

NATIONAL UNIVERSITY OF SINGAPORE

FACULTY OF SCIENCE

DEPARTMENT OF PHYSICS

**Generalized Matrix Numerov
Solutions to
the Schrödinger Equation**

A thesis presented for the degree of Bachelor of Science with Honours in Physics

Author:

TANG Dongjiao

Supervisors:

Dr. YEO Ye

Mr. Andreas DEWANTO

April 21, 2014

Abstract

The 1-D time-independent Schrödinger equation is an ordinary differential equations that can be solved numerically using the well-known Numerov method. Recently the basic version of Numerov method has been recast into the Basic Matrix Numerov Method which has great advantage when used in modern high-level programming environments but produces results only to a limited accuracy. In this thesis, we recast the generalized version of the Numerov method into the Generalized Matrix Numerov Method based on the algorithm of the existing Basic Matrix Numerov Method. The Generalized Matrix Numerov Method is capable of producing results to any desired accuracy. It is illustrated by finding stationary states with the corresponding energies and simulating the dynamics of Simple Harmonic Oscillator and Coupled Harmonic Oscillators, with the aid of MATHEMATICA. Results of varied accuracy are obtained, and highly accurate results are obtained with short CPU time, thus confirming the validity and effectiveness of this method.

Contents

1	Introduction	2
2	Matrix Numerov Method	4
2.1	Basic Matrix Numerov Method	4
2.2	Generalization	6
2.3	Determination of Grids	9
2.4	Example: Simple Harmonic Oscillator	9
3	Entanglement	13
3.1	Key Concepts	13
3.2	Example: Coupled Harmonic Oscillators	15
4	Conclusions	23
5	Appendix	24
5.1	Mathematica Codes for Finding Elements in the Kinetic Energy Matrix .	24
5.2	Mathematica Codes for Simple Harmonic Oscillator	25
5.3	Mathematica Codes for Coupled Harmonic Oscillators	28

Chapter 1

Introduction

The Numerov method, which was developed by Boris Vasil'evich Numerov, is a well-known numerical method for solving ordinary differential equations of second order that does not contain first order terms. An example of such differential equations is the most fundamental equation in Quantum Mechanics - the 1-D time-independent Schrödinger equation. Depending on the form of the potential, it might be difficult or even impossible to solve the equation analytically. Thus the Numerov method is very useful in providing numerical solutions in such cases.

Modern programming tools such as MATHEMATICA and MATLAB increase the value of numerical methods since the routine computations can be done with less human effort in a relatively shorter time than computation by hand. In 2012, Mohandas Pillai, Joshua Goglio, and Thad G. Walker brought up the idea of Matrix Numerov Method for solving the Schrödinger equation¹. This method further simplifies the calculations by discretizing the wave function on a linear lattice with appropriate boundary conditions, thus allowing the Hamiltonian to be represented by a square matrix whose eigenvalues and eigenvectors can be computed simultaneously.

While Pillai, Goglio, and Walkera placed their emphasis on simplicity and obtained results to a limited accuracy, with a reasonable increase in complexity it is possible to modify this method to attain much higher accuracy and to solve more complicated problems such as the dynamics of entangled quantum systems. Quantum entanglement has extreme importance in the emerging technologies of quantum computing and quantum cryptography, yet the intricacy in analysing the dynamics of entangled systems is even greater than that of simple systems. Therefore it is very helpful if we could solve such problems numerically to give us a general sense of the solution before making any attempt to get to the solution analytically, if it is even possible.

In this thesis, we will first introduce the most basic ideas of the Numerov method which has limited accuracy, and how it is developed into the Basic Matrix Numerov Method.

Next we will proceed to explain the generalized Numerov method, which produces results to any accuracy we choose. The main section of this thesis is to modify the generalized Numerov method into the matrix form in a similar manner as the Basic Matrix Numerov Method, and to apply the Generalized Matrix Numerov Method in solving for simple systems and entangled systems, with the aid of MATHEMATICA.

Chapter 2

Matrix Numerov Method

2.1 Basic Matrix Numerov Method

The Numerov method is a numerical method for solving ordinary differential equations of the form

$$\frac{d^2\psi(x)}{dx^2} = f(x)\psi(x). \quad (2.1)$$

The time-independent 1-D Schrödinger equation

$$-\frac{\hbar^2}{2m} \frac{d^2}{dx^2} \psi(x) + V(x)\psi(x) = E\psi(x) \quad (2.2)$$

can be written into the form of (2.1) as

$$\psi^{(2)}(x) = -\frac{2m}{\hbar^2} [E - V(x)]\psi(x) = f(x)\psi(x) \quad (2.3)$$

where

$$\psi^{(n)}(x) = \frac{d^n}{dx^n} \psi(x), \quad f(x) = -\frac{2m}{\hbar^2} [E - V(x)].$$

Taylor series expansions of the wave function $\psi(x)$ gives

$$\psi(x \pm d) = \psi(x) \pm d\psi^{(1)}(x) + \frac{1}{2!}d^2\psi^{(2)}(x) \pm \frac{1}{3!}d^3\psi^{(3)}(x) + \frac{1}{4!}d^4\psi^{(4)}(x) + \dots \quad (2.4)$$

It follows that

$$\psi(x + d) + \psi(x - d) = 2\psi(x) + d^2\psi^{(2)}(x) + \frac{1}{12}d^4\psi^{(4)}(x) + O(d^6) \quad (2.5)$$

which can be rearranged as

$$\psi^{(2)}(x) = \frac{\psi(x + d) + \psi(x - d) - 2\psi(x)}{d^2} - \frac{1}{12}d^2\psi^{(4)}(x) + O(d^4). \quad (2.6)$$

Substituting (2.6) into (2.3), we have

$$\psi^{(4)}(x) = \frac{f(x+d)\psi(x+d) + f(x-d)\psi(x-d) - 2f(x)\psi(x)}{d^2} + O(d^2). \quad (2.7)$$

Substitution of (2.7) into (2.6) yields an equation to $O(d^4)$

$$f_i\psi_i = \frac{\psi_{i+1} + \psi_{i-1} - 2\psi_i}{d^2} - \frac{1}{12}(f_{i+1}\psi_{i+1} + f_{i-1}\psi_{i-1} - 2f_i\psi_i) \quad (2.8)$$

where

$$\begin{aligned} f_{i-1} &\equiv f(x-d), & f_i &\equiv f(x), & f_{i+1} &\equiv f(x+d), \\ \psi_{i-1} &\equiv \psi(x-d), & \psi_i &\equiv \psi(x), & \psi_{i+1} &\equiv \psi(x+d). \end{aligned}$$

Rearranging (2.8), we have

$$\frac{\psi_{i+1} + \psi_{i-1} - 2\psi_i}{d^2} = \frac{1}{12}(f_{i+1}\psi_{i+1} + f_{i-1}\psi_{i-1} + 10f_i\psi_i). \quad (2.9)$$

Recall that

$$f_{i-1} = -\frac{2m}{\hbar^2}(E - V_{i-1}), \quad f_i = -\frac{2m}{\hbar^2}(E - V_i), \quad f_{i+1} = -\frac{2m}{\hbar^2}(E - V_{i+1}),$$

we have

$$-\frac{\hbar^2}{2m} \frac{\psi_{i-1} - 2\psi_i + \psi_{i+1}}{d^2} + \frac{V_{i-1}\psi_{i-1} + 10V_i\psi_i + V_{i+1}\psi_{i+1}}{12} = E \frac{\psi_{i-1} + 10\psi_i + \psi_{i+1}}{12}. \quad (2.10)$$

It follows that

$$-\frac{\hbar^2}{2m}A\psi + BV\psi = EB\psi \quad (2.11)$$

where

$$A = \frac{1}{d^2} \begin{pmatrix} -2 & 1 & 0 & 0 & 0 & \cdots \\ 1 & -2 & 1 & 0 & 0 & \cdots \\ 0 & 1 & -2 & 1 & 0 & \cdots \\ 0 & 0 & 1 & -2 & 1 & \cdots \\ 0 & 0 & 0 & 1 & -2 & \ddots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ddots \end{pmatrix}, \quad B = \frac{1}{12} \begin{pmatrix} 10 & 1 & 0 & 0 & 0 & \cdots \\ 1 & 10 & 1 & 0 & 0 & \cdots \\ 0 & 1 & 10 & 1 & 0 & \cdots \\ 0 & 0 & 1 & 10 & 1 & \cdots \\ 0 & 0 & 0 & 1 & 10 & \ddots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ddots \end{pmatrix},$$

$$V = \begin{pmatrix} V_1 & 0 & 0 & 0 & 0 & \cdots \\ 0 & V_2 & 0 & 0 & 0 & \cdots \\ 0 & 0 & V_3 & 0 & 0 & \cdots \\ 0 & 0 & 0 & V_4 & 0 & \cdots \\ 0 & 0 & 0 & 0 & V_5 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ddots \end{pmatrix}, \quad \psi = \begin{pmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \\ \psi_4 \\ \psi_5 \\ \vdots \end{pmatrix}.$$

Multiplying both sides of (2.11) by B^{-1} , we get

$$-\frac{\hbar^2}{2m}B^{-1}A\psi + V\psi = E\psi \quad (2.12)$$

which is in the form of

$$H\psi = E\psi, \quad H = -\frac{\hbar^2}{2m}B^{-1}A + V \quad (2.13)$$

where the square matrix H is the Hamiltonian of the system, i.e. the sum of the kinetic energy matrix $-\frac{\hbar^2}{2m}B^{-1}A$ and the potential energy matrix V . The eigenvectors of H are the stationary states of the time-independent Schrödinger equation, and the eigenvalues are the corresponding energies of the stationary states.

2.2 Generalization

Again we consider Taylor Series expansion of the wave function $\psi(x)$ in (2.4), it follows that

$$\psi(x+d) + \psi(x-d) - 2\psi(x) = 2\frac{d^2}{2!}\psi^{(2)}(x) + 2\frac{d^4}{4!}\psi^{(4)}(x) + \cdots + 2\frac{d^{2r}}{2r!}\psi^{(2r)}(x) + O(d^{2r+2}) \quad (2.14)$$

and

$$\psi^{(2)}(x) = \frac{\psi(x+d) + \psi(x-d) - 2\psi(x)}{d^2} - \cdots - \frac{2d^{2r}}{(2r+2)!}\psi^{(2r+2)}(x) + O(d^{2r+2}). \quad (2.15)$$

Similarly,

$$\begin{aligned}
& \psi(x+d) + \psi(x-d) - 2\psi(x) \\
&= 2\frac{d^2}{2!}\psi^{(2)}(x) + 2\frac{d^4}{4!}\psi^{(4)}(x) + \dots + 2\frac{d^{2r}}{2r!}\psi^{(2r)}(x) + O(d^{2r+2}) \\
& \psi(x+2d) + \psi(x-2d) - 2\psi(x) \\
&= 2\frac{(2d)^2}{2!}\psi^{(2)}(x) + 2\frac{(2d)^4}{4!}\psi^{(4)}(x) + \dots + 2\frac{(2d)^{2r}}{2r!}\psi^{(2r)}(x) + O((2d)^{2r+2}) \\
& \dots \\
& \psi(x+rd) + \psi(x-rd) - 2\psi(x) \\
&= 2\frac{(rd)^2}{2!}\psi^{(2)}(x) + 2\frac{(rd)^4}{4!}\psi^{(4)}(x) + \dots + 2\frac{(rd)^{2r}}{2r!}\psi^{(2r)}(x) + O((rd)^{2r+2}).
\end{aligned}$$

