SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master Thesis: Final report

Optimized basis sets for electronic structure calculations

Utkarsh Saraswat

SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master Thesis: Final report

Optimized basis sets for electronic structure calculations

Author: Utkarsh Saraswat
Supervisor: Prof. Dr. Christian Mendl
Advisor: MSc. Tobias Bauer

I confirm that this master thesis: final report is my own work and I have documented all sources and material used.

Munich, 12 May, 2025 Utkarsh Saraswat

Acknowledgments

I would like to express my deepest gratitude to my supervisor, Professor Christian Mendl, and my advisor, Tobias Bauer, for their unwavering support and guidance throughout the course of my research. Their insightful feedback, encouragement, and expertise have been invaluable to me at every stage of this journey.

I am especially thankful for the enriching discussions we had on complex topics, which greatly helped me clarify my thoughts and deepen my understanding. The willingness to dedicate time to these conversations, as well as to share reading material, truly made a difference in shaping this work.

I am also deeply appreciative of Tobias for his help in the writing process, particularly with structuring the discussion, refining ideas, and meticulously proofreading the manuscript. Your attention to detail and high standards have greatly improved the quality of this thesis.

Finally, I am sincerely grateful for the opportunity to work under the mentorship of Prof Christian and Tobias. It has been a privilege to learn from you, and this experience has significantly contributed to my academic and personal growth.

Contents

Acknowledgments **iii**

1. Introduction **1**

2. Technical Preliminaries **3**
2.1. Gaussian function and Gausslets . . . 3
2.2. Electronic Hamiltonian . . . 5
2.3. Second quantization of Electronic Hamiltonian . . . 6
2.4. Multi-resolution Analysis and Wavelet transform . . . 7
2.5. Orthogonal wavelets . . . 12
2.6. Tensor network methods . . . 14

3. Harmonium model for simulation of Quantum dots and trapped ions **19**
3.1. Emergent behavior of inverse square law forces . . . 19
3.2. Simulation of Quantum Dots and Ion Traps . . . 20
3.3. Evaluation of Hamiltonian terms in Harmonium models using ternary Gausslets 23
3.4. Hermite polynomial and Hermite basis function . . . 24

4. Method **26**
4.1. Problem statement . . . 26
4.2. Direct optimization method to get orthogonal Gausslet . . . 26
4.3. Unitary circuit method to obtain orthogonal Haar wavelets . . . 29
4.3.1. Calculation of dyadic wavelet coefficients using Unitary transformation 30
4.3.2. Calculation of ternary wavelet coefficients using Unitary transformation 33
4.4. Retrieving smooth basis functions from Haar basis coefficients: Gausslets . . . 39
4.4.1. Direct inverse method . . . 40
4.4.2. Discretization-based approximation method . . . 42
4.5. Numerical methods and algorithms . . . 45
4.5.1. Scaled sampling method . . . 45
4.5.2. Measuring orthogonality of a basis . . . 46
4.6. Symmetric permutation method to obtain orthogonal wavelet ansatz . . . 49

5. Results and Conclusion **53**
5.1. Approximation of Hermite polynomials using Gausslets . . . 53
5.2. Approximation of Coulomb Integral using Gausslets . . . 55
5.3. Conclusion and future work . . . 56

A. Derivation of Mathematical Results **60**
A.1. Product of two Gaussian 60
A.2. Electronic integrals for Harmonic model with Gausslets 60

Bibliography **65**

1. Introduction

The earliest attempt to study the electronic structure of atoms is nearly as old as the discovery of subatomic particles itself. Several speculations by the best minds of the 20th century wondered how negatively charged particles could sit around a positively charged atomic nucleus. It was not until 1927 that Erwin Schrödinger, an Austrian-Irish physicist, proposed one of the most influential equations in history, finally allowing us to visualize the shapes of the hydrogen atom. While the solution of the Schrödinger equation for the Hydrogen atom is relatively simple, spherical harmonic eigenfunctions, any increase in the number of electrons and nuclei makes the problem exponentially harder to solve. Several theories to approximate molecular orbitals have been developed; nevertheless, determining the precise electronic structure of large molecules is one of the most challenging problems, so much so that we are looking forward to quantum computers to solve it.

While it is not always required to get the precise shape of electronic wave functions in a molecular system, often we are interested in retrieving the energies of electronic states. The retrieval of electronic structure and energies is the eigenvector and eigenvalue problem, respectively, and in general, it has to be solved simultaneously. Molecules, especially larger ones, often exhibit complex and arbitrary geometry, which ultimately determines the final form of the Hamiltonian in its Schrödinger equation. Consequently, obtaining an expression for eigenfunctions/eigenvalues in a closed-form mathematical expression for such a system is impossible. Additionally, the fermionic nature of electrons and the particle number conservation impose additional constraints on eigenvalue problems and allow only a subset of possible solutions. Thankfully, the mathematical framework of tensor networks was developed by Roger Penrose in the 1970s to study complex quantum systems. Through tensor networks, one could i) represent states of many-body systems as n-dimensional tensors as nodes, ii) keep track of correlations between different states as edges in a graph. We will discuss the tensor network in detail later.

In any scientific research, converting a system of study into a linear system is highly desirable, as linear algebra is one of the most well-developed branches of mathematics. If you know linear algebra, you can study any linear system. Secondly, all data and computing operations in any modern computer occur in matrices and vectors (arrays to be precise), i.e., if you want to feed your problem to a computer, it has to be linearized first. Dealing with a linear system, though, comes with its issues. Computational complexity of a general matrix-vector multiplication and matrix-matrix multiplication is $O(n^2)$ and $O(n^3)$, respectively, which is already significant for a single large matrix and in tensor networks, where contraction algorithms have an overall complexity that is a multiple of the individual complexity of a single tensor. As a result, scaling has a drastic impact on computational resources. Therefore, most tensor network algorithms involve singular value decomposition of tensors, which

allows diagonal/sparse representation of tensors. A better option is to start with diagonal tensors in the first place. Representing molecular systems as a diagonal tensor network has a relatively straightforward way of calculating energy (eigenvalues) using tensor contraction algorithms. The procedure is multi-step. First, we represent the Schrödinger equation in second quantization, which consists of separate terms for single-electron and two-electron interactions. Each term has a tensor form (2-d matrix for single and 4-d tensor for two-electron terms) along with respective fermionic operators. These tensors can be constructed by the weak formulation of the Schrödinger equation (using an appropriate basis set similar to finite element analysis) using a suitable basis set. Therefore, we need to find a basis with vanishing non-diagonal terms, or in other words, an orthogonal basis. However, finding an orthogonal basis is insufficient, as the numerical solution may have significant errors relative to the actual solution because any approximation lies in the space of the basis set, which may or may not be a subset of the actual solution space. A basis set that spans the true solution space is called a complete basis. Usually, finding an orthogonal basis with excellent completeness is hard.

Our thesis aims to find an orthogonal and complete basis to solve the Schrödinger equation of multi-electron systems. We will start with the foundations of the multi-electron Hamiltonian in second quantization. Then we will introduce the theory of multi-resolution analysis, which is an essential tool to find our desired basis set. Then, we will describe the unitary circuit methodology discovered by Steven White to generate an orthogonal basis using a sequence of linear transformations. Later, we will go into the detailed implementation of the orthogonal Gausslet basis and apply it to trapped ion and quantum dot systems. The scope of our work can be further extended to the discovery of computationally efficient algorithms to calculate the expectation value (and hence energy corresponding to a given electronic state).

2. Technical Preliminaries

2.1. Gaussian function and Gausslets

Since the Gaussian function plays the most significant role in the thesis, in this section, we will describe the Gaussian functions and the Gausslet basis, which allows us to have a continuous, orthogonal, and complete basis that can be used to simulate Fermionic wavefunctions. The general mathematical expression of Gaussian functions is given by

$$f(x) = e^{-\frac{(x-\mu)^2}{2\sigma}}$$

With $\mu, \sigma \in \mathbb{R}$ being the mean and the variance, respectively. While the indefinite integral of a Gaussian cannot be expressed in a closed mathematical expression, the definite integral with limits from $-\infty$ to $+\infty$ is given by

$$\int_{-\infty}^{\infty} e^{-x^2}\, dx = \sqrt{\pi}$$
$$\int_{-\infty}^{\infty} e^{\frac{-x^2}{k^2}}\, dx = k\sqrt{\pi}$$

Gaussian functions are found in several naturally occurring phenomena spanning all domains, from natural science to economics, which is one of the motivations for using them as an electronic wave function basis. To get more interesting shapes out of Gaussians to represent complex wave functions, we can simply combine Gaussians at different centers (a_i) and amplitudes (c_i) to form a new wave function centered at β_j as follows.

Definition 1. *A function $\phi : \mathbb{R} \to \mathbb{R}$ given by linear combination of Gaussian with $\beta \in \mathbb{R}, \alpha \in \mathbb{R}$ of form*

$$\phi(x, \beta) = \sum_i c_i e^{-\frac{(x-\alpha_i-\beta)^2}{2\sigma}} \tag{2.1}$$

*is known as **Gausslet**.*

Forming a basis set constituent of Gausslet gives us the freedom to choose parameters c_i, α_i, and β, and fine-tune them to satisfy the LOC conditions (Locality, Orthogonality, and Completeness). Before we can proceed, we will make the following simplifying assumptions to make our life easier-

1. All basis functions have to be symmetric at their respective centers (β_i)
2. the centers α_i of individual Gaussian and the centers of Gausslets β in equation (2.1) are always integer multiples of some constants α, β, i.e $\alpha_i = i\alpha; \beta_j = j\beta : i, j \in \mathbb{Z}$

Definition 2. *A Gausslet which is symmetric and centered around $x = \beta_j$ and degree n (with total number of terms as $2n + 1$) as -*

$$\phi^n(x, \beta_j) = c_n e^{-\frac{(x+n\alpha-\beta_j)^2}{2\sigma}} + ... + c_1 e^{-\frac{(x+\alpha-\beta_j)^2}{2\sigma}} + c_0 e^{-\frac{(x-\beta_j)^2}{2\sigma}} + c_1 e^{-\frac{(x-\alpha-\beta_j)^2}{2\sigma}} + ... + c_n e^{-\frac{(x-n\alpha-\beta_j)^2}{2\sigma}}$$

with $n \in \mathbb{N} \cup \{0\}, \beta_j, \alpha \in \mathbb{N}$ is defined as a symmetric Gausslet

For symmetric Gausslets, it must follow that-

$$c_{N-i} = c_i$$

with $N =$ total number of Gaussian terms (or c_i), also implying the total number of c_i must be odd, since c_i, c_{N-i} pair exists for all i and there is exactly one c_0, i.e there are $2n + 1$ basis Gaussian function for symmetric Gausslet basis. Let us look at a few examples-
The simplest cases of symmetric Gausslets are

$$\phi^0(x, \beta_j) = c_0 e^{-\frac{(x-\beta_j)^2}{2\sigma}}$$
$$\phi^1(x, \beta_j) = c_1 e^{-\frac{(x+\alpha-\beta_j)^2}{2\sigma}} + c_0 e^{-\frac{(x-\beta_j)^2}{2\sigma}} + c_1 e^{-\frac{(x-\alpha-\beta_j)^2}{2\sigma}}$$

and for the case of $\beta = 0$ and $\sigma = 1$ we have

$$\phi^1(x, 0) = c_1 e^{-\frac{(x+\alpha)^2}{2}} + c_0 e^{-\frac{x^2}{2}} + c_1 e^{-\frac{(x-\alpha)^2}{2}} \tag{2.2}$$

with $\sigma = 1/2$ and $\beta_j = 3j$, i.e., the shift in the center is in multiples of 3, then j^{th} Gausslet is

$$\phi^n(x, 0) = e^{-(x-3j)^2} \text{ such that } j \in \{-n, -n+1, ... -1, 0, 1, 2, ...n\}$$

A Gausslet basis centered at 0 with 5 Gaussian terms will be given by

$$\phi^2(x, 0) = c_2 e^{-(x+6)^2} + c_1 e^{-(x+3)^2} + c_0 e^{-x^2} + c_1 e^{-(x-3)^2} + c_2 e^{-(x-6)^2}$$

Same basis centered at $x = 3$ or $j = 1$ with 5 Gaussian will be

$$\phi^2(x, 0) = c_2 e^{-(x+3)^2} + c_1 e^{-x^2} + c_0 e^{-(x-3)^2} + c_1 e^{-(x-6)^2} + c_2 e^{-(x-9)^2}$$

2.2. Electronic Hamiltonian

The Schrodinger equation for steady state in its most native form is

$$\hat{H}\psi = E\psi$$

Here, $\hat{H}$ represents the Hamiltonian operator acting on ψ, representing a wavefunction. Essentially, the Hamiltonian takes the states of a system as input and outputs the net energy of the system represented by E in the equation. For a single electron, we have-

$$\hat{H} = \hat{T} + \hat{V}$$

where T and V are Kinetic and potential energy operators, respectively. In quantum mechanics momentum of a particle is calculated using operator $\hat{p} = -\hbar \mathrm{i}\nabla = -\hbar \mathrm{i}[\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}]^T$ applied to wavefunction $\Psi(x, y, z) : \mathbb{R}^3 \to \mathbb{C}$. To get kinetic energy from Ψ, we can use classical (non-relativistic) operator $\frac{\hat{p}^2}{2m}$ as

$$\frac{\hat{p}^2}{2m} = (-\hbar \mathrm{i}\nabla).(-\hbar \mathrm{i}\nabla)/2m = \frac{-\hbar^2}{2m}(\nabla.\nabla) = \frac{-\hbar^2}{2m}\nabla^2 = \frac{-\hbar^2}{2m}\Delta = \frac{-\hbar^2}{2m}(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2})$$

The potential energy (E_p) of an electron can be calculated using Coulomb's law

$$E_p = \sum_{i \neq e} \kappa {q_e}^2 / |r_e - r_i| - \sum_N \kappa Z_N {q_e}^2 / |r_N - r_e|$$

with q_e being charge of an electron, k electrostatic constant (also given by expression as $\frac{1}{4\pi\epsilon_0}$ with ϵ_0 being vacuum electric permittivity), r_n, r_e position of nucleus and electron respectively in 3d space; r_i as positions of other electrons and Z_N atomic number (considering existence of multiple nucleic centers). Since E_p is independent of Ψ and only a function of $\mathbf{r} = [x, y, z]$, we can write potential energy operator as $\hat{V}\Psi = E_p\Psi = E_{ee}\Psi + E_{ne}\Psi = (V_{ee} + V_{ne})\Psi$. Where V_{ee}, V_{ne} represent the electrostatic potential energy from electron-electron repulsion and electron-nuclear attraction, respectively. The resulting expression for a single electron interacting with multiple nuclei is

$$\hat{H}\Psi = (\frac{-\hbar^2}{2m}\Delta + V_{ee} + V_{ne})\Psi \tag{2.3}$$

It is important to note that the expression in (2.3) doesn't imply that we can add Δ and V_{ne} or Δ and V_{ee}, it simply means that the operator can be applied on the wavefunction distributively and the resultant energies can be added together.

We are not interested in just one electron but the entire system comprising multiple electrons and nuclear centers. It modifies the equation to

$$H = \sum_i \frac{-\hbar^2}{2m}\Delta_{r_i} + \sum_{i,j} \kappa q_e^2 \frac{1}{r_{ij}} - \sum_{i,N} \kappa q_e^2 \frac{Z_N}{r_i}$$

with i, j representing indices of electrons and N for nucleic centers. Ignoring all constant factors for a more convenient analysis (or assuming all of them to be equal to 1), the equation can be rewritten as

$$H = \sum_i -\frac{1}{2}\Delta_{r_i} + \sum_{i,j}\frac{1}{r_{ij}} - \sum_{i,N}\frac{Z_N}{r_i}$$

Using **Born-Oppenheimer approximation** through which we can assume that the wave functions of the nucleus and electrons in a molecule can be treated separately. Instead of grouping the terms based on Kinetic and Potential Energy, a more convenient way is based on single electron terms vs multiple (dual) electron terms, leading to

$$H = (\sum_i -\frac{1}{2}\Delta_{r_i} - \sum_{i,N}\frac{Z_N}{r_i}) + \sum_{i,j}\frac{1}{r_{ij}} H = \sum_i \hat{h}(r_i) + \sum_{i,j}\frac{1}{r_{ij}} \tag{2.4}$$

where nucleic and kinetic terms for a single electron are clubbed into the term $\hat{h}$.

Intuitive explanation

Based on **Born-Oppenheimer approximation**[1], we can separately treat wave functions of two quantum systems, the electron and the nucleus. The consequence is that when we are interested in studying electronic structure, there will be some component of the nuclear wave function that we ideally have to consider. Since the effect is infinitesimally small in most cases due to the huge mass of the nucleus in comparison to the electron, it can be safely ignored. It is similar to the case when making orbital calculations of space probes. However, any fluctuations in the position of planets and moons also occur due to mutual gravitational attraction; we can safely ignore them. Another interpretation is that a nucleus's orders of magnitude higher mass than electrons, there is negligible uncertainty in their position, and hence they can be considered stationary; we can safely ignore their quantum nature in electronic structure calculations.

2.3. Second quantization of Electronic Hamiltonian

One of the main objectives of using Hamiltonian formalism is finding stable states of the system under study. The Hamiltonian as written in equation 2.4 is known as **first quantization**, and stable states can be found by solving eigenvalue problems.

Without explicitly proving mathematically, we will intuitively explain the second quantization terms. The molecular Hamiltonian in second quantization is written as

$$H = \sum_{ij} t_{ij} a_i^\dagger a_j + \sum_{ijkl} V_{ijkl} a_i^\dagger a_j^\dagger a_k a_l$$

with

$$t_{ij} = \int_R \phi_i^*(r)\hat{f}\phi_j(r)\,dr \tag{2.5}$$

$$V_{ijkl} = \int_R \int_R \phi_i^*(r_1)\phi_j^*(r_1)\frac{1}{|r_1 - r_2|}\phi_k(r_2)\phi_l(r_2)\,dr_1\,dr_2 \tag{2.6}$$

Here, the first term constitutes the single-electron part where $\hat{f}$ is the Fock operator, which is the Hamiltonian obtained via variational optimization of the Slater determinant (using calculus of variations to solve optimization problems of ground-state energy for a basis set) [1]. The second term is for two-electron interactions derived from Coulombic repulsion. In t_{ij} states $\phi_i(r)$ may or may not be orthogonal, therefore while calculating expectation value of energy given by $\hat{f}\psi = E_f\psi$, along with the obvious $\int_R \phi_i^*(r)\hat{f}\phi_i(r)\,dr$ terms, we have taken all the possible overlaps $\int_{R(i\neq j)} \phi_j^*(r)\hat{f}\phi_i(r)\,dr$ into account as well.

Note

Note that it doesn't mean an electron is present at states i and j simultaneously, but it only means that an electron can be present at the overlap of state i and j, and we have to consider that case. If i and j are orthogonal states, there is no overlap, and all the terms with $i \neq j$ in t_{ij} will be zero. Essentially, net energy at its state i can be calculated by $E_i = \sum_j t_{ij}$. One more way to understand this is a calculation of partial distribution in probability theory: if we have a joint distribution $P(x, y)$, the partial distribution of x will be given by $\sum_y P(x, y) = X(x)$. Similar inference can be made about the $V_{i_1 j_1 i_2 j_2}$ expression. Since we have two electrons with possible overlap of states, we have a $2 \times 2 = 4$ combination of indices leading to four pairs, and the partial distribution is calculated with respect to the overlapping states of both electrons.

2.4. Multi-resolution Analysis and Wavelet transform

Since most of the data we come across is often represented as functions of two variables, it would be very convenient if there were a formal procedure through which a given function could be projected to any arbitrary number of coarseness. Based on the paper "Orthonormal Bases of Compactly Supported Wavelets" by Ingrid Daubechies and further explained by Keinert in the book "Wavelets and Multi-wavelets", a framework of **Multi-resolution analysis or MRA** was developed, through which arbitrary square integrable functions can be approximated in terms of family of orthogonal basis functions. Orthogonal basis is not a new concept, and various orthogonal spaces were known earlier (including Legendre polynomial, Chebyshev polynomial, and Fourier space). Still, the theory of MRA formalized the procedure of orthogonal basis creation and controlled the accuracy of the approximation.

The natural question is, why are we talking about multi-resolution analysis in the thesis about electronic structure calculation? First of all, the states of electrons are represented by wavefunctions, and as discussed before, getting an exact mathematical representation of complex wave functions is not always possible. We can utilize MRA to approximate wave

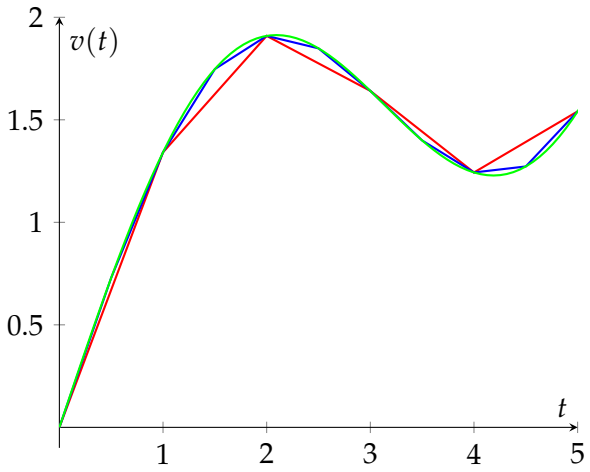


Figure 2.1.: Example to demonstrate use case of MRA via velocity time graph of particle. i) The green curve is a fine-grained plot of the hypothetical trajectory. ii) In the red spline curve, we are sampling for a coarseness level of 1 second (5 sampling points). iii) In the blue curve, we have a coarseness level of 0.5 seconds (10 sampling points). Using MRA, we aim to project any given function to any basis (linear spline in this case) to any coarseness level.

functions using familiar functions, which would also allow us to calculate energy levels (eigenvalues) of states. Secondly, as we will see later, MRA also allows us to efficiently calculate the electronic structure by providing a framework to obtain orthogonal basis functions. To understand MRA in detail, we will go step-by-step, first defining essential terms used in MRA, and then proving key results with illustrative examples.

Definition 3. *Given a function $f : \mathbb{R} \to \mathbb{C}$. If $\int |f(x)|^2 dx < \infty$, then f is a **square integrable function** in $\mathbb{R}$. A family of functions ϕ_n in $L^2(R)$. The set of all square integrable functions over $\mathbb{R}$ is represented as $L^2(R)$*

Definition 4. ***Refinement relation**: a function ϕ in $L^2(R)$ and a set of c_i which returns the same function on the left-hand side of the refinement equation as the input function on the right-hand side, i.e*

$$\phi = \sqrt{2} \sum_{k=1}^{N} h_k \phi(2x - k) \tag{2.7}$$

*Such a function is known as a **scaling function** or **refinable function***

The most obvious case of a refinable function is the Haar basis function as shown in Figure 2.2 with $\mathbf{h} = [1/\sqrt{2}, 1/\sqrt{2}]$.

