Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

A mortar finite element method for full-potential Kohn-Sham density functional theory

A master thesis submitted to
ETH Zürich

in partial fulfillment of the
requirements for the degree of
Master of Science ETH in Physics

presented by

Lukas Drescher

Supervisors:

Prof. Dr. Matthias Troyer
Department of Physics
ETH Zürich

Prof. Dr. Reinhold Schneider,
Dr. Kersten Schmidt
Department of Mathematics
TU Berlin

July 28, 2014

Abstract

Based on a recent regularity analysis, we develop a non-conforming discretization for the Kohn-Sham equations in density functional theory. For this purpose, we decompose the computational domain into a set of atomic spheres and an intermediate region. These domains are discretized separately using spherical harmonics tensored with radial polynomials on spheres and finite elements in the intermediate domain. The discretizations are coupled at the spherical interfaces using the mortar element method. Based on assumptions on the smoothness of the exchange-correlation potential this allows us to derive an a priori error estimate for the linearized Kohn-Sham problem.

We proceed with the implementation of the described mortar discretization in the hp-finite element solver `concepts`. In particular, we resolve the mortaring conditions for the coupling of two discretization spaces in an abstract framework, which results in a generic algorithm for the construction of the mortar space. We implement this algorithm subsequently in `concepts`. To study the Kohn-Sham equations in the self-consistent limit, we formulate and implement a version of the optimal damping algorithm for the extended Kohn-Sham model. This facilitates the robust computation of the self-consistent density, which will be used in our final numerical experiment to verify our a priori error estimate on a mono-atomic system.

Contents

Acknowledgements

First of all, I would like to thank Kersten Schmidt for initiating this project. If it wasn't for him, I would probably not have had the pleasure to write my master thesis in Berlin. He supported me with all his experience in coding and numerical analysis making lots invaluable suggestions during this project. At the same time as a member of his group I was offered great freedom in research. Thank you very much for providing such pleasant environment to work. I am also indebted to Prof. Schneider for suggesting such a fascinating topic and taking the time to sit down and discuss research topics any time with me. The open atmosphere has made it a pleasure to work with you. In the same breath, I also would like to thank Prof. Troyer in Zurich, who only made this work possible with his support.

Here in Berlin, I should also mention my office colleague, Dirk Klindworth, who patiently took the time to answer all the little questions on- and off-topic during this project. Despite being surrounded by four walls, this office has been a fun place to work. I would also like to thank Adrien Semin and Anastasia Zueva for their helpful advice and their sociability. This also applies to my student colleagues Robert Gruhlke, Philipp Kliewe and Maxim Zeinaliyev. Especially, I would like to thank Philipp for supporting me in the beginning of this project.

Finally, I would like to thank all my friends at home, that supported me during this intense time in Berlin. And last but not least, I am deeply grateful for the continuous support I obtain from my family. You are a safe haven to me.

1 Introduction

The computational complexity of quantum many body problems has been a central difficulty in modeling large systems in solid state physics and quantum chemistry since the early days of quantum mechanics. Successful approaches that avoid the high dimensionality posed by the Schroedinger equation date back to the 1920s, when Thomas and Fermi [1] developed an approximate model in terms of the density, that was shown to asymptotically reproduce the ground state energy in the limit of heavy nuclei [2]. Although, Thomas-Fermi theory was the first success in describing quantum many-body systems in terms of their density, it actually took more than thirty years to popularize and refine the density-based approach.

The beginning of modern density functional theory (DFT) is marked by the papers of Hohenberg and Kohn [3] and Kohn and Sham [4]. The former was concerned with theoretical foundations of the field, which were refined by the articles of Levy [5] and Lieb [6]. More important for practical purposes, was the development by Kohn and Sham who suggested an approximation of the energy functional making DFT a computationally tractable tool for the study of materials. The full power of their approach has only become clear in the last 20 years with the advent of modern computers.

Despite the computational power, discretization of the Kohn-Sham problem remains a subtle task due to the nonlinearity. The difficulty is amplified in the case of full-potential calculations. In this setting, plane waves, being the most widely used discretization in DFT, turn out to give disappointingly slow convergence [7]. Despite very efficient FFT schemes, plane waves are, in fact, likely to be outperformed by a less optimized but more adapted basis set that is able to resolve the wave-function near the nuclei. In this respect, methods such as augmented plane waves (APW) and linearized augmented plane waves (LAPW) have been developed [7].

The underlying idea of APW-related methods is to perform a domain decomposition into atomic spheres and an intercellular space. On the intercellular domain, one uses smooth functions that are coupled to eigenstates of the Kohn-Sham Hamiltonian on the atomic spheres over a spherical interface. On the atomic spheres, the Kohn-Sham potential is approximated as rotationally symmetric, which leads to tensor products of spherical harmonics and solutions of a radial Schrödinger equation as eigenfunctions. These eigenfunctions depend on the eigenvalue of the solution of the full eigenvalue problem, thus APW and related methods are inherently non-linear discretizations. Despite the resulting mathematical complexity, the APW-type methods are amongst the most precise techniques available today for electronic structure calculations.

Motivated by recent numerical analysis of an APW-related non-conforming method [8] for full-potential calculations that replaces solutions of the radial Schrödinger equation by radial polynomials, we will investigate a similar discretization technique in this thesis. Precisely, we will couple the functions in the atomic spheres to finite elements using the mortar element method [9]. Paying careful attention to numerical subtleties in this coupling, we expect to preserve the approximation properties of either discretization on the isolated domain. We emphasize that, ultimately, from a numerical analysis perspective it is the regularity of the solution that determines the adequacy discretization. Qualitatively, we expect the solution of the Kohn-Sham problem to have the same singular behavior as the eigenstates of the hydrogen atom near the nuclei, while it is smooth away

from them. Our suggested mortar discretization is exactly designed to capture these features. This is stated more precisely in section 3, where we give a rigorous a priori error analysis for the linearized Kohn-Sham equations under some assumptions on the potentials.

The outline is as follows. In chapter 2 we present the mathematical foundations of density functional theory pointing out approximations due to the Kohn-Sham model. We derive the Kohn-Sham equations and investigate the algorithmically relevant linearized problem for well-posedness. In chapter 3, we define the discretization spaces along with their approximation properties. We then derive an a priori error estimate for our mortar element method that can be applied to the linearized Kohn-Sham problem under certain assumptions on the potentials. In the subsequent chapter 4, we introduce several algorithmic aspects that are important for the implementation of the discretization spaces. We then present an adapted version of the optimal-damping algorithm [10] to solve the extended version of the Kohn-Sham problem in chapter 5. Chapter 6 then proceeds with an outline of the implementation of these parts in the hp-finite element solver `concepts` [11, 12]. In a final chapter, we present some numerical experiments documenting our progress towards a molecular Kohn-Sham solver using the suggested mortar discretization and the optimal damping algorithm.

2 Foundations of Density Functional Theory

We present the foundations of density functional theory (DFT). Using the approximations due to Kohn and Sham, we obtain a computationally tractable problem. We then derive the Euler-Lagrange equations of the Kohn-Sham density functional, widely known as the Kohn-Sham equations. We proceed with an analysis of the linearized Kohn-Sham problem, which appears in each step of iterative solution algorithms for the non-linear Kohn-Sham problem. In particular, we identify suitable spaces that guarantee existence of electronic orbitals as required by the so-called Aufbau principle. To conclude, we cite a regularity result motivating our mortar discretization in the subsequent chapter. Our main references for this chapter are [13] for the model development and [8] for the analysis and regularity of the linearized Kohn-Sham problem.

2.1 The quantum many-body problem and density functional theory

The basic problem in electronic structure theory is the determination of the electronic ground state. We work in the Born-Oppenheimer approximation, where we assume the cores to be fixed.[1] The task is then as follows. Given a molecular system of M nuclei at fixed positions $X_1, \dots, X_M \in \mathbb{R}^3$ with charge numbers $Z_1, \dots, Z_M$, find the electronic N-body state $\psi \in \mathcal{H}^{(N)}$, where $N = \sum_{k=1}^{M} Z_k$, that saturates the lower bound

$$E_0 = \inf_{\psi \in \mathcal{H}^{(N)}} \langle H_N \rangle_\psi \,, \tag{1}$$

where $\langle H_N \rangle_\psi := (H_N \psi, \psi)$ is the expectation value of the Hamiltonian in the state ψ. The space $\mathcal{H}^{(N)}$ consists of all the fermionic N-electron states. That is, it incorporates the Pauli principle through completely antisymmetric wavefunctions

$$\mathcal{H}^{(N)} = \left\{ \psi \in \bigwedge^N L^2(\mathbb{R}^3) \times \Sigma | \ \|\psi\|_{(L^2(\mathbb{R}^3)\times\Sigma)^{\otimes N}} = 1, \ \|\nabla\psi\|_{(L^2(\mathbb{R}^3)\times\Sigma)^{\otimes N}} < \infty \right\},$$

where $\Sigma = \{|\uparrow\rangle, |\downarrow\rangle\}$ is a basis of the one-particle spin space.[2] Explicitly, the norm is given by

$$\|\psi\|_{(L^2(\mathbb{R}^3)\times\Sigma)^{\otimes N}} = \sum_{\sigma_1,\dots,\sigma_N \in \Sigma} \int_{\mathbb{R}^{3N}} |\psi(x_1, \sigma_1, \dots, x_N, \sigma_N)|^2 dx_1 \cdots dx_N.$$

The N-body Hamiltonian can be decomposed into

$$H_N = T_N + V_N + U_N,$$

[1] Therefore, we ignore the associated contribution to the energy of the system. If the positions of the nuclei are not fixed, e.g. in structure optimization, the corresponding optimization can be done in an outer loop.

[2] For this work, we will only be concerned with real-valued wave-functions. This is not a restriction in the case of molecular systems since for any eigenpair (λ, ϕ) of some Hamiltonian H, $(\lambda_i, \bar{\phi}_i)$ will also be an eigenpair and thus real and imaginary part of ϕ are both real-valued eigenstates spanning the same eigenspace.

where T_N represents the kinetic energy, V_N the external Coulomb potential due to cores and U_N the electron-electron interaction. The observables T_N and V_N are obtained by summing up the corresponding one-particle observables

$$T_1(x) = -\frac{1}{2}\Delta_x,$$
$$V_{nuc}(x) = -\sum_{k=1}^{M} \frac{Z_k}{|x - X_k|}$$

over all electron coordinates $x_i \in \mathbb{R}^3$, $1 \leq i \leq N$, while the two-particle Coulomb interaction U_N results from a summation of

$$U(x, y) = \frac{1}{|x - y|}$$

over all electron pairs (x_i, x_j), $1 \leq i < j \leq N$.

The idea of density functional theory is to reformulate the optimization problem (1) in terms of the electron density. In particular, the expectation value of the electron-core Coulomb interaction can be written purely in terms of the electron density,

$$\langle V_N \rangle_\psi = \langle \psi | \, V_N \, | \psi \rangle = \int_{\mathbb{R}^3} V_{nuc} \; \rho_\psi \; dx = \langle V_{nuc} \rangle_{\rho_\psi} \,,$$

where we denoted the density[3] of ψ by ρ_ψ. Optimizing the remaining terms in the expectation value of the N-body Hamiltonian in a first step over all states ψ compatible with a fixed density $\rho_\psi = \rho$, we can write

$$E_0 = \inf_{\rho, \sqrt{\rho} \in H^1(\mathbb{R}^3)} \Big\{ F[\rho] + \langle V_{nuc} \rangle_\rho \Big\} \,, \tag{2}$$

where F is a functional of the density[4],

$$F[\rho] = \inf_{\psi \in \mathcal{H}^{(N)}, \rho_\psi = \rho} \langle T_N \rangle_\psi + \langle U \rangle_\psi \,. \tag{3}$$

Apart from the fact that this functional is formulated on the N-particle state space $\mathcal{H}^{(N)}$ it does not refer any further to the concrete system under consideration. Thus, it is *universal* for all systems.

Although F acts on functions over $\mathbb{R}^3$, the situation is not yet satisfying from a computational point of view. In particular, the high dimensionality still enters the

[3]The density is characterized by the diagonal elements of the N-fold reduced one-particle state via

$$\rho_\psi = \sum_{i=1}^{N} \int_{\mathbb{R}^{3(N-1)}} |\psi(x_1, \ldots, x_N)|^2 \; dx_1 \cdots \widehat{dx_i} \cdots dx_N$$
$$= N \int_{\mathbb{R}^{3(N-1)}} |\psi(x_1, \ldots, x_N)|^2 \; dx_2 \cdots dx_N$$

where the hat excludes the corresponding factor from the product and the last line follows from antisymmetry of ψ.

[4]We restrict here to closed-shell systems for simplicity. A generalization is provided by spin-density functional theory.

definition of F implicitly. To make this approach computationally tractable, we need an explicit formula for F in terms of the density ρ. Unfortunately, such an expression is not known and thus one resorts to approximations. The two main approaches are to either approximate F directly as a function of the density in spirit of Thomas-Fermi (leading to orbital-free DFT) or to introduce an orbital-based model to compute the expectation values (3). In applications the latter group dominates by far, the most important representative being the Kohn-Sham model.

2.2 The Kohn-Sham formalism

The Kohn-Sham approach recasts the minimization problem of density functional theory (2) using an orbital-based approach. Precisely, the idea is to approximate the expectation values in (3) restricting the minimization to Slater-determinants instead of general states. Slater determinants are antisymmetric tensor-products of one-electron orbitals[5],

$$\mathcal{S}^{(N)} := \left\{ \psi = \frac{1}{N!} \wedge_{i=1}^{N} \psi_i |\ \psi_i \in \mathcal{H}^{(1)},\ \psi_i \perp \psi_j \right\}.$$

In particular, one can evaluate the kinetic energy $\langle T_N \rangle_\psi$ for such states explicitly, which motivates

$$T_{KS}[\rho] = \inf_{\psi \in \mathcal{S}^{(N)},\, \rho_\psi = \rho} \sum_{i=1}^{N} \langle T_1 \rangle_{\psi_i}. \tag{4}$$

As a consequence of the restriction, the Kohn-Sham kinetic energy is an upper bound to the minimal kinetic energy over all compatible states compatible with a given density. The electron-electron interaction is approximated by the classical electrostatic self-energy of the density,

$$J[\rho] = \frac{1}{2} \int_{\mathbb{R}^3 \times \mathbb{R}^3} \frac{\rho(x)\rho(y)}{|x-y|} dxdy,$$

which only coincides with the Coulomb energy of the quantum state if the corresponding two-particle density is uncorrelated. In contrast to the Hartree-Fock method, where exchange terms of the orbitals due to the Pauli-principle arise due to the strict minimization in (1) over Slater-determinants, the Kohn-Sham formalism does not include this contribution directly. Instead the latter aims to solve the N-body problem in full generality by compensating the error due to the negligence of the Pauli-principle (exchange) and the restriction to Slater-determinants (correlation) collected in

$$E_{xc}[\rho] := F[\rho] - T_{KS}[\rho] - J[\rho].$$

This is called the exchange-correlation functional. Thus the N-body Schrödinger problem (1) reformulated in the setting of Kohn-Sham density functional theory is to find the minimizer of

$$E_0 = \inf_{\rho, \sqrt{\rho} \in H^1(\mathbb{R}^3)} T_{KS}[\rho] + E_n[\rho] + J[\rho] + E_{xc}[\rho]. \tag{5}$$

[5]The expansion in terms of one-electron orbitals $\{\psi_i\}_{1 \le i \le N}$ is unique up to unitaries in $U(N)$. Since, however, all considered N-body observables are invariant under conjugation with a unitary, the concrete expansion does not matter.

From a computational perspective, this does not amount to a simplification of (1) since the complicated N-body wave-function dependence has only been moved from F into E_{xc}. The basic achievement of Kohn and Sham in [4] was the suggestion of the local density approximation (LDA) that made this problem computationally tractable. In this framework the exchange-correlation energy is approximated by

$$E_{xc}^{LDA}[\rho] = \int_{\mathbb{R}^3} \rho \quad \mathcal{E} \circ \rho \, dx,$$

where the energy density exhibits a local dependence on the density through the function $\mathcal{E} : \mathbb{R}_+ \to \mathbb{R}$. Usually, the homogeneous electron gas is used as a reference for the function $\mathcal{E}$. Physically, the contributions to $\mathcal{E}$ are split into an exchange and a correlation term. The exchange part is taken as

$$\mathcal{E}_x(\rho(x)) = -\frac{3}{4\pi}(3\pi^2\rho(x))^{\frac{1}{3}}, \tag{6}$$

while correlations are approximated using an interpolation between low and high-density limit analytical expressions and results of quantum Monte Carlo simulations in between. In this thesis we will use the approximation of Perdew and Wang [14]

$$\begin{aligned}\mathcal{E}_c(\rho(x)) = &- 2c_0(1+\alpha_1 r_s(x))\times \\ &\log\left(1+\frac{1}{2c_0(\beta_1 r_s(x)^{\frac{1}{2}}+\beta_2 r_s(x)+\beta_3 r_s(x)^{\frac{3}{2}}+\beta_4 r_s(x)^2)}\right),\end{aligned} \tag{7}$$

where the specific radius is given by $r_s(x) = \left(\frac{3}{4\pi\rho(x)}\right)^{\frac{1}{3}}$. The constants are not written out explicitly here, but can be found in [14]. There is a wealth of more elaborate approximations to the exchange-correlation functional such as the generalized gradient approximation (GGA) that includes further dependence of $\mathcal{E}$ on the gradient of the density. However, for the purpose of studying different discretization schemes for the Kohn-Sham equations, we will restrict to LDA for this thesis.

2.2.1 The Kohn-Sham equations

We derive the Kohn-Sham equations as the Euler-Lagrange equations of the Kohn-Sham energy functional (5). The Kohn-Sham energy functional can be defined in terms of Slater determinants as

$$E^{KS}[\psi] := \sum_{i=1}^{N} \langle h \rangle_{\psi_i} + J[\rho_\psi] + E_{xc}[\rho_\psi], \quad \psi \in \mathcal{S}^{(N)} \tag{8}$$

where $\psi = \frac{1}{N!}\wedge_{i=1}^{N}\psi_i$. The operator $h = T_1 + V_{nuc}$ is called the one-particle Hamiltonian. A critical point ψ of this functional is characterized by

$$\frac{d}{dt}E^{KS}[\psi_t]\big|_{t=0} = 0 \tag{9}$$

for any curve $\psi_\cdot : (-\varepsilon, \varepsilon) \to \mathcal{S}^{(N)}$, with $\psi_0 = \psi$. It is sufficient to consider variations involving only a single factor,

$$\psi_t = \frac{1}{N!}\,\psi_1 \wedge \ldots \wedge \psi_{i,t} \wedge \ldots \wedge \psi_N,$$

with $\psi_{i,t} \in \mathcal{H}^{(1)}$, orthogonal to all ψ_j with $j \neq i$, since those span the tangent space of $\mathcal{S}^{(N)}$ at ψ[6]. The critical point condition (9) then becomes

$$\frac{d}{dt} E^{KS}[\psi_t]\big|_{t=0} = 2\,\langle \dot{\psi}_i|\, h + V_H[\rho_\psi] + V_{xc}[\rho_\psi]\,|\psi_i\rangle = 0, \tag{10}$$

where $\dot{\psi}_i = \frac{d\psi_i}{dt}(0)$ can be any element in $H^1(\mathbb{R}^3)$ orthogonal to the subspace spanned by $\{\psi_j\}_{1\leq j\leq N}$. The electrostatic potential $V_H[\rho]$ and the exchange-correlation potential $V_{xc}[\rho]$ are the Frechet derivatives of the corresponding energy functionals $J[\rho]$ and $E_{xc}[\rho]$ with respect to the density ρ,

$$\begin{aligned} V_H &: \rho \mapsto \rho * \frac{1}{|x|} \\ V_{xc} &: \rho \mapsto \mathcal{E} \circ \rho + \rho\, \mathcal{E}' \circ \rho. \end{aligned} \tag{11}$$

Note that the Hartree operator $V_H : L^2(\mathbb{R}^3) \to L^2(\mathbb{R}^3)$ is self-adjoint. We denote the density-dependent Kohn-Sham Hamiltonian at ψ by

$$H_{\rho_\psi} := h + V_H[\rho_\psi] + V_{xc}[\rho_\psi].$$

Condition (10) by reflexivity of $H^1(\mathbb{R}^3)$ is equivalent to the fact that $\mathrm{span}\{\psi_j\}_{1\leq j\leq N}$ is an invariant subspace of H_{ρ_ψ}. Since H_{ρ_ψ} is self-adjoint, we can diagonalize it in an orthonormal basis of that subspace. To obtain a local minimum of the Kohn-Sham energy functional (5), we further need the *Aufbau principle* [10] which requires the eigenvalues of the ψ_i to be the lowest lying ones of H_{ρ_ψ}. In total, we arrive at the strong form of the Kohn-Sham equations:

$$\boxed{\begin{aligned} &\text{Find eigenpairs } (\lambda_i, \phi_i)_{i=1}^N \subset \mathbb{R} \times H^1(\mathbb{R}^3),\ (\phi_i, \phi_j) = \delta_{ij}, \\ &\text{satisfying the Aufbau principle s.t. } H_{\rho_\phi}\phi_i = \lambda_i \phi_i \end{aligned}} \tag{12}$$

The brackets $(\,\cdot,\cdot)$ here denote the L^2-inner product on $\mathbb{R}^3$. The associated weak formulation can be obtained by testing against functions $v \in H^1(\mathbb{R}^3)$ so that the last expression in (12) is replaced by

$$\boxed{a_{\rho_\phi}(\phi_i, v) = \lambda_i(\phi_i, v),\ \forall v \in H^1(\mathbb{R}^3)} \tag{13}$$

with $a_{\rho_\phi}(\phi, v) := (H_{\rho_\phi}\phi, v)$. Using Gauss' theorem, the kinetic energy part of a_{ρ_ϕ} can be written in terms of an L^2-inner product of gradients.

Equation (12) possesses a high degree of similarity to a single-particle Schrödinger equation. For this reason, the Kohn-Sham model is sometimes referred to as a set of non-interacting particles that move in an effective one-particle system. Note, however, that in crucial difference to a proper Schrödinger problem on $\mathbb{R}^3$, the Kohn-Sham problem (12) is inherently a many-body problem. In particular, the density ρ_ϕ couples all electron orbitals ϕ_i, $1 \leq i \leq N$, through the non-linear terms in the potentials of H_{ρ_ϕ}. A particular consequence is that the sum of the eigenvalues in (12) cannot be interpreted as the energy

[6]The derivative of a general variation will be the sum of derivatives hitting only a single factor by the product rule.

of the state ϕ, which is given by $E^{KS}[\phi]$. Rather it is the expectation value of the Kohn-Sham Hamiltonian H_{ρ_ϕ} in the state ϕ. To summarize the analogy to the single-particle scenario, the Kohn-Sham formalism requires an effective single-particle system to be self-consistent with its ground state.

