Due Monday, December 14, 24:00

- I have named the relevant Python functions with web links so far, but I think it is time you make the habit of finding them yourself. Naturally, the easiest way is searching for what you want. However, note that Google sometimes lists an older version of the modules at the top.
- If you want to see everything together, all projects under SciPy (NumPy, matplotlib, etc) are listed at https://scipy.org/docs.html. For example, if you click on the specific link for SciPy, you reach https://docs.scipy.org/doc/scipy/reference/. All SciPy submodules are listed under *API Reference*, e.g. if you want to see all the functions related to linear algebra, click the link to scipy.linalg. This is a comprehensive list of the most up to date functions. Remember what I said about older versions above, so you might want to check here directly (it will be updated when there is a new version).
- NumPy is too big to list everything, and there are sub-sub modules. You pick a version of NumPy, typically the last stable one (currently here).[1] Then you click the "NumPy Reference" link to see the submodules categorized under different topics:
https://numpy.org/devdocs/reference/index.html. For example, if you click "Polynomials", you reach a list of sub-sub packages such as numpy.polynomials.chebyshev. You know the internet as well as me to figure out how to navigate the other branches of the modules.
- If you intend to use a very novel feature that came out only recently, make sure the module you have on your system is up to date. If you want to use code at the development stage, then know that there is a higher probability to have bugs and back-compatibility issues, so watch out for that as well. I advise you not to deal with development-stage code at all in this class, real life research might call for drastic measures though.
- From now on, i will only describe the functions you are going to use, without explicit links , you will find them yourselves.

Problem 21: Spectra of linear operators and symmetry (6 points)

We have seen how to transform an eigenfunction problem to an eigenvector problem using discretization. Eqs. 7.8.15-17 of the textbook give the specific matrices for the second derivative for Dirichlet, Neumann and periodic boundary conditions, respectively.

You learned in quantum mechanics that d^2/dx^2 is an Hermitian operator (or just take it from me if you have not). This is also reflected in the discretized versions of this operator as well, the derivative matrices are also Hermitian (i.e. symmetric in this case) for fixed and periodic boundary conditions. However, there is a curious aspect of the Neumann B.C. (Eq. 7.8.16)

$$X'(a) = 0 \quad , \quad X'(b) = 0 \ .$$

The textbook uses the discretization

$$x_i = a + i \frac{\overbrace{b-a}^{h}}{N} \quad , \quad i = 0, 1, \ldots, N \quad ,$$

with $x_0 = a$ and $x_N = b$. Elimination of the end points to write the second derivative matrix for the vector

[1] If you are a hardcore researcher, you may want to have a look at the "latest" (development) version (here). But no need unless you want to contribute to the NumPy project itself.

of the internal points $[x_1\ x_2 \dots x_{N-1}]$ gives

$$\frac{d^2}{dx^2} \to D_2 \equiv \frac{1}{h^2} \begin{pmatrix} -2/3 & 2/3 & 0 & 0 & \dots & 0 \\ 1 & -2 & 1 & 0 & \dots & 0 \\ 0 & 1 & -2 & 1 & \dots & 0 \\ & & \vdots & & & \\ 0 & \dots & 0 & 1 & -2 & 1 \\ 0 & \dots & 0 & 0 & 2/3 & -2/3 \end{pmatrix} . \tag{1}$$

Alternatively, we used the *ghost-point* method in class to obtain the following matrix for the vector $[x_0\ x_2 \dots x_N]$

$$\frac{d^2}{dx^2} \to D_2 \equiv \frac{1}{h^2} \begin{pmatrix} -2 & 2 & 0 & 0 & \dots & 0 \\ 1 & -2 & 1 & 0 & \dots & 0 \\ 0 & 1 & -2 & 1 & \dots & 0 \\ & & \vdots & & & \\ 0 & \dots & 0 & 1 & -2 & 1 \\ 0 & \dots & 0 & 0 & 2 & -2 \end{pmatrix} . \tag{2}$$

Even though these matrices are *mostly* symmetric, this property is lost at the endpoints. Does the Neumann boundary condition make the operator non-Hermitian? Not quite.

