

**The study of modeled atomtronic barrier potentials  
through the impedance method**

by

**Ava Ashby**

B.S., University of Colorado at Boulder, 2016

M.S., University of Colorado at Boulder, 2016

A thesis submitted to the  
Faculty of the Graduate School of the  
University of Colorado in partial fulfillment  
of the requirements for the degree of  
Master of Science  
Department of Physics

2016

This thesis entitled:  
The study of modeled atomtronic barrier potentials through the impedance method  
written by Ava Ashby  
has been approved for the Department of Physics

---

Prof. Dana Anderson

---

Prof. Heather Lewandowski

---

Prof. Charles Rogers

Date \_\_\_\_\_

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Ashby, Ava (M.S., Physics)

The study of modeled atomtronic barrier potentials through the impedance method

Thesis directed by Prof. Dana Anderson

The general impedance method developed by Khondker *et.al* [1] is applied to multiple barrier potentials to study the transmission behaviors of a single Gaussian barrier potential, double rectangular barrier potential, and a double Gaussian barrier potential. Using the impedance method we will find quantum mechanical transmission behaviors like resonant tunneling for the quasi bound-states of the double barrier potentials. The single Gaussian barrier potential is an approximation for the atomtronic battery and the double Gaussian barrier potential is an approximation for the atomtronic transistor. Thus far, the atomtronic work presented in reference [2] was done in the classical regime where thermal currents dominate. The work presented in this thesis analyzes the quantum mechanical characteristics, such as tunneling, to be incorporated and perhaps used to explore coherent transistor action.

## Dedication

To all of the fluffy puppies.

## Acknowledgements

Thank you to everyone who has helped make this project a success. Thank you to Dana Anderson and the entirety of the Anderson group for all of their support and wisdom. Thank you to all of my professors through out the years for with out their tutelage I would not be where I am today. Thank you to all of my peers who have struggled through the last few years with me. Last but not least thank you to my parents who have always been as supportive as possible during the most stressful of times.

## Contents

<b>Chapter</b>	
<b>1</b>	<b>1</b>
<b>2</b>	<b>4</b>
2.1	4
2.2	8
2.3	11
<b>3</b>	<b>13</b>
3.1	13
3.2	15
3.3	24
<b>4</b>	<b>25</b>
4.1	25
4.2	28
4.3	37
<b>5</b>	<b>39</b>
5.1	39
5.2	41
5.3	63

<b>6</b>	<b>Conclusion</b>	<b>64</b>
	<b>Bibliography</b>	<b>67</b>
	<b>Appendix</b>	
<b>A</b>	The Mathematica package for the impedance method	<b>68</b>
<b>B</b>	The step potential barrier Mathematica notebook	<b>72</b>
<b>C</b>	The single Gaussian barrier potential Mathematica notebook	<b>79</b>
<b>D</b>	The Mathematica package for the the eigenenergies	<b>98</b>
<b>E</b>	The double rectangular barrier potential Mathematica notebook	<b>102</b>
<b>F</b>	The Mathematica package for the reflection amplitude coefficients	<b>126</b>
<b>G</b>	The double Gaussian barrier potential Mathematica notebook	<b>130</b>
<b>H</b>	Eigenenergy tables for the double Gaussian barrier potential	<b>186</b>

## Tables

### Table

3.1	The first tunneling energy calculated from the impedance method for the single Gaussian barrier. . . . .	17
3.2	The last energy before full transmission is achieved, calculated from the impedance method for the single Gaussian . . . . .	18
3.3	The central rate of change or slope of the Transmission spectrum between the first tunneled energy and the last energy before full transmission is achieved for the single Gaussian . . . . .	19
4.1	The eigenenergies calculated using the impedance method for a double rectangular barrier . . . . .	31
4.2	The eigenenergies for a double rectangular barrier potential . . . . .	35
H.1	Table of the resonant energies for a double Gaussian barrier potential . . . . .	187
H.2	Table of the resonant energies for a double Gaussian barrier potential . . . . .	187
H.3	Table of the resonant energies for a double Gaussian barrier potential . . . . .	187
H.4	Table of the resonant energies for a double Gaussian barrier potential . . . . .	188
H.5	Table of the resonant energies for a double Gaussian barrier potential . . . . .	188
H.6	Table of the resonant energies for a double Gaussian barrier potential . . . . .	188
H.7	Table of the resonant energies for a double Gaussian barrier potential . . . . .	189

## Figures

### Figure

2.1	An arbitrary potential function showing the iteration scheme for the impedance method	7
2.2	A flow chart depicting the iteration scheme for the impedance method . . . . .	10
2.3	Transmission spectrum for the step potential . . . . .	12
3.1	The Gaussian barrier potential with varied $\sigma$ . . . . .	14
3.2	The transmission spectrum of the Gaussian barrier with varied $\sigma$ . . . . .	17
3.3	The transmission spectrum of the Gaussian barrier with varied height . . . . .	18
3.4	The transmission spectrum for a the single Gaussian potential with peak potential height at 30 kHz with the variance at the extreme limits of $\sigma \ll 1$ and $\sigma \gg 1$ . . .	21
3.5	The wavepacket interacting with the Gaussian barrier potential . . . . .	22
4.1	The double rectangular barrier potential whose potential height is {30 kHz, 31 kHz, 32 kHz} and barrier width is $0.333 \mu\text{m}$ with a separation distance of $0.625 \mu\text{m}$ . . . .	27
4.2	Comparison of analytical solutions and the impedance method . . . . .	29
4.3	The transmission spectrum for the double rectangular barrier potential described by Eq. (4.1) . . . . .	30
4.4	A compilation of transmission spectrums of the double rectangular barrier . . . . .	32
4.5	The transmission spectrum for the double rectangular barrier potential . . . . .	34
4.6	The transmission spectrum for the double rectangular barrier potential . . . . .	34

4.7	A representation of a wavepacket interacting with the double rectangular barrier potential . . . . .	36
5.1	Atomtronic transistor triple-well potential . . . . .	40
5.2	The double Gaussian barrier potential . . . . .	42
5.3	The double Gaussian barrier potential . . . . .	43
5.4	The double Gaussian barrier potential . . . . .	44
5.5	The finite difference method Gaussian well potential . . . . .	45
5.6	The comparison of two different methods for generating the eigenenergies of the double Gaussian barrier potential . . . . .	46
5.7	Double Gaussian barrier potential . . . . .	48
5.8	The transmission spectrum for a double Gaussian barrier potential . . . . .	49
5.9	The transmission spectrum for a double Gaussian barrier potential . . . . .	51
5.10	The transmission spectrum for a double Gaussian barrier potential . . . . .	52
5.11	The double Gaussian barrier potential . . . . .	54
5.12	The double Gaussian barrier potential transmission spectrum . . . . .	55
5.13	The double Gaussian barrier potential transmission spectrum . . . . .	56
5.14	The double Gaussian barrier potential . . . . .	57
5.15	The transmission spectrum for a double Gaussian barrier potential . . . . .	58
5.16	The transmission spectrum for a double Gaussian barrier potential . . . . .	59
5.17	The optimized transmission spectrum for the double Gaussian barrier potential . . .	61
5.18	The optimized transmission spectrum for the double Gaussian barrier potential . . .	62
H.1	Summary of resonant energy trends . . . . .	190

# Chapter 1

## Introduction

### Purpose of Study

This thesis details the use of the impedance method [1, 3] as a means to develop and understand atomtronic circuits. The impedance method is a numerical method that draws an analogy between the one-dimensional time-independent Schrödinger equation and transmission line theory in electronics and has been applied to double rectangular barrier potentials in [4] and [5] as well as other other well potentials, as in [6], for finding the bound-state solutions. The impedance method is used to calculate transmission/reflection probabilities, complex reflection coefficients, and eigenenergies of an arbitrary potential. In this thesis the impedance method is applied to a single Gaussian barrier potential, a double rectangular barrier potential, and a double Gaussian barrier potential. For the double rectangular barrier potential and the double Gaussian barrier potential, the impedance method can also be used to evaluate the eigenenergies for the resulting well between the barriers. In this work, application to atomtronics is made through a single Gaussian barrier potential and a double Gaussian barrier potential. Atomtronics is the atomic analog to electronics, in which neutral atoms replace electrons resulting in an atom current rather than an electron current. The chemical potential,  $\mu$ , of the system replaces the electric potential to further the analogy between atomtronics and electronics. The work done in reference [2] was done in the classical regime where thermal affects dominate. The work done in this thesis is an exploration of quantum mechanical effects like tunneling and can be used to incorporate and design potentials that use tunneling as a feature to atomtronic devices.

The atomtronic battery is created by the superposition of a harmonic well and a single Gaussian barrier potential [2]. The single Gaussian barrier potential will be treated using the impedance method in chapter 3 to find the behaviors of transmission associated with transmission. The atomtronic transistor is modeled by the superposition of a harmonic well and a double Gaussian barrier potential. The triple-well formed by this superposition is analogous to the three terminals of an electronic transistor, and the chemical potential in each well is analogous to the node voltage of the electronic transistor. The double Gaussian barrier potential will be treated by using the impedance method in chapter 4.

Because the impedance method is a numerical approximation, there are limitations to its effectiveness. Numerical limitations depend on the computation program used, and for more complex potentials, like that of the double barrier, energy spectra with much higher density or non-uniform optimized density must be used. Other limitations come from approximation of the potential barrier itself; since the impedance method incurs error for potentials with infinite slope, e.g. the step potential or double rectangular barrier. However, unlike other methods such as the Wentzel-Kramers-Brillouin (WKB) method, there is not a limitation on the variation of the potential. The impedance method is effective, simple to use, and can be applied to almost any potential.

This thesis is arranged as follows:

Chapter 2 introduces the general impedance method, a numerical method developed by Khondker *et al* in reference [1]. The analytically solvable step potential which is used as an example of the application of the impedance method. The impedance method is extended to the potential barriers covered in chapters 3, 4, and 5. Chapter 3 covers the single Gaussian barrier potential. Using the impedance method the transmission spectrum of a single Gaussian barrier potential is evaluated and used to determine the characteristics of the potential that lead to certain behaviors of the transmission. The single Gaussian barrier potential is an approximation of the atomtronics battery potential. Chapter 4 covers the double rectangular barrier potential. The double rectangular barrier presents an interesting case that can be solved analytically. The double rectangular barrier allows for resonant tunneling; meaning that incoming energies less than that of

the barrier height may have unity transmission when on resonance with the middle well. Resonant tunneling and the quasi-bound eigenenergies are explored and compared to its analytical solution. Chapter 5 explores the double Gaussian barrier potential. The double Gaussian barrier potential is an approximation to the atomtronic transistor potential. Manipulating the characteristics of the barriers and analyzing the resulting transmission spectrum and resonant eigenenergies allows us to optimize the atomtronic transistor potential to include quantum mechanical properties. The atomtronic transistor uses two blue detuned light sheets superposed on a harmonic magnetic trap to form a triple-well geometry. The purpose of analyzing the transmission spectrum of the double Gaussian barrier potential is to gain further understanding of quantum mechanical behaviors, like that of resonant tunneling, and apply them to the triple-well potential. Chapter 6 concludes the thesis. For the three barrier potentials studied here, the single Gaussian barrier potential, the double rectangular barrier potential, and the double Gaussian barrier potential, apparent relations between barrier characteristics and transmission energies are made through the use of the impedance method. The double Gaussian barrier potential can be applied to the atomtronic transistor potential in order to optimize certain behaviors, such as a coupling of resonant energy to the dipole moment of the gate condensate.

## Chapter 2

### Introduction to the General Impedance Method

#### 2.1 The transmission line analogy: The general impedance method

The transmission line analogy, also known as the general impedance method or impedance method, was developed by Khondker, *et al.* [1] in 1987 as a method to solve the Schrödinger equation for the quantum mechanical transmission spectrum of an arbitrary potential barrier. Originally this method was developed to describe tunneling in electronic devices, here the method is applied to atomtronic devices. Knowing the quantum mechanical transmission probability is useful for determining tunneling conditions, bound state conditions, resonant tunneling energies, wavefunction behavior, etc. Using Schrödinger's equation to find the transmission probabilities is only analytically possible with a handful of potential barrier functions, such as the infinite/finite square well, the delta potential, and the finite square barrier; many of these are common examples in physics textbooks [7, 8]. Other numerical and approximation methods can be used to solve for the transmission spectrum. For example, the Wentzel-Kramers-Brillouin approximation (WKB), provides an analytical solution for a slowly varying potential, or the Schrödinger equation can be solved directly via the finite difference method.

The general impedance method is a numerical method that calculates the quantum mechanical wave impedance allowing for the quantum mechanical transmission probabilities to be evaluated as well as the eigenenergies. The method is based on the suggestion that two similar problems will have similar solutions and draws a parallel between the Schrödinger equation and transmission line theory for a generalized distributed impedance. The barrier of an arbitrary potential in the

quantum scheme is analogous to the junction between two different transmission lines, and the wavefunction and its derivative are analogous to the transmission line voltage and current, respectively. The qualitative comparison below show the relationship between Schrödinger's equation and transmission line theory,

$$\begin{array}{ll}
\text{Schrödinger :} & \text{Transmission line theory :} \\
\psi(x) = A^+(e^{\kappa_1 x} - \rho e^{-\kappa_1 x}) & I(x) = I^+(e^{\gamma x} - \Gamma e^{-\gamma x}) \\
-\frac{i\hbar}{m} \frac{d\psi}{dx} = A^+ \zeta_0 (e^{\kappa_1 x} + \rho e^{-\kappa_1 x}) & \iff V(x) = I^+ \zeta_0 (e^{\gamma x} + \Gamma e^{-\gamma x}) \quad , \quad (2.1) \\
R = |\rho|^2 = \left| \frac{\kappa_2 - \kappa_1}{\kappa_2 + \kappa_1} \right|^2 & R_L = |\Gamma|^2 = \left| \frac{\zeta_{LOAD} - \zeta_{0,LINE}}{\zeta_{LOAD} + \zeta_{0,LINE}} \right|^2
\end{array}$$

where  $\gamma = \kappa = \frac{i\sqrt{2m(E-U)}}{\hbar}$  is the propagation constant and  $\kappa_1, \kappa_2$  denote the propagation constant in the reflected region 1 and barrier region 2 respectively. The wave amplitude reflection coefficient is  $\rho = \Gamma$ ,  $\zeta_0 = -\frac{i\hbar\kappa}{m}$  is the characteristic impedance, and  $R$  is the quantum reflection coefficient, which is directly related to  $R_L$ , the reflection coefficient defined in transmission line equations. The load and characteristic impedances of the transmission line are  $\zeta_{LOAD}$  and  $\zeta_{0,LINE}$ , respectively. The amplitude constants,  $A^+$  and  $I^+$ , are defined by initial conditions. Here,  $\hbar$  is Planck's constant,  $m$  is the atom mass,  $E$  is the energy of the incoming atom wave, and  $U$  is the potential barrier function.

The total voltage,  $V(x)$ , and total current,  $I(x)$ , between two different transmission lines must be continuous across the boundary, analogous to the boundary conditions of the quantum mechanical wavefunction,  $\psi(x)$ , and its derivative,  $\frac{d\psi}{dx}$ , must also be continuous across a barrier. The impedance, given a complex current and voltage, which is defined by Ohm's law as  $\zeta = V(x)/I(x)$  therefore, the quantum mechanical wave impedance is defined as

$$\zeta(x) = V(x)/I(x) = \frac{-i\hbar}{m} \frac{1}{\psi(x)} \frac{d\psi}{dx} \longrightarrow \zeta_{\text{input}} = \zeta_0 \frac{\zeta_{\text{load}} \cosh(\kappa l) - \zeta_0 \sinh(\kappa l)}{\zeta_0 \cosh(\kappa l) - \zeta_{\text{load}} \sinh(\kappa l)} \quad , \quad (2.2)$$

where  $\zeta_{\text{input}}$  is the input impedance,  $\zeta_0$  is the characteristic impedance defined in Eq. (2.5), and

$\zeta_{\text{load}}$  is the load impedance [1, 3]. The quantum mechanical wave impedance,  $\zeta_{\text{input}}$ , must be continuous across the boundaries; at the boundaries the load impedance,  $\zeta_{\text{load}}$ , in the reflection region is equivalent to the input impedance,  $\zeta_{\text{input}}$ , inside the barrier. From the boundary conditions the definition of the wave amplitude reflection coefficient becomes  $\Gamma = \frac{\zeta_{\text{input}} - \zeta_0}{\zeta_{\text{input}} + \zeta_0}$ , allowing for the transmission probabilities to be determined from  $T = 1 - |\Gamma|^2$ . The quantum mechanical wave impedance also allows for the eigenenergies of potential wells to be evaluated. The energy condition to find the eigenstate energies becomes

$$\zeta_{\text{input},L}(x_{\text{ref}}) = \zeta_{\text{input},R}(x_{\text{ref}}) , \quad (2.3)$$

where  $\zeta_{\text{input},L}(x_{\text{ref}})$  is the left looking impedance from a chosen reference plane and  $\zeta_{\text{input},R}(x_{\text{ref}})$  is the right looking impedance from the same chosen reference plane. The Mathematica package for calculating the eigenenergies is provided in Appendix B.

The general impedance method uses the quantum mechanical wave impedance in an iterative manner where,

$$k_j = \frac{i\sqrt{2m(E_j - V_j)}}{\hbar} , \quad (2.4)$$

$$\zeta_{0,j} = -\frac{i\hbar k_j}{m} , \quad (2.5)$$

$$\zeta_{\text{input},j} = \zeta_{0,j} \frac{\zeta_{\text{load},j} \cosh(k_j l) - \zeta_{0,j} \sinh(k_j l)}{\zeta_{0,j} \cosh(k_j l) - \zeta_{\text{load},j} \sinh(k_j l)} , \quad (2.6)$$

where  $k_j$  is the propagation constant for iteration  $j$  and  $\zeta_{0,j}$  is the characteristic impedance for iteration  $j$ .  $\zeta_{\text{input},j}$  is the input impedance of the  $j$ th segment and  $\zeta_{0,j}$  is the characteristic impedance of the  $j$ th segment (Note: for  $j = n$  where  $n$  is the total number of iterations,  $\zeta_{\text{load},n} = \zeta_{0,n}$ ). The load impedance of the  $j$ th segment is  $\zeta_{\text{load},j}$  and is equal to  $\zeta_{\text{input},n-1}$ . From definition Eq. (2.2) one can iterate backwards from  $j = n$  to  $j = 1$  along the potential function in order to calculate  $\zeta_{\text{LOAD}}$  and  $\zeta_{0,\text{LINE}}$  and find the reflection coefficient,

$$R = \left| \frac{\zeta_{\text{input}} - \zeta_0}{\zeta_{\text{input}} + \zeta_0} \right|^2 = \left| \frac{\zeta_{\text{LOAD}} - \zeta_{0,\text{LINE}}}{\zeta_{\text{LOAD}} + \zeta_{0,\text{LINE}}} \right|^2 = R_L . \quad (2.7)$$

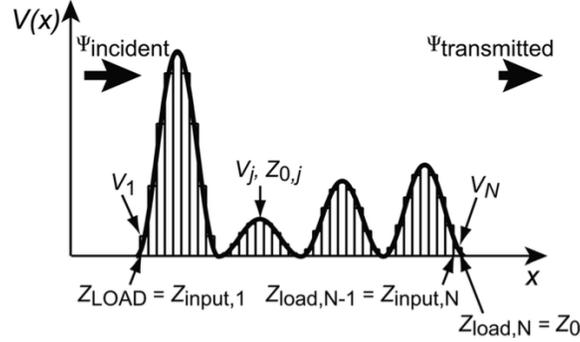


Figure 2.1: An arbitrary potential function showing the iteration scheme for the impedance method that begins on the right most point and ends on the left most. Image credit: [3].

Fig. 2.1 shows an incident wavefunction  $\psi_{\text{incident}}$  with some incoming energy and an arbitrary potential barrier function  $V(x_n)$ . Using Fig. 2.1 for reference, start with some  $\psi_{\text{incident}}$  at an energy  $E$  and a potential barrier defined by  $n$  discrete points. Then the impedance method initializes at the  $(x_n, V_n)$  position, where the step size  $x_n - x_{n-1} = l$  is defined and is the same through out the region of interest. We first apply Eq. (2.4) with  $E$  and  $V_j = V_n$  giving a value for  $k_n$ . From here  $\zeta_{0,n}$  is found from Eq. (2.5) and defines  $\zeta_{\text{load},n}$  giving  $\zeta_{\text{input},n} = \zeta_{0,n}$ . For the next iteration we move to position  $(V_{n-1}, x_{n-1})$  and set  $\zeta_{\text{load},n-1} = \zeta_{\text{input},n} = \zeta_{0,n}$ . Because  $V_{n-1} \neq V_n$ , it follows that  $k_{n-1} \neq k_n$  and  $k_{n-1}$  is used to find  $\zeta_{0,n-1}$ . Given  $l$ ,  $k_{n-1}$ ,  $\zeta_{0,n-1}$ ,  $\zeta_{\text{load},n-1} = \zeta_{0,n}$  and plugging into Eq. (2.2), we now have  $\zeta_{\text{input},n-1}$ . Continuing with this pattern all the way down to  $j = 1$  we are left with  $\zeta_{0,1}$  and  $\zeta_{\text{input},1}$  which is substituted into Eq. (2.1) for the reflection probability of our arbitrary potential. See Appendix A for the Mathematica package for the impedance method.

As an example, the simple and analytically solvable step potential barrier is presented. From there the impedance method will be applied to a single Gaussian barrier potential, a double rectangular barrier potential, and a double Gaussian barrier potential. The single Gaussian barrier potential and the double Gaussian barrier potential are simplifications of an atomtronics battery-like potential and transistor-like potential, respectively. The single Gaussian barrier potential will

be discussed in chapter 3 and the double Gaussian barrier potential is discussed in chapter 5.

## 2.2 An Example: The step barrier potential

The one-dimensional step potential, whose function is defined by Eq. (2.8) and shown in Fig. ??, may be solved analytically by the Schrödinger equation. The potential energy function is zero in the range  $x = (-\infty, 0)$  and a constant  $V_0$  over the range  $x = [0, \infty)$ .

The step potential function is defined by

$$V(x) = \begin{cases} 0 & : x < 0 \\ V_0 & : x \geq 0 \end{cases}, \quad (2.8)$$

where  $V_0$  is the height of the step potential [7]. The analytical solution for the reflection probability,  $R_{theory}$ , and transmission probability,  $T_{theory}$ , are

$$R_{theory} = \begin{cases} 1 & : 0 \leq E \leq V_0 \\ \left( \frac{E - \sqrt{E(E - V_0)}}{E + \sqrt{E(E - V_0)}} \right)^2 & : E > V_0 \end{cases}, \quad (2.9)$$

$$T_{theory} = 1 - R_{theory} = \begin{cases} 0 & : 0 \leq E \leq V_0 \\ \frac{4E^{3/2}\sqrt{E - V_0}}{\left( E + \sqrt{E(E - V_0)} \right)^2} & : E > V_0 \end{cases}, \quad (2.10)$$

where  $E$  is the energy and  $V_0$  is the height of the potential [7, 8] and is solved directly from the Schrödinger equation. In Fig. 2.2 the potential height is given a value of  $V_0 = 1$  kHz with a position range of  $x = [-3.0 \text{ mm}, 3.0 \text{ mm}]$  in steps of  $l = 0.40 \text{ } \mu\text{m}$  so that the assumption of the potential extending to infinity can be approximated for the analytic case. The necessary inputs for the impedance method are the step size,  $l$ , the potential barrier array,  $\vec{V}$ , and the incoming energy array,  $\vec{E}$ ;

$$\begin{aligned}
l &= x_n - x_{n-1} = x_{n-1} - x_{n-2} = \dots = x_2 - x_1 = 0.40 \mu\text{m} , \\
\vec{V} &= [0, \dots, 0, V_0, \dots, V_0]^T = [0, \dots, 0, 1, \dots, 1]^T \text{kHz} , \\
\vec{E} &= [E_1, \dots, E_m, \dots, E_n]^T = [0, \dots, 3.998, \dots, 5.0]^T \text{kHz} .
\end{aligned} \tag{2.11}$$

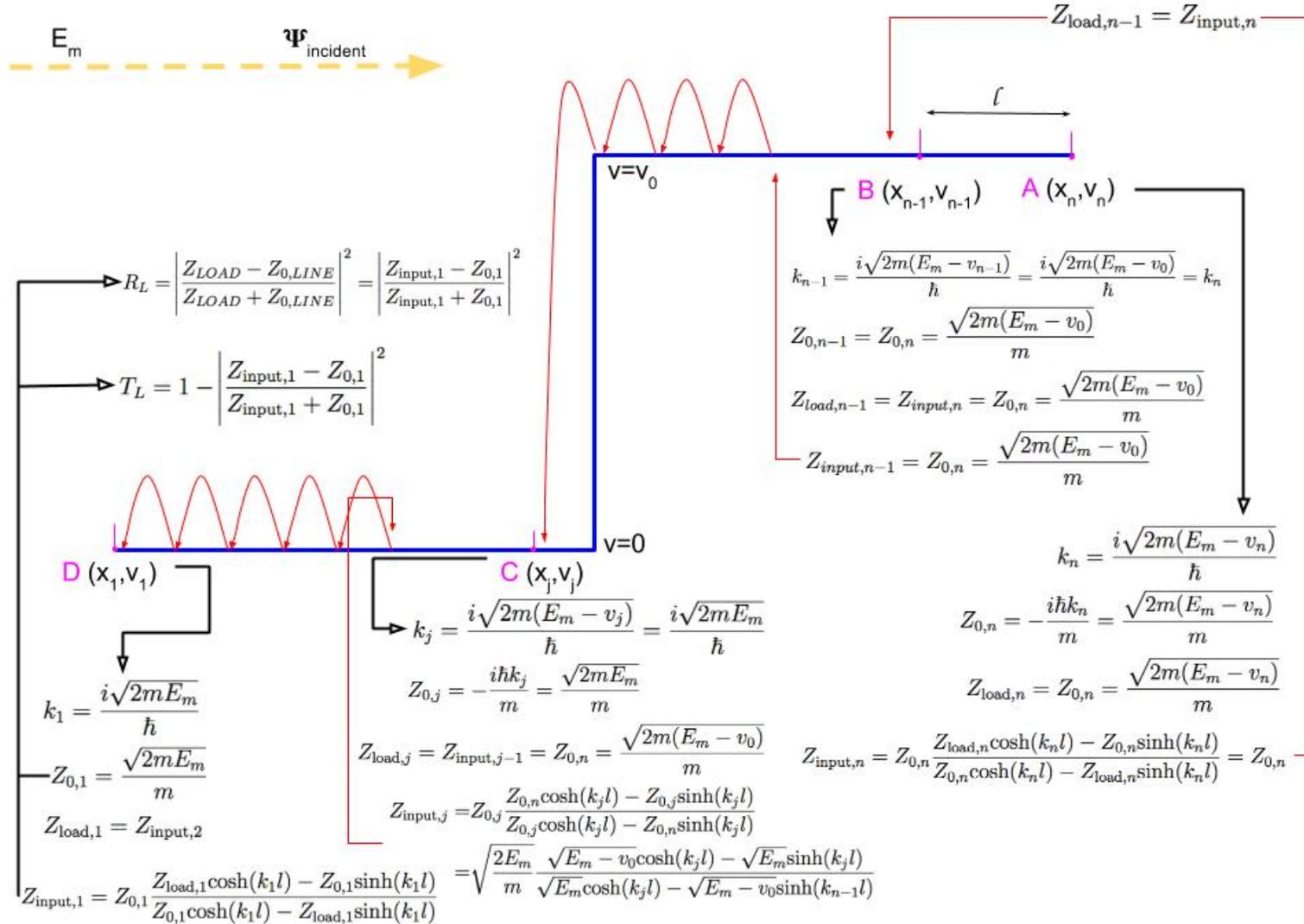


Figure 2.2: A flow chart depicting the iteration scheme for the impedance method using a step potential (blue) and some incident wave with energy  $E_m$  denoted by the yellow dashed arrow. Starting at point A and following through all the way to the end of the defined potential at point D. The step potential is the blue solid line. The points A, B, C, and D are denoted in pink. The iteration steps are denoted by red arrows and the impedance method scheme at steps A, B, C, and D are black.

Using Fig. 2.2 and starting with the nontrivial case of incoming energy  $E_m > v_0$ , the method begins at point A =  $(x_n, v_n)$ . The first calculation finds  $k_n$  from Eq. (2.4) where  $v_n = v_0$  and from there  $\zeta_{0,n}$  is defined by Eq. (2.5). For this initial iteration we set  $\zeta_{\text{load},n} = \zeta_{0,n}$  which it clearly follows that  $\zeta_{\text{input},n} = \zeta_{\text{load},n} = \zeta_{0,n}$ . Moving one step in the negative  $x$ -direction to point B =  $(x_{n-1}, v_{n-1})$  where  $v_{n-1} = v_0$ , and plugging in  $\zeta_{\text{load},n-1} = \zeta_{\text{input},n}$  we are left with all the same values; that is to say that  $k_{n-1} = k_n$ ,  $\zeta_{0,n-1} = \zeta_{0,n}$ ,  $\zeta_{\text{load},n-1} = \zeta_{\text{load},n} = \zeta_{0,n}$ , and  $\zeta_{\text{input},n-1} = \zeta_{\text{input},n} = \zeta_{0,n}$ . This is true for all following steps to point C =  $(x_j, v_j)$  where  $v_j = 0$ , and C is the step where  $v_{j+1} = v_0 \rightarrow v_j = 0$ . At point C  $k_j \neq k_n$  and it follows that  $\zeta_{0,j} \neq \zeta_{0,n}$  and  $\zeta_{\text{input},j} \neq \zeta_{0,n}$  as well. However  $\zeta_{\text{load},j} = \zeta_{0,n}$  is still true at this point because  $\zeta_{\text{load},j} = \zeta_{\text{load},j+1}$ . As we move from point C down to point D =  $(x_1, v_1)$  where  $v_1 = 0$ , it is clear to see that at each step it is true that  $k_j = k_{j-1} = \dots = k_1$  and  $\zeta_{0,j} = \zeta_{0,j-1} = \dots = \zeta_{0,1}$ , but  $\zeta_{\text{load},j} \neq \zeta_{\text{load},j-1} \neq \dots \neq \zeta_{\text{load},1}$  and thus changing the value of  $\zeta_{\text{input},i}$  along the way as well. At the end of the potential, the values of  $\zeta_{0,1}$  and  $\zeta_{\text{input},1}$  and are used in Eq. (2.7) to find the reflection probability  $R$ . The transmission spectrum for the step potential barrier is shown in Fig. 2.3 along with the analytically derived transmission probability from Eq. (2.10). The Mathematica notebook for the step potential barrier is given in Appendix B.

### 2.3 Summary of the numerical method

The impedance method is a numerical method that draws parallels between Schrödinger's equation and the transmission line theory by defining a quantum mechanical wave impedance. The impedance method is an iterative scheme that uses the propagation constant  $k_i$ , the characteristic impedance  $\zeta_{0,i}$ , and the input impedance  $\zeta_{\text{input},i}$ , i.e. the quantum wave impedance, to find the quantum mechanical transmission spectrum for an arbitrary potential energy function. For the case of a double barrier potential or well potential, the method can be used to find the eigenenergies of the quasi-bound states.

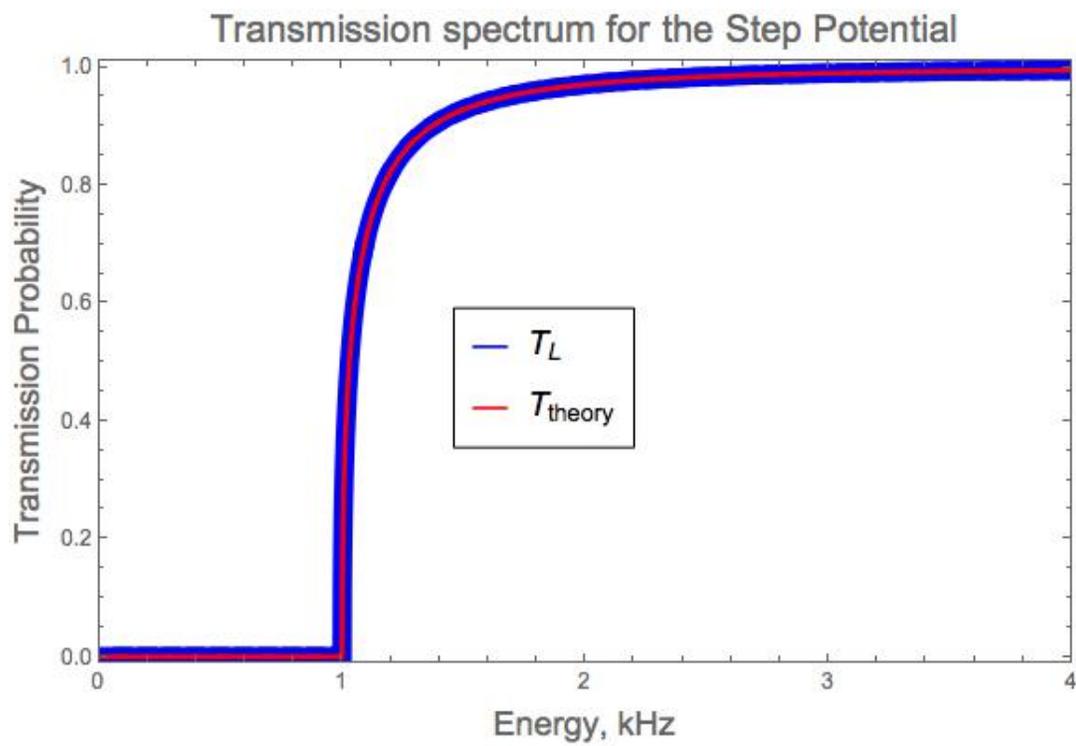


Figure 2.3: The transmission spectrum for the step potential barrier. The blue solid line is the evaluated transmission spectrum by the impedance method and the red solid line is the analytical solution for the transmission spectrum defined in Eq. (2.10).

## Chapter 3

### The Single Gaussian Potential

#### 3.1 The single Gaussian barrier potential

The Gaussian barrier potential, described by Eq. (3.1), is commonly used as the continuous counterpart of the finite square barrier in discussing tunneling behaviors. The single Gaussian barrier potential is described by,

$$V(x) = v_0 e^{-\frac{x^2}{\sigma^2}}, \quad (3.1)$$

where  $v_0$  is the peak potential height and  $\sigma$  is the variance, or width, of the barrier. The variance of the single Gaussian potential is related to the full width half maximum (FWHM) by  $\Gamma = \sqrt{\ln 2} \sigma$ . Figure 3.1 shows the Gaussian barrier and its variations of the potential height and variance. An increase in  $\sigma$  increases the FWHM which decreases the tunneling probability for the wavepacket of a given energy. Increasing the peak potential height makes the barrier taller and has the effect of shifting the tunneling energies to higher values.

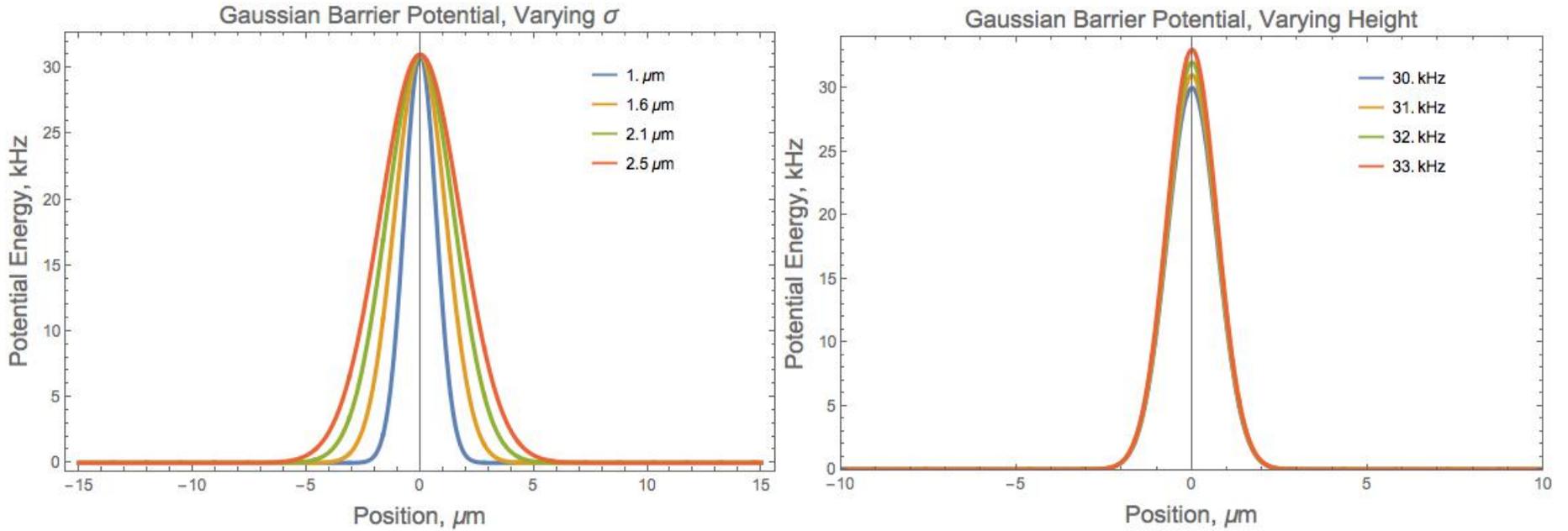


Figure 3.1: The Gaussian barrier potential with varied  $\sigma$ . The two barriers have values  $\sigma = \{1.0 \mu\text{m}$  (Blue solid line),  $1.6 \mu\text{m}$  (Orange solid line),  $2.1 \mu\text{m}$  (Green solid line),  $2.5 \mu\text{m}$  (Red solid line)} while the potential barrier height  $v_0$  is held fixed at 30 kHz. Notice how an increase of  $\sigma$  increases the FWHM (full width half max) of the potential, this increase in the FWHM will decrease the tunneling probability. The Gaussian barrier potential with varied potential height,  $v_0$ . The potential height has values  $v_0 = \{30 \text{ kHz}$  (Blue solid line),  $31 \text{ kHz}$  (Orange solid line),  $32 \text{ kHz}$  (Green solid line),  $33 \text{ kHz}$  (Red solid line)} while  $\sigma$  is held fixed at  $1.0 \mu\text{m}$ . An increase in  $v_0$  clearly increases the peak of the Gaussian and this will shift the tunneling probability to have a higher energy range.

The tunneling condition for the Gaussian barrier potential, as derived in reference [9], is

$$\frac{1}{8\sigma^2} > (v_0 - E), \quad (3.2)$$

where  $E$  is the incoming energy of the wavepacket. Eq. (3.2) was derived from the time-energy uncertainty principle  $\Delta E \Delta t \geq \frac{\hbar}{2}$  where  $\hbar$  has been taken to be unity. The left hand side of the tunneling condition is the kinetic energy of the wavepacket and the right hand side defines the energy difference. The condition states that the kinetic energy must be greater than the difference between the peak potential energy height and the incoming energy of the wavepacket. However, this tunneling condition is inconsistent; the values of the first tunneling energy in Table 3.1 violate this condition, and, as is discussed in reference [10], the dimensions on either side of Eq. (3.2) do not agree. When Eq. (3.2) is put into dimensionless notation,  $v_0 - \epsilon < \frac{1}{8}$ , the transmission probability for  $0 < \epsilon < v_0$  still violates this condition, i.e. there is still tunneling [10].

The single Gaussian barrier is a precursor to the double Gaussian barrier which is the approximated barrier function for the atomtronics transistor potential. The Gaussian barrier was studied for variations in the peak potential height,  $v_0$ , and in variance,  $\sigma$  as is shown in Fig. 3.1. The first tunneling energies are calculated using the impedance method's transmission spectrum for the different combinations of height and  $\sigma$ . These are given in Table 3.1.

### 3.2 Applying the impedance method to the single Gaussian barrier potential

Studying the transmission spectrum of the single Gaussian barrier potential allows for optimization of barrier characteristics such that only certain specific incoming energies can be transmitted or have full reflection. The transmission spectra for a single Gaussian barrier are shown in Fig. 3.2, and Fig. 3.3. The transmission spectrum for a Gaussian potential barrier whose height is held fixed at 30 kHz with varied  $\sigma$  is shown in Fig. 3.2. Notice that the transmission slope increases with increased  $\sigma$ , in other words the allowed tunneling energy values increase and approach the energy of the peak potential height from both sides. This is because as  $\sigma$  is increased, the FWHM

is increased and the wavepacket has more barrier to tunnel through. The 50% transmission energy is the same for all four different  $\sigma$  values and is directly related to the Gaussian barrier height. When the incoming energy of the wavepacket is equivalent to that of the barrier height, 50% of the incoming wavepacket is reflected back and the other 50% continues in the forward direction. This is because the incoming wavepacket is described by a plane wave, so it oscillates about its incident energy with half the wave being below the energy and half the wave being above it, thus 50% gets reflected back from the Gaussian barrier. At the lower end of the tunneling energy range one can clearly see from Fig. 3.2 and Table 3.1 that for a smaller  $\sigma$  one can have lower energies tunneling through the barrier and that the relation between tunneling and  $\sigma$  is not a linear one. At the upper end of the energy range in Fig. 3.2 and Table 3.2, when the incoming energies are at and above that of the peak barrier height  $v_0$  we see that for smaller  $\sigma$ , higher energies are still affected by the Gaussian barrier as compared to larger  $\sigma$  where transmission of the incoming wavepacket approaches unity faster. This behavior suggests that at the limits of  $\sigma \ll 1$  and  $\sigma \gg 1$  the transmission spectrum will behave as a transmission spectrum for a delta function potential or the classical transmission spectrum for a step potential respectively as shown in Fig. 3.4.

The transmission spectrum for a Gaussian potential barrier whose  $\sigma$  is held fixed at  $1.0 \mu\text{m}$  with varied peak potential height is shown in Fig. 3.3. The entire transmission spectrum is shifted to the right for higher peak potential energy heights, which is due to the barrier now being present at higher incoming energies. A change in the peak potential height has the effect of shifting the transmission spectrum's 50% transmission energy to the energy level of the peak potential height, changing both  $\sigma$  and the peak potential height leads to the unsurprising result of a shift in 50% transmission energy to  $v_0$  for the different variances. In other words, the first tunneling energy for a change from  $\sigma = 1.0 \mu\text{m}$  to  $\sigma = 1.6 \mu\text{m}$  for a potential height of 30 kHz differs by approximately 2 kHz and for the same change in  $\sigma$  at a potential height of 31 kHz, 32 kHz, and 33 kHz the first tunneling energies differ by approximately 2 kHz as well. This behavior is relevant for the optimization of the potential to specific tunneling energies.

The first tunneling energies in Table 3.1 are found using the impedance method rather than

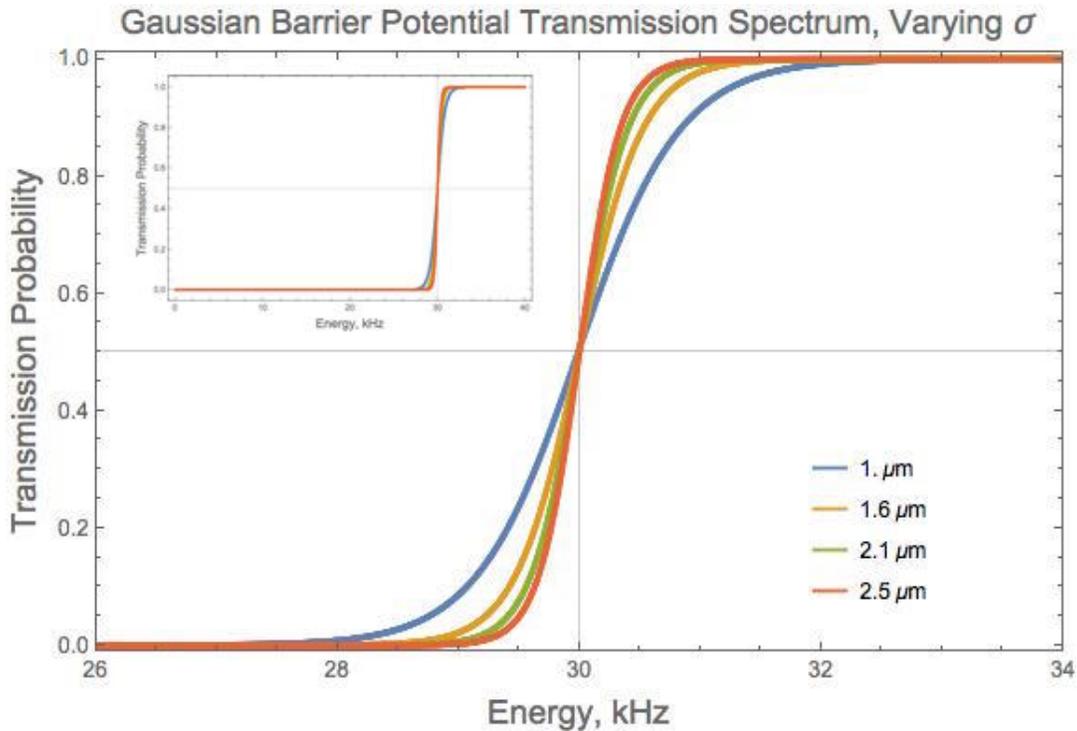


Figure 3.2: The transmission spectrum of the Gaussian barrier with  $\sigma = \{1.0 \mu\text{m} \text{ (Blue solid line)}, 1.6 \mu\text{m} \text{ (Orange solid line)}, 2.1 \mu\text{m} \text{ (Green solid line)}, 2.5 \mu\text{m} \text{ (Red solid line)}\}$  with the peak potential barrier height held fixed at 30 kHz. The light gray grid lines locate the 50% transmission point, or in other words 50% of the wavepacket is reflected off the Gaussian barrier. Notice how the energy of 50% transmission is equivalent to the Gaussian barrier height. The transmission spectrum with the largest variance,  $\sigma = 2.5 \mu\text{m}$  has the steepest slope, and this correlates to having a smaller energy range where tunneling is possible, i.e. the first allowed energy that can tunnel through the barrier is much closer to the Gaussian barrier height.

First Tunneling Energy in kHz				
$\frac{\text{Variance}}{\text{Height}}$	1. $\mu\text{m}$	1.6 $\mu\text{m}$	2.1 $\mu\text{m}$	2.5 $\mu\text{m}$
30. kHz	24.3768	26.4372	27.271	27.7014
31. kHz	25.2804	27.3769	28.2251	28.6629
32. kHz	26.1856	28.3177	29.18	29.625
33. kHz	27.0922	29.2593	30.1355	30.5876

Table 3.1: The first tunneling energy calculated from the impedance method for the single Gaussian barrier. The horizontal row entry denotes the variance  $\sigma$  and the vertical column entry denotes the peak potential height  $v_0$ .

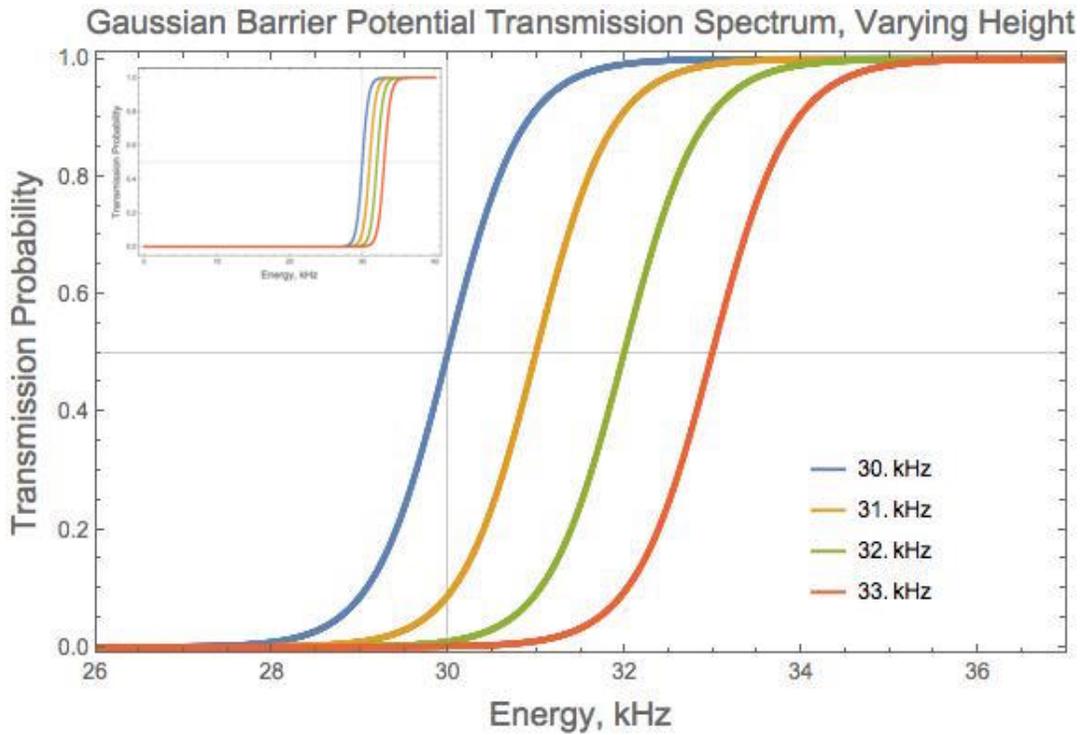


Figure 3.3: The transmission spectrum of the Gaussian barrier with varied height,  $v_0 = \{30 \text{ kHz (Blue solid line), } 31 \text{ kHz (Orange solid line), } 32 \text{ kHz (Green solid line), } 33 \text{ kHz (Red solid line)}\}$  with  $\sigma$  held fixed at  $1.0 \mu\text{m}$ . The light gray grid lines denote the 50% transmission point where 50% of the wavepacket is reflected off the barrier. Notice how the energy of 50% transmission is again equivalent to the Gaussian barrier height. The slope of the transmission line is the same for each variation but the tunneling energies have shifted by the same amount that the potential height has.

Last Energy Before Full Transmission in kHz				
Variance Height	$1. \mu\text{m}$	$1.6 \mu\text{m}$	$2.1 \mu\text{m}$	$2.5 \mu\text{m}$
30. kHz	36.0249	33.7197	32.8201	32.3628
31. kHz	37.1212	34.7799	33.8659	33.4013
32. kHz	38.216	35.8391	34.9111	34.4392
33. kHz	39.3094	36.8975	35.9555	35.4766

Table 3.2: The last energy before full transmission is achieved, calculated from the impedance method for the single Gaussian. The horizontal row entry denotes the variance of the Gaussian potential barrier and the vertical column entry denotes the peak potential height of the Gaussian potential barrier.

Transmission Rate of Change				
Variance Height	1. $\mu\text{m}$	1.6 $\mu\text{m}$	2.1 $\mu\text{m}$	2.5 $\mu\text{m}$
30. kHz	0.0858507	0.137315	0.180209	0.214527
31. kHz	0.0844536	0.13508	0.177279	0.211041
32. kHz	0.0831226	0.132954	0.174486	0.207718
33. kHz	0.0818517	0.130921	0.171821	0.20454

Table 3.3: The central rate of change or slope of the Transmission spectrum between the first tunneled energy and the last energy before full transmission is achieved for the single Gaussian. The horizontal row entry denotes the variance of the Gaussian potential barrier and the vertical column entry denotes the peak potential height of the Gaussian potential barrier.

the tunneling condition Eq. (3.2). The first tunneling energy was found by finding the first incoming energy in the set  $E_{incoming} = \{0.001, 0.0011, 0.0012, \dots, 39.9999, 40.0\}$  kHz that corresponds to the first reflection probability being less than unity. The values for each different variation of the potential are given in Table 3.1. Notice how an increase in  $\sigma$  increases the first tunneling energy, as does an increase in peak potential height of the Gaussian barrier. Changing the barrier height by 1 kHz causes the energy to shift by just under 1 kHz, but changing  $\sigma$  does not have such a near 1:1 relation. As  $\sigma$  is increased the growth in the energy is slowed, i.e. when  $\sigma = 1.0 \mu\text{m}$  versus when  $\sigma = 1.6 \mu\text{m}$  the energy increases by approximately 2, but when compared to  $\sigma = 1.6 \mu\text{m}$  and  $\sigma = 2.1 \mu\text{m}$  whose difference is  $0.5 \mu\text{m}$  the energy increases, but by less than unity. This can also be seen in Fig. 3.2 and Table 3.3, where the slope of transmission for  $\sigma = 1.0 \mu\text{m}$  is the slowest and the slope for  $\sigma = 2.5 \mu\text{m}$  is the fastest, while the  $\sigma$  values are between the two have slopes much closer to that of  $\sigma = 2.5 \mu\text{m}$  than  $\sigma = 1.0 \mu\text{m}$  even when the difference between them is lessened when taken with respect to  $\sigma = 1.0 \mu\text{m}$ . This behavior illustrates that the relation between  $\sigma$  and tunneling energies is not a linear one, although Eq. (3.2) cannot be used quantitatively, this is expected due to the tunneling condition. The slopes of the transmission spectrums are tabulated in Table 3.3, there the relation to  $\sigma$  and peak potential height is clear. The increase in peak potential height provides a slight decrease in the rate of change of the transmission spectrum over the energies of interest, but on the order of  $10^{-3}$ , where as the increase in  $\sigma$  has a more significant impact in the rate of change of the transmission spectrum over the energies of interest, that being on the order of  $10^{-1}$ . Table 3.2 shows the opposite trend for  $\sigma$  as Table 3.1 does; as  $\sigma$  is increased the last energy before full transmission decreases. This behavior is as expected since in the limiting case of a very large  $\sigma$  we expect the transmission spectrum to be that of a classical step potential; the energies which are transmitted are those greater than the peak potential height, otherwise full reflection happens. For each change in the potential height with the variance fixed at  $\sigma = 2.5 \mu\text{m}$  the last energy is approximately 2.5 kHz above the peak potential height, again illustrating a linear relationship. As is discussed above, when the limits of  $\sigma$  are very small or very large, corresponds to a delta function potential and a classical transmission for a step potential, illustrated in Fig. 3.4.

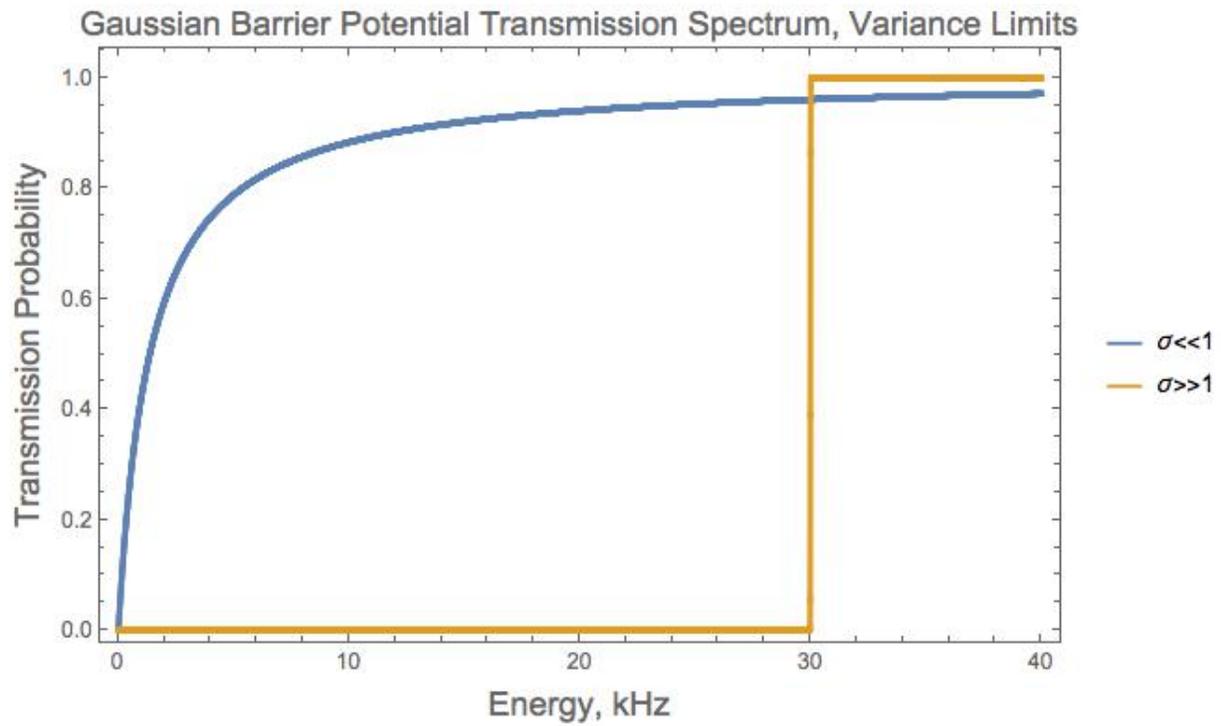


Figure 3.4: The transmission spectrum for a the single Gaussian potential with peak potential height at 30 kHz with the variance at the extreme limits of  $\sigma \ll 1$  and  $\sigma \gg 1$ . In the limit of  $\sigma \ll 1$  a transmission spectrum similar to that of a delta potential is expected and in the limit of  $\sigma \gg 1$  a transmission spectrum similar to that of a step potential is expected.

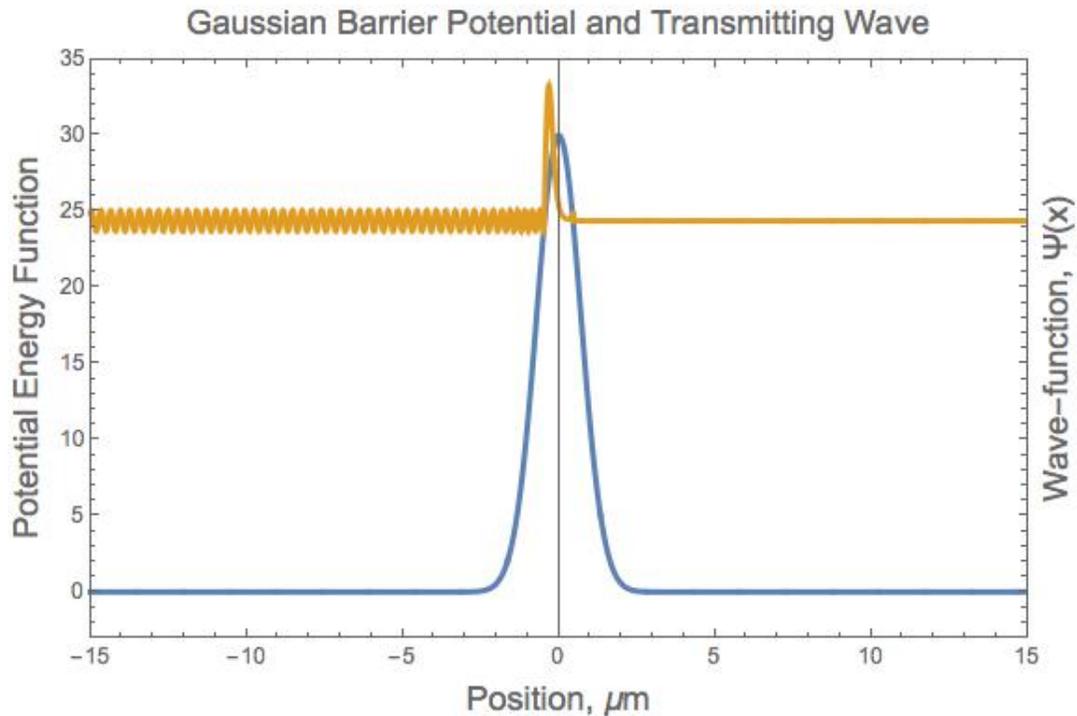


Figure 3.5: The wavepacket interacting with the Gaussian barrier potential. The potential has  $\sigma = 1.0 \mu\text{m}$  and  $v_0 = 30 \text{ kHz}$ . The wavepacket is described by a plane wave whose incoming energy is that of the first tunneling energy for this particular Gaussian barrier,  $E = 24.38 \text{ kHz}$ . The wavepacket on the left hand side of the barrier clearly oscillates about its incoming energy and is the superposition of the incident wave and the reflected wave. At the barrier one see's a spike in amplitude of the wavepacket which is from the reflected wave and incident wave having constructive interference. Within the barrier there is exponential decay until it exits the barrier and can be described by a plane wave traveling to the right, the amplitude of this plane wave is significantly reduced from the incident waves amplitude because we are at the lowest possible energy that still allows for tunneling, so a very small percent of the wave is transmitted through the barrier. It is key to note that this is a time-independent depiction.

An approximate depiction of what the wavepacket looks like when interacting with the Gaussian barrier potential is shown in Fig. 3.5. The Gaussian barrier potential is described by Eq. (3.1)  $\sigma = 1.0 \mu\text{m}$  and  $v_0 = 30 \text{ kHz}$ . The wavepacket has an incoming energy of  $E = 24.38 \text{ kHz}$ , which corresponds to the first tunneling energy for this particular Gaussian barrier, as stated in Table 3.1. The wavepacket is described by a plane wave, with an incident amplitude coefficient of unity and the reflection and transmission amplitude coefficients are determined by the general impedance method described in Chapter 2. As expected the wavepacket in region I, the region to the left of the barrier, has a clear oscillation about the energy  $E = 24.38 \text{ kHz}$  and is a superposition of the incident wave and the reflected wave. The spike at the barrier is in part due to the Gibbs phenomenon, which is a numerical and physical effect caused by large oscillations near the jump discontinuity, at the barrier, of the piecewise continuously differentiable periodic wavefunction which is defined in Eq. (3.3) and in part due to the constructive interference between the reflected wave and the incident wave. At the classical tuning point, the probability of finding the wave is higher because the kinetic energy is lower at the reflection point. An analogy for this is a classical pendulum, where probability of finding the pendulum at the turning points is highest. The piecewise periodic wavefunction is,

$$\Psi(x) = \begin{cases} e^{ikx} + re^{-ikx} & : x \in \text{Region I: Left of the barrier} \\ e^{-\kappa x} & : x \in \text{Region II: Inside the barrier} \\ te^{ikx} & : x \in \text{Region III: Right of the barrier} \end{cases}, \quad (3.3)$$

where  $k = \sqrt{\frac{2m(E-V(x))}{\hbar^2}}$  and  $\kappa = \sqrt{\frac{2m(V(x)-E)}{\hbar^2}}$ ,  $r$  is the reflection amplitude found from the impedance method (see Appendix C),  $t = \sqrt{1-R}$  where  $R$  is found by the impedance method (see Appendix A) and defined in Eq. (2.7). The behavior of the wavepacket within the barrier is that of exponential decay, and again at the jump discontinuity, as the wave exits the barrier, we see a slight spike from the Gibbs phenomenon and reflection within the barrier. The amplitude of the wavepacket that has successfully tunneled through is significantly reduced from that of the incident amplitude showing that there is indeed a small fraction of transmission occurring for this energy. The transmitted wave oscillates about the energy  $E = 24.38 \text{ kHz}$  but the amplitude is

comparatively much smaller.

### 3.3 Chapter conclusion

The atomtronic battery potential is a super position of a harmonic well and single Gaussian barrier potential. The atomtronic battery is analogous to the electric battery, where the electric potential is now defined by the chemical potential and the current is now defined by the atomic flux that depends on the barrier height of the single Gaussian potential barrier. For further discussion on the atomtronic battery see [2] and [11].

The single Gaussian barrier potential allows for transmission at energies lower than the peak potential height, i.e. tunneling occurs. Increasing the height of the Gaussian barrier has a near one-to-one relation with the first tunneling energy and keeps the general shape and slope of the transmission spectrum. Unlike the relationship with the height of the Gaussian barrier,  $\sigma$  of the Gaussian barrier does not scale linear with the tunneling energies. Changing  $\sigma$  changes the shape and slope of the transmission spectrum, and an increase in  $\sigma$  steepens the slope of transmission for tunneling and transmitting energies and shrinks the range of those energies as well. Knowing how transmission depends on the barrier characteristics allows for the single Gaussian barrier potential to be used as a spectrum analyzer. In the limiting cases where  $\sigma \ll 1$  and  $\sigma \gg 1$ , the transmission spectrum of the single Gaussian barrier approaches the transmission spectrum for a delta potential and the classical transmission spectrum for a step potential, as expected. The Mathematica notebook for the single Gaussian barrier potential is given in Appendix C.

## Chapter 4

### The Double Rectangular Potential

#### 4.1 The double rectangular barrier potential

For a double barrier potential, e.g. a double rectangular barrier or double Gaussian barrier, the quantum mechanical phenomenon of resonant tunneling occurs. This tunneling occurs when the incoming energy of the wavepacket is resonant with the eigenenergies of the inner potential well. Thus allowing the wave packet to be fully transmitted through the potential even when the incoming energy is less than that of the potential's height. In the impedance method, the energy condition for resonance is  $\zeta_{Left}(R_c) = \zeta_{Right}(R_c)$ , where  $R_c$  is the chosen center reference plane [1]. In effect, the left looking impedance is equivalent to the right looking impedance from a chosen center reference plane. See Appendix D. For the double rectangular case the potential is described by Eq. (4.1),

$$v(x) = \begin{cases} v_{01} & : -b \leq x \leq -a & \text{Region II} \\ 0 & : -a < x < a & \text{Region III} \\ v_{02} & : a \leq x \leq b & \text{Region IV} \\ 0 & : |x| > b & \text{Region I and V} \end{cases} . \quad (4.1)$$

Here, we choose  $b = 0.6455 \mu\text{m}$ ,  $a = 0.3125 \mu\text{m}$ ,  $v_{01} = 30 \text{ kHz}$ , and  $v_{02} = 30 \text{ kHz}$ . Figure 4.1 shows the potential barrier, described in Eq. (4.1), with the different regions labeled for clarity. The double rectangular barrier potential is an analytically solvable problem. The solution is found

by separating the problem into five different regions and calculating the respective periodic solutions to the Schrödinger equation, as denoted in Fig. 4.1 as region *I*, *II*, *III*, *IV*, and *V*. The boundary conditions require that the periodic functions and their derivatives are continuous over the boundary.

The analytical reflection/transmission is derived using the one-dimensional time-independent Schrödinger equation  $\frac{-i\hbar}{2m} \frac{d^2\psi}{dx^2} + v(x)\psi(x) = E\psi(x)$  as shown in Eq. (4.2). Although solvable, the calculation is quite involved and is generally done numerically as in [4]. References [12] and [13] have published analytical transmission coefficient functions, but they are not in agreement with each other nor with Eq. (4.2), Eq. (4.3), and Eq. (4.4). The analytically solved transmission probability is

$$T = 1 - \left| \frac{B}{A} \right|^2, \quad (4.2)$$

where

$$\begin{aligned} B = & e^{-2ikb} \left[ \left( \frac{k^2}{\kappa_2} + \kappa_2 \right) e^{2ika} \cosh(\kappa_1 \Delta) \sinh(\kappa_2 \Delta) \right. \\ & + \left( \frac{k^2}{\kappa_1} + \kappa_1 \right) e^{-2ika} \sinh(\kappa_1 \Delta) \cosh(\kappa_2 \Delta) \\ & \left. + \left[ \left( \frac{k^3}{\kappa_1 \kappa_2} - \frac{\kappa_1 \kappa_2}{k} \right) \sin(2ka) + ik \left( \frac{\kappa_1}{\kappa_2} - \frac{\kappa_2}{\kappa_1} \right) \cos(2ka) \right] \sinh(\kappa_1 \Delta) \sinh(\kappa_2 \Delta) \right], \end{aligned} \quad (4.3)$$

and

$$\begin{aligned} A = & e^{-2ika} \left[ \left( \frac{k^2}{\kappa_2} - \kappa_2 \right) \sinh(\kappa_2 \Delta) \cosh(\kappa_1 \Delta) \right. \\ & \left. + \left( \frac{k^2}{\kappa_1} - \kappa_1 \right) \sinh(\kappa_1 \Delta) \cosh(\kappa_2 \Delta) - 2ik \cosh(\kappa_2 \Delta) \cosh(\kappa_1 \Delta) \right] \\ & + \left[ \left( \frac{k^3}{\kappa_1 \kappa_2} + \frac{\kappa_1 \kappa_2}{k} \right) \sin(2ka) - ik \left( \frac{\kappa_1}{\kappa_2} + \frac{\kappa_2}{\kappa_1} \right) \cos(2ka) \right] \sinh(\kappa_2 \Delta) \sinh(\kappa_1 \Delta), \end{aligned} \quad (4.4)$$

where  $k = \sqrt{\frac{2m}{\hbar^2} E}$  is the propagation constant in region *I*, *III*, and *V*,  $\kappa_1 = \sqrt{\frac{2m}{\hbar^2} (v_{01} - E)}$  is the propagation constant in region *II*, and  $\kappa_2 = \sqrt{\frac{2m}{\hbar^2} (v_{02} - E)}$  is the propagation constant in region *IV*, with  $\Delta = a - b$ . This derivation was done by hand and confirmed using Mathematica. Despite disagreement with references [12] or [13], it is in close agreement with the eigenenergies calculated by the impedance method and is closer to the impedance method's transmission spectrum than

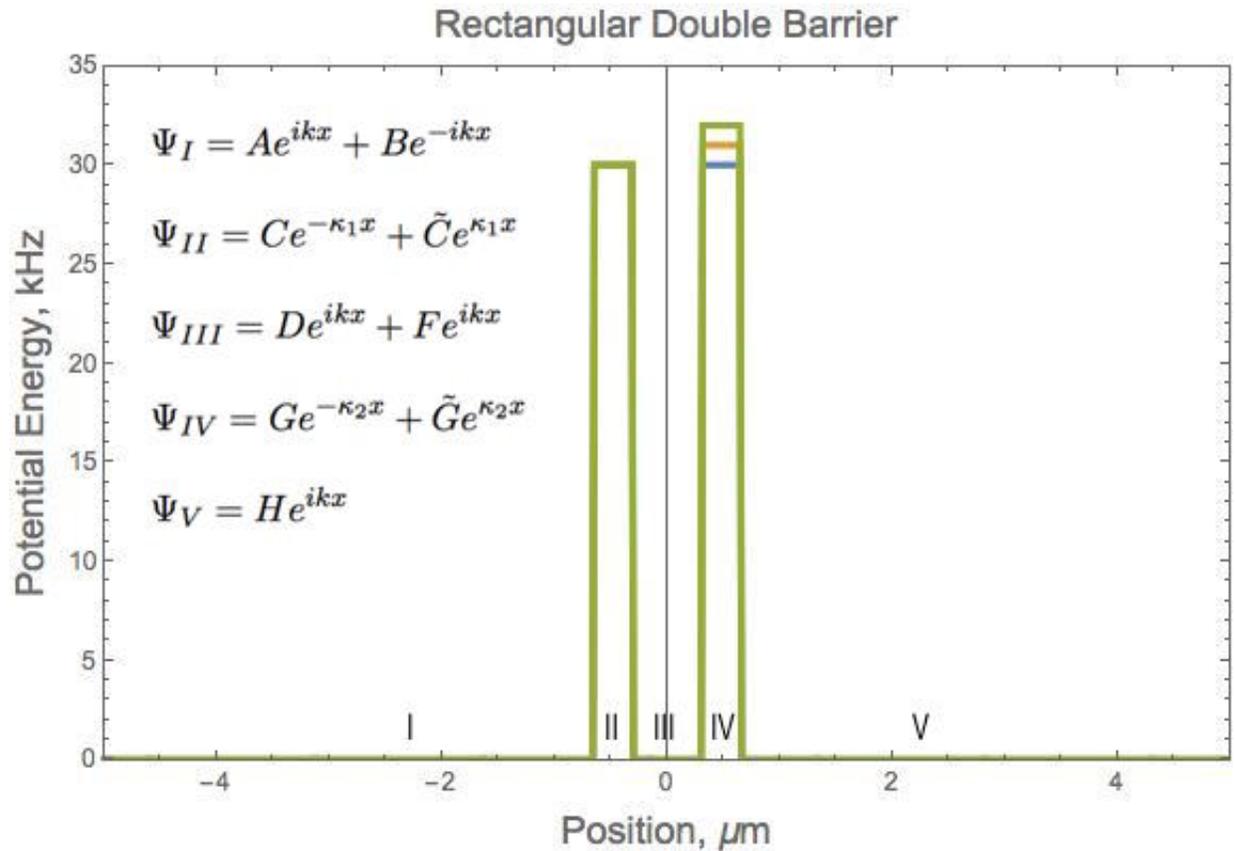


Figure 4.1: The double rectangular barrier potential whose potential height is  $v_{02} = \{30 \text{ kHz (Blue solid line), } 31 \text{ kHz (Orange solid line), } 32 \text{ kHz (Green solid line)}\}$  and barrier width is  $0.333 \mu\text{m}$  with a separation distance of  $0.625 \mu\text{m}$ . The wavepacket is described by the superposition of  $\Psi_I$ ,  $\Psi_{II}$ ,  $\Psi_{III}$ ,  $\Psi_{IV}$ , and  $\Psi_V$  where the subscript relates to the region.  $\Psi_I$  is the superposition of the incident and reflected wave,  $\Psi_{II}$  is the superposition of the tunneling waves,  $\Psi_{III}$  is the superposition of the tunneled wave and reflected tunneled wave,  $\Psi_{IV}$  is again a superposition of tunneling waves, and  $\Psi_V$  is the transmitted wave.

references [12] and [13]. The analytical solutions presented in [12], [13], as well as the solutions given in Eq. (4.2), Eq. (4.3), and Eq. (4.4) are compared in Fig. 4.2

## 4.2 Applying the impedance method to the double rectangular barrier potential

The transmission spectrum for the symmetric double rectangular barrier potential is illustrated in Fig. 4.3. The resonances are shown by peaks of full transmission at incoming energies less than the barrier height. The peaks occur when the incoming energy is matched to the quasi-bound-state solutions of the middle well between the two barriers. When the incoming energy is matched to one of these resonant energies the wavefunction between the two barriers is peaked due to constructive interference from the reflected waves off both interior barriers. A depiction of the wavefunction between the barriers is shown in Fig. 4.7 and discussed in further detail later. When the incoming wavepacket is not resonant the transmission is as though the well does not exist, i.e., transmission is less than unity for energies less than the barrier height. The double barrier acts as an energy filter, allowing the resonant wavepacket through both barriers with full transmission. All other energies less than the height of the potential are reflected off of the barrier. This filtering is illustrated in Fig. 4.3. Incoming energies less than that of  $v_{01}$  and on resonance fully transmit, where other off resonant energies, will fully reflect at the barrier. A higher energy density around the lower energy resonances is needed for the impedance method transmission spectrum to register and show full transmission. This is due to the resonant energies having a resonant width that gets smaller and smaller for the lower resonant energies, where the resonant width approaches a delta function. This is why the lower resonant spikes on the spectrum are not shown to be at unity transmission, when in reality they are.