The non-linearity in (13) also implies that mathematical techniques required for existence results of the ground state of (5) are much more sophisticated than for linear Schrödinger equations. Recently, progress has been made in this respect with important contributions in [15] for the above standard Kohn-Sham model using LDA and in [13] for an extension to mixed states and more general LDA- and GGA exchange-correlation functionals.

2.3 The linearized Kohn-Sham problem: Existence and regularity of eigenpairs

To solve the non-linear eigenvalue problem (13) numerically one usually resorts to iterative methods requiring the solution of the linearized Kohn-Sham problem in every step. Before starting with a discretization, we therefore analyze the well-posedness of this problem in terms of existence of a solution according to the Aufbau principle. Moreover, we cite a regularity result on which our discretization using mortar elements in the subsequent chapter is based. Our main reference for this subsection is [8].

2.3.1 Existence of an Aufbau solution

The linearized Kohn-Sham problem corresponds to (13) relaxing the self-consistency requirement by fixing the density ρ. In contrast to the original Kohn-Sham problem, one then obtains a true one-particle Schrödinger problem. We will consider this problem here on a bounded computational domain Ω with smooth boundary and homogeneous Dirichlet boundary conditions[7]. The linearized Kohn-Sham problem for a given density ρ, $\sqrt{\rho} \in H_0^1(\Omega)$, then is:

$$\boxed{\begin{aligned}&\text{Find eigenpairs } (\lambda_i, \phi_i)_{i=1}^N \subset \mathbb{R} \times H_0^1(\Omega),\ (\phi_i, \phi_j) = \delta_{ij},\\ &\text{satisfying the Aufbau principle s.t. } a_\rho(\phi_i, v) = \lambda_i(\phi_i, v),\ \forall v \in H_0^1(\Omega)\end{aligned}} \tag{14}$$

We can then formulate the following theorem for the linearized LDA Kohn-Sham model.

Theorem 2.1. *Let ρ such that $\sqrt{\rho} \in H_0^1(\Omega)$, and $V_{xc}[\rho] \in L^\infty(\Omega)$ for V_{xc} as in (11). Then there exists a (possibly degenerate) solution to (14).*

The proof of this Theorem is sketched in [8] and analogous to most other existence proofs for Schrödinger eigenvalue problems. Therefore, we reduce the presentation to the essentials.

[7] The problem can analogously be formulated for other boundary conditions without affecting the statement in Theorem 2.1.

The basic idea is to recast the variational eigenvalue problem (14) as an operator eigenvalue problem $\phi_i = \tilde{\lambda}_i T\phi_i$, where T is the Lax-Milgram solution operator of the class of boundary value problems

$$\tilde{a}(T\phi, v) = (\phi, v),\ \forall v \in H_0^1(\Omega)$$

for $\phi \in H_0^1(\Omega)$. In order for T to be well-defined and continuous we need continuity and coerciveness of $\tilde{a}$ according to the Lax-Milgram Lemma. The bilinear form a_ρ does not have the latter property by default due to negative-valued potentials. However, the lowest orbital energy of a_ρ can be bounded by some $\nu \in \mathbb{R}$ via

$$\begin{aligned} a_\rho(\phi,\phi) &\geq \frac{1}{2}|\phi|_1^2 - \|V_{nuc}\|_{L^2}\|\phi\|_{L^2}^{\frac{1}{2}}\|\phi\|_{L^6}^{\frac{3}{2}} - \underbrace{\|V_H[\rho]\|_{L^2}}_{\lesssim\|\rho\|_{L^2}}\|\phi\|_{L^4}^2 - \|V_{xc}[\rho]\|_{L^\infty}\|\phi\|_{L^2}^2 \\ &\geq \frac{1}{2}|\phi|_1^2 - \nu\|\phi\|_{L^2}^2, \end{aligned}$$

where we used that V_H is continuous [16] (c.f. [8]). Then, the bilinear form $\tilde{a}(\cdot,\cdot) = a_\rho(\cdot,\cdot) + \nu(\cdot,\cdot)$ is coercive. Thus for every eigenpair $(\tilde{\lambda}, \phi)$, the orbital ϕ will be an eigenvector of (14) with eigenvalue $\tilde{\lambda} - \nu$. Since the embedding of $H_0^1(\Omega)$ in $L^2(\Omega)$ is compact (Rellich's theorem [17]), the operator T restricts to a compact, self-adjoint operator on $H_0^1(\Omega)$ (the latter by symmetry of $\tilde{a}$). The spectral theorem for compact self-adjoint operators on a separable Hilbert space then guarantees the existence of a solution according to the Aufbau principle.

2.3.2 Regularity of the solutions for the modified linearized Kohn-Sham problem

We motivate our discretization of the linearized Kohn-Sham problem (14) by a regularity result from [18]. For this purpose, we introduce weighted Sobolev spaces on a sphere B_R. In particular, we consider a subspace of functions admitting an expansion of their singular part at the center in products of spherical harmonics with radial polynomials. Such a smoothness property can be recast in terms of usual Sobolev spaces on the open stretched cone $[0, R]\times S^2$, which will be the basis of our discussion of the a priori estimate in section 3.5.

Let the weighted Sobolev space $\mathcal{K}^{k,\gamma}$, $k \in \mathbb{N}_0$, $\gamma \in \mathbb{R}$ be the set of functions $u \in L^2(B_R)$ that satisfy

$$r^{|\alpha|-\gamma}\partial^\alpha u \in L^2(B_R)\ \forall \alpha \in \mathbb{N}_0^3,\ |\alpha| \leq k.$$

The role of γ is to measure the regularity of u. The higher γ, the weaker the singularity at the origin, e.g.

$$u \in \mathcal{K}^{k,\gamma} \Rightarrow |\partial^\alpha u| \lesssim r^{\gamma-\frac{3}{2}-|\alpha|}\ \forall \alpha,\ |\alpha| \leq k.$$

This space can further be refined by selecting out functions $u \in \mathcal{K}^{k,\gamma}$ whose singular part at the center admits an expansion

$$v_n = \sum_{j=0}^{n} r^j Y_j$$

for $n \in \mathbb{N}$, where Y_j is a linear combination of spherical harmonics up to maximum angular momentum[8] j. In terms of Sobolev spaces, this is formulated using a smooth cut-off function ω supported near the center such that

$$\forall n \in \mathbb{N} : \; u - \omega \cdot v_n \in \mathcal{K}^{k,\gamma+n}.$$

This means that all partial derivatives up to k-th order can be expanded around the center to arbitrary order in terms of radial polynomials and spherical harmonics, leaving an ever more regular residual $u - \omega \cdot v_n$ as $n \to \infty$. The functions that satisfy this property in all their derivatives for any $\gamma < \frac{3}{2}$ are called *asymptotically well-behaved*.

Importantly, such a characterization of the singular part of a function in $\mathcal{K}^{k,\gamma}$ immediately implies Sobolev-regularity on the open stretched cone,

$$u \in \mathcal{K}^{k,\gamma} \Rightarrow u \in H^{k-\frac{3}{2}}([0,R] \times S^2)$$

as was shown in [8]. Especially, asymptotically well-behaved functions achieve any regularity on $[0,R] \times S^2$.

The important observation in our context is captured in the following Lemma that applies to a modified version of the linearized Kohn-Sham problem (14). It is an adapted version of Lemma 2.1 in [8].

Lemma 2.2. *Assume the exchange-correlation potential is smooth, $V_{xc}[\rho] \in C^\infty(\Omega)$. Then the eigenfunctions to* (14) *are asymptotically well-behaved.*

We do not present a proof of this Lemma here, but assume the argument from [18] that leads to the result in [8] to carry over to the present setting.

[8] A spherical harmonic $Y \in C^\infty(S^2)$ is said to have angular momentum $l \in \mathbb{N}_0$ if it is an eigenfunction of the spherical Laplacian with eigenvalue $-l(l+1)$.

3 Discretization of the linearized Kohn-Sham problem: Theory

The linearized Kohn-Sham problem (14) is discretized in a general framework using the Galerkin method. Decomposing the domain of the linearized Kohn-Sham problem into a set of atomic spheres and an interstitial region, we introduce discrete domain spaces based on tensor products of spherical harmonics with radial polynomials and finite elements. These spaces are coupled via the mortar element method. We derive an a priori error estimate for our mortar element discretization with respect to an increase of the polynomial degree (p-refinement) for a set of Schroedinger problems that generalize (14) predicting super-algebraic convergence under some regularity assumptions on the potential.

3.1 Galerkin discretization

The Galerkin approach for the discretization of (14) consists of replacing the infinite-dimensional space $H_0^1(\Omega)$ by a finite dimensional space V_n to obtain the discrete problem,

$$\begin{aligned} &\text{Find eigenpairs } (\lambda_i, \phi_i)_{i=1}^N \subset \mathbb{R} \times V_n,\ (\phi_i, \phi_j)_{L^2} = \delta_{ij}, \\ &\text{satisfying the Aufbau principle s.t. } a_\rho(\phi_i, v) = \lambda_i (\phi_i, v)_{L^2},\ \forall v \in V_n. \end{aligned} \tag{15}$$

If $V_n \subset H_0^1(\Omega)$, the discretization is called conforming and we obtain well-posedness in the sense of the Aufbau principle as an immediate consequence of the infinite-dimensional case. Otherwise, the discretization is called non-conforming and well-posedness must be proven separately.

In this chapter, we will analyze a non-conforming domain decomposition approach for the discretization of the linearized Kohn-Sham equations. In particular, we decompose the domain Ω in (14) into a set of non-overlapping atomic spheres $\mathcal{C}_i = B_{R_i}(X_i)$ of radii R_i centered over the cores in X_i, $1 \leq i \leq M$ and an intermediate domain $\mathcal{D}$. An example geometry for a mono-atomic system is presented in figure 1. On the atomic spheres $\{\mathcal{C}_i\}_{1 \leq i \leq M}$ we will use spherical function spaces, while on the intermediate domain $\mathcal{D}$ hexahedral finite elements are employed. The corresponding discretization spaces will be defined in the following sections forming the basis for their coupling in the mortar element method, subsequently.

3.2 Finite elements

On the finite element domain, the Galerkin space is built from continuous piecewise smooth functions on a hexahedral mesh[9]. A mesh $\mathcal{T} = \{K_i\}_{i \in I}$ is a discretization of the domain Ω into elements $K_i \subset \Omega$ so that

$$\bar{\Omega} = \bigcup_{i \in I} \bar{K}_i,$$

where every element is the diffeomorphic image of some canonical reference domain,

$$F_{K_i} : \hat{K}_i \to K_i.$$

[9]For simplicity, we restrict to hexahedral elements in this work.

Throughout this thesis, the reference domain will be $\hat{K}_i = (0,1)^3 \equiv \hat{K}$. The mappings $\{F_{K_i}\}_{i\in I}$ must furthermore be compatible across element boundaries, i.e. their continuation must be such that the coordinate changes

$$F_{K_j} \circ F_{K_i}^{-1} : F_{K_i}(\bar{K}_i \cap \bar{K}_j) \to F_{K_j}(\bar{K}_i \cap \bar{K}_j) \tag{16}$$

are affine maps $\forall i, j \in I$. In the a priori error analysis of the mortar method, we will, furthermore, call the mesh $\mathcal{T}$ smooth, if the parametrizations F_{K_i} extend to a diffeomorphism on an open cover of $\hat{K}_i$ and K_i. This is not strictly required by the finite element method, however, it simplifies approximation proofs considerably and will be satisfied by all meshes considered in this work. For this chapter, we also focus exclusively on regular meshes, that is any two adjacent elements geometrically intersect on a common face, edge or vertex.

The global finite element space on the mesh $\mathcal{T} = \{K_i\}_{i\in I}$ is defined by pushing forward the local polynomial space[10] $\mathcal{Q}_p(\hat{K}_i) := \mathcal{P}_p(0,1)^{\otimes 3}$ along the mappings $\{F_{K_i}\}_{i\in I}$. Continuity is enforced across the element interfaces in order to obtain a conforming discretization for H^1-elliptic problems. The resulting Galerkin space is denoted by

$$S^p(\Omega, \mathcal{T}) := \{u \in C^0(\Omega) |\ u \circ F_{K_i} \in \mathcal{Q}_p(\hat{K}_i)\ \forall i \in I\}. \tag{17}$$

The approximation properties of $S^p(\Omega, \mathcal{T})$ presented in the following Lemma.

Lemma 3.1. *Let $u \in H^s(\Omega)$, $s > \frac{3}{2}$, and $\mathcal{T}$ be a smooth mesh. Then there exists a best-approximation $v \in S^p(\Omega, \mathcal{T})$ that satisfies*

$$\inf_{v\in S^p(\Omega,\mathcal{T})} \|u - v\|_{H^1(\Omega)} \leq C_{s,\mathcal{T}} p^{-(s-1)} \|u\|_{H^s(\Omega)},$$

where $C_{s,\mathcal{T}}$ is a constant independent of p.

The proof of this Lemma is analogous to the case of p-FEM on affine elements [19]. The only difference arises in the transformation behavior of Sobolev norms when switching between the physical and the reference element. In the affine case, the derivatives of F_{K_i} and $F_{K_i}^{-1}$ are constant. Thus they allow straight-forward estimation of the scaling of Sobolev norms under transformation in terms of the geometry of the mesh, which is necessary if one studies geometric refinement (h-FEM). However, in contrast to h-FEM, refinement in the polynomial degree (p-FEM) only requires the transformation constant to be finite, i.e. that the Sobolev norms on the physical and the reference element are equivalent at any given order.

The equivalence of Sobolev norms follows from the smoothness of the mesh $\mathcal{T}$. In particular, each F_{K_i} can be extended to a smooth diffeomorphism on an open cover of $\bar{K}_i$. Due to compactness of $\bar{K}_i$, partial derivatives of every F_{K_i} up to any given order $k \in \mathbb{N}$ are uniformly bounded. The same applies to the inverse $F_{K_i}^{-1}$. Thus, by Lemma 1 in [20] the the Sobolev norms of $H^s(\hat{K}_i)$ and $H^s(K_i)$ are equivalent.

For the proof of the appproximation properties of the mortar coupled space, we require the following approximation result for p-FEM on the spherical surface S^2.

[10] The set $\mathcal{P}_p(0,1)$ the spaces of polynomials up to degree p.

Lemma 3.2. *Let $u \in H^s(S^2)$ and $\mathcal{T}$ be a smooth mesh on S^2 such that the closure of every element is contained in an open half-sphere. The best-approximation error satisfies*

$$\inf_{v \in S^p(S^2,\mathcal{T})} \|u - v\|_{H^1(S^2)} \leq C_{s,\mathcal{T}} p^{-(s-1)} \|u\|_{H^s(S^2)},$$

where $C_{s,\mathcal{T}}$ is a constant independent of p.

This Lemma follows by techniques from approximation theory for p-FEM in two dimensions [21] and a similar Sobolev norm equivalence as in the remark following Lemma 3.1. Precisely, one can introduce suitably chosen element-wise spherical coordinates

$$G_{K_i} : \tilde{K}_i \subset (0, 2\pi) \times (0, \pi) \to S^2$$

on an open cover of K_i. The H^s-Sobolev norm on $G_{K_i}^{-1}(K_i)$ induced by the spherical metric can be shown to be equivalent to that induced by the euclidean metric. Uniform boundedness of the derivatives of the coordinate change $G_{K_i}^{-1} \circ F_{K_i}$ and its inverse at any given order then establishes the desired equivalence of Sobolev norms $H^s(K_i)$ and $H^s(\hat{K}_i)$.

The following Lemma related to [22], which is an adapted version of a basic property used in approximation proofs for p-FEM. We will use it in section 3.4.

Lemma 3.3. *Let $\mathcal{T}$ be a smooth mesh on Ω with boundary Γ. There exists a sequence of lifting operators for polynomials*

$$L_p : S^p(\Gamma, \mathcal{T}\big|_\Gamma) \subset H^{\frac{1}{2}}(\Gamma) \to S^p(\Omega, \mathcal{T}) \subset H^1(\Omega).$$

that is uniformly continuous in p.

Finally, we need a stability property of the L^2-orthogonal projector on the sphere. Since, however, we do not know a strict proof of this result, we present it as a conjecture.

Conjecture 3.4. *Let $\mathcal{T}$ be a smooth mesh on S^2. The L^2-orthogonal projector $\Pi_p : H^{\frac{1}{2}}(S^2) \to S^p(S^2, \mathcal{T}) \subset H^{\frac{1}{2}}(S^2)$ is continuous uniformly in p.*

3.3 Spherical space

On every atomic sphere we use a discrete space composed of a tensor products of radial polynomials and spherical harmonics. Spherical harmonics are characterized as the H^1-complete set of eigenfunctions of the spherical Laplacian,

$$-\Delta_{S^2} Y = \lambda Y, \tag{18}$$

They are smooth and the eigenvalues are of the form $\lambda = l(l+1)$, $l \in \mathbb{N}_0$. The corresponding eigenspaces will be denoted by E_l and are $(2l+1)$-dimensional. We define the spherical shape function space on the sphere $\mathcal{C} := B_1(0)$ as the tensor product of spherical harmonics with radial polynomials up to same order L via

$$\mathcal{S}_L(\mathcal{C}) = \mathcal{P}_L(0,1) \otimes \left(\bigoplus_{0 \leq l \leq L} E_l. \right) \tag{19}$$

This space is $H^1(\mathcal{C})$-conforming since any function $q \otimes Y \in \mathcal{S}_L$ satisfies

$$\|\nabla(q \otimes Y)\|^2 = \|r\partial_r q\|_{L^2}^2 \|Y\|_{L^2}^2 + \|q\|_{L^2}^2 \|\nabla_{S^2} Y\|_{L^2}^2 < \infty,$$

where $\|\nabla_{S^2} Y\|_{L^2}^2 \leq L(L+1)\|Y\|_{L^2}^2$ due to Stokes theorem[11] and orthogonality of the eigenspaces E_l. However, in contrast to finite elements, functions in $\mathcal{S}_L(\mathcal{C})$ are not necessarily continuous, e.g. $1 \otimes Y$ can have arbitrarily rapid oscillations at the center of the sphere. Ignoring this fact for the moment, the push-forward of shape functions along a translation by $X_i \in \mathbb{R}^3$ and a radial dilation by R_i yields the discrete space i-th atomic sphere $\mathcal{C}_i$, which we denote by $\mathcal{S}_L(\mathcal{C}_i)$.

The eigenvalue problem (18) is also the basis of spectral approximation theory for spherical harmonics. Tensorizing an orthogonal projector on the space $\bigoplus_{0 \leq l \leq L} E_l$ with an appropriate one for radial polynomials one can construct an explicit best-approximations in $\mathcal{S}_L(B_R(0))$ for functions defined on the open stretched cone [8].

Lemma 3.5. *For $u \in H^s([0,R] \times S^2)$ there exists $v \in \mathcal{S}_L(B_R(0))$ s.t.*

$$\inf_{v_L \in S_L} \|u - v\|_{H^1([0,R]\times S^2)} \leq C_s L^{-(s-1)} \|u\|_{H^s([0,R]\times S^2)},$$

where C_s is a constant independent of L.

3.4 Mortar element space

The mortar element method [9] is a domain decomposition approach that couples different discretization schemes on adjacent subdomains using a weak continuity constraint on the common interface. In our case, we decompose the domain Ω in (14) into a set of non-overlapping atomic spheres $\{\mathcal{C}_i\}_{1 \leq i \leq M}$ and an intermediate domain $\mathcal{D}$. The atomic spheres will be discretized in terms of

$$\mathcal{S}_p\left(\bigcup_{i=1}^{M} \mathcal{C}_i\right) := \bigoplus_{i=1}^{M} \mathcal{S}_p(C_i),$$

while on $\mathcal{D}$ we will use the finite element space $S^p(\mathcal{D}, \mathcal{T})$ for some smooth mesh $\mathcal{T}$. The direct sum of these domain spaces is a subspace of the broken Sobolev space

$$H_\delta := \left\{ u \in L^2(\Omega) \middle| \; u\big|_{\mathcal{C}_i} \in H^1(\mathcal{C}_i) \; , 1 \leq i \leq M, \; u\big|_{\mathcal{D}} \in H^1_{\partial\Omega}(\mathcal{D}) \right\}$$

with associated norm

$$\|v\|_\delta := \sum_{i=1}^{M} \|v\|_{H^1(\mathcal{C}_i)} + \|v\|_{H^1(\mathcal{D})}.$$

To impose weak continuity over the interfaces $\Gamma_i = \partial\mathcal{C}_i$, we define the bilinear form

$$b : \left\{ \begin{array}{ccc} H_\delta \times \mathcal{M} & \to & \mathbb{R} \\ (u, w) & \mapsto & \int_\Gamma [u] w \, dS, \end{array} \right.$$

[11] The gradient of a function $f \in C^1(S^2)$ at $x \in S^2$ is defined as the tangent space representation, $\nabla_{S^2} f(x) \in T_x S^2$, of the differential $Df(x) : T_x S^2 \to \mathbb{R}$ so that $\langle \nabla_{S^2} f(x), v \rangle := Df(x)v$ for all $v \in T_x S^2$.

that tests the trace jump $[u] := u\big|_{\partial \bigcup_i \mathcal{C}_i} - u\big|_{\partial \mathcal{D}}$ against a mortar test function $w \in \mathcal{M}$. The mortaring constraint on a discrete function u then is

$$b(u, w) = 0 \quad \forall w \in \mathcal{M}. \tag{20}$$

The choice of the mortar test space $\mathcal{M}$ must be appropriate in order to obtain a stable discretization. A common approach is to pick one of the trace spaces of the coupled discretizations. In this thesis, we will use the trace space

$$\mathcal{M}_p := S^p(\partial\mathcal{D}, \mathcal{T}\big|_\Gamma)$$

of finite elements, although spherical harmonics are also a conceivable choice. Discretizing every atomic sphere $\mathcal{C}_i$ as well as the intermediate domain $\mathcal{D}$ with the same order p we define the mortar element space as

$$V_p = \left\{ u \in S^p(\mathcal{D}, \mathcal{T}) \oplus \mathcal{S}_p \left(\bigcup_{i=1}^{M} \mathcal{C}_i \right) \;\middle|\; b(u, w) = 0 \; \forall w \in \mathcal{M}_p \right\}. \tag{21}$$

Due to the non-vanishing trace jump, the mortar space is non-conforming in terms of $H^1(\Omega)$. However, it is conforming in the broken Sobolev space H_δ. The linearized Kohn-Sham problem (14) can be extended to this space altering the kinetic energy term in a_ρ to obtain

$$\begin{aligned} a_{\rho,\delta}(u, v) := \frac{1}{2} \sum_{i=1}^{M} \int_{\mathcal{C}_i} \nabla u \cdot \nabla v dx + \frac{1}{2} \int_{\mathcal{D}} \nabla u \cdot \nabla v dx \\ + ((V_{nuc} + V_H[\rho] + V_{xc}[\rho])u, v) \end{aligned} \tag{22}$$

For the associated variational problem we have the following Lemma, whose proof is analogous to that of Theorem 2.1.

