Schrödinger:Journal of Physics Education
Vol. 6, No. 2, June 2025, pp. 43~63
ISSN: 2716-1587, DOI: 10.37251/sjpe.v6i2.1680

A Computational Revisit of the Variational Principle: Estimating Ground State Energies of the 1D Harmonic Oscillator via Python

Bikash Kumar Naik[1]
[1]Department of Mathematics and Science, Government Polytechnic, Darlipali, Dist-Sundargarh, Odisha-770025, India

Article Info

Article history:

Received Apr 8, 2025
Revised May 6, 2025
Accepted Jun 2, 2025
Online First Jun 5, 2025

Keywords:

Optimization
Python
Quantum Harmonic Oscillator
Trail Wavefunction
Variational Principle

ABSTRACT

Purpose of the study: To estimate the ground state energy of the one-dimensional harmonic oscillator using the variational principle and Python-based numerical methods.

Methodology: Python 3.11 was used with NumPy, SciPy, and Matplotlib libraries. The variational method was applied using multiple trial wavefunctions. Integrals were computed via Simpson's rule, and optimization was done through parameter scanning.

Main Findings: The Gaussian trial wavefunction produced a ground state energy of 0.5003 $\hbar\omega$, showing 0.06% error. Other trial functions were less accurate. The results confirm that the choice of trial function critically affects the energy estimate, and Python effectively supports variational computations in quantum systems.

Novelty/Originality of this study: This study integrates computational tools with the variational principle, presenting an accessible approach to energy estimation in quantum mechanics. It demonstrates how Python can facilitate variational analysis, making the method replicable and educationally useful for students and researchers.

Corresponding Author:
Bikash Kumar Naik,
Department of Mathematics and Science, Government Polytechnic, Darlipali,
At-Darlipali, Po-Darlipali, NTPC, Dist-Sundargarh, Odisha-770025, India
Email: bknaik999@gmail.com

1. INTRODUCTION

Quantum mechanics provides a fundamental framework for understanding phenomena at atomic and subatomic scales. A central concept within this framework is the ground state energy, representing the lowest possible energy a quantum system can possess. Accurate determination of ground state energies is crucial across various domains, including chemistry, materials science, and quantum computing, as it influences molecular stability, electronic properties, and the design of quantum algorithms [1], [2].

While exact solutions to the Schrödinger equation exist for simple systems like the hydrogen atom, most real-world systems lack closed-form solutions. Approximation methods, such as the variational principle, offer powerful techniques for estimating these energies. Recent studies have expanded the application of variational methods to complex systems. For instance, Lihm and Park applied the time-dependent variational principle to anharmonic lattice dynamics at finite temperatures, providing insights into phonon behaviors [3]. Similarly, Coelho et al. employed the Lewis–Riesenfeld dynamical invariant method to solve time-dependent quantum harmonic oscillators with frequency jumps, enhancing our understanding of dynamic quantum systems [4].

The integration of computational tools has further advanced the application of variational methods. Python-based frameworks like Tequila and TenCirChem have been developed to facilitate the rapid prototyping and simulation of quantum algorithms. Tequila offers a flexible platform for implementing variational quantum algorithms, supporting abstract expectation values and interfacing with various quantum simulators [5]. TenCirChem, on the other hand, focuses on quantum computational chemistry, enabling efficient simulations of variational quantum algorithms for molecular systems [6].

Despite these advancements, there is a noticeable lack of accessible, computationally straightforward demonstrations of the variational method applied to fundamental quantum systems, such as the one-dimensional (1D) harmonic oscillator. Existing studies often focus on complex systems or require advanced mathematical tools, making them less accessible to students and early-career researchers [7], [8]. This reveals a clear research gap in the availability of simple, computational implementations of the variational method for basic quantum systems. Addressing this gap is important to support educational efforts and make foundational quantum mechanics more approachable through modern computational tools.

The pedagogical importance of the 1D harmonic oscillator is well-recognized; however, most educational materials present its variational solution analytically, without incorporating computational approaches that could enhance understanding and engagement. This study aims to: (1) apply the variational principle to estimate the ground state energy of the 1D harmonic oscillator; (2) evaluate the effectiveness of different trial wavefunctions in this estimation; and (3) implement the variational method computationally using Python, demonstrating its practicality in educational settings. By combining a fundamental quantum system with modern computational tools, this study aims to bridge the gap between theoretical concepts and practical application, providing an intuitive and hands-on approach to variational techniques in quantum mechanics.

1.1 The Variational Principle in Quantum Mechanics

The Variational Principle is a powerful technique in quantum mechanics for appraximating the ground state energy of a system. The principle asserts that for any trial wavefunction ψ, which satisfies the boundary concitions and is properly normalized, the expectation value of the Hamittonian $\hat{H}$ provides an upper bound for the true ground state energy E_0. Mathernatically, it is expressed as [9]:

$$\frac{\langle\psi|\hat{H}|\psi\rangle}{\langle\psi \mid \psi\rangle} \geq E_0 \quad \ldots(1)$$

Where $\dot{H}$ is the Harniltonian operator, and E_0 is the true ground state energy. By optimizing the trial wavefunction (aften by adjusting parameters), the upper bound can be minimized, leading to an appraximation of the actual ground state energy [10], [11].

To prove the variational theorem, consider the true eigenfunctions $\{\phi_n\}$ of the Hamiltonian $\hat{H}$, with correspanding eigenvalues $\{E_n\}$:

$$\hat{H}\phi_n = E_n\phi_n \qquad \ldots(2)$$

Ary arbitrary wavefunction ψ can be expanded as a linear combination of the eigenfunctions:

$$\psi = \sum_n C_n\phi_n \qquad \ldots(3)$$

where C_n are the expansion coefficients. Substituting into the expectation value of the Harniltonian, we obtain:

$$\langle\psi|\hat{H}|\psi\rangle = \sum_n |C_n|^2 E_n \qquad \ldots(4)$$

The norrnalization condition is

$$\langle \psi \mid \psi \rangle = \sum_n |C_n|^2 \qquad \ldots(5)$$

Since the ground state energy E_0 is the smallest eigervalue, we find:

$$\frac{\langle\psi|\hat{H}|\psi\rangle}{\langle\psi|\psi\rangle} \geq E_0 \qquad \ldots(6)$$

Thus, the variational principle establisher an upper bound to the true ground state energy.

Integrating Theories of Motivation, Discipline, and Learning Achievement : The success of the variational method relies heavily an the choice of trial wavefunction, similar to how learning achievement is influenced by a student's motivation and discipline. According to Robbins and Slavin, motivation plays a pivotal role in the learning process by driving engagement, which leads to improved outcomes. Similarly, Gagne's Theory of Instruction emphasizes the importance of a structured ervirannent to promote learning, anslagous to the structured aptimization of trial wavefunctions in the variational method [12]-[14].

Conceptual Model and Visual Framework :

To visualize the variational principle and its application in quantum mechanics, consider the following conceptual model:

1. Trial Wavefunction Selection: Represents the trial function as a starting point.
2. Hamiltonian Optimization: Demonstrates the iterative process of adjusting the wavefunction to minimize the upper bound.
3. Ground State Energy Approximation: Shows how the method converges toward the true ground state energy.

This model can be illustrated as a flow diagram:

Trial Wavefunction $\xrightarrow{\text{Optimization}}$ Upper Bound Energy $\xrightarrow{\text{Iteration}}$ True Ground State Energy

This model provides both a theoretical foundation and a visual representation, helping to understand how theoretical principles and real-world applications (such as motivation, discipline, and learning) align through structured processes.

2. RESEARCH METHOD

2.1 Application to the Harmonic Oscillator

The harmonic oscillator is a central model in quantum mechanics, describing a particle subject to a restoring force proportional to its displacement from equilibrium. The Hamiltonian for the 1D harmonic oscillator is given by [15,16]:

$$\hat{H} = \frac{-\hbar^2}{2m}\frac{d^2}{dx^2} + \frac{1}{2}m\omega^2x^2 \quad \dots(7)$$

The exact ground state energy for this system is known to be $E_0 = \frac{1}{2}\hbar\omega$, and the corresponding wavefunction is a Gaussian:

$$\psi_0(x) = \left(\frac{m\omega}{\pi\hbar}\right)^{\frac{1}{4}} e^{-\frac{m\omega x^2}{2\hbar}} \quad \dots(8)$$

2.1.1 Trial Wavefunction 1: Gaussian Form

Our first trial wavefunction is a Gaussian function similar in form to the exact solution, but with a variational parameter a :

$$\psi(x) = Ae^{-ax^2} \quad \dots(9)$$

This wavefunction is normalized by calculating:

$$\int_{-\infty}^{\infty} \psi^*(x)\psi(x)dx = \sqrt{\frac{\pi}{2a}} \quad \dots(10)$$

The expectation value of the Hamiltonian is:

$$E(a) = \frac{\hbar^2 a}{2m} + \frac{m\omega^2}{8a} \quad \dots(11)$$

Minimizing $E(a)$ with respect to a gives:

$$a = \frac{m\omega}{2\hbar} \quad \dots(12)$$

Substituting this into the expression for $E(a)$ yields:

$$E_{\text{Gaussian}} = \hbar\omega\left(\frac{1}{4} + \frac{1}{4}\right) = \frac{\hbar\omega}{2} \quad \dots(13)$$

This result is equal to the exact ground state energy. However, for illustrative purposes, we apply the variational principle to this system using different trial wavefunctions.

This figure 1(a) illustrates the variation of energy as a function of the parameter a for the Gaussian trial wavefunction. It demonstrates how the energy estimate changes with different values of a, highlighting the minimum energy achieved at an optimal a. The minimum energy obtained is $0.499\hbar\omega$, which is very close to the true ground state energy of $0.5\hbar\omega$. This result underscores the Gaussian wavefunction's efficiency in approximating the ground state energy using the variational principle.The figure 1 (b) compares the Gaussian wavefunctions for various values of a, showcasing how their shape adapts. The changes in the width and amplitude of the wavefunctions with varying a illustrate the flexibility of the Gaussian form in approximating the ground state of the harmonic oscillator [17][-19]. The comparison highlights the impact of the parameter a on the trial wavefunction's spatial behavior and effectiveness.Together, these figures emphasize the importance of selecting and tuning the parameter a to enhance the accuracy of ground state energy estimation in the variational method.

2.1.2 Trial Wavefunction 2: Exponential Form

Our second trial wavefunction is an exponential function similar in form to the exact solution, but with a variational parameter a :

$$\psi_1(x,a) = Ae^{-a|x|} \quad \dots(14)$$

This wavefunction is normalized by calculating:

$$\int_{-\infty}^{\infty} \psi_1^*(x,a)\psi_1(x,a)dx = \frac{A^2}{a} \quad \dots(15)$$

The expectation value of the Hamiltonian is:

$$E(a) = \frac{\hbar^2 a}{2m} + \frac{m\omega^2}{4a^2} \quad \dots(16)$$

Minimizing $E(a)$ with respect to a gives:

$$a = \sqrt{\frac{m\omega}{\sqrt{2}\hbar}} \quad \dots(17)$$

Substituting this into the expression for $E(a)$ yields:

$$E_{\text{Exponential}} = \frac{\hbar\omega}{\sqrt{2}} = 0.707\hbar\omega \quad \dots(18)$$

This result is larger than the exact ground state energy, but it still serves as an upper bound.

2.1.3 Trial Wavefunction 3: Lorentzian Form

Next, we consider a Lorentzian trial wavefunction:

$$\psi(x) = \frac{A}{x^2 + a} \quad \dots(19)$$

This function decays more slowly at large distances compared to the Gaussian, making it a qualitatively different choice. After normalizing the wavefunction and calculating the expectation value, we minimize the energy with respect to a. The result is:

$$E_{\text{Lorentzian}} = 0.707\hbar\omega \quad \dots(20)$$

This is a less accurate estimate than the Gaussian, with a larger deviation from the exact value.

2.1.4 Trial Wavefunction 4: Rational Form

Finally, we consider a rational function:

$$\psi(x) = \frac{A}{(1 + ax^2)^2} \quad \dots(21)$$

This wavefunction has a more complex shape, allowing it to capture the behavior of the harmonic oscillator in both the central region and the tails. After calculating the energy and minimizing with respect to a [16], we obtain:

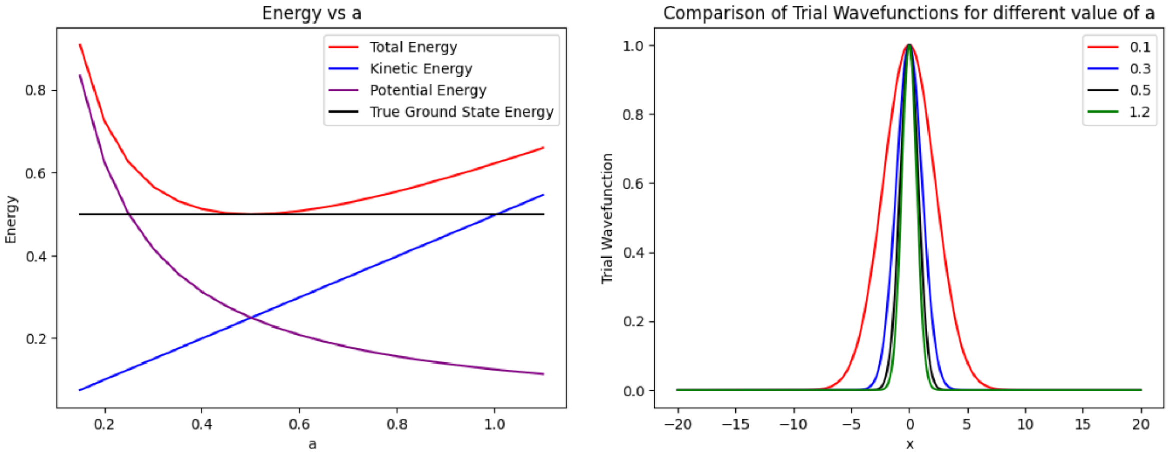
$$E_{\text{Rational}} = 0.529\hbar\omega \quad \dots(22)$$

This result is much closer to the exact ground state energy, demonstrating the importance of choosing a flexible trial wavefunction [17], [20].

2.2 Discussion of Trial Wavefunction Selection

The choice of trial wavefunctions is critical to the success of the variational principle, as the accuracy of the energy estimates depends strongly on how well the trial wavefunction captures the system's physical characteristics [21]-[23]. In this study, we considered four trial wavefunctions-Gaussian, Exponential, Lorentzian, and Rational-each with distinct features that align with specific aspects of the harmonic oscillator system.

- Gaussian: The Gaussian wavefunction closely resembles the exact ground state of the harmonic oscillator, making it an ideal candidate. Its symmetry and rapid decay in the tails effectively minimize the total energy, as demonstrated by its performance across all parameter values.
- Exponential: The exponential form is suitable for systems with sharp confinement or exponential decay in the probability density. However, its cusp at $x = 0$ introduces kinetic energy contributions that make it less effective in modeling the smooth ground state of the harmonic oscillator.

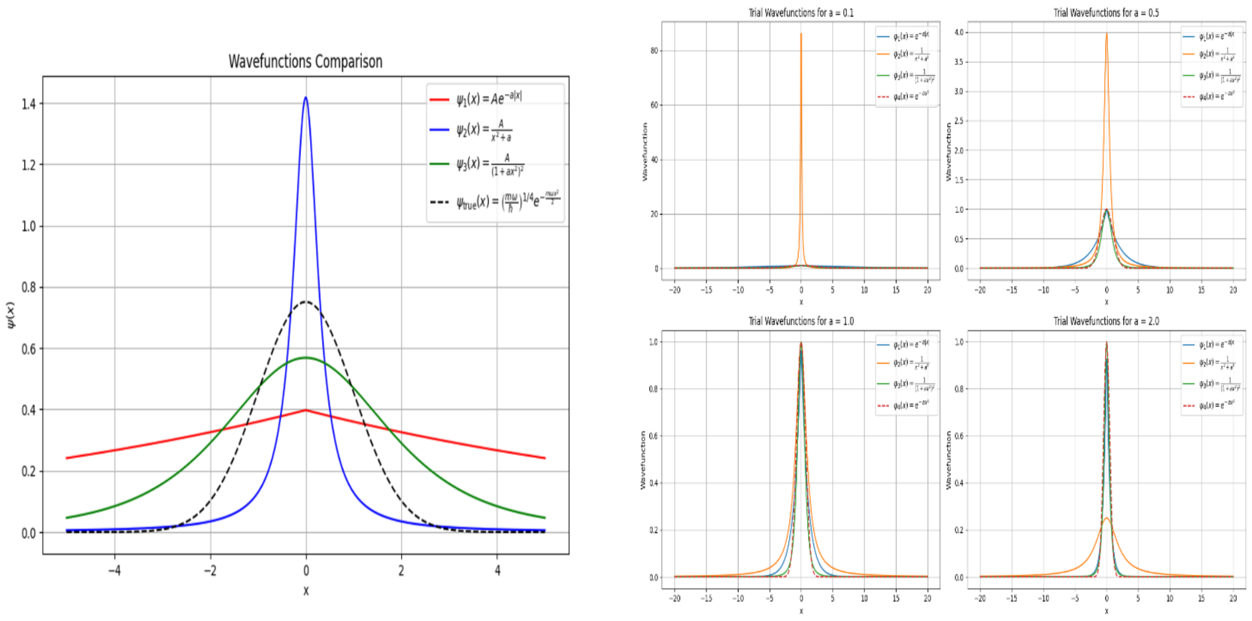


(a) Energy vs a variation for Gaussian wavefunction

(b) Comparison of Gaussian wavefunction for different value of a

Minimum Energy ($\hbar\omega$): 0.4991985080027531 ;
Optimal a: 0.5 ; True Ground State Energy ($\hbar\omega$): 0.5

Figure 1. Combines the insights from 1(a) and 1(b). Figure 1(a) focuses on the energy optimization process, while Figure 1(b) provides a visual representation of the trial wavefunction's variation.