The eigenenergies of a double barrier potential can be found by use of the impedance method as discussed in references [1] and [6]. The condition for an eigenenergy, defined in Eq. (2.3), is that the left-looking impedance, defined in Eq. (2.6), is equivalent to the right-looking impedance from a given reference plane between the two barriers. Specifically the program written to find the

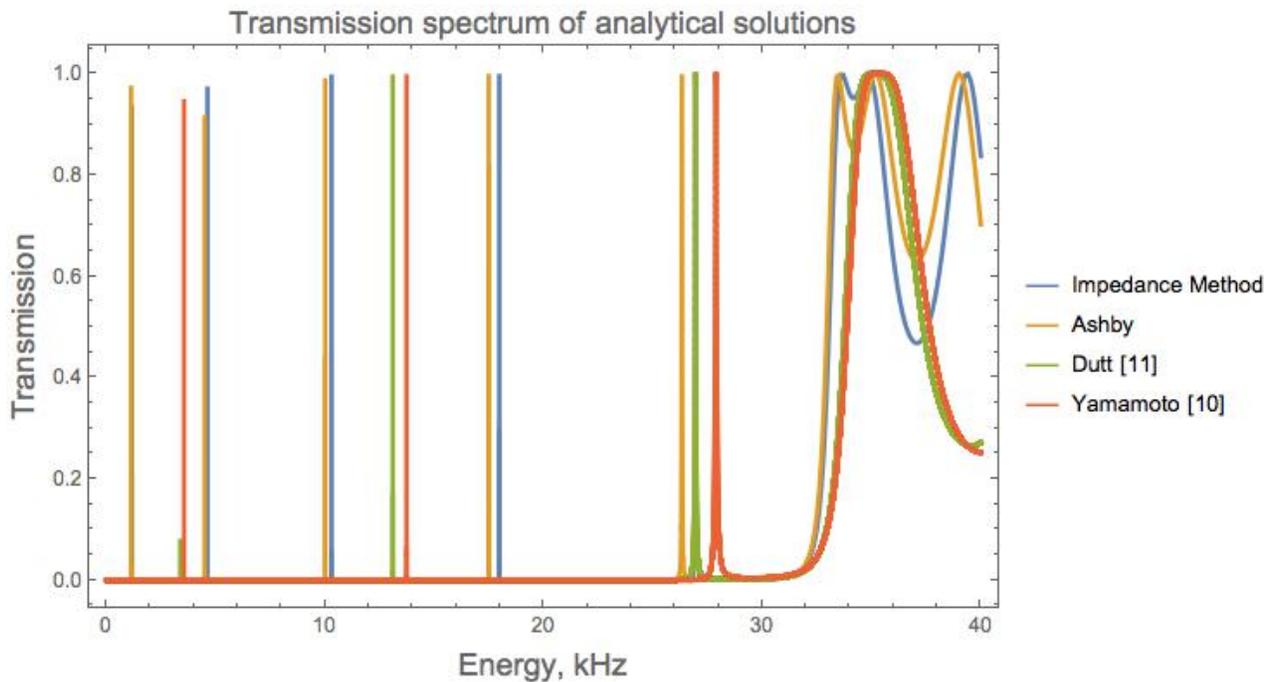


Figure 4.2: A comparison of the impedance method spectrum (Blue solid line), analytical solution from Eq. (4.2), Eq. (4.3), and Eq. (4.4) (Orange solid line), and analytical solutions done in [13] (Green solid line), and [12] (Red solid line). Notice that the both the impedance method transmission and transmission from Eq. (4.2), Eq. (4.3), and Eq. (4.4) show five resonant energies, but the transmission spectra of [12] and [13] only show three resonant energies. The general structure of [12] and [13] transmission spectra are similar and some what in agreement with each other, while the impedance method and Eq. (4.2), Eq. (4.3), and Eq. (4.4) are in closer agreement to each other. Interestingly the transmission spectra of [12] and [13] appear to have a somewhat uniform spacing of the resonant energies, unlike what we would expect from the bound-state solutions of the finite square well.

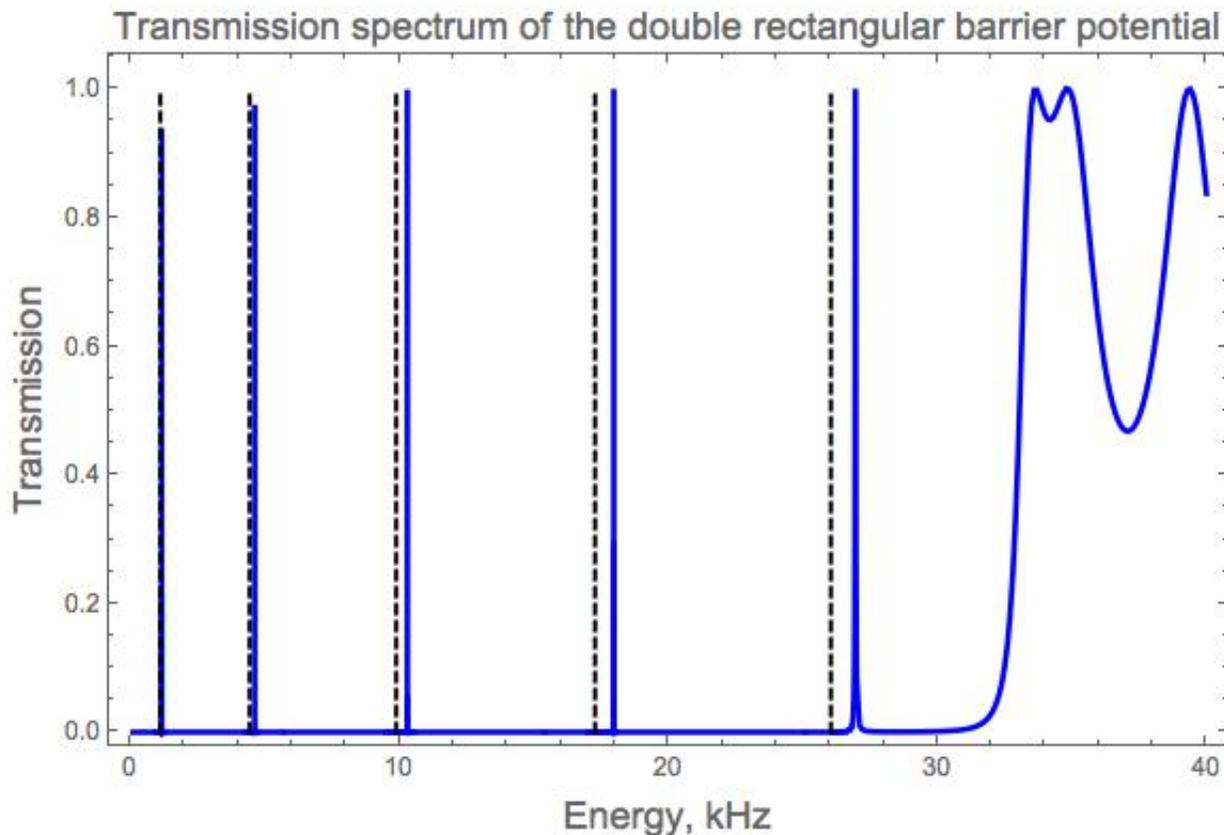


Figure 4.3: The transmission spectrum for the double rectangular barrier potential described by Eq. (4.1). The blue solid line is the transmission spectrum calculated by the impedance method, and the vertical black dashed lines are the eigenenergies calculated by the impedance method. The peaks are the resonant tunneling energies where full transmission is achieved. Notice though that the two lowest resonant energies are not at unity transmission in this plot, this is due to numerical memory limitation. The range of incoming energies is not at a high enough density around the two lowest peaks causing the impedance method to not see unity transmission. The resonant energies from this plot are 1.16198 kHz, 4.62346 kHz, 10.2976 kHz, 17.9696 kHz, and 26.9463 kHz.

eigenenergies of the double barrier potentials, rectangular and Gaussian, sets a reference plane to be centered between the two inside walls of the barriers and evaluates the impedance from the left (right) end of the potential function to the reference plane,  $\zeta_L$  ( $\zeta_R$ ). The eigenenergies satisfy the condition  $\left| \frac{\zeta_L}{\zeta_R} \right| = 1$ . The energies that satisfy this condition for the specific potentials in Fig. 4.1 are given in Table 4.1.

Eigen-Energies in kHz					
30. kHz	1.11401	4.43412	9.883	17.2732	26.039
31. kHz	1.11628	4.44363	9.90643	17.3226	26.1509
32. kHz	1.11844	4.45268	9.92868	17.369	26.2519

Table 4.1: The eigenenergies calculated using the impedance method for a double rectangular barrier whose potential height is {30 kHz, 31 kHz, 32 kHz}, width is  $0.333 \mu\text{m}$  with a separation distance of  $0.625 \mu\text{m}$ . Increasing the height of the second barrier increases the eigenenergies slightly. The eigenenergies for the double rectangular barrier whose barriers have heights both at 30 kHz are also shown in Fig. 4.3 and Fig. 4.4.

A comparison of the transmission spectrum evaluated using the impedance method, the derived analytical solution, and the evaluated eigenenergies is shown in Fig. 4.4. Notice that the lowest resonant energy all three methods are in agreement with each other, but at higher energies the analytical resonances are shifted slightly to the left of those found by the impedance method. The shift between the analytical solution's spectrum and the impedance method's spectrum is due to the numerical approximation of the infinite slope of the step potential. The calculated eigenenergies, via the reference plane method, are slightly shifted to the left of the analytical resonances, where the shift becomes greater for each higher resonance. The eigenenergies agree more with the analytical solution as compared to the impedance method's spectrum, this is interesting because the eigenenergies and the spectrum are both using the same general method of numerically calculated impedances. The difference in agreement to the analytical solution is again explained by the numerical approximation of the infinite slope of the boundaries of the potential function.

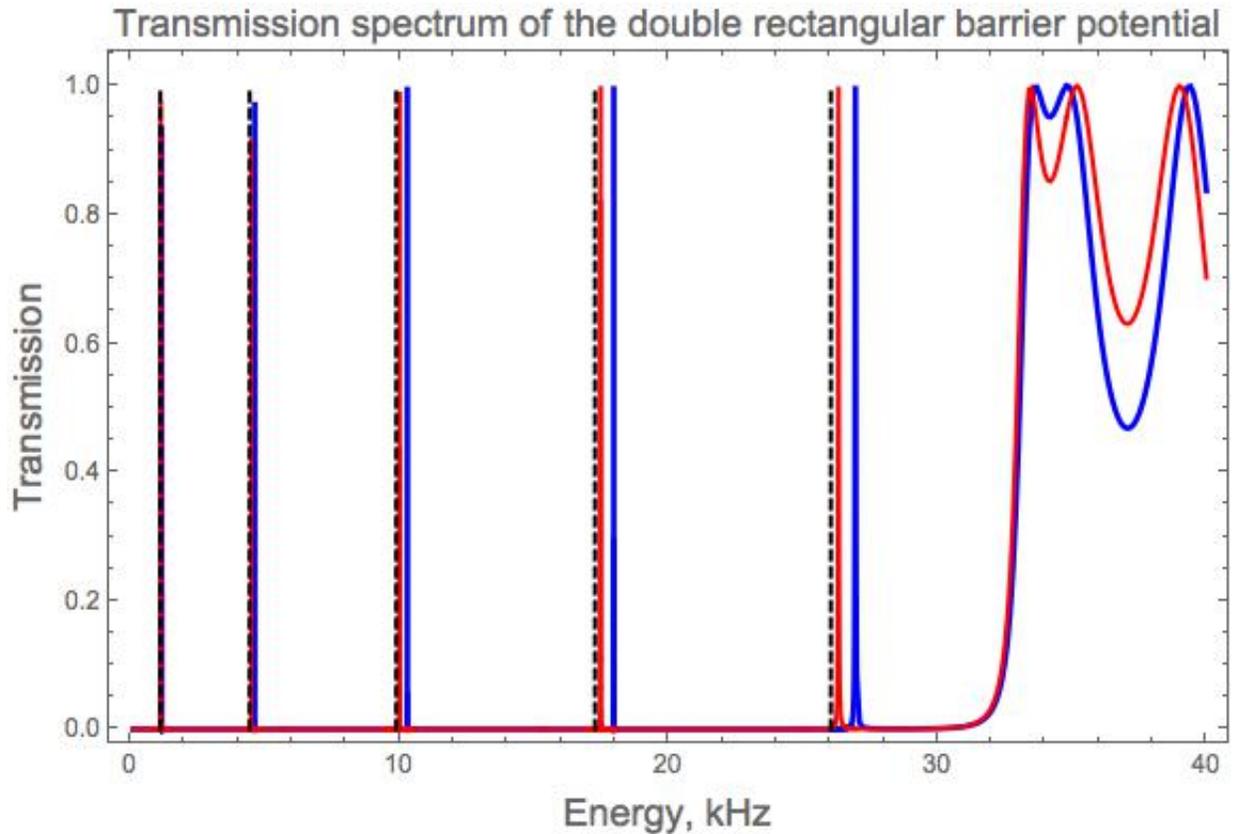


Figure 4.4: A compilation of transmission spectrums of the double rectangular barrier with  $v_{01} = v_{02} = 30$  kHz, width =  $0.333 \mu\text{m}$ , and separation distance =  $0.625 \mu\text{m}$ . The blue solid line is the transmission spectrum from the impedance method also shown in Fig. 4.3. The red solid line is the analytical solution for the transmission spectrum and the vertical black dashed lines are the values of the eigenenergies from the impedance method. Notice how the eigenenergies from the impedance method agree more with the analytical solution than the transmission spectrum from the impedance method does. All three methods are in agreement for the first quasi-bound state, but start to shift from one another as they near the height of the potential. This shift is due to the numerical approximation of the infinite slope of the barriers.

The impedance method spectrum encounters an infinite slope four times while the eigenenergy evaluation does the numerical approximation for the infinite slope twice, so the error in the spectrum is higher than the error in the evaluation of the eigenenergies. Although this error has visible effects at the higher energy resonances, it provides a very useful way to optimize the transmission spectrum. Using the eigenenergy package with an energy range of uniform density allows for the discovery of resonant energies that would not have otherwise shown in the spectrum. Therefore an optimized energy range can have a higher density of energies around the evaluated resonant energies, and these can be input into the impedance method package allowing for the lower resonant energies to be visible in the transmission spectrum. This is described further in chapter five.

A comparison of the transmission spectra for the double rectangular barrier potential where the second barrier has a height that is larger than the first is shown in Fig. 4.5. Notice that an increase in the second barrier height slightly increases the resonant energies, this can also be seen in Table 4.1. Increasing the separation of the barriers has a more noticeable and interesting effect on the resonant energies, this is shown in Fig. 4.6 and Table 4.2. As expected, increasing the separation of the barriers increases the number of resonant energies. From Fig. 4.6, one can see that as the separation is increased, the spacing between the resonant energies decreases and becomes more uniform, with the spacing between the higher resonances being larger than the spacing between the lower resonances. This type of behavior is expected in light of the finite square well bound-state energies defined by [7]

$$E_n = \frac{2\hbar^2 v_n^2}{m(2a)}, \quad (4.5)$$

where  $2a$  is the length of the well, and  $v_n$  is the dimensionless variable  $v_n = k_n a$ ,  $k_n$  being the propagation constant inside the well. Table 4.2 shows the spacing between energy resonances up to order  $10^{-1}$  as the separation between the barriers is increased.

The wavepacket is represented in Fig. 4.7 for the symmetric double rectangular barrier potential whose two barriers have equal height  $v_{01} = v_{02} = 30$  kHz, barrier width of  $0.333 \mu\text{m}$  and separation of  $0.625 \mu\text{m}$ . The incoming wave has an incident amplitude coefficient of unity and

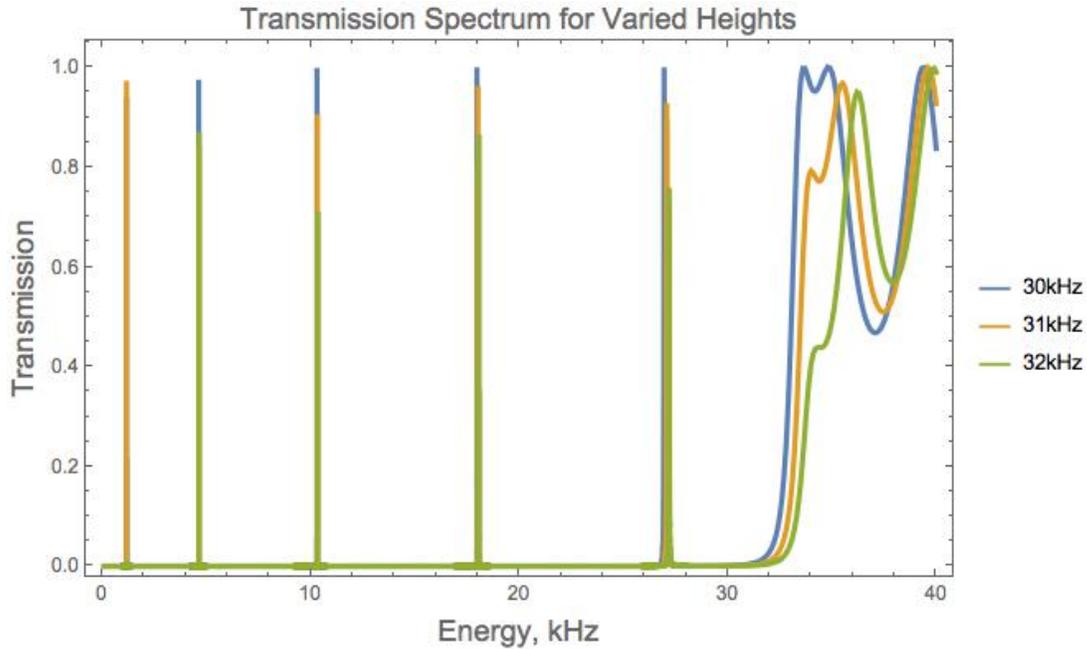


Figure 4.5: The transmission spectrum for the double rectangular barrier potential whose second barrier height is  $v_{02} = \{30 \text{ kHz (Blue solid line), } 31 \text{ kHz (Orange solid line), } 32 \text{ kHz (Green solid line)}\}$ , width of  $0.333 \mu\text{m}$ , and separation of  $0.625 \mu\text{m}$ . An increase in the second barrier height only slightly increases the resonant energies.

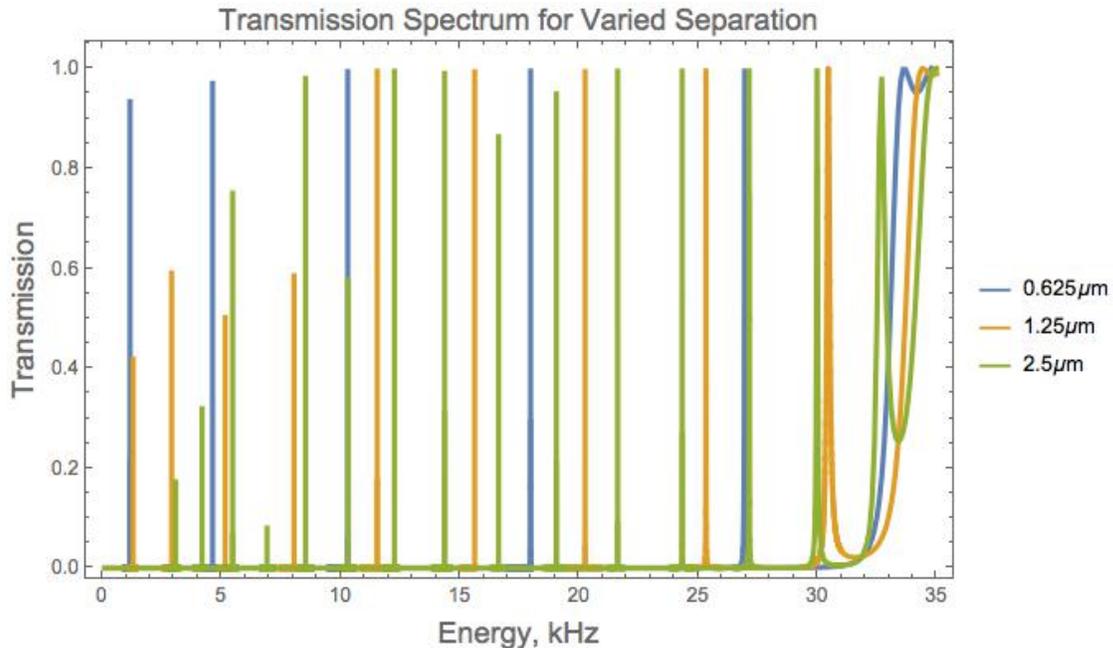


Figure 4.6: The transmission spectrum for the double rectangular barrier potential whose second barrier height is  $sep = \{0.625 \mu\text{m (Blue solid line), } 1.25 \mu\text{m (Orange solid line), } 2.5 \mu\text{m (Green solid line)}\}$ , width of  $0.333 \mu\text{m}$ , and height of  $v_{01} = v_{02} = 30 \text{ kHz}$ . Notice that an increase in separation increases the number of resonant energies.

Eigen-Energies in kHz, $v01=v02=30\text{kHz}$													
0.625 $\mu\text{m}$	1.11	4.43	9.88	17.27	26.04								
1.25 $\mu\text{m}$	1.27	2.84	5.05	7.86	11.29	15.29	19.83	24.81	29.93				
2.5 $\mu\text{m}$	3.04	4.14	5.4	6.83	8.43	10.19	12.11	14.2	16.44	18.84	21.38	24.06	26.85

Eigen-Energy differences in kHz, $v01=v02=30\text{kHz}$													
0.625 $\mu\text{m}$	3.3	5.4	7.4	8.8									
1.25 $\mu\text{m}$	1.6	2.2	2.8	3.4	4.	4.5	5.	5.1					
2.5 $\mu\text{m}$	1.1	1.3	1.4	1.6	1.8	1.9	2.1	2.2	2.4	2.5	2.7	2.8	

Table 4.2: The eigenenergies for a double rectangular barrier potential with the second barrier held fixed at 30 kHz and a width of 0.333  $\mu\text{m}$  for different values of separation and the differences or spacing between the resonant energies. Notice that as separation is increased so do the number of resonant energies. Also the spacing between those energies is similar to the bound-state solutions of the finite square well. As the separation is increase the spacing between the resonant energies is decreased, yet still follows the finite square well pattern, where lower resonant energies are spaced closer together than the higher resonant energies.

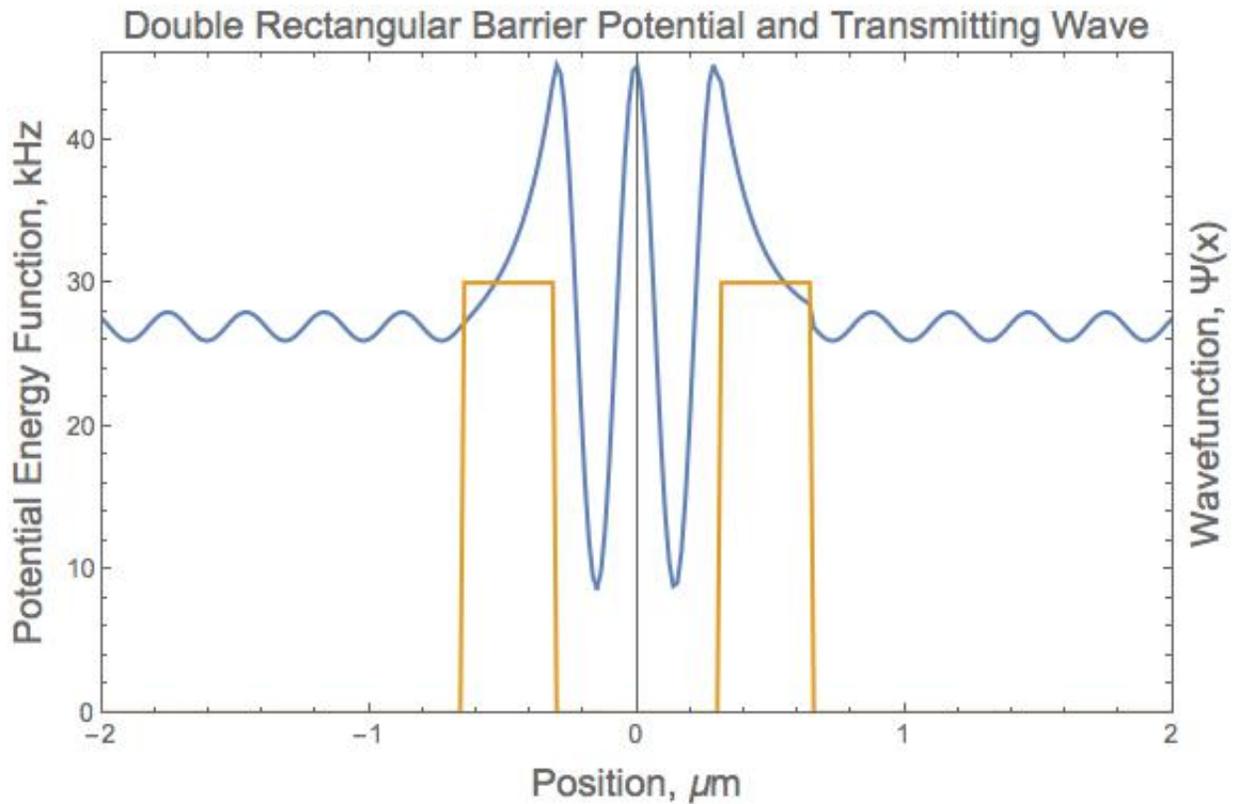


Figure 4.7: A representation of a wavepacket interacting with the double rectangular barrier potential. The wavepacket has an incoming energy on resonance at  $E = 26.95$  kHz thus having a transmission probability of unity. It is important to note that this is a representation of the wavefunction and the waves amplitude and frequency are not denoted by the axes of the figure. Notice that between the barriers the wavefunction is peaked due to the constructive interference between the reflected waves off the two inner barrier surfaces. The build up of wave amplitude allows for full transmission of the wavepacket through the second barrier. In the barriers themselves exponential decay occurs, as expected. The wavefunction on the left end of the barrier is a superposition of the incident and reflected waves. The reflected wave has a complex component from the reflected wave within the barrier well, but only the real part of the wave is shown here.

a reflection amplitude coefficient calculated by the impedance method, with an incoming energy at the resonant energy of  $E = 26.95$  kHz. As expected, the wavefunction peaks between the two barriers with exponential decay occurring in the two barriers, and the real part of the transmission amplitude coefficient is unity. The amplitude coefficients were calculated using the derived analytical coefficients which depend on the reflection amplitude coefficient calculated using the impedance method. The derived coefficients are,

$$\begin{aligned} C &= \frac{1}{2}e^{-\kappa_1 b} \left[ \frac{B}{A} e^{ikb} \left( \frac{ik}{\kappa_1} + 1 \right) - e^{-ikb} \left( \frac{ik}{\kappa_1} - 1 \right) \right] \\ \tilde{C} &= \frac{1}{2}e^{\kappa_1 b} \left[ e^{-ikb} \left( 1 + \frac{ik}{\kappa_1} \right) + \frac{B}{A} e^{ikb} \left( 1 - \frac{ik}{\kappa_1} \right) \right] \end{aligned} \quad (4.6)$$

$$\begin{aligned} D &= \frac{1}{2}e^{ika} \left[ C e^{\kappa_1 a} \left( 1 - \frac{\kappa_1}{ik} \right) + \tilde{C} e^{-\kappa_1 a} \left( 1 + \frac{\kappa_1}{ik} \right) \right] \\ F &= \frac{1}{2}e^{-ika} \left[ C e^{\kappa_1 a} \left( 1 + \frac{\kappa_1}{ik} \right) + \tilde{C} e^{-\kappa_1 a} \left( 1 - \frac{\kappa_1}{ik} \right) \right] \end{aligned} \quad (4.7)$$

$$\begin{aligned} G &= \frac{1}{2}e^{\kappa_2 a} \left[ D e^{ika} \left( 1 - \frac{ik}{\kappa_2} \right) + F e^{ika} \left( 1 + \frac{ik}{\kappa_2} \right) \right] \\ H &= \sqrt{1 - \left| \frac{B}{A} \right|} \end{aligned} \quad (4.8)$$

where  $C$ ,  $\tilde{C}$ ,  $D$ ,  $F$ ,  $G$ , and  $H$  are the amplitude coefficients denoted in Fig. 4.1. See Appendix E for further detail on the coefficients and Appendix F for the reflection amplitude package.

### 4.3 Chapter conclusion

The double rectangular barrier potential gives rise to resonant tunneling energies; full transmission is possible for incoming energies less than the barrier height when the incoming energy is at a resonant energy. When the incoming energy is not on resonance the wavepacket is reflected and the transmission spectrum is that of a box barrier potential, as if the well does not exist. The double rectangular barrier potential can be used as an energy filter, where full transmission only occurs at resonant energies. Increasing the separation of the barriers allows for more energies to pass through the filter, which was expected from the finite square well energy condition, Eq. (4.5).

The lower resonant energies are spaced closer together and as the resonant energies approach the height of the barrier the spacing between them increases, this behavior is similar to that of the finite square well bound-state energies. The Mathematica notebook for the double rectangular barrier potential is shown in Appendix E.

## Chapter 5

### The Double Gaussian Potential

The atomtronics project combines optical and magnetic trapping forces to produce a triple-well potential, as illustrated in Fig. 5.1, to develop an atomic transistor. The triple-well potential of the atomtronic transistor is a superposition of a harmonic well and a double Gaussian barrier potential, where the three wells are analogous to the three terminals of the electronic transistor. With an atom current flowing out of the source well, atoms become trapped in the gate well producing a condensate. The positive chemical potential of the condensate alters the original central potential well, shifting the energy levels of the gate well. A height difference between the barriers allows for control of the chemical potential and temperature of the gate condensate relative to the source condensate. The ability to manipulate the behavior in such a way demonstrates the potential for an operational atomtronic circuit [2]. The previous work done in reference [2] was a classical study of transport dynamics where the thermal current regime dominates. Application of the impedance method to the atomtronics transistor potential could be used to include quantum mechanical effects like tunneling into those transport dynamics already studied.

#### 5.1 The atomtronic transistor potential

The atomtronics triple-well potential is simplified and only the longitudinal direction is treated by approximating the experimental potential as a double Gaussian barrier potential centered at the origin. The approximate double Gaussian barrier potential is defined by

$$v(x, u_{02,j}, x_{\text{sep},k}, \sigma_n) = u_{01} \exp\left(-2\left(\frac{x + \frac{x_{\text{sep},k}}{2}}{\sigma_n}\right)^2\right) + u_{02,j} \exp\left(-2\left(\frac{x - \frac{x_{\text{sep},k}}{2}}{\sigma_n}\right)^2\right), \quad (5.1)$$

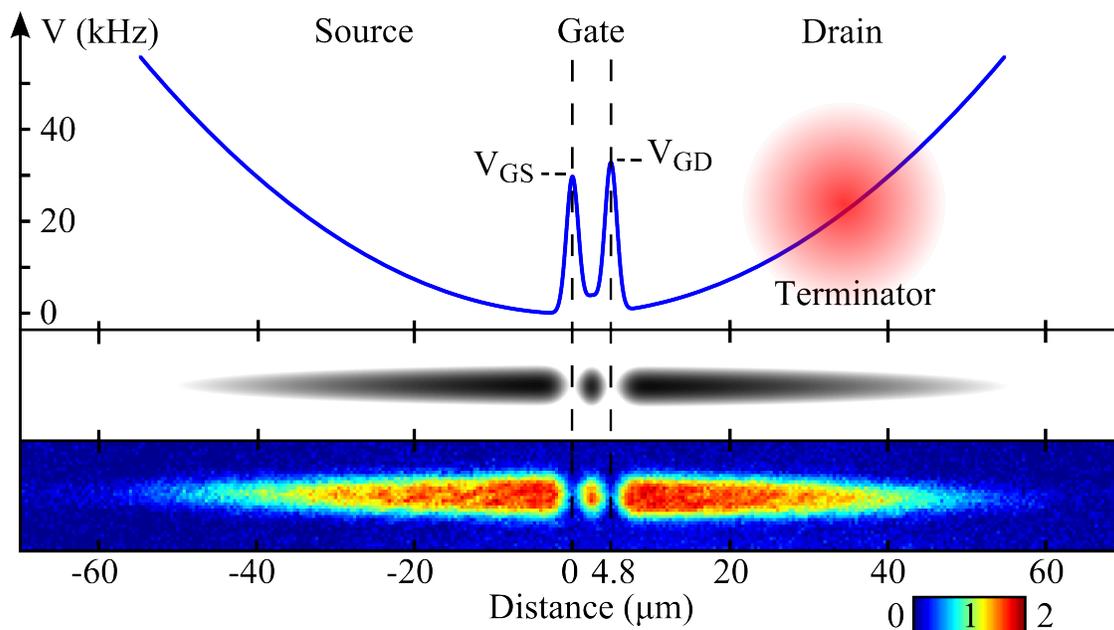


Figure 5.1: Atomtronic transistor triple-well potential showing the source, gate and drain wells as well as the relative height difference between the source and drain. The terminator beam removes the population in the drain via resonant light and replenishes them back into the source. Image credit: [2]

where  $u_{01}$  is the height of the first Gaussian barrier and is fixed,  $u_{02,j}$  is the height of the second Gaussian barrier and can be varied,  $x_{sep,k}$  is the separation between the two Gaussian barriers and can be varied,  $\sigma_n$  is the variance of the Gaussian barriers and can be varied. The values of  $u_{01}$ ,  $u_{02,j}$ ,  $x_{sep,k}$ , and  $\sigma_n$  studied are similar to those of the atomtronics experiment. The full width half maximum (FWHM) of the Gaussian barriers is defined by  $\Gamma = \sqrt{\frac{1}{2} \ln 2} \sigma_n$ . Increasing  $x_{sep}$  will increase the distance between the two barriers as shown in Fig. 5.2, increasing the number of resonant energies that can be found. An increase in  $\sigma$  will increase the FWHM, shifting the resonant energies to higher values because the width of the barriers has increased and so has decrease the separation at a given energy level. Looking back at the double rectangular barrier potential we would expect that an increase in  $u_{02}$  will also shift the resonances higher in energy. The double Gaussian barrier potentials for a varied variance and varied height of the second barrier are shown in Fig. 5.3 and Fig. 5.4 respectively.

## 5.2 Applying the impedance method to the double Gaussian barrier potential

Knowing the quasi-bound states for the double Gaussian barrier potential is important to construct an optimized atomtronics transistor. Knowledge of the quasi-bound states would allow control over the coupling between the transistor potential and the dipole moment of the condensate in the gate well. By exciting the dipole mode of the gate well condensate, interatomic interactions with the higher lying resonant states causes correlations between the gate condensate and the output current of the transistor [2].

There are many different numerical methods for solving Schrödinger's equation, one of which is the impedance method, which is used to generate the transmission spectrum and in the double barrier case the eigenenergies. The calculated eigenenergies for a potential barrier with variance  $\sigma = 1.6 \mu\text{m}$ , separation  $x_{sep} = 4.8 \mu\text{m}$  and second barrier height  $u_{02} = 30 \text{ kHz}$ , as shown in Fig. 5.4 and Fig. 5.12, are compared to the eigenenergies generated by solving Schrödinger's equation directly using the finite difference method, illustrated in Fig. 5.6. The figure is a one to one plot of the generated eigenenergies from the finite difference method and the impedance method.

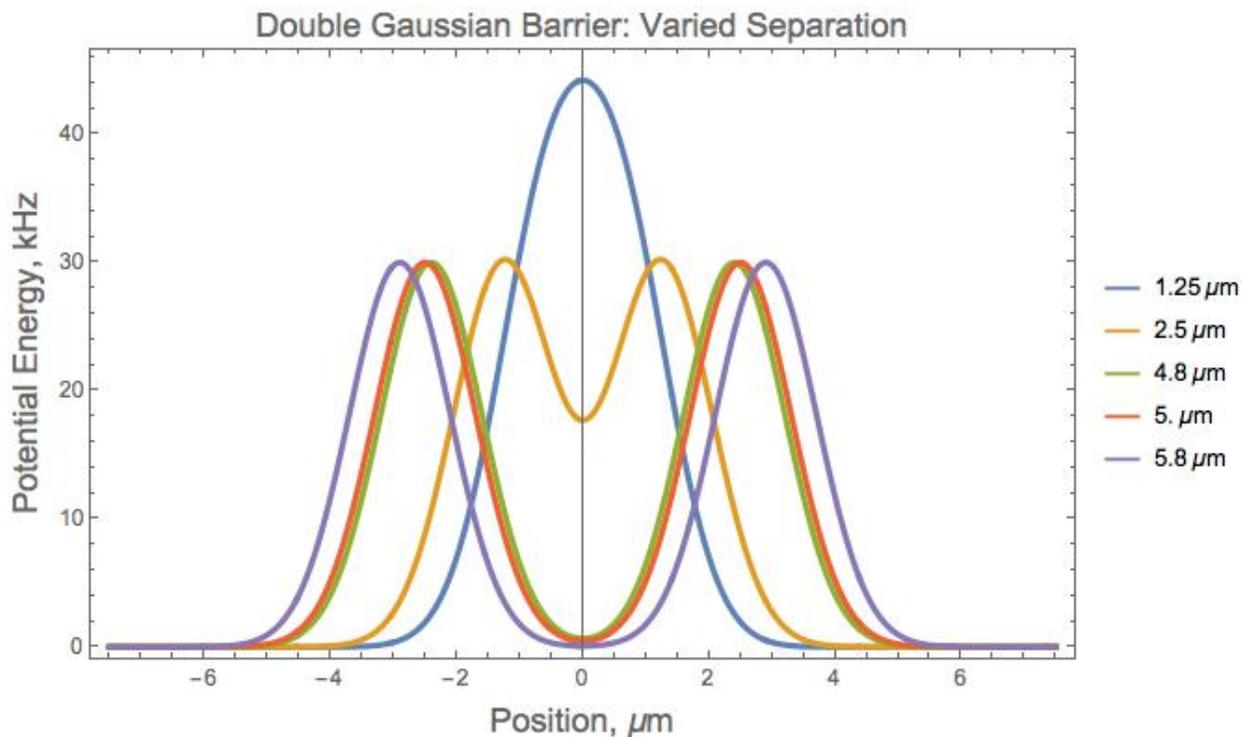


Figure 5.2: The double Gaussian barrier potential for  $\sigma = 1.6 \mu\text{m}$ ,  $u_{02} = 30 \text{ kHz}$  and the separation of the barriers  $x_{sep} = \{1.25 \mu\text{m}(\text{Blue solid line}), 2.5 \mu\text{m}(\text{Orange solid line}), 4.8 \mu\text{m}(\text{Green solid line}), 5.0 \mu\text{m}(\text{Red solid line}), 5.8 \mu\text{m}(\text{Purple solid line})\}$ . For  $x_{sep} = 1.25 \mu\text{m}$ (Blue solid line) one expects the transmission spectrum to be similar to that of a single Gaussian barrier and so  $x_{sep} = 1.25 \mu\text{m}$  is neglected, expect for special cases. As the separation is increased we would expect that the number of resonant energies to increase as well. For too small a separation,  $x_{sep} = 1.25 \mu\text{m}$  we have a single Gaussian barrier potential. As the separation of the barriers is increased the well begins to flatten at the bottom.

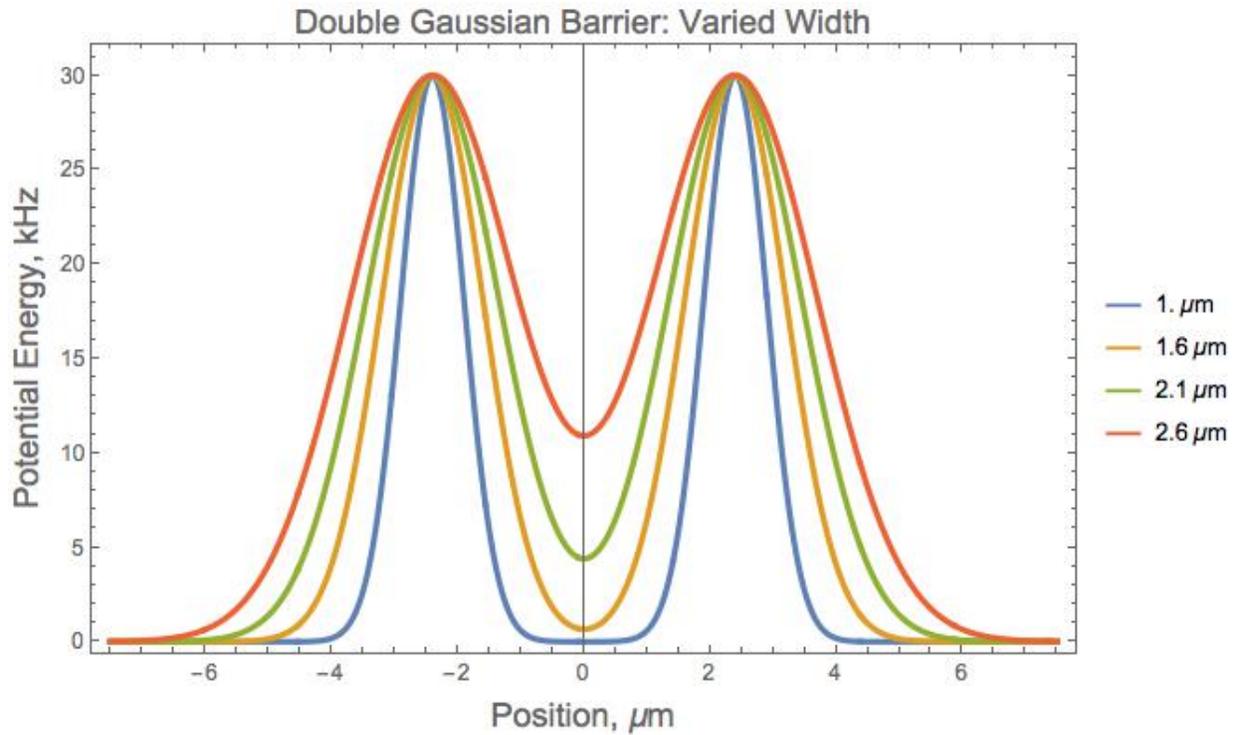


Figure 5.3: The double Gaussian barrier potential for  $x_{sep} = 4.8 \mu\text{m}$ ,  $u_{02} = 30 \text{ kHz}$  and the variance of the barriers  $\sigma = \{1.0 \mu\text{m}(\text{Blue solid line}), 1.6 \mu\text{m}(\text{Orange solid line}), 2.1 \mu\text{m}(\text{Green solid line}), 2.6 \mu\text{m}(\text{Red solid line})\}$ . For  $\sigma = 1.0 \mu\text{m}$ (Blue solid line) one expects the highest degree of resonant tunneling and a more uniform distribution of the resonant energies. Notice that the double Gaussian barrier potential with the smallest variance,  $\sigma = 1.0 \mu\text{m}$  has a well that is flattened at the bottom.

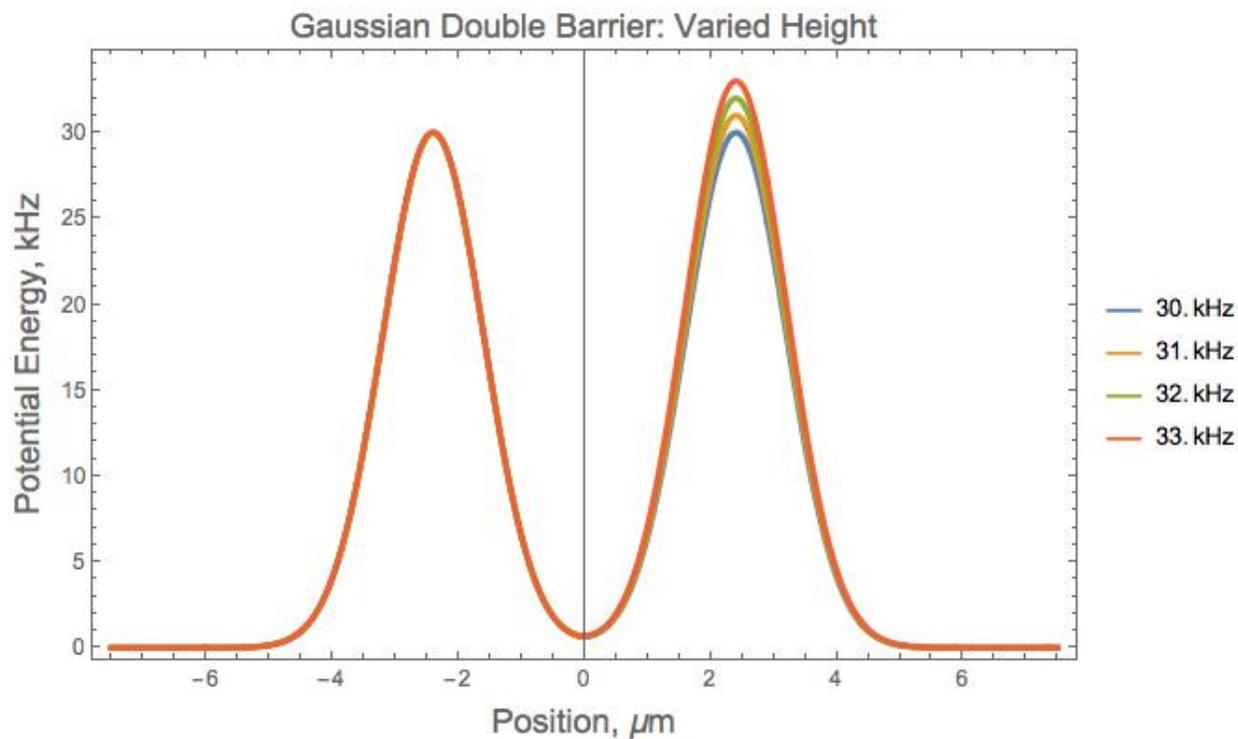


Figure 5.4: The double Gaussian barrier potential for  $x_{sep} = 4.8 \mu\text{m}$ ,  $\sigma = 1.6 \mu\text{m}$  and the height of the second barrier  $u_{02} = \{30 \text{ kHz}(\text{Blue solid line}), 31 \text{ kHz}(\text{Orange solid line}), 32 \text{ kHz}(\text{Green solid line}), 33 \text{ kHz}(\text{Red solid line})\}$ . Increasing the height of the second barrier one expects some type of shift in the resonant energies. Changing the height of the second barrier makes the double Gaussian barrier potential asymmetric about the origin, which has a  $T < 1$  on resonance effect on the transmission of a given wavepacket.

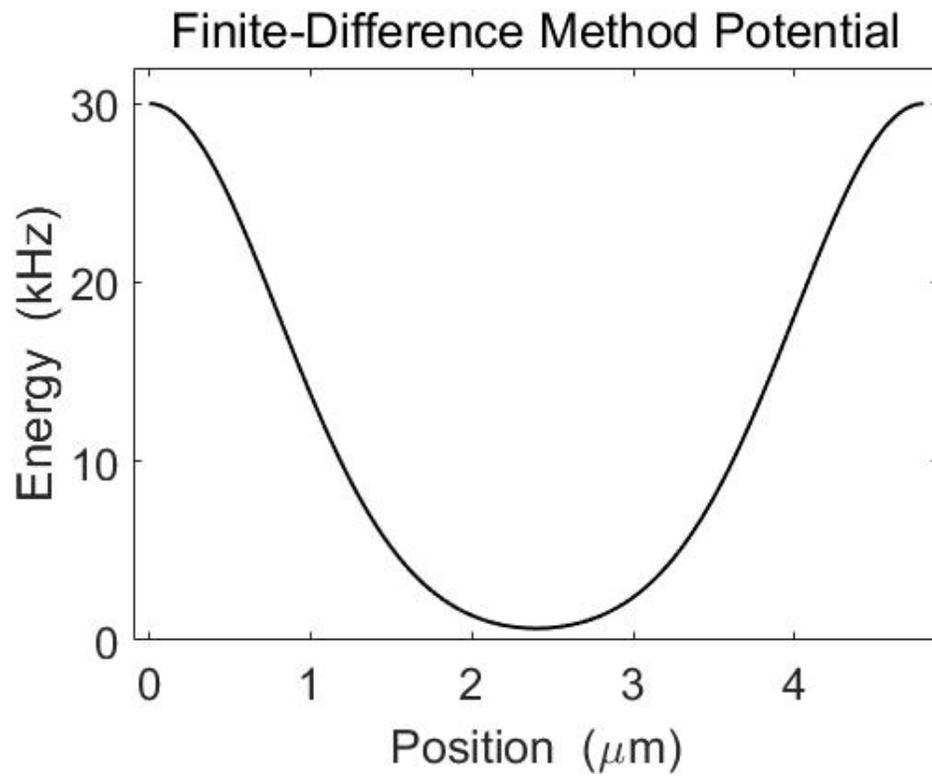


Figure 5.5: The Gaussian well potential used for the finite difference method.

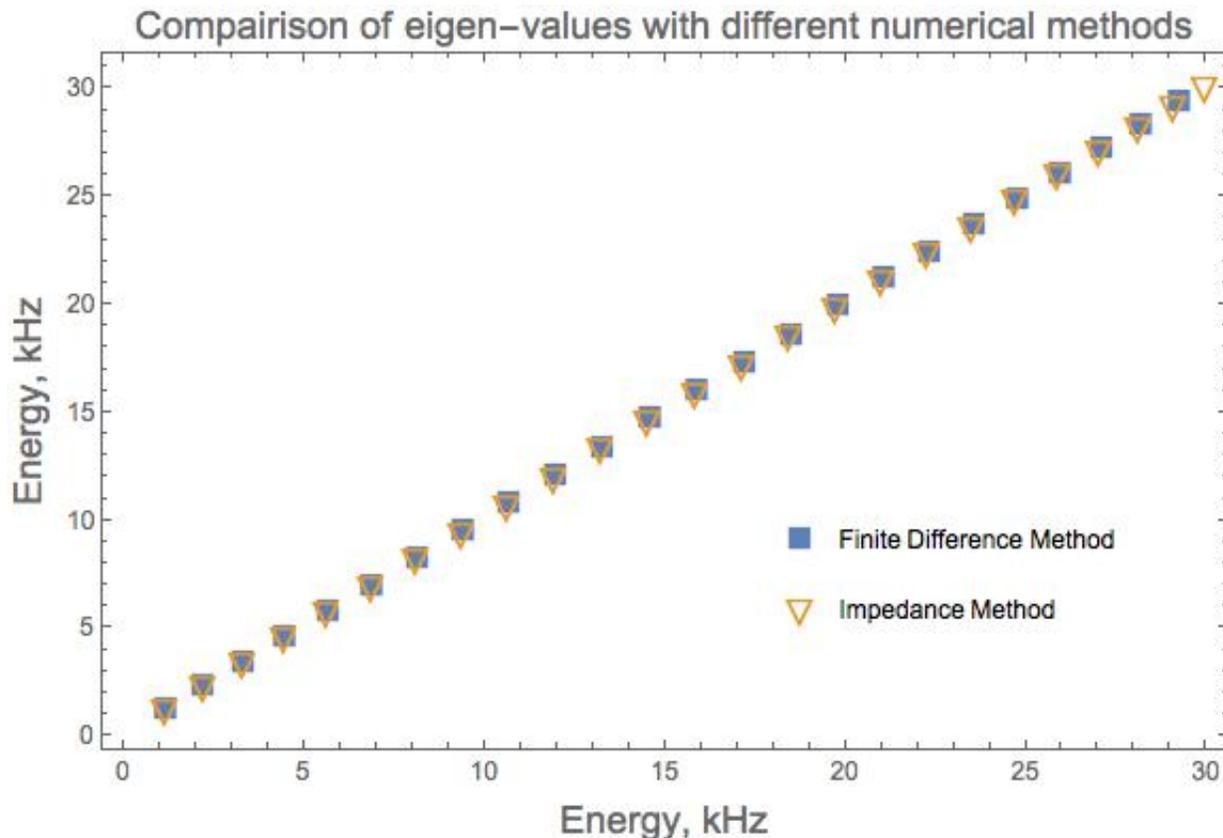


Figure 5.6: The comparison of two different methods for generating the eigenenergies of the double Gaussian barrier potential. The blue solid squares are the resonant energies that were found from solving Schrödinger's equation directly using the finite difference method and the orange empty triangles are the resonant energies that were found from the impedance method energy condition Eq. (2.3). The Gaussian barrier potential was defined to have a variance  $\sigma = 1.6 \mu\text{m}$ , a separation  $x_{sep} = 4.8 \mu\text{m}$  and a second barrier height  $u_{02} = 30 \text{ kHz}$ . Notice how the two agree at lower resonant energies, but as the barrier height is approached, the results start to deviate. This is in part due to the approximation of the barrier shape for the finite difference method; the potential, illustrated in Fig. 5.5, was taken such that the quasi-bound states could be found. Although the finite difference method potential was defined to be different near the barrier height, it is still a check on the lower bound energy levels.

The comparison between the two different numerical methods is a check to make sure that the impedance method is finding the correct eigenenergies for the double Gaussian barrier potential case. Figure 5.6 illustrates that the two methods are in agreement with one another, particularly at lower eigenenergies. It is important to note that the finite difference method used an approximated potential such that only the quasi-bound states of the Gaussian well would be found. The potential used in the finite difference method calculations is a Gaussian well and is illustrated in Fig. 5.5. The difference in the definition of the potentials accounts for the discrepancy between the two numerical methods at higher energies. In Fig. 5.6 the lower energy appear to have a uniform spacing between the eigenenergies. This is due to the fact that near the bottom of the gate well the potential is harmonic. Thus we expect a uniform spacing between the eigenenergies as in the simple harmonic oscillator case. As the incoming energies get closer to the top of our gate well, the harmonic approximation is no longer valid and the energy level spacing is no longer uniform.

The characteristics of the double Gaussian barrier potential can define the amount of resonant energies as well as the spacings between the resonant energies. Figure 5.7 shows the double Gaussian barrier potential for three separate cases, each of which have the same variance,  $\sigma = 1.0 \mu\text{m}$ , and the same second barrier height,  $u_{02} = 30 \text{ kHz}$ , but with different barrier separations. Given a larger separation between the barriers one expects the amount of resonant energies to increase as well as having a more uniform spacing between those resonant energies such that the well is still harmonic in shape. But since the larger separations are no longer harmonic we would expect the spacings for the lower resonant energies to be smaller than the spacings of the higher resonant energies. Each potential was evaluated using a uniform energy density ranging from 0.01 kHz to 40 kHz, and using the impedance method packages for the transmission spectrum and eigenenergies (Appendices A and D).

The transmission spectrum of the double Gaussian barrier potential with  $x_{sep} = 2.5 \mu\text{m}$  is illustrated in Fig. 5.8. Table H.1 of eigenenergies shows that there are a total of 13 resonant energies, although only five can be seen in Fig. 5.8 due to the uniform energy density used for the incoming energies. The resonant energies that are not shown in Fig. 5.8 require a much higher

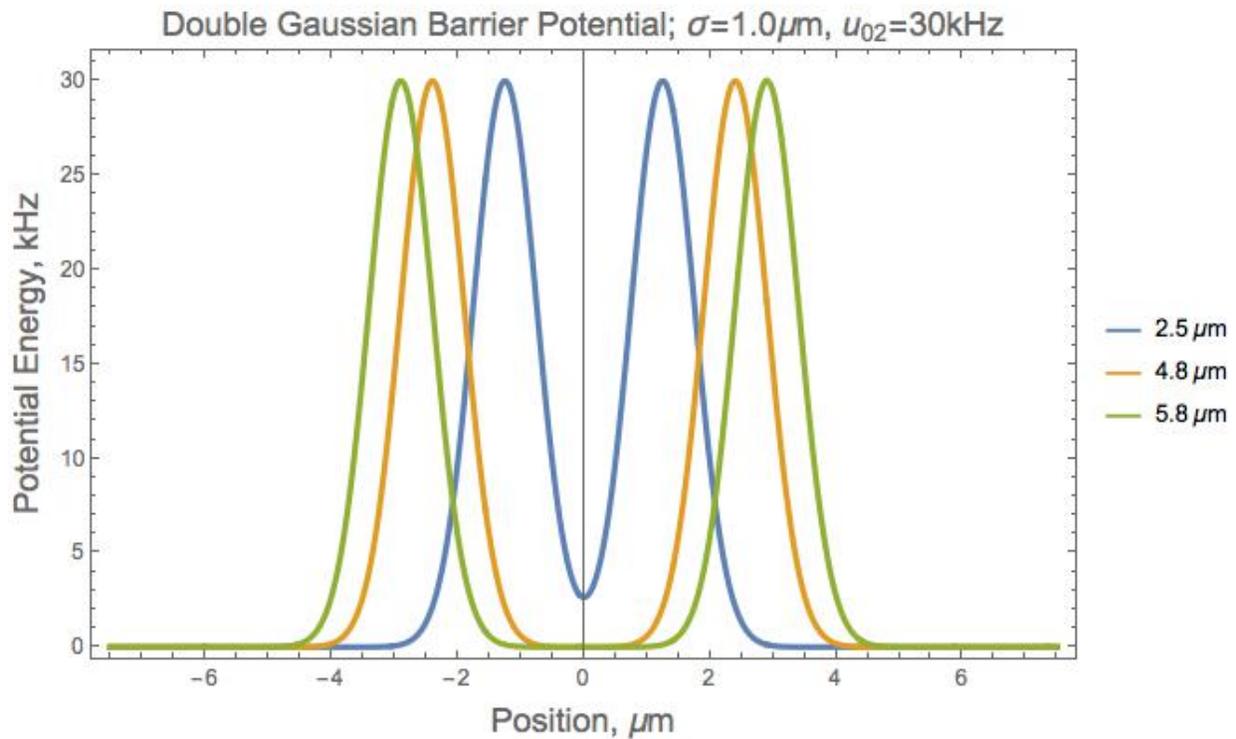


Figure 5.7: Double Gaussian barrier potential with barrier heights are fixed at 30kHz and variance is  $1.0 \mu\text{m}$ . The separation of the barriers is varied where  $x_{sep} = \{2.8 \mu\text{m}(\text{Blue solid line}), 4.8 \mu\text{m}(\text{Orange solid line}), 5.8 \mu\text{m}(\text{Green solid line})\}$ . Increasing the separation is expected to increase the amount of resonant energies possible. The spacing between the resonant energies will also have increased uniformity for the potential well that is harmonic in shape. Notice that the well for  $x_{sep} = 4.8 \mu\text{m}$  and  $5.8 \mu\text{m}$  is no longer similar to the harmonic well, and so we expect the resonant spacing to be have characteristics like that of the finite square well or double rectangular barrier, i.e. closer together for the lower energies than the higher energies.

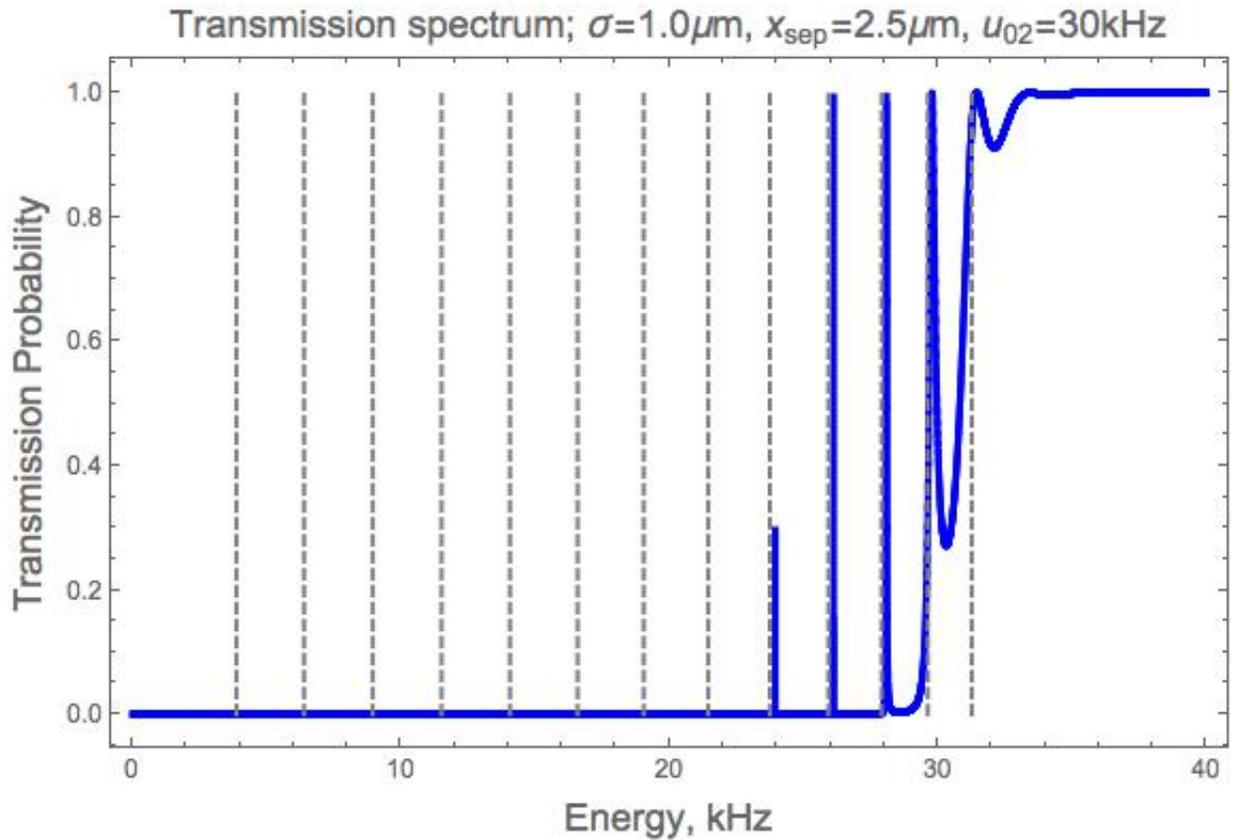


Figure 5.8: The transmission spectrum for a double Gaussian barrier potential defined by  $\sigma = 1.0 \mu\text{m}$ ,  $x_{\text{sep}} = 2.5 \mu\text{m}$ , and  $u_{02} = 30 \text{ kHz}$ . The blue solid line shows the transmission spectrum and the gray dashed vertical lines correspond to the calculated eigenenergies. There are 13 resonant energies. Only the last four resonant energies are shown on the transmission spectrum. This is due to the lower resonant energies having a smaller resonant width than the higher resonances. The resonant width is also why the first visible resonant energy does not show full transmission, although in reality it does have full transmission as demonstrated in Fig. 5.17 and Fig. 5.18. The spacing between the resonant energies is uniform, unlike the spacing shown in Fig. 5.9 and Fig. 5.10.

energy density to resolve them. This is due to the lower eigenenergies having an increasingly smaller resonant width than those resonant energies closer to the potential's peak height. This is why the lowest eigenenergy that is seen in Fig. 5.8 is not at full transmission, when in reality full transmission is achieved for each resonant energy. It is important to note this type of numerical limitation so that misinterpretation is avoided. Together with the evaluated eigenenergies, a new optimized energy density is developed in order to capture lower resonant energies for the transmission spectrum, this is shown in Fig. 5.17 and Fig. 5.18 and discussed later on in the chapter. Increasing the separation between the barriers to  $x_{sep} = 4.8 \mu\text{m}$ , whose transmission spectrum and eigenenergy values are shown in Fig. 5.9 and Table H.2 respectively, increase the amount of resonant energies allowed in the potential barrier well structure. It is not surprising that the first clearly visible resonant energy is around 24 kHz due to resonant widths, nor that the number of resonant energies has increased due to the increased barrier separation. The first visible resonant energy is at approximately the same energy for all three cases because the resonant widths are proportionally similar at the corresponding incoming energies. As is expected, increasing the separation allowed for a larger number of resonant energies, as was also seen for the double rectangular barrier in chapter 4.

Again increasing the separation of the Gaussian barriers to  $x_{sep} = 5.8 \mu\text{m}$ , and as seen in Fig. 5.10, the number of resonant energies visible has increased. As was the case for separations of  $x_{sep} = 2.5 \mu\text{m}$  and  $x_{sep} = 4.8 \mu\text{m}$ , the first visible resonant energy is around 24 kHz, and the number of resonant energies has increased. There are 13 resonant energies for  $x_{sep} = 2.5 \mu\text{m}$ , four of which are visible in Fig. 5.8. Increasing the separation to  $x_{sep} = 4.8 \mu\text{m}$  results in 28 resonant energies, six of which are visible in Fig. 5.12. Further increasing the separation to  $x_{sep} = 5.8 \mu\text{m}$ , leads to 35 resonant energies, eight of which are visible in Fig. 5.10. The spacing of the low eigenenergies in Fig. 5.9 and Fig. 5.10 is nonuniform, showing that the potential of the well is no longer harmonic and so uniform spacing of the quasi-bound states is no longer expected. Clearly an increase in separation increases the number of resonant energies available, but what does an increase in the variance do? To answer that we refer to Fig. 5.12, Table H.4 and Fig. 5.13, Table H.5. The double Gaussian barrier potential defined by  $\sigma = 1.6 \mu\text{m}$ ,  $x_{sep} = 4.8 \mu\text{m}$ , and  $u_{02} = 30 \text{ kHz}$  is illustrated in

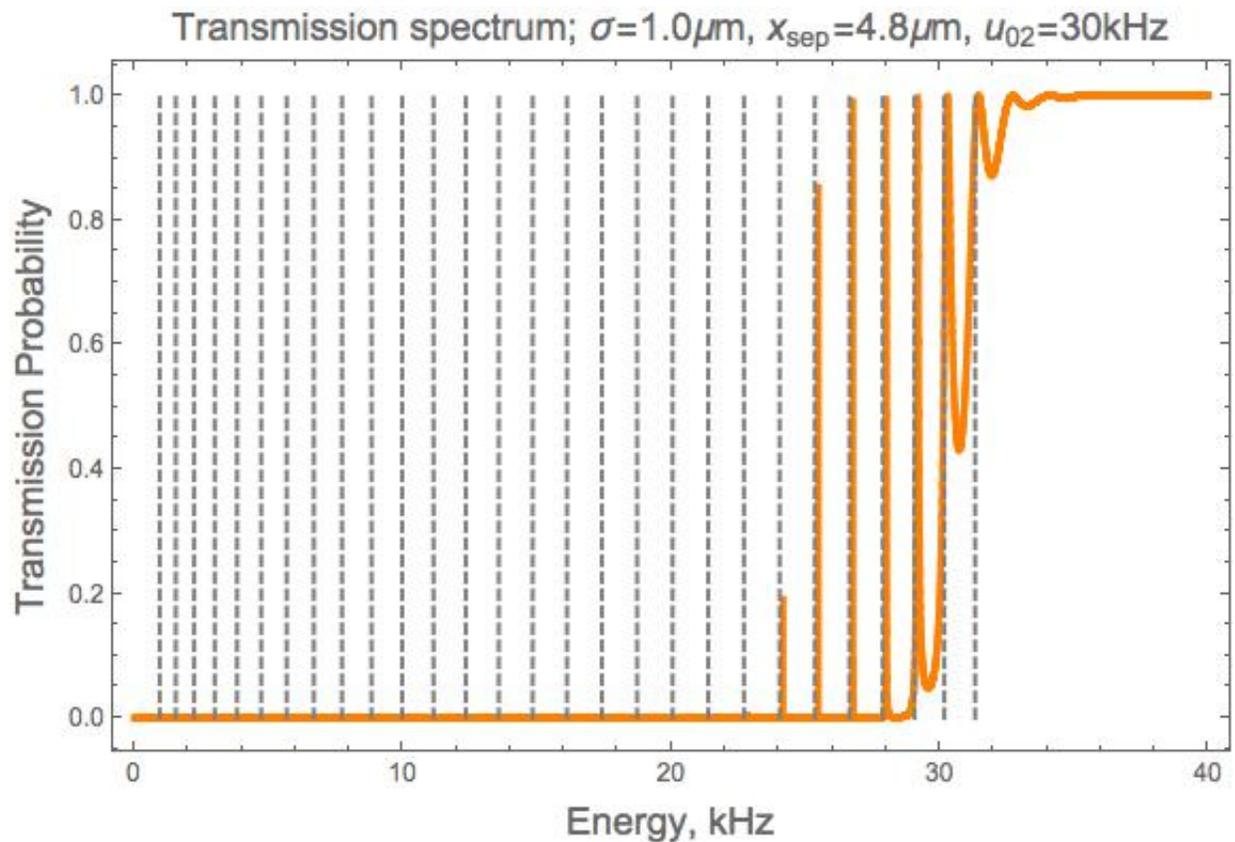


Figure 5.9: The transmission spectrum for a double Gaussian barrier potential defined by  $\sigma = 1.0 \mu\text{m}$ ,  $x_{\text{sep}} = 4.8 \mu\text{m}$ , and  $u_{02} = 30 \text{ kHz}$ . The orange solid line shows the transmission spectrum and the gray dashed vertical lines correspond to the calculated eigenenergies. There are 28 resonant energies. Notice the first resonant energy that can be seen is  $E = 24.03 \text{ kHz}$  but there are six more resonant energies compared to only three seen in Fig. 5.8. This implies that as the separation between the barriers is increased, so is the number of resonant energies.

