An approximation strategy on the Grassmann manifold to compute accurate SCF initial guesses for repeated calculations at different geometries

E. Polack, A. Mikhalev, Geneviève Dusson, B. Stamm, F. Lipparini

An approximation strategy on the Grassmann manifold to compute accurate SCF initial guesses for repeated calculations at different geometries

É. Polack[a] A. Mikhalev[b] G. Dusson[a] B. Stamm[b] and F. Lipparini[c]

[a]Laboratoire de Mathématiques de Besançon, UMR CNRS 6623, Université Bourgogne Franche-Comté, 16 route de Gray, 25030 Besançon, France
[b]Center for Computational Engineering Science, RWTH Aachen University, Schinkelstr. 2, 52062 Aachen, Germany
[c]Dipartimento di Chimica e Chimica Industriale, Univeristà di Pisa, Via G. Moruzzi 13, I-56124 Pisa, Italy

ARTICLE HISTORY
Compiled May 15, 2020

ABSTRACT
Repeated computations on the same molecular system, but with different geometries, are often performed in quantum chemistry, for instance, in ab-initio molecular dynamics simulations or geometry optimizations. While many efficient strategies exist to provide a good guess for the self-consistent field procedure, which is usually the main computational task to be performed, little is known on how to efficiently exploit in this direction the abundance of information generated during the many computations. In this article, we present a strategy to provide an accurate initial guess for the density matrix, expanded in a set of localized basis functions, within the self-consistent field iterations for parametrized Hartree-Fock problems where the nuclear coordinates are changed along a few user-specified collective variables, such as the molecule's normal modes. Our approach is based on an offline-stage where the Hartree-Fock eigenvalue problem is solved for some particular parameter values and an online-stage where the initial guess is computed very efficiently for *any* new parameter value. The method allows non-linear approximations of density matrices, which belong to a non-linear manifold that is isomorphic to the Grassmann manifold. The so-called Grassmann exponential and logarithm map the manifold onto the tangent space and thus provides the correct geometrical setting accounting for the manifold structure when working with subspaces rather than functions itself. Numerical tests on different amino acids show promising initial results.

KEYWORDS
Self-consistent field, density guess, ab-initio molecular dynamics, geometry optimization

1. Introduction

Computational quantum chemistry allows nowadays to describe, model and predict a very large variety of chemical phenomena. Thanks to a combination of new methods, computational techniques and hardware developments, quantum chemistry can be used to compute molecular structures, spectroscopic and response properties, reaction paths, aggregation properties and much more. A typical computational setup starts usually with the prediction, at a given level of theory, of the molecular geometry, which is obtained by minimizing the Born-Oppenheimer energy with respect to the nuclear coordinates [1]. Properties calculations are then carried out. For large molecules, as several stable conformers can exist, these operations may need to be repeated in order to account for the existence of multiple minima. The number of calculations required can be further increased if more complex systems are considered, for instance, a large biological polymer or a solvated molecule, as a correct statistical sampling of the system's configurations becomes mandatory in order to achieve correct results. In such cases, calculations can be performed on snapshots taken from classical or ab-initio molecular dynamics. As a

consequence, a computational study often requires to perform several calculations on the same system at different geometries.

One of the most common task performed during a quantum chemical calculation is the solution to the self-consistent field (SCF) equations, that is at the basis of Hartree-Fock [2] (HF) and Density Functional Theory [3] (DFT). The latter can often be the method of choice for the overall computational study, while the former is at least a necessary starting point for more refined post-HF treatments. The SCF equations are a set of coupled, non-linear differential equations that are solved iteratively. As such equations can exhibit notorious convergence problems [4], in the last years a number of different numerical techniques have been developed to achieve reliable and fast convergence. These new developments include not only convergence acceleration techniques, such as the popular Direct Inversion in the Iterative Subspace [5, 6] (DIIS) and its many extensions and generalizations [7, 8], but also methods to provide a better guess to the iterative procedure [9–15]. The latter point is of particular importance, as the SCF procedure can be particularly problematic when starting from an unrealistic guess and exhibit large oscillations and other pathological behaviors [14]. Thanks to all these recent developments, many existing SCF implementations manage to achieve convergence, at least for closed-shell systems, in as little as 15-20 iterations.

The guessing procedure developed in the years for SCF are usually focused on providing a good estimate of the electronic density for single point calculations. Much less has been done to specifically address the issue of repeated calculations, other than common-sense practices, such as using the density of a previous point as a guess for the next energy and forces evaluation in a geometry optimization and other related strategies. A notable and particularly successful exception, that directly aims at providing a better guess for the SCF procedure in ab-initio MD simulations (AIMD), is based on extended-Lagrangian techniques [16, 17], which introduce an auxiliary density that is propagated along the dynamics and used to provide a guess that is usually sufficiently good, so that, at the precision required by AIMD simulations, only a few SCF iterations are required per step. These techniques, that use the density, or guess density, at a collection of previous steps (usually, from a couple to about ten steps), successfully exploit this information to improve the guessing procedure. However, extended-Lagrangian techniques rely on the fact that the nuclei configurations at the various steps are produced by a deterministic process, such as MD, and are therefore not applicable to a general repeated calculation, as in geometry optimizations or QM/MM snapshots originating from uncorrelated frames extracted from a MD simulation.

In this work, we try to address the problem of forming a guess for repeated calculations that is as general and robust as possible. In particular, our aim is to develop a procedure that is able to reuse as much information as possible from previous calculations at different geometries, independent of their provenance, to provide an optimal guess for a further calculation. We assume that a set of atom-centered, localized basis functions, such as gaussian-type orbitals, is employed. The main idea can be stated as follows. Let us consider a set of configurations for which the SCF density is known and a further, new configuration for which we want to guess the density. A naive strategy would be to linearly interpolate the configurations, i.e., their Cartesian coordinates, for instance, and apply the same interpolation to the density matrices. However, there would be no guarantee that the density obtained with such a procedure would indeed be a density matrix, stemming from a monodeterminantal wavefunction. In order to enforce the correct properties of the new, approximated density, we adopt a geometric perspective. From a mathematical point of view, the density matrices live in a so-called Grassmann Manifold which, as it is not a vector space, does not allow for linear interpolation to be used. However, we will show how it is possible to map a point in such a manifold to its tangent space, which indeed is a vector space, perform the interpolation, or any kind of approximation there, and then go back to the manifold, ensuring that the interpolated density has all the properties that are required for it to be a genuine density matrix.

The techniques that we use are conceptually related to notions that are not new to chemists. Indeed, it is known that orbital rotations can be parametrized in terms of exponential maps, and that such maps can be used to parametrize the effect of orbital rotations on the density

matrix. This is commonly done for direct orbital optimization techniques, used for quadratically convergent SCF [18, 19] and multiconfigurational SCF implementations [20–25]. In this contribution, we use a different notion of exponential which allows to efficiently parametrize the set of density matrices. However, these exponentials are in practice very different, due to the structure difference between orbital rotations and density matrices. By applying geometrical techniques to the problem of repeated calculations, we will show how a very effective and rigorous SCF guessing procedure can be developed.

On the other hand, solving problems repeatedly for different parameter values is common in many engineering applications and can be put under the context of many-query computations. In such scenario, the concept of reduced order modelling for parametrized problems has been established and it has become a mature tool in computational engineering science. The roots of modern reduced order modelling lie in structural mechanics and an overview of literature, methods, concepts and applications can be found in the monograph [26]. The concept of reduced order modelling is only little known and exploited in computational chemistry. The few contributions in this field [27–30] involve methods based on finite elements, with only a limited amount of work having been done for Gaussian-type atomic orbitals. It can be noted that the numerical results in these papers deal with rather small molecules and do not contain any geometrical considerations as presented in this work. We hope that our further contribution shades a different angle at reduced order modelling for parametrized problems in electronic structure calculation.

In this preliminary study, we develop the methodology and apply the newly developed technique to a simple problem, where we assume that no level crossing occurs between the states due to geometry displacements. In particular, we generate one- and two-dimensional grids of molecular geometries by displacing the equilibrium geometry of a few chosen molecular systems along one resp. two different normal coordinates, using displacements of up to one atomic unit times the normalized coordinates. While this is a very simplified problem with respect to the general one, it provides an example of small, but non negligible oscillations of the geometry around an equilibrium point that are typical of MD simulations or of anharmonic force field calculations. We show that using a small number of data, we are able to predict the density at all other points with remarkable accuracy, providing an almost already converged density matrix.

This paper is organized as follows. In Section 2, we describe the addressed problem, namely the development of good initial guesses for the solution of the SCF problem parametrized with respect to the atomic positions, and we present the corresponding equations. We then present the methodology in Section 3, starting in Section 3.1 with the geometrical structure of the object of interest: the density matrix. We continue by describing the process of computing an approximation of the density matrix in Section 3.2, first in a case where the parameter dependency is one-dimensional, and second in the more complicated case of a multi-dimensional parameter space. In Section 4, we present some numerical results illustrating the accuracy of the initial guesses as well as the low computational cost obtained by this method. We close this article by pointing out some perspectives in Section 5.

2. Problem statement

While there exists a map between the geometry of a molecule and, for a given basis set, its SCF density matrix, such a map is unknown and certainly highly nonlinear. Finding the exact approximation of this map seems thus an impossible task. We have therefore to resort to some kind of approximation. The problem that we want to address in this article can be stated as follows. Suppose that a set of SCF computations has to be performed on the same molecule, or cluster of molecules, at different geometries, for instance, in a geometry optimization or molecular dynamics simulation. Suppose also that we allow the problem to be solved at some few specific geometries in order to access the density matrices for those points. The following question arises: how can the pre-computed density matrices be used to approximate the solution

at any new point, or to provide a very robust guess for the SCF at this new point?

To address this problem, a strategy needs to be developed in order to actually define geometries where we first compute the density matrix and then, in a second phase, use them to provide a guess and thus this task has the flavour of an interpolation or more generally an approximation problem. However, this is not an easy task due to the fact that the SCF density matrices are not elements of a vector space. This means that in general, a linear combination of two density matrices is not a density matrix. Therefore, the first goal of this paper is to find a strategy to perform an interpolation, or more generally an approximation, of the available densities in the appropriate set (manifold), so that the resulting density has all the properties that are needed.

A further point concerns the overall efficiency of this process, that strongly depends on how much data is needed to get a good approximation to the density. In other words, if we need to solve the SCF equations for a large number, say N_{g}, different geometries, we want to be able to provide a good guess based on pre-computed density matrices at $Q \ll N_{\mathrm{g}}$ points. Therefore, the second problem that we want to address is how can one find a *minimal* number of points that allow one to build a good density approximation at all other points.

