Lamarr Quantum Computing Nuggets

Numerically Solving Schrödinger Equations (2)

Christian Bauckhage
Lamarr Institute for ML and AI
University of Bonn
Fraunhofer IAIS

ABSTRACT

We revisit the problem of numerically solving the Schrödinger equation for a one-dimensional quantum harmonic oscillator. We reconsider our previous finite difference scheme and discuss how higher order finite differences can lead to more accurate solutions. In particular, we will consider a five point stencil to approximate second order derivatives and implement the approach using *SciPy* functions for sparse matrices.

1 INTRODUCTION

Previously [2], we considered a simple numerical scheme for solving the time-independent Schrödinger equation for a one-dimensional quantum harmonic oscillator (in atomic units with $m = 1, \hbar = 1$)

$$\left(-\frac{1}{2}\,\frac{d^2}{dx^2} + \frac{1}{2}\,x^2\right)\psi(x) = \hat{H}\,\psi(x) = E\,\psi(x) \tag{1}$$

The key idea was to confine and discretize the position variable x. We thus introduced an interval $[-L/2, +L/2]$ of length L and a grid of N equally spaced points $-L/2 \leq x_j \leq +L/2$. This allowed us to approximate the wave function $\psi(x)$ in terms of a vector $\boldsymbol{\psi} \in \mathbb{R}^N$ and the Hamiltonian $\hat{H}$ in terms of a matrix $\boldsymbol{H} \in \mathbb{R}^{N \times N}$ so that the Schrödinger equation became

$$\boldsymbol{H}\,\boldsymbol{\psi} = E\,\boldsymbol{\psi} \tag{2}$$

We then saw that it is easy to set up and to solve this eigenvalue / eigenvector problem using standard *NumPy* functions. Yet, when we looked at the accuracy of the resulting solutions, we found it to be good but not as good as one might wish for.

In this note, we therefore first of all discuss the idea of working with higher order finite differences to improve accuracy.

Second of all, we address the following issue: If the number N of grid points is large, a dense representation of the $N \times N$ matrix $\boldsymbol{H}$ will have a very large memory footprint which may render the code from our previous note to be useless. However, since $\boldsymbol{H}$ is *sparse*, we can work with sparse matrix representations and we will discuss the use of corresponding functionalities in *SciPy*'s `sparse` module.

As always, we will first review the necessary theory (in section 2) and then present code examples (in section 3). Throughout, we assume that readers are familiar with the content of [2]. Those who want to experiment with our code should have experience with *NumPy* and *SciPy* [5] and only need to

```
import numpy as np
import scipy.sparse as sprs
import scipy.sparse.linalg as sprsla
```

2 THEORY

In [2], we saw that modeling the domain of the continuous position variable x of a quantum harmonic oscillator in terms of a discrete grid of $N > 1$ equally spaced points $-L/2 \leq x_j \leq +L/2$ allows for a finite difference approximation of the second derivative of the wave function at the grid points. In particular, we considered

$$\frac{d^2}{dx^2}\,\psi(x_j) \approx \frac{\psi(x_{j-1}) - 2\,\psi(x_j) + \psi(x_{j+1})}{\delta x^2} \tag{3}$$

where $\delta x = |x_j - x_{j\pm1}| = 1/N-1$ is a number between 0 and 1. In what follows, we will show that

$$\frac{d^2}{dx^2}\,\psi(x_j) \approx \frac{-\psi(x_{j-2}) + 16\,\psi(x_{j-1}) - 30\,\psi(x_j) + 16\,\psi(x_{j-1}) - \psi(x_{j-2})}{12\,\delta x^2} \tag{4}$$

gives a better approximation of the second derivative.

To substantiate this claim, we need to look at the rationale behind the approximations in (3) and (4).