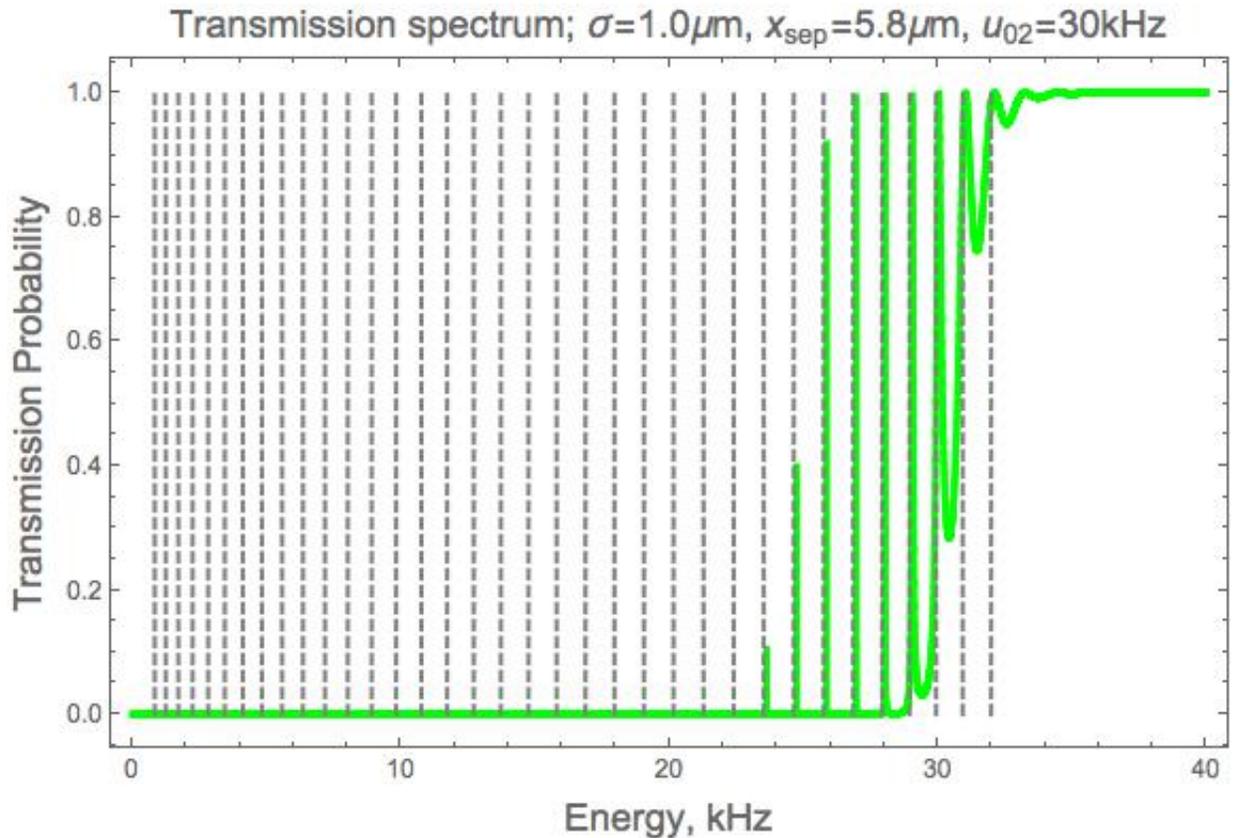


Figure 5.10: The transmission spectrum for a double Gaussian barrier potential defined by  $\sigma = 1.0 \mu\text{m}$ ,  $x_{\text{sep}} = 5.8 \mu\text{m}$ , and  $u_{02} = 30 \text{ kHz}$ . The green solid line shows the transmission spectrum and the gray dashed vertical lines correspond to the calculated eigenenergies. There are 35 resonant energies. Once again the first noticeable resonant energy is around  $E = 23.51 \text{ kHz}$  and there are eight resonant energies seen here. The spacing between the resonant energies is not uniform for the lower eigenenergies, because the shape of the potential well is no longer harmonic. Compared to Fig. 5.9 we see that the resonant energy spacing is even less uniform due to the flattening of the potential well.

Fig. 5.11 and its transmission spectrum is illustrated in Fig. 5.12. The first visible resonant energy for this particular case is  $E = 27.05$  kHz, and four resonances are shown in Fig. 5.12. Note that an increase in variance will decrease the separation of the barriers and cause a decrease the number of resonant energies. The energy level spacing for the potential whose variance is  $\sigma = 1.6 \mu\text{m}$  is approximately uniform, whereas the potential whose variance is  $\sigma = 1.0 \mu\text{m}$  did not have uniform eigenenergy spacing. Reviewing Fig. 5.7 and Fig. 5.12 its seen that the potential well bottoms differ significantly. The double Gaussian barrier potential in Fig. 5.7 has a potential whose gate well is flat that the bottom, while the double Gaussian barrier potential in Fig. 5.12 has a potential whose gate well is more harmonic in nature. Thus, the latter is more uniform. To gain further insight of how the variance of the double Gaussian potential affects the transmission spectrum and eigenenergies, please review Fig. 5.10 and Fig. 5.13. The double Gaussian potential giving the transmission spectrum in Fig. 5.13 is illustrated in Fig. 5.11 and has  $\sigma = 1.6 \mu\text{m}$ ,  $x_{sep} = 5.8 \mu\text{m}$ , and  $u_{02} = 30$  kHz. Because the double Gaussian potential in Fig. 5.11 has a potential well that is no longer harmonic, one expects the eigenenergy spacing to be nonuniform. interestingly the difference in the number of resonant energies in, Fig. 5.10 and Fig. 5.13, is two. Thus an increase of  $\Delta\sigma = 0.6 \mu\text{m}$  corresponds to two less available resonant energies. The decrease in available resonant energies is due to the shape of the potential well, for Fig. 5.10 the potential well has a fully flattened well bottom and so one would expect it to behave more like that of a finite square well where the lower resonant energies are spaced closer together. Comparatively, Fig. 5.13 is defined by a potential well that is just barely flattened at the bottom and so is more of a mixture between a harmonic well and finite square well, thus expecting the spacing of the lower eigenenergies to be nonuniform but have a larger spacing than those seen in Fig. 5.10.

Based on the work done with the single Gaussian barrier potential and the double rectangular barrier potential, one might expect that changing the relative height between the barriers, as shown in Fig. 5.14, will shift the eigenenergies slightly. However, as seen in Fig. 5.15 and Fig. 5.16, the change in symmetry does more than just a shift in resonance. The transmission spectrum for both figures significantly changes when compared to one another and further compared to the

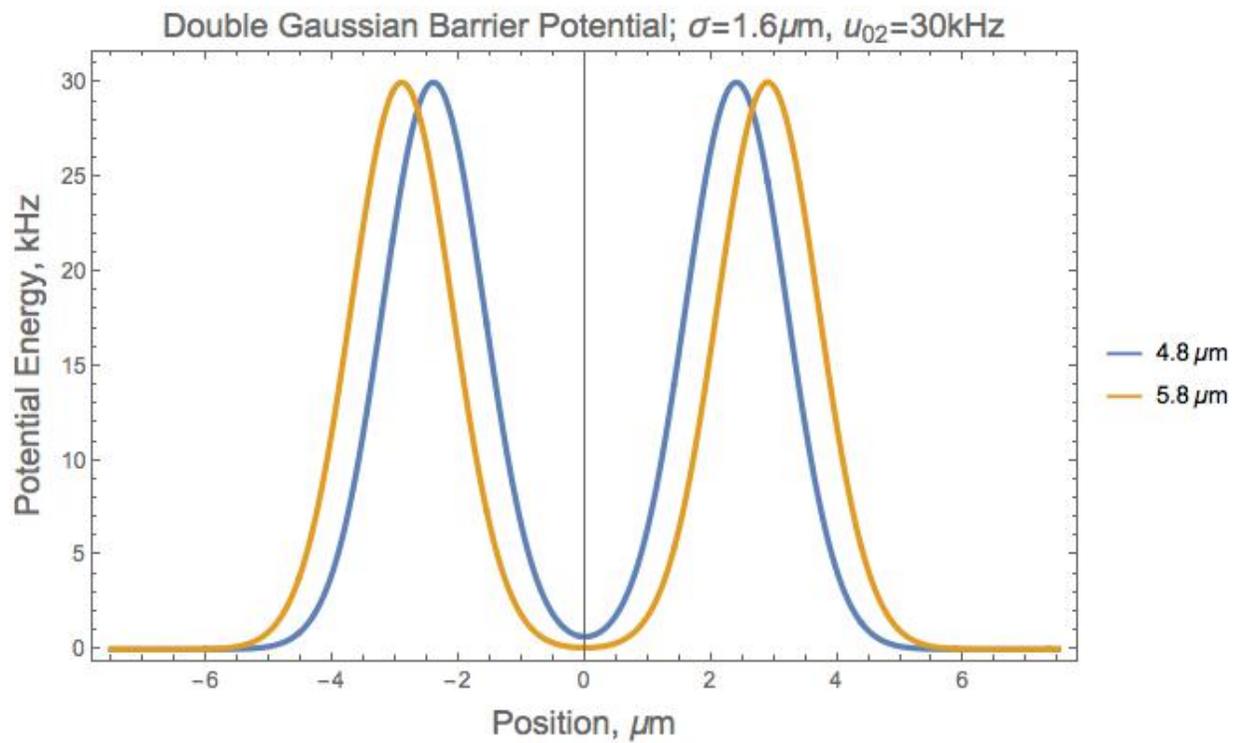


Figure 5.11: The double Gaussian barrier potential defined by  $\sigma = 1.6 \mu\text{m}$ ,  $x_{sep} = \{4.8 \mu\text{m}, 5.8 \mu\text{m}\}$ , and  $u_{02} = 30 \text{ kHz}$

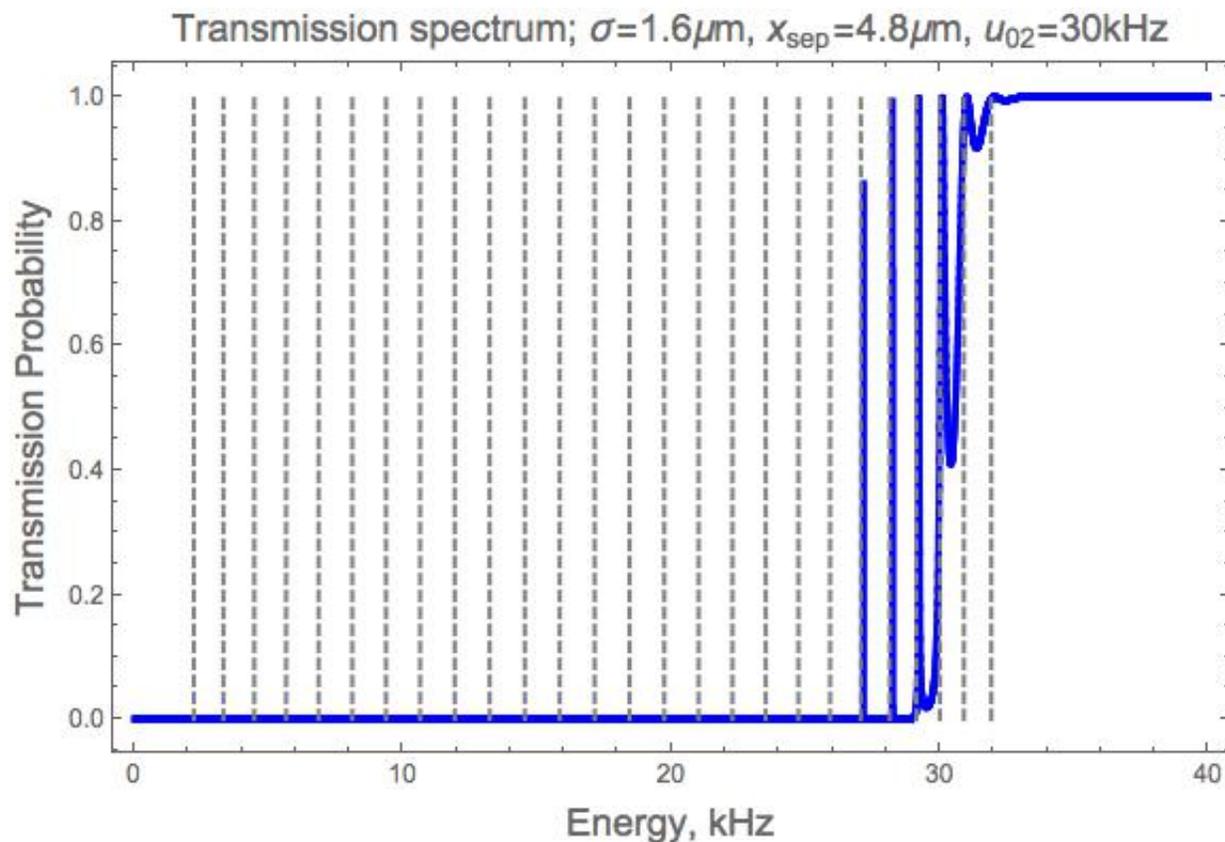


Figure 5.12: The transmission spectrum for the double Gaussian barrier potential defined by  $\sigma = 1.6 \mu\text{m}$ ,  $x_{\text{sep}} = 4.8 \mu\text{m}$ , and  $u_{02} = 30 \text{ kHz}$ , this is the potential that was also used in Fig. 5.6. The blue solid line shows the transmission spectrum and the gray dashed vertical lines correspond to the calculated eigenenergies. There are 27 resonant energies. Notice that there are four visible resonant energies, this is because the lower resonant energies have a much smaller resonant width. This is also why the first visible resonant energy does not show full transmission. The spacing between the resonances is uniform, as shown in Table H.4. The first noticeable resonant energy is  $E = 27.05 \text{ kHz}$ .

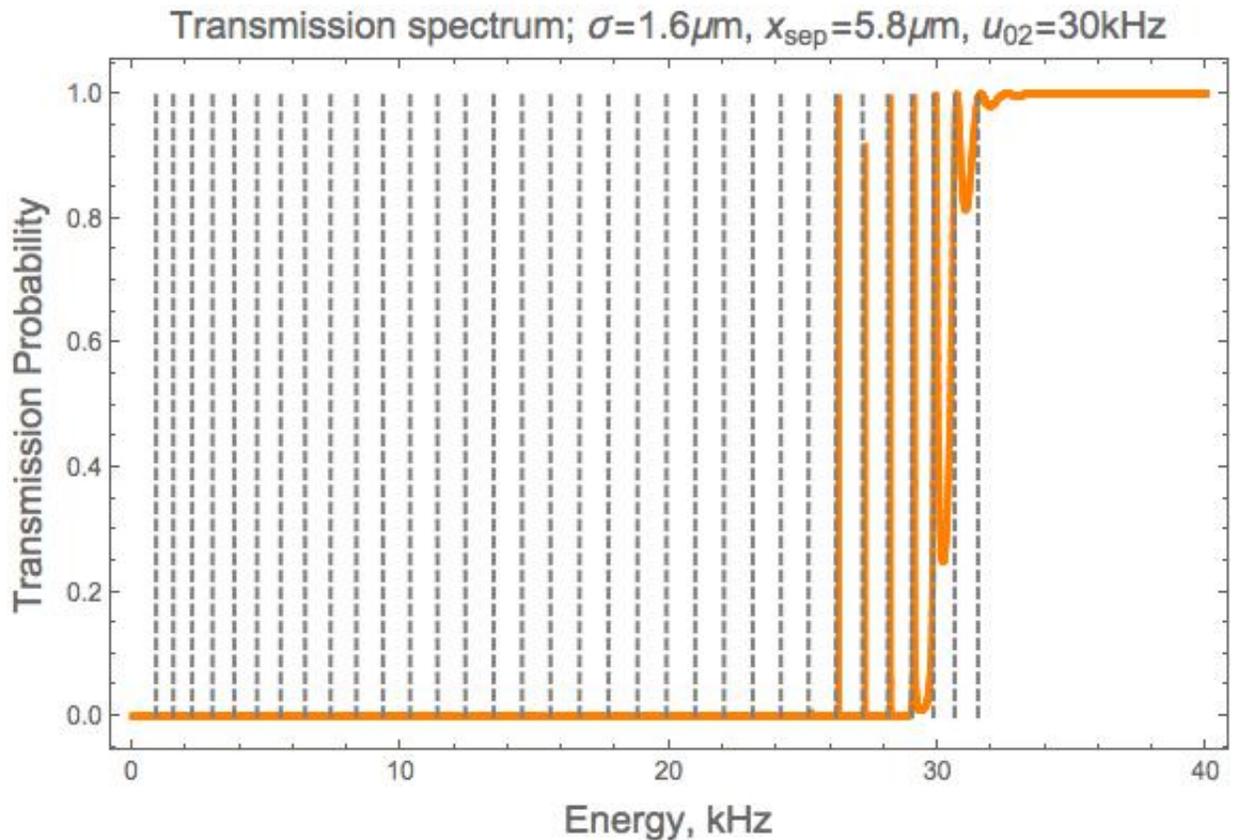


Figure 5.13: The transmission spectrum of the double Gaussian barrier potential defined by  $\sigma = 1.6 \mu\text{m}$ ,  $x_{\text{sep}} = 5.8 \mu\text{m}$ , and  $u_{02} = 30 \text{ kHz}$ . The orange solid line shows the transmission spectrum and the gray dashed vertical lines correspond to the calculated eigenenergies. There are 33 resonant energies. Notice that there are five visible resonant energies, due to the lower resonant energies have a much smaller resonant width than the final few, this is also why the first visible resonant energy does not show full transmission, although in reality it does have full transmission. The spacing between the resonances is not uniform, and decreases for the lower eigenenergies. This is because the potential well is no longer harmonic. The first noticeable resonant energy is  $E = 27.19 \text{ kHz}$ .

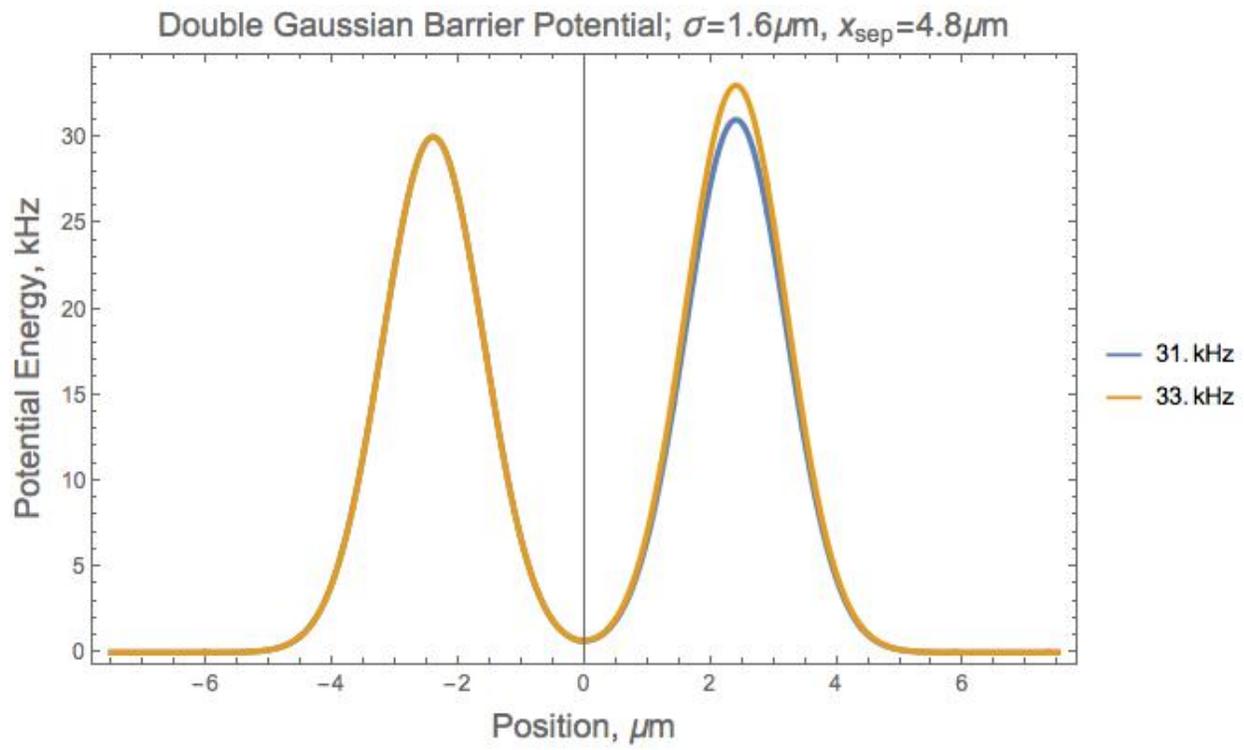


Figure 5.14: The double Gaussian barrier potential defined by  $\sigma = 1.6 \mu\text{m}$ ,  $x_{\text{sep}} = 4.8 \mu\text{m}$ , and  $u_{02} = \{31 \text{ kHz}(\text{Blue solid line}), 33 \text{ kHz}(\text{Orange solid line})\}$ . The difference in relative height between the two Gaussian barriers results in an asymmetric potential.

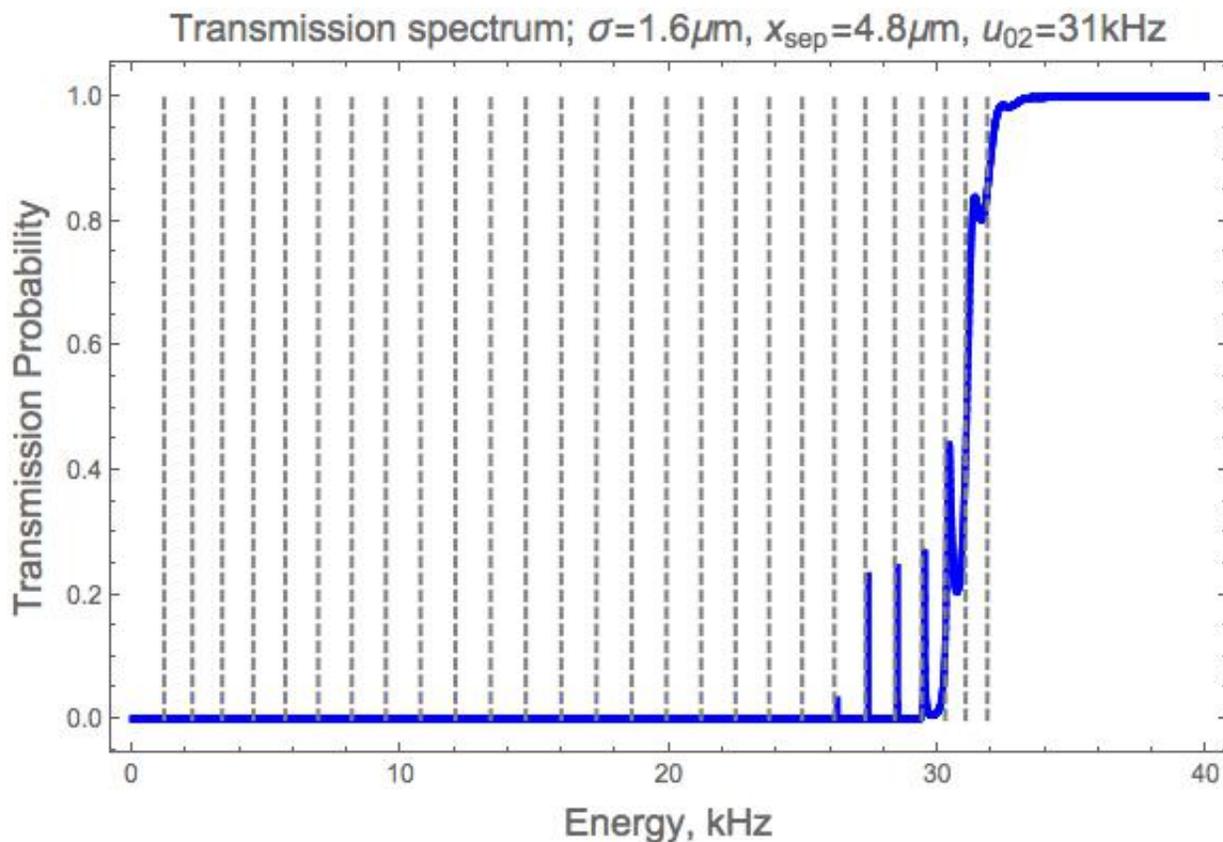


Figure 5.15: The transmission spectrum for a double Gaussian barrier potential defined by  $\sigma = 1.6 \mu\text{m}$ ,  $x_{\text{sep}} = 4.8 \mu\text{m}$ , and  $u_{02} = 31 \text{kHz}$ . The blue solid line shows the transmission spectrum and the gray dashed vertical lines correspond to the calculated eigenenergies. There are 27 resonant energies. The asymmetry of the double Gaussian barrier potential changes the transmission probability through the second barrier. The only difference between the potential for Fig. 5.12 is the height of the second barrier. The potential is now asymmetric and thus reduces the transmission probability even when on resonance.

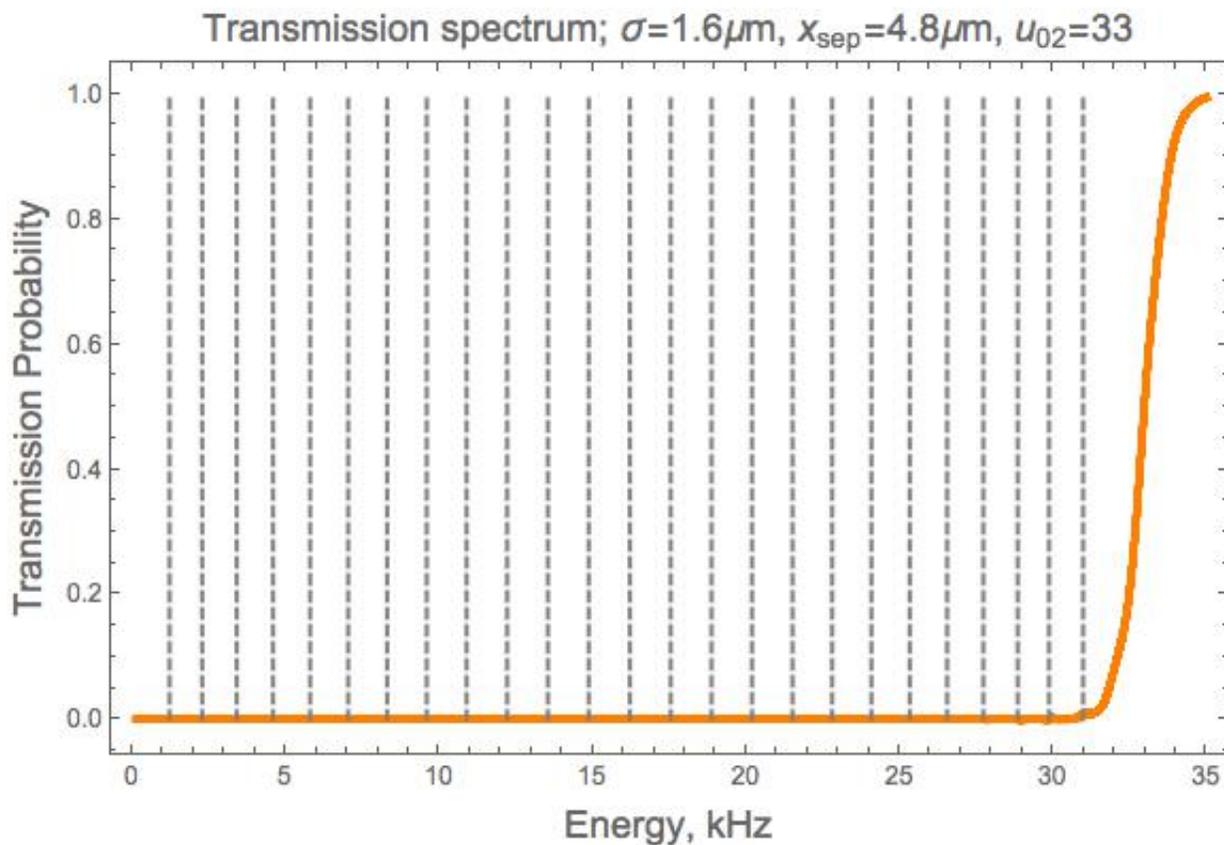


Figure 5.16: The transmission spectrum for a double Gaussian barrier potential defined by  $\sigma = 1.6 \mu\text{m}$ ,  $x_{\text{sep}} = 4.8 \mu\text{m}$ , and  $u_{02} = 33 \text{ kHz}$ . The orange solid line shows the transmission spectrum and the gray dashed vertical lines correspond to the calculated eigenenergies. There are 25 resonant energies. The transmission spectrum shown is similar to that of a single Gaussian barrier whose peak potential height is 33 kHz, the resonances do not have a transmission probability of unity for the asymmetric barrier. As seen in Table H.7 there are resonant energies where full transmission is not achieved even when on a resonant energy, a characteristic of an asymmetric barrier.

symmetric double Gaussian barrier potential in Fig. 5.12; the increase in the height of the second barrier significantly reduces the visible resonant energies, and in the case where  $u_{02} = 33$  kHz, there is no noticeable resonance. This drastic change in the transmission spectrum is surprising but reasonable since the barriers are no longer symmetric and the affect of the asymmetry of the barriers is a significant reduction of resonant transmission. Asymmetric double barriers were also studied in references [13] and [5], both of which found that asymmetry of the barriers significantly reduces the transmission of the wavepacket when on resonance. The lower resonant energies of the asymmetric double Gaussian barrier potentials are close to the resonant energies of the respective symmetric double Gaussian barrier potential, and the difference increases as the resonant energy increases. However the transmission of the wavepacket through the double Gaussian barrier potential is significantly reduced for the asymmetric double Gaussian barrier potentials. This type of behavior might prove useful for trapping the wavepacket within the gate well for a longer period of time, this behavior is explored further in references [4] and [14]. Increasing the incident amplitude of the wavepacket will allow for the resonant energies to be transmitted as discussed in reference [5].

Optimization of the transmission spectrum for the double Gaussian barrier potential can be done using the evaluated eigenenergies for a particular double Gaussian barrier potential. Using the eigenenergies, a related energy spectrum is made such that there is a higher density around the evaluated eigenenergies, illustrated in Fig. 5.17 and Fig. 5.18. The optimized transmission spectrum has two more visible resonance energies than the non-optimized transmission spectrum in Fig. 5.9. Figure 5.17 shows unity transmission for lower resonant energies, however this behavior is not captured in Fig. 5.9. The optimization illustrated in Fig. 5.18 is for the transmission spectrum of a double Gaussian barrier potential being defined by  $\sigma = 1.6 \mu\text{m}$ ,  $x_{sep} = 4.8 \mu\text{m}$ , and  $u_{02} = 30$  kHz. This potential is also used to make Fig. 5.12. With the optimized energy spectrum there are two more visible resonances and unity transmission in Fig. 5.18 than in Fig. 5.12. Optimization of the transmission spectra is useful in visualization of the resonances and their spacing as seen in Fig. 5.17 and Fig. 5.18. Qualitatively, from Fig. 5.18, the spacing of the resonant energies looks

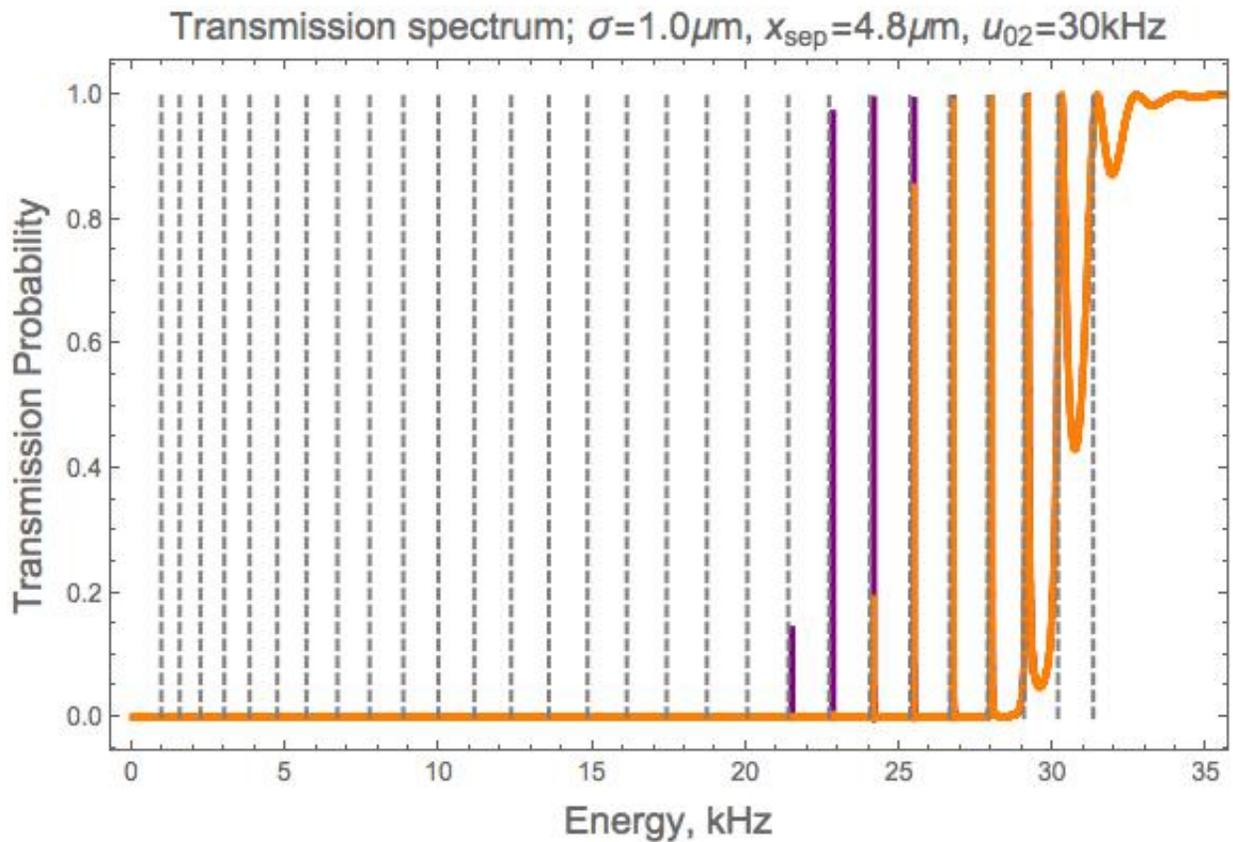


Figure 5.17: The optimized transmission spectrum for the double Gaussian barrier potential defined by  $\sigma = 1.0 \mu\text{m}$ ,  $x_{\text{sep}} = 4.8 \mu\text{m}$  and  $u_{02} = 30 \text{ kHz}$ . The purple solid line plots the optimized transmission spectrum, the orange solid line shows the transmission spectrum seen in Fig. 5.9 and the gray dashed vertical lines correspond to the calculated eigenenergies. Notice that the optimized spectrum shows unity transmission for  $E = 24.03 \text{ kHz}$ ,  $25.34 \text{ kHz}$ , and  $26.63 \text{ kHz}$  when Fig. 5.9 did not. Also notice that two more resonant energies are now visible,  $E = 21.37 \text{ kHz}$  and  $22.7 \text{ kHz}$ . The eigenenergies using the optimized energy density is the same as in Table H.2 to an order of  $10^{-4}$ .

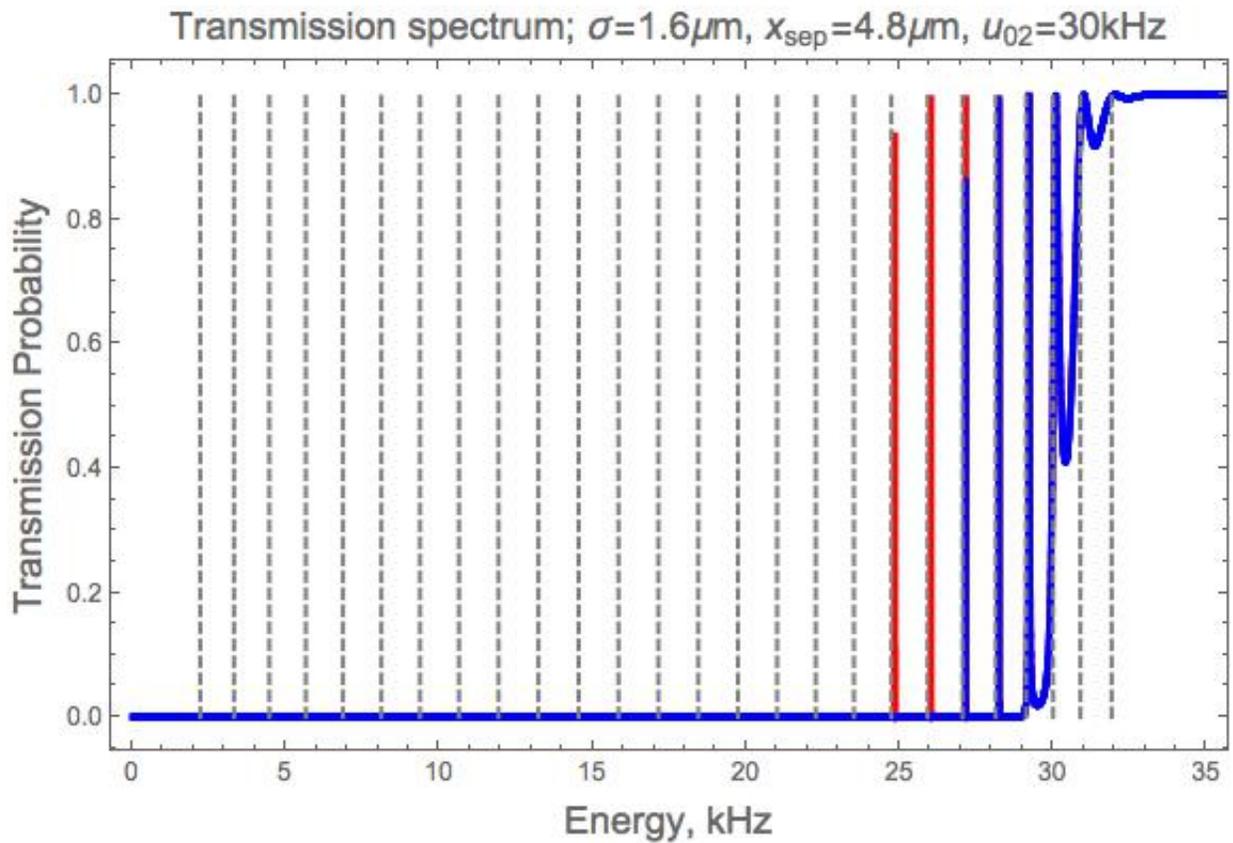


Figure 5.18: The optimized transmission spectrum for the double Gaussian barrier potential defined by  $\sigma = 1.6 \mu\text{m}$ ,  $x_{\text{sep}} = 4.8 \mu\text{m}$ , and  $u_{02} = 30 \text{ kHz}$ . The red solid line shows the optimized transmission spectrum, the blue solid line plots the transmission spectrum seen in Fig. 5.12 and the gray dashed vertical lines correspond to the calculated eigenenergies. Notice that the optimized spectrum has unity transmission for the resonant energy of  $E = 27.05 \text{ kHz}$  and two more resonant energies are visible now as well, those being  $E = 24.73 \text{ kHz}$  and  $25.91 \text{ kHz}$ . The eigenenergies from the optimized energy density are the same as those in Table H.4 within  $10^{-4}$ .

uniform and near 1 kHz. Knowing the resonant energies and their spacings in relation to each other can allow us to engineer state coupling and coherence.

### 5.3 Chapter conclusion

The atomtronics transistor potential is approximated by a superposition of a harmonic well and a double Gaussian barrier potential created by a magnetic potential. The three wells of the atomtronics transistor are analogous to the three terminals of an electronic transistor. By using the impedance method to evaluate the transmission spectrum and eigenenergies for double Gaussian barrier potential, it is found that an increase in separation and decrease in variance of a symmetric double Gaussian barrier increases the amount of resonant energies. This is due to many factors, but the most influential being the shape of the potential well, i.e. if it is harmonic or not. A symmetric double Gaussian barrier with a harmonic well allows for more resonant energies and a more uniform spacing of the resonances. Altering the double Gaussian barrier potential such that the potential is no longer symmetric about the origin reduces the transmission probability when on resonance. Using asymmetric double Gaussian barriers, or double rectangular barriers, might allow for trapping of the wavepacket. When the wavepacket is on resonance with an asymmetric double barrier the wavepacket will be at a maximum within the well, but transmission through the second barrier will be significantly reduced. Thus the wavepacket is trapped within the barriers for some amount of time before tunneling out. This type of trapping is explored further in [4], [5], and [14]. Studying the effects that different characteristics of the double Gaussian potential have on transmission and resonances shows the effectiveness and simplicity of the impedance method. Figure H.1 gives overall trends of the resonant energies studied thus far for the double Gaussian barrier potential. The Mathematica notebook for the double Gaussian barrier potential is shown in Appendix G, and all Tables referenced in this chapter are shown in Appendix H.

## Chapter 6

### Conclusion

The general impedance method is a numerical method that draws an analogy between the Schrödinger equation and transmission line theory. The impedance method is used to analyze the transmission spectrums of a single Gaussian barrier potential, a double rectangular barrier potential, and a double Gaussian barrier potential as well as extract the eigenenergies of the trapped states of the double barrier potentials.

Potentials with finite height and width, like the single Gaussian barrier potential, exhibit tunneling in its transmission spectrum. The transmission spectrum of a single Gaussian barrier potential varies according to the characteristics of the barrier. The slope of the spectrum shows a non-linear dependence on  $\sigma$ . Furthermore, the position of the 50% transmission energy varies proportionally with the Gaussian barrier height. Analysis of the single Gaussian barrier transmission spectrum was motivated by the application of the impedance method to the atomtronic battery. The single Gaussian barrier potential is discussed in regards to the atomtronic battery, which is generated by the superposition of a harmonic well and a single Gaussian barrier potential.

The transmission spectrum of the double rectangular barrier exhibits resonant tunneling behavior, which is a phenomena that occurs in periodic potentials. Resonant tunneling occurs when the incoming energy of the wavepacket matches the resonant energy of the states supported between the barriers, allowing for full transmission to occur when the energies are matched. From the transmission spectra of the double rectangular barrier potential, the separation of the barriers corresponds to the amount of allowed resonant energies and the difference in the heights of the

barriers slightly shifted the resonant energies.

The double Gaussian barrier potential approximates of the atomtronic transistor potential. The atomtronic transistor potential is formed by the superposition of a harmonic well and a double Gaussian potential to produce three wells. These wells are named the source, gate, and drain and correspond to the three terminals of an electronic transistor. The chemical potentials of the atoms in each well are analogous to the node voltages of an electronic transistor. As in the double rectangular barrier potential, the double Gaussian barrier potential supports resonant tunneling behavior. It was shown that the characteristics of the double Gaussian barrier can be tuned to control the transmission of resonant energies. The separation of the two barriers determines the number of allowed trapped states, in a similar manner to the double rectangular barrier potential. Asymmetry of the double Gaussian barrier potential was explored and was found to reduce transmission of the resonant energies. This mechanism can be used to control population trapping in the central well. The double Gaussian barrier has a more uniform distribution of resonant energies when the central well mimics the curvature of a harmonics well, as expected. Analysis of the double Gaussian barrier potential revealed many different characteristics of the potential which can be tuned to affect the resonant energies.

Armed with this information a discussion was provided that indicated the possibility of manipulating the triple-well potential such that the resonant energies are coupled to the dipole moment of the gate condensate. Coupling the center of mass motion of the condensate to the atoms incident on the barrier at resonant energies will allow for energy transfer via inter-particle interactions between the wave packet and condensate. If the wavepacket takes energy from the gate condensate its tunneling probability is increased otherwise if the condensate takes energy from the wavepacket the tunneling probability decreases and atoms become trapped within the gate well. Coupling between the resonant energy levels and the dipole moment of the condensate leads to correlation of the output atom current of the transistor and the gate well condensate motion. Thus far, the work presented in reference [2] was done in the classical regime where thermal currents dominate. The work presented in this thesis analyzes the quantum mechanical characteristics of the

atomic transistor potential such that tunneling can be incorporated and perhaps used to explore coherent transistor action.

## Bibliography

- [1] A. N. Khondker, M. R. Khan, and A. F. M. Anwar, "Transmission line analogy of resonance tunneling phenomena: The generalized impedance concept," J. Appl. Phys., vol. 63, no. 10, p. 5191, 1988.
- [2] S. Caliga, "Experimental realization of atomtronic circuit elements in non-equilibrium ultracold atomic systems," Ph.D. dissertation, University of Colorado at Boulder, 2016.
- [3] B. Gutierrez-Medina, "Wave transmission through periodic, quasiperiodic, and random one-dimensional finite lattices," Am. J. Phys., vol. 81, no. 2, p. 104, 2013.
- [4] A. Stamp and G. McIntosh, "A time-dependent study of resonant tunneling through a double barrier," Am. J. Phys., vol. 64, no. 3, 1996.
- [5] A. F. M. Anwar, A. N. Khondker, and M. R. Khan, "Calculation of the traversal time in resonant tunneling devices," J. Appl. Phys., vol. 65, no. 7, 1989.
- [6] S. F. Kabir, M. Khan, and M. Alam, "Application of quantum mechanical wave impedance in the solution of schrodinger's equation in quantum wells," Solid-State Electronics, vol. 34, no. 12, p. 1466, 1991.
- [7] D. J. Griffiths, Introduction to Quantum Mechanics, 2nd ed., J. Chalice, Ed. Pearson Prentice Hall, 2005.
- [8] R. Shankar, Principles of Quantum Mechanics, 2nd ed. Plenum Press, 1994.
- [9] S. Nandi, "The quantum gaussian well," Am. J. Phys., vol. 78, p. 1341, 2010.
- [10] F. M. Fernández, "Quantum gaussian wells and barriers," Am. J. Phys., vol. 79, no. 7, 2011.
- [11] A. A. Zozulya and D. Z. Anderson, "Principles of an atomtronic battery," Physical Review A, vol. 88, no. 4, 2013.
- [12] H. Yamamoto, "Resonant tunneling condition and transmission coefficient in a symmetrical one-dimensional rectangular double-barrier system," Appl. Phys. A, vol. 42, 1987.
- [13] A. Dutt and S. Kar, "Smooth double barriers in quantum mechanics," Am. J. Phys., vol. 78, no. 12, p. 1352, 2010.
- [14] B. Ricco and M. Y. Azbel, "Physics of resonant tunneling. the one-dimensional double-barrier," American Physical Society, vol. 29, no. 4, 1984.
- [15] C. M. Fabre, P. Cheiney, G. L. Gattobigio, F. Vermersch, S. Faure, R. Mathevet, T. Lahaye, and D. Guery-Odelin, "Realization of a distributed bragg reflector for propagating guided matter waves," American Physical Society, vol. 107, no. 230401, 2011.
- [16] P. Wellin, Programming with Mathematica An Introduction. Cambridge University Press, 2013.

## Appendix A

The Mathematica package for the impedance method

**ImpedanceMethod.m**

# General Impedance Method

08.07.2015

Anderson Group

## MetaPackage:

Transmission line analogy of resonance tunneling phenomena: The generalized impedance concept by A.N. Khondker

J. Appl. Phys., vol. 63, no. 10, p.5191, 1988

Wave transmission through periodic, quasiperiodic, and random one-dimensional finite lattices by Braulio Gutierrez-Medina

Am. J. Phys., vol. 81, no. 2, p. 104, 2013

This package used the general impedance method to numerically evaluate the reflection probabilities of an arbitrary potential energy function. The general impedance method evaluates the potential energy function from the output end to the input end of the potential.

The input values for this package are:

$v = \text{Table}[\text{function}[i], \{i, i_{\min}, i_{\max}, \text{stepsize}\}]$  : the arbitrary potential energy function as a list of points.

$\text{energy} = \text{Table}[i, \{i, i_{\min}, i_{\max}, \text{stepsize}\}]$  : the span of energy values that an arbitrary wavefunction might have. Note, this could also be a list of one.

$l = \text{constant}$  : step size, or more specifically the value  $x[[2]] - x[[1]]$ . This is the step size at which the potential energy function takes.

The output values for this package is the values of the reflection probability for each value of energy.  $\text{Length}[\text{ReflectionProbability}] = \text{Length}[\text{energy}]$ .

The variables defined within the package are:

$m =$  effective mass of the wave packet in units of kg (for rubidium)

$\hbar =$  Planck's constant in units of Js

$k =$  propagation constant

$z_0 =$  characteristic impedance

$z_{\text{load}} =$  load impedance

$z_{\text{input}} =$  input impedance

## Begin Package

```
BeginPackage ["ImpedanceMethod`ImpedanceMethod`"]
```

```
ImpedanceMethod`ImpedanceMethod`
```

## Usage Message

```
ReflectionProbability::usage =
  "ReflectionProbability[v,energy,l] computes the reflection
    coefficient by using the general impedance method. The arguments
    are v, energy, and l where they are the potential energy vector
    function, energy vector function, and the step size, respectively.
  m is the effective mass of the wave packet and  $\hbar$  is Planck's constant
    in units of Js, both are defined within the package.
  v must be of the form v=Table[function[i],{i,imin,imax,l}]
  energy must be of the form energy=Table[i,{i,imin,imax,l}]"
```

## Warning Messages

```
ReflectionProbability::badv =
  "The potential energy argument `1` must be in the form of a list.";

ReflectionProbability::badenergy =
  "The energy argument `1` must be in the form of a list.";

ReflectionProbability::badl =
  "The step size argument `1` is not within the appropriate bounds. l must be
    greater than 0 and less than the Thermal de Broglie Wavelength, `2`.";
```

## Options

### Private Context

```
Begin["`Private`"]
ImpedanceMethod`ImpedanceMethod`Private`
```

### Function Definitions (Public and Private)

```
m = 1.44*^-25;
 $\hbar$  = 1.054571726*^-34;
```

```

ReflectionProbability[v_, energy_, l_] :=
Module[{k, z0, zload, zinput, reflectionL, i},
Which[
! (VectorQ[v]), Message[ReflectionProbability::"badv", v],
! (VectorQ[energy]), Message[ReflectionProbability::"badenergy", energy],
! (0 < 1 ≤  $\frac{(2 * N[\pi, 10] * \hbar)}{\text{Sqrt}[2 * m * \text{Max}[\text{energy}] ]}$ ),
Message[ReflectionProbability::"badl", l,  $\frac{(2 * N[\pi, 10] * \hbar)}{\text{Sqrt}[2 * m * \text{Max}[\text{energy}] ]}$ ]
];
Do[
k = i * Sqrt[2 * m * (energy - v[[i]])] / ħ;
z0 = -i * ħ * k / m;
If[i == Length[v], zload = z0, zload = zinput];
zinput = z0 *  $\frac{zload * \text{Cosh}[k * l] - z0 * \text{Sinh}[k * l]}{z0 * \text{Cosh}[k * l] - zload * \text{Sinh}[k * l]}$ , {i, Length[v], 1, -1}];
reflectionL = Abs[ $\frac{zinput - z0}{zinput + z0}$ ] ^ 2

```

## End Private Context

```
End[]
```

```
ImpedanceMethod`ImpedanceMethod`Private`
```

## End Package

```
EndPackage[]
```

## Appendix B

The step potential barrier Mathematica notebook

`stepPotentialTest.nb`

# Step Potential Test:

This is a test of the ImpedanceMethod package by using a step potential.

```
Quit (*Quits out of local kernal*)

ClearAll["Global`"] (*clears all set variables with in the global context*)

$ContextPath (*gives the current context*)
{StreamingLoader`, IconizeLoader`,
 CloudObjectLoader`, PacletManager`, System`, Global`}

Get["/Users/avaashby/Desktop/masters-thesis/Mathematica
 Code/ImpedanceMethod/ImpedanceMethod.m"]
(*loads the ImpedanceMethod package. There are several ways to do this,
 here is a couple options:
 1) For newer versions of Mathematica you can start with
 Get["dkfj and a small icon will appear to let you search your computers files.
 2) If you know the file path use Get["filePath"] or Needs["filePath"]
 3) If you dont know the file path and the
 icon does not appear, try going to File -> Install -> "
 Type of Item to Install:" select Package, "Source:
 " select from file, and choose the appropriate file, ie
 ImpedanceMethod.m. Now use <<ImpedanceMethod` for the input*)

$ContextPath (*gives the current context,
 ie shows that the ImpedanceMethod package
 loaded correctly and we are now with in its context*)
{ImpedanceMethod`ImpedanceMethod`, StreamingLoader`,
 IconizeLoader`, CloudObjectLoader`, PacletManager`, System`, Global`}

m = 1.44*^-25; (*mass*)
ħ = 1.054571726*^-34; (*Planck's constant*)
v0 = 1000 * 2 * π * ħ; (*potential energy height*)
l = 4.0*^-9; (*potential energy step size*)
energy0 = Table[5000 * 2 * π * ħ * i, {i, 0, 1, 2.0*^-4}];
(*list of the energy that ranges from 0 to 3 with a step size of l*)

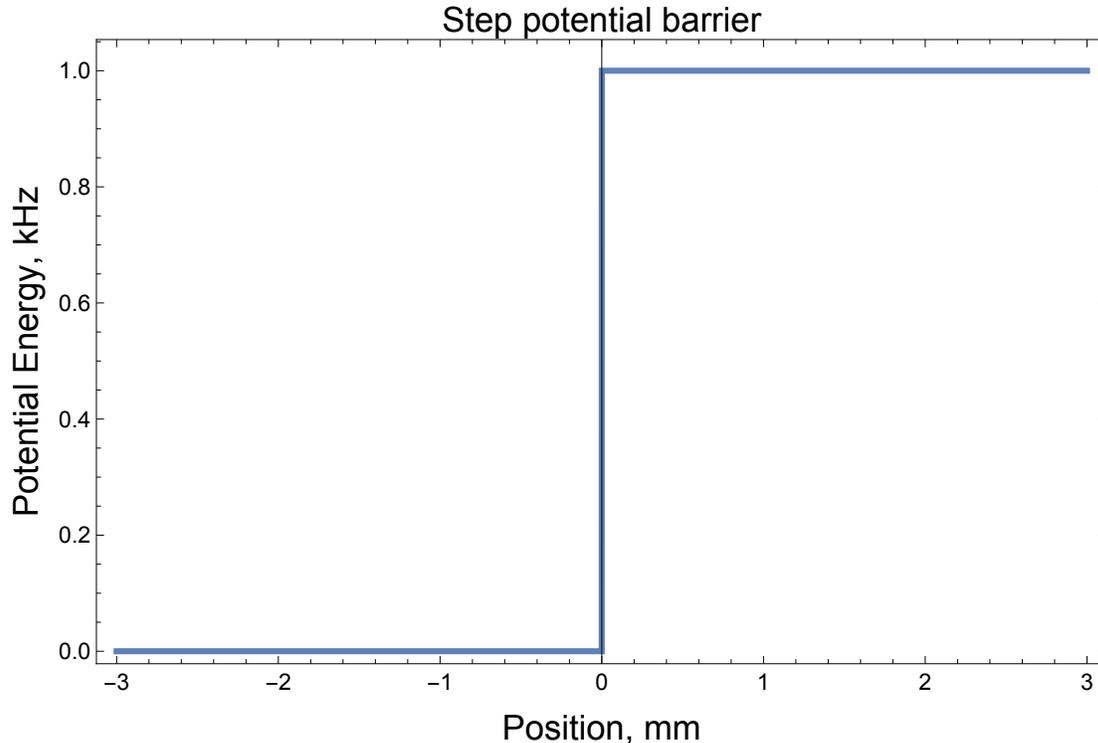
v = Table[v0 * UnitStep[i], {i, -3.0*^-3, 3.0*^-3, 1}];
(*list of the potential energy with a step size l*)

x = Table[i, {i, -3.0*^-3, 3.0*^-3, 1}];
(*list of the postion values for the potential energy function,
 only nessicary for plotting the potential energy function*)
```

```

stepPot = ListLinePlot[
  Transpose[{x * 10^3, v / (2 * π * ħ * 10^3)}, ImageSize → Large, Frame → True,
  FrameLabel → {Style["Position, mm", 18], Style["Potential Energy, kHz", 18]},
  PlotLabel → Style["Step potential barrier", 18],
  FrameTicksStyle → Medium, ImageSize → Large, PlotStyle → Thickness[.006]]

```



```

Export["/Users/avaashby/Desktop/masters-thesis/Mathematica
  Code/ImpedanceMethod/thesis nb's/possible figs/stepPot.jpeg", stepPot];

energy = If[Intersection[v, energy0] == {}, energy0,
  Delete[energy0, Flatten[Position[energy0, Intersection[v, energy0][[1]]]]];
(*checks to see if any value of energy0 is equal to any value of v,
and if so it deletes that value from energy0*)

Dimensions[energy]

{5000}

rP = ReflectionProbability[v, energy, 1];
(*uses ImpedanceMethod package to evaluate the reflection probabilities*)

stepReflection = Table[If[energy[[i]] ≤ v0, 1, rP[[i]]], {i, 1, Length[energy]}];
(*because the step potential is special,
in that for all values energy ≤ v0 ReflectionProbability=1,
this replaces all "indeterminant" values of ReflectionProbability with 1*)

data = Transpose[{energy / (2 * π * ħ * 10^3), stepReflection};
(*creates a 2-dimensional list*)

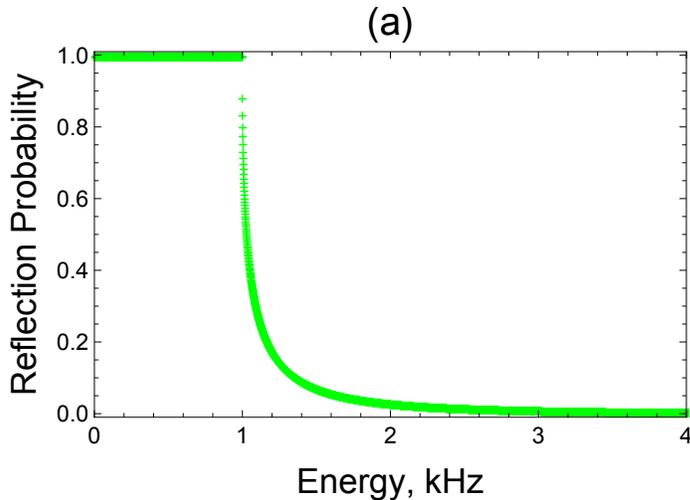
(*PlotLabel→Style["Reflection & Transmission",18],*)

```

```

reflectionLPlot = ListPlot[data, Frame -> True,
  FrameLabel -> {Style["Energy, kHz", 18], Style["Reflection Probability", 18]},
  PlotRange -> {{0, 4}, {-0.01, 1.01}}, PlotMarkers -> "+", PlotStyle -> Green,
  PlotLabel -> Style["(a)", 20], FrameTicksStyle -> Medium]
(*plots the resulting reflection behavior*)

```

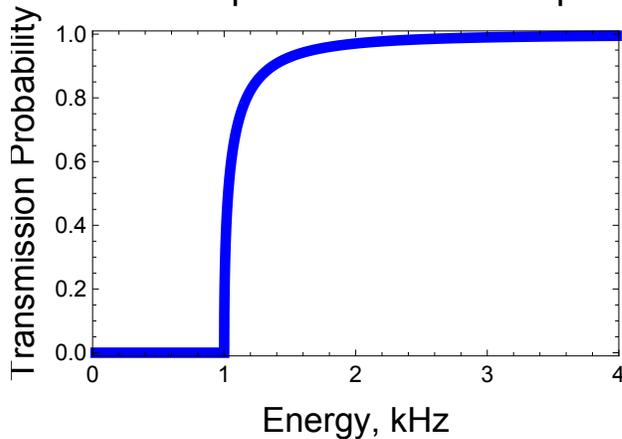


```

transmissionLPlot =
  ListLinePlot[Transpose[{energy / (2 * π * ħ * 10^3), 1 - stepReflection}], Frame -> True,
  FrameLabel -> {Style["Energy, kHz", 18], Style["Transmission Probability", 18]},
  PlotRange -> {{0, 4}, {-0.01, 1.01}}, (*PlotMarkers -> {"+", Large}, *)
  PlotStyle -> Directive[Blue, Thickness[.02]],
  PlotLabel -> Style["Transmission spectrum for the Step Potential", 20],
  FrameTicksStyle -> Medium] (*Plots resulting transmission behavior*)

```

ansmission spectrum for the Step Potenti

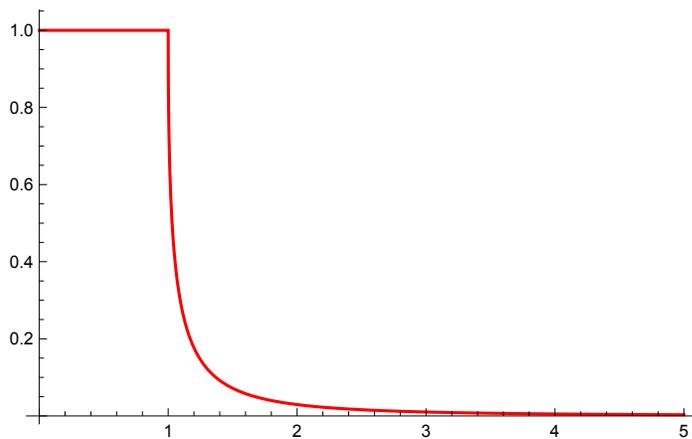


```

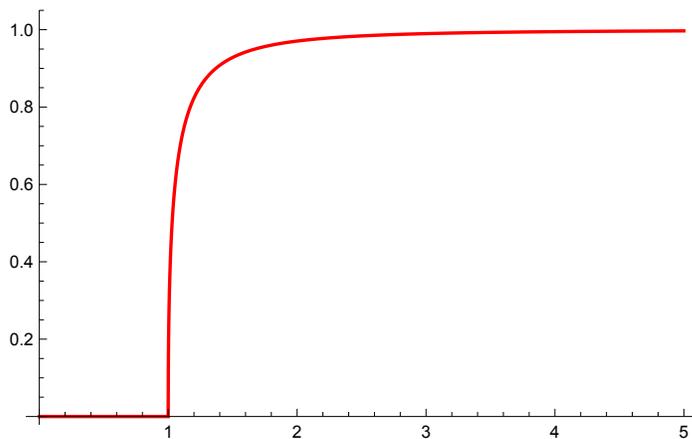
anarefl = Table[If[energy[[i]] ≤ v0, 1,
  ((energy[[i]] - Sqrt[energy[[i]] * (energy[[i]] - v0)]) / (energy[[i]] +
    Sqrt[energy[[i]] * (energy[[i]] - v0)))]^2, {i, 1, Length[energy]}];

```

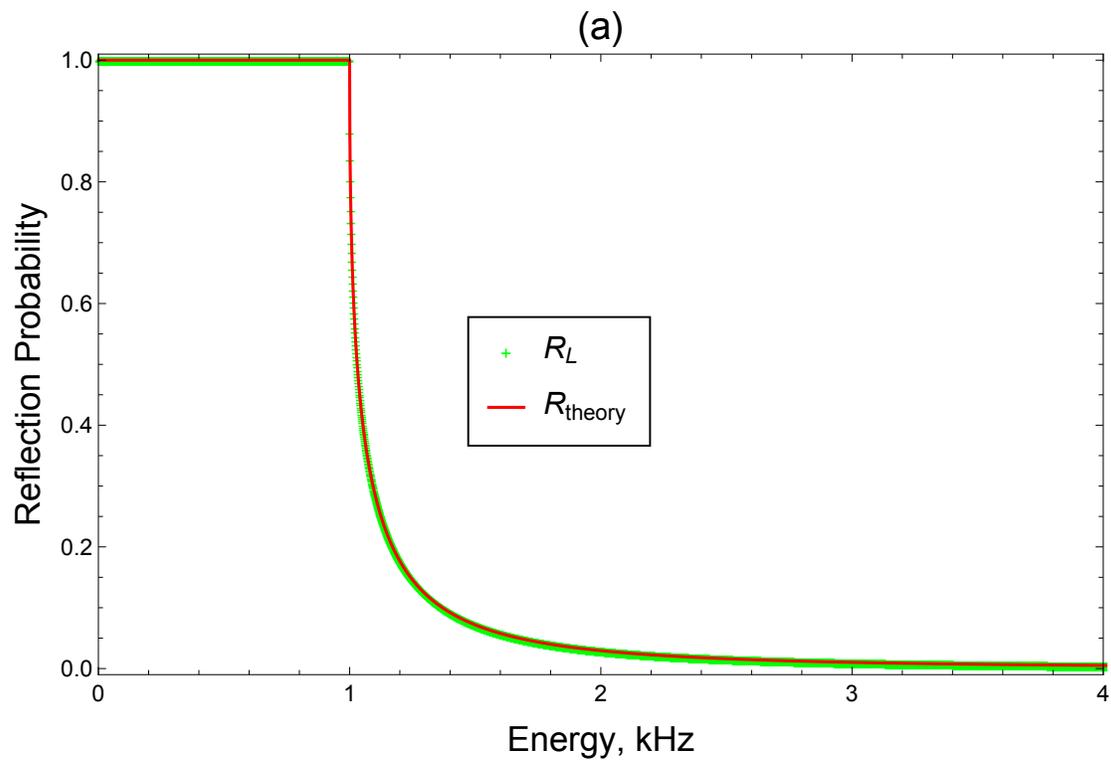
```
anareflPlot = ListLinePlot[  
  Transpose@{energy / (2 *  $\pi$  *  $\hbar$  * 103), anarefl}, PlotStyle -> Red, PlotRange -> All]
```



```
anatransPlot = ListLinePlot[Transpose@{energy / (2 *  $\pi$  *  $\hbar$  * 103), 1 - anarefl},  
  PlotStyle -> {Red, Thickness[.005]}, PlotRange -> All]
```



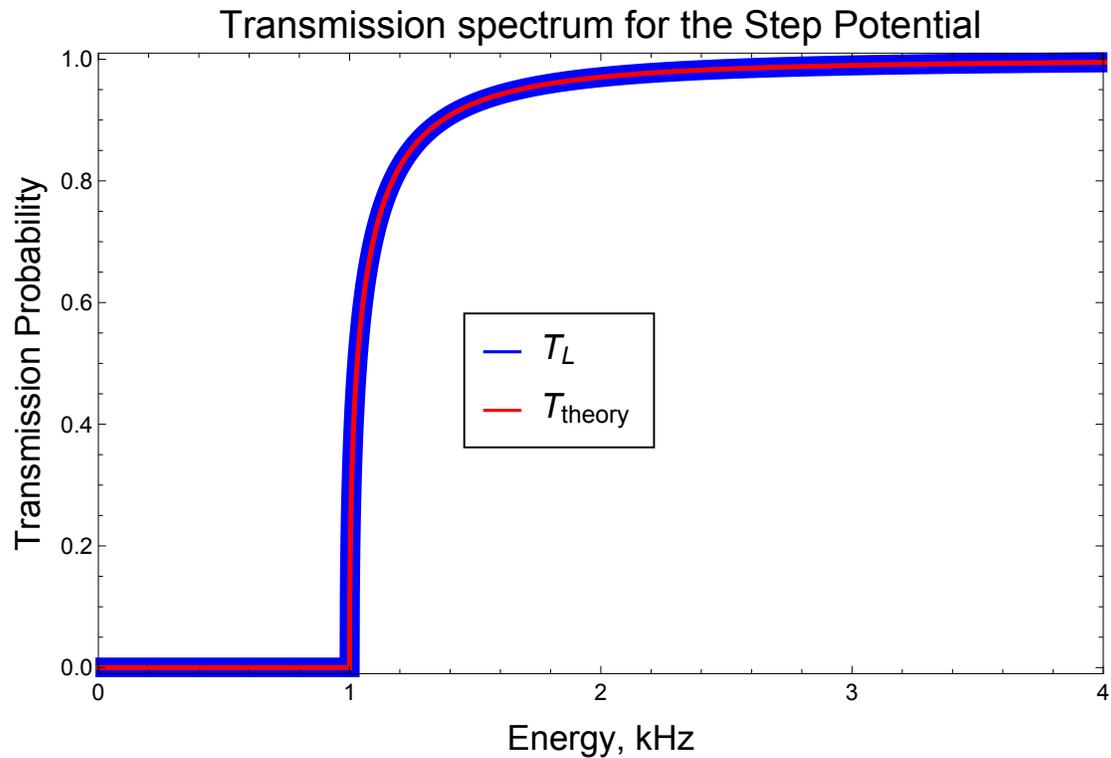
```
Show[reflectionLPlot, anareflPlot, ImageSize -> Large,  
  Epilog -> Inset[Framed[Column[{PointLegend[{Green}, {Style[" RL", 16]},  
    LegendMarkers -> "+"}, LineLegend[{Red}, {Style[" Rtheory", 16]}]}]]]]]
```



```

stepTrans =
Show[transmissionLPlot, anatransPlot, ImageSize -> Large, Epilog -> Inset[Framed[
  Column[{LineLegend[{Blue}, {Style[" TL", 18]}(*LegendMarkers->{"+", Large}*),
    LineLegend[{Red}, {Style[" Ttheory", 18]}]}]]]]

```



```

Export[
"/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis
nb's/possible figs/stepTrans.jpeg", stepTrans];

