Numerical Method for Eigenvalue Problem and Its Application in Physics

Mengzhi Chen[1] and Tong Li[1]

[1]Department of Physics and Astronomy, Michigan State University

Abstract

Eigenvalue decomposition of a Hermitian (or real symmetric) matrix is often the last step to solve a physics problem, and thus in this report we investigate different numerical methods for the eigenvalue problem. First two examples, buckling beam problem and quantum dot, are given to illustrate how a differential equation is transformed to an eigenvalue problem of a real symmetric matrix. Then we discuss details of Jacobi's and bisection methods, two important numerical algorithms for the eigenvalue problem. Applications of these two methods on the quantum dot are studied, which shows us when our approximation is good. In addition, we find that bisection method has a better performance and accuracy than Jacobi's. Therefore, bisection algorithm is usually more appropriate to utilize in physics.

1 Introduction

Eigenvalue problem is an important mathematical problem that appears commonly in science and engineering. For matrix $A \in \mathbb{C}^{n\times n}$, its eigenvalues λ_i and corresponding eigenvectors $\mathbf{v}_i \in \mathbb{C}^n$ satisfy

$$A\mathbf{v}_i = \lambda_i \mathbf{v}_i \ . \tag{1}$$

If A is Hermitian (or real symmetric), we can use its eigenvalues and eigenvectors to construct a unitary (or orthogonal) similarity transformation to diagonalize A, i.e.

$$S^\dagger AS = \mathrm{diag}\{\lambda_1, \lambda_2, \cdots, \lambda_n\} \text{ for Hermitian } A \in \mathbb{C}^{n\times n}, \tag{2}$$

$$S^T AS = \mathrm{diag}\{\lambda_1, \lambda_2, \cdots, \lambda_n\} \text{ for symmetric } A \in \mathbb{R}^{n\times n}, \tag{3}$$

where S is a unitary (or orthogonal) matrix and the ith column of S is the eigenvector of A with eigenvalue λ_i.

The diagonalization of a Hermitian (or real symmetric) matrix is usually the final step of a physics problem. In Sec. 2 we give two examples of eigenvalue problems in physics, which shows how a differential equation is transformed to an eigenvalue problem of a real symmetric matrix. Sec. 3 gives two algorithms, Jacobi's methods and bisection method, to solve the eigenvalue problem. In Sec. 4 our numerical solutions are compared with the analytical solution to show under what condition our approximation is valid. We also discuss the performance of the two algorithms we implement in Sec. 4. Conclusions and outlook are given in Sec. 5.

2 Examples of eigenvalue problems in physics

2.1 The buckling beam problem

The buckling beam problem involves the following differential equation:

$$\gamma \frac{d^2u(x)}{dx^2} = -Fu(x), \tag{4}$$

where $u(x)$ is the vertical displacement of the beam in y direction. L is the length of the beam, $x \in [0, L]$ and F is a force applied at $(L, 0)$. The parameter γ is a constant defined by the rigidity of the beam. The boundary condition of this problem is set to be $u(0) = u(L) = 0$.

To scale the equation, we define a dimensional variable

$$\rho = \frac{x}{L},$$

and thus $\rho \in [0, 1]$. Eq. 4 becomes

$$\frac{d^2u(\rho)}{d\rho^2} = -\frac{FL^2}{R}u(\rho) = -\lambda u(\rho), \tag{5}$$

with $\lambda = FL^2/R$. The discretized approximation to function u is defined as u_i on $(N + 1)$ equidistant points $\rho_i \in [0, 1]$, where $\rho_0 = 0$ and $\rho_N = 1$. Thus, the grid spacing is $h = 1/N$ and $\rho_i = ih$. Then we approximate the second derivative as

$$u_i'' = u''(\rho_i) = \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + O(h^2) \tag{6}$$

And thus Eq. 5 for a given value ρ_i can be rewritten as

$$-\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} = \lambda u_i. \tag{7}$$

We can rewrite Eq. 7 as an eigenvalue problem:

$$\begin{bmatrix} d & a & 0 & 0 & \dots & 0 & 0 \\ a & d & a & 0 & \dots & 0 & 0 \\ 0 & a & d & a & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots & a & d & a \\ 0 & \dots & \dots & \dots & \dots & a & d \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \dots \\ u_{N-2} \\ u_{N-1} \end{bmatrix} = \lambda \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \dots \\ u_{N-2} \\ u_{N-1} \end{bmatrix}. \tag{8}$$

where $d = 2/h^2$ and $a = -1/h^2$. Endpoints u_0 and u_N are not included in the matrix. This eigenvalue problem has analytical eigenvalues as

$$\lambda_j = d + 2a\cos\left(\frac{j\pi}{N+1}\right) \quad j = 1, 2, \dots N - 1. \tag{9}$$

which will be used for unit tests in Sec. 3.3.