(a) We will change how we pick our sample points to obtain a symmetric second derivative matrix for the Neumann BCs:

$$x_i = a + (i + \frac{1}{2}) \overbrace{\frac{b-a}{N}}^{h} \quad , \quad i = 0, 1, \dots, N-1$$

The endpoints would be at $x_{-1/2} = a$ and $b = x_{N-1/2}$, but they are *not* parts of our discrete sample.[2] Second, we use the *ghost point method* instead of the elimination method in the textbook, and consider $x_{-1} = a - h/2$ and $x_N = b + h/2$ right outside the two ends of our interval. The ghost points are not a part of our sample, nevertheless, we can use them to impose the Neumann BCs in an $\mathcal{O}(h^2)$ manner right on the end points as

$$y'(a) \approx \frac{y_0 - y_{-1}}{h} = 0 \quad , \quad y'(b) \approx \frac{y_N - y_{N-1}}{h} = 0 \ ,$$

where $y_i \equiv y(x_i)$. Explicitly give the resulting second derivative matrix for the vector $[y_0 \dots y_{N-1}]$, and observe that it is symmetric. [3]

(b) Find the second derivative matrix for the Robin BCs

$$\alpha_1 x(a) + \alpha_2 x'(a) = 0 \quad , \quad \beta_1 x(b) + \beta_2 x'(b) = 0 \ .$$

Even though the symmetric version is superior, you can use the non-symmetric but easier-to-obtain one if you choose.

[2] This is called a cell-centered grid, and it is quite commonly used in 2D and 3D as well.

[3] We should add that Eq. 7.8.16 of the textbook is not incorrect, it provides the correct eigenvalues and eigenvectors with some truncation error in the $N \to \infty$ limit as far as I know. However there are cases where symmetric discretizations are reported to be superior in terms of conservation laws or stability when the continuous operator is Hermitian. We will also discuss some computational advantages below. This issue is sometimes called *dual* or *adjoint consistency*, and details are quite advanced for our class.

The quintessential problem for calculating the spectrum of a linear operator is finding the normal modes and frequencies of a string (the wave equation). In summary, the transverse waves on a string obey

$$\frac{\partial^2 y(t,x)}{\partial x^2} - \frac{1}{v^2}\frac{\partial^2 y(t,x)}{\partial t^2} = 0 \ ,$$

and the modes are solutions of the form

$$y(t,x) = T(t)X(x) \ .$$

These imply that the spatial part of a mode obeys

$$\frac{d^2X}{dx^2} = \lambda X,$$

with boundary conditions imposed on the end points of the string at $x = 0, L$. We will calculate these modes and their respective eigenvalues numerically. With correct scaling, you can take the endpoints to be $x = 0, 1$. You can check your PHYS201 notes or Marion and Thornton chapter 13 if you need to refresh your knowledge on waves, but you do not really need anything more than the above explanation.

(c) Calculate the first 8 normal modes and their eigenvalues for Dirichlet BCs. The exact solution is known for each mode number n in this case:

$$X_n = \sin\frac{n\pi x}{L} \quad , \quad \lambda_n = -\frac{n^2\pi^2}{L^2} \quad , \quad n = 1, 2, \ldots$$

so you can check the accuracy of your result.

There are various issues in this calculation.

- If you use N points to represent X, there will be at most N modes (the matrix is $N \times N$). So, you need at least N sampling points. Solve the problem with $N = 8$, and comment on the accuracy of your solution. Increase N, and report how the modes and eigenvalues are affected. How many sampling points did you need to have accurate values of all 8 eigenvalues to a 1% level? Plot the eigenvectors in comparison to the exact solution above, and also the eigenvalues vs the mode number for each N value you tried to explain your reasoning.
- You can use the builtin functions to calculate the eigenvalues and eigenvectors. There are various choices here, there are generic eigen problem solvers suitable for any matrix, and specific ones for banded, symmetric, sparse matrices etc. For one of the cases in the previous item, use all the different options from the most general to the most specialized eigen-solver, and report their accuracies and speed.[4] What would be your expectation, is it realized? Follow my advice in the beginning of the problem set to search for such functions in SciPy and NumPy. [5]
- The discrete second derivative matrix diverges as the number of sample points increase as N^2. An overall factor does not change the eigenvalue problem analytically, but a matrix with entries might run into many problems in numerics. It might be wise to multiply the matrix with h^2.[6]

(d) Calculate the first 9 normal modes and their eigenvalues for Neumann BCs. You do not need to repeat your test with different values of N, just use the number you surmised to be ideal for the Dirichlet BCs. However, provide two results, one for the symmetric second derivative matrix we obtained in

[4] In the most specific sense, the matrix you have is a tridiagonal Toeplitz matrix which is symmetric, tridiagonal and banded. It is consequently sparse as well but you do not need to use sparse matrix type for this problem.