```

## Appendix C

The single Gaussian barrier potential Mathematica notebook

`gaussianTest.nb`

# Gaussian Test

This is a test of the ImpedanceMethod package by using a gaussian potential.

```
Quit[] (*Quits out of local kernel*)

ClearAll["Global`"] (*clears all set variables with in the global context*)

$ContextPath (*gives the current context*)
{StreamingLoader`, IconizeLoader`,
 CloudObjectLoader`, PacletManager`, System`, Global`}

Get["/Users/avaashby/Desktop/masters-thesis/Mathematica
 Code/ImpedanceMethod/ImpedanceMethod.m"]
Get["/Users/avaashby/Desktop/masters-thesis/Mathematica
 Code/ImpedanceMethod/ImpedanceMethodAmplitude.m"]
Get["/Users/avaashby/Desktop/masters-thesis/Mathematica
 Code/ImpedanceMethod/ImpedanceMethodEigenEnergy.m"]

(*loads the ImpedanceMethod package. There are several ways to do this,
 here is a couple options:
 1) For newer versions of Mathematica you can start with
 Get["dkfj and a small icon will appear to let you search your computers files.
 2) If you know the file path use Get["filePath"] or Needs["filePath"]
 3) If you dont know the file path and the
 icon does not appear, try going to File -> Install -> "
 Type of Item to Install:" select Package, "Source:
 " select from file, and choose the appropriate file, ie
 ImpedanceMethod.m. Now use <<ImpedanceMethod` for the input*)

$ContextPath (*gives the current context,
 ie shows that the ImpedanceMethod package
 loaded correctly and we are now with in its context*)
{ImpedanceMethodEigenEnergy`ImpedanceMethodEigenEnergy`,
 ImpedanceMethodAmplitude`ImpedanceMethodAmplitude`,
 ImpedanceMethod`ImpedanceMethod`, StreamingLoader`, IconizeLoader`,
 CloudObjectLoader`, PacletManager`, System`, Global`}

m = 1.44*^-25; (*mass*)
ħ = 1.05*^-34; (*Planck's constant*)
v0 = 2 * π * ħ * Table[(i) * 10^3, {i, 30, 33}]; (*potential energy height*)
jv0 = Length[v0]; (*length of the potential energy height*)
wx = 15.0*^-6; (*imin/imax*)
l = wx / 1000; (*potential energy step size*)
energy0 = Table[2 * π * ħ * 10^3 * i, {i, 0.001, 40, .0001}];
(*list of the energy that ranges from 0 to 3 with a step size of 1.0101e-3*)
x0 = 0.; (*horizontal shift*)
σ = Sort[{2.1*^-6, 2.5*^-6, 1.6*^-6, 1.0*^-6}, Less]; (*variance*)
nσ = Length[σ]; (*length of the variance*)
```

```
( $\hbar / (m * \sigma[[1]]^2)$ ) / 10^3 (* kinetic energy side of the tunneling condition*)
```

```
0.729167
```

```
newv = Table[v0[[j]] * Exp[ $-\frac{(i - x0)^2}{\sigma[[n]]^2}$ ], {n, 1, n $\sigma$ }, {j, 1, jv0}, {i, -wx, wx, 1}];
```

```
(*list of the potential energy that ranges from -  
5e-6 to 5e-6 with a step size 1*)
```

```
x = Table[i, {i, -wx, wx, 1}]; (*list of the position values for the potential  
energy function, only necessary for plotting the potential energy function*)
```

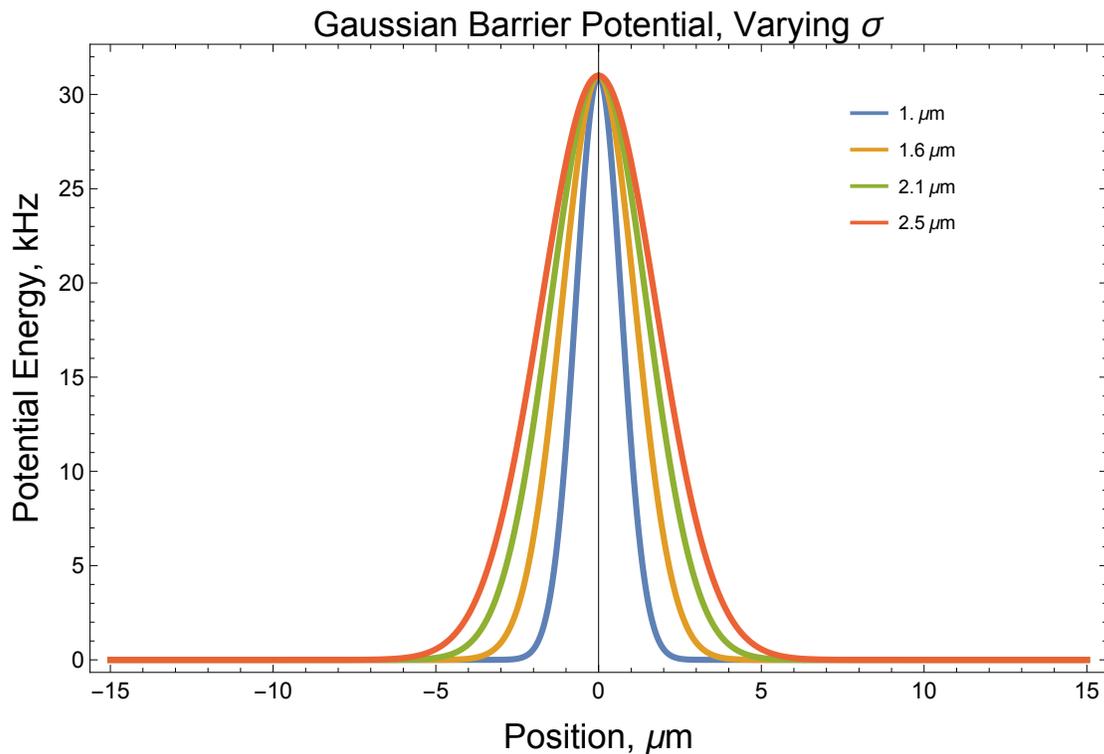
```
Dimensions[newv]
```

```
Dimensions[energy0]
```

```
{4, 4, 2001}
```

```
{399 991}
```

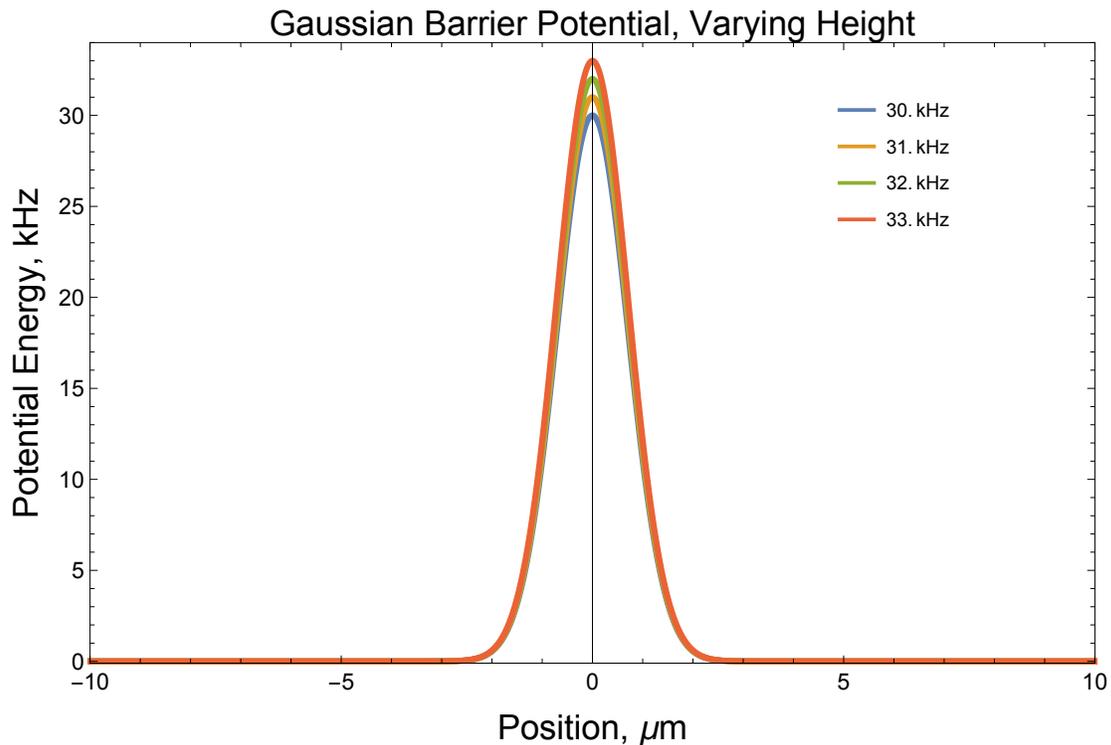
```
potentialEnergy = ListLinePlot [  
Table[Transpose@{x / 10^-6, newv[[i, 2, All]] / (2 *  $\pi$  *  $\hbar$  * 10^3)}, {i, 1, n $\sigma$ }],  
PlotRange -> All, Frame -> True,  
FrameLabel -> {Style["Position,  $\mu$ m", 18], Style["Potential Energy, kHz", 18]},  
PlotLabel -> Style["Gaussian Barrier Potential, Varying  $\sigma$ ", 18],  
FrameTicksStyle -> Medium, ImageSize -> Large, PlotStyle -> Thickness[.006],  
PlotLegends -> Placed[{Table[ $\sigma[[n]] / 10^{-6}$  " $\mu$ m", {n, 1, n $\sigma$ ]}], {.8, .8}]]
```



```
Export [
```

```
"/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis  
nb's/possible figs/potentialEnergy.n.jpeg", potentialEnergy];
```

```
potentialEnergyj = ListLinePlot[
  Table[Transpose[{x/10^-6, newv[[1, j, All]]/(2 * π * ħ * 10^3)}], {j, 1, jv0}],
  PlotRange → {{-10, 10}, {-0.1, 34}}, Frame → True,
  FrameLabel → {Style["Position, μm", 18], Style["Potential Energy, kHz", 18]},
  PlotLabel → Style["Gaussian Barrier Potential, Varying Height", 18],
  FrameTicksStyle → Medium, ImageSize → Large, PlotStyle → Thickness[.006],
  PlotLegends → Placed[{Table[v0[[j]]/(2 * π * ħ * 10^3) "kHz", {j, 1, 4}]}, {.8, .8}]]
```



```
Export[
  "/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis
  nb's/possible figs/potentialEnergyj.jpeg", potentialEnergyj];
```

```
energyoh =
  Table[If[Intersection[newv[[n, j, All]], energy0] == {}, energy0, Delete[energy0,
    Flatten[Position[energy0, Intersection[newv[[n, j, All]], energy0][[1]]]]],
  {n, 1, nσ}, {j, 1, jv0}];
```

```
Dimensions[energyoh]
```

```
{4, 4, 399 990}
```

```
reflectionPoh = Table[ReflectionProbability[
  newv[[n, j, All]], energyoh[[n, j, All]], 1], {n, 1, 4}, {j, 1, 4}];
```

```
Dimensions[reflectionPoh]
```

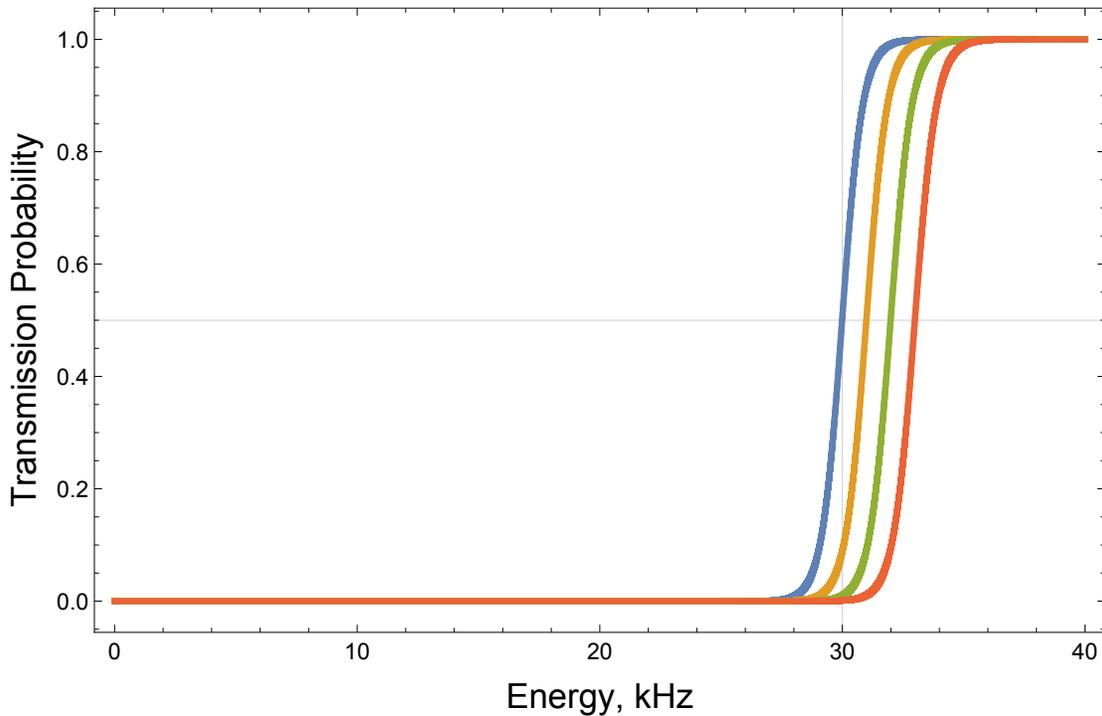
```
{4, 4, 399 990}
```

```
amplitudePoh = Table[ReflectionAmplitude[
  newv[[n, j, All]], energyoh[[n, j, All]], 1], {n, 1, 4}, {j, 1, 4}];
```

```
Dimensions[amplitudePoh]
```

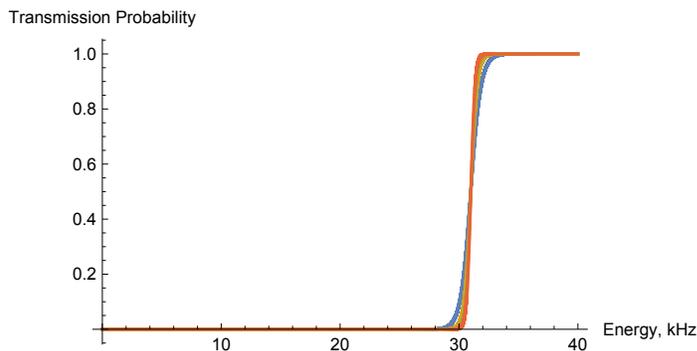
```
{4, 4, 399 990}
```

```
heightTrans = ListLinePlot[Table[
  Transpose@{energyoh[[1, j, All]] / (2 * π * ħ * 10^3), 1 - reflectionPoh[[1, j, All]]},
  {j, 1, 4}], PlotRange → All,
GridLines → {{v0[[1]] / (2 * π * ħ * 10^3)}, {.5}}, Frame → True,
FrameLabel → {Style["Energy, kHz", 18], Style["Transmission Probability", 18]},
(*PlotLabel→Style[
  "Gaussian Barrier Potential Transmission Spectrum, Varying Height", 18, *)
FrameTicksStyle → Medium, ImageSize → Large, PlotStyle → Thickness[.006]
(*, PlotLegends→Placed[{Table[v0[[j]] / (2 * π * ħ * 10^3) "kHz", {j, 1, 4}], {.2, .8}] *)]
```



```
Export[
"/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis
nb's/possible figs/heightTrans.jpeg", heightTrans];

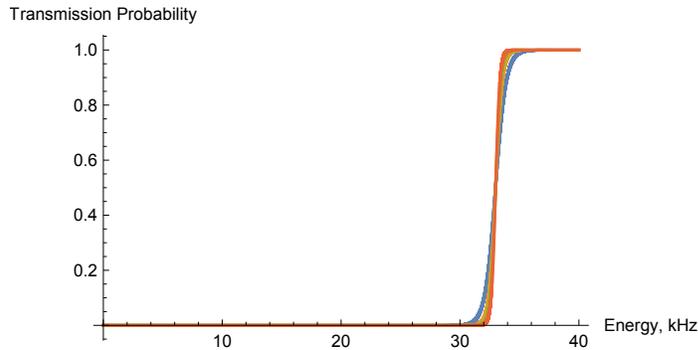
a = ListLinePlot[Table[
  Transpose@{energyoh[[n, 2, All]] / (2 * π * ħ * 10^3), 1 - reflectionPoh[[n, 2, All]]},
  {n, 1, 4}], PlotRange → All, (*GridLines→{{v0[[4]] / (2 * π * ħ * 10^3)}, {.5}}, *)
  AxesLabel → {"Energy, kHz", "Transmission Probability"}]
```



```

b = ListLinePlot[Table[
  Transpose@{energyoh[[n, 4, All]] / (2 * π * ħ * 10^3), 1 - reflectionPoh[[n, 4, All]]},
  {n, 1, 4}], PlotRange → All, (*GridLines→{{v0[[4]]/(2*π*ħ*10^3)}, {.5}},*)
  AxesLabel → {"Energy, kHz", "Transmission Probability"}]

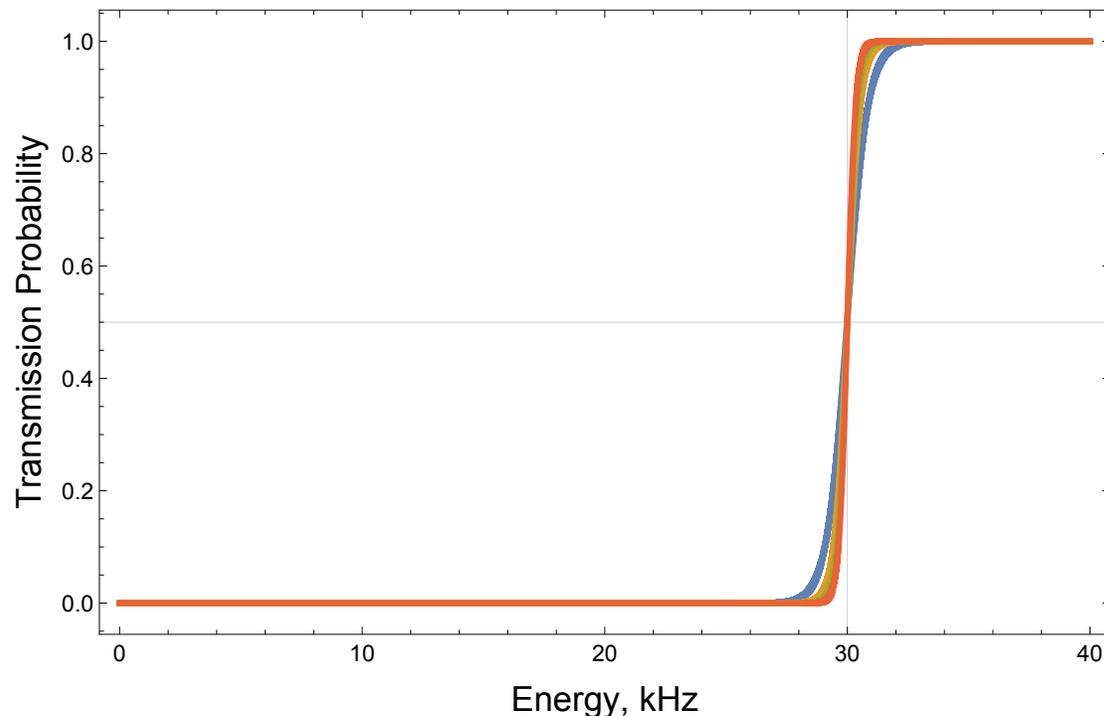
```



```

varTrans = ListLinePlot[Table[
  Transpose@{energyoh[[n, 1, All]] / (2 * π * ħ * 10^3), 1 - reflectionPoh[[n, 1, All]]},
  {n, 1, 4}], PlotRange → All,
  GridLines → {{v0[[1]] / (2 * π * ħ * 10^3)}, {.5}}, Frame → True,
  FrameLabel → {Style["Energy, kHz", 18], Style["Transmission Probability", 18]},
  (*PlotLabel→Style[
    "Gaussian Barrier Potential Transmission Spectrum, Varying Variance", 18],*)
  FrameTicksStyle → Medium, ImageSize → Large, PlotStyle → Thickness[.006]
  (*, PlotLegends→Placed[{Table[σ[[n]]/10^-6 "μm", {n, 1, nσ}]}], {.2, .8}]*)]

```



```

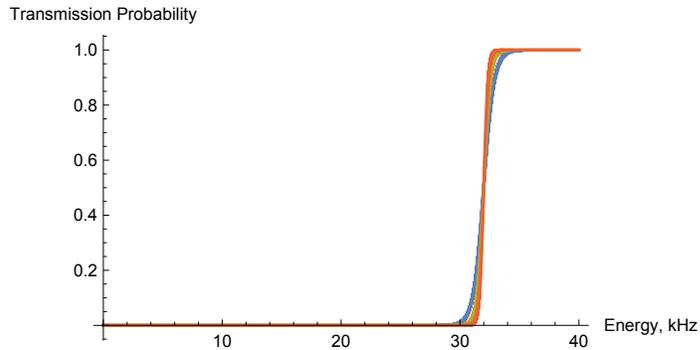
Export [
  "/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis
  nb's/possible figs/varTrans.jpeg", varTrans];

```

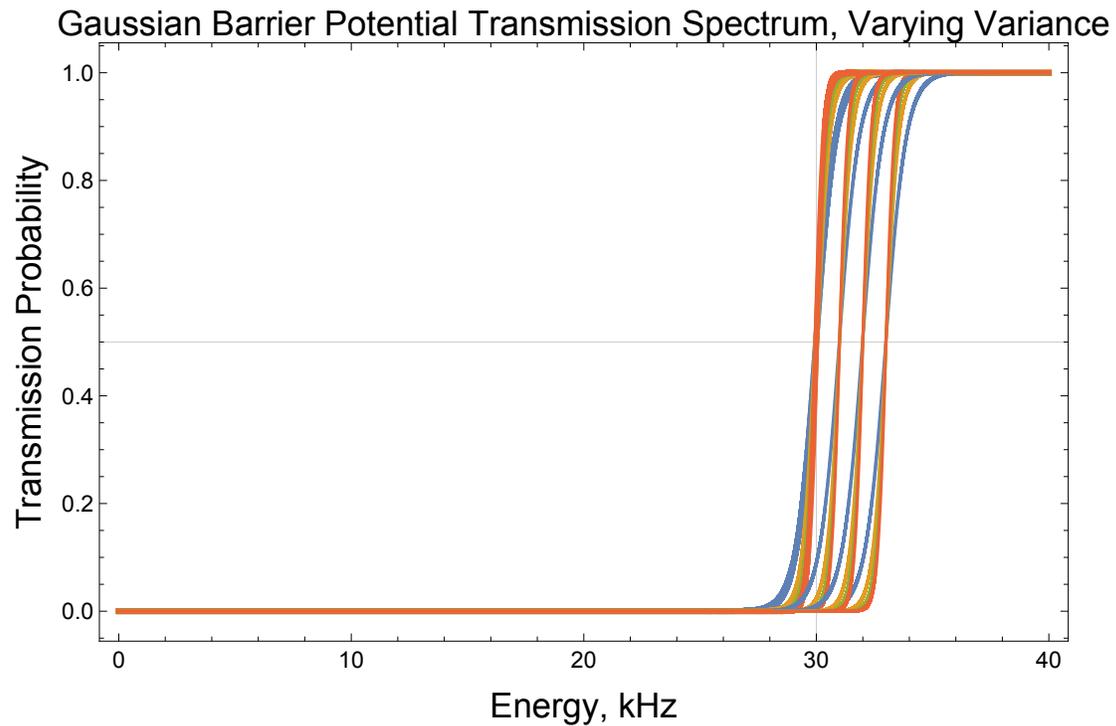
```

c = ListLinePlot[Table[
  Transpose@{energyoh[[n, 3, All]] / (2 * π * ħ * 10^3), 1 - reflectionPoh[[n, 3, All]]},
  {n, 1, 4}], PlotRange → All, (*GridLines→{{v0[[4]]/(2*π*ħ*10^3)}, {.5}},*)
  AxesLabel → {"Energy, kHz", "Transmission Probability"}]

```



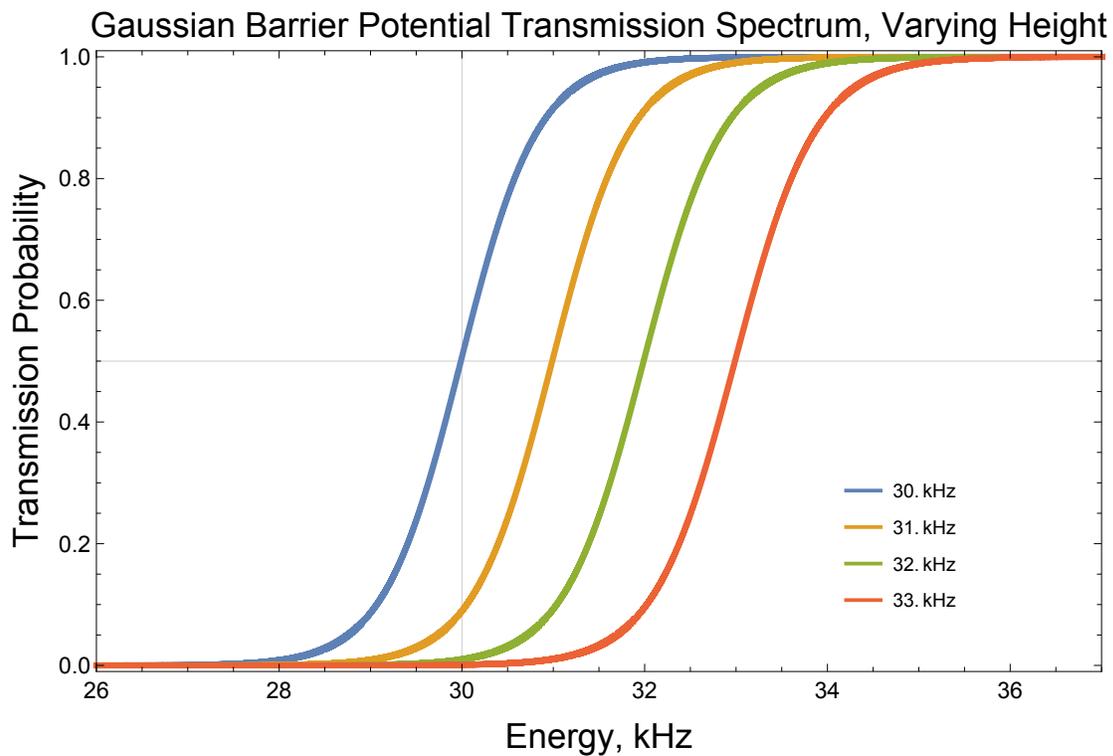
```
Show[varTrans, a, b, c]
```



```

hTransClose = ListLinePlot[Table[
  Transpose@{energyoh[[1, j, All]] / (2 * π * ħ * 10^3), 1 - reflectionPoh[[1, j, All]]},
  {j, 1, 4}], PlotRange → {{26, 37}, {-0.01, 1.01}},
  GridLines → {{v0[[1]] / (2 * π * ħ * 10^3)}, {.5}}, Frame → True, FrameLabel →
  {Style["Energy, kHz", 18], Style["Transmission Probability", 18]}, PlotLabel →
  Style["Gaussian Barrier Potential Transmission Spectrum, Varying Height", 18],
  FrameTicksStyle → Medium, ImageSize → Large, PlotStyle → Thickness[.006],
  PlotLegends → Placed[{Table[v0[[j]] / (2 * π * ħ * 10^3) "kHz", {j, 1, 4}]}, {.8, .2}]]

```

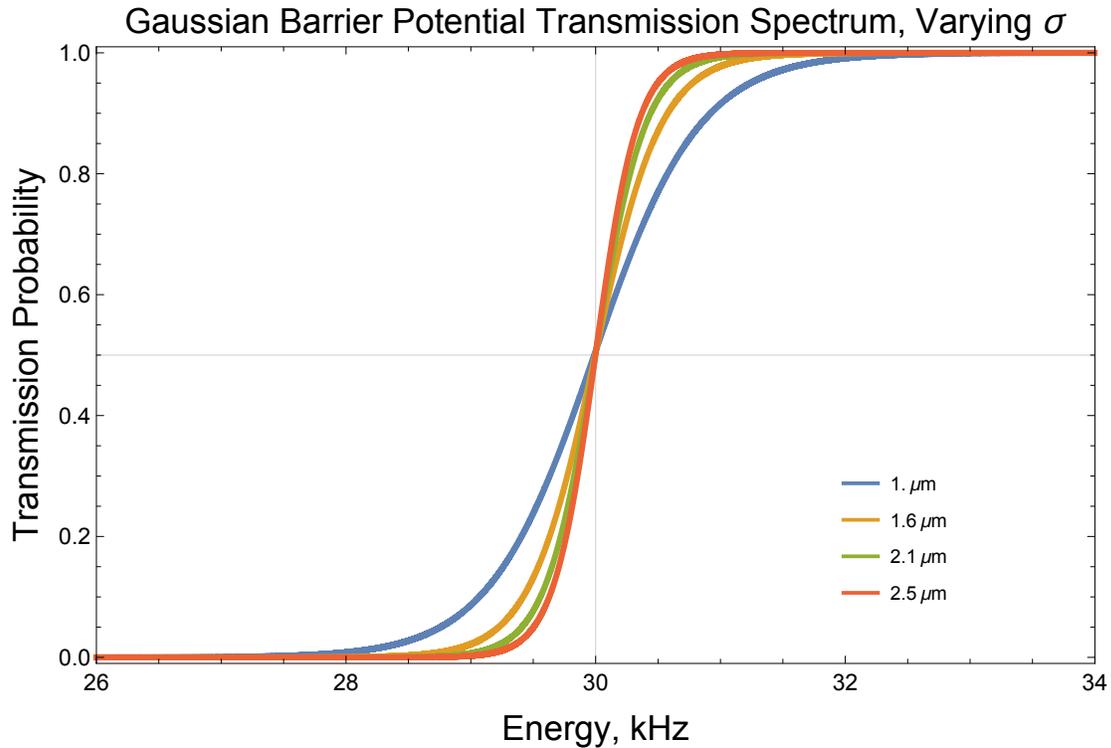


```

Export[
  "/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis
  nb's/possible figs/hTransClose.jpeg", hTransClose];

```

```
wTransClose = ListLinePlot[Table[
  Transpose[{energyoh[[n, 1, All]] / (2 * π * ħ * 10^3), 1 - reflectionPoh[[n, 1, All]]},
  {n, 1, 4}], PlotRange → {{26, 34}, {-0.01, 1.01}},
  GridLines → {{v0[[1]] / (2 * π * ħ * 10^3)}, {.5}}, Frame → True, FrameLabel →
  {Style["Energy, kHz", 18], Style["Transmission Probability", 18]}, PlotLabel →
  Style["Gaussian Barrier Potential Transmission Spectrum, Varying σ", 18],
  FrameTicksStyle → Medium, ImageSize → Large, PlotStyle → Thickness[.006],
  PlotLegends → Placed[{Table[σ[[n]] / 10^-6 "μm", {n, 1, nσ}], {.8, .2}}]
```



```
Export[
  "/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis
  nb's/possible figs/wTransClose.jpeg", wTransClose];
```

```
positionσ1 = Flatten[Position[1 - reflectionPoh[[1, 1, All]], _? (# > .000001 &)]];
Dimensions[positionσ1]
```

```
positionσ2 = Flatten[Position[1 - reflectionPoh[[2, 1, All]], _? (# > .000001 &)]];
Dimensions[positionσ2]
```

```
positionσ3 = Flatten[Position[1 - reflectionPoh[[3, 1, All]], _? (# > .000001 &)]];
Dimensions[positionσ3]
```

```
positionσ4 = Flatten[Position[1 - reflectionPoh[[4, 1, All]], _? (# > .000001 &)]];
Dimensions[positionσ4]
```

```
{156 232}
```

```
{135 628}
```

```
{127 290}
```

```
{122 986}
```

```

tunnelingEnergyσFirst =
  {energyσh[[1, 1, positionσ1[[1]]]], energyσh[[2, 1, positionσ2[[1]]]], energyσh[[
    3, 1, positionσ3[[1]]]], energyσh[[4, 1, positionσ4[[1]]]]} / (2 * π * ħ * 1.0*^3);
tunnelingEnergyσFirst
{24.3768, 26.4372, 27.271, 27.7014}

positionσLast11 =
  Flatten[Position[1 - reflectionPoh[[1, 1, All]], _? (# < .999999 &)]];
Dimensions[positionσLast11]
positionσLast12 =
  Flatten[Position[1 - reflectionPoh[[2, 1, All]], _? (# < .999999 &)]];
Dimensions[positionσLast12]
positionσLast13 =
  Flatten[Position[1 - reflectionPoh[[3, 1, All]], _? (# < .999999 &)]];
Dimensions[positionσLast13]
positionσLast14 =
  Flatten[Position[1 - reflectionPoh[[4, 1, All]], _? (# < .999999 &)]];
Dimensions[positionσLast14]
tunnelingEnergyσLast1 =
  {energyσh[[1, 1, positionσLast11[[Length[positionσLast11]]]],
    energyσh[[2, 1, positionσLast12[[Length[positionσLast12]]]],
    energyσh[[3, 1, positionσLast13[[Length[positionσLast13]]]],
    energyσh[[4, 1, positionσLast14[[Length[positionσLast14]]]]]} / (2 * π * ħ * 1.0*^3)
{360 239}

{337 187}

{328 191}

{323 618}

{36.0249, 33.7197, 32.8201, 32.3628}

positionσ21 = Flatten[Position[1 - reflectionPoh[[1, 2, All]], _? (# > .000001 &)]];
Dimensions[positionσ21]
positionσ22 = Flatten[Position[1 - reflectionPoh[[2, 2, All]], _? (# > .000001 &)]];
Dimensions[positionσ22]
positionσ23 = Flatten[Position[1 - reflectionPoh[[3, 2, All]], _? (# > .000001 &)]];
Dimensions[positionσ23]
positionσ24 = Flatten[Position[1 - reflectionPoh[[4, 2, All]], _? (# > .000001 &)]];
Dimensions[positionσ24]
{147 196}

{126 231}

{117 749}

{113 371}

```

```

tunnelingEnergyσSecond = {energyoh[[1, 2, positionσ21[[1]]],
    energyoh[[2, 2, positionσ22[[1]]]], energyoh[[3, 2, positionσ23[[1]]]],
    energyoh[[4, 2, positionσ24[[1]]]]} / (2 * π * ħ * 1.0*^3);
tunnelingEnergyσSecond
{25.2804, 27.3769, 28.2251, 28.6629}

positionσLast21 =
    Flatten[Position[1 - reflectionPoh[[1, 2, All]], _? (# < .999999 &)]];
Dimensions[positionσLast21]
positionσLast22 =
    Flatten[Position[1 - reflectionPoh[[2, 2, All]], _? (# < .999999 &)]];
Dimensions[positionσLast22]
positionσLast23 =
    Flatten[Position[1 - reflectionPoh[[3, 2, All]], _? (# < .999999 &)]];
Dimensions[positionσLast23]
positionσLast24 =
    Flatten[Position[1 - reflectionPoh[[4, 2, All]], _? (# < .999999 &)]];
Dimensions[positionσLast24]
tunnelingEnergyσLast2 =
    {energyoh[[1, 2, positionσLast21[[Length[positionσLast21]]]],
    energyoh[[2, 2, positionσLast22[[Length[positionσLast22]]]],
    energyoh[[3, 2, positionσLast23[[Length[positionσLast23]]]],
    energyoh[[4, 2, positionσLast24[[Length[positionσLast24]]]]]} / (2 * π * ħ * 1.0*^3)
{371 202}

{347 789}

{338 649}

{334 003}

{37.1212, 34.7799, 33.8659, 33.4013}

positionσ31 = Flatten[Position[1 - reflectionPoh[[1, 3, All]], _? (# > .000001 &)]];
Dimensions[positionσ31]
positionσ32 = Flatten[Position[1 - reflectionPoh[[2, 3, All]], _? (# > .000001 &)]];
Dimensions[positionσ32]
positionσ33 = Flatten[Position[1 - reflectionPoh[[3, 3, All]], _? (# > .000001 &)]];
Dimensions[positionσ33]
positionσ34 = Flatten[Position[1 - reflectionPoh[[4, 3, All]], _? (# > .000001 &)]];
Dimensions[positionσ34]
{138 144}

{116 823}

{108 200}

{103 750}

```

```

tunnelingEnergyσThird = {energyoh[[1, 3, positionσ31[[1]]],
  energyoh[[2, 3, positionσ32[[1]]]], energyoh[[3, 3, positionσ33[[1]]]],
  energyoh[[4, 3, positionσ34[[1]]]]} / (2 * π * ħ * 1.0*^3);
tunnelingEnergyσThird
{26.1856, 28.3177, 29.18, 29.625}

positionσLast31 =
  Flatten[Position[1 - reflectionPoh[[1, 3, All]], _? (# < .999999 &)]];
Dimensions[positionσLast31]
positionσLast32 =
  Flatten[Position[1 - reflectionPoh[[2, 3, All]], _? (# < .999999 &)]];
Dimensions[positionσLast32]
positionσLast33 =
  Flatten[Position[1 - reflectionPoh[[3, 3, All]], _? (# < .999999 &)]];
Dimensions[positionσLast33]
positionσLast34 =
  Flatten[Position[1 - reflectionPoh[[4, 3, All]], _? (# < .999999 &)]];
Dimensions[positionσLast34]
tunnelingEnergyσLast3 =
  {energyoh[[1, 3, positionσLast31[[Length[positionσLast31]]]],
    energyoh[[2, 3, positionσLast32[[Length[positionσLast32]]]],
    energyoh[[3, 3, positionσLast33[[Length[positionσLast33]]]],
    energyoh[[4, 3, positionσLast34[[Length[positionσLast34]]]]]} / (2 * π * ħ * 1.0*^3)
{382 150}

{358 381}

{349 101}

{344 382}

{38.216, 35.8391, 34.9111, 34.4392}

positionσ41 = Flatten[Position[1 - reflectionPoh[[1, 4, All]], _? (# > .000001 &)]];
Dimensions[positionσ41]
positionσ42 = Flatten[Position[1 - reflectionPoh[[2, 4, All]], _? (# > .000001 &)]];
Dimensions[positionσ42]
positionσ43 = Flatten[Position[1 - reflectionPoh[[3, 4, All]], _? (# > .000001 &)]];
Dimensions[positionσ43]
positionσ44 = Flatten[Position[1 - reflectionPoh[[4, 4, All]], _? (# > .000001 &)]];
Dimensions[positionσ44]
{129 078}

{107 407}

{98 645}

{94 124}

```

```

tunnelingEnergyσFourth = {energyσh[[1, 4, positionσ41[[1]]],
  energyσh[[2, 4, positionσ42[[1]]]], energyσh[[3, 4, positionσ43[[1]]]],
  energyσh[[4, 4, positionσ44[[1]]]]} / (2 * π * ħ * 1.0*^3);
tunnelingEnergyσFourth
{27.0922, 29.2593, 30.1355, 30.5876}

positionσLast41 =
  Flatten[Position[1 - reflectionPoh[[1, 4, All]], _? (# < .999999 &)]];
Dimensions[positionσLast41]
positionσLast42 =
  Flatten[Position[1 - reflectionPoh[[2, 4, All]], _? (# < .999999 &)]];
Dimensions[positionσLast42]
positionσLast43 =
  Flatten[Position[1 - reflectionPoh[[3, 4, All]], _? (# < .999999 &)]];
Dimensions[positionσLast43]
positionσLast44 =
  Flatten[Position[1 - reflectionPoh[[4, 4, All]], _? (# < .999999 &)]];
Dimensions[positionσLast44]
tunnelingEnergyσLast4 =
  {energyσh[[1, 4, positionσLast41[[Length[positionσLast41]]]],
    energyσh[[2, 4, positionσLast42[[Length[positionσLast42]]]],
    energyσh[[3, 4, positionσLast43[[Length[positionσLast43]]]],
    energyσh[[4, 4, positionσLast44[[Length[positionσLast44]]]]]} / (2 * π * ħ * 1.0*^3)
{393 084}

{368 965}

{359 545}

{354 756}

{39.3094, 36.8975, 35.9555, 35.4766}

tableData = {"First Tunneling Energy in kHz", SpanFromLeft}, {" $\frac{\text{Variance}}{\text{Height}}$ ",
  σ[[1]] / 10^-6 "μm", σ[[2]] / 10^-6 "μm", σ[[3]] / 10^-6 "μm", σ[[4]] / 10^-6 "μm"},
  Join[{v0[[1]] / (2 * π * ħ * 1.0*^3) "kHz"}, tunnelingEnergyσFirst],
  Join[{v0[[2]] / (2 * π * ħ * 1.0*^3) "kHz"}, tunnelingEnergyσSecond],
  Join[{v0[[3]] / (2 * π * ħ * 1.0*^3) "kHz"}, tunnelingEnergyσThird],
  Join[{v0[[4]] / (2 * π * ħ * 1.0*^3) "kHz"}, tunnelingEnergyσFourth]};

```

```
singleGaussTable = Grid[tableData, Alignment → Center, Spacings → {2, 2},
  Frame → All, Background → {{LightGray, None}, {None, LightGray}}]
```

First Tunneling Energy in kHz				
<u>Variance</u> Height	1. $\mu\text{m}$	1.6 $\mu\text{m}$	2.1 $\mu\text{m}$	2.5 $\mu\text{m}$
30. kHz	24.3768	26.4372	27.271	27.7014
31. kHz	25.2804	27.3769	28.2251	28.6629
32. kHz	26.1856	28.3177	29.18	29.625
33. kHz	27.0922	29.2593	30.1355	30.5876

```
Export [
  "/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis
  nb's/possible figs/singleGaussTable.jpeg", singleGaussTable];
```

```
tableDataLast =
```

```
{{"Last Energy Before Full Transmission in kHz", SpanFromLeft}, {"Variance
  Height
   $\sigma[[1]]/10^{-6}$  " $\mu\text{m}$ ",  $\sigma[[2]]/10^{-6}$  " $\mu\text{m}$ ",  $\sigma[[3]]/10^{-6}$  " $\mu\text{m}$ ",  $\sigma[[4]]/10^{-6}$  " $\mu\text{m}$ "},
  Join[{v0[[1]]/(2 *  $\pi$  *  $\hbar$  * 1.0*^3) "kHz"}, tunnelingEnergy $\sigma$ Last1],
  Join[{v0[[2]]/(2 *  $\pi$  *  $\hbar$  * 1.0*^3) "kHz"}, tunnelingEnergy $\sigma$ Last2],
  Join[{v0[[3]]/(2 *  $\pi$  *  $\hbar$  * 1.0*^3) "kHz"}, tunnelingEnergy $\sigma$ Last3],
  Join[{v0[[4]]/(2 *  $\pi$  *  $\hbar$  * 1.0*^3) "kHz"}, tunnelingEnergy $\sigma$ Last4]}];
```

```
singleGaussLastTable = Grid[tableDataLast, Alignment → Center, Spacings → {2, 2},
  Frame → All, Background → {{LightGray, None}, {None, LightGray}}]
```

Last Energy Before Full Transmission in kHz				
<u>Variance</u> Height	1. $\mu\text{m}$	1.6 $\mu\text{m}$	2.1 $\mu\text{m}$	2.5 $\mu\text{m}$
30. kHz	36.0249	33.7197	32.8201	32.3628
31. kHz	37.1212	34.7799	33.8659	33.4013
32. kHz	38.216	35.8391	34.9111	34.4392
33. kHz	39.3094	36.8975	35.9555	35.4766

```
Export [
  "/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis
  nb's/possible figs/singleGaussLastTable.jpeg", singleGaussLastTable];
```

```
slopeGauss = (.999999 - .000001) / {tunnelingEnergyσLast1 - tunnelingEnergyσFirst,
  tunnelingEnergyσLast2 - tunnelingEnergyσSecond, tunnelingEnergyσLast3 -
  tunnelingEnergyσThird, tunnelingEnergyσLast4 - tunnelingEnergyσFourth}
{{0.0858507, 0.137315, 0.180209, 0.214527}, {0.0844536, 0.13508, 0.177279, 0.211041},
  {0.0831226, 0.132954, 0.174486, 0.207718}, {0.0818517, 0.130921, 0.171821, 0.20454}}
```

```
Dimensions[slopeGauss]
```

```
{4, 4}
```

```
tableDataSlope = {"Transmission Rate of Change", SpanFromLeft}, {" $\frac{\text{Variance}}{\text{Height}}$ ",
  σ[[1]] / 10-6 "μm", σ[[2]] / 10-6 "μm", σ[[3]] / 10-6 "μm", σ[[4]] / 10-6 "μm"},
  Join[{v0[[1]] / (2 * π * ħ * 1.0*3) "kHz"}, slopeGauss[[1, All]]],
  Join[{v0[[2]] / (2 * π * ħ * 1.0*3) "kHz"}, slopeGauss[[2, All]]],
  Join[{v0[[3]] / (2 * π * ħ * 1.0*3) "kHz"}, slopeGauss[[3, All]]],
  Join[{v0[[4]] / (2 * π * ħ * 1.0*3) "kHz"}, slopeGauss[[4, All]]]};
singleGaussSlopeTable = Grid[tableDataSlope, Alignment → Center, Spacings → {2, 2},
  Frame → All, Background → {{LightGray, None}, {None, LightGray}}]
```

Transmission Rate of Change				
$\frac{\text{Variance}}{\text{Height}}$	1. μm	1.6 μm	2.1 μm	2.5 μm
30. kHz	0.0858507	0.137315	0.180209	0.214527
31. kHz	0.0844536	0.13508	0.177279	0.211041
32. kHz	0.0831226	0.132954	0.174486	0.207718
33. kHz	0.0818517	0.130921	0.171821	0.20454

```
Export[
```

```
"/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis
nb's/possible figs/singleGaussSlopeTable.jpeg", singleGaussSlopeTable];
```

```
Dimensions[amplitudePoh]
```

```
{4, 4, 399 990}
```

```
Abs[amplitudePoh[[1, 1, positionσ1[[1]]]]]
```

```
0.999999
```

```

positionEnergyBelow = Flatten[Position[newv[[1, 1, 1001 ;; 2001]],
  _? (# ≥ tunnelingEnergyσFirst[[1]] * (2 * π * ħ * 1.0*^3) - 1 * (2 * π * ħ * 1.0*^3) &)]];
Dimensions[positionEnergyBelow]
positionEnergyAbove = Flatten[Position[newv[[1, 1, All]],
  _? (# ≤ 10 * (2 * π * ħ * 1.0*^3) + 1 * (2 * π * ħ * 1.0*^3) &)]];
Dimensions[positionEnergyAbove]
positionEnergy = Union[positionEnergyBelow, positionEnergyAbove];
Dimensions[positionEnergy]
{34}

{1868}

{1868}

positionEnergyBelow
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,
  18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34}

newv[[1, 1, 1032]] / (2 * π * ħ * 1.0*^3)
24.1666

psi[i_, time_] := Re[Exp[
  ħ * Sqrt[ $\frac{2 * m}{\hbar^2} * (\text{tunnelingEnergy}\sigma\text{First}[[1]] * (2 * \pi * \hbar * 1.0*^3) - \text{newv}[[1, 1, i]])$ ] *
  (x[[i]] - time)] + amplitudePoh[[1, 1, positionσ1[[1]]]] * Exp[- ħ *
  Sqrt[ $\frac{2 * m}{\hbar^2} * (\text{tunnelingEnergy}\sigma\text{First}[[1]] * (2 * \pi * \hbar * 1.0*^3) - \text{newv}[[1, 1, i]])$ ] *
  (x[[i]] - time)]]

psiB[i_, time_] := Re[Exp[
  ħ * Sqrt[ $\frac{2 * m}{\hbar^2} * (\text{tunnelingEnergy}\sigma\text{First}[[1]] * (2 * \pi * \hbar * 1.0*^3) - \text{newv}[[1, 1, i]])$ ] *
  (x[[i]] - time)]]

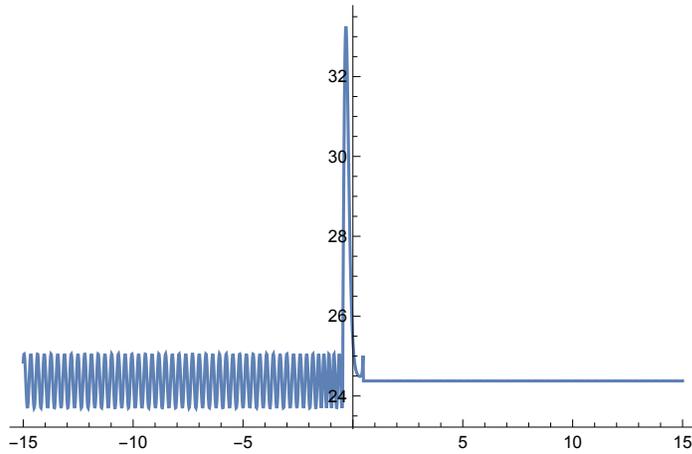
psiT[i_, time_] := Re[Sqrt[1 - Abs[amplitudePoh[[1, 1, positionσ1[[1]]]]]^2] * Exp[
  ħ * Sqrt[ $\frac{2 * m}{\hbar^2} * (\text{tunnelingEnergy}\sigma\text{First}[[1]] * (2 * \pi * \hbar * 1.0*^3) - \text{newv}[[1, 1, i]])$ ] *
  (x[[i]] - time)]]

ψ11 = tunnelingEnergyσFirst[[1]] + Join[Table[psi[i, 0], {i, 1, 970}],
  Table[psiB[i, 0], {i, 971, 1032}], Table[psiT[i, 0], {i, 1033, 2001}]];
Dimensions[ψ11]
x[[900]]
{2001}

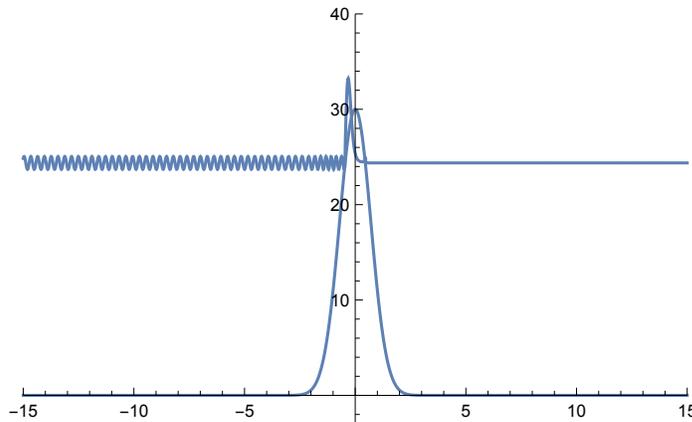
-1.515 × 10-6

```

```
ListPlot[Transpose[{x/10^-6,  $\psi_{11}$ }, Joined → True, PlotRange → All]
```



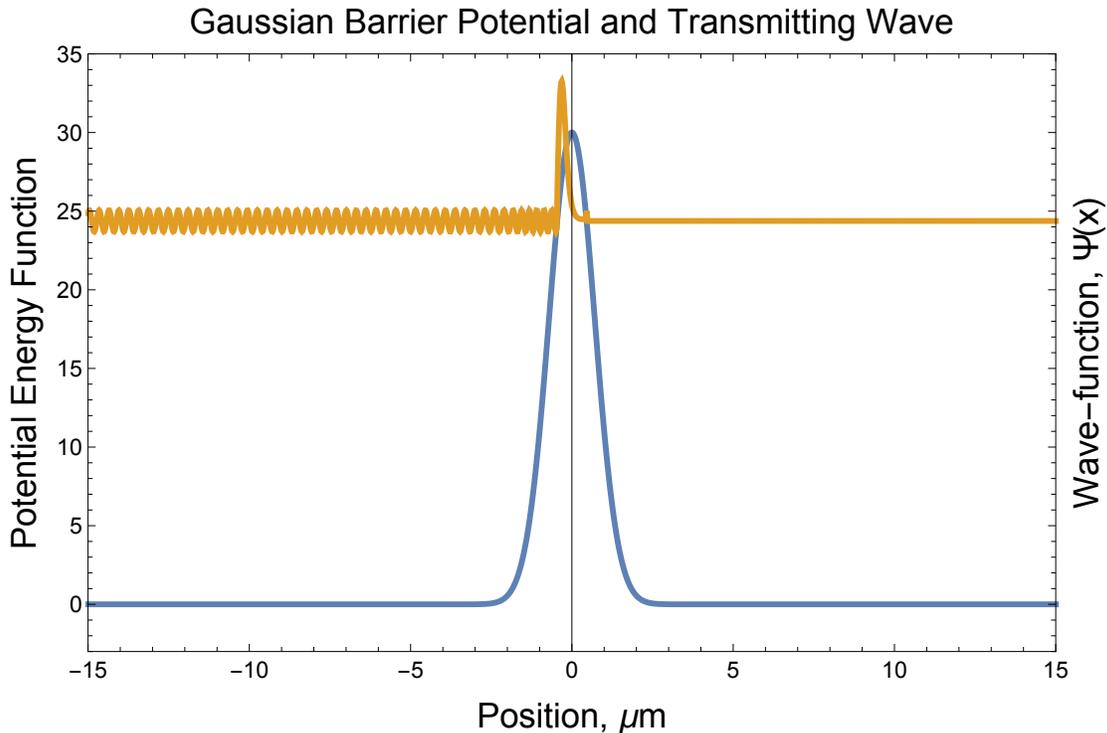
```
Show[ListLinePlot[Transpose[{x/10^-6, newv[[1, 1, All]] / (2 *  $\pi$  *  $\hbar$  * 10^3)},  
PlotRange → {{-15, 15}, {-3, 40}},  
ListLinePlot[Transpose[{x/10^-6,  $\psi_{11}$ }, PlotRange → All]]
```



```

gaussWave = ListLinePlot[{Transpose@{x/10^-6, newv[[1, 1, All]]/(2*π*ħ*10^3)},
  Transpose@{x/10^-6, ψ11}}, PlotRange → {{-15, 15}, {-3, 35}},
  Frame → True, FrameLabel → {{Style["Potential Energy Function", 18],
  Style["Wave-function, Ψ(x)", 18]}, {Style["Position, μm", 18], None}},
  PlotLabel → Style["Gaussian Barrier Potential and Transmitting Wave", 18],
  FrameTicksStyle → Medium, ImageSize → Large, PlotStyle → Thickness[.006]]

```



```

Export[
  "/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis
  nb's/possible figs/gaussWave.jpeg", gaussWave];

σσ = {.01*^-6, 100.0*^-6};
newv = Table[v0[[1]] * Exp[-(i - x0)^2 / σσ[[n]]^2], {n, 1, 2}, {i, -wx, wx, 1}];
Dimensions[newv]
{2, 2001}

energyLimit = Table[If[Intersection[newv[[n, All]], energy0] == {},
  energy0, Delete[energy0, Flatten[Position[energy0,
  Intersection[newv[[n, All]], energy0][[1]]]]]], {n, 1, 2}];
Dimensions[energyLimit]
{2, 399 990}

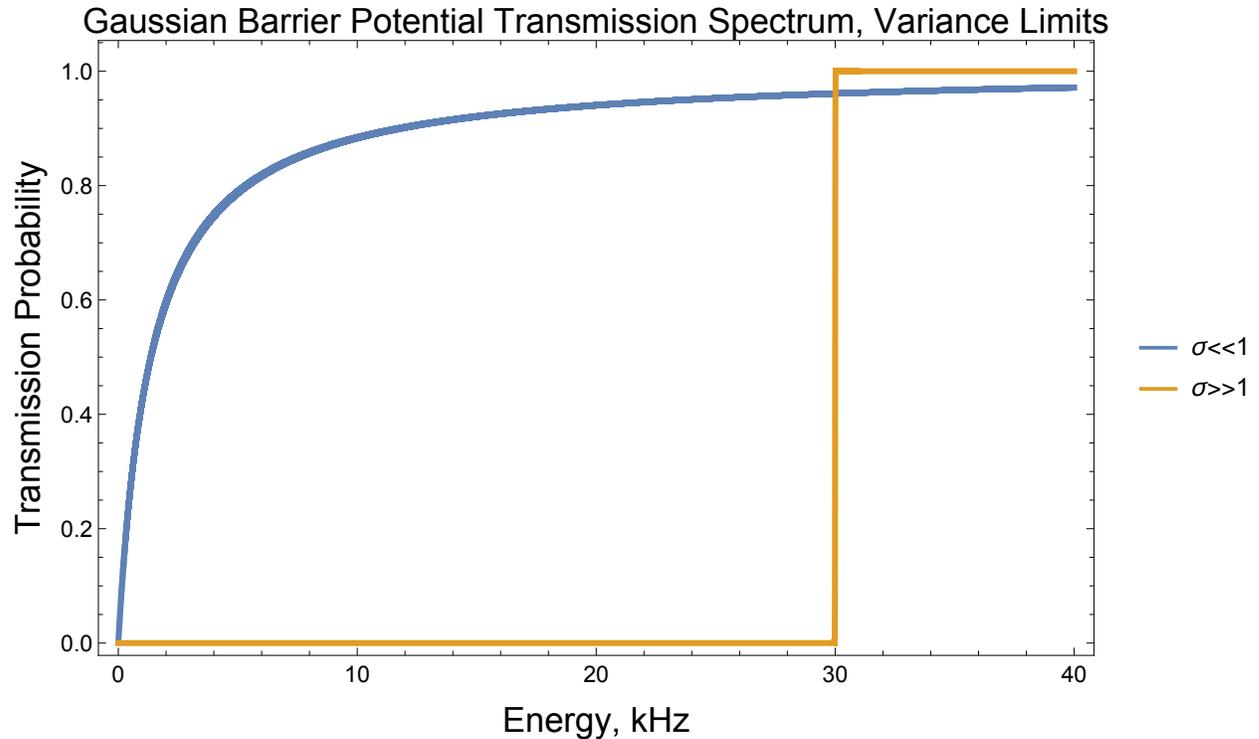
reflectionLimit = Table[
  ReflectionProbability[newv[[n, All]], energyLimit[[n, All]], 1], {n, 1, 2}];
Dimensions[reflectionLimit]
{2, 399 990}

```

```

varianceLimits = ListLinePlot[
  {Transpose@{energyLimit[[1, All]] / (2 *  $\pi$  *  $\hbar$  * 10^3), 1 - reflectionLimit[[1, All]]},
  Transpose@{energyLimit[[2, All]] / (2 *  $\pi$  *  $\hbar$  * 10^3),
    Join[Table[0, {i, 1, 298474}], 1 - reflectionLimit[[2, 298475 ;; 399990]]]},
  PlotRange -> All, Frame -> True, FrameLabel -> {Style["Energy, kHz", 18],
    Style["Transmission Probability", 18]}, PlotLabel ->
    Style["Gaussian Barrier Potential Transmission Spectrum, Variance Limits", 18],
  FrameTicksStyle -> Medium, ImageSize -> Large,
  PlotStyle -> Thickness[.006], PlotLegends -> {" $\sigma \ll 1$ ", " $\sigma \gg 1$ "}]

```



```

Export[
  "/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis
  nb's/possible figs/varianceLimits.jpeg", varianceLimits];

```

## Appendix D

The Mathematica package for the the eigenenergies

`ImpedanceMethodEigenEnergy.m`

# General Impedance Method Eigen Energy

02.11.2016

Anderson Group

## MetaPackage:

Transmission line analogy of resonance tunneling phenomena: The generalized impedance concept by A.N. Khondker

J. Appl. Phys., vol. 63, no. 10, p.5191, 1988

Wave transmission through periodic, quasiperiodic, and random one-dimensional finite lattices by Braulio Gutierrez-Medina

Am. J. Phys., vol. 81, no. 2, p. 104, 2013

This package used the general impedance method to numerically evaluate the eigen-energies of an arbitrary potential energy function. The general impedance method evaluates the potential energy function from the output end to the reference plane for the left looking input impedance,  $\zeta_{\text{inputL}}$ , and the input end of the potential to the reference plane for the right looking input impedance,  $\zeta_{\text{inputR}}$ . The condition for the eigen-energies is  $\zeta_{\text{inputL}} = \zeta_{\text{inputR}}$ .

The input values for this package are:

$v = \text{Table}[\text{function}[i], \{i, i_{\text{min}}, i_{\text{max}}, \text{stepsize}\}]$  : the arbitrary potential energy function as a list of points.

$\text{energy} = \text{Table}[i, \{i, i_{\text{min}}, i_{\text{max}}, \text{stepsize}\}]$  : the span of energy values that an arbitrary wavefunction might have. Note, this could also be a list of one.

$l = \text{constant}$  : step size, or more specifically the value  $x[[2]] - x[[1]]$ . This is the step size at which the potential energy function takes.

The output values for this package are the eigen-energies for the potential.

The variables defined within the package are:

$m =$  effective mass of the wave packet in units of kg (for rubidium)

$\hbar =$  Planck's constant in units of Js

$k =$  propagation constant

$\zeta_0 =$  characteristic impedance

$\zeta_{\text{load}} =$  load impedance

$\zeta_{\text{input}} =$  input impedance

## Begin Package

```
BeginPackage ["ImpedanceMethodEigenEnergy`ImpedanceMethodEigenEnergy`"]
```

```
ImpedanceMethodEigenEnergy`ImpedanceMethodEigenEnergy`
```

## Usage Message

```
EigenEnergy::usage =
  "EigenEnergy[v,energy,l] computes the eigen-energies by using the
    general impedance method. The arguments are v, energy,
    and l where they are the potential energy vector function,
    energy vector function, and the step size, respectively.
  m is the effective mass of the wave packet and  $\hbar$  is Planck's constant
    in units of Js, both are defined within the package.
  v must be of the form v=Table[function[i],{i,imin,imax,l}]
  energy must be of the form energy=Table[i,{i,imin,imax,l}]"
```

## Warning Messages

```
EigenEnergy::badv =
  "The potential energy argument `1` must be in the form of a list.";
EigenEnergy::badenergy = "The energy argument `1` must be in the form of a list.";
EigenEnergy::badl =
  "The step size argument `1` is not within the appropriate bounds. l must be
    greater than 0 and less than the Thermal de Broglie Wavelength, `2`.";
```

## Options

### Private Context

```
Begin["`Private`"]
ImpedanceMethodEigenEnergy`ImpedanceMethodEigenEnergy`Private`
```

## Function Definitions (Public and Private)

```
m = 1.44*^-25;
 $\hbar$  = 1.054571726*^-34;
```

```

EigenEnergy[v_, energy_, l_] :=
Module[{kL, ζ0L, ζloadL, ζinputL, kR, ζ0R, ζloadR, ζinputR,
  i, reference, above, below, aboveAndbelow, resonantEnergy},
Which[
  ! (VectorQ[v]), Message[EigenEnergy::"badv", v],
  ! (VectorQ[energy]), Message[EigenEnergy::"badenergy", energy],
  !  $\left(0 < 1 \leq \frac{(2 * N[\pi, 10] * \hbar)}{\text{Sqrt}[2 * m * \text{Max}[energy] ]}\right)$ ,
  Message[EigenEnergy::"badl", l,  $\frac{(2 * N[\pi, 10] * \hbar)}{\text{Sqrt}[2 * m * \text{Max}[energy] ]}$ ]
];
reference = If[EvenQ[Length[v]], Length[v] / 2, (Length[v] - 1) / 2];
Do[
  kL =  $i * \text{Sqrt}[2 * m * (\text{energy} - v[[i]])] / \hbar$ ;
  ζ0L =  $-i * \hbar * kL / m$ ;
  If[i == Length[v], ζloadL = ζ0L, ζloadL = ζinputL];
  ζinputL =  $\zeta_{0L} * \frac{\zeta_{loadL} * \text{Cosh}[kL * l] - \zeta_{0L} * \text{Sinh}[kL * l]}{\zeta_{0L} * \text{Cosh}[kL * l] - \zeta_{loadL} * \text{Sinh}[kL * l]}$ ,
  {i, Length[v], reference, -1}];
Do[
  kR =  $i * \text{Sqrt}[2 * m * (\text{energy} - v[[i]])] / \hbar$ ;
  ζ0R =  $-i * \hbar * kR / m$ ;
  If[i == 1, ζloadR = ζ0R, ζloadR = ζinputR];
  ζinputR =  $\zeta_{0R} * \frac{\zeta_{loadR} * \text{Cosh}[kR * l] - \zeta_{0R} * \text{Sinh}[kR * l]}{\zeta_{0R} * \text{Cosh}[kR * l] - \zeta_{loadR} * \text{Sinh}[kR * l]}$ , {i, 1, reference, 1}];

above = Flatten[Position[Abs[ζinputL/ζinputR], _? (# > .999 &)]];
below = Flatten[Position[Abs[ζinputL/ζinputR], _? (# < 1.001 &)]];
aboveAndbelow = Intersection[above, below];
resonantEnergy =
  Table[energy[[aboveAndbelow[[i]]]], {i, 1, Length[aboveAndbelow]}]
]

```

## End Private Context

```
End[]
```

```
ImpedanceMethodEigenEnergy`ImpedanceMethodEigenEnergy`Private`
```

## End Package

```
EndPackage[]
```

## Appendix E

The double rectangular barrier potential Mathematica notebook

`doubleRectPot.nb`

# Impedance Method Rectanglar Double Barrier

```

Quit[]

ClearAll["Global`"]

Get["/Users/avaashby/Desktop/masters-thesis/Mathematica
Code/ImpedanceMethod/ImpedanceMethod.m"]
Get["/Users/avaashby/Desktop/masters-thesis/Mathematica
Code/ImpedanceMethod/ImpedanceMethodAmplitude.m"]
Get["/Users/avaashby/Desktop/masters-thesis/Mathematica
Code/ImpedanceMethod/ImpedanceMethodEigenEnergy.m"]

$ContextPath

{ImpedanceMethodEigenEnergy`ImpedanceMethodEigenEnergy`,
 ImpedanceMethodAmplitude`ImpedanceMethodAmplitude`,
 ImpedanceMethod`ImpedanceMethod`, StreamingLoader`, IconizeLoader`,
 CloudObjectLoader`, PacletManager`, System`, Global`}

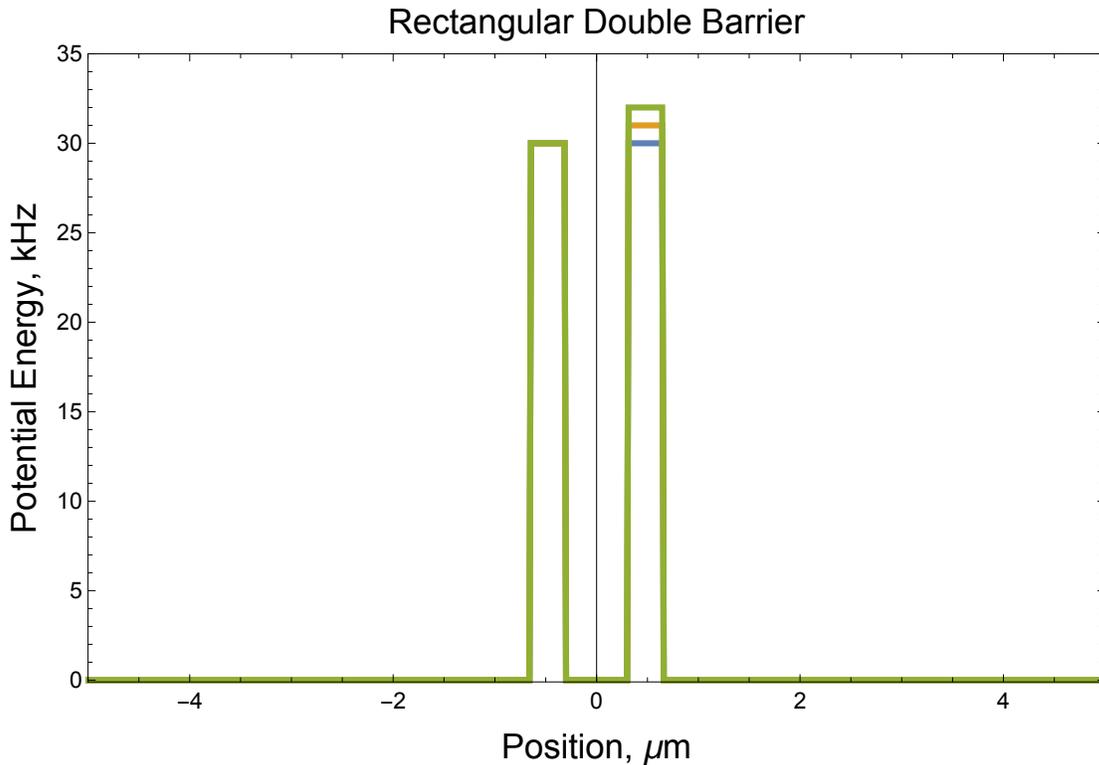
ħ = 1.054*^-34;
v01 = 2 * π * ħ * 30.0*^3;
v02 = 2 * π * ħ * 1.0*^3 * Table[i, {i, 30, 32}];
wid = .333*^-6;
sep =  $\frac{.625*^-6}{2}$ ;
mxm = 15.0*^-6;
l = mxm / 1000;

energy = 2 * π * ħ * 10^3 * Table[i, {i, .001, 31 (*56*), .0001 (*0.001*)}];
Dimensions[energy]
{309 991}

potential = Table[-v01 * UnitStep[i + sep] + v01 * UnitStep[i + (sep + wid)] +
  v02[[j]] * (-UnitStep[i - (sep + wid)] + UnitStep[i - sep]),
  {j, 1, Length[v02]}, {i, -mxm, mxm, 1}];
x = Table[i, {i, -mxm, mxm, 1}];
Dimensions[potential]
{3, 2001}

```

```
doubRectPot =
ListLinePlot[Table[Transpose@{x/10^-6, potential[[j, All]]/(2 * π * ħ * 10^3)},
  {j, 1, Length[v02]}], PlotRange → {{-5, 5}, {-0.1, 35}}, Frame → True,
FrameLabel → {Style["Position, μm", 18], Style["Potential Energy, kHz", 18]},
PlotLabel → Style["Rectangular Double Barrier", 18],
FrameTicksStyle → Medium, ImageSize → Large, PlotStyle → {Thickness[.006]}
```



```
Export [
"/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis
nb's/possible figs/doubRectPot.jpeg", doubRectPot];

energyGood = Table[If[Intersection[potential[[j, All]], energy] == {},
  energy, Delete[energy, Flatten[Position[energy,
  Intersection[potential[[j, All]], energy][[1]]]]]], {j, 1, Length[v02]}];
Dimensions[energyGood]
{3, 309 990}

t = AbsoluteTime[];
reflectCoef1 =
  ReflectionProbability[potential[[1, All]], energyGood[[1, All]], 1];
Dimensions[reflectCoef1]
AbsoluteTime[] - t

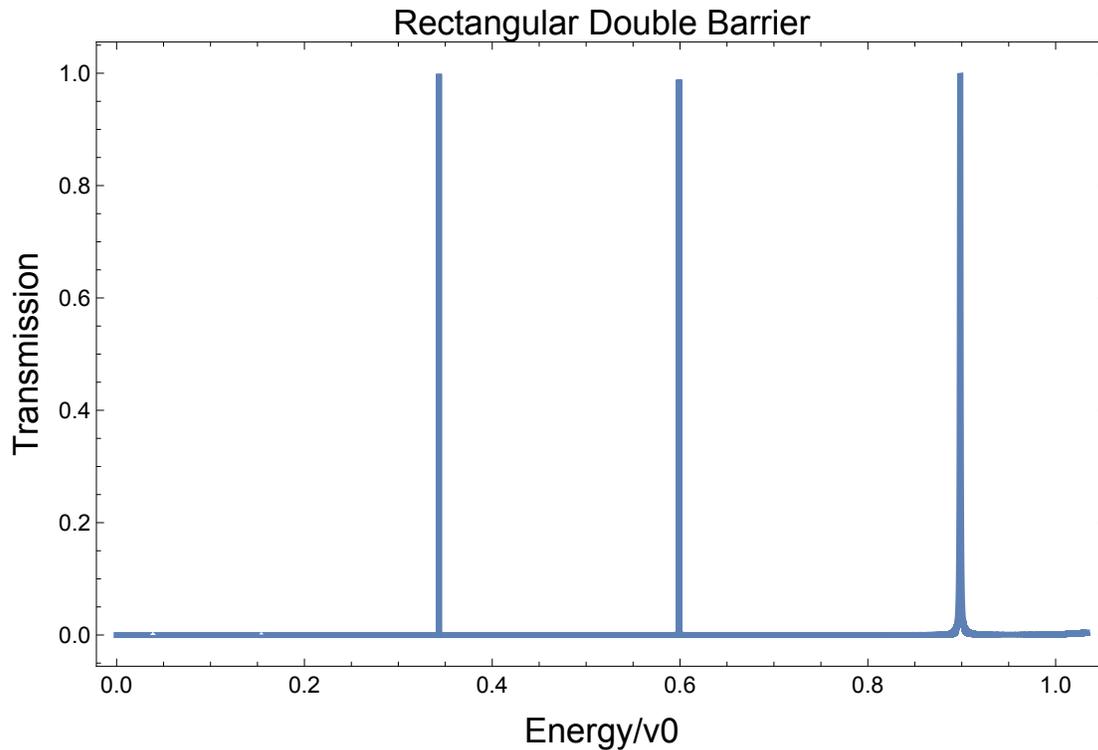
{309 990}

190.303008
```

```

p1 =
ListLinePlot[Transpose[{energyGood[[1, All]] / (2 *  $\pi$  *  $\hbar$  * 30.0*^3), 1 - reflectCoef1},
PlotRange -> All, Frame -> True,
FrameLabel -> {Style["Energy/v0", 18], Style["Transmission", 18]},
PlotLabel -> Style["Rectangular Double Barrier", 18],
FrameTicksStyle -> Medium, ImageSize -> Large, PlotStyle -> {Thickness[.006]}]