We thus have a set of simultaneous equations with r equations and r unknowns

$$\begin{aligned}
& \frac{d^2}{2!}\psi^{(2)}(x) + \frac{d^4}{4!}\psi^{(4)}(x) + \dots + \frac{d^{2r}}{2r!}\psi^{(2r)}(x) + O(d^{2r+2}) \\
&= \frac{\psi(x+d) + \psi(x-d) - 2\psi(x)}{2} \\
& \frac{(2d)^2}{2!}\psi^{(2)}(x) + \frac{(2d)^4}{4!}\psi^{(4)}(x) + \dots + \frac{(2d)^{2r}}{2r!}\psi^{(2r)}(x) + O((2d)^{2r+2}) \\
&= \frac{\psi(x+2d) + \psi(x-2d) - 2\psi(x)}{2} \\
& \dots \\
& \frac{(rd)^2}{2!}\psi^{(2)}(x) + \frac{(rd)^4}{4!}\psi^{(4)}(x) + \dots + \frac{(rd)^{2r}}{2r!}\psi^{(2r)}(x) + O((rd)^{2r+2}) \\
&= \frac{\psi(x+rd) + \psi(x-rd) - 2\psi(x)}{2},
\end{aligned}$$

which can be solved to obtain $\psi^{(2)}(x)$ and $\psi^{(2r)}(x)$. Suppose that

$$\psi^{(2)}(x) = \frac{1}{d^2} \sum_{i=-r}^r c_i \psi_i - \frac{2d^{2r}}{(2r+2)!} \psi^{(2r+2)}(x) + O(d^{2r+2}) \quad (2.16)$$

and

$$\psi^{(2r)}(x) = \frac{1}{d^{2r}} \sum_{i=-r}^r k_i \psi_i. \quad (2.17)$$

Substituting (2.15) into (2.3), we get

$$\psi^{(2r+2)}(x) = \frac{1}{d^{2r}} \sum_{i=-r}^r k_i f_i \psi_i. \quad (2.18)$$

Substituting (2.16) into (2.14), we get

$$f_i \psi_i = \frac{1}{d^2} \sum_{i=-r}^r c_i \psi_i - \sum_{i=-r}^r \frac{2}{(2r+2)!} k_i f_i \psi_i. \quad (2.19)$$

Therefore, to $O(d^{2r+2})$,

$$\frac{1}{d^2} \sum_{i=-r}^r c_i \psi_i = \sum_{i=-r}^r \frac{2}{(2r+2)!} k_i f_i \psi_i + f_i \psi_i. \quad (2.20)$$

(2.18) is in a similar form to (2.9), and can be expanded into the form of (2.10). The non-zero elements of matrix A are given by the coefficients of ψ_i , while that of matrix B are given by the coefficients of $f_i \psi_i$. With this, we can use the Matrix Numerov Method to an error of any order $2r+2$. For example, to $O(d^8)$, the matrices A and B are given by

$$A = \frac{1}{180d^2} \begin{pmatrix} -490 & 270 & -27 & 2 & 0 & 0 & 0 & 0 & \dots \\ 270 & -490 & 270 & -27 & 2 & 0 & 0 & 0 & \dots \\ -27 & 270 & -490 & 270 & -27 & 2 & 0 & 0 & \dots \\ 2 & -27 & 270 & -490 & 270 & -27 & 2 & 0 & \dots \\ 0 & 2 & -27 & 270 & -490 & 270 & -27 & 2 & \dots \\ 0 & 0 & 2 & -27 & 270 & -490 & 270 & -27 & \ddots \\ 0 & 0 & 0 & 2 & -27 & 270 & -490 & 270 & \ddots \\ 0 & 0 & 0 & 0 & 2 & -27 & 270 & -490 & \ddots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots \end{pmatrix},$$

$$B = \frac{1}{20160} \begin{pmatrix} 20140 & 150 & -6 & 1 & 0 & 0 & 0 & 0 & \dots \\ 150 & 20140 & 150 & -6 & 1 & 0 & 0 & 0 & \dots \\ -6 & 150 & 20140 & 150 & -6 & 1 & 0 & 0 & \dots \\ 1 & -6 & 150 & 20140 & 150 & -6 & 1 & 0 & \dots \\ 0 & 1 & -6 & 150 & 20140 & 150 & -6 & 1 & \dots \\ 0 & 0 & 1 & -6 & 150 & 20140 & 150 & -6 & \ddots \\ 0 & 0 & 0 & 1 & -6 & 150 & 20140 & 150 & \ddots \\ 0 & 0 & 0 & 0 & 1 & -6 & 150 & 20140 & \ddots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots \end{pmatrix}.$$

The Basic Matrix Numerov Method in Section 2.1 is a special case with $r=1$ to an

error $O(d^4)$. While the generalized Numerov method is not a new idea, in this paper we will apply it in the matrix form as Generalized Matrix Numerov Method so as to solve the Schrödinger equation to any desired accuracy.

2.3 Determination of Grids

Suppose that the dimensions of the square matrices A and B are both $N \times N$. The dimension of the eigenvector ψ is N as well; effectively we have thus implemented a boundary condition $\psi_0 = \psi_{N+1} = 0$, which is equivalent to placing the potential of interest inside an infinite-walled box.

The value of N is related to the maximum energy E_m of all stationary states we wish to find. It is possible to fix E_m and find N , or vice versa.

First we fix the maximum energy E_m . The minimum local de Broglie wavelength is $\lambda = h/\sqrt{2mE_m}$. Sufficient accuracy is generally obtained by taking the grid spacing d corresponding to about one point per radian, i.e., $d = \lambda/2\pi = \hbar/\sqrt{2mE_m}$. The number of grid points needed can be estimated by finding the outer turning points x_t such that $V(x_t) = E_m$, and allowing for an extra 2λ in the classically forbidden region. Thus, $N = 2(x_t/d + 4\pi)$ rounded to the nearest integer.

On the other hand, if we first have a fixed N value, we may express x_t as $x_t = (N/2 - 4\pi)d = (N/2 - 4\pi)\hbar/\sqrt{2mE_m}$. By substituting x_t into $V(x_t) = E_m$, we can solve for the maximum potential energy E_m .

2.4 Example: Simple Harmonic Oscillator

Matrix Numerov Method can be used to solve the time-independent Schrödinger equation even if the potential V is in a form that is not analytically solvable. However, in this section, we will apply the Generalized Matrix Numerov Method to Simple Harmonic Oscillator with potential $V(x) = \frac{1}{2}\omega^2x$, which is analytically solvable and thus useful for comparison to the numerical results.

By using scaled variables $s \equiv \sqrt{\frac{m\omega}{\hbar}}x$ and $\epsilon = \frac{E}{\hbar\omega}$, we have

$$-\frac{1}{2}\frac{d^2}{dx^2}\psi + \frac{1}{2}s^2\psi = \epsilon\psi \quad (2.21)$$

with stationary states $\psi_n(s) = \frac{\pi^{1/4}}{\sqrt{2^n n!}}H_n(s)e^{-s^2/2}$ and energies $\epsilon = n + \frac{1}{2}$.

(2.21) can be written into the matrix form

$$-\frac{1}{2}B^{-1}A\psi + \frac{1}{2}s^2\psi = \epsilon\psi. \quad (2.22)$$

Setting $\epsilon_m = 50$, we have $d_s = \frac{1}{\sqrt{2\epsilon_m}} = 0.1$ and $N = 2(4\pi + \frac{s_t}{d_s}) \approx 225$.

A comparison of the exact and numerical results for selected energy levels is shown in Table 2.1.

n	0	1	2	3	10	20	35	49
Exact	0.5	1.5	2.5	3.5	10.5	20.5	35.5	49.5
Numerov $O(d_s^4)$	0.5000	1.5000	2.5000	3.5000	10.499	20.495	35.476	49.434
Numerov $O(d_s^8)$	0.5000	1.5000	2.5000	3.5000	10.500	20.499	35.495	49.480

Table 2.1: Comparison of exact and numerical results for selected eigenenergies of the SHO.

Table 2.2 gives an overall comparison of the exact and numerical results for all the eigenenergies within the effective range $\epsilon < \epsilon_m = 50$. This is achieved by doing a linear regression of the numerical results versus the exact results, and a perfect match is represented by the equation $y = x$, with correlation coefficient $r_c = 1$.

It is shown that, numerical results with higher accuracy can be produced at the cost of longer CPU time. Nevertheless, the time taken to obtain sensible results is generally shorter than solving analytically.

	Equation	r_c	CPU time*
Exact Match	$y = x$	1	-
Numerov $O(d_s^4)$	$y = 0.9988x + 0.0135$	0.999999700	1.78 seconds
Numerov $O(d_s^8)$	$y = 0.9997x + 0.0042$	0.999999964	11.95 seconds

Table 2.2: Overall comparison of exact and numerical results for eigenenergies of the SHO.

For $O(d_s^4)$, the comparison between the analytical solution and the Numerov results for the $n = 49$ state of the SHO, with $\epsilon = 49.5$ i.e. the highest energy state below the maximum value $\epsilon_m = 50$, is shown in Figure 2.1.

Given the solutions $\psi_n(x)$ of the time-independent Schrödinger equation, the solution $\Psi(x, t)$ of the time-dependent Schrödinger equation can be expressed as

$$\Psi(x, t) = \sum_{n=0}^{\infty} c_n \psi_n(x) e^{-iE_n t/\hbar}. \quad (2.23)$$

The coefficients c_n can be found, if the initial wave function $\Psi(x, 0)$ is known, by

$$c_n = \int dx \psi_n^*(x) \Psi(x, 0). \quad (2.24)$$

*Varies with computer configuration.

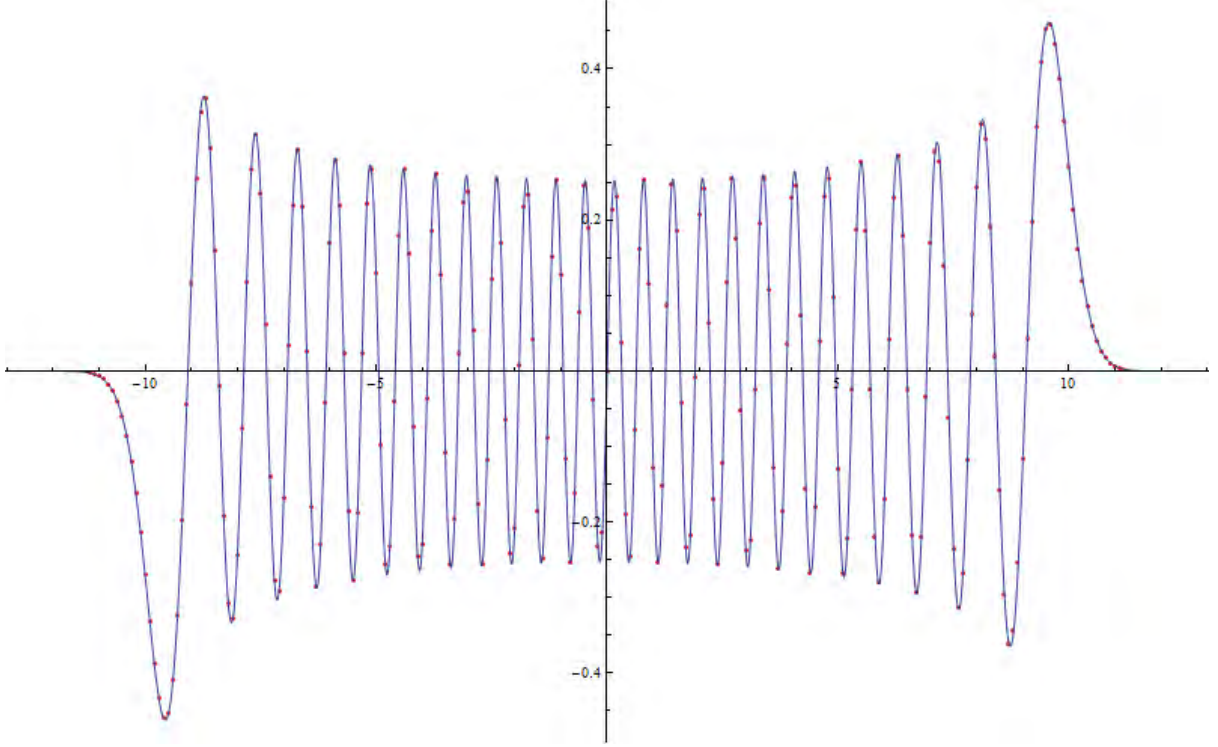


Figure 2.1: The blue solid line shows the analytical solution to the Schrödinger equation for the $n = 49$ state of the SHO, while the numerical results are shown by the red dots, for $O(d_s^4)$.

After substituting $s \equiv \sqrt{\frac{m\omega}{\hbar}}$ and $\epsilon = \frac{E}{\hbar\omega}$, we have

$$\Psi(s, t) = \sum_{n=0}^{\infty} c_n \psi_n(s) e^{-i\epsilon_n t}, \quad c_n = \int ds \psi_n^*(s) \Psi(s, 0). \quad (2.25)$$

Here we use the Gaussian wave packet $\Psi(s, 0) = (\frac{1}{4\pi})^{1/4} e^{-s^2/8}$ as the initial wave function.

