Masterarbeit

zur Erlangung des akademischen Grades
Master of Science (M. Sc.)
im Fach Physik

Implementation of the Sternheimer equation in the *GW* approximation

Eingereicht an der
Mathematisch-Naturwissenschaftlichen Fakultät I
Institut für Physik
Humboldt-Universität zu Berlin
am
29.11.2024

von
Hannah Kleine

Erstgutachter/in:	Prof. Dr. Claudia Draxl
Zweitgutachter/in:	Dr. Denis Usvyat
Betreuung:	Sven Lubeck

Contents

List of Figures

List of Tables

1. Introduction

The electronic band structure is an important property for characterizing a solid. Experimentally, it is commonly measured using photoemission spectroscopy [3]. For the theoretical description of the electronic structure, Kohn-Sham (KS) density functional theory (DFT) [4, 5] has long been the primary method. In combination with the widely used local density approximation (LDA) or generalized gradient approximation (GGA) functionals, it yields good results for ground-state properties, but fails to give accurate results for the electronic band structure. While more advanced functionals like meta-GGA and hybrid functionals yield more accurate results for the electronic band structure, they come with significantly higher computational cost. An alternative to DFT is many-body perturbation theory, which is based on Green's functions. The electronic self-energy is the central property in this framework. It describes many-body exchange and correlation effects beyond Hartree theory. A commonly used approximation to the self-energy is the GW approximation, first proposed by Hedin in 1965 [6, 7]. Today, the GW approximation is a state-of-the-art approach for the description of electronic properties of solids. However, according to the original proposal of Hedin, the GW equations form a loop. To obtain the quasiparticle (QP) energies, these equations must be solved self-consistently, making this approach too computationally expensive for most applications. In order to reduce computational cost, the self-consistency can be removed. In the one-shot GW or G_0W_0 approximation, the quasiparticle energies are obtained by a perturbative correction to Kohn-Sham (KS) (or Hartree-Fock) eigenvalues. However, the benefit of the reduced computational cost comes with additional drawbacks. Results obtained with the G_0W_0 method depend on the reference used as a starting point, as shown by Bruneval and coworkers [8], among others. Nonetheless, in most cases G_0W_0 provides QP band gaps that are in good agreement with experiment, especially for medium band gap materials [9]. In practice, a main challenge arises from the slow convergence of the correlation part of the self-energy with respect to the number of unoccupied bands [9, 10, 11, 12]. The correlation part of the self-energy is computed from the polarizability, which involves a summation over an infinite number of unoccupied states. In practical calculations, this sum needs to be truncated. Short-range electron correlations due to Coulomb forces cause a slow convergence of this sum with respect to the number of included states [13]. In addition, the basis used to compute ground-state properties within DFT is not optimized for describing high-energy states, causing a slow convergence with respect to the number of basis functions included in the calculation [9, 10, 11, 12]. These convergence issues are inherent to all flavours of G_0W_0, regardless of whether pseudopotentials are utilized or what kind of basis set is used. In the last years, a number of approaches have been proposed that help to avoid the unfavorable convergence behavior of G_0W_0 calculations. For example, it has been suggested to replace the band energies above a certain threshold with a constant energy. However, this so-called extra-polar approximation [14] is not parameter-free and can still lead to a slow convergence with respect to the number of included bands. Further development led to a similar, but parameter-free method. Here, the energies of the unoccupied states are not replaced by a parameter, but by a band-index independent effective-energy function [15, 16]. Unfortunately, this effective-energy function is not known and needs to be approximated. An alternative method that avoids the summation over unoccupied states, while being parameter-free and requiring no additional approximations, is the so-called Sternheimer *GW* [17, 18, 19]. In this method, the self-energy is computed by solving the linear Sternheimer equation in a basis. While this method avoids explicit summation over unoccupied states, the solution still lives in the Hilbert space spanned by a finite number of basis functions. These basis functions are optimized for the description of unperturbed wave functions. Therefore, a large number of such basis functions are required to accurately capture the wave function response.

In this work, we explore a method proposed by Betzinger and coworkers [20, 21, 22]. This method helps to strongly reduce the number of required unoccupied states and basis functions without introducing parameters or additional approximations. This method, called incomplete basis set correction (IBC), is a correction for the wave function response, consisting of two terms, namely the basis response (BR) term and the Pulay term. These two terms are added to the sum-over-states (SOS) expression of the polarizability, which contains the sum over unoccupied states. In the limit of an infinite sum, both correction terms vanish. In the practical case of a truncated sum, the BR term compensates for the incompleteness of the basis, and the Pulay term makes up for the fact, that eigenfunctions of a Hamiltonian computed in a finite, incomplete basis are not point-wise exact eigenfunctions of this Hamiltonian. The IBC relies on the the fact that the basis functions depend on the effective potential. While this is not the case for all types of basis sets, it applies to the all-electron full-potential (linearized) augmented planewaves ((L)APW) plus local orbitals (LO) method used in the all-electron full-potential computer package `exciting` [23]. The (L)APW+LO method is often taken as a reference, since it avoids crude approximations in the treatment of core states as ,e.g., required by pseudopotential methods. In spherical regions around the nuclei, basis functions in the (L)APW+LO method are computed by solving the radial Schrödinger equation. In these regions, the Schrödinger equation and therefore the basis functions depend on the effective potential. The IBC makes use of this dependence by considering the response of the basis functions to the change in potential. This response is calculated by solving the radial Sternheimer equation. The basis-function response rotates the Hilbert space spanned by the unperturbed basis functions, which allows to capture response contributions from outside of the space spanned by the original basis without increasing the number of basis functions. In this work, we take first steps towards implementing the IBC in the computer package `exciting` [23]. We implement an explicit computation of the SOS-expression of the polarizability which, in future work, will be corrected by the BR and Pulay correction terms. In addition, we implement and compare different radial solvers for inhomogeneous ordinary differential equations (ODE), which are needed to solve the radial Sternheimer equation.

2. Theoretical background

2.1. *GW* approximation

In GW, the many-body excitation energies $E^{\sigma}_{i\boldsymbol{q}}$ are defined by the poles of the interacting single-particle Green's function $G^{\sigma}(\boldsymbol{r},\boldsymbol{r}',\omega)$

$$G^{\sigma}(\boldsymbol{r},\boldsymbol{r}',\omega)=\sum_{i\boldsymbol{k}}\frac{\Psi^{\sigma}_{i\boldsymbol{k}}(\boldsymbol{r})\;[\Psi^{\sigma}_{i\boldsymbol{k}}(\boldsymbol{r}')]^{*}}{\omega-E^{\sigma}_{i\boldsymbol{q}}+E_0+i\,\eta\;\mathrm{sgn}(E^{\sigma}_{i\boldsymbol{q}}-E_0)}. \tag{2.1}$$

Here, $\Psi^{\sigma}_{i\boldsymbol{k}}(\boldsymbol{r})$ is the quasiparticle wave function, E_0 stands for the many body ground-state energy, σ the electron spin, and η represents an infinitesimal, real, positive number. The sum runs over the band index i, as well as the Bloch wave vector $\boldsymbol{k}$. To get the many-body excitation energies $E^{\sigma}_{i\boldsymbol{q}}$, we need to solve the quasi-particle equation

$$\left[-\frac{1}{2}\nabla^2+v_{\mathrm{ext}}(\mathbf{r})+v_{\mathrm{H}}(\mathbf{r})\right]\Psi^{\sigma}_{i\boldsymbol{k}}(\boldsymbol{r})+\int\Sigma^{\sigma}(\boldsymbol{r},\boldsymbol{r}';E^{\sigma}_{i\boldsymbol{k}})\;\Psi^{\sigma}_{i\boldsymbol{k}}(\boldsymbol{r}')\;d\boldsymbol{r}'=E^{\sigma}_{i\boldsymbol{k}}\;\Psi^{\sigma}_{i\boldsymbol{k}}(\boldsymbol{r}), \tag{2.2}$$

with the external potential $v_{\mathrm{ext}}(\mathbf{r})$ originating from the nuclei, the Hartree potential $v_{\mathrm{H}}(\mathbf{r})$, and the self-energy $\Sigma^{\sigma}(\boldsymbol{r},\boldsymbol{r}';\epsilon^{\sigma}_{i\boldsymbol{k}})$. The self-energy can be expanded in terms of the dynamically screened Coulomb interaction $W(\boldsymbol{r},\boldsymbol{r}';\omega)$. In the GW approximation [6], only the first-order term of this expansion is taken. This results in the following expression for the self-energy

$$\Sigma^{\sigma}(\boldsymbol{r},\boldsymbol{r}';\omega)=\frac{i}{2\pi}\int G^{\sigma}(\boldsymbol{r},\boldsymbol{r}';\omega+\omega')\;W(\boldsymbol{r},\boldsymbol{r}';\omega)\;e^{i\omega'\eta}\;d\omega'. \tag{2.3}$$

The screened Coulomb potential can be expressed as a Dyson-like equation:

$$W(\boldsymbol{r},\boldsymbol{r}';\omega)=v(\boldsymbol{r},\boldsymbol{r}')\;+\iint v(\boldsymbol{r},\boldsymbol{r}_1)P(\boldsymbol{r}_1,\boldsymbol{r}_2;\omega)\;W(\boldsymbol{r}_2,\boldsymbol{r}';\omega)\;d\boldsymbol{r}_1\;d\boldsymbol{r}_2, \tag{2.4}$$

where $v(\boldsymbol{r},\boldsymbol{r}')$ is the bare Coulomb potential:

$$v(\boldsymbol{r},\boldsymbol{r}')=\frac{1}{|\boldsymbol{r}-\boldsymbol{r}'|}, \tag{2.5}$$

screened by the polarizability $P(\boldsymbol{r},\boldsymbol{r}';\omega)$. The polarizability in the commonly used random-phase approximation (RPA) [24] takes the form

$$P(\boldsymbol{r},\boldsymbol{r}';\omega)=-\frac{i}{2\pi}\sum_{\sigma}\int G^{\sigma}(\boldsymbol{r},\boldsymbol{r}';\omega+\omega')\;G^{\sigma}(\boldsymbol{r},\boldsymbol{r}';\omega')\;e^{i\omega'\eta}\;d\omega'. \tag{2.6}$$

The Eqs. 2.1-2.6 form a loop, that, in theory, needs to be solved self-consistently. Since this is computationally quite expensive, in practice these equations are often solved only once, taking quantities calculated on KS DFT level as a starting point. This flavor of GW is called one-shot GW or G_0W_0 and is discussed in the following section.

2.1.1. G_0W_0

In the G_0W_0 approximation, the self energy is calculated similar to Eq. 2.3, but from the non-interacting single particle Green's function G_0 instead of the interacting single particle Green's function G, as shown here:

$$\Sigma^\sigma(\boldsymbol{r},\boldsymbol{r}';\omega) = \frac{i}{2\pi}\int G_0^\sigma(\boldsymbol{r},\boldsymbol{r}';\omega+\omega')\; W_0(\boldsymbol{r},\boldsymbol{r}';\omega)\; e^{i\omega'\eta}\; d\omega'. \tag{2.7}$$

The non-interacting single particle Green's function

$$G_0^\sigma(\boldsymbol{r},\boldsymbol{r}';\omega) = \sum_{i\boldsymbol{k}} \frac{\psi_{i\boldsymbol{k}}^\sigma(\boldsymbol{r})\;[\psi_{i\boldsymbol{k}}^\sigma(\boldsymbol{r}')]^*}{\omega - \epsilon_{i\boldsymbol{k}}^\sigma + i\;\eta\;\mathrm{sgn}(\epsilon_\mathrm{F} - \epsilon_{i\boldsymbol{k}}^\sigma)}. \tag{2.8}$$

is computed from the Kohn-Sham states $\psi_{i\boldsymbol{k}}^\sigma(\boldsymbol{r})$ instead of the actual quasi particle wave functions $\Psi_{i\boldsymbol{k}}^\sigma(\boldsymbol{r})$. Here, ϵ_F is the Fermi energy and $\epsilon_{i\boldsymbol{k}}^\sigma$ are the Kohn-Sham energies. We formulate W_0 in terms of the dielectric function $\varepsilon(\boldsymbol{r},\boldsymbol{r}';\omega)$

$$W_0(\boldsymbol{r},\boldsymbol{r}';\omega) = \int \varepsilon^{-1}(\boldsymbol{r},\boldsymbol{r}_1;\omega)\; v(\boldsymbol{r}_1,\boldsymbol{r}')\; d\boldsymbol{r}_1. \tag{2.9}$$

The dielectric function

$$\varepsilon(\boldsymbol{r},\boldsymbol{r}';\omega) = \delta(\boldsymbol{r},\boldsymbol{r}') - \int v(\boldsymbol{r},\boldsymbol{r}_1)\; P_0(\boldsymbol{r}_1,\boldsymbol{r}';\omega)\; d\boldsymbol{r}_1 \tag{2.10}$$

depends on the polarizability $P_0(\boldsymbol{r},\boldsymbol{r}';\omega)$. The Polarizability is calculated as described in Eq. 2.6, but replacing G with G_0, which allows us to carry out the integration over ω':

$$P_0(\boldsymbol{r},\boldsymbol{r}';\omega) = -\sum_\sigma \frac{i}{2\pi}\int G_0^\sigma(\boldsymbol{r},\boldsymbol{r}';\omega+\omega')\; G_0^\sigma(\boldsymbol{r}',\boldsymbol{r};\omega')\; e^{i\omega'\eta}\; d\omega' \tag{2.11}$$

$$= -\sum_\sigma \frac{i}{2\pi}\int \sum_{i\boldsymbol{k}} \frac{\psi_{i\boldsymbol{k}}^\sigma(\boldsymbol{r})\;[\psi_{i\boldsymbol{k}}^\sigma(\boldsymbol{r}')]^*}{\omega+\omega'-\epsilon_{i\boldsymbol{k}}^\sigma + i\,\eta\,\mathrm{sgn}(\epsilon_\mathrm{F}-\epsilon_{i\boldsymbol{k}}^\sigma)} \times \sum_{j\boldsymbol{k}'} \frac{\psi_{j\boldsymbol{k}'}^\sigma(\boldsymbol{r}')\;[\psi_{j\boldsymbol{k}'}^\sigma(\boldsymbol{r})]^*}{\omega'-\epsilon_{j\boldsymbol{k}'}^\sigma + i\,\eta\,\mathrm{sgn}(\epsilon_\mathrm{F}-\epsilon_{j\boldsymbol{k}'}^\sigma)}\; e^{i\omega'\eta}\; d\omega' \tag{2.12}$$

$$= -\sum_\sigma \sum_{ij} \sum_{\boldsymbol{k}\boldsymbol{k}'} f_{i\boldsymbol{k}}^\sigma(1-f_{j\boldsymbol{k}'}^\sigma)\; \psi_{i\boldsymbol{k}}^\sigma(\boldsymbol{r})\;\; [\psi_{j\boldsymbol{k}'}^\sigma(\boldsymbol{r})]^*\; \psi_{j\boldsymbol{k}'}^\sigma(\boldsymbol{r}')\; [\psi_{i\boldsymbol{k}}^\sigma(\boldsymbol{r}')]^* \times \left(\frac{1}{\omega+\epsilon_{i\boldsymbol{k}}^\sigma-\epsilon_{j\boldsymbol{k}'}^\sigma+i\,\eta} - \frac{1}{\omega-\epsilon_{i\boldsymbol{k}}^\sigma+\epsilon_{j\boldsymbol{k}'}^\sigma-i\,\eta}\right). \tag{2.13}$$

$f_{i\boldsymbol{k}}^\sigma$ represents the occupation number of the state specified by $\boldsymbol{ik}$ and σ. In the following, we omit the spin dependency and therefore can replace the sum over σ by a factor of 2, as shown here for the polarizability:

$$P_0(\boldsymbol{r},\boldsymbol{r}';\omega) = -2\sum_{ij}\sum_{\boldsymbol{k}\boldsymbol{k}'} f_{i\boldsymbol{k}}(1-f_{j\boldsymbol{k}'})\; \psi_{i\boldsymbol{k}}(\boldsymbol{r})\;\; [\psi_{j\boldsymbol{k}'}(\boldsymbol{r})]^*\; \psi_{j\boldsymbol{k}'}(\boldsymbol{r}')\; [\psi_{i\boldsymbol{k}}(\boldsymbol{r}')]^* \times \left(\frac{1}{\omega+\epsilon_{i\boldsymbol{k}}-\epsilon_{j\boldsymbol{k}'}+i\,\eta} - \frac{1}{\omega-\epsilon_{i\boldsymbol{k}}+\epsilon_{j\boldsymbol{k}'}-i\,\eta}\right). \tag{2.14}$$

In order to implement these equations, we rewrite them in matrix form. To do so, we employ basis sets. We distinguish between single-particle and two-particle functions. Single-particle functions, as the wave functions themselves, are expanded in the augmented planewaves plus local orbitals basis set described in the following section. Two-particle functions, such as the polarizability, are represented in the mixed-product basis, which we will discuss in more detail in section 2.2.3.

2.2. Linearized augmented planewaves plus local orbitals method

The linearized augmented planewave (LAPW) plus local orbitals (LO) method is known to be the gold standard for DFT calcualations. The main concept of the method is to divide the unit cell into different parts. The so-called muffin-tin (MT) spheres are centered at the atomic positions. The size of these spheres is adjustable, with the restriction that the spheres do not overlap. The region in between the MT spheres is called the interstitial region (I). This partitioning of space, first introduced in 1937 by Slater [25], is useful because wave functions close to the nuclei adapt atomic-like behavior and therefore can be described by atomic-like basis function, whereas wave functions in the I region behave much more smoothly and can be efficiently described by plane waves. In the following, a brief introduction to the different kinds of basis functions used in the linearized augmented planewaves plus local orbitals method is given.

2.2.1. Linearized augmented planewave method

The use of augmented planewaves (APW) was first proposed by Slater [25] in combination with the partition of the unit cell. The APWs he proposed are energy-dependent, making his method non-linear in energy. In 1975, Anderson developed the more practical linearized augmented planewave method [26]. He removed the non-linearity in energy by introducing energy parameters. For simplicity, we call these APW basis functions with fixed energy parameters linearized augmented planewaves (LAPW).
An example for LAPWs is shown here:

$$\phi_{\boldsymbol{G}+\boldsymbol{k}}(\boldsymbol{r}) = \begin{cases} \sum_{lm} A_{lm\alpha}^{\boldsymbol{G}+\boldsymbol{k}}\, u_{l\alpha}(r_\alpha; E_{l\alpha})\, Y_{lm}(\hat{\boldsymbol{r}}_\alpha), & r_\alpha \leq R_{MT} \\ \\ \frac{1}{\sqrt{\Omega}} e^{i(\boldsymbol{G}+\boldsymbol{k})\boldsymbol{r}}, & \boldsymbol{r} \in I. \end{cases} \tag{2.15}$$

In the I region, LAPWs take the form of planewaves that depend on the reciprocal lattice vectors $\boldsymbol{G}$ and are normalized by the unit cell volume Ω. The atomic-like functions in the MTs consist of a linear combination of radial functions $u_{l\alpha}(r_\alpha; E_{l\alpha})$ multiplied with spherical harmonics $Y_{lm}(\hat{\boldsymbol{r}}_\alpha)$. The matching coefficients $A_{lm\alpha}^{\boldsymbol{G}+\boldsymbol{k}}$ are chosen so that the LAPW are continuous at the MT-boundary. The radial functions themselves are solutions of the radial Schrödinger equation

$$\left[-\frac{1}{2}\frac{d^2}{dr^2} + \frac{l(l+1)}{2r^2} + v_0(r) - E_{l\alpha} \right] r u_{l\alpha}(r; E_{l\alpha}) = 0, \tag{2.16}$$

where l is the azimuthal quantum number, $v_0(r)$ is the spherical average of the KS-potential $v_{\mathrm{KS}}(\boldsymbol{r})$. The radial Schrödinger equation contains the energy parameters mentioned in the beginning. This so-called linearization energy $E_{l\alpha}$ can not be known. Hence, we introduce an error by estimating $E_{l\alpha}$. This linearization error can be reduced, by adding the first-order energy-derivative of the radial function $\dot{u}_{l\alpha}(r_\alpha; E_{l\alpha})$ to the linear combination, resulting in the following example of an LAPW

$$\phi_{\boldsymbol{G}+\boldsymbol{k}}(\boldsymbol{r}) = \begin{cases} \sum_{lm} [A_{lm\alpha}^{\boldsymbol{G}+\boldsymbol{k}} u_{l\alpha}(r_\alpha; E_{l\alpha}) + B_{lm\alpha}^{\boldsymbol{G}+\boldsymbol{k}} \dot{u}_{l\alpha}(r_\alpha; E_{l\alpha})] Y_{lm}(\hat{\boldsymbol{r}}_\alpha), & r_\alpha \leq R_{MT} \\ \\ \frac{1}{\sqrt{\Omega}} e^{i(\boldsymbol{G}+\boldsymbol{k})\boldsymbol{r}}, & \boldsymbol{r} \in I. \end{cases} \tag{2.17}$$

This addition introduces new matching coefficients $B_{lm\alpha}^{\boldsymbol{G}+\boldsymbol{k}}$. These are chosen, so that the LAPW is smooth at the MT-boundary. As a a supplement to LAPWs, local orbitals

(LO) can be used to further reduce the linearization error by adding high-order energy derivatives. Said LOs are described in the next section.