```



```

t = AbsoluteTime[];
reflectAmpl = ReflectionAmplitude[potential[[1, All]], energyGood[[1, All]], 1];
Dimensions[reflectAmpl]
AbsoluteTime[] - t

```

```

t = AbsoluteTime[];
eigenEnergy1 = EigenEnergy[potential[[1, All]], energyGood[[1, All]], 1];
Dimensions[eigenEnergy1]
AbsoluteTime[] - t

```

```
{42}
```

```
194.415754
```

```

DeleteDuplicates[eigenEnergy1 / (2 *  $\pi$  *  $\hbar$  * 1.0*^3), Abs[#1 - #2] < .1 &]
{1.1141, 4.4342, 9.883, 17.2733, 26.039}

```

Avik Dutt analytical solution:

```

m = 1.44*^-25;
k1[nrg_] := Sqrt[2 * m * nrg / ħ^2]
k2[nrg_] := Sqrt[2 * m * (v01 - nrg) / ħ^2]
k3[nrg_] := Sqrt[2 * m * (v02[[1]] - nrg) / ħ^2]
g = 2 * wid;
b = 2 * wid + sep;

tProb[nrg_] :=
16 * ((4 * Cosh[k3[nrg] * wid]^2 + ((k1[nrg] - k3[nrg]) / k1[nrg])^2 * Sinh[k3[nrg] * wid]^2) *
(4 * Cosh[k2[nrg] * wid]^2 + ((k1[nrg] - k2[nrg]) / k1[nrg])^2 * Sinh[k2[nrg] * wid]^2) +
(((k1[nrg] + k3[nrg]) / k1[nrg]) * ((k1[nrg] + k2[nrg]) / k1[nrg]) * Sinh[k2[nrg] * wid] *
Sinh[k3[nrg] * wid])^2 + ((k1[nrg] + k3[nrg]) / k1[nrg]) * ((k1[nrg] + k2[nrg]) / k1[nrg]) *
Sinh[k2[nrg] * wid] * Sinh[k3[nrg] * wid] * (8 * Cos[k1[nrg] * (wid + sep)] *
Cosh[k3[nrg] * wid] * Cosh[k2[nrg] * wid] - 4 * Sin[k1[nrg] * (wid + sep)]) *
(((k1[nrg] - k2[nrg]) / k1[nrg]) * Sinh[k2[nrg] * wid] * Cosh[k3[nrg] * wid] +
((k1[nrg] - k3[nrg]) / k1[nrg]) * Sinh[k3[nrg] * wid] * Cosh[k2[nrg] * wid]) -
2 * Cos[k1[nrg] * (wid + sep)] * ((k1[nrg] - k3[nrg]) / k1[nrg]) * Sinh[k3[nrg] * wid] *
((k1[nrg] - k2[nrg]) / k1[nrg]) * Sinh[k2[nrg] * wid]))^-1

```

H. Yamamoto analytical solution

```

β[nrg_] := Sqrt[2 * m * (v01 - nrg) / ħ^2]
kk[nrg_] := Sqrt[2 * m * nrg / ħ^2]
h[nrg_] := 2 * Sqrt[nrg * (v01 - nrg)] * Cosh[β[nrg] * wid] * Cos[kk[nrg] * sep] -
(2 * nrg - v01) * Sinh[β[nrg] * wid] * Sin[kk[nrg] * sep]
trans[nrg_] :=
(1 + (v01^2 * Sinh[β[nrg] * wid]^2 * h[nrg]^2) / (4 * nrg^2 * (v01 - nrg)^2))^-1

```

Ava Ashby analytic solution:

```

m = 1.44*^-25;
k[nrg_] := Sqrt[2 * m * nrg / ħ^2];
κ1[nrg_] := Sqrt[2 * m * (v01 - nrg) / ħ^2];
κ2[nrg_] := Sqrt[2 * m * (v02[[1]] - nrg) / ħ^2];
x1 = sep;
x2 = wid + sep;
Δ = x1 - x2;

```

```

rProbAva[nrg_] :=
  Abs [ ( Exp [ - 2 * i * k [ nrg ] * x2 ] * ( ( ( k [ nrg ] ^ 2
    + κ2 [ nrg ] ) * Exp [ 2 * i * k [ nrg ] * x1 ] *
      Cosh [ κ1 [ nrg ] * Δ ] * Sinh [ κ2 [ nrg ] * Δ ] + ( ( k [ nrg ] ^ 2
        + κ1 [ nrg ] ) *
          Exp [ - 2 * i * k [ nrg ] * x1 ] * Sinh [ κ1 [ nrg ] * Δ ] * Cosh [ κ2 [ nrg ] * Δ ] +
            ( ( ( k [ nrg ] ^ 3
              - κ1 [ nrg ] * κ2 [ nrg ] )
                k [ nrg ] ) * Sin [ 2 * k [ nrg ] * x1 ] +
                  ( ( κ1 [ nrg ]
                    - κ2 [ nrg ] )
                      κ1 [ nrg ] ) * i * k [ nrg ] * Cos [ 2 * k [ nrg ] * x1 ] ) *
                    Sinh [ κ1 [ nrg ] * Δ ] * Sinh [ κ2 [ nrg ] * Δ ] ) ) / ( Exp [ - 2 * i * k [ nrg ] * x1 ] *
                      ( ( ( k [ nrg ] ^ 2
                        - κ2 [ nrg ] )
                          κ2 [ nrg ] ) * Sinh [ κ2 [ nrg ] * Δ ] * Cosh [ κ1 [ nrg ] * Δ ] +
                            ( ( k [ nrg ] ^ 2
                              - κ1 [ nrg ] )
                                κ1 [ nrg ] ) * Sinh [ κ1 [ nrg ] * Δ ] * Cosh [ κ2 [ nrg ] * Δ ] - 2 * i * k [ nrg ] *
                                  Cosh [ κ2 [ nrg ] * Δ ] * Cosh [ κ1 [ nrg ] * Δ ] ) + Sinh [ κ2 [ nrg ] * Δ ] * Sinh [ κ1 [ nrg ] * Δ ] *
                                ( ( ( k [ nrg ] ^ 3
                                  + κ1 [ nrg ] * κ2 [ nrg ] )
                                    κ1 [ nrg ] * κ2 [ nrg ] ) * Sin [ 2 * k [ nrg ] * x1 ] -
                                      ( ( κ1 [ nrg ]
                                        + κ2 [ nrg ] )
                                          κ1 [ nrg ] ) * i * k [ nrg ] * Cos [ 2 * k [ nrg ] * x1 ] ) ) ) ] ^ 2 ;

c [ nrg_ , rAmp_ ] := Exp [ - κ1 [ nrg ] * x2 ] * ( 1 / 2 *
  ( rAmp * Exp [ i * k [ nrg ] * x2 ] * ( ( i * k [ nrg ] / κ1 [ nrg ] + 1 ) - Exp [ - i * k [ nrg ] * x2 ] * ( ( i * k [ nrg ] / κ1 [ nrg ] - 1 ) ) )

cc [ nrg_ , rAmp_ ] := Exp [ κ1 [ nrg ] * x2 ] * ( 1 / 2 *
  ( Exp [ - i * k [ nrg ] * x2 ] * ( 1 + i * k [ nrg ] / κ1 [ nrg ] ) + rAmp * Exp [ i * k [ nrg ] * x2 ] * ( 1 - i * k [ nrg ] / κ1 [ nrg ] ) )

psiI [ i_ , nrg_ , rAmp_ ] := Re [ Exp [ i * k [ nrg ] * x [ [ i ] ] ] + rAmp * Exp [ - i * k [ nrg ] * x [ [ i ] ] ] ]
psiII [ i_ , nrg_ , rAmp_ ] :=
  Re [ c [ nrg , rAmp ] * Exp [ - κ1 [ nrg ] * x [ [ i ] ] ] + cc [ nrg , rAmp ] * Exp [ κ1 [ nrg ] * x [ [ i ] ] ] ]

d [ nrg_ , rAmp_ ] :=
  1 / 2 * Exp [ i * k [ nrg ] * x1 ] * ( rAmp * Exp [ i * k [ nrg ] * x2 ] * ( ( i * k [ nrg ] / κ1 [ nrg ] - κ1 [ nrg ] / i * k [ nrg ] ) *
    Sinh [ κ1 [ nrg ] * Δ ] + Exp [ - i * k [ nrg ] * x2 ] *
      ( 2 * Cosh [ κ1 [ nrg ] * Δ ] - ( κ1 [ nrg ] / i * k [ nrg ] + i * k [ nrg ] / κ1 [ nrg ] ) * Sinh [ κ1 [ nrg ] * Δ ] ) )

f [ nrg_ , rAmp_ ] := 1 / 2 * Exp [ - i * k [ nrg ] * x1 ] * ( rAmp * Exp [ i * k [ nrg ] * x2 ] *
  ( 2 * Cosh [ κ1 [ nrg ] * Δ ] + ( i * k [ nrg ] / κ1 [ nrg ] + κ1 [ nrg ] / i * k [ nrg ] ) * Sinh [ κ1 [ nrg ] * Δ ] ) -
  Exp [ - i * k [ nrg ] * x2 ] * ( i * k [ nrg ] / κ1 [ nrg ] - κ1 [ nrg ] / i * k [ nrg ] ) * Sinh [ κ1 [ nrg ] * Δ ] )

```

```

psiIII[i_, nrg_, rAmp_] :=
  Re[d[nrg, rAmp] * Exp[i * k[nrg] * x[[i]]] + f[nrg, rAmp] * Exp[-i * k[nrg] * x[[i]]]]

tAmp[rAmp_] := Sqrt[1 - Abs[rAmp]^2]
gg[nrg_, rAmp_] := Sqrt[1 - Abs[rAmp]^2] * Exp[i * k[nrg] * x2 + κ2[nrg] * x2]

g4[nrg_, rAmp_] :=
  
$$\frac{1}{2} * \text{Exp}[\kappa2[\text{nrg}] * x1] * \left( d[\text{nrg}, \text{rAmp}] * \text{Exp}[i * k[\text{nrg}] * x1] * \left( 1 - \frac{i * k[\text{nrg}]}{\kappa2[\text{nrg}]} \right) + \right.$$


$$\left. f[\text{nrg}, \text{rAmp}] * \text{Exp}[-i * x1] * \left( 1 + \frac{i * k[\text{nrg}]}{\kappa2[\text{nrg}]} \right) \right)$$


gg4[nrg_, rAmp_] := 
$$\frac{1}{2} * \text{Exp}[-\kappa2[\text{nrg}] * x1] * \left( d[\text{nrg}, \text{rAmp}] * \text{Exp}[i * k[\text{nrg}] * x1] * \left( \frac{i * k[\text{nrg}]}{\kappa2[\text{nrg}]} + 1 \right) + \right.$$


$$\left. f[\text{nrg}, \text{rAmp}] * \text{Exp}[-i * x1] * \left( 1 - \frac{i * k[\text{nrg}]}{\kappa2[\text{nrg}]} \right) \right)$$


psiIV[i_, nrg_, rAmp_] :=
  Re[g4[nrg, rAmp] * Exp[-κ2[nrg] * x[[i]]] (*+gg4[nrg, rAmp] * Exp[κ2[nrg] * x[[i]]] *)
  (*gg[nrg, rAmp] * Exp[-κ2[nrg] * x[[i]]] *)

psiV[i_, nrg_, rAmp_] := Re[tAmp[rAmp] * Exp[i * k[nrg] * x[[i]]]]

positII = Flatten[Position[potential[[1, All]], _? (# == v01 &)]]
{958, 959, 960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 970, 971, 972, 973, 974, 975,
 976, 977, 978, 979, 980, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030,
 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044}

positIV = Flatten[Position[potential[[1, All]], _? (# == v02[[1]] &)]]
{958, 959, 960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 970, 971, 972, 973, 974, 975,
 976, 977, 978, 979, 980, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030,
 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044}

positAmp = Flatten[Position[energyGood[[1, All]], _? (# == eigenEnergy1[[13]] &)]];

rAmp = reflectAmp1[[269 501]]
Sqrt[1 - Abs[rAmp]^2]
-0.000825029 + 0.00075589 i

0.999999

energyGood[[1, 269 501]] / (2 * π * ħ * 1.0*^3)
26.951

```

```

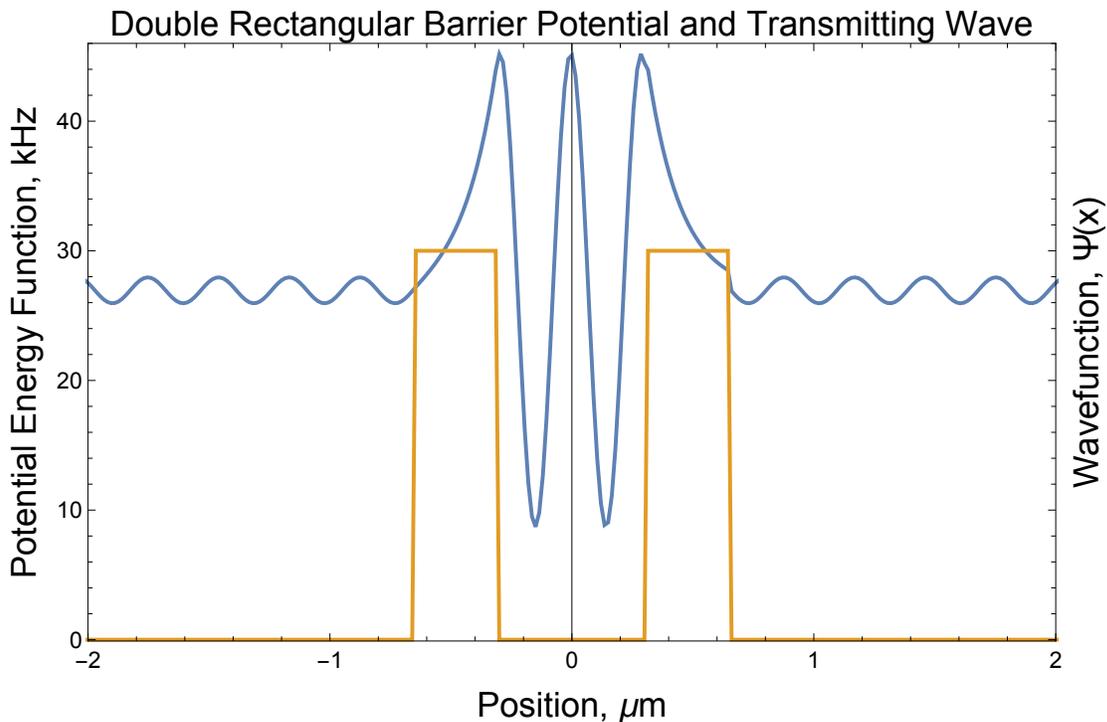
ψI = Flatten@Table[psiI[i, energyGood[[1, 269 501]], rAmp], {i, 1, 957}];
ψII = Flatten@Table[psiII[i, energyGood[[1, 269 501]], rAmp] * 1, {i, 958, 980}];
ψIII = Flatten@Table[psiIII[i, energyGood[[1, 269 501]], rAmp] * 1, {i, 981, 1021}];
ψIV = Flatten@Table[psiIV[i, energyGood[[1, 269 501]], rAmp] * 1, {i, 1022, 1044}];
ψV = Flatten@Table[psiV[i, energyGood[[1, 269 501]], rAmp], {i, 1045, Length[x]}];
Ψ = energyGood[[1, 269 501]] / (2 * π * ħ * 1.0*^3) + Join[ψI, ψII, ψIII, ψIV, ψV];
Dimensions[Ψ]
{2001}

```

```

doubRectWave = ListPlot[{{Transpose[{x / 10^-6, Ψ},
  Transpose[{x / 10^-6, potential[[1, All]] / (2 * π * ħ * 10^3)}]},
  Joined → True, PlotRange → {{-2, 2}, {-0.1, 46}}, Frame → True, FrameLabel →
  {{Style["Potential Energy Function, kHz", 18], Style["Wavefunction, Ψ(x)", 18]},
  {Style["Position, μm", 18], None}}, PlotLabel →
  Style["Double Rectangular Barrier Potential and Transmitting Wave", 18],
  FrameTicksStyle → Medium, ImageSize → Large, PlotStyle → Thickness[.004]}]

```



```

Export [
  "/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis
  nb's/possible figs/doubRectWave.jpeg", doubRectWave];

```

```

DeleteDuplicates[eigenEnergy1 / (2 * π * ħ * 1.0*^3), Abs[#1 - #2] < .1 &]
{1.1141, 4.4342, 9.883, 17.2733, 26.039}

```

```

energyBetter = (2 * π * ħ * 1.0*^3) * Join[Table[i, {i, .1, 1.0, .1}],
  Table[i, {i, 1.0, 1.2, .00000004}], Table[i, {i, 1.2, 4.3, .1}],
  Table[i, {i, 4.3, 4.7, .000001}], Table[i, {i, 4.7, 9.5, .1}],
  Table[i, {i, 9.5, 10.5, .00001}], Table[i, {i, 10.5, 17.0, .1}],
  Table[i, {i, 17.0, 18.0, .00001}], Table[i, {i, 18.0, 26.0, .1}],
  Table[i, {i, 26.0, 27.0, .001}], Table[i, {i, 27.0, 40.0, .1}]];
Dimensions[energyBetter]
{5 601 374}

t = AbsoluteTime[];
reflectCoefBetter1 = ReflectionProbability[potential[[1, All]], energyBetter, 1];
Dimensions[reflectCoefBetter1]
AbsoluteTime[] - t
{5 601 374}

3859.546387

t = AbsoluteTime[];
reflectAmpBetter1 = ReflectionAmplitude[potential[[1, All]], energyBetter, 1];
Dimensions[reflectAmpBetter1]
AbsoluteTime[] - t
{3 852 347}

2495.542985

t = AbsoluteTime[];
eigenEnergyBetter1 = EigenEnergy[potential[[1, All]], energyBetter, 1];
Dimensions[eigenEnergyBetter1]
AbsoluteTime[] - t

{2888}

9532.782682

dataTable = {"Eigen-Energies in kHz", SpanFromLeft},
  DeleteDuplicates[eigenEnergyBetter1 / (2 * π * ħ * 1.0*^3), Abs[#1 - #2] < .1 &]];
eigenEnergy30 = Grid[dataTable, Alignment → Center, Spacings → {2, 2},
  Frame → All, Background → {{None, None}, {LightGray, None}}]

```

Eigen-Energies in kHz				
1.11401	4.43412	9.883	17.2732	26.039

```

Export[
"/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis
nb's/possible figs/eigenEnergy30.jpeg", eigenEnergy30];

eigs30 = DeleteDuplicates[eigenEnergyBetter1 / (2 * π * ħ * 1.0*^3), Abs[#1 - #2] < .1 &];

```

```

energyAnal1 = 2 * π * ħ * 10^3 * Table[i, {i, .01, 40, .00001}];
Dimensions[energyAnal1]
energyAnal = If[Intersection[potential[[1, All]], energyAnal1] == {},
  energyAnal1, Delete[energyAnal1, Flatten[Position[energyAnal1,
    Intersection[potential[[1, All]], energyAnal1][[1]]]]];
Dimensions[energyAnal]
{3 999 001}
{3 999 000}

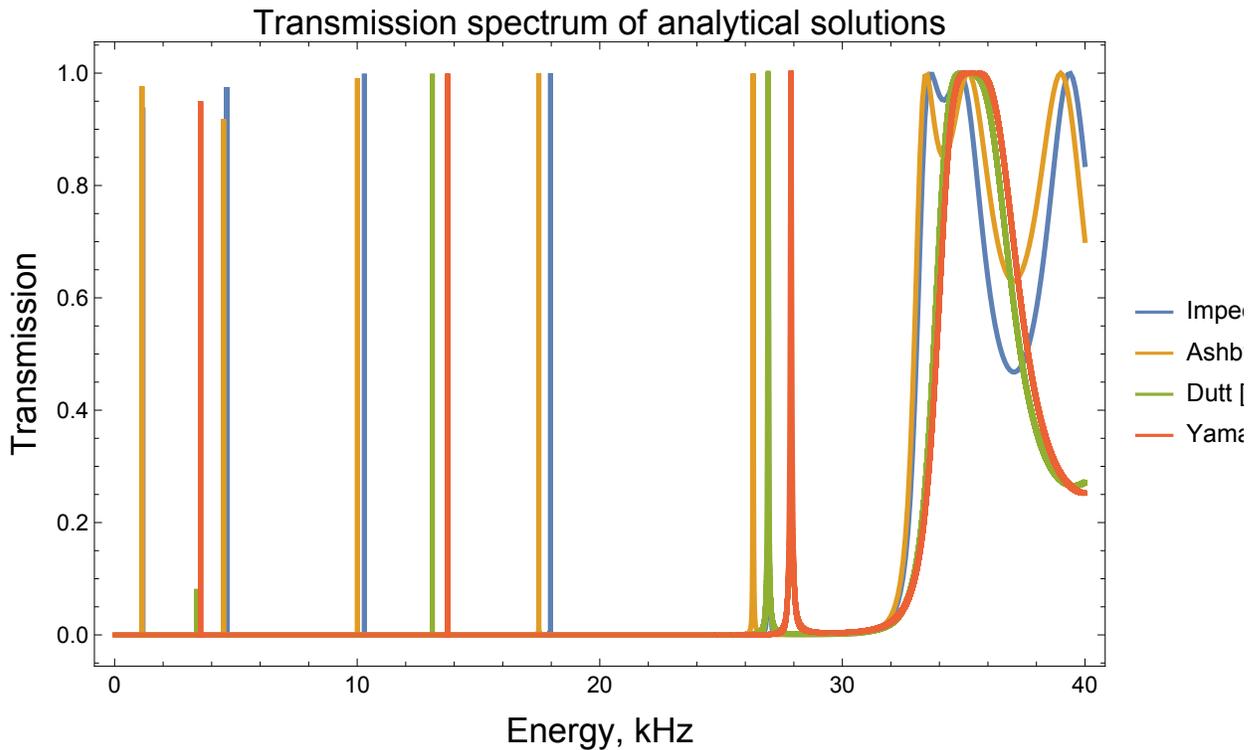
aBimped = Transpose@{energyBetter / (2 * π * ħ * 1.0*^3), 1 - reflectCoefBetter1};
Dimensions[aBimped]
aBAva = Transpose@{energyBetter / (2 * π * ħ * 1.0*^3), 1 - rProbAva[energyBetter]};
Dimensions[aBAva]
aBDutt = Transpose@{energyAnal / (2 * π * ħ * 1.0*^3), tProb[energyAnal]};
Dimensions[aBDutt]
aBYama = Transpose@{energyAnal / (2 * π * ħ * 1.0*^3), trans[energyAnal]};
Dimensions[aBYama]
{5 601 374, 2}
{5 601 374, 2}
{3 999 000, 2}
{3 999 000, 2}

```

```

avasBest = ListLinePlot[{aBimped, aBAva, aBDutt, aBYama}, PlotRange → All,
  Frame → True, FrameLabel → {Style["Energy, kHz", 18], Style["Transmission", 18]},
  PlotLabel → Style["Transmission spectrum of analytical solutions", 18],
  FrameTicksStyle → Medium, ImageSize → Large, PlotStyle → Thickness[.005],
  PlotLegends → {"Impedance Method", "Ashby", "Dutt [11]", "Yamamoto [10]"}
(*, Epilog → {Directive[{Thick, Black, Dashed}],
  Table[Line[{{eigs30[[i]], 0}, {eigs30[[i]], 1}}, {i, 1, 5}]}*)]

```

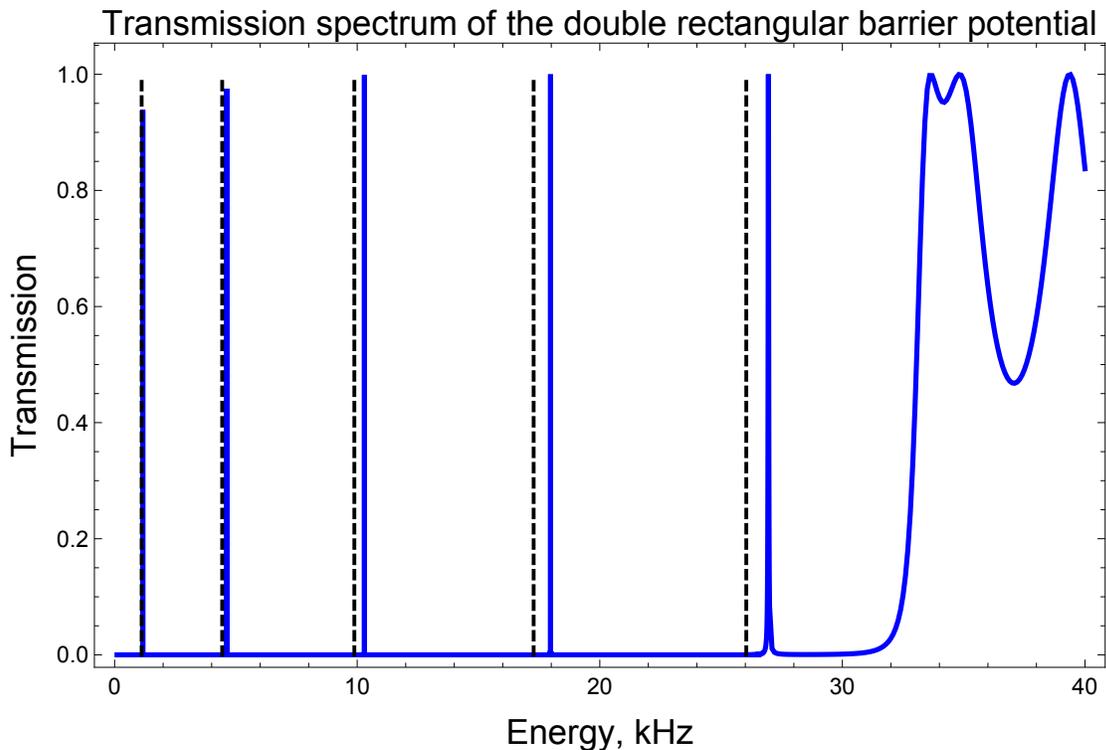


```

Export [
  "/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis
  nb's/possible figs/avasBest.jpeg", avasBest];

```

```
doubRectTrans =
ListLinePlot[Transpose[{energyBetter / (2 * π * ħ * 1.0*^3), 1 - reflectCoefBetter1},
PlotRange → All, Frame → True,
FrameLabel → {Style["Energy, kHz", 18], Style["Transmission", 18]}, PlotLabel →
Style["Transmission spectrum of the double rectangular barrier potential", 18],
FrameTicksStyle → Medium, ImageSize → Large,
PlotStyle → Directive[Blue, Thickness[.005]],
Epilog → {Directive[{Thick, Black, Dashed}],
Table[Line[{{eigs30[[i]], 0}, {eigs30[[i]], 1}}, {i, 1, 5}]}]
```



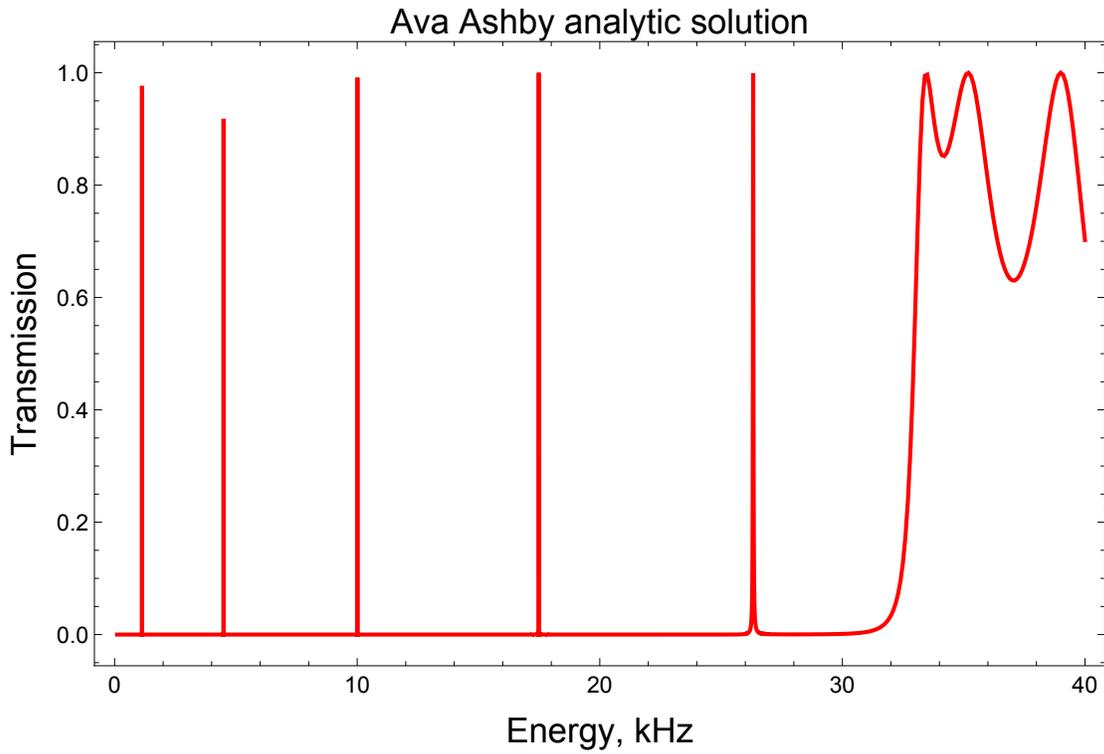
```
Export[
"/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis
nb's/possible figs/doubRectTrans.jpeg", doubRectTrans];

positionBet = Flatten[Position[1 - reflectCoefBetter1, _? (# > .4 &)]];
energyResonanceBetter = Table[
energyBetter[[positionBet[[i]]]] / (2 * π * ħ * 1.0*^3), {i, Length[positionBet]}];
refEnergy = DeleteDuplicates[energyResonanceBetter, Abs[#1 - #2] < .1 &]
{1.16198, 4.62346, 10.2976, 17.9693, 26.934, 33., 33.1, 33.2, 33.4, 33.5,
33.6, 33.7, 33.9, 34., 34.1, 34.2, 34.4, 34.5, 34.6, 34.7, 34.9, 35., 35.1,
35.2, 35.4, 35.5, 35.6, 35.7, 35.9, 36., 36.1, 36.2, 36.4, 36.5, 36.6, 36.7,
36.9, 37., 37.1, 37.2, 37.4, 37.5, 37.6, 37.7, 37.9, 38., 38.1, 38.2, 38.4,
38.6, 38.7, 38.8, 38.9, 39., 39.1, 39.2, 39.4, 39.5, 39.7, 39.8, 39.9, 40.}
```

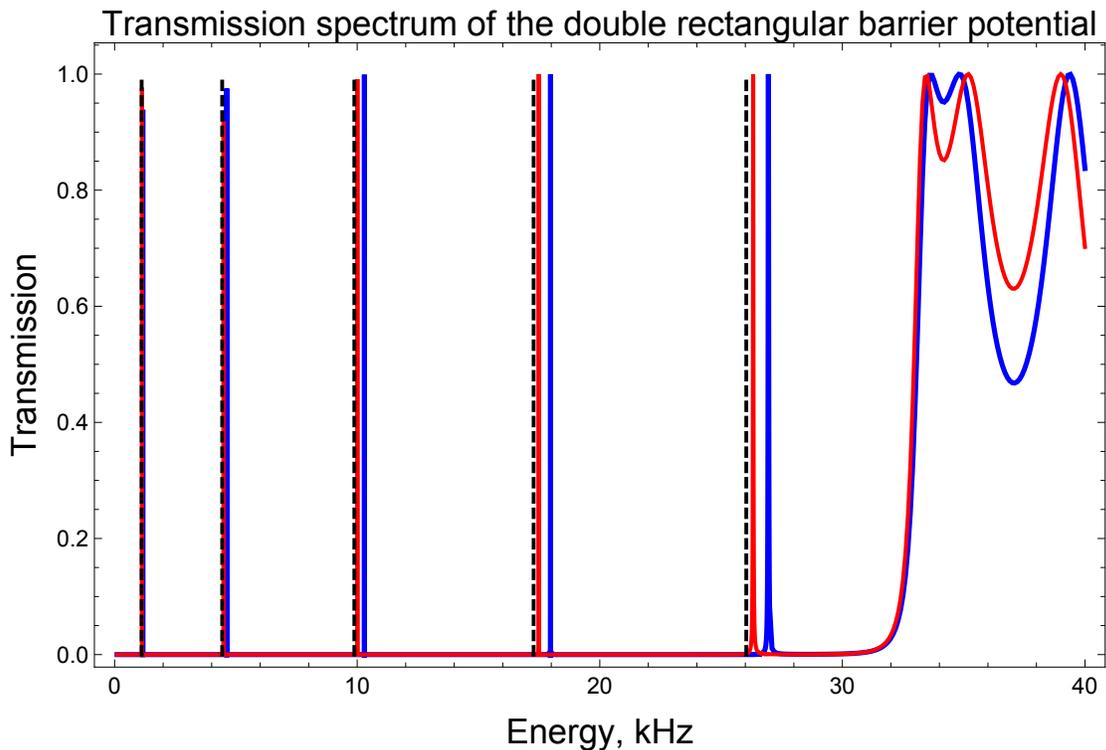
```

analytDoubRect = ListLinePlot[
  Transpose@{energyBetter / (2 *  $\pi$  *  $\hbar$  * 1.0*^3), 1 - rProbAva[energyBetter]},
  PlotRange -> All, PlotStyle -> Directive[Red, Thickness[.004]],
  PlotLabel -> Style["Ava Ashby analytic solution", 18], Frame -> True,
  FrameLabel -> {Style["Energy, kHz", 18], Style["Transmission", 18]},
  FrameTicksStyle -> Medium, ImageSize -> Large]

```



```
doubRectAnalyt = Show[doubRectTrans, analytDoubRect]
```



```
Export[
```

```
  "/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis  
  nb's/possible figs/doubRectAnalyt.jpeg", doubRectAnalyt];
```

```
energyBetter31 = (2 *  $\pi$  *  $\hbar$  * 1.0*^3) * Join[Table[i, {i, .1, 1.1, .1}],  
  Table[i, {i, 1.1, 1.3, .000001}], Table[i, {i, 1.3, 4.4, .1}],  
  Table[i, {i, 4.4, 4.9, .000001}], Table[i, {i, 4.9, 9.3, .1}],  
  Table[i, {i, 9.3, 10.6, .00001}], Table[i, {i, 10.6, 17.0, .1}],  
  Table[i, {i, 17.0, 18.5, .00001}], Table[i, {i, 18.5, 25.9, .1}],  
  Table[i, {i, 25.9, 28.0, .0001}], Table[i, {i, 28.0, 40.0, .1}]];
```

```
Dimensions[energyBetter31]
```

```
{1 001 354}
```

```
reflectCoefBetter31 =
```

```
  ReflectionProbability[potential[[2, All]], energyBetter31, 1];
```

```
Dimensions[reflectCoefBetter31]
```

```
{1 001 354}
```

```
eigenEnergyBetter31 = EigenEnergy[potential[[2, All]], energyBetter31, 1];
```

```
Dimensions[eigenEnergyBetter31]
```

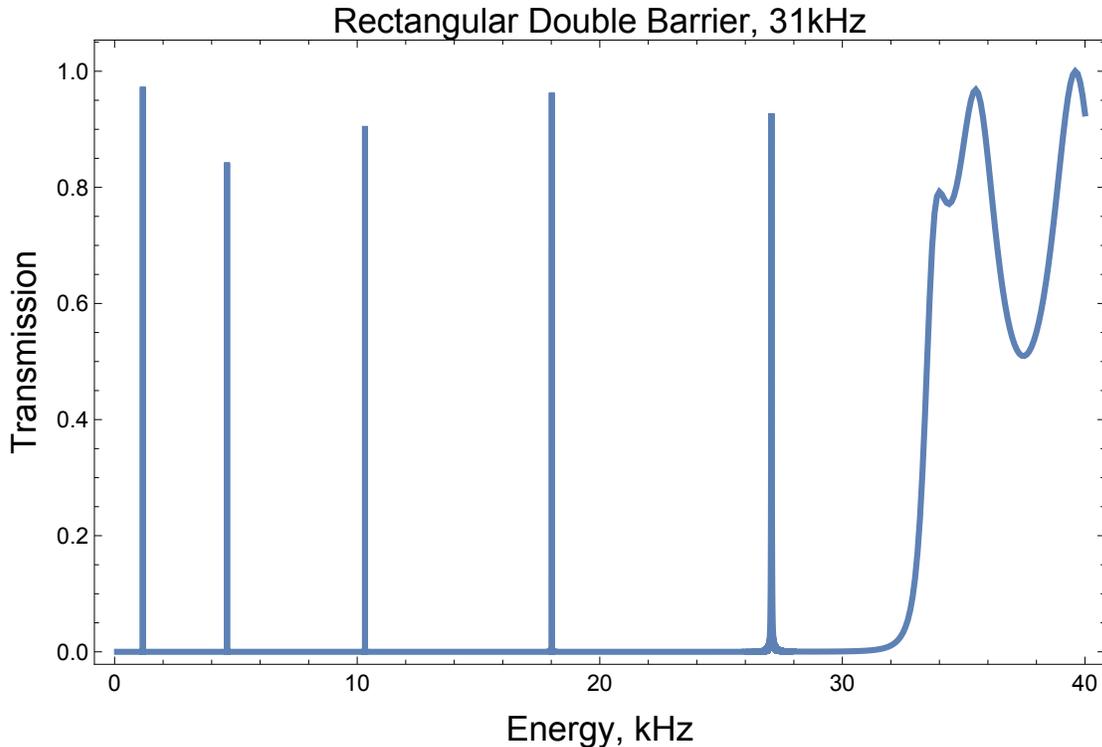
```
data31 =
```

```
  DeleteDuplicates[eigenEnergyBetter31 / (2 *  $\pi$  *  $\hbar$  * 1.0*^3), Abs[#1 - #2] < .1 &]
```

```
{665}
```

```
{1.11628, 4.44363, 9.90643, 17.3226, 26.1509}
```

```
ListLinePlot[Transpose[{energyBetter31 / (2 *  $\pi$  *  $\hbar$  * 1.0*^3), 1 - reflectCoefBetter31},
  PlotRange -> All, Frame -> True,
  FrameLabel -> {Style["Energy, kHz", 18], Style["Transmission", 18]},
  PlotLabel -> Style["Rectangular Double Barrier, 31kHz", 18],
  FrameTicksStyle -> Medium, ImageSize -> Large, PlotStyle -> {Thickness[.006]}]
```



```
better32 = (2 *  $\pi$  *  $\hbar$  * 1.0*^3) * Join[Table[i, {i, .1, 1.0, .1}],
  Table[i, {i, 1.0, 1.3, .000001}], Table[i, {i, 1.3, 4.4, .1}],
  Table[i, {i, 4.4, 4.9, .000001}], Table[i, {i, 4.9, 9.3, .1}],
  Table[i, {i, 9.3, 10.7, .00001}], Table[i, {i, 10.7, 17.0, .1}],
  Table[i, {i, 17.0, 18.5, .00001}], Table[i, {i, 18.5, 25.9, .1}],
  Table[i, {i, 25.9, 28.0, .0001}], Table[i, {i, 28.0, 40.0, .1}]];
energyBetter32 = If[Intersection[potential[[3, All]], better32] == {},
  better32, Delete[better32, Flatten[
    Position[better32, Intersection[potential[[3, All]], better32][[1]]]]];
Dimensions[energyBetter32]
{1 111 351}
```

```

reflectCoefBetter32 =
  ReflectionProbability[potential[[3, All]], energyBetter32, 1];
Dimensions[reflectCoefBetter32]
eigenEnergyBetter32 = EigenEnergy[potential[[3, All]], energyBetter32, 1];
Dimensions[eigenEnergyBetter32]
data32 =
  DeleteDuplicates[eigenEnergyBetter32 / (2 *  $\pi$  *  $\hbar$  * 1.0*^3), Abs[#1 - #2] < .1 &]

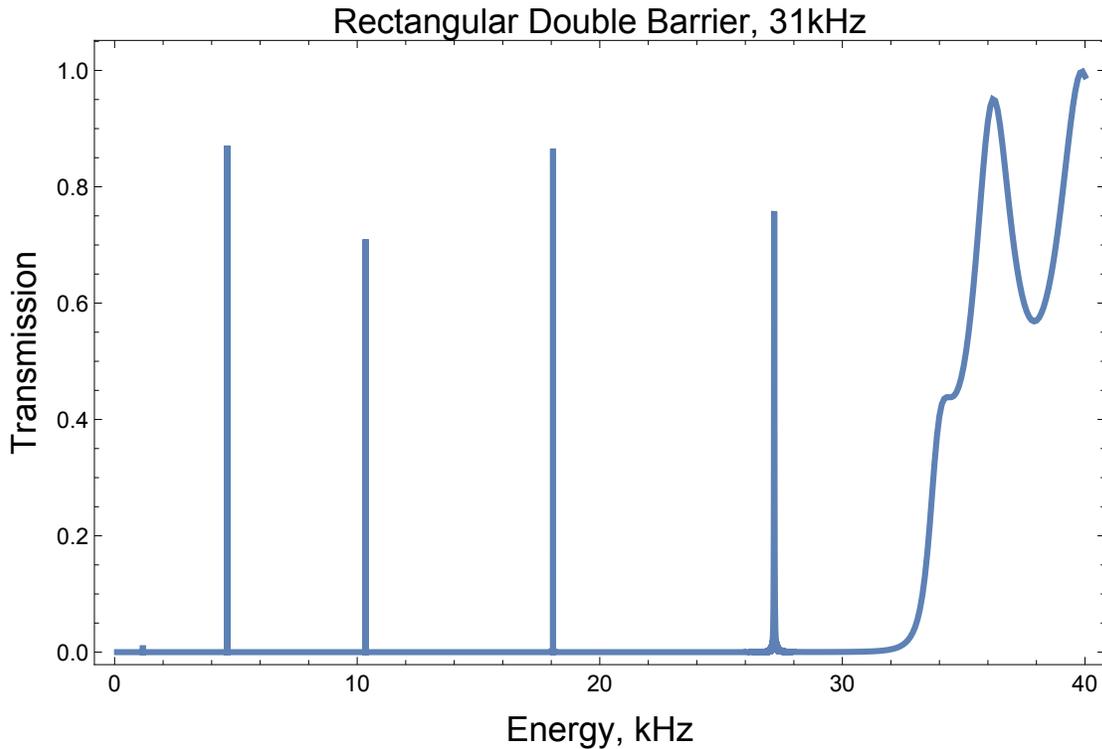
{1 111 351}

{650}

{1.11844, 4.45268, 9.92868, 17.369, 26.2519}

ListLinePlot[Transpose@{energyBetter32 / (2 *  $\pi$  *  $\hbar$  * 1.0*^3), 1 - reflectCoefBetter32},
  PlotRange -> All, Frame -> True,
  FrameLabel -> {Style["Energy, kHz", 18], Style["Transmission", 18]},
  PlotLabel -> Style["Rectangular Double Barrier, 32kHz", 18],
  FrameTicksStyle -> Medium, ImageSize -> Large, PlotStyle -> {Thickness[.006]}]

```



```

position32 = Flatten[Position[1 - reflectCoefBetter32, _? (# > .001 &)]];
energyResonance32 = Table[
  energyBetter32[[position32[[i]]]] / (2 *  $\pi$  *  $\hbar$  * 1.0*^3), {i, Length[position32]};
DeleteDuplicates[energyResonance32, Abs[#1 - #2] < .1 &]
{1.1667, 4.64326, 10.3463, 18.0676, 26.9207, 27.0208, 27.1208, 27.2208,
  27.3208, 27.4208, 27.5209, 31., 31.1, 31.3, 31.5, 31.6, 31.8, 32.1, 32.2,
  32.4, 32.5, 32.6, 32.7, 32.9, 33., 33.1, 33.2, 33.4, 33.5, 33.6, 33.7,
  33.9, 34., 34.1, 34.2, 34.4, 34.5, 34.6, 34.7, 34.9, 35., 35.1, 35.2,
  35.4, 35.5, 35.6, 35.7, 35.9, 36., 36.1, 36.2, 36.4, 36.5, 36.6, 36.7,
  36.9, 37., 37.1, 37.2, 37.4, 37.5, 37.6, 37.7, 37.9, 38., 38.1, 38.2, 38.4,
  38.6, 38.7, 38.8, 38.9, 39., 39.1, 39.2, 39.4, 39.6, 39.7, 39.8, 39.9, 40.}

dataTable = {"Eigen-Energies in kHz", SpanFromLeft},
  Join[{v02[[1]] / (2 *  $\pi$  *  $\hbar$  * 1.0*^3) "kHz"}, Flatten@
    DeleteDuplicates[eigenEnergyBetter1 / (2 *  $\pi$  *  $\hbar$  * 1.0*^3), Abs[#1 - #2] < .1 &]],
  Join[{v02[[2]] / (2 *  $\pi$  *  $\hbar$  * 1.0*^3) "kHz"}, Flatten@data31],
  Join[{v02[[3]] / (2 *  $\pi$  *  $\hbar$  * 1.0*^3) "kHz"}, data32]];
eigenEnergies = Grid[dataTable, Alignment -> Center, Spacings -> {2, 2},
  Frame -> All, Background -> {{LightGray, None}, {LightGray, None}}]

```

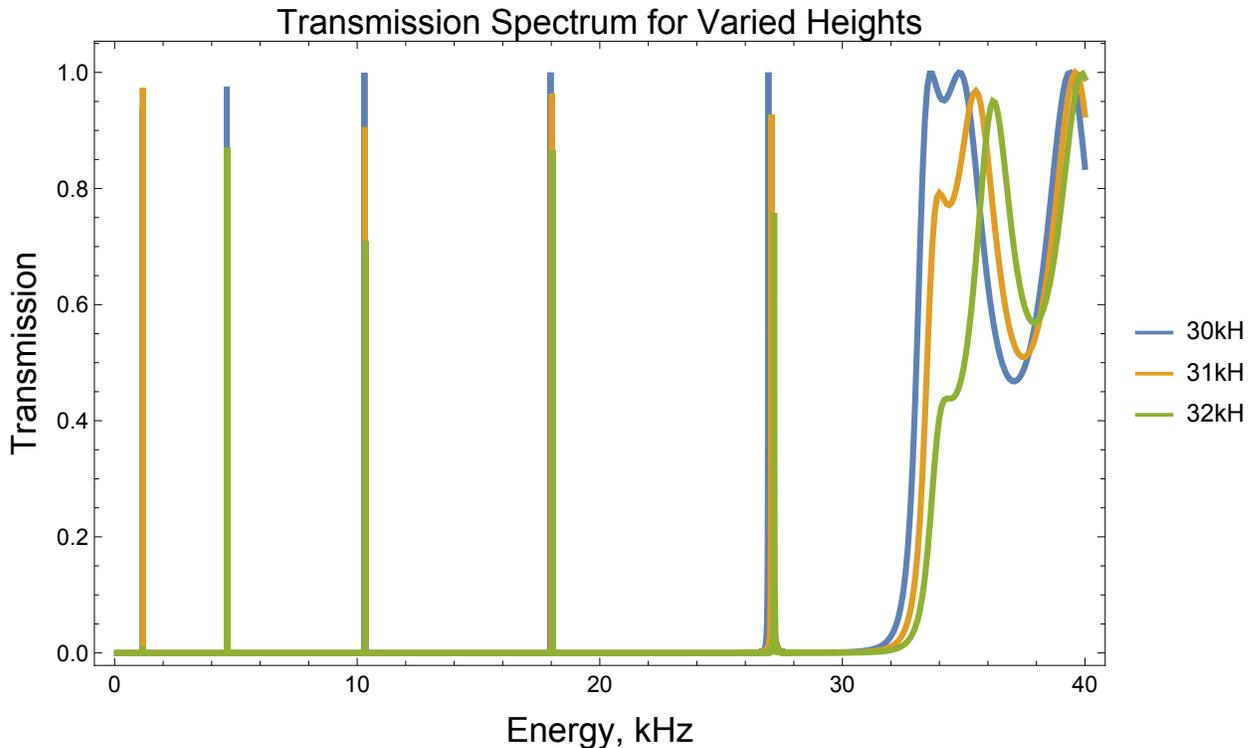
Eigen-Energies in kHz					
30. kHz	1.11401	4.43412	9.883	17.2732	26.039
31. kHz	1.11628	4.44363	9.90643	17.3226	26.1509
32. kHz	1.11844	4.45268	9.92868	17.369	26.2519

```

Export[
  "/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis
  nb's/possible figs/eigenEnergies.jpeg", eigenEnergies];

```

```
doubRTransH =
ListLinePlot[{Transpose@{energyBetter / (2 * π * ħ * 1.0*^3), 1 - reflectCoefBetter1},
  Transpose@{energyBetter31 / (2 * π * ħ * 1.0*^3), 1 - reflectCoefBetter31},
  Transpose@{energyBetter32 / (2 * π * ħ * 1.0*^3), 1 - reflectCoefBetter32}},
PlotRange → All, Frame → True,
FrameLabel → {Style["Energy, kHz", 18], Style["Transmission", 18]},
PlotLabel → Style["Transmission Spectrum for Varied Heights", 18],
FrameTicksStyle → Medium, ImageSize → Large,
PlotStyle → {Thickness[.006]}, PlotLegends → {"30kHz", "31kHz", "32kHz"}]
```

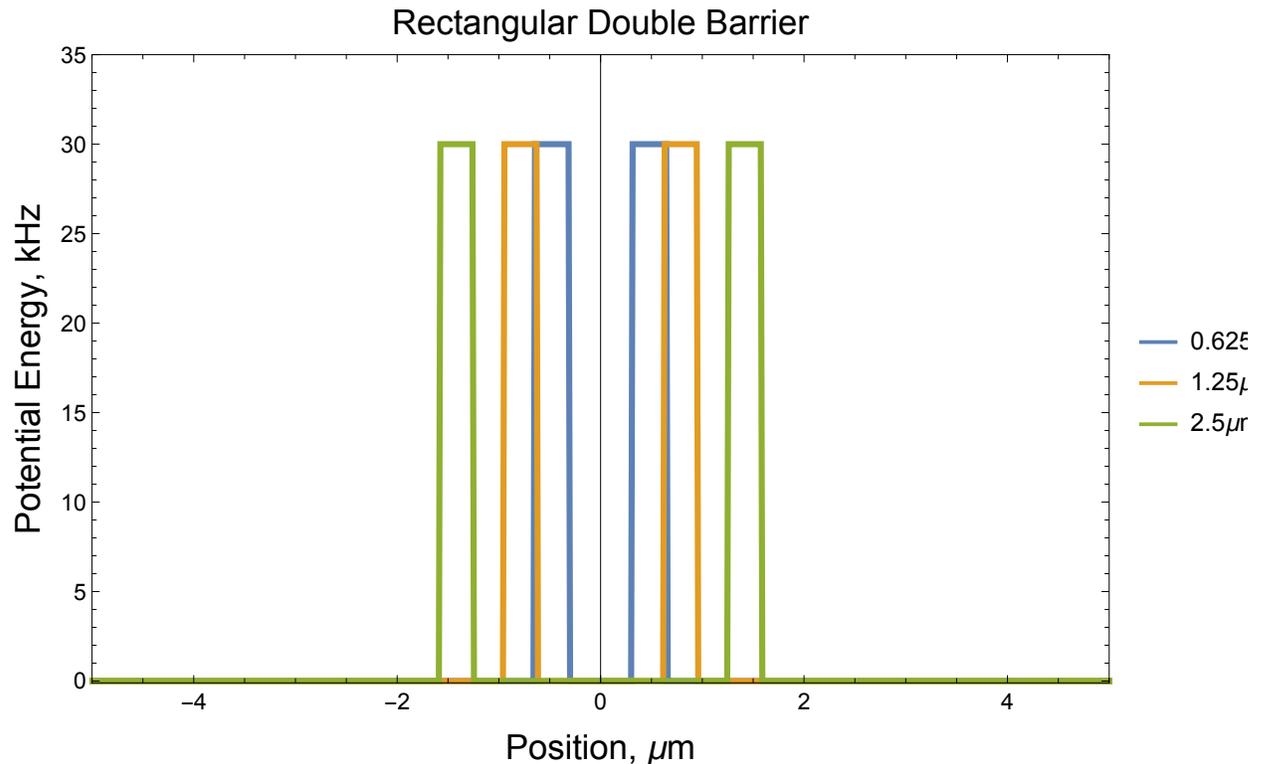


```
Export[
"/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis
nb's/possible figs/doubRTransH.jpeg", doubRTransH];
```

```
sepV = { $\frac{.625 \times 10^{-6}}{2}$ ,  $\frac{1.25 \times 10^{-6}}{2}$ ,  $\frac{2.5 \times 10^{-6}}{2}$ };
```

```
potentialSep =
Table[-v01 * UnitStep[i + sepV[[n]]] + v01 * UnitStep[i + (sepV[[n]] + wid)] +
  v02[[1]] * (-UnitStep[i - (sepV[[n]] + wid)] + UnitStep[i - sepV[[n]]]),
{n, 1, Length[sepV]}, {i, -mxm, mxm, 1}];
x = Table[i, {i, -mxm, mxm, 1}];
Dimensions[potentialSep]
{3, 2001}
```

```
doubRectSepPot =
ListLinePlot[Table[Transpose@{x/10^-6, potentialSep[[n, All]]/(2 * π * ħ * 10^3)},
  {n, 1, Length[sepV]}], PlotRange → {{-5, 5}, {- .1, 35}}, Frame → True,
FrameLabel → {Style["Position, μm", 18], Style["Potential Energy, kHz", 18]},
PlotLabel → Style["Rectangular Double Barrier", 18],
FrameTicksStyle → Medium, ImageSize → Large, PlotStyle → {Thickness[.006]},
PlotLegends → {"0.625μm", "1.25μm", "2.5μm"}]
```



```
Export [
"/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis
nb's/possible figs/doubRectSepPot.jpeg", doubRectSepPot];

betterSep2 = (2 * π * ħ * 1.0*^3) * Join[Table[i, {i, .1, 1.1, .1}],
Table[i, {i, 1.1, 1.4, .000001}], Table[i, {i, 1.4, 2.6, .1}],
Table[i, {i, 2.6, 3.0, .00001}], Table[i, {i, 3.0, 4.9, .1}],
Table[i, {i, 4.9, 5.3, .00001}], Table[i, {i, 5.3, 7.5, .1}],
Table[i, {i, 7.5, 8.5, .00001}], Table[i, {i, 8.5, 10.9, .1}],
Table[i, {i, 10.9, 11.9, .00001}], Table[i, {i, 11.9, 15.0, .1}],
Table[i, {i, 15.0, 15.7, .0001}], Table[i, {i, 15.7, 19.5, .1}],
Table[i, {i, 19.5, 21.0, .0001}], Table[i, {i, 21.0, 24.5, .1}],
Table[i, {i, 24.5, 26.0, .0001}], Table[i, {i, 26.0, 29.5, .1}],
Table[i, {i, 29.5, 31.0, .0001}], Table[i, {i, 31.0, 35.0, .1}]];
energyBetterSep2 = If[Intersection[potential[[2, All]], betterSep2] == {},
betterSep2, Delete[betterSep2, Flatten[
Position[betterSep2, Intersection[potential[[2, All]], betterSep2][[1]]]]];
Dimensions[energyBetterSep2]
{632 285}
```

```

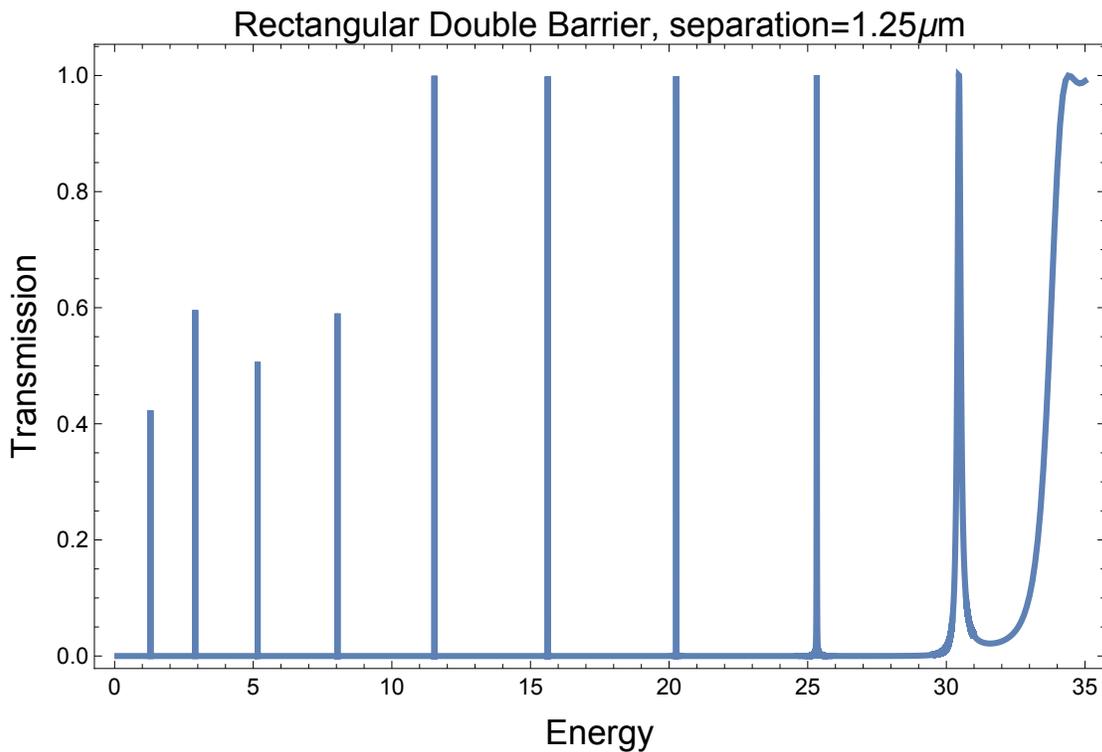
reflectCoefSepBetter2 =
  ReflectionProbability[potentialSep[[2, All]], energyBetterSep2, 1];
Dimensions[reflectCoefSepBetter2]
eigenEnergySepBetter2 = EigenEnergy[potentialSep[[2, All]], energyBetterSep2, 1];
Dimensions[eigenEnergySepBetter2]
dataSep2 =
  DeleteDuplicates[eigenEnergySepBetter2 / (2 *  $\pi$  *  $\hbar$  * 1.0*^3), Abs[#1 - #2] < .1 &]
{632 285}

{206}

{1.26529, 2.84344, 5.04584, 7.86416, 11.2851, 15.2864, 19.8268, 24.8131, 29.9303}

transH2 = ListLinePlot[Transpose@
  {energyBetterSep2 / (2 *  $\pi$  *  $\hbar$  * 1.0*^3), 1 - reflectCoefSepBetter2}, PlotRange -> All,
  Frame -> True, FrameLabel -> {Style["Energy", 18], Style["Transmission", 18]},
  PlotLabel -> Style["Rectangular Double Barrier, separation=1.25 $\mu$ m", 18],
  FrameTicksStyle -> Medium, ImageSize -> Large, PlotStyle -> {Thickness[.006]}]

```



```

Export["/Users/avaashby/Desktop/masters-thesis/Mathematica
Code/ImpedanceMethod/thesis nb's/possible figs/transH2.jpeg", transH2];

```

```

betterSep3 = (2 *  $\pi$  *  $\hbar$  * 1.0*^3) * Join[Table[i, {i, .1, 2.8, .1}],
  Table[i, {i, 2.8, 3.2, .00001}], Table[i, {i, 3.2, 3.9, .1}],
  Table[i, {i, 3.9, 4.3, .000001}], Table[i, {i, 4.3, 5.2, .1}],
  Table[i, {i, 5.2, 5.6, .000001}], Table[i, {i, 5.6, 6.7, .1}],
  Table[i, {i, 6.7, 7.2, .00001}], Table[i, {i, 7.2, 8.3, .1}],
  Table[i, {i, 8.3, 8.8, .00001}], Table[i, {i, 8.8, 9.9, .1}],
  Table[i, {i, 9.9, 10.6, .00001}], Table[i, {i, 10.6, 11.9, .1}],
  Table[i, {i, 11.9, 12.4, .00001}], Table[i, {i, 12.4, 13.9, .1}],
  Table[i, {i, 13.9, 14.6, .00001}], Table[i, {i, 14.6, 16.3, .1}],
  Table[i, {i, 16.3, 16.9, .0001}], Table[i, {i, 16.9, 18.5, .1}],
  Table[i, {i, 18.5, 19.5, .0001}], Table[i, {i, 19.5, 21.2, .1}],
  Table[i, {i, 21.2, 21.9, .0001}], Table[i, {i, 21.9, 23.8, .1}],
  Table[i, {i, 23.8, 24.8, .0001}], Table[i, {i, 24.8, 26.5, .1}],
  Table[i, {i, 26.5, 27.5, .0001}], Table[i, {i, 27.5, 29.4, .1}],
  Table[i, {i, 29.4, 31.0, .001}], Table[i, {i, 31.0, 35.0, .1}]];
energyBetterSep3 = If[Intersection[potential[[3, All]], betterSep3] == {},
  betterSep3, Delete[betterSep3, Flatten[
    Position[betterSep3, Intersection[potential[[3, All]], betterSep3][[1]]]]];
Dimensions[energyBetterSep3]
{1 174 877}

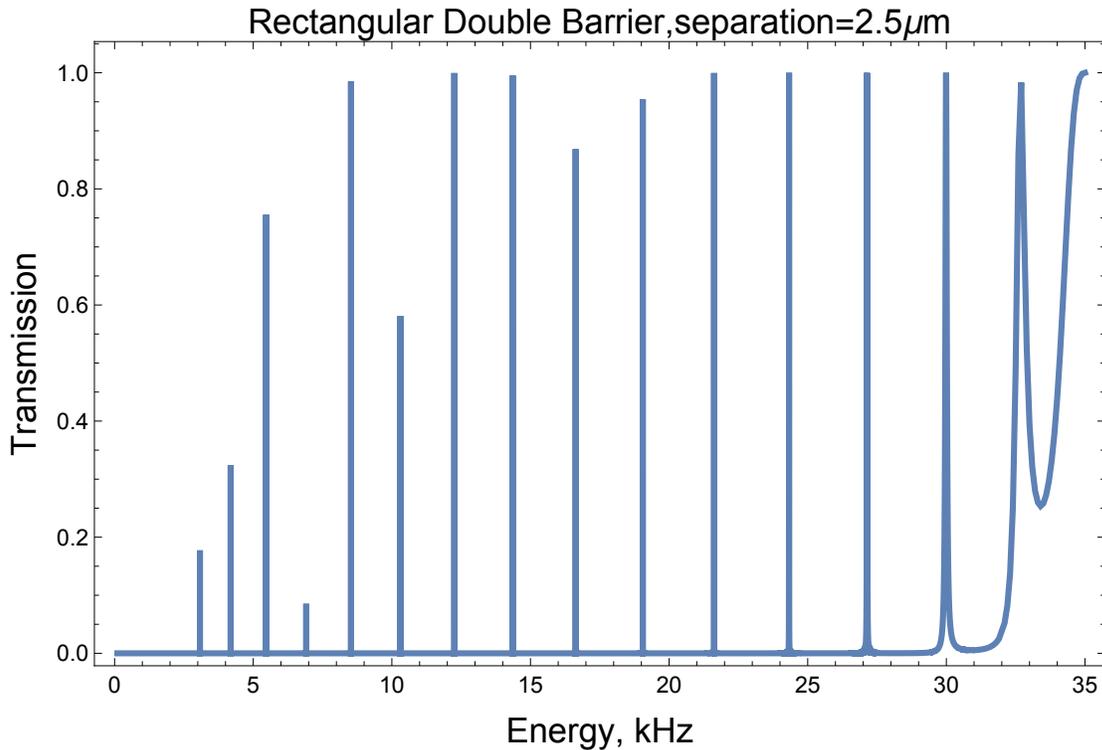
reflectCoefSepBetter3 =
  ReflectionProbability[potentialSep[[3, All]], energyBetterSep3, 1];
Dimensions[reflectCoefSepBetter3]
eigenEnergySepBetter3 = EigenEnergy[potentialSep[[3, All]], energyBetterSep3, 1];
Dimensions[eigenEnergySepBetter3]
dataSep3 =
  DeleteDuplicates[eigenEnergySepBetter3 / (2 *  $\pi$  *  $\hbar$  * 1.0*^3), Abs[#1 - #2] < .1 &]
{1 174 877}

{361}

{3.04189, 4.13844, 5.40232, 6.83282, 8.42903, 10.1898,
  12.1136, 14.1983, 16.441, 18.8375, 21.381, 24.0596, 26.8487}

```

```
transH3 = ListLinePlot[
  Transpose@{energyBetterSep3 / (2 * π * ħ * 1.0*^3), 1 - reflectCoefSepBetter3},
  PlotRange → All, Frame → True,
  FrameLabel → {Style["Energy, kHz", 18], Style["Transmission", 18]},
  PlotLabel → Style["Rectangular Double Barrier, separation=2.5μm", 18],
  FrameTicksStyle → Medium, ImageSize → Large, PlotStyle → {Thickness[.006]}
```



```
Export["/Users/avaashby/Desktop/masters-thesis/Mathematica
  Code/ImpedanceMethod/thesis nb's/possible figs/transH3.jpeg", transH3];
```

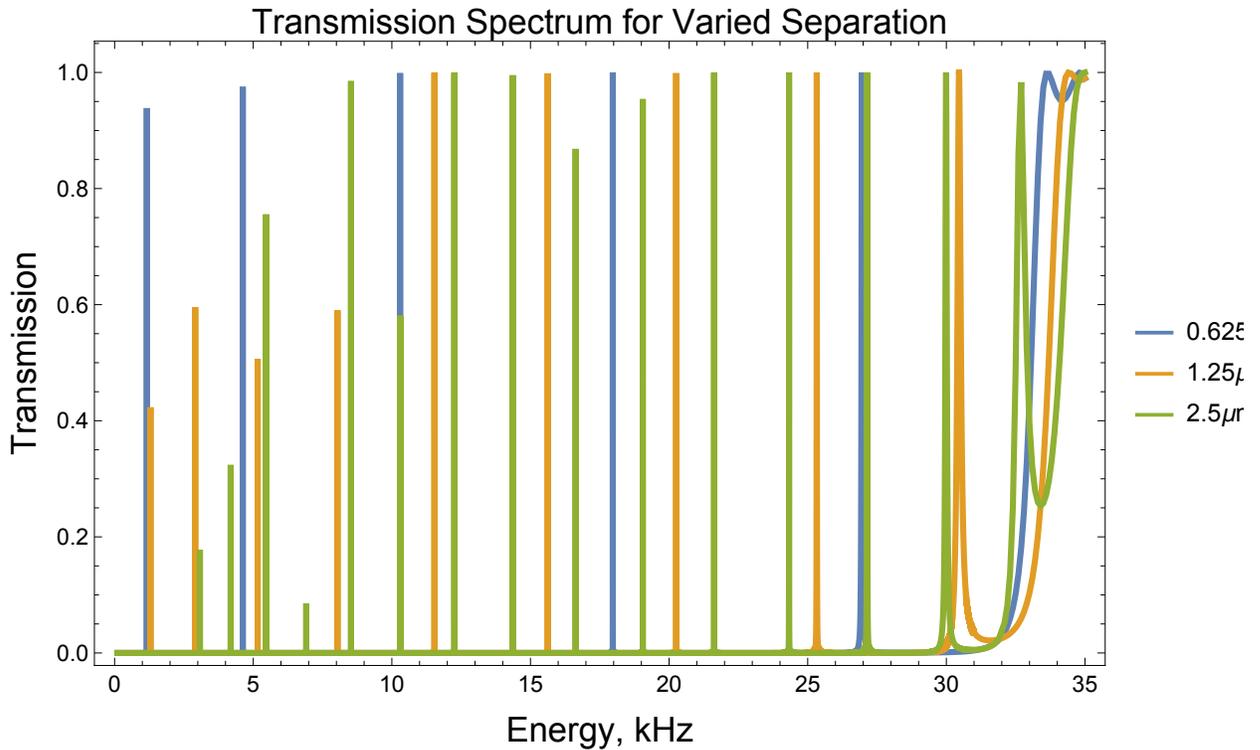
```
positionughhhh = Flatten[Position[energyBetter / (2 * π * ħ * 1.0*^3), _? (# >= 35.0 &)]];
positionughhhh[[1]]
```

```
5 601 324
```

```
energyBetter[[positionughhhh[[1]]]] / (2 * π * ħ * 1.0*^3)
```

```
35.
```

```
doubRtransSep = ListLinePlot[
  {Transpose@{energyBetter[[1 ;; positionughhhh[[1]]]] / (2 *  $\pi$  *  $\hbar$  * 1.0*^3),
    1 - reflectCoefBetter1[[1 ;; positionughhhh[[1]]]]},
  Transpose@{energyBetterSep2 / (2 *  $\pi$  *  $\hbar$  * 1.0*^3), 1 - reflectCoefSepBetter2},
  Transpose@{energyBetterSep3 / (2 *  $\pi$  *  $\hbar$  * 1.0*^3), 1 - reflectCoefSepBetter3}},
  PlotRange -> All, Frame -> True,
  FrameLabel -> {Style["Energy, kHz", 18], Style["Transmission", 18]},
  PlotLabel -> Style["Transmission Spectrum for Varied Separation", 18],
  FrameTicksStyle -> Medium, ImageSize -> Large, PlotStyle -> {Thickness[.006]},
  PlotLegends -> {"0.625 $\mu$ m", "1.25 $\mu$ m", "2.5 $\mu$ m"}]
```



```
Export[
  "/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis
  nb's/possible figs/doubRtransSep.jpeg", doubRtransSep];
```

```
Round[dataSep3, .001]
```

```
{3.042, 4.138, 5.402, 6.833, 8.429, 10.19,
  12.114, 14.198, 16.441, 18.838, 21.381, 24.06, 26.849}
```

```

dataTableSep = {"Eigen-Energies in kHz, v01=v02=30kHz", SpanFromLeft},
  Join[{sepV[[1]] * 2 / 10^-6 "μm"}, Round[Flatten@DeleteDuplicates[
    eigenEnergyBetter1 / (2 * π * ħ * 1.0*^3), Abs[#1 - #2] < .1 &], .01]],
  Join[{sepV[[2]] * 2 / 10^-6 "μm"}, Round[dataSep2, .01]],
  Join[{sepV[[3]] * 2 / 10^-6 "μm"}, Round[dataSep3, .01]]];
eigenEnergiesSep = Grid[dataTableSep, ItemSize → Full, Frame → All,
  Background → {{LightGray, None}, {LightGray, None}}]

```

Eigen-Energies in kHz, v01=v02=30kHz												
0.625 μm	1.11	4.43	9.88	17.27	26.04							
1.25 μm	1.27	2.84	5.05	7.86	11.29	15.29	19.83	24.81	29.93			
2.5 μm	3.04	4.14	5.4	6.83	8.43	10.19	12.11	14.2	16.44	18.84	21.38	24.06

```

Export[
  "/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis
  nb's/possible figs/eigenEnergiesSep.jpeg", eigenEnergiesSep];

```

```

diffData1 = Round[Differences[Flatten@
  DeleteDuplicates[eigenEnergyBetter1 / (2 * π * ħ * 1.0*^3), Abs[#1 - #2] < .1 &]], .1]
diffData2 = Round[Differences[dataSep2], .1]
diffData3 = Round[Differences[dataSep3], .1]
{3.3, 5.4, 7.4, 8.8}
{1.6, 2.2, 2.8, 3.4, 4., 4.5, 5., 5.1}
{1.1, 1.3, 1.4, 1.6, 1.8, 1.9, 2.1, 2.2, 2.4, 2.5, 2.7, 2.8}

```

```

dataTableSepDiff =
  {"Eigen-Energy differnces in kHz, v01=v02=30kHz", SpanFromLeft},
  Join[{sepV[[1]] * 2 / 10^-6 "μm"}, diffData1], Join[{sepV[[2]] * 2 / 10^-6 "μm"},
  diffData2], Join[{sepV[[3]] * 2 / 10^-6 "μm"}, diffData3]};
eigenEnergiesSepDiff = Grid[dataTableSepDiff, Alignment → Center, ItemSize → Full,
  Frame → All, Background → {{LightGray, None}, {LightGray, None}}]

```

Eigen-Energy differnces in kHz, v01=v02=30kHz												
0.625 μm	3.3	5.4	7.4	8.8								
1.25 μm	1.6	2.2	2.8	3.4	4.	4.5	5.	5.1				
2.5 μm	1.1	1.3	1.4	1.6	1.8	1.9	2.1	2.2	2.4	2.5	2.7	2.8

```

Export[
  "/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis
  nb's/possible figs/eigenEnergiesSepDiff.jpeg", eigenEnergiesSepDiff];

```

## Appendix F

The Mathematica package for the reflection amplitude coefficients

`ImpedanceMethodAmplitude.m`

# General Impedance Method reflection amplitude

12.21.2015

Anderson Group

## MetaPackage:

Transmission line analogy of resonance tunneling phenomena: The generalized impedance concept by A.N. Khondker

J. Appl. Phys., vol. 63, no. 10, p.5191, 1988

Wave transmission through periodic, quasiperiodic, and random one-dimensional finite lattices by Braulio Gutierrez-Medina

Am. J. Phys., vol. 81, no. 2, p. 104, 2013

This package used the general impedance method to numerically evaluate the reflection probabilities of an arbitrary potential energy function. The general impedance method evaluates the potential energy function from the output end to the input end of the potential.