Lemma 3.6. *The linearized Kohn-Sham problem (14) on H_δ using $a_{\rho,\delta}$ instead of a_ρ is well-posed.*

For the a priori error analysis, we introduce two further subspaces. To bound the consistency error, we define

$$H_{\delta,p} = \{u \in H_\delta |\; b(u, w) = 0 \; \forall w \in \mathcal{M}_p\}.$$

We will be interested in functions on this space that satisfy a Sobolev regularity on the open stretched cone, thus we set

$$\tilde{H}^s_{\delta,p}(\Omega) := \{u \in H_{\delta,p} |\; u\big|_{\mathcal{C}_i} \in H^s([0, R] \times S^2), u\big|_{\mathcal{D}} \in H^s(\mathcal{D})\}.$$

with associated norm

$$\|u\|_{\tilde{H}^s_{\delta,p}(\Omega)} = \sum_{i=1}^{M} \|u\big|_{\mathcal{C}_i}\|_{H^s([0,R]\times S^2)} + \|u\big|_{\mathcal{D}}\|_{H^s(\mathcal{D})}.$$

The following result that follows from Lemma 2.3 in [8] implies that it is sufficient to analyze the approximation error in the latter norm.

Lemma 3.7. *The inclusion $\tilde{H}^s_{\delta,p}(\Omega) \hookrightarrow H_\delta(\Omega)$ is continuous.*

3.5 A priori error estimate for Schrödinger problems

Based on the regularity result in section 2.3.2, we derive an a priori error analysis for the proposed mortar element discretization of Schrödinger problems. The basic results of our analysis are estimates of the best-approximation and the consistency errors in the Lax-Milgram setting measured in the broken Sobolev-norm of H_δ. By an analogous argument to [8] this implies an a priori error estimate for Schrödinger eigenvalue problems with asymptotically well-behaved eigenfunctions. For simplicity, we will assume that the domain Ω is decomposed using only a sphere $\mathcal{C}$, the generalization to multiple spheres is straight-forward.

3.5.1 Lax-Milgram estimates

We consider variational problems for the extended bilinear form (22) shifted by a lower bound on the eigenvalues ν so that

$$\tilde{a}_\delta \begin{cases} H_\delta \times H_\delta \to \mathbb{R} \\ (u,v) \mapsto a_{\rho,\delta}(u,v) + \nu(u,v)_{L^2} \end{cases}$$

is continuous and coercive. Let $f \in L^2(\Omega)$. Then, by Lax-Milgram there exist unique $u \in H_0^1(\Omega)$, $u_p \in V_p$ such that

$$\begin{aligned} \tilde{a}_\delta(u,v) &= (f,v), & \forall v \in H_0^1(\Omega), \\ \tilde{a}_\delta(u_p,v_p) &= (f,v_p), & \forall v_p \in V_p. \end{aligned} \tag{23}$$

Throughout this section we will assume that the mesh $\mathcal{T}$ of the domain Ω is smooth as described in section 3.2. Using a nonconforming analogue of Céa's Lemma [23][12], we can bound the error by

$$\|u-u_p\|_\delta \lesssim \inf_{v_p \in V_p} \|u-v_p\|_\delta + \sup_{v_p \in V_p} \frac{\tilde{a}_\delta(u-u_p,v_p)}{\|v_p\|_\delta}. \tag{24}$$

The first term is the best-approximation error and the second is the consistency error. We first analyze the latter.

Lemma 3.8. *If $u \in \tilde{H}^s_{\delta,p}(\Omega)$ for $s > \frac{3}{2}$, the consistency error satisfies*

$$\begin{aligned} \sup_{v_p \in V_p} \frac{\tilde{a}_\delta(u-u_p,v_p)}{\|v_p\|_\delta} &= \sup_{v_p \in V_p} \inf_{w_p \in \mathcal{M}_p} \frac{b(v_p, \partial_n u - w_p)}{\|v_p\|_\delta} \\ &\lesssim p^{-(s-1)} \|u\|_{\tilde{H}^s_{\delta,p}(\Omega)} \end{aligned} \tag{25}$$

[12] The triangle inequality applied to $\|u-u_p\|_\delta$ and the fact that

$$\|u_p - v_p\|_\delta^2 \lesssim \tilde{a}_\delta(u_p-v_p,u_p-v_p) \lesssim \underbrace{\tilde{a}_\delta(u-v_p,u_p-v_p)}_{\lesssim \|u-v_p\|_\delta \|u_p-v_p\|_\delta} + \tilde{a}_\delta(u_p-u,u_p-v_p)$$

for any $v_p \in V_p$.

Proof The first inequality follows from $\tilde{a}_\delta(u, v_p) = (f, v_p) + b(v_p, \partial_n u)$ due to $u \in H^s_{\delta,p}(\Omega)$, (23) and the characteristic property (20) of $v_p \in V_p$ that $b(v_p, w_p) = 0$ for all $w_p \in \mathcal{M}_p$. The second inequality is due to

$$\begin{aligned} b(v_p, \partial_n u - w_p) &\lesssim \|[v_p]\|_{H^{\frac{1}{2}}(\Gamma)} \|\partial_n u - w_p\|_{H^{-\frac{1}{2}}(\Gamma)} \\ &\lesssim p^{-(s-1)} \|v_p\|_\delta \|\partial_n u\|_{H^{s-\frac{3}{2}}(\Gamma)} \\ &\lesssim p^{-(s-1)} \|v_p\|_\delta \|u\|_{H^s(\mathcal{D})}. \end{aligned}$$

Here, we used triangle inequality and the trace theorem for the first factor. For the second factor, we applied the best-approximation property of the mortar test space $\mathcal{M}_p$ (Lemma 3.2) and again the trace theorem.

□

The best-approximation error is characterized in the following Lemma.

Lemma 3.9. *If $u \in \tilde{H}^s_p(\Omega)$ for $s > \frac{3}{2}$, then, provided conjecture 3.4 is satisfied, there exists a constant such that*

$$\inf_{v_p \in V_p} \|u - v_p\|_\delta \lesssim p^{-(s-1)} \|u\|_{\tilde{H}^s_p(\Omega)}$$

Proof We use the standard mortar proof technique applied to spectral methods [24]. The idea is to construct optimal spectral approximations on both the finite element domain $\mathcal{D}$ and the spherical domain $\mathcal{C}$ yielding a function in the discontinuous space $S^p(\mathcal{D}, \mathcal{T}) \oplus \mathcal{S}_p(\mathcal{C})$. Subsequently, the mortaring condition (20) is enforced subtracting the mortar-transmitted jump across the interface.

On the sphere $\mathcal{C}$, we use Lemma 3.7 so that we may measure the error in the $H^1([0, R] \times S^2)$-norm. According to Lemma 3.5, we pick a best-approximation $u_1 \in \mathcal{S}_p(\mathcal{C})$. On the intermediate domain $\mathcal{D}$, we choose a best-approximation $u_2 \in S^p(\mathcal{D}, \mathcal{T})$ according to Lemma 3.1. The function $\tilde{u} = (u_1, u_2) \in \mathcal{S}_p(\mathcal{C}) \oplus S^p(\mathcal{D}, \mathcal{T})$ then is a discontinuous domain-wise best-approximation of u.

To enforce weak continuity (20), conjecture 3.4 is used together with Lemma 3.3 to guarantee stability of the mortar projector $\Pi_p : H^{\frac{1}{2}}(\Gamma) \to S^p(\Gamma, \mathcal{T}\big|_\Gamma)$ and the polynomial lifting operator $L_p : S^p(\Gamma, \mathcal{T}\big|_\Gamma) \to S^p(\mathcal{D}, \mathcal{T})$. Thus, we can subtract the trace jump $u_1\big|_\mathcal{C} - u_2\big|_\mathcal{D}$ subject to a mortar projection and subsequent lifting on the finite element side from $\tilde{u}$ to enforce the mortaring condition. We bound this term by

$$\begin{aligned} \|L_p \Pi_p [\tilde{u}]\|_{H^1(D)} &\lesssim \|\Pi_p [\tilde{u}]\|_{H^{\frac{1}{2}}(\Gamma)} \\ &\lesssim \|[\tilde{u}]\|_{H^{\frac{1}{2}}(\Gamma)} \\ &\lesssim \|u - \tilde{u}\big|_{\partial C}\|_{H^{\frac{1}{2}}(\Gamma)} + \|u - \tilde{u}\big|_{\partial D}\|_{H^{\frac{1}{2}}(\Gamma)} \\ &\lesssim \|u - \tilde{u}\|_\delta, \end{aligned}$$

where we used the triangle inequality and the trace theorem in the last two steps. The assertion then follows from the estimates in the first part of the proof.

□

Lemma 3.9 completes the error analysis in the Lax-Milgram setting. We summarize the results of this section in the following theorem.

Theorem 3.10. *Let $u \in \tilde{H}^s_{\delta,p}(\Omega)$, $s > \frac{3}{2}$, and $u_p \in V_p$ be the solutions to the boundary value problems (23). Then*

$$\|u - u_p\|_\delta \lesssim p^{-(s-1)} \|u\|_{\tilde{H}^s_p(\Omega)}.$$

Remark. The spherical harmonics spaces

$$\mathcal{M}_p = \bigoplus_{i=1}^{M} \mathcal{S}_p\left(\mathcal{C}_i\right)\big|_{\partial\mathcal{C}_i}$$

is also a feasible choice. However, a continuous trace-lifting operator must be proven to exist separately for the spherical space. In addition, the argument presented in [8] does not apply in this setting, since finite elements are not globally smooth on S^2, but only H^1-regular in general. □

3.5.2 A priori error estimate for Schrödinger problems

Following the argument in [8], the results of the previous section can be applied to derive an a priori error estimate for Schrödinger eigenvalue problems using Babuska-Osborn theory [25]. Precisely, a duality argument together with Theorem 3.10 allows to conclude the convergence of the discrete Lax-Milgram solution operator (second variational problem in (23)) to that of the continuous one (first problem in (23)) as an operator in $\mathcal{L}(L^2(\Omega), L^2(\Omega))$. This establishes convergence of eigenvalues and recasts the eigenspace approximation error, which is measured in terms of

$$d(E, F) = \max\{\delta(E, F), \delta(F, E)\},$$

where

$$\delta(E, F) := \sup_{x \in E,\ \|x\|_{L^2(\Omega)}=1} \inf_{y \in F,\ \|y\|_{L^2(\Omega)}=1} \|x - y\|_\delta$$

for closed subspaces $E, F \subset L^2(\Omega)$, in the Lax-Milgram setting. The regularity of the eigenfunctions of a_δ described in section 2.3.2 is then exploited so that Theorem 3.10 can again be applied. Since the consistency error (Lemma 3.8) is of the same order as the best-approximation error (Lemma 3.9), the eigenvalues converge twice as fast as the eigenspaces. We omit further details of this proof and instead directly state the theorem.

Theorem 3.11. *Let λ be an eigenvalue of (14) with eigenspace E^λ, $\dim E^\lambda = m$. If the potentials satisfy the regularity requirements in section 2.3.2 and conjecture 3.4 holds, for sufficiently large p there exists eigenvalues $(\lambda_{k,p})_{k=1}^m$ of the discrete problem with eigenspaces $(E_p^{\lambda_{k,p}})_{k=1}^m$ such that*

$$d\left(E^\lambda, \bigoplus_{k=1}^{m} E_p^{\lambda_{k,p}}\right) \lesssim p^{-(s-1)}$$

and

$$|\lambda - \lambda_{k,p}| \lesssim p^{-2(s-1)}.$$

4 Algorithmic aspects

In this chapter, we describe the basic algorithmic tasks involved in the solution of the discrete linearized Kohn Sham problem (15). First of all, this includes the transfer to a generalized matrix eigenvalue problem and its solution. We will then specify important algorithmic concepts for the construction of the finite element space (17) forming the basis of our discussion of the spherical discretization space. In the remainder, we will derive a generic algorithm to construct a basis for the mortar element space in an abstract setting that directly applies to our concrete discretization proposed in chapter 3.

4.1 General remarks on Galerkin discretization

Computationally, the discrete linearized Kohn-Sham problem (15) is reformulated as an eigenvalue problem in $\mathbb{R}^n$. Introducing a basis $\{b_i\}_{1\leq i\leq n}$ of V_n, we can expand both sides of (15) into matrix-vector products. The discretized Hamiltonian at ρ is given by

$$H_{\rho_{ji}} := a_\rho(b_i, b_j) = \int_\Omega b_j H_\rho b_i dx \tag{26}$$

and the overlap matrix

$$S_{ji} = (b_i, b_j)_{L^2} = \int_\Omega b_i b_j dx. \tag{27}$$

Problem (15) then is equivalent to the generalized matrix eigenvalue problem

$$\boxed{\begin{array}{c}\text{Find the eigenpairs } (\varepsilon_L, \mathbf{C}_L)_{L=1}^N \in \mathbb{R}\times\mathbb{R}^n \\ \text{obeying the Aufbau principle s.t. } H_\rho \mathbf{C}_L = \varepsilon_L S \mathbf{C}_L\end{array}} \tag{28}$$

on $\mathbb{R}^n$, where $\mathbf{C}_L = (C_{iL})_{1\leq i\leq n}$ is the vector of coefficients of the orbital ϕ_L relative to the basis $\{b_i\}_{1\leq i\leq n}$.

It is important to note that the transfer from the discrete (15) to the matrix eigenvalue problem (28) is in general not exact since the integrals in (26), (27) can often only be approximated by numerical quadrature. We will, however, ignore these effects in this work.

There are two important general algorithmic remarks to be made in the context of a general discretization of the linearized Kohn-Sham problem. First, we explain an efficient approach to the computation of the Hartree potential via a Poisson problem. Secondly, we describe an algorithm for the solution of the generalized eigenvalue problem posed in (28).

4.1.1 Evalutation of the Hartree potential

For the Hartree potential we need to evaluate a convolution of the electron density with the kernel of the Laplacian

$$V_H[\rho] = \rho * \frac{1}{|x|}.$$

Instead of numerically integrating this singular integrand at every point the Hartree potential is evaluated, we may instead use the fact that the Hartree potential is the solution

to a Poisson problem on the global domain $\mathbb{R}^3$ with the constraint $u(x) \to 0$ as $|x| \to \infty$. Therefore, we can approximate $V_H[\rho]$ by the solution to

$$\boxed{\text{Find } u \in H_0^1(\tilde{\Omega}) \text{ s.t. } -\Delta u = 4\pi\rho,} \tag{29}$$

on a truncated domain $\tilde{\Omega} \subset \mathbb{R}^3$. This will not affect the existence result in Theorem 2.1 for the linearized Kohn-Sham equation, since the Lax-Milgram lemma for the Poisson equation guarantees that u is bounded in terms of ρ. The error in the Hartree potential corresponds to a harmonic function on $\tilde{\Omega}$ (in the weak sense) that agrees with the exact Hartree potential on $\partial\tilde{\Omega}$. By Lax-Milgram, it can be estimated in terms of the $H^{\frac{1}{2}}$-norm of the trace of the exact Hartree potential on $\partial\tilde{\Omega}$ assuming the boundary is sufficiently regular so that we can continuously lift $V_H[\rho]\big|_{\partial\tilde{\Omega}} \in H^{\frac{1}{2}}(\partial\tilde{\Omega})$ into $H^1(\tilde{\Omega})$.

In the present work, we will use to the same domain for the Poisson problem (29) as for the linearized Kohn-Sham problem (15). For future work, it remains to be investigated whether a larger domain $\tilde{\Omega}$ for the Poisson problem enhances the results.

Solution of the linear eigenvalue problem

We describe a feasible approach to solve the generalized eigenvalue problem (28) posed by the linearized Kohn-Sham equations. First of all, we note that both H_ρ and S are both real, symmetric matrices. This is an ideal situation for transforming the problem back into a symmetric standard eigenvalue problem of the form

$$A\tilde{C} = \lambda\tilde{C},$$

where $A \in M_n(\mathbb{R})$ is symmetric, $C \in \mathbb{R}^n$. For such an eigenvalue problem one can efficiently determine the largest part (in modulus) of the spectrum. Since the Aufbau principle requires the algebraically smallest eigenvalues, we need to first shift the eigenvalues by a lower bound ν on the spectrum of a_ρ as in the proof of Theorem 2.1,

$$(H_\rho + \nu S)\mathbf{C}_L = (\varepsilon_L + \nu)S\mathbf{C}_L.$$

so that the left hand side matrix becomes positive definite. To obtain an eigenvalue problem in terms of the largest part of the spectrum, we need to move $H_\rho + \nu S$ to the other side. Precisely, a factorization into a symmetric product LL^t is required, where L should be efficiently invertible. A feasible candidate is Cholesky factorization [26]

$$(H_\rho + \nu S) = L_{H_\rho+\nu S} L^t_{H_\rho+\nu S}$$

which produces a triangular factor $L_{H_\rho+\nu S}$. Thus, we arrive at the standard eigenvalue problem

$$\begin{gathered}\text{Find the eigenpairs } (\tilde{\varepsilon}_L, \tilde{\mathbf{C}}_L)_{L=1}^N \text{ on } \mathbb{R}^n \text{ with largest eigenvalues } \tilde{\varepsilon}_L \text{ s.t.} \\ L^{-1}_{H_\rho+\nu S} S L^{-t}_{H_\rho+\nu S} \tilde{\mathbf{C}}_L = \tilde{\varepsilon}_L \tilde{\mathbf{C}}_L.\end{gathered} \tag{30}$$

This problem can be solved efficiently using Lanczos iteration [27]. The solution to the original problem (28) can then be obtained by transforming back via

$$\begin{aligned}\mathbf{C}_L &= L^{-t}_{H_\rho+\nu S}\, \tilde{\mathbf{C}}_L, \\ \varepsilon_L &= \frac{1}{\tilde{\varepsilon}_L} - \nu.\end{aligned}$$

4.2 Finite elements

To lay the groundwork for our discussion of discretization spaces, we describe algorithmic aspects of the finite element method as used in the hp-FEM solver `concepts`. Especially, this includes the concepts involved in the construction of an appropriate basis and the efficient computation of the system matrices in (26) and (27). This defines a conceptual framework, in which we treat the spherical space and the mortar element coupling of FEM to spherical harmonics in the following sections.

4.2.1 Construction of a basis

An important algorithmic difficulty of the finite element method stems from the construction of an appropriate basis $\{b_i\}_{1\leq i\leq n}$ of $S^p(\Omega, \mathcal{T})$ in (17) that incorporates inter-element continuity constraints. It can be resolved adopting an interior-boundary decomposition on the element spaces $\mathcal{Q}_p(\hat{K})$ as presented in [28]. Considering the hexahedron as a topological cell complex[13] $\{c_k\}_{k\in I_K}$, one constructs a basis $\{\hat{b}_i\}_{1\leq i\leq (p+1)^3}$ of shape functions that decomposes the element space $\mathcal{Q}_p(\hat{K})$ into a set of subspaces, each associated to a single cell c_k, $k \in I_K$. The element space then decays into a sum of the hexahedral, face, edge and vertex cell spaces,

$$\mathcal{Q}_p(\hat{K}) = \bigoplus_{k\in I_K} \mathcal{Q}_p^{c_k}(\hat{K}_i),$$

The space $\mathcal{Q}_p^{c_k}(\hat{K})$ is such that when restricted to the cell c_k it either coincides with $\mathcal{P}_{p,0}(0,1)^{\otimes \dim c_k}$ with homogeneous trace if $\dim c_k > 0$ or the space of constants if c_k is a vertex. Moreover, the functions in $\mathcal{Q}_p^{c_k}(\hat{K})$ are chosen to decay linearly along all orthogonal directions in the hexahedron $\hat{K}_i$ so that they vanish on all other cells of dimension less than or equal to $\dim c_k$. As a consequence of this decomposition, to obtain globally continuous basis functions of $S^p(\Omega, \mathcal{T})$ one only has to match the spaces $\mathcal{Q}_p^{c_k}(\hat{K})$ for geometrically coincident cells $F_{K_i}(c_k) = F_{K_j}(c_l)$ for any pair of adjacent elements in the mesh. In terms of basis functions, this eventually boils down to studying the group action of affine transformations (16) on cell shape functions. In this manner, the interior-boundary decomposition produces localized basis functions with the property that for any subset $\mathcal{T}'$ of cells in $\mathcal{T}$ the non-zero restrictions of basis functions forms a basis of the restricted space $S^p(\Omega, \mathcal{T})\big|_{\mathcal{T}'}$. As a consequence, it is straight-forward to construct arbitrary trace spaces, which will turn out to be useful in the construction of the mortar element space.

For completeness, note that the choice of shape functions is not only governed by ease of implementation but also the efficiency of the solution step. Important points are also the efficient computation and the condition number of system matrices as well as the associated memory requirements. In this respect, `concepts` uses a highly optimized basis of tensorized integrated Legendre polynomials [28] that facilitates $\mathcal{O}(p^7)$ evaluation of bilinear forms on element spaces via sum-factorization and produces well-conditioned system matrices (c.f. section 6 in [30]). The assembly of system matrices is described in the next section.

[13] The hexahedron can be topologically described as a cell-complex, so that the hexahedron, every face, edge and vertex are individual cells (c.f. [29])

4.2.2 Assembly

The correspondence between global basis functions $\{b_i\}$ and the local shape functions $\{\hat{b}_j\}$ can be expressed by the discrete restriction operator[14]

$$T_K : \begin{cases} S^p(\Omega, \mathcal{T}) & \to & F_{K*}\left(\mathcal{Q}_p(\hat{K})\right) \\ b_i & \mapsto & b_i\big|_K = \sum_j T_{K,ji}\, \hat{b}_j \circ F_K^{-1} \end{cases}$$

The associated matrix relative to these bases is called the T-matrix. It facilitates the assembly, that is the computation of system matrices (26) and (27) from element-wise contributions. For a local xc-potential in H_ρ (e.g. LDA), we can evaluate the bilinear form for a_ρ iteratively over all elements

$$H_{\rho,ji} = \sum_{K\in\mathcal{T}} a_\rho\big|_K(b_i, b_j) \quad = \sum_{K\in\mathcal{T}} \sum_{k,l=1}^{(p+1)^3} T_{K,ki}\, T_{K,lj}\, a_\rho\big|_K(\hat{b}_k \circ F_K^{-1}, \hat{b}_l \circ F_K^{-1}) \tag{31}$$

and analogously the L^2-inner products for the overlap matrix S.