2.2.2. Local orbitals

LOs can be used in addition to APW basis functions to reduce the linearization error [27]. Similar to LAPWs they take the form of atom-like orbitals in one specific MT sphere α, but vanish in the I region and all other MT spheres, as shown here:

$$\phi_\mu(\boldsymbol{r}) = \begin{cases} \delta_{\alpha\alpha_\mu}\delta_{ll_\mu}\delta_{mm_\mu}[a_\mu u_{l\alpha}(r_\alpha; E_{l\alpha}) + b_\mu \dot{u}_{l\alpha}(r_\alpha; E_{l\alpha})]Y_{lm}(\hat{r}_\alpha), & r_\alpha \leq R_{MT} \\ 0, & \boldsymbol{r} \in I. \end{cases} \tag{2.18}$$

The coefficients a_μ and b_μ are chosen so that the LO vanishes at the MT boundary $\phi_\mu(\boldsymbol{R}_{\mathrm{MT}}) = 0$ and it is normalized, $\int_\Omega |\phi_\mu(\boldsymbol{r})|^2 d\boldsymbol{r} = 1$. The linearization error can be further reduced by including higher order energy derivatives of the radial functions. This can be done by adding an additional LO as defined in Eq. 2.18, but replacing $\dot{u}_{l\alpha}(r_\alpha; E_{l\alpha})$ with e.g. $\ddot{u}_{l\alpha}(r_\alpha; E_{l\alpha})$. Likewise, combinations of $\dot{u}_{l\alpha}(r_\alpha; E_{l\alpha})$ and $\ddot{u}_{l\alpha}(r_\alpha; E_{l\alpha})$ or a combination with any other energy derivative of higher order are possible. Alternatively, a higher order derivative can be added as an additional term in the linear combination. This results in the following expression for an LO

$$\phi_\mu(\boldsymbol{r}) = \begin{cases} \delta_{\alpha\alpha_\mu}\delta_{ll_\mu}\delta_{mm_\mu}[a_\mu u_{l\alpha}(r_\alpha; E_{l\alpha}) + b_\mu \dot{u}_{l\alpha}(r_\alpha; E_{l\alpha}) + c_\mu \ddot{u}_{l\alpha}(r_\alpha; E_{l\alpha})]Y_{lm}(\hat{r}_\alpha), & r_\alpha \leq R_{MT} \\ 0, & \boldsymbol{r} \in I. \end{cases} \tag{2.19}$$

The addition of $\ddot{u}_{l\alpha}(r_\alpha; E_{l\alpha})$ introduces extra matching coefficients c_μ, which are defined by requiring the LO to go to zero smoothly at the MT boundary.

Another way to utilize LOs is to describe semicore states [28]. To do so, LOs are build from two radial functions, where the first radial function at linearization energy $E^{\mathrm{v}}_{l\alpha}$ corresponds to the valence state with azimuthal quantum number l, and the second radial function at $E^{\mathrm{sc}}_{l\alpha}$ to a semicore state with in the same l-channel. Such an LO describing semicore states, is shown here:

$$\phi_\mu(\boldsymbol{r}) = \begin{cases} \delta_{\alpha\alpha_\mu}\delta_{ll_\mu}\delta_{mm_\mu}[a_\mu u_{l\alpha}(r_\alpha; E^{\mathrm{v}}_{l\alpha}) + b_\mu u_{l\alpha}(r_\alpha; E^{\mathrm{sc}}_{l\alpha})]Y_{lm}(\hat{r}_\alpha), & r_\alpha \leq R_{MT} \\ 0, & \boldsymbol{r} \in I. \end{cases} \tag{2.20}$$

As for the valence states, we do not know $E^{\mathrm{sc}}_{l\alpha}$ beforehand, and this may introduce a linearization error. In order to minimize this error, we add higher-order energy derivatives as described at the beginning of this section.

2.2.3. Mixed-product basis

In addition to an efficient basis for the description of single-particle functions, we need to define a suitable basis for the two-particle functions. The main requirement for this basis is the efficient description of wave-function products:

$$\psi_{i\boldsymbol{k}}(\boldsymbol{r}) \;\; [\psi_{j\boldsymbol{k}-\boldsymbol{q}}(\boldsymbol{r})]^* = \sum_I M^I_{ij}(\boldsymbol{k},\boldsymbol{q}) \; \chi^{\boldsymbol{q}}_I(\boldsymbol{r}), \tag{2.21}$$

with the expansion coefficients $M^I_{ij}(\boldsymbol{k},\boldsymbol{q})$ given by

$$M^I_{ij}(\boldsymbol{k},\boldsymbol{q}) = \int_\Omega \left[\chi^{\boldsymbol{q}}_I(\boldsymbol{r}) \; \psi_{j\boldsymbol{k}-\boldsymbol{q}}(\boldsymbol{r})\right]^* \psi_{i\boldsymbol{k}}(\boldsymbol{r}) \; d\boldsymbol{r}. \tag{2.22}$$

It is useful to employ the same distinction of MTs and I region as before for the (L)APW+LO basis. Both requirements are fulfilled by the mixed-product basis (MPB) proposed by Kotani and Schilfgaarde [29]. The MPB functions exist either in one MT sphere or in the I region and vanish in the other regions, which simplifies the removal of linear dependencies. It is feasible to use planewaves as basis functions in the I region, since the product of two planewaves is still a planewave. However, we require an orthonormal basis, but these interstitial planewaves (IPW) do not satisfy orthogonality. To orthogonalize the IPW basis, we need to diagonalize the overlap matrix defined as

$$\mathbb{O}_{\boldsymbol{G}\boldsymbol{G}'} = \frac{1}{\Omega}\int_\Omega \theta_I(\boldsymbol{r}) e^{i(\boldsymbol{G}-\boldsymbol{G}')\boldsymbol{r}} \; d\boldsymbol{r} \equiv \frac{1}{\Omega} l_{\boldsymbol{G}-\boldsymbol{G}'}. \tag{2.23}$$

Here, $\theta_I(\boldsymbol{r})$ is a step function that is one in the I region and zero everywhere else. We are able to carry out the integral, which gives us an expression for $l_{\boldsymbol{G}}$:

$$l_{\boldsymbol{G}} = \begin{cases} \Omega - \sum_\alpha V^\alpha_{\mathrm{MT}} & \boldsymbol{G} = 0 \\ \\ -3\sum_\alpha V^\alpha_{\mathrm{MT}} \; e^{i\boldsymbol{G}\boldsymbol{r}_\alpha} \dfrac{sin(GR^\alpha_{\mathrm{MT}}) - GR^\alpha_{\mathrm{MT}} \; cos(GR^\alpha_{\mathrm{MT}})}{(GR^\alpha_{\mathrm{MT}})^3} & \boldsymbol{G} \neq 0, \end{cases} \tag{2.24}$$

with V^α_{MT} being the volume of the MT sphere α.
In order to diagonalize $\mathbb{O}_{\boldsymbol{G}\boldsymbol{G}'}$, we need to solve the eigenvalue problem

$$\sum_{\boldsymbol{G}'} \mathbb{O}_{\boldsymbol{G}\boldsymbol{G}'} \; S_{\boldsymbol{G}'i} = \lambda^I_i \; S_{\boldsymbol{G}'i}. \tag{2.25}$$

This results in our new orthonormal basis set $P^{\boldsymbol{q}}_i(\boldsymbol{r})$ for the I region

$$P^{\boldsymbol{q}}_i(\boldsymbol{r}) = \frac{1}{\sqrt{\Omega}} \sum_{\boldsymbol{G}} \frac{S_{\boldsymbol{G}'i}}{\sqrt{\lambda^I_i}} \; e^{i(\boldsymbol{G}+\boldsymbol{q})\boldsymbol{r}} \; \theta_I(\boldsymbol{r}). \tag{2.26}$$

In the MT spheres, we can make use of the fact that the product of two spherical harmonics can again be expanded in spherical harmonics using Clebsch-Gordan coefficients:

$$Y_{lm}(\hat{\boldsymbol{r}})\,Y_{l'm'}(\hat{\boldsymbol{r}}) = \sum_{L=|l-l'|}^{l+l'} C_{lm,l'm'}^{LM}\,Y_{LM}(\hat{\boldsymbol{r}}). \tag{2.27}$$

Therefore, it is useful to write the MPB function in the MT spheres as a product of spherical harmonics $Y_{LM}(\hat{\boldsymbol{r}})$ and radial product functions $U_{\alpha NL}(r)$:

$$\gamma_{\alpha NLM}(\boldsymbol{r}) = e^{i\boldsymbol{q}\cdot\boldsymbol{r}_\alpha}\,U_{\alpha NL}(r)\,Y_{LM}(\hat{\boldsymbol{r}}). \tag{2.28}$$

The computation of the radial part $U_{\alpha NL}(r)$ of the MPB function is more involved. To get $U_{\alpha NL}(r)$, we have to consider all products $u_{\alpha l}(r)u_{\alpha l'}(r)$ into that satisfy the condition $|l-l'| \leq L \leq l+l'$. In a next step, the overlap matrix between theses products is computed according to:

$$\mathbb{O}_{ll',l_1l_1'} = \int_0^{R_{\mathrm{MT}\alpha}} u_{\alpha l}(r)u_{\alpha l'}(r)u_{\alpha l_1}(r)u_{\alpha l_1'}(r)\,r^2\,dr. \tag{2.29}$$

The diagonalization of this overlap matrix, gives a set of eigenvalues λ_N^{MT} with corresponding eigenvectors $c_{ll',N}$. To avoid (near) linear dependencies, eigenvectors with small eigenvalues are removed from the basis set. When normalized, we receive the orthonormal radial MPB basis functions:

$$U_{\alpha NL}(r) = \sum_{ll'} c_{ll',N}\,u_{l\alpha}(r)\,u_{l'\alpha}(r). \tag{2.30}$$

Combined with the spherical harmonics and multiplied by the factor $e^{i\boldsymbol{q}\boldsymbol{r}}$ to take crystal symmetry into consideration, we get the full MPB functions for the MT regions defined as

$$\gamma_{\alpha NLM}^{\boldsymbol{q}}(\boldsymbol{r}) = e^{i\boldsymbol{q}\boldsymbol{r}}\,U_{\alpha NL}(r)\,Y_{LM}(\hat{\boldsymbol{r}}). \tag{2.31}$$

The orthonormal MPB set $\chi_I^{\boldsymbol{q}}(\boldsymbol{r})$ is given by the union of the MT and I basis sets:

$$\{\chi_I^{\boldsymbol{q}}(\boldsymbol{r})\} = \{\gamma_{\alpha NLM}^{\boldsymbol{q}}(\boldsymbol{r}), P_i^{\boldsymbol{q}}(\boldsymbol{r})\}. \tag{2.32}$$

2.3. Polarizability in the Mixed-Product-Basis

Using the MPB, we can rewrite properties needed in the *GW* approximation in a matrix representation. For now, we focus on deriving the matrix representation of the polarizability. The polarizability $P(\boldsymbol{r},\boldsymbol{r'};\omega)$ can be understood as the linear response of the electron density $\rho(\boldsymbol{r},\omega)$ to changes in the effective potential $v_{\text{eff}}(\boldsymbol{r'},\omega)$, as shown here:

$$P\left(\boldsymbol{r},\boldsymbol{r'};\omega\right) = \frac{\delta\rho(\boldsymbol{r},\omega)}{\delta v_{\text{eff}}(\boldsymbol{r'},\omega)}. \tag{2.33}$$

The electron density $\rho(\boldsymbol{r})$ can be represented in terms of single-particle wave functions $\psi_{i\boldsymbol{k}}(\boldsymbol{r})$,

$$\rho(\boldsymbol{r}) = 2\sum_{i\boldsymbol{k}} |\psi_{i\boldsymbol{k}}(\boldsymbol{r})|^2 = 2\sum_{i\boldsymbol{k}} \psi^*_{i\boldsymbol{k}}(\boldsymbol{r})\ \psi_{i\boldsymbol{k}}(\boldsymbol{r}). \tag{2.34}$$

Using this expression, we get the following representation of the polarizability

$$P\left(\boldsymbol{r},\boldsymbol{r'};\omega\right) = 2\sum_{i\boldsymbol{k}} \frac{\delta}{\delta v_{\text{eff}}(\boldsymbol{r'},\omega)} \Big\{ \psi^*_{i\boldsymbol{k}}(\boldsymbol{r})\ \psi_{i\boldsymbol{k}}(\boldsymbol{r}) \Big\}, \tag{2.35}$$

in terms of the wave functions. Employing the product and chain rules for functional derivatives we can rewrite Eq. 2.35 as:

$$P\left(\boldsymbol{r},\boldsymbol{r'};\omega\right) = 2\sum_{i\boldsymbol{k}} \left\{ \psi^*_{i\boldsymbol{k}}(\boldsymbol{r})\ \frac{\delta\psi_{i\boldsymbol{k}}(\boldsymbol{r})}{\delta v_{\text{eff}}(\boldsymbol{r'},\omega)} + \psi^*_{i\boldsymbol{k}}(\boldsymbol{r})\ \frac{\delta\psi_{i\boldsymbol{k}}(\boldsymbol{r})}{\delta v_{\text{eff}}(\boldsymbol{r'},-\omega)} \right\}. \tag{2.36}$$

In order to get the matrix form of the polarizability, we need to expand $P(\boldsymbol{r},\boldsymbol{r'};\omega)$ in a basis. As described earlier, it is beneficial to use MPB functions $\chi_I^{\boldsymbol{q}}(\boldsymbol{r})$ for the representation of two-particle functions. By applying the MPB expansion we get

$$\begin{aligned} P_{IJ}\left(\boldsymbol{q},\omega\right) &= \iint [\chi_I^{\boldsymbol{q}}(\boldsymbol{r})]^*\ P\left(\boldsymbol{r},\boldsymbol{r'};\omega\right)\ \chi_J^{\boldsymbol{q}}(\boldsymbol{r'})\ d^3r'\ d^3r \\ &= \iint [\chi_I^{\boldsymbol{q}}(\boldsymbol{r})]^*\ 2\sum_{i\boldsymbol{k}} \left\{ \psi^*_{i\boldsymbol{k}}(\boldsymbol{r})\ \frac{\delta\psi_{i\boldsymbol{k}}(\boldsymbol{r})}{\delta v_{\text{eff}}(\boldsymbol{r'},\omega)} + \psi^*_{i\boldsymbol{k}}(\boldsymbol{r})\ \frac{\delta\psi_{i\boldsymbol{k}}(\boldsymbol{r})}{\delta v_{\text{eff}}(\boldsymbol{r'},-\omega)} \right\} \chi_J^{\boldsymbol{q}}(\boldsymbol{r'})\ d^3r'\ d^3r \\ &= 2\sum_{i\boldsymbol{k}} \int [\chi_I^{\boldsymbol{q}}(\boldsymbol{r})]^*\ \psi^*_{i\boldsymbol{k}}(\boldsymbol{r})\ \int \left\{ \frac{\delta\psi_{i\boldsymbol{k}}(\boldsymbol{r})}{\delta v_{\text{eff}}(\boldsymbol{r'},\omega)}\ \chi_J^{\boldsymbol{q}}\left(\boldsymbol{r'}\right) + \frac{\delta\psi_{i\boldsymbol{k}}(\boldsymbol{r})}{\delta v_{\text{eff}}(\boldsymbol{r'},-\omega)}\ \chi_J^{\boldsymbol{q}}\left(\boldsymbol{r'}\right) \right\} d^3r'\ d^3r. \end{aligned} \tag{2.37}$$

This quite complex expression can be simplified by recognizing that part of Eq. 2.37 looks similar to the expression of a wave function $\psi_{i\boldsymbol{k}}(\boldsymbol{r})$ perturbed by a MPB function $\chi_I^{\boldsymbol{q}}(\boldsymbol{r})$. This expression is the response function

$$\psi^{(1)}_{i\boldsymbol{k},I\boldsymbol{q}}\left(\boldsymbol{r},\omega\right) = \int \frac{\delta\psi_{i\boldsymbol{k}}(\boldsymbol{r})}{\delta v_{\text{eff}}(\boldsymbol{r'},\omega)}\ \chi_I^{\boldsymbol{q}}\left(\boldsymbol{r'}\right)\ d^3r'. \tag{2.38}$$

The polarizability in terms of response functions then reads

$$P_{IJ}\left(\boldsymbol{q},\omega\right) = 2\sum_{\boldsymbol{k}i}^{occ} \langle \chi_I^{\boldsymbol{q}}\ \psi_{i\boldsymbol{k}} \mid \psi^{(1)}_{i\boldsymbol{k},J\boldsymbol{q}}\left(\omega\right) + \psi^{(1)}_{i\boldsymbol{k},J\boldsymbol{q}}\left(-\omega\right) \rangle. \tag{2.39}$$

The following section will focus on finding an expression for said response function.

2.4. Sternheimer equation

As described above, the polarizability can be represented in terms of response functions defined as:

$$\psi_{i\boldsymbol{k}}^{(1)}\left(\boldsymbol{r},\omega\right)=\int\frac{\delta\psi_{i\boldsymbol{k}}^{(0)}(\boldsymbol{r})}{\delta v_{\text{eff}}(\boldsymbol{r'},\omega)}\;\hat{H}^{(1)}\left(\boldsymbol{r'}\right)\;d^3r'. \tag{2.40}$$

It represents the linear response of the wave function $\psi_{i\boldsymbol{k}}^{(0)}(\boldsymbol{r})$ to perturbations $\hat{H}^{(1)}(\boldsymbol{r})$ of the effective potential $v_{\text{eff}}(\boldsymbol{r})$. In the MPB-representation of the polarizability these perturbations are parameterized by MPB-functions $\chi_I^{\boldsymbol{q}}(\mathbf{r})$. To get an expression for the linear response of the wave function, the Kohn-Sham (KS) equations

$$\hat{H}^{(0)}|\psi_{i\mathbf{k}}^{(0)}\rangle=\epsilon_{i\mathbf{k}}^{(0)}|\psi_{i\mathbf{k}}^{(0)}\rangle, \tag{2.41}$$

with

$$\hat{H}^{(0)}\left(\mathbf{r}\right)=-\frac{1}{2}\nabla^2+v_{\text{eff}}\left(\mathbf{r}\right), \tag{2.42}$$

are linearized with respect to changes in the effective potential. In the following, we show first-order perturbation theory as a way to linearize the KS equations. This theory is valid for time- and momentum-independent spherical perturbations. Depending on the complexity of the perturbation, more advanced perturbation theories need to be used. Rayleigh-Schrödinger perturbation theory for degenerate states is employed for non-spherical perturbations and time-dependent perturbation theory is required for frequency-dependent perturbations.