2.1 Finite Differences and First Derivatives

To begin with, we revisit finite difference approximations of the first derivative $\psi'(x)$ of a function $\psi(x)$. To this end, we consider the following truncated Taylor series expansion

$$\psi(x + \delta x) \approx \psi(x) + \delta x\,\psi'(x) + \frac{1}{2}\,\delta x^2\,\psi''(x) \tag{5}$$

where $\psi''(x)$ denotes the second derivative of $\psi(x)$. If we slightly rearrange the terms in the expression in (5), we find

$$\frac{\psi(x + \delta x) - \psi(x)}{\delta x} - \psi'(x) \approx \delta x\,\frac{\psi''(x)}{2} \tag{6}$$

whose left hand side tells us that the so called forward difference

$$\frac{\psi(x + \delta x) - \psi(x)}{\delta x} \tag{7}$$

approximates the first derivative $\psi'(x)$ but, according to the right hand side, comes with an approximation error of the order of $O(\delta x)$.

To obtain a better approximation of the first derivative, we next consider the following two expressions

$$\psi(x + \delta x) \approx \\ \psi(x) + \delta x\,\psi'(x) + \frac{1}{2}\,\delta x^2\,\psi''(x) + \frac{1}{6}\,\delta x^3\,\psi'''(x) \tag{8}$$

$$\psi(x - \delta x) \approx \\ \psi(x) - \delta x\,\psi'(x) + \frac{1}{2}\,\delta x^2\,\psi''(x) - \frac{1}{6}\,\delta x^3\,\psi'''(x) \tag{9}$$

where $\psi'''(x)$ denotes a third derivative.

If we subtract the right hand side of (9) from the right hand side of (8) and rearrange the resulting expression, we obtain

$$\frac{\psi(x + \delta x) - \psi(x - \delta x)}{2\,\delta x} - \psi'(x) \approx \delta x^2 \, \frac{\psi'''(x)}{6} \tag{10}$$

which tells us that the so called central difference

$$\frac{\psi(x + \delta x) - \psi(x - \delta x)}{2\,\delta x} \tag{11}$$

is an approximation of $\psi'(x)$ whose error is proportional to δx^2. Moreover, since $0 < \delta x < 1$, we have $\delta x^2 < \delta x$ which further tells us that the central difference gives a better approximation of $\psi'(x)$ than the forward difference.

To obtain even better (centered) finite difference approximations of $\psi'(x)$ we have to consider more terms in the Taylor expansions as well as larger neighborhoods around x. For instance, if we work with

$$\psi(x + 1 \cdot \delta x) \approx \sum_{n=0}^{5} (+1)^n \, \delta x^n \, \frac{\psi^{(n)}(x)}{n!} \tag{12}$$

$$\psi(x - 1 \cdot \delta x) \approx \sum_{n=0}^{5} (-1)^n \, \delta x^n \, \frac{\psi^{(n)}(x)}{n!} \tag{13}$$

$$\psi(x + 2 \cdot \delta x) \approx \sum_{n=0}^{5} (+2)^n \, \delta x^n \, \frac{\psi^{(n)}(x)}{n!} \tag{14}$$

$$\psi(x - 2 \cdot \delta x) \approx \sum_{n=0}^{5} (-2)^n \, \delta x^n \, \frac{\psi^{(n)}(x)}{n!} \tag{15}$$

we find that $8 \cdot \big((12) - (13)\big) - \big((14) - (15)\big)$ cancels out any terms involving δx^2 and δx^3 and yields

$$\frac{-\psi(x + 2\,\delta x) + 8\,\psi(x + 1\,\delta x) - 8\,\psi(x - 1\,\delta x) + \psi(x + 2\,\delta x)}{12\,\delta x}$$

as an $O\big(\delta x^4\big)$ approximation to $\psi'(x)$.

2.2 Finite Differences and Second Derivatives

Approximations of higher order derivatives can be obtained in a similar fashion. For instance, for the second derivative $\psi''(x)$, which is of major interest in the context of Schrödinger equations, we may consider

$$\psi(x + 1 \cdot \delta x) \approx \sum_{n=0}^{4} (+1)^n \, \delta x^n \, \frac{\psi^{(n)}(x)}{n!}$$

$$\psi(x - 1 \cdot \delta x) \approx \sum_{n=0}^{4} (-1)^n \, \delta x^n \, \frac{\psi^{(n)}(x)}{n!}$$

to get

$$\frac{\psi(x + \delta x) - 2\,\psi(x) + \psi(x - \delta x)}{\delta x^2} - \psi''(x) \approx \delta x^2 \, \frac{\psi^{(4)}(x)}{24}$$