Figure 2.2 shows the graphs of the real part $\Psi(s, 600)$, i.e. solution to the time-dependent Schrödinger equation at $t = 600$, for $O(d_s^4)$ and $O(d_s^8)$ respectively. By comparing the Numerov results with the analytical solution, it can be seen that the accuracy of the numerical results is significantly improved by increasing the value of r in $O(d^{2r+2})$.

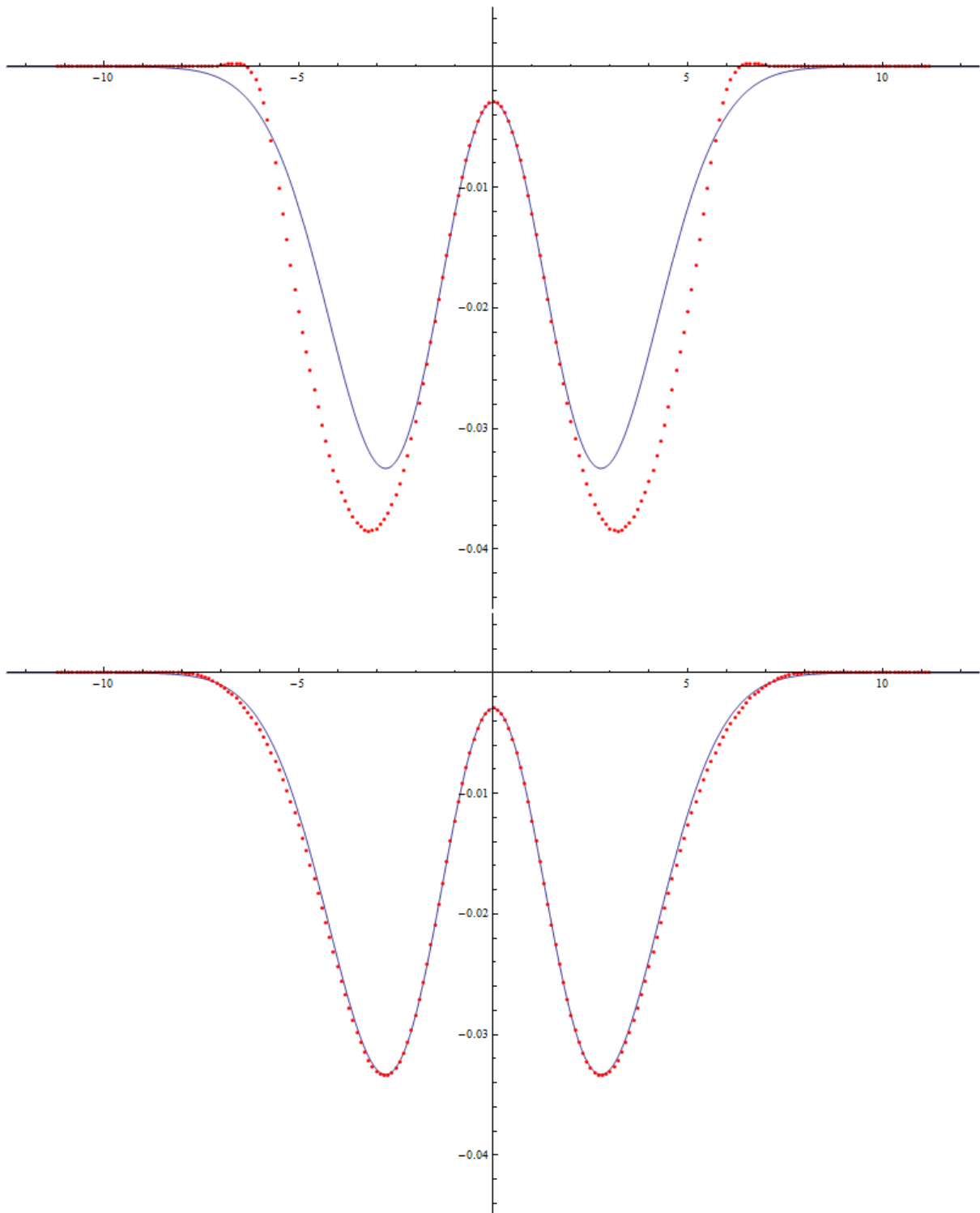


Figure 2.2: The upper graph shows the real part of $\Psi(s, 600)$ for $O(d_s^4)$; the lower graph shows the real part of $\Psi(s, 600)$ for $O(d_s^8)$. The analytical solution is represented by the blue solid line and the Numerov results are represented by the red dots.

Chapter 3

Entanglement

3.1 Key Concepts

Quantum entanglement is a physical phenomenon that occurs when pairs or groups of particles are generated or interact in ways such that the quantum state of each particle cannot be described independently.

Consider two noninteracting systems A and B , in Hilbert spaces H_A and H_B respectively. The Hilbert space of the composite system is the tensor product

$$H = H_A \otimes H_B. \quad (3.1)$$

If A is in state $|\psi_A\rangle$ and B is in state $|\phi_B\rangle$, the state of the composite system is

$$|\Pi_{AB}\rangle = |\psi_A\rangle \otimes |\phi_B\rangle. \quad (3.2)$$

A states of the composite system which can be represented in this form is called a *product state*. However, not all states in H are product states.

Let $\{|i_A\rangle\}$ be the basis vectors of H_A , and $\{|j_B\rangle\}$ be the basis vectors of H_B . Then the most general state in H is of the form

$$|\Psi_{AB}\rangle = \sum_{i,j} c_{ij} |i_A\rangle \otimes |j_B\rangle. \quad (3.3)$$

This state is a product state if there exist c_i^A and c_j^B such that $c_{ij} = c_i^A c_j^B$. On the other hand, if $c_{ij} \neq c_i^A c_j^B$ for all c_i^A and c_j^B , this state is called an *entangled state*.

The *density operator* $\hat{\rho}$ for the composite system is given by

$$\hat{\rho} = |\Psi_{AB}\rangle \langle \Psi_{AB}|. \quad (3.4)$$

and the elements of the density matrix is given by

$$\rho_{mn} = \langle m_{AB} | \Psi_{AB} \rangle \langle \Psi_{AB} | n_{AB} \rangle \quad (3.5)$$

where $\{|m_{AB}\rangle\}$ is a set of basis vectors of the composite system H .

The *reduced density operator* of system A is by definition²

$$\hat{\rho}_A = \text{Tr}_B(\hat{\rho}) \quad (3.6)$$

where the partial trace over B is defined by

$$\text{Tr}_B(|i_A\rangle\langle k_A| \otimes |j_B\rangle\langle l_B|) = |i_A\rangle\langle k_A| \text{Tr}(|j_B\rangle\langle l_B|). \quad (3.7)$$

Similarly,

$$\hat{\rho}_B = \text{Tr}_A(\hat{\rho}), \quad (3.8)$$

$$\text{Tr}_A(|i_A\rangle\langle k_A| \otimes |j_B\rangle\langle l_B|) = \text{Tr}(|i_A\rangle\langle k_A|) |j_B\rangle\langle l_B|. \quad (3.9)$$

The reduced density operator describes all the properties or outcomes of measurements of one system, given that the other system is left unobserved. To determine whether a state is a product state or entangled state, we may find the purity

$$\gamma \equiv \text{Tr}(\rho_A^2) = \text{Tr}(\rho_B^2). \quad (3.10)$$

The condition $\gamma = 1$ corresponds to a product state whereas $0 < \rho < 1$ corresponds to an entangled state. An entangled state means that we cannot distinguish, or "trace out" one system from the other.

For example, if both A and B are two-level systems, we may take

$$\{|i_A\rangle\} = \{|j_B\rangle\} = \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}, \quad \{|m_{AB}\rangle\} = \left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \right\}.$$

For a state $|\Psi_1\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ -1 \\ 0 \end{pmatrix}$ in H , we have

$$\rho = |\Psi_1\rangle\langle\Psi_1| = \frac{1}{2} \begin{pmatrix} 1 \\ 0 \\ -1 \\ 0 \end{pmatrix} \begin{pmatrix} 1^* & 0^* & -1^* & 0^* \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix};$$

$$\rho_A = \frac{1}{2} \begin{pmatrix} \text{Tr} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} & \text{Tr} \begin{pmatrix} -1 & 0 \\ 0 & 0 \end{pmatrix} \\ \text{Tr} \begin{pmatrix} -1 & 0 \\ 0 & 0 \end{pmatrix} & \text{Tr} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}, \quad \rho_A^2 = \frac{1}{2} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} = \rho_A;$$

$$\rho_B = \frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + \frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad \rho_B^2 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} = \rho_B.$$

Since $\gamma \equiv \text{Tr}(\rho_A^2) = \text{Tr}(\rho_B^2) = 1$, we conclude that $|\Psi_1\rangle$ is a product state. In fact, it can be expressed as $|\Psi_1\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 0 \\ -1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix}$.

For another state $|\Psi_2\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 1 \\ -1 \\ 0 \end{pmatrix}$, we have

$$\rho = |\Psi_2\rangle\langle\Psi_2| = \frac{1}{2} \begin{pmatrix} 0 \\ 1 \\ -1 \\ 0 \end{pmatrix} \begin{pmatrix} 0^* & 1^* & -1^* & 0^* \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix};$$

$$\rho_A = \frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \rho_A^2 = \frac{1}{4} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \neq \rho_A;$$

$$\rho_B = \frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \rho_B^2 = \frac{1}{4} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \neq \rho_B.$$

Since $\gamma \equiv \text{Tr}(\rho_A^2) = \text{Tr}(\rho_B^2) = 0.5 < 1$, $|\Psi_2\rangle$ is an entangled state.

3.2 Example: Coupled Harmonic Oscillators

The Hamiltonian of Coupled Harmonic Oscillators is represented by

$$H = -\frac{\hbar^2}{2m_1} \frac{\partial}{\partial x_1^2} + \frac{1}{2} m_1 \omega_1^2 x_1^2 - \frac{\hbar^2}{2m_2} \frac{\partial}{\partial x_2^2} + \frac{1}{2} m_2 \omega_2^2 x_2^2 + g_x x_1 x_2 \quad (3.11)$$

where g_x is the coupling constant. By using scaled variables

$$s_1 \equiv \sqrt{\frac{m}{\hbar}} \mu x_1, \quad s_2 \equiv \sqrt{\frac{m}{\hbar}} \mu^{-1} x_2, \quad g \equiv \frac{1}{\hbar^2} g_x$$

with $\mu = (m_1/m_2)^{1/4}$ and $m = (m_1 m_2)^{1/2}$, we have

$$H = -\frac{1}{2} \frac{\partial}{\partial s_1^2} + \frac{1}{2} \omega_1^2 s_1^2 - \frac{1}{2} \frac{\partial}{\partial s_2^2} + \frac{1}{2} \omega_2^2 s_2^2 + g s_1 s_2. \quad (3.12)$$

The ground state wave function of this Hamiltonian is given by³

$$\psi_0(\vec{y}) \equiv \langle y_1, y_2 | 0, 0 \rangle = \sqrt{\frac{\omega}{\pi}} \exp \left\{ -\frac{\omega}{2} (e^\eta y_1^2 + e^{-\eta} y_2^2) \right\} \quad (3.13)$$

where

$$y_1 \equiv s_1 \cos \frac{\theta}{2} - s_2 \sin \frac{\theta}{2}, \quad y_2 \equiv s_1 \sin \frac{\theta}{2} + s_2 \cos \frac{\theta}{2}, \quad (3.14)$$

$$\tan \theta = \frac{2g}{\omega_2^2 - \omega_1^2}, \quad e^{2\eta} = \frac{\omega_1^2 + \omega_2^2 + \sqrt{(\omega_1^2 - \omega_2^2)^2 + 4g^2}}{2\omega^2}, \quad \omega = (\omega_1^2 \omega_2^2 - g^2)^{1/4}$$

provided that the condition $\omega_1^2 \omega_2^2 > g^2$ must be fulfilled.