Applying a perturbation $\lambda\hat{H}^{(1)}$ results in the following equation

$$\hat{H}\left(\lambda\right)|\psi_{i\mathbf{k}}\left(\lambda\right)\rangle=\epsilon_{i\mathbf{k}}\left(\lambda\right)|\psi_{i\mathbf{k}}\left(\lambda\right)\rangle \tag{2.43}$$

with the perturbed Hamiltonian

$$\hat{H}\left(\lambda\right)=\hat{H}^{(0)}+\lambda\hat{H}^{(1)}, \tag{2.44}$$

the perturbed wave functions

$$\psi_{i\mathbf{k}}\left(\lambda\right)=\psi_{i\mathbf{k}}^{(0)}+\lambda\psi_{i\mathbf{k}}^{(1)}, \tag{2.45}$$

and the perturbed eigenvalues

$$\epsilon_{i\mathbf{k}}\left(\lambda\right)=\epsilon_{i\mathbf{k}}^{(0)}+\lambda\epsilon_{i\mathbf{k}}^{(1)}. \tag{2.46}$$

Here, λ stands for the strength of the perturbation, witch needs to be small. When the expressions for $\hat{H}$, $\psi_{i\mathbf{k}}(\boldsymbol{r})$, and $\epsilon_{i\mathbf{k}}$ are plugged back into the perturbed KS equations, and the terms not linear in λ are eliminated, one obtains the following equation:

$$\hat{H}^{(0)}|\psi_{i\mathbf{k}}^{(0)}\rangle+\lambda\hat{H}^{(0)}|\psi_{i\mathbf{k}}^{(1)}\rangle+\lambda\hat{H}^{(1)}|\psi_{i\mathbf{k}}^{(0)}\rangle=\epsilon_{i\mathbf{k}}^{(0)}|\psi_{i\mathbf{k}}^{(0)}\rangle+\lambda\epsilon_{i\mathbf{k}}^{(0)}|\psi_{i\mathbf{k}}^{(1)}\rangle+\lambda\epsilon_{i\mathbf{k}}^{(1)}|\psi_{i\mathbf{k}}^{(0)}\rangle. \tag{2.47}$$

In a next step, the unperturbed KS equations are subtracted from Eq. 2.47, and the resulting expression is divided by λ. The subsequent equation is called the Sternheimer equation

$$\hat{H}^{(0)}|\psi_{i\mathbf{k}}^{(1)}\rangle + \hat{H}^{(1)}|\psi_{i\mathbf{k}}^{(0)}\rangle = \epsilon_{i\mathbf{k}}|\psi_{i\mathbf{k}}^{(1)}\rangle + \epsilon_{i\mathbf{k}}^{(1)}|\psi_{i\mathbf{k}}^{(0)}\rangle. \tag{2.48}$$

By left-multiplying Eq. 2.48 with the unperturbed eigenstate $\psi_{i\mathbf{k}}^{(0)}(\boldsymbol{r})$, subtracting $\langle\psi_{i\mathbf{k}}^{(0)}|\hat{H}^{(1)}|\psi_{i\mathbf{k}}^{(1)}\rangle$, and using the orthonormality of $\psi_{i\mathbf{k}}^{(1)}(\boldsymbol{r})$, we obtain an expression for the linear change of the eigenvalue $\epsilon_{i\mathbf{k}}^{(1)}$

$$\epsilon_{i\mathbf{k}}^{(1)} = \langle\psi_{i\mathbf{k}}^{(0)}|\hat{H}^{(1)}|\psi_{i\mathbf{k}}^{(0)}\rangle. \tag{2.49}$$

In order to reach the initial goal of acquiring an expression for the response function (or linear change of the eigenstate), we make use of the resolution of the identity:

$$\hat{H}^{(1)}|\psi_{i\mathbf{k}}^{(0)}\rangle = \left(\sum_{j}|\psi_{j\mathbf{k}}^{(0)}\rangle\langle\psi_{j\mathbf{k}}^{(0)}|\right)\hat{H}^{(1)}|\psi_{i\mathbf{k}}^{(0)}\rangle. \tag{2.50}$$

The sum over states j can be split up into terms with $j \neq i$ and $j = i$

$$\hat{H}^{(1)}|\psi_{i\mathbf{k}}^{(0)}\rangle = \left(\sum_{j(\neq i)}|\psi_{j\mathbf{k}}^{(0)}\rangle\langle\psi_{j\mathbf{k}}^{(0)}|\right)\hat{H}^{(1)}|\psi_{i\mathbf{k}}^{(0)}\rangle + \epsilon_{i\mathbf{k}}^{(1)}|\psi_{i\mathbf{k}}^{(0)}\rangle. \tag{2.51}$$

By subtracting Eq. 2.49 we get

$$\left(\hat{H}^{(1)} - \epsilon_{i\mathbf{k}}^{(1)}\right)|\psi_{i\mathbf{k}}^{(0)}\rangle = \left(\sum_{j(\neq i)}|\psi_{j\mathbf{k}}^{(0)}\rangle\langle\psi_{j\mathbf{k}}^{(0)}|\right)\hat{H}^{(1)}|\psi_{i\mathbf{k}}^{(0)}\rangle. \tag{2.52}$$

Using Eq. 2.48, the term $(\hat{H}^{(1)} - \epsilon_{i\mathbf{k}}^{(1)})|\psi_{i\mathbf{k}}^{(0)}\rangle$ can be replaced by $(\hat{H}^{(0)} - \epsilon_{i\mathbf{k}}^{(0)})|\psi_{i\mathbf{k}}^{(1)}\rangle$ resulting in the equation

$$\left(\epsilon_{i\mathbf{k}}^{(0)} - \hat{H}^{(0)}\right)|\psi_{i\mathbf{k}}^{(1)}\rangle = \left(\sum_{j(\neq i)}|\psi_{j}^{(0)}\rangle\langle\psi_{j\mathbf{k}}^{(0)}|\right)\hat{H}^{(1)}|\psi_{i\mathbf{k}}^{(0)}\rangle. \tag{2.53}$$

After renaming the summation index to j' for convenience, we left-multiply Eq. 2.53 with an arbitrary unperturbed eigenstate $\psi_{j\mathbf{k}}^{(0)}(\boldsymbol{r})$ unequal to $\psi_{i\mathbf{k}}^{(0)}(\boldsymbol{r})$. Once more using the orthonormality of eigenstates and dividing by $(\epsilon_i - \epsilon_j)$ we then obtain

$$\langle\psi_{j\mathbf{k}}^{(0)}|\psi_{i\mathbf{k}}^{(1)}\rangle = \frac{\langle\psi_{j\mathbf{k}}^{(0)}|\hat{H}^{(1)}|\psi_{i\mathbf{k}}^{(0)}\rangle}{\left(\epsilon_{i}^{(0)} - \epsilon_{j}^{(0)}\right)}. \tag{2.54}$$

The above expression can be understood as the first order response component along $\psi_{j\mathbf{k}}^{(0)}(\boldsymbol{r})$. Therefore, the full expression for the response function reads

$$|\psi^{(1)}_{i\mathbf{k}}\rangle = \sum_{j(\neq i)}^{N} \frac{\langle\psi^{(0)}_{j\mathbf{k}}|\hat{H}^{(1)}|\psi^{(0)}_{i\mathbf{k}}\rangle}{(\epsilon^{(0)}_{i\mathbf{k}} - \epsilon^{(0)}_{j\mathbf{k}})}|\psi^{(0)}_{j\mathbf{k}}\rangle, \tag{2.55}$$

where the sum runs over an infinite number of eigenstates N. In order to get a frequency- and $\boldsymbol{q}$-dependent response function, we have to use time-dependent perturbation theory for degenerate states. Using this, leads to the following expression for the response function

$$|\psi^{(1)}_{i\boldsymbol{k},\boldsymbol{q}}(\omega)\rangle = \sum_{j(\neq i)}^{N} \frac{\langle\psi^{(0)}_{j\boldsymbol{k}+\boldsymbol{q}}|H^{(1)}_{\boldsymbol{q}}|\psi^{(0)}_{i\boldsymbol{k}}\rangle}{\epsilon^{(0)}_{i\boldsymbol{k}} - \epsilon^{(0)}_{j\boldsymbol{k}+\boldsymbol{q}} + \omega}\,|\psi^{(0)}_{j\boldsymbol{k}+\boldsymbol{q}}\rangle. \tag{2.56}$$

This expression is often implemented in practice, since it is a straightforward way to compute the response function. However, it is not possible to sum over an infinite number of KS states N, because N is limited by the number of basis functions N_{basis}. In other words, if Eq. 2.56 is used, only the part of the response that lies in the Hilbert space spanned by the basis functions is accounted for. In order to avoid a significant loss of accuracy, the Hilbert space and consequently the number of basis functions N_{basis} need to be increased with respect to the number of basis functions used for usual DFT calculations, leading to large computational cost. This problem is described comprehensively by Dmitrii Nabok and coworkers [10]. In the next section, we show how to overcome this issue by employing the incomplete-basis-set correction (IBC). Markus Betzinger and coworkers formulated this correction for the LAPW+LO method in 2012 [20]and applied it to the optimized-effective-potential approach. They later generalized it to accommodate non-spherical and frequency dependent perturbations and applied it to the polarizability [21, 22].

2.5. Incomplete-basis-set correction

As shown above, a sum over occupied and a large number of unoccupied states is needed to get an accurate description of the wave function response. Both occupied and unoccupied states are represented by the same (L)APW + LO basis set. The basis functions in this set are optimized to resemble occupied states, whereas they are in general not optimal to represent unoccupied states lying at higher energies. Therefore, such set is a poor choice for the wave function response. Another way to think about this is to imagine that it is possible to find a perturbation out of the MPB-functions $\chi^{\boldsymbol{q}}_I(\boldsymbol{r};\omega)$ that turns the perturbed wave function out of the Hilbert space spanned by the basis set. The common way to avoid this problem is to increase the Hilbert space by including more basis functions, resulting in high computational cost. In the following, we describe how to avoid this by not increasing the size of the spanned Hilbert space, but rotating the space in the same way as the wave functions when perturbed. This can be done by applying the perturbation directly to the basis functions. As a result we find two kinds of corrections to the usual expression of the response function, the basis-response and Pulay correction. To get expressions for these corrections, we follow the derivations of Betzinger and coworkers [20, 21, 22]. We start by applying a perturbation δv_{eff} to the basis representation of the wave function:

$$\frac{\delta\psi_{i\boldsymbol{k}}\left(\boldsymbol{r}\right)}{\delta v_{\text{eff}}(\boldsymbol{r'};\omega)} = \sum_{\boldsymbol{G}} \frac{\delta C_{\boldsymbol{k}\boldsymbol{G}i}}{\delta v_{\text{eff}}(\boldsymbol{r'};\omega)}\phi_{\boldsymbol{G}+\boldsymbol{k}}\left(\boldsymbol{r}\right) + C_{\boldsymbol{k}\boldsymbol{G}i}\frac{\delta\phi_{\boldsymbol{G}+\boldsymbol{k}}\left(\boldsymbol{r}\right)}{\delta v_{\text{eff}}(\boldsymbol{r'};\omega)}. \tag{2.57}$$

Note that from here on, to simplify the notation, we will omit the (0) marker for unperturbed properties. Perturbations of the effective potential can be parameterized by MPB functions. This allows us to derive an expression for the linear response of the wave function $\psi^{(1)}_{i\boldsymbol{k},I}\left(\boldsymbol{r};\omega\right)$ with respect to $\chi^{\boldsymbol{q}}_{I}$. We show in the following, that $\psi^{(1)}_{i\boldsymbol{k},I}\left(\boldsymbol{r};\omega\right)$ can be split into two terms:

$$\begin{align}
\psi^{(1)}_{i\boldsymbol{k},I}\left(\boldsymbol{r};\omega\right) &= \int \frac{\delta\psi_{i\boldsymbol{k}}\left(\boldsymbol{r}\right)}{\delta v_{\text{eff}}(\boldsymbol{r'};\omega)}\ \chi^{\boldsymbol{q}}_{I}\left(\boldsymbol{r};\omega\right)d^3r \tag{2.58}\\
&= \int \sum_{\boldsymbol{G}} \frac{\delta C_{\boldsymbol{k}\boldsymbol{G}i}}{\delta v_{\text{eff}}(\boldsymbol{r'};\omega)}\phi_{\boldsymbol{G}+\boldsymbol{k}}\left(\boldsymbol{r}\right)\ \chi^{\boldsymbol{q}}_{I}\left(\boldsymbol{r};\omega\right)d^3r \nonumber\\
&\quad + \sum_{\boldsymbol{G}} C_{\boldsymbol{k}\boldsymbol{G}i}\int \frac{\delta\phi_{\boldsymbol{G}+\boldsymbol{k}}\left(\boldsymbol{r}\right)}{\delta v_{\text{eff}}(\boldsymbol{r'};\omega)}\ \chi^{\boldsymbol{q}}_{I}\left(\boldsymbol{r};\omega\right)d^3r \tag{2.59}\\
&= \int \sum_{\boldsymbol{G}} \frac{\delta C_{\boldsymbol{k}\boldsymbol{G}i}}{\delta v_{\text{eff}}(\boldsymbol{r'};\omega)}\phi_{\boldsymbol{G}+\boldsymbol{k}}\left(\boldsymbol{r}\right)\ \chi^{\boldsymbol{q}}_{I}\left(\boldsymbol{r};\omega\right)d^3r + \sum_{\boldsymbol{G}} C_{\boldsymbol{k}\boldsymbol{G}i}\ \phi^{(1)}_{\boldsymbol{G}+\boldsymbol{k},I\boldsymbol{q}}\left(\boldsymbol{r},\omega\right) \tag{2.60}\\
&= \hat{\psi}^{(1)}_{i\boldsymbol{k},I\boldsymbol{q}}\left(\boldsymbol{r};\omega\right) + \tilde{\psi}^{(1)}_{i\boldsymbol{k},I\boldsymbol{q}}\left(\boldsymbol{r};\omega\right). \tag{2.61}
\end{align}$$

The first term $\hat{\psi}_{i\boldsymbol{k},I\boldsymbol{q}}(\boldsymbol{r};\omega)$ arises from the response of the expansion coefficients $C_{\boldsymbol{k}\boldsymbol{G}i}$, the second term $\tilde{\psi}^{(1)}_{i\boldsymbol{k},I\boldsymbol{q}}(\boldsymbol{r};\omega)$ stems from the response of the basis functions $\phi_{\boldsymbol{G}+\boldsymbol{k}}(\boldsymbol{r})$ themselves. We will focus on the derivation of the basis function response in the next section, before discussing the expansion coefficient response in section 2.5.2.

2.5.1. Basis function response

The following derivation is done for the example of the LAPW basis. The derivation for LOs is not shown, but closely follows the steps presented here. As shown before, the MT part of the LAPW basis functions depends on the effective potential. On the contrary, the plane waves used in the I region result in a vanishing response of the basis function in the I region. To simplify the derivation, we will for now only consider time and momentum independent spherical perturbations parameterized by:

$$\chi^{\boldsymbol{q}}_{I}\left(\boldsymbol{r};\omega\right) = \chi_{I}\left(r\right), \tag{2.62}$$

where the index I stands for the combination of the angular momentum quantum number L, the magnetic quantum number M, and the atomic index α. Later in this chapter we will introduce more complex perturbations.

To get the expression for the basis function response, we apply the perturbation to the usual expression of an LAPW, Eq. 2.17, and linearize the result:

$$\phi^{(1)}_{\boldsymbol{G}+\boldsymbol{k},I}(\boldsymbol{r}) = \begin{cases} \sum_{lm} \Big[A^{\boldsymbol{G}+\boldsymbol{k}}_{lm\alpha}\ u^{(1)}_{l\alpha,I}(r_\alpha) + A^{(1)\ \boldsymbol{G}+\boldsymbol{k}}_{lm\alpha,I}\ u_{l\alpha}(r_\alpha) & \\ \quad + B^{\boldsymbol{G}+\boldsymbol{k}}_{lm}\ \dot{u}^{(1)}_{l\alpha,I}(r_\alpha) + B^{(1)\ \boldsymbol{G}+\boldsymbol{k}}_{lm\alpha,I}\ \dot{u}_{l\alpha}(r_\alpha) \Big] Y_{lm}(\hat{\boldsymbol{r}}_\alpha), & r_\alpha \leq R^\alpha_{\mathrm{MT}} \\ 0, & \boldsymbol{r} \in I. \end{cases} \tag{2.63}$$

As shown above, the perturbation has an influence on the radial functions as well as the matching coefficients. Due to the linearization, the linear combination contains only products of unperturbed matching coefficients with perturbed radial functions $u^{(1)}_{l\alpha,I}(r_\alpha)$ and products of perturbed matching coefficients $A^{(1)}_{lm\alpha,I}$ and $B^{(1)}_{lm\alpha,I}$ with unperturbed radial functions. Since we apply a spherical perturbation, the basis function response stays in the same lm-channel as the unperturbed basis function, leaving the spherical harmonics unchanged. This allows us to multiply the whole linear combination with the same spherical harmonic.

To get the response of the radial function to the perturbation, we need to linearize the radial SE. Similar to the derivation shown in section 2.4, this will result in a radial Sternheimer equation of the following form:

$$\Big[h_{l\alpha} - \epsilon_{l\alpha}\Big]\ r\ u^{(1)}_{l\alpha,I}(r) = \Big[\epsilon^{(1)}_{l\alpha,I} - \chi_I(r)\Big] r u_{l\alpha}(r), \tag{2.64}$$

with the energy shift:

$$\epsilon^{(1)}_{l\alpha,I} = \langle u_{l\alpha} | \chi_I | u_{l\alpha} \rangle. \tag{2.65}$$

The perturbed matching coefficients are chosen so that $\phi^{(1)}_{\boldsymbol{G}+\boldsymbol{k},I}(\boldsymbol{r})$ and its radial derivative continuously go to zero at the MT boundaries [20].

In a next step we allow for non-spherical perturbations which are parameterized by the following functions

$$\chi^{\boldsymbol{q}}_I(\boldsymbol{r};\omega) = \chi_I(\boldsymbol{r}) = \chi_I(r)\ Y_{LM}\left(\hat{\boldsymbol{r}}_\alpha\right), \tag{2.66}$$

Such a perturbation leads to response functions that have contributions in more than one lm-channel. This means that radial functions and spherical harmonics cannot be treated independently from each other. For convenience, we introduce a new combined quantity defined as

$$\tilde{u}_{lm\alpha}(\boldsymbol{r}) = u_{l\alpha}(r_\alpha)\ Y_{lm}(\hat{\boldsymbol{r}}_\alpha). \tag{2.67}$$

The basis function response then takes the form:

$$\phi^{(1)}_{\boldsymbol{G}+\boldsymbol{k},I}\left(\boldsymbol{r}\right) = \begin{cases} \sum_{lm} \left[A^{\boldsymbol{G}+\boldsymbol{k}}_{lm\alpha}\ \tilde{u}^{(1)}_{lm\alpha,I}(\boldsymbol{r}) + A^{(1)\,\boldsymbol{G}+\boldsymbol{k}}_{lm\alpha,I}\ \tilde{u}_{lm\alpha}(\boldsymbol{r}) \right. & \\ \qquad \left. + B^{\boldsymbol{G}+\boldsymbol{k}}_{lm}\ \tilde{\dot{u}}^{(1)}_{lm\alpha,I}(\boldsymbol{r}) + B^{(1)\,\boldsymbol{G}+\boldsymbol{k}}_{lm\alpha,I}\ \tilde{\dot{u}}_{lm\alpha}(\boldsymbol{r}) \right] & r_\alpha \leq R^{\alpha}_{\mathrm{MT}} \\ 0 & \boldsymbol{r} \in I. \end{cases} \tag{2.68}$$

To get the response of Eq. 2.67, we need to sum over all possible lm-channels:

$$\tilde{u}^{(1)}_{lm\alpha,I}(\boldsymbol{r}) = \sum_{l'm'} u^{(1)}_{lm\alpha,I,l'm'}(r)\ Y_{l'm'}(\hat{\boldsymbol{r}}_\alpha). \tag{2.69}$$

The expression for the new radial function response $u^{(1)}_{lm,I,l'm'\alpha}(r)$ now also depends on the magnetic quantum number m. To get $u^{(1)}_{lm\alpha,I,l'm'}(r)$, we need to linearize the radial SE using Rayleigh-Schrödinger perturbation theory for degenerate states. This results in an inhomogeneous ordinary differential equation (ODE) similar to Eq. 2.64,

$$\left[h_{l'\alpha} - \epsilon_{l\alpha}\right]\ r\ u^{(1)}_{lm\alpha,I,l'm'}(r) = \mathcal{G}^{l'm'}_{LM;lm}\ \left[\delta_{ll'}\ \epsilon^{(1)}_{l\alpha,I} - \chi_I(r)\right] r\ u_{l\alpha}(r), \tag{2.70}$$

with the Gaunt coefficients $\mathcal{G}^{l'm'}_{LM;lm}$ defined as

$$\mathcal{G}^{l'm'}_{LM;lm} = \int Y_{LM}(\hat{\boldsymbol{r}}_\alpha)\ \left[Y_{l'm'}(\hat{\boldsymbol{r}}_\alpha)\right]^*\ Y_{lm}(\hat{\boldsymbol{r}}_\alpha)\ d\Omega. \tag{2.71}$$

Since the Gaunt coefficients are different from zero only in cases were $l + l' + L$ is even and the conditions $|l - l'| \leq L \leq l + l'$ and $m - m' + m = 0$ are fulfilled, the number of times Eq. 2.70 needs to be solved, reduces drastically. The condition for the matching coefficients stays the same as for the spherical perturbation described earlier.