Let us start by stating the first problem in a more precise way. We consider the electronic Schrödinger problem where the M nuclear positions $\boldsymbol{r} \in \mathbb{R}^{3M}$ are parametrized by a given, possibly non-linear, map $\mathbb{P} \ni \mathsf{p} \mapsto \psi(\mathsf{p}) = \boldsymbol{r} \in \mathbb{R}^{3M}$, where the map ψ may consist of reaction-coordinates, optimization steps, normal modes or any collective variable in general. We refer to the bounded domain $\mathbb{P} \subset \mathbb{R}^P$, for a given $P \in \mathbb{N}$, as the parameter domain. The parameter-dependency plays a key role in the methodology and we therefore highlight the dependency on p in the following with a subscript. We consider a level of theory that corresponds to the Hartree-Fock equations or Density Functional Theory (DFT) but without loss of generality we present in the following our approach for the Hartree-Fock (HF) method. Using a given basis set within the LCAO-framework (Linear Combination of Atomic Orbitals), the discrete energy can be written as

$$\mathcal{E}_{\mathsf{p}}(\widetilde{\mathsf{C}}) = \mathrm{Tr}\left(\widetilde{\mathsf{C}}^{\top}\mathsf{h}_{\mathsf{p}}\widetilde{\mathsf{C}} + \tfrac{1}{2}\widetilde{\mathsf{C}}^{\top}\mathsf{G}_{\mathsf{p}}(\widetilde{\mathsf{C}}\widetilde{\mathsf{C}}^{\mathrm{T}})\widetilde{\mathsf{C}}\right) \tag{1}$$

where h_{p} and G_{p} are the customary one and two electron integral matrices in the atomic orbitals (AO) basis for the parameter value p. The matrix $\widetilde{\mathsf{C}} \in \mathbb{R}^{N_{\mathsf{b}} \times N}$ contains the N_{b} coefficients of the N occupied molecular orbitals within the given N_{b}-dimensional basis. The SCF problem can be stated as the variational minimization of the SCF energy

$$\min_{\widetilde{\mathsf{C}} \in \mathcal{M}(\mathsf{p})} \mathcal{E}_{\mathsf{p}}(\widetilde{\mathsf{C}}), \tag{2}$$

where the coefficients $\widetilde{\mathsf{C}}$ need to satisfy the usual orthonormality constraints or, in other words, belong to the manifold $\mathcal{M}(\mathsf{p})$ defined as

$$\mathcal{M}(\mathsf{p}) = \left\{ \widetilde{\mathsf{C}} \in \mathbb{R}^{N_{\mathsf{b}} \times N} \;\middle|\; \widetilde{\mathsf{C}}^{\top}\mathsf{S}_{\mathsf{p}}\widetilde{\mathsf{C}} = 1_N \right\}, \tag{3}$$

with S_{p} denoting the overlap matrix. Writing the first-order optimality conditions, we obtain the following non-linear eigenvalue problem: Find a matrix $\widetilde{\mathsf{C}}_{\mathsf{p}} \in \mathcal{M}(\mathsf{p})$ and a diagonal matrix $\mathsf{E}_{\mathsf{p}} \in \mathbb{R}^{N \times N}$ containing the orbital energies $(\varepsilon_1, \ldots, \varepsilon_N)$ such that

$$\mathsf{F}_{\mathsf{p}}(\widetilde{\mathsf{D}}_{\mathsf{p}})\widetilde{\mathsf{C}}_{\mathsf{p}} = \mathsf{S}_{\mathsf{p}}\widetilde{\mathsf{C}}_{\mathsf{p}}\mathsf{E}_{\mathsf{p}}, \tag{4}$$

$$\widetilde{\mathsf{C}}_{\mathsf{p}}\widetilde{\mathsf{C}}_{\mathsf{p}}^{\top} = \widetilde{\mathsf{D}}_{\mathsf{p}}, \tag{5}$$

where $\mathsf{F}_{\mathsf{p}}(\widetilde{\mathsf{D}}) = \mathsf{h}_{\mathsf{p}} + \mathsf{G}_{\mathsf{p}}(\widetilde{\mathsf{D}})$ denotes the Fock operator and $\widetilde{\mathsf{D}}_{\mathsf{p}} \in \mathbb{R}^{N_{\mathsf{b}} \times N_{\mathsf{b}}}$ the density matrix. We note that the input data F_{p} and S_{p} depend explicitly on p whereas the solution $\widetilde{\mathsf{C}}_{\mathsf{p}}$ respectively $\widetilde{\mathsf{D}}_{\mathsf{p}}$

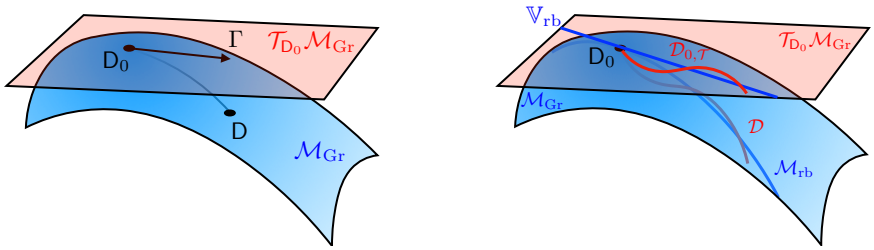


Figure 1. Schematic illustration of the geometrical setting. In both figures, we illustrate by the blue hypersurface the Grassmann manifold $\mathcal{M}_{\mathrm{Gr}}$ and by the red plane the tangent space $\mathcal{T}_{\mathsf{D}_0}\mathcal{M}_{\mathrm{Gr}}$ to $\mathcal{M}_{\mathrm{Gr}}$ at D_0. On the left, we illustrate the one-to-one relationship between a close density matrix $\mathsf{D} \in \mathcal{M}_{\mathrm{Gr}}$ and the corresponding vector $\Gamma = \mathrm{Log}_{\mathrm{Gr},0}\mathsf{D}$ in the tangent space. On the right, we further schematically illustrate the notions of $\mathcal{D}_{0,\mathcal{T}}$ and $\mathbb{V}_{\mathrm{rb}}$, respectively defined in (17) and (18), as well as their equivalent sets $\mathcal{D} = \{\mathsf{D}_\mathsf{p} | \mathsf{p} \in \mathbb{P}\}$ and $\mathcal{M}_{\mathrm{rb}} = \mathrm{Exp}_{\mathrm{Gr},0}(\mathbb{V}_{\mathrm{rb}})$ on $\mathcal{M}_{\mathrm{Gr}}$.

to the eigenvalue problem depend implicitly on p through the relations (4)–(5).

When the computation is done without any previous history (single step calculation as opposed to molecular dynamics, for example, where a predictor can be employed), an initial guess contains no a priori information on the solution and provides an error of order one. As already mentioned, the goal of this paper is to establish an approximation scheme to provide a good guess of the density matrix $\widetilde{\mathsf{D}}_\mathsf{p}$ when some known density matrices $\left\{\widetilde{\mathsf{D}}_{\mathsf{p}_i}\right\}_{i=1}^{Q}$ for some parameter values $\{\mathsf{p}_i\}_{i=1}^{Q}$ are given. Our strategy tackles the two main issues stated at the beginning of this section as follows.

First, in order to be able to perform an approximation based on known densities, we look at the problem from a geometrical point of view. The orthogonal projectors onto the space spanned by the N orbitals in the atomic orbital basis belong to the manifold

$$\mathsf{S}_\mathsf{p}^{\frac{1}{2}}\widetilde{\mathsf{D}}_\mathsf{p}\mathsf{S}_\mathsf{p}^{\frac{1}{2}} \in \mathcal{M}_{\mathrm{Gr}} = \left\{\mathsf{D} \in^{N_\mathrm{b}\times N_\mathrm{b}} \,\middle|\, \mathsf{D} = \mathsf{D}^\top, \mathsf{D}^2 = \mathsf{D}, \mathrm{Tr}(\mathsf{D}) = N\right\}, \tag{6}$$

is well known in mathematics under the name "Grassmann manifold". To be completely rigorous, the former is isomorphic to the latter, but we omit such technical details in the following. Here, using the properties of such manifold, we develop a strategy that maps the densities obtained at the various points to a vector space, namely the tangent space, performs the linear approximation there and then maps the interpolation back to the Grassmann manifold. From an intuitive point of view, the process can be seen as depicted in Figure 1 (left). The differentiable manifold can be thought of as a curve hypersurface (in blue) of lower dimension. We map the manifold to the hyperplane, which is tangent to the surface at a given point, and are projecting then all the data points (density matrices) to such plane and perform the approximation. Then, once the approximation is built, we use the inverse map and go back to the manifold. The key point here is that this guarantees to obtain a density matrix that satisfies all the physical requirements and we are therefore sure that such a matrix corresponds to a single Slater Determinant.

Second, we address the computational problem of making this scheme efficient. Indeed, if one computes the density matrix for a large number of molecular geometries, it is very likely that the information will be redundant. In this case, the corresponding density matrices can all be approximated by (different) linear combinations of very few common elementary matrices. In applied mathematics, those elements are called reduced basis of the parametrized problem since they build a basis of a vector space that approximates any density matrix to high accuracy. As elaborated above, the issue with this approach in our context is that any linear combination of density matrices is in general not a density matrix. However, we can apply this concept on the tangent space. Thus, after having mapped all density matrices to the tangent space, one can, for example, find a low dimensional basis by performing a singular value decomposition (SVD) of all tangent vectors. In consequence, any tangent vector can be represented with few degrees of freedom if expressed in this "reduced basis" on the tangent space. Mapping this

approximation back to the manifold of density matrices guarantees then that the approximation has the structure of a density matrix.

3. Methodology

3.1. The geometrical structure

We note that for any value of the parameter p, the matrix $\widetilde{\mathsf{C}}_\mathsf{p} \in \mathcal{M}(\mathsf{p})$, solution to (4)–(5), can be transformed and we define $\mathsf{C}_\mathsf{p} := \mathsf{S}_\mathsf{p}^{\frac{1}{2}} \widetilde{\mathsf{C}}_\mathsf{p}$ as is usual within the Löwdin orthonormalization. In consequence, we observe that C_p belongs to the Stiefel manifold of orthonormal N-frames in $\mathbb{R}^{N_\mathsf{b}}$. The corresponding density matrix $\mathsf{D}_\mathsf{p} = \mathsf{C}_\mathsf{p}\mathsf{C}_\mathsf{p}^\top$, belongs to the manifold of rank N projectors in $\mathbb{R}^{N_\mathsf{b}}$, already defined in eq. (6), which is isomorphic to the Grassmann manifold, hence designated with the same name. We will not insist on a very precise description of the setting in terms of differential geometry as this is not the purpose of this article. For interested readers we refer to [31, 32]. We will rather point out the practically important considerations, give some intuitive explanations and try to keep technical considerations to a minimum. We note that the energy $\mathcal{E}_\mathsf{p}$ defined in (1) is invariant under orthogonal transformation of the N-frames and we thus conclude that the solution of (4)–(5) is uniquely represented by D_p rather than C_p.

We are thus facing the situation where we are given the possibility to access the density matrix D_p for specific parameter values p, but we would like to keep those computations to a minimum. This will be done in the so-called *offline*-stage, where two tasks will be assigned. First, the choice of the points $\{\mathsf{p}_i\}_{i=1}^Q$ and second, the computation of the density matrix $\{\mathsf{D}_{\mathsf{p}_i}\}_{i=1}^Q$ at each of those points.

In the *online*-stage, we are then given parameter-solution pairs $\{\mathsf{p}_i, \mathsf{D}_{\mathsf{p}_i}\}_{i=1}^Q$ with $\mathsf{D}_{\mathsf{p}_i} \in \mathcal{M}_{\mathrm{Gr}}$ and we aim to approximate the mapping

$$\mathbb{P} \ni \mathsf{p} \mapsto \mathsf{D}_\mathsf{p} \in \mathcal{M}_{\mathrm{Gr}}. \tag{7}$$

Since the Grassmann manifold is not a vector space, it is obvious that a linear combination of density matrices does not belong to $\mathcal{M}_{\mathrm{Gr}}$ in general. In consequence, approximating $\mathcal{M}_{\mathrm{Gr}}$ with a vector space does not respect the geometric structure of the problem and some of the properties of $\mathcal{M}_{\mathrm{Gr}}$ would be lost in general.

For the Grassmann manifold, which is a differential manifold, for any given $\mathsf{D}_0 = \mathsf{C}_0\mathsf{C}_0^\top$ with $\mathsf{D}_0 := \mathsf{D}_{\mathsf{p}_0}$ and $\mathsf{C}_0 := \mathsf{C}_{\mathsf{p}_0}$ for fixed p_0, the tangent space is

$$\mathcal{T}_{\mathsf{D}_0}\mathcal{M}_{\mathrm{Gr}} = \left\{ \Gamma \in \mathbb{R}^{N_\mathsf{b} \times N} \,\middle|\, \mathsf{C}_0^\top \Gamma = 0 \right\} \subset \mathbb{R}^{N_\mathsf{b} \times N}. \tag{8}$$

Note that the tangent space is an affine space. One can then introduce the Grassmann exponential which maps tangent vectors on $\mathcal{T}_{\mathsf{D}_0}\mathcal{M}_{\mathrm{Gr}}$ to the manifold $\mathcal{M}_{\mathrm{Gr}}$ in a locally bijective manner around D_0. Indeed, it is not only an abstract tool from differential geometry, but it can be computed in practice involving the matrix exponential. By complementing C_0 with orthonormal columns to obtain $(\mathsf{C}_0, \mathsf{C}_\perp) \in O(N_\mathsf{b})$ and $\Gamma \in \mathcal{T}_{\mathsf{D}_0}\mathcal{M}_{\mathrm{Gr}}$ we have

$$\mathrm{Exp}_{\mathrm{Gr},0}(\Gamma) = \mathsf{C}\mathsf{C}^\top, \qquad \mathsf{C} = \left(\mathsf{C}_0, \mathsf{C}_\perp\right) \exp \begin{pmatrix} 0 & -\mathsf{B}^\top \\ \mathsf{B} & 0 \end{pmatrix} \mathsf{I}_{N_\mathsf{b},N}, \tag{9}$$

where the matrix $\mathsf{B} \in \mathbb{R}^{(N_\mathsf{b}-N)\times N}$ contains expansion coefficients of columns of Γ in a span of columns of $\mathsf{C}_\perp$ such that $\Gamma = \mathsf{C}_\perp \mathsf{B}$ and $\mathsf{I}_{N_\mathsf{b},N} = (\mathsf{I}_N, 0)^\top \in \mathbb{R}^{N_\mathsf{b}\times N}$ are the first N columns of the $N_\mathsf{b} \times N_\mathsf{b}$ identity matrix. As one can see, it is an exponential ansatz of a skew-symmetric matrix that leaves out any redundant parametrization (the zero diagonal blocks) due to the mixing of the virtual resp. occupied orbitals. In this manner the mapping between $\mathcal{T}_{\mathsf{D}_0}\mathcal{M}_{\mathrm{Gr}}$ and $\mathcal{M}_{\mathrm{Gr}}$

becomes locally bijective. Further, the Grassmann exponential can then be expressed by

$$\mathrm{Exp}_{\mathrm{Gr},0}(\Gamma) = \mathsf{C}\mathsf{C}^\top, \qquad \mathsf{C} = [\mathsf{C}_0\mathsf{V}_e \cos(\Sigma_e) + \mathsf{U}_e \sin(\Sigma_e)]\, \mathsf{V}_e^\top, \tag{10}$$

by means of a singular value decomposition (SVD) $\Gamma = \mathsf{U}_e\Sigma_e\mathsf{V}_e^\top$ of Γ. A schematic representation can be found in Figure 1 (left) and we refer to [31, 32] for further details and its derivation.

