Systematic Asymptotic Approximation Based On Variational Methods

Z.X. Wang
Shanghai Jiaotong University

We present a method for approximating the ground state wavefunction in quantum systems using an n-th order series expansion. This approach hypothesises that the estimated ground state energy will asymptotically approach the true ground state energy as the number of free parameters in the series expansion increases. As the order of the expansion increases, the method promises to converge to the true wavefunction, providing a robust framework for high-precision quantum simulations. Compared to traditional numerical methods, the variational asymptotic approximation can effectively deal with the infinity problem.

INTRODUCTION

The variational method is a cornerstone of quantum mechanics for approximating the ground state energy and wavefunction of a quantum system. It is based on the variational principle, which states that the expectation value of the Hamiltonian for any trial wavefunction is an upper bound on the true ground state energy. Mathematically, for a given trial wavefunction ψ_T, the variational energy E_T is given by

$$E_T = \frac{\langle\psi_T|\hat{H}|\psi_T\rangle}{\langle\psi_T|\psi_T\rangle}$$

where $\hat{H}$ is the Hamiltonian operator. The variational principle guarantees that $E_T \geq E_0$, with E_0 being the exact ground state energy. By minimizing E_T over all possible ψ_T, one can approach E_0.

Despite its elegance, the variational method faces two major challenges. First, it is impractical to infinitely increase the degrees of freedom in an analytical function to refine the approximation, as this would require an infinite dimensional parameter space. Second, it is difficult to estimate and systematically improve the accuracy of the approximate wave function. The variational energy provides an upper bound, but there is no direct measure of how close this bound is to the true ground state energy. Since we can't know how accurate the ground state energy is, we can't know the ground state wavefunction.

The limitations described above motivate the exploration of alternative approaches based on the variational method. This paper presents a method that leverages the series expansion to approximate the ground state wavefunction. By incrementally increasing the number of terms in the series, it is possible to systematically enhance the approximation's accuracy and address the challenges inherent in the conventional variational method.

CONVERGENCE OF VARIATIONAL ENERGY

In this section, we explore the convergence of the variational energy as the number of degrees of freedom in our trial wavefunction increases. The central idea is that for a given n-dimensional trial wavefunction, we can find a minimum energy E_n. However, by adding an additional degree of freedom, we can find a new minimum energy E_{n+1} that is lower than E_n. This process can be iteratively applied to demonstrate that the variational energy converges to the true ground state energy as the number of degrees of freedom approaches infinity.

Let's consider a trial wavefunction $\psi(s_1, s_2, \cdots, s_n)$ that depends on n parameters. According to the variational principle, the energy expectation value E_n for this wavefunction is given by:

$$E_{T_n} = E(s_1, s_2, \cdots, s_n)|_{\frac{dE}{ds_i}=0}$$

Now, consider extending our trial wavefunction to $\psi(s_1, s_2, \cdots, s_{n+1})$ by adding one more parameter. This new wavefunction has an additional degree of freedom, allowing it to explore a larger space of possible wavefunctions. The energy expectation value is:

$$E_{T_{n+1}} = E(s_1, s_2, \cdots, s_{n+1}) = E_{T_n}(s_{n+1})|_{\frac{dE_{T_n}}{ds_{n+1}}=0}$$

which is apparent that:

$$E_{T_{n+1}} \leq E_{T_n}$$

Although adding freedom make the Variational Energy lower, it is still an upper bound of ground state energy, which means $\lim_{n\to\infty} E_n \geq E_0$. However, if we hypothesize that the fundamental wave function, such a function that converges to 0 at a distance and has no singularities, can be convergently fitted by a set of bases, then we can assume that

$$\lim_{n\to\infty} E_{T_n} = E_0, \psi_0 = \psi(s_1, s_2, \cdots)$$

In this way we not only approximate the ground state energy, but also give an asymptotic form of the ground state wave function.