2.2 Quantum dot in three dimensions

The eigenvalue problem commonly appears in quantum mechanics. For illustration, we consider a quantum dot in a three-dimensional harmonic oscillator potential with frequency ω. At first only one electron

is inside the trap and the system is assumed to be spherically symmetric. The radial part of Schrödinger's equation for one electron

$$-\frac{\hbar^2}{2m}\left(\frac{1}{r^2}\frac{d}{dr}r^2\frac{d}{dr}-\frac{l(l+1)}{r^2}\right)R(r)+V(r)R(r)=ER(r). \tag{10}$$

where l is the quantum number of orbital angular momentum and $V(r)=\frac{1}{2}m\omega^2r^2$ is the harmonic oscillator potential. The energies of the system is

$$E_{nl}=\hbar\omega\left(2n+l+\frac{3}{2}\right), \tag{11}$$

with $n=0,1,2,\ldots$ and $l=0,1,2,\ldots$ By substituting $R(r)=u(r)/r$ we obtain a more compact form of equation:

$$-\frac{\hbar^2}{2m}\frac{d^2}{dr^2}u(r)+\left(\frac{1}{2}m\omega^2r^2+\frac{l(l+1)}{r^2}\frac{\hbar^2}{2m}\right)u(r)=Eu(r). \tag{12}$$

The boundary conditions are $u(0)=0$ and $u(\infty)=0$.

For simplicity, we only discuss s wave, i.e. $l=0$ case. By defining

$$\alpha=\sqrt{\frac{\hbar}{m\omega}},\quad \rho=\frac{r}{\alpha},$$

Eq. 12 can be written in a simpler form as

$$-\frac{d^2}{d\rho^2}u(\rho)+\rho^2u(\rho)=\lambda u(\rho), \tag{13}$$

with

$$u(\rho=0)=0,\quad u(\rho\to\infty)=0.$$

Here $\lambda=\frac{2E}{\hbar\omega}$ and the centrifugal term disappears as $l=0$. The analytical solutions of λ are $\lambda_0=3,\ \lambda_1=7,\ \lambda_2=11,\ldots$

Since it is impossible to go to infinity when we numerically solve Eq. 13, we choose a maximum radius $\rho_{\max}$ and set $u(\rho_{\max})=0$. Similar to the bulking beam problem discussed in Sec. 2.1, we can define the discretized approximation to function u as u_i on $(N+1)$ equidistant points $\rho_i\in[0,\rho_{\max}]$, where $\rho_0=0$ and $\rho_N=1$. Thus, the grid spacing is $h=1/N$ and $\rho_i=ih$. Then Eq. 13 can be discretized and approximated as

$$-\frac{u_{i+1}-2u_i+u_{i-1}}{h^2}+\rho_i^2u_i=\lambda u_i, \tag{14}$$

where ρ_i^2 comes from the harmonic oscillator potential. Eq. 14 can be rewritten as an eigenvalue problem of a matrix:

$$\begin{bmatrix} d_1 & e_1 & 0 & 0 & \ldots & 0 & 0\\ e_2 & d_2 & e_2 & 0 & \ldots & 0 & 0\\ 0 & e_3 & d_3 & e_3 & 0 & \ldots & 0\\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots\\ 0 & \ldots & \ldots & \ldots & \ldots e_{N-2} & d_{N-2} & e_{N-2}\\ 0 & \ldots & \ldots & \ldots & \ldots & e_{N-1} & d_{N-1}\end{bmatrix}\begin{bmatrix}u_1\\u_2\\\ldots\\\ldots\\\ldots\\u_{N-1}\end{bmatrix}=\lambda\begin{bmatrix}u_1\\u_2\\\ldots\\\ldots\\\ldots\\u_{N-1}\end{bmatrix}. \tag{15}$$