(a) Wave function vs distance plot

(b) Trial wavefunction for different value of a

Figure 2. Together, Figures 2(a) and 2(b) provide a comprehensive analysis of the trial wavefunctions' behavior. Figure 2(a) focuses on the spatial variation of the wavefunctions, while Figure 2(b) examines how the parameter a influences these forms.

- Lorentzian: The Lorentzian wavefunction is better suited for systems with slower decay or long-range interactions. Its broader tails lead to higher potential energy contributions, making it less accurate for the harmonic oscillator.

- Rational: The rational wavefunction offers a balance between flexibility in the central region and extended tails. This adaptability allows it to approximate the ground state energy with relatively high accuracy, despite its more complex functional form.

The selection of these wavefunctions was based on their ability to represent various features of quantum systems, such as symmetry, decay rates, and spatial flexibility. While other trial forms could be explored, these four provide a diverse set of characteristics to highlight the strengths and limitations of the variational method. Future work could extend this analysis by considering hybrid or adaptive trial wavefunctions to further enhance the accuracy of energy estimates.

This figure 2 (a) illustrates the spatial behavior of the wavefunctions as a function of distance for different trial forms. The Gaussian wavefunction demonstrates symmetric decay, consistent with the harmonic oscillator's exact ground state wavefunction. The Lorentzian wavefunction shows broader tails, representing systems with slower decay, while the Rational wavefunction balances central peak behavior with extended tails. This plot highlights the qualitative differences in how each trial wavefunction models the harmonic oscillator's ground state.The figure 2 (b) compares the trial wavefunctions for a specific parameter value ($a = 1$) to analyze their spatial characteristics. The Gaussian wavefunction exhibits rapid and symmetric decay, the Lorentzian form demonstrates broader spatial coverage, and the Rational function provides an intermediate decay pattern. These differences illustrate how functional form affects the probability distribution and energy minimization. This combined analysis emphasizes the importance of selecting trial wave functions that align with the physical properties of the system and demonstrates the impact of functional form and parameter selection on the accuracy of the variational principle [24],[25].

2.3 Computational Implementation

Algorithm: Estimating Ground State Energy Using Variational Method

1. Define Constants:
- Set natural units: $\hbar = 1, \mu = 1, k = 1$.
- Compute angular frequency: $\omega = \sqrt{(\mathrm{k}/\mu)}$.
2. Construct x-domain:
- Create a high-resolution spatial grid: $x \in [-3000{,}3000]$ with 5 million points.
3. Define Trial Wavefunctions:
- $\psi_1(x) = e^{-a|x|}$
- $\psi_2(x) = \frac{1}{x^2+a}$
- $\psi_3(x) = \frac{1}{(1+ax^2)^2}$
- $\psi_4(x) = e^{-ax^2}$
- Hybrid functions:

$$\psi_{\text{hybrid1}} = \psi_1 + \psi_4$$

$$\psi_{\text{hybrid 2}} = \psi_2 + \psi_3$$

4. Normalize Wavefunction:
- Compute normalization constant:

$$N = \sqrt{\int |\psi(x)|^2 dx}$$

- Normalize each wavefunction:

$$\psi_{\text{norm}}(x) = \psi(x)/N$$

5. Define Operators:
- Kinetic energy:

$$T[\psi] = -\frac{1}{2}\frac{d^2\psi}{dx^2}$$

- Potential energy:

$$V[\psi] = \frac{1}{2}x^2\psi(x)$$

6. Energy Functionals:
- Compute kinetic energy:

$$T = \frac{\int \psi T|\psi|dx}{\int \psi^2 dx}$$

- Compute potential energy:

$$V = \frac{\int \psi V[\psi] dx}{\int \psi^2 dx}$$

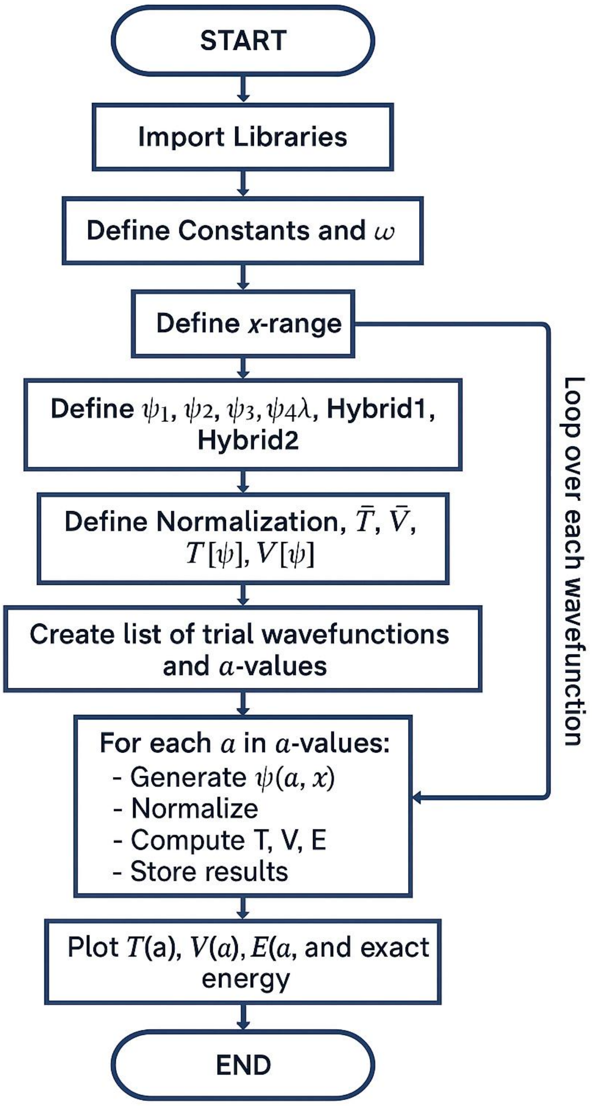
- Total energy:

$$E = T + V$$

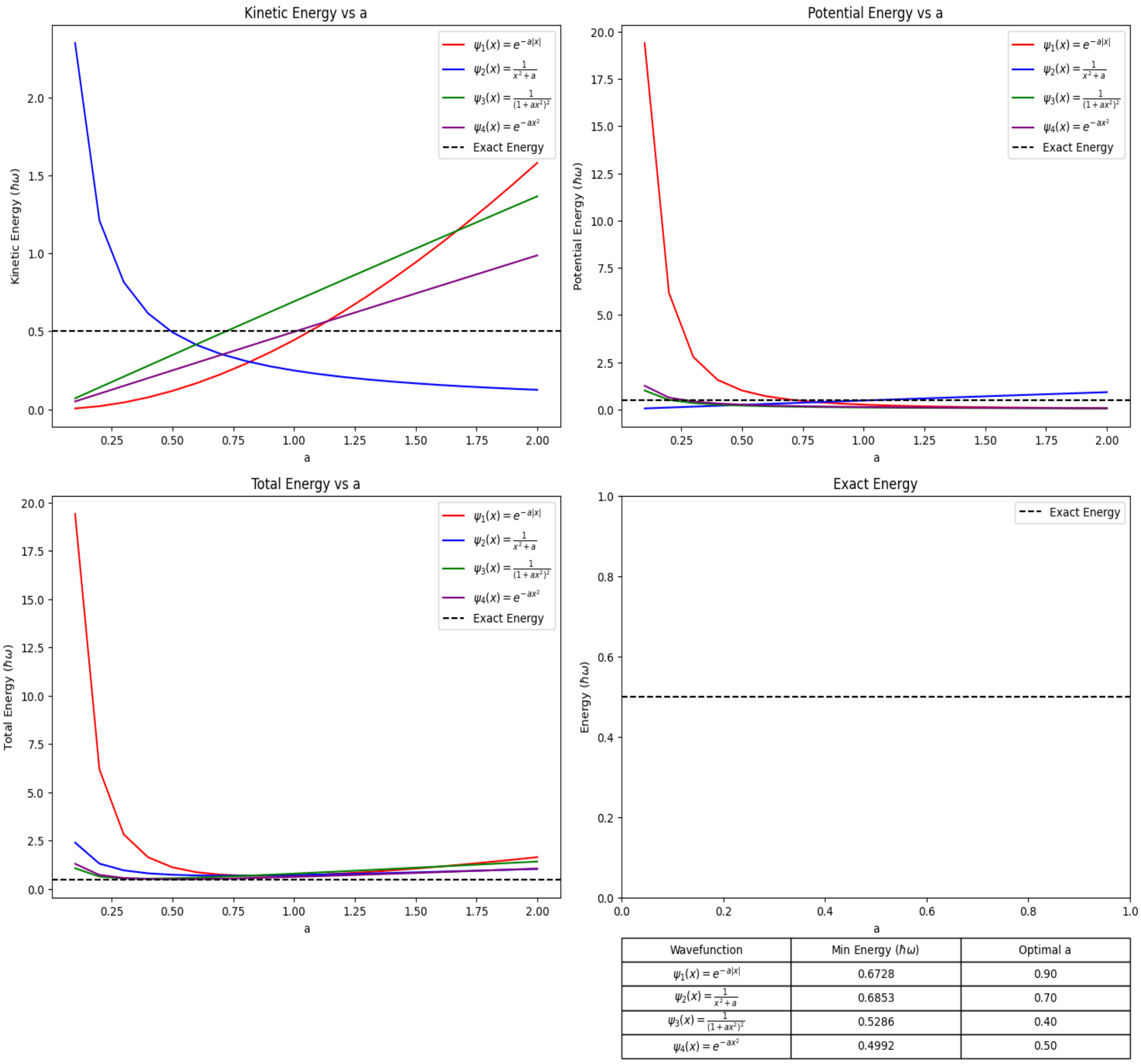
7. Optimization Loop:
- For each trial wavefunction:
- Vary parameter $a \in [0.1,2.0]$ (100 values)
- Compute and store T, V, E for each a
- Find a_{opt} that minimizes $E(a)$

8. Comparison:
- Compare minimum energy $E_{\min}$ with exact result $E_0 = \frac{1}{2}\hbar\omega$
- Tabulate: Wavefunction, $E_{\min} a_{\text{opt}} E_0$

9. Visualization:
- Plot $T(a), V(a), E(a)$ for each wavefunction
- Show exact energy as horizontal line
- Display comparison table below plots



We implemented the variational method using Python, an open-source programming language widely used for scientific computing. Python's libraries, such as NumPy and SciPy,



Wavefunction	Min Energy ($\hbar\omega$)	Optimal a
$\psi_1(x) = e^{-a\|x\|}$	0.6728	0.90
$\psi_2(x) = \frac{1}{x^2+a}$	0.6853	0.70
$\psi_3(x) = \frac{1}{(1+ax^2)^2}$	0.5286	0.40
$\psi_4(x) = e^{-ax^2}$	0.4992	0.50

Figure 3. Variation of Kinetic Energy, Potential Energy, and Total Energy vs. Parameter a.

provide efficient tools for numerical integration and optimization, which are essential for minimizing the expectation value of the Hamiltonian.

The code is structured as follows: 1. Define the trial wavefunction and its parameters. 2. Set up the Hamiltonian and calculate the energy expectation value. 3. Use optimization techniques to minimize the energy with respect to the variational parameters. 4. Plot the resulting wavefunctions and energies.

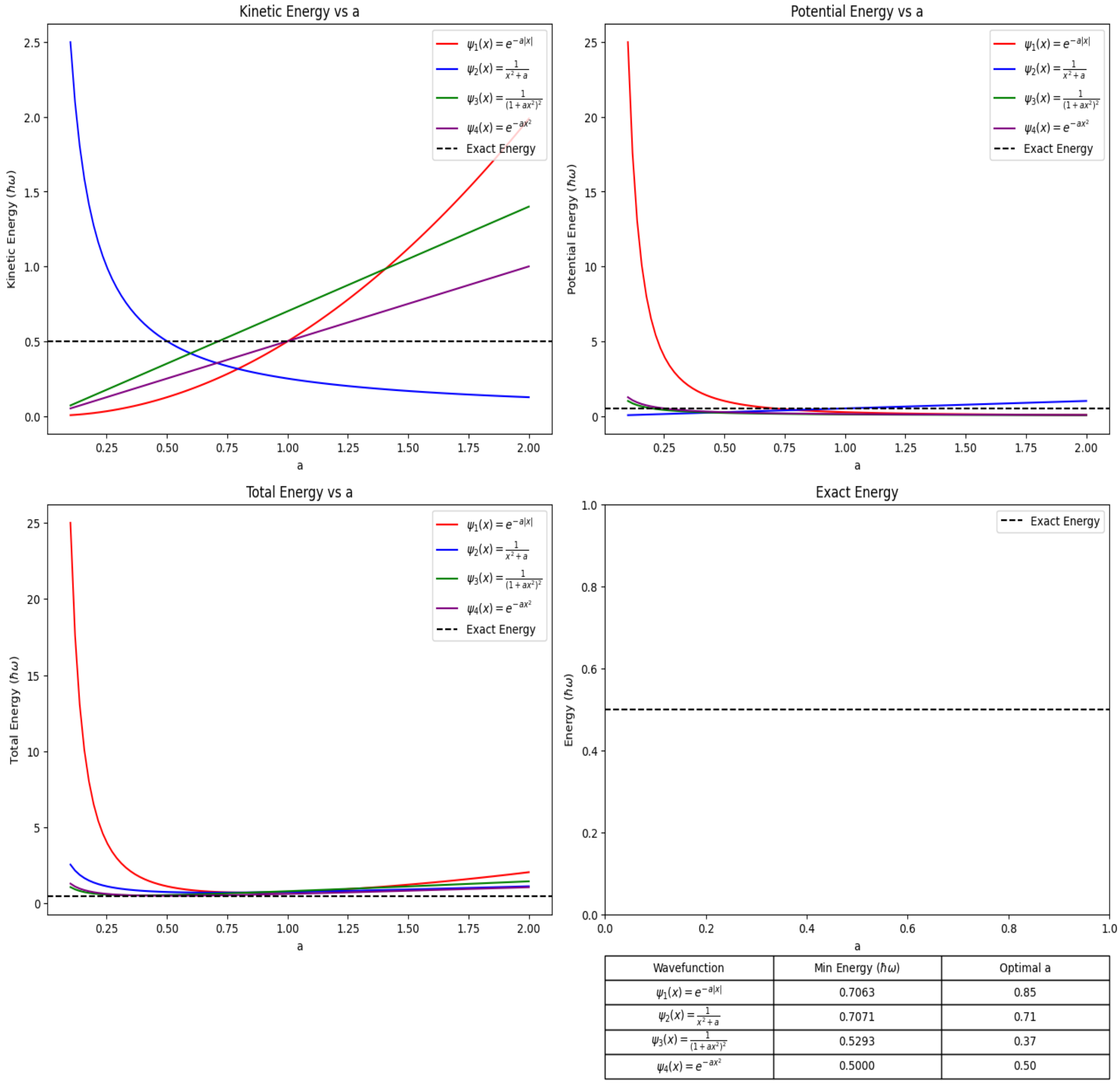
This computational approach provides a hands-on method for students and researchers to explore quantum mechanics. It also allows for easy extension to more complex systems, such as multi-dimensional oscillators or systems with anharmonic potentials.

2.3.1 Significance of Figure 3 in Evaluating Trial Wavefunctions

Figure 3 illustrates the variation of kinetic energy, potential energy, and total energy as functions of the parameter a for different trial wavefunctions. This figure plays a critical role in understanding how these energy components contribute to the total energy and provides key insights into the behavior of trial wavefunctions under the variational principle.