4.3 Spherical function space

We construct a set of shape functions as well as a global basis for the subspace of continuous functions in $\mathcal{S}_L(\bigcup_i \mathcal{C}_i)$. Furthermore, we explain an optimized evaluation of bilinear forms on element spaces analogous to sum-factorization for tensorial finite elements.

In analogy to the FEM setting, the construction of a global basis is based on shape functions defined on a reference element. For this purpose, we use spherical coordinates,

$$\begin{array}{rcl} [0,1]\times[0,\pi]\times[0,2\pi] & \to & B_{R_i}(X_i) \\ (r,\theta,\phi) & \mapsto & \begin{pmatrix} R_i\, r\sin\theta\cos\phi \\ R_i\, r\sin\theta\sin\phi \\ R_i\, r\cos\theta \end{pmatrix} + X_i \end{array} \tag{32}$$

on $\mathcal{C}_i = B_{R_i}(X_i)$. The element space is of the form

$$\mathcal{S}_L(B_1(0)) = \mathcal{P}_L(0,1) \otimes \left(\bigoplus_{l=0}^{L} E_l\right). \tag{33}$$

The spherical harmonic space E_l of angular momentum l is spanned by $(2l+1)$ basis functions indexed by pairs (l,m), $-l \le m \le l$, that are usually chosen as

$$Y_{lm}(\theta,\phi) = \tilde{P}_{lm}(\cos\theta)\frac{1}{\sqrt{2\pi}} e^{im\phi}.$$

[14] We denote the push-forward of $f \in L^2(\hat{K})$ along F_K by

$$F_{K*}(f) := f \circ F_K^{-1}.$$

The $\tilde{P}_{lm}$ are the associated Legendre functions characterized by the Rodrigues formula

$$\tilde{P}_{lm}(z) = (1-z^2)^{\frac{m}{2}} \frac{d^m}{dz^m} P_l(z), \tag{34}$$

in terms of Legendre polynomials P_l, which indicates their non-polynomial structure for m odd. Instead of (34), for numerical purposes, it is more appropriate to work with normalized basis functions. Thus, we will denote by P_{lm} the $L^2(-1,1)$-normalized associated Legendre functions in the following. Since we restrict to real-valued function spaces, we perform a change of basis in the ϕ-modes and define

$$Y_{lm+}(\theta,\phi) := P_{lm}(\cos\theta)\frac{1}{\sqrt{\pi}}\cos(m\phi) \tag{35}$$

$$Y_{lm-}(\theta,\phi) := P_{lm}(\cos\theta)\frac{1}{\sqrt{\pi}}\sin(m\phi) \tag{36}$$

for $m > 0$ and in case $m = 0$

$$Y_{l0+}(\theta,\phi) := P_{l0}(\cos\theta)\frac{1}{\sqrt{2\pi}}, \tag{37}$$

where $l \geq m \geq 0$.[15] The concrete shape functions for the element space are constructed by a tensor product of radial polynomials with spherical harmonics,

$$\hat{b}_{(l,m,s)_{\text{lin}}+(L+1)^2 p} = q_p \otimes Y_{lms}, \tag{38}$$

for all $0 \leq p, l \leq L$, and m, s correspondingly. The function $q_p \in \mathcal{P}_L(0,1)$ here denotes the p-th integrated Legendre polynomial. We use a linear enumeration convention for spherical harmonics given by

$$(l,m,s)_{\text{lin}} = \begin{cases} l & ,m = 0 \\ (L+1) + 2\sum_{\tilde{m}=1}^{m-1}(L+1-\tilde{m}) + 2(l-m) + s & ,m > 0 \end{cases} \tag{39}$$

identifying $+ \equiv 0$ and $- \equiv 1$ for s.

Due to the absence of inter-element matching conditions compared to the finite element space, the assembly of shape functions into a global basis only needs to consider the continuity restrictions imposed by the space $\mathcal{S}_L(\mathcal{C}_i) \cap C^0(\mathcal{C}_i)$ on the i-th element. Hence, the $(L+1)^2 - 1$ modes that vary on the interior boundary of the stretched cone need to be discarded during the construction of the space. For the chosen shape functions, these are exactly

$$q_0(r) \otimes Y_{lms}(\theta,\phi) = (1-r)Y_{lms}(\theta,\phi), \quad l > 0,$$

so that we are left with a single global degree of freedom, $q_0 \otimes Y_{00+}$, that does not vanish at the center of the sphere. All other integrated Legendre polynomials for $p > 0$ vanish at $r = 0$ and thus all the corresponding modes are directly integrated into the global basis of

$$\bigoplus_i S_L(\mathcal{C}_i) \cap C^0(\mathcal{C}_i).$$

[15] Note that this basis does not involve a Condon-Shortley phase for the associated Legendre polynomial often used by the physics community.

To obtain the matrix eigenvalue problem (28) for the discrete linearized Kohn-Sham problem on $\mathcal{S}_L(\bigcup_i \mathcal{C}_i)$, we need to evaluate the integrals in (26) and (27) restricted to the spheres $\mathcal{C}_i$ numerically. In general, this will involve quadrature and, thus the appropriate rules must be determined. We distinguish between the integration required by the kinetic energy operator T_1, which corresponds to an inner product of gradients, and integrals for the potentials, which are multiplication operators[16], where the integrand is a product of usual functions.

For the gradient bilinear form, we have on the sphere $\mathcal{C} = B_R(X)$,

$$\begin{aligned}
&\int_{B_R(X)} \nabla(q_p \otimes Y_{lms}) \cdot \nabla(q_{p'} \otimes Y_{l'm's'}) dx \\
&\qquad = \int_0^R \partial_r q_p \partial_r q_{p'} r^2 dr \int_{S^2} Y_{lms} Y_{l'm's'} dS \\
&\qquad + \int_0^R q_p q_{p'} dr \underbrace{\int_{S^2} \nabla_{S^2} Y_{lms} \cdot \nabla_{S^2} Y_{l'm's'} dS}_{= l(l+1) \int_{S^2} Y_{lms} Y_{l'm's'} dS} \\
&\qquad = \delta_{l,l'} \delta_{m,m'} \delta_{s,s'} \left(\int_0^R \partial_r q_p \partial_r q_{p'} r^2 dr + l(l+1) \int_0^R q_p q_{p'} dr \right) \qquad (40)
\end{aligned}$$

due to Stokes' Theorem and orthogonality of spherical harmonics. This integral can, thus, be exactly evaluated using Gaussian quadrature with $L+1$ nodes. As an inherent property of the spectral basis, the effort for calculating these integrals is small at $\mathcal{O}(L^3)$ floating point operations[17].

For multiplication operators, the evaluation of element integrals is more involved. For some general potential $V \in C^0(\mathcal{C})$ the expression to be evaluated is

$$\int_{B_R} V \; q_p \otimes Y_{lms} \; q_{p'} \otimes Y_{l'm's'} \; dx = \int_{S^2} Y_{lms} \; Y_{l'm's'} \underbrace{\int_0^R V q_p q_{p'} r^2 dr}_{=: V_{pp'}} \; dS, \qquad (41)$$

where we integrate first along the radial direction. In the next step, we perform the ϕ-

[16] The integrals for the overlap matrix (27) also falls into this category.

[17] Precisely, we need to evaluate $(L+1)^2$ radial integrals using $L+1$ Gaussian nodes.

integration on the functions $\{V_{pp'}\}_{0\le p,p'\le L}$ via Fourier coefficients[18]

$$V_{pp'mm'ss'} := \begin{cases} \Re(\widehat{V_{pp'}}_{m+m'}) + \Re(\widehat{V_{pp'}}_{m-m'}) & , m,m' > 0, s = s' = + \\ -\Re(\widehat{V_{pp'}}_{m+m'}) + \Re(\widehat{V_{pp'}}_{m-m'}) & , m,m' > 0, s = s' = - \\ \Im(\widehat{V_{pp'}}_{m+m'}) - \Im(\widehat{V_{pp'}}_{m-m'}) & , m,m' > 0, s = +, s' = - \\ \Re(\widehat{V_{pp'}}_{m}) + \Re(\widehat{V_{pp'}}_{-m}) & , m > 0, s = +, m' = 0 \\ -\Im(\widehat{V_{pp'}}_{m}) - \Im(\widehat{V_{pp'}}_{-m}) & , m' > 0, s = -, m' = 0 \\ \widehat{V_{pp'}}_{0} & , m = 0, m' = 0 \end{cases} \tag{42}$$

where we used the real-valuedness of the $V_{pp'}$ and real linearity of the real and imaginary part, $\Re$ and $\Im$, respectively. We are then left with the z-integrals

$$\int_{B_R} V\ q_p \otimes Y_{lms}\ q_{p'} \otimes Y_{l'm's'}\ dx = \int_{-1}^{1} P_{lm}(z)\ P_{l'm'}(z) V_{pp'mm'ss'}(z)\ dz. \tag{43}$$

For general V, none of these integrals can be evaluated exactly and thus appropriate quadrature rules must be chosen. For the radial and the z-integration Gauss quadrature is appropriate with a minimum of $(L+2)$ and $(L+1)$ nodes, respectively. This will be exact in case V is constant. For the ϕ-direction a trapezoidal rule with at least $(2L+1)$ nodes is suitable to exactly integrate trigonometric polynomials. For more general V, we have the following Lemma.

Lemma 4.1. *If $V \in \mathcal{S}_{L'}(\mathcal{C})$, the integral (41) can be evaluated exactly using $(L+2+\lceil \frac{L'}{2} \rceil)$ Gaussian nodes in r-, $(2L+1+L')$ trapezoidal nodes in ϕ- and $(L+1+\lceil \frac{L'}{2} \rceil)$ Gaussian nodes in z-direction.*

Proof The proof of this Lemma is elementary up to z-integrals of the form

$$\int_{-1}^{1} P_{lm}\ P_{l'm'}\ P_{l''m''}\ dz,$$

where it is not a priori clear that the integrand is a polynomial. However, this integral occurs only if

$$m \pm m' \pm m'' = 0,$$

since the ϕ-integration eliminates all other terms. Moving any variable to the right-hand side, we see that this can only hold if either two or none of the elements in $\{m, m', m''\}$

[18] E.g. for $m, m' > 0$, $s = s' = +$ using

$$\cos(m\phi)\cos(m'\phi) = \frac{1}{2}\left(\cos((m+m')\phi) + \cos((m-m')\phi)\right),$$

we obtain:

$$\begin{aligned} \frac{1}{\pi}\int_0^{2\pi} V_{pp'}\cos(m\phi)\cos(m'\phi)d\phi &= \frac{1}{2\pi}\int_0^{2\pi} V_{pp'}\Re(e^{i(m+m')\phi})d\phi + \frac{1}{2\pi}\int_0^{2\pi} V_{pp'}\Re(e^{i(m-m')\phi})d\phi \\ &= \Re(\widehat{V_{pp'}}_{m+m'}) + \Re(\widehat{V_{pp'}}_{m-m'}) \end{aligned}$$

are odd. But in this case, by the Rodrigues formula (34) the above integrand will again be a polynomial and thus Gauss quadrature exact. □

An interesting fact is that the observation in the proof and thus the statement of Lemma 4.1 can be generalized to the case, where V is a product of arbitrary many functions in the spherical space $\mathcal{S}_{L'}(\mathcal{C})$, which can also be seen from representation theory. Especially, this has interesting consequences for the approximation of two-particle expectation values in DFT. Lemma 4.1 also marks the starting point for an a priori error analysis of the mortar element discretization (21) that includes quadrature errors, which we, however, do not pursue in this thesis.

Finally, we analyze the algorithmic complexity of the evaluation of (41) on all local shape functions. For the radial integral (41) we have two fixed quadrature nodes and two radial degrees of freedom so that we need to integrate $\mathcal{O}(L^4)$ functions over $L+2$ nodes each. For the ϕ-transformation, we can resort to FFT, which computes the Fourier coefficients of the $(L+1)^3$ functions in $\mathcal{O}(L \log L)$ operations each [31]. Finally, the associated Legendre integral (43), being the most expensive operation, requires $\mathcal{O}(L^6)$ functions[19] to be integrated over $L+1$ nodes each. In total, we have an $\mathcal{O}(L^7)$ algorithm for the evaluation of the multiplication operator bilinear form (41), which is identical to the computational complexity on finite elements using sum-factorization.

In addition to bilinear forms restricted to elements, we must also declare a linear form corresponding to the inner product of the density in the Poisson problem for the Hartree potential (29) with test functions. The optimizations to perform there are entirely analogous to the multiplication bilinear form. As a result, one obtains an $\mathcal{O}(L^4)$ algorithm, the involved spherical harmonic transform costing $\mathcal{O}(L^3)$ operations.

In terms of optimizations, there are several further options not yet realized in this framework. The most obvious is to use the symmetry of the P_{lm}. However, this comes at the cost of giving up the flexibility in choosing the quadrature rule. To be consistent with `concepts`[20], we chose to avoid this optimization. Furthermore, the spherical harmonic quadrature grid can be thinned towards the poles due to exponential decay of the associated Legendre functions. This can be seen by a quasi-classical WKB-approximation [32]. Further approximations lead to several asymptotically faster algorithms for spherical harmonic transforms with a complexity of e.g. $\mathcal{O}(L^{\frac{5}{2}} \log L)$ or $O(L^2 \log^2 L)$ in [32]. However, from a practical perspective, these algorithms are vastly ouperformed by an explicitly vectorized implementation of the same $\mathcal{O}(L^3)$-algorithm we used above in [33]. For the purpose of this thesis, we will restrict to the transforms presented above and leave optimizations to the future.

4.4 The mortar element space

The resolution of the mortaring condition (20) for the coupling of two discretization schemes is derived in an abstract framework. As a consequence, we obtain a generic algorithm to construct a basis for the mortar element space. In chapter 6, we apply this algorithm to construct a generic mortar element space for Kohn-Sham DFT coupling finite elements to spherical harmonics using either trace space as a mortar test space.

[19] That is $(L+1)^2$ spherical harmonic and $L+1$ radial configurations in both factors.

[20] Note that `concepts` does not exploit the symmetry in the p-FEM shape functions

4.4.1 Construction of a basis

Let the domain $\Omega \subset \mathbb{R}^n$, $n = 2, 3$, be decomposed into Ω_1, Ω_2 with a smooth interface Γ. Let the discretization on Ω_i be denoted by V_i with trace space V_i^Γ on Γ for both $i = 1, 2$. Suppose that the basis functions on both V_i perform an interior-boundary decomposition in the sense that $(b_j^i)_{1 \le j \le \dim V_i^\Gamma}$ is a basis of V_i^Γ and $(b_j^i)_{\dim V_i^\Gamma < j \le \dim V_i}$ satisfy homogeneous Dirichlet boundary conditions on Γ[21]. Furthermore, let $\mathcal{M}_\Gamma \subset H^{-\frac{1}{2}}(\Gamma)$ be the mortar test space of $\dim \mathcal{M}_\Gamma = \dim V_2^\Gamma$ spanning the dual space of V_2^Γ by means of the isomorphism

$$\begin{array}{ccc} \mathcal{M}_\Gamma & \to & V_2^{\Gamma\prime} \\ w & \mapsto & (\cdot, w)_{L^2(\Gamma)} \end{array} \tag{44}$$

and $\{w_k\}_{1 \le k \le \dim \mathcal{M}_\Gamma}$ a basis of $\mathcal{M}_\Gamma$. In analogy to (20), we define the mortaring condition by

$$b(u, w) = \int_\Gamma [u]\, w \, dS = 0, \quad \forall w \in \mathcal{M}_\Gamma \tag{45}$$

for $u \in V_1 \oplus V_2$. The basis functions $(b_j^1)_{1 \le j \le \dim V_1^\Gamma}$ are then called the mortars, while the basis functions $(b_j^2)_{1 \le j \le \dim V_2^\Gamma}$ are the non-mortars.

The mortaring condition (45) enforces the continuation of mortars by non-mortars. In particular, for $u = (\sum_{i=1}^{\dim V_j} c_i^j b_i^j)_{j=1,2} \in V_1 \oplus V_2$, the mortaring condition is equivalent to

$$\sum_{i=1}^{\dim V_1^\Gamma} (b_i^1, w_k)_{L^2(\Gamma)} c_i^1 = \sum_{i=1}^{\dim V_2^\Gamma} (b_i^2, w_k)_{L^2(\Gamma)} c_i^2 \quad \text{for all } 1 \le k \le \dim \mathcal{M}_\Gamma,$$

which we can restate as

$$S_\Gamma^{2\,-t} S_\Gamma^{1\,t} \mathbf{C}_\Gamma^1 = \mathbf{C}_\Gamma^2, \tag{46}$$

for the overlap matrices defined by

$$\left(S_\Gamma^i\right)_{kl} := \int_\Gamma b_k^i w_l \, dS, \quad 1 \le k \le \dim V_i^\Gamma,\ 1 \le l \le \dim \mathcal{M}_\Gamma. \tag{47}$$

By duality of $\mathcal{M}_\Gamma$ to V_2^Γ (44) the overlap matrix S_Γ^2 is regular and its inverse, thus, well-defined. If the kernel of S_Γ^2 was non-zero, then there would exist $0 \neq u_2 \in V_2^\Gamma$ such that

$$(u_2, w) = 0 \quad \forall w \in \mathcal{M}_\Gamma.$$

But in this case, the space $\mathcal{M}_\Gamma$ does not span the dual of V_2^Γ, since $(\cdot, u_2)_{L^2(\Gamma)} \in V_2^{\Gamma\prime}$ does not vanish on u_2. This is in contradiction to the isomorphism (44). Hence, equation (46) tells us that the non-mortar coefficients of a function $u \in V_b$,

$$V_b = \{u \in V_1 \oplus V_2 |\ b(u, w) = 0\ \forall w \in \mathcal{M}_\Gamma\}$$

[21] Such a property can be satisfied by any discretization. If a given basis does not satisfy this property, we can always construct a new basis that incorporates it using Gram-Schmidt orthogonalization with respect to the inner product of $L^2(\Gamma)$.

are uniquely determined by the mortar coefficients. We, therefore, call the product $S_\Gamma^{2\,-t} S_\Gamma^{1\,t}$ the mortar transmission matrix. It is the non-zero block of the linear mortar transmission operator

$$S : \begin{array}{ccc} V_1 & \to & V_2 \\ b_j^1 & \mapsto & \begin{cases} \sum_{k=1}^{\dim V_2^\Gamma} \left(S_\Gamma^{2\,-t} S_\Gamma^{1\,t} \right)_{kj} b_k^2 & , \quad 1 \leq j \leq \dim V_1^\Gamma \\ 0 & , \quad \dim V_1^\Gamma < j \leq \dim V_1 \end{cases} \end{array} \tag{48}$$

The following Lemma then characterizes the mortar space.

Lemma 4.2. *The familiy of interface functions*

$$\left\{ b_j^1, S(b_j^1) \right\}_{1 \leq j \leq \dim V_1^\Gamma} \tag{49}$$

grouped with the interior basis functions $\{b_j^1\}_{\dim V_1^\Gamma < j \leq \dim V_1}$ *and* $\{b_j^2\}_{\dim V_2^\Gamma < j \leq \dim V_2}$ *forms a basis of the mortar space* V_b*.*

Proof Linear independence is clear, since for the interface basis functions it holds on the mortar side. Moreover, the constraint (45) can be formulated as a map

$$f : \begin{array}{ccc} V_1 \oplus V_2 & \to & \mathbb{R}^{\dim \mathcal{M}_\Gamma} \\ u = (u_1, u_2) & \mapsto & \begin{pmatrix} b(u, w_1) \\ \vdots \\ b(u, w_{\dim \mathcal{M}_\Gamma}) \end{pmatrix} \end{array} ,$$

for some basis $\{w_i\}_{1 \leq i \leq \dim \mathcal{M}_\Gamma}$ of $\mathcal{M}_\Gamma$ so that its kernel coincides with the mortar space. This map is onto due to the isomorphism (44). Therefore, the mortar space V_b has dimension

$$\dim V_b = \dim V_1 + \dim V_2 - \dim \mathcal{M}_\Gamma,$$

which due to $\dim \mathcal{M}_\Gamma = \dim V_2^\Gamma$ is exactly the size of our linearly independent family. □

Remark. The statement of Lemma 4.2 can be generalized to the mortar coupling of multiple discretization schemes across pairwise interfaces. This is e.g. the situation in DFT, where the finite element spherical harmonic mortar element coupling can be defined differently on every atomic sphere $\mathcal{C}_i$. This provides extra variability, that we will, however, not be concerned with in this work. □

Remark. There is a particular subtlety about mortar element methods associated with the structure of the transmission operator (48). In general, the non-zero block $S_\Gamma^{2\,-t} S_\Gamma^{1\,t}$ will be dense if we use the trace space V_2^Γ as the test space $\mathcal{M}_\Gamma$. An alternative way to see this is to consider the entry $(S_\Gamma^{2\,-t} S_\Gamma^{1\,t})_{kj}$ as the $L^2(\Gamma)$-inner product of b_j^1 with the k-th element of the bi-orthogonal basis $\{\tilde{b}_k^2\}_{1 \leq k \leq \dim V_2^\Gamma}$ of V_2^Γ that is characterized by

$$\left(b_r^2, \tilde{b}_s^2 \right) = \delta_{rs}, \quad \forall 1 \leq r, s \leq \dim V_2^\Gamma .$$

This basis usually has global support on the interface Γ and, therefore, enforces global continuations of mortars on the non-mortar side. This is especially disadvantageous if

V_1 and V_2 are both discretizations with localized basis functions such as finite elements. The literature is well aware of this phenomenon [34] and there are several techniques to choose the mortar test space differently in order to keep the support of the continuations more localized. An example is the dual basis method [34], where a subspace of $H^{-\frac{1}{2}}(\Gamma)$ is constructed spanned by localized discontinuous basis functions that are $L^2(\Gamma)$-bi-orthogonal to the non-mortars $\{b_j^2\}_{1\leq j\leq \dim V_2^\Gamma}$. While this provides a substantial optimization for couplings of two finite element schemes, it is unlikely to result in a significant benefit in the coupling of finite elements to spherical harmonics. □

4.4.2 Assembly

We embed the mortar space V_b in the assembly framework presented in section 4.2.2. For this purpose, we denote the basis of the mortar space V_b constructed defined in Lemma 4.2 by $(b_i^m)_i$. We define the domain-wise projectors

$$T_{\Omega_j} : \left\{ \begin{array}{ccc} V_b & \to & V_j \\ b_i^m & \mapsto & b_i^m\big|_{\Omega_j}. \end{array} \right. \tag{50}$$

The matrix entry $(T_{\Omega_1})_{ij}$ is equal to one if and only if $b_j^m\big|_{\Omega_1} = b_i^1$. The same holds for T_{Ω_2} for interior degrees of freedom on Ω_2. For an interface degree of freedom b_i^m (49) that restricts to b_j^1 on Ω_1 we have

$$T_{\Omega_2 ki} = \left({S_\Gamma^2}^{-t} {S_\Gamma^1}^t \right)_{kj}, \quad 1 \leq k \leq \dim V_2^\Gamma.$$

If the discretization V_j consist of a mesh $\mathcal{T}$ of elements with T-matrices $\{T_K\}_{K\in\mathcal{T}}$, then these concatenate naturally with the domain projectors T_{Ω_1}, T_{Ω_2} (50) to a restriction operator from the mortar space V_b to the element space V_K,

$$T_K \circ T_{\Omega_j} : V_b \to V_K \tag{51}$$

so that the original elements with the modified T-matrices can directly be embedded in the mortar space V_b.