where

$$d_i=\frac{2}{h^2}+\rho_i^2,\quad e_i=-\frac{1}{h^2}. \tag{16}$$

When two electrons are put inside the harmonic oscillator potential, the radial part of Schrödinger's equation can be written as

$$\left(-\frac{\hbar^2}{m}\frac{d^2}{dr^2}-\frac{\hbar^2}{4m}\frac{d^2}{dR^2}+\frac{1}{4}m\omega^2r^2+m\omega^2R^2+\frac{\beta e^2}{r}\right)u(r,R)=Eu(r,R). \tag{17}$$

where $\mathbf{R}$ is the center of mass coordinate and $\mathbf{r}$ is the relative coordinate. $\frac{\beta e^2}{r}$ is the Coulomb repulsion between two electrons, with $\beta e^2 = 1.44\ \text{eV}\cdot\text{nm}$. Eq. 17 can be separated into two parts:

$$\left(-\frac{\hbar^2}{m}\frac{d^2}{dr^2}+m\omega^2R^2\right)\chi(R)=E_R\chi(R), \tag{18}$$

and

$$\left(-\frac{\hbar^2}{m}\frac{d^2}{dr^2}+\frac{1}{4}m\omega^2r^2+\frac{\beta e^2}{r}\right)\psi(r)=E_r\psi(r), \tag{19}$$

where

$$E=E_R+E_r,\quad u(r,R)=\chi(R)\psi(r). \tag{20}$$

Similar to the one-electron case, we define a natural length

$$\alpha'=\frac{\hbar^2}{m\beta e^2}$$

and let $\rho = r/\alpha'$. Then Eq. 19 can be rewritten as

$$-\frac{d^2}{d\rho^2}\psi(\rho)+\omega_r^2\rho^2\psi(\rho)+\frac{1}{\rho}=\lambda\psi(\rho), \tag{21}$$

where

$$\omega_r^2=\frac{m^2\omega^2}{4\hbar^2}\alpha'^4,\quad \lambda=\frac{m\alpha'^2}{\hbar^2}E_r. \tag{22}$$

Similar to Eq. 13, Eq. 21 can be numerically solved by transforming it into an eigenvalue problem

$$\begin{bmatrix} d_1 & e_1 & 0 & 0 & \dots & 0 & 0 \\ e_2 & d_2 & e_2 & 0 & \dots & 0 & 0 \\ 0 & e_3 & d_3 & e_3 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots & \dots e_{N-2} & d_{N-2} & e_{N-2} \\ 0 & \dots & \dots & \dots & \dots & e_{N-1} & d_{N-1} \end{bmatrix} \begin{bmatrix} \psi_1 \\ \psi_2 \\ \dots \\ \dots \\ \dots \\ \psi_{N-1} \end{bmatrix} = \lambda \begin{bmatrix} \psi_1 \\ \psi_2 \\ \dots \\ \dots \\ \dots \\ \psi_{N-1} \end{bmatrix}, \tag{23}$$

where

$$d_i=\frac{2}{h^2}+\omega_r^2\rho_i^2+\frac{1}{\rho_i},\quad e_i=-\frac{1}{h^2}. \tag{24}$$

3 Numerical methods

In order to solve the eigenvalue problems that occur naturally in physics, we will investigate two relevant numerical algorithms (Jacobi's method and bisection method) in this section.

3.1 Jacobi's method

Jacobi's method gives a simple algorithm to diagonalize a real symmetric (or Hermitian) matrix. It is based on Givens rotation

$$G(k,l,\theta) = \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & c & \cdots & s & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & -s & \cdots & c & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix} \tag{25}$$

where $c = \cos\theta$ and $s = \sin\theta$ appear at the intersections of kth and lth rows and columns. Therefore, for $k > l$, non-zero matrix elements of $G(k,l,\theta)$ are given by

$$\begin{aligned} g_{ii} &= 1 \quad \text{if } i \neq k, l \ ; \\ g_{kk} &= g_{ll} = c \ ; \\ g_{kl} &= -g_{lk} = -s \ . \end{aligned}$$

We can see that $G(k,l,\theta)$ is an orthogonal matrix.