The input values for this package are:

$v = \text{Table}[\text{function}[i], \{i, i_{\min}, i_{\max}, \text{stepsize}\}]$  : the arbitrary potential energy function as a list of points.

$\text{energy} = \text{Table}[i, \{i, i_{\min}, i_{\max}, \text{stepsize}\}]$  : the span of energy values that an arbitrary wavefunction might have. Note, this could also be a list of one.

$l = \text{constant}$  : step size, or more specifically the value  $x[[2]] - x[[1]]$ . This is the step size at which the potential energy function takes.

The output values for this package is the values of the reflection probability for each value of energy.  $\text{Length}[\text{ReflectionProbability}] == \text{Length}[\text{energy}]$ .

The variables defined within the package are:

$m =$  effective mass of the wave packet in units of kg (for rubidium)

$\hbar =$  Planck's constant in units of Js

$k =$  propagation constant

$z_0 =$  characteristic impedance

$z_{\text{load}} =$  load impedance

$z_{\text{input}} =$  input impedance

## Begin Package

```
BeginPackage["ImpedanceMethodAmplitude`ImpedanceMethodAmplitude`"]
```

```
ImpedanceMethodAmplitude`ImpedanceMethodAmplitude`
```

## Usage Message

```
ReflectionAmplitude::usage =
  "ReflectionAmplitude[v,energy,l] computes the complex reflection
  amplitude by using the general impedance method. The arguments
  are v, energy, and l where they are the potential energy vector
  function, energy vector function, and the step size, respectively.
  m is the effective mass of the wave packet and  $\hbar$  is Planck's constant
  in units of Js, both are defined within the package.
  v must be of the form v=Table[function[i],{i,imin,imax,l}]
  energy must be of the form energy=Table[i,{i,imin,imax,l}]"
```

## Warning Messages

```
ReflectionAmplitude::badv =
  "The potential energy argument `1` must be in the form of a list.";

ReflectionAmplitude::badenergy =
  "The energy argument `1` must be in the form of a list.";

ReflectionAmplitude::badl =
  "The step size argument `1` is not within the appropriate bounds. l must be
  greater than 0 and less than the Thermal de Broglie Wavelength, `2`.";
```

## Options

### Private Context

```
Begin["`Private`"]
ImpedanceMethodAmplitude`ImpedanceMethodAmplitude`Private`
```

## Function Definitions (Public and Private)

```
m = 1.44*^-25;
 $\hbar$  = 1.054571726*^-34;
```

```

ReflectionAmplitude[v_, energy_, l_] :=
Module[{k, z0, zload, zinput, reflectionL, i},
Which[
! (VectorQ[v]), Message[ReflectionAmplitude::"badv", v],
! (VectorQ[energy]), Message[ReflectionAmplitude::"badenergy", energy],
!  $\left(0 < l \leq \frac{(2 * N[\pi, 10] * \hbar)}{\text{Sqrt}[2 * m * \text{Max}[energy]]}\right)$ ,
Message[ReflectionAmplitude::"badl", l,  $\frac{(2 * N[\pi, 10] * \hbar)}{\text{Sqrt}[2 * m * \text{Max}[energy]]}$ ]
];
Do[
k = i * Sqrt[2 * m * (energy - v[[i]])] /  $\hbar$ ;
z0 = -i *  $\hbar$  * k / m;
If[i == Length[v], zload = z0, zload = zinput];
zinput = z0 *  $\frac{zload * \text{Cosh}[k * l] - z0 * \text{Sinh}[k * l]}{z0 * \text{Cosh}[k * l] - zload * \text{Sinh}[k * l]}$ , {i, Length[v], 1, -1}];
reflectionL =  $\frac{zinput - z0}{zinput + z0}$ 
]

```

## End Private Context

```
End[]
```

```
ImpedanceMethodAmplitude`ImpedanceMethodAmplitude`Private`
```

## End Package

```
EndPackage[]
```

## Appendix G

The double Gaussian barrier potential Mathematica notebook

`doubleGaussPot.nb`

# Impedance Method Double Gaussian Potential

```
In[1]:= Quit []
```

```
In[1]:= ClearAll["Global`"]
```

Load the ImpedanceMethod package:

There are several ways to do this, here is a couple options:

- 1) For newer versions of Mathematica you can start with Get["dkf] and a small icon will appear to let you search your computers files.
- 2) If you know the file path use Get["filePath"] or Needs["filePath"], or sometimes Get["context`"], Needs["context`"] works too.
- 3) If you dont know the file path and the icon does not appear, try going to File -> Install -> "Type of Item to Install:" select Package, "Source:" select from file, and choose the appropriate file, ie ImpedanceMethod.m. Now use <<ImpedanceMethod` for the input

```
In[2]:= Get["/Users/avaashby/Desktop/masters-thesis/Mathematica
Code/ImpedanceMethod/ImpedanceMethod.m"]
```

```
In[3]:= Get["/Users/avaashby/Desktop/masters-thesis/Mathematica
Code/ImpedanceMethod/ImpedanceMethodAmplitude.m"]
```

```
In[4]:= Get["/Users/avaashby/Desktop/masters-thesis/Mathematica
Code/ImpedanceMethod/ImpedanceMethodEigenEnergy.m"]
```

```
In[5]:= $ContextPath
```

```
Out[5]= {ImpedanceMethodEigenEnergy`ImpedanceMethodEigenEnergy`,
ImpedanceMethodAmplitude`ImpedanceMethodAmplitude`,
ImpedanceMethod`ImpedanceMethod`, StreamingLoader`, IconizeLoader`,
CloudObjectLoader`, PacletManager`, System`, Global`}
```

```
phase[vec_List] := Module[{phi, df, len, i},
  phi = Im[Log[vec]];
  df = Differences@phi;
  len = Length@phi;
  i = Flatten@Position[df, x_ /; Abs[x] > 3.5];
  Do[phi = phi - (2 Pi * Sign[df[[j]]] * UnitStep[# - (j + 1)] & /@Range[len]), {j, i}];
  phi]

(*unWrapPhase[data_?VectorQ, tol_ : π, inc_ : 2 π] :=
  data + inc FoldList[Plus, 0, Sign[Chop[Apply[Subtract, Partition[data, 2, 1], {1}]]tol]]]
```

Define all global constants:

```

In[6]:= ħ = 1.054*^-34;
m = 1.44*^-25;
wx = 15.0*^-6; (*imin/imax*)
l = wx / 1000;
sigx = Sort[{2.1*^-6, 1.0*^-6, 1.6*^-6, 2.6*^-6}, Less];
nL = Length[sigx];
u01 = 2 * π * ħ * 30*^3;
u02 = 2 * π * ħ * Table[(i) * 10^3, {i, 30, 33}];
jL = Length[u02];
xsep = Sort[{2.5*^-6, 1.25*^-6, 5.0*^-6, 4.8*^-6, 5.8*^-6}, Less];
kL = Length[xsep];

```

Define the potential function,  $v[i]$ , and put into list of data points:

```

In[17]:= v[i_, j_, k_, n_] :=
  u01 * Exp[-2 *  $\left(\frac{i + \frac{xsep[[k]]}{2}}{\frac{sigx[[n]]}{1}}\right)^2$ ] + u02[[j]] * Exp[-2 *  $\left(\frac{i - \frac{xsep[[k]]}{2}}{\frac{sigx[[n]]}{1}}\right)^2$ ];
potential = Table[v[i, j, k, n], {n, 1, nL}, {k, 1, kL},
  {j, 1, jL}, {i, -wx / 2, wx / 2, 1}];
x = Table[i, {i, -wx / 2, wx / 2, 1}];
Dimensions[potential]

```

```
Out[20]= {4, 5, 4, 1001}
```

Define the energies you would like to be evaluated:

```

In[253]:= energy = 2 * π * ħ * 10^3 * Table[i, {i, 1.0, 35, .00004}];
Dimensions[energy]

```

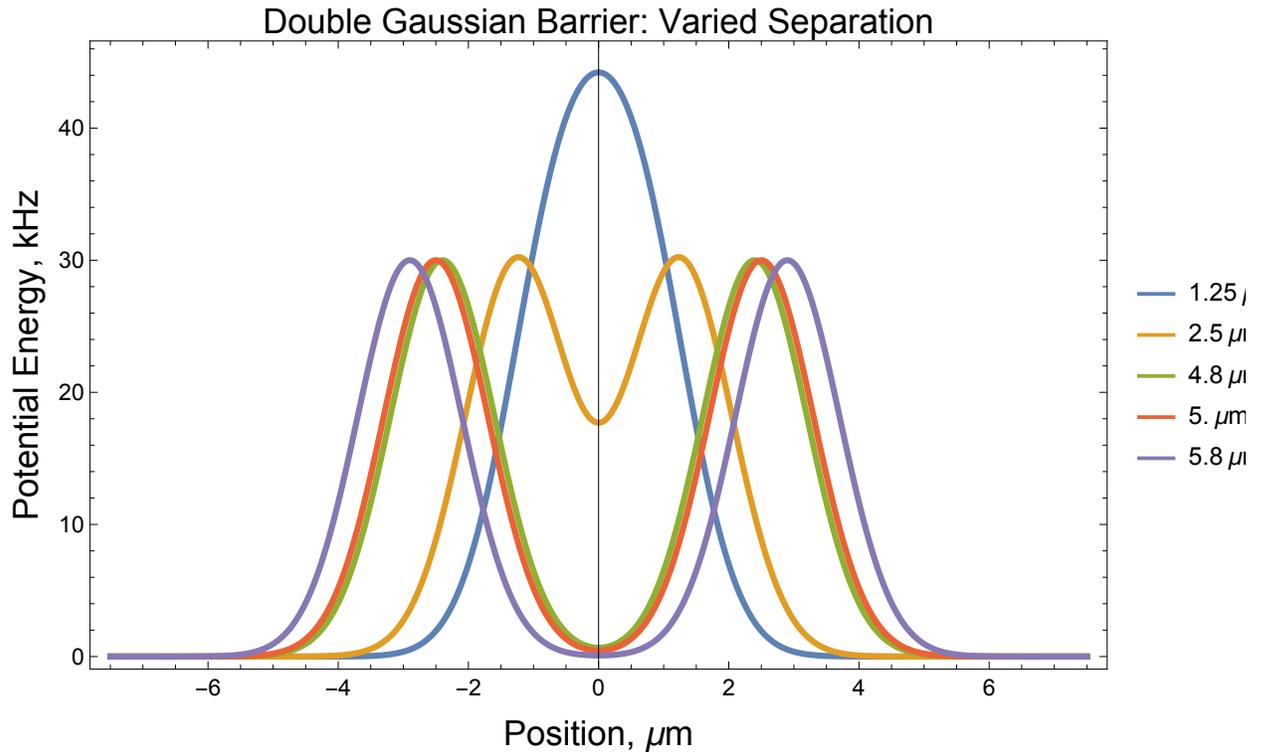
```
Out[254]= {850 001}
```

Optional- Plot potential function:

```

varsepPot = ListLinePlot[Table[
  Transpose[{x / 10^-6, potential[[2, k, 1, All]] / (2 *  $\pi$  *  $\hbar$  * 10^3)}, {k, 1, kL}],
  FrameLabel -> {Style["Position,  $\mu\text{m}$ ", 18], Style["Potential Energy, kHz", 18]},
  PlotLabel -> Style["Double Gaussian Barrier: Varied Separation", 18],
  PlotLegends -> {Table[" $\mu\text{m}$ " xsep[[k]] / 10^-6, {k, 1, kL}]},
  PlotRange -> All, FrameTicksStyle -> Medium, ImageSize -> Large,
  PlotStyle -> {Thickness[.006]}, Frame -> True]

```



```

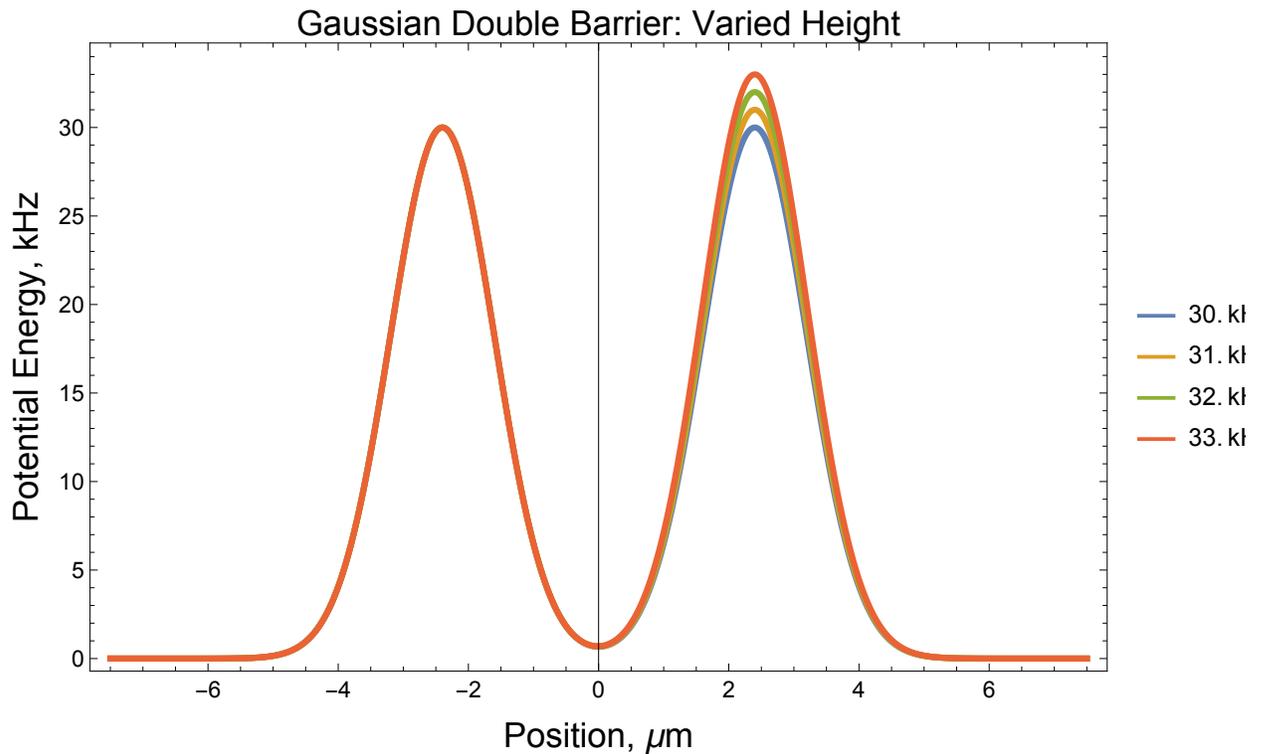
Export[
  "/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis
  nb's/possible figs/varsepPot.jpeg", varsepPot];

```

```

varheightPot = ListLinePlot[Table[
  Transpose[{x/10^-6, potential[[2, 3, j, All]]/(2 * π * ħ * 10^3)}, {j, 1, jL}],
  FrameLabel → {Style["Position, μm", 18], Style["Potential Energy, kHz", 18]},
  PlotLabel → Style["Gaussian Double Barrier: Varied Height", 18],
  PlotLegends → {Table["kHz" u02[[j]]/(2 * π * ħ * 10^3), {j, 1, jL}]},
  FrameTicksStyle → Medium, ImageSize → Large,
  PlotStyle → {Thickness[.006]}, Frame → True, PlotRange → All]

```

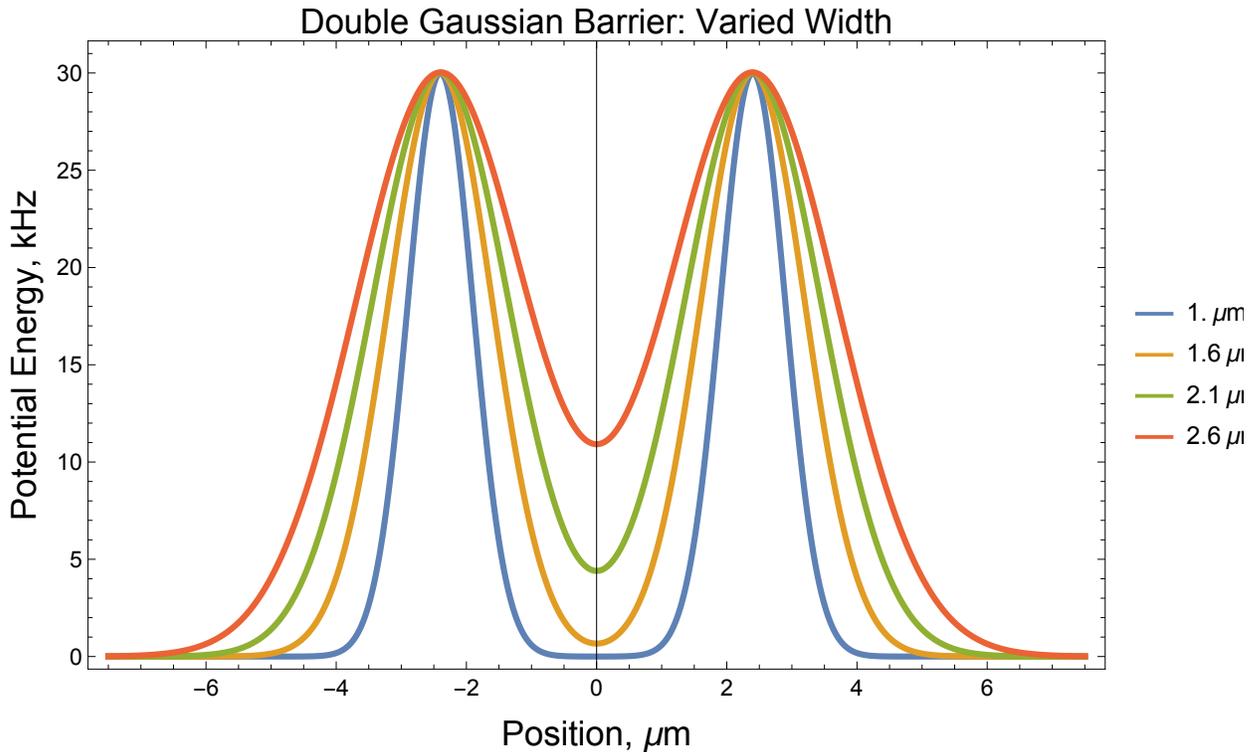


```

Export[
  "/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis
  nb's/possible figs/varheightPot.jpeg", varheightPot];

```

```
varwidthPot = ListLinePlot[Table[
  Transpose[{x/10^-6, potential[[n, 3, 1, All]]/(2 * π * ħ * 10^3)}, {n, 1, nL}],
  FrameLabel → {Style["Position, μm", 18], Style["Potential Energy, kHz", 18]},
  PlotLabel → Style["Double Gaussian Barrier: Varied Width", 18],
  PlotLegends → {Table["μm sigx[[n]]/10^-6, {n, 1, nL}]}, FrameTicksStyle → Medium,
  ImageSize → Large, PlotStyle → {Thickness[.006]}, Frame → True]
```



Export [

```
"/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis
nb's/possible figs/varwidthPot.jpeg", varwidthPot];
```

Evaluate ReflectionProbability[ ]:

```
cams3030 = {1.17740582696143, 2.23133776667272, 3.34253664358537, 4.49894536987933,
5.69190681957433, 6.91468142122128, 8.16173082465847, 9.42831606824111,
10.7102482710843, 12.0037210432943, 13.3051893293914, 14.6112751117543,
15.9186878599275, 17.2241511179226, 18.5243278612551, 19.8157367704978,
21.0946492913658, 22.3569532744475, 23.5979665672744, 24.8122060464905,
25.9932613055158, 27.1344544972186, 28.2320728685983, 29.2928309974024};
cams3033 = {1.22170862465289, 2.29953553327378, 3.43503981668997,
4.61627663809360, 5.83463517411243, 7.08339968535128, 8.35704776481764,
9.65085465197591, 10.9606470048600, 12.2826371626870, 13.6133033601839,
14.9492966688016, 16.2873627355717, 17.6242698148082, 18.9567357552994,
20.2813459902520, 21.5944520010844, 22.8920350827527, 24.1695178150228,
25.4215358162169, 26.6418780099316, 27.8245036657676, 28.9675945788482};
```

```

In[23]:= energyBetterer = 2 *  $\pi$  *  $\hbar$  * 10^3 * Table[i, {i, .0001, 10, .00001}];
Dimensions[energyBetterer]
eigenEnergyBetterer = EigenEnergy[potential[[2, 3, 1, All]], energyBetterer, 1];
Dimensions[eigenEnergyBetterer]
energyBetterer2 = 2 *  $\pi$  *  $\hbar$  * 10^3 * Table[i, {i, 10, 15, .00001}];
Dimensions[energyBetterer2]
eigenEnergyBetterer2 = EigenEnergy[potential[[2, 3, 1, All]], energyBetterer2, 1];
Dimensions[eigenEnergyBetterer2]
energyBetterer3 = 2 *  $\pi$  *  $\hbar$  * 10^3 * Table[i, {i, 15, 20, .00001}];
Dimensions[energyBetterer3]
eigenEnergyBetterer3 = EigenEnergy[potential[[2, 3, 1, All]], energyBetterer3, 1];
Dimensions[eigenEnergyBetterer3]
energyBetterer4 = 2 *  $\pi$  *  $\hbar$  * 10^3 * Table[i, {i, 20, 25, .00001}];
Dimensions[energyBetterer4]
eigenEnergyBetterer4 = EigenEnergy[potential[[2, 3, 1, All]], energyBetterer4, 1];
Dimensions[eigenEnergyBetterer4]
energyBetterer5 = 2 *  $\pi$  *  $\hbar$  * 10^3 * Table[i, {i, 25, 30, .00001}];
Dimensions[energyBetterer5]
eigenEnergyBetterer5 = EigenEnergy[potential[[2, 3, 1, All]], energyBetterer5, 1];
Dimensions[eigenEnergyBetterer5]

```

```
Out[24]= {999 991}
```

```
Out[26]= {74}
```

```
Out[28]= {500 001}
```

```
Out[30]= {69}
```

```
Out[32]= {500 001}
```

```
Out[34]= {82}
```

```
Out[36]= {500 001}
```

```
Out[38]= {88}
```

```
Out[40]= {500 001}
```

```
Out[42]= {103}
```

```

In[46]:= waveEnergy = Join[energyBetterer, energyBetterer2,
    energyBetterer3, energyBetterer4, energyBetterer5];
Dimensions[waveEnergy]

```

```
Out[47]= {2 999 995}
```

```

In[43]:= a3030 = DeleteDuplicates[Join[eigenEnergyBetterer / (2 *  $\pi$  *  $\hbar$  * 1.0*^3),
    eigenEnergyBetterer2 / (2 *  $\pi$  *  $\hbar$  * 1.0*^3), eigenEnergyBetterer3 /
    (2 *  $\pi$  *  $\hbar$  * 1.0*^3), eigenEnergyBetterer4 / (2 *  $\pi$  *  $\hbar$  * 1.0*^3),
    eigenEnergyBetterer5 / (2 *  $\pi$  *  $\hbar$  * 1.0*^3)], Abs[#1 - #2] < .1 &];

```

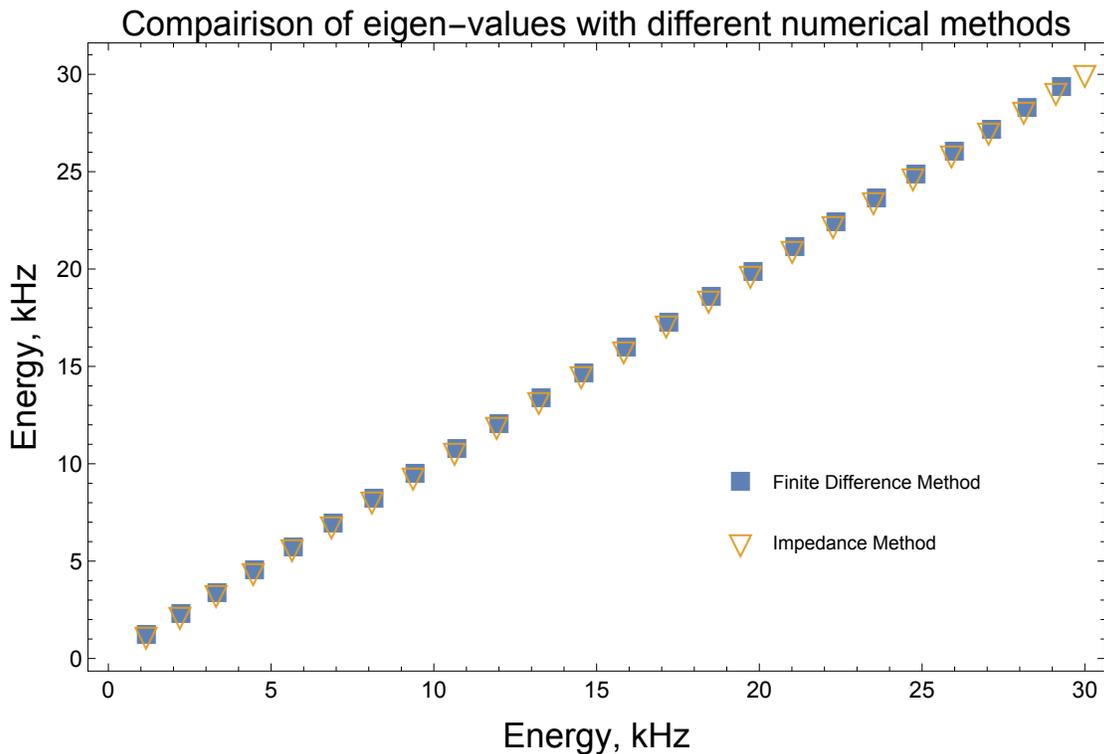
```
In[44]:= a3030
```

```
Out[44]= {1.16551, 2.20647, 3.31029, 4.45918, 5.6462, 6.86311, 8.10521, 9.367, 10.6448,
    11.9344, 13.2325, 14.5355, 15.8403, 17.1435, 18.4419, 19.7318, 21.0098,
    22.2716, 23.5127, 24.7274, 25.9088, 27.0471, 28.1263, 29.1165, 29.9996}
```

In[45]= **Differences[a3030]**

Out[45]= {1.04096, 1.10382, 1.14889, 1.18702, 1.21691, 1.2421, 1.26179, 1.27782,  
1.28956, 1.29813, 1.30299, 1.30483, 1.30317, 1.29838, 1.28995, 1.27799,  
1.26179, 1.24108, 1.2147, 1.18141, 1.13826, 1.07927, 0.99015, 0.88308}

```
compair3030 = ListPlot[{Transpose@{cams3030, cams3030}, Transpose@{a3030, a3030}},
  ImageSize → Large, PlotMarkers → {{■, 23}, {▽, 23}},
  FrameLabel → {Style["Energy, kHz", 18], Style["Energy, kHz", 18]}, PlotLabel →
  Style["Comparison of eigen-values with different numerical methods", 18],
  PlotLegends → Placed[{"Finite Difference Method", "Impedance Method"},
  {.75, .25}], FrameTicksStyle → Medium, ImageSize → Large,
  PlotStyle → Directive[Thickness[.006]], Frame → True]
```



**Export[**

```
"/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis
nb's/possible figs/compair3030.jpeg", compair3030];
```

```

energyBetterer3033 = 2 *  $\pi$  *  $\hbar$  * 10^3 * Table[i, {i, .0001, 10, .00001}];
Dimensions[energyBetterer3033]
eigenEnergyBetterer3033 =
  EigenEnergy[potential[[2, 3, 4, All]], energyBetterer3033, 1];
Dimensions[eigenEnergyBetterer3033]
energyBetterer23033 = 2 *  $\pi$  *  $\hbar$  * 10^3 * Table[i, {i, 10, 15, .00001}];
Dimensions[energyBetterer23033]
eigenEnergyBetterer23033 =
  EigenEnergy[potential[[2, 3, 4, All]], energyBetterer23033, 1];
Dimensions[eigenEnergyBetterer23033]
energyBetterer33033 = 2 *  $\pi$  *  $\hbar$  * 10^3 * Table[i, {i, 14, 20, .00001}];
Dimensions[energyBetterer33033]
eigenEnergyBetterer33033 =
  EigenEnergy[potential[[2, 3, 4, All]], energyBetterer33033, 1];
Dimensions[eigenEnergyBetterer33033]
energyBetterer43033 = 2 *  $\pi$  *  $\hbar$  * 10^3 * Table[i, {i, 19, 25, .00001}];
Dimensions[energyBetterer43033]
eigenEnergyBetterer43033 =
  EigenEnergy[potential[[2, 3, 4, All]], energyBetterer43033, 1];
Dimensions[eigenEnergyBetterer43033]
energyBetterer53033 = 2 *  $\pi$  *  $\hbar$  * 10^3 * Table[i, {i, 24, 30, .00001}];
Dimensions[energyBetterer53033]
eigenEnergyBetterer53033 =
  EigenEnergy[potential[[2, 3, 4, All]], energyBetterer53033, 1];
Dimensions[eigenEnergyBetterer53033]
{999 991}

{97 036}

{500 001}

{44}

{600 001}

{69}

{600 001}

{106}

{600 001}

{198}

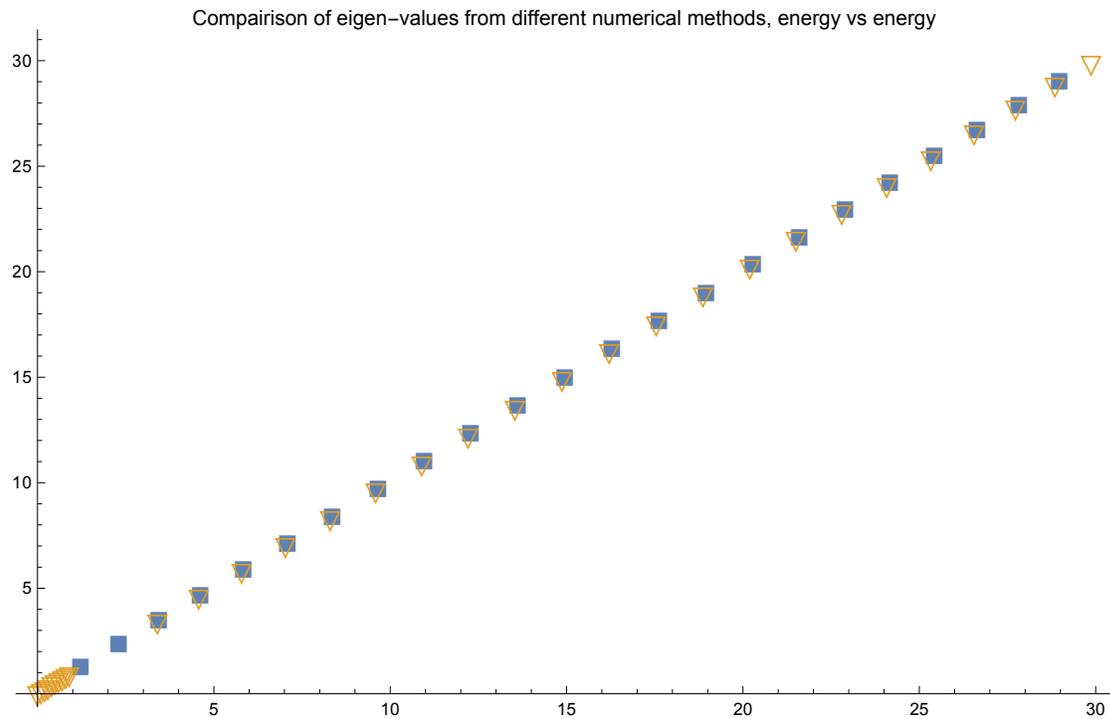
a3033 = DeleteDuplicates[Join[eigenEnergyBetterer3033 / (2 *  $\pi$  *  $\hbar$  * 1.0*^3),
  eigenEnergyBetterer23033 / (2 *  $\pi$  *  $\hbar$  * 1.0*^3), eigenEnergyBetterer33033 /
  (2 *  $\pi$  *  $\hbar$  * 1.0*^3), eigenEnergyBetterer43033 / (2 *  $\pi$  *  $\hbar$  * 1.0*^3),
  eigenEnergyBetterer53033 / (2 *  $\pi$  *  $\hbar$  * 1.0*^3)], Abs[#1 - #2] < .1 &]
{0.0001, 0.1001, 0.2001, 0.3001, 0.4001, 0.5001, 0.6001, 0.7001,
  0.8001, 0.9001, 3.40171, 4.57519, 5.7874, 7.0301, 8.29861, 9.58744,
  10.8929, 12.2109, 13.538, 14.8707, 16.206, 17.5405, 18.871, 20.194,
  21.506, 22.8029, 24.0802, 25.3325, 26.5526, 27.7302, 28.8457, 29.8734}

```

**Differences[a3033]**

```
{0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 2.50161, 1.17348, 1.21221, 1.2427,
 1.26851, 1.28883, 1.3055, 1.31791, 1.32715, 1.33273, 1.33531, 1.33445, 1.33049,
 1.32299, 1.312, 1.29691, 1.27735, 1.25226, 1.22015, 1.17755, 1.11548, 1.02776}
```

```
compair3033 = ListPlot[{Transpose@{cams3033, cams3033}, Transpose@{a3033, a3033}},
  ImageSize → Large, PlotRange → All, PlotMarkers → {{■, 20}, {▽, 20}},
  PlotLabel → "Comparison of eigen-values
  from different numerical methods, energy vs energy"]
```



**Export[**

```
"/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis
nb's/possible figs/compair3033.jpeg", compair3033];
```

```

energyBetter =
  2 *  $\pi$  *  $\hbar$  * 10^3 * Join[Table[i, {i, .01, .6, .01}], Table[i, {i, .6, 1.0, .001}],
    Table[i, {i, 1.0, 1.5, .01}], Table[i, {i, 1.5, 2.0, .0001}],
    Table[i, {i, 2.0, 2.5, .01}], Table[i, {i, 2.5, 3.0, .0001}],
    Table[i, {i, 3.0, 3.5, .01}], Table[i, {i, 3.5, 4.2, .0001}],
    Table[i, {i, 4.2, 4.7, .01}], Table[i, {i, 4.7, 5.4, .0001}],
    Table[i, {i, 5.4, 5.8, .01}], Table[i, {i, 5.8, 6.4, .0001}], Table[i,
      {i, 6.4, 6.9, .01}], Table[i, {i, 6.9, 7.9, .0001}], Table[i, {i, 7.9, 9.2, .01}],
    Table[i, {i, 9.2, 10.0, .0001}], Table[i, {i, 10.0, 10.5, .01}],
    Table[i, {i, 10.5, 11.3, .0001}], Table[i, {i, 11.3, 11.8, .01}],
    Table[i, {i, 11.8, 12.4, .0001}], Table[i, {i, 12.4, 12.9, .01}],
    Table[i, {i, 12.9, 13.5, .0001}], Table[i, {i, 13.5, 14.2, .01}],
    Table[i, {i, 14.2, 14.7, .0001}], Table[i, {i, 14.7, 15.5, .01}],
    Table[i, {i, 15.5, 16.0, .0001}], Table[i, {i, 16.0, 16.8, .01}],
    Table[i, {i, 16.8, 17.2, .0001}], Table[i, {i, 17.2, 17.9, .01}],
    Table[i, {i, 17.9, 18.4, .0001}], Table[i, {i, 18.4, 19.2, .001}],
    Table[i, {i, 19.2, 19.8, .00001}], Table[i, {i, 19.8, 20.5, .001}],
    Table[i, {i, 20.5, 20.9, .00001}], Table[i, {i, 20.9, 21.7, .001}],
    Table[i, {i, 21.7, 22.1, .00001}], Table[i, {i, 22.1, 22.9, .001}],
    Table[i, {i, 22.9, 23.3, .00001}], Table[i, {i, 23.3, 24.0, .001}],
    Table[i, {i, 24.0, 24.5, .00001}], Table[i, {i, 24.5, 25.3, .001}],
    Table[i, {i, 25.3, 25.7, .00001}], Table[i, {i, 25.7, 26.5, .001}],
    Table[i, {i, 26.5, 26.9, .00001}], Table[i, {i, 26.9, 27.5, .001}],
    Table[i, {i, 27.5, 27.9, .00001}], Table[i, {i, 27.9, 28.5, .001}],
    Table[i, {i, 28.5, 29.1, .00001}], Table[i, {i, 29.1, 29.4, .01}],
    Table[i, {i, 29.4, 29.9, .0001}], Table[i, {i, 29.9, 40, .1}]];
Dimensions[energyBetter]
reflectCoefBetter3030 =
  ReflectionProbability[potential[[2, 3, 1, All]], energyBetter, 1];
Dimensions[reflectCoefBetter3030]
eigenEnergyBetter3030 = EigenEnergy[potential[[2, 3, 1, All]], energyBetter, 1];
Dimensions[eigenEnergyBetter3030]

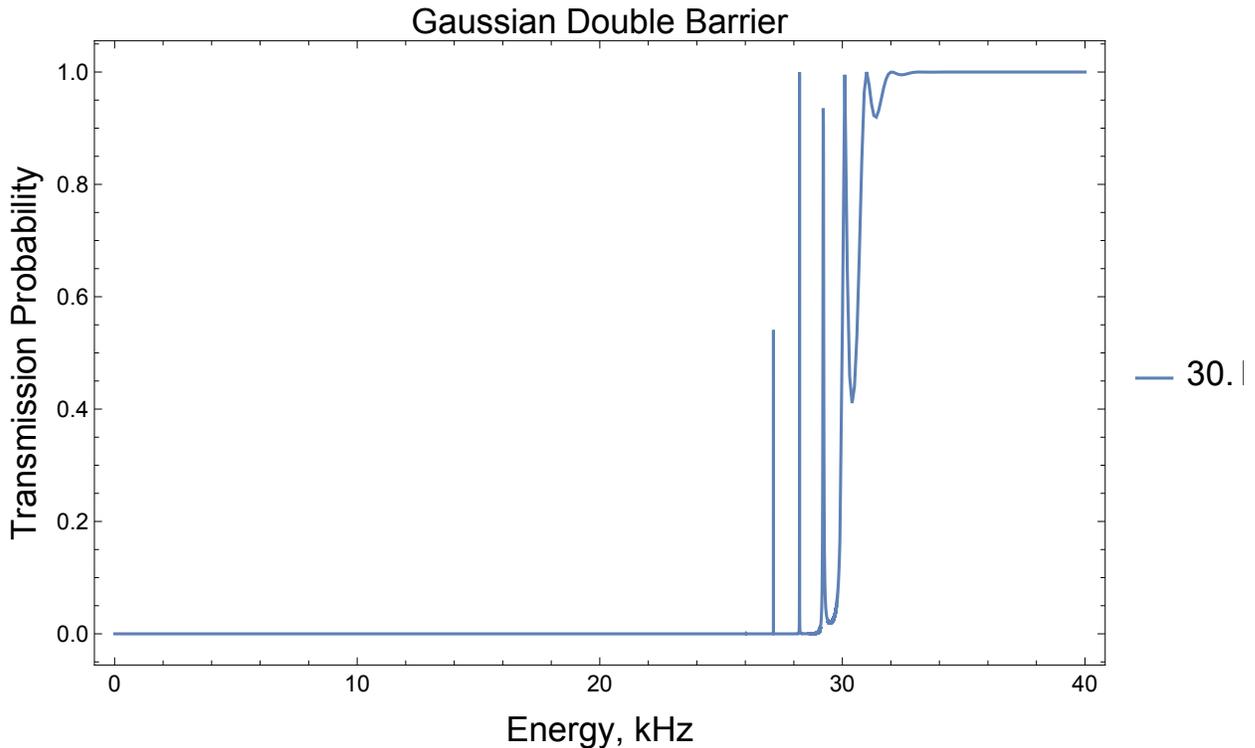
{513 711}

{513 711}

{84}

```

```
ListLinePlot[Transpose@{energyBetter/(2 * π * ħ * 1.0*^3), 1 - reflectCoefBetter3030},
  PlotRange → All (*{{20,38},{-0.01,1.005}}*), GridLines → {{}, {}},
  FrameLabel → {Style[(*"Energy/v0"*)"Energy, kHz", 18],
    Style["Transmission Probability", 18]},
  PlotLabel → Style[(*"Varied Potential Height"*)"Gaussian Double Barrier", 18],
  PlotLegends → {Table[Style["kHz = u02" u02[[j]]/(2 * π * ħ * 10^3), 18], {j, 1, jL}]},
  FrameTicksStyle → Medium, ImageSize → Large, Frame → True]
```



```
positBetter3030 = Flatten[Position[1 - reflectCoefBetter3030, _? (# > .01 &)]];
energyResonanceBetter3030 =
  Table[energyBetter[[positBetter3030[[i]]]]/(2 * π * ħ * 1.0*^3),
    {i, Length[positBetter3030]}];
Dimensions[energyResonanceBetter3030]
{6394}
```

```
DeleteDuplicates[energyResonanceBetter3030, Abs[#1 - #2] < .1 &]
```

```
{27.16, 28.221, 29.0877, 29.19, 29.29, 29.39, 29.4901, 29.5901, 29.6902, 29.7902,
  29.8902, 30., 30.2, 30.4, 30.5, 30.7, 30.9, 31., 31.2, 31.4, 31.5, 31.7, 31.9,
  32., 32.1, 32.3, 32.4, 32.5, 32.6, 32.8, 32.9, 33., 33.1, 33.3, 33.4, 33.5,
  33.6, 33.8, 33.9, 34., 34.1, 34.3, 34.4, 34.5, 34.6, 34.7, 34.9, 35., 35.1,
  35.2, 35.4, 35.5, 35.6, 35.7, 35.9, 36., 36.1, 36.2, 36.4, 36.5, 36.6, 36.7,
  36.9, 37., 37.1, 37.2, 37.4, 37.5, 37.6, 37.7, 37.9, 38., 38.1, 38.2, 38.4,
  38.6, 38.7, 38.8, 38.9, 39., 39.1, 39.2, 39.4, 39.6, 39.7, 39.8, 39.9, 40.}
```

```
DeleteDuplicates[eigenEnergyBetter3030 / (2 * π * ħ * 1.0^3), Abs[#1 - #2] < .1 &]
{9.367, 10.6449, 11.9344, 13.2326, 14.5355, 15.8404, 17.1435, 18.442, 19.7318,
 21.01, 25.909, 34.3, 35.4, 35.5, 35.6, 35.7, 35.9, 36., 36.1, 36.2, 36.4, 36.5,
 36.6, 36.7, 36.9, 37., 37.1, 37.2, 37.4, 37.5, 37.6, 37.7, 37.9, 38., 38.1, 38.2,
 38.4, 38.6, 38.7, 38.8, 38.9, 39., 39.1, 39.2, 39.4, 39.5, 39.7, 39.8, 39.9, 40.}
```

```
In[255]= meFirst = Table[If[Intersection[potential[[n, k, j, All]], energy] == {},
  energy, Delete[energy, Flatten[
    Position[energy, Intersection[potential[[n, k, j, All]], energy][[1]]]]],
  {n, 1, nL}, {k, 1, kL}, {j, 1, jL}];
Dimensions[meFirst]
energyGood =
  Table[If[Intersection[potential[[n, k, j, All]], meFirst[[n, k, j]]] == {},
    meFirst[[n, k, j]], Delete[meFirst[[n, k, j]],
      Flatten[Position[meFirst[[n, k, j]], Intersection[potential[[n, k, j, All]],
        meFirst[[n, k, j]][[1]]]]], {n, 1, nL}, {k, 1, kL}, {j, 1, jL}];
Dimensions[energyGood]
```

```
Out[256]= {4, 5, 4}
```

```
Out[258]= {4, 5, 4}
```

```
In[259]= Table[Length[energyGood[[n, k, j]]], {n, 1, nL}, {k, 1, kL}, {j, 1, jL}] // MatrixForm
```

```
Out[259]//MatrixForm=
```

$$\begin{pmatrix} \begin{pmatrix} 850\,001 \\ 850\,001 \\ 850\,001 \\ 850\,001 \end{pmatrix} & \begin{pmatrix} 850\,001 \\ 850\,001 \\ 850\,001 \\ 850\,001 \end{pmatrix} & \begin{pmatrix} 850\,000 \\ 849\,999 \\ 849\,999 \\ 849\,999 \end{pmatrix} & \begin{pmatrix} 850\,001 \\ 850\,001 \\ 850\,001 \\ 850\,001 \end{pmatrix} & \begin{pmatrix} 850\,001 \\ 850\,001 \\ 850\,001 \\ 850\,001 \end{pmatrix} \\ \begin{pmatrix} 850\,001 \\ 850\,001 \\ 850\,001 \\ 850\,001 \end{pmatrix} & \begin{pmatrix} 850\,001 \\ 850\,001 \\ 850\,001 \\ 850\,001 \end{pmatrix} & \begin{pmatrix} 850\,001 \\ 850\,001 \\ 850\,001 \\ 850\,001 \end{pmatrix} & \begin{pmatrix} 850\,001 \\ 850\,001 \\ 850\,001 \\ 850\,001 \end{pmatrix} & \begin{pmatrix} 850\,001 \\ 850\,001 \\ 850\,001 \\ 850\,001 \end{pmatrix} \\ \begin{pmatrix} 850\,001 \\ 850\,001 \\ 850\,001 \\ 850\,001 \end{pmatrix} & \begin{pmatrix} 850\,001 \\ 850\,001 \\ 850\,001 \\ 850\,001 \end{pmatrix} & \begin{pmatrix} 850\,001 \\ 850\,001 \\ 850\,001 \\ 850\,001 \end{pmatrix} & \begin{pmatrix} 850\,001 \\ 850\,001 \\ 850\,001 \\ 850\,001 \end{pmatrix} & \begin{pmatrix} 850\,001 \\ 850\,001 \\ 850\,001 \\ 850\,001 \end{pmatrix} \\ \begin{pmatrix} 850\,001 \\ 850\,001 \\ 850\,001 \\ 850\,001 \end{pmatrix} & \begin{pmatrix} 850\,001 \\ 850\,001 \\ 850\,001 \\ 850\,001 \end{pmatrix} & \begin{pmatrix} 850\,001 \\ 850\,001 \\ 850\,001 \\ 850\,001 \end{pmatrix} & \begin{pmatrix} 850\,001 \\ 850\,001 \\ 850\,001 \\ 850\,001 \end{pmatrix} & \begin{pmatrix} 850\,001 \\ 850\,001 \\ 850\,001 \\ 850\,001 \end{pmatrix} \\ \begin{pmatrix} 850\,001 \\ 850\,001 \\ 850\,001 \\ 850\,001 \end{pmatrix} & \begin{pmatrix} 850\,001 \\ 850\,001 \\ 850\,001 \\ 850\,001 \end{pmatrix} & \begin{pmatrix} 850\,001 \\ 850\,001 \\ 850\,001 \\ 850\,001 \end{pmatrix} & \begin{pmatrix} 850\,001 \\ 850\,001 \\ 850\,001 \\ 850\,001 \end{pmatrix} & \begin{pmatrix} 850\,001 \\ 850\,001 \\ 850\,001 \\ 850\,001 \end{pmatrix} \end{pmatrix}$$

```
t = AbsoluteTime[];
reflectCoef = Monitor[
  Table[ReflectionProbability[potential[[n, k, j, All]], energyGood[[n, k, j]], 1],
    {n, 1, nL}, {k, 1, kL}, {j, 1, jL}], Row[ProgressIndicator[j, {1, jL}], j], " "];
Dimensions[reflectCoef]
AbsoluteTime[] - t
{4, 5, 4}
```

```
22 656.839627
```

```
In[260]:= t = AbsoluteTime[];
eigenEnergy =
  Monitor[Table[EigenEnergy[potential[[n, k, j, All]], energyGood[[n, k, j]], 1],
    {n, 1, nL}, {k, 1, kL}, {j, 1, jL}], Row[ProgressIndicator[k, {1, kL}], k], " "];
Dimensions[eigenEnergy]
AbsoluteTime[] - t
```

```
Out[262]= {4, 5, 4}
```

```
Out[263]= 41759.347563
```

```
DeleteDuplicates[eigenEnergy[[1, 3, 1]] / (2 *  $\pi$  *  $\hbar$  * 1.0 * 10^3), Abs[#1 - #2] < .1 &]
{2.221, 4.7191, 5.6741, 6.6809, 7.7348, 8.8317, 9.9678, 11.1392, 12.3427, 13.5747,
 14.8318, 16.1108, 17.408, 18.7198, 20.0423, 21.371, 22.7009, 24.0261, 25.3387,
 26.6274, 27.8735, 29.0493, 30.1596, 31.3055, 32.5645, 33.927, 35.3674, 36.858,
 38.3423, 38.4423, 38.5424, 38.6424, 39.6653, 39.7653, 39.8654, 39.9654}
```

```
Length[eigenEnergy[[2, 2, 3]]]
```

```
1304
```

```
reflectAmp = Monitor[
  Table[ReflectionAmplitude[potential[[n, k, j, All]], energyGood[[n, k, j]], 1],
    {n, 1, nL}, {k, 1, kL}, {j, 1, jL}], Row[ProgressIndicator[k, {1, kL}], k], " "];
Dimensions[reflectAmp]
{3, 3, 4, 4000}
```

Plot Reflection and Transmission behavior:

```
(*yticks=Table[{i*10, Round[u02[[i*10]]/(2* $\pi$ * $\hbar$ *10^3)]}, {i, 1, jL/10}];
xticks=Table[{i*1000, Round[energy[[i*1000]]/(2* $\pi$ * $\hbar$ *10^3)]},
  {i, 1, Length[energy]/1000}];*)

(*jet[u_?NumericQ] :=
  Blend[{{0, RGBColor[0, 0, 9/16]}, {1/9, Blue}, {23/63, Cyan}, {13/21, Yellow},
    {47/63, Orange}, {55/63, Red}, {1, RGBColor[1/2, 0, 0]}], u] /; 0 <= u <= 1*)

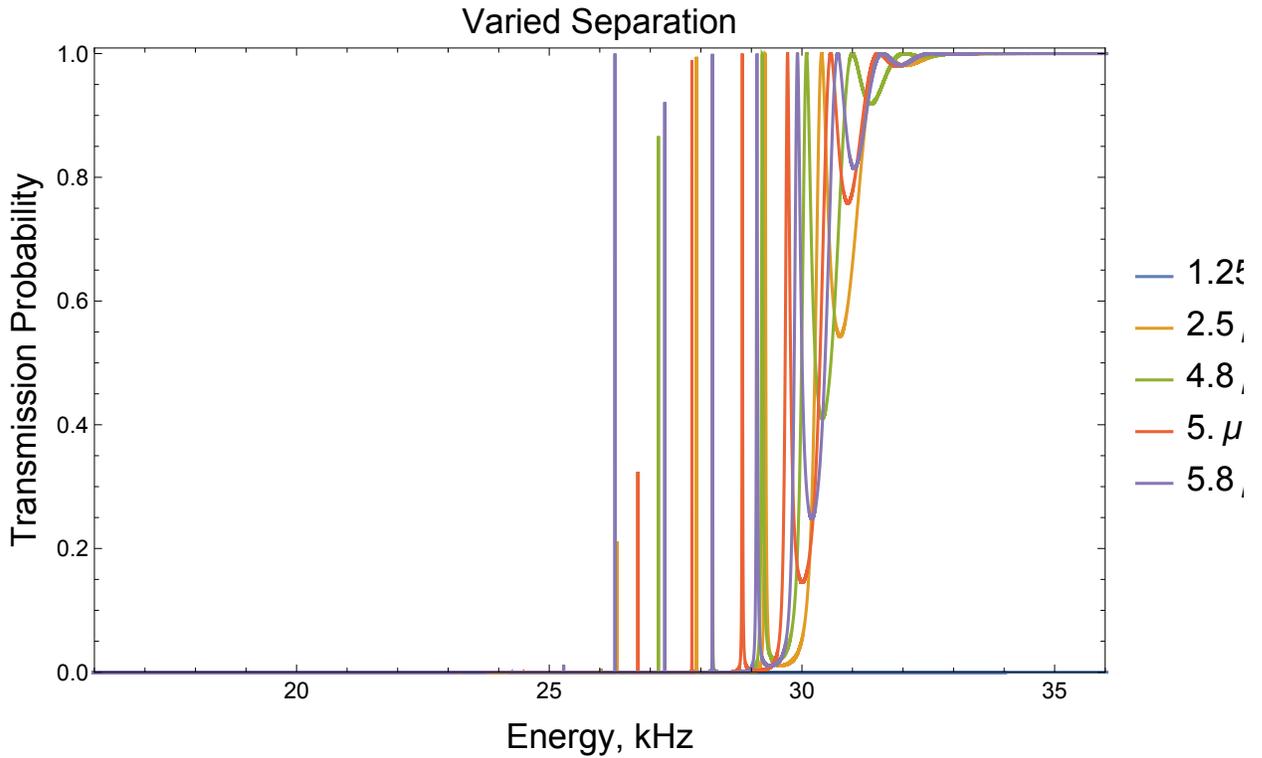
(*ListDensityPlot[1-reflectCoef[[1, 1, All, All]], PlotLegends->Automatic,
  Frame->True, ColorFunction->jet, FrameTicks->{{yticks, None}, {xticks, None}},
  FrameLabel->{{"U02, kHz", "Transmission"}, {"Incoming Energy, kHz", None}}]*)

(*ListLinePlot[
  Table[Transpose@{energy/(2* $\pi$ * $\hbar$ *10^3), reflectCoef[[1, k, 2, All]]}, {k, 1, kL}],
  PlotRange->{{27, 36}, {-0.01, 1.01}}, GridLines->{{}, {}},
  AxesLabel->{"Energy, kHz", "Reflection Probability"},
  PlotLabel->"Varied separation",
  PlotLegends->{Table[" $\mu$ m = xsep" xsep[[k]]/10^-6, {k, 1, kL}]}]*)
```

```

varsepTrans2k1 = ListLinePlot[Table[
  Transpose@{energyGood[[2, k, 1]] / (2 *  $\pi$  *  $\hbar$  * 10^3), 1 - reflectCoef[[2, k, 1]]},
  {k, 1, kL}], PlotRange -> {{16.0, 36.0}, {0, 1.009}}, GridLines -> {{}, {}},
  FrameLabel -> {Style["Energy, kHz", 18], Style["Transmission Probability", 18]},
  PlotLabel -> Style["Varied Separation", 18],
  PlotLegends -> {Table[Style[" $\mu$ m = xsep" xsep[[k]] / 10^-6, 18], {k, 1, kL}]},
  FrameTicksStyle -> Medium, ImageSize -> Large, Frame -> True]

```



```

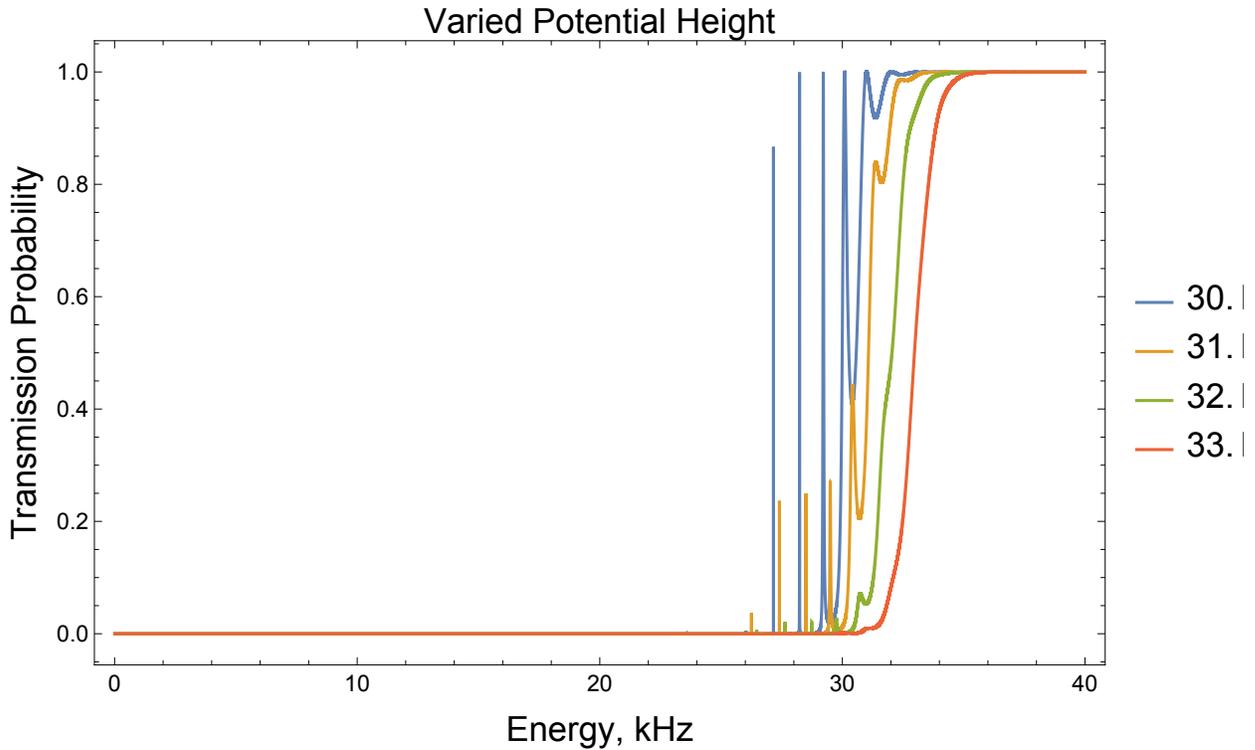
Export[
  "/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis
  nb's/possible figs/varsepTrans2k1.jpeg", varsepTrans2k1];

```

```

varheightTrans23j = ListLinePlot[Table[
  Transpose[{energyGood[[2, 3, j]] / (2 *  $\pi$  *  $\hbar$  * 1.0*^3), 1 - reflectCoef[[2, 3, j]]},
  {j, 1, jL}], PlotRange -> All(*{{20, 38}, {-0.01, 1.005}}*),
  GridLines -> {{}, {}}, FrameLabel -> {Style[(*"Energy/v0"*)"Energy, kHz", 18],
  Style["Transmission Probability", 18]},
  PlotLabel -> Style["Varied Potential Height" (*"Gaussian Double Barrier"*) , 18],
  PlotLegends -> {Table[Style["kHz = u02" u02[[j]] / (2 *  $\pi$  *  $\hbar$  * 10^3), 18], {j, 1, jL}]},
  FrameTicksStyle -> Medium, ImageSize -> Large, Frame -> True]

```

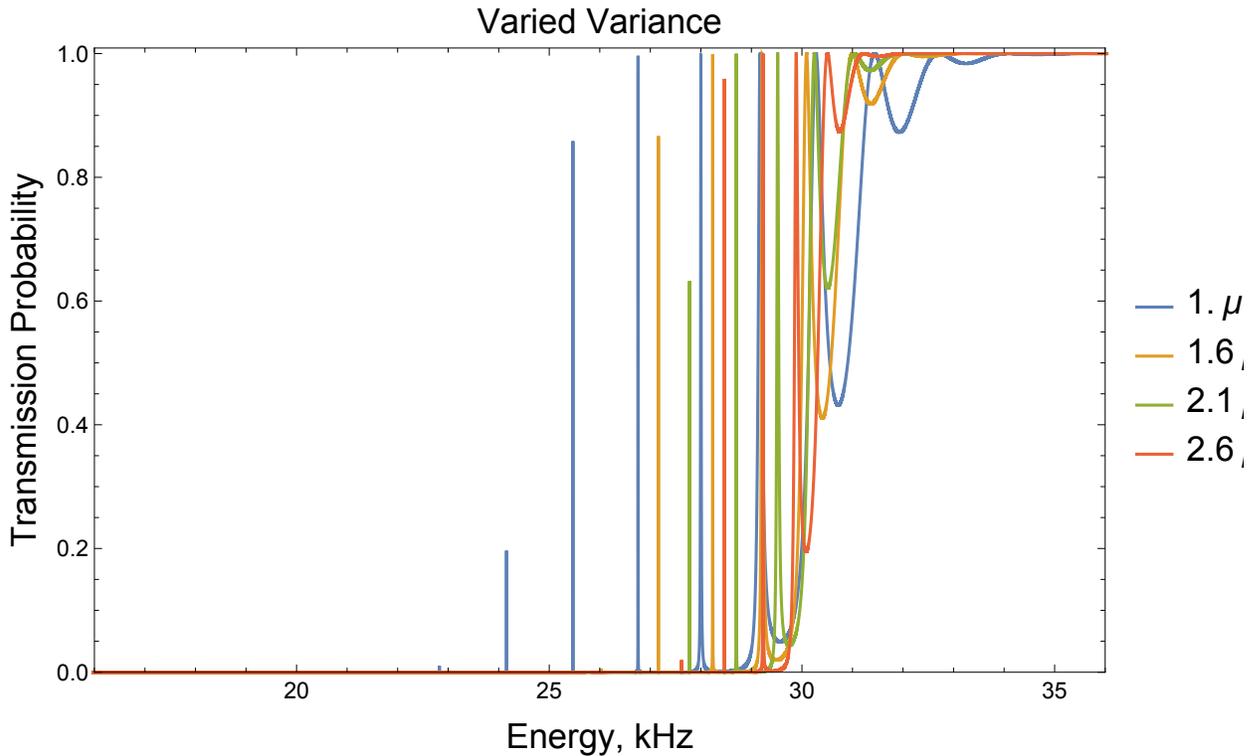


```

Export[
  "/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis
  nb's/possible figs/varheightTrans23j.jpeg", varheightTrans23j];

```

```
varVarTransn31 = ListLinePlot[Table[
  Transpose@{energyGood[[n, 3, 1]] / (2 *  $\pi$  *  $\hbar$  * 10^3), 1 - reflectCoef[[n, 3, 1]]},
  {n, 1, nL}], PlotRange -> {{16.0, 36.0}, {0, 1.009}}, GridLines -> {{}, {}},
  FrameLabel -> {Style["Energy, kHz", 18], Style["Transmission Probability", 18]},
  PlotLabel -> Style["Varied Variance", 18],
  PlotLegends -> {Table[Style[" $\mu$ m =  $\sigma$ " sigx[[n]] / 10^-6, 18], {n, 1, nL}]},
  FrameTicksStyle -> Medium, ImageSize -> Large, Frame -> True]
```



```
Export[
  "/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis
  nb's/possible figs/varVarTransn31.jpeg", varVarTransn31];
```

```

sep1V30 =
  Table[Round[Select[DeleteDuplicates[eigenEnergy[[1, k, 1]] / (2 * π * ħ * 1.0*^3),
    Abs[#1 - #2] < .1 &], # ≤ 30 &], .01], {k, 2, kL}];
sep2V30 = Table[Round[Select[DeleteDuplicates[eigenEnergy[[2, k, 1]] /
  (2 * π * ħ * 1.0*^3), Abs[#1 - #2] < .1 &], # ≤ 30 &], .01], {k, 2, kL}];
sep3V30 = Table[Round[Select[DeleteDuplicates[eigenEnergy[[3, k, 1]] /
  (2 * π * ħ * 1.0*^3), Abs[#1 - #2] < .1 &], # ≤ 30 &], .01], {k, 2, kL}];
sep4V30 = Table[Round[Select[DeleteDuplicates[eigenEnergy[[4, k, 1]] /
  (2 * π * ħ * 1.0*^3), Abs[#1 - #2] < .1 &], # ≤ 30 &], .01], {k, 2, kL}];
Dimensions[sep1V30]
Dimensions[sep2V30]
Dimensions[sep3V30]
Dimensions[sep4V30]

{4}

{4}

{4}

{4}

Table[Length[sep1V30[[i]]], {i, 1, 4}]
Table[Length[sep2V30[[i]]], {i, 1, 4}]
Table[Length[sep3V30[[i]]], {i, 1, 4}]
Table[Length[sep4V30[[i]]], {i, 1, 4}]

{12, 22, 25, 26}

{39, 24, 26, 30}

{231, 20, 21, 27}

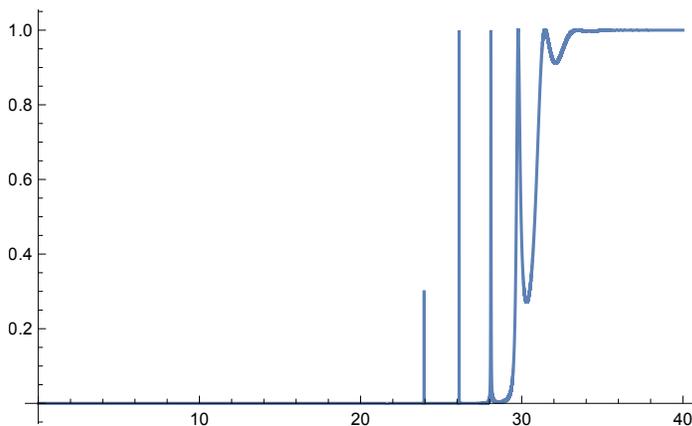
{300, 46, 32, 23}

```

```

ListLinePlot[
  Transpose@{energyGood[[1, 2, 1]] / (2 * π * ħ * 10^3), 1 - reflectCoef[[1, 2, 1]]}]

```



```

sep1V31 =
  Table[Round[Select[DeleteDuplicates[eigenEnergy[[1, k, 2]] / (2 * π * ħ * 1.0*^3),
    Abs[#1 - #2] < .1 &], # ≤ 31 &], .01], {k, 2, kL}];
Dimensions[sep1V31]
sep1V32 =
  Table[Round[Select[DeleteDuplicates[eigenEnergy[[1, k, 3]] / (2 * π * ħ * 1.0*^3),
    Abs[#1 - #2] < .1 &], # ≤ 32 &], .01], {k, 2, kL}];
Dimensions[sep1V32]
sep1V33 =
  Table[Round[Select[DeleteDuplicates[eigenEnergy[[1, k, 4]] / (2 * π * ħ * 1.0*^3),
    Abs[#1 - #2] < .1 &], # ≤ 33 &], .01], {k, 2, kL}];
Dimensions[
  sep1V33]
{4}
{4}
{4}

```

```

data12Vh =
  {"Resonant energies, kHz, whose variance is 1.0μm and spacing is 2.5μm",
  SpanFromLeft}, Join[{u02[[1]] / (2 * π * ħ * 1.0*^3)}, sep1V30[[1]]],
  Join[{u02[[2]] / (2 * π * ħ * 1.0*^3)}, sep1V31[[1]]],
  Join[{u02[[3]] / (2 * π * ħ * 1.0*^3)}, sep1V32[[1]]],
  Join[{u02[[4]] / (2 * π * ħ * 1.0*^3)}, sep1V33[[1]]]};

```

```

sep2sig1VhTable = Grid[data12Vh(*, Alignment → {Left}, Spacings → {12, -1}*),
  Frame → All, Background → {{LightGray, None}, {None, None}}]

```

Resonant energies, kHz, whose variance is 1.0μm and spacing is 2.5μm													
30.	3.87	6.39	8.95	11.51	14.06	16.58	19.04	21.44	23.74	25.92	27.9	29.62	
31.	3.92	6.46	9.05	11.63	14.2	16.74	19.23	21.64	23.97	26.17	28.19	30.77	
32.	6.53	9.14	11.74	14.33	16.89	19.4	22.55	24.19	26.42	28.46	30.21	31.69	
33.	4.02	6.6	11.48	19.57	22.03	24.41	26.65	28.71	30.51	32.21			

```

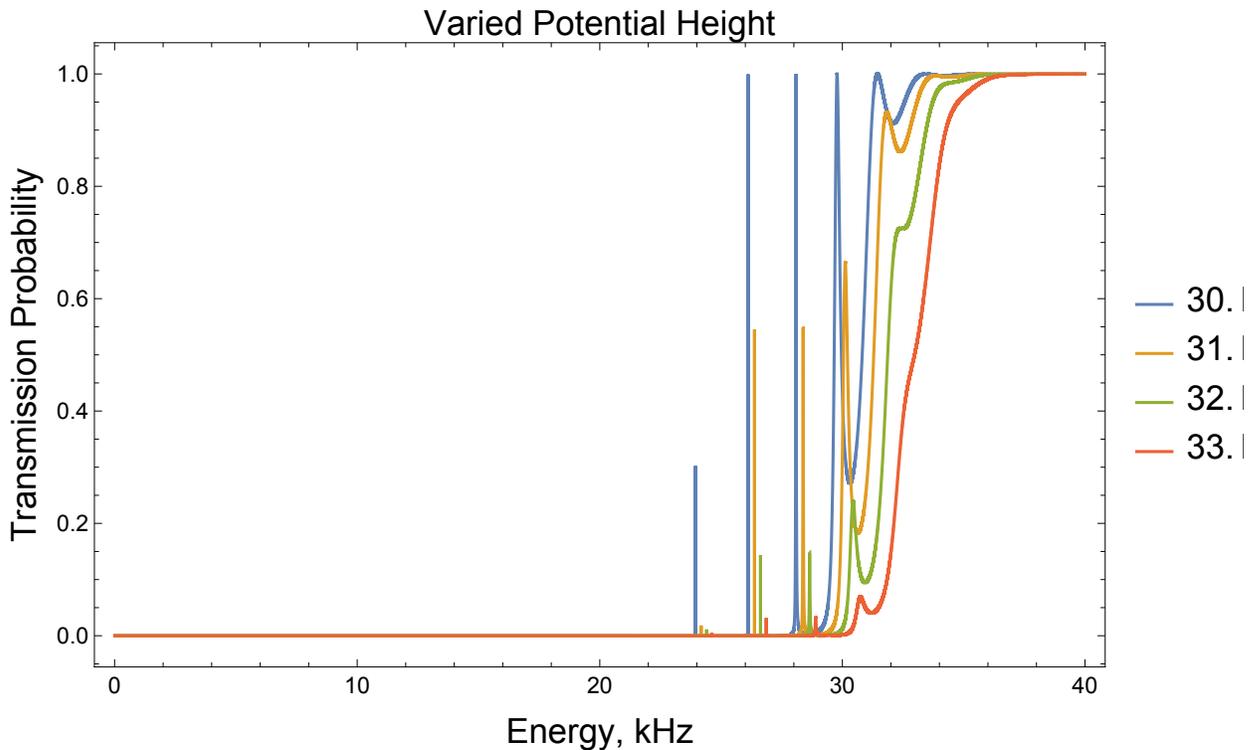
Export[
  "/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis
  nb's/possible figs/sep2sig1VhTable.jpeg", sep2sig1VhTable];