4.4.3 Evaluation of the coupling bilinear form

In the computation of the transmission matrix (48) numerical subtleties arise in case of a geometric non-conformity of the adjacent discretizations on the interface Γ. The situation is especially intricate if the interface cannot be exactly resolved from both sides. For the coupling of finite elements to spherical harmonics this is not the case, fortunately. Still, the integrals between the mortars and the mortar test functions in the computation of the overlap matrix S_Γ^1 (47) can only be approximated by quadrature. The error will directly impact the precision of the interface basis functions (49) of the mortar element space. Those integrals should, therefore, be evaluated at high precision.

In principle, we can choose between quadrature from finite elements or spherical harmonics to approximate these integrals. A priori, these may seem equivalent, however, their numerical performance can be anticipated to differ markedly. The reason for this is lies in the approximation properties of the trace spaces of $\mathcal{S}_L(\bigcup_i \mathcal{C}_i)$ and $S^p(\mathcal{D},\mathcal{T})$. In

particular, the idea is to trace the quadrature error back to the best-approximation error on the reference elements of the associated discrete space using a stability argument for the Lagrange interpolation operator (c.f. [24]). Since spherical harmonics are smooth on the spherical quadrilaterals, they can be approximated by finite elements with an error that converges exponentially under p-refinement. In contrast, finite elements are not globally smooth on the sphere, but generally only lie in $H^1(S^2)$. Therefore, we cannot bound the error for a spherical harmonics quadrature by high-order Sobolev-norms. Thus, we expect only slow algebraic convergence for spherical harmonics quadrature. Therefore, finite element quadrature is to be preferred to compute the spherical harmonic transforms required at the construction of our mortar element space (21).

5 The optimal damping algorithm

We discuss the optimal damping algorithm developed in [10] for the solution of the extended Kohn-Sham problem. In contrast to classic self-consistent field iteration algorithms, such as Roothaan's algorithm [35], the optimal damping algorithm relaxes the integer occupation number constraint by a convexification of the state space. As a result, it possesses better robustness properties and can be proven to always converge [10] to a local minimum of the extended Kohn-Sham functional. In the following, we will first introduce the generalization of the standard Kohn-Sham model to mixed states. Then, the algorithm will be presented in a framework that will be useful for implementation.

5.1 The extended Kohn-Sham problem

The N-body Schrödinger problem (1) can be reformulated for statistical ensembles of pure states[22]. Such ensembles are represented by density operators. For short, density operators are self-adjoint, positive operators with unit trace and finite energy on the N-particle Hilbert space $\mathcal{H}^{(N)}$,

$$\mathcal{D}^{(N)} = \left\{\Theta \in \mathcal{L}(\mathcal{H}^{(N)}, \mathcal{H}^{(N)})|\ \Theta^* = \Theta,\ \Theta \geq 0,\ \mathrm{tr}(\Theta) = 1, \mathrm{tr}(-\Delta\Theta) < \infty\right\}.$$

They represent a statistical mixture of pure states $\psi_i \in \mathcal{H}^{(N)}$ via the expansion

$$\Theta = \sum_i n_i \left|\psi_i\right\rangle \left\langle\psi_i\right|,$$

where $n_i \in [0,1]$, $\sum_i n_i = 1$, are the occupation numbers[23]. The expectation value of an observable O_N is defined correspondingly,

$$\left\langle O_N\right\rangle_\Theta = \sum_i n_i \left\langle O_N\right\rangle_{\psi_i} = \mathrm{tr}(O_N\Theta).$$

Kohn-Sham DFT is then analogously developed for density operators as in chapter 2 resulting in the extended Kohn-Sham model, where

$$E^{KS}[\tau] = \left\langle h\right\rangle_\tau + J[\rho_\tau] + E_{xc}[\rho_\tau], \quad \tau \in \mathcal{D}^{(1,N)} := \left\{N\sigma|\ \sigma \in \mathcal{D}^{(1)}\right\} \tag{52}$$

which is defined in terms of one-particle reduced density operators instead of Slater determinants. The only difference to the pure state model is that the minimization of the N-particle kinetic energy needs not be approximated. This is due to the fact that every density operator on the one-particle Hilbert space is mixed state N-representable, while it is generally unknown which one-particle operators correspond to a pure state in $\mathcal{H}^{(N)}$[24]. The kinetic energy for the extended Kohn-Sham model is given by the Janak functional

$$T_J[\rho] := \inf_{\tau\in\mathcal{D}^{(1,N)},\rho_\tau=\rho} \left\langle T_1\right\rangle_\tau,$$

[22] An argument leading to the same model relies on a generalization to open quantum systems

[23] Such a decomposition may always be found according to the spectral theorem.

[24] In particular, for this reason the minimization in (4) was restricted over Slater determinants that have an immediate N-particle representation.

where

$$\langle T_1 \rangle_\tau = \operatorname{tr}\left(-\frac{1}{2}\Delta\tau\right) = \frac{1}{2}\sum_{L=1}^{\infty} n_L \|\nabla\phi_L\|_{L^2}^2$$

for $\tau = \sum_{L=1}^{\infty} n_L \left|\phi_L\right\rangle \left\langle\phi_L\right| \in \mathcal{D}^{(1,N)}$. For the exchange-correlation functional the same approximations are used as for pure states (c.f. chapter 2).

5.2 The optimal damping algorithm

The idea of the optimal damping algorithm (ODA) is to find a local minimum of E^{KS} (52) by iterative steepest descent. We will focus on closed-shell systems of $2N$ electrons, where every spatial orbital is occupied by two electrons due to the spin degree of freedom. For this purpose, let us denote the set of reduced discrete density operators in terms of the discretization space V of the orbitals by

$$\mathcal{D}_V^{(1,2N)} = \{\tau \in \mathcal{L}(V \times \Sigma, V \times \Sigma) |\ \tau^* = \tau,\ \tau \geq 0,\ \operatorname{tr}(\tau) = 2N\}. \tag{53}$$

The direction of steepest descent at $\tau \in \mathcal{D}_V^{(1,2N)}$ then corresponds to finding $\sigma \in \mathcal{D}_V^{(1,2N)}$ that minimizes

$$\frac{d}{dt} E^{KS}[\tau + t(\sigma - \tau)]\big|_{t=0} = \langle H_{\rho_\tau} \rangle_\sigma - \langle H_{\rho_\tau} \rangle_\tau .$$

The second term is constant, thus, we can focus on the first one. It is exactly minimal if σ is the orthogonal projector on the space spanned by the Aufbau solution of the linearized Kohn-Sham problem (15) for H_{ρ_τ}. Once we have found this Aufbau solution, we can proceed by replacing τ with the state τ' that minimizes E^{KS} on the line segment

$$\operatorname{Seg}[\tau, \sigma] := \{\tau + t(\sigma - \tau) |\ t \in [0, 1]\}.$$

This procedure can then be repeated until the distance between τ and τ' falls below a certain threshold indicating convergence to a local minimum of E^{KS}.

In the following, we work out the details of the ODA in a setting that is suitable for implementation. We will use spatial density operators $\tau \in \mathcal{D}_V^{(1,N,s)}$ operating only on V that satisfy the same properties as in (53) except for the fact that they normalize to N. Thus, in the state $\tau = \sum_{L=1}^{\tilde{N}} n_L \left|\phi_L\right\rangle \left\langle\phi_L\right| \in \mathcal{D}_V^{(1,N,s)}$ we have the particle density

$$\rho_\tau(x) = 2\sum_{L=1}^{\tilde{N}} |\phi_L(x)|^2.$$

The Kohn-Sham energy functional then takes the form

$$E^{KS}[\tau] = E^{1e}[\tau] + E^H[\tau] + E^{xc}[\tau],$$

where the energies are

$$E^{1e}[\tau] = \langle h \rangle_\tau = 2\sum_{L=1}^{\tilde{N}} n_L \left\langle\phi_L\right| h \left|\phi_L\right\rangle, \text{ where } h = -\frac{1}{2}\Delta + V_{nuc} \tag{54}$$

$$E^H[\tau] = \frac{1}{2}\langle V_H[\rho_\tau] \rangle_\tau = \frac{1}{2}\int_\Omega V_H[\rho_\tau]\ \rho_\tau\ dx, \text{ where } V_H[\rho_\tau] = \rho_\tau * \frac{1}{|x|} \tag{55}$$

$$E^{xc}[\tau] = \int_\Omega \rho_\tau\ \mathcal{E}_{xc} \circ \rho_\tau\ dx. \tag{56}$$

In the k-th step, we will be provided with the k-th iterate $\tilde{\tau}^{(k)} \in \mathcal{D}_V^{(1,N,s)}$ with density $\tilde{\rho}^{(k)}$. Solving the linearized Kohn-Sham problem (15) at $\rho_{\tilde{\tau}^{(k)}}$, we obtain an Aufbau solution $\Phi^{(k+1)} = (\phi_1^{(k+1)}, \dots, \phi_N^{(k+1)}) \in V^N$ with eigenvalues $(\varepsilon_1^{(k+1)}, \dots, \varepsilon_N^{(k+1)})$. We denote the corresponding density operator by $\tau^{(k+1)}$. In order to perform the line search on $\mathrm{Seg}[\tilde{\tau}^{(k)}, \tau^{(k+1)}]$, we approximate E^{KS} by a cubic polynomial q. For this purpose, we use the values and derivatives of E^{KS} at the endpoints. The energies can be computed from (56), while for the derivatives we use that

$$\langle H_{\rho_\tau} \rangle_\tau = E^{1e}[\tau] + 2E^H[\tau] + \langle V_{xc}[\rho_\tau] \rangle_\tau \,, \tag{57}$$

for $\tau = \tilde{\tau}^{(k)}, \tau^{(k+1)}$, where

$$\langle V_{xc}[\rho_\tau] \rangle_\sigma = \int_\Omega V_{xc}[\rho_\tau] \rho_\sigma dx. \tag{58}$$

as well as

$$\left\langle H_{\tilde{\rho}^{(k)}} \right\rangle_{\tau^{(k+1)}} = 2 \sum_{L=1}^{N} \varepsilon_L^{(k+1)} \tag{59}$$

$$\left\langle H_{\rho^{(k+1)}} \right\rangle_{\tilde{\tau}^{(k)}} = E^{1e}[\tilde{\tau}^{(k)}] + \left\langle V_H[\rho^{(k+1)}] \right\rangle_{\tilde{\tau}^{(k)}} + \left\langle V_{xc}[\rho^{(k+1)}] \right\rangle_{\tilde{\tau}^{(k)}} . \tag{60}$$

One then constructs q and determines the optimal damping parameter

$$\lambda^{(k+1)} = \mathrm{arginf}_{\lambda \in [0,1]} \, q(\lambda)$$

solving a quadratic equation. The update of the iterate then is

$$\tilde{\tau}^{(k+1)} := \lambda^{(k+1)} \tau^{(k+1)} + (1 - \lambda^{(k+1)}) \tilde{\tau}^{(k)} \tag{61}$$

$$= \sum_{l=0}^{k} \lambda_l^{(k+1)} \tau^{(l)} \tag{62}$$

with $\lambda_l^{(k+1)} = \lambda^{(k+1)} \prod_{m=l+1}^{k+1} (1 - \lambda^{(m)})$. If the initial state $\tau^{(0)}$ is pure, we thus have an explicit form for all iterates in terms of a convex combination of pure states. The update of the density is correspondingly

$$\tilde{\rho}^{(k+1)} := \lambda^{(k+1)} \rho^{(k+1)} + (1 - \lambda^{(k+1)}) \tilde{\rho}^{(k)} \tag{63}$$

and the new Hartree potential as well as the new one-particle energy can be found by linear combination

$$V_H[\tilde{\rho}^{(k+1)}] = \lambda^{(k+1)} V_H[\rho^{(k+1)}] + (1 - \lambda^{(k+1)}) V_H[\tilde{\rho}^{(k)}] \tag{64}$$

$$E^{1e}[\tilde{\rho}^{(k+1)}] = \lambda^{(k+1)} E^{1e}[\rho^{(k+1)}] + (1 - \lambda^{(k+1)}) E^{1e}[\tilde{\rho}^{(k)}] \tag{65}$$

due to linearity of the Hartree operator V_H and E^{1e}. The xc-energy density and potential are also redefined in terms of the density $\tilde{\rho}^{(k+1)}$. It remains to define a termination condition. We measure progress in terms of the density, hence, as soon as

$$\|\tilde{\rho}^{(k+1)} - \tilde{\rho}^{(k)}\|_{L^2(\Omega)}$$

falls below a certain threshold ε the iteration is quit and the result $(E[\tilde{\rho}^{(k+1)}], \tilde{\rho}^{(k+1)})$ returned. All in all, we obtain the optimal damping algorithm in the form presented in algorithm 1.

Algorithm 1: Optimal damping algorithm

Input : Initial state $\tilde{\tau}^{(0)}$, tolerance ε
Output: Local minimum of E^{KS}: $(E^{KS}[\rho], \rho)$

for *k=1,...* **do**

$\left(\varepsilon_L^{(k+1)}, \phi_L^{(k+1)}\right)_{1\leq L\leq N} \leftarrow$ **solve linearized KS-problem** (15) for $H_{\tilde{\rho}^{(k)}}$;

Define KS orbital state and evaluate energies
$\tau^{(k+1)} \leftarrow \sum_{L=1}^{N} |\phi_L^{(k+1)}\rangle \langle \phi_L^{(k+1)}|$;
Compute $\left(E^{1e}[\tau^{(k+1)}], E^{H}[\tau^{(k+1)}], E^{xc}[\tau^{(k+1)}], \langle V_{xc}[\tau^{(k+1)}]\rangle_{\tau^{(k+1)}}\right)$ acc. to (56), (58) ;

Evaluate potential overlap integrals
Compute $\left\langle V_H[\rho^{(k+1)}]\right\rangle_{\tilde{\rho}^{(k)}}, \left\langle V_{xc}[\rho^{(k+1)}]\right\rangle_{\tilde{\rho}^{(k)}}$;

Line search

$$\text{Find } q^{(k+1)} \in \mathcal{P}_3(0,1) \text{ s.t. } \begin{cases} q^{(k+1)}(0) = E^{[}\tilde{\tau}^{(k)}] \\ q^{(k+1)}(1) = E^{[}\tau^{(k+1)}] \\ q^{(k+1)\prime}(0) = \left\langle H_{\rho_{\tilde{\tau}^{(k)}}} \right\rangle_{\tau^{(k+1)}} - \left\langle H_{\rho_{\tilde{\tau}^{(k)}}} \right\rangle_{\tilde{\tau}^{(k)}} \\ q^{(k+1)\prime}(1) = \left\langle H_{\rho_{\tau^{(k+1)}}} \right\rangle_{\tau^{(k+1)}} - \left\langle H_{\rho_{\tau^{(k+1)}}} \right\rangle_{\tilde{\tau}^{(k)}} \end{cases}$$

$\lambda^{(k+1)} \leftarrow \operatorname{argmin}_{\lambda\in[0,1]} q^{(k+1)}(\lambda)$;

Update iterate
Update $\tilde{\tau}^{(k+1)}, \tilde{\rho}^{(k+1)}, V_H[\tilde{\tau}^{(k+1)}], E^{1e}[\tilde{\tau}^{(k+1)}]$ acc. to (62), (63), (65) ;
Compute $\left(E^{H}[\tilde{\tau}^{(k+1)}], E^{xc}[\tilde{\tau}^{(k+1)}], \langle V_{xc}[\tilde{\tau}^{(k+1)}]\rangle_{\tau^{(k+1)}}\right)$ acc. to (56), (58) ;

Check for termination condition
if $\|\tilde{\rho}^{(k+1)} - \tilde{\rho}^{(k)}\|_{L^2} < \varepsilon$ **then**
  return $(E^{1e}[\tilde{\tau}^{(k+1)}] + E^{H}[\tilde{\tau}^{(k+1)}] + E^{xc}[\tilde{\tau}^{(k+1)}], \tilde{\rho}^{(k+1)})$;
end
end

6 Implementation in `concepts`

For the numerical experiments, we employ the `C++` hp-finite element solver `concepts`, for which we describe the newly implemented components in this chapter. The choice for `concepts` has been made since it provides high order finite elements for our mortar element discretization in the intermediate domain. During this project a spherical discretization, a generic mortar element and a DFT framework have been added to `concepts`. The structure of the individual sections in this chapter will be such that we first specify the requirements on our implementation. We emphasize concepts that are directly modeled by types in the code in bold face letters. In this way, we separate abstractions from implementation details that are explained in the follow-up, respectively. Throughout this chapter, the typewriter font will be reserved for `C++`-classes, software libraries and filenames.

6.1 Geometry

To model smooth spherical interfaces between the finite element discretization and the spherical domain the geometry section of `concepts` was extended to handle $3D$ curved elements. To explain our implementation, we first outline the general structure of the geometry in `concepts`. Throughout this chapter, we will use the notion of a hexahedral cell complex as introduced in section 4.2.1 to refer to vertices, edges, quadrilaterals and hexahedra and their mutual relations in the mesh.

6.1.1 Representation of the geometry in `concepts`

The geometry representation in `concepts` builds on a strict separation of **topological relations** between cells in the complex and its **geometric embedding**. The former are represented as abstract cells that include first order top-down incidence and orientation relations among n-cells and their $(n-1)$ dimensional boundary cells. In this manner, they form an incidence graph data structure that completely characterizes the topological structure of the mesh. On the other hand, the geometric embedding is merged with the abstract complex only on the level of top-dimensional cells, in the sense of a pairing of every abstract cell with a corresponding geometric mapping in a **geometric cell**. These cells are then aggregated in a container to form a **mesh**.

6.1.2 Implementation

We describe the extension of the geometry section of `concepts` focusing first on a smooth spherical discretization, from which we then pass to the general framework of representing curved hexahedral meshes. Our main intention during this project is to discretize the spherical surface smoothly in terms of a curved quadrilateral finite element mesh. We achieve this in a two step procedure. First, we approximate the spherical surface by a mesh of bilinear quadrilaterals whose vertices lie on the sphere. Subsequently, the mesh is stretched pointwise in the radial direction to coincide with the spherical surface. This discretization strategy is common in geosciences, where the radial stretching is known as the gnomonic projection. We implemented this quadrilateral element mapping in the geometric type `concepts::BallQuad3d`. Importantly, the construction via

gnomonic projection guarantees the property that coordinate changes (16) on boundary edge cells are affine, which is required to achieve a consistent finite element discretization.

The affinity of the element interface coordinate changes (16) is also important for the general inductive construction of the geometric complex formed by the finite element mesh. Precisely, it implies that once the boundary of a cell has been fixed by other cells, its parametrization can only interpolate between the corresponding mappings of the boundary cells (up to affine coordinate changes). An interpolation strategy that naturally incorporates these constraints is transfinite interpolation [36]. The associated interpolation concept of boolean interpolation operator sums generalizes straight-forwardly to $3D$. The transfinite geometric mappings are implemented in the types `concepts::BlendingHexahedron3d`, `concepts::BlendingQuad3d` and `concepts::StraightEdge3d` and are constructed inductively by specifying their boundary cells.

At the construction of these geometric types, we perform a change of paradigm in `concepts`. Up to this point, geometric mappings and topological objects were completely separated during their construction. Especially, no incidence or orientation information was exchanged between them. Instead, those properties were detected separately on both complexes performing two-dimensional geometric searches for the geometric complex. Such an approach is no longer feasible in three dimensions. Moreover, the required information is already present in the abstract topological complex at the construction of the geometric mappings and, thus, can be reused building topology and geometry side by side. As an additional benefit, this establishes consistency between the abstract topological cell complex and its geometric embedding.

6.1.3 The `Gmsh` interface

To construct complex meshes with curved boundaries in a scalable fashion, we designed a new interface to the mesh generator `Gmsh`. Precisely, `Gmsh` allows to construct $3D$ domains using built-in CAD functionality that can then be discretized into a mesh. The mesh can be curved with adjustable polynomial orders for the mappings. However, the meshes built by `Gmsh` are always non-conforming on general curved boundaries such as a sphere. Since we consider a spectral approximation technique, we need to circumvent this non-conformity by post-processing the mesh output by `Gmsh` at the boundary of the domain. For this purpose, we parse the mesh (`.msh`) as well as the unrolled geometry (`.geo`) file from `Gmsh`, the latter containing the information on the original mesh. In `concepts`, we then reconstruct a conforming curved boundary discretization according to the `Gmsh` mesh representing the cells in the geometry as abstract factories [37]. The mesh can then be obtained using the types `FiniteElementMeshGmsh` for hexahedra and `SphereMeshGmsh` for spheres, which are described in the next section.

6.2 Spherical Space

We describe the implementation of the continuous spherical space

$$\bigoplus_{i=1}^{M} \mathcal{S}_L(\mathcal{C}_i) \cap C^0(\mathcal{C}_i)$$

in `concepts`. To be able to reuse existing functionality, we conform to the established design principles for discrete spaces in `concepts`. We, thus, first outline the general structure of discrete spaces in `concepts` and then focus on our implementation of continuous spherical subspace of $\mathcal{S}_L(\cup_{i=1}^{M}\mathcal{C}_i)$.

6.2.1 General structure of discrete function spaces in `concepts`

In `concepts`, a discrete function space $\mathcal{S}(\mathcal{T})$ over a mesh $\mathcal{T} = \{K_i\}_{i\in I}$ is a **sequence of elements**. It provides a forward iterator over its elements by means of a **scanner**. Every element K_i aggregates an embedding in the cell complex formed by the mesh $\mathcal{T}$ in the sense of **topological incidence and adjacency relations** in the abstract cell complex paired with a **geometric parametrization** F_{K_i}. Moreover, an element also provides a local function space spanned by a set of **shape functions**[25] evaluated at quadrature nodes and corresponding **quadrature rules** for integration over the element. In addition, every element specifies a T**-matrix** T_{K_i} encoding the expansion of global basis functions from $\mathcal{S}(\mathcal{T})$ in local shape functions. This enables switching between element and global space facilitating the **assembly** (31). On top of this, extra-functionality for **graphical output** during the post-processing phase can be provided. In contrast, both **operator discretizations** as well as **synthesis of discrete functions** relative to the element spaces are outsourced entities.