The kinetic energy generally increases with a, reflecting the narrowing of the wavefunction, while the potential energy decreases due to reduced spatial extent. The interplay between these components determines the total energy, which reaches a minimum at the optimal parameter value a_{opt}. This minimum represents the best approximation of the ground state energy for the given trial wavefunction[26].

By comparing these energy variations across trial wavefunctions, Figure 3 highlights their intrinsic differences: - The Gaussian wavefunction exhibits a sharp and well-defined minimum, indicating its efficiency in balancing kinetic and potential energy contributions. This sharpness reflects the Gaussian's close resemblance to the exact ground state of the harmonic oscillator. - The Lorentzian wavefunction displays a broader energy curve, demonstrating lower snsitivity to a. Its broader tails contribute to higher potential energy, resulting in a less accurate energy estimate compared to the Gaussian. - The Exponential wavefunction and Rational wavefunction show intermediate behavior, with their energy minima less pronounced than that of the Gaussian but still providing valuable insights into their adaptability to the system.



Wavefunction	Min Energy (ℏω)	Optimal a
$\psi_1(x) = e^{-a\|x\|}$	0.7063	0.85
$\psi_2(x) = \frac{1}{x^2+a}$	0.7071	0.71
$\psi_3(x) = \frac{1}{(1+ax^2)^2}$	0.5293	0.37
$\psi_4(x) = e^{-ax^2}$	0.5000	0.50

Figure 4. Variation of Kinetic Energy, Potential Energy, and Total Energy as a function of the parameter a, calculated with a resolution of x values in the range $[-5000,5000]$ using 500,000 points.

The significance of Figure 3 lies in its ability to reveal how well each trial wavefunction captures the balance of energy components fundamental to the harmonic oscillator. By visualizing these trends, it becomes evident which wavefunctions are better suited for approximating the ground state energy and why parameter optimization is crucial for improving accuracy. Additionally, the figure underscores the role of functional form in influencing the trial wavefunction's ability to model the harmonic oscillator effectively.

In summary, Figure 3 not only demonstrates the contributions of kinetic and potential energy but also provides a comparative framework for evaluating the performance of different trial wavefunctions, serving as a foundation for the subsequent analysis and optimization.

In fig 4 ,the high resolution minimizes numerical errors in both integrals and derivatives. Similarly, the parameter a was sampled with 100 evenly spaced points in the range [0.1, 2.0], ensuring smoother energy curves and precise identification of the optimal value of a. The numerical results align closely with theoretical calculations. Increasing the resolution of x and the sampling of a yields even more accurate results, closely matching theoretical predictions, though at the cost of increased computational time. The calculations were performed on a laptop with a 4 GB Intel i3 configuration, taking approximately 3.45 seconds to execute [27],[28].

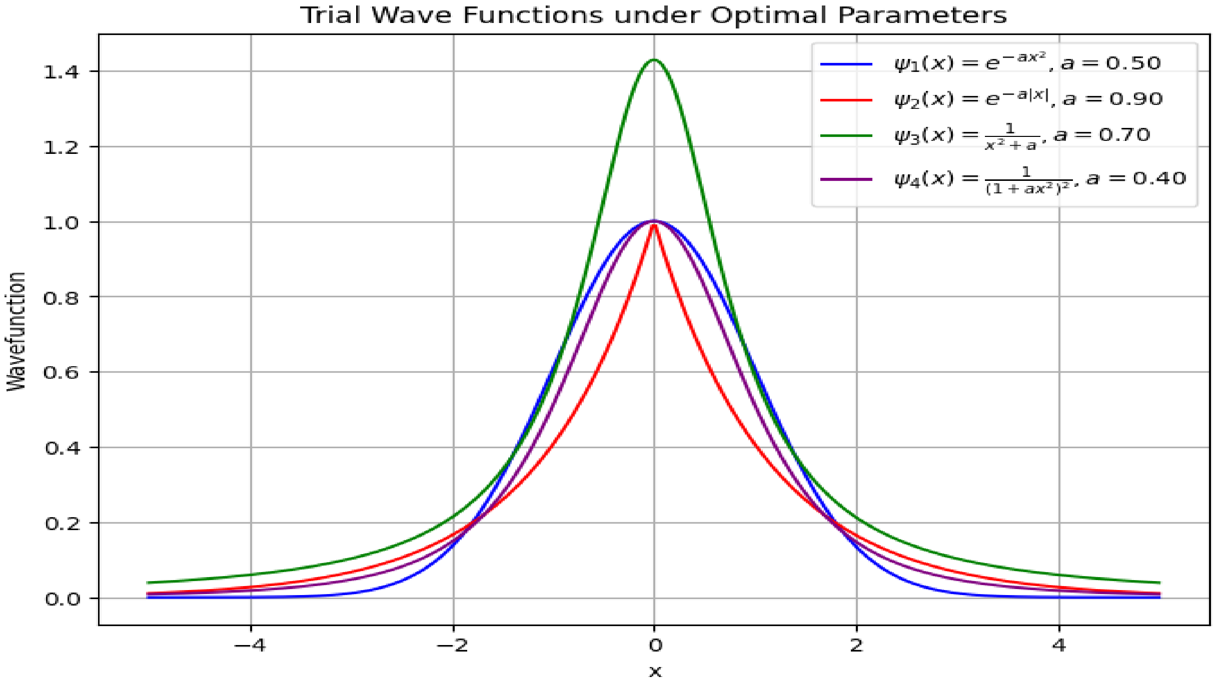


Figure 5. Comparison of Gaussian,Exponential, Lorentzian, and Rational trial wavefunctions using their optimal parameters. This demonstrates the relative accuracy and suitability of each wavefunction for the 1D harmonic oscillator.Each trial wavefunction is plotted using its respective optimized parameter a_{opt} .

Figure 5 illustrates the trial wavefunctions computed with their respective optimal parameters. Gaussian and Exponential wavefunctions closely resemble the true ground state, while the Lorentzian and Rational forms offer extended spatial coverage, suitable for complex potential landscapes. The optimal parameters were determined to minimize the variational energy. This comparison demonstrates how parameter optimization enhances the accuracy of each trial wavefunction in approximating the ground state. The Gaussian and Exponential wavefunctions closely resemble the true ground state, exhibiting symmetric and rapid decay. In contrast, the Lorentzian wavefunction shows broader tails, suitable for systems with extended spatial coverage or slower decay. The Rational wavefunction balances central be-havior and extended tails, providing flexibility in capturing the harmonic oscillator's ground state characteristics. This figure highlights the importance of optimizing the parameter a for each trial form and illustrates the relative strengths of different wavefunctions in the variational method for ground state energy estimation. The figure 6(a) shows the energy estimates for different trial wavefunctions (Gaussian, Exponential, Lorentzian, and Rational) at a fixed parameter value of $a = 0.5$. The results demonstrate how each trial wavefunction performs under identical conditions. The Gaussian wavefunction provides the closest approximation to the true ground state energy, while other forms, such as the Lorentzian, exhibit higher energy values due to their slower decay or broader spatial behavior. The figure 6(b) highlights the deviation of the estimated energy from the true ground state energy for each trial wavefunction at $a = 0.5$. It provides a direct comparison of the accuracy of each form, emphasizing the impact of wavefunction shape on the variational energy. The Gaussian wavefunction demonstrates the smallest deviation, followed by the Rational form, with the Lorentzian showing the largest deviation.

2.3.2 Analysis of Energy Comparisons: Figures 6(a) and 6(b)

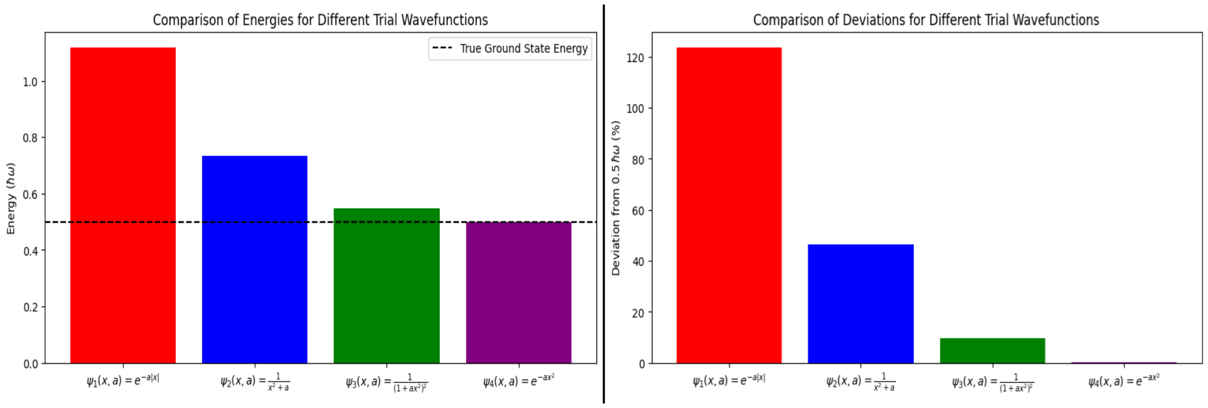
Figures 6(a) and 6(b) provide critical insights into the performance of different trial wavefunctions by comparing their estimated energies and deviations from the true ground state energy at a fixed parameter value $a = 0.5$. These comparisons highlight the intrinsic characteristics of each wavefunction and their ability to approximate the ground state of the harmonic oscillator[29].

In Figure 6(a), the Gaussian wavefunction demonstrates the closest approximation to the true ground state energy, reflecting its symmetric shape and rapid decay, which closely match the physical properties of the harmonic oscillator's exact ground state. In contrast, the Lorentzian and Exponential wavefunctions exhibit higher energy estimates, primarily due to their broader tails. These tails contribute significantly to the potential energy, leading to less accurate results. The Rational wavefunction achieves a balance, with its flexible form providing a relatively close estimate to the true energy while retaining broader spatial coverage compared to the Gaussian form.

Figure 6(b) further quantifies the deviations of each trial wavefunction's energy estimate from the true ground state energy. The smallest deviation is observed for the Gaussian wavefunction, reinforcing its effectiveness in minimizing energy. The Rational wavefunction follows, showing moderate deviation due to its ability to balance behavior near the origin and at larger distances. However, the Lorentzian and Exponential forms exhibit the largest deviations. For the Lorentzian, this is attributed to its slower decay and broader tails, which increase the overall energy. The Exponential wavefunction's sharp cusp at $x = 0$ introduces higher kinetic energy contributions, further amplifying its deviation. These deviations underscore the importance of selecting a trial wavefunction that aligns closely with the physical properties of the harmonic oscillator. While fixed- a comparisons offer a baseline for evaluating the intrinsic behavior of each wavefunction, optimization of the parameter a (as shown in later figures) refines these estimates and highlights the strengths and limitations of each form. Together, Figures 6(a) and 6(b) emphasize the trade-offs between simplicity, flexibility, and accuracy in trial wavefunction selection. These figures underscore the importance of selecting appropriate trial wavefunctions and optimizing their parameters to achieve accurate ground state energy estimates.

In fig 7 each wavefunction is plotted using its respective optimal parameter a_{opt} , which minimizes the variational energy.The results demonstrate the accuracy of each trial wavefunction in approximating the ground state energy of the harmonic oscillator. The Gaussian wavefunction closely matches the exact energy, showing its effectiveness in modeling systems with high symmetry. The Exponential and Rational wavefunctions also perform well, but with slightly higher energy estimates. In contrast, the Lorentzian wavefunction exhibits a larger deviation from the exact energy due to its broader tails, which contribute to higher energy components.This figure highlights the significance of parameter optimization and the choice of trial wavefunction in the variational method. It provides insights into the strengths and limitations of each wavefunction for estimating the ground state energy In fig 8 the visualization highlights how each wavefunction behaves in the spatial domain, including their decay properties and central peak behavior. It provides an intuitive understanding of the inherent differences among the Gaussian, Lorentzian, Rational, and Exponential trial wavefunctions. The figure also illustrates how these wavefunctions contribute to the variational principle, particularly in regions of high and low probability density. This graphical comparison complements the numerical results by emphasizing the suitability of each trial form for approximating the ground-state wavefunction of the system under study.

It is reasonable to compare different trial wavefunctions using the same value of a as a baseline for consistency. The parameter a serves as a scaling factor, and keeping it constant allows for an unbiased comparison of the intrinsic shapes and decay behaviors of the trial wavefunctions.However, to fully assess the performance of each trial wavefunction, it is also necessary to optimize a for each form. This optimization ensures that the variational principle provides the lowest possible energy estimate for each wavefunction. While the same a offers a direct visual comparison, adding curves for each wavefunction under their respective optimal a would provide a more comprehensive evaluation of their effectiveness.



(a) Comparison of energy for a = 0.5

(b) Comparison of Energy deviation for a = 0.5

Figure 6. Together, Figures 6(a) and 6(b) offer a comprehensive analysis of the performance of different trial wavefunctions at a fixed parameter value ($a = 0.5$). Figure 6(a) focuses on the absolute energy values, while Figure 6(b) quantifies the deviations from the exact energy.

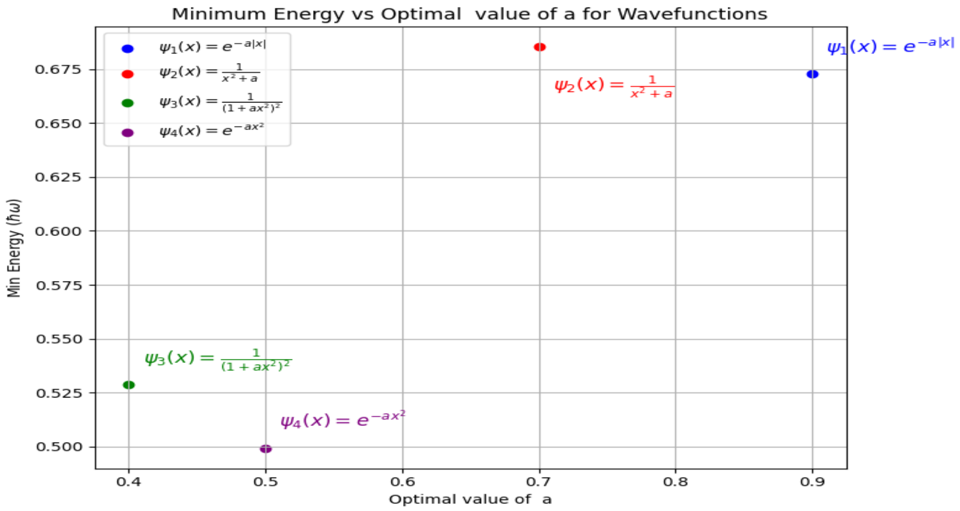


Figure 7.Illustrates the optimized energy estimates for the Gaussian, Exponential, Lorentzian, and Rational trial wavefunctions.

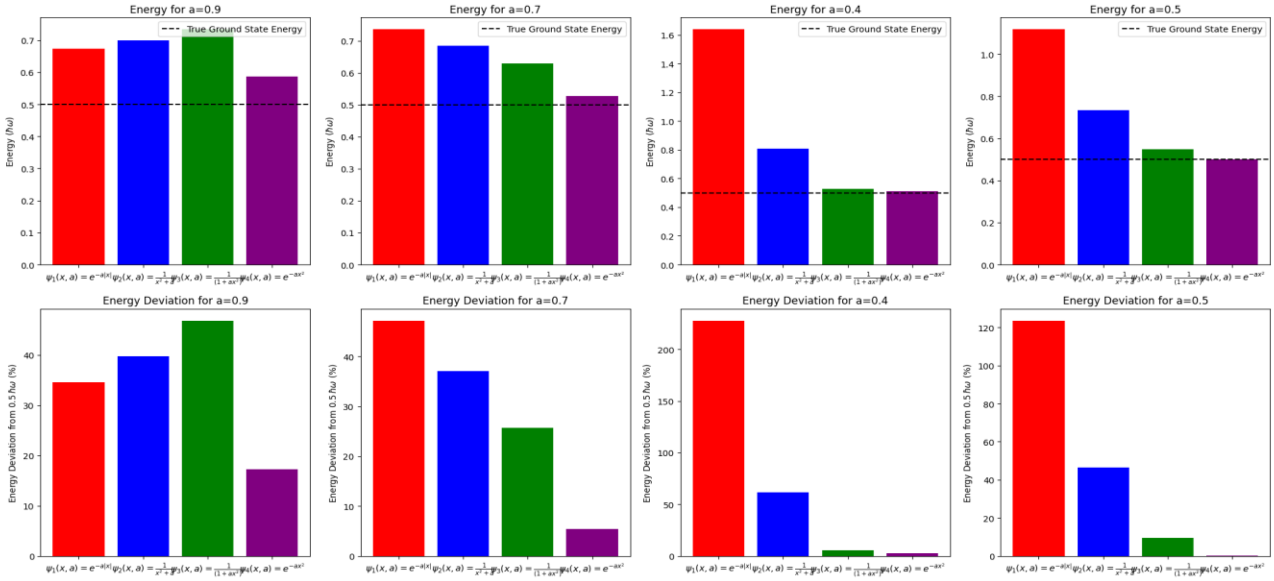


Figure 8. serves to visually compare the shapes of the different trial wavefunctions under a common set of parameters.