The inverse function is the so-called Grassmann logarithm $\mathrm{Log}_{\mathrm{Gr},0}$ (see, e.g., [31, 32]) which maps any $\mathsf{D} = \mathsf{C}\mathsf{C}^\top \in \mathcal{M}_{\mathrm{Gr}}$ in a neighborhood of D_0 to the tangent space $\mathcal{T}_{\mathsf{D}_0}\mathcal{M}_{\mathrm{Gr}}$ by

$$\mathrm{Log}_{\mathrm{Gr},0}(\mathsf{D}) = \mathsf{U}_\ell \arctan(\Sigma_\ell)\mathsf{V}_\ell^\top, \tag{11}$$

using the following SVD decomposition

$$\mathsf{U}_\ell\Sigma_\ell\mathsf{V}_\ell^\top = \mathsf{L} \qquad \text{with} \qquad \mathsf{L} = \mathsf{C}\left(\mathsf{C}_0^\top\mathsf{C}\right)^{-1} - \mathsf{C}_0. \tag{12}$$

Note that we respectively denote by $\mathsf{U}_\ell\Sigma_\ell\mathsf{V}_\ell^\top$ and $\mathsf{U}_e\Sigma_e\mathsf{V}_e^\top$ the thin Singular Value Decompositions (SVD) of L and Γ with the asymptotic cost of $\mathcal{O}(N_\mathsf{b}N^2)$, see e.g. [31, 32]. Such a cost is comparable with the cost of a traditional dense diagonalization, which is commonly used in SCF codes working with localized basis functions. We remark here that the diagonalization itself is seldom the rate-determining step for medium-large calculations, which are dominated by the cost of building the Fock matrix.

In this manner we map each density matrix $\mathsf{D}_{\mathsf{p}_i}$ to the tangent space at the reference point D_0 in order to obtain $\Gamma_i = \mathrm{Log}_{\mathrm{Gr},0}(\mathsf{D}_{\mathsf{p}_i})$. The reference point can in principle be chosen arbitrarily but it is the most intuitive to place it in the center of the parameter domain $\mathbb{P}$. Since the tangent space is a vector space we have now transformed our problem to a standard approximation problem of pairs of data (p_i, Γ_i) belonging to Euclidian vector spaces. In the next sections, we will precise how the map $\mathbb{P} \ni \mathsf{p} \mapsto \Gamma(\mathsf{p}) \in \mathcal{T}_{\mathsf{D}_0}\mathcal{M}_{\mathrm{Gr}}$ is approximated.

Before that, we summarize the global picture of our strategy: using the Grassmann logarithm allows us to map density matrices on the tangent space at a particular point of the manifold. Then we can rely on classical approximations techniques between the parameter domain and the tangent space being a vector space. Having the approximation defined on the tangent space, we use the Grassmann exponential to map back to the Grassmann manifold and thus can provide a density matrix obeying the exact geometrical structure of the problem, i.e. belonging to $\mathcal{M}_{\mathrm{Gr}}$.

3.2. Approximation of density matrices

The case of a one-dimensional parameter space provides a simple intuitive way to illustrate a first version of the approximation method using Lagrange interpolation. We proceed therefore in two steps, explaining first the one-dimensional case before extending the methodology to higher-dimensional parameter spaces.

3.2.1. One-dimensional parameter space

We predefine the offline-stage here in the sense that we choose $Q+1$ interpolation points p_i, $i = 0, \ldots, Q$, and compute the corresponding density matrices $\mathsf{D}_i = \mathsf{D}_{\mathsf{p}_i}$ at those points. We choose p_0 and consider the tangent space $\mathcal{T}_{\mathsf{D}_0}\mathcal{M}_{\mathrm{Gr}}$ as above. For the remaining Q points $\mathsf{p}_i \in \mathbb{P}$, we build the Lagrange basis functions $L_i : \mathbb{P} \subset \mathbb{R} \to \mathbb{R}$:

$$L_i(\mathsf{p}) = \frac{\prod_{j\neq i}(\mathsf{p} - \mathsf{p}_j)}{\prod_{j\neq i}(\mathsf{p}_i - \mathsf{p}_j)}. \tag{13}$$

In the online-stage, for any new $\mathsf{p} \in \mathbb{P}$ we build the following approximation, using $\Gamma_i =$

$\mathrm{Log}_{\mathrm{Gr},0}(\mathsf{D}_i)$,

$$\Gamma(\mathsf{p}) = \sum_{i=1}^{Q} L_i(\mathsf{p})\, \Gamma_i, \tag{14}$$

upon which we apply the Grassmann exponential to finally obtain the approximate density matrix

$$\mathsf{D}_{\mathrm{app}}(\mathsf{p}) = \mathrm{Exp}_{\mathrm{Gr},0}\left(\sum_{i=1}^{Q} L_i(\mathsf{p})\, \Gamma_i \right). \tag{15}$$

By construction, the interpolation property $\mathsf{D}_{\mathrm{app}}(\mathsf{p}_i) = \mathsf{D}_i$ is satisfied due to the property $L_i(\mathsf{p}_j) = \delta_{ij}$ of the Lagrange polynomials.

We note that when only two density matrices D_0 and D_1 are available, the application

$$\mathsf{D}_{\mathrm{app}}(\mathsf{p}) = \mathrm{Exp}_{\mathrm{Gr},0}\left(\frac{\mathsf{p} - \mathsf{p}_0}{\mathsf{p}_1 - \mathsf{p}_0} \Gamma_1 \right) \tag{16}$$

parametrizes the geodesic between D_0 and D_1 on $\mathcal{M}_{\mathrm{Gr}}$, as long as the exponential map is bijective, which is at least satisfied when p_0 and p_1 are close. This is the most natural way to define an approximation on $\mathcal{M}_{\mathrm{Gr}}$ for values $\mathsf{p} \in [\mathsf{p}_0, \mathsf{p}_1]$.

3.2.2. Multi-dimensional parameter space

We now extend our considerations to arbitrary dimensional parameter domains. The previous case of a one-dimensional parameter space suggests that accurate approximations of Γ can be obtained in the form of linear combinations of polynomials in p times known vectors Γ_i belonging to the tangent space.

We state now two remarks that seem appropriate at this point. First, a possible generalization of the approach to higher dimensions can be realized by tensor-products of the Lagrange-polynomials. This would, however, require an exponential increase (with respect to the dimension) of data-points p_i on a structured grid where the solutions D_i and Γ_i, respectively, are required to be known. A remedy can consist of the use of sparse grids on the parameter domain but we will propose in the following a more adaptive framework.

Second, the set of all $\Gamma_i = \mathrm{Log}_{\mathrm{Gr},0}(\mathsf{D}_i)$, $i = 1, \ldots, Q$, might be highly linearly dependent. In such cases, there exists a low-dimensional basis $\{\Theta_1, \ldots, \Theta_n\}$, with $n \ll Q$, such that the manifold

$$\mathcal{D}_{0,\mathcal{T}} := \{\Gamma(\mathsf{p}) = \mathrm{Log}_{\mathrm{Gr},0}(\mathsf{D}_{\mathsf{p}}) \,|\, \mathsf{p} \in \mathbb{P}\} \subset \mathcal{T}_{\mathsf{D}_0}\mathcal{M}_{\mathrm{Gr}}, \tag{17}$$

on $\mathcal{T}_{\mathsf{D}_0}\mathcal{M}_{\mathrm{Gr}}$ can be well-approximated by suitable elements of the n-dimensional space

$$\mathbb{V}_{\mathrm{rb}} = \mathrm{Span}\{\Theta_1, \ldots, \Theta_n\} \subset \mathcal{T}_{\mathsf{D}_0}\mathcal{M}_{\mathrm{Gr}}, \tag{18}$$

see Figure 1 (right) for a schematic illustration of the situation. The approximate density matrix $\mathsf{D}_{\mathrm{app}}(\mathsf{p})$ will be defined as $\mathsf{D}_{\mathrm{app}}(\mathsf{p}) := \mathrm{Exp}_{\mathrm{Gr},0}\left(\Gamma_{\mathrm{app}}(\mathsf{p})\right)$, with

$$\Gamma_{\mathrm{app}}(\mathsf{p}) = \sum_{i=1}^{n} L_i(\mathsf{p})\, \Theta_i, \tag{19}$$

where the functions $L_i : \mathbb{P} \to \mathbb{R}$ and the reduced basis $\{\Theta_1, \ldots, \Theta_n\}$ have to be appropriately chosen. We focus for now on the practical aspects of the method. A more theoretical approach is presented in Appendix A.

We start by choosing a rather large number N_{p} of parameters $\mathsf{p} \in \mathbb{P}$ (of the order 100 in our test cases), covering the parameter space $\mathbb{P}$ in a reasonable way. For example, one can take a uniform grid (as in our numerical tests) or (quasi-) random points on the parameter space $\mathbb{P}$. Then, the offline part can be summarized by the following two main steps.

First, d parameter points $\{\mathsf{q}_1, \dots, \mathsf{q}_d\}$ among the N_{p} points $\{\mathsf{p}_1, \dots, \mathsf{p}_{N_{\mathsf{p}}}\}$ are selected, for which the density matrices are computed and, as well, their Grassmann logarithms which we denote by $\{\Gamma(\mathsf{p}_1), \dots, \Gamma(\mathsf{p}_d)\}$. Second, a reduced basis $\{\Theta_1, \dots, \Theta_n\}$ with $n \leq d$ (hopefully $n \ll d$) of Grassmann logarithms is computed using a singular value decomposition (SVD), from which the functions $L_i(\mathsf{p})$ are also deduced.

More precisely, we first choose $d \in \mathbb{N}$ multivariate functions $\{P_1, \dots, P_d\}$ with $P_j : \mathbb{P} \to \mathbb{R}$ for j from 1 to d. For simplicity, we take all multivariate monomials on $\mathbb{P}$ of cumulative degree up to M with a total of d monomials. However, other choices for a basis are possible and do not change the substance of the method. We then assemble the matrix $\widetilde{P} \in \mathbb{R}^{N_{\mathsf{p}} \times d}$ containing the values of these functions at the parameters $\{\mathsf{p}_1, \dots, \mathsf{p}_{N_{\mathsf{p}}}\}$, i.e. $\widetilde{P}_{i,j} = P_j(\mathsf{p}_i)$.