Other states can then be calculated by

$$|n_1, n_2\rangle = \frac{(a_1^\dagger)^{n_1} (a_2^\dagger)^{n_2}}{\sqrt{n_1! n_2!}} |0, 0\rangle \quad (3.15)$$

where a_1^\dagger and a_2^\dagger are the creation operators

$$a_1^\dagger = \sqrt{\frac{\omega}{2}} e^{\eta/2} y_1 - \frac{1}{\sqrt{2\omega}} e^{-\eta/2} \frac{\partial}{\partial y_1}, \quad a_2^\dagger = \sqrt{\frac{\omega}{2}} e^{-\eta/2} y_2 - \frac{1}{\sqrt{2\omega}} e^{\eta/2} \frac{\partial}{\partial y_2}.$$

The corresponding eigenenergies are

$$\epsilon_{n_1, n_2} = \omega (e^\eta n_1 + e^{-\eta} n_2 + \cosh \eta). \quad (3.16)$$

To obtain expressions of the wavefunctions in terms of (s_1, s_2) , one can simply substitute (3.14) back into the expression. For example, the ground state wavefunction (3.13) can be written as

$$\begin{aligned} \psi_0(\vec{s}) &\equiv \langle s_1, s_2 | 0, 0 \rangle \\ &= \sqrt{\frac{\omega}{\pi}} \exp \left\{ -\frac{\omega}{2} \left[e^\eta \left(s_1 \cos \frac{\theta}{2} - s_2 \sin \frac{\theta}{2} \right)^2 + e^{-\eta} \left(s_1 \sin \frac{\theta}{2} + s_2 \cos \frac{\theta}{2} \right)^2 \right] \right\}. \end{aligned} \quad (3.17)$$

In terms of the Matrix Numerov Method, the Hamiltonian (3.5) is represented by

$$H = \left(-\frac{1}{2} B_1^{-1} A_1 + V_1 \right) \otimes I_1 + I_2 \otimes \left(-\frac{1}{2} B_2^{-1} A_2 + V_2 \right) + g S_1 \otimes S_2, \quad (3.18)$$

where

$$S = \text{diag}(\cdots, s_{i-1}, s_i, s_{i+1}, \cdots).$$

Here we follow the same standard to determine the grids for individual systems. This time $N = 50$ is first fixed, with $\omega_1 = 1.5$, $\omega_2 = 1.0$ and $g = 0.2$, we then have $d_1 \approx 0.23$,

$d_2 \approx 0.28$, $\epsilon_{m1} \approx 9.33$ and $\epsilon_{m2} \approx 6.22$. The maximum energy of the composite system is taken to be the sum of those of the individual systems, i.e. $\epsilon_m = \epsilon_{m1} + \epsilon_{m2} = 15.54$.

A comparison of the exact and numerical results for selected energy levels is shown in Table 3.1.

n_1, n_2	0,0	0,1	1,0	1,1	1,2	3,2	3,5	6,5
Exact	1.2473	2.2316	2.7577	3.7420	4.7262	7.7470	10.700	15.231
Numerov $O(d_s^4)$	1.2473	2.2314	2.7574	3.7415	4.7253	7.7436	10.690	15.207
Numerov $O(d_s^8)$	1.2473	2.2316	2.7577	3.7419	4.7261	7.7462	10.697	15.223
Numerov $O(d_s^{12})$	1.2473	2.2316	2.7577	3.7420	4.7262	7.7470	10.700	15.231

Table 3.1: Comparison of exact and numerical results for selected eigenenergies of the coupled oscillators.

Table 3.2 gives an overall comparison of the exact and numerical results for all the eigenenergies within the effective range $\epsilon < \epsilon_m = 15.54$.

	Equation	r_c	CPU time
Exact Match	$y = x$	1	-
Numerov $O(d_s^4)$	$y = 0.9963x + 0.0192$	0.999985	19.27 seconds
Numerov $O(d_s^8)$	$y = 0.9983x + 0.0097$	0.999995	79.76 seconds
Numerov $O(d_s^{12})$	$y = 0.9999x + 0.0008$	1.000000	96.33 seconds

Table 3.2: Overall comparison of exact and numerical results for eigenenergies of the coupled oscillators.

Figure 3.1 shows a 3-D plot of the state $(n_1, n_2) = (6, 5)$, with $\epsilon = 15.23$ which is close to the maximum effective value $\epsilon_m = 15.54$.

The solution $\Psi(s_1, s_2, t)$ of the time-dependent Schrödinger equation can be expressed as

$$\Psi(s_1, s_2, t) = \sum_{n_1=0}^{\infty} \sum_{n_2=0}^{\infty} c_{n_1, n_2} \psi_{n_1, n_2}(s_1, s_2) e^{-i\epsilon_{n_1, n_2} t}. \quad (3.19)$$

The coefficients c_{n_1, n_2} can be found, if the initial wave function $\Psi(s_1, s_2, 0)$ is known, by

$$c_{n_1, n_2} = \int \int ds_1 ds_2 \psi_{n_1, n_2}^*(s_1, s_2) \Psi(s_1, s_2, 0). \quad (3.20)$$

Here we use the Gaussian wave packet $\Psi(s_1, s_2, 0) = (\frac{1}{2\pi})^{1/4} e^{-(s_1-1)^2/4} (\frac{1}{2\pi})^{1/4} e^{-(s_2+2)^2/4}$ as the initial wave function.

Figure 3.2 gives a sketch of the real part of the wave function at $t = 150$, for $O(d_s^4)$, $O(d_s^8)$ and $O(d_s^{12})$ respectively. It can be observed that the Basic Matrix Numerov Method with an error $O(d_s^4)$ does not have sufficient accuracy to produce sensible results.

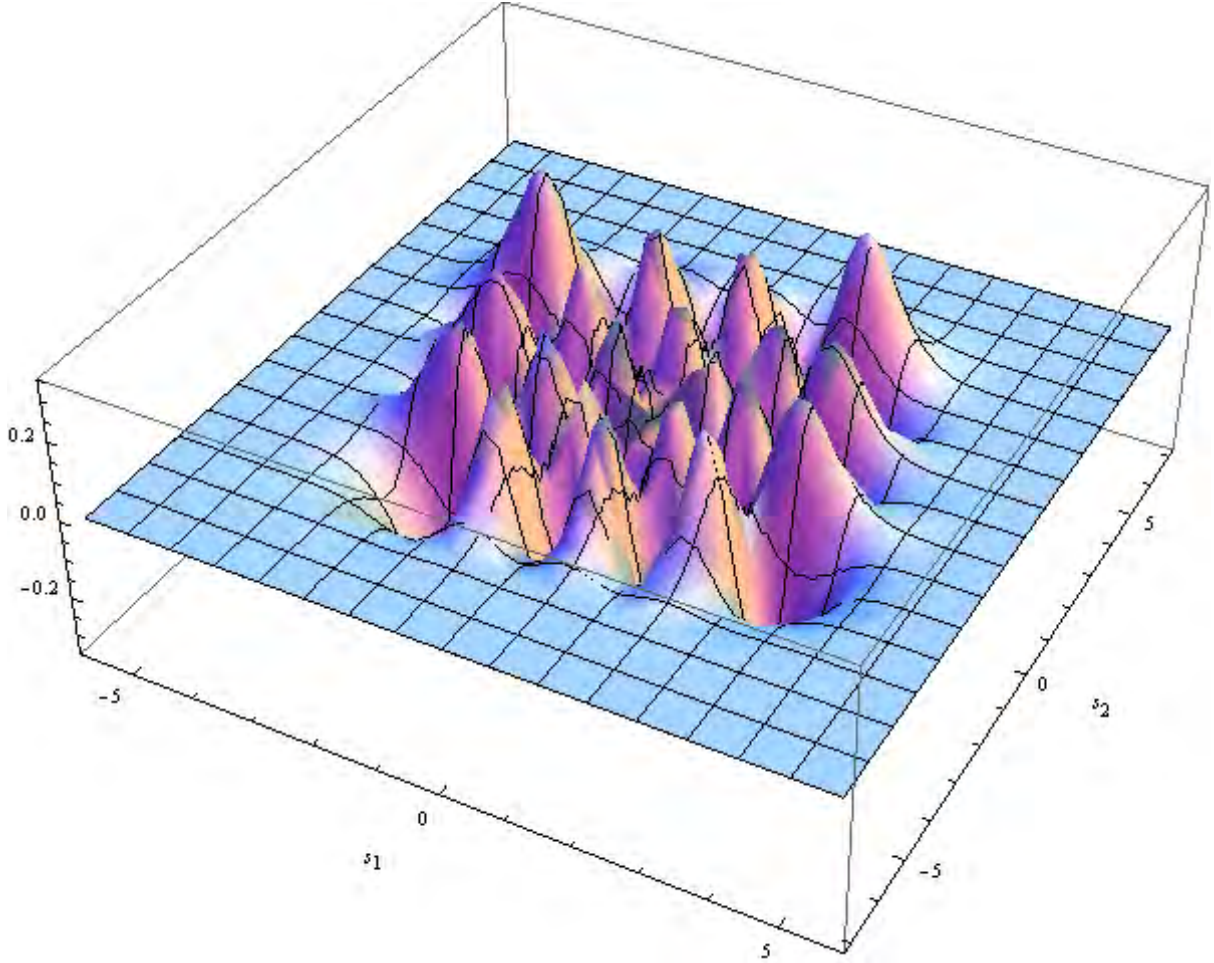


Figure 3.1: 3-D plot of the state $(n_1, n_2) = (6, 5)$ to $O(d_s^8)$. The curved surface represents the analytical solution and the black lines represent the numerical results.

By increasing r in the Generalized Matrix Numerov Method, it is possible to obtain results that are very close to the analytical solutions.

Nevertheless, in order to achieve satisfactory results, it is insufficient to increase r solely. Again we set $N = 50$, $\omega_1 = 1.5$, $\omega_2 = 1.0$, and $\Psi(s_1, s_2, 0) = (\frac{1}{2\pi})^{1/4}e^{-(s_1-1)^2/4}(\frac{1}{2\pi})^{1/4}e^{-(s_2+2)^2/4}$ as the initial wave function; but this time with coupling constant $g = 0$. The single qubit states are expected to be pure, i.e. $\gamma \equiv \text{Tr}(\rho^2) = 1$. Yet in Table 3.3 it is shown that $\gamma < 1$, and an increase in r has little effect on improving its accuracy.

However, if we modify the initial wave function either by reducing its width or shifting its centre towards the origin, the accuracy of γ improves significantly as shown in Table 3.4. The reason is that when the modified initial wave functions (Ψ_2 and Ψ_3) are decomposed into a sum of the stationary states, the states of lower energies will have larger coefficients and states of higher energies will have smaller coefficients compared to those in the original initial wave function (Ψ_1). Since in Matrix Numerov Method we

	$t = 0$	150	300	500	1000
Exact	1	1	1	1	1
Numerov $O(d_s^4)$	0.96482	0.96482	0.96482	0.96482	0.96482
Numerov $O(d_s^8)$	0.96505	0.96505	0.96505	0.96505	0.96505
Numerov $O(d_s^{12})$	0.96520	0.96520	0.96520	0.96520	0.96520

Table 3.3: Purity γ for $g = 0$, with $N = 50$, $\omega_1 = 1.5$, $\omega_2 = 1.0$, and $\Psi(s_1, s_2, 0) = (\frac{1}{2\pi})^{1/4} e^{-(s_1-1)^2/4} (\frac{1}{2\pi})^{1/4} e^{-(s_2+2)^2/4}$.

only keep stationary states with energy $\epsilon < \epsilon_m$, if the coefficients of the stationary states with $\epsilon > \epsilon_m$ are larger, the error caused by excluding these states will be larger, and vice versa.

	$t = 0$	150	300	500	1000
Exact	1	1	1	1	1
Numerov $\Psi(s_1, s_2, 0) = \Psi_1$	0.96520	0.96520	0.96520	0.96520	0.96520
Numerov $\Psi(s_1, s_2, 0) = \Psi_2$	0.99999	0.99999	0.99999	0.99999	0.99999
Numerov $\Psi(s_1, s_2, 0) = \Psi_3$	0.99943	0.99943	0.99943	0.99943	0.99943

$$\Psi_1 = (\frac{1}{2\pi})^{1/4} e^{-(s_1-1)^2/4} (\frac{1}{2\pi})^{1/4} e^{-(s_2+2)^2/4},$$

$$\Psi_2 = (\frac{2}{\pi})^{1/4} e^{-(s_1-1)^2} (\frac{2}{\pi})^{1/4} e^{-(s_2+2)^2}, \quad \Psi_3 = (\frac{1}{2\pi})^{1/4} e^{-s_1^2/4} (\frac{1}{2\pi})^{1/4} e^{-s_2^2/4}.$$

Table 3.4: Purity γ for $g = 0$, with $N = 50$, $\omega_1 = 1.5$, $\omega_2 = 1.0$ for $O(d_s^{12})$.

