

# Computational Physics Project 2

Curtis Rau and Benjamin Brophy  
Michigan State University

In this project we sought to solve the Schrödinger Equation for two interacting electrons in a cylindrically symmetric harmonic oscillator potential. Potentials of this form are useful for simulating quantum dots. Of particular interest are the low lying energy levels for this system which only have known analytical solutions for a few special cases [1]. This paper gives an overview of the process of boiling down the Schrodinger equation to an easily solvable one dimensional differential equation. Then the method of discretizing the differential equation, and translating it into a matrix equation is discussed. The Jacobi Method is introduced as the way in which the eigenvalues were obtained. The Householder Algorithm was used to check the results from Jacobi's Method and both methods were in excellent agreement.

## I. INTRODUCTION

In this project we aim to solve Schrödinger's equation for two electrons in a three dimensional harmonic oscillator well with and without a repulsive Coulomb interaction, assuming that the move in a three dimensional harmonic oscillator potential. Here we have the Schrödinger for two electrons at  $r_1$  and  $r_2$  in such a potential

$$\left[-\frac{\hbar^2}{2m}(\nabla_1^2 + \nabla_2^2) + \frac{1}{2}m\omega^2(r_1^2 + r_2^2) + \frac{ke^2}{|r_1^2 - r_2^2|}\right]\psi(r_1, r_2) = E\psi(r_1, r_2) \quad (1)$$

The Schrödinger equation can be transformed to the radial equation in center of mass frame, where  $r$  represents the distance between electrons, and  $R$  is the position of the center of mass.

$$\left[\left(-\frac{\hbar^2}{m}\frac{1}{r^2}\partial_r r^2 \partial_r + \frac{1}{4}m\omega^2 r^2 + \frac{ke^2}{r}\right) + \left(-\frac{\hbar^2}{4m}\frac{1}{R^2}\partial_R R^2 \partial_R + m\omega^2 R^2 + \frac{\hbar l(l+1)}{R^2}\right)\right]\psi(r)\theta(R) = (E_r + E_R)\psi(r)\theta(R) \quad (2)$$

We are only interested in the radial solution

$$\left(-\frac{\hbar^2}{m}\frac{1}{r^2}\partial_r r^2 \partial_r + \frac{1}{4}m\omega^2 r^2 + \frac{ke^2}{r}\right)\psi(r) = E_r \psi(r) \quad (3)$$

We can make the equation dimensionless which gives

$$\left(-\partial_\rho^2 + \rho^2 + \frac{\beta}{\rho}\right)u(\rho) = \lambda u(\rho) \quad \left(-\partial_\rho^2 + V(\rho)\right)u(\rho) = \lambda u(\rho) \quad (4)$$

## II. METHODS

We discretize the equation as follows, where  $b$  and  $a$  are the end points of our model using  $N$  internal points.

(There are  $N+2$  total including end points)

$$\begin{aligned} u(\rho) &\rightarrow u(\rho_i) = u_i \\ \rho &\rightarrow \rho_i = \rho_0 + ih \\ h &= \frac{b-a}{n+1} \\ a &= 0 \end{aligned} \quad (5)$$

And use the following expression as an approximation of the second derivative:

$$u'' = \frac{u(\rho+h) - 2u(\rho) + u(\rho-h)}{h^2}$$

We can approximate the Schrödinger equation as

$$\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + V_i u_i = \lambda u(\rho_i)$$

In terms of matrices, we have

$$\begin{pmatrix} \frac{2}{h^2} + V_1 & -\frac{1}{h^2} & & 0 \\ -\frac{1}{h^2} & \frac{2}{h^2} + V_2 & -\frac{1}{h^2} & \\ & & \ddots & \\ 0 & & -\frac{1}{h^2} & \frac{2}{h^2} + V_{N-1} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \dots \\ u_N \end{pmatrix} = \lambda \begin{pmatrix} u_1 \\ u_2 \\ \dots \\ u_N \end{pmatrix} \quad (6)$$

We will call the  $N \times N$  matrix  $\hat{A}$ . We will perform a Jacobi rotation on this matrix, represented by multiplying both sides by  $S_{ij}^T$ , which as an identity matrix except with  $c = \cos\theta$  in positions  $i, i$  and  $j, j$ ,  $s = \sin\theta$  at position  $i, j$ , and  $-s$  at position  $j, i$  for some angle  $\theta$ .

$$S_{ij}^T = \begin{pmatrix} 1 & & & & 0 \\ & \ddots & & & \\ & & c & \dots & s \\ & & \vdots & & \vdots \\ & & -s & \dots & c \\ & & & & \ddots \\ 0 & & & & & 1 \end{pmatrix} \quad (7)$$

