Computational Physics - FYS3150
Project 2

Håvard Sinnes Haughom and Ida Risnes Hansen

Sep 30, 2019

Abstract

Differential equations concerning a buckling beam and quantum dots in three dimensions have been re-written as eigenvalue problems. The eigenvalue problems have been solved using an implementation of Jacobi's rotation algorithm. The resulting eigenvalues were equal to analytical eigenvalues to the third digit after the decimal point. In addition we have compared the performance of Jacobi's rotation algorithm to functionality in the C++ library Armadillo (Curtin & Sanderson, 2016). The Jacobi algorithm could not compete, and was, in the worst case, over 12000 times slower than Armadillo.

.

1 Introduction

No doubt, many fond memories have been made while solving for the eigenvalues of well behaved 3x3 matrices in introductory linear algebra. In applications, however, the situation is often not as ideal. For instance, one family of applied eigenvalue problem arises from discretizing second order differential equations. In this case, high dimensional matrices are often necessary for a good result.

While the same techniques, in theory, can be used to find the eigenvalues of 3x3 and 100x100 matrices, in practice, it is necessary to turn to more efficient algorithms for the latter case.

In this project, we will look at the differential equation for a buckling beam, as well as for one and two electrons in an harmonic oscillator potential. All the matrices arising from discretizing these are tridiagonal, and differ only by their diagonals.

Many algorithms exist for approximating the eigenvalues of matrices. We will use one of the simpler ones, Jacobi's rotation algorithm. Here, off-diagonal elements are eliminated, one by one, by similarity transformations that rotate the plane.

For the simplest case, the buckling beam, we assess the performance of the algorithm by comparing it to optimized functions from the Armadillo library. For the two quantum cases, we will look at the accuracy of Jacobi's algorithm by comparing its result to the analytical eigenvalues or approximations.

2 Theory

2.1 Three differential equations

2.1.1 The buckling beam

The vertical displacement $u(x)$ of a beam of length L along $x \in [0, L]$, can be described by the differential equation

$$\gamma \frac{d^2u(x)}{dx^2} = -Fu(x),$$

where F is a force applied at $(L, 0)$ towards the origin and γ is a constant defined by properties of the beam. The beam is fastened to the edges, so that $u(0) = u(L) = 0$. This can be rewritten in terms of a dimensionless variable $\rho \equiv x/L \in [0, 1]$,

$$\frac{d^2u(\rho)}{d\rho^2} = -\lambda u(\rho) \qquad \lambda = FL^2/\gamma$$

The second derivative can then be approximated with stepsize h as

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

and the equation can be discretized

$$-\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} = \lambda u_i \qquad i = 1, 2, ..., N \tag{1}$$

where

$$h = \frac{\rho_N - \rho_0}{N} = \frac{1}{N}.$$

2.1.2 One electron in three dimensions

The radial Schroedinger's equation for an electron in a three-dimensional harmonic oscillator potential $V(r) = kr^2/2$ is

$$-\frac{\hbar^2}{2m}\left(\frac{1}{r^2}\frac{d}{dr}r^2\frac{d}{dr} - \frac{l(l+1)}{r^2}\right)R(r) + \frac{kr^2}{2}R(r) = ER(r) \qquad r \in [0, \infty)$$

where m is the mass and l is the orbital momentum of the electron. Substituting $R(r) = (1/r)u(r)$ and again introducing a dimensionless $\rho = (1/\alpha)r$,

$$-\frac{\hbar^2}{2m\alpha^2}\frac{d^2}{d\rho^2}u(\rho)+\left(\frac{k\alpha^2\rho^2}{2}+\frac{l(l+1)}{\rho^2}\frac{\hbar^2}{2m\alpha^2}\right)u(\rho)=Eu(\rho), \qquad u(0)=0,\, u(\infty)=0$$

For the purposes of this project, we choose $l = 0$, and

$$-\frac{\hbar^2}{2m\alpha^2}\frac{d^2}{d\rho^2}u(\rho)+\frac{k}{2}\alpha^2\rho^2u(\rho)=Eu(\rho)$$
$$-\frac{d^2}{d\rho^2}u(\rho)+\frac{mk}{\hbar^2}\alpha^4\rho^2u(\rho)=\frac{2m\alpha^2}{\hbar^2}Eu(\rho)$$

