Assignment 2: Eigenvalue problems I

deadline Tuesday April 5 2022

When we want to treat a quantum mechanical system, we usually have to solve an eigenvalue problem

$$H\Psi = E\Psi, \tag{1}$$

where H is the Hamilton operator. For a one dimensional problem;

$$H = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + V(x). \tag{2}$$

As a test case we will use the one dimensional harmonic oscillator Hamiltonian. This problem can be solved analytically and we will use that to check our numerical approach. It is a good idea to read through the chapter on the harmonic oscillator in your quantum mechanics textbook. The importance of the harmonic oscillator problem stems from the fact that whenever there is a local potential minimum the harmonic oscillator model gives the first approximation to the physics. If the potential $V(x)$ has a minimum in $x = x_0$ we can expand it in a Taylor series around the minimum;

$$\begin{aligned} V(x) &= V(x_0) + V'(x_0)(x - x_0) + \frac{1}{2}V''(x_0)(x - x_0)^2 + ... \\ \Rightarrow V(x) - V(x_0) &\approx \frac{1}{2}k(x - x_0)^2, \end{aligned} \tag{3}$$

where we have used that $V'(x_0) = 0$ since $x = x_0$ is a minimum. We have further put $k = V''(x_0)$, where $V''(x_0) \geq 0$ (which follows since we are in a minimum). A classical harmonic oscillator (a spring for example) is governed by a restoring force $F = -k(x - x_0)$ and its potential energy is $V(x) = k(x-x_0)^2/2$, i.e. identical to the last equality in Eq. 3 if we subtract the constant background potential. Such an oscillator will oscillate with period $T = 2\pi/\omega$, with $\omega = \sqrt{(k/m\ )}$. Using ω instead of k, we can write the Hamilton operator as:

$$H = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + \frac{1}{2}m\omega^2 x^2$$

where we have put x_0 in the origin. There are several ways to solve the eigenvalue equation (Eq. 1) with this Hamiltonian. We shall start with a so called finite difference approach.

In principle there is an infinite number of eigenstates, $\Psi(x)$, to H, and these can extend to $x \to \pm\infty$. However, we are usually only interested in a finite number of states. We search for states with rather low energy and which essentially are confined to a region of space close to $x = 0$.

• Span a space which you think is appropriate for what you are interested in with a linear grid from xmin to xmax. Outside these boundaries you will assume that the wave function is zero. In principle this means that you have put your harmonic oscillator in a potential well. At the well boundaries we assume that the potential goes to infinity, and thus the eigenstates have to go to zero there.

• Discretize the kinetic energy term with a five-point formula, i.e. use that fact that

$$\begin{aligned} f''(x) &= c_{-2h}f(x - 2h) - c_{-h}f(x - h) + c_{0h}f(x) + c_h f(x + h) + c_{2h}f(x + 2h) + O(h^4) \\ &= -\frac{1}{12h^2}\left(f(x - 2h) - 16f(x - h) + 30f(x) - 16f(x + h) + f(x + 2h)\right) + O(h^4). \end{aligned} \tag{4}$$

If you have forgotten how this relation is obtained you should repeat the section on finite difference methods in L. Edsberg "Introduction to computation and modelling for Differential equations", or some other textbook on numerical methods. You can now set up the *matrix eigenvalue equation.*

$$\mathbb{H}\mathbb{X} = E\,\mathbb{X},$$

where

$$\mathbb{H} = \begin{pmatrix} c_{0h}+V(x_1) & c_h & c_{2h} & 0 & 0 & 0 & 0 & 0 & 0 \\ c_{-h} & c_{0h}+V(x_2) & c_h & c_{2h} & 0 & 0 & 0 & 0 & 0 \\ c_{-2h} & c_{-h} & c_{0h}+V(x_3) & c_h & c_{2h} & 0 & 0 & 0 & 0 \\ 0 & c_{-2h} & c_{-h} & c_{0h}+V(x_4) & c_h & c_{2h} & 0 & 0 & 0 \\ 0 & 0 & c_{-2h} & c_{-h} & c_{0h}+V(x_5) & c_h & c_{2h} & 0 & 0 \\ . & . & . & & & & & & \\ . & . & . & & & & & & \\ . & . & . & & & & & & \end{pmatrix}$$

and

$$\mathbb{X} = \begin{pmatrix} f(x_1) \\ f(x_2) \\ f(x_3) \\ f(x_4) \\ . \\ . \\ . \end{pmatrix}$$

Note that the matrix above is banded. Since $c_{nh} = c_{-nh}$ it is also symmetric

H is an Hermitian operator. We know then that eigenstates corresponding to different eigenvalues should be orthogonal, i.e.

$$\int \Psi_i^*(x)\Psi_j(x)\,dx = \delta_{ij},$$

where normalization is assumed. Further the eigenstates to a Hermitian operator form a complete set. The matrix $\mathbb{H}$ we just obtained by discretization of H is an Hermitian matrix (this means that it is *self adjoint*; $H = H^\dagger = (H^T)^*$) and its eigenvectors will be orthogonal in a similar manner

$$\mathbb{X}_i^\dagger \mathbb{X}_j = \delta_{ij},$$

when $E_i \neq E_j$. Our matrix above belongs actually to the subclass of Hermitian matrices that are real and symmetric. Then also the eigenvectors $\mathbb{X}_i$ will be real.