In order to get the response function for representing the polarizability, we have to allow for time dependent perturbations. These are parameterized by the MPB functions as defined in Section 2.2.3 multiplied with the time-dependents, as shown here:

$$\chi^{\boldsymbol{q}}_I\left(\boldsymbol{r};\omega\right) = \chi_I\left(\boldsymbol{r};\omega\right) = \chi_I\left(\boldsymbol{r}\right)\ e^{-i\omega t}. \tag{2.72}$$

The linear response of the KS wave function $\psi_{i\boldsymbol{k}}(\boldsymbol{r})$ takes the same time dependence, $e^{-i\omega t}$. As before, we can represent $\psi_{i\boldsymbol{k}}(\boldsymbol{r})$ in the LAPW basis, leading to a contribution containing the perturbed basis function $\phi^{(1)}_{\boldsymbol{G}+\boldsymbol{k},I}(\boldsymbol{r};\omega)$. It takes a form similar to the one shown in Eq. 2.68, but with the perturbed radial functions now depending on ω. In order to get these perturbed radial functions, we need to use time-dependent perturbation theory for degenerate states. This leads to another inhomogeneous ODE similar to Eqs. 2.64 and ´ 2.70:

$$\left[h_{l'\alpha} - \epsilon_{l\alpha} - \omega\right] r\, u^{(1)}_{lm,I,l'm'\alpha}(r;\omega) = \mathcal{G}^{l'm'}_{LM;lm} \left[\delta_{ll'}\, \epsilon^{(1)}_{l\alpha,I} - \chi_I(r)\right] r\, u_{l\alpha}(r). \tag{2.73}$$

The last generalization step that needs to be done is the generalization to momentum dependent perturbations parameterized by:

$$\chi_I^{\boldsymbol{q}}\left(\boldsymbol{r};\omega\right) = \chi_I\left(\boldsymbol{r};\omega\right)\; e^{-i\boldsymbol{q}\cdot\boldsymbol{R}}, \tag{2.74}$$

where $\boldsymbol{q}$ represents the momentum vector and $\boldsymbol{R}$ is the vector pointing to the MT-center. The wave function response and therefore the basis function response inherit the same phase factor $e^{-i\boldsymbol{q}\cdot\boldsymbol{R}}$. The basis function response including all generalizations described in this Subsection then reads

$$\phi^{(1)}_{\boldsymbol{G}+\boldsymbol{k},I}\left(\boldsymbol{r},\omega\right) = \begin{cases} e^{i(\boldsymbol{k}+\boldsymbol{q})\cdot\boldsymbol{R}}\, \sum_{lm} \Big[A^{\boldsymbol{G}+\boldsymbol{k}}_{lm\alpha}\; \tilde{u}^{(1)}_{lm\alpha,I}(\boldsymbol{r},\omega) + A^{(1)\,\boldsymbol{G}+\boldsymbol{k}}_{lm\alpha,I}\; \tilde{u}_{lm\alpha}(\boldsymbol{r},\omega) & \\ \quad + B^{\boldsymbol{G}+\boldsymbol{k}}_{lm}\; \tilde{\dot{u}}^{(1)}_{lm\alpha,I}(\boldsymbol{r},\omega) + B^{(1)\,\boldsymbol{G}+\boldsymbol{k}}_{lm\alpha,I}\; \tilde{\dot{u}}_{lm\alpha}(\boldsymbol{r},\omega) \Big] & r_\alpha \leq R^{\alpha}_{\mathrm{MT}} \\ 0 & \boldsymbol{r} \in I. \end{cases} \tag{2.75}$$

2.5.2. Expansion coefficient response

To derive an expression for the expansion coefficient response $\hat{\psi}^{(1)}_{i\boldsymbol{k},I\boldsymbol{q}}\left(\boldsymbol{r};\omega\right)$, we need to expand $\hat{\psi}^{(1)}_{i\boldsymbol{k},I\boldsymbol{q}}(\boldsymbol{r};\omega)$ in terms of the unperturbed eigenfunctions $\psi_{i\boldsymbol{k}+\boldsymbol{q}}(\boldsymbol{r})$:

$$\hat{\psi}^{(1)}_{i\boldsymbol{k},I\boldsymbol{q}}\left(\boldsymbol{r};\omega\right) = \sum_{j}^{N} \langle \psi_{j\boldsymbol{k}+\boldsymbol{q}} | \hat{\psi}^{(1)}_{i\boldsymbol{k},I\boldsymbol{q}}(\omega)\rangle\; \psi_{j\boldsymbol{k}+\boldsymbol{q}}\left(\boldsymbol{r}\right). \tag{2.76}$$

This is possible, because this part of the response lies completely in the Hilbert space spanned by the unperturbed eigenfunctions. The sum over the projection coefficients can be split in to parts, $j = i$ and $j \neq i$:

$$\hat{\psi}^{(1)}_{i\boldsymbol{k},I\boldsymbol{q}}\left(\boldsymbol{r};\omega\right) = \sum_{j(\neq i)}^{N} \langle \psi_{j\boldsymbol{k}+\boldsymbol{q}} | \hat{\psi}^{(1)}_{i\boldsymbol{k},I\boldsymbol{q}}(\omega)\rangle\; \psi_{j\boldsymbol{k}+\boldsymbol{q}}\left(\boldsymbol{r}\right) + \langle \psi_{i\boldsymbol{k}+\boldsymbol{q}} | \hat{\psi}^{(1)}_{i\boldsymbol{k},I\boldsymbol{q}}(\omega)\rangle\; \psi_{i\boldsymbol{k}+\boldsymbol{q}}\left(\boldsymbol{r}\right). \tag{2.77}$$

The projection coefficient for the case $j = i$ can be straightforwardly found by using the expression for the complete response function Eq. 2.58. This allows us to represent the projection coefficient in terms of the full response function and basis-function response we derived before, resulting in:

$$\langle \psi_{i\boldsymbol{k}+\boldsymbol{q}} | \hat{\psi}^{(1)}_{i\boldsymbol{k},I\boldsymbol{q}}(\omega)\rangle = \langle \psi_{i\boldsymbol{k}+\boldsymbol{q}} | \psi^{(1)}_{i\boldsymbol{k},I\boldsymbol{q}}(\omega)\rangle - \langle \psi_{i\boldsymbol{k}+\boldsymbol{q}} | \tilde{\psi}^{(1)}_{i\boldsymbol{k},I\boldsymbol{q}}(\omega)\rangle. \tag{2.78}$$

Using the normalization condition:

$$\int [\psi_{i\boldsymbol{k}+\boldsymbol{q}}]^* \ \psi^{(1)}_{i\boldsymbol{k},I\boldsymbol{q}}(\omega) d^3r = 0, \tag{2.79}$$

we can show that the term $\langle \psi_{i\boldsymbol{k}+\boldsymbol{q}} | \psi^{(1)}_{i\boldsymbol{k},I\boldsymbol{q}}(\omega)\rangle$ in Eq. 2.78 vanishes. Therefore, we can express the projection coefficient solely in terms of known quantities:

$$\langle \psi_{i\boldsymbol{k}+\boldsymbol{q}} | \hat{\psi}^{(1)}_{i\boldsymbol{k},I\boldsymbol{q}}(\omega)\rangle = -\langle \psi_{i\boldsymbol{k}+\boldsymbol{q}} | \tilde{\psi}^{(1)}_{i\boldsymbol{k},I\boldsymbol{q}}(\omega)\rangle. \tag{2.80}$$

To obtain the projection coefficient for the case $j \neq i$, we make use of the fact that the complete response function (Eq. 2.58) is the solution to the Sternheimer Equation Eq. 2.48 with the perturbation parameterized by $\chi^{\boldsymbol{q}}_I(\boldsymbol{r};\omega)$. By left multiplying Eq. 2.48 with $\psi^*_{j\boldsymbol{k}+\boldsymbol{q}}(\boldsymbol{r})$ and integrating over the whole space we get the equation

$$\big(\epsilon_{j\boldsymbol{k}+\boldsymbol{q}} - \epsilon_{i\boldsymbol{k}} - \omega\big)\langle \psi_{j\boldsymbol{k}+\boldsymbol{q}} | \hat{\psi}^{(1)}_{i\boldsymbol{k},I\boldsymbol{q}}(\omega)\rangle + \langle D_{j\boldsymbol{k}+\boldsymbol{q}} | \hat{\psi}^{(1)}_{i\boldsymbol{k},I\boldsymbol{q}}(\omega)\rangle = \langle \psi_{j\boldsymbol{k}+\boldsymbol{q}} | \epsilon^{(1)}_{i\boldsymbol{k},I} - \chi^{\boldsymbol{q}}_I(\omega) | \psi_{i\boldsymbol{k}}\rangle, \tag{2.81}$$

with

$$D_{j\boldsymbol{k}}(\boldsymbol{r}) = (H^0 - \epsilon_{j\boldsymbol{k}})\psi_{j\boldsymbol{k}}(\boldsymbol{r}). \tag{2.82}$$

At first glance, $D_{j\boldsymbol{k}}(\boldsymbol{r})$ seems to be always equal to zero, but if we consider the possibility that the computed eigenfunctions $\psi_{j\boldsymbol{k}}(\boldsymbol{r})$ deviate from the true eigenfunctions of the Hamiltonian H^0, we need to treat $D_{j\boldsymbol{k}}(\boldsymbol{r})$ as a non-zero contribution. By plugging Eq. 2.58 into Eq. 2.81 and using the normalization condition, $\langle \psi_{i\boldsymbol{k}} | \psi_{j\boldsymbol{k}}\rangle = 0$, as well as $\langle D_{j\boldsymbol{k}+\boldsymbol{q}} | \psi^{(1)}_{i\boldsymbol{k},I\boldsymbol{q}}(\omega)\rangle = 0$, we obtain

$$\big(\epsilon_{j\boldsymbol{k}+\boldsymbol{q}} - \epsilon_{i\boldsymbol{k}} - \omega\big)\langle \psi_{j\boldsymbol{k}+\boldsymbol{q}} | \hat{\psi}^{(1)}_{i\boldsymbol{k},I\boldsymbol{q}}(\omega) + \tilde{\psi}^{(1)}_{i\boldsymbol{k},I\boldsymbol{q}}(\omega)\rangle + \langle D_{j\boldsymbol{k}+\boldsymbol{q}} | \tilde{\psi}^{(1)}_{i\boldsymbol{k},I\boldsymbol{q}}(\omega)\rangle = -\langle \psi_{j\boldsymbol{k}+\boldsymbol{q}} | \chi^{\boldsymbol{q}}_I | \psi_{i\boldsymbol{k}}\rangle. \tag{2.83}$$

Rearranging Eq. 2.83, leads to the projection coefficient $\langle \psi_{j\boldsymbol{k}} | \hat{\psi}^{(1)}_{i\boldsymbol{k},I\boldsymbol{q}}\rangle$. By plugging both expressions for the projection coefficients back in to Eq. 2.77, we obtain the full expression for the expansion coefficient response function:

$$\begin{aligned}\hat{\psi}^{(1)}_{i\boldsymbol{k},I\boldsymbol{q}}(\boldsymbol{r};\omega) = &\sum_{j(\neq i)}^{N} \left[\frac{\langle \psi_{j\boldsymbol{k}+\boldsymbol{q}} | \chi^{\boldsymbol{q}}_I | \psi_{i\boldsymbol{k}}\rangle}{\epsilon_{i\boldsymbol{k}} - \epsilon_{j\boldsymbol{k}+\boldsymbol{q}} + \omega} + \frac{\langle D_{j\boldsymbol{k}+\boldsymbol{q}} | \tilde{\psi}^{(1)}_{i\boldsymbol{k},I\boldsymbol{q}}(\omega)\rangle}{\epsilon_{i\boldsymbol{k}} - \epsilon_{j\boldsymbol{k}+\boldsymbol{q}} + \omega} \right] \psi_{j\boldsymbol{k}+\boldsymbol{q}}(\boldsymbol{r}) \\ &- \sum_{j}^{N} \langle \psi_{j\boldsymbol{k}+\boldsymbol{q}} | \tilde{\psi}^{(1)}_{i\boldsymbol{k},I\boldsymbol{q}}(\omega)\rangle \ \psi_{j\boldsymbol{k}+\boldsymbol{q}}(\boldsymbol{r}).\end{aligned} \tag{2.84}$$

2.5.3. Corrected response function

Now that we have an expression for both the basis function response and the expansion coefficient response, we can combine them to get the full response function. To do so, we have to rewrite the last term of the expansion coefficient response in the following way:

$$\sum_{j}^{N} \langle \psi_{j\boldsymbol{k}+\boldsymbol{q}} | \tilde{\psi}^{(1)}_{i\boldsymbol{k},I\boldsymbol{q}}(\omega) \rangle \; \psi_{j\boldsymbol{k}+\boldsymbol{q}}\left(\boldsymbol{r}\right) = \sum_{j}^{N} \int d^3r' \; \psi_{j\boldsymbol{k}+\boldsymbol{q}}\left(\boldsymbol{r}\right) \; \psi_{j\boldsymbol{k}+\boldsymbol{q}}\left(\boldsymbol{r'}\right) \; \tilde{\psi}^{(1)}_{i\boldsymbol{k},I\boldsymbol{q}}\left(\boldsymbol{r'};\omega\right). \tag{2.85}$$

In addition, we can represent the basis function response as an integral over a delta function:

$$\tilde{\psi}^{(1)}_{i\boldsymbol{k},I\boldsymbol{q}}\left(\boldsymbol{r};\omega\right) = \int d^3r' \; \delta(\boldsymbol{r}-\boldsymbol{r'}) \; \tilde{\psi}^{(1)}_{i\boldsymbol{k},I\boldsymbol{q}}\left(\boldsymbol{r'};\omega\right). \tag{2.86}$$

Using both of these expression combined with Eq. 2.84, leads to the full response function:

$$\begin{aligned} \psi^{(1)}_{i\boldsymbol{k},I\boldsymbol{q}}\left(\boldsymbol{r};\omega\right) = & \sum_{j(\neq i)}^{N} \left[\frac{\langle \psi_{j\boldsymbol{k}+\boldsymbol{q}} | \chi^{\boldsymbol{q}}_{I} | \psi_{i\boldsymbol{k}} \rangle}{\epsilon_{i\boldsymbol{k}} - \epsilon_{j\boldsymbol{k}+\boldsymbol{q}} + \omega} + \frac{\langle D_{j\boldsymbol{k}+\boldsymbol{q}} | \tilde{\psi}^{(1)}_{i\boldsymbol{k},I\boldsymbol{q}}(\omega) \rangle}{\epsilon_{i\boldsymbol{k}} - \epsilon_{j\boldsymbol{k}+\boldsymbol{q}} + \omega} \right] \psi_{j\boldsymbol{k}+\boldsymbol{q}}\left(\boldsymbol{r}\right) \\ & + \sum_{j}^{N} \int d^3r' \left[\delta(\boldsymbol{r}-\boldsymbol{r'}) - \psi_{j\boldsymbol{k}+\boldsymbol{q}}\left(\boldsymbol{r}\right) \; \psi^{*}_{j\boldsymbol{k}+\boldsymbol{q}}\left(\boldsymbol{r'}\right) \right] \tilde{\psi}^{(1)}_{i\boldsymbol{k},I\boldsymbol{q}}\left(\boldsymbol{r'};\omega\right). \end{aligned} \tag{2.87}$$

It consists of three terms. The first term is the expression that we have got from perturbation theory, Eq. 2.56. This term is the only one giving a non-zero contribution when going to the limit of a summation over an infinite number of states. In the following, we call it sum-over-states (SOS) term. The second term,

$$\psi^{\text{Pulay}}_{i\boldsymbol{k},I\boldsymbol{q}}\left(\boldsymbol{r};\omega\right) = \sum_{j(\neq i)}^{N} \frac{\langle \psi_{j\boldsymbol{k}+\boldsymbol{q}} | H - \epsilon_{j\boldsymbol{k}+\boldsymbol{q}} | \tilde{\psi}^{(1)}_{i\boldsymbol{k},I\boldsymbol{q}}(\omega) \rangle}{\epsilon_{i\boldsymbol{k}} - \epsilon_{j\boldsymbol{k}+\boldsymbol{q}} + \omega} \; \psi_{j\boldsymbol{k}+\boldsymbol{q}}\left(\boldsymbol{r}\right), \tag{2.88}$$

introduces a usually small correction to the SOS term, which is necessary, since the computed wave functions are not pointwise exact solutions to the SE. It is called the Pulay term because this correction is similar to the Pulay term in atomic force calculations. The last term,

$$\psi^{\text{BR}}_{i\boldsymbol{k},I\boldsymbol{q}}\left(\boldsymbol{r};\omega\right) = \sum_{j}^{N} \int d^3r' \left[\delta(\boldsymbol{r}-\boldsymbol{r'}) - \psi_{j\boldsymbol{k}+\boldsymbol{q}}\left(\boldsymbol{r}\right) \; \psi^{*}_{j\boldsymbol{k}+\boldsymbol{q}}\left(\boldsymbol{r'}\right) \right] \tilde{\psi}^{(1)}_{i\boldsymbol{k},I\boldsymbol{q}}\left(\boldsymbol{r'};\omega\right), \tag{2.89}$$

is called the basis-response (BR) term. This term corrects for the incompleteness of the basis introduced by the truncation of the infinite sum in the SOS term. It has been shown that this term accounts for most of the correction. By plugging the full response function into the polarizability Eq. 2.39, we can split the polarizability in a similar manner. We obtain the well-known SOS term, corresponding to the usual Adler-Wiser expression, a Pulay-correction term, and a BR correction term. The full expression for the corrected polarizability can be found in Appendix A.