Note

Consider a hypothetical case where we are keeping track of a car taking a tour across Europe spanning over a month. Let's say we want to record data about how far the car travels. Since the journey spanned across borders with a net distance in the order of thousands, we have an acceptable error tolerance of 10 km. Assuming the error in the distance is 1 percent of displacement between the two recorded positions, we have to sample the position of the car after every 1000 km or just once in two days (assuming average speed of 50 kmph and 10 10-hour drives in a day). A rough estimate of the distance can be made simply by joining all sampled positions through a straight line and summing up the length of all segments. Now we also want to calculate the net cost of a night's stay at a motel for passengers on the trip. Assuming that the night stay cost can be approximated within reasonable accuracy just by determining the town/city in which the motel is situated and its distance from the town/city center, we have to sample the position of the passengers again. It is known that the price of a motel can change drastically with distance and location; therefore, for a reasonably accurate measure, this time we have to sample the location of the car every day. Similarly, we can extend this case to calculate food cost, considering the passengers stopped three times per day for meals, and the prices of meals are also dependent on the distance from the city center. We notice that for the same trajectory of the trip, we require different sampling frequencies based on our use case. To approximate the net distance covered, we can use coarse sampling of 0.5 times a day, and for the calculation of accommodation and food costs, we need a finer sampling of 1 or 3 times per day. It also makes sense to avoid over-sampling (like in the case of distance) as it will require additional efforts with little gain in accuracy. This Example illustrates that there can be instances where, to produce a particular insight out of a given dataset, we would need to change the resolution of the data specific to the requirement.

In general, any function can be scaled by a factor a as

$$\phi_a = \sqrt{a}\phi(ax - k) \tag{2.8}$$

Which shrinks the width of a basis function a times (hence increasing the resolution of the function a times) and shifts it by a distance k/a while also preserving the normalization condition

$$\int_{-\infty}^{\infty} \phi^2(x).dx = \int_{-\infty}^{\infty} \phi_a^2(x).dx = 1$$

Definition 5. *Given a function f in $L^2(R)$ and subspace V_m spanned by integral translation $\phi_{mn} = \phi_m(x - n) : i \in \mathbb{Z}$ of refinable function ϕ_m with scaling coefficients $\boldsymbol{h} = \{h_1, h_2, ..h_l\}$. The Projector*

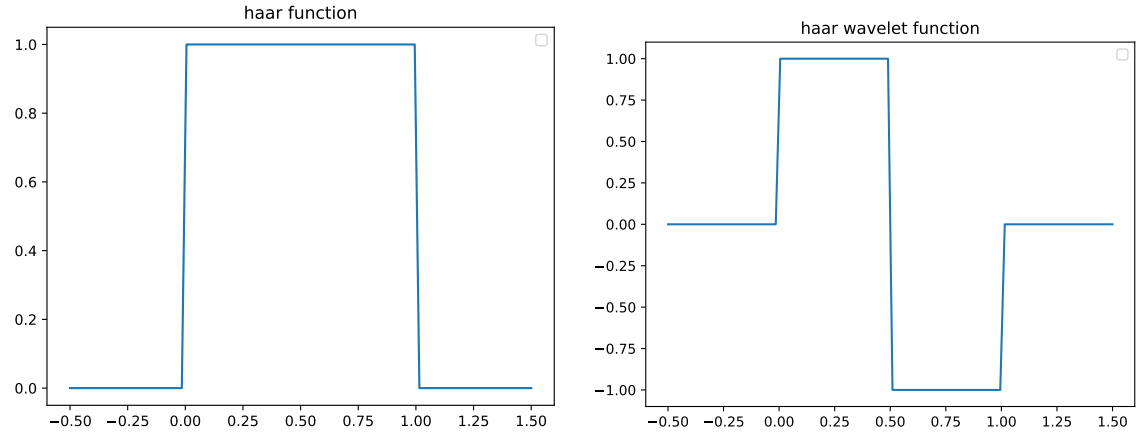


Figure 2.2.: Plot of Haar function and corresponding wavelet function. Support of Haar and Haar wavelet function is $[0,1]$ which is strictly local and satisfy $\int_{-\infty}^{\infty} \phi(x)\phi(x-n) = \delta_{n0}$ with $n \in \mathbb{N} \bigcup \{0\}$

operation P_m on f with respect to V_m is defined as

$$P_m f = \sum_i c_i \phi_{mi}$$
$$c_i = \int_{-\infty}^{\infty} \phi_{mi}.f(x).dx$$

We can see here why the condition of our function being square integrable is necessary. When performing integration of the product of two functions ranging from $-\infty$ to ∞, we expect the value of c_i to be bounded.

The projector is analogous to a linear algebra projector on a vector defined as

$$P_u v = uu^T v \tag{2.9}$$

Physically, the projector represents the component of vector v on another vector u; similarly, in MRA, the projector represents a component of a function on the function space of ϕ_{mn}

Definition 6. *Wavelet transformation for a function f in $L^2(R)$ is defined as*

$$\int_{-\infty}^{\infty} h\left(\frac{x-b}{a}\right) f(x)\,\mathrm{d}x$$

where $a \in \mathbb{R}\backslash\{0\}$, $b \in \mathbb{R}$ and $h : \mathbb{R} \to \mathbb{R}$ is function satisfying condition

$$\int_{-\infty}^{\infty} |\hat{h}(\eta)|^2 / |\eta|\,\mathrm{d}\eta < \infty$$

where $\hat{h}$ represents Fourier transformation of h.

The parameters $a,\ b$ can be discretized, and h can be chosen from a family of functions defined by

$$h_{mn} = 2^{-m/2} h(2^{-m}x - nb_0) \tag{2.10}$$

$$m,\ n \in \mathbb{Z} \tag{2.11}$$

We want h_{mn} to be orthogonal for fixed m. The key idea behind MRA is to find a family of multiple orders, such that the span of all functions in the family of a given order is always a subset of the span of all functions belonging to orders higher than that. More formally, if ϕ_{mn} is family of functions with order m and V_m represents space spanned by ϕ_{mn}, then it must follow that

$$V_m \subset V_{m+1}\ \forall m\ \in\ \mathbb{Z} \tag{2.12}$$

We can understand this using a very simple example. Consider function

$$f_0(x) = \begin{cases} 1: & x \in 0 < x \leq 1 \\ 0: & x \notin 0 < x \leq 1 \end{cases}$$

In terms of the h_{mn} formalism, we have $f_0(x)\ =\ h_{00}$ and $f_0(x-n)\ =\ h_{0n}$

If we want to create a staircase function with two steps, as shown below. It can be represented as

$$s_0(x)\ =\ f_0(x) + 2f_0(x-1)\ =\ h_{00} + 2h_{01}$$

Now we will try to perform the same set of operations for h_{1n}, i.e based on equation (2.11) $h_{1n} = \frac{h_{00}(x/2-n)}{\sqrt{2}}$ or simply-

$$f_1(x) = \begin{cases} 1/\sqrt{2}: & x \in 0 < x \leq 1/2 \\ 0: & x \notin 0 < x \leq 1/2 \end{cases}$$

$$s_1(x)\ =\ f_1(x) + 2f_1(x-1/2)\ =\ h_{10} + 2h_{11}$$

If we want recreate $s_0(x)$ just using the functions from h_{1n} only, we can do that simply by

$$s_0(x)\ =\ \sqrt{2}h_{10}(x) + \sqrt{2}h_{11} + 2\sqrt{2}h_{12} + 2\sqrt{2}h_{13}$$

However, the other way, i.e, recreating $s_1(x)$ just using the functions from h_{0n} is impossible. This can be understood from the figure 2.3 that no matter what combination of coefficients we choose, we cannot split the function value between 0 and 1 since the h_{0n} basis has only a single Haar function within this range. Therefore, reiterating equation (2.12), it can be concluded that the space spanned by h_{0n} is always a subset of the space spanned by h_{1n}. This allows us to define MRA as follows-

Definition 7. *A multiresolution approximation (MRA) of function in $L^2(R)$ is a doubly infinite nested sequence of subspaces V_m of $L^2(R)$ with $V_m \subset V_{m+1} \forall m \in \mathbb{Z}$*

Since V_m is a subset of V_{m+1}, it means there exists a component of V_{m+1} disjoint to V_m which, when combined with V_m, completes the subspace. The linear algebra analogue is to find an orthogonal component to a vector v with respect to a larger space V_{m+1}. We take any element from S and project it onto v, then subtract it from S as

$$v_\perp = (I - P_u)v \tag{2.13}$$

This vector represents component of S orthogonal to v and space and span of S is completed by v and $v_\perp$

Definition 8. *Given any function f in $L^2(R)$, an MRA is orthogonal if we have the projection of f in V_m and V_{m+1} as $P_m f$ and $P_{m+1} f$ result in the space spanned by the orthogonal projector W_m defined as*

$$W_m f = P_{m+1} f - P_m f \tag{2.14}$$

satisfying conditions $V_m \perp W_m$, $W_m \subset V_{m+1}$ and $V_m \oplus W_m = V_{m+1}$

It can be proven that corresponding to orthogonal MRA, there exists a wavelet function $\psi(x)$ spanning the orthogonal component W_m

$$\psi(x) = \sum_{k=0}^{N-1} g_k \phi(2x - k)$$
$$g_k = (-1)^k h_{N-1-k}$$

The set of coefficients for W_m is known as wavelet coefficients. Furthermore, since W_m orthogonal to V_m , W_{m+1} is orthogonal to V_{m+1}, but W_m is subset of V_{m+1}, which implies that W_m is orthogonal to W_{m+1}. Inductively, it can be inferred that W_m is orthogonal to V_m for all values of m and W_n for all $m \neq n$.

2.5. Orthogonal wavelets

Orthogonality in general mathematical terms means linear independence between any two vectors, which is defined by the vanishing inner product of the two vectors. Just like orthogonal basis of vectors allows us to represent any vector in general lying in same space as a linear combination of basis vectors, we can extend the same concept of orthogonality to the functions analogously by requiring orthogonal functions ϕ_1, ϕ_2 to have $\int_R \phi_1 \phi_2 \, dr = 0$. In discrete form the same expression can be written as $\sum \frac{\phi_{1i} \phi_{2i}}{N} = 0$ which is similar to inner product of two vectors. One of the most important and computationally intense calculations

in retrieving electronic structure is the electron repulsion integral, as we stated in equation (2.6). It is easy to see that the complexity of the equation is $O(N^4)$, where N is the number of functions in the basis set, due to taking all possible combinations of four indices. Our aim is to reduce the complexity to $O(N^2)$.

In the previous section, we saw that the MRA allows us to project any real function to an arbitrary resolution. Let's say we are given a basis set $\Phi = \{\phi_1, \phi_2 ... \phi_n\}$ and we have a real function $f : \mathbb{R} \to \mathbb{R}$ which we want to approximate in terms of Φ. Denoting the approximated f as $\bar{f}$, we will have-

$$\bar{f}(x) = a_1\phi_1(x) + a_2\phi_2(x)..a_N\phi(x) \ : \ a_i \in \mathbb{R}$$

Integrating both sides on $\mathbb{R}$ with $\phi_i(x)$ for each i, we will have

$$\int \bar{f}(x).\phi_i(x)\ dx = \int a_1\phi_1(x).\phi_i(x)\ dx + \int a_2\phi_2(x).\phi_i(x)\ dx.. \int a_N\phi_N(x).\phi_i(x)\ dx \tag{2.15}$$

Denoting $\int \phi_i(x).\phi_j(x)\ dx = I_{ij}$ and $\int f(x).\phi_i(x)\ dx = f_i$ (assuming $\bar{f}(x) \approx f(x)$), we will have to solve equation

$$\begin{bmatrix} I_{11} & I_{12} & ... & I_{1N} \\ I_{21} & I_{22} & ... & I_{2N} \\ ... & ... & ... & ... \\ I_{N1} & I_{N2} & ... & I_{NN} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ .. \\ a_N \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ .. \\ f_N \end{bmatrix}$$

The overlap integral for any two basis functions $\Psi_i(x)$ and $\Psi_j(x)$ is given by

$$I_{ij} = \int_{-\infty}^{\infty} \Psi_i(x)\Psi_j(x).dx$$

The equation will be much easier to solve if the matrix I_{ij} is diagonal, or in other words, our basis Φ is orthogonal with $\int \phi_i(x).\phi_j(x)\ dx = \delta_{ij}$. In MRA, we can have an orthogonal dyadic wavelet basis if it satisfies the condition

$$\sum_{i}^{N-2j} h_i h_{i+2j} = 2\delta_{j0} \tag{2.16}$$

$$\sum_{r=1}^{N} r^{\alpha} g_r = 0 : \alpha \in \{0, 1, 2, ...N/2 - 1\} \tag{2.17}$$

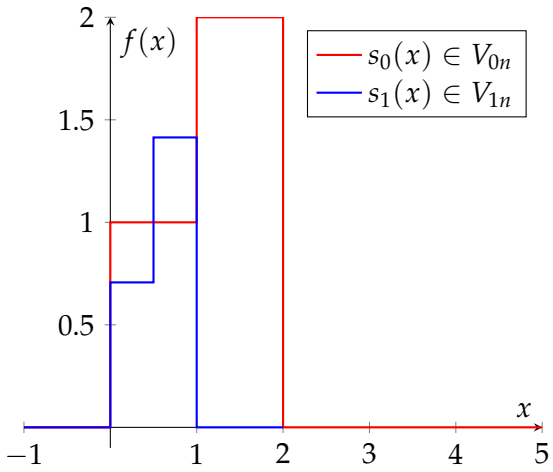


Figure 2.3.: Plots of functions belonging to V_{0n} and V_{1n} space. While s_0 belongs to both V_{0n} and V_{0n}, s_1 only belongs to V_{1n}

2.6. Tensor network methods

Tensors are a generalized representation of multi-dimensional linear data structures. The most common examples of tensors are matrices and vectors as two- and one-dimensional tensors respectively. There are various applications where tensors of higher dimensions are required. Most notably in quantum computing and condensed matter physics, tensors are used to represent entanglement between particles. Tensors are represented as $T_{i_1 i_2 \dots i_n}$ where i_j are indices, and the total number of indices n is called the rank of the tensor. A rank-one tensor (vector) can be written as v_i, and a rank-2 tensor (matrix) can be written as T_{ij}. Any rank-n tensor can be graphically represented as shown in 2.4, with the number of open edges equal to the rank-

Definition 9. *Tensor contraction : Given any two tensors $T_{i_1 i_2 .. k_1 k_2 \dots k_l i_n}$ and $U_{j_1 j_2 .. k_1 k_2 \dots k_l .., j_m}$, a tensor contraction across the indices is $k_1 k_2 \dots k_l$ between T and U is given by a new tensor*

$$\sum_{k_1} \sum_{k_2} \dots \sum_{k_l} T_{i_1 i_2 .. k_1 k_2 \dots k_l .. i_n} U_{j_1 j_2 .. k_1 k_2 \dots k_l .., j_m}$$

A common example of tensor contraction is matrix-vector multiplication, as shown in the figure and the vector inner product. The tensor network contraction always results in net-reduction of rank, i.e, we contract a set of tensors C_i to get a new tensor T, then $\sum_i rank(C_i) \geq rank(T)$. A graphical representation of tensor networks is shown in Figure 2.4.

Definition 10. *A representation of all possible combinations of a pair of tensors t, u with $t \in T$ and $u \in U$ (similar to the product of two sets), a tensor product is defined as*

$$(T \otimes U)_{i_1 i_2 .. i_m j_1 j_2 .. j_n} = T_{i_1 i_2 .. i_m} . U_{j_1 j_2 .. j_n}$$

The tensor product is distributive but not commutative. The new tensor has a rank sum of the two initial tensors, and graphically can be represented by just aligning the two tensors across the indices. In this case, no dimensionality reduction occurs. If the tensors are state matrices/vectors, then the tensor product implies that the two distributions are uncorrelated and hence independent of each other. Quantum particles exhibit entanglement, which makes them fundamentally different from their classical counterparts. Through entanglement, the states in which quantum particles can exist are correlated with each other. One of the most prominent examples is Bell's state, which has four possible entanglements between two particles. To understand, let's imagine two particles (say photons) which, on measurement, can produce either outcome 0 or 1 (vertically polarized or horizontally polarized). Assuming the particles have equal probabilities of both outcomes, we can represent states of each particle before measurement as

$$|\Psi\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

Where $|0\rangle, |1\rangle$ are the tensor representation of states given by

$$|0\rangle = [1, 0]\ ;\ |1\rangle = [0, 1]$$

Intuitive explanation

In the theory of probability, the joint distribution of two independent random variables is given by taking the direct multiplication of individual probabilities as

$$P(x, y) = P(x)P(y)$$

The same can be analogously applied to the tensor representation of qubits, which means that if there are two uncorrelated qubits with individual states represented by tensors $|\Psi_1\rangle$ and $|\Psi_2\rangle$, their joint state is given by the tensor product-

$$|\Psi_1\rangle \otimes |\Psi_2\rangle$$

For example, given two qubits $|\Psi_1\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$ and $|\Psi_2\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$
The joint state will be

$$\begin{aligned}\Psi &= \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + |1\rangle}{\sqrt{2}} \\ &= \frac{|0\rangle\,|0\rangle + |0\rangle\,|1\rangle + |1\rangle\,|0\rangle + |1\rangle\,|1\rangle}{2}\end{aligned}$$

Often when describing the state of a multiple-qubit system of the form $|\Psi_1\rangle \otimes |\Psi_2\rangle$, we drop the $\otimes$ notation and simply write $|\Psi_1\Psi_2\rangle$ (for example, $|0\rangle \otimes |0\rangle$ can be written as $|00\rangle$). The general expression of a 2-qubit system can be given by

$$\Psi_{ent} = a\,|00\rangle + b\,|01\rangle + c\,|10\rangle + \sqrt{d}\,|11\rangle \tag{2.18}$$

with $a, b, c, d \in \mathbb{C}$ and $|a|^2 + |b|^2 + |c|^2 + |d|^2 = 1$. In case the two particles are entangled: for example, either both of them are zero or both of them are 1 with equal likelihood, we don't expect all possible outcomes in the (2.18) to appear, and the joint state will be given by

$$\Psi_{ent} = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

or more generally, when likelihoods are not equal as

$$\Psi_{12} = a\,|00\rangle + e^{i\theta}\sqrt{1 - |a|^2}\,|11\rangle$$

For any 2-qubit state, we need to store four values corresponding to each possible state (3, considering normalization). However, for entangled states, we just need to store three values: 2 for probabilities and phase, and 1 for one of the four possible entangled states, which can be coded as a single-digit integer from 0 to 3. This feels like a minute difference, but since the number of possible states scales as 2^n for n-qubits (or more generally d^n for dubits), for a large system with a general expression for the state

$$\Psi = \sum_{i_1 i_2..i_n}^{\{0,1\}} p_{i_1 i_2..i_n}\,|i_1 i_2...i_n\rangle$$

We require 2^n probability values to be stored. On the other hand, for a maximally entangled system with n qubits, for example

$$\Psi_{ent} = \frac{|000..0\rangle + |111..1\rangle}{\sqrt{2}}$$

We still need just three values. Therefore, it makes sense to have a more compact way of storing quantum states' information, especially for entangled systems. Using **Matrix product state** [2] or MPS, we can store entangled qubits efficiently using a tensor train network-

Definition 11. *Given a quantum multi-body system with d independent (orthogonal) states and n particles, the matrix product state or MPS representation of the system is given by*

$$|\Psi\rangle = \sum_{s_1 s_2..s_n}^{\{0,1\}} tr[A_1^{s_1} A_2^{s_2}...A_n^{s_n}]\,|s_1 s_2...s_n\rangle$$

with $A_j^{s_i}$ being $c \times c$ matrices (rank 2 tensors) for some constant c and for state $s_i \in \{0, 1, ...d-1\}$ at the j_{th} position. For the translationally invariant case, we have $A_1^{s_i} = A_2^{s_i} .. = A_n^{s_i}$ for all s_i and the MPS simplifies to

$$|\Psi\rangle = \sum_{s_1 s_2 .. s_n}^{\{0,1\}} tr[A^{s_1} A^{s_2} ... A^{s_n}] \, |s_1 s_2 ... s_n\rangle$$

It can be proved that MPS can be used to represent any quantum state, and the representation requires a spatial complexity of $\mathcal{O}(ndc^2)$ against d^n required earlier. The diagrammatic representation of a matrix product state is shown in figure 2.5

MPS representation of some well-known quantum states is given below

- Greenberger–Horne–Zeilinger state

  $$|\Psi\rangle = \frac{|0\rangle^{\otimes n} + |1\rangle^{\otimes n}}{\sqrt{2}}$$

  This is a translationally invariant state, and we have

  $$A^0 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}; A^1 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

  It can be easily verified that

  $$A^0.A^0 = A^0 \; ; \; A^1.A^1 = A^1 \; ; \; A^1.A^0 = A^0.A^1 = 0$$

  leading to the only non zero terms in the MPS being $(A^0)^n$ and $(A^1)^n$ corresponding to states $|0\rangle^{\otimes n}$ and $|1\rangle^{\otimes n}$ as expected.

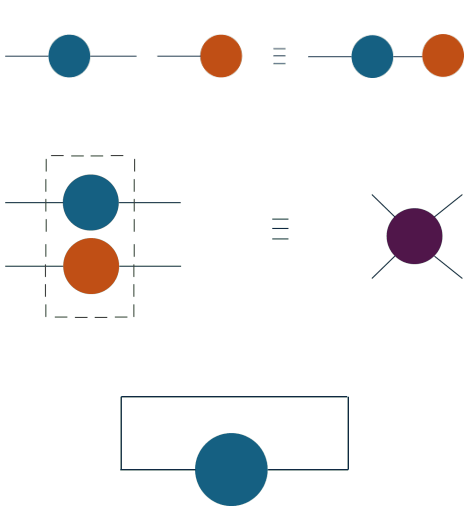
- W - state

  $$|\Psi\rangle = \frac{|001\rangle + |010\rangle + |100\rangle}{\sqrt{3}}$$

  This state requires a different matrix for each state, unlike the previous case, as

  $$A_1^0 = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}, A_1^1 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}; A_2^0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, A_2^1 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}; A_3^0 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, A_3^1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

  A_2^0 is just identity, so $A_1^i A_2^0 A_3^j = A_1^i A_3^j$. $A_1^0 A_3^0 = 0$ and $A_1^0 A_3^1 = A_1^0$ with $tr(A_1^0) = 1$. Similarly, $A_1^1 A_2^0 A_3^0 = A_3^0 = A_1$ and $A_1^1 A_2^0 A_3^1 = 0$, leading to the only non-zero terms of form $|i0j\rangle$ being $|001\rangle$ and $|100\rangle$. In the same way, it can be shown that the only other term with non-zero trace of MPS will be $|010\rangle$



$$\mathbf{A}_{ij}.\mathbf{u}_j = \sum_j \mathbf{A}_{ij}\mathbf{u}_j = \mathbf{v}_i$$

$$\mathbf{A}_{ij} \otimes \mathbf{B}_{kl} = \mathbf{C}_{ijkl}$$

$$tr(\mathbf{A}_{ij}) = \sum_{i=j} \mathbf{A}_{ij}$$

Figure 2.4.: Some common tensor operations. (from top to bottom) i) simple tensor contraction as matrix vector multiplication, ii) tensor product of two rank-2 tensors to generate a rank-4 tensor, iii) trace of a matrix represented as self-contraction operation.