```

```

varheightTrans12j = ListLinePlot[Table[
  Transpose[{energyGood[[1, 2, j]] / (2 *  $\pi$  *  $\hbar$  * 1.0*^3), 1 - reflectCoef[[1, 2, j]]},
  {j, 1, jL}], PlotRange -> All(*{{20, 38}, {-0.01, 1.005}}*),
  GridLines -> {{}, {}}, FrameLabel -> {Style[(*"Energy/v0"*)"Energy, kHz", 18],
  Style["Transmission Probability", 18]},
  PlotLabel -> Style["Varied Potential Height" (*"Gaussian Double Barrier"*) , 18],
  PlotLegends -> {Table[Style["kHz = u02" u02[[j]] / (2 *  $\pi$  *  $\hbar$  * 10^3), 18], {j, 1, jL}]},
  FrameTicksStyle -> Medium, ImageSize -> Large, Frame -> True]

```



```

Export[
  "/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis
  nb's/possible figs/varheightTrans12j.jpeg", varheightTrans12j];

```

```

data13Vhfirst =
  {"Resonant energies, kHz, whose variance is 1.0 $\mu$ m and spacing is 2.5 $\mu$ m",
  SpanFromLeft}, Join[{u02[[1]] / (2 *  $\pi$  *  $\hbar$  * 1.0*^3)}, sep1V30[[2]]],
  Join[{u02[[2]] / (2 *  $\pi$  *  $\hbar$  * 1.0*^3)}, sep1V31[[2]]]};

```

```
sep3sig1VhTablefirst = Grid[data13Vhfirst(*,Alignment->{Left},
    Spacings->{2,2}*), Frame -> All, Background -> {{LightGray, None}, {None, None}}]
```

Resonant energies, kHz, whose variance is 1.0μm and spacing is 2.5μm																
30.	2.2	4.7	5.6	6.6	7.3	8.3	9.7	11.1	12.4	13.5	14.8	16.1	17.4	18.7	20.0	21.3
	2	2	7	8	3	3	7	1	3	5	8	1	4	7	0	3
	2	2	7	8	3	3	7	1	3	5	8	1	4	7	0	3
31.	3.8	4.7	5.7	6.7	7.7	8.7	10.0	11.1	12.4	13.5	14.8	16.1	17.4	18.7	20.0	21.3
	4	4		1	7	7	0	1	4	6	9	1	5	8	1	4
	4	4		1	7	7	0	1	4	6	9	1	5	8	1	4

```
data13Vhsecond = {"Resonant energies, kHz, whose variance is 1.0μm and spacing is 2.5μm",
    SpanFromLeft}, Join[{u02[[3]] / (2 * π * ħ * 1.0*^3)}, sep1V32[[2]]],
    Join[{u02[[4]] / (2 * π * ħ * 1.0*^3)}, sep1V33[[2]]]};
```

```
sep3sig1VhTablesecond = Grid[data13Vhsecond(*,Alignment->{Left},
    Spacings->{2,2}*), Frame -> All, Background -> {{LightGray, None}, {None, None}}]
```

Resonant energies, kHz, whose variance is 1.0μm and spacing is 2.5μm													
32.	0.95	3.01	5.72	11.24	12.46	14.97	17.58	23.39	26.95	28.23	29.44	30.57	31.61
33.	0.46	2.25	13.24	20.34	21.7	23.06	24.42	25.77	27.1	28.39	29.62	30.79	31.96

```
In[270]:= energyEigsH31sig2sep3 = 2 * π * ħ * 10^3 * Table[i, {i, .5, 35, .00001}];
Dimensions[energyEigsH31sig2sep3]
meFirstEigsH31 =
    If[Intersection[potential[[2, 3, 2, All]], energyEigsH31sig2sep3] == {},
        energyEigsH31sig2sep3,
        Delete[energyEigsH31sig2sep3, Flatten[Position[energyEigsH31sig2sep3,
            Intersection[potential[[2, 3, 2, All]], energyEigsH31sig2sep3][[1]]]]];
Dimensions[meFirstEigsH31]
energyGoodEigsH31 = If[Intersection[potential[[2, 3, 2, All]], meFirstEigsH31] == {},
    meFirstEigsH31, Delete[meFirstEigsH31, Flatten[Position[meFirstEigsH31,
        Intersection[potential[[2, 3, 2, All]], meFirstEigsH31][[1]]]]];
Dimensions[energyGoodEigsH31]
```

Out[271]= {3 450 001}

Out[273]= {3 450 001}

Out[275]= {3 450 001}

```

In[276]:= t = AbsoluteTime[];
eigenEnergyEigsig2sep3H31 =
  EigenEnergy[potential[[2, 3, 2, All]], energyGoodEigsH31, 1];
Dimensions[eigenEnergyEigsig2sep3H31]
AbsoluteTime[] - t
Out[278]= {4658}
Out[279]= 2856.751361

In[280]:= energyEigs =
  2 *  $\pi$  *  $\hbar$  * 10^3 * Join[Table[i, {i, .5, 10, .00001}], Table[i, {i, 10, 35, .0001}]];
Dimensions[energyEigs]
meFirstEigs =
  Table[If[Intersection[potential[[n, k, j, All]], energyEigs] == {}, energyEigs,
    Delete[energyEigs, Flatten[Position[energyEigs, Intersection[potential[[n, k,
      j, All]], energyEigs][[1]]]]], {n, 1, nL}, {k, 1, kL}, {j, 1, jL}];
Dimensions[meFirstEigs]
energyGoodEigs =
  Table[If[Intersection[potential[[n, k, j, All]], meFirstEigs[[n, k, j]]] == {},
    meFirstEigs[[n, k, j]],
    Delete[meFirstEigs[[n, k, j]], Flatten[Position[meFirstEigs[[n, k, j]],
      Intersection[potential[[n, k, j, All]], meFirstEigs[[n, k, j]][[1]]]]],
    {n, 1, nL}, {k, 1, kL}, {j, 1, jL}];
Dimensions[energyGoodEigs]
Out[281]= {1 200 002}
Out[283]= {4, 5, 4}
Out[285]= {4, 5, 4}

In[286]:= t = AbsoluteTime[];
eigenEnergyEigsig2sep3H = Table[EigenEnergy[
  potential[[2, 3, j, All]], energyGoodEigs[[2, 3, j]], 1], {j, 2, jL, 2}];
Dimensions[eigenEnergyEigsig2sep3H]
AbsoluteTime[] - t
Out[288]= {2}
Out[289]= 1807.404821

Round[Select[DeleteDuplicates[eigenEnergyEigsig2sep3H[[1]] / (2 *  $\pi$  *  $\hbar$  * 1.0*^3),
  Abs[#1 - #2] < .1 &], # <= 32 &], .01]
{1.18, 2.23, 3.34, 4.5, 5.69, 6.92, 8.17, 9.44, 12.03, 13.34, 15.97, 17.28,
  18.59, 19.89, 21.18, 22.46, 23.71, 26.43, 28.39, 29.4, 30.27, 31.03, 31.84}

```

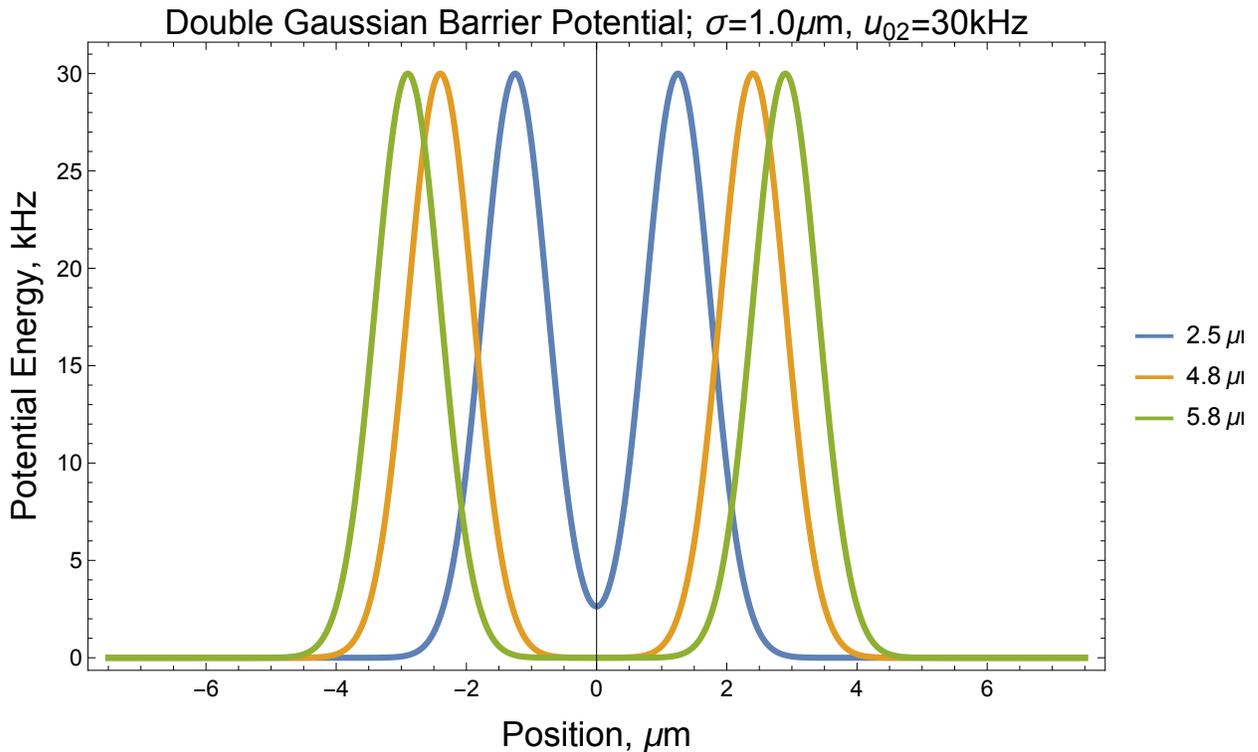
```
In[290]:= t = AbsoluteTime[];
eigenEnergyEigs30sig2sep5 =
  EigenEnergy[potential[[2, 5, 1, All]], energyGoodEigs[[2, 5, 1]], 1];
Dimensions[eigenEnergyEigs30sig2sep5]
eigenEnergyEigs30sig1 = Table[EigenEnergy[
  potential[[1, k, 1, All]], energyGoodEigs[[1, k, 1]], 1], {k, 3, kL, 2}];
Dimensions[eigenEnergyEigs30sig1]
AbsoluteTime[] - t
```

```
Out[292]= {1563}
```

```
Out[294]= {2}
```

```
Out[295]= 2213.023531
```

```
varPot30sig1 =
  ListLinePlot[{Transpose@{x/10^-6, potential[[1, 2, 1, All]]/(2*π*ħ*10^3)},
    Transpose@{x/10^-6, potential[[1, 3, 1, All]]/(2*π*ħ*10^3)},
    (*Transpose@{x/10^-6, potential[[1, 4, 1, All]]/(2*π*ħ*10^3)}, *)
    Transpose@{x/10^-6, potential[[1, 5, 1, All]]/(2*π*ħ*10^3)}},
  FrameLabel → {Style["Position, μm", 18], Style["Potential Energy, kHz", 18]},
  PlotLabel → Style["Double Gaussian Barrier Potential; σ=1.0μm, u02=30kHz", 18],
  PlotLegends → {"μm" xsep[[2]]/10^-6, "μm" xsep[[3]]/10^-6,
    (*"xsep" xsep[[4]]/10^-6, *) "μm" xsep[[5]]/10^-6}, FrameTicksStyle → Medium,
  ImageSize → Large, PlotStyle → {Thickness[.006]}, Frame → True]
```

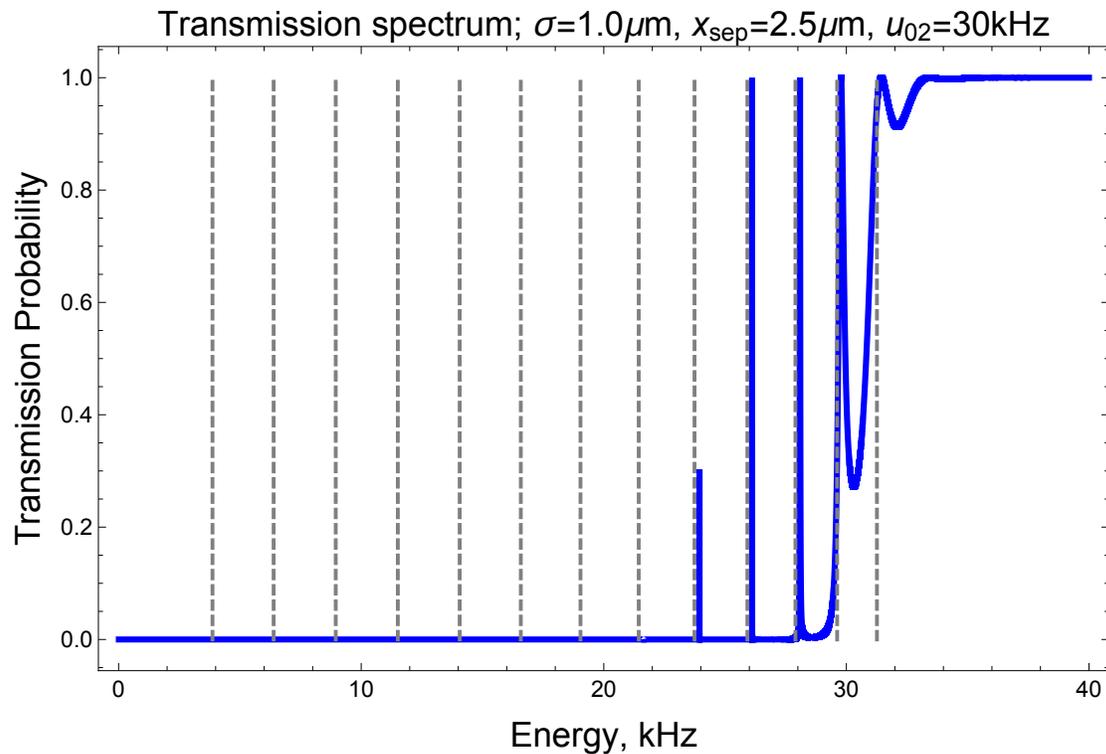


```
Export [
  "/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis
  nb's/possible figs/varPot30sig1.jpeg", varPot30sig1];
```

```
varsEigen30sig1[[1]][[13]]
```

```
31.26
```

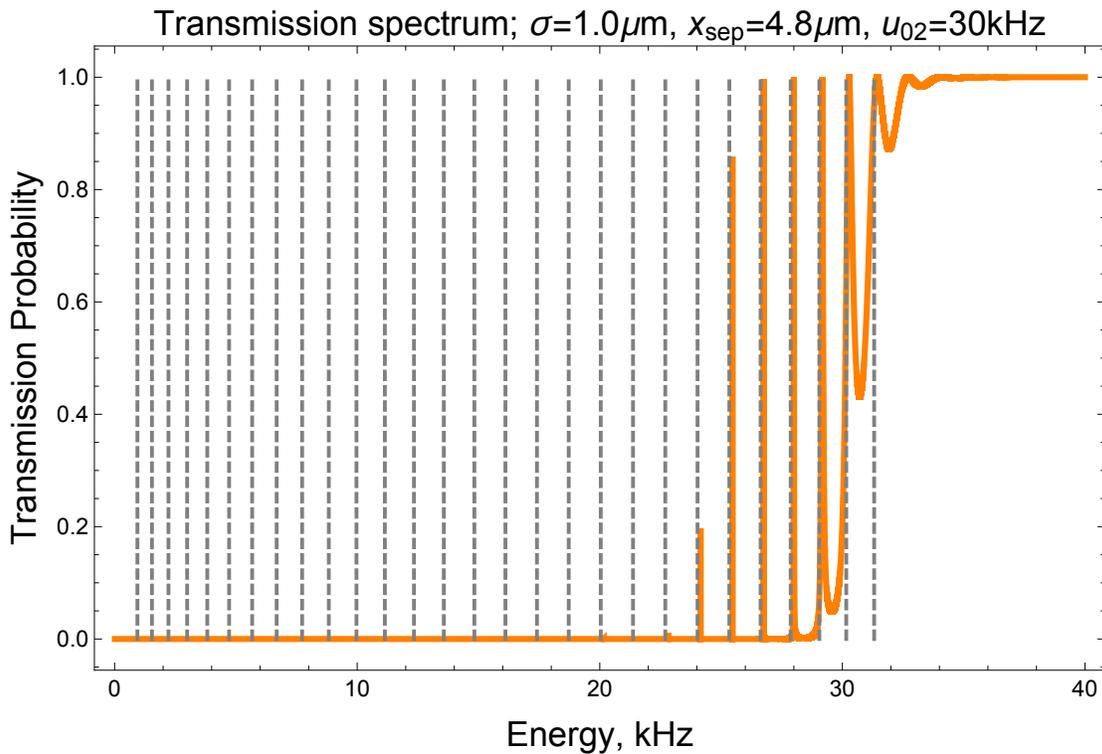
```
varTrans30sig1sep2 = ListLinePlot[
  {Transpose@{energyGood[[1, 2, 1]]/(2 *  $\pi$  *  $\hbar$  * 1.0*^3), 1 - reflectCoef[[1, 2, 1]]},
    (*Transpose@{energyGood[[1, 3, 1]]/(2* $\pi$ * $\hbar$ *1.0*^3), 1-reflectCoef[[1, 3, 1]]},
    Transpose@{energyGood[[1, 5, 1]]/(2* $\pi$ * $\hbar$ *1.0*^3), 1-reflectCoef[[1, 5, 1]]}*},
  PlotRange -> All(*{{20, 38}, {-0.01, 1.005}}*), GridLines -> {{}, {}},
  FrameLabel -> {Style[(*"Energy/v0")"Energy, kHz", 18],
    Style["Transmission Probability", 18]},
  PlotLabel -> Style["Transmission spectrum;  $\sigma=1.0\mu\text{m}$ ,  $x_{\text{sep}}=2.5\mu\text{m}$ ,  $u_{02}=30\text{kHz}$ ", 18],
  FrameTicksStyle -> Medium, ImageSize -> Large, Frame -> True,
  PlotStyle -> Directive[Blue, Thickness[.006]], Epilog ->
    {Directive[{Thick, Gray, Dashed}], Table[Line[{{varsEigen30sig1[[1]][[i]], 0},
      {varsEigen30sig1[[1]][[i]], 1}}, {i, 1, 13}]}
```



```
Export[
```

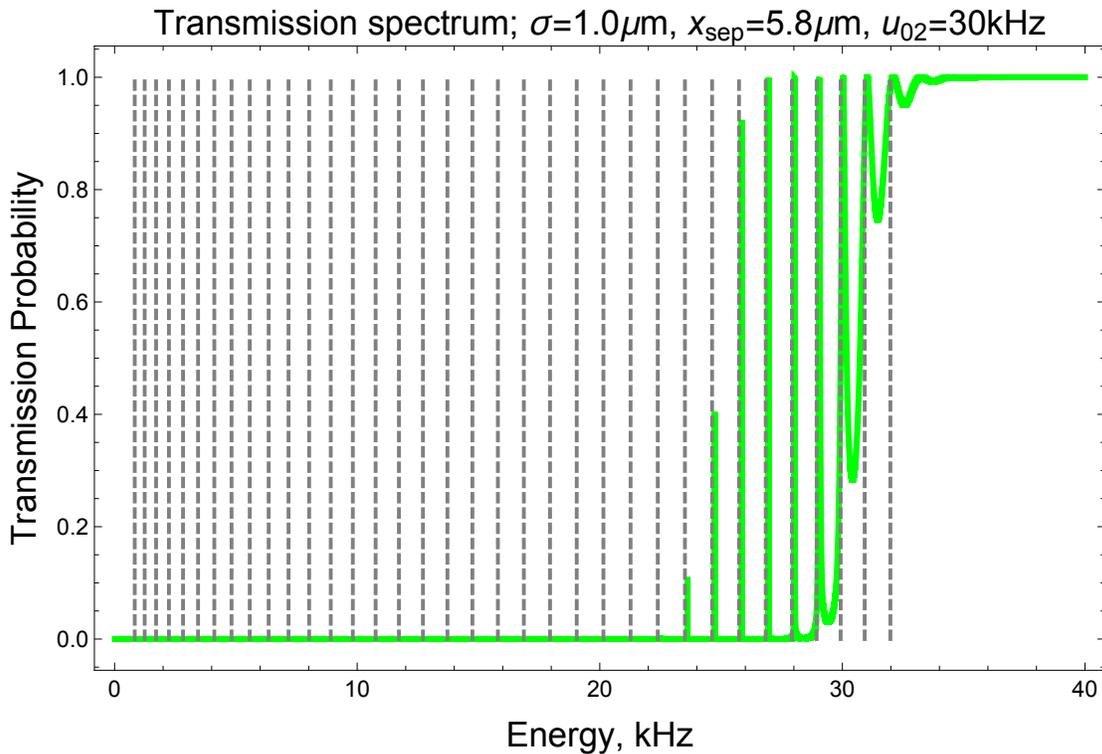
```
  "/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis
  nb's/possible figs/varTrans30sig1sep2.jpeg", varTrans30sig1sep2];
```

```
varTrans30sig1sep3 = ListLinePlot[
  {Transpose@{energyGood[[1, 3, 1]]/(2 * π * ħ * 1.0*^3), 1 - reflectCoef[[1, 3, 1]]}
  (*, Transpose@{energyGood[[1, 5, 1]]/(2 * π * ħ * 1.0*^3), 1 - reflectCoef[[1, 5, 1]]}*)},
  PlotRange → All(*{{20, 38}, {-0.01, 1.005}}*), GridLines → {{}, {}},
  FrameLabel → {Style[(*"Energy/v0"*)"Energy, kHz", 18],
  Style["Transmission Probability", 18]},
  PlotLabel → Style["Transmission spectrum; σ=1.0μm, xsep=4.8μm, u02=30kHz", 18],
  (*PlotLegends→{ "=xsep"xsep[[3]]/10^-6(*, "=xsep"xsep[[5]]/10^-6*}, *)
  FrameTicksStyle → Medium, ImageSize → Large, Frame → True,
  PlotStyle → Directive[Orange, Thickness[.006]],
  Epilog → {Directive[{Thick, Gray, Dashed}],
  Table[Line[{{varsEigen30sig1[[2]][[i]], 0}, {varsEigen30sig1[[2]][[i]], 1}}],
  {i, 1, Length[varsEigen30sig1[[2]]}]}]}]
```



```
Export[
  "/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis
  nb's/possible figs/varTrans30sig1sep3.jpeg", varTrans30sig1sep3];
```

```
varTrans30sig1sep5 = ListLinePlot[
  Transpose@{energyGood[[1, 5, 1]] / (2 *  $\pi$  *  $\hbar$  * 1.0*^3), 1 - reflectCoef[[1, 5, 1]]},
  PlotRange -> All (*{{20, 38}, {-0.01, 1.005}}*), GridLines -> {{}, {}},
  FrameLabel -> {Style[(*"Energy/v0"*)"Energy, kHz", 18],
    Style["Transmission Probability", 18]},
  PlotLabel -> Style["Transmission spectrum;  $\sigma=1.0\mu\text{m}$ ,  $x_{\text{sep}}=5.8\mu\text{m}$ ,  $u_{02}=30\text{kHz}$ ", 18],
  FrameTicksStyle -> Medium, ImageSize -> Large, Frame -> True,
  PlotStyle -> Directive[Green, Thickness[.006]],
  Epilog -> {Directive[{Thick, Gray, Dashed}],
    Table[Line[{{varsEigen30sig1[[3]][[i]], 0}, {varsEigen30sig1[[3]][[i]], 1}}],
      {i, 1, Length[varsEigen30sig1[[3]]}]}]}]
```



```
Export[
  "/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis
  nb's/possible figs/varTrans30sig1sep5.jpeg", varTrans30sig1sep5];
```

```
In[360]= varsEigen30sig1 =
  {Round[Select[DeleteDuplicates[eigenEnergy[[1, 2, 1]] / (2 * π * ħ * 1.0*^3),
    Abs[#1 - #2] < .1 &], # ≤ 32 &], .01],
  Round[Select[DeleteDuplicates[eigenEnergyEigs30sig1[[1]] / (2 * π * ħ * 1.0*^3),
    Abs[#1 - #2] < .1 &], # ≤ 32 &], .01],
  Round[Select[DeleteDuplicates[eigenEnergyEigs30sig1[[2]] / (2 * π * ħ * 1.0*^3),
    Abs[#1 - #2] < .1 &], # ≤ 32 &], .01]}
Dimensions[varsEigen30sig1]
Length[varsEigen30sig1[[1]]]
Length[varsEigen30sig1[[2]]]
Length[varsEigen30sig1[[3]]]
eigenDiff30sig1sepk = {Join[Differences[varsEigen30sig1[[1]]], {0}],
  Join[Differences[varsEigen30sig1[[2]]], {0}],
  Join[Differences[varsEigen30sig1[[3]]], {0}]}
Table[Dimensions[eigenDiff30sig1sepk[[i]]], {i, 1, 3}]
```

```
Out[360]= {{3.87, 6.39, 8.95, 11.51, 14.06, 16.58, 19.04, 21.44, 23.74,
  25.92, 27.9, 29.62, 31.26}, {0.94, 1.54, 2.22, 2.99, 3.82, 4.72, 5.67,
  6.68, 7.73, 8.83, 9.97, 11.14, 12.34, 13.57, 14.83, 16.11, 17.41, 18.72,
  20.04, 21.37, 22.7, 24.03, 25.34, 26.63, 27.87, 29.05, 30.16, 31.31},
  {0.83, 1.24, 1.71, 2.24, 2.82, 3.44, 4.11, 4.82, 5.57, 6.35, 7.17, 8.02, 8.91,
  9.82, 10.76, 11.72, 12.71, 13.72, 14.75, 15.8, 16.87, 17.95, 19.05, 20.15,
  21.27, 22.39, 23.51, 24.63, 25.74, 26.84, 27.92, 28.94, 29.93, 30.92, 31.98}}
```

```
Out[361]= {3}
```

```
Out[362]= 13
```

```
Out[363]= 28
```

```
Out[364]= 35
```

```
Out[365]= {{2.52, 2.56, 2.56, 2.55, 2.52, 2.46, 2.4, 2.3, 2.18, 1.98, 1.72, 1.64, 0},
  {0.6, 0.68, 0.77, 0.83, 0.9, 0.95, 1.01, 1.05, 1.1, 1.14, 1.17, 1.2, 1.23, 1.26,
  1.28, 1.3, 1.31, 1.32, 1.33, 1.33, 1.33, 1.31, 1.29, 1.24, 1.18, 1.11, 1.15, 0},
  {0.41, 0.47, 0.53, 0.58, 0.62, 0.67, 0.71, 0.75, 0.78, 0.82, 0.85,
  0.89, 0.91, 0.94, 0.96, 0.99, 1.01, 1.03, 1.05, 1.07, 1.08, 1.1, 1.1,
  1.12, 1.12, 1.12, 1.12, 1.11, 1.1, 1.08, 1.02, 0.99, 0.99, 1.06, 0}}
```

```
Out[366]= {{13}, {28}, {35}}
```

```
In[306]= varsEigen30sig1sep2Table =
  Grid[{"Eigen-energies for a double Gaussian defined by σ=1.0μm, xsep=2.5μm",
  SpanFromLeft}, Join[{"30kHz"}, varsEigen30sig1[[1]]],
  Join[{"Δ", ""}, Round[Differences[varsEigen30sig1[[1]]], .01]}],
  Frame → All, Background → {{LightBlue, None}, {None, None}}
```

Eigen-energies for a double Gaussian defined by $\sigma=1.0\mu\text{m}$ , $x_{\text{sep}}=2.5\mu\text{m}$													
30kHz	3.87	6.39	8.95	11.51	14.06	16.58	19.04	21.44	23.74	25.92	27.9	29.62	31.26
H <sub>z</sub>				51	06	58	04	44	74	92		62	26
Δ		2.52	2.56	2.56	2.55	2.52	2.46	2.4	2.3	2.18	1.98	1.72	1.64

```
Out[306]=
```

```
In[307]:= Export["/Users/avaashby/Desktop/masters-thesis/Mathematica
Code/ImpedanceMethod/thesis nb's/possible
figs/varsEigen30sig1sep2Table.jpeg", varsEigen30sig1sep2Table];
```

```
In[308]:= varsEigen30sig1sep3Table =
Grid[{"Eigen-energies for a double Gaussian defined by  $\sigma=1.0\mu\text{m}$ ,  $x_{\text{sep}}=4.8\mu\text{m}$ ",
SpanFromLeft}, Join[{"30kHz"}, varsEigen30sig1[[2]]],
Join[{" $\Delta$ ", ""}, Round[Differences[varsEigen30sig1[[2]]], .01]]],
Frame  $\rightarrow$  All, Background  $\rightarrow$  {{LightOrange, None}, {None, None}}, ItemSize  $\rightarrow$  Full]
```

Eigen-energies for a double G														
30kHz	0.94	1.54	2.22	2.99	3.82	4.72	5.67	6.68	7.73	8.83	9.97	11.14	12.34	13.5
$\Delta$		0.6	0.68	0.77	0.83	0.9	0.95	1.01	1.05	1.1	1.14	1.17	1.2	1.2

```
In[310]:= Export["/Users/avaashby/Desktop/masters-thesis/Mathematica
Code/ImpedanceMethod/thesis nb's/possible
figs/varsEigen30sig1sep3Table.jpeg", varsEigen30sig1sep3Table];
```

```
In[309]:= varsEigen30sig1sep5Table =
Grid[{"Eigen-energies for a double Gaussian defined by  $\sigma=1.0\mu\text{m}$ ,  $x_{\text{sep}}=5.8\mu\text{m}$ ",
SpanFromLeft}, Join[{"30kHz"}, varsEigen30sig1[[3]]],
Join[{" $\Delta$ ", ""}, Round[Differences[varsEigen30sig1[[3]]], .01]]],
Frame  $\rightarrow$  All, Background  $\rightarrow$  {{LightGreen, None}, {None, None}}, ItemSize  $\rightarrow$  Full]
```

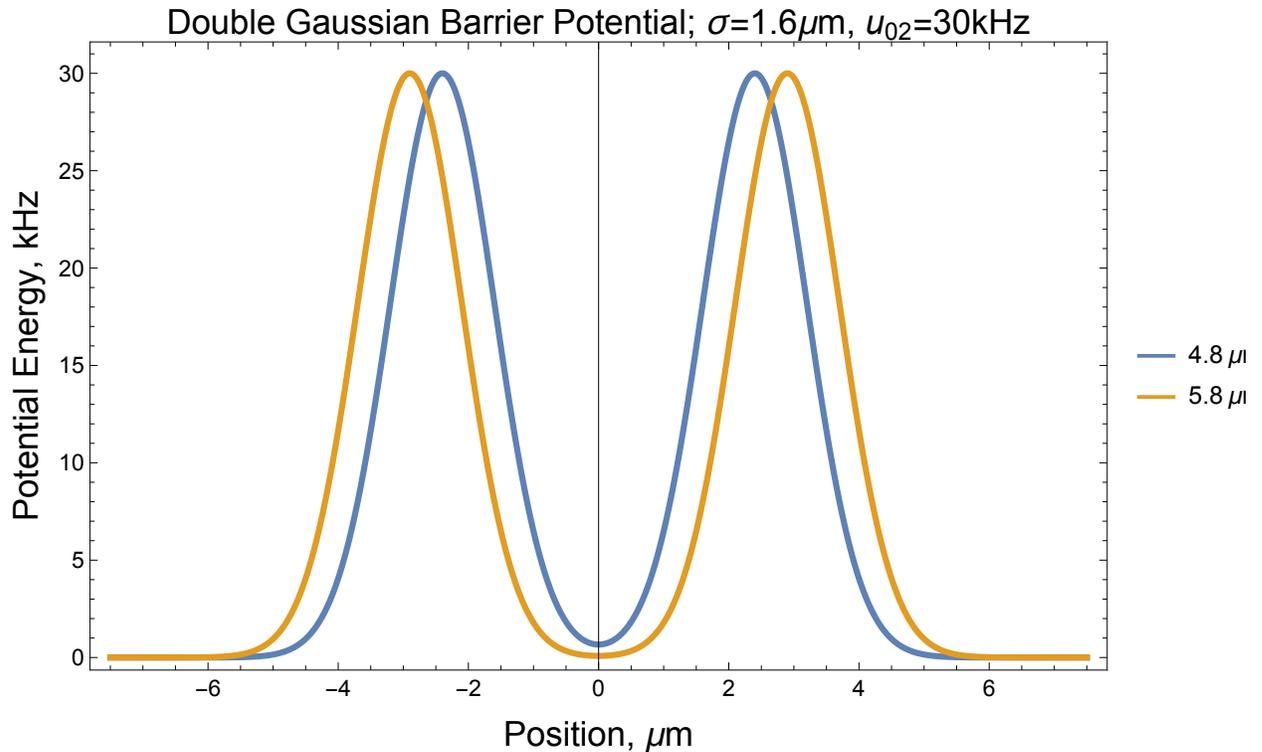
Eigen-en														
30kHz	0.83	1.24	1.71	2.24	2.82	3.44	4.11	4.82	5.57	6.35	7.17	8.02	8.91	9.82
$\Delta$		0.41	0.47	0.53	0.58	0.62	0.67	0.71	0.75	0.78	0.82	0.85	0.89	0.91

```
In[311]:= Export["/Users/avaashby/Desktop/masters-thesis/Mathematica
Code/ImpedanceMethod/thesis nb's/possible
figs/varsEigen30sig1sep5Table.jpeg", varsEigen30sig1sep5Table];
```

```

varPot30sig2 =
ListLinePlot[{Transpose@{x/10^-6, potential[[2, 3, 1, All]]/(2 * π * ħ * 10^3)},
  Transpose@{x/10^-6, potential[[2, 5, 1, All]]/(2 * π * ħ * 10^3)}},
FrameLabel → {Style["Position, μm", 18], Style["Potential Energy, kHz", 18]},
PlotLabel → Style[
  "Double Gaussian Barrier Potential; σ=1.6μm, u02=30kHz" (*, xsep=4.8 μm*), 18],
PlotLegends → {"μm" xsep[[3]]/10^-6, "μm" xsep[[5]]/10^-6},
FrameTicksStyle → Medium, ImageSize → Large,
PlotStyle → {Thickness[.006]}, Frame → True]

```

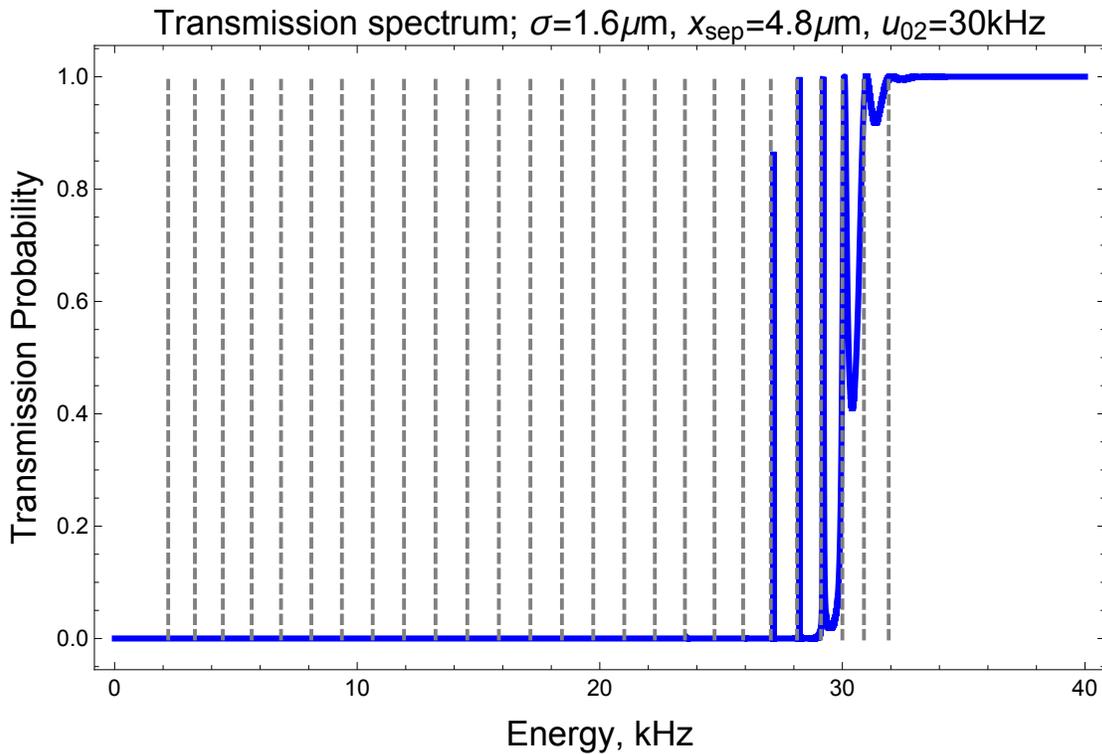


```

Export[
"/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis
nb's/possible figs/varPot30sig2.jpeg", varPot30sig2];

```

```
varTrans30sig2sep3 = ListLinePlot[
  Transpose@{energyGood[[2, 3, 1]] / (2 *  $\pi$  *  $\hbar$  * 1.0*^3), 1 - reflectCoef[[2, 3, 1]]},
  PlotRange -> All (*{{20, 38}, {-0.01, 1.005}}*), GridLines -> {{}, {}},
  FrameLabel -> {Style[(*"Energy/v0"*) "Energy, kHz", 18],
    Style["Transmission Probability", 18]},
  PlotLabel -> Style["Transmission spectrum;  $\sigma=1.6\mu\text{m}$ ,  $x_{\text{sep}}=4.8\mu\text{m}$ ,  $u_{02}=30\text{kHz}$ ", 18]
  (*, PlotLegends -> { "=x_sep"xsep[[3]]/10^-6*}, FrameTicksStyle -> Medium,
  ImageSize -> Large, Frame -> True, PlotStyle -> Directive[Blue, Thickness[.006]],
  Epilog -> {Directive[{Thick, Gray, Dashed}],
    Table[Line[{{varsEigen30sig2[[1]][[i]], 0}, {varsEigen30sig2[[1]][[i]], 1}],
      {i, 1, Length[varsEigen30sig2[[1]]}]}]
```

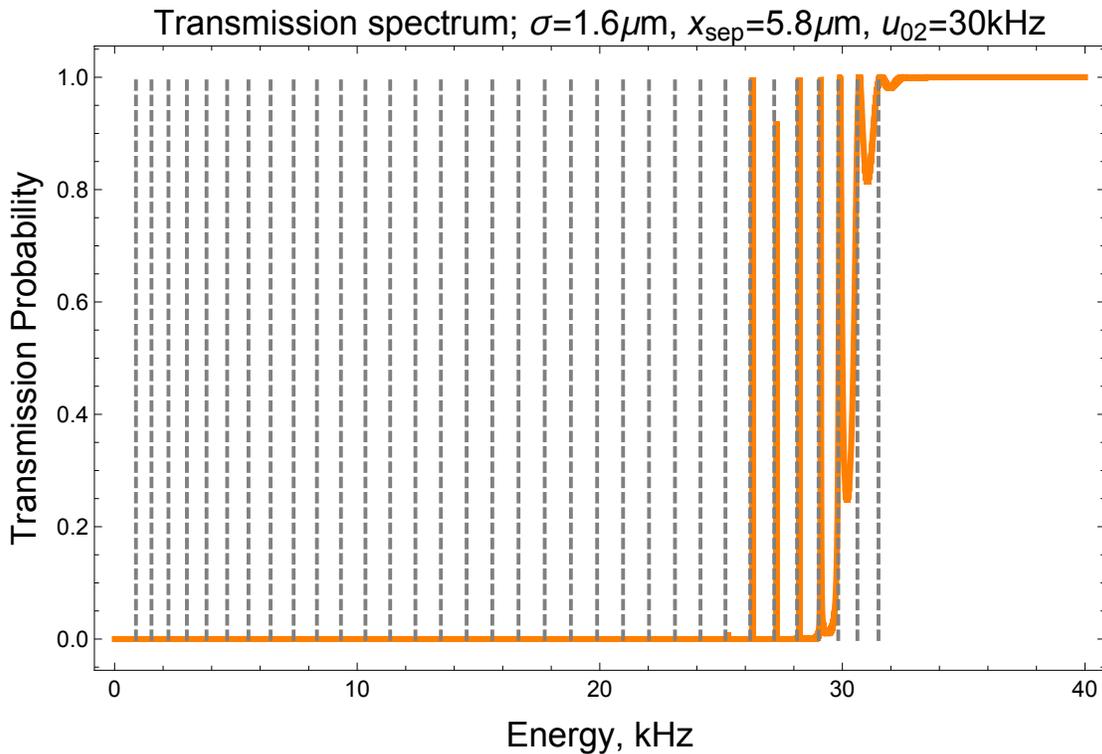


```
Export[
  "/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis
  nb's/possible figs/varTrans30sig2sep3.jpeg", varTrans30sig2sep3];
```

```

varTrans30sig2sep5 = ListLinePlot[
  Transpose@{energyGood[[2, 5, 1]] / (2 *  $\pi$  *  $\hbar$  * 1.0*^3), 1 - reflectCoef[[2, 5, 1]]},
  PlotRange -> All (*{{20, 38}, {-0.01, 1.005}}*), GridLines -> {{}, {}},
  FrameLabel -> {Style[(*"Energy/v0"*)"Energy, kHz", 18],
    Style["Transmission Probability", 18]},
  PlotLabel -> Style["Transmission spectrum;  $\sigma=1.6\mu\text{m}$ ,  $x_{\text{sep}}=5.8\mu\text{m}$ ,  $u_{02}=30\text{kHz}$ ", 18],
  FrameTicksStyle -> Medium, ImageSize -> Large, Frame -> True,
  PlotStyle -> Directive[Orange, Thickness[.006]],
  Epilog -> {Directive[{Thick, Gray, Dashed}]},
  Table[Line[{{varsEigen30sig2[[2]][[i]], 0}, {varsEigen30sig2[[2]][[i]], 1}},
    {i, 1, Length[varsEigen30sig2[[2]]}]]]

```



```

Export[
  "/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis
  nb's/possible figs/varTrans30sig2sep5.jpeg", varTrans30sig2sep5];

```

```
In[368]= varsEigen30sig2 =
  {Round[Select[DeleteDuplicates[eigenEnergy[[2, 3, 1]] / (2 * π * ħ * 1.0*^3),
    Abs[#1 - #2] < .1 &], # ≤ 32 &], .01],
  Round[Select[DeleteDuplicates[eigenEnergyEigs30sig2sep5 / (2 * π * ħ * 1.0*^3),
    Abs[#1 - #2] < .1 &], # ≤ 32 &], .01]}
Dimensions[varsEigen30sig2]
Length[varsEigen30sig2[[1]]]
Length[varsEigen30sig2[[2]]]
diffEigen30sig2 = {Join[Differences[varsEigen30sig2[[1]]], {0}],
  Join[Differences[varsEigen30sig2[[2]]], {0}]}
```

```
Out[368]= {{1.17, 2.21, 3.31, 4.46, 5.65, 6.86, 8.11, 9.37, 10.64,
  11.93, 13.23, 14.54, 15.84, 17.14, 18.44, 19.73, 21.01, 22.27,
  23.51, 24.73, 25.91, 27.05, 28.13, 29.12, 30., 30.89, 31.91}},
  {{0.88, 1.52, 2.22, 2.98, 3.79, 4.64, 5.52, 6.43, 7.38, 8.34, 9.33, 10.34,
  11.36, 12.4, 13.45, 14.51, 15.57, 16.65, 17.73, 18.81, 19.89, 20.96,
  22.03, 23.1, 24.15, 25.18, 26.2, 27.19, 28.13, 29.02, 29.83, 30.62, 31.49}}
```

```
Out[369]= {2}
```

```
Out[370]= 27
```

```
Out[371]= 33
```

```
Out[372]= {{1.04, 1.1, 1.15, 1.19, 1.21, 1.25, 1.26, 1.27, 1.29, 1.3, 1.31, 1.3, 1.3, 1.3,
  1.29, 1.28, 1.26, 1.24, 1.22, 1.18, 1.14, 1.08, 0.99, 0.88, 0.89, 1.02, 0},
  {{0.64, 0.7, 0.76, 0.81, 0.85, 0.88, 0.91, 0.95, 0.96, 0.99, 1.01,
  1.02, 1.04, 1.05, 1.06, 1.06, 1.08, 1.08, 1.08, 1.08, 1.07, 1.07,
  1.07, 1.05, 1.03, 1.02, 0.99, 0.94, 0.89, 0.81, 0.79, 0.87, 0}}}
```

```
In[323]= varsEigen30sig2sep3Table =
  Grid[{"Eigen-energies for a double Gaussian defined by  $\sigma=1.6\mu\text{m}$ ,  $x_{\text{sep}}=4.8\mu\text{m}$ ",
  SpanFromLeft}, Join[{"30kHz"}, varsEigen30sig2[[1]]],
  Join[{"Δ", ""}, Round[Differences[varsEigen30sig2[[1]]], .01]]],
  Frame → All, Background → {{LightBlue, None}, {None, None}}, ItemSize → All]
```

Eigen-energies for a double Gaussian defined by $\sigma=1.6\mu\text{m}$ , $x_{\text{sep}}=4.8\mu\text{m}$												
30kHz	1.17	2.21	3.31	4.46	5.65	6.86	8.11	9.37	10.64	11.93	13.23	14.54
Δ		1.04	1.1	1.15	1.19	1.21	1.25	1.26	1.27	1.29	1.3	1.31

```
In[324]= Export["/Users/avaashby/Desktop/masters-thesis/Mathematica
Code/ImpedanceMethod/thesis nb's/possible
figs/varsEigen30sig2sep3Table.jpeg", varsEigen30sig2sep3Table];
```

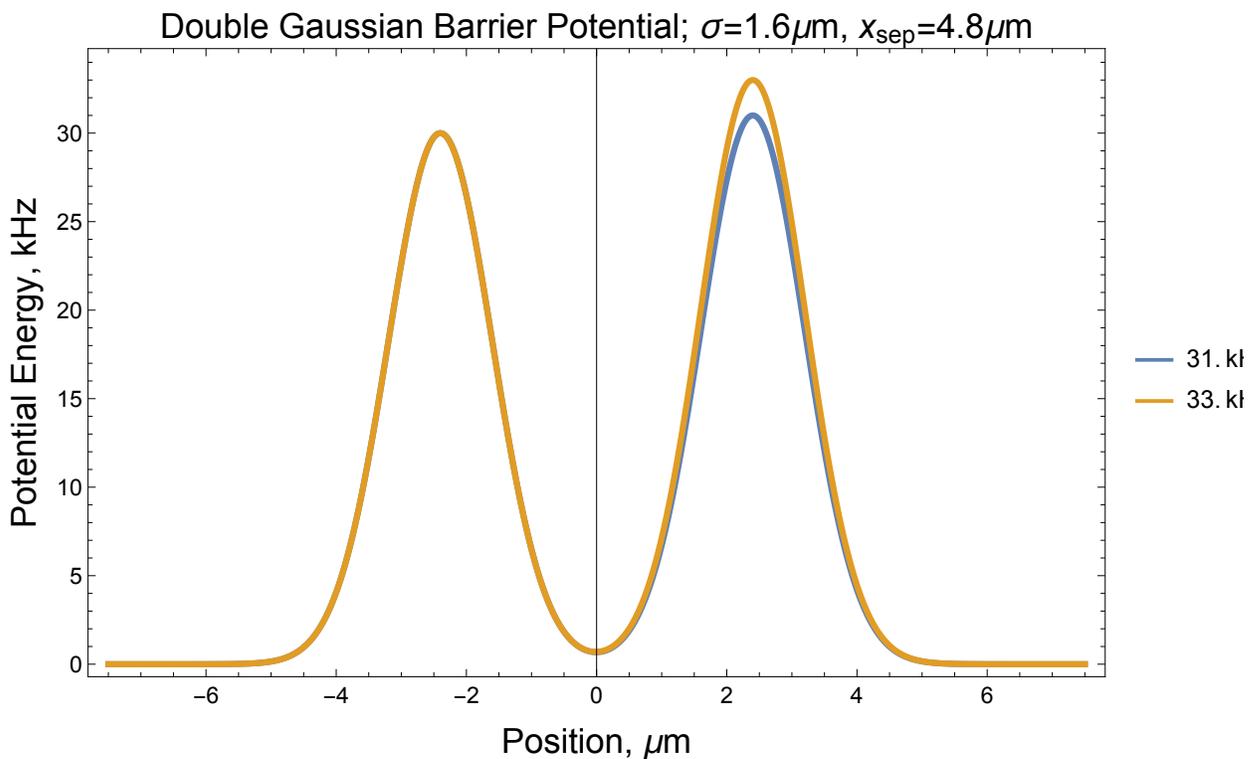
```
In[325]= varsEigen30sig2sep5Table =
  Grid[{"Eigen-energies for a double Gaussian defined by  $\sigma=1.6\mu\text{m}$ ,  $x_{\text{sep}}=5.8\mu\text{m}$ ",
  SpanFromLeft}, Join[{"30kHz"}, varsEigen30sig2[[2]]],
  Join[{"Δ", ""}, Round[Differences[varsEigen30sig2[[2]]], .01]]],
  Frame → All, Background → {{LightOrange, None}, {None, None}}, ItemSize → All]
```

Eigen-energies for a double Gaussian defined by $\sigma=1.6\mu\text{m}$ , $x_{\text{sep}}=5.8\mu\text{m}$												
30kHz	0.88	1.52	2.22	2.98	3.79	4.64	5.52	6.43	7.38	8.34	9.33	10.34
Δ		0.64	0.7	0.76	0.81	0.85	0.88	0.91	0.95	0.96	0.99	1.01

```
In[326]:= Export["/Users/avaashby/Desktop/masters-thesis/Mathematica
Code/ImpedanceMethod/thesis nb's/possible
figs/varsEigen30sig2sep5Table.jpeg", varsEigen30sig2sep5Table];
```

```
varPotsig2h31h33 =
```

```
ListLinePlot[{Transpose@{x/10^-6, potential[[2, 3, 2, All]]/(2 * π * ħ * 10^3)},
Transpose@{x/10^-6, potential[[2, 3, 4, All]]/(2 * π * ħ * 10^3)}
(*, Transpose@{x/10^-6, potential[[1, 4, 1, All]]/(2 * π * ħ * 10^3)},
Transpose@{x/10^-6, potential[[1, 5, 1, All]]/(2 * π * ħ * 10^3)}*)},
FrameLabel → {Style["Position, μm", 18], Style["Potential Energy, kHz", 18]},
PlotLabel → Style["Double Gaussian Barrier Potential; σ=1.6 μm, xsep=4.8 μm", 18],
PlotLegends → {"kHz" u02[[2]]/(2 * π * ħ * 10^3), "kHz" u02[[4]]/(2 * π * ħ * 10^3)},
FrameTicksStyle → Medium, ImageSize → Large,
PlotStyle → {Thickness[.006]}, Frame → True]
```

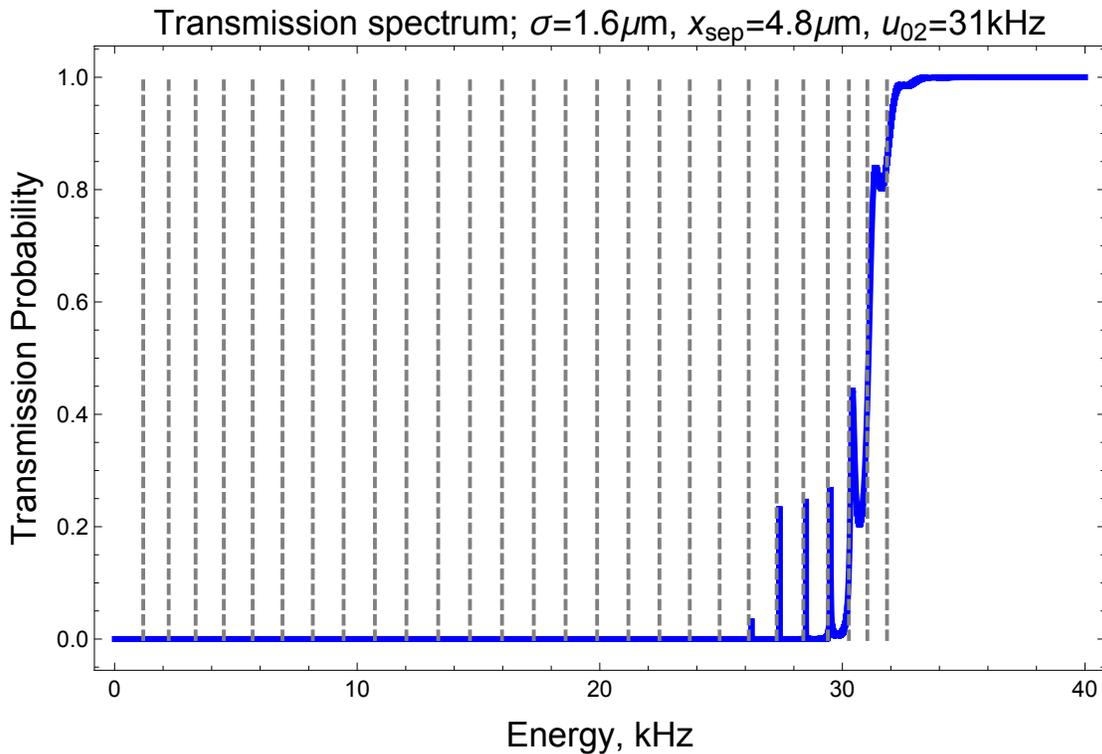


```
Export [
"/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis
nb's/possible figs/varPotsig2h31h33.jpeg", varPotsig2h31h33];
```

```

varTrans31sig2sep3 = ListLinePlot[
  Transpose@{energyGood[[2, 3, 2]] / (2 *  $\pi$  *  $\hbar$  * 1.0*^3), 1 - reflectCoef[[2, 3, 2]]},
  PlotRange -> All (*{{20, 38}, {-0.01, 1.005}}*), GridLines -> {{}, {}},
  FrameLabel -> {Style[(*"Energy/v0"*)"Energy, kHz", 18],
    Style["Transmission Probability", 18]},
  PlotLabel -> Style["Transmission spectrum;  $\sigma=1.6\mu\text{m}$ ,  $x_{\text{sep}}=4.8\mu\text{m}$ ,  $u_{02}=31\text{kHz}$ ", 18],
  FrameTicksStyle -> Medium, ImageSize -> Large, Frame -> True,
  PlotStyle -> Directive[Blue, Thickness[.006]],
  Epilog -> {Directive[{Thick, Gray, Dashed}], Table[
    Line[{{varsEigensig2sep3[[1]][[i]], 0}, {varsEigensig2sep3[[1]][[i]], 1}},
    {i, 1, Length[varsEigensig2sep3[[1]]}]}]}]

```



```

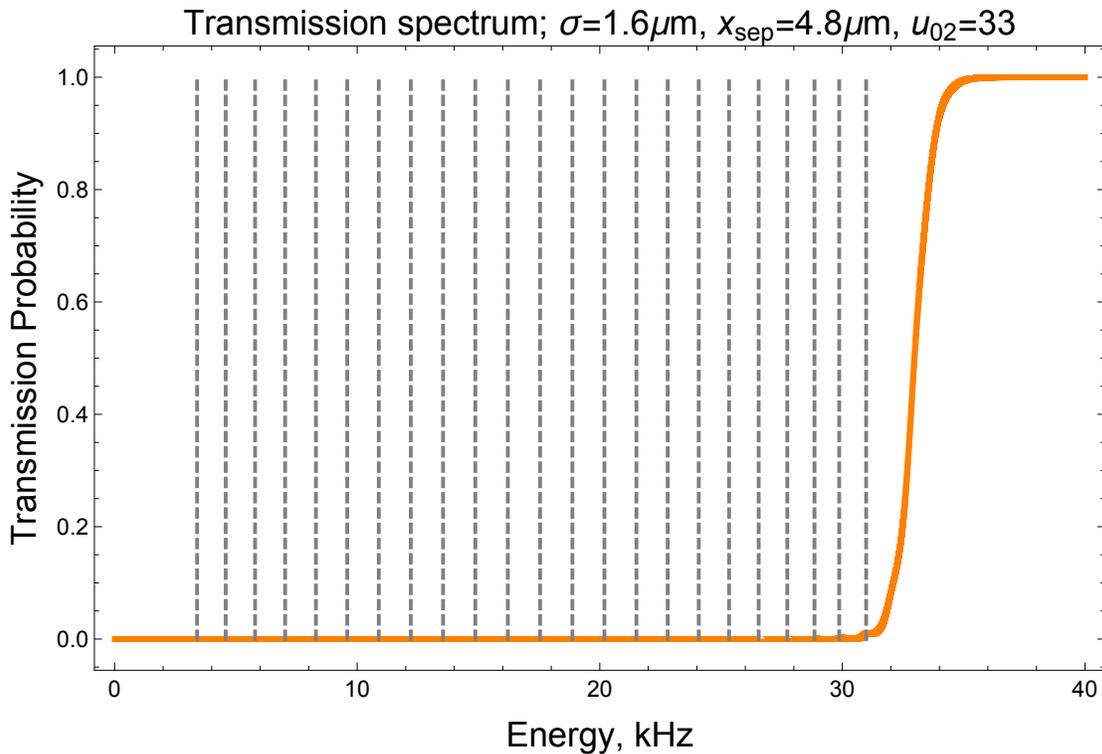
Export[
  "/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis
  nb's/possible figs/varTrans31sig2sep3.jpeg", varTrans31sig2sep3];

```

```

varTrans33sig2sep3 = ListLinePlot[
  Transpose@{energyGood[[2, 3, 4]] / (2 *  $\pi$  *  $\hbar$  * 1.0*^3), 1 - reflectCoef[[2, 3, 4]]},
  PlotRange -> All (*{{20, 38}, {-0.01, 1.005}}*), GridLines -> {{}, {}},
  FrameLabel -> {Style[(*"Energy/v0"*)"Energy, kHz", 18],
    Style["Transmission Probability", 18]},
  PlotLabel -> Style["Transmission spectrum;  $\sigma=1.6\mu\text{m}$ ,  $x_{\text{sep}}=4.8\mu\text{m}$ ,  $u_{02}=33$ ", 18],
  FrameTicksStyle -> Medium, ImageSize -> Large, Frame -> True,
  PlotStyle -> Directive[Orange, Thickness[.006]],
  Epilog -> {Directive[{Thick, Gray, Dashed}], Table[
    Line[{{varsEigensig2sep3[[2]][[i]], 0}, {varsEigensig2sep3[[2]][[i]], 1}},
    {i, 6, Length[varsEigensig2sep3[[2]]]}]}]}

```



```

Export[
  "/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis
  nb's/possible figs/varTrans33sig2sep3.jpeg", varTrans33sig2sep3];

```

```

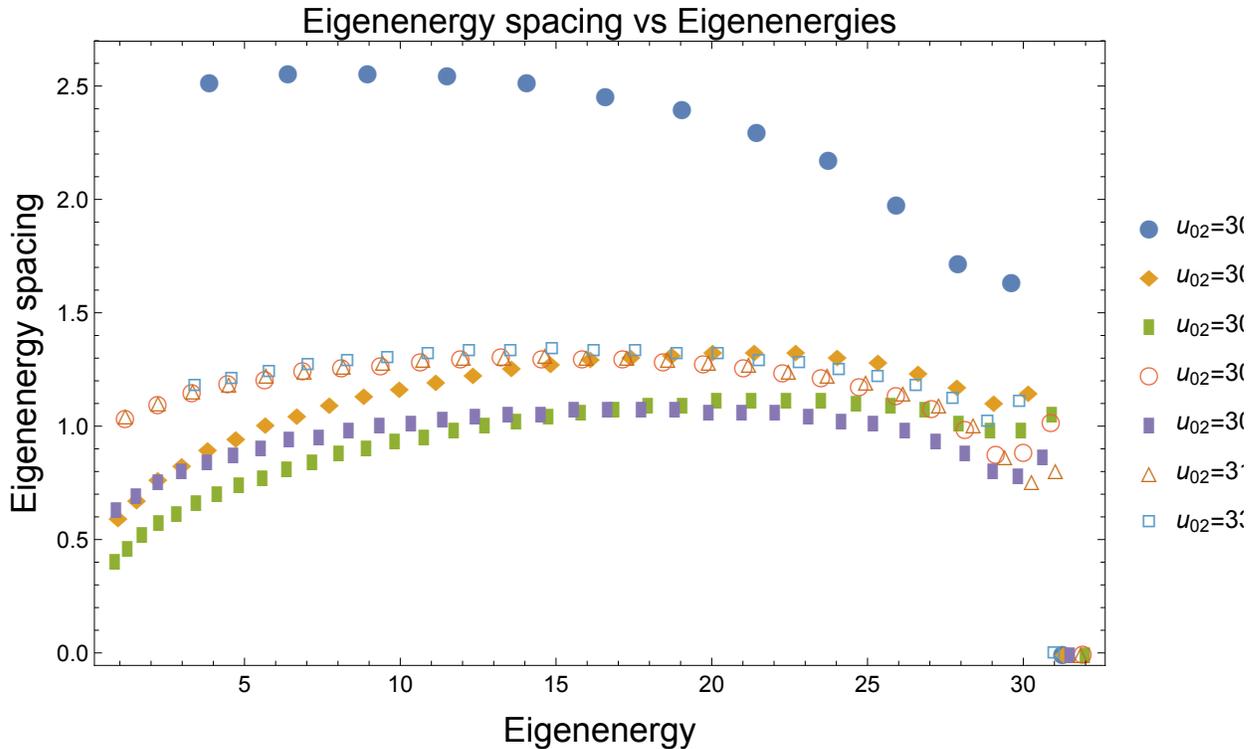
In[376]:= varsEigensig2sep3 =
  {Round[Select[DeleteDuplicates[eigenEnergyEigsig2sep3H31 / (2 * π * ħ * 1.0*^3),
    Abs[#1 - #2] < .4 &], # ≤ 32 &], .01],
  Round[Select[DeleteDuplicates[eigenEnergyEigsig2sep3H[[2]] / (2 * π * ħ * 1.0*^3),
    Abs[#1 - #2] < .1 &], # ≤ 32 &], .01]}
Dimensions[varsEigensig2sep3]
Length[varsEigensig2sep3[[1]]]
Length[varsEigensig2sep3[[2]]]
diffEigensig2sep3 = {Join[Differences[varsEigensig2sep3[[1]]], {0}],
  Join[Differences[varsEigensig2sep3[[2]]], {0}]}
Out[376]= {{1.18, 2.23, 3.34, 4.5, 5.69, 6.92, 8.17, 9.44, 10.73,
  12.03, 13.34, 14.65, 15.97, 17.28, 18.59, 19.89, 21.18, 22.46,
  23.71, 24.94, 26.14, 27.29, 28.39, 29.4, 30.27, 31.03, 31.84},
  {0.5, 0.6, 0.7, 0.8, 0.9, 3.4, 4.58, 5.79, 7.03, 8.3, 9.59, 10.89,
  12.21, 13.54, 14.87, 16.21, 17.54, 18.87, 20.19, 21.51,
  22.8, 24.08, 25.33, 26.55, 27.73, 28.85, 29.87, 30.98}}
Out[377]= {2}
Out[378]= 27
Out[379]= 28
Out[380]= {{1.05, 1.11, 1.16, 1.19, 1.23, 1.25, 1.27, 1.29, 1.3, 1.31, 1.31, 1.32, 1.31,
  1.31, 1.3, 1.29, 1.28, 1.25, 1.23, 1.2, 1.15, 1.1, 1.01, 0.87, 0.76, 0.81, 0},
  {0.1, 0.1, 0.1, 0.1, 2.5, 1.18, 1.21, 1.24, 1.27, 1.29, 1.3, 1.32, 1.33, 1.33,
  1.34, 1.33, 1.33, 1.32, 1.32, 1.29, 1.28, 1.25, 1.22, 1.18, 1.12, 1.02, 1.11, 0}}

```

```

In[394]= eigsVdiffs = ListPlot[{Transpose@{varsEigen30sig1[[1]], eigenDiff30sig1sep3[[1]]},
  Transpose@{varsEigen30sig1[[2]], eigenDiff30sig1sep3[[2]]},
  Transpose@{varsEigen30sig1[[3]], eigenDiff30sig1sep3[[3]]},
  Transpose@{varsEigen30sig2[[1]], diffEigen30sig2[[1]]},
  Transpose@{varsEigen30sig2[[2]], diffEigen30sig2[[2]]},
  Transpose@{varsEigensig2sep3[[1]], diffEigensig2sep3[[1]]},
  Transpose@{varsEigensig2sep3[[2]][[6 ;; 28]],
    diffEigensig2sep3[[2]][[6 ;; 28]]}], PlotRange -> All,
  FrameLabel -> {Style["Eigenenergy", 18], Style["Eigenenergy spacing", 18]},
  PlotLabel -> Style["Eigenenergy spacing vs Eigenenergies", 18],
  FrameTicksStyle -> Medium, ImageSize -> Large, Frame -> True,
  PlotLegends -> {"u02=30kHz, sigma=1.0um, xsep=2.5um",
    "u02=30kHz, sigma=1.0um, xsep=4.8um", "u02=30kHz, sigma=1.0um, xsep=5.8um",
    "u02=30kHz, sigma=1.6um, xsep=4.8um", "u02=30kHz, sigma=1.6um, xsep=5.8um",
    "u02=31kHz, sigma=1.6um, xsep=4.8um", "u02=33kHz, sigma=1.6um, xsep=4.8um"},
  PlotMarkers -> {{Circle, 15}, {Diamond, 15}, {Square, 15}, {Circle, 15}, {Square, 15}, {Triangle, 15}, {Square, 15}}

```



```

In[395]= Export [
  "/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis
  nb's/possible figs/eigsVdiffs.jpeg", eigsVdiffs];

```

```

In[332]= varsEigen31sig2sep3Table =
  Grid[{"Eigen-energies for a double Gaussian defined by sigma=1.6um, xsep=4.8um",
  SpanFromLeft}, Join[{"31kHz"}, varsEigensig2sep3[[1]]],
  Join[{"Delta", ""}, Round[Differences[varsEigensig2sep3[[1]]], .01]]],
  Frame -> All, Background -> {{LightBlue, None}, {None, None}}, ItemSize -> All]

```

		Eigen-energies for a double Gaussian defined by $\sigma=1.6\mu\text{m}$ , $x_{\text{sep}}=4.8\mu\text{m}$											
31kHz		1.18	2.23	3.34	4.5	5.69	6.92	8.17	9.44	10.73	12.03	13.34	14.65
$\Delta$			1.05	1.11	1.16	1.19	1.23	1.25	1.27	1.29	1.3	1.31	1.31

```
In[333]= Export["/Users/avaashby/Desktop/masters-thesis/Mathematica
Code/ImpedanceMethod/thesis nb's/possible
figs/varsEigen31sig2sep3Table.jpeg", varsEigen31sig2sep3Table];
```

```
In[334]= varsEigen33sig2sep3Table =
Grid[{"Eigen-energies for a double Gaussian defined by  $\sigma=1.6\mu\text{m}$ ,  $x_{\text{sep}}=4.8\mu\text{m}$ ",
SpanFromLeft}, Join[{"33kHz"}, varsEigensig2sep3[[2]][[6 ;; 28]]],
Join[{" $\Delta$ ", ""}, Round[Differences[varsEigensig2sep3[[2]][[6 ;; 28]]], .01]]],
Frame  $\rightarrow$  All, Background  $\rightarrow$  {{LightOrange, None}, {None, None}}, ItemSize  $\rightarrow$  All]
```

Eigen-energies for a double Gaussian de												
33kHz	3.4	4.58	5.79	7.03	8.3	9.59	10.89	12.21	13.54	14.87	16.21	17.54
$\Delta$		1.18	1.21	1.24	1.27	1.29	1.3	1.32	1.33	1.33	1.34	1.33

```
In[335]= Export["/Users/avaashby/Desktop/masters-thesis/Mathematica
Code/ImpedanceMethod/thesis nb's/possible
figs/varsEigen33sig2sep3Table.jpeg", varsEigen33sig2sep3Table];
```

## optimization:

```
varsEigen30sig2sep3Table
varsEigen30sig2[[1]]
```

Eigen-energies for a double G												
30kHz	2.21	3.31	4.46	5.65	6.86	8.11	9.37	10.64	11.93	13.23	14.54	15.84