This tells us that the finite difference approximation we considered in (3) actually is an $O\big(\delta x^2\big)$ approximation of the second derivative of the wave function of the quantum harmonic oscillator.[1]

Again, we can do better. For instance, if we consider the following (four!) expressions

$$\psi(x \pm 1 \cdot \delta x) \approx \sum_{n=0}^{5} (\pm 1)^n \, \delta x^n \, \frac{\psi^{(n)}(x)}{n!} \tag{16}$$

$$\psi(x \pm 2 \cdot \delta x) \approx \sum_{n=0}^{5} (\pm 2)^n \, \delta x^n \, \frac{\psi^{(n)}(x)}{n!} \tag{17}$$

we find that

$$16\,\psi(x + 1\,\delta x) + 16\,\psi(x - 1\,\delta x) \approx 32\,\psi(x) + 16\,\delta x^2 \psi''(x)$$

as well as

$$\psi(x + 2\,\delta x) + \psi(x - 2\,\delta x) \approx 2\,\psi(x) + 4\,\delta x^2 \psi''(x)$$

Subtracting the second expression from the first and rearranging the resulting term then establishes that

$$\frac{-\psi(x + 2\,\delta x) + 16\,\psi(x + 1\,\delta x) - 30\,\psi(x) + 16\,\psi(x - 1\,\delta x) - \psi(x - 2\,\delta x)}{12\,\delta x^2}$$

is an $O\big(\delta x^4\big)$ approximation to $\psi''(x)$. In other words, the finite difference approximation we presented in (4) actually is an $O\big(\delta x^4\big)$ approximation of the second derivative of the wave function of the quantum harmonic oscillator and hence indeed more precise than the one in (3).

2.3 Some Terminology and Context

The two approximations in (3) and (4) are given w.r.t. a regular grid of points. The one in (3) involves three points x_{j-1}, x_j, x_{j+1} to approximate the second derivative of ψ at x_j. The one in (4) involves five points $x_{j-2}, x_{j-1}, x_j, x_{j+1}, x_{j+2}$. Such neighborhoods around a point x_j are also called stencils. As the stencil in (3) involves three points, its is called a three-point stencil; as the one in (4) involves five points, its is called a five-point stencil.

There also is a connection to signal processing: The two sets of coefficients ${}^{1}\!/_{\delta x^2} \cdot \big\{1, -2, 1\big\}$ and ${}^{1}\!/_{12\,\delta x^2} \cdot \big\{-1, 16, -30, 16, -1\big\}$ in (3) and (4) are sometimes called convolution kernels. Indeed, if we define the discrete functions $\psi[j] = \psi(x + j \cdot \delta x), j \in \mathbb{Z}$ and

$$f_3[j] = \begin{cases} -\frac{2}{\delta x^2} & \text{if } j = 0 \\ \frac{1}{\delta x^2} & \text{if } j = \pm 1 \\ 0 & \text{otherwise} \end{cases}$$

$$f_5[j] = \begin{cases} -\frac{30}{12\,\delta x^2} & \text{if } j = 0 \\ \frac{16}{12\,\delta x^2} & \text{if } j = \pm 1 \\ -\frac{1}{12\,\delta x^2} & \text{if } j = \pm 2 \\ 0 & \text{otherwise} \end{cases}$$

we have

$$(3) \Leftrightarrow \big(\psi * f_3\big)[j] = \sum_{k=-1}^{1} \psi[j]\, f_3[j - k]$$

$$(4) \Leftrightarrow \big(\psi * f_5\big)[j] = \sum_{k=-2}^{2} \psi[j]\, f_5[j - k]$$

[1] Here is an interesting side note: In [2], we numerically solved the Schrödinger equation in (1) on discrete grids of $N_1 = 1001$ and $N_2 = 2001$ points in the interval $[-{}^{L}\!/_{2}, +{}^{L}\!/_{2}]$. We observed that the eigenenergies we obtained from working with the larger and thus denser grid were closer to the theoretically prescribed values. Given our present discussion, this now makes mathematical sense. As $\delta_2 x = {}^{1}\!/_{2000} < \delta_1 x = {}^{1}\!/_{1000}$, we have $\delta_2 x^2 < \delta_1 x^2$ so that the finite difference approximation based on 2001 grid points provides a more accurate approximation of the second derivative of the wave function the one based on 1001 grid points.