EXAMPLE: INFINITE WELL

Take the infinite well in natural units as an example:

$$\hat{H} = -\frac{\partial^2}{\partial x^2} + V, V = \{0, |x| < 1; \infty, |x| > 1\}$$

The ground state energy is:

$$E_0 = \frac{\pi^2}{4} = 2.46740, \psi_0 = 1 - \frac{\pi^2}{8}x^2 = 1 - 1.2337x^2$$

If we use traditional variational methods, we can only presuppose an analytic function and add a finite number of degrees of freedom to it. But if we use Taylor expansion to approximate it, we can set:

$$\psi_0 = \sum_{n=0}^{\infty} s_n(1 - x^n)$$

For formal calculations, we take the first five terms as particles and remove the chi-square terms and normalise them:

$$\psi_0 = (1 - x^2) + s_4(1 - x^4)$$

Using `Mathematica` to solve, we have:

$$E_T = 2.46744, \psi = 1 - x^2 - 0.180831(1 - x^4)$$

This result is surprising so we can see the benefits of the systematic approach when we compute higher order approximations numerically. Using `Python` to plot the accuracy vs the highest order:

FIG. 1. accuracy vs the highest order

Before the 10th order, the energy and wave function errors decrease exponentially with the order; after the 10th order, they cannot continue to decrease due to the effect of the tolerance of the multi-objective optimisation algorithm.

EXAMPLE: HARMONIC OSCILLATOR

It has been determined that, in the context of diverse problems, a concerted effort is required to meticulously design the foundation of the wave function, with the objective of averting non-self-adjointness. In the case of infinitely deep potential wells, the wave function must be constrained to ensure that its value on both sides is zero. In the context of simple harmonic oscillators, the wave function must be constrained to converge at infinity.

For the problem under consideration:

$$\hat{H} = -\frac{\partial^2}{\partial x^2} + x^2$$

We are aware that:

$$E_0 = 1, \psi_0(x) = e^{-\frac{1}{2}x^2}$$

Set:

$$\psi = \sum_{n=0} s_{2n} x^{2n} e^{-x^2}$$

We have:

$$\sum_{n=0} s_{2n} x^{2n} = e^{\frac{1}{2}x^2} = 1 + 0.5x^2 + 0.125x^4 + \cdots$$

Use `Python` to solve the problem(first 6 order):

$$E_T = 1.00061505, \psi = 1 + 0.53x^2 + 0.066x^4 + 0.0496x^6$$

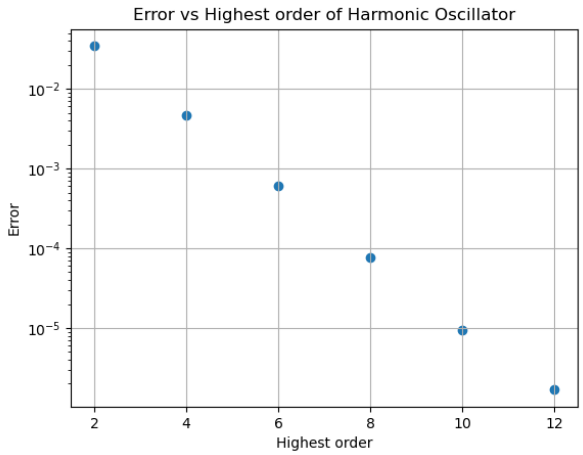

FIG. 2. accuracy vs the highest order

The error decays as fast as the exponential decay!

PROSPECTS AND LIMITATIONS

For the ground state energy of the helium atom discussed in class, we use this equation to estimate the ground state wave function:

$$\psi(r_1, r_2) = \frac{Z^3}{\pi a^3} e^{-Z\frac{r_1 + r_2}{a}}$$

As we discussed earlier, this function adds at most powers of distance like:

$$\psi(r_1, r_2) = Ae^{-Z\frac{(r_1+r_2)^n}{a}}$$

to the equivalent number of electrons Z and cannot add more parameter space. However, the use of the systematic approximation method we have proposed opens up the possibility of solving the ground state wave function for helium atoms and even for other multielectron atoms. On the basis of numerically solving the ground state wave function, it is of course possible to solve the excited state wave function.

However, it is evident that the convergence of this method is significantly less than that of the analytical method, as the contribution of the added degrees of freedom gradually diminishes. Furthermore, the presence of local minima and arithmetic problems in optimisation algorithms with multiple degrees of freedom poses significant challenges.