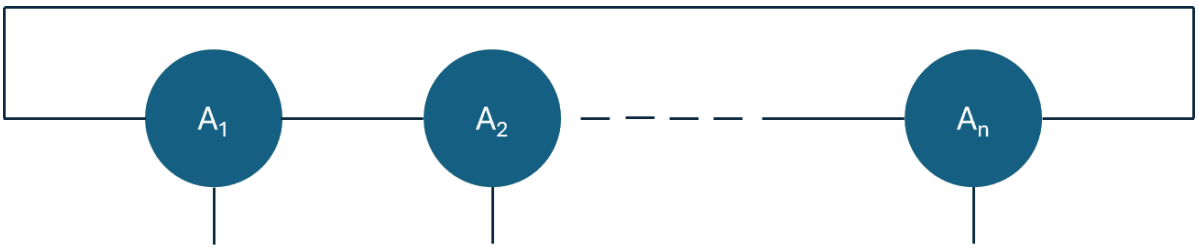


Figure 2.5.: Graphical representation of a matrix product state. Each matrix A_i can be interpreted as a rank-3 tensor, with the bottom edge of the tensor corresponding to the number of states d each dubit can occupy. The chain of the tensor represents the product of matrices.

3. Harmonium model for simulation of Quantum dots and trapped ions

3.1. Emergent behavior of inverse square law forces

In order to study the quantum mechanics of quantum dots and ion traps, we need to first understand the environment the electrons in semiconductor devices are exposed to. The form of the nuclear Hamiltonian, as we saw in equation (2.4), is fundamental and takes the inverse square law of Coulomb interaction into account, which means the force of attraction/repulsion is inversely proportional to the square of the distance between two interacting bodies. However, in a more complex system, the emergent behavior of multiple centers of Coulomb attraction/repulsion leads to a different kind of dependency on distance than the simple inverse of the square. One more fundamental concept from classical mechanics is force from a stretched or compressed spring, which is directly proportional to the distance (more precisely, the displacement) of the object from the stable position of a spring as represented in the figure 3.1. Any displacement from the center leads to oscillations of the object, causing a simple harmonic motion. While the two phenomena (Coulombic attraction and force from a spring) might appear unrelated at first, we will show how a certain arrangement of charges can give rise to simple harmonic motion as if the particle under study is attached to a spring.

Let's say there are three charged bodies of the same sign, two of which are separated by a large distance and one of them is placed exactly at the middle of the two, as shown in the figure 3.1. Assume we have $q_1 = q_2 = Q$ and $q_3 = q$. We further assume that out of three particles, only q_3 is allowed to move with a degree of freedom only along the x direction. Taking the right-hand side as the positive direction, the net force on the particle is given by

$$F = \frac{KQq}{r_1^2} - \frac{KQq}{r_2^2}$$

where K is the *electrostatic constant*

For $r_1 = r_2 = r$, the forces from either side cancel out and are equal to 0. If the q_3 is perturbed by a distance $d \ll r$ along the x direction, then the net force will be

$$F = \frac{KQq}{(r+d)^2} - \frac{KQq}{(r-d)^2} \tag{3.1}$$

$$F = -\frac{KQq.2rd}{(r^2-d^2)^2} \tag{3.2}$$

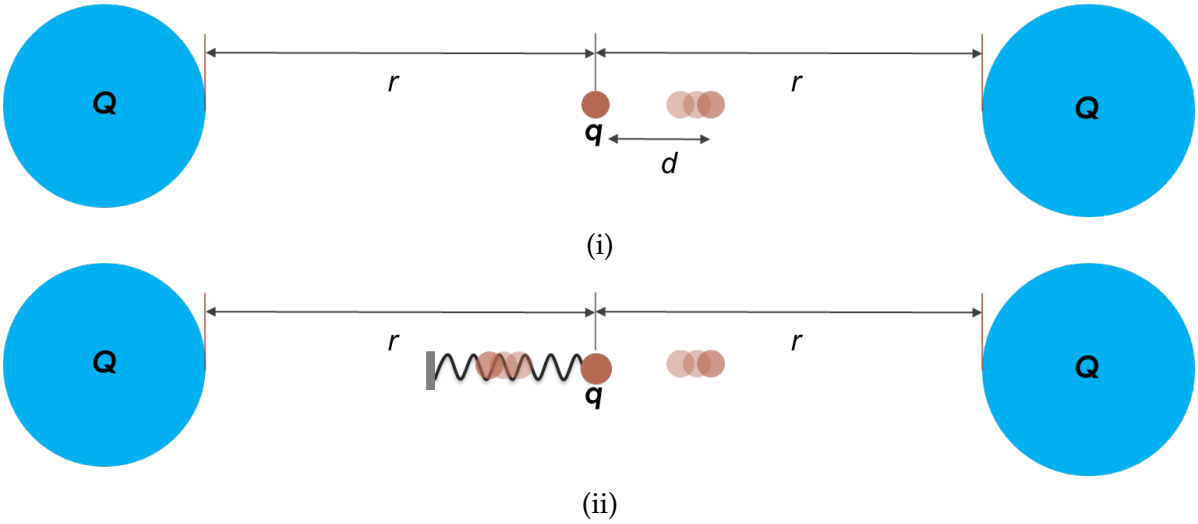


Figure 3.1.: Three charged particles with the same sign of charge placed in the arrangement

Since $d \ll r$ we can assume $(r^2 - d^2)^2 = r^4$ leading to

$$F = -\frac{KQq.2rd}{r^4} = -\frac{2KQq.d}{r^3}$$

Since r, K, Q, q are all constants, we can write

$$F = -kd$$

which is identical to a particle oscillating around the center as if a spring is connected to it. Intuitively, it is easy to see that the particle at the center is in stable equilibrium, and a perturbation of either side would increase the repulsive force in the opposite direction and try to restore the particle to the stable state. Hence, it can be concluded that there exist configurations of charged particles that can give rise to emergent behavior different than simple Coulombic interaction.

3.2. Simulation of Quantum Dots and Ion Traps

Quantum dots are nano-scale semiconductor components formed by laterally confining electrons in an electric potential [3]. Given a setup where a number of free electrons with a certain degree of translational freedom (as in a two-dimensional electron gas prepared on an AlGaN/GaN interface semiconductor [4]) and collectively exposed to an artificial electric potential instead of an atomic nucleus, it would result in a Quantum dot. To calculate corresponding eigenstates and excitation energies, we need to solve the Schrodinger equation of many-body dynamics. Quantum dots demonstrate properties similar to atoms, including

distinct emission and absorption spectra, discrete energy levels, and are also sometimes referred to as "artificial atoms" [5].

Quantum dots can be realized in semiconductor devices via layering of components with varying chemical composition, creating a potential well. The electrons from n-doped substrates are allowed to be confined in a potential well. In one of the fabrication methods, a substrate of n-doped GaAs is used as a substrate (including source and sink) and in between source and sink, a very thin sandwich of undoped $Al_xGa_{1-x}As$, $In_yGa_{1-y}As$, $Al_xGa_{1-x}As$ is placed [5]. The $Al_xGa_{1-x}As$ [1] layers act as a barrier to the potential drop created from $In_yGa_{1-y}As$. A bias potential is applied to induce movements of the electron from the source to the sink. The presence of $Al_xGa_{1-x}As$ barrier acts as a quantum tunnel, so that only a few electrons are able to reach from the source to the potential well. Inside the potential well, the trapped electrons are even less likely to make it to the sink, hence forming a quantum dot.

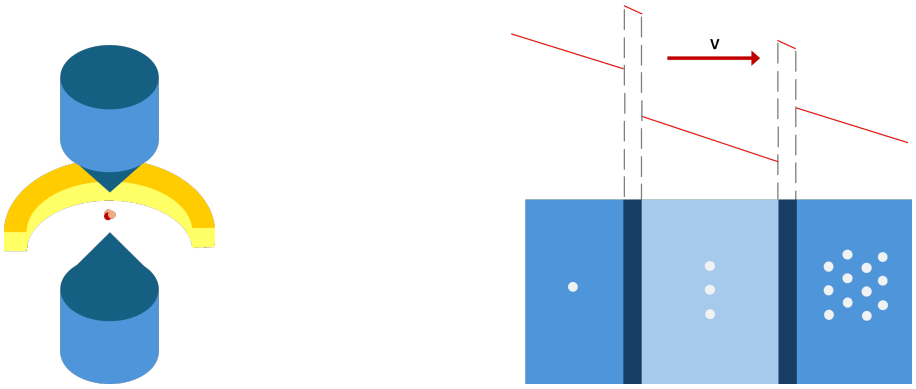


Figure 3.2.: (left to right) i) experimental apparatus for trapped ion system (also known as a Paul trap) via application of an electric field through ring potential (yellow ring figure) and static electrode potential (two blue conical figure), at the center of the electrodes there is a rough illustration of a confined ion ii) preparation of quantum dots via creating potential well using combination of p-doped and n-doped semiconductor (electron density indicated by shade of blue in the regions) and the applying bias potential (indicated by red line). The small blue circles and their numbers represent electrons and a rough estimate of their density in the regions.

Similar to the case of quantum dots, where electrons are confined in a potential well, a confinement of ions prepared to study their quantum properties is known as trapped ions. Trapped ions and electrons in quantum dots are very different systems, but they can be treated in a similar way when studying their Hamiltonian dynamics, i.e., they are also charged entities confined in a potential well exhibiting emission and absorption properties, but the method of preparation differs vastly. Unlike quantum dots, trapped ions are prepared by isolating ions from their medium, often levitating freely in a vacuum using electric fields. As an example, an ion trap of Hg^+ ions can be prepared by laser cooling mercury ions and

[1] Al : Aluminium, In : Indium, As : Arsenic, Ga : Gallium

trapping them between the electric potential well created from ring and end cap electrodes (known as Paul radio frequency trap) [6]. In the arrangement, one can use spectroscopy as well as photographic images to probe into the ion cluster, which allows us to obtain different energy levels and configurations. In recent times, trapped ions are also becoming promising candidates for quantum computing.

To come up with the Hamiltonian of a quantum dot system, we have to model i) the external potential well, ii) the electron-electron interaction, and iii) the momentum of electrons. The quadratic nature of the external potential well can be well established from the discussion in the previous section. Interestingly, we can also model electron-electron interaction as Harmonic oscillators. The free electrons in a doped semiconductor have their wave function smeared along the axis perpendicular to the plane of confinement. This distribution of charge results in saturation of the electric potential between the two electrons as their distance reduces to zero, instead of shooting up to infinity as expected from Coulomb's law [3]. We can write inter-electron potential V_{ij} in quantum dots as-

$$V_{ij} = V_0 - k(r_i - r_j)^2$$

with k a real constant and V_0 saturation potential.

To calculate momenta of electrons, we have to take the canonical momentum ($\mathbf{p} = m\mathbf{v}$) as well as electromagnetic momentum (eA/c) into account, with A being a vector quantity, Magnetic Potential. In classical electrodynamics, A is interpreted as an abstract quantity which relates to the magnetic field B as $B = \nabla \times A$. However in quantum mechanics, this abstract quantity becomes much more significant as it was shown in the Aharonov–Bohm effect that even in the absence of any electric or magnetic field, the electrons are still affected by electromagnetic potential, which can only be explained on the basis that E and B are not sufficient to describe electromagnetic phenomena, and instead one must use Electromagnetic four-potential, i.e combination of scalar electric potential and vector magnetic potential. The additional term of eA/c can be derived by writing the Lagrangian of an electron as

$$L = \frac{1}{2}m\mathbf{v}^2 + e\mathbf{v}.A - V$$

Using the momentum formula for Lagrangian $dL/d\mathbf{v}$), we get

$$\mathbf{p} = \frac{d}{d\mathbf{v}}(\frac{1}{2}m|\mathbf{v}|^2 + e\frac{\mathbf{v}.A}{c} - eV)$$
$$\mathbf{p} = m\mathbf{v} + \frac{eA}{c}$$

It can be proven more rigorously using Maxwell equations and relativistic dynamics [7]. Finally, we also have to take energy from the spin of electrons as [3]

3.3. Evaluation of Hamiltonian terms in Harmonium models using ternary Gausslets

The Hamiltonian of the Harmonium model is given by

$$H = \sum_k \frac{p_k^2}{2m_k} + \frac{1}{2} m_k x_k^2 \; + \; K \sum_{0 \le k < l \le N} |x_k - x_l|^s \tag{3.3}$$

where k, l are the indices of particles out of N and m_k, p_k represent mass and momentum respectively for position x_k. $K \in \mathbb{R}$ and $s \in \mathbb{N}$ are some constants. The corresponding equation in second quantization is given by

$$\sum_{ij} h_{ij} a_i^\dagger a_j + \sum_{i_1 i_2 j_1 j_2} V_{i_1 i_2 j_1 j_2} a_{i_1}^\dagger a_{i_2}^\dagger a_{j_1} a_{j_2} \tag{3.4}$$

$$h_{ij} = \langle \phi_i | \frac{p_k^2}{2m_k} + \frac{1}{2} m_k \omega^2 x_k^2 | \phi_j \rangle \; ; \; V_{i_1 i_2 j_1 j_2} = \langle \phi_{i_1} \phi_{i_2} | K | x_k - x_l |^s | \phi_{j_1} \phi_{j_2} \rangle$$

The corresponding steady state Schrodinger equation of this Hamiltonian has analytical solutions for the case s = 2, but we are more interested in calculating numerical approximations of h_{ij} and $V_{i_1 i_2 j_1 j_2}$. Based on the earlier discussion, we can assume that electrons occupy stable states discretized grids represented by orthogonal ternary Gausslets ϕ_i, which don't have to be exactly equal to a true solution but a sufficiently close and well-behaved function. Accordingly the expressions of h_{ij} and $V_{i_1 i_2 j_1 j_2}$ can be simplified (proof in Appendix A.2) with final expressions given by-

$$h_{ij} = \frac{\hbar^2}{2m_k} (J_{i-j}(0) - J_{i-j}(2)) + \frac{1}{2} m_k \omega^2 I_{ij}(2) \tag{3.5}$$

$$V_{i_1 i_2 j_1 j_2} = K(\; \delta_{i_2 j_2} \; . \; I_{i_1 j_1}(2) \; + \; \delta_{i_1 j_1} \; . \; I_{i_2 j_2}(2) - 2 \; I_{i_1 j_1}(1) \; . \; I_{i_2 j_2}(1) \;) \tag{3.6}$$

with

$$I_{ij}(1) := \sqrt{\pi} \left\{ \sum_{kl} c_k c_l \mu_{i+j,kl} \; e^{-(\lambda_{i-j,kl}^2 / 4\sigma)} \right\}$$

$$I_{ij}(2) := \sqrt{\pi} \left\{ \sum_{kl} c_k c_l (\frac{1}{2} + {\mu_{i+j,kl}}^2) \; e^{-({\lambda_{i-j,kl}}^2 / 4\sigma)} \right\}$$

$$J_{i-j}(0) := \sqrt{\pi} \left\{ \sum_{kl} (\lambda_{i-j,kl}/2)^2 \; . \; c_k c_l \; . \; e^{-(\lambda_{i-j,kl}^2 / 4\sigma)} \; \right\}$$

$$J_{i-j}(2) := \frac{\sqrt{\pi}}{2} \left\{ \sum_{kl} c_k c_l . \; e^{-(\lambda_{i-j,kl}^2 / 4\sigma)} \right\}$$

$$\lambda_{i-j,kl} := \alpha(k-l) + \beta(i-j) \tag{3.7}$$
$$\mu_{i+j,kl} := (\alpha(l+k) + \beta(i+j))/2 \tag{3.8}$$

The indices of form $i+j$ in $\mu_{i+j,kl}$ and $i-j$ in $\lambda_{i-j,kl}$, J_{i-j} represent reduced dependency of the electronic states on the basis functions, in a way that same of value of $i-j$ and $i+j$ map to same values in respective cases and we can iterate over just one reduced index of form $i \pm j$ instead of two independent indices i, j. Alternatively, we can also evaluate the integrals directly using numerical integration if the values of ϕ_i and $\frac{d}{dx}\phi_i$ are already calculated on discrete points. However, the accuracy is expected to be low in this method.

$$\begin{aligned} h_{ij} &= \int_{-\infty}^{\infty} \frac{d}{dx}\phi_i(x)\frac{d}{dx}\phi_j(x)\ .\ dx + \frac{1}{2}\int_{-\infty}^{\infty} \phi_i(x)x^2\phi_j(x)\ .\ dx \\ &\approx \sum_k \phi_i'(x_k)\phi_j'(x_k)\ .\ \Delta x + \frac{1}{2}\sum_k \phi_i(x_k)x_k^2\phi_j(x_k)\ .\ \Delta x \end{aligned}$$

Before evaluating Hamiltonian from equation 3.6, we need to calculate $I_{ij}(1)$, $I_{ij}(2)$, $J_{i-j}(0)$ and $J_{i-j}(2)$ for all possible pairs i, j. While both i and $j \in \mathbb{N}$ representing all possible states an electron can take in 1-d space, in the case of quantum dots, the electrons are assumed to be bounded within a finite spatial domain and hence we can restrict our range to $0 \leq i, j < N$ for some finite $N \in \mathbb{N}$. Also, based on discussions in section 4.3, we will choose $\alpha = 1$ and $\beta = 3$ in equation 3.7.

3.4. Hermite polynomial and Hermite basis function

A quantum Harmonic oscillator defined by the Hamiltonian of the form

$$\hat{H} = \frac{\hat{p}^2}{2m} + \frac{1}{2}m\omega^2x^2$$

where $\hat{H}$ is Hamiltonian and $\hat{p} = -i\hbar\frac{d}{dx}$ and we have to solve ψ in the equation $\hat{H}\psi = E\psi$. The equation directly translates to a system where a quantum particle is left in a parabolic potential well. The general term for a solution in terms of the Physicist's Hermite polynomial $H_n(x)$ is given by

$$\psi_n(x) = \mathcal{N}k^{1/4}e^{-k^2x^2/2}H_n(kx) \tag{3.9}$$

with

$$\begin{aligned} k &= m\omega/\hbar \\ \mathcal{N}_{\backslash} &= \frac{1}{\sqrt{\sqrt{\pi}2^n n!}} \\ H_n(x) &= (-1)^n e^{x^2}\frac{d^n}{dx^n}(e^{-x^2}) \end{aligned}$$

Hermite polynomials have the following properties

$$\int_{-\infty}^{\infty} H_m(x)H_n(x)e^{-x^2}.dx = \frac{\sqrt{\pi}}{\mathcal{N}^2}\delta_{mn} \tag{3.10}$$

which implies the basis function in 3.9 forms an orthogonal basis. Another interesting property of Hermite functions when multiplied with $e^{-x^2/2}$ they are eigenfunctions of Fourier transform, i.e

$$\mathcal{F}\{e^{-x^2/2}H_n(x)\}(\omega) = (-i)^n e^{-\omega^2/2}H_n(\omega) \tag{3.11}$$

Since Hermite functions can directly express solutions in the Harmonic quantum system, they can be used to represent solutions of the Harmonium model in 3.3.

4. Method

4.1. Problem statement

In the previous section, we saw how to map the electronic Hamiltonian terms in second quantization into electron interaction tensors. We also saw that the Coulomb tensor describing two-electron interaction can incur a huge computational load that is not properly diagonalized due to its $O(n^4)$ complexity. We aim to discover methods to generate continuous orthogonal basis functions to solve electronic structure problems. To understand the importance of the objective, we will go through the previous approaches to reduce the computation time of electronic structure calculation. We will also briefly examine the tensor network methods to optimize these calculations and how an orthogonal basis can further enhance the tensor algorithms. Later, we will use the Harmonious model described in the previous section as a benchmark to determine the accuracy and efficiency of our new basis.

Even in the earlier approaches to tackle the electron repulsion integral, the usefulness of the Gaussian function basis was realized [8]. The proposed methods included variations of optimizations to choose functions from the basis set, using representing different components of the electronic hamiltonian as auxiliary functions [9], using hybrid algorithm schemes consisting of multiple schemes and optimal placement of contraction steps [10] [11] [12]. When the Density Matrix Renormalization Group algorithm, or DMRG in short, was discovered, it provided a more rigorous and generalized framework centered around finding the ground state of the electron system using a tensor network.

In this section, we will describe the Gausslet basis, which allows us to have an orthogonal and complete basis. After that, we will attempt to fit Hermite basis functions using Gausslet basis and use it as our benchmark to evaluate the performance of the Gausslet basis based on the discussion in 2.15. If we represent $H_n(x)e^{-x^2/2} = \sum g_i \phi_i(x)$ (ignoring constant term $\mathcal{N}$) where $\phi_i(x)$ are orthogonal Gausslet basis functions, we will have

$$g_i = \int_{-\infty}^{\infty} H_n(x)e^{-x^2/2}.\phi_i(x).\mathrm{d}x \tag{4.1}$$

and we aim to find g_i with highest accuracy.