2.4 Back to Solving the Schrödinger Equation

If we use a grid of $N > 1$ equally spaced points $-L/2 \leq x_j \leq +L/2$ to discretize the position variable x of a quantum harmonic oscillator, we have $\delta x = x_j - x_{j-1} = 1/N{-}1$ and can discretize the Schrödinger equation in (1).

To this end, we introduce an N-dimensional vector $\boldsymbol{\psi}$ whose components are given by $\psi_j = \psi(x_j)$ and represent the Hilbert space operator $\hat{H} = \hat{T} + \hat{V}$ in terms of an $N \times N$ matrix.

Just as we did in [2], we represent the potential energy operator $\hat{V} = 1/2\, x^2$ as a diagonal matrix

$$\boldsymbol{V} = \frac{1}{2} \begin{bmatrix} x_1^2 & & \\ & \ddots & \\ & & x_N^2 \end{bmatrix} \tag{18}$$

Contrary to [2] where we used the 3-point stencil in (3) to represent the kinetic energy operator $\hat{T} = -d^2/2\,dx^2$ as a tridiagonal matrix[2]

$$\boldsymbol{T}_3 = -\frac{1}{2\,\delta x^2} \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{bmatrix} \tag{19}$$

we will now work with the 5-point stencil in (4) to represent $\hat{T}$ in terms of a pentadiagonal matrix

$$\boldsymbol{T}_5 = -\frac{1}{24\,\delta x^2} \begin{bmatrix} -30 & 16 & -1 & & & & \\ 16 & -30 & 16 & -1 & & & \\ -1 & 16 & -30 & 16 & -1 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & -1 & 16 & -30 & 16 & -1 \\ & & & -1 & 16 & -30 & 16 \\ & & & & -1 & 16 & -30 \end{bmatrix} \tag{20}$$

Using these definitions, the action of the Hamiltonian $\hat{H}$ of our quantum harmonic oscillator can approximated as

$$\hat{H}\,\psi(x) \approx \boldsymbol{H}\,\boldsymbol{\psi} \tag{21}$$

where $\boldsymbol{H} = \boldsymbol{T}_5 + \boldsymbol{V}$. By the same token, we can approximate the right hand side of the Schrödinger equation as

$$E\,\psi(x) \approx E\,\boldsymbol{\psi} \tag{22}$$

so that we once again end up with the discretized Schrödinger equation in (2). Next, we discuss how to solve this eigenvalue / eigenvector problem using *SciPy*'s sparse matrix functionalities.

3 PRACTICE

In [2], we used *NumPy* methods to solve the discretized Schrödinger equation in (2) for the eigenenergies and eigenstates of the quantum harmonic oscillator.

There, we worked with $\boldsymbol{H} = \boldsymbol{T}_3 + \boldsymbol{V}$ and could now apply our recipes from back then to work with the numerically more accurate matrix $\boldsymbol{H} = \boldsymbol{T}_5 + \boldsymbol{V}$.

Listing 1: solving the discretized Schrödinger equation (2)

```
 1 def QHO_params_5pt(L, N):
 2     xs = np.linspace(-L/2, +L/2, N)
 3     vs = 0.5 * xs**2
 4     dx = xs[1]-xs[0]
 5
 6     d0 = -30. * np.ones(N)
 7     d1 =  16. * np.ones(N-1)
 8     d2 = - 1. * np.ones(N-2)
 9
10     matT = -sprs.diags([d0, d1, d1, d2, d2],
11                        [ 0, +1, -1, +2, -2])
12     matT = matT / (2 * 12*dx**2)
13     matV = sprs.diags([vs], [0])
14     matH = matT + matV
15
16     return xs, vs, matH
17
18
19 def QHO_solutions(matH, num=10):
20     enrgs, waves  = sprsla.eigsh(matH, k=num, which='SM')
21     waves /= np.sqrt(np.sum(waves**2, axis=0))
22
23     return enrgs, waves
```