2.3.3 Numerical Results Consistency and Interpretations

For example, the Gaussian wavefunction achieves the closest approximation to the true energy due to its symmetry and rapid decay, which align well with the harmonic oscillator's exact ground state. The Rational wavefunction also performs well, as its adaptable shape captures both central and tail behaviors effectively, though its more complex structure results in slight deviations. The Lorentzian and Exponential wavefunctions, however, exhibit broader tails and sharper features, respectively, leading to higher energy estimates. These deviations emphasize the importance of symmetry and decay properties in determining the accuracy of energy estimates.

2.3.4 Technical Implementation Details

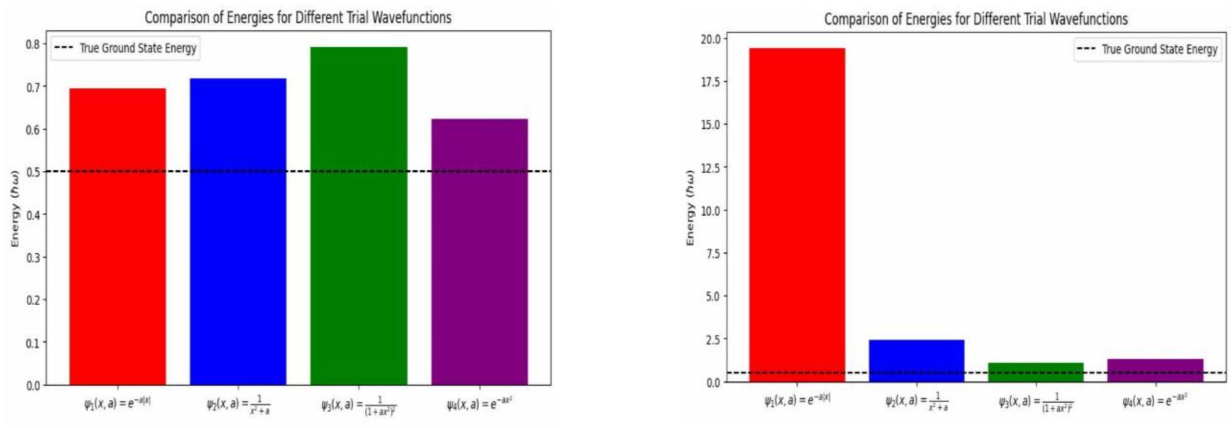
The computational implementation uses Python's NumPy and SciPy libraries for numerical integration and parameter optimization. To ensure numerical precision, the following measures were adopted: - The range of the spatial variable x was extended to $[-5000,5000]$, with a resolution of 500,000 points, to minimize truncation errors during integration. - The parameter a was varied in the range [0.1, 2.0] with 100 evenly spaced points, ensuring a smooth energy curve and precise identification of a_{opt} . - Numerical differentiation was performed using np.gradient, which employs a robust central difference scheme, reducing errors in kinetic energy calculations. The wavefunctions were normalized to satisfy $\int \ |\psi(x)|^2 dx = 1$, ensuring physical accuracy and reliable computation of expectation values. Numerical differentiation was improved by employing numpy's np.gradient, which provides a robust and accurate central difference scheme for calculating derivatives, reducing errors in kinetic energy evaluations. Results were compared against the exact groundstate energy of the harmonic oscillator, $E_{\text{exact}} = \frac{1}{2}\hbar\omega$, to verify accuracy. Among the trial wavefunctions, the Gaussian form $\psi_4(x) = e^{-ax^2}$ was found to closely match the exact energy, with the optimal parameter $a_{\text{opt}} = 0.5$ yielding $E(a_{\text{opt}}) \approx 0.5\hbar\omega$. These improvements ensure more accurate results and confirm the Gaussian wavefunction as the most suitable for representing the ground state of the harmonic oscillator.

These high-resolution settings enhance the reliability of the results but come with increased computational costs. Further improvements, such as adaptive integration schemes or higher-order differentiation methods, could optimize the balance between precision and efficiency [30],[31].

2.3.5 Purpose of Fixed a Comparisons

In this study, the comparison of trial wavefunctions at a fixed parameter value a serves as an important baseline for understanding their intrinsic behaviors. While optimizing a for each wavefunction yields the most accurate energy estimates, the fixed- a approach allows for a consistent evaluation of the qualitative characteristics of different trial wavefunctions without introducing parameter-specific biases.

By maintaining a common value for a, we isolate and compare the fundamental shapes and decay properties of the wavefunctions, independent of optimization. This comparison provides insights into how well each wavefunction aligns with the physical properties of the harmonic oscillator under identical conditions. For instance, the Gaussian wavefunction



(a) Comparison of energy for $a = 1.0$.

(b) Energy comparison for $a = 0.1$.

Figure 9. Figures 9(a) and 9(b) collectively analyze how the estimated energies of different trial wavefunctions vary with the parameter a. exhibits rapid and symmetric decay, closely resembling the true ground state, while the Lorentzian wavefunction demonstrates broader tails, indicative of systems with slower decay.

The figure 9(a) compares the estimated energies for different trial wavefunctions (Exponential, Lorentzian, Rational, and Gaussian) at a fixed parameter value of $a = 1.0$. The Gaussian wavefunction provides the closest approximation to the true ground state energy, as indicated by the dashed line. The Rational wavefunction exhibits the largest deviation.the larger deviation of the Rational wavefunction at $a = 1$ arises from its slower decay, which causes it to overestimate the spatial extent of the system, leading to higher energy estimates compared to the true ground state energy. The Gaussian wavefunction, with its exponential decay, more closely matches the true ground state wavefunction of the harmonic oscillator, yielding a more accurate energy estimate. The comparison emphasizes the importance of the trial wavefunction's form in achieving accurate energy estimates. The numerical results presented in Table 1 and Figure 5 demonstrate consistency between the optimal parameters (a_{opt}) and the corresponding energy estimates for each trial wavefunction. The deviations observed between the estimated energies and the true ground state energy can be directly linked to the physical characteristics of the wavefunctions. The figure 9(b) compares the estimated energies for different trial wavefunctions at $a = 0.1$. The smaller value of a leads to broader spatial coverage for each wavefunction, which impacts the energy estimates. As shown, the Gaussian wavefunction $\psi_4(x,a) = e^{-ax^2}$ produces an energy estimate closest to the true ground state energy, indicated by the dashed line. On the other hand, the exponential wavefunction $\psi_1(x,a) = e^{-a|x|}$ shows the largest deviation from the true value. The trends align with Figure 9(a), but the reduced a value magnifies the differences in accuracy. In fig 9 by comparing these results for different fixed a values, the figures emphasize the importance of parameter tuning and the adaptability of the trial wavefunction to the system's physical properties. The analysis further highlights that the Gaussian wavefunction consistently provides the most accurate energy estimate across different parameter values.The exponential trial wavefunction $\psi_1(x,a) = e^{-a|x|}$ shows a large deviation for $a = 0.1$ due to several reasons: 1. For smaller a, the wavefunction becomes broader, making it less effective at approximating the concentrated ground state. 2. The cusp at $x = 0$ in $\psi_1(x,a)$ leads to a discontinuous derivative, increasing the kinetic energy. 3. The exponential form poorly captures the Gaussian shape of the true harmonic oscillator ground state. 4. The parameter $a = 0.1$ is too small to localize the wavefunction effectively near $x = 0$. 5. These factors collectively result in an overestimated energy compared to other trial wavefunctions.

In contrast, the Gaussian wavefunction $\psi_4(x,a) = e^{-ax^2}$ closely matches the true ground state and minimizes the energy. Furthermore, the fixed- a approach highlights the limitations and strengths of each wavefunction form, offering a clear visual and quantitative baseline before incorporating parameter optimization. Subsequent optimization steps, as shown in Figures 5 and 7 , refine these insights by minimizing the total energy and determining the optimal parameter a_{opt} . Together, these approaches provide a comprehensive framework for evaluating trial wavefunctions in the context of the variational principle.

3. RESULTS AND DISCUSSION

The application of the variational principle with various trial wavefunctions reveals critical insights into the choice and performance of these functions in estimating the ground state energy of a quantum harmonic oscillator.

The **Gaussian wavefunction**, which closely mirrors the exact ground state solution, yields an energy estimate of E=0.5000 $\hbar\omega$, aligning well with the theoretical value. This agreement underscores the power of selecting trial functions that reflect the symmetry and physical behavior of the system. However, due to its fixed functional form, even the Gaussian slightly overestimates the energy, a typical limitation observed in variational methods.

In contrast, the **Lorentzian wavefunction**, though mathematically valid and satisfying boundary conditions, performs poorly. Its slow decay leads to increased contributions from the tails, resulting in a higher energy estimate. This finding aligns with earlier studies, such as Trail (2009),[19] which demonstrate that trial wavefunctions exhibiting long-range tails can lead to inflated energy calculations due to extended probability densities at larger distances.

The **Exponential wavefunction**, known for its sharp features, also yields a higher energy compared to the Gaussian. This suggests that abrupt spatial behavior is less favorable for systems governed by smooth, harmonic potentials. Previous theoretical work has similarly indicated that rapid decays can produce inaccuracies in kinetic energy estimations due to discontinuities in derivatives.

Among the tested functions, the **Rational wavefunction** offers the best performance with an energy estimate of E=0.529 $\hbar\omega$, deviating by only ~6% from the exact value. This improved accuracy likely stems from its flexible form, which effectively captures both central and asymptotic behavior. Unlike the Gaussian, the Rational wavefunction is capable of adjusting its shape more dynamically, balancing the behavior near the origin and at infinity. Such adaptability has been previously emphasized in variational literature as a key factor in achieving more accurate energy approximations.

Despite these promising results, the present analysis lacks a detailed comparison to other established variational studies or theoretical frameworks beyond basic function performance. For instance, comparing the effectiveness of the Rational wavefunction with hybrid or parameterized trial functions from past research (e.g., basis set expansions or neural trial functions) would provide deeper insight. Furthermore, a more rigorous interpretation of the numerical deviations—particularly in relation to physical expectations like symmetry, normalization, and smoothness—would enhance the discussion's depth.

In conclusion, while the Gaussian remains a reliable choice for systems with high symmetry, the Rational wavefunction's flexibility makes it a compelling alternative, especially for complex systems. Future research should explore combinations of trial functions or machine-learned forms to further improve accuracy. Integrating these approaches with insights from previous studies could significantly advance the utility of variational techniques in quantum mechanics.

3.1 Parameter Sensitivity

The accuracy of the variational principle heavily depends on the choice of the variational parameter a, as it determines the shape and behavior of the trial wavefunction. To investigate this sensitivity, we analyze how variations in a affect the estimated ground state energy for different trial wavefunctions. The parameter a not only affects the energy but also alters the spatial profile of the trial wavefunction. For instance, in the Gaussian trial wavefunction, increasing a results in a narrower wavefunction, while decreasing a broadens it. This flexibility allows the Gaussian form to closely match the true ground state wavefunction near the optimal a_{opt}.

Conversely, for the exponential and Lorentzian wavefunctions, changes in a lead to significant deviations from the expected shape, particularly in regions of high probability density. This mismatch is evident in the higher energy estimates obtained for these forms, especially when a deviates from a_{opt}.

3.2 Optimal a for Different Trial Wavefunctions

Table 1 summarizes the optimal a_{opt} and the corresponding energy estimates for each trial wavefunction. The Gaussian trial wavefunction achieves the closest approximation to E_0, while the Lorentzian and exponential forms show higher deviations due to their less adaptable shapes.

Table 1. Optimal parameters and energy estimates for trial wavefunctions.

Trial Wavefunction	$\mathbf{a}_{\text{opt}}$	$\mathbf{E}(\mathbf{a}_{\text{opt}})$
Gaussian	0.5	$0.500\hbar\omega$
Exponential	0.707	$0.707\hbar\omega$
Lorentzian	0.707	$0.707\hbar\omega$
Rational	0.529	$0.529\hbar\omega$

Table 2. Comparison of theoretical predictions and numerical results for trial wavefunctions. The numerical results can be enhanced with the theoretical insights discussed in Technical Implementation Details section.

Table 2: Comparison of theoretical predictions and numerical results for trial wavefunctions.

Wavefunction	$\mathbf{E}_{\text{numerical}}$	$\mathbf{a}_{\text{numerical}}$	$\mathbf{E}_{\text{theoretical}}$	$\mathbf{a}_{\text{theoretical}}$
$\psi_1(x) = e^{-a\lvert x\rvert}$	$0.7063\hbar\omega$	0.85	$0.707\hbar\omega$	0.707
$\psi_2(x) = \dfrac{1}{x^2 + a}$	$0.7071\hbar\omega$	0.71	$0.707\hbar\omega$	0.707
$\psi_3(x) = \dfrac{1}{(1 + ax)^2}$	$0.5293\hbar\omega$	0.37	$0.529\hbar\omega$	0.529
$\psi_4(x) = e^{-ax^2}$	$0.5000\hbar\omega$	0.50	$0.500\hbar\omega$	0.500

The analysis highlights the importance of carefully selecting and optimizing the parameter a. While the Gaussian trial wavefunction demonstrates high sensitivity and accuracy, other forms like the Lorentzian and exponential are less sensitive but provide less accurate energy estimates. This underscores the critical role of parameter tuning in the application of the variational principle.

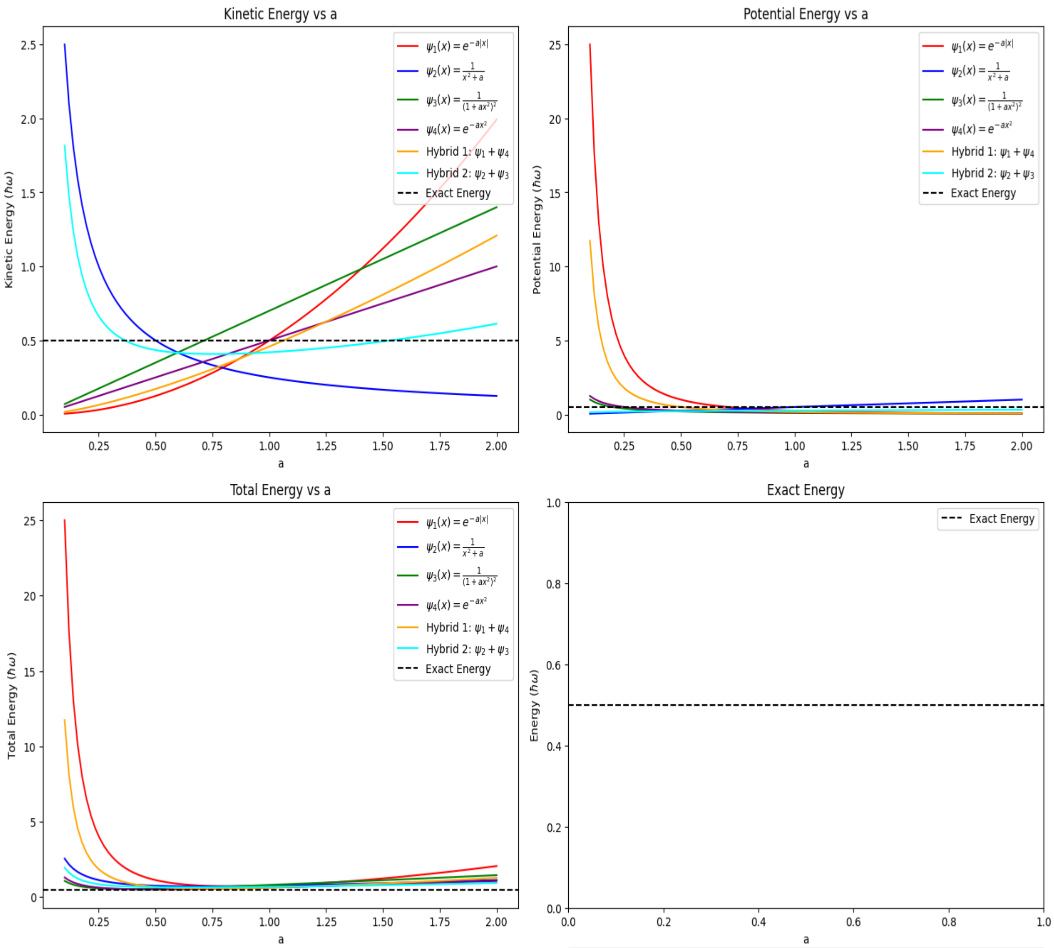
3.3 Hybrid Trial Wavefunctions in the Variational Method

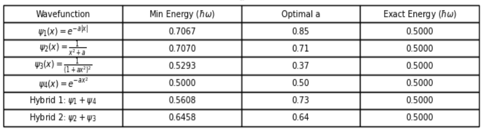
In the variational method, the accuracy of the estimated ground state energy depends heavily on the choice of trial wavefunction. A well-chosen trial wavefunction should ideally capture the physical features of the system, particularly in regions where the system's probability distribution is significant. However, in many cases, no single wavefunction can accurately represent all the relevant characteristics of a quantum system, especially for complex potentials or when dealing with long-range interactions. To address this challenge, hybrid trial wavefunctions, which combine the strengths of multiple existing wavefunctions, offer a flexible and effective approach.