4.2. Direct optimization method to get orthogonal Gausslet

For the Gausslet basis, the expression of the overlap integral can be simplified as

$$I_{r_1 r_2} = \sum_{kl} c_k c_l e^{-(x-ak-\beta i)^2} e^{-(x-al-\beta j)^2}$$

Based on the orthogonality criteria as mentioned, we expect the above integral to be zero (or infinitesimally small) for all except $i = j$

$$\int_R \phi(x, \beta_k)\phi(x, \beta_l)\, \mathrm{d}x = \delta_{kl} \tag{4.2}$$

$$\int_R \phi(x, \beta_j)\phi(x, \beta_j)\, dx = \int_R \phi(x, \beta_j)^2\, dx = 1 \tag{4.3}$$

as a normalization condition. The equation (4.2) can be re-written as

$$\int_R \sum_{ij} c_i c_j e^{\frac{-(x-\alpha_i-\beta_k)^2}{2}} e^{\frac{-(x-\alpha_j-\beta_l)^2}{2}}\, dx = \delta_{kl}$$

Multiplication of any two Gaussian functions separated by distance results in another Gaussian centered at the midpoint of the original Gaussian (detailed proof in Appendix A.1). If we denote **distance between centers of Gausslet as multiples of** d and **distance between centers of Gaussian as multiples of** a. Cleary we have $\alpha_i = ia$ and $\alpha_j = ja$. Using result from A.1 we get -

$$\begin{aligned} \lambda &:= (i-j)\alpha + \beta(k-l) \\ \mu_{ik} &:= i\alpha + k\beta \\ \mu_{jl} &:= j\alpha + l\beta \end{aligned}$$

$$\int_{-\infty}^{\infty} \sum_{ij} c_i c_j e^{-(\frac{\lambda}{2})^2} e^{-(x-\frac{(\mu_{ik}+\mu_{jl})}{2})^2}\, dx = \sum_{ij} c_i c_j e^{-(\frac{\lambda}{2})^2} \int_{-\infty}^{\infty} e^{-(x-\frac{(\mu_{ik}+\mu_{jl})}{2})^2}\, dx$$

Integration over $\mathbb{R}$ of any Gaussian doesn't depend on center shift and we substitute $\int_R e^{-(x-\frac{(\alpha_i+\alpha_j)}{2})^2}\, dx = \int_R e^{(\frac{-x^2}{2})}\, dx = \sqrt{\pi}$. For $k \neq l$, it must satisfy

$$\sqrt{\pi} \sum_{ij} c_i c_j e^{-(\frac{\lambda}{2})^2} = \sum_{ij} c_i c_j e^{-(\frac{\lambda}{2})^2} = 0 \tag{4.4}$$

for (4.3) the condition will be

$$\sum_{ij} c_i c_j e^{-\frac{((i-j)\alpha)^2}{2}} = 1 \tag{4.5}$$

For simplicity, we will assume $a = 1$ and our objective is to solve (4.4) and (4.5) for the simplest nontrivial case of Gausslet as in (2.2) with c_0, c_1, β as variables. We have a non-linear

set of equations which are hard to solve, but for the case in (2.2) we can simplify things. To generate all possible combinations of coefficients, we define vectors

$$C = \begin{bmatrix} c_1 \\ c_0 \\ c_1 \end{bmatrix} \quad C^T = \begin{bmatrix} c_1 & c_0 & c_1 \end{bmatrix}$$

Then we define the outer product as-

$$M = CC^T = \begin{bmatrix} c_1^2 & c_1c_0 & c_1^2 \\ c_1c_0 & c_0^2 & c_1c_0 \\ c_1^2 & c_1c_0 & c_1^2 \end{bmatrix}$$

The matrix M has three distinct values of elements c_0^2, c_1c_0, c_1^2. Let's name them l_1, l_2, l_3. (though we have dependency $l_1l_3 = (l_2)^2$, we will take care of it later). We will now define new matrix $\bar{M}$ to get the expression in (4.4) and (4.5) as

$$\bar{M}_{ij} = e^{(-(\beta+\alpha(i-j))^2)} M_{ij} \tag{4.6}$$

Our equations can be formed by simply summing up all the elements of matrix M, and if we ignore the value of β for now, we will obtain a linear equation with three variables (l_1, l_2, l_3). Assuming all three variables are independent, we need three equations for a nontrivial solution. The first most obvious one is the normalization condition (4.5) for the pair of Gausslets with distance between their centers $\lambda = 0$. The second two equations can be obtained by requiring condition (4.5) for two more pairs with $\lambda = \beta$ and $\lambda = 2\beta$. If we define the matrix M from (4.6) as a function of central distance λ, then we will have three equations as

$$\sum_{ij} \bar{M}_{ij}|_{\lambda=0} = 1$$
$$\sum_{ij} \bar{M}_{ij}|_{\lambda=\beta} = 0$$
$$\sum_{ij} \bar{M}_{ij}|_{\lambda=2\beta} = 0$$

We will solve the equations above in the following steps

- Choose a guess value of β
- Solve the linear equation (4.2) for l_1, l_2, l_3
- Tune β such that $|l_1l_3 - l_2^2|$ in minimum (or zero)
- Retrieve c_1 and c_0 from l_1 and l_3

The above equation was solved in the Python notebook, and it was found that the values of the three variables (c_0, c_1, β) were

$$c_0 = 1.3642541146148037$$
$$c_1 = -0.23540487848582725$$
$$\beta = 2.877551020408163$$

The approach discussed above can be extended to a Gausslet composed of more than three Gaussians (and hence more than three variables), with orthogonality conditions defined for much larger λ as

$$\sum_{ij} \bar{M}_{ij}|_{\lambda=0} = 1$$
$$\sum_{ij} \bar{M}_{ij}|_{\lambda=\beta} = 0$$
$$.....$$
$$\sum_{ij} \bar{M}_{ij}|_{\lambda=n\beta} = 0$$

Unlike the simple constraint of form $l_1 l_3 = l_2^2$, we will have rather cumbersome and numerous constraints to solve simultaneously. As the number of Gaussians directly corresponds to the size of wavelet coefficients in 2.5, we will see in our next section that an orthogonal Gausslet basis with more than 3 Gaussians can be constructed using the Unitary circuits method.

4.3. Unitary circuit method to obtain orthogonal Haar wavelets

In this section, we will describe the work by Glen Evenbly and Steven R White from their paper "Representation and design of wavelets using unitary circuits" [13] in simplified and concise form, and we will cover only the topic relevant to our thesis. Given a wavelet defined by scaling sequence **h** and wavelet sequence **g** such that

$$\mathbf{h} = [h_1, h_2, ...h_N] \; ; \; \mathbf{g} = [g_1, g_2, ...g_N]$$
$$g_i = (-1)^i h_{N-i} \implies \mathbf{h}.\mathbf{g} = 0$$

for a scaling factor m one must have

$$\sum_{r=1}^{N-mj} h_r h_{r+mj} = 0 \forall\ j \in \{1, 2, ...N/2 - 1\} \tag{4.7}$$

$$\sum_{r=1}^{N} {h_r}^2 = 1 \tag{4.8}$$

The objective is to find all possible sets of $\mathbf{h}$ and corresponding $\mathbf{g}$. For standard basis vectors

$$\mathbf{e}^i = [e_1^i, e_2^i, ...e_N^i] : e_j^i = \delta_{ij} \tag{4.9}$$

we have $\mathbf{e}^i$ and $\mathbf{e}^j$ orthogonal for all $i \neq j$. This result will be used later, and in the next section, we find $\mathbf{h}$ for the case of dyadic wavelets, where we have $N = 2$.

4.3.1. Calculation of dyadic wavelet coefficients using Unitary transformation

The simplest case with a dyadic wavelet with depth = 1 , we have $N = 2$ or $\mathbf{h} = \{h_1, h_2\}$. Based on 4.7, we have ${h_1}^2 + {h_2}^2 = 1$ and since $h_{1+2\times1} = h_3$ doesn't exist, there is no case for 4.8. We have a straightforward solution for this trivial case: $h_1 = cos(\theta), h_2 = sin(\theta)$ for any real θ.

Analyzing the next depth with $N = 3$, we have $\mathbf{h} = \{h_1, h_2, h_3\}$ and hence there will case for both 4.7 and 4.8 as

$$h_1^2 + h_2^2 + h_3^2 = 1$$
$$h_1.h_3 = 0$$

$h_1.h_3$ means either $h_1 = 0$ or $h_3 = 0$, which leads to $\mathbf{h} = [h_1, h_2, 0]$ or $\mathbf{h} = [0, h_2, h_3]$, which ultimately reduces to case $N = 2$ as we will have $h_1^2 + h_2^2 = 0$ or $h_2^2 + h_3^2 = 0$.

Analyzing next case for N = 4, or $\mathbf{h} = [h_1, h_2, h_3, h_4]$, we have

$$h_1^2 + h_2^2 + h_3^2 + h_4^2 = 1$$
$$h_1.h_3 + h_2.h_4 = 0$$

The equation 4.3.1 can be interpreted as dot product of two orthogonal vectors $\mathbf{v}_1 = [h_1, h_3]^T$ and $\mathbf{v}_2 = [h_2, h_4]^T$. Any 2-d orthogonal pair of vectors with real values can be represented as $\mathbf{v}_1 = r_1[cos(\theta), sin(\theta)]^T$ and $\mathbf{v}_1 = r_2[-sin(\theta), cos(\theta)]^T$ for any real r_1, r_2. Each of these vectors can be obtained by unitary transformation of $r_1\mathbf{e}^1 = r_1[1, 0]$ and $r_2\mathbf{e}^2 = r_2[0, 1]$ as

$$\mathbf{v}_1 = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} r_1 \\ 0 \end{bmatrix} \quad ;\ \mathbf{v}_2 = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} 0 \\ r_2 \end{bmatrix} \tag{4.10}$$

If we represent unitary transform as $U(\theta)$ we have $\mathbf{v}_1 = U(\theta)\mathbf{e}^1$ and $\mathbf{v}_2 = U(\theta)\mathbf{e}^2$. This simple methodology allows us to generate all possible sets $\mathbf{h} = [h_1, h_2, h_3, h_4]$ for dyadic wavelets by varying a single parameter θ. We will see that the same can be extended for depths higher than $N = 4$.

Definition 12. *Direct sum $\oplus$ operation between two vector spaces V_m, V_n as $\oplus : V_m \times V_n \rightarrow V_{m+n}; \boldsymbol{u} \oplus \boldsymbol{v} = [u_1, u_2, ...u_m] \oplus [v_1, v_2, ...v_n] = [u_1, u_2, ...u_m, v_1, v_2, ...v_n]$. Also let $\boldsymbol{0} = [0]$ a 1-d vector.*

Then if any vector $\mathbf{h}$ satisfied the conditions in 4.7 and 4.8, then it can be easily verified that $[0, h_1, h_2, ..., h_n, 0] = \mathbf{0} \oplus \mathbf{h} \oplus \mathbf{0}$ also satisfies the same conditions for dyadic wavelets. For verification of orthogonality, we can define **shifted inner product** as

Definition 13. *Given any vector $\boldsymbol{v} = [v_1, v_2, ..., v_n]$, let us denote trimmed vectors from $\boldsymbol{v}$ as*

$$\begin{aligned}
\boldsymbol{v}[s:] &= [v_s, v_{s+1}, ..., v_n] \\
\boldsymbol{v}[:s] &= [v_1, v_2, ..., v_s] \\
\boldsymbol{v}[-s:] &= [v_{n-s}, v_{n-s+1}, ..., v_n] \\
\boldsymbol{v}[:-s] &= [v_1, v_2, ..., v_{n-s}]
\end{aligned}$$

Definition 14. *Given any two vectors v_1, v_2 in vector spaces R^m, R^n respectively, then the **shifted inner product** between u and v, denoted by notation $[u.v]_s$ with shift $s \in \mathbb{N} \cup \{0\}$ and $s \leq min(m, n)$ is defined as inner product of vectors $u[-s:].v[:s]$. The shifted inner product between two vectors can be represented with an **intuitive symbolic notation** by writing them in row form and aligning them vertically as*

$$\begin{aligned}
[\boldsymbol{u}.\boldsymbol{v}]_s = &[u_1, u_2 \,..\, u_s,\ u_{s+1} \,..\, u_m] \\
&\quad [v_1,\ v_2\ ..\ \ v_{m-s}, v_{m-s+1} \,..\, v_n]
\end{aligned}$$

with the second operand vector displaced by s number of positions, such that the aligned vertical elements correspond to the inner product that takes place

Below we have performed verification of 4.7 for the case N = 4 while using the pictorial representation of the shifted inner product (as described in definition 14) on $\mathbf{h}$ with different multiples of the dilation factor (2) applied as a shift. [1]

$$\begin{aligned}
&\textbf{Case : } [\mathbf{h}.\mathbf{h}]_2 \\
&[0, h_1, h_2, h_3, h_4, 0] \\
&\qquad [0, h_1, h_2, h_3, h_4, 0] \\
&= h_2.0 + h_3.h_1 + h_4.h_2 + 0.h_3 \\
&= h_3.h_1 + h_4.h_2 = 0
\end{aligned}$$

[1] Blue and purple colors are just added for readability and have no other significance

Case : $[\mathbf{h}.\mathbf{h}]_4$

$$[0, h_1, h_2, h_3, h_4 0]$$
$$[0, h_1, h_2, h_3, h_4, 0]$$
$$= h_4.0 + h_2.0 = 0$$

It can also be proven analytically: let us define $\bar{\mathbf{h}} = \mathbf{0} \oplus \mathbf{h} \oplus \mathbf{0}$, given $\mathbf{h}$ already satisfies 4.7. We will have $\bar{h}_1 = \bar{h}_{N+2} = 0$ and $\bar{h}_i = h_{i-1}$ for $1 < i < N+2$. Substituting in 4.7 for $\bar{\mathbf{h}}$ with dilation factor $m = 2$

$$\sum_{r=1}^{N+2-2j} \bar{h_r}\bar{h}_{r+2j} = 0.\bar{h}_{2j} + \sum_{r=2}^{N+1-2j} \bar{h_r}\bar{h}_{r+2j} + 0.\bar{h}_{N+2-2j}$$
$$= \sum_{r=2}^{N+1-2j} h_{r-1}h_{r+2j-1} = \sum_{t=1}^{N+2j} h_t h_{t+2j} = 0$$

Before moving on further, it will be more useful to write 4.7 by coupling adjacent values into 2-d vectors (given N for dyadic wavelet is always even) as

$$\mathbf{h} = [h_1, h_2, h_3, h_4 ... h_{N-1}, h_N] = [\mathbf{v}_1, \mathbf{v}_2, ... \mathbf{v}_{N/2}]$$

and can re-write 4.7 for dylation factor $m = 2$ as

$$\sum_{r=1}^{N/2-j} \mathbf{v}_r.\mathbf{v}_{r+j} = 0 \tag{4.11}$$

From 4.11 it clearly follows that $\mathbf{v}_1$ and $\mathbf{v}_{N/2}$ are always orthogonal. This we already verified in case $N = 4$ with $v_1 = [0, r_1]$ or $[0, h_1]$ and $v_2 = [r_2, 0]$ or $[h_4, 0]$. If we have wavelet coefficients $\mathbf{h}$ for depth N, we can obtain coefficients for depth $N/2$ by appending zeroes on either side of the original $\mathbf{h}$. But an $\mathbf{h}$ with zeroes at either end is not very useful as, effectively, it will produce the same functions with depth N. But if subject all vectors from $\mathbf{v}_1$ and $\mathbf{v}_{N/2}$ to an identical unitary transformation, we can obtain the $\mathbf{h}$ for next depth with non-zero values and still maintain the condition in 4.11. It works because an identical unitary transformation doesn't affect the inner product of two vectors. Since this scheme can be applied for any N, it means we can start with just vector $\mathbf{e}^{N/2+1}$, apply unitary transformation at each level until N and theoretically we can obtain any orthogonal dyadic wavelet basis by varying parameter θ (angle of unitary transformation) at each level.

Definition 15. ***Unitary circuit formalism****: Given a Unitary map $V_n : \mathbb{R}^n \to \mathbb{R}^n$ represented as $\mathbb{R}^n \times \mathbb{R}^n$ matrix which can be decomposed as matrix product*

$$V \equiv \prod_{i=1}^{d} (U_{i1} \oplus U_{i2} \oplus + ... U_{im_i}) \tag{4.12}$$

*with $d \in \mathbb{N}$, U_{ij} being a $c_{ij} \times c_{ij}$ unitary matrices with constraint $\sum_{j=1}^{m_i} c_{ij} = n$, a diagrammatic representation of the above decomposition is defined as **Unitary circuit diagram** as shown in figure 4.1 and the operation $v = Uv$ with u being the **input vector** and v being the **output vector** (both in $\mathbb{R}^n$) is defined as **Unitary circuit transformation***

Parameters related to a unitary circuit

- **depth**: the d in the equation 4.12 of a Unitary circuit, in other words, the total number of matrix product operators applied in the transformation
- **Level**: the i in the equation 4.12 of a Unitary circuit, in other words, the state of the unitary circuit where i_{th} matrix product operators $(U_{i1} \oplus U_{i2} \oplus +....U_{im_i})$ is applied in the transformation
- **Width**: at any given level i of the unitary circuit transformation, the smallest value of $l - k$ with $1 \leq k < l \leq n$ such that $(U_{i1} \oplus U_{i,2} \oplus +....U_{lm_i})$ can be decomposed as $(I_{k-1} \oplus U_{ik} \oplus U_{i,k+1} \oplus +....U_{il} \oplus I_{n-l+k-1})$ with I_j representing $j \times j$ Identity matrix. The width may or may not be a function of level.
- **Net width**: the width at the level equal to the depth of the circuit

The diagrammatic representation of a circuit with dilation two is shown in the figure 4.3. However, as discussed in the previous section, there are additional constraints that our wavelet basis must satisfy. We want 4.8 to fulfilled, but since we already start with vector $\mathbf{e}^{N/2+1}$ with $|\mathbf{e}| = 1$ and the resultant circuit is nothing but a giant unitary transform; this condition is already taken care of. We also want our wavelet coefficients ($\mathbf{g}$) to have zero momentum as described in 2.17, i.e

For dyadic case specifically, we have $N/2$ equations in 2.17 and $N/2$ variables (θ) in our unitary circuit, the system may have a unique solution and we can precisely satisfy the momentum constraint. However, it's non linear nature makes it difficult to determine. We will see later that the exact solution satisfying may not exist for cases with dilation factor $m > 2$, and we have to solve optimization problems such that moments are as close to zero as possible. In the table below, we performed optimization of θ for $m = 2$ and different values of N.

4.3.2. Calculation of ternary wavelet coefficients using Unitary transformation

In the previous section, we saw how conveniently we can calculate coefficients for dyadic orthogonal wavelets. In this section, we will want to repeat the procedure for ternary wavelets with $n = 3$. However, our setup was very specific for the case $m = 2$ and cannot be replicated entirely for $m = 3$. Instead, we have to drastically transform our unitary circuit and the ansatz for the unitary matrix.

To begin with, we can start with 4.11, which still holds, except v_r in $\mathbf{h}$ are a set of 3-d vectors instead of 2-d vectors. Given we are already provided with $\mathbf{h}$ at depth N (this time, N is always a multiple of 3), we would like to find coefficients for the next level with depth

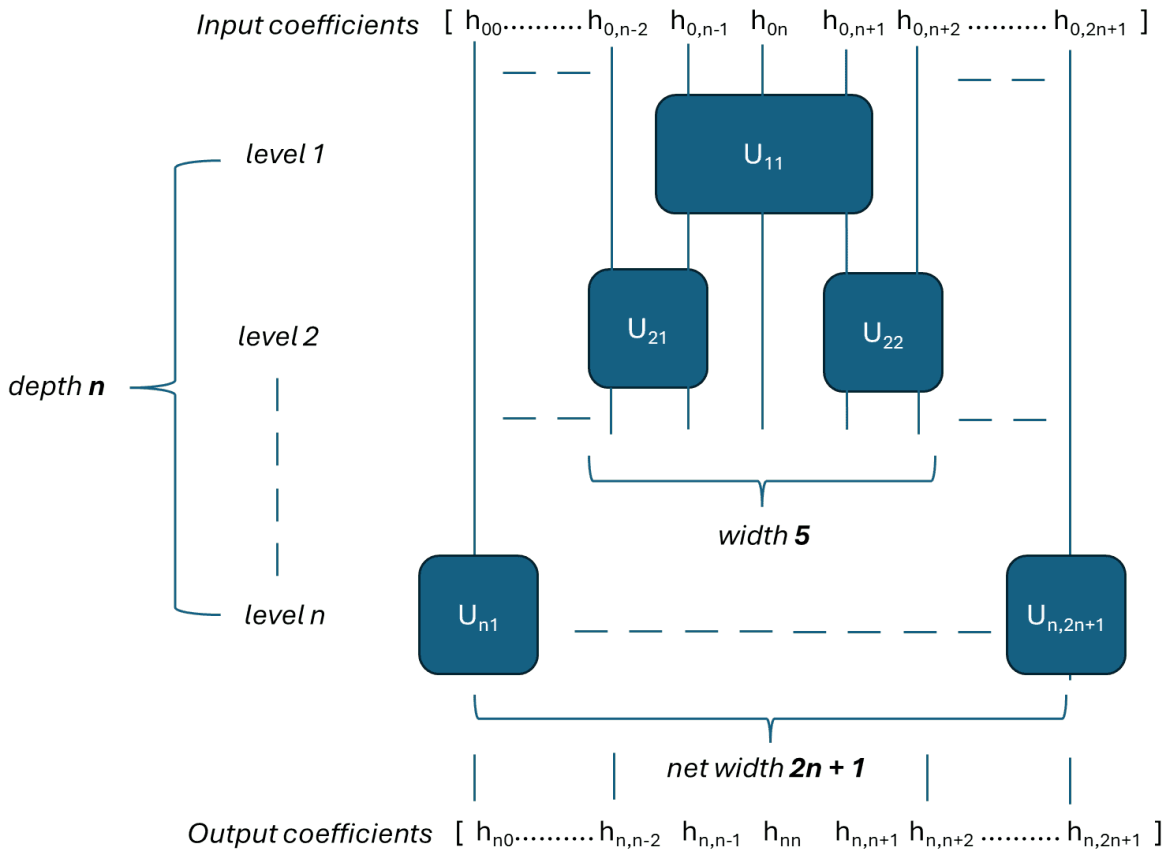


Figure 4.1.: General representation of a unitary circuit transformation for the case where the width of the unitary circuit is a function of the depth

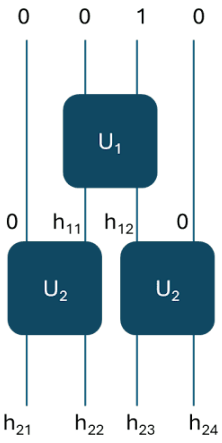


Figure 4.2.: Unitary circuit design to obtain orthogonal wavelet coefficients with dilation factor $m = 2$ and depth $N = 2$

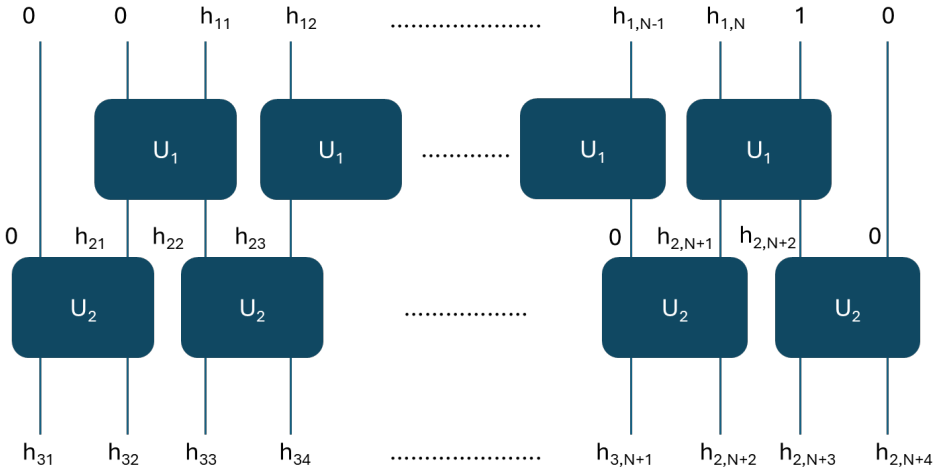


Figure 4.3.: General scheme for unitary circuit design at a given level to obtain orthogonal wavelet coefficients with dilation factor $m = 2$ and depth $N = 2$

$N + 3$. An obvious approach would be to add zero at either end just as in the case $m = 2$. Previously, we had to add two zeros, which could be easily divided into two and appended on either end, hence maintaining symmetry (which we will discuss shortly). But now, since going from N to $N + 3$, we have to add three zeros, and there is no way we can symmetrically distribute them just by appending them to the ends. It indicates a major difference from the relatively simpler $m = 2$ case. One possible way is placing one-one zero at the ends and the third zero exactly at the middle, but this is only possible if N is also divisible by two and even with this arrangement, as we will see, cannot preserve the condition in 4.11. We can verify it for simpler cases with $N = 6$ below.

$$\mathbf{h} = [\mathbf{v}_1, \mathbf{v}_2] = [h_1, h_2, h_3, h_4, h_5, h_6]$$
$$\bar{\mathbf{h}} = [0, h_1, h_2, h_3, 0, h_4, h_5, h_6, 0]$$

Case : $[\bar{\mathbf{h}}.\bar{\mathbf{h}}]_3$

$$[0, h_1, h_2, h_3, 0, h_4, h_5, h_6, 0]$$
$$[0, h_1, h_2, h_3, 0, h_4, h_5, h_6, 0]$$
$$= h_4.h_2 + h_5.h_3 = 0$$

Case : $[\bar{\mathbf{h}}.\bar{\mathbf{h}}]_6$

$$[0, h_1, h_2, h_3, 0, h_4, h_5, h_6, 0]$$
$$[0, h_1, h_2, h_3, 0, h_4, h_5, h_6, 0]$$
$$= h_6.h_1 = 0$$

We have to simultaneously satisfy $h_1.h_6 = 0$ and $h_4.h_2 + h_5.h_3 = 0$, which means at least one out of h_6, h_1 is equal to 0. It has two issues: first, we have destroyed at least one by setting it to zero instead of creating new non-zero coefficients. Secondly, it becomes apparent that we have ended up with a condition of dyadic wavelets instead of ternary wavelets with $h_4.h_2 + h_5.h_3 = 0$.