The main idea is to minimize the error between the exact and approximate Grassmann logarithms on these N_{p} samples, i.e. solve

$$\min_{\Theta \in \mathbb{R}^{d \times (N_{\mathsf{b}} \cdot N)}} \|\Gamma_{\text{train}} - \widetilde{P}\Theta\|, \tag{20}$$

where $\Gamma_{\text{train}} \in \mathbb{R}^{N_{\mathsf{p}} \times (N_{\mathsf{b}} \cdot N)}$ contains as rows the $\Gamma(\mathsf{p}_i)$ reshaped in vectors, and where $\|\cdot\|$ is a suitable norm. An approximate solution to this problem is found by selecting a square submatrix of $\widetilde{P}$ using the so-called *maxvol* method as introduced in [33]. It finds a quasi-dominant square $d \times d$ submatrix denoted by $\widehat{P} \in \mathbb{R}^{d \times d}$ of $\widetilde{P}$ by selecting d samples $\{\mathsf{q}_i\}_{i=1}^d$. The approximate Θ is then written in the form

$$\Theta = \widehat{P}^{-1}\widehat{\Gamma}, \tag{21}$$

where $\widehat{\Gamma} \in \mathbb{R}^{d \times (N_{\mathsf{b}} \cdot N)}$ contains as rows the reshaped Grassmann logarithms $\Gamma(\mathsf{q}_i)$. A great feature of this method is that it requires only the computation of the density matrices for the selected parameters $\{\mathsf{q}_1, \dots, \mathsf{q}_d\}$ and not for all N_{p} parameters. At this stage, the Grassmann logarithm for a new parameter p can be computed via

$$\Gamma_{\text{app}}(\mathsf{p}) = \sum_{i=1}^{d} \left[P(\mathsf{p})\, \widehat{P}^{-1}\right]_i \Gamma(\mathsf{q}_i), \tag{22}$$

with $P(\mathsf{p}) = (P_1(\mathsf{p}), P_2(\mathsf{p}), \dots, P_d(\mathsf{p}))$.

The second part consists of further reducing the dimensionality by performing a SVD on the matrix $\widehat{\Gamma}$, noting that its rows can be highly linearly dependent. The SVD writes

$$\widehat{\Gamma} = \widehat{U}_n \widehat{S}_n \widehat{V}_n + \widehat{E}_n, \qquad \widehat{U}_n \in \mathbb{R}^{d \times n}, \quad \widehat{S}_n \in \mathbb{R}^{n \times n}, \quad \widehat{V}_n \in \mathbb{R}^{n \times (N_{\mathsf{b}} \cdot N)}, \tag{23}$$

where $\widehat{E}_n$ is the remaining error term due to truncation. The truncation order n is determined based on a user-specified error tolerance ε by requiring $\sigma_{n+1}(\widehat{\Gamma}) < \varepsilon \sigma_1(\widehat{\Gamma})$, where $\sigma_i(\widehat{\Gamma})$ denotes the i-th singular value of $\widehat{\Gamma}$. We denote by $(\Theta_1, \dots, \Theta_n)$ the rows of the matrix $\widehat{V}_n$ reshaped into matrices of size $N_{\mathsf{b}} \times N$. Substituting the truncated SVD into (22) leads to

$$\Gamma_{\text{app}}(\mathsf{p}) = \sum_{i=1}^{n} \left[P(\mathsf{p})\, Z\right]_i \Theta_i, \tag{24}$$

where $Z = \widehat{P}^{-1} U_n S_n \in \mathbb{R}^{d \times n}$ and Θ_i can be precomputed offline. Thus, the online stage consists of building, for any new parameter $\mathsf{p} \in \mathbb{P}$, the matrix $P(\mathsf{p})$, building $\Gamma_{\text{app}}(\mathsf{p})$ according to (24)

Algorithm 1: Offline stage

Data: Domain $\mathbb{P}$ of parameter p; d multivariate monomials $\{P_i\}_{i=1}^{d}$; relative truncation threshold ε for the SVD.

Result: A reduced basis $(\Theta_1, \ldots, \Theta_n)$ along with its size n and a $d \times n$ matrix Z, that define approximation in (24).

Total complexity: $O((N_\mathsf{p} + N_\mathsf{b}N)d^2 + N_\mathsf{b}^\beta d)$ if $N_\mathsf{p} > d > n$ and $N_\mathsf{b}N \gg d$.

1 Define a uniform grid of N_p points $\mathsf{p}_j \in \mathbb{P}$ such that $N_\mathsf{p} \geq d$. *This requires* $O(N_\mathsf{p})$ *operations.*

2 Compute the matrix $\widetilde{P} \in \mathbb{R}^{N_\mathsf{p} \times d}$ given by $\widetilde{P}_{i,j} = P_j(\mathsf{p}_i)$. *Since all* $P_j(\mathsf{p})$ *are monomials, complexity is* $O(N_\mathsf{p} d)$.

3 Apply the *maxvol* method to the matrix $\widetilde{P}$ to obtain d indices of rows $\{\mathsf{piv}(i)\}_{i=1}^{n}$ and compute the corresponding submatrix $\widehat{P}$. *The number of operations is* $O(N_\mathsf{p} d^2)$.

4 Define the set $\{\mathsf{q}_i\}_{i=1}^{d}$ such that $\mathsf{q}_i = \mathsf{p}_{\mathsf{piv}(i)}$. *The operation count is* $O(d)$.

5 For each q_i define $\widehat{\Gamma}_{i,:}$ by reshaping the computed value of $\Gamma(\mathsf{q}_i)$ into a $N_\mathsf{b} \cdot N$ row vector. *The complexity is* $O(N_\mathsf{b}^\beta d)$, *where* β *depends on the eigenvalue solver.*

6 Compute the SVD of the matrix $\widehat{\Gamma} \in \mathbb{R}^{d \times (N_\mathsf{b} \cdot N)}$, truncate it to the rank-n approximation $U_n S_n V_n$ such that $\sigma_{n+1}(\widehat{\Gamma}) < \varepsilon \sigma_1(\widehat{\Gamma})$. *This step is done in* $O(N_\mathsf{b} N d^2)$ *operations.*

7 Reshape each row of $n \times (N_\mathsf{b} \cdot N)$ factor V_n into a corresponding $N_\mathsf{b} \times N$-matrix Θ_i. *No need to perform any operations, since reshape does not require any actions.*

8 Output $d \times n$ matrix as the product $\widehat{P}^{-1} U_n S_n$. *Inverting, multiplying and diagonal scaling in* $O(d^3 + nd^2 + nd)$ *operations.*

9 Output the reduced basis $\{\Theta_1, \ldots, \Theta_n\}$.

and finally computing the Grassmann exponential thereof in order to obtain the approximate density matrix $\mathsf{D}_{\text{app}}(\mathsf{p})$.

The algorithms presenting the computations done in the offline and online stages are described in Algorithm 1 and 2, together with the complexity of their different operations. Note that the most time-consuming step in the online calculation is the application of the Grassmann exponential.

3.3. Summary of the method

To summarize, the proposed approach returns an approximate density matrix $\mathsf{D}_{\mathsf{app}}(\mathsf{p})$ of $\mathsf{D}(\mathsf{p})$ at any given point p inside the parameter domain $\mathbb{P}$. This density matrix $\mathsf{D}_{\mathsf{app}}(\mathsf{p})$ is then used as an initial guess for the SCF solver. The goal is to reduce the number of required SCF iterations. The starting guess is found with the two following steps:

(1) Offline, precomputations: define points in $\mathbb{P}$ where the exact density matrix and functionals thereof are computed.
(2) Online, runtime computations: use the precomputed density matrices and functionals to reconstruct an approximate density matrix $\mathsf{D}_{\mathsf{app}}(\mathsf{p})$ at any parameter point $\mathsf{p} \in \mathbb{P}$.

The above mentioned steps are different for one-dimensional and multi-dimensional cases. In the case of a one-dimensional domain $\mathbb{P}$, the data points are chosen in a greedy hierarchical manner, as described in [34]. Then, a Lagrange interpolation is built upon these points. In the multi-dimensional case, we use Algorithm 1, performed offline, to obtain both the points and the data. Then, for any given value of $\mathsf{p} \in \mathbb{P}$ we use Algorithm 2 to compute an initial guess.

Algorithm 2: Online stage

Data: A point $\mathsf{p} \in \mathbb{P}$; d multivariate monomials $\{P_i\}_{i=1}^d$; the reduced basis $\{\Theta_1, \dots, \Theta_n\}$; the matrix $Z \in \mathbb{R}^{d \times n}$ appearing in equation (24)

Result: The approximate value of $\mathsf{D}(\mathsf{p})$

Total complexity: $O(N_\mathsf{b} N(n+N))$ if $N_\mathsf{b} N \gg d$.

1 Compute the vector $P(\mathsf{p}) = (P_1(\mathsf{p}), \dots, P_d(\mathsf{p}))$ at the new point p. *The number of operations is $O(d)$.*

2 Compute the vector of scalars $(L_1(\mathsf{p}), \dots, L_n(\mathsf{p})) = P(\mathsf{p})Z$. *This multiplication is done with $O(nd)$ operations.*

3 Compute the matrix $\Gamma_{\mathsf{app}}(\mathsf{p}) = \sum_{i=1}^n L_i(\mathsf{p})\Theta_i$. *Summation with $O(nN_\mathsf{b}N)$ operations.*

4 Apply Grassmann exponential: $\mathsf{D}_{\mathsf{app}}(\mathsf{p}) = \mathrm{Exp}_{\mathrm{Gr},0}(\Gamma_{\mathsf{app}}(\mathsf{p}))$ *Complexity of this step is mostly defined by the SVD leading to $O(N_\mathsf{b}N^2)$ operations*

4. Numerical results

To demonstrate the method's accuracy and robustness, we illustrate it on four different small- to medium-sized molecules, namely, the amino acids alanine, asparagine, phenylalanine, and tryptophan (13, 17, 23, and 27 atoms, respectively). If not explicitly stated otherwise, all the SCF calculations in the following have been performed using the CFOUR [35] suite of program, employing Dunning's cc-pVDZ basis set [36]. The SCF program was modified so that a guess density matrix, obtained with the newly developed method, could be provided as an input. The default convergence criterion was used for all the calculation: 10^{-7} for the root-mean-square (RMS) change of the density and 10^{-6} for the maximum change. The algorithm developed to generate the guess density, presented in Section 3.2 has been implemented in Julia [37]. The program works with input densities which are generated by CFOUR, and writes as output the computed guess density matrix in a file, that can be read by CFOUR.

In order to generate displaced geometries, normal modes are computed for the molecules using analytical second derivatives. For each molecule, we choose two different normal modes, one corresponding to the carbonyl C-O stretching, the second to a low-frequency collective vibration. All the starting structures, including the normal modes used to generate displacement geometries, are reported in the supporting information.

As parameter values p, we consider the coefficients corresponding to each normal mode, i.e. the nuclear coordinates are constructed by

$$\boldsymbol{r} = \boldsymbol{r}_0 + \sum_{i=1}^{P} \mathsf{p}_i \boldsymbol{n}_i, \tag{25}$$

where $\boldsymbol{r}_0$ denotes the equilibrium geometry, p_i the i-th component of the parameter p and $\boldsymbol{n}_i$ the i-th normal mode. For one-dimensional parameter domains, we consider thus one normal mode (the one reported first in the supporting information) whereas for two-dimensional domains, we consider both normal modes. The parameters p_i are chosen in the range $[-1, 1]$ bohr and discretized using an 11 points grid, i.e., we displace the geometries of $-1, -0.8, \dots$ times the normal coordinate. For alanine, we repeat the calculations taking the larger parameters domain $[-10, 10]$ bohr, still using an 11 points grid. The latter example is denoted by "Alanine*". The grids for two-dimensional domains are formed by a tensor product of the one-dimensional grids. For any given parameter p, the corresponding molecular geometry can then be generated and used for a SCF calculation.

In the following, we provide several numerical tests. We illustrate how we can provide accurate initial density matrices for one-dimensional and two-dimensional parameter spaces. To assess the quality of the guess, we report the number of SCF iterations required to achieve convergence and we compare it with the number of iterations required starting from a guess

Table 1. Number of SCF iterations required to achieve convergence (max change in the density smaller than 10^{-6}) using different initial guesses. As the computations were carried out using different packages, that offer different SCF implementations, this cannot be considered an accurate comparison between the various guesses, but only a qualitative estimate of the number of required iterations. Core: diagonalization of the core Hamitlonian (with CFOUR). Harris: diagonalization of the Harris functional (Gaussian 16). Hückel: using the extended Hückel method (PySCF). MinAO: start from a SCF calculation using a minimal AO basis set, which is then projected onto the chosen basis (PySCF). SAD: superposition of atomic densities (PySCF).