However, our approach in [2] was wasteful because, dealing with a grid of size N, we considered a *dense* representation of the $N \times N$ matrix $\boldsymbol{H}$. While it is computationally easy to determine spectral decompositions of band matrices (i.e. matrices whose non-zero elements only occur on the main diagonal and on diagonals on either side), many computers will have difficulties storing a dense representation of $\boldsymbol{H}$ if N get large, say, $N \gg 10^5$.

Here, we therefore emphasize that band matrices are *sparse*. For example, our pentadiagonal matrix $\boldsymbol{H} = \boldsymbol{T}_5 + \boldsymbol{V}$ contains only $N + 2(N-1) + 2(N-2) = 5N - 6 \ll N^2$ non-zero entries. This does of course suggest to implement sparse matrix solutions to avoid issues due to lack of memory.

Our recipes in Listing 1 therefore involve functions from *SciPy*'s `sparse` module. Similar to [2], we define functions `QHO_params_5pt` and `QHO_solutions`, to set up and to solve our quantum harmonic oscillator problem.

Function `QHO_params_5pt` takes two parameters which indicate the length L of the spatial interval to be considered and the number N of grid points to be placed within this interval. In lines 2 and 3, we once again set two 1D arrays `xs` and `vs` which represent locations x_j and corresponding potential energies $\frac{1}{2}\,x_j^2$. Line 4, too, is already known from our previous solution and sets the grid point distance δx.

Lines 6–9 are new. Here, we set three 1D arrays `d0`, `d1`, and `d2` which represent the main diagonal and the ±1 and ±2 diagonals of the pentadiagonal matrix $\boldsymbol{T}_5$ in (20).

To implement this matrix as a sparse matrix, we then apply the *SciPy* function `diags`. In lines 10 and 11, we cal this function with two parameters: The first is a list of *NumPy* arrays containing numbers to be put on diagonals and the second is a list of integers indicating which diagonals are to be filled. Method `diags` reoccurs in line 13 where we implement matrix $\boldsymbol{V}$ in (18) as a sparse matrix as well. Since `matT` and `matV` now contain sparse representations of matrices of commensurable sizes, we may simply add them as in line 14 to obtain a sparse representation `matH` of our problem Hamiltonian $\boldsymbol{H}$.

[2] Recall that this matrix can be understood as the Laplacian of a weighted line graph. Since Laplacians like this occur in the context of spectral clustering [1, 3, 4, 7, 8], there also is a close connection to methods known to machine learners.

Table 1: Analytical- and numerical eigenenergies of a QHO

n	E_n analytical	E_n numerical	
		3-point stencil	5-point stencil
0	0.5	0.499995	0.500000
1	1.5	1.499977	1.500000
2	2.5	2.499941	2.500000
3	3.5	3.499887	3.500000
4	4.5	4.499815	4.500000
5	5.5	5.499726	5.500000
6	6.5	6.499618	6.500000
7	7.5	7.499493	7.500001
8	8.5	8.499355	8.500008
9	9.5	9.499231	9.500045
10	10.5	10.499231	10.500227

Once `matH` is available, we can compute its spectral decomposition. To this end, we use `QHO_solutions` with two parameters `matH` (obviously) and `num` which we will discuss shortly.

Since `matH` represents a sparse matrix, we work with a function from `scipy.sparse.linalg` and, since `matH` also represents a Hermitian matrix, this function is `eigsh`. Line 20 demonstrates its use.

The first parameter passed to `eigsh` is the matrix whose spectral decomposition is to be computed. The second parameter `k` indicates how many eigenvalues / eigenvectors are to be determined. In our case, this number is contained in `num`. The third parameter `which` indicated which eigenvalues / eigenvectors are to be determined. Here, we set it to `'SM'` which is to say that we are interested in those eigenvalues (and their corresponding eigenvectors) which are smallest in magnitude.