In fact, there is no arrangement of zeros that can provide ternary wavelet coefficients of depth $N + 3$ from depth N. Thankfully, it is possible to append six zeros instead of 3 and retrieve depth coefficients $N + 6$ from N. If we are already provided with coefficients for depth $N == 3 \times 1$ and $N = 3 \times 2$, we can obtain coefficients for any depth $3 \times n : n \in \mathbb{N}$. Our circuit design, however, is going to be slightly more convoluted. Let's start with $N = 3$ and adding three zeros at either end of $\mathbf{h}$ as

$$\mathbf{h} = [\mathbf{v}_1] = [h_1, h_2, h_3] \tag{4.13}$$
$$\bar{\mathbf{h}} = [0, 0, 0, h_1, h_2, h_3, 0, 0, 0] = [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3] \tag{4.14}$$

It's easy to see we will have $[\bar{\mathbf{h}}\bar{\mathbf{h}}]_3$ and $[\bar{\mathbf{h}}\bar{\mathbf{h}}]_6$ both sum up to be zero as needed in the condition. But as always, we also need to convert zeroes in 4.14 to non-zero values. We cannot apply 3×3 unitary transform on $\mathbf{h}$ in this state as both v_1, v_2 are $[0, 0, 0]$ and any transform $Uv_1 or Uv_2$ would amount to nothing. However, we can apply a 2×2 unitary transform to prepare our $\mathbf{h}$ for next steps. If we just swap h_1 and h_3 with their neighboring zeroes in 4.14, we will get

$$\bar{\mathbf{h}} = [0, 0, h_1, 0, h_2, 0, h_3, 0, 0] = [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3] \tag{4.15}$$

The two-swap operation (also known as **CNOT** in quantum computing and denoted by X) has a matrix representation $X = \begin{bmatrix}0,1\\1,0\end{bmatrix}$ and it is easy to verify its unitary nature as $XX^\dagger = XX = I$. Now we have non trivial $\mathbf{v}_1, \mathbf{v}_3$ also satisfying $\mathbf{v}_1.\mathbf{v}_3 = 0$. We can apply unitary transform of 3×3 matrix to each of $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ as per the circuit below in Figure 4.4. We can design a similar scheme starting with $\mathbf{h}$ of size $N = 6$ as shown in Figure 4.5

It is obvious that if we want an orthogonal wavelet basis which also happens to generate symmetric functions in the basis, we expect corresponding $\mathbf{h}$ to be symmetric as well. In the case of dyadic wavelets, the symmetry of the unitary circuit arose naturally, since all the identical unitary operations at each level were placed exactly at the center of the circuit. But the $\mathbf{h}$ values were not symmetric. To understand, consider $\mathbf{h}_0^4 = [0,0,1,0]$. If we take the mirror image of the entire circuit while keeping unitaries as it is, we will have $\mathbf{h}_0^4 = [0,1,0,0]$, which is orthogonal and **anti-symmetric** to the previous $\mathbf{h}_0^4$. The same is true for any $\mathbf{h}_0^N$ dyadic wavelet basis, and it can be shown that there cannot be any symmetric MRA basis for dilation factor 2. But in ternary case, since we have our basis set size in odd multiples of 3, $\mathbf{h}_0^N$ will always be symmetric, for example we have $\mathbf{h}_0^9 = [0,0,0,0,1,0,0,0,0]$ providing us freedom to generate a basis with symmetric basis. But if we apply an arbitrary unitary transformation, it will break the symmetry of $\mathbf{h}$. To prevent that, we have to use specific 3×3 unitary matrices, such that $h_i = h_{N-i}$ holds at any depth in the circuit. Let's say we have a symmetric $\mathbf{h}$, we create $\bar{\mathbf{h}} = \mathbf{0} \oplus \mathbf{h} \oplus \mathbf{0} = [0,0,0,h_1,h_2,h_3] \oplus \mathbf{h}_{6:N-6} \oplus [h_3,h_2,h_1,0,0,0]$ and we apply swap operation to get $\bar{\mathbf{h}} = [0,0,h_1,0,h_2,h_3] \oplus \mathbf{h}_{6:N-6} \oplus [h_3,h_2,0,h_1,0,0]$. To preserve symmetry of $\mathbf{h}_{6:N-6}$ given for any 3-d vector $\mathbf{v}$

$$U\mathbf{v} = v_1 U\mathbf{e}^1 + v_2 U\mathbf{e}^2 + v_1 U\mathbf{e}^3$$
$$X_3 U\mathbf{v} = v_1 X_3 U\mathbf{e}^1 + v_2 X_3 U\mathbf{e}^2 + v_1 X_3 U\mathbf{e}^3$$

where $X_3 = \begin{bmatrix}0,0,1\\0,1,0\\1,0,0\end{bmatrix}$. We must have $U\mathbf{v} = X_3 U\mathbf{v}$. A straightforward term-wise comparison would require

$$U\mathbf{e}^1 = X_3 U\mathbf{e}^3; U\mathbf{e}^2 = X_3 U\mathbf{e}^2; U\mathbf{e}^3 = X_3 U\mathbf{e}^1 \tag{4.16}$$

$U\mathbf{e}^i$ just means i^{th} column of U, therefore 4.16 implies the first and third column of U should be the reverse of each other and the second column should be symmetric. Denoting i^{th} column of U as U_i, a most primitive ansatz we can propose for U_1 and U_3 is $U_1 = [a,b,c]; U_2 = [c,b,a] : a,b,c \in \mathbb{R}$ Since U is also unitary, we must have $U_1.U_3 = 0 \implies 2ac + b^2 = 0 \implies b = \sqrt{-2ac}$. We want our $\mathbf{h}$ to be real, which means we expect U to be real as well, which means one of a or c is positive and the other is negative for b to be real. Let's take the c term to be negative, and for avoiding confusion, we will convert $c \to -c$, leading to $U_1 = [a, \sqrt{2ac}, -c]; U_2 = [-c, \sqrt{2ac}, a]$. On further imposing $|U_i| = 1$ condition, we will have

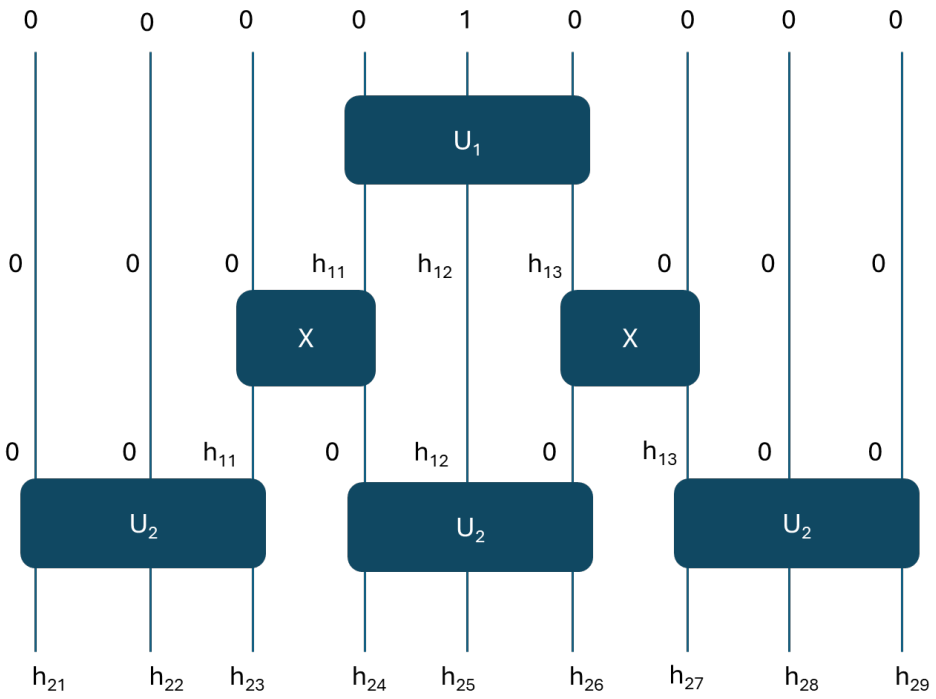


Figure 4.4.: Unitary circuit design to obtain orthogonal wavelet coefficients with dilation factor $m = 3$ and depth $N = 9$

$$U = \begin{bmatrix} cos(\theta)+1 & .. & cos(\theta)-1 \\ \sqrt{2}sin(\theta) & .. & \sqrt{2}sin(\theta) \\ cos(\theta)-1 & .. & cos(\theta)+1 \end{bmatrix}$$

Remaining U_2 can be calculated as $U_2 = \mathbf{e}^1 - (\mathbf{e}^1.U_1)U_1 - (\mathbf{e}^1.U_3)U_3$ and imposing symmetry and unitary condition, finally resulting in

$$U = \begin{bmatrix} cos(\theta)+1 & \sqrt{2}sin(\theta) & cos(\theta)-1 \\ \sqrt{2}sin(\theta) & 2cos(\theta) & \sqrt{2}sin(\theta) \\ cos(\theta)-1 & \sqrt{2}sin(\theta) & cos(\theta)+1 \end{bmatrix}$$

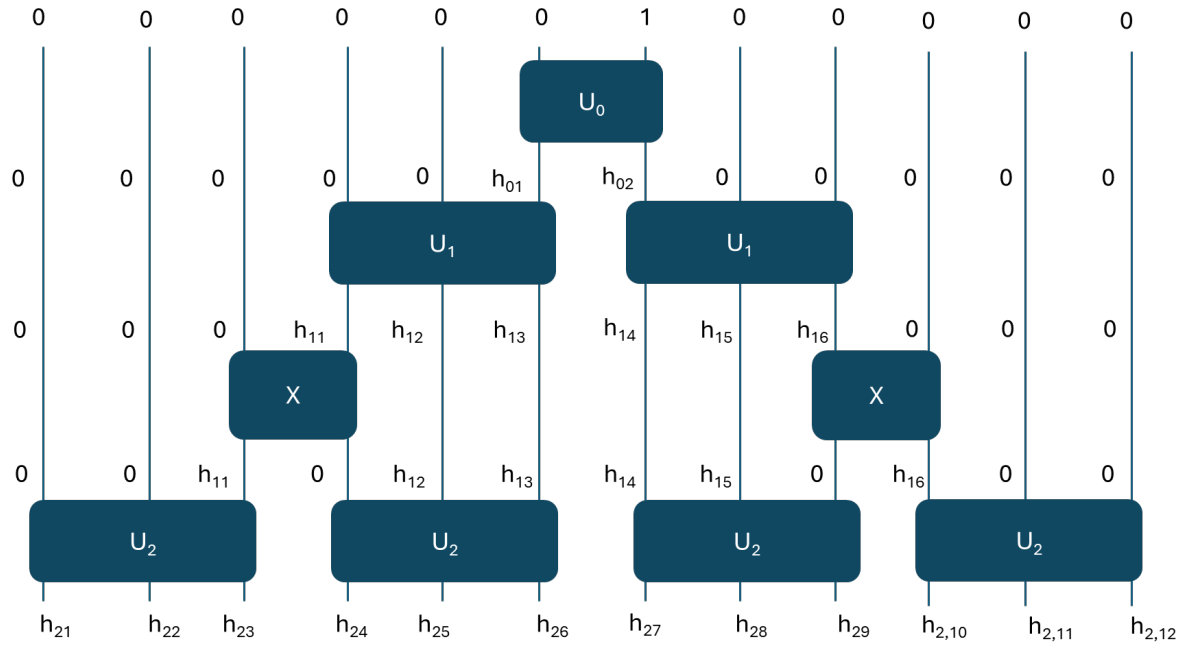


Figure 4.5.: Unitary circuit design to obtain orthogonal wavelet coefficients with dilation factor $m = 3$ and depth $N = 12$

We define the basis formed by using output coefficients from the optimized ternary unitary circuit as scaling coefficients and Haar function as the scaling function as **Unitary circuit Haar basis**

4.4. Retrieving smooth basis functions from Haar basis coefficients: Gausslets

We have seen how we can construct an orthogonal basis using the Haar function and the unitary circuit method of arbitrary depth. In principle, we can use the Haar basis to express

any function of interest using MRA, as increasing the resolution (or scaling) makes the interpolation more and more accurate to the true value of the function. However, there are multiple reasons why using them is not the best option. Firstly, for any finite scaling, the Haar basis would almost always produce a non-differentiable and discontinuous step function, which might not be suitable for further analysis. Secondly, the Haar basis has a highly localized support, which means that when plotting numerically, any minute overlap in the support would cause a sudden spike in the reconstructed function values, further deteriorating the accuracy. It is a common issue when using the Haar basis wavelet function to reconstruct function values. As an alternative, we can use hat functions, which are popularly used in finite elements, as they are continuous with overlapping support. But having an overlapping support leads to the neighboring hat functions not being orthogonal, hence a non-diagonal overlap matrix. Additionally, they still produce non-differentiable functions.

We can look back at Gaussian functions if we desire smooth and orthogonal functions. While there are other alternatives in $L^2(\mathbb{R})$ space, such as the sinc function $sinc(x) = sin(x)/x$, two reasons why Gaussians are a preferred choice are that i) the product of any two Gaussian functions produces another Gaussian function. It is a useful property, as we will see later in calculating Coulomb integrals. ii) Gaussian functions have non-oscillating and shorter tails, which is important as an ideal basis function has to be maximally local. Moreover, applying wavelet transformation can introduce additional high frequencies to the basis functions, which we want to avoid.

4.4.1. Direct inverse method

The coefficients calculated for Haar function using the unitary circuit method will not produce orthogonal basis if we replace the Haar function with a Gaussian, since the fundamental assumption behind the framework is to begin with a highly localized orthogonal function with vanishing overlap integral for any translation in multiples of the dilation factor. Gaussians are not strictly local as the overlap integral of any two Gaussian decays exponentially, but is never exactly zero for any finite distance. Linear combinations of multiple Gaussians, though, certainly can produce a basis in a way that the sum of products between different Gaussians nullifies each other. Therefore, we have to find appropriate linear combinations, or scaling and wavelet coefficients, which are orthogonal with a certain level of precision.

When finding an appropriate basis function, completeness is the most important requirement, i.e, if the span of basis functions can accurately approximate all polynomials of degree n and less, then our basis is n-complete. A quick and simple verification of completeness is if the basis set can produce a constant function (a polynomial with degree zero). For translationally invariant and symmetric basis functions, the condition reduces to

$$\sum_{i=-n}^{n} \phi(x - i\beta - t) \approx k : x \in (-sn + t, sn + t)$$

$$t, s \in \mathbb{R}$$

Where s is the support of the function, t is any arbitrary translation. Figure 4.6 illustrates this with the Gaussian basis with $\sigma = 1$ for cases $\beta = 1, \beta = 2$. We can see that for $\beta = 1$, Gaussian functions can be a great choice for a complete basis.

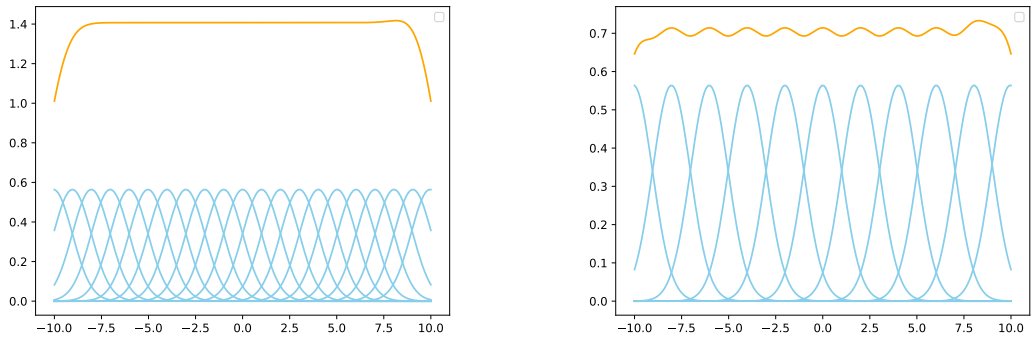


Figure 4.6.: Quick check of completeness. A complete basis which is translationally invariant and able to fit any polynomial up to degree $n > 0$ and less should necessarily (though not sufficiently) satisfy the condition that $\sum_i \phi_i(x - \beta_i)$ is constant within the support if basis functions are combined. In the left figure, we have a Gaussian basis set (blue) with the dilation factor 1 and 23 basis functions, which yield a constant function (orange) on summation in $x \in (-7, 7)$. In the right figure, we have 11 basis functions with a dilation factor 2. Though the sum of all basis sets(blue) in this case results in a close constant function, the oscillations in the summation (orange) indicate a relatively poor completeness.

Therefore, approximating Haar ternary wavelets using the Gaussian basis is a good initial approach. We can use the method described in (2.15) by calculating the overlap matrix of the Gaussians. We then have to take the inverse of the overlap matrix, and the issue is that the Gaussian overlap matrix is poorly conditioned [14] (i.e, the ratio between the lowest and highest eigenvalue of the matrix is very high). Given a matrix A the condition number is defined as

$$\kappa(A) = ||A||.||A^{-1}||$$

where $||A||$ is the induced norm of matrix A determined by

$$||A|| = max_v(\frac{|A.v|}{|v|}) = \frac{\sigma_{max}(A)}{\sigma_{min}(A)}$$

Where $\sigma(A)$ are the singular values of A. When A is a hermitian matrix, the singular values are just absolute values of corresponding eigenvalues, and the condition number can be written as the ratio $|\lambda_{max}|/|\lambda_{min}|$. For matrices close to being singular, we expect at least one eigenvalue very close to zero, since the determinant of a matrix is given by the product of all eigenvalues, as for at least one $i \in 1, 2, ..n$-

$$det(A) = \lambda_1\lambda_2 \ldots \lambda_n \approx 0 \implies \lambda_i \approx 0$$

It means the condition number can explode as a matrix approaches singularity. As a result, a poorly conditioned matrix is undesirable due to high inaccuracy in the results when attempting to solve the inverse of the matrix. Nevertheless, the Gaussian overlap matrix can still get fairly accurate results by incurring additional computational resources. In the 4.7, we see that Gaussian overlap matrices can still produce Gausslets very close to the expected one.

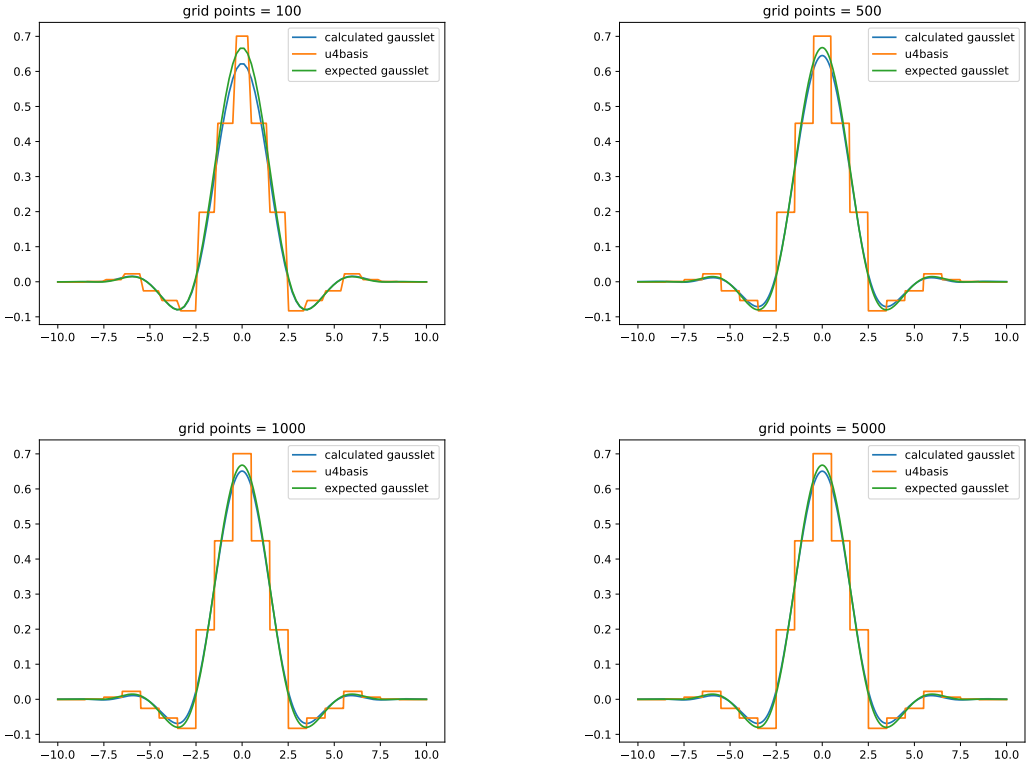


Figure 4.7.: Comparison of Gausslet calculated from direct inverse method, expected Gausslet (from Steven White's paper), and the Unitary circuit Haar basis function. We can see that even with a small grid size, the direct inverse method can yield a Gausslet that is very close to the expected one.