3. Implementation

3.1. Polarizability in the LAPW method

In G_0W_0, the IBC can be applied directly to the polarizability, as shown in Appendix A. However, in the G_0W_0 implementation of the `exciting` code [30], a trick is used that reduces the required memory and computational cost by skipping the step of explicitly computing the polarizability. The SOS part of the polarizability,

$$P^{(\mathrm{SOS})}_{IJ}(\boldsymbol{q},\omega) = 2\sum_{\boldsymbol{k}}^{BZ}\sum_{i}^{\text{occ.}}\sum_{j}^{\text{unocc.}} \langle\chi_J^{\boldsymbol{q}}\ \psi_{i\boldsymbol{k}}|\psi_{j\boldsymbol{k}+\boldsymbol{q}}\rangle\ \langle\psi_{j\boldsymbol{k}+\boldsymbol{q}}|\psi_{i\boldsymbol{k}}\ \chi_J^{\boldsymbol{q}}\rangle \left\{\frac{1}{\epsilon_{i\boldsymbol{k}}-\epsilon_{j\boldsymbol{k}+\boldsymbol{q}}+\omega}+\frac{1}{\epsilon_{i\boldsymbol{k}}-\epsilon_{j\boldsymbol{k}+\boldsymbol{q}}-\omega}\right\}, \tag{3.1}$$

can be simplified by using the expression for the MPB expansion coefficients (Eq. 2.22) as well as the occupation number $f_{i\boldsymbol{k}}$ of state $i\boldsymbol{k}$:

$$P^{(\mathrm{SOS})}_{IJ}(\boldsymbol{q},\omega) = 2\sum_{\boldsymbol{k}}^{BZ}\sum_{ij} f_{i\boldsymbol{k}}\ (1-f_{j\boldsymbol{k}})\ \left[M^I_{ij}(\boldsymbol{k},\boldsymbol{q})\right]^*\ M^J_{ij}(\boldsymbol{k},\boldsymbol{q})\ \left\{\frac{1}{\epsilon_{i\boldsymbol{k}}-\epsilon_{j\boldsymbol{k}+\boldsymbol{q}}+\omega}+\frac{1}{\epsilon_{i\boldsymbol{k}}-\epsilon_{j\boldsymbol{k}+\boldsymbol{q}}-\omega}\right\} \tag{3.2}$$

$$= \sum_{\boldsymbol{k}}^{BZ}\sum_{ij} F_{ij}(\boldsymbol{k},\boldsymbol{q};\omega)\ \left[M^I_{ij}(\boldsymbol{k},\boldsymbol{q})\right]^*\ M^J_{ij}(\boldsymbol{k},\boldsymbol{q}). \tag{3.3}$$

We then represent the dielectric function (Eq. 2.10) in the MPB:

$$\epsilon_{ij}(\boldsymbol{q},\boldsymbol{w}) = \delta_{IJ} - \sum_{lm} v^{\frac{1}{2}}_{Il}(\boldsymbol{q})\ P_{lm}(\boldsymbol{q},\omega)\ v^{\frac{1}{2}}_{mJ}(\boldsymbol{q}), \tag{3.4}$$

and plug the Eq. 3.2 into the matrix representation of the dielectric function

$$\begin{aligned}\epsilon_{IJ}(\boldsymbol{q},\omega) &= \delta_{ij} - \sum_{ls} v^{\frac{1}{2}}_{il}(\boldsymbol{q})\ \left[\frac{1}{N_{\mathrm{c}}}\sum_{\boldsymbol{k}}\sum_{nm} F_{nm}(\boldsymbol{k},\boldsymbol{q};\omega) M^l_{nm}(\boldsymbol{k},\boldsymbol{q})[M^s_{nm}(\boldsymbol{k},\boldsymbol{q})]\right]\ v^{\frac{1}{2}}_{sj}(\boldsymbol{q}) \\ &= \delta_{ij} - \frac{1}{N_{\mathrm{c}}}\sum_{ls}\sum_{\boldsymbol{k}}\sum_{nm} F_{nm}(\boldsymbol{k},\boldsymbol{q};\omega) v^{\frac{1}{2}}_{il}(\boldsymbol{q}) M^l_{nm}(\boldsymbol{k},\boldsymbol{q})[M^s_{nm}(\boldsymbol{k},\boldsymbol{q})]^*\ v^{\frac{1}{2}}_{sj}(\boldsymbol{q}).\end{aligned} \tag{3.5}$$

By introducing a new set of expansion coefficients,

$$\tilde{M}^{i}_{nm}(\boldsymbol{k},\boldsymbol{q}) = \sum_{p} v^{\frac{1}{2}}_{ip}(\boldsymbol{q}) M^{p}_{nm}(\boldsymbol{k},\boldsymbol{q}), \tag{3.6}$$

the dielectric function simplifies to

$$\epsilon_{IJ}(\boldsymbol{q},\omega) = \delta_{ij} - \sum_{\boldsymbol{k}} \sum_{nm} F_{nm}(\boldsymbol{k},\boldsymbol{q};\omega)\ \tilde{M}^{i}_{nm}(\boldsymbol{k},\boldsymbol{q})\ [\tilde{M}^{j}_{nm}(\boldsymbol{k},\boldsymbol{q})]^{*}. \tag{3.7}$$

Therefore, we do not have to compute the polarizability explicitly. However, this trick only works for the SOS part of the polarizability, because of its symmetric structure. In this work we implement an explicit computation of the SOS part of the polarizability in the G_0W_0 framework of `exciting`, to then be able to compute the dielectric function from the (later corrected) polarizability. To do so, we have to express the matching coefficients of the MPB $M^{I}_{nm}(\boldsymbol{k},\boldsymbol{q})$ in terms of (L)APWs and LOs.

3.1.1. Evaluation of $M^{I}_{nm}(\boldsymbol{k},\boldsymbol{q})$ in the (L)APW+LO basis

The expansion coefficients of the MPB play a central role in the construction of the polarizability matrix. In the following, we show how they are constructed in the (L)APW+LO basis set [30].

We start from the representation of the Kohn-Sham orbitals in terms of the basis,

$$\psi_{n\boldsymbol{k}}(\boldsymbol{r}) = \sum_{\boldsymbol{G}} C_{n\boldsymbol{k},\boldsymbol{G}}\ \phi_{\boldsymbol{G}+\boldsymbol{k}}(\boldsymbol{r}) + \sum_{\mu} \tilde{C}_{n\boldsymbol{k},\mu}\ \phi_{\mu}(\boldsymbol{r}), \tag{3.8}$$

and rewrite them by differentiating between MT and I regions:

$$\psi_{n\boldsymbol{k}}(\boldsymbol{r}) = \begin{cases} \sum_{\zeta\nu lm} \mathcal{A}_{n\boldsymbol{k},\alpha\zeta\nu lm}\ u_{l\alpha\zeta}(r_\alpha, E_{l\alpha\nu})\ Y_{lm}(\hat{\boldsymbol{r}}^{\alpha}) & r_\alpha \le R^{\alpha}_{\mathrm{MT}} \\ \\ \frac{1}{\sqrt{\Omega}} \sum_{\boldsymbol{G}} C_{n\boldsymbol{k},\boldsymbol{G}}\ e^{i(\boldsymbol{G}+\boldsymbol{k})\boldsymbol{r}} & \boldsymbol{r} \in I. \end{cases} \tag{3.9}$$

Here, ζ represents the order of radial derivatives of $u_{l\alpha\zeta}(r_\alpha, E_{l\alpha\nu})$, and ν stands for the different linearization energies used to build a local orbital. The same partitioning applies to the expansion coefficients of the MPB. For a MPB function $\chi^{\boldsymbol{q}}_{I}(\boldsymbol{r}) = \gamma_{\alpha NLM}(\boldsymbol{r})$, which is in the MT-region, we plug Eq. 3.9 into Eq. 2.22. This leads to the expansion coefficient

$$M^{I}_{nm}(\boldsymbol{k},\boldsymbol{q}) = e^{-i\boldsymbol{q}\cdot\boldsymbol{r}_\alpha} \sum_{l_1m_1} \sum_{l_2m_2} [\mathcal{G}^{l_2m_2}_{LM;l_1m_1}]^{*} \sum_{\zeta_1\zeta_2} \sum_{\nu_1\nu_2} \mathcal{A}_{n\boldsymbol{k},\alpha\zeta_1\nu_1 l_1 m_1} \mathcal{A}^{*}_{n\boldsymbol{k},\alpha\zeta_2\nu_2 l_2 m_2} \langle NL|l_1\zeta_1\nu_1, l_2\zeta_2\nu_2\rangle, \tag{3.10}$$

with

$$\langle NL|l_1\zeta_1\nu_1, l_2\zeta_2\nu_2\rangle = \int_0^{R^{\alpha}_{\mathrm{MT}}} U_{\alpha NL}(r)\ u_{l_1\alpha\zeta_1}(r_\alpha, E_{l_1\alpha\nu_1})\ u_{l_1\alpha\zeta_1}(r_\alpha, E_{l_1\alpha\nu_1})\ r^2\ dr. \tag{3.11}$$

For a MPB function in the I region, $\chi_I^{\boldsymbol{q}}(\boldsymbol{r}) = P_i^{\boldsymbol{q}}(r)$, the expansion coefficients take the form

$$M_{nm}^{I}(\boldsymbol{k},\boldsymbol{q}) = \frac{1}{\Omega^{\frac{3}{2}}} \sum_{\boldsymbol{G}\boldsymbol{G}'} C_{n\boldsymbol{k},\boldsymbol{G}} \left[C_{m\boldsymbol{k}-\boldsymbol{q},\boldsymbol{G}'}\right]^{*} \sum_{\boldsymbol{G}_1} l_{\boldsymbol{G}-\boldsymbol{G}'-\boldsymbol{G}_1} \tilde{S}^{*}_{\boldsymbol{G}_1 i}. \tag{3.12}$$

Using this expression, the implementation of the polarizability is straightforward.

3.2. Solver for the Radial Sternheimer Equation

The first ingredient needed for the IBC, is a method to solve the radial Sternheimer equation 2.73 numerically. In this chapter, we describe a number of different algorithms that we implement and later compare, to find the best method for our application.

3.2.1. The Radial Sternheimer Equation

The radial Sternheimer equation is an inhomogeneous, ordinary differential equation (ODE) of second order. It is well known, that we can rewrite a single ODE of second-order into two coupled first-order ODEs, which simplifies the numerical solution. To shorten the following derivations, we can rewrite Eq. 2.73 into a more general form

$$\left[-\frac{d^2}{dr^2} + \tilde{h}(r)\right] r\, u^{(1)}(r) = I(r), \tag{3.13}$$

where

$$\tilde{h}(r) = \frac{l(l+1)}{2r^2} + v_0(r) - \epsilon_{l\alpha} - \omega \tag{3.14}$$

and the inhomogeneity $I(r)$ is

$$I(r) = \mathcal{G}^{\ell' m'}_{LM;\ell m} \left[\delta_{\ell\ell'}\, \epsilon^{(1)}_{\ell\alpha,I} - \chi_I(r)\right] r\, u_{\ell\alpha}(r). \tag{3.15}$$

To get first-order ODEs, we introduce two substitutions

$$P^{(1)}(r) = r\, u^{(1)}(r) \tag{3.16}$$

and

$$Q^{(1)}(r) = r\, \frac{d}{dr} u^{(1)}(r). \tag{3.17}$$

By taking the derivative of $P^{(1)}(r)$ with respect to r, we directly get the first first-order ODE

$$\frac{d}{dr} P^{(1)}(r) = u^{(1)}(r) + r\, \frac{d}{dr} u^{(1)}(r) = \frac{1}{r}\, P^{(1)}(r) + Q^{(1)}(r). \tag{3.18}$$

We derive the second first-order ODE by taking the radial derivative of $Q^{(1)}(r)$

$$\frac{d}{dr}Q^{(1)}(r) = \frac{d}{dr}u^{(1)}(r) + r\,\frac{d^2}{dr^2}u^{(1)}(r) = \frac{d^2}{dr^2}\Big[r\,u^{(1)}(r)\Big] - \frac{d}{dr}u^{(1)}(r), \tag{3.19}$$

and afterwards using Eq. 3.13 to find an expression for $\frac{d^2}{dr^2}u^{(1)}$, resulting in the second first-order ODE:

$$\frac{dQ^{(1)}(r)}{dr} = \tilde{h}(r)\;P^{(1)}(r) - I(r) - \frac{1}{r}\;Q^{(1)}(r). \tag{3.20}$$

These two ODEs are coupled through the quantities $P^{(1)}(r)$ and $Q^{(1)}(r)$. The values for the radial function $u^{(1)}(r)$, as well as its first derivative $\frac{d}{dr}u^{(1)}(r)$ at the MT center are known. This allows us to compute $P^{(1)}(r_0)$ and $Q^{(1)}(r_0)$. There are a number of different methods that are able to solve such a system of coupled first-order ODEs with known initial values. In the following, two classes of those methods, with there advantages as well as disadvantages are presented. The two methods we present, are the Runge-Kutta (RK) and the Bulirsch-Stoer method. The Bulirsch-Stoer method is known for its high accuracy, whereas RK methods are known to be more efficient, as long as the evaluation of the functional derivatives is inexpensive. However, newer RK methods of higher order are able to yield high accuracy as well. We show the following derivations for a very general form of a system of first-order ODEs

$$\frac{dy^i(r)}{dr} = f_i(r, y^0, y^1, ..., y^{N-1}); \qquad i = 0, ..., N-1, \tag{3.21}$$

where the initial values $y^i(r_0) = y_0^i$ are known. For the case of the radial Sternheimer equation, we choose $y^0(r) = P(r)$ and $y^1(r) = Q(r)$, where f_0 is then equal to the right side of Eq. 3.18, and f_1 is equal to the right side of Eq. 3.20. For simplicity we will from now on consider only a single ODE $\frac{dy(r)}{dr} = f(r, y)$, though the generalization to a system of linear ODEs is straightforward. The derivations in this chapter are mainly based on Ref. [31], Ref. [32], and Ref. [33].

3.2.2. Runge-Kutta solver

The RK method is an improved version of the basic Euler method, which Euler first proposed in 1768 [34]. Here, the value of a function $y(r)$ at the next grid point r_{n+1} is approximated from the function and its derivative at the previous grid point r_n,

$$y_{n+1} = y(r_n) + h\,\frac{d}{dr}y(r)\bigg|_{r_n} = y_n + h\;f(r_n, y_n), \tag{3.22}$$

where $h = r_{n+1} - r_n$ is the step size and y_n is the approximation of the function $y(r)$ at the grid point r_n [31]. This concept is illustrated in Fig. 3.1.

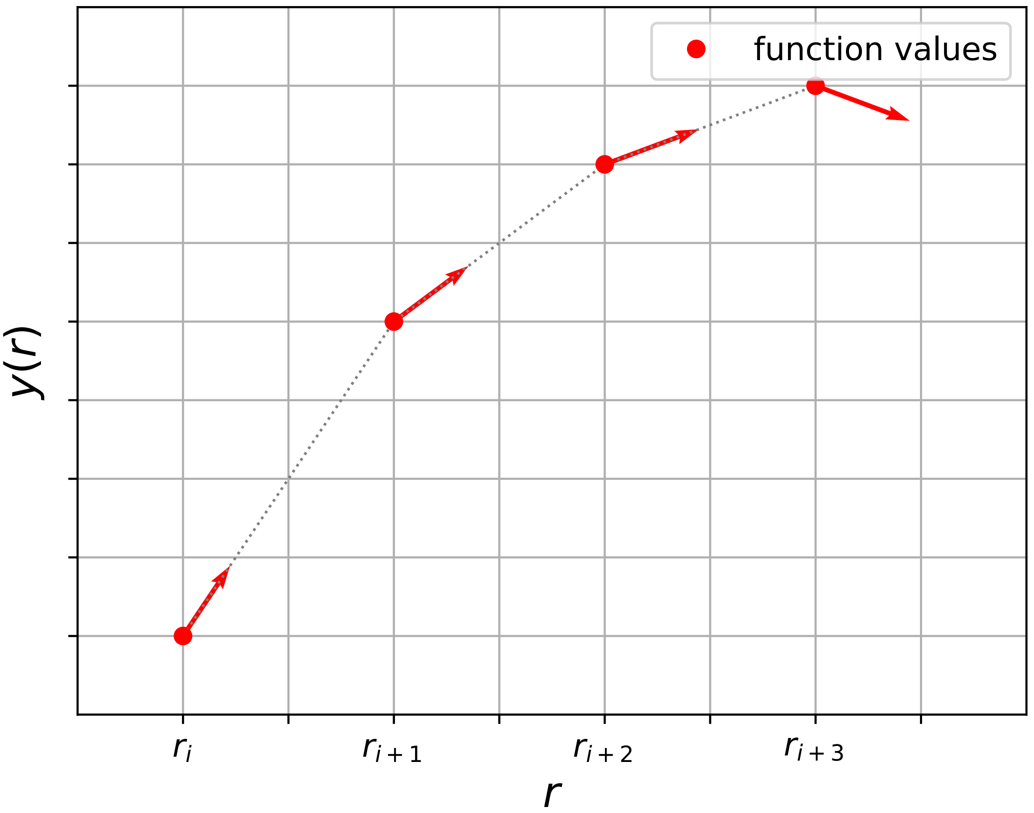

Figure 3.1.: Illustration of the Euler method. The derivatives (represented by the arrows) at each grid point r_n are used to find an estimate for the function value $y(r)$ at the next grid point r_{n+1}.

To get an estimate for the quality of the approximation, we compute the local truncation error $\mathcal{E}_{\text{lt}}$, which determines the error made in the computation of a single step. By expanding $y(r)$ into a Taylor series,

$$\mathcal{T}y(r;r_n) = y(r_n) + (r - r_n)\left.\frac{dy(r)}{dr}\right|_{r_n} + \frac{h^2}{2}\left.\frac{d^2y(r)}{dr^2}\right|_{r_n} + \mathcal{O}((r-r_n)^3) \tag{3.23}$$

$$= y_n + h\,f(r_n, y_n) + \frac{h^2}{2}\left.\frac{d^2y(r)}{dr^2}\right|_{r_n} + \mathcal{O}(h^3), \tag{3.24}$$

we are able to compare Eq. 3.23 to Eq. 3.22

$$\mathcal{E}_{\text{lt}} = y(r_n + h) - y_{n+1} = \frac{h^2}{2}\left.\frac{d^2y(r)}{dr^2}\right|_{r_n} + \mathcal{O}(h^3)\;. \tag{3.25}$$

With that, we show that the local truncation error for small step sizes is of order $\mathcal{O}(h^2)$, if we assume that the third derivative of $y(r)$ is bounded. However, this is only the error in a single step, assuming that the starting point y_n is an exact solution of $y(r_n)$. To get the global truncation error, i.e., the cumulative error after a number of iterations, we have to assume that f is Lipschitz continuous, i.e.,

$$L = \max\left(\left|\frac{d}{dr}[f(r,y)]\right|\right) \tag{3.26}$$

for the interval of r we are interested in, and that the local truncation error is bounded:

$$|\mathcal{E}_{\mathrm{lt}}| \leq M. \tag{3.27}$$

With that, we get an upper limit for the global truncation error, i.e.,

$$\mathcal{E}_{\mathrm{gt}} = |y(r_i) - y_i| \leq \frac{HM}{L}\left(e^{l(r_i - r_0)} - 1\right), \tag{3.28}$$

which is of order $\mathcal{O}(h)$. Therefore, the Euler method is considered a first-order method. A detailed derivation of truncation errors can be found in Ref [35]. The relatively large global truncation error makes the method not suitable for most applications. However, it is a good starting point for more involved methods, like the second order Runge-Kutta method described in the next section.

2$^{\text{nd}}$-order Runge-Kutta method

In order to reduce the truncation errors, the 2$^{\text{nd}}$ order RK method utilizes a trial step in the center of the interval h. It then takes the values of $f(r, y(r))$ evaluated at the initial point r_n,

$$k_1 = f(r_n, y_n), \tag{3.29}$$

and a point somewhere in the interval $r_n + ch$ defined by the coefficient $0 \leq c \leq 1$

$$k_2 = f(r_n + ch, y_n + ahk_1), \tag{3.30}$$

to compute y_{n+1}

$$y_{n+1} = y_n + h(b_1k_1 + b_2k_2). \tag{3.31}$$

By choosing the coefficients a, b_1, b_2, and c in a certain way, the local truncation error can be reduced by cancelling out the first-order error term as shown in the following.

We can expand Eq. 3.30 in a two-dimensional Taylor series:

$$\begin{aligned}\mathcal{T}k_2 &= \mathcal{T}f(r_n + ch, y_n + ahk_1) &(3.32)\\ &= f(r_n, y_n) + ah\ k_1\ f_y(r_n, y_n) + ch\ f_r(r_n, y_n) + \mathcal{O}(h^3) &(3.33)\\ &= f(r_n, y_n) + ch\ f_r(r_n, y_n) + ah\ f(r_n, y_n)\ f_y(r_n, y_n) + \mathcal{O}(h^3), &(3.34)\end{aligned}$$

where we define the partial derivatives as $f_y(r_n, y_n) = \frac{\partial}{\partial y} f(r, y(r))$ and $f_r(r_n, y_n) = \frac{\partial}{\partial r} f(r, y(r))$. Plugging this Taylor expansion back into Eq. 3.31, we obtain an expression for y_{n+1}:

$$y_{n+1} = y_n + hb_1\ f(r_n, y_n) + hb_2 \Big(f(r_n, y_n) + ch\ f_r(r_n, y_n) + ah\ f_y(r_n, y_n)\ f(r_n, y_n) + \mathcal{O}(h^3) \Big). \tag{3.35}$$

As for the Euler method, we then compare this expression to the Taylor expansion of the actual function $y(r)$. To do so, we rewrite Eq. 3.23 as

$$\begin{aligned}\mathcal{T}y(r; r_n) &= y_n + h\ f(r_n, y(r_n)) + \frac{h^2}{2}\frac{d}{dr}\left[\frac{d}{dr}y(r)\right]\Bigg|_{r_n} + \mathcal{O}(h^3)\\ &= y_n + h\ f(r_n, y(r_n)) + \frac{h^2}{2}\frac{d}{dr}f(r, y(r))\Bigg|_{r_n} + \mathcal{O}(h^3)\\ &= y_n + h\ f(r_n, y(r_n)) + \frac{h^2}{2} f_t(r_n, y(r_n)) + \frac{h^2}{2} f_y(r_n, y(r_n))\ f(r_n, y(r_n)) + \mathcal{O}(h^3).\end{aligned} \tag{3.36}$$

By comparing the coefficients of Eq. 3.35 and Eq. 3.36, we observe that they agree up to third-order terms, if a, b_1, b_2, and c satisfy the following order conditions:

$$\begin{aligned}b_1 + b_2 &= 1\\ b_2 c &= \frac{1}{2}\\ b_2 a &= \frac{1}{2}.\end{aligned} \tag{3.37}$$

Therefore, the local truncation error is of order $\mathcal{O}(h^3)$, which then leads to a global truncation error of order $\mathcal{O}(h^2)$, thus the name 2$^{\text{nd}}$-order RK. The fact that we have four coefficients that are determined by only three equations, gives us the possibility to define a variety of implementations of second-order RK methods. However, the most popular choice of coefficients is $b_1 = 0$, $b_2 = 1$, $a = c = \frac{1}{2}$. Using these coefficients, the trial point is in the center of the interval. It is only used to get a better estimate of the gradient needed to estimate $y(r_n)$. This variant of second-order RK is illustrated in Fig. 3.2. In the same manner as shown in this section, different combinations of first-order evaluations of $f(r, y(r))$ can help to cancel out error terms of even higher order. A generalization of this concept is shown in the following section.