Finally, by setting

$$\frac{mk}{\hbar^2}\alpha^4=1 \qquad \lambda=\frac{2m\alpha^2}{\hbar^2}E$$

the equation can be written

$$-\frac{d^2}{d\rho^2}u(\rho)+\rho^2u(\rho)=\lambda u(\rho), \qquad \rho\in[0,\infty),$$

or on a discretized form

$$-\frac{u_{i+1}-2u_i+u_{i-1}}{h^2}+\rho_i^2=\lambda u_i. \tag{2}$$

The parameter ρ is in the range $[0, \infty)$, however, in the implementation, it is necessary to pick some finite ρ_N, so that

$$h=\frac{\rho_N-\rho_0}{N}=\frac{\rho_N}{N} \qquad \text{and} \qquad \rho_i=\rho_0+ih$$

2.1.3 Two electrons in three dimensions

Building on the previous case, it is possible to separate the radial Schroedinger equation for two electrons into a center of mass part and a relative part. The relative part is then

$$\left(-\frac{\hbar^2}{m}\frac{d^2}{dr^2}+\frac{1}{4}kr^2+\frac{\beta e^2}{r}\right)u(r)=E_r\psi(r),$$

where the relative coordinate $\mathbf{r}\equiv\mathbf{r}_1-\mathbf{r}_2$, the Coulomb potential $V(r)=\frac{\beta e^2}{r}$ and E_r is the relative energy of the two electrons. Introducing the dimensionless variable $\rho = r/\alpha$, the equation becomes

$$-\frac{d^2}{d\rho^2}u(\rho)+\frac{1}{4}\frac{mk}{\hbar^2}\alpha^4\rho^2u(\rho)+\frac{m\alpha\beta e^2}{\rho\hbar^2}u(\rho)=\frac{m\alpha^2}{\hbar^2}E_ru(\rho).$$

In this case,

$$\omega_r^2 \equiv \frac{1}{4}\frac{mk}{\hbar^2}\alpha^4, \qquad \frac{m\alpha\beta e^2}{\hbar^2} \equiv 1, \qquad \lambda \equiv \frac{m\alpha^2}{\hbar^2}E,$$

and the more manageable Schroedinger equation becomes

$$-\frac{d^2}{d\rho^2}u(\rho) + \omega_r^2\rho^2 u(\rho) + \frac{1}{\rho}u(\rho) = \lambda u(\rho).$$

As before, this can be discretized to the form

$$-\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + (\omega_r^2\rho_i^2 + \frac{1}{\rho_i})u_i = \lambda u_i, \tag{3}$$

with

$$h = \frac{\rho_N}{N} \qquad \text{and} \qquad \rho_i = \rho_0 + ih.$$

2.2 From differential equation to eigenvalue problem

In the cases above, equation (1)-(3), can be rephrased as eigenvalue problems on the form

$$\mathbf{Au} \equiv \begin{bmatrix} d_1 & a & 0 & 0 & \dots & 0 & 0 \\ a & d_2 & a & 0 & \dots & 0 & 0 \\ 0 & a & d_3 & a & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots & a & d_{N-1} & a \\ 0 & \dots & \dots & \dots & \dots & a & d_N \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \dots \\ u_{N-2} \\ u_{N-1} \end{bmatrix} = \lambda \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \dots \\ u_{N-2} \\ u_{N-1} \end{bmatrix} = \lambda\mathbf{u}, \tag{4}$$

where the off-diagonal element $a = -1/h^2$, while the diagonal differs in the three cases.

2.2.1 The buckling beam

For the buckling beam example, the diagonal elements are equal for all rows,

$$d_i = \frac{2}{h^2}, \qquad i = 1, 2, \dots, N.$$

For a Toeplitz matrix like this, the eigenvalues are known analytically,

$$\lambda_i = d + 2a\cos(\frac{i\pi}{N+1}), \qquad i = 1, 2, \dots N.$$

2.2.2 One electron in three dimensions

In the single electron example, the diagonal elements are different.

$$d_i = \frac{2}{h^2} + \rho_i^2 = \frac{2}{h^2} + (ih)^2$$

Here, the first few analytical eigenvalues are

$$\lambda = 3,\ 7,\ 11,\ 15,\ \ldots$$

2.2.3 Two electrons in three dimensions

Finally, for the two electron case

$$d_i = \frac{2}{h^2} + \omega_r^2\rho_i^2 + \frac{1}{\rho_i} = \frac{2}{h^2} + \omega_r^2(ih)^2 + \frac{1}{ih}$$