6.2.2 Implementation of the spherical space

For the implementation of the spherical space $\mathcal{S}_L(\bigcup_i \mathcal{C}_i)$ a new namespace `sphere` has been designed. In the following, we will describe the individual types presenting new classes along with existing ones. The separation between newly added and existing functionality should be self-explanatory, since no spherical functionality was previously available in `concepts`.

The spherical space $\mathcal{S}_L(\bigcup_i \mathcal{C}_i)$ is encapsulated in the type `sphere::Space`, which is derived from the template `concepts::SpaceOnCells<Real>` that serves as base class for essentially all cell-based discrete function spaces in `concepts`[26]. It contains a sequence of elements of type `sphere::Sphere`, that can be forward iterated over using the `concepts::Scan` template.

Topology

The embedding of the `sphere::Sphere` elements in the abstract cell complex of the mesh $\mathcal{T}$ is achieved by the topological type `concepts::Sphere`. It is aggregated together with geometric information on the center and radius of the sphere in the geometric cell type `concepts::Sphere3d`. The purpose of the topology here is to specify the incidence relations between the `concepts::Sphere` and its topological boundary, which is of type `concepts::SphericalSurface`. Importantly, the topological spherical surfaces consistently integrate the spherical cells into a hexahedral complex by referencing topological quadrilaterals that refine them. This is achieved in

[25] and possibly their gradient

[26] We omit the namespace identifier in front of `concepts::Real` for brevity, the type being just an alias for `double`.

a sequence of pointers to `concepts::Quad` children that can be obtained from every `concepts::SphericalSurface` object. Thus, the `Connector` incidence graph data structure entails the full topological information on the cell complex generated by hexahedral finite elements and atomic spheres.

Element spaces and shape functions

To compute shape functions on sphere elements, we reuse the integrated Legendre polynomials and their derivatives from `concepts::Karniadakis` for the radial direction, while we added the types `concepts::Trigonometric` and `concepts::AssociatedLegendre`. All these types implement a $1D$ shape function interface that requires them to provide shape functions evaluated over prescribed quadrature points in a contiguous array. For memory efficiency, these arrays are shared by all objects of a concrete shape function type according to the flyweight pattern [37]. For spherical harmonics, the type `concepts::Trigonometric` provides the $L^2(0, 2\pi)$-orthonormalized cosine/sine factors in ascending order of m[27], which can be conveniently computed using the `C` standard library. In contrast, for the associated Legendre functions we resort to the `C`-library `libftsh` [32] that uses a recurrence via Jacobi polynomials which is extra-stabilized for over- and underflow by storing mantissa and exponent in separate variables. The enumeration scheme here is identical to that for spherical harmonics presented in (39). Tensorization of these $1D$ shape functions is then done by clients, so that types that discretize differential operators or synthesize discrete functions in the element space can be implemented efficiently.

The shape functions are complemented with a set of quadrature rules chosen according to Lemma 4.1. In `concepts`, these can be obtained from the factory type `concepts::QuadRuleFactory`. The caching according to the flyweight pattern is then left to the concrete quadrature rule. By default, `concepts` provides quadrature rules over the domain $[-1, 1]$. However, the domain of geometric parametrizations F_K in the mesh is canonically a cartesian product of $[0, 1]$ intervals. Unfortunately, this is a major source of errors in client code. The situation is especially critical for spheres, we use the domain

$$(r, \theta, \phi) \in [0, 1] \times [0, \pi] \times [0, 2\pi],$$

according to the element map (32). Firstly, this requires different quadrature rules in r- and θ- than in ϕ-direction (c.f. Lemma 4.1) and, secondly, needs all of these to be transformed differently. The `libftsh` library furthermore requires the θ-quadrature not to include the endpoints, while the ϕ-quadrature must be inherently periodic. To tackle these issues with reasonable development effort, we chose the pragmatic solution to avoid global caching of spherical quadrature rules and instead store them element-wise in the newly created type `sphere::QuadratureRule1d` that encapsulates all the required transformations. As long as the number of atoms is small compared to the polynomial degree, this is a minor inefficiency and can be neglected[28]. The element

[27] The ordering of shape functions is $\left(\frac{1}{\sqrt{2\pi}}, \frac{1}{\sqrt{\pi}}\cos(\phi), \frac{1}{\sqrt{\pi}}\sin(\phi), \ldots, \frac{1}{\sqrt{\pi}}\cos(L\phi), \frac{1}{\sqrt{\pi}}\sin(L\phi)\right)$. Note that no odd 0-frequency mode occurs.

[28] Take e.g. the number of associated Legendre functions to be stored, which scales with the square of the polynomial degree, as a reference.

space functionality of spheres is completed with the shape function and quadrature rule interfaces.

Assembly

For the T-matrices of spheres, we used the standard `concepts` template `concepts::TMatrix`, which stores the non-zero coefficients in a compressed column sparse matrix format. As described in section 4.3 shape functions that are discontinuous at the center are discarded, while all others are directly integrated in the global space. To be able to efficiently construct such a sparse T-matrix, a new constructor for the matrix entries in coordinate storage format has been added to the class template `concepts::TMatrix`. The assembly (31) itself is then encapsulated in the constructor of system matrices and vectors that iteratively collect all element contributions for this purpose.

Graphics

We conclude this overview on the sphere element by mentioning that for graphical output using VTK an extra type `sphere::SphereGraphics` was implemented. It can be used together with `concepts`' VTK output facilities to create files for $3D$-visualization of functions in the space $\mathcal{S}_L(\bigcup_i \mathcal{C}_i)$.

Discrete Function synthesis

The handling of discrete functions in the spherical element space (33) by means of coefficient vectors is encapsulated in the class template `sphere::Value`. This is a function object that evaluates a discrete function specified by a coefficient vector relative to local shape functions at a prescribed point. A discrete function from the global spherical space can be synthesized by first applying the element T-matrix to the coefficient vector. For DFT purposes we would also like the evaluation of discrete functions such as the electron density or Hartree potential to be efficiently performed in one step over all quadrature points. Using optimization techniques analogous to sum-factorization (c.f. section 4.3), this could reduce the cost of this operation from $\mathcal{O}(L^6)$ to $\mathcal{O}(L^4)$ and therefore bring orders of magnitude of performance[29]. Unfortunately, such an optimization is avoided in `concepts` by design. Nevertheless, the value type and its associates provide a powerful abstraction enabling easy high-level manipulation of discrete functions.

[29] In two dimensions, for a discrete function in the element space $\mathcal{Q}_p([0,1]^2)$ with coefficients $(c_{ij})_{0\le i,j\le p}$ relative to the basis induced by $1D$ shape functions $\{q_r\}_{0\le r\le p}$ the evaluation is done according to the following two-step procedure,

$$(c_{ij})_{0\le i,j\le p} \xrightarrow{\sum_{i=0}^{p} \cdot\, q_i(x_{1,k})} \left(\sum_{i=0}^{p} c_{ij} q_i(x_{1,k})\right)_{0\le k,j\le p} \xrightarrow{\sum_{j=0}^{p} \cdot\, q_j(x_{2,l})} \left(\sum_{j=0}^{p}\sum_{i=0}^{p} c_{ij} q_i(x_{1,k}) q_i(x_{2,l})\right)_{0\le k,l\le p},$$

where $(x_{s,k})$, $s = 1, 2$, are the quadrature nodes. The complexity of each of the steps is $\mathcal{O}(p^3)$.

Operator discretization: Bilinear and linear forms

To discretize the linearized Kohn-Sham problem (15) as well as the Poisson problem (29), objects that evaluate compute the element matrices for the integrals (40) and (41) are required. This is achieved by the newly added bilinear forms `sphere::Laplace` and `sphere::Identity3D`, where the latter is constructed specifying the potential at construction. They compute the element integrals for all pairings of shape functions in a single function call as described in section 4.3. The type `sphere::Riesz` does the corresponding task for linear forms, which is required to solve the Poisson problem. We note that for the FFT in the azimuthal integration, we use calls to the highly optimized FFTW3 library [31] throughout the implementation of these types.

Trace Space

For completeness, we note that in order to couple the finite element discretization to the spherical space the implementation of the trace space $\mathcal{S}_L(\bigcup_i \mathcal{C}_i)\big|_{\bigcup_i \partial\mathcal{C}_i}$ in `sphere::TraceSpace` was necessary. This includes an analogous class framework with `sphere::SphericalSurface` as element type with much functionality in common with `sphere::Sphere` and `sphere::Identity2D` to evaluate the L^2-inner product of spherical harmonics, which can be done analytically. Due to the analogy with `sphere::Space`, we do not further elaborate on the implementation of these parts here.

6.3 Mortar element space

Based on the presentation in section 4.4, we design a generic framework to couple discrete spaces via the mortar element method in the namespace `mortar`. In doing so, we perform a change of paradigm from traditional object-oriented programming using inheritance to generic programming. To use the full power of this approach, we add a few extra traits templates in the new namespace `traits` that encapsulate type information for discrete spaces. The new C++11 feature of template aliases is then used to facilitate the writing of intuitive generic code that syntactically resembles much the original `concepts` while being fully generic on the discretization space. To cover the details sufficiently, the subsequent discussion of the implementation will be somewhat more detailed than in the previous chapter.

6.3.1 Requirements

The idea is to construct a **composite discrete function space** V_b that adopts the elements from separate domains with associated discrete spaces V_1 and V_2. To incorporate the mortaring conditions for a particular choice of interface trace space as the mortar test space, the **projectors** from the mortar element space to either domain space must be constructed using appropriate **overlap bilinear forms**. The original T-matrices of the elements can then be concatenated with the corresponding projector. To be used naturally, abstractions for the **operator discretization** and **discrete function synthesis** on the level of the composite space are required.

6.3.2 Implementation

The mortar element framework is parametrized throughout by the types of two discrete spaces on separate domains. Unless mentioned otherwise, all types in the namespace `mortar` are parametrized by those discretization spaces. In order to keep verbosity low, we thus omit their extra mentioning with every type they parametrize.

Construction of domain projectors

The main task in the construction of the mortar element space is the construction of the appropriate domain projectors (50). For this purpose, we designed a template `mortar::MortarProjector`. In addition to the discretization spaces, it is parametrized by a policy index that specifies the trace space to be used as the mortar test space. For the sake of this presentation, we will assume the mortar test space to be V_2^Γ. By the technique of traits, the overlap bilinear form to compute S_Γ^1 and the overlap bilinear form to compute the inner product matrix S_Γ^2 will be found automatically by the compiler. Both of the latter are bilinear forms defined on the trace spaces. While the second bilinear form performs a simple inner product of shape functions, the implementation of the interface overlap bilinear form is more involved since a product of shape-functions of non-matching elements must be integrated. We will return to this issue at the end of this section.

A `mortar::MortarProjector` object builds the actual domain projectors at construction according to the recipe shown in section 4.4. To obtain the matrices S_Γ^1 and S_Γ^2, the system matrices assembled from the bilinear forms must be contracted to act only on boundary degrees of freedom[30]. Moreover, we need to invert S_Γ^2, perform matrix multiplication and reexpand the product to the original indexing convention of the global spaces. This linear algebra procedure is encapsulated in the function template `mortar::solveMatrixEquationOnSubspace`. It uses the `SuperLU`-interface provided by `concepts` to perform the matrix inversion. The restriction and reexpansion operations on system matrices corresponding to a pair of subspaces (or trace spaces) is implemented in the template `mortar::SubspaceIsomorphism`. The resulting coefficient matrix can then be directly be used to construct the domain projectors. Once construction is finished, the built domain projectors are available from the `mortar::MortarProjector` object through a `std::tuple`-like interface.

To concatenate domain element T-matrices with domain projectors, we added the member

```
TMatrix<Real>::concatenate(const TMatrix<Real>& ).
```

This implements exactly the operation described in (51), so that once called on an element T-matrix it will in the following act as a restriction operator from the mortar element space.

[30] The size of the coefficient vectors and matrices `concepts` uses is the same for trace spaces as for their parent space. Here, this necessitates an extra technical step. However, for boundary value problems this is especially convenient to handle inhomogeneous Neumann data.

Mortar element space

For the mortar element space itself it is not a priori clear whether we need an extra computational representation. An option is to simply drag along the two component spaces V_1 and V_2 and their associated types and use the projectors from the mortar element space whenever we e.g. assemble a linear system by hand. However, this is error-prone, will cause a lot of identical code and inhibits the composite function space from being used in generic programs. A much more effective approach is to fit the mortar space into the abstraction of a `concepts::SpaceOnCells`. In fact, providing a few extra types that abstract operator discretization, discrete function synthesis, etc. on the level of the mortar space, we can reuse all of `concepts` functionality. As a consequence, from a user's perspective the assembly of a linear system will require no more code for the mortar space than for simple linear finite elements. Moreover, we will be able to manipulate discrete functions from the mortar space on a high level which simplifies the implementation of DFT code substantially.

We, thus, create a template `mortar::Space` that is parametrized by the discrete space types. Objects of these types are aggregated in a tuple at construction. In addition, we pass a projector object as a policy to the constructor. This has two benefits. First, we outsource code related to specific spaces to a policy parameter and, secondly, this makes the `mortar::Space` reusable in other domain decomposition contexts[31]. During construction, the T-matrices of all the elements from both domain spaces are concatenated with the corresponding domain projector so that the `mortar::MortarProjector` can safely go out of scope at the end of the constructor.

The elements of a `mortar::Space` are of the domain-tagged type `mortar::Element` wrapping the actual domain element according to the decorator pattern [37]. The `mortar::Element` instantiations are derived from the common base `mortar::ElementBase` so that the domain tag can be queried at runtime. To satisfy the `concepts::SpaceOnCells` concept for the `mortar::Space`, we consequently provide a scanner over a sequence of pointers to `mortar::ElementBase`[32].

The introduction of element decorators enables a straight forward implementation of adaptors to build the familiar high-level abstractions of discrete function spaces for the `mortar::Space`. The core functionality is a generic element-based dispatcher differentiating by domain tag[33] that can elegantly be implemented by means of variadic templates. As a consequence, we obtain the adaptor [37] types `ElementFunction`, `ElementFormula`, `BilinearForm` and `LinearForm` that are all part of the `mortar` namespace. Via template aliasing they provide all the abstractions of the corresponding `concepts` types such as `Laplace`, `Identity`, `Riesz`, etc. that is familiar from other contexts for the `mortar::Space`. Thus, the namespace `mortar` provides the full PDE discretization functionality as the other namespaces in `concepts`.

[31] As an example, the projector type `mortar::DGProjector` builds a composite space that is discontinuous across the domain interface.

[32] For backward compatibility with non-polymorphic code (e.g. the `graphics` section), we provide an alternative view of the `mortar::Space` via the type `mortar::BlankView`. Modelling also the `concepts::SpaceOnCells` concept, it provides a scanner over the unwrapped elements of the domain spaces.

[33] Note that for backward compatibility with non-polymorphic code, the dispatcher falls back to differentiation based on element type if called with other elements than the ones from the `mortar` namespace.

Mortar overlap bilinear form

To compute the overlap integrals between finite element shape functions and spherical harmonics required for S_Γ^1, we defined the bilinear form `sphere::SphereFeIdentity`. The quadrature used to evaluate this bilinear form can be chosen by the user to be either based on spherical harmonics or on finite elements. In case of spherical harmonics quadrature, to evaluate finite elements the quadrature grid must be decomposed over the quadrilateral mesh on the sphere. This decomposition requires the inverse of the spherical quadrilateral element parametrization, which is provided in the `concepts::BallQuad`-member `inverseMapping`. During evaluation of the overlap bilinear form, the finite element shape functions must then be evaluated in local coordinates. Conversely, when the finite element quadrature grid is chosen, spherical harmonics must be evaluated over the FE quadrature grid. The inverse of this parametrization, however, is a simple transformation to spherical coordinates. The discrepancy in numerical performance of these options anticipated in chapter 4 will be discussed in chapter 7.

6.4 The DFT framework

For problems in density functional theory a new namespace `dft` was designed. The basic components implemented are a generic solver for the linearized Kohn-Sham equations and the optimal-damping algorithm. The framework can be extended to study other quantum many-body problems (e.g. via Hartree-Fock theory) or other one-particle Schrödinger problems. We start our presentation with a description of the Kohn-Sham solver. Subsequently, we will then focus on the implementation of the optimal-damping algorithm.

6.4.1 Solution of the linearized Kohn-Sham problem

A generic solver for the linearized Kohn-Sham eigenvalue problem (15) as required by iterative algorithms that is parametrized by the discretization space is implemented in `concepts`. Since the linearized Kohn-Sham equations represent a special class of Schrödinger problems, we separate the functionality for the solution of Schrödinger equations from the specific requirements of the Kohn-Sham problem.

Requirements

The basic task is to **construct and solve the generalized matrix eigenvalue problem** (28) for a prescribed discretization space and a set of potentials V_n, $V_H[\rho]$ and $V_{xc}[\rho]$. For the solution of the matrix eigenvalue problem, one can employ the **Cholesky-Lanczos algorithm** described in chapter 4. A specific requirement is the efficient assembly of the discretized Hamiltonian H_ρ (26) during the operation of an iterative algorithm. An algorithm such as ODA (algorithm 1) will alter the density ρ in every step and, therefore, input different Hartree and exchange-correlation potentials into the Kohn-Sham solver. This necessitates a **reassembly of the Hartree-xc matrix** corresponding to the multiplication operator $V_H[\rho] + V_{xc}[\rho]$ in every step. On the other hand, the **one-particle Hamiltonian matrix**,

$$h_{ij} = \int_\Omega b_i h b_j dx$$

is constant as is the **overlap matrix** S_{ij} (27). Therefore these matrices have to be assembled only once at the beginning of the iterative procedure and can then reused in every step.

Implementation

To provide a reusable implementation, we separate the concrete requirements of the Kohn-Sham solver from that of more generally any one-particle Schrödinger solver. In particular, the form of the discrete eigenvalue problem (15) is the same for other one-particle Schrödinger problems. In general, only the potential varies. Therefore, the solution technique via the Cholesky-Lanczos algorithm applies to Schrödinger problems in general.

Thus, we encapsulate solution procedure for generalized eigenvalue problems of the form (28) in a base class `dft::SchroedingerSolver`. This template is parametrized by the discretization space so that by virtue of traits and template aliasing

the choice of appropriate bilinear form type for kinetic energy and potential multiplication operators is delegated to the compiler.

For the Cholesky-Lanczos algorithm, we employ the `Arpack++` [38] interface to the Fortran library `Arpack` for Lanczos-iteration and the `Cholmod` C-library [39] for Cholesky factorization. For resource management, a new interface has been added in the type `concepts::Cholmod` performing Cholesky-factorization of a `concepts::SparseMatrix`. It provides several members for the solution of linear systems. The implementation is inspired by the Cholmod-interface of the C++ library `Eigen` [40]. The transformation and solution of the generalized matrix eigenvalue problem (28) is entirely encapsulated in the type `ArpackHelperSA`. It is constructed by specifying the Hamiltonian and overlap matrices as well as the shift parameter ν for the lower bound on the Hamiltonian. For the transformation to a standard eigenvalue problem it then factorizes the shifted Hamiltonian in `concepts::Cholmod`. `ArpackHelperSA` objects furthermore provide several members performing matrix-vector multiplication so that `Arpack++` can directly execute Lanczos-iteration on them. To be adapted to the requirements of ODA, the Hamiltonian matrix can be reset any time using the member `setA`. For reusability, we implemented the functionality common to smallest- and largest-eigenvalue Lanczos-iteration in the fashion of the curiously recurring template pattern in a base template, so that `ArpackHelperSA` and `ArpackHelperLA`[34], which we can provide essentially at no cost, are derived from appropriate instantiations of the `ArpackHelperBase` template.

Due to the frequently changing Hamiltonian in iterative algorithms, the base class `dft::SchroedingerSolver` delegates the computation of the corresponding matrix (26) to derived classes via a virtual function. A concrete derived type is `dft::KohnShamSolver` that caches the one-particle Hamiltonian matrix h, while it reassembles the Hartree-xc matrix at every call of its application operator with an `concepts::ElementFormula` corresponding to $V_H[\rho] + V_{xc}[\rho]$. Equipped with the discretized Hamiltonian, the `solve` member from the base type `dft::SchroedingerSolver` then calls the Cholesky-Lanczos solver, whose result is then returned as a sequence of coefficient vectors and eigenvalues.

Finally, we note that for the electron-core interaction V_{nuc}, we implemented the type `concepts::NuclearPotential` that is constructed specifying charge numbers and positions of atoms. Alternatively, in the mortar element setting, the user can also specify a sequence of topological spheres, where the cores reside. To prevent ill-defined quadrature and numerical cancellation at the discretization of the electron-core potential (41) an additional flag can be set if the r^2-term of the volume form on every `sphere::Sphere` is already included in the potential, so that the $\frac{1}{r}$-singularities of V_{nuc} effectively does not appear in the numerical integration.

6.5 The Optimal Damping Algorithm for Kohn-Sham DFT

We describe the implementation of algorithm 1 to solve the non-linear minimization problem posed by the extended Kohn-Sham model . In a first step we identify the abstract

[34] `ArpackHelperLA` is designed to solve generalized eigenvalue problems for the largest part of the spectrum.

entities required by the ODA. This facilitates a modular and extensible framework for DFT calculations allowing easy adjustment of individual components in the Kohn-Sham model according to custom needs. The mapping to classes is described in a second step.

6.5.1 Requirements

To start, we need to store all electronic states $\Phi^{(k)} = (\phi_1^{(k)}, \dots, \phi_N^{(k)}) \in V^N$ appearing throughout the iteration as solutions to the Kohn-Sham equations. Structurally, they are Slater determinants and, therefore, we will refer to such states as **Slater states**.

In terms of observables, the most important one in the ODA is the particle density. It appears in two forms. First, in every iteration step we need the density of a Slater state, for short **Slater density**. Moreover, we need to be able to form convex combinations of Slater densities as required by the update of the ODA resulting in **mixed Slater densities**[35].

Furthermore, the **Hartree** and **exchange-correlation operators** V_H, $\mathcal{E}_{xc}$, V_{xc} are required by the Kohn-Sham model. They should be provided as (non-)linear operators defined on the density discretization space mapping a given density to the corresponding observable. In the case of the Hartree operator we require in addition linearity semantics in the sense that a linear combination of its outputs must satisfy

$$\lambda_1 V_H[\rho_1] + \lambda_2 V_H[\rho_2] \equiv V_H[\lambda_1\rho_1 + \lambda_2\rho_2].$$

This is required to implement the convex mixing update for the Hartree potential efficiently.