On the other hand, the accuracy of γ can be increased by increasing N as well. An increase in N corresponds to an increase in ϵ_m , hence more stationary states with higher energies are included in our results. There are fewer stationary states with $\epsilon > \epsilon_m$, thus smaller error for truncating them off, as observed in Table 3.5.

	$t = 0$	150	300	500	1000
Exact	1	1	1	1	1
Numerov $N = 50$ ($\epsilon_m = 15.54$)	0.96520	0.96520	0.96520	0.96520	0.96520
Numerov $N = 60$ ($\epsilon_m = 21.79$)	0.99659	0.99659	0.99659	0.99659	0.99659
Numerov $N = 70$ ($\epsilon_m = 28.04$)	0.99964	0.99964	0.99964	0.99964	0.99964

Table 3.5: Purity γ for $g = 0$, with $\omega_1 = 1.5$, $\omega_2 = 1.0$ and $\Psi(s_1, s_2, 0) = (\frac{1}{2\pi})^{1/4} e^{-(s_1-1)^2/4} (\frac{1}{2\pi})^{1/4} e^{-(s_2+2)^2/4}$, for $O(d_s^{12})$.

For a fixed initial wave function, the accuracy of results is determined by both r and N (ϵ_m). An increase in r , which is possible only in the Generalized Matrix Numerov Method

but not in the Basic Matrix Numerov Method, corresponds to an increase of non-zero elements in the kinetic energy matrix and makes each stationary state within the effective energy range closer to the analytical solution; an increase in N (ϵ_m) corresponds to an increase in dimension of the kinetic energy matrix, extending the effective energy range thus reducing the number of stationary states being truncated off.

To find the dynamics of coupled oscillators with a given initial wave function, we can choose r and N such that when $g = 0$, we have $\gamma \approx 1$ to our desired accuracy. For example, if we were to solve a problem with

$$\omega_1 = 1.5, \quad \omega_2 = 1.2, \quad g = 0.5,$$

$$\Psi(s_1, s_2, 0) = \left(\frac{1}{2\pi}\right)^{1/4} e^{-s_1^2/4} \left(\frac{1}{\pi}\right)^{1/4} e^{-(s_2+1)^2/2},$$

we first set $g = 0$ and keep all other parameters unchanged. It is found that with $r = 5$ and $N = 60$, $\gamma_0 = 0.999976$, approximated to 1 with an error of order 10^{-5} . We may then proceed with $g = 0.5$; by comparing the value of γ at different time t with γ_0 , we have a rough estimate of when the system is entangled and relatively how much it is entangled.

Figure 3.3 is an illustration for the purity change of the coupled oscillators. The blue horizontal line indicates the γ_0 value for the $g = 0$ case and the red dots represents the γ values for the $g = 0.5$ case, with time t varying from 0 to 100. Dots which lie well below γ_0 imply that the coupled oscillators are entangled at those points of time while dots close to γ_0 show that they are nearly non-entangled. The red broken line is obtained by joining the dots, providing a clearer view of how γ varies with t .

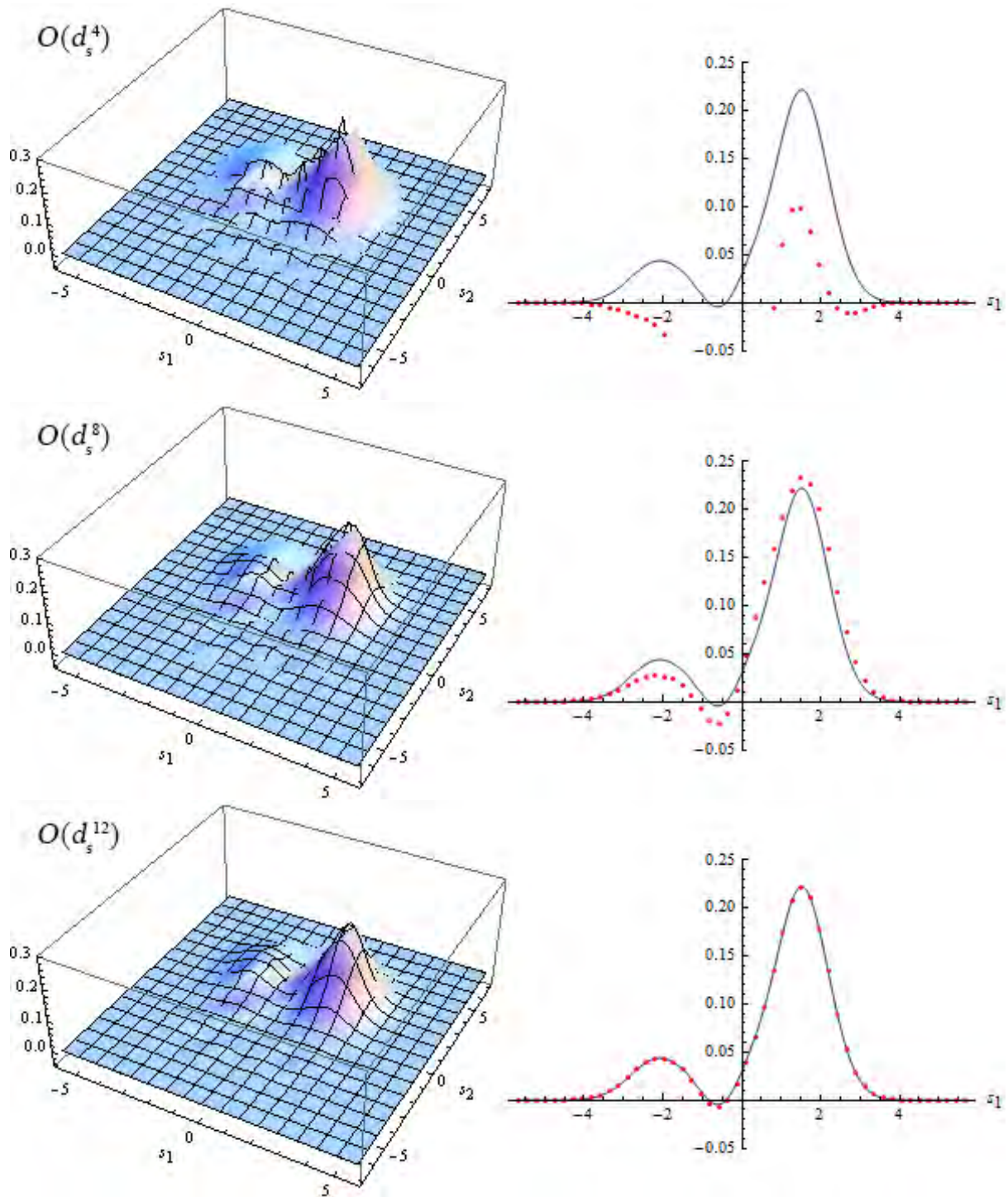


Figure 3.2: Graphs of the real part of the wave function at $t = 150$, for $O(d_s^4)$, $O(d_s^8)$ and $O(d_s^{12})$ respectively. The left column shows 3D plots of the wave function; the curved surface represents the analytical solution and the black lines represent the numerical results. The right column is the cross-section of the wave function at $s_2 = 2.69$; the blue solid line represents the analytical solution and the red dots represent the numerical results.

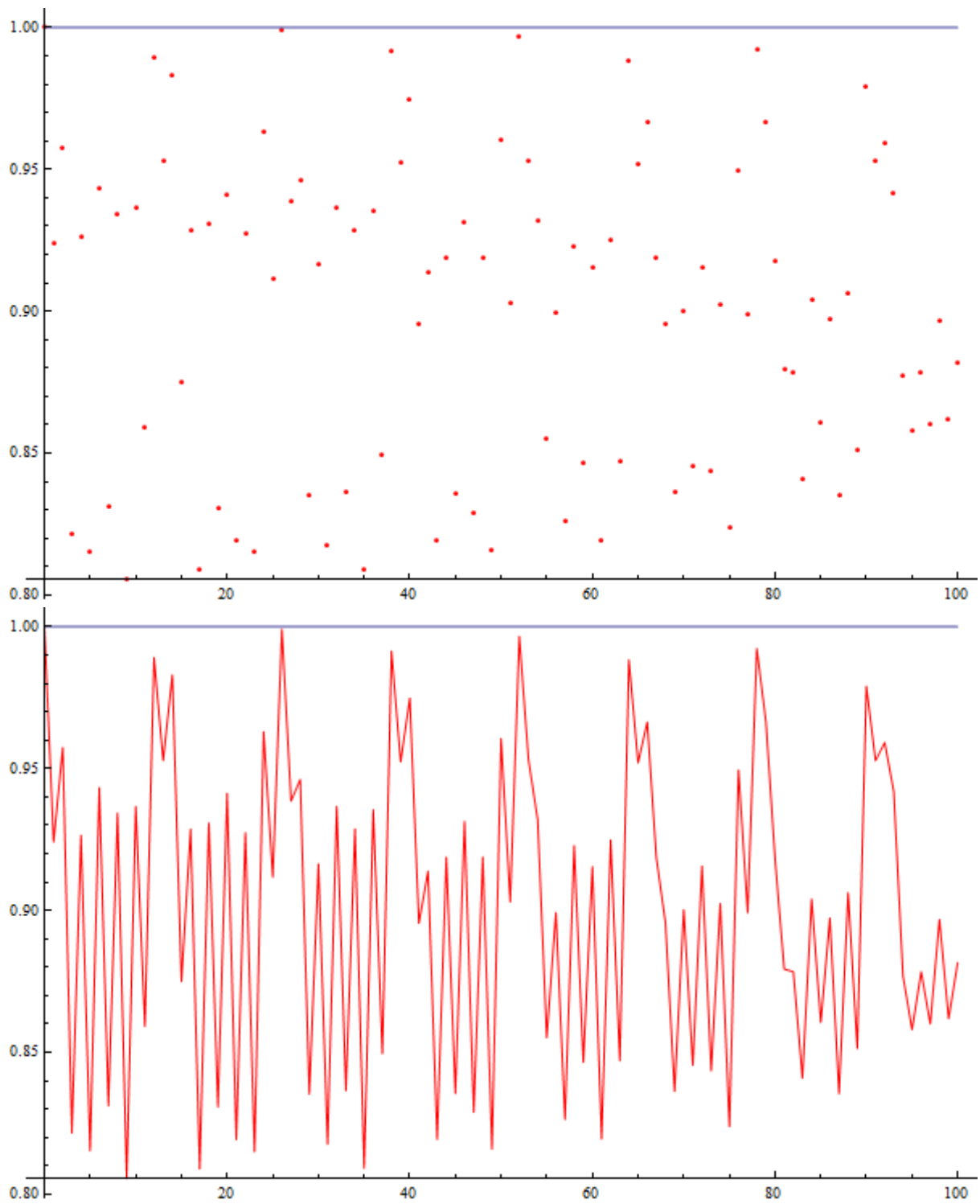


Figure 3.3: The blue horizontal line indicates the γ_0 value for $g = 0$. The red dots in the upper graph show the γ values for $g = 0.5$, from $t = 0$ to $t = 100$. The red broken line in the lower graph is obtained by joining the red dots.

Chapter 4

Conclusions

With reference to the existing Basic Matrix Numerov method, we have successfully developed the generalized Numerov method into the Generalized Matrix Numerov Method, and applied it to the Simple Harmonic Oscillator and Coupled Harmonic Oscillators problems.

Basic Matrix Numerov Method has the advantage of giving a large number of eigenstates simultaneously by discretizing the wave function on a linear lattice with appropriate boundary conditions and representing the Hamiltonian with a square matrix. However, the accuracy of results is limited, making it unsuitable for solving problems with greater complexity.

Generalized Matrix Numerov Method preserves the advantage and, at the same time, overcomes the limitation. Its validity is confirmed by comparing the numerical results with the analytical solutions, and we are able to conclude that with this method it is possible to solve rather complex quantum dynamics problems to infinitely high accuracy, provided that there is sufficient computing power. Given a certain computing environment, optimum results can be obtained with minimum CPU time by choosing suitable parameters.

The potential application of the Generalized Matrix Numerov Method is to simulate quantum dynamics for pedagogical purposes, as well as for comparison and reference in scientific research. Apparently the utility of this method is not limited to solving the Schrödinger equation; with slight modification it will be suitable for finding solutions to other Numerov-type problems as well.