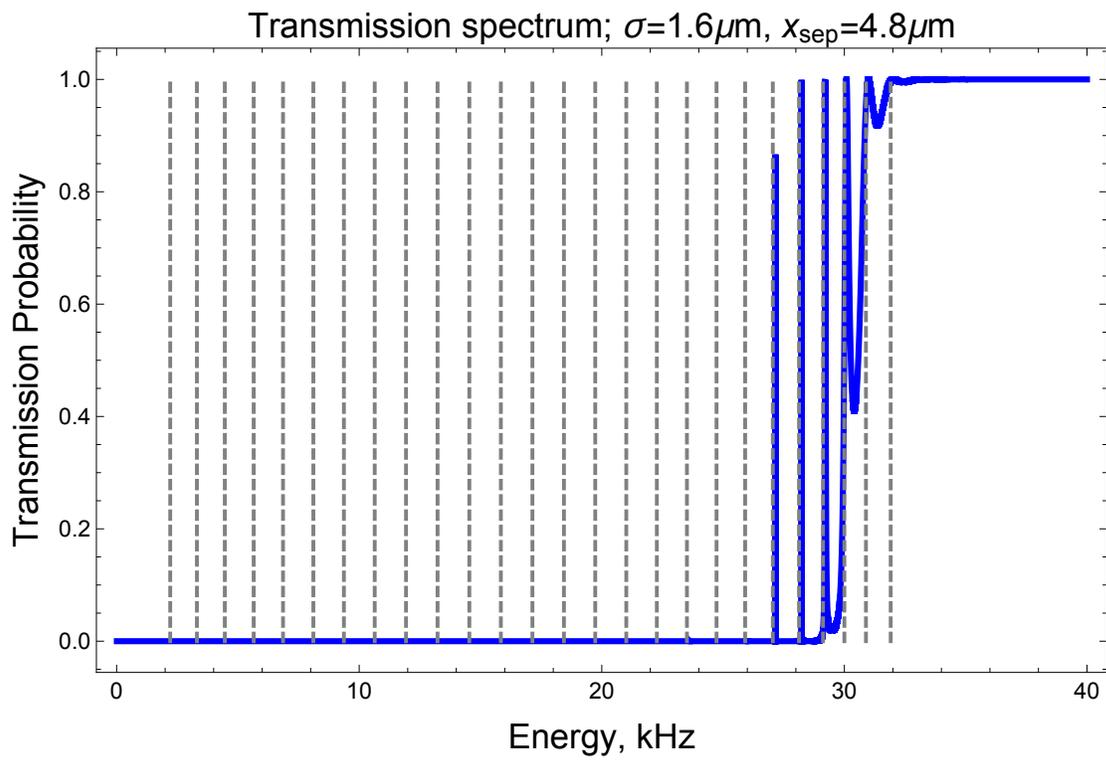
```
{2.21, 3.31, 4.46, 5.65, 6.86, 8.11, 9.37, 10.64,
11.93, 13.23, 14.54, 15.84, 17.14, 18.44, 19.73, 21.01, 22.27,
23.51, 24.73, 25.91, 27.05, 28.13, 29.12, 30., 30.89, 31.91}
```

```

energyBetter30sig2sep3 =
(2 * π * ħ * 1.0 * 10^3) * Join[Table[i, {i, .1, 1.7, .1}], Table[i, {i, 1.7, 2.7, .1}],
  Table[i, {i, 2.7, 2.8, .1}], Table[i, {i, 2.8, 3.8, .1}],
  Table[i, {i, 3.8, 4.0, .1}], Table[i, {i, 4.0, 5.0, .1}],
  Table[i, {i, 5.0, 5.2, .1}], Table[i, {i, 5.2, 6.2, .1}],
  Table[i, {i, 6.2, 6.4, .1}], Table[i, {i, 6.4, 7.4, .1}],
  Table[i, {i, 7.4, 7.6, .1}], Table[i, {i, 7.6, 8.6, .1}], Table[i,
  {i, 8.6, 8.9, .1}], Table[i, {i, 8.9, 9.9, .1}], Table[i, {i, 9.9, 10.1, .1}],
  Table[i, {i, 10.1, 11.1, .1}], Table[i, {i, 11.1, 11.4, .1}],
  Table[i, {i, 11.4, 12.4, .1}], Table[i, {i, 12.4, 12.7, .1}],
  Table[i, {i, 12.7, 13.7, .1}], Table[i, {i, 13.7, 14.0, .1}],
  Table[i, {i, 14.0, 15.0, .1}], Table[i, {i, 15.0, 15.3, .1}],
  Table[i, {i, 15.3, 16.3, .1}], Table[i, {i, 16.3, 16.6, .1}],
  Table[i, {i, 16.6, 17.6, .1}], Table[i, {i, 17.6, 17.9, .1}],
  Table[i, {i, 17.9, 18.9, .1}], Table[i, {i, 18.9, 19.2, .1}],
  Table[i, {i, 19.2, 20.2, .01}], Table[i, {i, 20.2, 20.5, .1}],
  Table[i, {i, 20.5, 21.5, .01}], Table[i, {i, 21.5, 21.7, .1}],
  Table[i, {i, 21.7, 22.7, .001}], Table[i, {i, 22.7, 23.0, .1}],
  Table[i, {i, 23.0, 24.0, .00001}], Table[i, {i, 24.0, 24.4, .001}],
  Table[i, {i, 24.4, 25.0, .0000001}], Table[i, {i, 25.0, 25.4, .001}],
  Table[i, {i, 25.4, 26.4, .000001}], Table[i, {i, 26.4, 26.5, .1}],
  Table[i, {i, 26.5, 27.5, .00001}], Table[i, {i, 27.5, 27.6, .1}],
  Table[i, {i, 27.6, 29.6, .0001}], Table[i, {i, 29.6, 35.0, .01}]];
Dimensions[energyBetter30sig2sep3]
{7 222 786}

reflectCoef30sig2sep3 =
  ReflectionProbability[potential[[2, 3, 1, All]], energyBetter30sig2sep3, 1];
Dimensions[reflectCoef30sig2sep3]
{7 222 786}

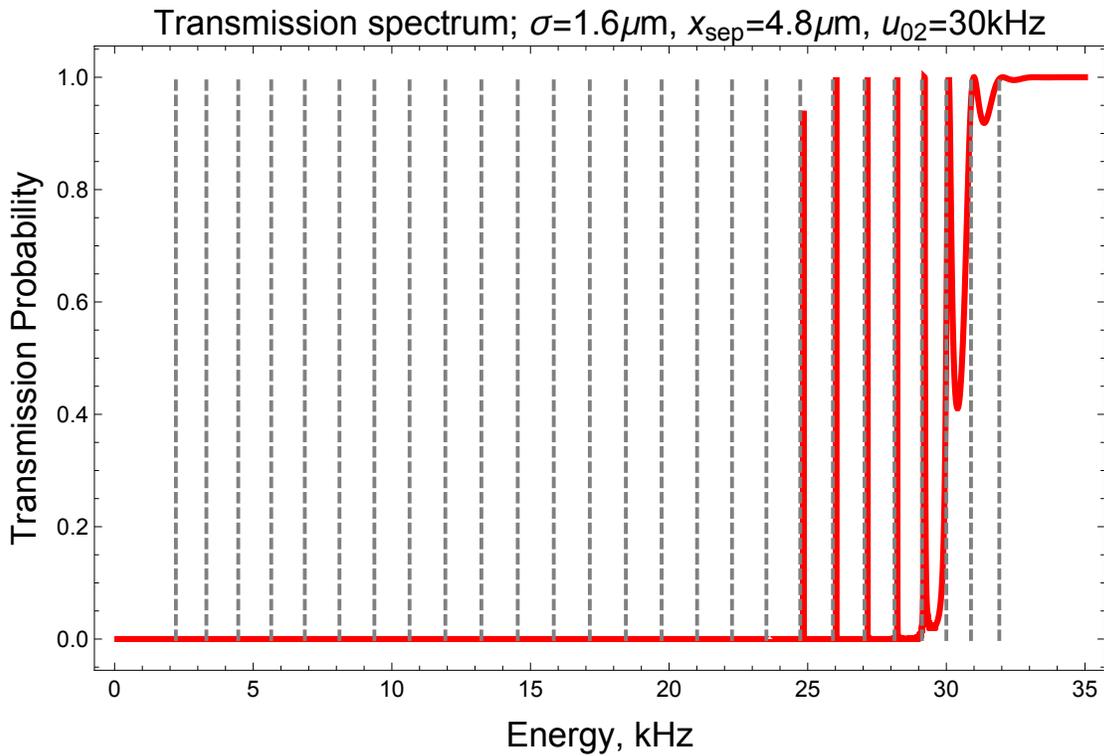
```

`varTrans30sig2sep3`

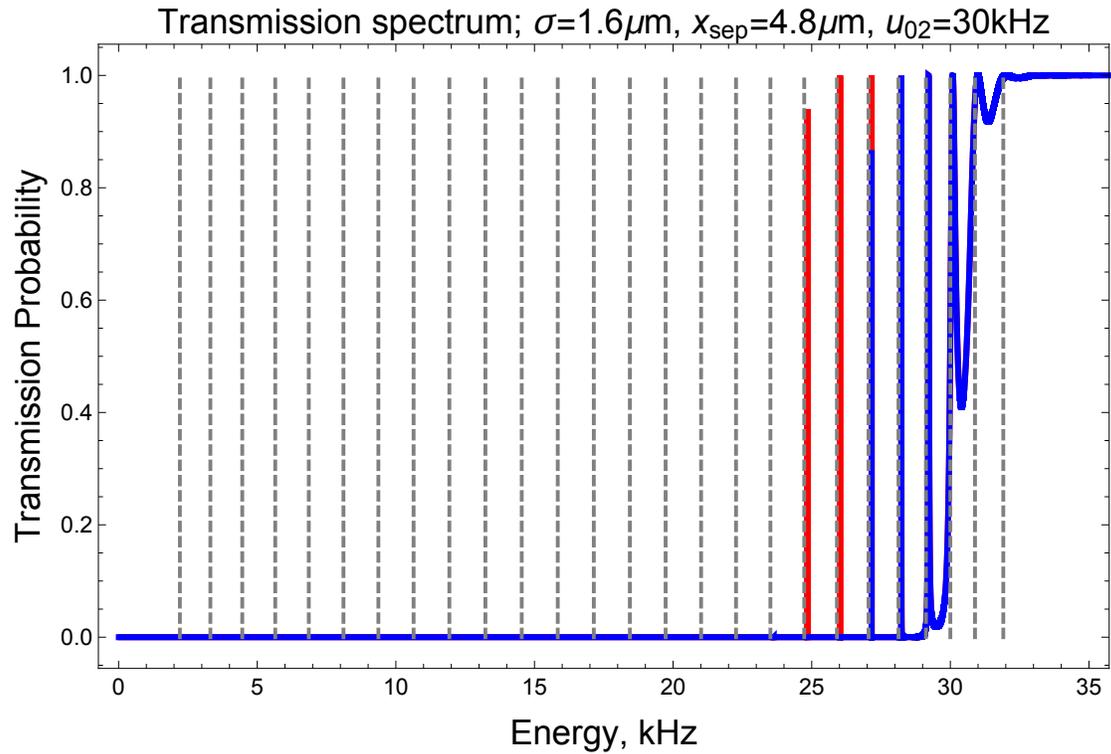
```

trans30sig2sep3 = ListLinePlot[
  Transpose@{energyBetter30sig2sep3 / (2 *  $\pi$  *  $\hbar$  * 1.0*^3), 1 - reflectCoef30sig2sep3},
  PlotRange -> All (*{{20,38},{-0.01,1.005}}*), GridLines -> {{}, {}},
  FrameLabel -> {Style[(*"Energy/v0"*)"Energy, kHz", 18],
    Style["Transmission Probability", 18]},
  PlotLabel -> Style["Transmission spectrum;  $\sigma=1.6\mu\text{m}$ ,  $x_{\text{sep}}=4.8\mu\text{m}$ ,  $u_{02}=30\text{kHz}$ ", 18],
  FrameTicksStyle -> Medium, ImageSize -> Large, Frame -> True,
  PlotStyle -> Directive[Red, Thickness[.006]],
  Epilog -> {Directive[{Thick, Gray, Dashed}],
    Table[Line[{{varsEigen30sig2[[1]][[i]], 0}, {varsEigen30sig2[[1]][[i]], 1}}],
      {i, 1, Length[varsEigen30sig2[[1]]}]}]

```



```
compairTrans30sig2sep3 = Show[trans30sig2sep3, varTrans30sig2sep3]
```



```
Export[
```

```
  "/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis  
  nb's/possible figs/compairTrans30sig2sep3.jpeg", compairTrans30sig2sep3];
```

```
varsEigen33sig2sep3Table
```

```
varsEigen33sig2sep3[[2]][[11 ;; 31]]
```

Eigen-energies for a double Gaussian defined												
33kHz	4.58	5.79	7.03	9.59	10.89	12.21	13.54	14.87	16.21	17.54	18.87	20.19

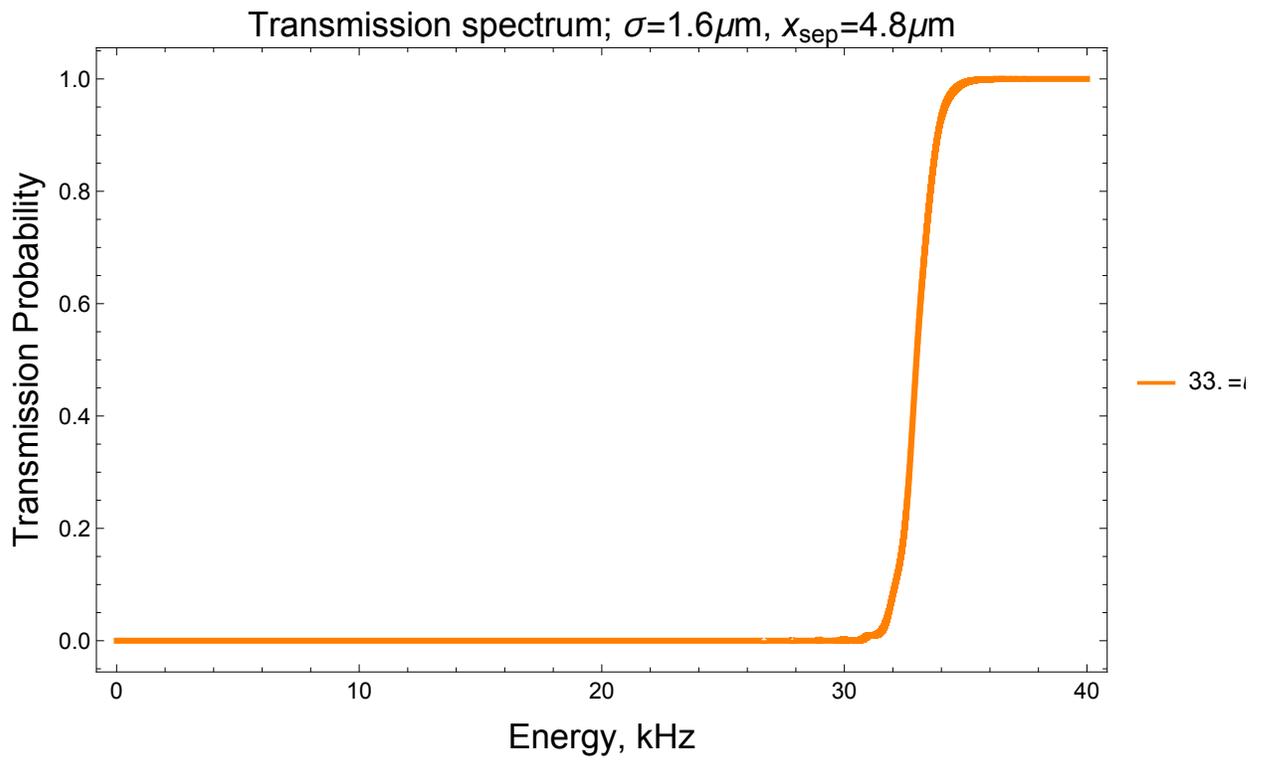
```
{4.58, 5.79, 7.03, 9.59, 10.89, 12.21, 13.54, 14.87, 16.21, 17.54, 18.87,  
 20.19, 21.51, 22.8, 24.08, 25.33, 26.55, 27.73, 28.85, 29.87, 30.98}
```

```

energyBetter33sig2sep3 =
(2 * π * ħ * 1.0 * 10^3) * Join[Table[i, {i, .1, 4.0, .1}], Table[i, {i, 4.0, 5.0, .1}],
  Table[i, {i, 5.0, 5.2, .1}], Table[i, {i, 5.2, 6.2, .1}], Table[i,
    {i, 6.2, 6.5, .1}], Table[i, {i, 6.5, 7.5, .1}], Table[i, {i, 7.5, 9.0, .1}],
  Table[i, {i, 9.0, 10.0, .1}], Table[i, {i, 10.0, 10.3, .1}],
  Table[i, {i, 10.3, 11.3, .1}], Table[i, {i, 11.3, 11.7, .1}],
  Table[i, {i, 11.7, 12.7, .1}], Table[i, {i, 12.7, 13.0, .1}],
  Table[i, {i, 13.0, 14.0, .1}], Table[i, {i, 14.0, 14.4, .1}],
  Table[i, {i, 14.4, 15.4, .1}], Table[i, {i, 15.4, 15.7, .1}],
  Table[i, {i, 15.7, 16.7, .1}], Table[i, {i, 16.7, 17.0, .1}],
  Table[i, {i, 17.0, 18.0, .1}], Table[i, {i, 18.0, 18.3, .1}],
  Table[i, {i, 18.3, 19.3, .1}], Table[i, {i, 19.3, 19.6, .1}],
  Table[i, {i, 19.6, 20.6, .1}], Table[i, {i, 20.6, 21.0, .1}],
  Table[i, {i, 21.0, 22.0, .1}], Table[i, {i, 22.0, 22.3, .1}],
  Table[i, {i, 22.3, 23.3, .01}], Table[i, {i, 23.3, 23.5, .1}],
  Table[i, {i, 23.5, 24.5, .001}], Table[i, {i, 24.5, 24.8, .1}],
  Table[i, {i, 24.8, 25.8, .0001}], Table[i, {i, 25.8, 26.0, .1}],
  Table[i, {i, 26.0, 27.0, .00001}], Table[i, {i, 27.0, 27.2, .01}],
  Table[i, {i, 27.2, 28.2, .000001}], Table[i, {i, 28.2, 28.3, .01}],
  Table[i, {i, 28.3, 29.3, .000001}], Table[i, {i, 29.3, 29.4, .01}],
  Table[i, {i, 29.4, 30.4, .000001}], Table[i, {i, 30.4, 30.5, .01}],
  Table[i, {i, 30.5, 31.5, .00001}], Table[i, {i, 31.5, 35.0, .01}]];
Dimensions[energyBetter33sig2sep3]
{3 211 772}

reflectCoef33sig2sep3 =
  ReflectionProbability[potential[[2, 3, 4, All]], energyBetter33sig2sep3, 1];
Dimensions[reflectCoef33sig2sep3]
{3 211 772}

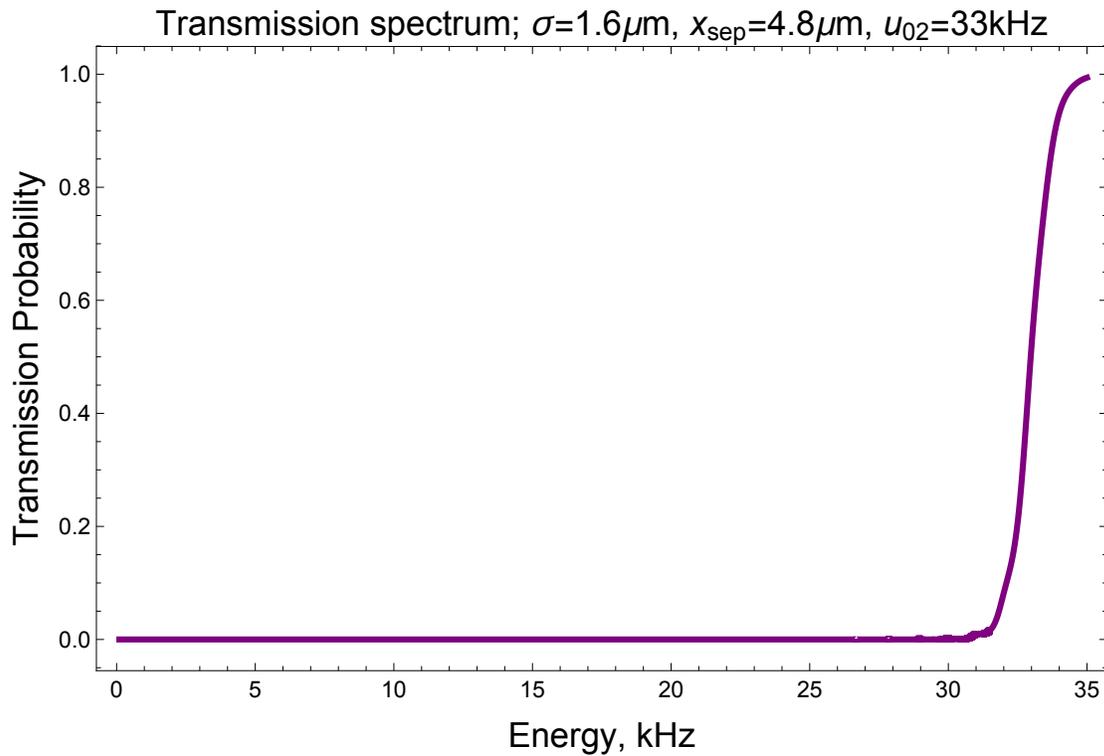
```

`varTrans33sig2sep3`

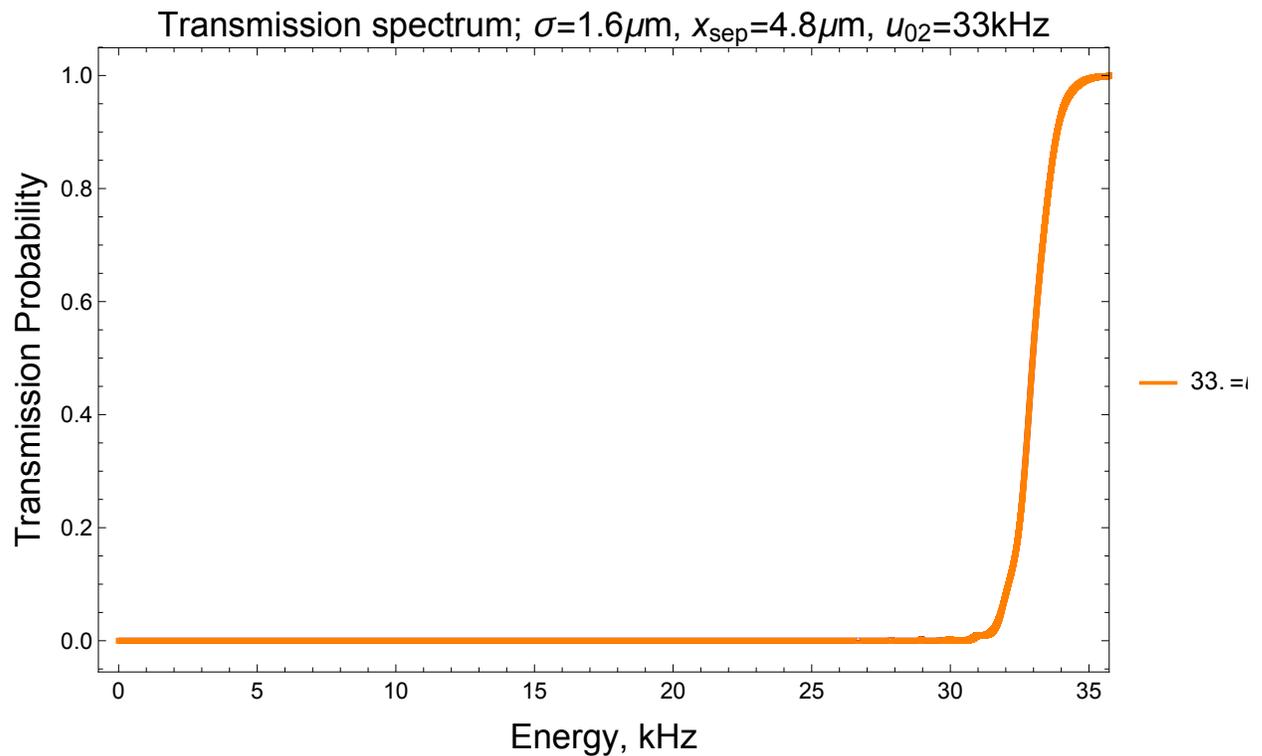
```

trans33sig2sep3 = ListLinePlot [
  Transpose@{energyBetter33sig2sep3 / (2 *  $\pi$  *  $\hbar$  * 1.0*^3), 1 - reflectCoef33sig2sep3},
  PlotRange -> All (*{{20,38},{-0.01,1.005}}*), GridLines -> {{}, {}},
  FrameLabel -> {Style[(*"Energy/v0"*)"Energy, kHz", 18],
    Style["Transmission Probability", 18]},
  PlotLabel -> Style["Transmission spectrum;  $\sigma=1.6\mu\text{m}$ ,  $x_{\text{sep}}=4.8\mu\text{m}$ ,  $u_{02}=33\text{kHz}$ ", 18],
  FrameTicksStyle -> Medium, ImageSize -> Large, Frame -> True,
  PlotStyle -> Directive[Purple, Thickness[.006]]]

```



Show[trans33sig2sep3, varTrans33sig2sep3]



varsEigen30sig2sep5Table

varsEigen30sig2[[2]]

	Eigen-ener											
30kHz	0.88	1.52	2.22	2.98	3.79	5.52	6.43	7.38	8.34	9.33	10.34	11.36

{0.88, 1.52, 2.22, 2.98, 3.79, 5.52, 6.43, 7.38, 8.34, 9.33, 10.34,  
11.36, 12.4, 13.45, 14.51, 15.57, 16.65, 17.73, 18.81, 19.89, 20.96,  
22.03, 23.1, 24.15, 25.18, 26.2, 27.19, 28.13, 29.02, 29.83, 30.62, 31.49}

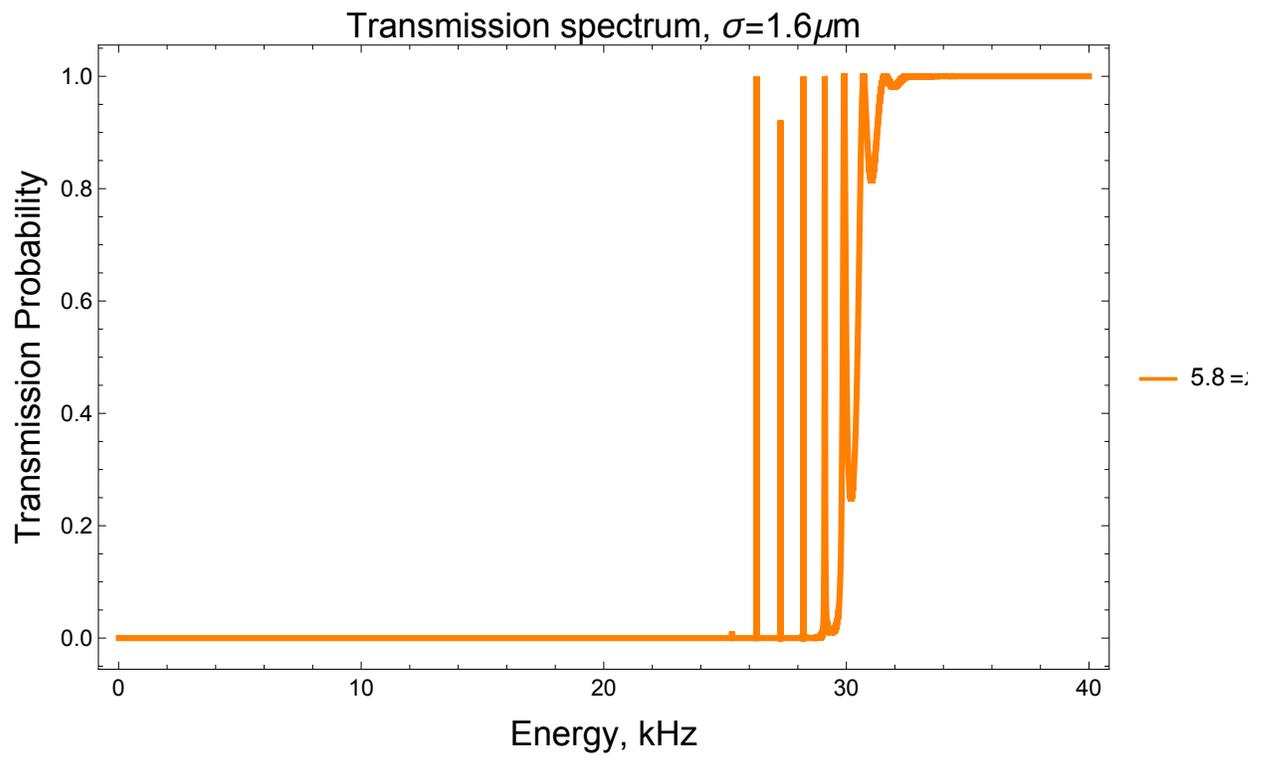
```

energyBetter30sig2sep5 =
(2 * π * ħ * 1.0 * 10^3) * Join[Table[i, {i, .1, .5, .01}], Table[i, {i, .5, 1.1, .1}],
  Table[i, {i, 1.1, 1.2, .1}], Table[i, {i, 1.2, 1.7, .1}],
  Table[i, {i, 1.7, 1.9, .1}], Table[i, {i, 1.9, 2.5, .1}],
  Table[i, {i, 2.5, 2.6, .1}], Table[i, {i, 2.6, 3.2, .1}],
  Table[i, {i, 3.2, 3.4, .1}], Table[i, {i, 3.4, 4.0, .1}],
  Table[i, {i, 4.0, 5.2, .1}], Table[i, {i, 5.2, 5.8, .1}],
  Table[i, {i, 5.8, 6.1, .1}], Table[i, {i, 6.1, 6.7, .1}],
  Table[i, {i, 6.7, 7.0, .1}], Table[i, {i, 7.0, 7.6, .1}],
  Table[i, {i, 7.6, 8.0, .1}], Table[i, {i, 8.0, 8.6, .1}], Table[i,
    {i, 8.6, 9.0, .1}], Table[i, {i, 9.0, 9.6, .1}], Table[i, {i, 9.6, 10.0, .1}],
  Table[i, {i, 10.0, 10.6, .1}], Table[i, {i, 10.6, 11.0, .1}],
  Table[i, {i, 11.0, 11.6, .1}], Table[i, {i, 11.6, 12.1, .1}],
  Table[i, {i, 12.1, 12.7, .1}], Table[i, {i, 12.7, 13.1, .1}],
  Table[i, {i, 13.1, 13.7, .1}], Table[i, {i, 13.7, 14.2, .1}],
  Table[i, {i, 14.2, 14.8, .1}], Table[i, {i, 14.8, 15.2, .1}],
  Table[i, {i, 15.2, 15.8, .1}], Table[i, {i, 15.8, 16.3, .1}],
  Table[i, {i, 16.3, 16.9, .1}], Table[i, {i, 16.9, 17.4, .1}],
  Table[i, {i, 17.4, 18.0, .1}], Table[i, {i, 18.0, 18.5, .1}],
  Table[i, {i, 18.5, 19.1, .1}], Table[i, {i, 19.1, 19.5, .1}],
  Table[i, {i, 19.5, 20.1, .1}], Table[i, {i, 20.1, 20.7, .1}],
  Table[i, {i, 20.7, 21.2, .1}], Table[i, {i, 21.2, 21.7, .1}],
  Table[i, {i, 21.7, 22.3, .1}], Table[i, {i, 22.3, 22.8, .1}],
  Table[i, {i, 22.8, 23.4, .1}], Table[i, {i, 23.4, 23.8, .1}],
  Table[i, {i, 23.8, 24.4, .0001}], Table[i, {i, 24.4, 24.8, .00001}],
  Table[i, {i, 24.8, 25.4, .000001}], Table[i, {i, 25.4, 25.8, .00001}],
  Table[i, {i, 25.8, 26.5, .000001}], Table[i, {i, 26.5, 26.8, .0001}],
  Table[i, {i, 26.8, 27.4, .000001}], Table[i, {i, 27.4, 27.8, .001}],
  Table[i, {i, 27.8, 28.4, .000001}], Table[i, {i, 28.4, 28.7, .001}],
  Table[i, {i, 28.7, 29.3, .000001}], Table[i, {i, 29.3, 35.0, .001}]];
Dimensions[energyBetter30sig2sep5]
{3 195 732}

reflectCoef30sig2sep5 =
  ReflectionProbability[potential[[2, 5, 1, All]], energyBetter30sig2sep5, 1];
Dimensions[reflectCoef30sig2sep5]
{3 195 732}

```

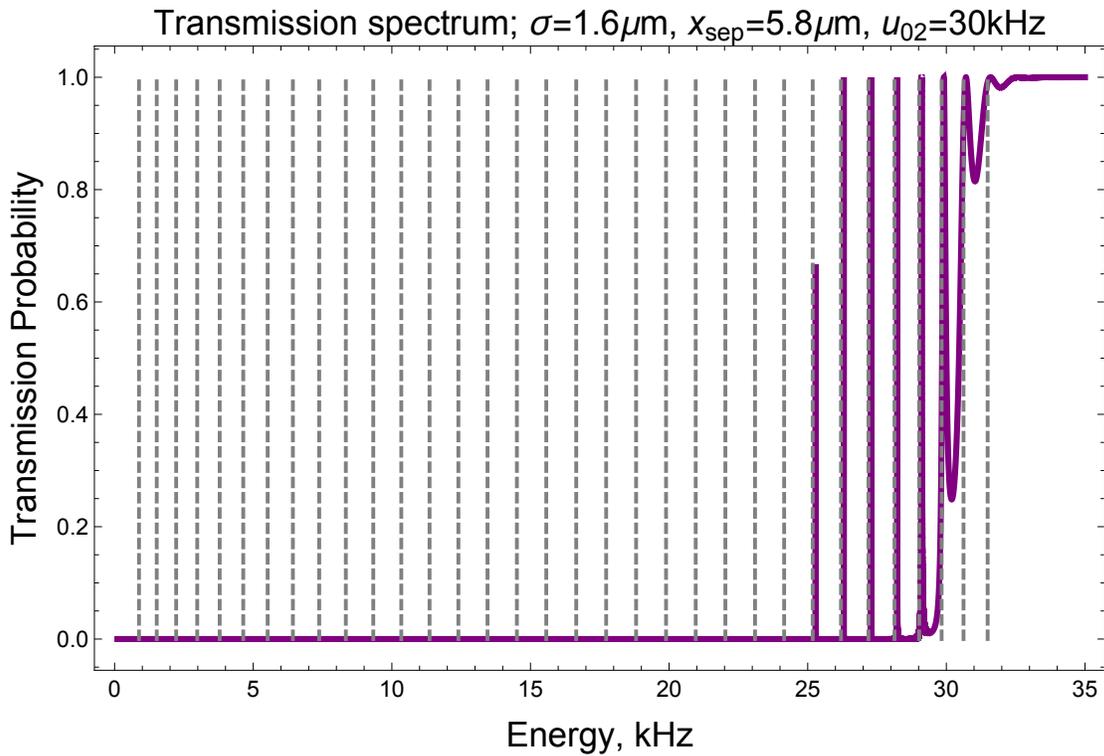
varTrans30sig2sep5



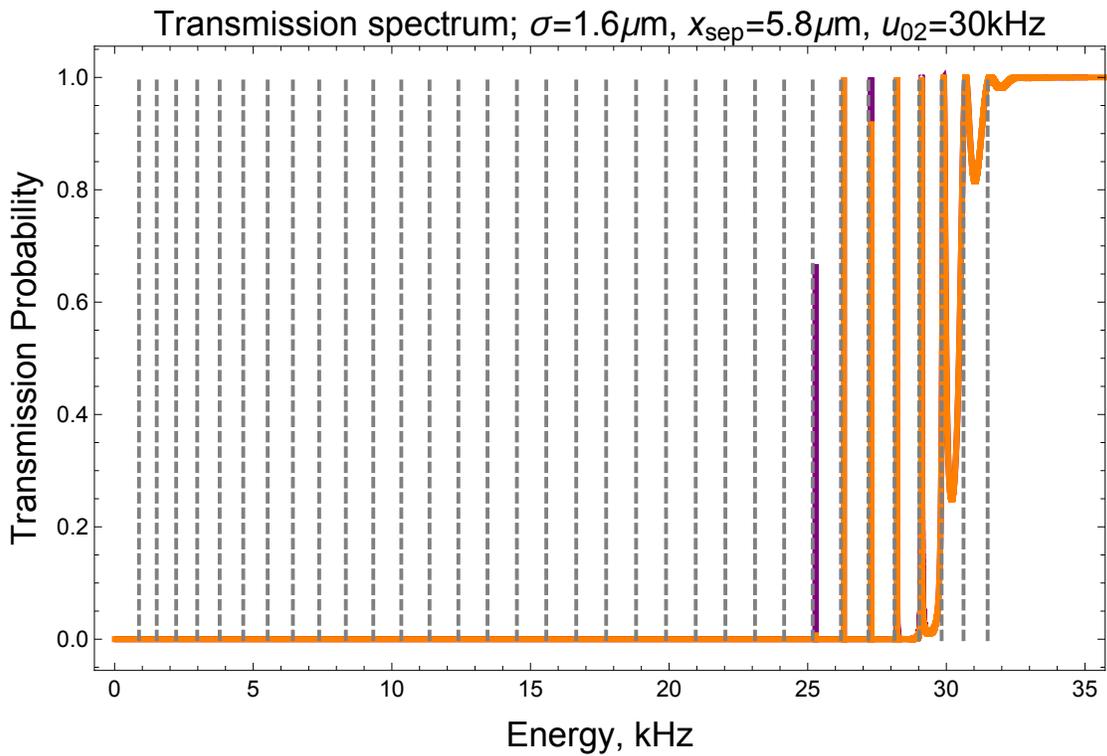
```

trans30sig2sep5 = ListLinePlot[
  Transpose@{energyBetter30sig2sep5 / (2 *  $\pi$  *  $\hbar$  * 1.0*^3), 1 - reflectCoef30sig2sep5},
  PlotRange -> All (*{{20,38},{-0.01,1.005}}*), GridLines -> {{}, {}},
  FrameLabel -> {Style[(*"Energy/v0"*)"Energy, kHz", 18],
    Style["Transmission Probability", 18]},
  PlotLabel -> Style["Transmission spectrum;  $\sigma=1.6\mu\text{m}$ ,  $x_{\text{sep}}=5.8\mu\text{m}$ ,  $u_{02}=30\text{kHz}$ ", 18],
  FrameTicksStyle -> Medium, ImageSize -> Large, Frame -> True,
  PlotStyle -> Directive[Purple, Thickness[.006]],
  Epilog -> {Directive[{Thick, Gray, Dashed}],
    Table[Line[{{varsEigen30sig2[[2]][[i]], 0}, {varsEigen30sig2[[2]][[i]], 1}},
      {i, 1, Length[varsEigen30sig2[[2]]}]}]}

```



```
compairTrans30sig2sep5 = Show[trans30sig2sep5, varTrans30sig2sep5]
```



```
Export[
```

```
  "/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis  
  nb's/possible figs/compairTrans30sig2sep5.jpeg", compairTrans30sig2sep5];
```

```
varsEigen30sig1sep3Table
```

```
varsEigen30sig1[[2]]
```

Eigen-energies for a double Gaussian def													
30kHz	2.22	4.72	5.67	6.68	7.73	8.83	9.97	11.14	12.34	13.57	14.83	16.11	17.41

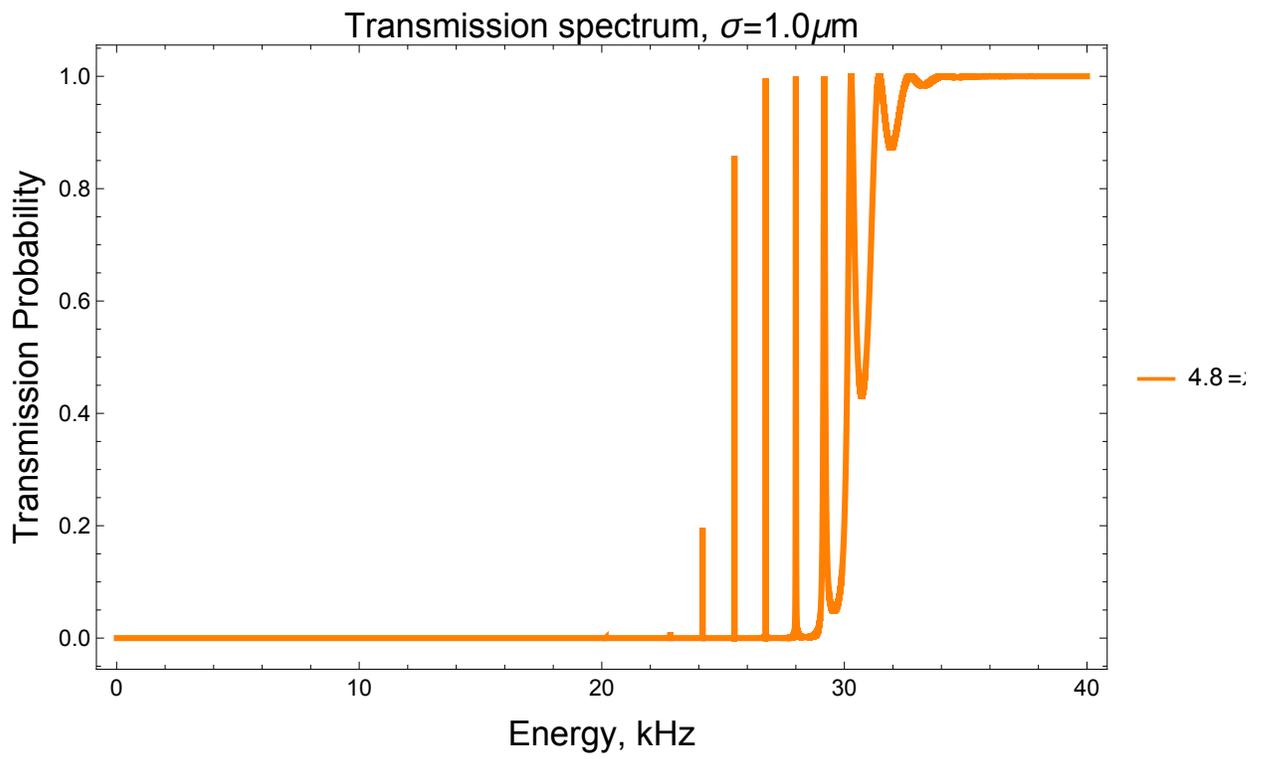
```
{2.22, 4.72, 5.67, 6.68, 7.73, 8.83, 9.97, 11.14, 12.34, 13.57, 14.83, 16.11, 17.41,  
18.72, 20.04, 21.37, 22.7, 24.03, 25.34, 26.63, 27.87, 29.05, 30.16, 31.31}
```

```

energyBetter30siglsep3 =
(2 * π * ħ * 1.0 * 10^3) * Join[Table[i, {i, .1, 1.9, .1}], Table[i, {i, 1.9, 2.5, .1}],
  Table[i, {i, 2.5, 4.3, .1}], Table[i, {i, 4.3, 5.1, .1}],
  Table[i, {i, 5.1, 5.2, .1}], Table[i, {i, 5.2, 6.0, .1}],
  Table[i, {i, 6.0, 6.2, .1}], Table[i, {i, 6.2, 7.0, .1}],
  Table[i, {i, 7.0, 7.3, .1}], Table[i, {i, 7.3, 8.1, .1}], Table[i,
    {i, 8.1, 8.4, .1}], Table[i, {i, 8.4, 9.2, .1}], Table[i, {i, 9.2, 9.5, .1}],
  Table[i, {i, 9.5, 10.3, .1}], Table[i, {i, 10.3, 10.7, .1}],
  Table[i, {i, 10.7, 11.5, .1}], Table[i, {i, 11.5, 11.9, .1}],
  Table[i, {i, 11.9, 12.7, .1}], Table[i, {i, 12.7, 13.1, .1}],
  Table[i, {i, 13.1, 13.9, .1}], Table[i, {i, 13.9, 14.4, .1}],
  Table[i, {i, 14.4, 15.2, .1}], Table[i, {i, 15.2, 15.7, .01}],
  Table[i, {i, 15.7, 16.5, .001}], Table[i, {i, 16.5, 17.0, .01}],
  Table[i, {i, 17.0, 17.8, .001}], Table[i, {i, 17.8, 18.3, .001}],
  Table[i, {i, 18.3, 19.1, .000001}], Table[i, {i, 19.1, 19.6, .001}],
  Table[i, {i, 19.6, 20.4, .000001}], Table[i, {i, 20.4, 20.9, .001}],
  Table[i, {i, 20.9, 21.7, .000001}], Table[i, {i, 21.7, 22.4, .001}],
  Table[i, {i, 22.4, 23.1, .000001}], Table[i, {i, 23.1, 23.6, .001}],
  Table[i, {i, 23.6, 24.4, .000001}], Table[i, {i, 24.4, 24.9, .001}],
  Table[i, {i, 24.9, 25.7, .00001}], Table[i, {i, 25.7, 26.2, .001}],
  Table[i, {i, 26.2, 27.0, .00001}], Table[i, {i, 27.0, 27.4, .01}],
  Table[i, {i, 27.4, 28.2, .0001}], Table[i, {i, 28.2, 28.6, .01}],
  Table[i, {i, 28.6, 29.4, .0001}], Table[i, {i, 29.4, 35, .001}]];
Dimensions[energyBetter30siglsep3]
{4 087 276}

reflectCoef30siglsep3 =
  ReflectionProbability[potential[[1, 3, 1, All]], energyBetter30siglsep3, 1];
Dimensions[reflectCoef30siglsep3]
{4 087 276}

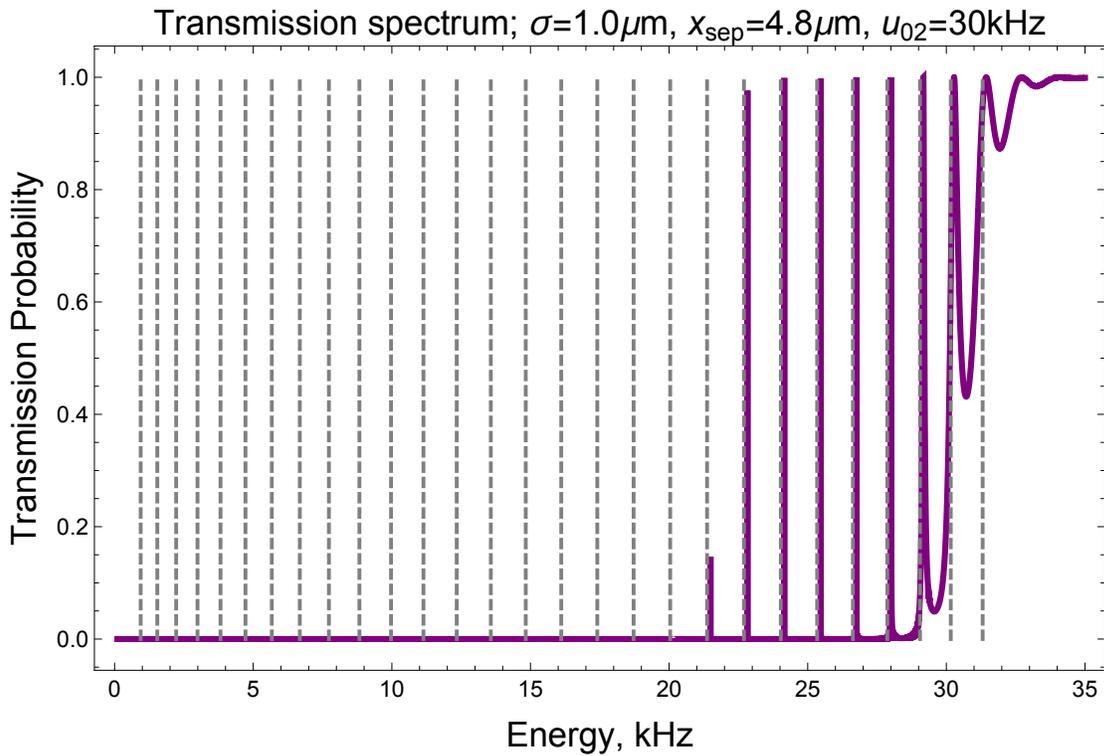
```

`varTrans30sig1sep3`

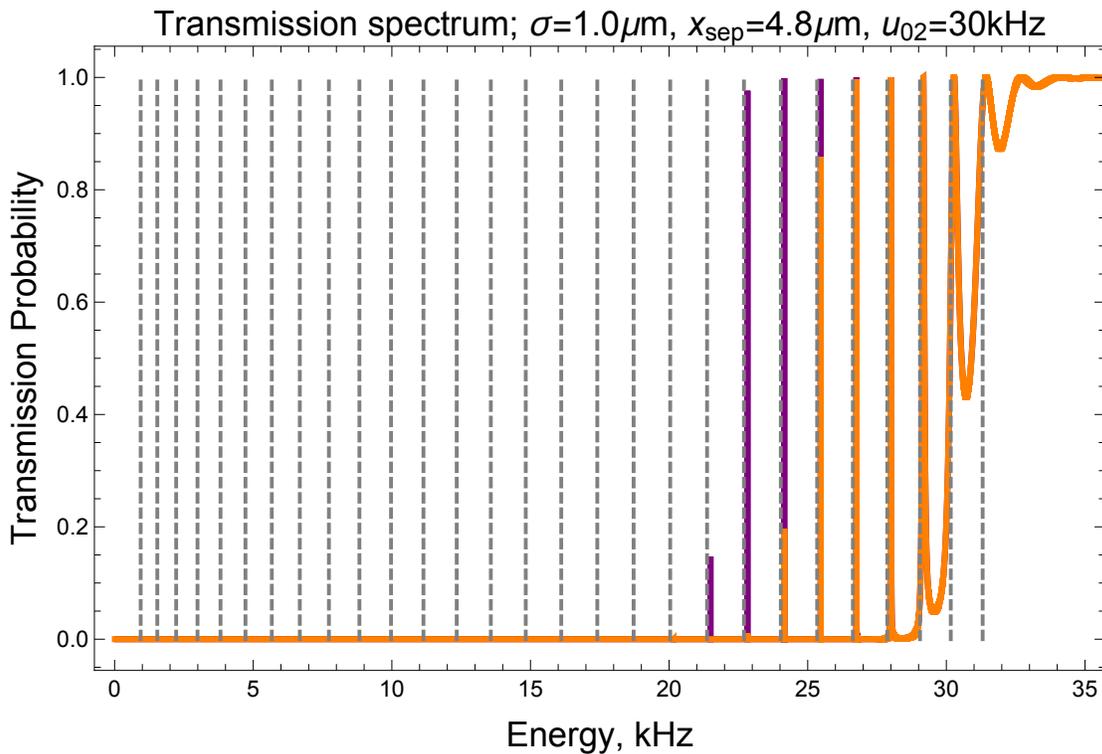
```

trans30sig1sep3 = ListLinePlot[
  Transpose@{energyBetter30sig1sep3 / (2 *  $\pi$  *  $\hbar$  * 1.0*^3), 1 - reflectCoef30sig1sep3},
  PlotRange -> All (*{{20,38},{-0.01,1.005}}*), GridLines -> {{}, {}},
  FrameLabel -> {Style[(*"Energy/v0"*)"Energy, kHz", 18],
    Style["Transmission Probability", 18]},
  PlotLabel -> Style["Transmission spectrum;  $\sigma=1.0\mu\text{m}$ ,  $x_{\text{sep}}=4.8\mu\text{m}$ ,  $u_{02}=30\text{kHz}$ ", 18],
  FrameTicksStyle -> Medium, ImageSize -> Large, Frame -> True,
  PlotStyle -> Directive[Purple, Thickness[.006]],
  Epilog -> {Directive[{Thick, Gray, Dashed}],
    Table[Line[{{varsEigen30sig1[[2]][[i]], 0}, {varsEigen30sig1[[2]][[i]], 1}}],
      {i, 1, Length[varsEigen30sig1[[2]]}]}]

```



```
compairTrans30sig1sep3 = Show[trans30sig1sep3, varTrans30sig1sep3]
```



```
Export [
```

```
  "/Users/avaashby/Desktop/masters-thesis/Mathematica Code/ImpedanceMethod/thesis  
  nb's/possible figs/compairTrans30sig1sep3.jpeg", compairTrans30sig1sep3];
```

## Possible wavefunction representation: using the double rectangular coefficients

```
In[48]= a3030
```

```
Out[48]= {1.16551, 2.20647, 3.31029, 4.45918, 5.6462, 6.86311, 8.10521, 9.367, 10.6448,  
  11.9344, 13.2325, 14.5355, 15.8403, 17.1435, 18.4419, 19.7318, 21.0098,  
  22.2716, 23.5127, 24.7274, 25.9088, 27.0471, 28.1263, 29.1165, 29.9996}
```

```
In[119]= waveRef = ReflectionProbability[potential[[2, 3, 1, All]], energyBetterer5, 1];  
  Dimensions [waveRef]
```

```
Out[120]= {500 001}
```

```
In[176]= positFullRef3030 = Flatten[Position[1 - waveRef, _? (# >= .999999 &)]];  
  Table[1 - waveRef[[positFullRef3030[[i]]]], {i, Length[positFullRef3030]}]
```

```
Out[177]= {0.999999, 1., 1., 0.999999}
```

```
In[141]= energyBetterer5[[positFullRef3030[[2]]]] / (2 *  $\pi$  *  $\hbar$  *  $10^3$ )
```

```
Out[141]= 29.2144
```

```
In[196]= posit3030 = Flatten[Position[potential[[2, 3, 1, All]] / (2 * π * ħ * 10^3),
  _? (# >= energyBetterer5[[positFullRef3030[[2]]]] / (2 * π * ħ * 10^3) &)]];
Dimensions[posit3030]
posit3030Pot =
  Table[potential[[2, 3, 1, posit3030[[i]]]], {i, Length[posit3030]}] / (2 * π * ħ * 10^3)
```

```
Out[197]= {50}
```

```
Out[198]= {29.2502, 29.3687, 29.4773, 29.5759, 29.6644, 29.7427, 29.8108, 29.8685,
  29.9157, 29.9526, 29.9789, 29.9947, 30., 29.9947, 29.9789, 29.9526, 29.9157,
  29.8685, 29.8108, 29.7427, 29.6644, 29.5759, 29.4773, 29.3687, 29.2502,
  29.2502, 29.3687, 29.4773, 29.5759, 29.6644, 29.7427, 29.8108, 29.8685,
  29.9157, 29.9526, 29.9789, 29.9947, 30., 29.9947, 29.9789, 29.9526, 29.9157,
  29.8685, 29.8108, 29.7427, 29.6644, 29.5759, 29.4773, 29.3687, 29.2502}
```

```
In[200]= posit3030[[1]]
posit3030[[25]]
posit3030[[26]]
posit3030[[50]]
```

```
Out[200]= 329
```

```
Out[201]= 353
```

```
Out[202]= 649
```

```
Out[203]= 673
```

```
In[218]= m = 1.44*^-25;
k[nrg_] := Sqrt[2 * m * nrg / ħ^2];
κ1[i_, nrg_] := Sqrt[2 * m * (potential[[2, 3, 1, i]] - nrg) / ħ^2];
κ2[i_, nrg_] := Sqrt[2 * m * (potential[[2, 3, 1, i]] - nrg) / ħ^2];
x1 = Abs[x[[649]]] - Abs[x[[353]]];
x2 = Abs[x[[673]]] - x[[649]] + x1;
Δ = x1 - x2;
```

```
In[227]= c[i_, nrg_, rAmp_] := Exp[-κ1[i, nrg] * x2] *  $\frac{1}{2}$  *
  (rAmp * Exp[i * k[nrg] * x2] * (  $\frac{i * k[nrg]}{\kappa1[i, nrg]} + 1$  ) - Exp[-i * k[nrg] * x2] * (  $\frac{i * k[nrg]}{\kappa1[i, nrg]} - 1$  ))
cc[i_, nrg_, rAmp_] := Exp[κ1[i, nrg] * x2] *  $\frac{1}{2}$  *
  (Exp[-i * k[nrg] * x2] * (  $1 + \frac{i * k[nrg]}{\kappa1[i, nrg]}$  ) + rAmp * Exp[i * k[nrg] * x2] * (  $1 - \frac{i * k[nrg]}{\kappa1[i, nrg]}$  ))
```

```
In[229]= psiI[i_, nrg_, rAmp_] := Re[Exp[i * k[nrg] * x[[i]]] + rAmp * Exp[-i * k[nrg] * x[[i]]]]
```

```
In[230]= psiII[i_, nrg_, rAmp_] := Re[c[i, nrg, rAmp] * Exp[-κ1[i, nrg] * x[[i]]] +
  cc[i, nrg, rAmp] * Exp[κ1[i, nrg] * x[[i]]]]
```

```

d[i_, nrg_, rAmp_] :=
  1/2 * Exp[i * k[nrg] * x1] * (rAmp * Exp[i * k[nrg] * x2] * (i * k[nrg] / κ1[nrg] - κ1[nrg] / i * k[nrg]) *
    Sinh[κ1[nrg] * Δ] + Exp[-i * k[nrg] * x2] *
    (2 * Cosh[κ1[nrg] * Δ] - (κ1[nrg] / i * k[nrg] + i * k[nrg] / κ1[nrg]) * Sinh[κ1[nrg] * Δ]))
f[nrg_, rAmp_] := 1/2 * Exp[-i * k[nrg] * x1] * (rAmp * Exp[i * k[nrg] * x2] *
  (2 * Cosh[κ1[nrg] * Δ] + (i * k[nrg] / κ1[nrg] + κ1[nrg] / i * k[nrg]) * Sinh[κ1[nrg] * Δ]) -
  Exp[-i * k[nrg] * x2] * (i * k[nrg] / κ1[nrg] - κ1[nrg] / i * k[nrg]) * Sinh[κ1[nrg] * Δ])
psiIII[i_, nrg_, rAmp_] :=
  Re[d[nrg, rAmp] * Exp[i * k[nrg] * x[[i]]] + f[nrg, rAmp] * Exp[-i * k[nrg] * x[[i]]]]
tAmp[rAmp_] := Sqrt[1 - Abs[rAmp]^2]
gg[nrg_, rAmp_] := Sqrt[1 - Abs[rAmp]^2] * Exp[i * k[nrg] * x2 + κ2[nrg] * x2]
g4[nrg_, rAmp_] :=
  1/2 * Exp[κ2[nrg] * x1] * (d[nrg, rAmp] * Exp[i * k[nrg] * x1] * (1 - i * k[nrg] / κ2[nrg]) +
  f[nrg, rAmp] * Exp[-i * x1] * (1 + i * k[nrg] / κ2[nrg]))
gg4[nrg_, rAmp_] := 1/2 * Exp[-κ2[nrg] * x1] *
  (d[nrg, rAmp] * Exp[i * k[nrg] * x1] * (i * k[nrg] / κ2[nrg] + 1) +
  f[nrg, rAmp] * Exp[-i * x1] * (1 - i * k[nrg] / κ2[nrg]))
psiIV[i_, nrg_, rAmp_] :=
  Re[g4[nrg, rAmp] * Exp[-κ2[nrg] * x[[i]]] (*+gg4[nrg, rAmp]*Exp[κ2[nrg]*x[[i]]*)
  (*gg[nrg, rAmp]*Exp[-κ2[nrg]*x[[i]]*)]
psiV[i_, nrg_, rAmp_] := Re[tAmp[rAmp] * Exp[i * k[nrg] * x[[i]]]]
ψI = Flatten@Table[psiI[i, energyGood[[1, 269 501]], rAmp], {i, 1, 957}];
ψII = Flatten@Table[psiII[i, energyGood[[1, 269 501]], rAmp] * 1, {i, 958, 980}];
ψIII = Flatten@Table[psiIII[i, energyGood[[1, 269 501]], rAmp] * 1, {i, 981, 1021}];
ψIV = Flatten@Table[psiIV[i, energyGood[[1, 269 501]], rAmp] * 1, {i, 1022, 1044}];
ψV = Flatten@Table[psiV[i, energyGood[[1, 269 501]], rAmp], {i, 1045, Length[x]}];
Ψ = energyGood[[1, 269 501]] / (2 * π * ħ * 1.0*^3) + Join[ψI, ψII, ψIII, ψIV, ψV];
Dimensions[Ψ]

doubRectWave = ListPlot[{Transpose[{x / 10^-6, Ψ},
  Transpose[{x / 10^-6, potential[[1, All]] / (2 * π * ħ * 10^3)}],
  Joined → True, PlotRange → {{-2, 2}, {-0.1, 46}}, Frame → True, FrameLabel →
  {{Style["Potential Energy Function, kHz", 18], Style["Wavefunction, Ψ(x)", 18]},
  {Style["Position, μm", 18], None}}, PlotLabel →
  Style["Double Rectangular Barrier Potential and Transmitting Wave", 18],
  FrameTicksStyle → Medium, ImageSize → Large, PlotStyle → Thickness[.004]}]

```

## Appendix H

Eigenenergy tables for the double Gaussian barrier potential

Eigen-energies for a double Gaussian defined by $\sigma=1.0\mu m$ , $x_{sep}=2.5\mu m$													
30kHz	3.87	6.39	8.95	11.51	14.06	16.58	19.04	21.44	23.74	25.92	27.9	29.62	31.26
$\Delta$		2.52	2.56	2.56	2.55	2.52	2.46	2.4	2.3	2.18	1.98	1.72	1.64

Table H.1: Table of the resonant energies for a double Gaussian barrier potential defined by  $\sigma = 1.0\mu m$ ,  $x_{sep} = 2.5\mu m$ , and  $u_{02} = 30kHz$ , and the difference between the resonant energies to order  $10^0$ . Notice that the first resonant energy seen in the transmission spectrum is about  $E \approx 23.74kHz$ , but there are eight resonant energies not shown in the spectrum. This will allow us to optimize the energy density such that we can see more of the resonant energies in the spectrum. The spacing between the resonant energies decreased and became more uniform as the resonant energy increased.

Eigen-energies for a double Gaussian defined by $\sigma=1.0\mu m$ , $x_{sep}=4.8\mu m$																												
30kHz	0.94	1.54	2.22	2.99	3.82	4.72	5.67	6.68	7.73	8.83	9.97	11.14	12.34	13.57	14.83	16.11	17.41	18.72	20.04	21.37	22.7	24.03	25.34	26.63	27.87	29.05	30.16	31.31
$\Delta$		0.6	0.68	0.77	0.83	0.9	0.95	1.01	1.05	1.1	1.14	1.17	1.2	1.23	1.26	1.28	1.3	1.31	1.32	1.33	1.33	1.33	1.31	1.29	1.24	1.18	1.11	1.15

Table H.2: Table of the resonant energies for a double Gaussian barrier potential defined by  $\sigma = 1.0\mu m$ ,  $x_{sep} = 4.8\mu m$ , and  $u_{02} = 30kHz$ , and the difference between the resonant energies to order  $10^0$ .

Eigen-energies for a double Gaussian defined by $\sigma=1.0\mu m$ , $x_{sep}=5.8\mu m$																																				
30kHz	0.83	1.24	1.71	2.24	2.82	3.44	4.11	4.82	5.57	6.35	7.17	8.02	8.91	9.82	10.76	11.72	12.71	13.72	14.75	15.8	16.87	17.95	19.05	20.15	21.27	22.39	23.51	24.63	25.74	26.84	27.92	28.94	29.93	30.92	31.98	
$\Delta$		0.41	0.47	0.53	0.58	0.62	0.67	0.71	0.75	0.78	0.82	0.85	0.89	0.91	0.94	0.96	0.99	1.01	1.03	1.05	1.07	1.08	1.1	1.1	1.12	1.12	1.12	1.12	1.12	1.11	1.1	1.08	1.02	0.99	0.99	1.06

Table H.3: Table of the resonant energies for a double Gaussian barrier potential defined by  $\sigma = 1.0\mu m$ ,  $x_{sep} = 5.8\mu m$ , and  $u_{02} = 30kHz$ , and the difference between the resonant energies to order  $10^0$ . It is clear that again the number of allowed resonant energies has increased.

Eigen-energies for a double Gaussian defined by $\sigma=1.6\mu\text{m}$ , $x_{\text{sep}}=4.8\mu\text{m}$																											
30kHz	1.17	2.21	3.31	4.46	5.65	6.86	8.11	9.37	10.64	11.93	13.23	14.54	15.84	17.14	18.44	19.73	21.01	22.27	23.51	24.73	25.91	27.05	28.13	29.12	30.	30.89	31.91
$\Delta$		1.04	1.1	1.15	1.19	1.21	1.25	1.26	1.27	1.29	1.3	1.31	1.3	1.3	1.3	1.29	1.28	1.26	1.24	1.22	1.18	1.14	1.08	0.99	0.88	0.89	1.02

Table H.4: Table of the resonant energies for a double Gaussian barrier potential defined by  $\sigma = 1.6\mu\text{m}$ ,  $x_{\text{sep}} = 4.8\mu\text{m}$ , and  $u_{02} = 30\text{kHz}$ , and the difference between the resonant energies to order  $10^0$ .

Eigen-energies for a double Gaussian defined by $\sigma=1.6\mu\text{m}$ , $x_{\text{sep}}=5.8\mu\text{m}$																																	
30kHz	0.88	1.52	2.22	2.98	3.79	4.64	5.52	6.43	7.38	8.34	9.33	10.34	11.36	12.4	13.45	14.51	15.57	16.65	17.73	18.81	19.89	20.96	22.03	23.1	24.15	25.18	26.2	27.19	28.13	29.02	29.83	30.62	31.49
$\Delta$		0.64	0.7	0.76	0.81	0.85	0.88	0.91	0.95	0.96	0.99	1.01	1.02	1.04	1.05	1.06	1.06	1.08	1.08	1.08	1.08	1.07	1.07	1.07	1.05	1.03	1.02	0.99	0.94	0.89	0.81	0.79	0.87

Table H.5: Table of the resonant energies for a double Gaussian barrier potential defined by  $\sigma = 1.6\mu\text{m}$ ,  $x_{\text{sep}} = 5.8\mu\text{m}$ , and  $u_{02} = 30\text{kHz}$ , and the difference between the resonant energies to order  $10^0$ .

Eigen-energies for a double Gaussian defined by $\sigma=1.6\mu\text{m}$ , $x_{\text{sep}}=4.8\mu\text{m}$																											
31kHz	1.18	2.23	3.34	4.5	5.69	6.92	8.17	9.44	10.73	12.03	13.34	14.65	15.97	17.28	18.59	19.89	21.18	22.46	23.71	24.94	26.14	27.29	28.39	29.4	30.27	31.03	31.84
$\Delta$		1.05	1.11	1.16	1.19	1.23	1.25	1.27	1.29	1.3	1.31	1.31	1.32	1.31	1.31	1.3	1.29	1.28	1.25	1.23	1.2	1.15	1.1	1.01	0.87	0.76	0.81

Table H.6: Table of the resonant energies for a double Gaussian barrier potential defined by  $\sigma = 1.6\mu\text{m}$ ,  $x_{\text{sep}} = 4.8\mu\text{m}$ , and  $u_{02} = 31\text{kHz}$ , and the difference between the resonant energies to order  $10^0$ . Notice that the resonant energies match within an order of  $10^{-1}$  with those in Fig. 5.12, but the transmission is no longer unity for those of similar resonance.

Eigen-energies for a double Gaussian defined by $\sigma=1.6\mu\text{m}$ , $x_{sep}=4.8\mu\text{m}$																							
33kHz	3.4	4.58	5.79	7.03	8.3	9.59	10.89	12.21	13.54	14.87	16.21	17.54	18.87	20.19	21.51	22.8	24.08	25.33	26.55	27.73	28.85	29.87	30.98
$\Delta$		1.18	1.21	1.24	1.27	1.29	1.3	1.32	1.33	1.33	1.34	1.33	1.33	1.32	1.32	1.29	1.28	1.25	1.22	1.18	1.12	1.02	1.11

Table H.7: Table of the resonant energies for a double Gaussian barrier potential defined by  $\sigma = 1.6\mu\text{m}$ ,  $x_{sep} = 4.8\mu\text{m}$ , and  $u_{02} = 33\text{kHz}$ , and the difference between the resonant energies to order  $10^0$ . Notice that the lower resonances are within an order of  $10^{-1}$  of the resonant energies of the symmetric double Gaussian barrier potential. The asymmetric double Gaussian barrier has resonances that do not achieve full transmission unlike its symmetric counter part.

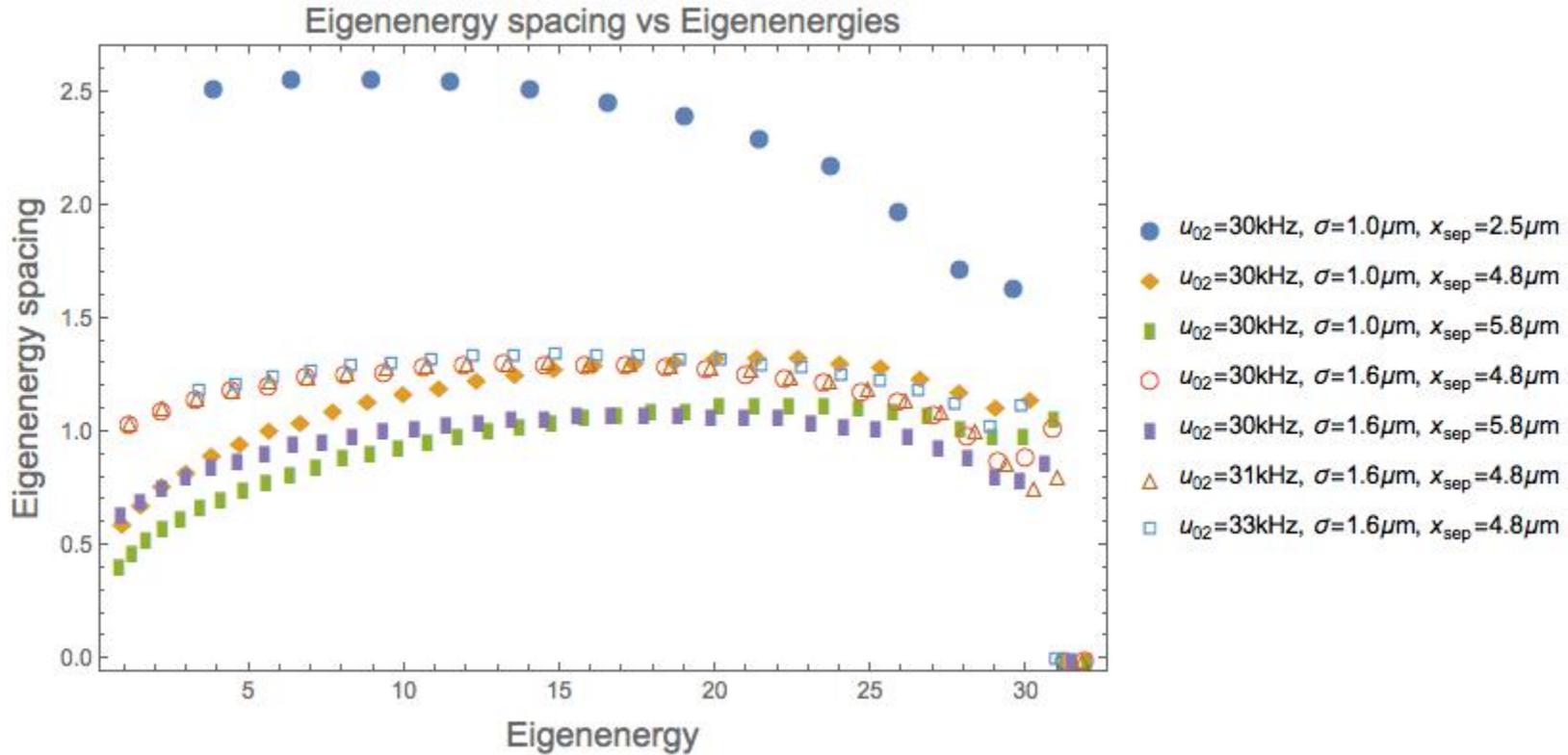


Figure H.1: Overall trends of the resonant energies for each double Gaussian barrier potential transmission spectrum studied thus far. The resonant energy spacing of the harmonic like potentials (blue solid circle, red open circle, orange open triangle, light blue open square) are, in general, more uniform in the spacing of the lower resonant energies. The non-harmonic potentials (orange solid diamond, green rectangle, purple rectangle) have resonant energy spacings closer together for the lower resonances and further apart for the higher resonances. For the potentials with only a difference in height of the second barrier (red open circle, orange open triangle, light blue open square) the trend shows that the lower resonant energies are similar in value and in spacing and while the energy is increased the differences between the resonant energies starts to shift further and further from each other.