In comparison to equivalent numerical iterative methods, the variational-based asymptotic approximation has been shown to be an effective solution to the infinity problem, which numerical iterative methods are unable to resolve. Furthermore, the asymptotic approximation method is capable of effectively measuring the magnitude of the co-contributions of the different terms and providing an analytical approximation, while the numerical iterative method can only provide interpolated values, which require subsequent fitting operations.

PYTHON CODE

Infinite Well

```
import numpy as np
from scipy.integrate import quad
from scipy.optimize import minimize

def calculate_denominator(s):
def integrand(x):
terms = [s[i-1] * (1-x**(2*i+2)) for i in range(1, len(s)+1)]
return (1 -x**2+ sum(terms))**2
result, _ = quad(integrand, -1, 1)
return result

def calculate_numerator(s):
def sum_of_terms(x):
return sum(s[i-1] * (1-x**(2*i+2)) for i in range (1,len(s)+1))

def second_derivative(x):
second_derivative = sum(2*(i+1)*(2*i+1)*s[i-1]*x**(2*i) for i in range(1, len(s)+1))
return second_derivative

def integrand(x):
return -(1-x**2 + sum_of_terms(x)) *(-2-second_derivative(x))
result, _ = quad(integrand, -1, 1)
return result

def objective_function(s):
denominator = calculate_denominator(s)
numerator = calculate_numerator(s)
return numerator / denominator
def optimize_integral(n):
initial_coefficients = np.ones((n,))
bounds = [(None, None) for _ in range(n)]
result = minimize(objective_function, initial_coefficients, method='SLSQP', bounds=bounds)
print('Optimized Coefficients:', result.x)
print('Minimum Value of the Objective Function:', result.fun)
return {'coefficients': result.x, 'objective_function_value': result.fun}

n = 1
optimized_result = optimize_integral(n)
```

Harmonic Oscillator

```
import matplotlib.pyplot as plt
import numpy as np
from scipy.integrate import quad
from scipy.optimize import minimize
import sympy as sp
from matplotlib.ticker import MaxNLocator, ScalarFormatter

def calculate_denominator(s):
def integrand(x):
return (np.exp(-x**2) + sum(s[i-1] * x**(2*i) *np.exp(-x**2) for i in range(1,len(s)+1)))**2
result, _ = quad(integrand, -np.inf, np.inf)
return result

def calculate_numerator(s):
x = sp.symbols('x')
sum_of_terms = sp.exp(-x**2) + sum(s[i-1] * x**(2*i) ) * sp.exp(-x**2) for i in range(1, len(s)+1))
second_derivative = sum_of_terms.diff(x, 2)
x2 = x**2*sp.exp(-x**2)+sum(s[i-1] * x**(2*i+2) * sp.exp(-x**2) for i in range(1, len(s)+1))
integrand = sum_of_terms * (-second_derivative + x2 )
result = sp.integrate(integrand, (x, -sp.oo, sp.oo) )
return result.evalf()

def objective_function(s):
denominator = calculate_denominator(s)
numerator = calculate_numerator(s)
return numerator / denominator
def optimize_integral(n):
initial_coefficients = np.ones((n,))
bounds = [(None, None) for _ in range(n)]
result = minimize(objective_function, initial_coefficients, method='SLSQP', bounds=bounds)
print('Optimized Coefficients:', result.x)
print('Minimum Value of the Objective Function:', result.fun)
return {'coefficients': result.x, 'objective_function_value': result.fun}
```

```

num=4
error=np.zeros(num)
for i in range(num):
n = i+1
optimized_result = optimize_integral(n)
error[i]=optimized_result['objective_function_value
    ']-1
np.save('error2.npy',error)
plt.scatter([2,4,6,8],error)
plt.grid(True)
plt.xscale('log')
plt.yscale('log')
plt.xlabel('Highest order')
plt.ylabel('Error')
plt.title('Error vs Highest order of Infinite Well
    Wavefunction')
ax = plt.gca()
ax.xaxis.set_major_formatter(ScalarFormatter(
    useMathText=True))
ax.xaxis.set_major_locator(MaxNLocator(4))
plt.show()

```