If $\mathbb{H}$ is an $n \times n$ matrix than there will be n eigenvectors. These eigenvectors, as the eigenvectors to the Hamilton operator, form a complete set. The difference is though that the eigenfunction to the *operator* H can span any function $\Phi(x)$ defined anywhere on the real x-axis, the finite set of eigenstates to our *matrix* $\mathbb{H}$ can span any function defined on our grid.
There are several methods to solve the matrix eigenvalue equation.
• *Diagonalize* and find all eigenvalues and (optionally) all eigenvectors. The scheme is then to find the similarity transformation of the matrix $\mathbb{H}$ such that

$$\mathbb{X}^{-1}\mathbb{H}\mathbb{X} = \mathbb{D}, \tag{5}$$

where $\mathbb{D}$ is a diagonal matrix. The eigenvalues are now found at the diagonal of $\mathbb{D}$ and the eigenvectors are the columns of $\mathbb{X}$, see further A. Ruhe Topics in Linear Algebra. There are standard routines for diagonalization available (e.g. in the LAPACK package). A method that uses the fact that the matrix is banded and symmetric is much faster than a general diagonalizer. LAPACK provides specialized routines for a number of matrix-types. The input to such a routine is generally only the non-zero elements of the matrix. Matlab judges itself how to diagonalize a matrix.
• Another possibility is to find a few eigenvalues (the lowest, the highest, in a specified energy region) and their eigenvector with an *iterative* method. This type of methods are often faster when you want

a particular solution and have a sparse matrix. One such method is the "(Inverse) Power Iteration" method. More advanced methods of this kind are the Algorithm Arnoldi or the Lanczos algorithm (which can handle truly big matrices). See further A. Ruhe Topics in Linear Algebra.

• Your tasks:
It is expected that you solve this Assignment in a low level language (C/C^{++}/Fortran). You will then use LAPACK routines directly, and can compare the performances of routines for banded matrices and for general matrices. If you think this is hard; write your routine first in (for example) MatLab, and then "translate" to the low level language. In this way you can also check your program at intermediate levels.

1. Use the finite difference approach described above to construct the Hamiltonian matrix for the one dimensional Harmonic oscillator. Diagonalize the matrix with a standard routine (LAPACK) and check that you get the expected results. As always it is a good idea to check that you really have found an eigenvector by inserting it into the eigenvalue equation. (Be careful, many standard routines destroys the original matrix).

2. Write an "Inverse Power Iteration with shift routine" to obtain the first few eigenvalues and eigenvectors to the matrix (see below for explanation). Compare with the standard routine results (LAPACK). Check how time consuming the two methods are. You might need to increase the size of the matrix to clearly see the difference in time.

3. Compare the result from your calculation with the analytical solutions to the Harmonic oscillator problem in some detail. Remember that you have at least two aspects to consider. The second order derivative is approximated with a finite difference formula. This gives a specific error. In addition you force the solution to be zero outside the last point in your grid.

4. Check the properties of your solutions also in other respects. Are they orthogonal to each other? normalized?

5. Consider a slightly different Hamiltonian, the harmonic oscillator Hamiltonian with an extra potential term added

$$H = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + \frac{1}{2}m\omega^2 x^2 + V(x) = H_\omega + V(x), \tag{6}$$

where

$$V(x) = C_1 e^{-x^2 C_2}, \tag{7}$$

and C_1 and C_2 are numerical constant. The extra potential is thus a "bump" in the middle of the harmonic oscillator potential. Choose some values of C_1 and C_2 so that the lowest energy solutions are such that a particle in these states have to tunnel through the "bump" in order to go from one side of the well to the other. Compare the wave functions for the first few solutions with and without the bump. You can do that by plotting them, but also by looking at the overlap integral. This integral is easy to calculate if you chose the same grid for the two problems.