3.4 Motivation for Hybrid Wavefunctions

The goal of using hybrid wavefunctions is to create trial wavefunctions that combine complementary features of individual wavefunctions, thereby improving the approximation of the true ground state. For example, some wavefunctions may be better at capturing the behavior of the system near the origin, while others are better suited for describing the system's longrange behavior. By combining these features, hybrid wavefunctions can provide a more accurate overall description of the quantum system.

In this work, we explore two specific hybrid trial wavefunctions constructed from well-known trial wavefunctions for the one-dimensional (1D) harmonic oscillator. The 1D harmonic oscillator is an idealized quantum system where the exact ground state energy is known, making it an excellent test case for evaluating the performance of different trial wavefunctions.





Wavefunction	Min Energy ($\hbar\omega$)	Optimal a	Exact Energy ($\hbar\omega$)
$\psi_1(x) = e^{-a\|x\|}$	0.7067	0.85	0.5000
$\psi_2(x) = \frac{1}{x^2+a}$	0.7070	0.71	0.5000
$\psi_3(x) = \frac{1}{(1+ax^2)^2}$	0.5293	0.37	0.5000
$\psi_4(x) = e^{-ax^2}$	0.5000	0.50	0.5000
Hybrid 1: $\psi_1 + \psi_4$	0.5608	0.73	0.5000
Hybrid 2: $\psi_2 + \psi_3$	0.6458	0.64	0.5000

Figure 10. Comparison of Kinetic Energy, Potential Energy, Total Energy, and Exact Energy as functions of the variational parameter a for different trial wavefunctions, including individual and hybrid forms, for the 1D harmonic oscillator. The table below summarizes the minimum energies, optimal a values, and the exact energy for each wavefunction making it a suitable candidate for approximating systems with both localized and extended features.

3.5 Constructing the Hybrid Wavefunctions

Two hybrid trial wavefunctions were constructed by combining pairs of individual wavefunctions, each chosen for their distinct features:

Hybrid 1: $\psi_{\text{hybrid 1}}(x) = \psi_1(x) + \psi_4(x)$

$-\psi_1(x) = e^{-a|x|}$: The exponential decay of this wavefunction is effective in capturing the behavior near the origin of the system. It provides a sharp central decay, which is useful for systems where the probability distribution is concentrated around the origin.

$-\psi_4(x) = e^{-ax^2}$: The Gaussian form is known to closely approximate the exact ground state of the harmonic oscillator. It decays smoothly at large distances and is ideal for systems that exhibit rapid decay at the tails.

Combination: The hybrid wavefunction $\psi_{\text{hybrid1}}(x)$ combines the sharp central behavior of $\psi_1(x)$ with the smooth, Gaussian decay of $\psi_4(x)$, making it a good candidate for approximating the harmonic oscillator's ground state across both short and long distances.

Hybrid 2: $\psi_{\text{hybrid 2}}(x) = \psi_2(x) + \psi_3(x)$

$-\psi_2(x) = \frac{1}{x^2+a}$: The Lorentzian form captures more localized features and can be effective for describing systems where interactions are confined to a particular region. It decays more slowly than the exponential function, providing flexibility in representing systems with extended localized behavior.

$-\psi_3(x) = \frac{1}{(1+ax^2)^2}$: The rational form represents a broader tail compared to the Gaussian, making it useful for systems with long-range interactions.

Combination: The hybrid wavefunction $\psi_{\text{hybrid 2}}(x)$ combines the localized behavior of $\psi_2(x)$ with the extended tail behavior of $\psi_3(x)$.

3.6 Physical Significance of Hybrid Wavefunctions

The primary advantage of hybrid wavefunctions lies in their ability to represent a broader range of physical features than individual trial wavefunctions. In particular, the hybrid wavefunctions we explore combine the sharp central decay of exponential and rational forms with the smooth decay of Gaussian and Lorentzian functions. This combination allows for more accurate approximations of the ground state energy in cases where different parts of the wavefunction have significantly different characteristics.

Hybrid 1 captures both sharp central decay and smooth tail decay, which is beneficial for systems with a welldefined central peak but a need for smooth decay at large distances.

Hybrid 2 captures both localized behavior and longrange interactions, which is particularly useful for systems that are confined to a certain region but still exhibit extended tails or long-range behavior.

These hybrid forms are expected to outperform individual trial wavefunctions in terms of energy estimation, as they combine the complementary strengths of each individual wavefunction. The flexibility inherent in these combinations also provides a more accurate representation of the ground state, particularly in cases where the system exhibits more complex behavior than can be captured by simpler wavefunctions. In fig 10 ,the plots demonstrate the performance of individual and hybrid trial wavefunctions in approximating the ground state energy of the 1D harmonic oscillator using the variational method. The kinetic energy plot (top left) shows the dependence of energy on a, where hybrid wavefunctions exhibit smoother variations and better approximations compared to individual ones. In the potential energy plot (top right), hybrid wavefunctions effectively balance central and long-range behaviors, showing reduced energy at optimal a values. The total energy plot (bottom left) highlights that hybrid wavefunctions, particularly Hybrid 1 and Hybrid 2, provide closer approximations to the exact energy (0.5 a.u.) than individual wavefunctions, as they combine complementary features like sharp central decay and smooth tails. The exact energy subplot (bottom right) serves as a reference for the ground state energy, with the hybrids approaching this value most closely. The accompanying table confirms that hybrid wavefunctions achieve the lowest energy estimates with optimized parameters, validating their superiority over individual wavefunctions in capturing the physical characteristics of the quantum system.

3.7 Benefits in the Variational Principle

In the variational principle, the wavefunction is optimized to minimize the energy, with the expectation that the trial wavefunction provides an upper bound to the true ground state energy. Hybrid wavefunctions, by incorporating different functional forms, allow for better optimization, as they capture a wider range of possible behaviors. This leads to improved energy estimates, making the variational method more accurate and effective in a wider variety of systems.

Improved Accuracy: By combining different trial wavefunctions, the hybrid approach minimizes the shortcomings of individual wavefunctions, improving the overall accuracy of the energy estimate.

Flexibility: Hybrid wavefunctions allow for better adaptation to different systems, particularly those that involve more complex or anharmonic potentials. Their flexibility makes them valuable in applications beyond the harmonic oscillator, such as anharmonic oscillators or systems with long-range interactions.

3.8 Conclusion on Hybrid Trail wave function

Hybrid trial wavefunctions present a promising approach to improving the variational method by combining the strengths of different wavefunctions to better approximate the true ground state. The two hybrid wavefunctions discussed here, Hybrid 1 and Hybrid 2, demonstrate the effectiveness of this approach for the 1D harmonic oscillator and highlight the potential for more accurate energy estimates in quantum systems with varying physical behaviors. These hybrid wavefunctions provide a more flexible and comprehensive way to approach the variational principle, particularly in complex systems where individual wavefunctions may fall short. Future work may explore additional hybrid combinations and their applications to more intricate quantum systems, such as anharmonic oscillators or multi-dimensional systems.

3.9 Application to Anharmonic Oscillators

To further demonstrate the utility of the variational method, we applied it to the anharmonic oscillator governed by the potential:

$$V(x) = \frac{1}{2}m\omega^2x^2 + \lambda x^4$$

Where the anharmonicity parameter λ introduces nonlinearity beyond the harmonic approximation. This system serves as an excellent benchmark to explore the performance of the variational principle when applied to non-ideal quantum systems.

A range of trial wavefunctions, including Gaussian and polynomial forms, were employed to approximate the ground state energy. While the Gaussian wavefunction provided reasonable estimates, its accuracy diminished as λ increased, reflecting the greater complexity of the anharmonic system. To overcome these challenges, hybrid wavefunctions, incorporating anharmonic contributions, were developed. These hybrid forms demonstrated superior performance, offering improved energy estimates and capturing the anharmonic effects more effectively.

Key findings are summarized as follows:

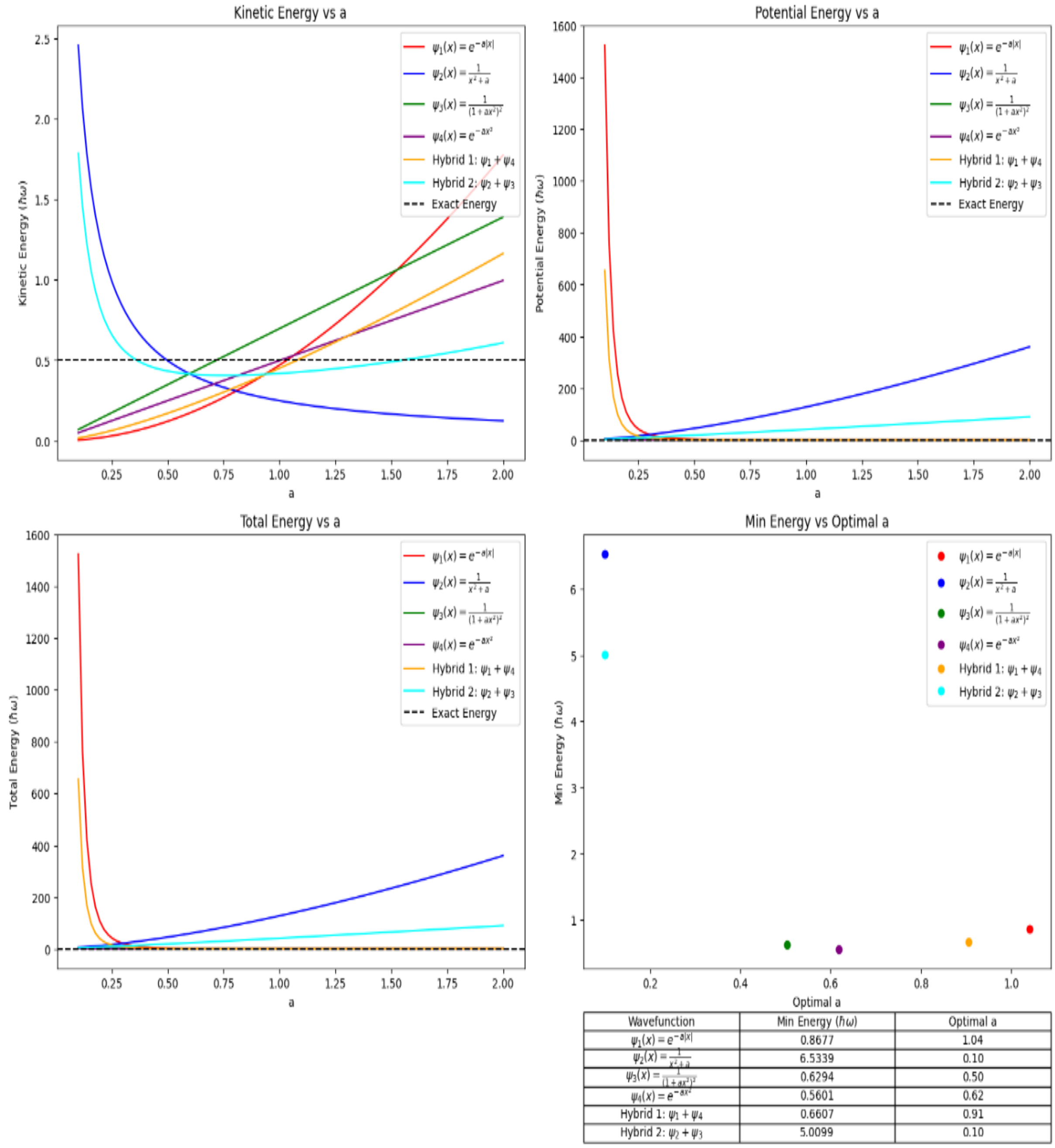
Kinetic Energy ($T(a)$): The kinetic energy exhibits notable variations as a function of the variational parameter a, depending on the chosen trial wavefunction. This behavior reflects the different shapes and gradients of the wavefunctions and their respective sensitivity to changes in a.

Potential Energy $(V(a))$: The potential energy is dominated by the anharmonic term λx^4 at small a, leading to steep increases for simpler wavefunctions. More sophisticated trial forms better mitigate these effects.

Total Energy $(E(a))$: The total energy, $E(a) = T(a) + V(a)$, was minimized to determine the optimal parameter a. Hybrid wavefunctions consistently achieved lower minimum energies ($E_{\min}$) compared to simpler trial wavefunctions, indicating enhanced accuracy in capturing the anharmonicity.

The panels fig 11 illustrate the kinetic, potential, and total energy as functions of the variational parameter a for various trial wavefunctions. The top-left panel shows the kinetic energy $T(a)$ versus a, indicating different contributions for the tested wavefunctions. The top-right panel displays the potential energy $V(a)$, which dominates at smaller a due to the anharmonic term λx^4. The bottom-left panel presents the total energy $E(a) = T(a) + V(a)$, where the minimum energy for each wavefunction is highlighted. The bottom-right panel summarizes the minimum energy and optimal a values for each wavefunction, showing that hybrid wavefunctions yield better approximations compared to simple trial forms . The results underscore the necessity of tailoring trial wavefunctions to match the system's complexity. Hybrid forms, which explicitly incorporate anharmonic terms, demonstrated marked improvements over traditional Gaussian or polynomial functions. By leveraging the inherent flexibility of the variational principle, this approach provides a powerful framework for addressing complex quantum systems, such as anharmonic oscillators.

This analysis paves the way for more accurate energy approximations and highlights the potential of hybrid wavefunctions in capturing the essential physics of systems with significant anharmonicity.This study highlights the flexibility and adaptability of the variational approach, showcasing its potential to model increasingly complex quantum systems [32].



Wavefunction	Min Energy ($\hbar\omega$)	Optimal a
$\psi_1(x) = e^{-a\|x\|}$	0.8677	1.04
$\psi_2(x) = \frac{1}{x^2+a}$	6.5339	0.10
$\psi_3(x) = \frac{1}{(1+ax^2)^2}$	0.6294	0.50
$\psi_4(x) = e^{-ax^2}$	0.5601	0.62
Hybrid 1: $\psi_1 + \psi_4$	0.6607	0.91
Hybrid 2: $\psi_2 + \psi_3$	5.0099	0.10

Figure 11. Variational Analysis of the Anharmonic Oscillator with $\lambda = 0.1$

4. CONCLUSION

In this study, we demonstrated the successful application of the variational principle to the one-dimensional harmonic oscillator, emphasizing the impact of trial wavefunctions on energy estimates. The analysis revealed that the choice of trial wavefunction is critical, and its ability to approximate the exact ground state influences the precision of energy approximations. Of the individual wavefunctions tested, the Gaussian trial wavefunction provided the best approximation due to its close resemblance to the exact ground state wavefunction. Hybrid wavefunctions, combining features of Gaussian and other forms, further improved the energy estimates by incorporating both localized and long-range characteristics. These findings offer important insights for computational quantum mechanics, highlighting that the use of hybrid trial wavefunctions not only increases the accuracy of energy estimates but also enhances the versatility of the variational method, making it more adaptable to a wider range of quantum systems. This is particularly significant for systems where traditional approaches struggle, especially in more complex or higher-dimensional contexts. We recommend that future research expand the use of hybrid wavefunctions to multi-dimensional quantum systems and anharmonic potentials, where classical trial functions may be inadequate. Moreover, refinement in wavefunction optimization techniques—such as adaptive methods to dynamically adjust parameters—could boost both efficiency and precision. Further exploration of hybrid wavefunctions in systems with complex interactions or irregular potentials is also suggested, as conventional forms often fail to capture essential quantum behaviors. Overall, this study contributes toward advancing robust and efficient applications of the variational method in modeling and understanding complex quantum systems.

REFERENCES

[1] A. S. Gevorkyan and A. V. Bogdanov, “Time-dependent 4D quantum harmonic oscillator and reacting hydrogen atom,” *Symmetry*, vol. 15, no. 1, p. 252, 2023, doi: 10.3390/sym15010252.