	Alanine	Asparagine	Phenylalanine	Tryptophan
Core	21	21	23	26
Harris	13	14	14	15
Hückel	16	17	17	18
MinAO	15	17	17	17
SAD	16	17	17	17

obtained by diagonalizing the core Hamiltonian, which is the default guess in CFOUR. For 1D grids, we use the method presented in Section 3.2.1 whereas for examples involving a two-dimensional parameter space, we use the general algorithm reported in Section 3.2.2. Before proceeding with the numerical tests, we report in Table 1 the number of SCF iterations needed to converge the Hartree-Fock equations, using different guess procedures, at the equilibrium geometry of the various test systems. The calculations were performed with different softwares, namely, CFOUR, Gaussian 16 [38] and PySCF 1.7 [39] and are therefore not directly comparable. However, they provide a qualitative estimate of the number of SCF iterations one can expect for such calculations and thus a benchmark for our algorithm. As convergence criteria are different in the various codes, we consider the SCF converged when the maximum variation of the density matrix between two subsequent iterations is smaller than 10^{-6}, as this information is reported in all codes used for the various calculations.

4.1. One-dimensional parameter domains

For this first batch of tests with $P = 1$, we compute an approximation of the density matrix using the method presented in Section 3.2.1 for every point in the parameter space (i.e., for each displaced geometry) and use it as a starting guess for a SCF calculation in CFOUR. We repeat such computations varying the order of interpolation, i.e., the number of precomputed densities used to build the guess. In order to select the interpolation points, we select them with a hierarchical greedy algorithm that chooses as next point the parameter value where the current approximation is worst, sometimes also referred to the *magic points* (see [34]). In this simple one-dimensional case, we consider the left-most, thus the smallest, parameter value as the root to build the tangent space. We observe numerically that all the results are independent on the choice of the root to build the tangent space, which is not obvious from the formulae.

The results obtained using our guessing procedure for the four amino acids selected as test molecules are reported in Figure 2. In the left panel, we show the maximal number of SCF iterations required to achieve convergence over all the points in the test grids. In the right panel, the accuracy of the guess with respect to the converged SCF density is also reported. The tests confirm the good accuracy of our guess, as using a Lagrange polynomial interpolation of degree 5 manages to reduce the number of required SCF iterations to only a few, namely, 3 for asparagine, 2 for alanine, and to 1 for phenylalanine and tryptophan. The latter result is particularly noteworthy as it demonstrates that, for these two systems, our guessing procedure can produce a guess density which is essentially already at convergence, as it can also be seen by looking at the norm of the error in the right panel. This makes in turn the overall SCF procedure unnecessary. For the other two molecules, convergence is achieved in 2 or 3 iterations, which is still a remarkable gain with respect to the standard procedure, that always requires at least 13

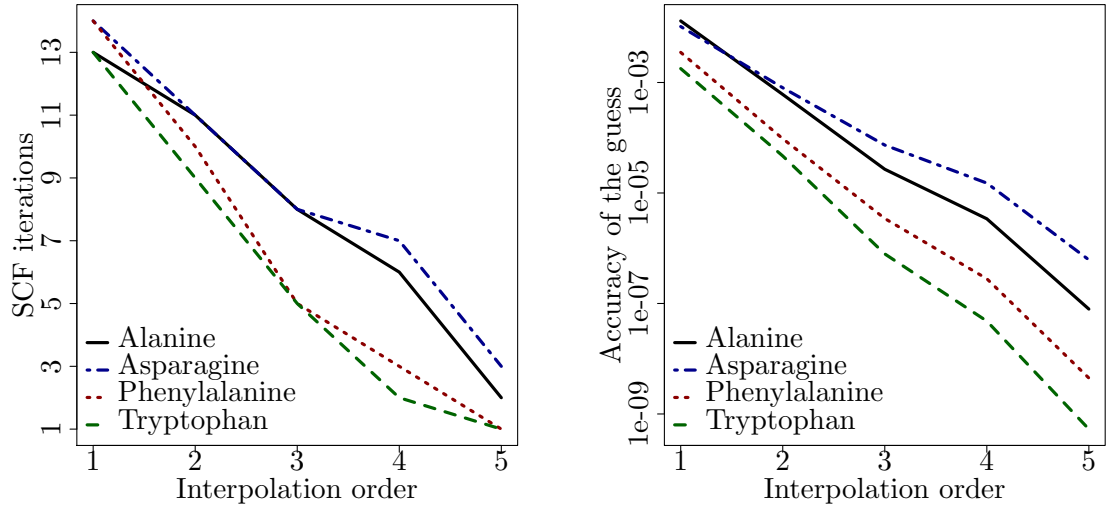


Figure 2. Results for the 1D parameter space. Number of SCF iterations required to achieve convergence (left panel) and Frobenius norm error on the density guess (right panel) as a function of the interpolation order for the various test systems. All the calculations were performed with CFOUR using the following convergence criteria for the increment of the density ΔP: RMS $\Delta P < 10^{-7}$ and max $|\Delta P| < 10^{-6}$.

iterations.

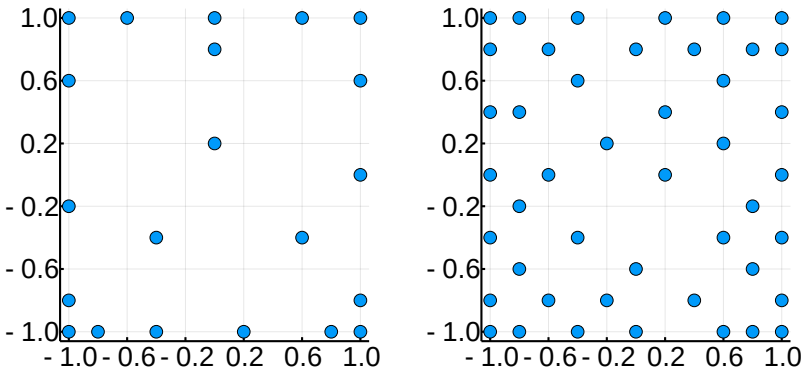
We point out that while we are considering small perturbations to the equilibrium geometry, these are not negligible. The SCF energy along the grid points varies of about 1.5-2.0 kcal/mol, which is a small, but significant oscillation if compared with the thermal energy at room temperature. In the following examples, we will explore larger energy fluctuations in order to assess the robustness of the method.

An interesting comparison can be made here with a common-practice strategy to provide a good guess for SCF calculations at similar geometries, i.e., using the converged SCF density as a guess for a calculation at a close geometry. We proceed as follows. We compute a fully converged SCF solution at the first gridpoint and then we advance along the 1D grid using each time the SCF density of the previous point as a guess. Considering the 10 points for which a guess density was available, the SCF converged on average in 10 iterations for tryptophan, 11 iterations for alanine and asparagine, and 12 iterations for phenylalanine. While these numbers are, as it could be easily expected, an improvement with respect to the ones reported in Table 1, it is apparent how our algorithm outperforms this strategy. We also repeated the calculation for alanine on the coarser grid, i.e., using displacements of 1 bohr along the normal coordinate. In this case, 13-14 iterations were needed to achieve convergence, which is close to what reported in Table 1, meaning that the geometry change considered for this example is already more than enough to produce sizeable changes in the density matrix and hindering thus the efficiency of a simple strategy such as using the density at the closest available geometry.

4.2. Two-dimensional parameter domains

We now present similar tests for the case where the parameter domain is two-dimensional, i.e., we allow the displacement of the atoms in the molecules in two normal directions ($P = 2$). The initial guess density matrix is computed with the method presented in Section 3.2.2, using a maximum cumulative degree of the monomials taken to $M = 8$ with a corresponding number of monomials $d = 45$. This ensures in the following numerical tests that the tolerances obtained in equation (23) are reasonably small.

For the two-dimensional grid used here, we generate a uniform 11×11 test-grid consisting of 121 points, i.e., displaced geometries. Note that in the offline part, the required SCF computa-



(a) $M = 5$ (b) $M = 8$

Figure 3. Maxvol-selected points for 2D case for different maximum cumulative degree M.

tions are only those of the selected parameters in the maxvol method presented in Section 3.2.2, i.e. only $d = 45$ calculations. Figure 3 shows the actual points selected by the maxvol-algorithm, for maximum cumulative degrees M equal to 5 and 8 respectively. The converged density matrices at the selected points are then used to build the reduced basis, the size of which is reported in the following. For the four considered molecules, using the $[-1, 1]$ parameter range, the SCF energy exhibits much larger fluctuations than the ones observed for the $P = 1$ examples. In particular, the energy fluctuates of 9.1, 8.9, 8.5, and 7.6 kcal/mol for alanine, asparagine, phenylalanine, and tryptophan, respectively. These are large energy fluctuations for a single molecule if compared, for instance, with the thermal energy at room temperature, and are likely not to be encountered when performing a molecular dynamics simulation. As in the 1D case, we chose the lower-left parameter value as the root to build the tangent space, and we observe numerically that the results are independent of the choice of the root to build the tangent space.

In Figure 4 (left panel) we report the maximum number of SCF iterations required to achieve convergence over the test grid of parameter values as a function of the size of the reduced basis used to build the approximation. In the right panel, the error of the computed guess with respect to the converged SCF density is reported.

These results show that, despite the sizeable fluctuations in the energy, our procedure is always able to reconstruct a guess density that is at convergence for every displaced geometry using no more than 17 basis vectors, with 13 being enough to obtain the same result in the best-case scenario (tryptophan). The convergence of the error in the density with respect to the number of basis vectors (right panel) is fast and smooth, which confirms the excellent performances of our procedure.

A computational remark is, at this point, mandatory. The guess procedure presented in Section 3 consists of two separate parts, named offline and online stages, respectively. In the offline part, the reduced basis is assembled. This is of course the expensive part of the procedure, as in order to compute the reduced basis, we need to solve the SCF problem at a given number of points, depending on the required accuracy. The online state is, on the other hand, completely inexpensive and can be performed in a fraction of a second for all the examples reported in this work. The key idea beyond the separation of the procedure in two different stages is that the offline one can be performed once and for all: as soon as the reduced basis is available, only the online stage has to be performed. In practice, this means that if we were to repeat our test calculations with a much finer grid, we would get a guess density for all the points using the reduced basis already assembled, and therefore at a cost that is completely negligible with respect to that of performing even a single SCF iteration.

In order to test the robustness of our procedure, we repeated the calculations on alanine using a two-dimensional grid, this time with a parameter domain of $[-10, 10]$. This grid encompasses

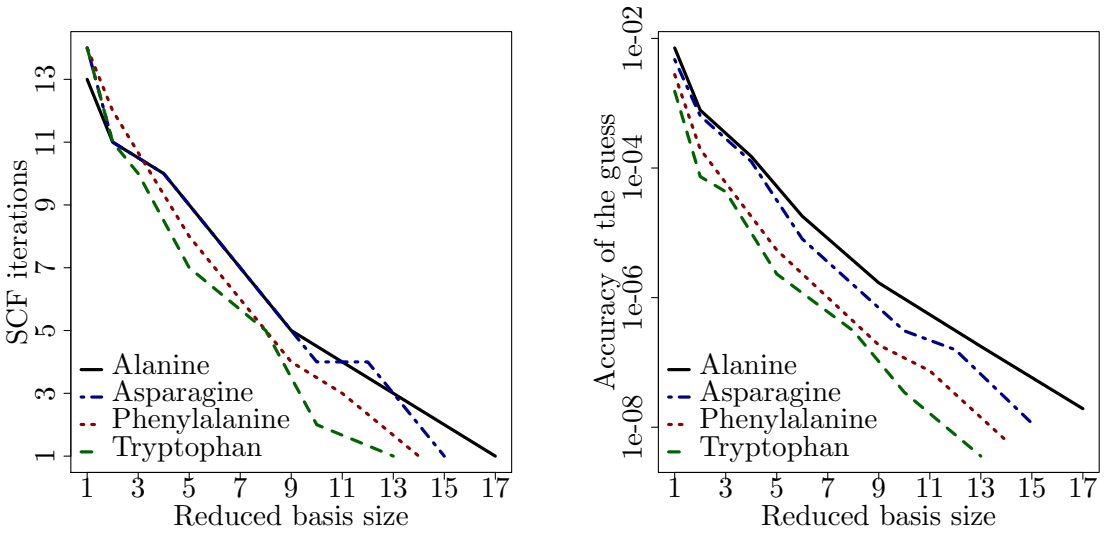


Figure 4. Results for the 2D parameter space. Number of SCF iterations required to achieve convergence (left panel) and Frobenius norm error on the density guess (right panel) as a function of the interpolation order for the various test systems. All the calculations were performed with CFOUR using the following convergence criteria for the increment of the density ΔP: RMS $\Delta P < 10^{-7}$ and max $|\Delta P| < 10^{-6}$.

large geometry variations, with the SCF energy varying in a range of more than 1000 kcal/mol, and provides a test for our algorithm in more extreme conditions. In Figure 5 we report, in the left panel, the results obtained for this case using the same setup used for the other 2D examples. The number of SCF iterations is reported on the right axis, while the error is on the left. For comparison, the results for the same molecule and the previous grid are always reported. We can immediately see how our guess procedure is now struggling to provide an accurate guess. Increasing the size of the reduced basis, we observe that the accuracy is stagnating, so that there is no gain by further increasing it. In order to better understand the source of this behavior, we allow the maximum cumulative degree of the monomials used in the algorithm to grow up to 14. The results are reported in the right panel of Figure 5. The guess density error and the number of SCF iterations exhibit now a convergent behavior, with as little as 5 iterations needed to converge the SCF in the worse case scenario when using the largest reduced basis. However, the size of the reduced basis required to observe a large reduction of the number of SCF iterations is much larger than what was observed before. We stress however that this is an extreme test case, and that we compute the SCF at geometries that are always quite distant from each other and it is hard to imagine a similar situation in a real-life application. However, the reduced basis built for this example allows one to explore a much larger portion of the potential energy surface of alanine than before, so that a larger number of vectors in the reduced basis appears justified. We stress that, even though the reduced basis is much larger than in the other examples, the online stage of the algorithm can still be performed in a negligible amount of time (less than 1 second).