4.4.2. Discretization-based approximation method

If we want to obtain the same results with fewer computations, as an alternative, we can replace the true overlap matrix with an approximation with a lower condition number. Sparser and diagonal matrices generally have better condition numbers than denser matrices. A straightforward approach is to set the overlap integral value of all pairs of Gaussians to zero except the immediate neighbors, or more generally, set the value of all pairs of Gaussians

Table 4.1.: Values of first four Gausslet coefficients obtained via direct inverse method for different levels of discretization (represented by n in c_n)

c_{actual}	c_{100}	c_{500}	c_{1000}	c_{5000}
0.3503355455763915	9.31412908e-01	9.68065664e-01	0.97743808	0.97707646
0.2653497528194342	-8.20676929e-03	1.02097902e-02	0.01462182	0.01490235
-0.009493682948873735	2.37047851e-03	-2.95337613e-03	-0.00417733	-0.00430602
-0.08104201303620272	-3.06368236e-04	7.68931357e-04	0.00101405	0.00104849

having a distance greater than a certain value. Another approach is to map Gaussian basis functions into discontinuous Haar basis-like functions, and find the combination of these vectors that maximally fits the Unitary circuit Haar basis, which can also be mapped into a vector. To understand the steps, we require the following formal definitions-

Definition 16. *A basis domain is any closed interval* $X = [l_1, l_2]; l_1, l_2 \in \mathbb{R}$

Definition 17. *Given a basis domain, an Equidistant sample* $E_n(X) = \{e_1, e_2..e_i..e_n\}$ *is a subset of* X *such that* $n \in \mathbb{N}, e_1 = l_1, e_n = l_2, e_{i+1} - e_i = (l_1 - l_2)/(n-1)$ *for all* $1 \leq i \leq n-1$

Definition 18. *Given a basis function* ϕ *in function space* $\mathcal{F}$ *and a basis domain* X*, the vector transformation* $V_n : \mathcal{F} \to \mathbb{R}^n$ *is defined* $V_n^X(\phi) = [\phi(E_n(X)_1), \phi(E_n(X)_2), ..., \phi(E_n(X)_n]^T$ *where* $E_n(X)_i$ *denotes the* i^{th} *element of Equidistant sample* $E_n(X)$

Definition 19. *Given a basis function set* $\{\phi_1, \phi_2, ..., \phi_m\}$ *in function space* $\mathcal{F}$ *and a basis domain* X*, the augmented vector transformation* $Au_n : \mathcal{F}^m \to \mathbb{R}^n$ *is defined* $(Au)_n^X(\phi) = [e_1 \otimes V_n^X(\phi_1), e_2 \otimes V_n^X(\phi_2, ..., e_m \otimes V_n^X(\phi_m))]^T$

An illustration vector transformation is provided in 4.8. The approximation method for Gausslet makes use of vector transformation with the detailed algorithm as follows-

Algorithm 1 Determine Basis Function Coefficients

1: **Input:** Set of basis functions $\Phi = \{\phi_1, \phi_2, ..., \phi_n\}$, function support domain X
2: **Output:** Coefficient vector $c = [c_1, c_2, ..., c_n]^T$
3: Select the function support X as the basis domain.
4: Convert the set of basis functions Φ into a design matrix A using the augmented vector transform
5: Determine the coefficient vector c of each basis function by solving the linear least squares problem:

$$c = (A^T A)^{-1} A^T y$$

where $y = [y_1, y_2, ..., y_m]^T$ is the vector of target values.

As a simple illustration, let **b** represent the vector transform of the Haar basis function, and u_i represent the vector transform of the Gaussian basis function, then we have to find coefficients c_i such that-

$$\mathbf{b} \approx c_0\mathbf{u}_0 + c_1\mathbf{u}_1 + ... + c_n\mathbf{u}_n$$

$$= c_0 \begin{pmatrix} u_{00} \\ u_{01} \\ . \\ . \\ u_{0n} \end{pmatrix} + c_1 \begin{pmatrix} u_{10} \\ u_{11} \\ . \\ . \\ u_{1n} \end{pmatrix} + ... + c_n \begin{pmatrix} u_{n0} \\ u_{n1} \\ . \\ . \\ u_{nn} \end{pmatrix}$$

$$\mathbf{b} \approx \begin{bmatrix} u_{00} & u_{10} & ... & u_{n0} \\ u_{01} & u_{11} & ... & u_{n1} \\ ... & ... & ... & ... \\ ... & ... & ... & ... \\ u_{0n} & u_{1n} & ... & u_{nn} \end{bmatrix} \begin{pmatrix} c_0 \\ c_1 \\ . \\ . \\ c_n \end{pmatrix} = \mathbf{Ac}$$

This is equivalent to solving linear least squares problems. Here, we want to find coefficients corresponding to a set of vectors (Gaussians) that can maximally fit another vector (orthogonal Haar wavelets). The linear least squares solution is given by $\mathbf{c} = (\mathbf{A}\mathbf{A}^T)^-1(\mathbf{A}^T\mathbf{A})\mathbf{b}$. Generation of a matrix $\mathbf{A}$, for a basis with disjoint support, can be done via augmented vector transformation as

$$A = \begin{pmatrix} e^{-2^2/2} \\ e^{-1^2/2} \\ e^{-0^2/2} \\ e^{-1^2/2} \\ e^{-2^2/2} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \oplus \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ e^{-2^2/2} \\ e^{-1^2/2} \\ e^{-0^2/2} \\ e^{-1^2/2} \\ e^{-2^2/2} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \oplus \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ e^{-2^2/2} \\ e^{-1^2/2} \\ e^{-0^2/2} \\ e^{-1^2/2} \\ e^{-2^2/2} \end{pmatrix} = \begin{bmatrix} e^{-2^2/2} & 0 & 0 \\ e^{-1^2/2} & 0 & 0 \\ e^{-0^2/2} & 0 & 0 \\ e^{-1^2/2} & 0 & 0 \\ e^{-2^2/2} & 0 & 0 \\ 0 & e^{-2^2/2} & 0 \\ 0 & e^{-2^2/2} & 0 \\ 0 & e^{-0^2/2} & 0 \\ 0 & e^{-1^2/2} & 0 \\ 0 & e^{-2^2/2} & 0 \\ 0 & 0 & e^{-2^2/2} \\ 0 & 0 & e^{-1^2/2} \\ 0 & 0 & e^{-0^2/2} \\ 0 & 0 & e^{-1^2/2} \\ 0 & 0 & e^{-2^2/2} \end{bmatrix}$$

And if we want to get $\mathbf{A}$ for overlapping support, we can define an overlap matrix Ω_n^m with n as the number of basis functions and m being the discrete representation of the dilation factor (anti-overlap). This means $m = 0$ implies maximum overlap, or all basis functions are identical, and $m = n$ implies no overlap. The new overlap matrix is given by $\hat{\mathbf{A}} = \Omega_n^m\mathbf{A}$

$$\Omega_3^0 = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & ... & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & ... & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & ... & 0 & 0 & 1 \end{bmatrix} \; ; \; \Omega_3^2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0... \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0... \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0... \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0... \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0... \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0... \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1... \end{bmatrix}$$

$$\hat{A} = \Omega_3^3 A = \begin{bmatrix} e^{-2^2/2} & 0 & 0 \\ e^{-1^2/2} & 0 & 0 \\ e^{-0^2/2} & 0 & 0 \\ e^{-1^2/2} & e^{-2^2/2} & 0 \\ e^{-2^2/2} & e^{-2^2/2} & 0 \\ 0 & e^{-0^2/2} & 0 \\ 0 & e^{-1^2/2} & e^{-2^2/2} \\ 0 & e^{-2^2/2} & e^{-1^2/2} \\ 0 & 0 & e^{-0^2/2} \\ 0 & 0 & e^{-1^2/2} \\ 0 & 0 & e^{-2^2/2} \end{bmatrix}$$

The key idea is to begin with an approximate Gaussian basis which is sparser, more diagonal, and well conditioned to get the first set of coefficients w.r.t. Haar wavelet basis and then use them to build our first Gausslet. Indeed, the resulting Gausslet will not be precisely orthogonal. Still, it will certainly have a more diagonal matrix than pure Gaussian (considering it fits the Haar wavelet basis, which we know has a strictly diagonal overall matrix). This process can be applied recursively, so the first set of Gausslets can be used as a new basis to find the next set of Gaussian coefficients.

4.5. Numerical methods and algorithms

4.5.1. Scaled sampling method

We encountered numerous instances of shifting a function over the x-coordinate with different values, also known as translation. Though the method analytically is straightforward, to apply it in a numerical program can involve a chain of repetitive and costly operations. As a simple example, given we have a Gaussian function and we want to scale it using the scaling relation in (2.8), we need to evaluate the Gaussian function (which involves computation of polynomials of very high degree) with exponential dependency on the scaling coefficient. Additionally, there will be instances of redundant evaluation of the Gaussian function at the same coordinate point multiple times. To make matters worse, in case of functions like the FEM hat function as in [15], the computer program requires an if statement which makes the function un-vectorizable. To tackle this issue, we can implement a scaled sampling of the

calculated value of a function. By scaled sampling, we only calculate the function value once for scaling = 1, and then sample a subset from the same values to retrieve values at a different scaling level. To understand this method, let's have $F(X)$ be set of all values a function F in $L^2(R)$ space takes in a uniform discrete domain $X = \{x_1, x_2, ..x_n\}$ such that $x_i \in \mathbb{R}$, $x_{i+1} - x_i$ is constant, support of F is subset in X, and $F(x_1), F(x_n) = 0$. If we want to calculate $F_{m,t}$, i.e F scaled to $m \geq 1$ and translated (shifted) by $t \in \mathbb{R}$ on the domain X, it can be shown that for integral scaling m, $F_m(X) \subseteq F(X)$. Once we have calculated $F(X)$ values for $m = 1$, we can re-use them as values of $F_m(X)$ as long as m is an integer.

Many square integrable functions, like the Gaussian function, are never exactly zero for any finite value in a finite domain. But since the value of all square functions tends towards zero at $\pm\infty$, we can still apply the scaled sampling method with reasonable accuracy by adjusting the size of the domain X. We can also use this methodology if we have $m = 1/n$ with $n \in \mathbb{N}$, and $F_m(X)$ is no longer a subset of $F(X)$. In this case, we will need values of $F(X)$ in between the interval x_i and x_{i+1}, which is calculated using simple barycentric linear interpolation, i.e

$$F(x_j) = \frac{F(x_i)(x_{i+1} - x_j) + F(x_{i+1})(x_j - x_i)}{x_{i+1} - x_i}$$

Both cases $m > 1$ or $m < 1$ have limitations when using the scaled sampling method. For any value of scale $m > \frac{x_{i+1}-x_i}{x_n-x_1}$ or $m < \frac{x_n-x_1}{x_{i+1}-x_i}$ scaled sampling method cannot be applied and accuracy of the $F_m(X)$ reduces as m moves away from the value of 1.

An illustration of this method is shown in 4.9, and a detailed algorithm is given as follows-

4.5.2. Measuring orthogonality of a basis

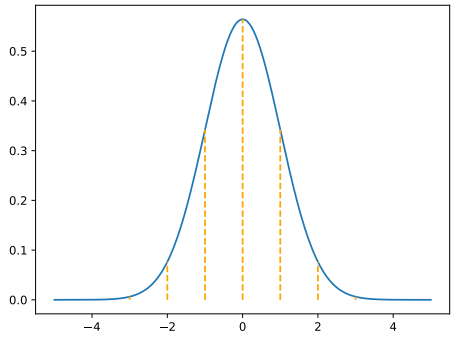
We earlier used the condition number to determine the singularity of a matrix, and the result was that the singularity of the overlap matrix of a basis also implied a denser overlap matrix or poor orthogonality. This measure can be good to discard a basis (with a high condition number). Still, given two bases very close to being diagonal, we need a better metric, as the eigenvalues of a diagonal matrix are the diagonal values themselves and ratio max/min eigenvalues would no longer be indicative of anything. A convenient way to determine how much diagonal a matrix is can be calculated by taking the ratio of the diagonal L_2 norm and the matrix L_2 norm.

$$O(A) = \frac{||d||_2}{||A||_2}$$

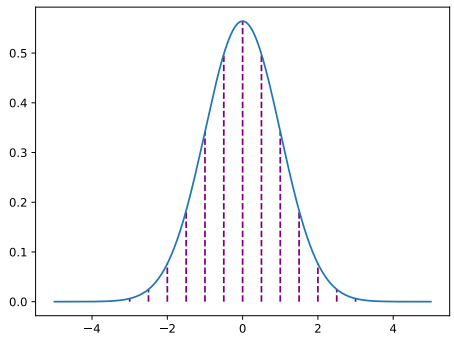
The value $O(A)$ is 1 for a strictly diagonal matrix, i.e, all the elements except the diagonal are equal to zero, and

Algorithm 2 Scaled Sampling

function ScaledSampling(func_val, X, translate, scaling $= 1$)
  $N \leftarrow \text{length}(X)$
  $X \leftarrow X - \frac{X[N-1]+X[0]}{2}$ ▷ Normalization
  if $N \neq \text{length}(\text{func\_val})$ **then**
    print "func_val and X not of same length!"
    return
  end if
  fitting_func $\leftarrow$ fit_select
  if scaling < 1 **then**
    fitting_func $\leftarrow$ fit_interpol
  end if
  $dx \leftarrow \frac{X[N-1]-X[0]}{N}$
  $y \leftarrow$ zeros of length N
  begin $\leftarrow \max(0 \lfloor (N-1) \cdot \frac{\text{translate}}{X[N-1]-X[0]} \rfloor)$
  for $i \leftarrow$ begin to $N-1$ **do**
    $x_{\text{new}} \leftarrow \text{scaling} \cdot (X[i] - \text{translate})$
    shifted_i $\leftarrow \lfloor (N-1) \cdot \frac{x_{\text{new}}-X[0]}{X[N-1]-X[0]} \rfloor$
    shifted_i $\leftarrow \min(\max(\text{shifted\_i}, 0), N-2)$
    $a \leftarrow \frac{x_{\text{new}}-X[\text{shifted\_i}]}{dx}$
    $y[i] \leftarrow \text{fitting\_func}(\text{func\_val}[\text{shifted\_i}], \text{func\_val}[\text{shifted\_i} + 1], a)$
  end for
  return y
end function



$$V_5^{[-2,2]}(\phi) = \phi \begin{pmatrix} -2 \\ -1 \\ 0 \\ 1 \\ 2 \end{pmatrix} = \begin{pmatrix} e^{-2^2/2} \\ e^{-1^2/2} \\ e^{-0^2/2} \\ e^{-1^2/2} \\ e^{-2^2/2} \end{pmatrix}$$



$$V_9^{[-3,3]}(\phi) = \phi \begin{pmatrix} -2.5 \\ -2 \\ -1.5 \\ -1 \\ 0 \\ 1 \\ 1.5 \\ 2 \\ 2.5 \end{pmatrix} = \begin{pmatrix} e^{-2.5^2/2} \\ e^{-2^2/2} \\ e^{-1.5^2/2} \\ e^{-1^2/2} \\ e^{-0^2/2} \\ e^{-1^2/2} \\ e^{-1.5^2/2} \\ e^{-2^2/2} \\ e^{-2.5^2/2} \end{pmatrix}$$

Figure 4.8.: Transformation of continuous basis function into a discrete vector: The Gaussian function is discretely sampled based on i) the width of sampling, ii) the unit of sampling. In the top figure, the unit size is 1 and the width is 4, resulting in a 5-element sample. In the bottom figure, the unit size is 5 and the width is 6, resulting in a 9-element sample. Each sample is later converted into a vector of corresponding function values, which serves as a new approximate basis.

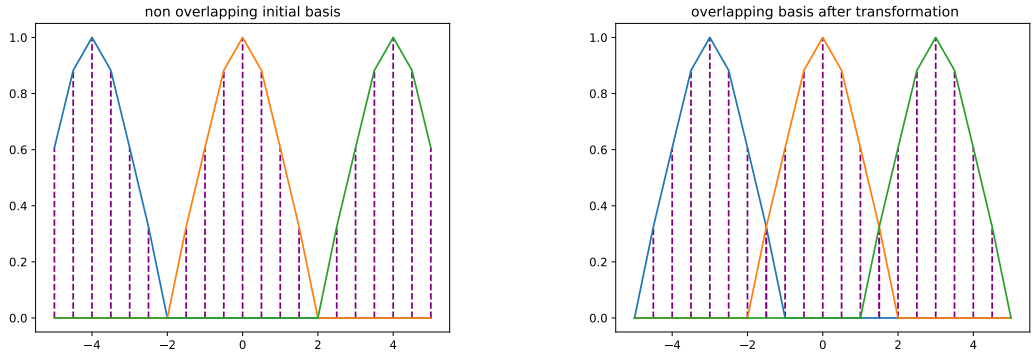


Figure 4.9.: The vectors from the new basis are augmented into higher dimensions based on the number of basis functions and their position. a) A basis matrix is formed by initially keeping the vectors disjoint and nonoverlapping. b) An overlap is performed later by applying the overlap matrix based on the distance of overlap and the number of functions. In this case, the overlap is between the two elements of the vectors, and therefore, the overlap operation of Ω_3^2 is applied

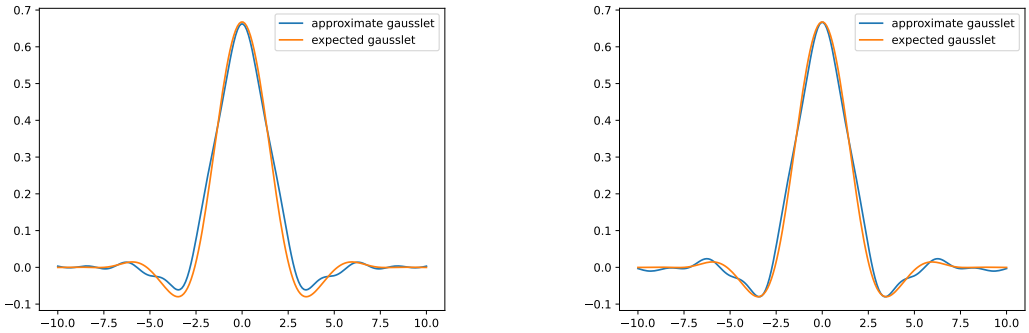


Figure 4.10.: Comparison of Gausslet calculated from the approximation method and the expected Gausslet. The method works optimally at unit size = 1/2 and width = 11 with 1-2 recursions. Any additional increment in parameters leads to overfitting and a less orthogonal basis

4.6. Symmetric permutation method to obtain orthogonal wavelet ansatz

In an alternative method, results similar to unitary circuits can be arrived at using Permutation operators on the wavelet coefficient. In the unitary circuit design method, we saw that the circuit can be expanded to infinite depth by augmenting zeroes to the coefficient and using appropriate unitary transformations so that the orthogonality of the scaling/wavelet coefficient is always preserved. Within the Unitary transformations, there were two broad categories: Constant Unitary transformation (SWAP gate) and Variational Unitary transformation (U_θ gates). The constant gate we came across can be generalized to broader Permutation gates (or matrices). In general, any permutation operation can be represented as a matrix $M \in \mathbb{R}^m \times \mathbb{R}^m$ of the form [16]

$$\mathcal{P}_{ij} = \delta_{i\pi(j)}$$

Where $\pi(j)$ is the $j_t h$ element of any permutation π on the set $1, 2, 3 \ldots .m$. However, not all permutation matrices can preserve the orthogonality of the scaling coefficients. The obvious reason is that any change in the arrangement of the scaling coefficient would affect their inner product in unpredictable ways (the only exception being the case where there is no translation). We can understand from a simple example below:

Given we have two orthogonal vectors in $\mathbb{R}^3$ as $\mathbf{u} = [u_1, u_2, u_3]$ and $\mathbf{v} = [v_1, v_2, v_3]$, constituting a scaling coefficient $\mathbf{h} = [u_1, u_2, u_3, v_1, v_2, v_3]$ with dilation factor 3; as expected we will have $[\mathbf{h}, \mathbf{h}]_3 = u_1v_1 + u_2v_2 + u_3v_3 = 0$. Now if we apply a simple permute operation swapping $h[2]$ an $h[3]$, we will end up with a different $\mathbf{h'} = [u_1, u_2, v_1, u_3, v_2, v_3]^T$ in which we do not expect $[\mathbf{h'}, \mathbf{h'}]_3 = u_1u_3 + u_2v_2 + v_1v_3$ to be always equal to zero. At the same time it can also be seen that any local permutation operation which acts identically and locally on the clusters of three values: i.e $[u_1, u_2, u_3]$ and $[v_1, v_2, v_3]$ will always preserve the orthogonality,

since any local rearrangement acting within range of the dilation factor doesn't change the relative positions of the coefficients while calculating their inner product. To understand with a simple example: if we apply swap operation between h[0] and h[1], and h[3] and h[4], and have $\mathbf{h'} = [u_2, u_1, u_3, v_2, v_1, v_3]$, calculating $[\mathbf{h'}, \mathbf{h'}]_3 = u_2v_2 + u_1v_1 + u_3v_3$ computes to zero again.