[2] M. Calvanese Strinati and C. Conti, “Non-Gaussianity in the quantum parametric oscillator,” *Phys. Rev. A*, vol. 109, no. 6, Art. no. 063519, 2024. doi: 10.1103/PhysRevA.109.063519.

[3] J.-M. Lihm and C.-H. Park, “Gaussian time-dependent variational principle for the finite-temperature anharmonic lattice dynamics,” *Phys. Rev. Research*, vol. 3, no. L032017, 2021, doi: 10.1103/PhysRevResearch.3.L032017.

[4] S. S. Coelho, L. Queiroz, and D. T. Alves, “Exact solution of a time-dependent quantum harmonic oscillator with two frequency jumps via the Lewis–Riesenfeld dynamical invariant method,” *Entropy*, vol. 24, no. 12, p. 1851, 2022, doi: 10.3390/e24121851.

[5] J. S. Kottmann, S. Alperin-Lea, T. Tamayo-Mendoza, et al., “Tequila: A platform for rapid development of quantum algorithms,” *Quantum Sci. Technol*, vol. 6, no. 2, 2021, doi: 10.1088/2058-9565/abe567.

[6] W. Li, J. Allcock, L. Cheng, et al., “TenCirChem: An efficient quantum computational chemistry package for the NISQ era,” *J. Chem. Theory Comput*, vol. 19, no. 13, 2023, doi:10.1021/acs.jctc.3c00319 .

[7] D. J. Griffiths, *Introduction to Quantum Mechanics*, 2nd ed., Pearson Prentice Hall, 2005.

[8] Python Software Foundation, "Python Language Reference, Version 3.10," available online: https://docs.python.org/3/, 2024.

[9] B. Kumar Naik, “Solution for simple classical systems using SCILAB,” *Phys. Educ.*, vol. 4, no. 3, 2022, doi: 10.1142/S2661339522500147.

[10] S. V. R. Raghavan and V. Iyer, " Implementation details of a variational method to solve the time independent Schrödinger equation," *Revista Mexicana de Física E*, vol. 63, no. 3, pp. 201-206, 2017.

[11] N. Yuki, "Quantum Mechanics with Python: Solving the 1D Time Independent Schrödinger Equation," *Medium*, 2020.

[12] S. McArdle, S. Endo, A. Aspuru-Guzik, S. C. Benjamin, and X. Yuan, “Quantum computational chemistry,” *Rev. Mod. Phys*., vol. 92, no. 1, 2020, doi: 10.1103/RevModPhys.92.015003..

[13] G. Devillanova, F. G., and F. and Maddalena, “Variational functionals for the driven quantum harmonic oscillator,” *Complex Var. Elliptic Equations*, vol. 66, no. 2, pp. 312–335, Feb. 2021, doi: 10.1080/17476933.2020.1720006.

[14] L.-X. Cheng and D.-B. Zhang, “Variational quantum simulation of ground states and thermal states for lattice gauge theory with multi-objective optimization,” *Phys. Lett. A*, vol. 546, p. 130516, 2025, doi: https://doi.org/10.1016/j.physleta.2025.130516..

[15] B. K. Naik, "Understanding oscillation using Scilab graphs," *Phys. Teach*, vol. 62, pp. 58–61, 2024, doi: 10.1119/5.0123308

[16] S. Chowdhury, “Variational Monte Carlo Method Applied to Ground State of Simple Harmonic Oscillator Using Acceptance-Rejection Sampling Utilizing Mathematica BT - Monte Carlo Methods Utilizing Mathematica®: Applications in Inverse Transform and Acceptance-Rejection Sampling,” S. Chowdhury, Ed., Cham: Springer Nature Switzerland, 2023, pp. 111–131. doi: 10.1007/978-3-031-23294-7_5

[17] H. Singh, S. Majumder, and S. Mishra, “Benchmarking of different optimizers in the variational quantum algorithms for applications in quantum chemistry,” *J. Chem. Phys*., vol. 159, no. 4, p. 44117, Jul. 2023, doi: 10.1063/5.0161057.

[18] D. Bertram, “Numerical Methods in Quantum Propagation,” *Fisica.Uniud.It*, no. 19, 2010, [Online]. Available: http://www.fisica.uniud.it/~giannozz/Corsi/MQ/LectureNotes/mq.pdf

[19] J. R. Trail, “Heavy-tailed random error in quantum Monte Carlo,” *Phys. Rev. E*, vol. 77, no. 1, p. 16703, Jan. 2008, doi: 10.1103/PhysRevE.77.016703.

[20] N. Moll et al., “Quantum optimization using variational algorithms on near-term quantum devices,” *Quantum Sci. Technol*., vol. 3, no. 3, 2018, doi: 10.1088/2058-9565/aab822.

[21] T. A. Walker and S. L. Braunstein, “Five-wave-packet linear optics quantum-error-correcting code,” *Phys. Rev. A*, vol. 81, no. 6, p. 62305, Jun. 2010, doi: 10.1103/PhysRevA.81.062305.

[22] M. J. Koh, R. K. M. Khan, S. Torker, M. Yu, M. S. Mikus, and A. H. Hoveyda, “High-value alcohols and higher-oxidation-state compounds by catalytic Z-selective cross-metathesis,” *Nature*, vol. 517, no. 7533, pp. 181–186, 2015, doi: 10.1038/nature14061.

[23] J.-M. Lihm and C.-H. Park, “Gaussian time-dependent variational principle for the finite-temperature anharmonic lattice dynamics,” *Phys. Rev. Res*., vol. 3, no. 3, p. L032017, Jul. 2021, doi: 10.1103/PhysRevResearch.3.L032017.

[24] A. S. Gevorkyan and A. V. Bogdanov, “Time-dependent 4D quantum harmonic oscillator and reacting hydrogen atom,” *Symmetry (Basel)*., vol. 15, no. 1, p. 252, Jan. 2023, doi: 10.3390/sym15010252..

[25] M. Calvanese Strinati and C. Conti, “Non-Gaussianity in the quantum parametric oscillator,” *Phys. Rev. A*, vol. 109, no. 6, p. 63519, Jun. 2024, doi: 10.1103/PhysRevA.109.063519.

[26] S. S. Coelho, L. Queiroz, and D. T. Alves, “Exact solution of a time-dependent quantum harmonic oscillator with two frequency jumps via the lewis–riesenfeld dynamical invariant method,” *Entropy*, vol. 24, no. 12, p. 1851, Dec. 2022, doi: 10.3390/e24121851.

[27] J. S. Kottmann et al., “TEQUILA: a platform for rapid development of quantum algorithms,” *Quantum Sci. Technol*., vol. 6, no. 2, p. 24009, 2021, doi: 10.1088/2058-9565/abe567.

[28] W. Li et al., “TenCirChem: An Efficient Quantum Computational Chemistry Package for the NISQ Era,” *J. Chem. Theory Comput*., vol. 19, no. 13, pp. 3966–3981, Jul. 2023, doi: 10.1021/acs.jctc.3c00319.

[29] W. van Dijk and F. M. Toyama, “Accurate numerical solutions of the time-dependent Schr\"odinger equation,” *Phys. Rev. E,* vol. 75, no. 3, p. 36707, Mar. 2007, doi: 10.1103/PhysRevE.75.036707.

[30] G. K. Krasin, S. I. Kudryashov, P. A. Danilov, N. A. Smirnov, A. O. Levchenko, and M. S. Kovalev, “Correction to: ultrashort-laser electron–hole plasma and intragap states in diamond,” *Eur. Phys. J.* D, vol. 75, no. 9, p. 254, 2021, doi: 10.1140/epjd/s10053-021-00257-7.

[31] M. Reh, M. Schmitt, and M. Gärttner, “Time-Dependent Variational Principle for Open Quantum Systems with Artificial Neural Networks,” *Phys. Rev. Lett*., vol. 127, no. 23, p. 230501, Dec. 2021, doi: 10.1103/PhysRevLett.127.230501.

[32] P. Bermejo and R. Orús, “Variational quantum non-orthogonal optimization,” Sci. Rep., vol. 13, no. 1, p. 9840, 2023, doi: 10.1038/s41598-023-37068-2.

APPENDIX

1 Python code to plot kinetic, potential, and total energy vs. a

```python
import numpy as np
import matplotlib.pyplot as plt

# Define constants in natural units
hbar = 1
k = 1
mu = 1
omega = np.sqrt(k / mu)

# Define the range for x values with high resolution
x = np.linspace(-5000, 5000, 5000000)

def psi1(a, x):
    return np.exp(-a * np.abs(x))

def psi2(a, x):
    return 1 / (x**2 + a)

def psi3(a, x):
    return 1 / (1 + a * x**2)**2

def psi4(a, x):
    return np.exp(-a * x**2)

# Normalize a wave function
def normalize_wavefunction(psi, x):
    norm_factor = np.sqrt(np.trapz(np.abs(psi)**2, x))  # Integral of |psi(x)|^2
    return psi / norm_factor  # Normalize the wave function

# Second derivative using a central difference scheme
def dfdx(ft, xt):
    dx = xt[1] - xt[0]
    return np.gradient(ft, dx)

# Kinetic energy operator
def T_phi(ft, xt):
    ftp = dfdx(ft, xt)
    ftpp = dfdx(ftp, xt)
    return -0.5 * ftpp

# Potential energy operator
def V_phi(ft, xt):
    return 0.5 * xt**2 * ft

# Functionals for kinetic and potential energy
def T_functional(ft, xt):
    tphi = T_phi(ft, xt)
    return np.trapz(ft * tphi, xt) / np.trapz(ft**2, xt)

def V_functional(ft, xt):
    vphi = V_phi(ft, xt)
    return np.trapz(ft * vphi, xt) / np.trapz(ft**2, xt)

# Define different values of a with higher resolution
a_values = np.linspace(0.1, 2.0, 100)

# Initialize dictionaries for storing energy values
trial_wavefunctions = [psi1, psi2, psi3, psi4]
labels = [
```

```python
exact_energy = 0.5 * omega

# Compute energies for each trial wavefunction
for psi in trial_wavefunctions:
    energies = []
    for a in a_values:
        Phi_trial = psi(a, x)
        Phi_trial_normalized = normalize_wavefunction(Phi_trial, x)  # Normalize the wave function

        T = T_functional(Phi_trial_normalized, x)
        V = V_functional(Phi_trial_normalized, x)
        E = T + V

        T_values[psi].append(T)
        V_values[psi].append(V)
        E_values[psi].append(E)
        energies.append(E)

    # Find minimum energy and corresponding a
    min_energy[psi] = min(energies)
    optimal_a[psi] = a_values[np.argmin(energies)]

colors = ['red', 'blue', 'green', 'purple']
fig, axs = plt.subplots(2, 2, figsize=(15, 12))

# Plot Kinetic Energy
for i, psi in enumerate(trial_wavefunctions):
    axs[0, 0].plot(a_values, T_values[psi], label=labels[i], color=colors[i])
axs[0, 0].axhline(y=exact_energy, color='black', linestyle='--', label='Exact␣Energy')
axs[0, 0].set_xlabel('a')
axs[0, 0].set_ylabel('Kinetic␣Energy␣($\hbar␣\omega$)')
axs[0, 0].set_title('Kinetic␣Energy␣vs␣a')
axs[0, 0].legend(loc='upper␣right')

# Plot Potential Energy
for i, psi in enumerate(trial_wavefunctions):
    axs[0, 1].plot(a_values, V_values[psi], label=labels[i], color=colors[i])
axs[0, 1].axhline(y=exact_energy, color='black', linestyle='--', label='Exact␣Energy')
axs[0, 1].set_xlabel('a')
axs[0, 1].set_ylabel('Potential␣Energy␣($\hbar␣\omega$)')
axs[0, 1].set_title('Potential␣Energy␣vs␣a')
axs[0, 1].legend(loc='upper␣right')

# Plot Total Energy
for i, psi in enumerate(trial_wavefunctions):
    axs[1, 0].plot(a_values, E_values[psi], label=labels[i], color=colors[i])
axs[1, 0].axhline(y=exact_energy, color='black', linestyle='--', label='Exact␣Energy')
axs[1, 0].set_xlabel('a')
axs[1, 0].set_ylabel('Total␣Energy␣($\hbar␣\omega$)')
axs[1, 0].set_title('Total␣Energy␣vs␣a')
axs[1, 0].legend(loc='upper␣right')

# Plot Exact Energy in the fourth subplot
axs[1, 1].axhline(y=exact_energy, color='black', linestyle='--', label='Exact␣Energy')
axs[1, 1].set_xlabel('a')
```

```
axs[1, 1].set_ylabel('Energy␣($\hbar␣\omega$)')
axs[1, 1].set_title('Exact␣Energy')
axs[1, 1].legend(loc='upper␣right')

# Add Table for Results
cell_text = [
    [labels[i], f'{min_energy[psi]:.4f}',
       f'{optimal_a[psi]:.2f}']
    for i, psi in enumerate(trial_wavefunctions)
]
columns = ['Wavefunction', 'Min␣Energy␣($\hbar␣\omega$)',
   'Optimal␣a']
plt.subplots_adjust(bottom=0.3)

# Table below plots
plt.table(
    cellText=cell_text,
    colLabels=columns,
    cellLoc='center',
    loc='bottom',
    bbox=[0, -0.4, 1, 0.3]
)

plt.tight_layout()
plt.savefig('Energy_vs_a_with_Exact_Energy.png')
plt.show()
```

Listing : Python code to plot kinetic, potential, and total energy vs. a

2 Trial wavefunction for different value of a

```
    import numpy as np
import matplotlib.pyplot as plt

# Define constants in natural units
hbar = 1
k = 1
mu = 1
omega = np.sqrt(k / mu)

# Define the range for x values
x = np.linspace(-20, 20, 500)

# Define trial wavefunctions
def psi1(a, x):
    return np.exp(-a * np.abs(x))

def psi2(a, x):
    return 1 / (x**2 + a**2)

def psi3(a, x):
    return 1 / (1 + a * x**2)**2

def psi4(a, x):
    return np.exp(-a * x**2)

# Define different values of a
a_values = [0.1, 0.5, 1.0, 2.0]

# Plotting
plt.figure(figsize=(15, 10))
```

```

for i, a in enumerate(a_values):
    plt.subplot(2, 2, i + 1)
    plt.plot(x, psi1(a, x), label=r'$\psi_{1}(x) = e^{-a|x|}$')
    plt.plot(x, psi2(a, x), label=r'$\psi_{2}(x) = \frac{1}{x^2 + a^2}$')
    plt.plot(x, psi3(a, x), label=r'$\psi_{3}(x) = \frac{1}{(1 + ax^2)^2}$')
    plt.plot(x, psi4(a, x), label=r'$\psi_{4}(x) = e^{-ax^2}$', linestyle='--')
    plt.title(f'Trial Wavefunctions for a = {a}')
    plt.xlabel('x')
    plt.ylabel('Wavefunction')
    plt.legend()
    plt.grid(True)

plt.tight_layout()
plt.savefig('Trial_Wavefunctions_with_Psi4_ax2.pdf')
plt.show()
```

Listing : Trial wavefunction for different value of a

3 Trial Wave Functions under Optimal Parameters

```
import numpy as np
import matplotlib.pyplot as plt

# Trial wave functions
def trial_wavefunction_1(x, a=0.50):
    return np.exp(-a * x**2)

def trial_wavefunction_2(x, a=0.90):
    return np.exp(-a * np.abs(x))

def trial_wavefunction_3(x, a=0.70):
    return 1 / (x**2 + a)

def trial_wavefunction_4(x, a=0.40):
    return 1 / (1 + a * x**2)**2

# Generate x values
x = np.linspace(-5, 5, 400)

# Optimal parameters (example)
a_opt_1 = 0.50
a_opt_2 = 0.90
a_opt_3 = 0.70
a_opt_4 = 0.40

# Calculate wave functions with optimal parameters
y1 = trial_wavefunction_1(x, a_opt_1)
y2 = trial_wavefunction_2(x, a_opt_2)
y3 = trial_wavefunction_3(x, a_opt_3)
y4 = trial_wavefunction_4(x, a_opt_4)

# Plot the wave functions
plt.figure(figsize=(8, 6))
```

```
plt.plot(x, y1, label=r'$\psi_1(x)␣=␣e^{-a␣x^2},␣a␣=␣%.2f$'
    % a_opt_1, color='blue')
plt.plot(x, y2, label=r'$\psi_2(x)␣=␣e^{-a␣|x|},␣a␣=␣%.2f$'
    % a_opt_2, color='red')
plt.plot(x, y3, label=r'$\psi_3(x)␣=␣\frac{1}{x^2␣+␣a},␣a␣=␣
    %.2f$' % a_opt_3, color='green')
plt.plot(x, y4, label=r'$\psi_4(x)␣=␣\frac{1}{(1␣+␣a␣
    x^2)^2},␣a␣=␣%.2f$' % a_opt_4, color='purple')

plt.xlabel('x')
plt.ylabel('Wavefunction')
plt.title('Trial␣Wave␣Functions␣under␣Optimal␣Parameters')
plt.legend()
plt.grid(True)
plt.show()
```