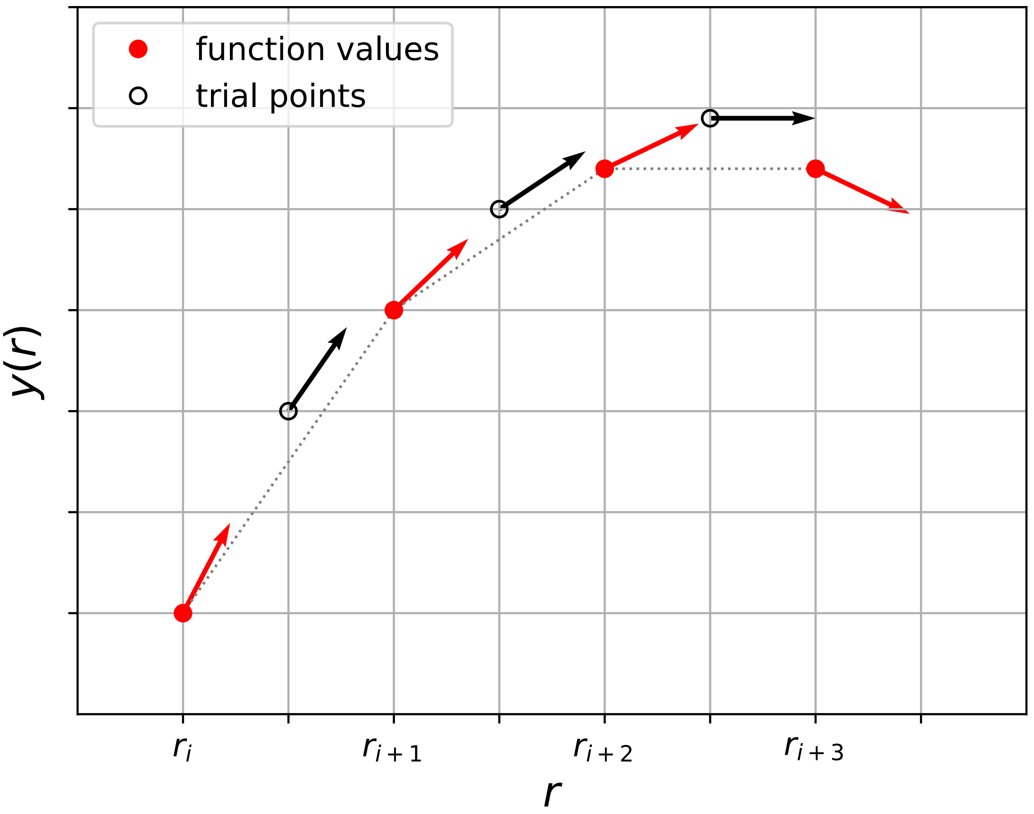

Figure 3.2.: Illustration of the second-order Runge-Kutta method. The derivatives (represented by the arrows) at each grid point r_n (red) are used to find an estimate for the gradient at the trial points $r_n + h/2$ (black). This new gradient (black arrow) is than used to approximate $y(r)$ at the next grid point, $r_{n+1} = r_n + h$.

General formulation of the Runge-Kutta method

In 1901, Kutta defined a general definition of the Runge-Kutta method [36]. A RK method is defined by the number of stages s, where s represents the number of times the function $f(r, y(r))$ is evaluated. For example, the second-order RK method shown above, is a two-stage method. In the s-stage RK method the approximation of $y(r)$ at the grid point $r_n + 1$ is given by

$$y_n = y_{n+1} + h(b_1 k_1 + b_2 k_2 + ...b_s k_s). \tag{3.38}$$

The evaluations K_I of $f(r, y(r))$ are defined as

$$
\begin{aligned}
&k_1 = f(r_n, y_n) \\
&k_2 = f(r_n + c_2 h, y_n + h a_{21} k_1) \\
&k_3 = f(r_n + c_3 h, y_n + h(a_{31} k_1 + a_{32} k_2)) \\
&... \\
&k_s = f(r_n + c_s h, y_n + h(a_{s1} k_1 + a_{s2} k_2 + ... + a_{ss-1} k_s - 1)),
\end{aligned}
\tag{3.39}
$$

where the coefficients a_{ij}, b_i, and c_i are real numbers. As shown for the second-order RK method, certain choices of the coefficients minimize the order of the local (and therefore also global) truncation error. Finding these order conditions becomes increasingly difficult for higher-order RK methods and is not shown in this work. However, there are ways to simplify the derivations with smart notation choices, as shown in detail by e.g. Hairer and coworkers [32]. In the following two sections we show examples of higher-order RK methods, which are utilized in this work.

4^{th}-order Runge-Kutta solver

The 4^{th}-order RK method is known as the workhorse of radial solvers, because it offers a good balance of accuracy and computational cost. As shown by Heun [37], a 4-step RK-method is of fourth-order, if the following eight order conditions are satisfied:

$$b_1 + b_2 + b_3 + b_4 = 1 \tag{3.40}$$
$$b_2 c_2 + b_3 c_3 + b_4 c_4 = 1/2 \tag{3.41}$$
$$b_2 c_2^2 + b_3 c_3^2 + b_4 c_4^2 = 1/3 \tag{3.42}$$
$$b_2 c_2^3 + b_3 c_3^3 + b_4 c_4^3 = 1/4 \tag{3.43}$$
$$b_3 a_{32} c_2 + b_4(a_{42} c_2 + a_{43} c_3) = 1/6 \tag{3.44}$$
$$b_3 c_3 a_{32} c_2 + b_4 c_4(a_{42} c_2 + a_{43} c_3) = 1/8 \tag{3.45}$$
$$b_3 a_{32} c_2^2 + b_4(a_{42} c_2^2 + a_{43} c_3^2) = 1/12 \tag{3.46}$$
$$b_4 a_{43} a_{32} c_2 = 1/24. \tag{3.47}$$

This system of equation is under-determined, since we have 13 degrees of freedom, but only eight equations, which allows to choose the coefficients in a couple of different ways. The most popular fourth-order method, also referred to as ”The” Runge-Kutta method was proposed by Kutta [36]. In his method all coefficients are equal to zero, except for

$$c_2 = c_3 = a_{21} = a_{32} = 1/2 \tag{3.48}$$
$$c_4 = a_{43} = 1 \tag{3.49}$$
$$2b_1 = b_2 = b_3 = 2b_4 = 1/3. \tag{3.50}$$

This configuration simplifies the expression for y_{n+1}

$$y_{n+1} = y_n + \frac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right) \tag{3.51}$$

as well as the evaluations of $f(r, y(r))$

$$k_1 = f(r_n, y_n) \tag{3.52}$$

$$k_2 = f(r_n + \frac{h}{2}, y_n + \frac{h}{2}k_1) \tag{3.53}$$

$$k_3 = f(r_n + \frac{h}{2}, y_n + \frac{h}{2}k_2) \tag{3.54}$$

$$k_4 = f(r_n + h, y_n + hk_3). \tag{3.55}$$

We have chosen to implement this specific configuration of the fourth-order RK in the all-electron full-potential computer package `exciting` [23] to use it for solving the radial Sternheimer equation, because it offers a good balance between accuracy and computational cost.

8th-order Runge-Kutta solver

The highest order of RK commonly used, are RK methods of order eight. The coefficients for these methods need to satisfy 115 order conditions. It has been proven that these conditions can only be satisfied using a RK method that performs at least 11 steps k_i, making it noticeable more expensive than, for example, fourth-order RK methods [32]. In this work, we us an implementation of this method from the python library SciPy [2]. Despite eighth-order RK having a truncation error that grows only slowly with increasing step size, it is difficult to estimate the actual error made by the approximation, even so being assumed small for most applications. A more systematic error estimation is intrinsic to the Bulirsch-Stoer method, described in the next section.

3.2.3. Bulirsch-Stoer Method

The Bulirsch-Stoer method was first proposed in 1980 [33]. It combines the basic concept of Richardson extrapolation with polynomial extrapolation and the modified-midpoint method.

Richardson-extrapolation

In the RK-method discussed above, the step size h is a preset parameter that heavily influences the accuracy of the method. Richardson extrapolation methods, on the other hand, treat h as an adjustable parameter. This parameter can be decreased, resulting in a more and more accurate approximation of the function $y(r)$ at r_n, but not necessarily reaching the desired accuracy as indicated in Fig. 3.3. However, the information on the approximation thus gained can be used to fit some analytic function, extrapolating the

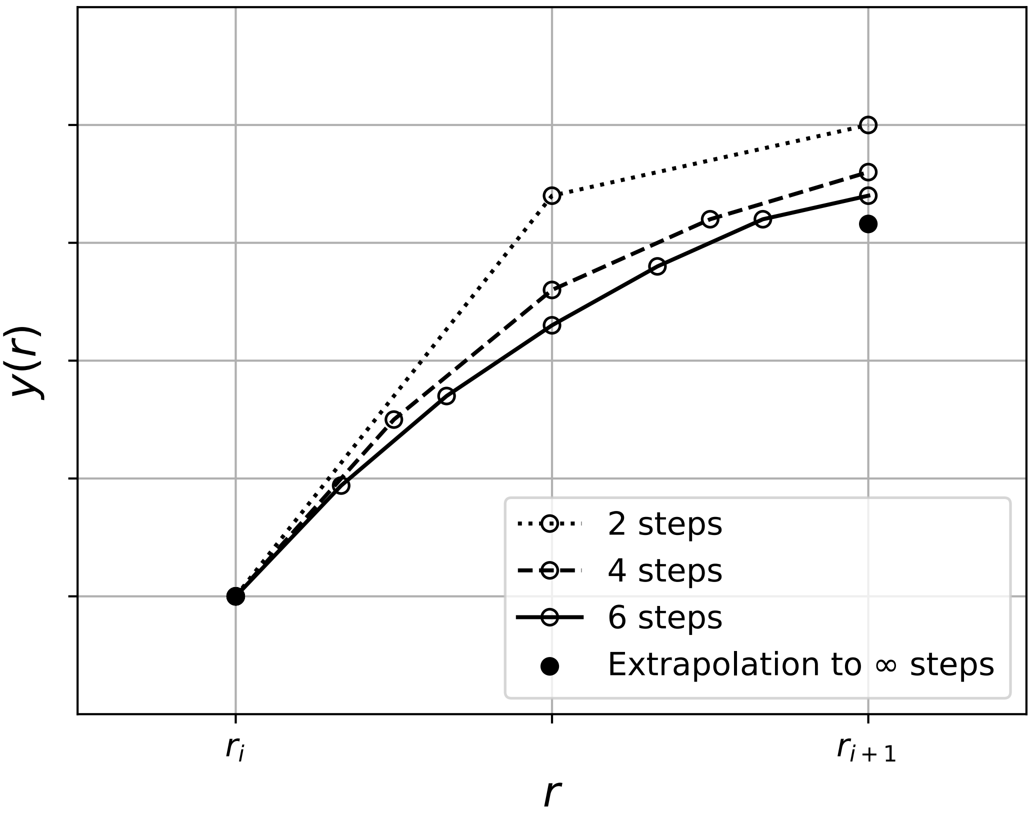

Figure 3.3.: Illustration of the Richardson extrapolation. The function value of $y(r)$ is propagated over a large interval $r_{n+1} - r_n = H$ via different sets of sub-grid points with decreasing step size. The different approximations of $y(r_n)$ are then extrapolated to an infinitesimal step size.

result for a vanishing step size [38]. The first ingredient needed for such a method is a scheme for advancing the function $y(r)$ from one grid point r_n to the next, $r_{n+1} = r_n + H$, using substeps of size $h = \frac{H}{N}$, where N is the number of substeps. Since we will later extrapolate the result to a vanishing step size, the method does not necessarily need to be a high-order approximation like the RK-methods. Having said this, it is more important that the method is computationally inexpensive, because it has to be executed several times for decreasing step sizes. A good choice is the modified-midpoint method.

Modified-midpoint method

The modified-midpoint method [37] propagates $y(r)$ from one sub-grid point to the next:

$$k_0 = y(r_n) \tag{3.56}$$
$$k_1 = k1 + h\ f(r, k_0) \tag{3.57}$$
$$k_{m+1} = k_{m-1} + 2h\ f(r + mh, k_m) \qquad \text{for} \qquad m = 1, 2, ..., N_{\text{sub}} - 1, \tag{3.58}$$

until the grid point r_{n+1} is reached

$$y(r_{n+1}) \approx y_N = \frac{1}{2}\left[k_N + k_{N-1} + hf(x + H, k_N)\right]. \tag{3.59}$$

For large N, this method requires approximately only one evaluation of $f(r, y(r))$ per step h, while still being a second-order method, such as second-order RK, which requires two evaluations of the gradient. However, the RK method, even though being of the same order, minimizes the actual truncation error. The modified-midpoint method provides another advantage. As shown by Gragg [39], the local truncation error contains only powers of h^2

$$y_N - y(r_n + H) = \sum_{i=1}^{\infty} \epsilon_i h^{2i}. \tag{3.60}$$

We use the same idea of combining steps to get rid of higher-order error terms as we did for the RK-methods, with the advantage that the error decreases by two orders at a time. It can be shown that, for example, combining the result of the modified-midpoint method executed with N steps with the result computed with $N/2$ steps (assuming N is even)

$$y(x + H) \approx \frac{y_N - y_{N/2}}{3} \tag{3.61}$$

is accurate up to fourth order [31] and needs for large N only 1.5 evaluations of the gradient per step. The polynomial extrapolation makes use of these advantages.

Polynomial extrapolation

Bulirsch and Stoer first proposed using rational function extrapolation in their method. However, it was later found that for smooth problems, the polynomial extrapolation is more efficient and also less intricate to implement. Polynomial interpolation uses the polynomial with the lowest degree that fits all points in a given set. For example, two data points can be met by a second-order polynomial [31]. A way to implement polynomial extrapolation is the Nevilles algorithm [40]. It is straightforward to implement and allows for error estimation. The basic idea is that polynomials of higher degrees can be computed from polynomials of lower degrees that, when combined, match the same set of data points. To make this abstract concept a bit more instructive, we illustrate it with the example of a polynomial that is supposed to fit four data points, as shown in Fig. 3.4.

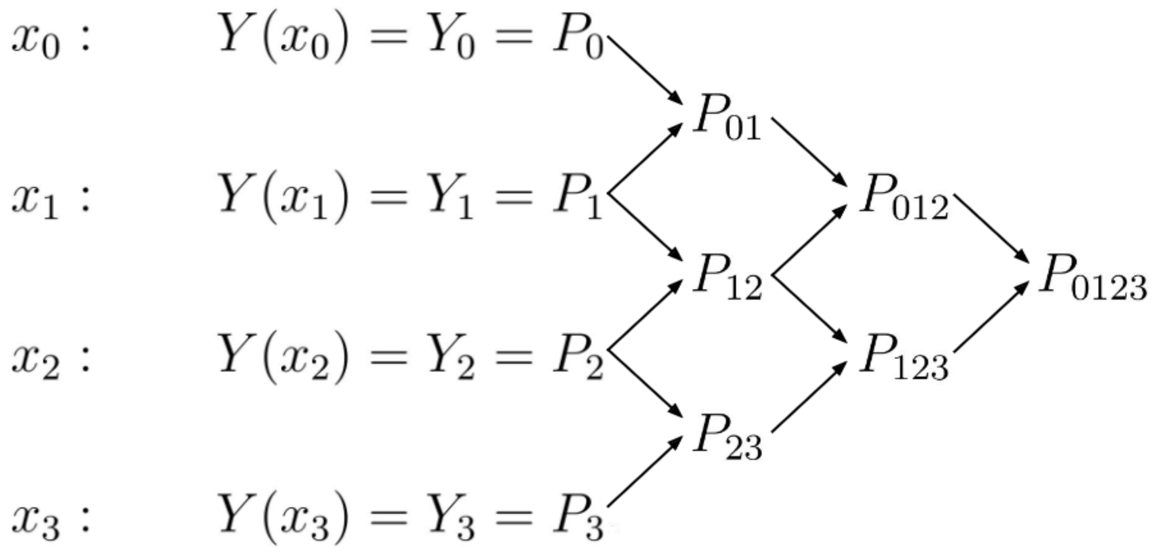

Figure 3.4.: This flowchart illustrates polynomial extrapolation. The zero-degree polynomials P_i in the first column are equal to the function value $Y(x)$ at x_i. Polynomials of higher degrees are calculated from polynomials of lower degrees as indicated by the arrows.

Here, we start computing the polynomials from the left. The polynomials in the first column are of degree zero, and therefore just the value of the function $Y(x)$ at x_i. The polynomials in the next column are of degree one and can be computed from the zero-degree polynomials using the relation

$$P_{ij}(x) = \frac{(x - x_j)P_i + (x_i - r)P_j}{x_i - x_j}. \tag{3.62}$$

In the same manner, each of the following columns is filled. The general relation to relate two "parent" polynomials of degree d to one "descendant" polynomial of degree $d + 1$ is given by [31]

$$P_{i(i+1)\ldots(i+m)} = \frac{(x - x_{i+m})P_{i(i+1)\ldots(i+m-1)} + (x_i - x)P_{(i+1)(i+2)\ldots(i+m)}}{x_i - x_{i+m}}. \tag{3.63}$$

Bulirsch-Stoer method in practice

Now, that we have discussed how the polynomial extrapolation works in general, we show how it can be used in the Bulirsch-Stoer method to extrapolate to an infinitesimal step size. To simplify the equations, we can represent the flowchart ilustrated in Fig. 3.4 as a lower triangular matrix:

$$\begin{pmatrix} T_{00} & & & \\ T_{10} & T_{11} & & \\ T_{20} & T_{21} & T_{22} & \\ T_{30} & T_{31} & T_{32} & T_{33} \\ \dots & \dots & \dots & \dots \end{pmatrix}. \tag{3.64}$$

The first column represents the zero-degree polynomials, the second column represents the first-degree polynomials, and so on. In the context of the Bulirsch-Stoer method, each data point represents the value $y(r_n)$ computed with a certain step size $h_i = H/N_i$. Therefore, the zero-degree polynomials are $T_{i0} = y_i$, where $y_i = y(r_n + H)$, computed with the step size h_i. This means that computing $y(r_n + H)$ with a new step size h_i, adds a new row to the matrix 3.64. With y_i as the first entry of a row, the row is filled using polynomial extrapolation. To do so, Eq. 3.63 can be rewritten in terms of T_{ij}

$$T_{i\,j+1} = T_{ij} + \frac{T_{ij} - T_{i-1\,j}}{(n_i/n_{i-j})^2 - 1}, \qquad \text{with } j = 0, 1, ..., i-1. \tag{3.65}$$

Elements in the same column of the matrix are approximations of the same order of h. The elements T_{ii} in the last column of each row are approximations of the highest order for the respective step size h_i. The difference between the last two elements of a row can be taken as an error estimate. In practice, the modified-midpoint method is performed with increasing numbers of substeps N_i until the error estimate drops below a certain tolerance [31]. Bulirsch and Stoer originally proposed to increase the number of steps following the scheme

$$N_i = 2N_{i-2}, \tag{3.66}$$

but Deuflhard later showed that increasing N_i following the series

$$N_i = 2(i+1) \tag{3.67}$$

is more efficient in most cases [41]. We have implemented this method in addition to the fourth-order RK solver in the `exciting` code. It should be noted that, due to numerical noise, too small step sizes can lead to a loss of precision. To avoid this, a minimal step size is set. If this minimal step size is reached, but the error estimate did not yet drop below the desired precision, we switch to the fourth-order RK method, for the one step r_n to r_{n+1} only.