In addition to merely representing observables, we need to be able to compute their **expectation values**. In the case of multiplication operators such as $V_H[\rho]$ and $V_{xc}[\rho]$, which are **diagonal in position space**[36], this is an integral of the observable over the density-weighted measure, whether the density corresponds to a pure Slater state or a mixing of such. The exchange-correlation energy is an integral over only the associated energy density $\mathcal{E}_{xc}[\rho]$. If the observable is **not diagonal in position space**, which is the case for the one-particle operator h due to the Laplacian in the kinetic energy, one has to differentiate between pure and mixed Slater states. For a pure Slater state $\Phi = (\phi_1, \dots, \phi_L) \in V^N$, we can compute its expectation value via

$$\langle h \rangle_\Phi = 2 \sum_{L=1}^{N} \langle \phi_L | \, h \, | \phi_L \rangle \, .$$

[35] Recall that the density in the k-th step is

$$\tilde{\rho}^{(k)} = \sum_{l=0}^{k} \lambda_l^{(k)} \rho^{(l)}.$$

[36] That is they are multiplication operators

$$V : \left\{ \begin{array}{lcl} L^2(\Omega) & \to & L^2(\Omega) \\ \phi(x) & \mapsto & V(x)\phi(x), \end{array} \right.$$

For **mixed Slater states**, such as the k-th ODA iterate

$$\tilde{\tau}^{(k)} = \sum_{l=0}^{k} \lambda_l^{(k)} \tau^{(l)}, \qquad \text{where } \tau^{(l)} = \sum_{L=1}^{N} |\phi_L^{(l)}\rangle \langle \phi_L^{(l)}| ,$$

the expectation value results from a convex mixing of Slater state expectation values,

$$\langle h \rangle_{\tilde{\tau}^{(k)}} = \sum_{l=0}^{k} \lambda_l^{(k)} \langle h \rangle_{\Phi^{(l)}} .$$

Since the update of the one-particle energy in algorithm 1 is a convex mixing of the expectation value in the last iteration step and the expectation value of the newly computed Kohn-Sham orbitals, such an expectation value is not directly required by the ODA, however. Still, this rule is useful if the evolution of general observables during the ODA is to be reconstructed. We conclude this overview on the requirements of the ODA by the **polynomial interpolation** and **minimization** needed for the line search.

6.5.2 Implementation

The overview on the implementation of the ODA for Kohn-Sham models is divided into two subsection. We first describe the individual components and then explain their cooperation in the implementation of ODA.

Individual components

Following our strategy in the previous paragraph, we design a framework for the ODA parametrized by the discretization space of the electronic orbitals. The ODA is governed by a central object of type `dft::ODADriver` that controls the iteration process and stores a set of variables that parametrize the Kohn-Sham model. In particular, the discretization space of the electronic orbitals is stored by the `dft::ODADriver`.

Slater states as outlined in the previous section are objects of type `dft::SlaterState` containing the coefficient vectors of their orbitals relative to the basis of V as well as a reference to V itself. In every step of the iteration, the `dft::ODADriver` saves a new such object as well as a new damping-parameter λ to its history.

The particle density of a Slater state is represented as a `dft::SlaterDensity` object. This is essentially a typedef for the newly created template `concepts::ElementFormulaMultiVector` that evaluates multiple functions over the same discrete space in parallel. It is parametrized by a discrete function synthesizer on the element level as well as a function object to be applied to the values of the discrete functions. In case of `dft::SlaterDensity` the latter is given in `dft::ClosedShellSlaterDensity`. Internally, the discrete functions are characterized by a reference to a sequence of coefficient vectors as provided by `dft::SlaterState` objects. To conform with the existing functionality in `concepts` instantiations of `concepts::ElementFormulaMultiVector` are derived from `concepts::ElementFormula`, which is the common base type for all functions defined in terms of elements of a space. In addition to Slater states, the `dft::ODADriver` also keeps track of the associated particle densities appearing throughout the iteration.

Mixed state densities for the current iterate are implemented in class `dft::MixedDensity`. It stores a sequence of pointers to Slater densities as well as a sequence of coefficients used for their linear combination. In order to perform the density update in algorithm 1 it provides the member `convexMixing` that accepts a damping parameter and a Slater density.[37] Due to the ring structure of the discretization spaces available in `concepts`, an optimized implementation of `dft::MixedDensity` might expand pointwise products of orbitals in a higher-order space requiring the storage and synthesis of only a single discrete function at evaluation. While this is a common optimization in quantum chemistry packages [7], we postpone its implementation to the future during this project.

The Hartree- and xc-operators are represented as function objects parametrizing the ODA. Assuming the electron density is provided in a `concepts::ElementFormula` the requirement on all these operators is to return the corresponding potential/energy density in a `concepts::ElementFormula`. The Hartree operator V_H is implemented in `dft::HartreeOperator` that provides the Hartree potential of a particular electron density solving the associated Poisson equation with homogeneous Dirichlet boundary conditions as described in chapter 4. For this purpose, the Hartree operator aggregates an object of type `dft::PoissonSolver`. The return type of the `dft::HartreeOperator` is `dft::HartreePotential` representing a function on the discrete space of the `dft::PoissonSolver` object[38]. The update is again implemented efficiently via a linear combination of coefficient vectors in the member function `convexMixing` that accepts a damping parameter and another `dft::HartreePotential` object.

We emphasize that the `dft::HartreeOperator` may in fact be used in a more general context to calculate expectation values of the Coulomb interaction. In particular, one can use it to compute arbitrary two-electron Coulomb integrals of the form

$$\int_{\Omega\times\Omega} \frac{\phi_i(x)\phi_j(x)\phi_k(y)\phi_l(y)}{|x-y|} dxdy,$$

which is necessary for wave-function methods. This is achieved inserting a pointwise product of electronic orbitals into the `dft::HartreeOperator` and then integrating with respect to the weight of another pointwise product of orbitals. However, the computational effort of this task involves the solution of N^2 Poisson equations and N^4 subsequent integrations which quickly becomes prohibitive[39]. The Coulomb expectation values are not further analyzed in this thesis and may be investigated in future work.

For the xc-operators a generic framework was designed that is parametrized by xc-tags. The xc-operators are available in the templates `dft::XCEnergyOperator` and `dft::XCPotentialOperator`, the energy density and potential are accessible in `dft::XCEnergy` and `dft::XCPotential`. They are implemented instantiating the `dft::LDAFormula` template by appropriate function objects implementing the

[37] Note that C++ as an alternative to the `convexMixing`-method allows operator-overloading. However, we chose to avoid this since the `convexMixing` member increases encapsulation.

[38] This includes aggregation of a triple of coefficient vector, discrete space and corresponding value object.

[39] The ring structure of the discretization space for the pointwise product of orbitals entails some potential for optimization if the dimension of the space and the number of electronic states are of the same order.

xc-functional. In particular, the xc-functional specified by the sum of (6) and (7) are implemented in the fully specialized templates `dft::XCEnergyFormula` and `dft::XCPotentialFormula` for the tag `dft::PW92`.

Expectation values of observables are computed in a function object `dft::ExpectationValue`. At its construction the discretization space V must be specified. To calculate the expectation value for a multiplication operator, one needs to pass the observable and the electron density both as `concepts::ElementFormulas` to the function application operator so that an integration can be performed internally. For non-multiplication operators, an overload is provided accepting the discretized observable in the form of a `concepts::SparseMatrix` as e.g. in the case of the one-particle operator h

$$h_{ij} := \int_\Omega b_i h b_j dx, \quad \forall i, j$$

that is assembled over the basis in V and a `dft::SlaterState` Φ containing the coefficient vectors $(C_{iL})_{1\leq i\leq \dim V, 1\leq L\leq N}$. It then internally computes the expectation value via

$$\langle h\rangle_\Phi = 2\sum_L \langle\phi_L|\, h\, |\phi_L\rangle = 2\sum_{L=1}^{N}\sum_{i,j=1}^{\dim V} h_{ij}C_{iL}C_{jL}.$$

The expectation value for non-multiplication operators on a mixed state is a convex combination of expectation values of Slater states as shown in the requirements section and therefore could be provided in another overload in principle. However, since the ODA does not strictly require the evaluation of expectation values of mixed states, we leave the implementation of this method to the future.

Finally, the implementation of the cubic interpolation and line search in the type `dft::CubicIntPolynomial` completes our outline of the components for the ODA.

Implementation of the ODA

The implementation of the optimal damping algorithm 1 is done in the `dft::ODADriver`. To encapsulate the computation of energies and Fock expectation values[40], we designed the types `dft:SlaterEnergies` and `dft:MixedEnergies`. The former uses a pair `dft::SlaterState` τ and `dft::SlaterObservables` objects passed at construction to calculate the energies

$$\left(E^{1e}[\tau], E^H[\tau], E^{xc}[\tau], \langle V_{xc}[\tau]\rangle_\tau\right)$$

acc. to equations (56). (58). The actual computations are delegated to a function object of type `dft::ExpectationValue` aggregated by `dft::ODADriver`. The struct `dft::SlaterObservables` contains all the relevant observables for the ODA, which are

$$(\rho_\tau, h, V_H[\rho_\tau], \mathcal{E}_{xc}\circ\rho_\tau, V_{xc}\circ\rho_\tau), \tag{66}$$

where h denotes the matrix of the one-particle Hamiltonian that can be obtained from the `dft::KohnShamSolver`. The type `dft::MixedEnergies` is defined analogously except that we need to pass `dft::MixedEnergies`, `dft::MixedObservables`

[40] The Fock expectation value is the expectation of the Hamiltonian.

objects from the previous iterate and, in addition, a `dft::SlaterEnergies` object for the current Kohn-Sham orbitals as well as the determined optimal damping parameter at construction. This will then update the energies according to the newly resulting iterate. In analogy to Slater states, the struct `dft::MixedObservables` also contains the data (66). It remains to compute the overlap expectation values (60). This is achieved in the members `fockOverlapExp`. In case of `dft::SlaterEnergies` this can be either called with the sequence of eigenvalues output by the Kohn-Sham solver, whereas for `dft::MixedEnergies` it must be computed passing the corresponding `dft::SlaterObservables` and `dft::MixedDensity` objects.

The structure of the implementation of the ODA is then outlined in algorithm 2, where the initialization phase, implemented in the member `initialize` and the iteration phase in `run` are patched together to form a single procedure.

Algorithm 2: Pseudo-code presentation of the optimal damping algorithm

```
Input  : error tolerance ε
Output: Local minimum (ρ_f, E_f) of E^KS
// solve initial linearized Kohn-Sham problem
slaterStates[0] ← ksSolver.solve(0);
dampingParameters[0] ← 1;
// set up Slater observables and energies
slaterDensities[0] ← SlaterDensity ( slaterStates[0] );
slaterObservables ← SlaterObservables ( ksSolver.h(), slaterDensities[0] );
slaterEnergies[0] ← SlaterEnergies ( expectationValue, slaterStates[0],
                                                        slaterObservables )
// initialize iterate observables and energies
iterateObservables ← MixedObservables ( ksSolver.h(), slaterDensities[0] );
iterateDensities[0] ← MixedDensity ( slaterObservables.density );
iterateEnergies[0] ← MixedEnergies ( slaterEnergies[0] );
for k = 1,... do
    // solve linearized Kohn-Sham problem
    ( slaterStates[k], slaterEnergies[k].setFockOverlapExp(·) )
            ← ksSolver.solve( iterateObservables.hartreePotential
                                    + iterateObservables.xcPotential ) ;
    // set up Slater observables and energies
    slaterDensities[k] ← SlaterDensity ( slaterStates[k] ) ;
    slaterObservables.reset( slaterDensities[k] ) ;
    slaterEnergies[k] ← SlaterEnergies ( expectationValue, slaterStates[k],
                                                slaterObservables ) ;
    iterateEnergies[k].fockOverlapExp( slaterObservables,
                                        iterateObservables.density ) ;
    // line search
    q ← CubicIntPolynomial (
              iterateEnergies[k].energy(), slaterEnergies[k].energy(),
              slaterEnergies[k].fockOverlapExp() -iterateEnergies[k].fockExp(),
              slaterEnergies[k].fockExp() - iterateEnergies[k].fockOverlapExp() )
    dampingParameters[k] ← q.argmin() ;
    // update the iterate
    iterateObservables.convexMixing(dampingParameters[k], slaterObservables) ;
    iterateDensities[k] ← MixedDensity ( iterateObservables.density )
    iterateEnergies[k] ← MixedEnergies (expectationValue,
                                    iterateEnergies[k-1], iterateObservables,
                                    slaterEnergies[k], dampingParameters[k] ) ;
    // check for termination condition
    if || iterateDensities[k] - iterateDensities[k-1] ||_{L^2} < ε || k == maxIterations
    then
        return ( iterateEnergies[k], iterateDensities[k] ) ;
    end
end
```

7 Numerical study

In this chapter, we present the numerical experiments performed in `concepts`. We first analyze the quadrature error in the construction of the mortar element space (21), which is intimately related to the computation of spherical harmonic transforms of finite elements. Subsequently, we then test the a priori error estimate from Theorem 3.11 on an isolated oxygen atom. An extension to real molecular systems is under way, but out of the scope of this thesis. The geometry we use for this chapter will be throughout the one presented in figure 1.

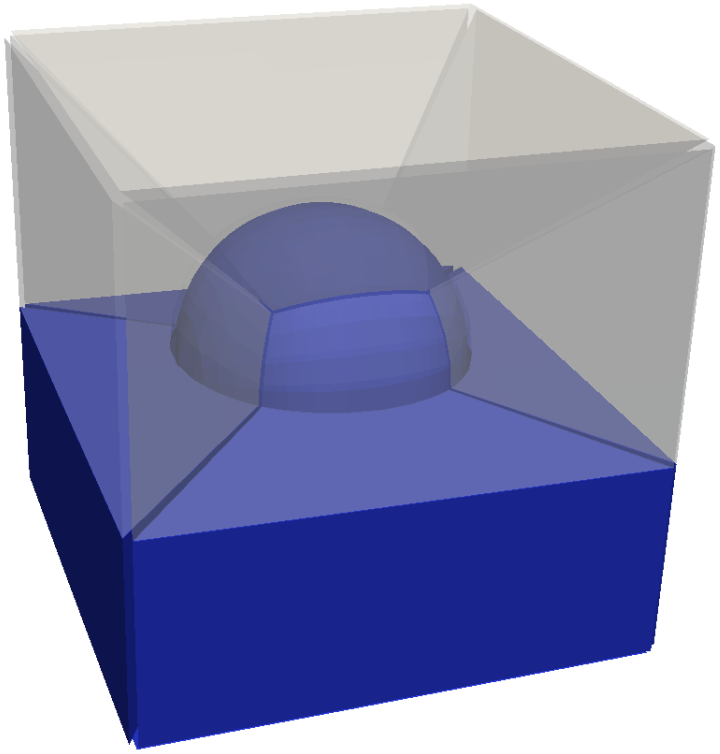

Figure 1: Mono-atomic geometry discretized by a single sphere surrounded by six curved hexahedra

7.1 Spherical harmonic transforms and the effect of quadrature on the mortar interface basis functions

As a first application, we investigate the effect of quadrature on the precision of the overlap matrix S_1^Γ. Since this matrix directly enters in the construction of the mortar transmission operator (48) and, therefore, in the interface basis functions (49), we should aim for high precision.

The coefficients in the matrix S_1^Γ admit another natural interpretation due to the orthonormality of spherical harmonics. They are the coefficients of the L^2-orthogonal

projection of finite elements on the corresponding spherical harmonics space, commonly known as the spherical harmonic transform. Due to this more natural interpretation, we study spherical harmonic transforms of finite elements.

In the following, we denote by $u^{FE} \in S^p(\Gamma, \mathcal{T}\big|_\Gamma)$ a finite element basis function on the spherical trace Γ that is either a vertex mode, edge mode or an interior quadrilateral degree of freedom. The approximation obtained for the L^2-orthogonal projection by quadrature from the overlap matrix S_1^Γ at spherical harmonic order l is denoted by $u_{(l)}^{SH} \in \bigoplus_{0\leq i\leq l} E_i$. To identify exponential convergence, we are specifically interested in the geometric means of the errors. Thus, for the spherical harmonic transform convergence, we consider

$$\frac{\|u^{FE} - u_{(l)}^{SH}\|_{L^2(\Gamma)}}{\|u^{FE}\|_{L^2(\Gamma)}}$$

averaged over all finite element degrees of freedom of equal polynomial degree. This is presented in figure 2. The vertical arrows identify the range of actual errors measured. As observed in the theoretical chapter 3, the low regularity of finite elements being only piecewise smooth on the sphere inhibits a high-order convergent approximation by means of spherical harmonics. Instead, in general, we can expect the convergence to be only algebraic. These unfavorable approximation properties of spherical harmonics on finite elements were also the reason why the approximation properties proven in [8] for the mortar coupling of spherical harmonics to plane-waves do not directly carry over to our setting.

The different approximation properties of finite elements versus spherical harmonics are represented in the markedly different convergence of the quadrature error. The finite element quadrature in figure 3 induces an exponentially decaying error, which is due to the smoothness of the integrand when pulled back to the reference quadrilaterals. This is in sharp contrast to spherical harmonics quadrature, where the error converges only algebraically due to the lack of global smoothness of finite elements on the sphere.

The conclusion of this observation is on the one hand that finite element quadrature must be used in the evaluation of the transmission matrix (48). On the other hand, this also means that spherical harmonic transforms of finite elements should never be performed using spherical harmonics quadrature despite the availability of highly optimized algorithms.

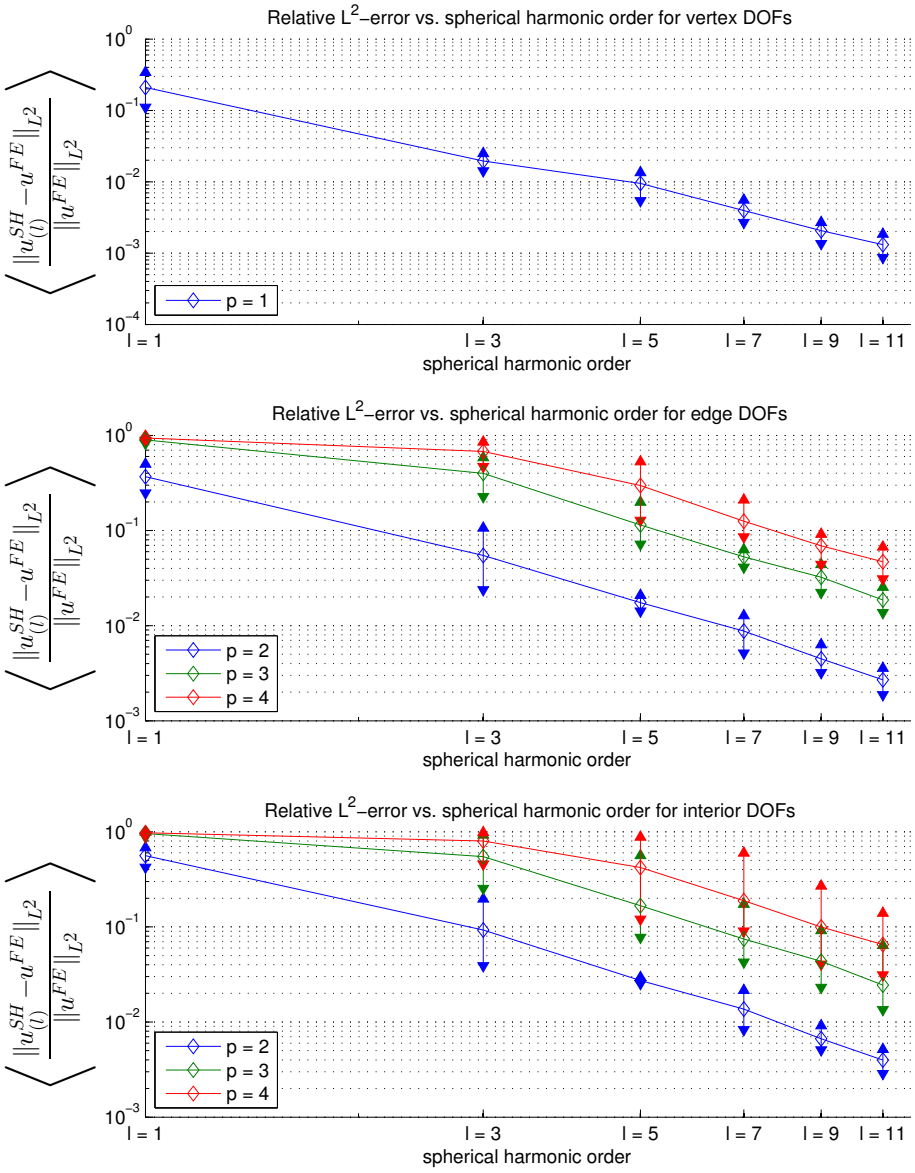

Figure 2: Convergence of the geometric average of the L^2-error of spherical harmonic transforms of finite element degrees of freedom (DOFs) for vertex modes (bilinear), edge modes (linear along one direction, higher order along the other), interior modes (higher order along both directions).