For small ω_r, the first eigenvalues of the ground state can be approximated (Taut, 1993).

$$\lambda \approx \frac{3}{2}\left(\frac{\omega_r}{2}\right)^{2/3} + \sqrt{3}\omega_r\left(m + \frac{1}{2}\right), \quad m = 0, 1, 2, \ldots$$

2.3 Jacobi's rotation algorithm

The idea behind Jacobi's rotation algorithm, is to perform N orthogonal transformations on A, so that

$$\mathbf{A}_N = \mathbf{P}_{p_N q_N}^T \ldots \mathbf{P}_{p_1 q_1}^T \mathbf{A} \mathbf{P}_{p_1 q_1} \ldots \mathbf{P}_{p_N q_N} \approx \begin{bmatrix} \lambda_1 & 0 & \ldots & \ldots & 0 \\ 0 & \lambda_2 & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \lambda_{N-1} & 0 \\ 0 & \ldots & \ldots & 0 & \lambda \end{bmatrix} \tag{5}$$

For unitary transformations like this, it is possible to show that the eigenvalues are the same for all A_i. In other words, the diagonal in $\mathbf{A}_N$ holds the N eigenvalues of the original matrix A.

After i transformations, let the biggest off-diagonal element in $\mathbf{A}_i$ be in row p_i and column q_i. The next transformation matrix is then on the form

$$\mathbf{P}_{p_{i+q}q_{i+1}} = \begin{bmatrix} & & & \downarrow p_i & & & \downarrow q_i & & \\ & 1 & \dots & \dots & \dots & \dots & \dots & \dots & 0 \\ & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ p_i \rightarrow & \dots & \dots & c & \dots & \dots & s & \dots & \dots \\ & \dots & \dots & \dots & 1 & \dots & \dots & \dots & \dots \\ & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ q_i \rightarrow & \dots & \dots & -s & \dots & \dots & c & \dots & \dots \\ & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ & 0 & \dots & \dots & \dots & \dots & \dots & \dots & 1 \end{bmatrix} \tag{6}$$

where $c \equiv \tan\phi$ and $s \equiv \sin\phi$. This will effectively rotate the plane an angle ϕ and change the elements in the rows and columns q_i and p_i in matrix $\mathbf{A}_i$. In particular, the previously biggest element

$$a_{p_iq_i} = (c^2 - s^2)a_{p_iq_i} + sc(a_{p_ip_i} - a_{q_iq_i}).$$

By setting this to zero, we can solve for the rotation angle ϕ. The process is then repeated until all the elements are below a certain threshold.

At a later iteration j, if the biggest element of A_j is in the column or row $p_j = p_i$ or $q_j = q_i$, the element $a_{p_iq_i}$ that was previously eliminated, might change during the new transformation. This slows down the algorithm. Even so, given enough iterations, matrix A will be close to diagonalized.

3 Method

All of the different cases discussed above have been solved using Jacobi's rotation algorithm. This has been implemented in a C++ script, and the implementation can be found in the GitHub repository HHaughom/FYS3150-Project2.

3.1 The buckling beam

For the buckling beam, we had a Toeplitz matrix with analytic eigenvalues given in (2.2.1). These were used to compare the results given by the implementation of Jacobi's rotation algorithm. We also implemented a solver using the functionality in the C++ Armadillo library (Curtin & Sanderson, 2016), to compare the performance of the different algorithms. All of these methods were run on matrices with dimensions varying from $n = 10$ to $n = 1000$.

3.2 Quantum dots in 3 dimensions

We find the eigenvalues of the systems of 1 and 2 electrons in 3 dimensions.

3.2.1 1 electron

As described in (3), we no longer have a Toeplitz matrix, but a tridiagonal matrix with varying diagonal elements and constant off-diagonal elements. We used Jacobi's algorithm with different values for the dimension of the matrix n and maximum value for the dimensionless parameter ρ. n varied from $n = 10$ to $n = 200$, and ρ_N from $\rho_N = 1$ to $\rho_N = 10$. The results were compared with the analytic eigenvalues given in (2.2.2) to see what values of n and ρ_N gave a result equal to the analytical with a precision of 4 digits after the decimal point.

3.2.2 2 electrons

The case of two electrons in three dimensions is very similar to the previous one with only one electron. The only difference is a change in the diagonal elements of the initial matrix. In addition to running the algorithm for different values of n and ρ_N, we also used different values of the strength of the oscillator potential with $\omega_r = 0.01, 0.5, 1.0, 5.0$. The results are compared with the analytical approximation given in (2.2.3).