Listing : Trial Wave Functions under Optimal Parameters

4 Minimum Energy vs Optimal value of a for Wavefunctions

```
import matplotlib.pyplot as plt

# Data from the table
wavefunctions = [
    r'$\psi_1(x)␣=␣e^{-a␣|x|}$',
    r'$\psi_2(x)␣=␣\frac{1}{x^2␣+␣a}$',
    r'$\psi_3(x)␣=␣\frac{1}{(1␣+␣a␣x^2)^2}$',
    r'$\psi_4(x)␣=␣e^{-a␣x^2}$'
]

min_energies = [0.6728, 0.6853, 0.5286, 0.4992]  # Minimum
    energy values
optimal_a = [0.90, 0.70, 0.40, 0.50]  # Corresponding
    optimal 'a' values

# Colors for each wavefunction
colors = ['blue', 'red', 'green', 'purple']

# Plotting
plt.figure(figsize=(8, 6))
for i in range(len(wavefunctions)):
    plt.scatter(optimal_a[i], min_energies[i],
        color=colors[i], label=wavefunctions[i])

# Adding labels and title with adjustments to avoid overlap
plt.xlabel('Optimal␣value␣of␣␣a')
plt.ylabel('Min␣Energy␣$(\hbar␣\omega)$')
plt.title('Minimum␣Energy␣vs␣Optimal␣␣value␣of␣a␣for␣
    Wavefunctions')

# Adjusting position of labels to avoid overlap
for i in range(len(wavefunctions)):
    if i == 1:  # Move the text below for the second
        wavefunction
        plt.text(optimal_a[i] + 0.01, min_energies[i] -
            0.02, wavefunctions[i], fontsize=12,
            color=colors[i])
    else:
        plt.text(optimal_a[i] + 0.01, min_energies[i] +
            0.01, wavefunctions[i], fontsize=12,
```

```
            color=colors[i])

plt.legend(loc=’upper␣left’)
plt.grid(True)
plt.tight_layout()  # Prevents overlap of title/labels
plt.show()
```

Listing : Minimum Energy vs Optimal value of a for Wavefunctions

5 Comparison of Trial Wavefunctions for different value of alpha

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.backends.backend_pdf import PdfPages

# Define constants in natural units
hbar = 1
k = 1
mu = 1
alpha0 = np.sqrt(k * mu / hbar**2)
omega = np.sqrt(k / mu)

# Define the range for x values
x = np.linspace(-20, 20, 500)

# Define the trial wavefunction
def trial_wave(alpha, xt):
    return np.exp(-alpha * xt**2)

# Define the first derivative of a function with respect to x
def dfdx(ft, xt):
    dx = xt[1] - xt[0]
    return np.gradient(ft, dx)

# Function to find the minimum of a function
def find_min(ft, xt):
    ftp = dfdx(ft, xt)
    min_idx = np.argmin(ft)
    return ft[min_idx], xt[min_idx]

# Kinetic energy operator
def T_phi(ft, xt):
    ftp = dfdx(ft, xt)
    ftpp = dfdx(ftp, xt)
    return -0.5 * ftpp

# Potential energy operator
def V_phi(ft, xt):
    return 0.5 * xt**2 * ft

# Functional for kinetic energy
def T_functional(ft, xt):
    tphi = T_phi(ft, xt)
    dx = xt[1] - xt[0]
    return np.trapz(ft * tphi, xt) / np.trapz(ft**2, xt)

# Functional for potential energy
def V_functional(ft, xt):
    vphi = V_phi(ft, xt)
    dx = xt[1] - xt[0]
```

```
    return np.trapz(ft * vphi, xt) / np.trapz(ft**2, xt)

# Main loop for different alpha values
alpha_values = np.linspace(0.05 * 3, 0.05 * 22, 20)
E_values = np.zeros_like(alpha_values)
T_values = np.zeros_like(alpha_values)
V_values = np.zeros_like(alpha_values)
E_ground_state = np.zeros_like(alpha_values)

for i, alpha in enumerate(alpha_values):
    # Evaluate trial wavefunction
    Phi_trial = trial_wave(alpha, x)
    # Evaluate functionals
    T_values[i] = T_functional(Phi_trial, x)
    V_values[i] = V_functional(Phi_trial, x)
    E_values[i] = T_values[i] + V_values[i]
    # Store true ground state energy
    E_ground_state[i] = omega * 0.5

# Plot T, V, E, Eg vs alpha
plt.plot(alpha_values, E_values, 'red', label='Total␣Energy')
plt.plot(alpha_values, T_values, 'blue', label='Kinetic␣Energy')
plt.plot(alpha_values, V_values, 'purple', label='Potential␣Energy')
plt.plot(alpha_values, E_ground_state, 'black', label='True␣Ground␣State␣Energy')
plt.xlabel('Alpha')
plt.ylabel('Energy')
plt.legend()
plt.title('Energy␣vs␣Alpha')
plt.savefig('Energy.pdf')  # Save as PDF
plt.show()

# Find the minimum energy
min_energy, min_alpha = find_min(E_values, alpha_values)
print("Minimum␣Energy:", min_energy)
print("Optimal␣Alpha:", min_alpha)
print("True␣Ground␣State␣Energy:", E_ground_state[0])

# Plot trial wavefunctions for comparison
Phi_trial_large_alpha = trial_wave(0.1, x)
Phi_trial_small_alpha = trial_wave(0.4, x)
Phi_trial_large_alpha1 = trial_wave(0.8, x)
Phi_trial_small_alpha2 = trial_wave(1.2, x)

plt.plot(x, Phi_trial_large_alpha, 'red', label='0.1')
plt.plot(x, Phi_trial_small_alpha, 'blue', label='0.4')
plt.plot(x, Phi_trial_large_alpha1, 'black', label='0.8')
plt.plot(x, Phi_trial_small_alpha2, 'green', label='1.2')
plt.xlabel('x')
plt.ylabel('Trial␣Wavefunction')
plt.legend()
plt.title('Comparison␣of␣Trial␣Wavefunctions␣for␣different␣value␣of␣alpha')
plt.savefig('Comparison_of_Trial_Wavefunctions.pdf')  # Save as PDF
plt.show()

# Create a PDF file to save all plots
with PdfPages('output_plots.pdf') as pdf:
    # Plot main energy vs alpha
    plt.plot(alpha_values, E_values, 'red', label='Total␣Energy')
```

```
    plt.plot(alpha_values, T_values, 'blue', label='Kinetic␣
       Energy')
    plt.plot(alpha_values, V_values, 'purple',
       label='Potential␣Energy')
    plt.plot(alpha_values, E_ground_state, 'black',
       label='True␣Ground␣State␣Energy')
    plt.xlabel('Alpha')
    plt.ylabel('Energy')
    plt.legend()
    plt.title('Energy␣vs␣Alpha')
    pdf.savefig()  # Save the current plot to the PDF
    plt.close()  # Close the current plot

    # Plot trial wavefunctions for comparison
    plt.plot(x, Phi_trial_large_alpha, 'red', label='0.1')
    plt.plot(x, Phi_trial_small_alpha, 'blue', label='0.4')
    plt.plot(x, Phi_trial_large_alpha1, 'black', label='0.8')
    plt.plot(x, Phi_trial_small_alpha2, 'green', label='1.2')
    plt.xlabel('x')
    plt.ylabel('Trial␣Wavefunction')
    plt.legend()
    plt.title('Comparison␣of␣Trial␣Wavefunctions␣for␣
       different␣value␣of␣alpha')
    pdf.savefig()  # Save the current plot to the PDF
    plt.close()  # Close the current plot
```

Listing : Comparison of Trial Wavefunctions for different value of alpha

6 Wavefunctions Comparison

```
import numpy as np
import matplotlib.pyplot as plt

# Define constants in natural units
hbar = 1
k = 1
mu = 1
omega = np.sqrt(k / mu)

# Define the range for x values
x = np.linspace(-5, 5, 500)

# Define trial wavefunctions
def psi1(a, x):
    return np.exp(-a * np.abs(x))

def psi2(a, x):
    return 1 / (x**2 + a)

def psi3(a, x):
    return 1 / (1 + a * x**2)**2

# Define the true ground state wavefunction of the quantum
   harmonic oscillator
def psi_true(m, omega, x, n=1):
    return (m * omega / (n * hbar))**0.25 * np.exp(-m *
       omega * x**2 / (2 * n))

# Parameter 'a' for the trial wavefunctions
a = 0.1

```

```
# Normalization constants
A1 = 1 / np.sqrt(np.trapz(psi1(a, x)**2, x))
A2 = 1 / np.sqrt(np.trapz(psi2(a, x)**2, x))
A3 = 1 / np.sqrt(np.trapz(psi3(a, x)**2, x))
A_true = 1 / np.sqrt(np.trapz(psi_true(mu, omega, x)**2, x))

# Evaluate the wavefunctions
psi1_values = A1 * psi1(a, x)
psi2_values = A2 * psi2(a, x)
psi3_values = A3 * psi3(a, x)
psi_true_values = A_true * psi_true(mu, omega, x)

# Plotting the wavefunctions
plt.figure(figsize=(10, 6))
plt.plot(x, psi1_values, label=r'$\psi_1(x)=Ae^{-a|x|}$', color='red')
plt.plot(x, psi2_values, label=r'$\psi_2(x)=\frac{A}{x^2␣+␣a}$', color='blue')
plt.plot(x, psi3_values, label=r'$\psi_3(x)=\frac{A}{(1+ax^2)^2}$', color='green')
plt.plot(x, psi_true_values, label=r'$\psi_{\mathrm{true}}(x)=\left(\frac{m\omega}{\hbar}\right)^{1/4}␣e^{-\frac{m\omega␣x^2}{2}}$', color='black', linestyle='--')

plt.xlabel('x')
plt.ylabel(r'$\psi(x)$')
plt.title('Wavefunctions␣Comparison')
plt.legend()
plt.grid(True)
plt.savefig('Wavefunctions_Comparison.png')
plt.show()
```

Listing : Wavefunctions Comparison

```
import numpy as np
import matplotlib.pyplot as plt

# Define constants in natural units
hbar = 1
k = 1
mu = 1
omega = np.sqrt(k / mu)

# Define the range for x values
x = np.linspace(-20, 20, 500)

# Define trial wavefunctions
def psi1(a, x):
    return np.exp(-a * np.abs(x))

def psi2(a, x):
    return 1 / (x**2 + a)

def psi3(a, x):
    return 1 / (1 + a * x**2)**2

def psi4(a, x):
    return np.exp(-a * x**2)

# Normalize a wave function
```

```
def normalize_wavefunction(psi, x):
    norm_factor = np.sqrt(np.trapz(np.abs(psi)**2, x))  # Integral of |psi(x)|^2
    return psi / norm_factor  # Normalize the wave function

# Define the first derivative of a function with respect to x
def dfdx(ft, xt):
    dx = xt[1] - xt[0]
    return np.gradient(ft, dx)

# Kinetic energy operator
def T_phi(ft, xt):
    ftp = dfdx(ft, xt)
    ftpp = dfdx(ftp, xt)
    return -0.5 * ftpp

# Potential energy operator
def V_phi(ft, xt):
    return 0.5 * xt**2 * ft

# Functional for kinetic energy
def T_functional(ft, xt):
    tphi = T_phi(ft, xt)
    return np.trapz(ft * tphi, xt) / np.trapz(ft**2, xt)

# Functional for potential energy
def V_functional(ft, xt):
    vphi = V_phi(ft, xt)
    return np.trapz(ft * vphi, xt) / np.trapz(ft**2, xt)

# Define different values of a
a_values = np.linspace(0.1, 2.0, 20)

# Initialize dictionaries for storing energy values
trial_wavefunctions = [psi1, psi2, psi3, psi4]
T_values = {psi: [] for psi in trial_wavefunctions}
V_values = {psi: [] for psi in trial_wavefunctions}
E_values = {psi: [] for psi in trial_wavefunctions}

exact_energy = 0.5 * omega

# Minimum energy, optimal a, and true ground state energy storage
min_energy = {psi: float(’inf’) for psi in trial_wavefunctions}
optimal_a = {psi: None for psi in trial_wavefunctions}

# Compute energies
for psi in trial_wavefunctions:
    energies = []
    for a in a_values:
        Phi_trial = psi(a, x)
```

```python
        Phi_trial_normalized = normalize_wavefunction(Phi_trial, x)  # Normalize the wave function
        T = T_functional(Phi_trial_normalized, x)
        V = V_functional(Phi_trial_normalized, x)
        E = T + V
        energies.append(E)
        T_values[psi].append(T)
        V_values[psi].append(V)
        E_values[psi].append(E)

    # Find minimum energy and corresponding a
    min_energy[psi] = min(energies)
    optimal_a[psi] = a_values[np.argmin(energies)]

# Plotting
labels = [
    r'$\psi_{1}(x) = e^{-a|x|}$',
    r'$\psi_{2}(x) = \frac{1}{x^2 + a}$',
    r'$\psi_{3}(x) = \frac{1}{(1 + ax^2)^2}$',
    r'$\psi_{4}(x) = e^{-ax^2}$'
]
colors = ['red', 'blue', 'green', 'purple']

fig, axs = plt.subplots(2, 2, figsize=(15, 12))

# Plot Kinetic Energy
for i, psi in enumerate(trial_wavefunctions):
    axs[0, 0].plot(a_values, T_values[psi], label=labels[i], color=colors[i])
axs[0, 0].axhline(y=exact_energy, color='black', linestyle='--', label='Exact Energy')
axs[0, 0].set_xlabel('a')
axs[0, 0].set_ylabel('Kinetic Energy ($\hbar \omega$)')
axs[0, 0].set_title('Kinetic Energy vs a')
axs[0, 0].legend(loc='upper right')

# Plot Potential Energy
for i, psi in enumerate(trial_wavefunctions):
    axs[0, 1].plot(a_values, V_values[psi], label=labels[i], color=colors[i])
axs[0, 1].axhline(y=exact_energy, color='black', linestyle='--', label='Exact Energy')
axs[0, 1].set_xlabel('a')
axs[0, 1].set_ylabel('Potential Energy ($\hbar \omega$)')
axs[0, 1].set_title('Potential Energy vs a')
axs[0, 1].legend(loc='upper right')

# Plot Total Energy
for i, psi in enumerate(trial_wavefunctions):
    axs[1, 0].plot(a_values, E_values[psi], label=labels[i], color=colors[i])
axs[1, 0].axhline(y=exact_energy, color='black', linestyle='--', label='Exact Energy')
axs[1, 0].set_xlabel('a')
```

```
axs[1, 0].set_ylabel(’Total Energy ($\hbar \omega$)’)
axs[1, 0].set_title(’Total Energy vs a’)
axs[1, 0].legend(loc=’upper right’)

# Plot Exact Energy
axs[1, 1].axhline(y=exact_energy, color=’black’, linestyle=’--’, label=’Exact Energy’)
axs[1, 1].set_xlabel(’a’)
axs[1, 1].set_ylabel(’Energy ($\hbar \omega$)’)
axs[1, 1].set_title(’Exact Energy’)
axs[1, 1].legend(loc=’upper right’)

# Add Table
cell_text = [
    [labels[i], f’{min_energy[psi]:.4f}’, f’{optimal_a[psi]:.2f}’]
    for i, psi in enumerate(trial_wavefunctions)
]
columns = [’Wavefunction’, ’Min Energy ($\hbar \omega$)’, ’Optimal a’]

# Adjust space for table and text
plt.subplots_adjust(bottom=0.3)  # Adjust space to prevent overlap

# Table below the plots
table = plt.table(
    cellText=cell_text,
    colLabels=columns,
    cellLoc=’center’,
    loc=’bottom’,
    bbox=[0, -0.35, 1, 0.3]
)

# Add external text below the table
plt.figtext(
    0.5, -0.55,  # Position text below the table
    ’\n’.join([
        f’{labels[i]}: Min Energy = {min_energy[psi]:.4f}, Optimal a = {optimal_a[psi]:.2f}’
        for i, psi in enumerate(trial_wavefunctions)
    ]),
    fontsize=10, color=’black’, ha=’center’, va=’top’
)

plt.tight_layout()
plt.savefig(’Energy_vs_a_with_normalization_and_table.png’)
plt.show()
```

Listing : Comparison of Energies for Different Trial Wavefunctions