Note that:

- Eigenvalues unchanged under orthogonal transforms such as a Jacobi rotation.
- Eigenvectors change, but the length is preserved.
- Dot product between vectors is conserved

$$\hat{A}x = \lambda x \quad (8)$$

$$S^T A x = \lambda S^T x \quad (9)$$

$$S^T A S(S^T x) = \lambda(S^T x) \quad (10)$$

The values in the new matrix  $\hat{A}' = S_{ij}^T \hat{A}$  can be found with the following relations, where  $s = \sin\theta$  and  $c = \cos\theta$  for some  $\theta$ :

$$\hat{A}'_{ii} = c^2 \hat{A}_{ii} - 2sc \hat{A}_{ij} + s^2 \hat{A}_{jj} \quad (11)$$

$$\hat{A}'_{jj} = s^2 \hat{A}_{ii} + 2sc \hat{A}_{ij} + c^2 \hat{A}_{jj} \quad (12)$$

$$\hat{A}'_{ij} = \hat{A}'_{ji} = (c^2 - s^2) \hat{A}_{ij} + sc(\hat{A}_{ii} - \hat{A}_{jj}) \quad (13)$$

$$\hat{A}'_{ik} = \hat{A}'_{ki} = c \hat{A}_{ik} - s \hat{A}_{jk}, k \neq i, j \quad (14)$$

$$\hat{A}'_{jk} = \hat{A}'_{kj} = s \hat{A}_{ik} - c \hat{A}_{jk}, k \neq i, j \quad (15)$$

$$\hat{A}'_{kl} = \hat{A}_{kl}, k, l \neq i, \quad (16)$$

We performed a Jacobian eigenvalue algorithm on this matrix to find the approximate eigenvalues. If we minimize all of the off-diagonal elements with orthogonal transforms, which preserve eigenvalues, then the remaining diagonal gives us the approximate eigenvalues of the matrix. A Jacobi rotation is one such orthogonal transform.

In our method, we first identify the largest off-diagonal element  $A_{i,j}$  to use as the "pivot" in the Jacobi Rotation. So, we find the angle of rotation such that  $A_{i,j}$  becomes zero. We perform a Jacobi rotation of that angle on  $\hat{A}$  and call the rotated matrix our new  $\hat{A}$ .

We repeat this process until either the largest off-diagonal element has absolute value less than a certain tolerance, 0.01, or until a maximum number of iterations is reached.

After many iterations, the off diagonal elements are minimized while the diagonal elements approach eigenvalues.

### III. ANALYSIS

Our first goal is to find the three lowest eigenstates for the case with no electrostatic potential to four leading digits. We have a few ways to go about this, as seen in the following table. In the case of the Householder method, our answer only depends on how much we discretize our matrix, or what value of N we choose. In this case we need  $N = 300$  in order to obtain the lowest 3 eigenvalues to 4 leading digits. This takes .25 seconds to calculate. In our crafted Jacobian method, the answer depends on both N and our tolerance. As described in our methods, we applied a special Jacobian rotation until the off diagonal elements are reduced below a certain tolerance. In the case of  $N=100$ , although the Householder method will not be accurate to 4 digits, we can obtain four leading correct digits with a tolerance of 0.001. This only takes 0.25 seconds and 13200 iterations to calculate. In the case of  $N=200$ , we can obtain four leading correct digits with a tolerance of 0.01. This only takes 2.9 seconds and 44012 iterations to calculate. In the case  $N=300$ , even though with a high tolerance of 0.1 gives us the eigenvalues within 4 leading digits, this brute force method takes 12.2 seconds compared to the .25 seconds for the Householder method. The number of iterations increases about as the square of N.