Finally, in order to check the method when a larger basis set is used, we repeated the calculations, once again chosing alanine and employing the fine 2-dimensional grid, using the augmented, triple zeta Dunning's basis set aug-cc-pVTZ. These sets of results are labeled "Alanine+" and reported in Figure 6, where they are compared vis-a-vis with the results obtained with the smaller cc-pVDZ basis set. As it can be seen from the figure, the use of a larger basis set has virtually no influence on our algorithm. This result is not surprising, as the methodology applies in principle to the non-discretized problem as well, i.e., for complete basis sets.

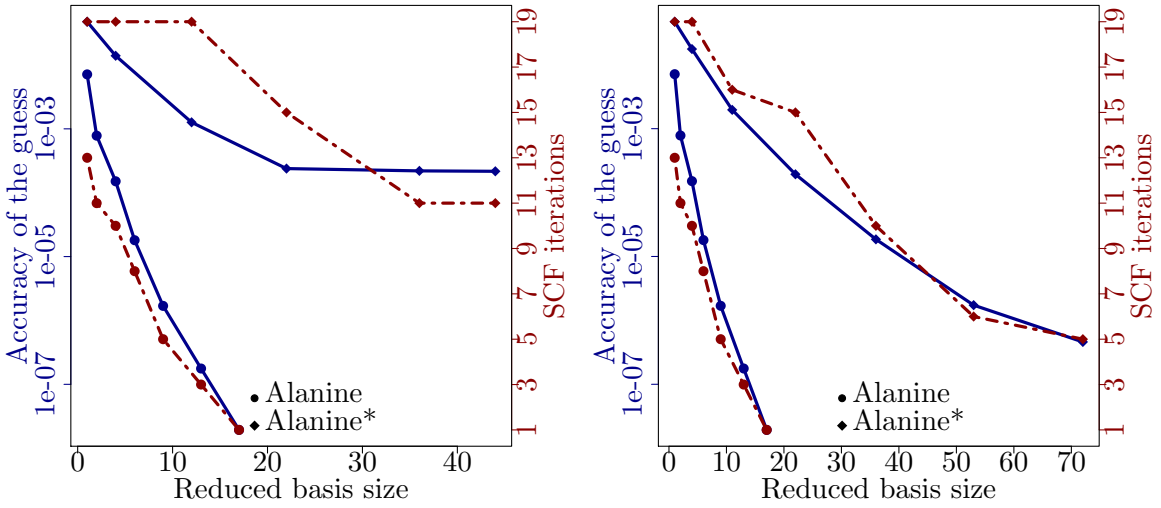


Figure 5. Comparison of alanine and alanine* in the 2D parameter space. Number of SCF iterations required to achieve convergence and Frobenius norm error on the density guess as a function of the interpolation order for $M = 8$ for both systems (left panel) and $M = 8$ for alanine and M=14 for alanine* (right panel). All the calculations were performed with CFOUR using the following convergence criteria for the increment of the density ΔP: RMS $\Delta P < 10^{-7}$ and max $|\Delta P| < 10^{-6}$.

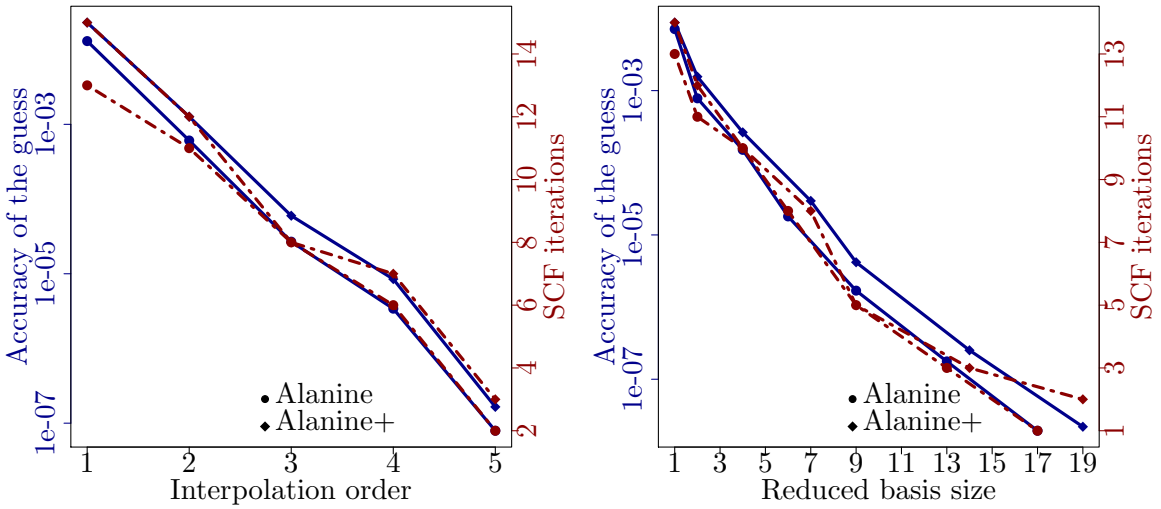


Figure 6. Comparison of alanine and alanine+. Number of SCF iterations required to achieve convergence and Frobenius norm error on the density guess as a function of the interpolation order for the 1D parameter space (left panel) and the 2D parameter space (right panel). All the calculations were performed with CFOUR using the following convergence criteria for the increment of the density ΔP: RMS $\Delta P < 10^{-7}$ and max $|\Delta P| < 10^{-6}$.

5. Perspectives

In this contribution, we presented a new method to compute a guess density matrix for the self-consistent field procedure at a given molecular geometry exploiting in an efficient way results available for other molecular geometries. The method is robust and is able to efficiently reconstruct a very good approximation to the SCF density at the a new geometry, often to the point that the SCF procedure itself becomes unnecessary. The proposed algorithm is divided in two different steps. In a first, offline phase, the building blocks for the approximation of the density are computed. This phase thus encompasses all the most expensive steps in the calculation, including solving the SCF problem at a number of geometries, which are chosen using a greedy strategy that attempts to add, at every new point, the most relevant information to improve the basis. Once this first stage is completed, the online phase comes into play. Starting with the results of the offline stage, the approximated density is built for any new molecular geometry. The cost of this second phase is negligible and the computational investment of the offline phase can be harvested in a many-query context where the online phase is used many times or, in other words, the effort done to assemble the reduced basis pays off when many other computations need to be performed, as a very good guess can be assembled for all such computations at a very little cost.

In this first work, we tested the algorithm on a few selected medium-sized molecules, namely, the amino acids alanine, asparagine, phenylalanine and tryptophan. In order to create displaced geometries, we computed normal modes and chose two particular vibrations, namely, the carbonyl stretching and a low-frequency collective mode, and used such coordinates to create either one- or two-dimensional grids, displacing the equilibrium geometry of 5 uniform increments per direction per dimension, generating thus 11 and 121 geometries, respectively, for the 1D and 2D cases. We tested our method both with displacements compatible with steps used in finite difference calculations of energy gradients, for which we observed variations in the SCF energy of about 7-9 kcal/mol, and also for much larger displacements, that gave rise to a range of SCF energies spreading well over 1000 kcal/mol. In both cases and for all molecules, the algorithm showed very good performances, generating a guess able to reduce the number of SCF iterations required to achieve convergence to only a few, if any was needed at all.

The main limitation of our strategy is that, at the moment, it was tested and applied only to low dimensional problems (in parameter domain) - as these are the only ones for which it is possible to generate uniform grids and compute reference data at each point with reasonable computational resources. The next natural stage is to test the algorithm on a more general set of data for a high-dimensional parameter domain and develop a strategy to handle the creation of a reduced basis when there is no simple connection between the different geometries. That would be the case if the geometries were generated randomly or with molecular dynamics. The latter application is of course particularly interesting. However, further understanding of the theory is still required and a new strategy to assemble the reduced basis on-the-fly has to be developed in order to circumvent the so far artificial offline-online decomposition.

Acknowledgements

This paper is dedicated to prof. Jürgen Gauss in honor of his sixtieth birthday. FL would like to express his deepest gratitude to prof. Gauss, not only for being an exceptional, dedicated mentor, who deeply cares for his pupils both from a scientific and a human point of view, but also for all the things done together, that range from going to the opera, to skiing, to enjoying a nice dinner and a good bottle of wine. Having the chance of working in Jürgen's group has made a big difference not only for my scientific growth, but also for my personal one.

This work has been created based on an interdisciplinary collaboration between theoretical chemistry and applied mathematics. The mathematics community had the chance to meet and interact with prof. Gauss in interdisciplinary workshops, such as the Oberwolfach workshop on "Mathematical Methods in Quantum Chemistry" held in March 2018 for example, and learnt

to know him as a researcher who is interested in fundamental concepts and answers to scientific questions, which of course is a common interest with mathematics. In this regards, we think that this article reflects this particular facet of prof. Gauss' research activities.

The roots of this project lie in a project given to students in the so-called CAMMP Week Pro[1] (where CAMMP stands for Computational And Mathematical Modelling Program) where students work during one week intensively in a team on a research-related problem. It was fascinating to see what students can achieve within one week and their work has definitively contributed to get the ball rolling in this project.

Finally, the authors acknowledge Eric Cancès and Yvon Maday for fruitful discussions.

Funding

This work was supported by the French "Investissements d'Avenir" program, project ISITE-BFC (contract ANR-15-IDEX-0003).

Appendix A. Appendix

We describe in this appendix the method used in the case of multi-dimensional parameter spaces. In particular, we detail the motivations and justifications behind the choices leading to the method presented in Section 3.2.2.

First, the approximate density matrix $\mathsf{D}_{\mathrm{app}}(\mathsf{p})$ will be defined as $\mathsf{D}_{\mathrm{app}}(\mathsf{p}) := \mathrm{Exp}_{\mathrm{Gr},0}\left(\Gamma_{\mathrm{app}}(\mathsf{p})\right)$, with

$$\Gamma_{\mathrm{app}}(\mathsf{p}) = \sum_{i=1}^{n} L_i(\mathsf{p})\,\Theta_i, \tag{A1}$$

where the functions $L_i : \mathbb{P} \to \mathbb{R}$ and the reduced basis $\{\Theta_1, \ldots, \Theta_n\}$ have to be appropriately chosen.

Since each Θ_i consists of $N_{\mathsf{b}} \cdot N$ elements, an obvious upper bound on a dimensionality of the reduced basis is $n \leq N_{\mathsf{b}}N$, but we hope to have a reduced basis of a much smaller i.e. $n \ll N_{\mathsf{b}}N$. Let us assume for now that some arbitrary set of functions $\{L_i(\mathsf{p})\}_{i=1}^{n}$ are given and that they are stored in a row-vector $L(\mathsf{p}) = (L_1(\mathsf{p}), L_2(\mathsf{p}), \ldots, L_n(\mathsf{p}))$. Let us denote by Θ the 3-dimensional tensor such that $\Theta(i,:,:) = \Theta_i$ which is simply a stack of all elements $\Theta_i \in \mathbb{R}^{N_{\mathsf{b}} \times N}$. Since we are looking for approximations of the form (A1), we can rewrite it in compact notation as

$$\Gamma_{\mathrm{app}}(\mathsf{p}) = L(\mathsf{p})\Theta. \tag{A2}$$

With these considerations, it now becomes clear that we have to optimize simultaneously the reduced basis $\mathbb{V}_{\mathrm{rb}}$ as well as the functions $L_i(\mathsf{p})$ contained in the vector $L(\mathsf{p})$, i.e., we consider the minimization problem

$$\min_{\Theta \in \mathbb{R}^{n \times N_{\mathsf{b}} \times N}} \min_{L} \|\Gamma(\cdot) - L(\cdot)\Theta\|_*, \tag{A3}$$

where the norm $\|\cdot\|_*$ is arbitrary and any suitable norm can be chosen.