Chapter 5

Appendix

5.1 Mathematica Codes for Finding Elements in the Kinetic Energy Matrix

$r = 3$; (* Enter the value of r in $O(d^{2r+2})$ *)

```
g = Solve [Table [  $\frac{(ad)^{2b}}{(2b)!}$ , {a, 1, r}, {b, 1, r} ] . Table[ $\Psi[2c]$ , {c, r}] ==  
Table [  $\frac{\Psi_{i-k} - 2\Psi_i + \Psi_{i+k}}{2}$ , {k, r} ] , Table[ $\Psi[2c]$ , {c, r}]] ;
```

```
coefficientA = Coefficient [g[[1]][[1]][[2]], Table [ $\Psi_{m+i}$ , {m, -r, r}]]
```

```
coefficientBraw =  $\frac{2}{(2r+2)!}d^{2r}$  Coefficient [g[[1]][[-1]][[2]], Table [ $\Psi_{m+i}$ , {m, -r, r}]] ;
```

```
coefficientB = ReplacePart[coefficientBraw, r + 1 → coefficientBraw[[r + 1]] + 1]
```

$$\left\{ \frac{1}{90d^2}, -\frac{3}{20d^2}, \frac{3}{2d^2}, -\frac{49}{18d^2}, \frac{3}{2d^2}, -\frac{3}{20d^2}, \frac{1}{90d^2} \right\}$$

$$\left\{ \frac{1}{20160}, -\frac{1}{3360}, \frac{1}{1344}, \frac{1007}{1008}, \frac{1}{1344}, -\frac{1}{3360}, \frac{1}{20160} \right\}$$

5.2 Mathematica Codes for Simple Harmonic Oscillator

$r = 1$; (* Enter the value of r in $O(d^{2r+2})$ *)

(* Find elements in matrices A and B *)

$g = \text{Solve} \left[\text{Table} \left[\frac{(ad)^{2b}}{(2b)!}, \{a, 1, r\}, \{b, 1, r\} \right] . \text{Table}[\Psi[2c], \{c, r\}] == \right.$
 $\left. \text{Table} \left[\frac{\Psi_{i-k} - 2\Psi_i + \Psi_{i+k}}{2}, \{k, r\} \right], \text{Table}[\Psi[2c], \{c, r\}] \right];$

$\text{coefficientA} = \text{Coefficient}[g[[1]][[1]][[2]], \text{Table}[\Psi_{m+i}, \{m, -r, r\}]];$

$\text{coefficientBraw} = \frac{2}{(2r+2)!} d^{2r} \text{Coefficient}[g[[1]][[-1]][[2]], \text{Table}[\Psi_{m+i}, \{m, -r, r\}]];$

$\text{coefficientB} = \text{ReplacePart}[\text{coefficientBraw}, r + 1 \rightarrow \text{coefficientBraw}[[r + 1]] + 1];$

(*Potential, desired max energy*)

$V[s.] := .5s^2; \epsilon m = 50.;$

(* Determine grid *)

$\text{return} = \text{FindRoot}[V[s] == \epsilon m, \{s, \epsilon m\}][[1, 2]]; d = \frac{1}{\sqrt{2\epsilon m}};$

$n = \text{Round} \left[2 \left(\frac{\text{return}}{d} + 4\pi \right) \right]; s = \text{Table} \left[-\frac{d(n+1)}{2} + di, \{i, n\} \right];$

(* Calculate KE matrix *)

$\mathbb{I}[n., d.] := \text{DiagonalMatrix}[1 + \text{ORange}[n - \text{Abs}[d]], d];$

$B = \text{Total}[\text{Table}[\text{coefficientB}[[u + r + 1]] * \mathbb{I}[n, u], \{u, -r, r\}]];$

$A = \text{Total}[\text{Table}[\text{coefficientA}[[v + r + 1]] * \mathbb{I}[n, v], \{v, -r, r\}]];$

$\text{KE} = \frac{-1}{2} \text{Inverse}[B].A;$

(* Hamiltonian *)

$H = \text{KE} + \text{DiagonalMatrix}[V[s]];$

(* Energies, wavefunctions *)

```
{eval, evec} = Eigensystem[H];
```

```
(* Ordering of eigenvalues and eigenfunctions *)
```

```
in = Ordering[eval];
```

```
eval = eval[[in]]; evec = evec[[in]];
```

```
(* Pick out eigenvalues within the energy range *)
```

```
evalnum = Select[eval, # ≤ εm&];
```

```
(* Find theoretical eigenvalues *)
```

```
l = Length[evalnum];
```

```
evaltheo = Table[w - 0.5, {w, l}];
```

```
(* Linear regression *)
```

```
lm = LinearModelFit[Transpose[{evaltheo, evalnum}], x, x]
```

```
rsquared = lm["RSquared"]
```

```
FittedModel[0.0135104 + 0.998789x]
```

1.

```
(* Plot Numerov wavefunctions, normalize to 1 *)
```

```
evecnum[n_]:=evec[[n + 1]] /  $\sqrt{d}$ ;
```

```
f[i_]:= - (d(n + 1))/2 + di
```

```
scale = Map[f, Range[n]];
```

```
result = Transpose[{scale, evecnum[49]}];
```

```
numerov = ListPlot[result, PlotRange → All, PlotStyle → {Red}];
```

```
(* Theoretical eigenvector *)
```

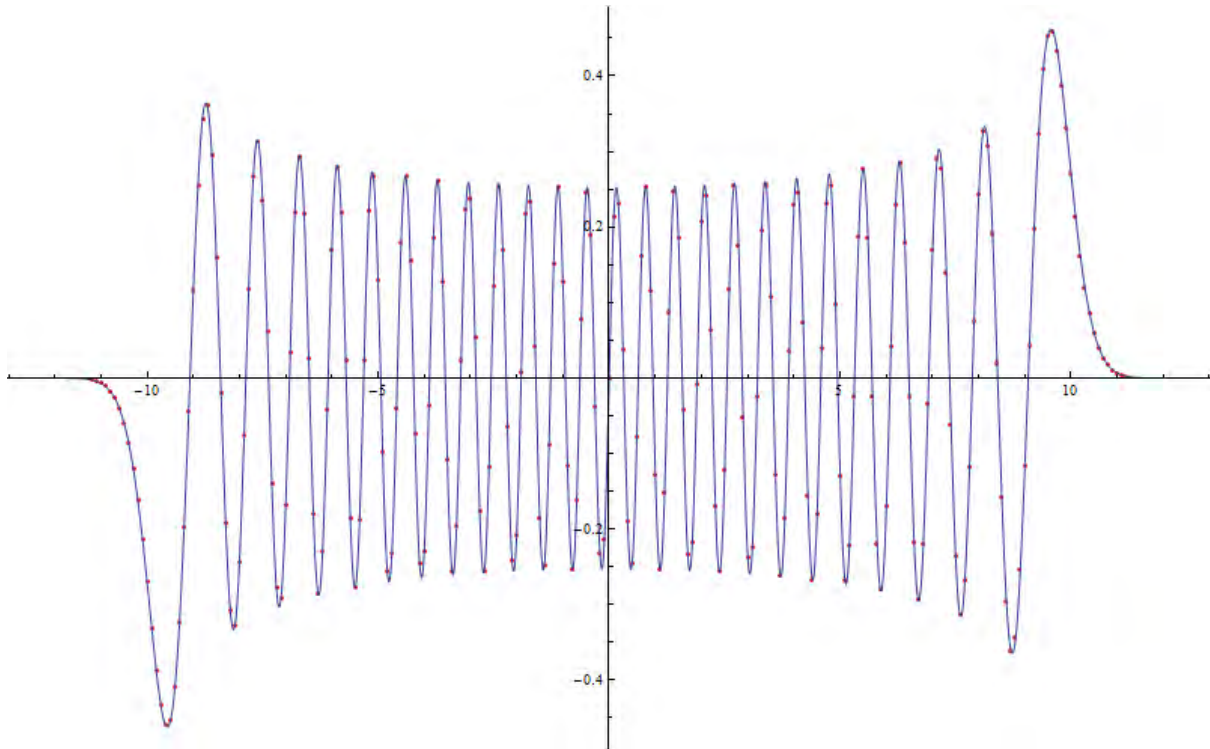
```

evectheo[s_, n_] :=  $\frac{\pi^{-1/4}}{\sqrt{2^n * (n)!}}$  Exp  $\left[-\frac{s^2}{2}\right]$  * HermiteH[n, s];
analytical = Plot[evectheo[s, 49], {s, -12.5, 12.5}, PlotRange -> All];

```

(* Plot graph for $n = 49$ state *)

```
Show[numerov, analytical, ImageSize -> 800]
```



(* Coefficient list for $\Psi(s, t)$ *)

```

ctheo = Table  $\left[\int_{-\infty}^{\infty} \text{Conjugate}[evectheo[x, n]] * \left(\frac{1}{4\pi}\right)^{1/4} e^{-x^2/8} dx, \{n, 0, l-1\}\right] // N;$ 
cnum = Table  $\left[\text{Total} \left[\text{Conjugate}[evecnum[n]] * \left(\frac{1}{4\pi}\right)^{1/4} e^{-s^2/8} * d\right], \{n, 0, l-1\}\right];$ 

```

(* Numerov and theoretical solutions for $\Psi(s, t)$ *)

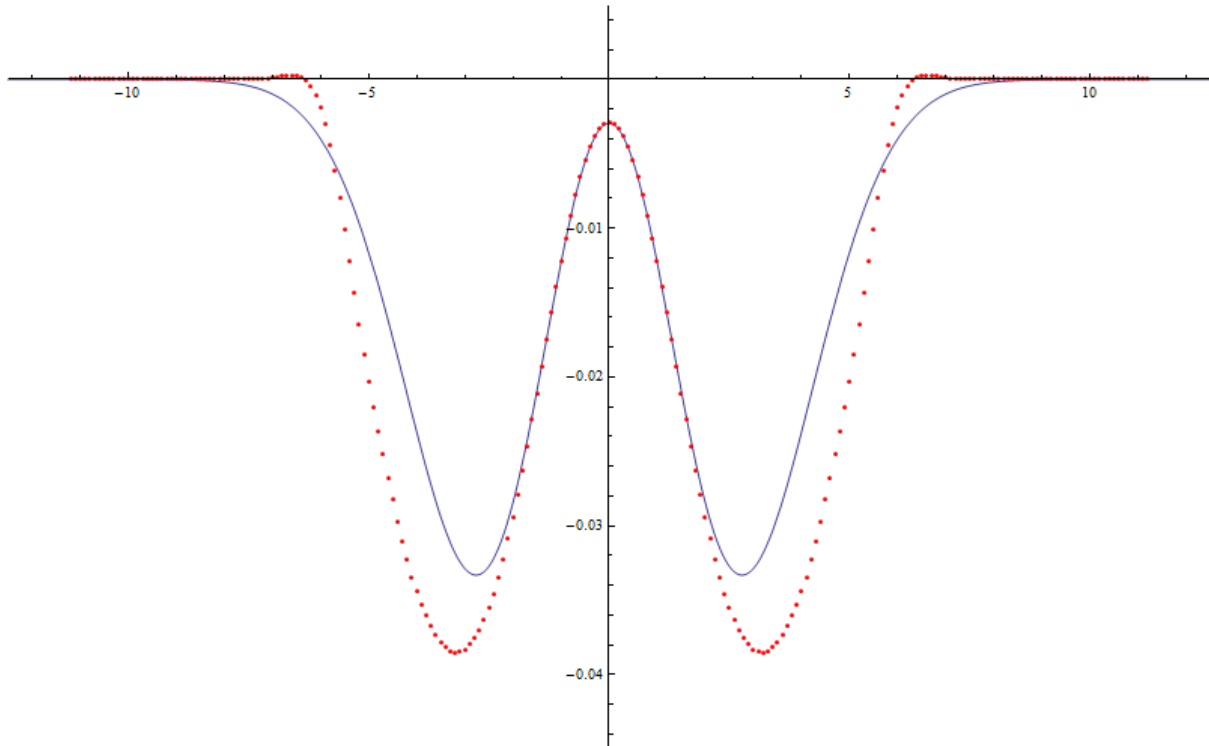
```

realpsinum[t_] := Re  $\left[\text{Total} \left[e^{-I*evalnum*t} * cnum * \text{Table}[evecnum[n], \{n, 0, l-1\}]\right]\right];$ 
realpsitheo[x_, t_] := Re  $\left[\text{Total} \left[e^{-I*evaltheo*t} * ctheo * \text{Table}[evectheo[x, n], \{n, 0, l-1\}]\right]\right];$ 
impsinum[t_] := Im  $\left[\text{Total} \left[e^{-I*evalnum*t} * cnum * \text{Table}[evecnum[n], \{n, 0, l-1\}]\right]\right];$ 
impsitheo[x_, t_] := Im  $\left[\text{Total} \left[e^{-I*evaltheo*t} * ctheo * \text{Table}[evectheo[x, n], \{n, 0, l-1\}]\right]\right];$ 