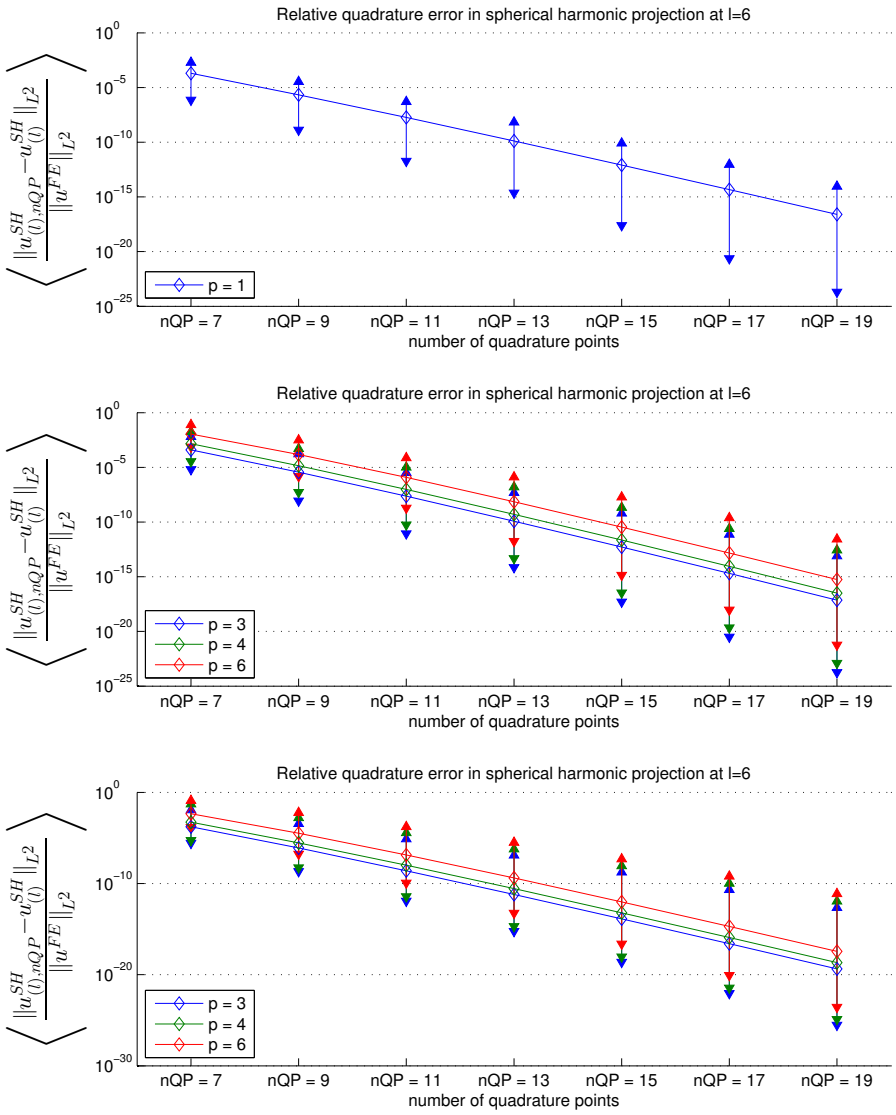

Figure 3: Relative quadrature error of the spherical harmonic transform $u^{SH}_{(l),nQP}$ of the finite element u^{FE} when using finite element quadrature. The reference transform $u^{SH}_{(l)}$ was computed at $nQP = 39$ quadrature points along every dimension of the reference Quad. Displayed is the geometric mean and the range of the quadrature errors for all degrees of freedom (DOFs) whose maximum polynomial degree is equal (top: vertex modes (bilinear), middle: edge modes (linear in one direction, higher order along the other), bottom: interior modes (higher order along both directions)).

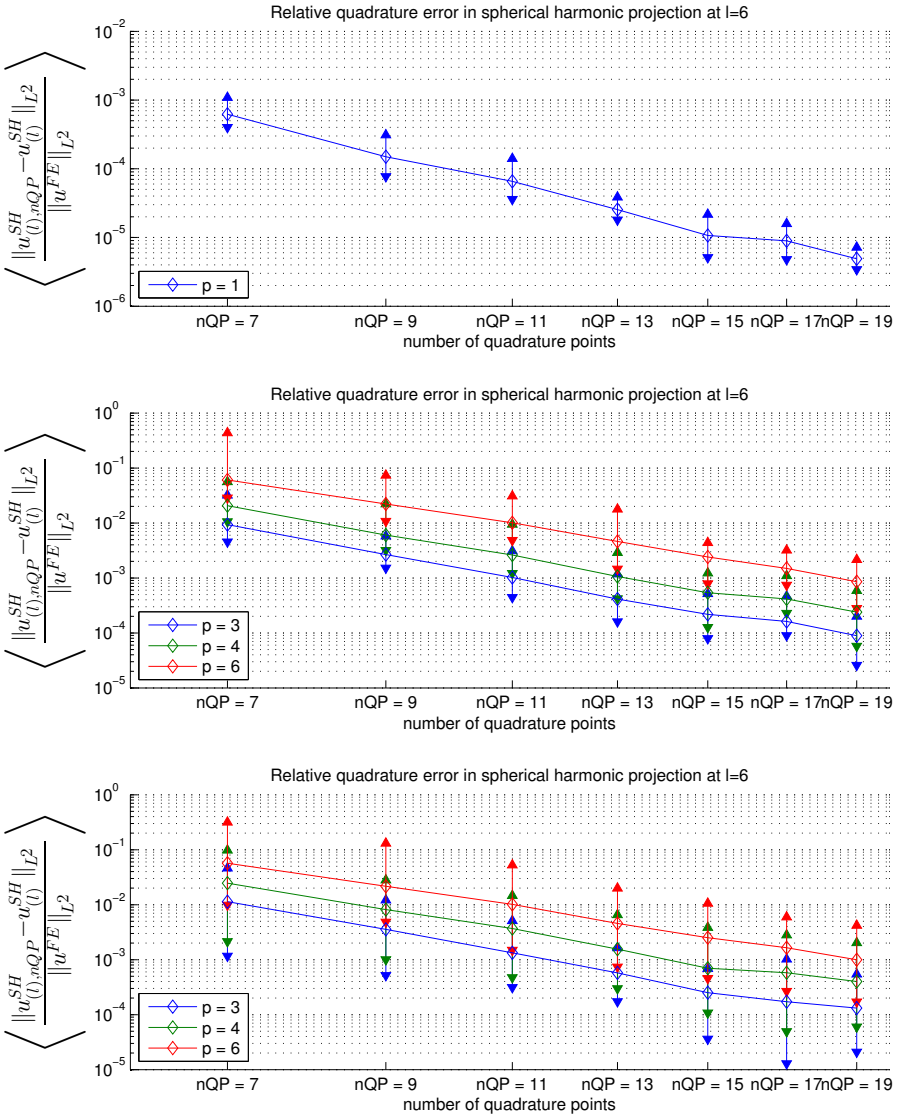

Figure 4: Relative quadrature error of the spherical harmonic transform $u^{SH}_{(l),nQP}$ of the finite element u^{FE} when using spherical harmonics quadrature. The reference transform $u^{SH}_{(l)}$ was computed at a Spherical Harmonic quadrature with $nQP = 39$ points in z-direction. Displayed is the geometric mean and the range of the quadrature errors for all degrees of freedom (DOFs) whose maximum polynomial degree is equal (top: vertex modes (bilinear), middle: edge modes (linear along one direction, higher order along the other), bottom: interior modes (higher order along both directions)).

7.2 Optimal damping convergence study and Schrödinger problem in the self-consistent limit

This final section is devoted to the verification of the a priori error estimate for the linearized Kohn-Sham problem in section 3. Due to time restrictions of this project, we do this on a model problem consisting of an isolated oxygen atom. We use the same mesh as in the previous section. Moreover, we ignore approximations for the correlation in the xc-potential during the following study. This restriction is not yet satisfactory from a physical point of view. However, it represents the current state of our implementation, which will eventually target the simulation of realistic molecules. From a mathematical point of view, the focus on a single atom is not necessarily a restriction, since the problem remains the resolution of the singularity at the core.

Our model is characterized by a charge with $Z = 8$ in the center of the atomic sphere so that we are looking for 4 spatial orbitals in the Kohn-Sham problem of closed-shell DFT. To obtain the Kohn-Sham Hamiltonian in the self-consistent limit as required by Theorem 3.11 an estimate for the self-consistent density is required.

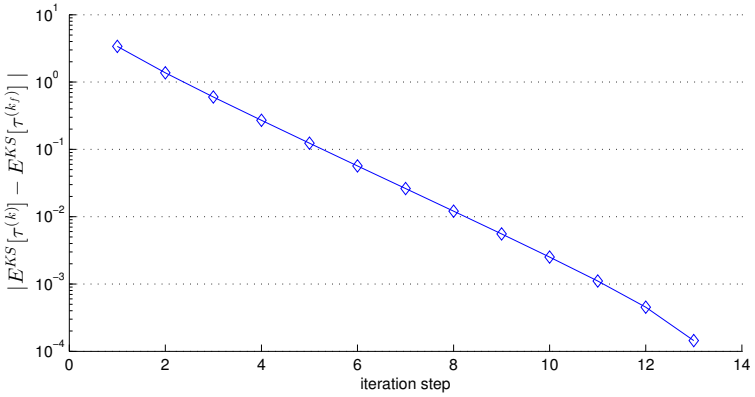

Figure 5: Convergence of the Kohn-Sham energies in the optimal damping algorithm as a function of the number of iterations.

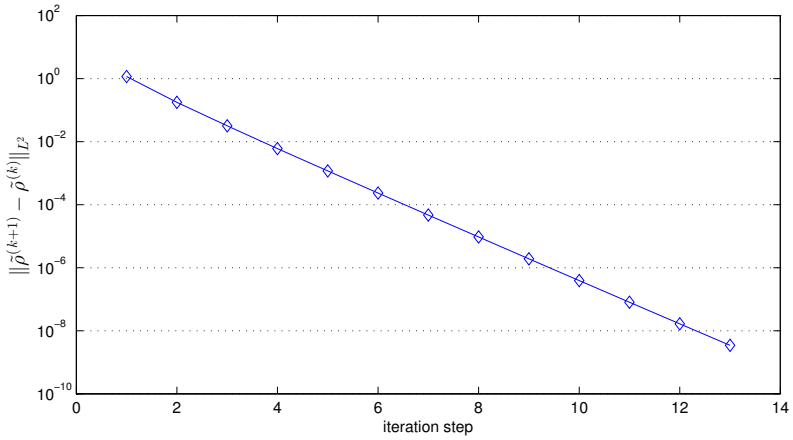

Figure 6: Convergence of the difference of the densities of adjacent iterates in the optimal damping algorithm as a function of the number of iterations.

We compute this estimate for the density using our implementation of the optimal damping algorithm 1. For this purpose, we use a uniform polynomial degree of $p = 6$ on all elements, so that our discretization space is V_6 (c.f. (21)). The maximum number of iterations is set to 15. We restrict to this moderate polynomial degree, since even with 15 iterations, at the end of the ODA a single evaluation of the electron density will require the synthesis of roughly 60 discrete orbitals, which causes for a drastic increase in the computational time compared to the beginning. The computations were performed on a local machine at the Institute of Mathematics of TU Berlin that is equipped with an AMD Phenom II X6 1090T six-core CPU at cycle frequency 3.2 GHz, 64 KB L1 Cache, 512 KB L2 Cache and 6 MB L3 Cache and 16 GB RAM. The total duration was around 90 minutes.

In the presented setting, we obtained convergence of the Kohn-Sham energies shown in figure 5. The slope can be determined at -0.35 in the semilogarithmic plot. The difference of densities shown in figure 6 converges exactly with twice this order. The final electron density is visualized in figure 7.

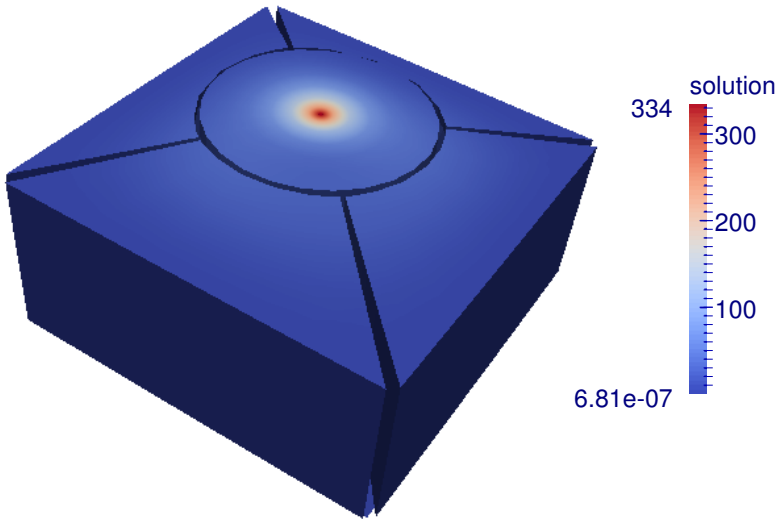

Figure 7: Plot of the particle density output by the ODA.

The Schrödinger problem is then initialized with the Hartree-potential and the exchange potential corresponding to the self-consistent density obtained from the ODA. We study the convergence of eigenvectors and eigenvalues under successive refinement the polynomial degree of the space V_p for the values between $p = 2$ and $p = 14$. The highest refinement level serves as a reference. The results are shown in figures 8 and 9. Both the eigenvectors and eigenvalues display exponential convergence in the asymptotic setting with a slope of -0.26 for the eigenvectors and -0.41 for the eigenvalues on average in the semilogarithmic plot. This does not quite correspond to the doubling of convergence order we expected from Theorem 3.11, however, we may still be satisfied with these results. All in all, the a priori error estimate is confirmed for this mono-atomic problem.

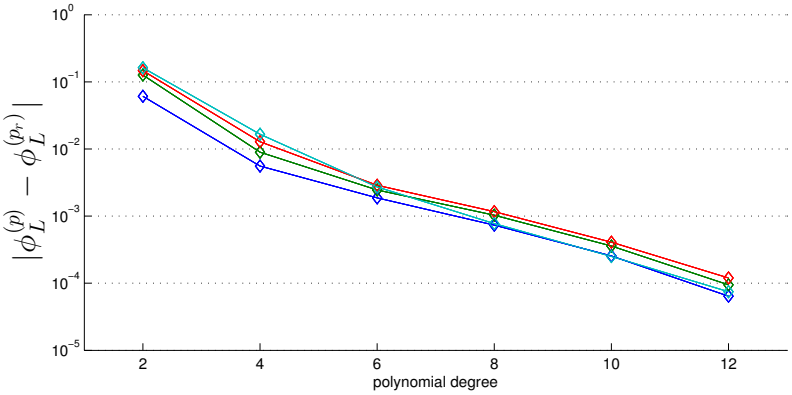

Figure 8: Convergence of the eigenvectors of the Kohn-Sham Hamiltonian for p-refinement.

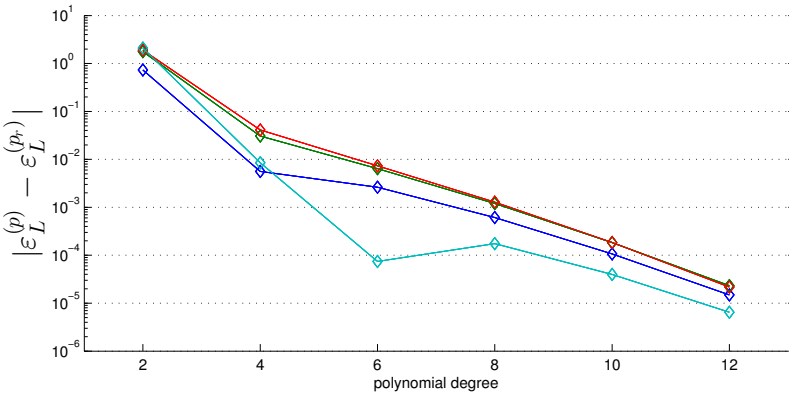

Figure 9: Convergence of the eigenvalues of the Kohn-Sham Hamiltonian for p-refinement.

8 Conclusion and outlook

We have presented a new mortar element method for full-potential density functional theory. Based on smoothness assumptions on the exchange-correlation potential, we derived an a priori error estimate for the linearized Kohn-Sham problem in the asymptotic setting. We derived a generic algorithm for the construction of mortar element spaces that was implemented in the hp-FEM library `concepts`. To study the Kohn-Sham problem in the self-consistent limit, furthermore, the optimal damping algorithm was implemented. A final numerical experiment on a mono-atomic system confirmed the a priori error estimate for the mortar element discretization by exponential convergence for both eigenvalues and eigenvectors.

There are several possible future directions not yet addressed in this framework. First and foremost, the fast convergence rate of our mortar discretization should be established on realistic physical systems, such as e.g. small molecules. The effort to achieve this in terms of implementation is limited, but was out of scope for this thesis. The generic mortar coupling framework will then also provide the interesting option to compare our non-conforming approach to other discretization schemes that can be reformulated in this setting, such as e.g. augmented plane waves.

However, fundamentally, the idea behind our discretization is not to compete with augmented plane waves. From the perspective of numerical analysis, we believe that our approach rather provides an enhancement for hp-finite element methods. In full-potential calculations, the efficiency of such discretizations lacks from the large number of degrees of freedom required to resolve the singularities at the nuclei. Here we expect our method to have a mild advantage while still providing fast convergence in a neighborhood of the cores. Simultaneously, we do not lose the strengths inherent to finite element methods, which are flexibility, stability and the ability to systematically generate convergence. In terms of the first point, we especially emphasize a posteriori error estimators that may use the extra-adaptivity provided by our mortar element approach to in order to further reduce the discretization error locally and shall, therefore, be investigated in a future work.

References

[1] L. H. Thomas. The calculation of atomic fields. *Mathematical Proceedings of the Cambridge Philosophical Society*, 23:542–548, 1927.

[2] Elliott H Lieb and Barry Simon. The Thomas-Fermi theory of atoms, molecules and solids. *Advances in Mathematics*, 23(1):22–116, 1977.

[3] P. Hohenberg and W. Kohn. Inhomogeneous electron gas. *Phys. Rev.*, 136:B864–B871, Nov 1964.

[4] W. Kohn and L. J. Sham. Self-consistent equations including exchange and correlation effects. *Phys. Rev.*, 140:A1133–A1138, 1965.

[5] Mel Levy. Universal variational functionals of electron densities, first-order density matrices, and natural spin-orbitals and solution of the V-representability problem. *Proceedings of the National Academy of Sciences*, 76(12):6062–6065, 1979.

[6] Elliott H. Lieb. Density functionals for Coulomb systems. *International Journal of Quantum Chemistry*, 24(3):243–277, 1983.

[7] Lars Nordstrom and David J Singh. *Planewaves, Pseudopotentials, and the LAPW method*. Springer, 2006.

[8] H. Chen and R. Schneider. Numerical analysis of augmented plane waves methods for full-potential electronic structure calculations. *Preprint 116, DFG-SPP 1324*, 2012.

[9] Christine Bernardi, Yvon Maday, and Anthony T Patera. Domain decomposition by the mortar element method. In *Asymptotic and numerical methods for partial differential equations with critical parameters*, pages 269–286. Springer, 1993.

[10] Eric Cancès. Self-consistent field algorithms for Kohn–Sham models with fractional occupation numbers. *The Journal of Chemical Physics*, 114(24):10616–10622, 2001.

[11] Philipp Frauenfelder and Christian Lage. Concepts - an object-oriented software package for partial differential equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 36(05):937–951, 2002.

[12] Concepts Development Team. *Webpage of Numerical C++ Library Concepts 2*. http://www.concepts.math.ethz.ch, 2013.

[13] Arnaud Anantharaman and Eric Cancès. Existence of minimizers for Kohn–Sham models in quantum chemistry. *Annales de l'Institut Henri Poincare (C) Nonlinear Analysis*, 26(6):2425–2455, 2009.

[14] John P Perdew and Yue Wang. Accurate and simple analytic representation of the electron-gas correlation energy. *Physical Review B*, 45(23):13244, 1992.

[15] C. LeBris. *Quelques problèmes mathématiques en chimie quanntique moléculaire*. PhD thesis, École Polytechnique, Paris, France, 1993.

[16] Eric Cancès, Rachida Chakir, and Yvon Maday. Numerical analysis of the planewave discretization of some orbital-free and Kohn-Sham models. *ESAIM: Mathematical Modelling and Numerical Analysis*, 46:341–388, 3 2012.

[17] F. Rellich. Ein Satz über mittlere Konvergenz. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse*, 1930:30–35, 1930.

[18] Heinz-Jürgen Flad, Reinhold Schneider, and Bert-Wolfgang Schulze. Asymptotic regularity of solutions to Hartree–Fock equations with Coulomb potential. *Mathematical Methods in the Applied Sciences*, 31(18):2172–2201, 2008.

[19] Benqi Guo. Approximation Theory for the P-Version of the Finite Element Method in Three Dimensions part II: Convergence of the P Version of the Finite Element Method. *SIAM Journal on Numerical Analysis*, 47(4):2578–2611, 2009.

[20] Nadine Große and Cornelia Schneider. Sobolev spaces on Riemannian manifolds with bounded geometry: General coordinates and traces. *Mathematische Nachrichten*, 286(16):1586–1613, 2013.

[21] B. Guo and I. Babuška. The h-p version of the finite element method. *Computational Mechanics*, 1(1):21–41, 1986.

[22] Benqi Guo and Jianming Zhang. Stable and compatible polynomial extensions in three dimensions and applications to the p and h-p finite element method. *SIAM Journal on Numerical Analysis*, 47(2):1195–1225, 2009.

[23] L. Ridgway Scott Susanne C. Brenner. *The Mathematical Theory of Finite Element Methods*, volume 15 of *Texts in Applied Mathematics*. Springer New York, 2008.

[24] Christine Bernardi and Yvon Maday. Spectral, spectral element and mortar element methods. In *Theory and Numerics of Differential Equations*, pages 1–57. Springer, 2001.

[25] John E. Osborn. Spectral approximation for compact operators. *Mathematics of Computation*, 29(131):712–725, 1975.

[26] Gene H Golub and Charles F van Van Loan. Matrix computations (Johns Hopkins studies in mathematical sciences). 1996.

[27] Richard B Lehoucq, Danny C Sorensen, and Chao Yang. *ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods*, volume 6. Siam, 1998.

[28] George Karniadakis and Spencer Sherwin. *Spectral/hp Element Methods for Computational Fluid Dynamics*. Oxford Scholarship, 2005.

[29] Allen Hatcher. Algebraic topology. *Cambridge UP, Cambridge*, 606, 2002.

[30] Philipp Frauenfelder. *hp-Finite element methods on anisotropically, locally refined meshes in three dimensions with stochastic data*. PhD thesis, Swiss Federal Institute of Technology, 2004.

[31] M. Frigo and S.G. Johnson. The Design and Implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231, Feb 2005.

[32] Martin J. Mohlenkamp. A fast transform for spherical harmonics. *Journal of Fourier Analysis and Applications*, 5(2-3):159–184, 1999.

[33] Nathanaël Schaeffer. Efficient spherical harmonic transforms aimed at pseudospectral numerical simulations. *Geochemistry, Geophysics, Geosystems*, 14(3):751–758, 2013.

[34] Barbara I Wohlmuth. A mortar finite element method using dual spaces for the lagrange multiplier. *SIAM Journal on Numerical Analysis*, 38(3):989–1012, 2000.

[35] C.C.J. Roothaan and PS Bagus. Atomic self-consistent field calculations by the expansion method. Technical report, DTIC Document, 1962.

[36] William J Gordon and Charles A Hall. Construction of curvilinear co-ordinate systems and applications to mesh generation. *International Journal for Numerical Methods in Engineering*, 7(4):461–477, 1973.

[37] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design patterns: elements of reusable object-oriented software*. Pearson Education, 1994.

[38] Francisco M Gomes and Danny C Sorensen. ARPACK++: A C++ implementation of ARPACK eigenvalue package. *CRPC, Rice University, Houston, TX, Tech. Rep. TR97729*, 1997.

[39] Timothy A. Davis. User Guide for CHOLMOD: a sparse Cholesky factorization and modification package, 2009.

[40] Gaël Guennebaud, Benoît Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010.