Let $A \in \mathbb{R}^{n\times n}$ be symmetric. Our purpose of orthogonal similarity transformation

$$B = G(k,l,\theta)^T A G(k,l,\theta) \tag{26}$$

is to eliminate non-zero off-diagonal elements $a_{kl} = a_{lk}$. To achieve this, we should have $0 = b_{kl} = a_{kl}(c^2 - s^2) + (a_{ll} - a_{kk})cs$, and thus $t = \tan\theta$ should be the solution of

$$t^2 + 2\tau t - 1 = 0 \tag{27}$$

where

$$\tau = \frac{a_{kk} - a_{ll}}{2a_{kl}}. \tag{28}$$

Eq. 27 yields

$$t = -\tau \pm \sqrt{1+\tau^2}. \tag{29}$$

Then c and s can be easily obtained via

$$\begin{aligned} c &= \frac{1}{\sqrt{1+t^2}} \ , \\ s &= tc \ . \end{aligned} \tag{30}$$

Usually we choose the smaller root of Eq. 29, which makes $|\theta| \leq \frac{\pi}{4}$ and minimizes the Frobenius norm [1] of $(B - A)$.

Based on Givens rotation, the main idea of Jacobi's method is to eliminate the off-diagonal element with maximum absolute value by Eq. 25, 26, 29 and 30, until the absolute values of all the off-diagonal elements are less than a specific tolerance. The pseudocode of Jacobi's method is given in Algorithm 1. The elmination can also be done in a cyclic way, which avoids the necessity to find the maximum

[1] The Frobenius norm of matrix $A \in \mathbb{C}^{n\times n}$ is defined as

$$||A||_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} |a_{ij}|^2} \ .$$

It is conserved in unitary (or orthogonal) similarity transformation.

Input: Symmetric matrix $A \in \mathbb{R}^{n\times n}$, tolerance and maxiter.
Output: Eigenvalues in vector **u** and corresponding eigenvectors in columns of matrix $S \in \mathbb{R}^{n\times n}$.

S=identity matrix;
while maxnondiag > tolerance && iterations <= maxiter **do**
  iterations++;
  Find the maximum non-diagonal element a_{kl} $(k > l)$ in matrix A;
  Calculate t, c and s via Eq. 29 and 30 and obtain $G(k, l, \theta)$;
  $A = G(k, l, \theta)^T AG(k, l, \theta)$ (Eq. 26);
  $S = SG(k, l, \theta)$;
end
for $i = 1; i <= n; i++$ **do**
  $u_i = a_{ii}$;
end
return **u**,S;

Algorithm 1: Jacobi's method for diagonalization of symmetric matrix $A \in \mathbb{R}^{n\times n}$.

off-diagonal element in each iteration. The implementation can be simply achieved by substituting the maximum off-diagonal searching element kernel in the classical Jacobi method with a loop over $A's$ elements.

3.2 Bisection eigenvalue algorithm

As we are usually interested in the symmetric tridiagonal eigenproblem, the bisection method can be applied. The eigen-polynomial of a n-by-n tridiagonal matrix

$$A = \begin{bmatrix} a_1 & b_1 & 0 & \cdots & \cdots & 0 \\ b_1 & a_2 & b_2 & 0 & \cdots & 0 \\ 0 & b_2 & a_3 & b_3 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & b_{n-2} & a_{n-1} & b_{n-1} \\ 0 & \cdots & \cdots & 0 & b_{n-1} & a_n \end{bmatrix} \tag{31}$$

satisfies a general recurrence relation

$$p_r(x) = (a_r - x)p_{r-1}(x) - b_{r-1}^2 p_{r-2}(x) \tag{32}$$

where $r = 2 : n$, $p_0(x) = 1$ and $p_1(x) = a_1$.