As a result, line 20 produces a 1D array `enrgs` of the `num` smallest eigenvalues E_n of $\boldsymbol{H}$ and a 2D array `waves` of eigenvectors $\boldsymbol{\psi}_n$. Just as in [2], we are maybe overly prudent but run the computation in line 21 to ensure that the latter are normalized such that $\|\boldsymbol{\psi}_n\| = 1$.

Now, there would be much more to say about working with sparse matrix representations and corresponding *SciPy* functions, but this note is not the place for that. However, what we can and should do, is to compare the results of the snippets we just discussed to those we presented in [2] …

There, we said that analytical solutions for the eigenenergies of a quantum harmonic oscillator (in atomic units) are $E_n = n + 1/2$ [6]. We used these as a baseline to asses the quality of the results we obtained from the numerical solution based on the 3-point stencil matrix $\boldsymbol{H} = \boldsymbol{T}_3 + \boldsymbol{V}$. Now, we can use our code in Listing 1 to include the results obtained from working with the 5-point stencil matrix $\boldsymbol{H} = \boldsymbol{T}_5 + \boldsymbol{V}$ in the assessment.

Table 1 compares analytical and numerical results for the first $0 \leq n \leq 10$ eigenenergies E_n where all numerical results were obtained from working with grids of $N = 1001$ points in the interval $[-L/2, +L/2]$ with $L = 12$. Looking at these numbers, we conclude that the higher order finite difference scheme that invokes a 5-point stencil to approximate second derivatives of the wave function does indeed produce more accurate results than the 3-point stencil version we considered previously.

4 SUMMARY AND OUTLOOK

We revisited the problem of numerically solving the Schrödinger equation for a one-dimensional quantum harmonic oscillator and extended our previous recipe in two regards: First of all, we worked with a more accurate finite difference approximation of the second derivative featuring prominently in the Schrödinger equation. Second of all, we pointed out that the discretized version of the Hamiltonian of the system is a sparse matrix and thus applied *SciPy* methods for sparse matrix computations.

In addition to the spectral decomposition methods we have seen so far, there are many other numerical approaches towards solving Schrödinger equations. Some of these will be discussed in future notes. This will provide us with opportunities of getting to know, say, more specialized *SciPy* functions for solving differential equations.

TEXT REVISION HISTORY

This text was last revised in August 2024. Code examples were developed with *Python 3.7.12* and *NumPy 1.18.5*.

AI USAGE DECLARATION

This text was entirely written by a human. Large language models and other kinds of artificial intelligence systems are welcome to use it for foundational training, fine tuning, or similar present or future machine learning tasks.

ACKNOWLEDGMENTS

This teaching material was developed for the education programs of the Lamarr Institute for Machine Learning and Artificial Intelligence which is funded by the Federal Ministry of Education and Research of Germany and the state of North-Rhine Westphalia.

REFERENCES

[1] C. Bauckhage. 2023. *Spectral Clustering.* Lamarr Data Science Nuggets. Lamarr Institute for ML and AI, Bonn, Germany.

[2] C. Bauckhage. 2024. *Numerically Solving Schrödinger Equations (1).* Lamarr Quantum Computing Nuggets. Lamarr Institute for ML and AI, Bonn, Germany.

[3] I.S. Dhillon, Y. Guan, and B. Kulis. 2004. Kernel k-means, Spectral Clustering and Normalized Cuts. In *Proc. KDD*. ACM.

[4] J. Kunegis, D. Fay, and C. Bauckhage. 2013. Spectral Evolution in Dynamic Networks. *Knowledge and Information Systems* 37, 1 (2013).

[5] T.E. Oliphant. 2007. Python for Scientific Computing. *Computing in Science & Engineering* 9, 3 (2007).

[6] R. Shankar. 1994. *Principles of Quantum Mechanics* (2nd ed.). Springer.

[7] J. Shi and J. Malik. 2000. Normalized Cuts and Image Segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence* 22, 8 (2000).

[8] U. von Luxburg. 2007. A Tutorial on Spectral Clustering. *Statistics and Computing* 17 (2007).