```

(* Plot graph for real part of $\Psi(s, 600)$ *)

```
Show[Plot[realpsitheo[x, 600], {x, -12.5, 12.5}, PlotRange -> {{-12.5, 12.5},  
{-0.045, 0.005}}, ListPlot[{s, realpsinum[600]}T, PlotStyle -> {Red},  
PlotRange -> {{-12.5, 12.5}, {-0.045, 0.005}}, ImageSize -> 800]
```



5.3 Mathematica Codes for Coupled Harmonic Oscillators

$\omega_1 = 1.5$; (* Enter the value for ω_1 *)

$\omega_2 = 1$; (* Enter the value for ω_2 *)

$cc = 0.2$; (* Enter the value for coupling constant g *)

$r = 5$; (* Enter the value of r in $O(d^{2r+2})$ *)

$n = 50$; (* Enter the value of N *)

(* Find elements in matrices A and B *)

$$g = \text{Solve} \left[\text{Table} \left[\frac{(ad)^{2b}}{(2b)!}, \{a, 1, r\}, \{b, 1, r\} \right] . \text{Table}[\Psi[2c], \{c, r\}] == \right. \\ \left. \text{Table} \left[\frac{\Psi_{i-k} - 2\Psi_i + \Psi_{i+k}}{2}, \{k, r\} \right], \text{Table}[\Psi[2c], \{c, r\}] \right];$$

$$\text{coefficientA} = \text{Coefficient} [g[[1]][[1]][[2]], \text{Table} [\Psi_{m+i}, \{m, -r, r\}]];$$

$$\text{coefficientBraw} = \frac{2}{(2r+2)!} d^{2r} \text{Coefficient} [g[[1]][[-1]][[2]], \text{Table} [\Psi_{m+i}, \{m, -r, r\}]];$$

$$\text{coefficientB} = \text{ReplacePart}[\text{coefficientBraw}, r + 1 \rightarrow \text{coefficientBraw}[[r + 1]] + 1];$$

(* Determine grid *)

$$\text{xt1} = \sqrt{\left(\frac{n}{2} - 4\pi\right) \left(\frac{1}{\omega_1}\right)};$$

$$\text{e1} = .5 * \omega_1^2 * \text{xt1}^2;$$

$$\text{d1} = \frac{1}{\sqrt{2 * \text{e1}}};$$

$$\text{x1} = \text{Table} \left[-\frac{\text{d1}(n+1)}{2} + \text{d1}i, \{i, n\} \right];$$

$$\text{V1} = .5 * \omega_1^2 * \text{x1}^2;$$

$$\text{xt2} = \sqrt{\left(\frac{n}{2} - 4\pi\right) \left(\frac{1}{\omega_2}\right)};$$

$$\text{e2} = .5 * \omega_2^2 * \text{xt2}^2;$$

$$\text{d2} = \frac{1}{\sqrt{2 * \text{e2}}};$$

$$\text{x2} = \text{Table} \left[-\frac{\text{d2}(n+1)}{2} + \text{d2}i, \{i, n\} \right];$$

$$\text{V2} = .5 * \omega_2^2 * \text{x2}^2;$$

(* Calculate KE matrix *)

$$\mathbb{I}[n, d] := \text{DiagonalMatrix}[1 + \text{ORange}[n - \text{Abs}[d]], d];$$

$$B = \text{Total}[\text{Table}[\text{coefficientB}[[u + r + 1]] * \mathbb{I}[n, u], \{u, -r, r\}]];$$

$$A = \text{Total}[\text{Table}[\text{coefficientA}[[v + r + 1]] * \mathbb{I}[n, v], \{v, -r, r\}]];$$

$$A1 = A / .d \rightarrow \text{d1};$$

$$A2 = A / .d \rightarrow \text{d2};$$

$$\text{KE1} = -\frac{1}{2} \text{Inverse}[B] . A1;$$

$$\text{KE2} = -\frac{1}{2} \text{Inverse}[B] . A2;$$

(* Hamiltonian *)

```
H = KroneckerProduct[KE1 + DiagonalMatrix[V1], IdentityMatrix[n]]  
+KroneckerProduct[IdentityMatrix[n], KE2 + DiagonalMatrix[V2]]  
+cc * KroneckerProduct[DiagonalMatrix[x1], DiagonalMatrix[x2]];
```

(* Energies, wavefunctions *)

```
{eval, evec} = Eigensystem[H];  
in = Ordering[eval];  
eval = eval[[in]]; evec = evec[[in]];
```

(* Theoretical eigenvalues *)

$$\omega = (\omega_1^2 \omega_2^2 - cc^2)^{1/4} // N;$$
$$\eta = \frac{1}{2} \text{Log} \left[\frac{\omega_1^2 + \omega_2^2 + \sqrt{(\omega_1^2 - \omega_2^2)^2 + 4cc^2}}{2\sqrt{\omega_1^2 \omega_2^2 - cc^2}} \right] // N;$$

$$\text{If} \left[\omega_2^2 == \omega_1^2, \theta = \frac{\pi}{2}, \theta = \text{ArcTan} \left[\frac{2cc}{\omega_2^2 - \omega_1^2} \right] \right];$$

```
evaltheo =  $\omega$ (Flatten[Table[( $E^\eta$ ) $i$  + ( $E^{-\eta}$ ) $j$  + Cosh[ $\eta$ ], { $i$ , 0,  $n - 1$ }, { $j$ , 0,  $n - 1$ }]]];  
in2 = Ordering[evaltheo];  
evaltheo = evaltheo[[in2]];
```

(* Pick out eigenvalues within the energy range *)

```
evaltheo = Select[evaltheo, # <= (e1 + e2)&];  
l = Length[evaltheo];  
evalnum = eval[[1;;l]];
```

(* Linear regression *)

```
lm = LinearModelFit[Transpose[{evaltheo, evalnum}], z, z]
```



```
rsquared = lm["RSquared"]
```

```
FittedModel[0.00078274 + 0.999875z]
```

1.

```
(* Set up noncommutative multiplication *)4
```

```
scalarQ[_]:=False
```

```
ncTimes[]:=1
```

```
ncTimes[a_]:=a
```

```
ncTimes[a___, ncTimes[b_, c_], d___]:=ncTimes[a, b, c, d]
```

```
ncTimes[a___, x_ + y_, b___]:=ncTimes[a, x, b] + ncTimes[a, y, b]
```

```
ncTimes[a___, i_Integer * c_, b___]:=i * ncTimes[a, c, b]
```

```
ncTimes[a___, i_Integer, b___]:=i * ncTimes[a, b]
```

```
ncTimes[a___, s[i_Integer], s[i_], b___]:=ncTimes[a, b]
```

```
ncTimes[a___, s[i_Integer], s[j_Integer], b___]/;j > i:=  
- ncTimes[a, s[j], s[i], b]
```

```
differentialOperate[a_, expr_]/;FreeQ[a, D]:=a * expr
```

```
differentialOperate[L1_ + L2_, expr_]:=
```

```
differentialOperate[L1, expr] + differentialOperate[L2, expr]
```

```
differentialOperate[a_ * L_, expr_]/;FreeQ[a, D]:=a * differentialOperate[L, expr]
```

```
differentialOperate[a : HoldPattern[D[_]&], expr_]:=a[expr]
```

```
differentialOperate[ncTimes[L1_, L2_], expr_]:=
```

```
Expand[differentialOperate[L1, differentialOperate[L2, expr]]]
```

```
differentialOperate[L1_^n_Integer, expr_]/;n > 1:=
```

```
Nest[Expand[differentialOperate[L1, #]]&, expr, n]
```

```
ddvar[x_, n_:1]:=D[#, {x, n}]&
```

(* Theoretical expression for eigenstates *)

```

ψ[0, 0] = √(ω/π) * Exp[-ω/2 (Exp[η] * y112 + Exp[-η] * y222)];
ψψ[i-, j-]:=differentialOperate [ncTimes [(√(ω/2) * Exp[η/2] * y11
-√(1/2ω) * Exp[-η/2] * ddvar[y11])i,
(√(ω/2) * Exp[-η/2] * y22 - √(1/2ω) * Exp[η/2] * ddvar[y22])j]
/√(i! * j!), ψ[0, 0]]
ψ[p-, q-]:=ψψ[p, q]/.ncTimes[a-, b-] → a * b;
y1 = x11 * Cos [θ/2] - x22 * Sin [θ/2];
y2 = x11 * Sin [θ/2] + x22 * Cos [θ/2];
evectheo[n1-, n2-]:=ψ[n1, n2]/.{y11 → y1, y22 → y2};

```

(* Numerov expression for eigenstates *)

```

evecnum[n1-, n2-]:=evec[[First[First[Position[in2, n1 * n + (n2 + 1)]]]]] / √(d1 * d2);

```

(* Plot (n₁, n₂) = (6, 5) state *)

```

Show[Plot3D[evectheo[6, 5], {x11, First[x1], Last[x1]}, {x22, First[x2], Last[x2]},
Mesh → None, PlotRange → All, ImageSize → 600],
ListPlot3D[Transpose[Partition[evecnum[6, 5], n]], PlotStyle → None,
PlotRange → All, ImageSize → 600, DataRange → {{First[x1], Last[x1]},
{First[x2], Last[x2]}, Automatic}], AxesLabel → {s1, s2}]

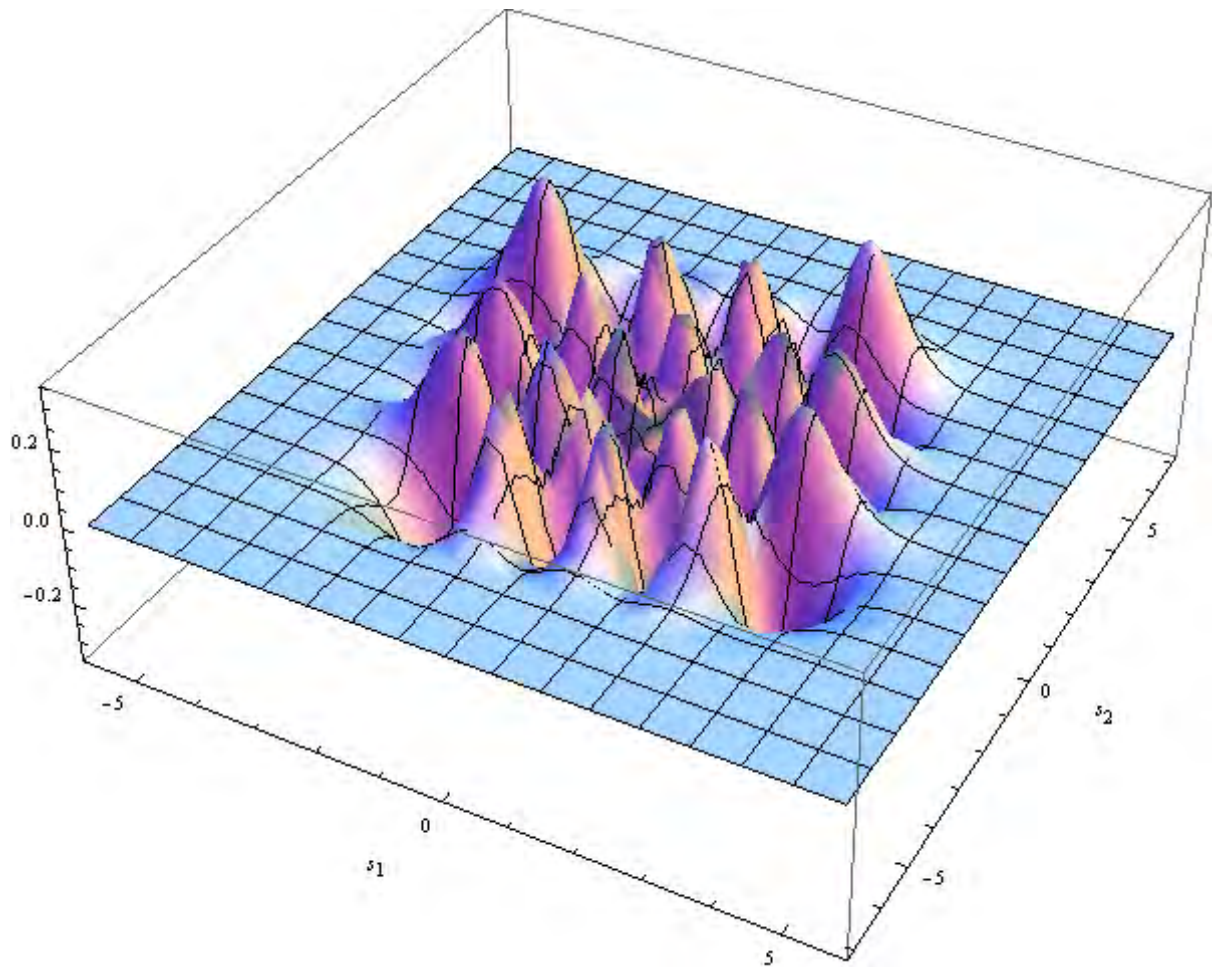
```