The Sturm sequence property[2] reveals the relation between eigenvalues neighboring eigen-polynomials. The relation is

$$\lambda_r(A_r) < \lambda_{r-1}(A_{r-1}) < \lambda_{r-1}(A_r) < \cdots < \lambda_2(A_r) < \lambda_1(A_{r-1}) < \lambda_1(A_r). \tag{33}$$

. Moreover, if $a(\alpha)$ denotes the number of sign changes in sequence

$$\{p_0(\alpha), p_1(\alpha), \cdots, p_n(\alpha)\},$$

then $a(\alpha)$ equals the number of p's eigenvalues that are less than α. Also the Gershgorin theorem fixes the lower and upper bounds of eigenvalues. The $\lambda_k(p) \in [y, z]$ where

$$y = \min_{1\le i\le n} a_i - |b_i| - |b_{i-1}|, \quad z = \max_{1\le i\le n} a_i + |b_i| + |b_{i-1}|. \tag{34}$$

Input: Diagonal vector **a**, off-diagonal vector **b**, tolerence *tol*
Output: Eigenvalues in vector **u** and corresponding eigenvectors in columns of matrix $S \in \mathbb{R}^{n\times n}$.

Calculate lower bound y and upper bound z via Eq. 34;
// Determine subintervals;
Calculate $p_n(y), p_n(z), a(y)$ and $a(z)$;
while $a(z) - a(y) > 1$ **do**
    bisectional searching for bounds of subintervals and store them in vector $subint(n+1)$;
end
// Apply the bisection method to every subintervals;
for $i = 0; i < n; i++$ **do**
    $y = subint(i)$, $z = subint(i+1)$;
    while $rmax - rmin > tol$ **do**
        $x = (y+z)/2$;
        if $p_n(x) * p_n(y) < 0$ **then**
            $z = x$;
        end
        else
            $y = x$;
        end
    end
    $u(i) = x$;
end
// Calculate correlated eigenvectors;
for $i = 0; i < n; i++$ **do**
    solve $(A - u(i)I)S(i,:) = 0$;
end

Algorithm 2: The bisection method for diagonalization of symmetric tridiagonal matrix $A \in R^{n\times n}$.

During the execution of the bisection method, information about the location of other eigenvalues is obtained. By keeping track of this information it is possible to extract subintervals for every eigenvalues. See [7]. Then, we can apply the bisection method for each subinterval to yield corresponding eigenvalue. In addition, in this method, we allocate space only for tridiagonal elements as others are not referred. It saves us a lot of memory which is crucial for huge size calculations.

3.3 Unit tests for development

As is stated in Sec. 2.1, the eigenvalue problem of Eq. 8 has an analytical expression (Eq. 9). Therefore, we employ Eq. 9 to construct a unit test for our eigenvalue solvers during the development, which is done with the help of Catch2 library. In addition, the conservation of Frobenius norm and the orthogonality of eigenvectors are also tested to confirm the correctness of our codes.

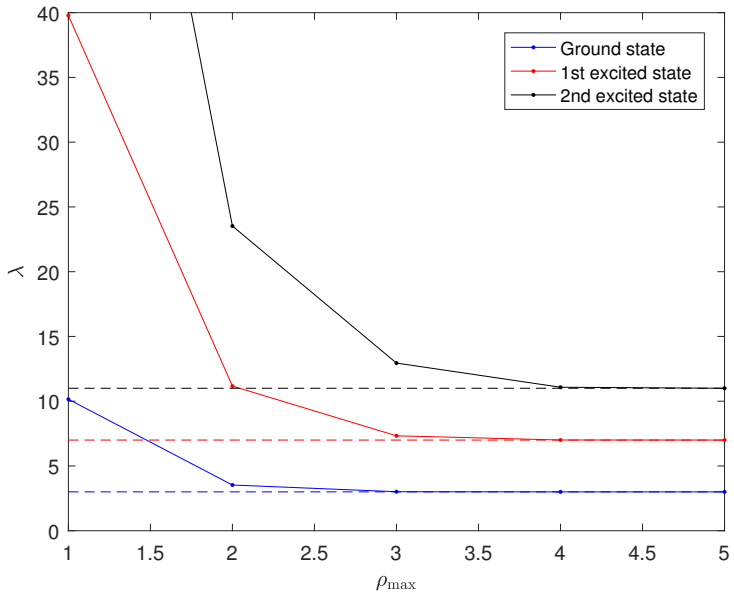

Figure 1: Relation between $\rho_{\max}$ and the lowest three eigenvalues of quantum dot with one electron inside a harmonic oscillator potential. Grid spacing h is fixed to be 0.01. Analytical values are given by dashed lines.