This problem can also be viewed from a different angle: For the given $N_{\mathsf{b}} \cdot N$ functions $\Gamma_{j,i}(\mathsf{p})$ and given functions $L_1(\mathsf{p}), L_2(\mathsf{p}), \ldots, L_n(\mathsf{p})$, one aims to approximate each $\Gamma_{j,i}(\mathsf{p})$ in the space spanned by elements of L, i.e., the L_i. Then, the ansatz (A1) can be seen as finding

[1] https://blog.rwth-aachen.de/cammp/angebot-fuer-studierende/

coefficients $\Theta_1, \Theta_2, \ldots, \Theta_n$, thus the reduced basis, for given row-vector $L(\mathsf{p})$. This corresponds to exchanging the order of the minima in (A3).

From this perspective we first prescribe d polynomial basis functions $P_1(\mathsf{p}), P_2(\mathsf{p}), \ldots, P_d(\mathsf{p})$, collected in the vector $P(\mathsf{p}) = (P_1(\mathsf{p}), P_2(\mathsf{p}), \ldots, P_d(\mathsf{p}))$, spanning a sufficiently large space such that the distance between $\Gamma(\mathsf{p})$ and its projection to the space spanned by $P(\mathsf{p})$ is smaller than a certain threshold. Just like the size n of the reduced basis, reasonable value of d is assumed to satisfy $d \ll N_{\mathsf{b}}N$. Then, for given functions $P_i(\mathsf{p})$ (and thus $P(\mathsf{p})$) one is aiming at a Θ that minimizes the following distance:

$$\Theta_P = \underset{\Theta \in \mathbb{R}^{d \times N_{\mathsf{b}} \times N}}{\arg\min} \|\Gamma(\cdot) - P(\cdot)\Theta\|_*. \tag{A4}$$

Note that the dimension d of the reduced basis, as constructed like this, will be reduced in a further step. As norm $\|\cdot\|_*$, we will first consider the ideal choice

$$\|\Gamma(\cdot) - P(\cdot)\Theta\|_*^2 = \int_{\mathbb{P}} \|\Gamma(\mathsf{p}) - P(\mathsf{p})\Theta\|_F^2 \, d\mathsf{p}, \tag{A5}$$

as starting point. Here $\|\cdot\|_F$ stands for the Frobenius norm for matrices. Having the exact $\Gamma(\mathsf{p})$ at every point $\mathsf{p} \in \mathbb{P}$ is not feasible in practice which motivates to introduce a quadrature rule based on points p_j and weights ω_j, $j = 1, \ldots, N_{\mathsf{p}}$ given by:

$$\|\Gamma(\cdot) - P(\cdot)\Theta\|_\circ^2 := \sum_{j=1}^{N_{\mathsf{p}}} w_j \|\Gamma(\mathsf{p}_j) - P(\mathsf{p}_j)\Theta\|_F^2 \approx \|\Gamma(\cdot) - P(\cdot)\Theta\|_*^2. \tag{A6}$$

Introducing $\widetilde{\Gamma} \in \mathbb{R}^{N_{\mathsf{p}} \times (N_{\mathsf{b}} \cdot N)}$, $\widetilde{\Theta} \in \mathbb{R}^{d \times (N_{\mathsf{b}} \cdot N)}$ and $\widetilde{P} \in \mathbb{R}^{N_{\mathsf{p}} \times d}$ defined by

$$\widetilde{\Gamma}_{j,:} = \mathsf{reshape}(\Gamma(\mathsf{p}_j), 1, N_{\mathsf{b}} \cdot N), \tag{A7}$$

$$\widetilde{\Theta} = \mathsf{reshape}(\Theta, d, N_{\mathsf{b}} \cdot N), \tag{A8}$$

$$\widetilde{P}_{j,i} = P_i(\mathsf{p}_j), \tag{A9}$$

we rewrite the optimization problem as follows:

$$\widetilde{\Theta}_{\widetilde{P}} = \underset{\widetilde{\Theta} \in \mathbb{R}^{d \times (N_{\mathsf{b}} \cdot N)}}{\arg\min} \|\widetilde{\Gamma} - \widetilde{P}\widetilde{\Theta}\|_F, \tag{A10}$$

assuming p_j is a uniform grid in $\mathbb{P}$, i.e. $\omega_j = \frac{|\mathbb{P}|}{N_{\mathsf{p}}}$.

In consequence, we transformed the problem to a least squares problem, whose solution, for given $\widetilde{P}$, is given by the pseudoinverse of $\widetilde{P}$ acting on $\widetilde{\Gamma}$:

$$\widetilde{\Theta}_{\widetilde{P}} = \widetilde{P}^\dagger \widetilde{\Gamma}. \tag{A11}$$

Thus, in the case where the matrix $\widetilde{P}$ is given, we have an explicit expression for the minimizer and one can easily compute the optimal coefficients Θ_i of the approximation (19). Returning to the global optimization problem (A3), this allows us to write the optimization problem in only one variable, namely the $N_{\mathsf{p}} \times d$ matrix $\widetilde{P}$. The minimization problem becomes

$$\widetilde{P}_{\mathsf{opt}} = \underset{\widetilde{P} \in \mathbb{R}^{N_{\mathsf{p}} \times d}}{\arg\min} \|\widetilde{\Gamma} - \widetilde{P}\widetilde{P}^\dagger\widetilde{\Gamma}\|_F. \tag{A12}$$

The solution of such an optimization problem is the best approximation of $\widetilde{\Gamma}$ by matrices of (given) rank d (the size of reduced basis) and can be obtained by performing the singular value

decomposition of the matrix $\widetilde{\Gamma} = U\Sigma V^\top$, so that $\widetilde{P}_{\mathsf{opt}}$ and $\widetilde{\Theta}_{\mathsf{opt}}$ are given by

$$\widetilde{P}_{\mathsf{opt}} = U_d, \qquad \widetilde{\Theta}_{\mathsf{opt}} = U_d^\top \widetilde{\Gamma}, \tag{A13}$$

where $U_d\Sigma_d V_d^\top$ is the rank d approximation of $\widetilde{\Gamma}$ with $U_d \in \mathbb{R}^{N_\mathsf{p}\times d}$, $\Sigma_d \in \mathbb{R}^{d\times d}$ and $V_d \in \mathbb{R}^{N_\mathsf{b}\cdot N\times d}$. This provides the solution to the optimization problem (A3) for the particular norm defined in (A6).

Unfortunately, the optimal $\widetilde{\Theta}_{\mathsf{opt}}$, as can be found in equation (A13), requires full knowledge of $\widetilde{\Gamma}$, i.e., $\Gamma(\mathsf{p}_j)$ for every quadrature point p_j. In the following we show how we can drastically reduce the amount of points p where we need to compute $\Gamma(\mathsf{p})$. In order to do so, we propose to replace the Frobenius norm in favour for the **max** norm for matrices, also known as Chebyshev norm, given by

$$\|A\|_\mathrm{C} = \max_{ij} |a_{ij}|. \tag{A14}$$

This leads to the following optimization problem

$$\widetilde{\Theta}_{\mathsf{opt},\mathrm{C}} = \operatorname*{arg\,min}_{\widetilde{\Theta}\in\mathbb{R}^{d\times(N_\mathsf{b}\cdot N)}} \|\widetilde{\Gamma} - \widetilde{P}\widetilde{\Theta}\|_\mathrm{C}. \tag{A15}$$

We then aim to find quasi-optimal solutions of this problem by so-called *interpolative* approximations of the form

$$\widetilde{\Gamma}_{\mathsf{app}} = CUR, \tag{A16}$$

where either C is a collection of "basis" columns of $\widetilde{\Gamma}$ or R is a collection of "basis" rows of $\widetilde{\Gamma}$ and U is a "core" matrix. If both C and R are submatrices of $\widetilde{\Gamma}$, then the "core" matrix is, usually, an inverse or pseudo-inverse of the intersection of the "basis" rows R and the "basis" columns C. Such an approximation is called cross approximation since the intersection of columns and rows reminds of a cross. A theoretical analysis of cross approximations, provided in [33, 40], proves that such a rank d decomposition exists, i.e., $U \in \mathbb{R}^{d\times d}$, such that

$$\|\widetilde{\Gamma} - CUR\|_\mathrm{C} \leq (d+1)\,\sigma_{d+1}(\widetilde{\Gamma}). \tag{A17}$$

Here, $\sigma_{d+1}(\widetilde{\Gamma})$ denotes the $(d+1)$-st singular value of $\widetilde{\Gamma}$ in descending order. More recent results on the error estimation in the Chebyshev norm can be found in [41, 42]. Although the Chebyshev norm is studied well in terms of theory and practical methods, building cross approximations with controlled error in the spectral or Frobenius norm is still ongoing research. We refer to the recent papers [43, 44] for further information.

Since we are looking for the interpolative approximation by rows of $\widetilde{\Gamma}$, the matrix C appearing in equation (A16) can be chosen arbitrary as long as the space spanned by its columns approximates columns of $\widetilde{\Gamma}$ with high enough precision. The best choice is, of course, the first left singular vectors of $\widetilde{\Gamma}$, which again requires the undesirable full knowledge of $\widetilde{\Gamma}$. However, any column of $\Gamma(\mathsf{p})$ can by construction be well approximated by an element in the space spanned by the elements of $P(\mathsf{p})$ (the polynomial basis functions), so the matrix C can be defined as the matrix $\widetilde{P}$, previously defined as the vector $P(\mathsf{p})$ at all quadrature points p_j. Then the "core" matrix U is simply an inverse of some submatrix of $\widetilde{P}$ and the matrix R is just a collection of d rows of the matrix $\widetilde{\Gamma}$, corresponding to d computations of $\Gamma(\mathsf{p})$.

This is realized by analyzing only the matrix $\widetilde{P}$ to select a few samples $\{\mathsf{q}_j\}_{j=1}^d$ where we need to compute subsequently the matrices $\Gamma(\mathsf{q}_j)$. For this purpose we use the so-called *maxvol* method as introduced in [33]. It finds a quasi-*dominant* square $d\times d$ submatrix of $\widetilde{P}$ in $O(N_\mathsf{p} d^2)$ operations. A $d\times d$ submatrix of the $N_\mathsf{p}\times d$ matrix $\widetilde{P}$ is called *dominant* if the modulus of its

determinant does not grow if we change one of its rows by any other row of $\widetilde{P}$. *Quasi-dominance* means that the modulus of the determinant does not grow by more than a factor of $1+\alpha$ with a small value of α. Such a property is necessary for the theoretical error estimation presented in equation (A17).

In practise, the *maxvol* method takes the matrix $\widetilde{P} \in \mathbb{R}^{N_\mathsf{p} \times d}$ as input and returns the square quasi-dominant submatrix $\widehat{P} \in \mathbb{R}^{d \times d}$ along with a matrix of coefficients $C \in \mathbb{R}^{N_\mathsf{p} \times d}$, such that the product of the coefficients by the submatrix is equal to the input $\widetilde{P}$, i.e.,

$$\widetilde{P} = C\widehat{P}, \qquad C = \widetilde{P}\widehat{P}^{-1}. \tag{A18}$$

Let us denote $\{\mathsf{piv}(i)\}_{i=1}^{d}$ the set of row-indices such that $\widehat{P}_{i,j} = \widetilde{P}_{\mathsf{piv}(i),j}$. We now define

$$\widehat{\Gamma}_{i,j} = \widetilde{\Gamma}_{\mathsf{piv}(i),j} \qquad \mathsf{q}_i = \mathsf{p}_{\mathsf{piv}(i)}. \tag{A19}$$

Then, the interpolative approximation of $\widetilde{\Gamma}$ is given by

$$\widetilde{\Gamma} \approx \widetilde{P}\widehat{P}^{-1}\widehat{\Gamma} = \widetilde{P}\widetilde{\Theta} \tag{A20}$$

with $\widetilde{\Theta} = \widehat{P}^{-1}\widehat{\Gamma}$ and we define the approximation

$$\Gamma_{\mathrm{app}}(\mathsf{p}) = \sum_{i=1}^{d} \left(\mathcal{P}(\mathsf{p})\widehat{P}^{-1}\right)_i \Gamma(\mathsf{q}_i). \tag{A21}$$

One of the main features of this approximation is that, in order to compute the value of $\Gamma(\mathsf{p})$ at a new point p, we only need to compute a row-vector $P(\mathsf{p})$ of values of the polynomials $P_i(\mathsf{p})$ at the new point p and that the functions $\left(P(\mathsf{p})\widehat{P}^{-1}\right)_i$ are polynomials in the prescribed space.