Appendix

Power Iteration; an example of an Iterative method

If we want to find the largest eigenvalue to a $n \times n$ matrix $\mathbb{A}$, we can proceed as follows. Take any vector of length n as a first guess, we call this first guess $\mathbb{Y}_1$. Sine the (still unknown) eigenvectors to $\mathbb{A}$ form a complete set, $\mathbb{Y}$ can be formally expanded in them

$$\mathbb{Y}_1 = \sum_n c_n \mathbb{X}_n. \tag{8}$$

Applying $\mathbb{A}$ from the left we get

$$\mathbb{Y}_2 = \mathbb{A}\mathbb{Y}_1 = \sum_n c_n \lambda_n \mathbb{X}_n. \tag{9}$$

where λ_n is the eigenvalue associated with eigenvector $\mathbb{X}_n$. We can apply $\mathbb{A}$ from the left a second time;

$$\mathbb{Y}_3 = \mathbb{A}\mathbb{Y}_2 = \sum_n c_n \lambda_n^2 \mathbb{X}_n, \tag{10}$$

and so on. After sufficiently many applications of $\mathbb{A}$ from the left the term in the sum with the largest absolute eigenvalue will dominate completely and

$$\frac{\mathbb{Y}_i^\dagger \mathbb{A}\mathbb{Y}_i}{\mathbb{Y}_i^\dagger \mathbb{Y}_i} \Rightarrow max \mid \lambda_n \mid, \text{when } i \Rightarrow \infty. \tag{11}$$

It is a good idea to normalize $\mathbb{Y}_i$ in every step, i.e. to keep $\mathbb{Y}_i^\dagger \mathbb{Y}_i = \mathbb{I}$. This is a simple and robust method. As long as the initial guess has some overlap with the eigenvector associated with the largest eigenvalue convergence will be achieved.

Inverse Power Iteration

For solutions of the Schrödinger equation we are , however, more interested in the smallest eigenvalue than in the largest. For this we use the Inverse iteration scheme which formally corresponds to the performance of power iteration for $\mathbb{A}^{-1}$. Since the computation of the inverse of a matrix is as time-consuming as the full eigenvalue problem the practical calculation is, however, through a different path. Note first that the eigenvalues of $\mathbb{A}^{-1}$ are λ_n^{-1}, if λ_n are the eigenvalues to $\mathbb{A}$.

$$\mathbb{A}\mathbb{X}_n = \lambda_n \mathbb{X}_n \Longleftrightarrow \lambda_n^{-1} \mathbb{X}_n = \mathbb{A}^{-1} \mathbb{X}_n. \tag{12}$$

We will now solve the *system of linear equations*

$$\mathbb{A}\mathbb{Y}_2 = \mathbb{Y}_1, \tag{13}$$

where $\mathbb{Y}_1$ is our first guess for an eigenvector. In the next step we put the solution, $\mathbb{Y}_2$ on the right-hand side, and solve again, i.e. we have an iterative scheme;

$$\mathbb{A}\mathbb{Y}_{i+1} = \mathbb{Y}_i. \tag{14}$$

To analyze the situation we note again that any vector can be expanded in eigenvectors to the matrix. After the first step we have for example;

$$\mathbb{Y}_2 = \mathbb{A}^{-1}\mathbb{Y}_1 = \mathbb{A}^{-1} \sum_n c_n \mathbb{X}_n = \sum_n \lambda_n^{-1} c_n \mathbb{X}_n. \tag{15}$$

It is clear that in the iterative procedure the solution $\mathbb{Y}_{i+1}$ will converge towards the eigenvector with the largest value of λ_n^{-1}, i.e. towards the eigenvector with the smallest eigenvalue. At every step in the iteration the current approximation of the inverse of the smallest eigenvalue is given by

$$\frac{\mathbb{Y}_i^\dagger \mathbb{Y}_{i+1}}{\mathbb{Y}_i^\dagger \mathbb{Y}_i} = \frac{\mathbb{Y}_i^\dagger \mathbb{A}^{-1} \mathbb{Y}_i}{\mathbb{Y}_i^\dagger \mathbb{Y}_i} \Rightarrow \frac{1}{min \mid \lambda_n \mid}, \text{when } i \Rightarrow \infty. \tag{16}$$

Also here it is a good idea to normalize in every step. Note finally that at every step in the iteration we solve a system of linear equations, Eq.14. The left-hand side matrix is the same every time, but the right-hand side change. This is a typical situation where it is an advantage to first perform an LU-decomposition, $\mathbb{A} = \mathbb{L}\mathbb{U}$, for fast solutions in the following iterations.

Inverse Power Iteration with Shift

If we shift the matrix with a scalar constant,ξ,

$$\mathbb{A} \Rightarrow \mathbb{A} - \xi\mathbb{I}, \tag{17}$$

the power iterations will converge to the largest (smallest) eigenvalue of the shifted matrix i. e to $max \mid \lambda_n - \xi \mid$, or to $1/min \mid \lambda_n - \xi \mid$. In this way it is possible to find more than one eigenvalue. The shift can also be used to improve the convergence since the rate of convergence depends on $\mid \lambda_1/\lambda_2 \mid$, where λ_1 is the largest and λ_2 is the second largest eigenvalue, or vice versa for the inverse power iteration.