4 Results

4.1 The buckling beam

The resulting eigenvalues for the buckling beam are shown in Table 1. For simplicity only the first 10 eigenvalues are shown.

$n = 10$			$n = 100$			$n = 1000$		
Analytic	Armadillo	Jacobi	Analytic	Armadillo	Jacobi	Analytic	Armadillo	Jacobi
8.1014	8.1014	8.1014	9.6743	9.6743	9.6743	9.8499	9.8499	9.8499
31.7493	31.7493	31.7493	38.6881	38.6881	38.6881	39.3994	39.3994	39.3994
69.0279	69.0279	69.0279	87.0130	87.0130	87.0130	88.6484	88.6484	88.6484
116.917	116.917	116.917	154.603	154.603	154.603	157.596	157.596	157.596
171.537	171.537	171.537	241.391	241.391	241.391	246.242	246.242	246.242
228.463	228.463	228.463	347.295	347.295	347.295	354.586	354.586	354.586
283.083	283.083	283.083	472.212	472.212	472.212	482.625	482.625	482.625
330.972	330.972	330.972	616.020	616.020	616.020	630.360	630.360	630.360
368.251	368.251	368.251	778.581	778.581	778.581	797.788	797.788	797.788
391.899	391.899	391.899	959.738	959.738	959.738	984.909	984.909	984.909

Table 1: Resulting eigenvalues obtained by running the analytic-, Armadillo- and Jacobi-solver on the buckling beam problem. The problem is solved with matrices with dimensions $n = 10$, $n = 100$ and $n = 100$, where the dimension is $(n \times n)$. Only the first 10 eigenvalues are shown.

The time used by the different algorithms for the different values of n is shown in Table 2.

	$n = 10$	$n = 100$	$n = 1000$
Analytic	0.0 2ms	0.06 ms	0.72 ms
Armadillo	0.23 ms	9.56 ms	597 ms
Jacobi	1 178 ms	37 689 ms	2 h+

Table 2: Time spent by the different algorithms for different values of n were n is the dimension of the matrix $(n \times n)$. The Jacobi algorithm ran for over 2 hours with $n = 1000$ without finding a result, before being terminated.

4.2 Quantum dots in 3 dimensions

4.2.1 1 electron

In Table 3 you can see the eigenvalues of the system of 1 electron in 3 dimensions. The eigenvalues are found for different values of n and ρ_N. The analytical eigenvalues are given in (2.2.2).

	$n = 10$, $\rho_N = 1$	$n = 100$, $\rho_N = 5$	$n = 200$, $\rho_N = 10$	$n = 400$, $\rho_N = 20$
λ_1	8.4414	2.9992	2.9992	3.0018
λ_2	32.1377	6.9961	6.9969	6.9977
λ_3	69.4244	10.9906	10.9905	10.9926
λ_4	117.3164	14.9865	14.9823	14.9870

Table 3: The first four eigenvalues of the system with 1 electron in 3 dimensions with different values of n and ρ_N. n is the dimension of the matrix $(n \times n)$ (integration points), and ρ_N is the max value of the dimensionless parameter described in (2.1.2). The analytical eigenvalues are 3, 7, 11, 15.

4.2.2 2 electrons

The eigenvalues for the system of 2 electrons in 3 dimensions are shown in Table 4. Here we have used $n = 200$ and $\rho_N = 10$. The strength of the oscillator potential ω_r is varying from $\omega_r = 0.01$ to $\omega_r = 5$. The corresponding analytic eigenvalues are shown in Table 5.

	$\omega_r = 0.01$	$\omega_r = 0.5$	$\omega_r = 1$	$\omega_r = 5$
λ_1	0.3095	2.2299	4.0571	17.4284
λ_2	0.6764	4.1336	7.9059	36.9753
λ_3	1.2124	6.0717	11.8100	56.6183
λ_4	1.9298	8.0267	15.7388	76.2671

Table 4: Eigenvalues of the system of 2 electrons in 3 dimensions with different strength of oscillator potential ω_r. In all runs the dimension of the matrix/integration points is $n = 200$ and the max value for the dimensionless variable ρ is $\rho_N = 10$.

