tensor as a function of energy calculated within RPA and including local field effects, in textual xml format.

epsilon.RPAEXC : contains the dielectric macroscopic tensor as a function of energy calculated within RPA, including local field effects and excitonic effects, in textual xml format.

repsilon.(lf/nlf). `calculation_kind` : real part of epsilon projected on the specified q direction, in format specified by option `output_format`.

oas.(lf/nlf). `calculation_kind` : imaginary part of epsilon (optical absorption spectrum) projected on the specified q direction, in format specified by option `output_format`.

eels.(lf/nlf). `calculation_kind` : imaginary part divided by the square of epsilon (electron energy loss spectrum) projected on the specified q direction, in format specified by option `output_format`.

3.5 pptools.x

`pptools.x` compute the Self-Interaction (SI). The Participation Ratio (PR) is in project but not yet in use.

3.5.1 Calculation options

As in the previous cases, the calculations options are setted in the xml tag `sax_options`. Most of the options introduced by keywords have self-explanatory names, as for instance:

```
cutoff_density
cutoff_fock
nelec
```

The following variables in `<input>` must always be specified:

```
calculation_kind
nbandmin
nbandmax
convert_from_pwscf
```

If `convert_from_pwscf="true."` then the following variables have to be setted:

```
convert_from_pwscf_file
nelec
```

Most variables of the xml tag `sax_options` have default values which may or may not fit your needs:

```
system_kind="3D"
start_from="DFT"

cutoff_fock="6.0"
cutoff_density="4.0"
```

Explanations for the meaning of variables are in file `INPUT_PPTOOLS`. Please read them carefully. `pptools.x` has just one kind of output file according to the requested kind of calculations: `self-interaction.ik` or `participation-ratio.ik`, where `ik` stands for the kpoint index on which SI or PR is calculated. Those files are textual in xml format.

3.6 memory.x

memory.x gives an estimation of the memory need by gw.x or bse.x. Only the most memory-demanding objects and operations (implying allocation of temporary big arrays) are taken into account. gw.x or bse.x input files are used as memory.x input files. Few additional options can be added to have memory statistics as a function of required number of process.

3.6.1 Additional Calculation options

As in the previous cases, the calculations options are setted in the xml tag `sax_options`. The proper memory.x options and their respective default values are:

```
guessed_nb_proc = "1"  
nb_memory_test_nproc = "4"  
memory_output_file = "memory.out"
```

Explanations for the meaning of variables are in file `INPUT_MEMORY`. Please read them carefully.

Memory.x as a unique output file where the maximum memory allocated by the programs gw.x and bse.x is printed according to the number of process targeted by the user (`guessed_nb_proc`) and to other (`nb_memory_test_nproc`) different process numbers. The file is written in textual xml format.

References

- [1] L. Hedin, Phys. Rev. **139**, A796 (1965).
- [2] L. Hedin and S. Lundqvist, Solid State Phys. **23**, 1 (1969).
- [3] M. S. Hybertsen and S. G. Louie, Phys. Rev. B **30**, 5777 (1984).
- [4] M. S. Hybertsen and S. G. Louie, Phys. Rev. Lett. **55**, 1418 (1985).
- [5] R. W. Godby, M. Schlüter, and L. J. Sham, Phys. Rev. B **37**, 10159 (1988).
- [6] B. Farid, R. Daling, D. Lenstra, and W. van Haeringen, Phys. Rev. B **38**, 7530 (1988).
- [7] G. E. Engel, B. Farid, C. M. M. Nex, and N. H. March, Phys. Rev. B **44**, 13356 (1991).
- [8] E. L. Shirley and R. M. Martin, Phys. Rev. B **47**, 15404 (1993).
- [9] H. N. Rojas, R. W. Godby, and R. J. Needs, Phys. Rev. Lett. **74**, 1827 (1995).
- [10] M. Rohlfing, P. Krüger, and J. Pollmann, Phys. Rev. B **52**, 1905 (1995).
- [11] P. Hohenberg and W. Kohn, Phys. Rev. **136**, B864 (1964).
- [12] W. Kohn and L. J. Sham, Phys. Rev. **140**, A1133 (1965).
- [13] F. Aryasetiawan and O. Gunnarsson, Rep. Prog. Phys. **61**, 237 (1998).
- [14] G. Onida, L. Reining, and A. Rubio, Rev. Mod. Phys. **74**, 601 (2002).
- [15] A. L. Fetter and J. D. Walecka, *Quantum theory of many-particle systems*, McGraw-Hill, New York, 1971.
- [16] B. Farid, Ground and low-lying excited states of interacting electron systems; a survey and some critical analyses, in *Electron Correlation in the Solid State*, edited by N. H. March, pages 103–261, Imperial College Press, London, 1999.
- [17] G. E. Engel and B. Farid, Phys. Rev. B **47**, 15931 (1993).
- [18] R. W. Godby and R. J. Needs, Phys. Rev. Lett. **62**, 1169 (1989).
- [19] M. S. Hybertsen and S. G. Louie, Phys. Rev. B **34**, 2920 (1986).
- [20] M. Rohlfing and S. G. Louie, Phys. Rev. B **62**, 4927 (2000).
- [21] G. Bussi, Phys. Scr. **T109**, 141 (2004).
- [22] D. J. Chadi and M. L. Cohen, Phys. Rev. B **8**, 5747 (1973).
- [23] S. Froyen, Phys. Rev. B **39**, 3168 (1989).
- [24] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical recipes in FORTRAN*, Cambridge University Press, Cambridge, second edition, 1992.

- [25] X. Gonze, J. M. Beuken, R. Caracas, M. F. F. Detraux, G. M. Rignanese, L. Sindic, M. Verstraete, G. Zerah, F. Jollet, M. Torrent, A. Roy, M. Mikami, P. Ghosez, J. Y. Raty, and D. C. Allan, *Comp. Mat. Sci* **25**, 478 (2002).
- [26] A. Marini, the `self` project <http://people.roma2.infm.it/~marini/self>.
- [27] S. V. Faleev, M. van Schlipfgaarde, and T. Kotani, *Phys. Rev. Lett.* **93**, 126406 (2004).