### IV. CONCLUSIONS

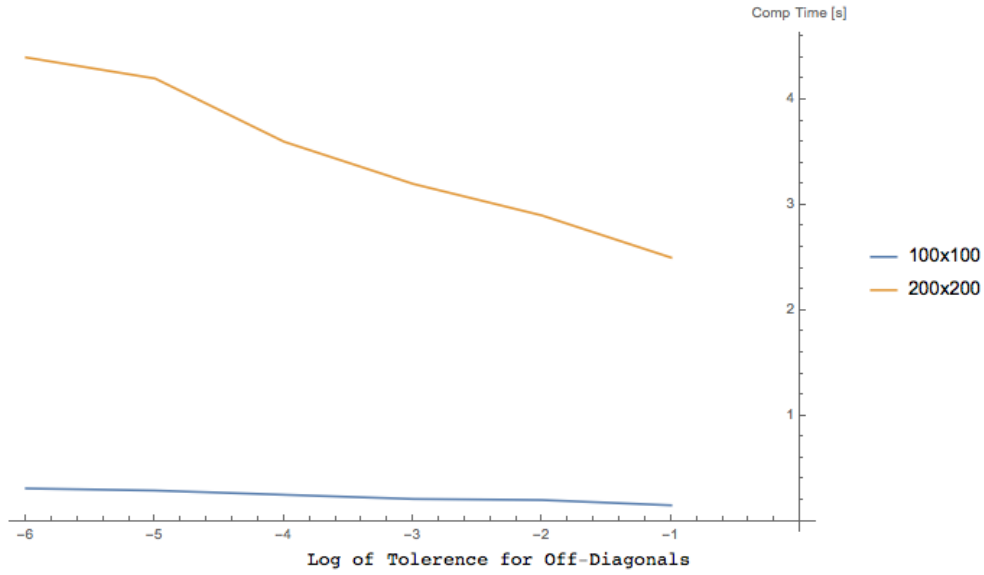
---

[1] M. Taut. Two electrons in an external oscillator potential: Particular analytic solutions of a coulomb cor-

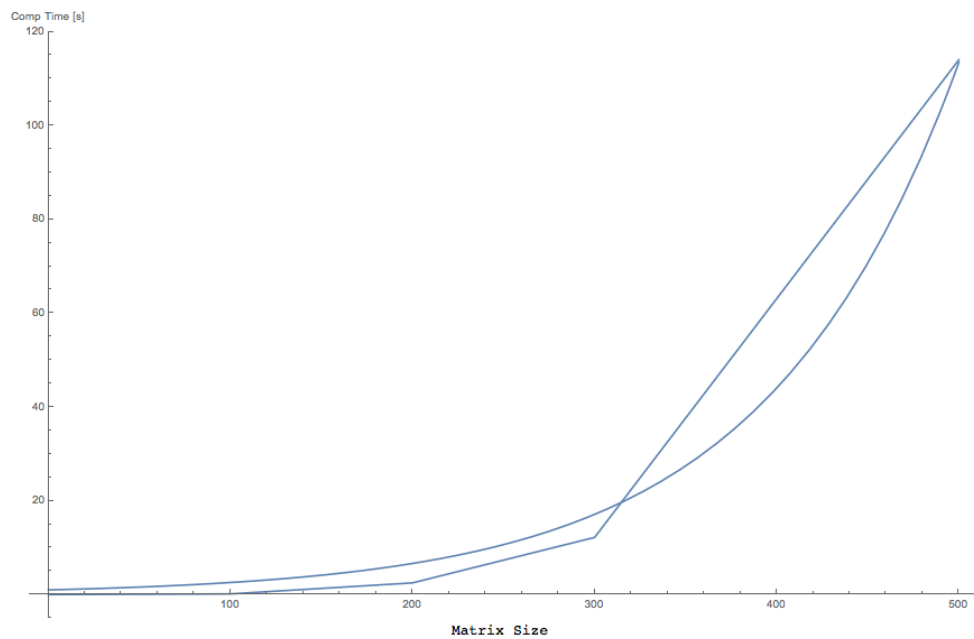
relation potential. *Physical Review A*, 48(5):3561–3566, November 1993.

Size of Matrix	Tolerance	Householder Time	Jacobi Time	Iterations of Jacobi Method	Jacobi E1	Householder E1	Jacobi E2	Householder E2	Jacobi E3	Householder E3
100	0.1	0.01	0.15	8684	3.00030	3.22937	6.96709	6.96536	10.9168	10.91526
"	0.01	"	0.20	10667	2.99924	"	6.96537	"	10.9153	"
"	0.001	"	0.21	12076	2.99923	"	6.96536	"	10.9153	
"	0.0001	"	0.25	13200	"	"				
200	0.1	0.1	2.5	36836	3.00228	2.99986				
"	0.01	"	2.9	44012	2.99981	"				
"	0.001	"	3.2	49387	"	"				
"	0.0001	"	3.6	53888	"	"				
300	0.1	0.38	12.2	85521	3.00024	3.00000				
500	"	2.9	114	246826	3.00047	3.00003	7.00098		11.0005	
				Expected Value:	3	3	7	7	11	11

**TABLE I:** rhoMax = 5.0



**FIG. 1:** A plot of the Jacobi Method's time dependence on off diagonal element tolerance. The plot legend gives the size of the matrix.



**FIG. 2:** A plot of the Jacobi Method's time dependence on matrix size.