Since we are working with $\widetilde{P}$ instead of $\widetilde{\Gamma}$, the actual error is different from the estimation in (A17). We omit the error analysis of our approximation in this paper and plan to release it in a follow-up article.

Note that the "basis" rows $\widehat{\Gamma}$ of the matrix $\widetilde{\Gamma}$ can be highly linearly dependent. We therefore consider the singular value decomposition of the matrix $\widehat{\Gamma}$ and truncate it up to rank n:

$$\widehat{\Gamma} = \widehat{U}_n \widehat{S}_n \widehat{V}_n + \widehat{E}_n, \tag{A22}$$

such that $\widehat{U}_n \in \mathbb{R}^{d \times n}, \widehat{S}_n \in \mathbb{R}^{n \times n}, \widehat{V}_n \in \mathbb{R}^{n \times (N_\mathsf{b} \cdot N)}$ and $\widehat{E}_n$ is the remaining term due to truncation. Substituting the truncated SVD into (A20) we get:

$$\widetilde{\Gamma} \approx \widetilde{P}\widehat{P}^{-1}\widehat{U}_n \widehat{S}_n \widehat{V}_n. \tag{A23}$$

Let us denote the i-th row of $\widehat{V}_n$, after reshaping into a $N_\mathsf{b} \times N$ matrix, as Θ_i and the product $\widehat{P}^{-1}\widehat{U}_n S_n$ as a matrix Z. Then, we approximate the value of $\Gamma(\mathsf{p})$ for any new value of p as

$$\Gamma_{\mathrm{app}}(\mathsf{p}) = \sum_{i=1}^{n} L_i(\mathsf{p})\Theta_i. \tag{A24}$$

with $L_i(\mathsf{p}) = (P(\mathsf{p})Z)_i$, which is exactly of the form (A1). The additional SVD further reduces the dimension of the reduced basis but of course introduces another error, which is nevertheless controlled by the singular values, but such an approximation requires a proper theoretical analysis, which is omitted in this paper. It can be derived in the same way as the theoretical estimations in [41, Theorem 4.8].

The proposed approximation technique can be formally divided into two parts: an offline stage, where the reduced basis is pre-computed, and an online stage, where an approximate value of $\mathsf{D}(\mathsf{p})$ is computed efficiently for a given p.

The offline part is schematically illustrated in Algorithm 1. This stage requires $O((N_\mathsf{p} + N_\mathsf{b}N)d^2 + N_\mathsf{b}^\beta d)$ operations, where N_p stands for the number of quadrature points p_j, N_b and N are the number of atomic orbital basis functions and the number of orbitals respectively. Further, d is the number of basis multivariate monomials and β is a power factor determined by the specific nature of the eigenvalue solver that is employed to solve (4)-(5).

The online part containing the approximation of $\mathsf{D}(\mathsf{p})$ for any new p is sketched by Alg. 2. It shall be noticed that this part is of much lower complexity: it uses only $O(n(N_\mathsf{b}N + d^2) + N_\mathsf{b}N^2)$ operations, where n is a size of the final reduced basis. It is worth to emphasize, that both procedures, offline as well as online, must use the same set of basis monomials $\{P_1, \ldots, P_d\}$.

References

[1] H.B. Schlegel, WIREs Comput. Mol. Sci. **1** (5), 790–809 (2011).
[2] C.C.J. Roothaan, Rev. Mod. Phys. **23**, 69–89 (1951).
[3] W. Kohn and L.J. Sham, Phys. Rev. **140**, A1133–A1138 (1965).
[4] H.B. Schlegel and J.J.W. McDouall, in *Computational Advances in Organic Chemistry: Molecular Structure and Reactivity*, edited by Cemil Ögretir and Imre G. Csizmadia (, , 1991), Chap. 2, pp. 167–185.
[5] P. Pulay, Chem. Phys. Lett. **73** (2), 393–398 (1980).
[6] P. Pulay, J. Comput. Chem. **3** (4), 556–560 (1982).
[7] K.N. Kudin, E. Cancès and G.E. Scuseria, J. Chem. Phys. **116** (19), 8255–8261 (2002).
[8] Y.A. Wang, C.Y. Yam, Y.K. Chen and G. Chen, J. Chem. Phys **134** (24), 241103 (2011).
[9] M. Wolfsberg and L. Helmholz, J. Chem. Phys **20** (5), 837–843 (1952).
[10] R. Hoffmann, J. Chem. Phys. **39** (6), 1397–1412 (1963).
[11] J. Almlf, K. Faegri Jr. and K. Korsell, J. Comput. Chem. **3** (3), 385–399 (1982).
[12] J. Harris, Phys. Rev. B **31**, 1770–1779 (1985).
[13] J.H. Van Lenthe, R. Zwaans, H.J.J. Van Dam and M.F. Guest, J. Comput. Chem. **27** (8), 926–932 (2006).
[14] S. Lehtola, J. Chem. Theory Comput. **15** (3), 1593–1604 (2019).
[15] S. Lehtola, L. Visscher and E. Engel, J. Chem. Phys. **152** (14), 144105 (2020).
[16] A.M.N. Niklasson, P. Steneteg, A. Odell, N. Bock, M. Challacombe, C.J. Tymczak, E. Holmstrm, G. Zheng and V. Weber, J. Chem. Phys. **130** (21), 214109 (2009).
[17] D. Loco, L. Lagardère, S. Caprasecca, F. Lipparini, B. Mennucci and J.P. Piquemal, J. Chem. Theory Comput. **13** (9), 4025–4033 (2017).
[18] J. Douady, Y. Ellinger, R. Subra and B. Levy, Comp. Phys. Comm. **17** (1), 23 – 25 (1979).
[19] G.B. Bacskay, Chem. Phys. **65** (3), 383 – 396 (1982).
[20] H.J. Werner and W. Meyer, J. Chem. Phys. **74** (10), 5794–5801 (1981).
[21] H.J.Aa. Jensen and P. Jørgensen, J. Chem. Phys. **80** (3), 1204–1214 (1984).
[22] H.J.Aa. Jensen and H. Ågren, Chem. Phys. Lett. **110** (2), 140–144 (1984).
[23] H.J. Werner and P.J. Knowles, J. Chem. Phys. **82** (11), 5053–5063 (1985).
[24] H.J.Aa. Jensen and H. Ågren, Chem. Phys. **104** (2), 229–250 (1986).
[25] F. Lipparini and J. Gauss, J. Chem. Theory Comput. **12**, 4284 (2016).
[26] J.S. Hesthaven, G. Rozza, B. Stamm *et al.*, *Certified reduced basis methods for parametrized partial differential equations*, Vol. 590 (, , 2016).
[27] E. Cancès, C. Le Bris, Y. Maday and G. Turinici, Journal of scientific computing **17** (1-4), 461–469 (2002).
[28] E. Cancès, C. Le Bris, N. Nguyen, Y. Maday, A.T. Patera and G.S.H. Pau, in *Proceedings of the workshop for high-dimensional partial differential equations in science and engineering (Montreal)*, Vol. 41, pp. 15–57.
[29] Y. Maday and U. Razafison, Comptes Rendus Mathematique **346** (3-4), 243–248 (2008).
[30] V. Schauer and C. Linder, Advances in Computational Mathematics **41** (5), 1035–1047 (2015).
[31] A. Edelman, T.A. Arias and S.T. Smith, SIAM journal on Matrix Analysis and Applications **20** (2), 303–353 (1998).

[32] R. Zimmermann, arXiv preprint arXiv:1902.06502 (2019).
[33] S.A. Goreinov, I.V. Oseledets, D.V. Savostyanov, E.E. Tyrtyshnikov and N.L. Zamarashkin, How to find a good submatrix, Research Report 08-10, ICM HKBU, Kowloon Tong, Hong Kong 2008 .
[34] Y. Maday, N.C. Nguyen, A.T. Patera and S. Pau, Communications on Pure & Applied Analysis **8** (1), 383 (2009).
[35] J.F. Stanton, J. Gauss, L. Cheng, M.E. Harding, D.A. Matthews and P.G. Szalay, CFOUR, Coupled-Cluster techniques for Computational Chemistry, a quantum-chemical program package, With contributions from A.A. Auer, R.J. Bartlett, U. Benedikt, C. Berger, D.E. Bernholdt, Y.J. Bomble, O. Christiansen, F. Engel, R. Faber, M. Heckert, O. Heun, M. Hilgenberg, C. Huber, T.-C. Jagau, D. Jonsson, J. Jusélius, T. Kirsch, K. Klein, W.J. Lauderdale, F. Lipparini, T. Metzroth, L.A. Mück, D.P. O'Neill, D.R. Price, E. Prochnow, C. Puzzarini, K. Ruud, F. Schiffmann, W. Schwalbach, C. Simmons, S. Stopkowicz, A. Tajti, J. Vázquez, F. Wang, J.D. Watts and the integral packages MOLECULE (J. Almlöf and P.R. Taylor), PROPS (P.R. Taylor), ABACUS (T. Helgaker, H.J. Aa. Jensen, P. Jørgensen, and J. Olsen), and ECP routines by A. V. Mitin and C. van Wüllen. For the current version, see http://www.cfour.de.
[36] J. Thom H. Dunning, J. Chem. Phys. **90** (2), 1007–1023 (1989).
[37] J. Bezanson, A. Edelman, S. Karpinski and V.B. Shah, SIAM Review **59** (1), 65–98 (2017).
[38] M.J. Frisch, G.W. Trucks, H.B. Schlegel, G.E. Scuseria, M.A. Robb, J.R. Cheeseman, G. Scalmani, V. Barone, G.A. Petersson, H. Nakatsuji, X. Li, M. Caricato, A.V. Marenich, J. Bloino, B.G. Janesko, R. Gomperts, B. Mennucci, H.P. Hratchian, J.V. Ortiz, A.F. Izmaylov, J.L. Sonnenberg, D. Williams-Young, F. Ding, F. Lipparini, F. Egidi, J. Goings, B. Peng, A. Petrone, T. Henderson, D. Ranasinghe, V.G. Zakrzewski, J. Gao, N. Rega, G. Zheng, W. Liang, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, K. Throssell, J.A. Montgomery, Jr., J.E. Peralta, F. Ogliaro, M.J. Bearpark, J.J. Heyd, E.N. Brothers, K.N. Kudin, V.N. Staroverov, T.A. Keith, R. Kobayashi, J. Normand, K. Raghavachari, A.P. Rendell, J.C. Burant, S.S. Iyengar, J. Tomasi, M. Cossi, J.M. Millam, M. Klene, C. Adamo, R. Cammi, J.W. Ochterski, R.L. Martin, K. Morokuma, O. Farkas, J.B. Foresman and D.J. Fox, Gaussian16 Revision A.03, Gaussian Inc. Wallingford CT, 2016.
[39] Q. Sun, T.C. Berkelbach, N.S. Blunt, G.H. Booth, S. Guo, Z. Li, J. Liu, J.D. McClain, E.R. Sayfutyarova, S. Sharma, S. Wouters and G.K. Chan, Wiley Interdisciplinary Reviews: Computational Molecular Science **8** (1), e1340 (2017).
[40] S.A. Goreinov and E.E. Tyrtyshnikov, Contemporary Mathematics **280**, 47–52 (2001).
[41] A.Y. Mikhalev and I.V. Oseledets, Linear Algebra Appl. **538** (1), 187–211 (2018).
[42] A.I. Osinsky and N.L. Zamarashkin, Linear Algebra and its Applications **537**, 221–249 (2018).
[43] N.L. Zamarashkin and A.I. Osinsky, in *Doklady Mathematics*, Vol. 97 (, , 2018), pp. 164–166.
[44] A. Cortinovis and D. Kressner, arXiv preprint arXiv:1908.06059 (2019).