4. Results

4.1. Polarizability

As shown in Section 3.1, we need to compute the polarizability explicitly to be able to add the IBC correction terms to it later. In this section, we compare the G_0W_0 algorithm that utilizes the trick of skipping the computation of the polarizability with our new implementation, where the polarizability is computed explicitly. We use two different materials as test cases. The first material is bulk silicon, which is a simple example often computed with the G_0W_0 method. The second example is zinc oxide. This material is known for demanding a large number of unoccupied states as well as local orbitals to obtain a converged G_0W_0 calculation, making its computation rather expensive. With the help of the IBC, we ultimately want to get rid of the need for such a large basis and, therefore, reduce computational cost. The structures for both materials are taken from the NOMAD repository [42, 43]. By comparing the individual elements of the dielectric function in matrix form computed with the different methods, we validate our implementation. Since our goal is not to get fully converged results, we use looser settings for the convergence parameters than we would do for production calculations. We use the same settings for both methods, which ensures that the choice of the convergence parameters does not affect the comparison of the two methods. For both materials, the differences in the matrix elements of the dielectric function are smaller than 10^{-15}. Since double precision floating-point numbers are used, these discrepancies can be explained by purely numerical errors, assuring us that our implementation is valid. However, when comparing the computational cost, the computation of the dielectric function with the explicit computation of the polarizability is considerably more expensive than the direct computation. The corresponding time for Si, averaged over all q-points, amounts to 18 seconds, when the polarizability is computed explicitly, while it takes only 10 seconds for the direct computation of the dielectric function. We find a similar behavior for ZnO, with times of approximately 252 seconds and 149 seconds, respectively.

4.2. Radial solver

The radial Sternheimer equation needs to be solved for every combination of MPB functions and radial functions, defined by l and the linearization energy. This shows that the solution of the radial Sternheimer equation is a crucial step in the construction of the IBC. Our goal is to find an ODE solver that gives sufficiently precise results while still being low in computational cost. To find a suitable solver, we first compare three different RK solvers of different orders. For this comparison we use already existing solvers implemented in Python [2, 1]. In a second step, we implement the RK solver with the best balance between computational cost and precision in the full-potential all-electron computer package `exciting` [23]. To validate our implementation, we compare it to results obtained with the Python implementation of the same order. In a last step, we implement the Bulirsch-Stoer algorithm in `exciting` and compare it to the RK solver. This comparison is done because the unperturbed radial functions are computed with a Bulirsch-Stoer algorithm and we want to make sure that the precision loss introduced through the use of the RK solver is still of an acceptable order. We test all of these solvers on examples of radial response functions in the same materials used in Section 4.1. It should be noted that it is less relevant to have a large number of different materials, because the muffin-tin potentials of their atoms produce response functions with similar shapes (e.g. number of nodes) as long as they correspond to the same state (defined by l and the linearization energy). Therefore, we compute perturbed radial functions $P^{(1)}_{\ell\alpha}(\boldsymbol{r}) = r\ u^{(1)}_{\ell\alpha}(\boldsymbol{r})$ with azimuthal quantum numbers $l \in \{0, 1, 2\}$ perturbed by MPB functions $\chi^{\boldsymbol{q}}_{I}$ with azimuthal quantum numbers $L \in \{0, 1, 2\}$ different values of $\omega \in \{0, 0.5, 10, 150\}$. This results in 108 different radial response functions with various shapes.

4.2.1. Metrics for the precision

To be able to compare the precision of different radial solvers, we first need to find a metric for the precision. For this, we use two different error estimates. On the one hand, we use the maximum absolute difference (MAD), which is the largest difference between two functions,

$$MAD = \max_{i=1,\dots,N} \left(|P^{(1)}_{\text{solver 1}}(r_i) - P^{(1)}_{\text{solver 2}}(r_i)| \right), \text{ with } r_N = R_{\text{MT}}, \tag{4.1}$$

where r_0 corresponds to the grid point closest to zero and R_{MT} represents the MT-boundary. This helps to give an estimate for the upper bound of the error. On the other hand, we use the root-mean-squared difference (RMSD)

$$RMSD = \sqrt{\frac{1}{R_{\text{MT}} - r_0} \int_{r_0}^{R_{\text{MT}}} \left(P^{(1)}_{\text{solver 1}}(r) - P^{(1)}_{\text{solver 2}}(r) \right)^2 dr}. \tag{4.2}$$

This gives us an error estimate for the whole range of r that we are interested in. To get a baseline for these two errors, we first compare the unperturbed with the perturbed radial functions. So, instead of taking the difference between $P^{(1)}_{\text{solver 1}}(r)$ and $P^{(1)}_{\text{solver 1}}(r)$, we take the difference between $P^{(1)}(r)$ and $P(r)$. As an example, we show the $l = 0$, $n = 4$ radial function of zinc (in ZnO) perturbed by a MPB function with $L = 2$ in Fig. 4.1. For this example we get a MAD of 0.44 and an RMSD of 0.29. If we look for the minimum differences over all calculated radial functions, we get similar results. The minimum MAD is equal to 0.29 and the averaged RMSD is equal to 0.18. Therefore, we can take a MAD and RMSD of order 10^{-1} as a baseline for later error estimations.

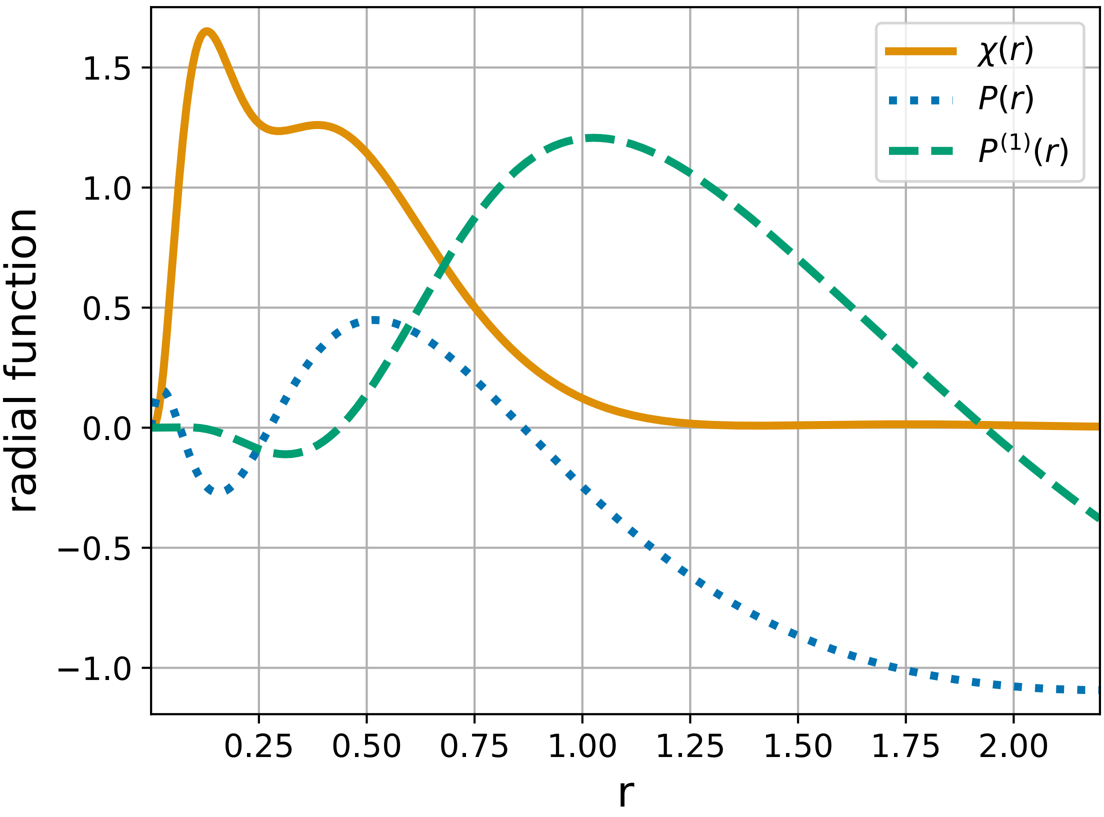

Figure 4.1.: Normalized radial function (blue dotted line) for the azimuthal quantum number $l = 0$ calculated in the MT-sphere of Zn in ZnO and a normalized radial response function (green dashed line), perturbed by the MPB function with azimuthal quantum number $L = 2$ (orange solid line).

4.2.2. Comparison of RK solvers of orders two, four, and eight

Second-order RK vs. Eighth-order RK Solver

In a first step, we compare a second-order RK solver with an eighth-order RK solver. The second-order RK solver we use here, is a Python implementation taken from Ref. [1]. The eighth-order solver is implemented in Python as well and part of the SciPy library [2]. This solver is taken as a reference for further considerations. As explained in Section 3.2.2, the second-order RK solver needs to evaluate the gradient only two times for every step, whereas the eighth-order RK demands eleven evaluations and is therefore considerably more expensive. By using the eighth-order RK solver as a reference, we estimate how much precision is lost by using the second-order RK solver. We again take the radial function mentioned above as an example. For this radial response function, we get an MAD of $7{\cdot}10^{-4}$

and an RMSD of $3 \cdot 10^{-4}$. The averaged MAD over all response functions computed is equal to $6 \cdot 10^{-2}$ and the averaged RMSD is equal to $8 \cdot 10^{-2}$. These deviations are already large. Having said this, considering calculations for each value of ω individually, shows a strong increase of the error for increasing values of ω, as shown in Table 4.1. In Fig. 4.2, we show how increasing values of ω lead to more strongly varying response functions. These variations are better captured, when the gradient is evaluated at a higher number of trial points, as done in the eighth-order RK method (Section 3.2.2). For the highest value of ω considered in this comparison, the largest MAD is approximately equal to $3.9 \cdot 10^{-1}$. This error is of the same order as the baseline defined in Section 4.2.1. This lets us conclude that a solver of higher order is required for the solution of the radial Sternheimer equation.

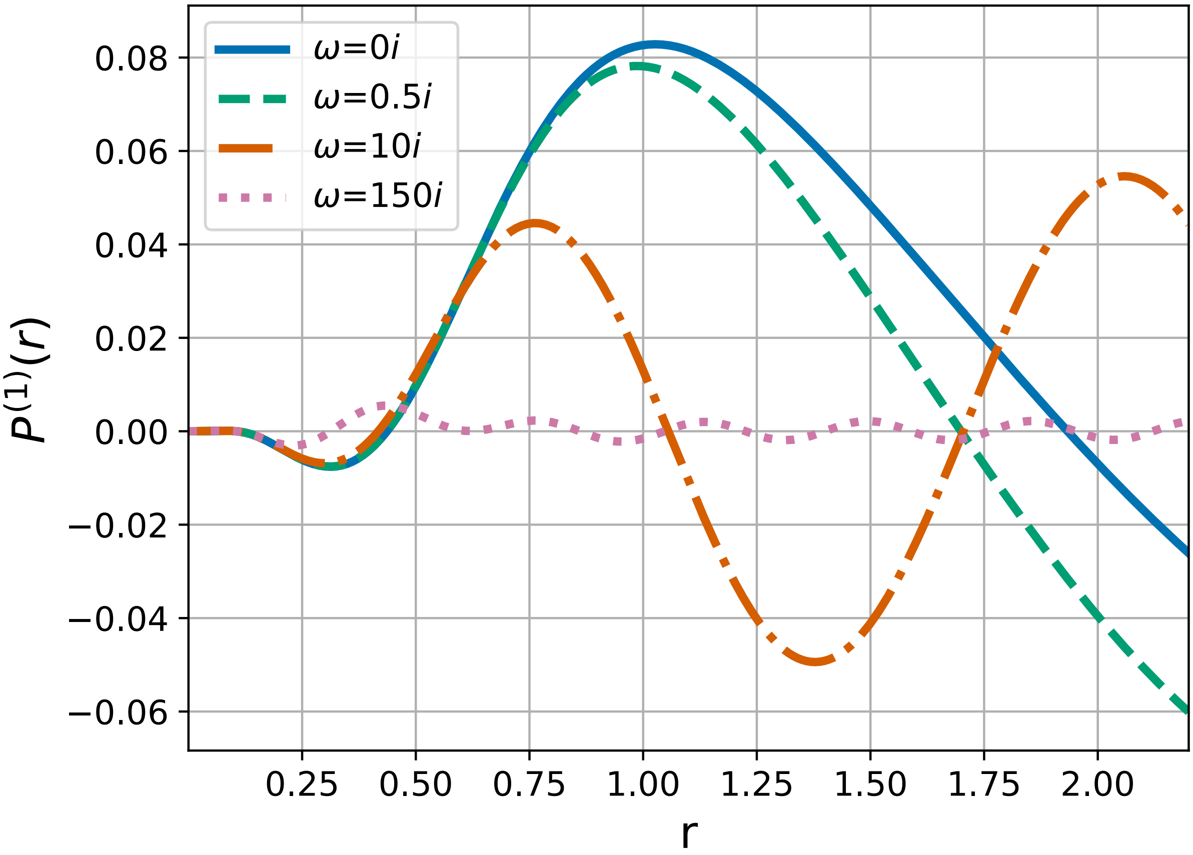

Figure 4.2.: Normalized radial response function calculated for $l = 0$ in the MT-sphere of Zn in ZnO, perturbed by the MPB function with azimuthal quantum number $L = 2$ for different values of ω.

Fourth-order RK vs. eighth-order RK solver

In a next step, we compare a fourth-order RK solver with the eighth-order RK solver considered before. The fourth-order RK solver we use here, is a Python implementation

Table 4.1.: Differences as defined in Section 4.2.1 between radial functions computed with the second-order RK solver [1] and the eighth-order RK solver [2] for different values of ω.

ω	**averaged MAD**	**averaged RMSD**	**maximum MAD**	**maximum RMSD**
0	$4.5 \cdot 10^{-4}$	$2.2 \cdot 10^{-4}$	$1.7 \cdot 10^{-3}$	$7.9 \cdot 10^{-4}$
0.5	$4.9 \cdot 10^{-4}$	$2.7 \cdot 10^{-4}$	$1.1 \cdot 10^{-3}$	$7.3 \cdot 10^{-4}$
10	$8.9 \cdot 10^{-3}$	$3.1 \cdot 10^{-3}$	$3.3 \cdot 10^{-2}$	$1.1 \cdot 10^{-2}$
150	$2.5 \cdot 10^{-1}$	$7.7 \cdot 10^{-2}$	$3.9 \cdot 10^{-1}$	$1.1 \cdot 10^{-1}$

taken from Ref. [1]. Since the fourth order RK solver needs to evaluate the gradient only four times for every step, it is still considerably faster than the eighth-order RK solver. As shown in Fig. 4.3 for the radial response function with $l = 0$, $L = 2$, and $\omega = 0$, there are no visible differences between the radial functions computed with the two solvers. For this example, we get a MAD of $4 \cdot 10^{-5}$ and a RMSD of $1 \cdot 10^{-5}$. The MAD and RMSD averaged over all computed response functions are of the order 10^{-4}, three orders smaller than our baseline deviations. Looking at the calculations for each value of ω individually, we observe that the errors increase for increasing values of ω, as expected (see Table 4.2). However, the maximum MAD over all response functions computed with different values of l, L, and ω is equal to $4.4 \cdot 10^{-3}$ and the maximum RMSD is approximately $2.1 \cdot 10^{-3}$. We consider the largest difference still sufficiently small to use the fourth-order RK solver for production calculations. Therefore, we have decided to implement this RK solver in the `exciting` computer package.

Table 4.2.: Differences as defined in Section 4.2.1 between radial functions computed with the fourth-order RK solver [1] and the eighth-order RK solver [2] for different values of ω.

ω	**averaged MAD**	**averaged RMSD**	**maximum MAD**	**maximum RMSD**
0	$1.5 \cdot 10^{-4}$	$5.2 \cdot 10^{-5}$	$6.0 \cdot 10^{-4}$	$3.4 \cdot 10^{-4}$
0.5	$9.1 \cdot 10^{-5}$	$2.4 \cdot 10^{-5}$	$3.6 \cdot 10^{-4}$	$1.3 \cdot 10^{-4}$
10	$3.5 \cdot 10^{-4}$	$1.0 \cdot 10^{-4}$	$8.9 \cdot 10^{-4}$	$7.3 \cdot 10^{-4}$
150	$2.0 \cdot 10^{-3}$	$6.1 \cdot 10^{-4}$	$4.4 \cdot 10^{-3}$	$2.1 \cdot 10^{-3}$

4.2.3. Validation of the fourth-order RK solver implemented in `exciting`

After implementing the fourth-order RK solver in `exciting`, we compare results computed with our new implementation with results form the fourth-order RK solver implemented in Python [1]. Comparing these two solvers, we get a maximum AMD of $1.1 \cdot 10^{-9}$ and a maximum RMSD of $7.7 \cdot 10^{-10}$. The MAD averaged over all computed radial functions

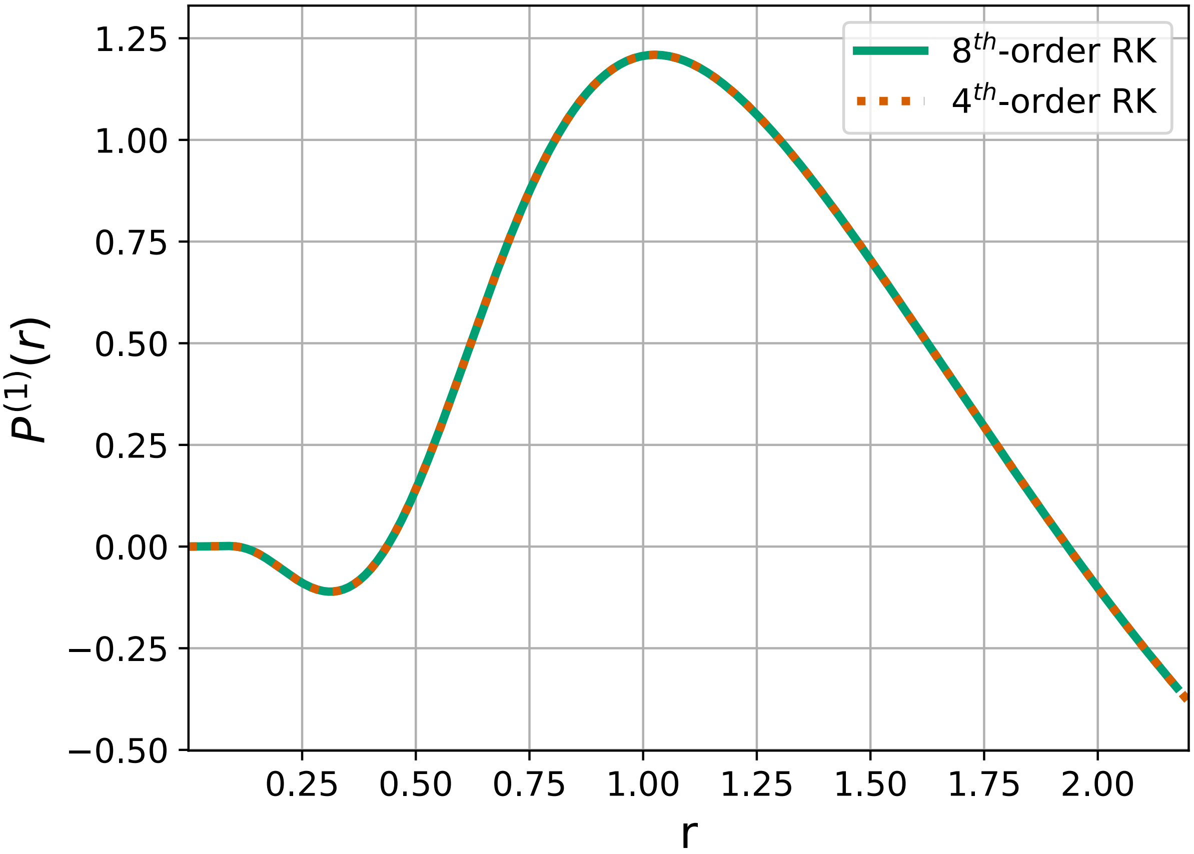

Figure 4.3.: Shown here, is a normalized radial response function calculated for $l = 0$ in the MT-sphere of Zn in ZnO, perturbed by the MPB function with azimuthal quantum number $L = 2$ computed with the Python implementation of the eighth-order RK solver [2] (green solid line) and computed with the Python implementation of the fourth-order RK solver [1] (orange dashed line).

is on the order of 10^{-10} and the averaged RMSD is of order 10^{-11}. These deviations are considered small. The errors are also independent of ω as shown in Table 4.3. This is expected, since the subgrid points used in both methods are identical.