	$\omega_r = 0.01$	$\omega_r = 0.5$	$\omega_r = 1$	$\omega_r = 5$
λ_1	0.1050	2.0566	3.6219	14.1863
λ_2	0.1397	3.7886	7.0860	31.5068
λ_3	0.1743	5.5207	10.5501	48.8273
λ_4	0.2090	7.2527	14.0142	66.1478

Table 5: Analytic eigenvalues for varying oscillator potential strength in the system of 2 electrons in 3 dimensions. Calculated from expressions described in (2.2.3).

5 Discussion

We have divided the discussion section up in to parts. One part regarding the performance of the Jacobi's algorithm and one about the result accuracy in the quantum mechanical cases.

5.1 Performance

The performance of the different algorithms was tested in the case with the buckling beam. The results are shown in Table 2. As you can see in the measurements the analytic solver is clearly fastest. This was also expected since this is only an analytic expression with no iterations.

The more interesting case is to compare the Armadillo solver and the implementation of Jacobi's rotation algorithm. Here we can see that the Armadillo solver heavily outperforms Jacobi's algorithm for all n, especially as n is increasing. With $n = 1000$ we got the eigenvalues in just over half a second with the armadillo solver, while the implementation of Jacobi's algorithm was running for over 2 hours without finalizing. The fact that the Jacobi solver is so slow is as expected because we know that the spent goes like $O(n^3)$. One reason for this is that during the diagonalization of the matrix, we zero out one by one off-diagonal element. In doing so, the elements we have previously zeroed out may become non-zero again, and we have to go back and zero them out again.

5.2 Electron dots in 3 dimensions

5.2.1 1 electron

In the case of 1 electron in 3 dimensions we found the eigenvalues as a function of dimension of matrix/integration points n and maximum value of the dimensionless parameter ρ. The eigenvalues we found with $n = 10$ and $\rho_N = 1$ were far off but already at $n = 100$ and $\rho_N = 5$ we can see that the eigenvalues are getting close to the analytical eigenvalues. At $n = 200$ and $\rho_N = 10$ the eigenvalues were stabilizing close to the analytical ones, but not getting any closer. Increasing n and ρ_N further did not increase the accuracy, and were very time demanding.

We did not achieve the 4 digits after decimal point accuracy that we wanted, but we got 3 digits.

Why we did not get higher accuracy by increasing n and ρ_N further is hard to say. Maybe our tolerance for 0 is to big when we see if the off-diagonal elements in the matrix is 0. Another possibility is that the round-off errors accumulated in every iteration is to big.

5.2.2 2 electrons

For the system of 2 electrons in 3 dimensions we found the eigenvalues numerically. The results we got were not the same as the analytical eigenvalues obtained in (2.2.3). They were of same magnitude and were increasing for higher ω_r in both the analytical and numerical results, but not identical. Since the implementation of the Jacobi algorithm has proven to work in all the previous cases discussed, we assume that the error is in the analytical eigenvalues.

Regardless of were the error lies, the eigenvalues seem to increase with increasing oscillator potential strength ω_r, and qualitatively matches the behaviour of the approximations.

6 Conclusion

Jacobi's rotation algorithm is a brute force method for finding eigenvalues. In practice, this means that, given enough iterations, the algorithm will spit out good approximations to the eigenvalues of non-singular matrices. For the simple, buckling beam case, the analytical eigenvalues of the matrix were reproduced to the third digit after the decimal point with a 100x100 matrix.

However, the condition of *enough iterations* becomes an important caveat. The algorithm is already many orders of magnitude slower than the Armadillo function in the beam case. In the more complicated, but still tridiagonal, quantum cases, the algorithm fails to reproduce the accuracy of the beam case, despite using matrices of dimension in the range 200-400. At this point, the performance of the algorithm becomes debilitating.

The straightforward implementation of Jacobi's algorithm makes it passable for low dimensional matrices. However, for second order magnitude matrices and beyond, one would do well to look for alternatives.

References

[1] Curtin, R. Sanderson, C. (2016) *Armadillo: a template-based C++ library for linear algebra.* Journal of Open Source Software, Vol. 1, pp. 26.

[2] Press, W.H. Teukolsky, S.A. Vetterling, W.T. Flannery, B.P. (2007) *Numerical Recipes: The Art of Scientific Computing* (Section 11.1.) (3rd ed.), New York: Cambridge University Press.

[3] Taut, M. (1993) *Two electrons in an external oscillator potential: Particular solutions of a Coulomb correlation problem* Phys. Rev. A., Vol. 48, pp. 3561.