4 Results and discussion

4.1 Results of the quantum dot problem

In this section we first discuss the results of quantum dot with one electron (Sec. 2.2), which are obtained from Jacobi's method (Sec. 3.1) with different $\rho_{\max}$ and grid points N. Tolerance of the maximum off-diagonal element is 10^{-5}.

The relation between $\rho_{\max}$ and the lowest three eigenvalues of quantum dot with one electron inside a harmonic oscillator potential is shown in Fig. 4.1. Grid spacing h is fixed to be 0.01. The numerical solutions of eigenvalues decrease quickly as $\rho_{\max}$ increases, and these values approach analytical solutions when $\rho_{\max}$ is larger than 4. When we fix $\rho_{\max} = 4$ and change grid number N, as shown in Table 1, the numerical solutions approach their analytical values gradually as N increases, and $N = 100$ is usually good enough for this problem.

Now we start to discuss the results of quantum dot with two electrons and Coulomb repulsion.

4.2 Performance comparison of different methods

To test the performances of the cyclic Jacobi and the bisection method, we use these methods to diagonalize the matrix 8. Treating Eq. 9 as a standard benchmark, we adjust the error tolerances in algorithms to reproduce first five eigenvalues deviates smaller than 10^{-5} form their analytic values. The execution time and speedup for different algorithms and sizes are shown in table 2.

First of all, we see the speedup of cyclic Jacobi method increases with growing sizes, because it cuts off the searching time which is proportional to the number of iterations, as well as the matrix size. Generally speaking, the promotion we got from the cyclic Jacobi method is acceptable. However, both these Jacobi's methods take a lot of time dealing with the matrix whose size is larger than 1,000. At that

Table 1: Relation between different grid number N and the lowest three eigenvalues of quantum dot with one electron inside a harmonic oscillator potential. $\rho_{\max}$ is fixed to be 4.

N	Ground state	1st excited state	2nd excited state
10	2.9491	6.7428	10.3971
20	2.9875	6.9402	10.9154
40	2.9969	6.9877	11.0383
80	2.9992	6.9995	11.0687
100	2.9995	7.0009	11.0724
Analytical	3	7	11

Table 2: Execution times and speedups for the classical Jacobi, cyclic Jacobi and bisection method for different inout matrix sizes.

Size	Classical Jacobi	Cyclic Jacobi		Bisection Method	
	Execution time (ms)	Execution time (ms)	Speedup	Execution time (ms)	Speedup
50	8.817	4.189	2.105	1.084	8.132
100	82.845	26.696	3.103	2.819	29.393
200	1340.239	208.239	6.436	7.845	170.833
500	35269.687	5085.778	6.935	34.520	1021.717
1000	···	···	···	119.494	···
2000	···	···	···	418.566	···

time, the bisection method shows its superiority. We see its speedup is super-linear and the execution time is in control as the size growing up to 2,000.

5 Conclusions and outlook

Wait to be filled.

Acknowledgments

We are grateful for the sincere guidance from Prof. Morten Hjorth-Jensen.

References

[1] Github - catchorg/Catch2: A modern, C++-native, header-only, test framework for unit-tests, TDD and BDD - using C++11, C++14, C++17 and later (or C++03 on the Catch1.x branch). `https://github.com/catchorg/Catch2`. Accessed Mar 5, 2018.

[2] Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.

[3] Morten Hjorth-Jensen. Computational physics lectures: Eigenvalue problems. `https://compphysics.github.io/ComputationalPhysicsMSU/doc/pub/eigvalues/html/eigvalues.html`. Accessed Mar 5, 2018.

[4] Morten Hjorth-Jensen. Project 2. `https://compphysics.github.io/ComputationalPhysicsMSU/doc/Projects/2018/Project2/html/Project2.html`. Accessed Mar 5, 2018.

[5] Conrad Sanderson and Ryan Curtin. Armadillo: a template-based C++ library for linear algebra. *Journal of Open Source Software*, 2016.

[6] Lloyd N Trefethen and David Bau III. *Numerical linear algebra*. Siam, 1997.

[7] JH Wilkinson. Calculation of the eigenvalues of a symmetric tridiagonal matrix by the method of bisection. *Numerische Mathematik*, 4(1):362–367, 1962.