4.2.4. Comparison of the fourth-order RK solver with the Bulirsch-Stoer solver

In the `exciting` code [23], the unperturbed radial functions are computed using a Bulirsch-Stoer solver for homogeneous ODEs. We have implemented a new Bulirsch-Stoer solver that is able to compute homogeneous as well as inhomogeneous ODEs. We compare this new Bulirsch-Stoer solver with the fourth-order RK solver in terms of precision and computational cost. The averaged MAD as well as RMSD for these two solvers are both of

Table 4.3.: Differences between radial functions as defined in Section 4.2.1, computed with our fourth-order RK solver implemented in `exciting` and the fourth-order RK solver implemented in Python [1] for different values of ω.

ω	**averaged MAD**	**averaged RMSD**	**maximum MAD**	**maximum RMSD**
0	$3.6 \cdot 10^{-11}$	$2.1 \cdot 10^{-11}$	$6.6 \cdot 10^{-10}$	$4.1 \cdot 10^{-10}$
0.5	$8.1 \cdot 10^{-11}$	$4.9 \cdot 10^{-11}$	$5.6 \cdot 10^{-10}$	$3.4 \cdot 10^{-10}$
10	$1.6 \cdot 10^{-10}$	$9.4 \cdot 10^{-11}$	$1.1 \cdot 10^{-9}$	$7.7 \cdot 10^{-10}$
150	$4.9 \cdot 10^{-10}$	$2.8 \cdot 10^{-11}$	$9.8 \cdot 10^{-10}$	$6.1 \cdot 10^{-10}$

the order of 10^{-5}, thus four orders of magnitude smaller than our baseline defined at the beginning of this section. However, the error depends strongly on the value of ω as shown in Table 4.4. The reason for this is the same, as explained in Section 4.2.2. More subgrid points are required to describe strong variations in the radial functions, as caused by large values of ω. Nevertheless, we consider even the maximum RMSD error for $\omega = 150$ sufficiently small and have decided to use the RK solver for production calculations, especially considering the computational costs. The difference in computational cost is compared by performing the radial solution for each computed response function 100 times and then averaging over all values. The computation time needed for the solution with the Bulirsch-Stoer solver is approximately three times higher than that for the solution with the fourth-order RK solver. For the Bulirsch-Stoer solver, we also see a slight increase in computation time for the largest value of ω, while the computation time remains constant for the RK solver. This can be explained by the increasing number of substeps needed for the Bulirsch-Stoer method to capture the strong variations with constant precision.

Table 4.4.: Differences between radial functions as defined in Section 4.2.1, computed with the fourth-order RK solver and the Bulirsch-Stoer solver, both implemented in `exciting` for different values of ω.

ω	**averaged MAD**	**averaged RMSD**	**maximum MAD**	**maximum RMSD**
0	$3.4 \cdot 10^{-8}$	$1.8 \cdot 10^{-8}$	$2.5 \cdot 10^{-7}$	$1.2 \cdot 10^{-7}$
0.5	$5.8 \cdot 10^{-8}$	$2.9 \cdot 10^{-8}$	$3.1 \cdot 10^{-7}$	$1.5 \cdot 10^{-7}$
10	$9.1 \cdot 10^{-7}$	$3.4 \cdot 10^{-7}$	$1.6 \cdot 10^{-6}$	$5.5 \cdot 10^{-7}$
150	$7.6 \cdot 10^{-4}$	$2.1 \cdot 10^{-4}$	$1.3 \cdot 10^{-3}$	$3.5 \cdot 10^{-4}$

5. Conclusions and Outlook

As shown in Section 4.1, we are able to reproduce G_0W_0 results that are generated with a method were the computation of the polarizability is skipped, using our new implementation, in which the polarizability is calculated explicitly. This comparison validates our implementation of the polarizability. In the course of this, we have compared the computational cost for these two methods. This is of importance, because we ultimately want to reduce the high computational cost of G_0W_0 calculations by reducing the required basis set size. We have found that the computation of the dielectric function including the computation of the polarizability is 1.5 to 2 times more expensive than the direct computation of the dielectric function. However, the current G_0W_0 implementation in `exciting` scales quartically with the number of basis functions. Therefore, we are confident that the increased cost of calculating the dielectric function is small compared to the speedup we expect from the reduced number of required basis functions.

We have also implemented two different solvers for the radial Sternheimer equation in the computer package `exciting`, as described in Section 4.2. Prior to this, we have compared RK solvers of different orders implemented in Python, which have allowed us to choose the RK solver with the optimal speed to precision ratio for our application. We have shown that the loss in precision of the second-order RK solver is relatively large, especially for high values of ω. Therefore we have decided to use a RK solver of higher order. The fourth-order RK solver produces errors that we find to be sufficiently small compared to the eighth-order RK solver, especially considering that the computed radial response functions are used only as basis functions. They need to be less precise than, for example, the actual wave functions. Also, the basis functions contain parameters like the linearization energies which need to be approximated. Typical variations of the trial energies lead to differences larger than the precision we achieve with the different solvers. Therefore, we have implemented the fourth-order RK solver in the `exciting` code. We have validated our new implementation by comparing to the fourth-order RK solver implemented in Python. The maximum differences are of order 10^{-9}, which gives us confidence, that our implementation is valid. In addition to that, we have implemented a Bulirsch-Stoer solver in `exciting` that is able to handle ODEs with inhomogeneities. Since the Bulirsch-Stoer method is the method currently used to compute the unperturbed basis functions in `exciting`, we have taken a closer look at the comparison between the fourth-order RK and the Bulirsch-Stoer solvers to make sure that the loss of precision is of acceptable magnitude. We have found very small discrepancies of the order of 10^{-8} for relatively weakly varying radial functions. We have found larger errors of the order of 10^{-3} for radial functions that vary more strongly, e.g. for large values of ω. Overall, these errors are still at least two orders of magnitude smaller than the defined baseline. We consider such errors to be sufficiently small. We also need to take into consideration that the Bulirsch-Stoer solver is about three times more expensive than the fourth-order RK solver. As mentioned above, the main goal of the IBC is to reduce computational cost by reducing the basis set size. This comes with the cost of computing a basis response function for every basis function perturbed by every MPB function used. This additional computational effort scales quadratically with the basis set size, while the computation of the self-energy in general scales quartically. For relatively large systems, this difference in scaling makes the added computational cost insignificant. For small systems, however, it is still important to keep the additional cost as low as possible while maintaining the required precision. For that, we decide to use the fourth-order RK solver for the solution of the radial Sternheimer equation. When particularly high values of ω are used, it may be advised to probe the use of the Bulirsch-Stoer solver. Switching between solver types can be done by simply changing the value of one input parameter. We plan to automatize the solver choice depending on the value of ω.

In a next step, we want to compute the BR correction term for the polarizability. As shown by Betzinger and coworkers [22], this correction has the largest influence on the number of required basis functions. When this correction term is implemented, we will perform first convergence tests with respect to the number of unoccupied states and local orbitals. Besides looking at the reduction in required basis functions due to the BR correction, we also need to examine the actual computational cost. This is done to make sure that the additional computational cost of explicitly computing the SOS-term of the polarizability as well as the correction terms is small compared to the computation time we save by reducing the basis set size. Finally, we will implement the Pulay correction term and perform the same tests. This will give us information on what level of correction is useful for what type of material to reduce the computational cost. We also expect to find materials for which the IBC is essential to obtain converged results.

A. Polarizability in the IBC

By plugging the corrected response function Eq. 2.87 in to the expression for the polarizability matrix, Eq. 2.39, the polarizability matrix can be divided into three terms, the usual Adler-Wiser sum-over-states term

$$\begin{aligned}P_{IJ}^{(\mathrm{SOS})}(\boldsymbol{q},\omega) =&\sum_{\boldsymbol{k}}^{BZ}\sum_{i}^{\mathrm{occ}}\sum_{j}^{\mathrm{unocc}} \left[M_{ij}^{I}(\boldsymbol{k},\boldsymbol{q})\right]^{*} M_{ij}^{J}(\boldsymbol{k},\boldsymbol{q}) \\ &\times\left\{\frac{1}{\epsilon_{i\boldsymbol{k}}-\epsilon_{j\boldsymbol{k}+\boldsymbol{q}}+\omega}+\frac{1}{\epsilon_{i\boldsymbol{k}}-\epsilon_{j\boldsymbol{k}+\boldsymbol{q}}-\omega}\right\},\end{aligned} \tag{A.1}$$

the basis-response term

$$\begin{aligned}P_{IJ}^{(\mathrm{BR})}(\boldsymbol{q},\omega) =&\sum_{\boldsymbol{k}}^{BZ}\sum_{i}^{\mathrm{occ}}\langle\chi_{I}^{\boldsymbol{q}}\ \psi_{i\boldsymbol{k}}|\tilde{\psi}_{i\boldsymbol{k},J\boldsymbol{q}}^{(1)}(\omega)+\tilde{\psi}_{i\boldsymbol{k},J\boldsymbol{q}}^{(1)}(-\omega)\rangle \\ &-\sum_{\boldsymbol{k}}^{BZ}\sum_{i}^{\mathrm{occ}}\sum_{j}^{\mathrm{unocc}}\langle\chi_{I}^{\boldsymbol{q}}\ \psi_{i\boldsymbol{k}}|\psi_{j\boldsymbol{k}+\boldsymbol{q}}\rangle\langle\psi_{j\boldsymbol{k}+\boldsymbol{q}}|\tilde{\psi}_{i\boldsymbol{k},J\boldsymbol{q}}^{(1)}(\omega)+\tilde{\psi}_{i\boldsymbol{k},J\boldsymbol{q}}^{(1)}(-\omega)\rangle,\end{aligned} \tag{A.2}$$

and the Pulay term

$$\begin{aligned}P_{IJ}^{(\mathrm{Pulay})}(\boldsymbol{q},\omega) =&\sum_{\boldsymbol{k}}^{BZ}\sum_{i}^{\mathrm{occ}}\sum_{j}^{\mathrm{unocc}}\langle\chi_{I}^{\boldsymbol{q}}\ \psi_{i\boldsymbol{k}}|\psi_{j\boldsymbol{k}+\boldsymbol{q}}\rangle \\ &\times\left[\frac{\langle\psi_{j\boldsymbol{k}+\boldsymbol{q}}|H-\epsilon_{j\boldsymbol{k}+\boldsymbol{q}}|\tilde{\psi}_{i\boldsymbol{k},J\boldsymbol{q}}^{(1)}(\omega)\rangle}{\epsilon_{i\boldsymbol{k}}-\epsilon_{j\boldsymbol{k}+\boldsymbol{q}}+\omega}+\frac{\langle\psi_{j\boldsymbol{k}+\boldsymbol{q}}|H-\epsilon_{j\boldsymbol{k}+\boldsymbol{q}}|\tilde{\psi}_{i\boldsymbol{k},J\boldsymbol{q}}^{(1)}(-\omega)\rangle}{\epsilon_{i\boldsymbol{k}}-\epsilon_{j\boldsymbol{k}+\boldsymbol{q}}-\omega}\right. \\ &\times\left[\frac{\langle\tilde{\psi}_{j\boldsymbol{k}+\boldsymbol{q},J\boldsymbol{q}}^{(1)}(-\omega)|H-\epsilon_{i\boldsymbol{k}}|\psi_{i\boldsymbol{k}}\rangle}{\epsilon_{i\boldsymbol{k}}-\epsilon_{j\boldsymbol{k}+\boldsymbol{q}}+\omega}+\frac{\langle\tilde{\psi}_{j\boldsymbol{k}+\boldsymbol{q},J\boldsymbol{q}}^{(1)}(\omega)|H-\epsilon_{i\boldsymbol{k}}|\psi_{i\boldsymbol{k}}\rangle}{\epsilon_{i\boldsymbol{k}}-\epsilon_{j\boldsymbol{k}+\boldsymbol{q}}-\omega}\right].\end{aligned} \tag{A.3}$$

Bibliography

[1] Jaan Kiusalaas. *Numerical methods in engineering with Python*. Cambridge university press, 2010.

[2] The SciPy community. Scipy.

[3] Stephan Hüfner. *Photoelectron spectroscopy: principles and applications*. Springer Science & Business Media, 2013.

[4] P Hohenberg and WJPR Kohn. Density functional theory (dft). *Phys. Rev*, 136(1964):B864, 1964.

[5] Walter Kohn and Lu Jeu Sham. Self-consistent equations including exchange and correlation effects. *Physical review*, 140(4A):A1133, 1965.

[6] Lars Hedin. New method for calculating the one-particle green's function with application to the electron-gas problem. *Physical Review*, 139(3A):A796, 1965.

[7] Lars Hedin and Stig Lundqvist. Effects of electron-electron and electron-phonon interactions on the one-electron states of solids. In *Solid state physics*, volume 23, pages 1–181. Elsevier, 1970.

[8] Fabien Bruneval and Miguel AL Marques. Benchmarking the starting points of the gw approximation for molecules. *Journal of chemical theory and computation*, 9(1):324–329, 2013.

[9] Jiří Klimeš, Merzuk Kaltak, and Georg Kresse. Predictive gw calculations using plane waves and pseudopotentials. *Physical Review B*, 90(7):075125, 2014.

[10] Dmitrii Nabok, Andris Gulans, and Claudia Draxl. Accurate all-electron G_0W_0 quasiparticle energies employing the full-potential augmented plane-wave method. *Phys. Rev. B*, 94:035118, Jul 2016.

[11] Hong Jiang and Peter Blaha. Gw with linearized augmented plane waves extended by high-energy local orbitals. *Physical Review B*, 93(11):115203, 2016.

[12] Christoph Friedrich, Mathias C Müller, and Stefan Blügel. Erratum: Band convergence and linearization error correction of all-electron gw calculations: The extreme case of zinc oxide [phys. rev. b10. 1103/physrevb. 83.081101 83, 081101 (r)(2011)]. *Physical Review B—Condensed Matter and Materials Physics*, 84(3):039906, 2011.

[13] Wim Klopper, Keld L Bak, Poul Jørgensen, Jeppe Olsen, and Trygve Helgaker. Highly accurate calculations of molecular electronic structure. *Journal of Physics B: Atomic, Molecular and Optical Physics*, 32(13):R103, 1999.

[14] Fabien Bruneval and Xavier Gonze. Accurate gw self-energies in a plane-wave basis using only a few empty states: Towards large systems. *Physical Review B—Condensed Matter and Materials Physics*, 78(8):085125, 2008.

[15] JA Berger, Lucia Reining, and Francesco Sottile. Ab initio calculations of electronic excitations: Collapsing spectral sums. *Physical Review B—Condensed Matter and Materials Physics*, 82(4):041103, 2010.

[16] JA Berger, Lucia Reining, and Francesco Sottile. Efficient calculation of the polarizability: a simplified effective-energy technique. *The European Physical Journal B*, 85:1–10, 2012.

[17] RM Sternheimer. Electronic polarizabilities of ions from the hartree-fock wave functions. *Physical Review*, 96(4):951, 1954.

[18] Feliciano Giustino, Marvin L Cohen, and Steven G Louie. Gw method with the self-consistent sternheimer equation. *Physical Review B—Condensed Matter and Materials Physics*, 81(11):115105, 2010.

[19] Martin Schlipf, Henry Lambert, Nourdine Zibouche, and Feliciano Giustino. Sternheimergw: A program for calculating gw quasiparticle band structures and spectral functions without unoccupied states. *Computer Physics Communications*, 247:106856, 2020.

[20] Markus Betzinger, Christoph Friedrich, Andreas Görling, and Stefan Blügel. Precise response functions in all-electron methods: Application to the optimized-effective-potential approach. *Phys. Rev. B*, 85:245124, Jun 2012.

[21] Markus Betzinger, Christoph Friedrich, and Stefan Blügel. Precise response functions in all-electron methods: Generalization to nonspherical perturbations and application to nio. *Phys. Rev. B*, 88:075130, Aug 2013.

[22] Markus Betzinger, Christoph Friedrich, Andreas Görling, and Stefan Blügel. Precise all-electron dynamical response functions: Application to cohsex and the rpa correlation energy. *Phys. Rev. B*, 92:245101, Dec 2015.

[23] Andris Gulans, Stefan Kontur, Christian Meisenbichler, Dmitrii Nabok, Pasquale Pavone, Santiago Rigamonti, Stephan Sagmeister, Ute Werner, and Claudia Draxl. Exciting: a full-potential all-electron package implementing density-functional theory and many-body perturbation theory. *Journal of Physics: Condensed Matter*, 26(36):363202, 2014.

[24] Stephen L Adler. Quantum theory of the dielectric constant in real solids. *Physical Review*, 126(2):413, 1962.

[25] J. C. Slater. Wave functions in a periodic potential. *Phys. Rev.*, 51:846–851, May 1937.

[26] O. Krogh Andersen. Linear methods in band theory. *Phys. Rev. B*, 12:3060–3083, Oct 1975.

[27] Elisabeth Sjöstedt, Lars Nordström, and DJ Singh. An alternative way of linearizing the augmented plane-wave method. *Solid state communications*, 114(1):15–20, 2000.

[28] David Singh. Ground-state properties of lanthanum: Treatment of extended-core states. *Physical Review B*, 43(8):6388, 1991.

[29] Takao Kotani and Mark Van Schilfgaarde. All-electron gw approximation with the mixed basis expansion based on the full-potential lmto method. *Solid State Communications*, 121(9-10):461–465, 2002.

[30] Hong Jiang, Ricardo I Gómez-Abal, Xin-Zheng Li, Christian Meisenbichler, Claudia Ambrosch-Draxl, and Matthias Scheffler. Fhi-gap: A gw code based on the all-electron augmented plane wave method. *Computer Physics Communications*, 184(2):348–366, 2013.

[31] William H Press, William T Vetterling, Saul A Teukolsky, and Brian P Flannery. *Numerical recipes.* Cambridge University Press, London, England, 1988.

[32] Norsett Hairer and Wanner. *Solving ordinary differential equations I.* Springer Berlin Heidelberg New York, 1993.

[33] Josef Stoer and Roland Bulirsch. *Introduction to numerical analysis*, volume 1993. Springer, 1980.

[34] Leonhard Euler. De integratione aequationum differentialium. *Novi Commentarii Academiae Scientiarum Petropolitanae*, pages 3–63, 1763.

[35] John Charles Butcher. *Numerical methods for ordinary differential equations.* John Wiley & Sons, 2016.

[36] Wilhelm Kutta. *Beitrag zur näherungsweisen Integration totaler Differentialgleichungen.* Teubner, 1901.

[37] Karl Heun et al. Neue methoden zur approximativen integration der differentialgleichungen einer unabhängigen veränderlichen. *Z. Math. Phys*, 45:23–38, 1900.

[38] Lewis Fry Richardson. Ix. the approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 210(459-470):307–357, 1911.

[39] William B Gragg. On extrapolation algorithms for ordinary initial value problems. *Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis*, 2(3):384–403, 1965.

[40] Eric Harold Neville. *Iterative interpolation*. St. Joseph's IS Press, 1934.

[41] Peter Deuflhard. Order and stepsize control in extrapolation methods. *Numerische Mathematik*, 41:399–422, 1983.

[42] Nomad repository. `https://nomad-lab.eu/prod/v1/gui/search/entries/entry/id/zk1qOPQ8CtLLmazIAs6eTAFG5mbi`.

[43] Nomad repository. `https://nomad-lab.eu/prod/v1/gui/search/entries/entry/id/9ZM_3WTsr5OUWkhApXLrWwLjOk_6`.

Selbstständigkeitserklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbstständig verfasst und noch nicht für andere Prüfungen eingereicht habe. Sämtliche Quellen einschließlich Internetquellen, die unverändert oder abgewandelt wiedergegeben werden, insbesondere Quellen für Texte, Grafiken, Tabellen, Bilder sowie die Nutzung von Künstlicher Intelligenz für die Erstellung von Texten und Abbildungen, sind als solche kenntlich gemacht. Mir ist bekannt, dass bei Verstößen gegen diese Grundsätze ein Verfahren wegen Täuschungsversuchs bzw. Täuschung eingeleitet wird.

Berlin, den 29.11.2024,

H. Kleine

Hannah Kleine