We aim to generalize a local permutation operator for any given orthogonal scaling coefficients $\mathbf{h} = [h_1h_2...h_dl] = [\mathbf{v}_1, \mathbf{v}_2, ..\mathbf{v}_l]$ with $\mathbf{v}_i = [h_{di}, h_{di+1}, ..h_{di+d-1}]$

Definition 20. ***Local Permutation of scaling coefficient vector****: Given scaling sequence* $\boldsymbol{h} = [h_1, h_2, ..., h_n]$ *with dilation factor* $d \in \mathbb{N}$ *such that n is always an integral multiple of d. A local permutation on h is defined as permutation operation* $P_d : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ *applied as matrix* $I_{n/d} \otimes \mathcal{P}_\pi$ *with* $\mathcal{P}_\pi$ *being* $d \times d$ *permutation on* $\boldsymbol{h}$

As an example on scaling coefficients with 9 elements, we will have a cyclic permutation[16] operator transforming a sequence of the form $[1, 2, 3]$ to $[2, 3, 1]$ as

$$\mathcal{P}_\pi = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \tag{4.17}$$

and the resulting local permutation operator P_π will be

$$I_3 \otimes \mathcal{P}_3 = \left[\begin{array}{ccc|ccc|ccc} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{array}\right]$$

Definition 21. ***cyclic right permutation****: An operation transforming* $v_{i,1} \to v_{mod(i-1,ld)+1,1}$ *with* $mod(x, y)$ *representing modulus of x w.r.t y. Similarly,* ***cyclic left permutation*** *is operation transforming*$v_{l-k,j} \to v_{mod(i+1,ld)+1,j}$

Another permutation h that will preserve the shifted inner products over $[\mathbf{h}, \mathbf{h}]_{id}$ (with certain conditions) is cyclic permutation of i_{th} elements of all $\mathbf{v}_1, \mathbf{v}_2, .., \mathbf{v}_l$. To illustrate given $\mathbf{h} = [\mathbf{u}, \mathbf{v}, \mathbf{w}] = [u_1, u_2, u_3, v_1, v_2, v_3, w_1, w_2, w_3]$, The orthogonality conditions are

$$\begin{aligned} u_1v_1 + u_2v_2 + u_3v_3 + v_1w_1 + v_2w_2 + v_3w_3 &= 0 \\ u_1w_1 + u_2w_2 + u_3w_3 &= 0 \end{aligned}$$

A cyclic right permutation permute $\mathcal{P}_3$ as in of first elements of $\mathbf{u}, \mathbf{v}, \mathbf{w}$ will result in $\mathbf{h'} = [w_1, u_2, u_3, u_1, v_2, v_3, v_1, w_2, w_3]$ and with new conditions

$$w_1u_1 + u_2v_2 + u_3v_3 + u_1v_1 + v_2w_2 + v_3w_3 = 0$$
$$w_1v1 + u_2w_2 + u_3w_3 = 0$$

The orthogonality conditions for h and h' can hold simultaneously only if $w_1, w_2, w_3 = 0$ or $u_1, u_2, u_3 = 0$. In fact, this condition holds for $\mathbf{h} = [h_1h_2...h_{dl}]$ $(1 \leq k \leq l-1)$

$$\sum_{j=1}^{d}\sum_{i=1}^{(l-k)} v_{i,j}v_{i+k,j} = \sum_{i=1}^{(l-k)} v_{i,1}v_{i+k,1} + \sum_{j=2}^{d}\sum_{i=1}^{(l-k)} v_{i,j}v_{i+k,j} \tag{4.18}$$

After cyclic right permutation, the previous expression becomes

$$v_{l,1}v_{k+1,1} + \sum_{i=2}^{(l-k)} v_{i-1,1}v_{i-1+k,1} + \sum_{j=2}^{d}\sum_{i=1}^{(l-k)} v_{i,j}v_{i+k,j} \tag{4.19}$$
$$= v_{l,1}v_{k+1,1} + \sum_{i=1}^{(l-k-1)} v_{i,1}v_{i+k,1} + \sum_{j=2}^{d}\sum_{i=1}^{(l-k)} v_{i,j}v_{i+k,j} \tag{4.20}$$

Equations 4.20 and 4.18 will be the same expression if $v_{l,1} = 0$, hence providing us another option for an orthogonality preserving invariant. The same can be extended to any cyclic right permutation $v_{l-k,j} \rightarrow v_{mod(i-1,ld)+1,j}$ requiring $v_{l,j} = 0$ and for cyclic left permutation the condition is $v_{0,j} = 0$. To define the matrix operator of such a transformation, we can re-use the matrix as defined in the equation 4.6 and **projector matrices** to only a selected set of elements in the scaling factor **h**. In the case of Local Permutation, we broadcast the permute operation to each sub-element of size d in he **h** vector. However, we must apply the permute operation across the sub-elements of size d at a particular position. A matrix operator which returns the first elements of all l sub-vectors of size d in h is given by

$$I_l \otimes \Pi_i : \Pi_i = (\mathbf{e}^i)^T\mathbf{e}^i$$

with $\mathbf{e}^i$ being d-dimensional standard basis vectors as defined in equation 4.9. As an example, if have $\mathbf{h} = [h_1, h_2, h_3, h_4]$, then

$$I_2 \otimes (\begin{bmatrix}1\\0\end{bmatrix}\begin{bmatrix}1 & 0\end{bmatrix}) \begin{pmatrix}h_1\\h_2\\h_3\\h_4\end{pmatrix} = \begin{pmatrix}h_1\\0\\h_3\\0\end{pmatrix} \tag{4.21}$$

We can obtain same results via the tensor product of the vectors-

$$\begin{pmatrix} h_1 \\ h_3 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} h_1 \\ 0 \\ h_3 \\ 0 \end{pmatrix}$$

It means we can also decompose h as

$$\begin{pmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{pmatrix} = \begin{pmatrix} h_1 \\ h_3 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} h_2 \\ h_4 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

To apply the permutation operator $\mathcal{P}$ at either $(h_2, h_4)^T$ or $(h_1, h_3)^T$, the linear operator will be $Q_2 = \mathcal{P} \otimes \Pi_1 + I_2 \otimes (I_2 - \Pi_1)$ if we want to permute $(h_1 h_3)^T$ and $Q_2 = \mathcal{P} \otimes \Pi_2 + I_2 \otimes (I_2 - \Pi_2)$ if we want permute $(h_2, h_4)^T$. A generalized operator will be

$$Q = \sum_k \mathcal{P}_k \otimes \Pi_{i_k} + I_d \otimes (I_l - \sum_k \Pi_{i_k})$$

We can propose an ansatz for creating an orthogonality-preserving unitary circuit based on P and Q permutation operators.

5. Results and Conclusion

5.1. Approximation of Hermite polynomials using Gausslets

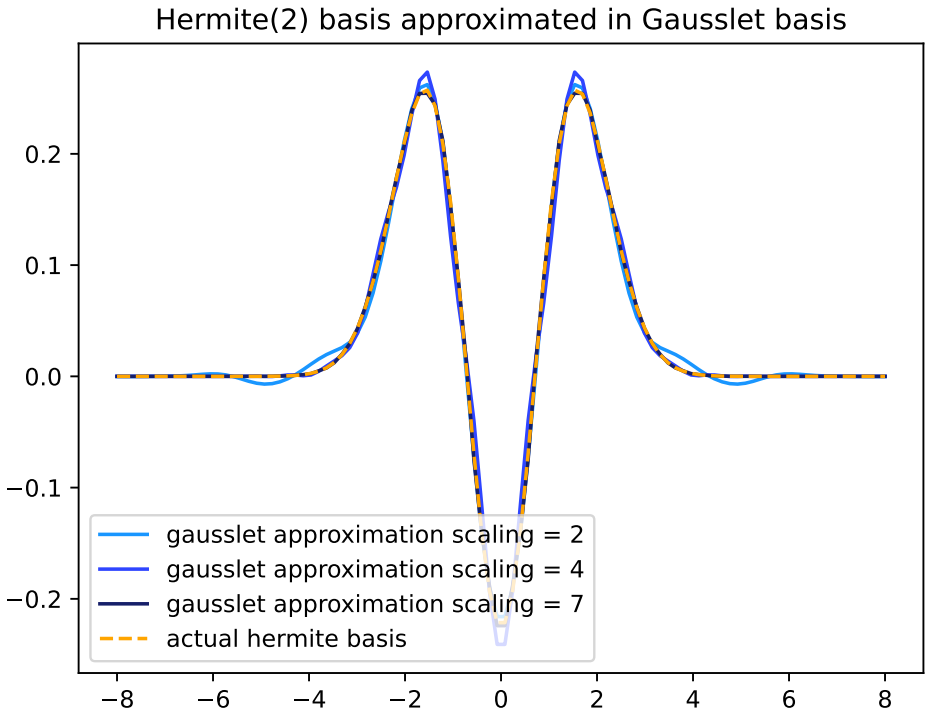


Figure 5.1.: Illustration of increasing accuracy of Gausslet approximate for fitting Hermite basis functions with increasing scaling factor

We have used the solutions of the electronic wavefunction in quantum dots, i.e, Hermite basis function, as our benchmark to evaluate the Orthogonal Gausslet basis. The accuracy of the solution is determined by the initial scaling parameter. Since higher scaling implies smaller support per basis, higher accuracy is expected as well. This can also be concluded from the argument that the space spanned by a higher scaling level is a superset of all lower scaling levels, as discussed in the theory of MRA. The illustration of increasing accuracy is depicted in 5.1

However, higher scaling also means a higher number of basis functions in the domain and hence higher computational costs. Therefore, it would be appropriate to use the right amount of scaling based on the shape of the function we want to fit. In the case of Hermite

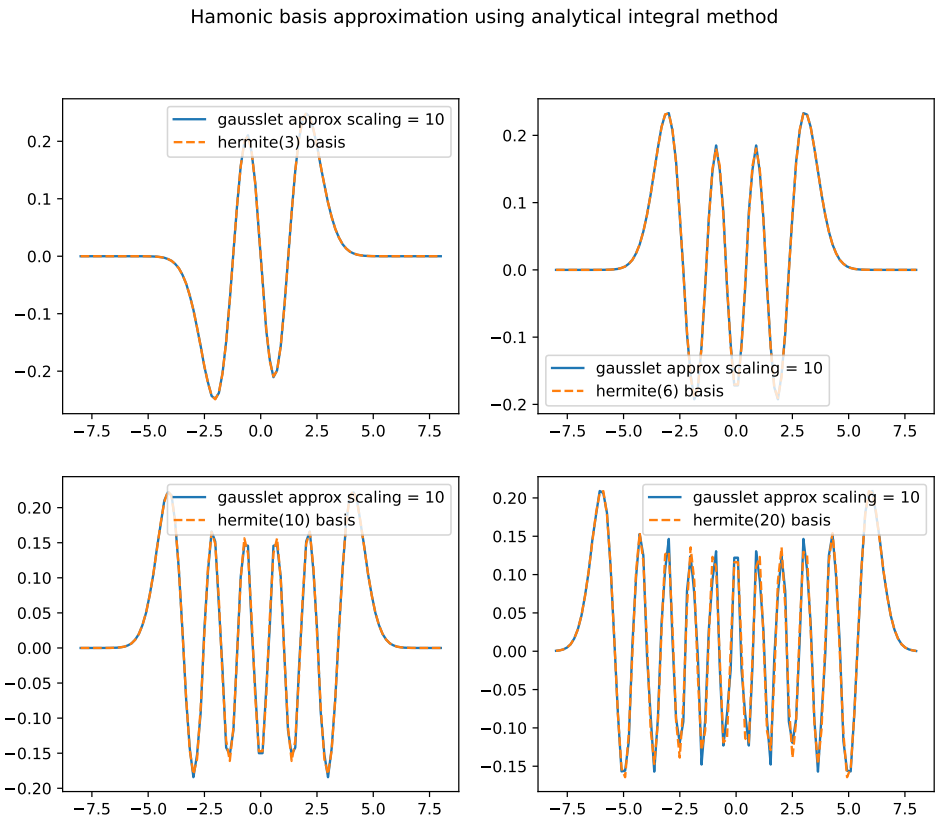


Figure 5.2.: Plot of Hermite basis functions and corresponding Gausslet approximation for different orders with the analytical integration method for determining Gausslet coefficients

polynomials, the number of zeros increases with degree within the same span of length over the x-axis. This means the frequency of oscillation also increases, which means we can directly adjust the scaling of the basis function with the degree of the polynomial. Since the Hermite basis is just a Hermite polynomial multiplied by a Gaussian, the same can be applied to it.

Since the Gausslet basis is very close to orthogonal, the coefficient corresponding to each Gausslet function for a Hermite basis function is given by (assuming Gausslets are normalized to 1 as well)

$$c_i = \int_{-\infty}^{-\infty} H_n(x) G_i(x).dx$$

The above integral can either be simplified analytically or can be computed entirely using a numerical integration scheme. In the 5.3 and 5.4, Hermite basis functions of four orders are expressed in terms of Gaussian. Qualitatively, it can be observed that the accuracy is nearly identical in both methods.

To analyse the errors more rigorously, a plot of absolute error in the values of the true

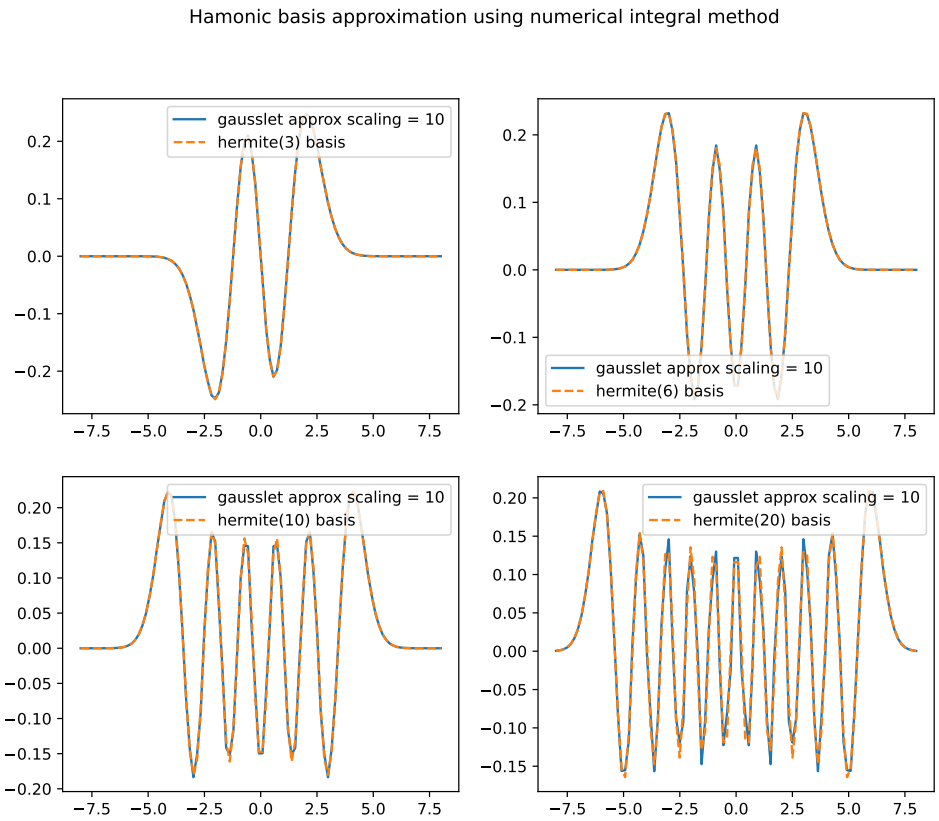


Figure 5.3.: Plot of Hermite basis functions and corresponding Gausslet approximation for different orders with numerical integration method for determining Gausslet coefficients

Hermite basis function and the fit in the Gausslet basis is shown in 5.2. Another plot of 5.5 shows maximum and average error as the order of the Hermite basis is increased while keeping the scaling of the Gausslet basis fixed. Both plots indicate that while the average error remains nearly constant, there is an increase in the maximum error with respect to the order of the Hermite basis for a constant scaling factor. However, this is an expected result, as discussed earlier, and can be fixed by using an appropriate scale for each order. In 5.6, it can be easily inferred that the errors can be significantly reduced for low and high orders by increasing the scaling level.

5.2. Approximation of Coulomb Integral using Gausslets

In order to use a basis set for simulating electronic structure, we have to compare the accuracy of the Coulomb repulsion tensor in addition to how well it fits the function. Since our ultimate goal is the calculation of ground state energy, which requires DMRG algorithms to run on Coulomb tensors, the accuracy of the tensors is equally crucial. In the 5.7, the values of the

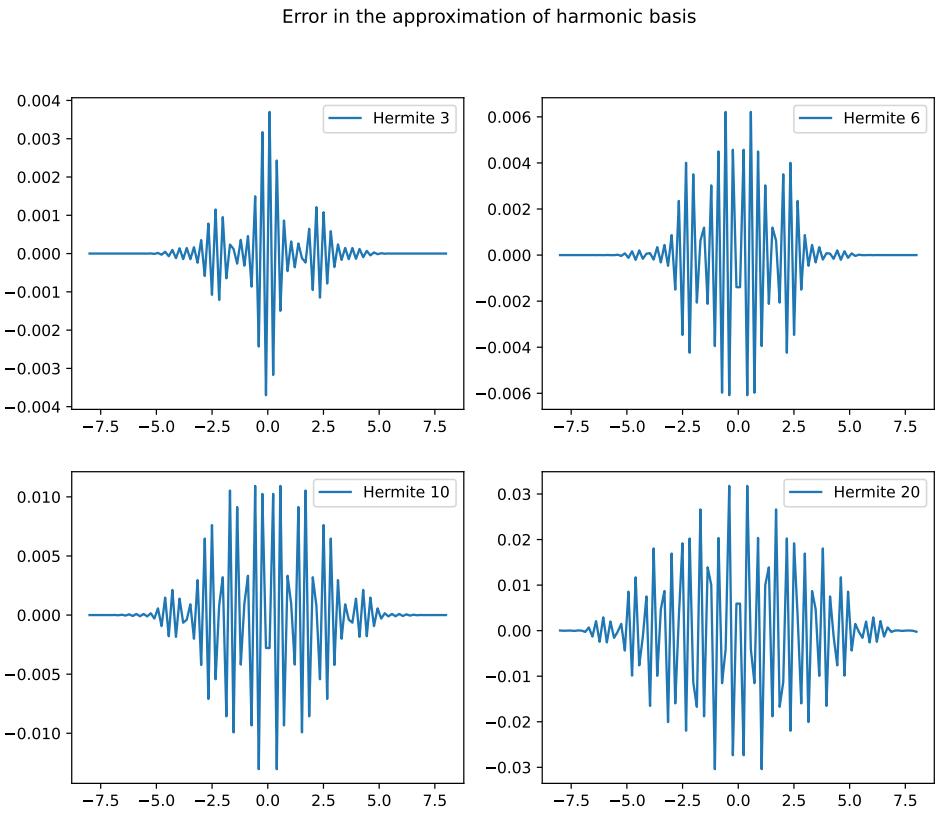


Figure 5.4.: Absolute error between Hermite basis function and Gausslet approximation for different orders with a constant scaling factor.

2-electron Coulomb tensor are shown.

5.3. Conclusion and future work

In our thesis, we were able to reconstruct the unitary circuits method to generate scaling and wavelet coefficients for wavelets with different dilation factors. We developed an in-depth understanding of the theory of wavelets and multi-resolution analysis. Using the fundamental equations of Coulomb's law and Pauli's exclusion principle, we derived the expression of the two-electron Hamiltonian used in the electronic structure calculation in the second quantization form and familiarize ourselves with the tensor networks representation of the Hamiltonian of a multi-electronic system and the DMRG algorithm to find the ground state of the tensor Hamiltonian. The wavelet and scaling coefficients were later used to derive the coefficients of the orthogonal Gausslet basis. The orthogonal wavelets were later applied to simulate the electronic wavefunction, more specifically, of the quantum dot system. We developed a general understanding of the Harmonium model, which can be used to simulate the trapped ion and quantum dot systems. We verified that the Gausslet basis fit really well

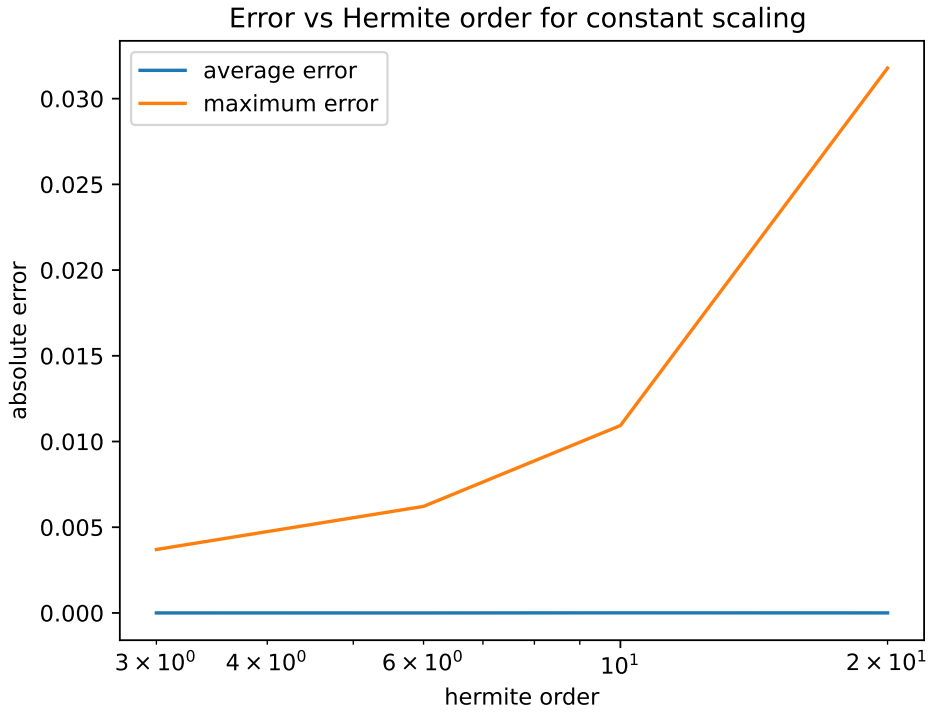


Figure 5.5.: Absolute error vs Hermite order graph for Gausslet approximation of Hermite basis functions for constant scaling factor. While the mean error remains negligible for all orders, the maximum error increases with positive second derivative as the function order is increased.

with the harmonic basis functions.

For future work, we can apply the DMRG algorithm [2] [17] to the Coulomb matrices for the Hydrogen atom and other atomic/molecular species to compare the speed and accuracy of the solution. The orthogonal Gausslet can also be reshaped to non-uniform grids and applied to systems where higher resolution is needed near the vicinity of atomic/molecular nuclei than in other regions. DMRG algorithm can also be combined with message passing algorithms [18] [19] [20], which can find the ground state energy of electrons even faster.

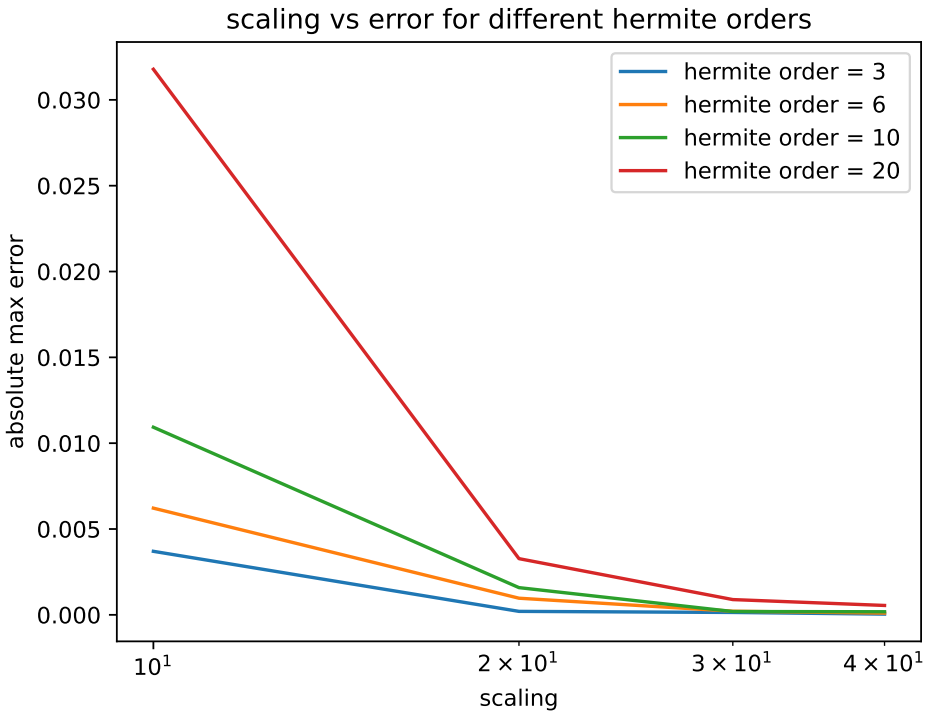


Figure 5.6.: Maximum absolute error vs scaling factor graph for different orders of Hermite basis functions. By increasing the scaling factor, absolute errors can be brought down to almost the same level for all orders.