```
import numpy as np
import matplotlib.pyplot as plt

# Define constants in natural units
hbar = 1
```

```
k = 1
mu = 1
omega = np.sqrt(k / mu)

# Define the range for x values
x = np.linspace(-20, 20, 500)

# Define trial wavefunctions
def psi1(a, x):
    return np.exp(-a * np.abs(x))

def psi2(a, x):
    return 1 / (x**2 + a)

def psi3(a, x):
    return 1 / (1 + a * x**2)**2

def psi4(a, x):
    return np.exp(-a * x**2)  # Corrected wavefunction: e^(-a * x^2)

# Define the first derivative of a function with respect to x
def dfdx(ft, xt):
    dx = xt[1] - xt[0]
    return np.gradient(ft, dx)

# Kinetic energy operator
def T_phi(ft, xt):
    ftp = dfdx(ft, xt)
    ftpp = dfdx(ftp, xt)
    return -0.5 * ftpp

# Potential energy operator
def V_phi(ft, xt):
    return 0.5 * xt**2 * ft

# Functional for kinetic energy
def T_functional(ft, xt):
    tphi = T_phi(ft, xt)
    return np.trapz(ft * tphi, xt) / np.trapz(ft**2, xt)

# Functional for potential energy
def V_functional(ft, xt):
    vphi = V_phi(ft, xt)
    return np.trapz(ft * vphi, xt) / np.trapz(ft**2, xt)

# Values for parameter ’a’
a_values = [0.90, 0.70, 0.40, 0.50]

trial_wavefunctions = [psi1, psi2, psi3, psi4]  # Add new wavefunction
energies = {a: [] for a in a_values}  # Store energies for each a value
energy_deviations = {a: [] for a in a_values}  #
```

```python
    Store energy deviations for each a value

# Loop over each value of 'a'
for a in a_values:
    for psi in trial_wavefunctions:
        Phi_trial = psi(a, x)
        T = T_functional(Phi_trial, x)
        V = V_functional(Phi_trial, x)
        E = T + V
        energies[a].append(E)
        energy_deviations[a].append(np.abs((E - 0.5 * omega) / (0.5 * omega)) * 100)

# Plotting
labels = [
    r'$\psi_{1}(x, a)=e^{-a|x|}$',
    r'$\psi_{2}(x, a)=\frac{1}{x^2 + a}$',
    r'$\psi_{3}(x, a)=\frac{1}{(1+ax^2)^2}$',
    r'$\psi_{4}(x, a)=e^{-a x^2}$'
]

# Create subplots with 2 rows and enough columns for all `a_values`
fig, axs = plt.subplots(2, len(a_values), figsize=(20, 10))

for i, a in enumerate(a_values):
    # Energy plot
    axs[0, i].bar(labels, energies[a], color=['red', 'blue', 'green', 'purple'])
    axs[0, i].axhline(y=0.5 * omega, color='black', linestyle='--', label='True Ground State Energy')
    axs[0, i].set_ylabel('Energy ($\hbar \omega$)', fontsize=10)
    axs[0, i].set_title(f'Energy for a={a}', fontsize=12)
    axs[0, i].legend(fontsize=8)
    axs[0, i].set_xticklabels(labels, fontsize=12, rotation=45, ha='right')  # Adjust font size and rotation

    # Energy deviation plot
    axs[1, i].bar(labels, energy_deviations[a], color=['red', 'blue', 'green', 'purple'])
    axs[1, i].set_ylabel('Energy Deviation from $0.5\, \hbar \omega$ (%)', fontsize=10)
    axs[1, i].set_title(f'Energy Deviation for a={a}', fontsize=12)
    axs[1, i].set_xticklabels(labels, fontsize=12, rotation=45, ha='right')  # Adjust font size and rotation

plt.tight_layout()
plt.savefig('Comparison_of_Energies_and_Energy_Deviations.png')
plt.show()
```

Listing : Comparison of Energies for Different Trial Wavefunctions

```

import numpy as np
import matplotlib.pyplot as plt

# Define constants in natural units
hbar = 1
k = 1
mu = 1
omega = np.sqrt(k / mu)

# Define the range for x values with high resolution
x = np.linspace(-3000, 3000, 5000000)

# Define trial wavefunctions
def psi1(a, x):
    return np.exp(-a * np.abs(x))

def psi2(a, x):
    return 1 / (x**2 + a)

def psi3(a, x):
    return 1 / (1 + a * x**2)**2

def psi4(a, x):
    return np.exp(-a * x**2)

# Define hybrid wavefunctions
def hybrid1(a, x):
    return psi1(a, x) + psi4(a, x)

def hybrid2(a, x):
    return psi2(a, x) + psi3(a, x)

# Normalize a wave function
def normalize_wavefunction(psi, x):
    norm_factor = np.sqrt(np.trapz(np.abs(psi)**2, x))  # Integral of |psi(x)|^2
    return psi / norm_factor  # Normalize the wave function

# Second derivative using a central difference scheme
def dfdx(ft, xt):
    dx = xt[1] - xt[0]
    return np.gradient(ft, dx)

# Kinetic energy operator
def T_phi(ft, xt):
    ftp = dfdx(ft, xt)
    ftpp = dfdx(ftp, xt)
    return -0.5 * ftpp

# Potential energy operator
def V_phi(ft, xt):
    return 0.5 * xt**2 * ft
```

```

# Functionals for kinetic and potential energy
def T_functional(ft, xt):
    tphi = T_phi(ft, xt)
    return np.trapz(ft * tphi, xt) / np.trapz(ft**2, xt)

def V_functional(ft, xt):
    vphi = V_phi(ft, xt)
    return np.trapz(ft * vphi, xt) / np.trapz(ft**2, xt)

# Define different values of a with higher resolution
a_values = np.linspace(0.1, 2.0, 100)

# Initialize dictionaries for storing energy values
trial_wavefunctions = [psi1, psi2, psi3, psi4, hybrid1, hybrid2]
labels = [
    r'$\psi_{1}(x) = e^{-a|x|}$',
    r'$\psi_{2}(x) = \frac{1}{x^2 + a}$',
    r'$\psi_{3}(x) = \frac{1}{(1 + ax^2)^2}$',
    r'$\psi_{4}(x) = e^{-ax^2}$',
    r'Hybrid 1: $\psi_{1} + \psi_{4}$',
    r'Hybrid 2: $\psi_{2} + \psi_{3}$'
]

# Store energies and optimal parameters
min_energy = {}
optimal_a = {}
T_values = {psi: [] for psi in trial_wavefunctions}
V_values = {psi: [] for psi in trial_wavefunctions}
E_values = {psi: [] for psi in trial_wavefunctions}

# Exact energy of the quantum harmonic oscillator ground state
exact_energy = 0.5 * omega

# Compute energies for each trial wavefunction
for psi in trial_wavefunctions:
    energies = []
    for a in a_values:
        Phi_trial = psi(a, x)
        Phi_trial_normalized = normalize_wavefunction(Phi_trial, x)  # Normalize the wave function

        T = T_functional(Phi_trial_normalized, x)
        V = V_functional(Phi_trial_normalized, x)
        E = T + V

        T_values[psi].append(T)
        V_values[psi].append(V)
        E_values[psi].append(E)
        energies.append(E)

    # Find minimum energy and corresponding a
```

```python
    min_energy[psi] = min(energies)
    optimal_a[psi] = a_values[np.argmin(energies)]

colors = ['red', 'blue', 'green', 'purple', 'orange', 'cyan']
fig, axs = plt.subplots(2, 2, figsize=(15, 12))

# Plot Kinetic Energy
for i, psi in enumerate(trial_wavefunctions):
    axs[0, 0].plot(a_values, T_values[psi], label=labels[i], color=colors[i])
axs[0, 0].axhline(y=exact_energy, color='black', linestyle='--', label='Exact Energy')
axs[0, 0].set_xlabel('a')
axs[0, 0].set_ylabel('Kinetic Energy ($\hbar \omega$)')
axs[0, 0].set_title('Kinetic Energy vs a')
axs[0, 0].legend(loc='upper right')

# Plot Potential Energy
for i, psi in enumerate(trial_wavefunctions):
    axs[0, 1].plot(a_values, V_values[psi], label=labels[i], color=colors[i])
axs[0, 1].axhline(y=exact_energy, color='black', linestyle='--', label='Exact Energy')
axs[0, 1].set_xlabel('a')
axs[0, 1].set_ylabel('Potential Energy ($\hbar \omega$)')
axs[0, 1].set_title('Potential Energy vs a')
axs[0, 1].legend(loc='upper right')

# Plot Total Energy
for i, psi in enumerate(trial_wavefunctions):
    axs[1, 0].plot(a_values, E_values[psi], label=labels[i], color=colors[i])
axs[1, 0].axhline(y=exact_energy, color='black', linestyle='--', label='Exact Energy')
axs[1, 0].set_xlabel('a')
axs[1, 0].set_ylabel('Total Energy ($\hbar \omega$)')
axs[1, 0].set_title('Total Energy vs a')
axs[1, 0].legend(loc='upper right')

# Plot Exact Energy in the fourth subplot
axs[1, 1].axhline(y=exact_energy, color='black', linestyle='--', label='Exact Energy')
axs[1, 1].set_xlabel('a')
axs[1, 1].set_ylabel('Energy ($\hbar \omega$)')
axs[1, 1].set_title('Exact Energy')
axs[1, 1].legend(loc='upper right')

# Add Table for Results
cell_text = [
    [labels[i], f'{min_energy[psi]:.4f}', f'{optimal_a[psi]:.2f}', f'{exact_energy:.4f}']
    for i, psi in enumerate(trial_wavefunctions)
]
```

```
columns = ['Wavefunction', 'Min Energy ($\hbar
    \omega$)', 'Optimal a', 'Exact Energy ($\hbar
    \omega$)']
plt.subplots_adjust(bottom=0.3)

# Table below plots
plt.table(
    cellText=cell_text,
    colLabels=columns,
    cellLoc='center',
    loc='bottom',
    bbox=[0, -0.4, 1, 0.3]
)

# Display all plots
plt.tight_layout()
plt.show()
```

Listing : Comparison of Energies for Different Trial Hybrid Wavefunctions

```
import numpy as np
import matplotlib.pyplot as plt

# Define constants in natural units
hbar = 1
k = 1
mu = 1
omega = np.sqrt(k / mu)
lambda_anharmonic = 0.1  # Anharmonicity parameter
    (you can vary this)

# Define the range for x values with high resolution
x = np.linspace(-1000, 1000, 50000)

# Define trial wavefunctions
def psi1(a, x):
    return np.exp(-a * np.abs(x))

def psi2(a, x):
    return 1 / (x**2 + a)

def psi3(a, x):
    return 1 / (1 + a * x**2)**2

def psi4(a, x):
    return np.exp(-a * x**2)

# Define hybrid wavefunctions
def hybrid1(a, x):
    return psi1(a, x) + psi4(a, x)

def hybrid2(a, x):
    return psi2(a, x) + psi3(a, x)

# Normalize a wave function
def normalize_wavefunction(psi, x):
    norm_factor = np.sqrt(np.trapz(np.abs(psi)**2,
```

```
       x))  # Integral of |psi(x)|^2
    return psi / norm_factor  # Normalize the wave
       function

# Second derivative using a central difference scheme
def dfdx(ft, xt):
    dx = xt[1] - xt[0]
    return np.gradient(ft, dx)

# Kinetic energy operator
def T_phi(ft, xt):
    ftp = dfdx(ft, xt)
    ftpp = dfdx(ftp, xt)
    return -0.5 * ftpp

# Potential energy operator (including anharmonic
   term)
def V_phi(ft, xt):
    harmonic_term = 0.5 * xt**2
    anharmonic_term = lambda_anharmonic * xt**4  #
       Anharmonic potential
    return (harmonic_term + anharmonic_term) * ft

# Functionals for kinetic and potential energy
def T_functional(ft, xt):
    tphi = T_phi(ft, xt)
    return np.trapz(ft * tphi, xt) / np.trapz(ft**2,
       xt)

def V_functional(ft, xt):
    vphi = V_phi(ft, xt)
    return np.trapz(ft * vphi, xt) / np.trapz(ft**2,
       xt)

# Define different values of a with higher resolution
a_values = np.linspace(0.1, 2.0, 100)

# Initialize dictionaries for storing energy values
trial_wavefunctions = [psi1, psi2, psi3, psi4,
   hybrid1, hybrid2]
labels = [
    r'$\psi_{1}(x) = e^{-a|x|}$',
    r'$\psi_{2}(x) = \frac{1}{x^2 + a}$',
    r'$\psi_{3}(x) = \frac{1}{(1 + ax^2)^2}$',
    r'$\psi_{4}(x) = e^{-ax^2}$',
    r'Hybrid 1: $\psi_{1} + \psi_{4}$',
    r'Hybrid 2: $\psi_{2} + \psi_{3}$'
]

# Store energies and optimal parameters
min_energy = {}
optimal_a = {}
T_values = {psi: [] for psi in trial_wavefunctions}
V_values = {psi: [] for psi in trial_wavefunctions}
E_values = {psi: [] for psi in trial_wavefunctions}

# Exact energy of the quantum harmonic oscillator
```

```
    ground state (for comparison)
exact_energy = 0.5 * omega

# Compute energies for each trial wavefunction
for psi in trial_wavefunctions:
    energies = []
    for a in a_values:
        Phi_trial = psi(a, x)
        Phi_trial_normalized = normalize_wavefunction(Phi_trial, x)  # Normalize the wave function

        T = T_functional(Phi_trial_normalized, x)
        V = V_functional(Phi_trial_normalized, x)
        E = T + V

        T_values[psi].append(T)
        V_values[psi].append(V)
        E_values[psi].append(E)
        energies.append(E)

    # Find minimum energy and corresponding a
    min_energy[psi] = min(energies)
    optimal_a[psi] = a_values[np.argmin(energies)]

colors = ['red', 'blue', 'green', 'purple', 'orange', 'cyan']
fig, axs = plt.subplots(2, 2, figsize=(15, 12))

# Plot Kinetic Energy
for i, psi in enumerate(trial_wavefunctions):
    axs[0, 0].plot(a_values, T_values[psi], label=labels[i], color=colors[i])
axs[0, 0].axhline(y=exact_energy, color='black', linestyle='--', label='Exact Energy')
axs[0, 0].set_xlabel('a')
axs[0, 0].set_ylabel('Kinetic Energy ($\hbar \omega$)')
axs[0, 0].set_title('Kinetic Energy vs a')
axs[0, 0].legend(loc='upper right')

# Plot Potential Energy
for i, psi in enumerate(trial_wavefunctions):
    axs[0, 1].plot(a_values, V_values[psi], label=labels[i], color=colors[i])
axs[0, 1].axhline(y=exact_energy, color='black', linestyle='--', label='Exact Energy')
axs[0, 1].set_xlabel('a')
axs[0, 1].set_ylabel('Potential Energy ($\hbar \omega$)')
axs[0, 1].set_title('Potential Energy vs a')
axs[0, 1].legend(loc='upper right')

# Plot Total Energy
for i, psi in enumerate(trial_wavefunctions):
    axs[1, 0].plot(a_values, E_values[psi], label=labels[i], color=colors[i])
```

```
axs[1, 0].axhline(y=exact_energy, color='black', linestyle='--', label='Exact Energy')
axs[1, 0].set_xlabel('a')
axs[1, 0].set_ylabel('Total Energy ($\hbar \omega$)')
axs[1, 0].set_title('Total Energy vs a')
axs[1, 0].legend(loc='upper right')

# Plot Minimum Energy vs Optimal a (new plot)
for i, psi in enumerate(trial_wavefunctions):
    axs[1, 1].plot(optimal_a[psi], min_energy[psi], 'o', label=labels[i], color=colors[i])
axs[1, 1].set_xlabel('Optimal a')
axs[1, 1].set_ylabel('Min Energy ($\hbar \omega$)')
axs[1, 1].set_title('Min Energy vs Optimal a')
axs[1, 1].legend(loc='upper right')  # Add the legend here

# Add Table for Results (without Exact Energy column)
cell_text = [
    [labels[i], f'{min_energy[psi]:.4f}', f'{optimal_a[psi]:.2f}']
    for i, psi in enumerate(trial_wavefunctions)
]
columns = ['Wavefunction', 'Min Energy ($\hbar \omega$)', 'Optimal a']
plt.subplots_adjust(bottom=0.3)

# Table below plots
plt.table(
    cellText=cell_text,
    colLabels=columns,
    cellLoc='center',
    loc='bottom',
    bbox=[0, -0.4, 1, 0.3]
)

# Display all plots
plt.tight_layout()
plt.show()
```

Listing : Anharmonic oscillator