(* Coefficient list for Ψ (s₁, s₂, t) *)

```

evaltheotable = ω(Table[(E^η)i + (E^(-η))j + Cosh[η], {i, 0, n - 1}, {j, 0, n - 1}]);
evaltheoposition = Position[evaltheotable, _?(# ≤ (e1 + e2)&)];
evaltheopositionnn = evaltheoposition - 1;
evectheolist = Table[evectheo[evaltheopositionnn[[n]][[1]], evaltheopositionnn[[n]][[2]], {n, l}];
ctheo = NIntegrate [Conjugate[evectheolist] * (1/2π)1/4 e-(x11-1)2/4 (1/2π)1/4 e-(x22+2)2/4,
{x11, -∞, ∞}, {x22, -∞, ∞}, MaxRecursion → 16]//N;

```



```
cnum = Table [Total [Conjugate [evec[[k]] /  $\sqrt{d1 * d2}$ ]
*Flatten [Table [( $\frac{1}{2\pi}$ )1/4 e-(x1[[i]-1)2/4 ( $\frac{1}{2\pi}$ )1/4 e-(x2[[j]+2)2/4, {i, n}, {j, n}]]]
*d1 * d2], {k, l}];
```

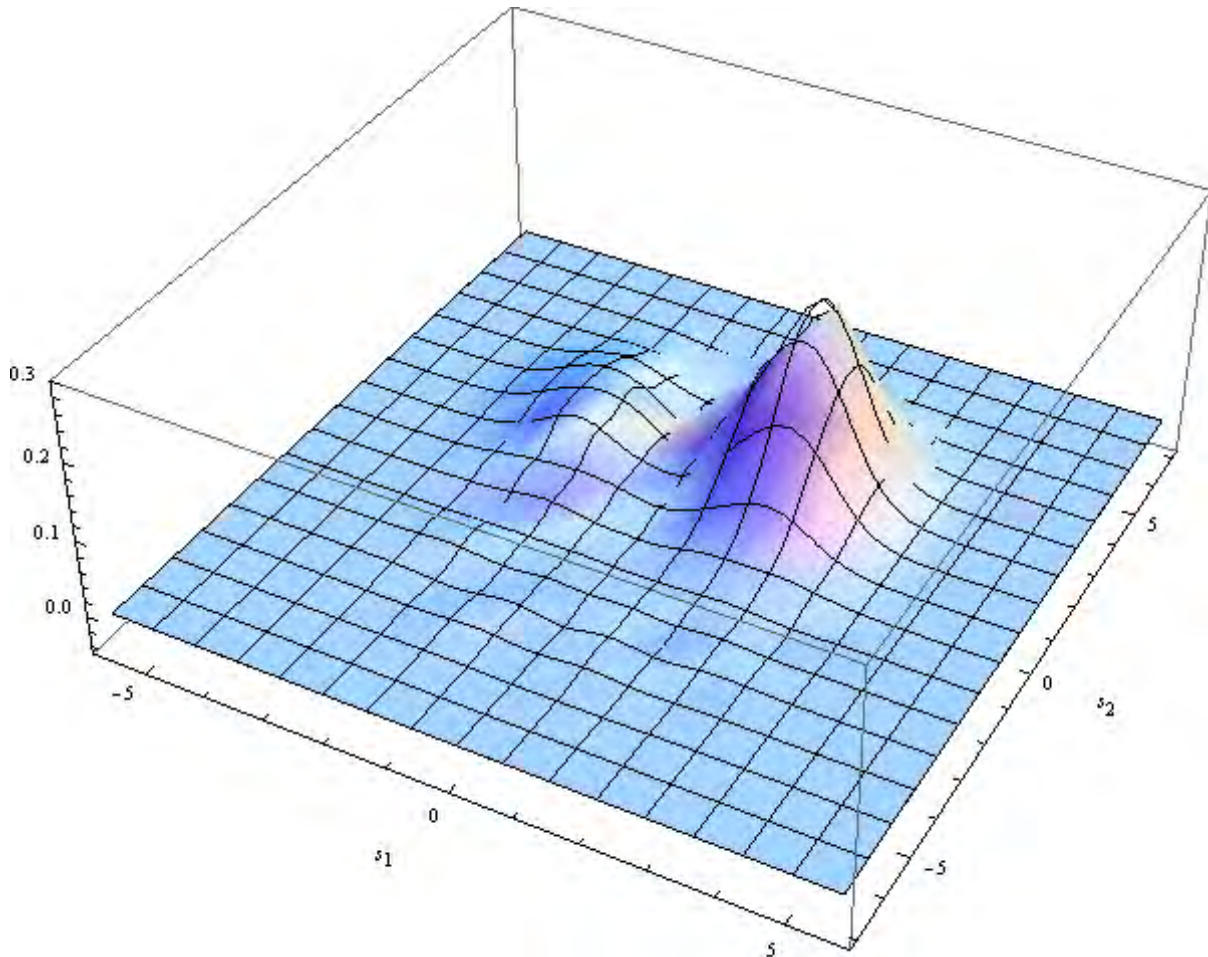
(* Numerical and theoretical solutions for $\Psi(s_1, s_2, t)$ *)

```
psinum[t.]:=Total [e-I*evalnum*t * cnum * Table [evec[[k]] /  $\sqrt{d1 * d2}$ , {k, l}]]];
psitheo[t.]:=Total [Table [e-I*Extract[evaltheotable,evaltheoposition[[n]]*t, {n, l}]]
*ctheo * evecetheolist];
```

(* 3-D plot for real part of $\Psi(s_1, s_2, 150)$ *)

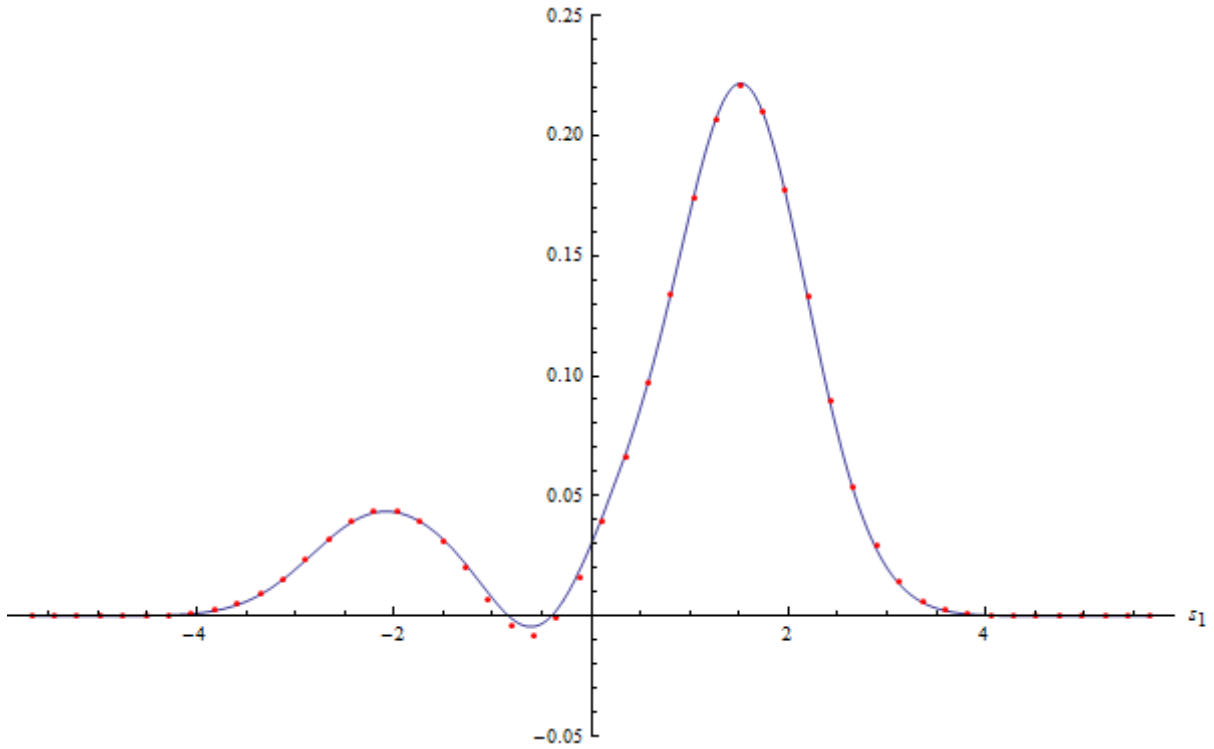
```
Show[Plot3D[Re[psitheo[150]], {x11, First[x1], Last[x1]}, {x22, First[x2], Last[x2]},
Mesh → None, ImageSize → 600, PlotRange → {-0.05, 0.3}],
```

```
ListPlot3D[Re[Transpose[Partition[psinum[150], n]], PlotStyle → None,
DataRange → {{First[x1], Last[x1]}, {First[x2], Last[x2]}}, ImageSize → 600,
PlotRange → {-0.05, 0.3}], AxesLabel → {s1, s2}
```



(* Cross section for real part of $\Psi(s_1, s_2, 150)$ at $s_2 = 2.69$ *)

```
Show[Plot[Re[psitheo[150]/. {x22 → x2[[35]]}], {x11, First[x1], Last[x1]},
PlotRange → {-0.05, 0.25}, ImageSize → 600],
ListPlot[Re[Transpose[Partition[psinum[150], n]][[35]],
DataRange → {First[x1], Last[x1]}, ImageSize → 600,
PlotRange → {-0.05, 0.25}, PlotStyle → {Red}], AxesLabel → {s1, None}]
```



(* Purity ρ_1 at $t = 150$ *)

$$\rho_1 = \text{Tr}[\text{MatrixPower}[\text{Table}[\text{Tr}[\text{Take}[\text{KroneckerProduct}[\text{psinum}[150], \text{Conjugate}[\text{psinum}[150]]], \{n * i - n + 1, n * i\}, \{n * j - n + 1, n * j\}], \{i, n\}, \{j, n\}], 2] * d1^2 * d2^2]$$

$$0.970268 + 5.12754 \times 10^{-19}i$$

(* Purity ρ_2 at $t = 150$ *)

$$\rho_2 = \text{Tr}[\text{MatrixPower}[\text{Sum}[\text{Take}[\text{KroneckerProduct}[\text{psinum}[150], \text{Conjugate}[\text{psinum}[150]]], \{n * i - n + 1, n * i\}, \{n * i - n + 1, n * i\}], \{i, n\}], 2] * d1^2 * d2^2]$$

$$0.970268 + 2.83854 \times 10^{-19}i$$

Bibliography

- [1] Pillai, Mohandas, Joshua Goglio, and Thad G. Walker. "Matrix Numerov method for solving Schrödingers equation." *American Journal of Physics* 80.11 (2012): 1017-1019.
- [2] Stefan Kröll, Andreas Walther, Peter Samuelsson and Göran Johansson. "Quantum Information (Lecture 4-5)." *Mathematical Physics - Education*. Lund University, 13 Mar 2013. Web. 4 Apr 2014. <<http://www.matfys.lth.se/education/quantinfo/>>.
- [3] Jellal, Ahmed, Fethi Madouri, and Abdeldjalil Merdaci. "Entanglement in coupled harmonic oscillators studied using a unitary transformation." *Journal of Statistical Mechanics: Theory and Experiment* 2011.09 (2011): P09015.
- [4] Daniel Lichtblau. "Symbolic FAQ." *Wolfram Library Archive*. Wolfram Research, Inc, Jan 2005. Web. 4 Apr 2014. <<http://library.wolfram.com/infocenter/Conferences/325>>.