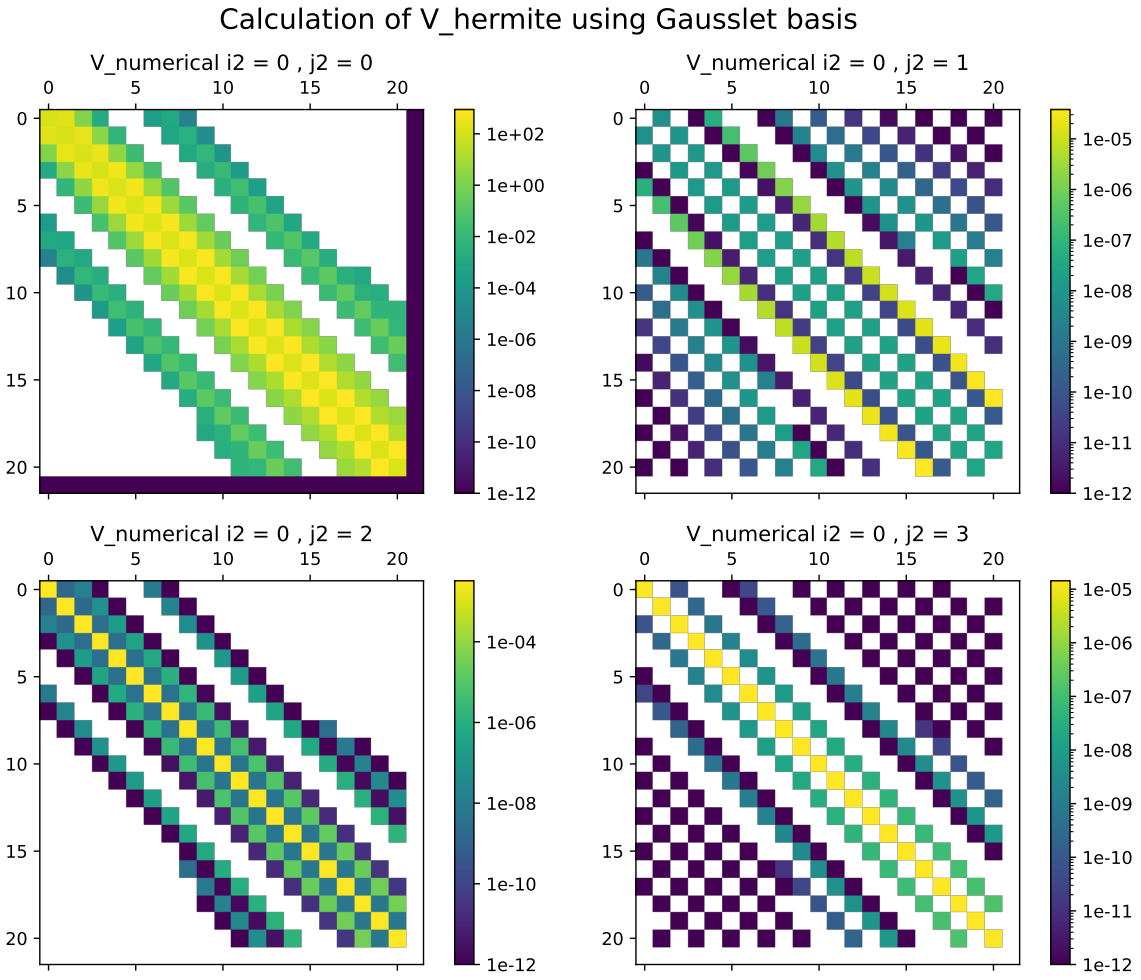


Figure 5.7.: Visualization of Coulomb tensor of Harmonium model of form $V_{i_1 i_2 j_1 j_2}$ for fixed values of i_1, j_2 obtained via Gausslet basis.

A. Derivation of Mathematical Results

A.1. Product of two Gaussian

Given two Gaussian $g_1,\ g_2$ separated by distance λ, we can choose distance halfway between them as the origin of coordinates and define them as $g_1 = e^{(-\frac{(x-\lambda/2)^2}{2\sigma})}$ and $g_2 = e^{(-\frac{(x+\lambda/2)^2}{2\sigma})}$. We will have

$$\begin{aligned} g_1 \cdot g_2 &= e^{(-\frac{(x-\lambda/2)^2}{2\sigma})} \cdot e^{(-\frac{(x+\lambda/2)^2}{2\sigma})} \\ &= e^{(-\frac{2(x^2+(\lambda/2)^2)-2(\lambda/2)x+2(\lambda/2)x}{2\sigma})} = e^{\frac{-\left(x^2+(\lambda/2)^2\right)}{\sigma}} \\ &= e^{\frac{-(x^2+\lambda/2)^2)}{\sigma}} = e^{\frac{-(\lambda/2)^2}{\sigma}} e^{\frac{-x^2}{\sigma}} \end{aligned}$$

For different values of σ

$$\begin{aligned} g_1 \cdot g_2 &= e^{(-\frac{(x-\lambda/2)^2}{2\sigma_1})} \cdot e^{(-\frac{(x+\lambda/2)^2}{2\sigma_2})} \\ &= e^{(-\frac{(\sigma_1+\sigma_2)(x^2+(\lambda/2)^2)-(\sigma_1-\sigma_2)(\lambda/2)x+2(\lambda/2)x}{2\sigma_1\sigma_2})} = e^{\frac{-\left(x^2+(\lambda/2)^2\right)}{\sigma}} \\ &= e^{\frac{-(x^2+\lambda/2)^2)}{\sigma}} = e^{\frac{-(\lambda/2)^2}{\sigma}} e^{\frac{-x^2}{\sigma}} \end{aligned}$$

Hence the product of any two gaussian with same width (σ) and different centers gives another Gaussian centered at mid-point of the original two Gaussian and width twice as big (σ) and peak value reduced by a factor of $e^{\frac{-(\lambda/2)^2}{\sigma}}$

A.2. Electronic integrals for Harmonic model with Gausslets

$$h_{ij} = \langle\phi_i|\frac{p_k^2}{2m} + \frac{1}{2}m\omega^2 x_k^2|\phi_j\rangle \ ;\ V_{i_1 i_2 j_1 j_2} = \langle\phi_{i_1}\phi_{i_2}|K|x_k - x_l|^s|\phi_{j_1}\phi_{j_2}\rangle$$

First simplifying h_{ij} and using the one dimensional momentum operator from quantum mechanics $p_k = -\mathrm{i}\hbar\frac{\partial}{\partial x}$

$$\begin{aligned} h_{ij} = \langle\phi_i|\frac{p^2}{2m} + \frac{1}{2}m\omega^2 x_k^2|\phi_j\rangle &= \langle\phi_i|\frac{p^2}{2m}|\phi_j\rangle + \langle\phi_i|\frac{1}{2}m\omega^2 x_k^2|\phi_j\rangle \\ = \frac{\hbar^2}{2m}\langle\phi_i|\frac{\partial^2}{\partial x^2}|\phi_j\rangle &+ \frac{1}{2}m\omega^2\langle\phi_i|x_k^2|\phi_j\rangle \end{aligned}$$

Simplifying the $\langle \phi_i | \frac{\partial^2}{\partial x^2} | \phi_j \rangle$ term

$$\begin{aligned}\langle \phi_i | \frac{\partial^2}{\partial x^2} | \phi_j \rangle &= \int_{-\infty}^{-\infty} \phi_i(x) \frac{\partial^2 \phi_j(x)}{\partial x^2} \, \mathrm{d}x \\ &= \left[\phi_i \frac{\partial \phi_j(x)}{\partial x} \right]_{-\infty}^{\infty} - \int_{-\infty}^{-\infty} \frac{\partial \phi_i(x)}{\partial x} \, \frac{\partial \phi_j(x)}{\partial x} \, \mathrm{d}x \\ &= -\int_{-\infty}^{-\infty} \frac{\partial \phi_i(x)}{\partial x} \, \frac{\partial \phi_j(x)}{\partial x} \, \mathrm{d}x\end{aligned}$$

Since $\phi_j(x) = \sum_k c_k e^{\frac{-(x-\alpha k-\beta j)^2}{2\sigma}}$ and $\frac{\partial}{\partial x} e^{-x^2/2\sigma} = \frac{-1}{\sigma} x e^{-x^2/2\sigma}$

$$\begin{aligned}\int_{-\infty}^{-\infty} \frac{\partial \phi_i(x)}{\partial x} \, \frac{\partial \phi_j(x)}{\partial x} \, \mathrm{d}x &= \frac{1}{\sigma^2} \int_{-\infty}^{-\infty} \sum_k (x - \alpha k - \beta i) \, ... \\ ... \, c_k e^{-\frac{(x-\alpha k-\beta i)^2}{2\sigma}/2\sigma} \, . \, \sum_l (x - \alpha l - \beta j) \, . \, c_l e^{-\frac{(x-\alpha l-\beta j)^2}{2\sigma}/\sigma} \mathrm{d}x\end{aligned}$$

Substituting $p = \alpha k + \beta i$, $q = \alpha l + \beta j$, $\lambda = |p - q| = p - q$ (letting p > q) and $t = x - (p + q)/2$

$$\begin{aligned}&= \int_{-\infty}^{-\infty} \sum_{kl} (x - p)(x - q) \, . \, c_k c_l \, . \, e^{-(x-p)^2/2\sigma} \, . \, e^{-(x-q)^2/2\sigma} \, . \mathrm{d}x \\ &= \int_{-\infty}^{-\infty} \sum_{kl} (t - \lambda/2)(t + \lambda/2) \, . \, c_k c_l \, . \, e^{-\lambda^2/4\sigma} \, . \, e^{-t^2/\sigma} \, . \mathrm{d}t \\ &= \int_{-\infty}^{-\infty} \sum_{kl} (t^2 - (\lambda/2)^2) \, . \, c_k c_l \, . \, e^{-\lambda^2/4\sigma} \, . \, e^{-t^2/\sigma} \, . \mathrm{d}t \\ &= \frac{1}{\sigma^2} \int_{-\infty}^{-\infty} \sum_{kl} c_k c_l \, . \, e^{-\lambda^2/4\sigma} \, . \, t^2 \, . \, e^{-t^2/\sigma} \, . \mathrm{d}t \; - \; \frac{1}{\sigma^2} \int_{-\infty}^{-\infty} \sum_{kl} (\lambda/2)^2 \, . \, c_k c_l \, . \, e^{-\lambda^2/4\sigma} \, . \, e^{-t^2/\sigma} \, . \mathrm{d}t\end{aligned}$$

since $\lambda = p - q = \alpha(k - l) + \beta(i - j)$ is function of differences $i - j$ and $k - l$. The terms $i - j$ and $k - l$ can be reduced to just one index, making λ dependent on just two variables. However, in our expression, we also have coefficients of Gaussian c_k and c_l, which cannot be further reduced. Therefore, we will write $\lambda_{i-j,kl}$ as a way to represent the dependency of the final expression on three variables instead of the expected four. After

$$\frac{1}{\sigma^2} \int_{-\infty}^{-\infty} \sum_{kl} c_k c_l \, . \, e^{-\lambda_{i-j,kl}^2/4\sigma} \, . \, t^2 \, . \, e^{-t^2/\sigma} \, . \mathrm{d}t \; - \; \frac{1}{\sigma^2} \int_{-\infty}^{-\infty} \sum_{kl} (\lambda_{i-j,kl}/2\sigma)^2 \, . \, c_k c_l \, . \, e^{-\lambda_{i-j,kl}^2/4\sigma} \, . \, e^{-t^2/\sigma} \, . \mathrm{d}t$$

Hence, the final expression is just dependent on $i - j$, which implies translational invariance of the basis functions as the integral are only dependent on relative distance. The terms in

expression can be evaluated numerically. For convenience of notation, we represent the above integrals as

$$\begin{aligned}
J_{i-j}(2) &:= \int_{-\infty}^{-\infty} \sum_{kl} c_k c_l \,.\, e^{-\lambda_{i-j,kl}^2/4\sigma} \,.\, t^2 \,.\, e^{-t^2/\sigma} \,.\mathrm{d}t \\
&= \left\{ \frac{1}{\sigma^2} \sum_{kl} c_k c_l \,.\, e^{-\lambda_{i-j,kl}^2/4\sigma} \right\} . \int_{-\infty}^{\infty} t^2 \, e^{-t^2/\sigma} \,.\mathrm{d}t \\
J_{i-j}(0) &:= \int_{-\infty}^{-\infty} \sum_{kl} (\lambda_{i-j,kl}/2\sigma)^2 \,.\, c_k c_l \,.\, e^{-\lambda_{i-j,kl}^2/4\sigma} \,.\, e^{-t^2/\sigma} \,.\mathrm{d}t \\
&= \left\{ \sum_{kl} (\lambda_{i-j,kl}/2\sigma)^2 \,.\, c_k c_l \,.\, e^{-\lambda_{i-j,kl}^2/4\sigma} \right\} . \int_{-\infty}^{-\infty} e^{-t^2/\sigma} \,.\mathrm{d}t
\end{aligned}$$

The net momentum integral can be written as

$$\langle \phi_i | \frac{\partial^2}{\partial x^2} | \phi_j \rangle = J_{i-j}(0) - J_{i-j}(2) \tag{A.1}$$

where n in $J_{i-j}(n)$ is used to represent power of t in the integrals. Before we calculate $\frac{1}{2} m\omega^2 \langle \phi_i | x_k^2 | \phi_j \rangle$, we will simplify two-electron integrals, which will also help us evaluate this component. Evaluating $V_{i_1 i_2 j_1 j_2}$ for the simple and relevant case s = 2.

$$\begin{aligned}
V_{i_1 i_2 j_1 j_2} &= \langle \phi_{i_1} \phi_{i_2} | K | x_k - x_l |^2 | \phi_{j_1} \phi_{j_2} \rangle = K \langle \phi_{i_1}(x_k) \phi_{i_2}(x_l) | (x_k - x_l)^2 | \phi_{j_1}(x_k) \phi_{j_2}(x_l) \rangle \\
&= \int_{x_l} \int_{x_k} \phi_{i_1}(x_k) \phi_{i_2}(x_l) (x_k - x_l)^2 \phi_{j_1}(x_k) \phi_{j_2}(x_l) \, \mathrm{d}x_k \mathrm{d}x_l \\
&= \int_{x_l} \int_{x_k} ({x_k}^2 + {x_l}^2 - 2x_k x_l) \phi_{i_1}(x_k) \phi_{j_1}(x_k) \phi_{i_2}(x_l) \phi_{j_2}(x_l) \, \mathrm{d}x_k \mathrm{d}x_l \\
&= \int_{x_l} \phi_{i_2}(x_l) \phi_{j_2}(x_l) \, \mathrm{d}x_l \,.\, \int_{x_k} {x_k}^2 \phi_{i_1}(x_k) \phi_{j_1}(x_k) \, \mathrm{d}x_k ..... \\
&..... + \int_{x_k} \phi_{i_1}(x_k) \phi_{j_1}(x_k) \, \mathrm{d}x_k \,.\, \int_{x_l} {x_l}^2 \phi_{i_2}(x_l) \phi_{j_2}(x_l) \, \mathrm{d}x_l ..... \\
&..... - 2 \int_{x_k} x_k \phi_{i_1}(x_k) \phi_{j_1}(x_k) \, \mathrm{d}x_k \,.\, \int_{x_l} x_k \phi_{i_2}(x_l) \phi_{j_2}(x_l) \, \mathrm{d}x_l \lambda_{i_2 j_2} \,.\, \int_{x_k} {x_k}^2 \phi_{i_1}(x_k) \phi_{j_1}(x_k) \, \mathrm{d}x_k ..... \\
&..... + \lambda_{i_1 j_1} \,.\, \int_{x_l} {x_l}^2 \phi_{i_2}(x_l) \phi_{j_2}(x_l) \, \mathrm{d}x_l - 2 \int_{x_k} x_k \phi_{i_1}(x_k) \phi_{j_1}(x_k) \, \mathrm{d}x_k \,.\, \int_{x_l} x_l \phi_{i_2}(x_l) \phi_{j_2}(x_l) \, \mathrm{d}x_l
\end{aligned}$$

Denoting the expression of form $\int x \phi_i(x) \phi_j(x) \, \mathrm{d}x$ as $I_{ij}(1)$ and $\int x^2 \phi_i(x) \phi_j(x) \, \mathrm{d}x$ as $I_{ij}(2)$ we can further simplify as

$$V_{i_1 i_2 j_1 j_2} = K(\, \delta_{i_2 j_2} \,.\, I_{i_1 j_1}(2) \, + \, \delta_{i_1 j_1} \,.\, I_{i_2 j_2}(2) - 2 \, I_{i_1 j_1}(1) \,.\, I_{i_2 j_2}(1) \,)$$

Integral $I(1)$ and $I(2)$ can be further simplified as

$$I(1) := \int x\phi_i(x)\phi_j(x)\,\mathrm{d}x = \int x \sum_k c_k e^{-\frac{(x-\alpha k-\beta i)^2}{2\sigma}} \,.\, \sum_k c_k e^{-(x-\alpha k-\beta j)^2}\,\mathrm{d}x$$
$$= \int x \sum_{kl} c_k c_l e^{-\frac{(x-\alpha k-\beta i)^2}{2\sigma}} e^{-\frac{(x-\alpha l-\beta j)^2}{2\sigma}}\,\mathrm{d}x = \sum_{kl} c_k c_l \int x e^{-\frac{(x-\alpha k-\beta i)^2}{2\sigma}} e^{-\frac{(x-\alpha l-\beta j)^2}{2\sigma}}\,\mathrm{d}x$$

Once again, the product of two Gaussians is also a Gaussian, and the constant terms can be separated out, leading to integrals of the form

$$I_{ij}(1) := \sum_{kl} c_k c_l \; e^{-(\lambda^2_{i-j,kl}/4\sigma)} \int_{-\infty}^{-\infty} (t + \mu_{i+j,kl}) e^{-t^2/\sigma}\,\mathrm{d}t \tag{A.2}$$

where

$$\lambda_{i-j,kl} := \alpha(l-k) + \beta(i-j)$$
$$\mu_{i+j,kl} := \frac{\alpha(l+k) + \beta(i+j)}{2}$$
$$t := x - \mu_{i+j,kl}$$

similarly we have

$$I_{ij}(2) := \sum_{kl} c_k c_l \; e^{-(\lambda_{i-j,kl}/2)^2} \int_{-\infty}^{-\infty} (t + \mu_{i+j,kl})^2 e^{-t^2}\,\mathrm{d}t$$

The t in (A.1) and (3.7) is not indexed like λ and μ as t is a continuous variable ranging from $-\infty$ to ∞ in the integral. The integrals evaluate to -

$$\int_{-\infty}^{\infty} t^2 e^{-t^2}\,\mathrm{d}t = \sqrt{\pi}/2$$
$$\int_{-\infty}^{\infty} e^{-t^2}\,\mathrm{d}t = \sqrt{\pi}$$
$$\int_{-\infty}^{\infty} (t-\mu) e^{-t^2}\,\mathrm{d}t = -\mu\sqrt{\pi}$$
$$\int_{-\infty}^{\infty} (t-\mu)^2 e^{-t^2}\,\mathrm{d}t = \sqrt{\pi}(\frac{1}{2} + \mu^2)$$

We observe that the single electron component $\langle\phi_i|\, x_k^2\, |\phi_j\rangle$, which we left for simplification, is nothing but $I_{ij}(2)$, therefore, the final expressions of single electron and two-electron components inthe harmonium model will be given by

$$h_{ij} = \frac{\hbar^2}{2m}(J_{i-j}(0) - J_{i-j}(2)) + \frac{1}{2}m\omega^2 I_{ij}(2) \tag{A.3}$$
$$V_{i_1 i_2 j_1 j_2} = K(\; \delta_{i_2 j_2} \,.\, I_{i_1 j_1}(2) \;+\; \delta_{i_1 j_1} \,.\, I_{i_2 j_2}(2) - 2\; I_{i_1 j_1}(1) \,.\, I_{i_2 j_2}(1)\;) \tag{A.4}$$

with

$$I_{ij}(1) := \sqrt{\pi}\left\{\sum_{kl} c_k c_l \mu_{i+j,kl}\ e^{-(\lambda^2_{i-j,kl}/4\sigma)}\right\}$$

$$I_{ij}(2) := \sqrt{\pi}\left\{\sum_{kl} c_k c_l (\frac{1}{2} + {\mu_{i+j,kl}}^2)\ e^{-({\lambda_{i-j,kl}}^2/4\sigma)}\right\}$$

$$J_{i-j}(0) := \sqrt{\pi}\left\{\sum_{kl} (\lambda_{i-j,kl}/2\sigma)^2\ .\ c_k c_l\ .\ e^{-(\lambda^2_{i-j,kl}/4\sigma)}\right\}$$

$$J_{i-j}(2) := \frac{\sqrt{\pi}}{2}\left\{\frac{1}{\sigma^2}\sum_{kl} c_k c_l .\ e^{-(\lambda^2_{i-j,kl}/4\sigma)}\right\}$$

Bibliography

[1] J. O. Trygve Helgaker Poul Jørgensen. *Molecular Electronic-Structure Theory*. John Wiley Sons, Ltd, 2000.

[2] Hand-waving and interpretive dance: an introductory course on tensor networks. "The Prism Algorithm for Two-Electron Integrals". In: (2017).

[3] N. F. Johnson and M. C. Payne. "Exactly Solvable Model of Interacting Particles in a Quantum Dots". In: (1991).

[4] M. Razeghi and M. Henini. *Optoelectronic Devices: III Nitrides: Chapter 4 - Technology of MOVPE Production Tools*. Elsevier, 2005.

[5] S. T. et al. "Electronic states in quantum dot atoms and molecules". In: (1998).

[6] D. J. W. et al. "Atomic-Ion Coulomb Clusters in an Ion Trap". In: (1987).

[7] D. J. Griffiths. "Resource Letter EM-1: Electromagnetic Momentum". In: (2011).

[8] S. F. . Boys. "A general method of calculation for the stationary states of any molecular system". In: (1950).

[9] L. E. MCMURCHIE and E. R. D. De. "One- and Two-Electron Integrals over Cartesian Gaussian Functions". In: (1977).

[10] M. H.-G. PETER M. W. GILL and J. A. POPLE. "An Efficient Algorithm for the Generation of Two-Electron Repulsion Integrals over Gaussian Basis Functions". In: (1989).

[11] *. M. H.-G. Peter M. W. Gill and J. A. Pople. "Efficient Computation of Two-Electron-Repulsion Integrals and Their nth-Order Derivatives Using Contracted Gaussian Basis Sets". In: (1989).

[12] P. M. GILL and J. A. POPLE. "The Prism Algorithm for Two-Electron Integrals". In: (1990).

[13] G. Evenbly and S. R. White. "Representation and design of wavelets using unitary circuits". In: (2018).

[14] S. R. White. "Hybrid grid/basis set discretizations of the Schrödinger equation". In: *Journal of Chemical Physics* (2017).

[15] J. W. W. Steven R. White. "Finite-element method for electronic structure". In: *Physical Review B* (1988).

[16] R. A. Brualdi. *Combinatorial Matrix Classes*. Cambridge University Press, 2006.

[17] J. B. e. a. Martin Ganahl1. "Density Matrix Renormalization Group with Tensor Processing Units". In: *American Physical Society, Physical Review Journals* (2023).

[18] G. T. C. e. a. Alec Kirkley. "Belief propagation for networks with loops". In: *science advances* (2021).

[19] e. a. Chu Guo Dario Poletti. *Block belief propagation algorithm for two-dimensional tensor networks*. 2023.

[20] Y. E. Z. e. a. Yijia Wang. *Tensor Network Message Passing*. 2023.