[5] Remember that general-purpose solvers do not have to be slow, MATLAB's $A\backslash b$ is a good example (though for linear systems, not eigenvalue problems).

[6] This may not be necessary since there is an overall factor for all entries, but it is good to be on the safe side. What we did here can be considered a very simple case of preconditioning : replacing the matrix in a question with another one that is more amenable to numerical manipulation while still solving the same problem.

part (a), and another for the non-symmetric one in Eq. 2. This problem is a good way to show that our theoretical efforts for obtaining symmetric second derivative matrices was not in vain. There are algorithms that only work on symmetric matrices which are typically faster, so if there is an option, using a symmetric matrix is a potential advantage. For each solution use the best suited (most specific) eigenvalue solver for the matrix. Note that the matrix of Eq. 2 is even more special, it is tridiagonal and consequently also sparse, so has its own specific eigenvalue finders.

For comparison, the exact solution is

$$X_n = \cos\frac{n\pi x}{L} \quad , \quad \lambda_n = -\frac{n^2\pi^2}{L^2} \quad , \quad n = 0, 1, 2, \dots$$

(e) Calculate and plot the first 8 normal modes and their eigenvalues for periodic BCs. One calculation with sufficient numerical parameters is enough.

Aside from the exact solutions for the continuous case, the eigenvalues and eigenvectors for some of the discrete cases can also be calculated analytically. You can use these to test your numerical results, but be careful about the indexing which can be different in different sources.

Problem 22: Half a system to rule them all[7] (6 points)

The harmonic oscillator is *the* most important system in physics. We will add a bit of sauce to the problem, and investigate the time-independent Schrödinger equation of the half harmonic oscillator

$$\left[-\frac{\hbar^2}{2m}\frac{d^2}{dx^2} + V(x)\right]\psi = E\psi \tag{3}$$

where

$$V(x) = \begin{cases} \infty, & x < 0 \\ \frac{1}{2}m\omega^2 x^2, & x > 0 \ . \end{cases}$$

In order for the solutions to represent physical quantum states, they should also be normalizable, i.e

$$\psi(0) = \psi(\infty) = 0 \ .$$

These boundary conditions can only be satisfied for discrete values E_n called the eigenvalues, and the corresponding solutions of the differential equations ψ_n are called the eigenfunctions. This problem can be solved exactly using analytical methods, so you can look up ψ_n and E_n to check your numerical results.

It is important to emphasize that the eigenvalue equation above has a solution for any E if we do not care about normalizability. This is easy to see: pick any E, $\psi(0)$ and $psi'(0)$, and evolve $psi(x)$ as an IVP in x. You will obtain a function that satisfies Eq. 3, which means it is an eigenfunction with eigenvalue E. The problem will be that $\psi(\infty)$ will almost always diverge. So those very special discrete values E_n for which we do get $\psi(\infty) = 0$ are the "true" eigenfunctions we study in quantum mechanics, and we are interested below.

(a) If we measure all lengths in the characteristic length scale x_0 and all energies similarly in the characteristic energy E_0, the equation simplifies to

$$\overbrace{\left[-\frac{1}{2}\frac{d^2}{dz^2} + \frac{1}{2}z^2\right]}^{H}\psi = \epsilon\psi \ ,$$

where $z \equiv x/x_0$ and $\epsilon = E/E_0$. Find x_0 and E_0. Observe that asymptotically $\psi(z \to \infty) \sim e^{\alpha z^2}$, and show that $\alpha = \pm 1/2$.

[7] "The career of a young theoretical physicist consists of treating the harmonic oscillator in ever-increasing levels of abstraction." *Sidney Coleman*

(b) Find the first 4 eigenvalues and eigenfunctions using the *shooting method* (Sec. 7.6 in the textbook). You can use any of the numerical integrators on SciPy for the evolution. We learnt how to use the shooting method to solve BVPs, but wow do we use the shooting method to find eigenvalues and eigenfunctons? Read on.

First, observe that the eigenvalues of the half harmonic oscillator are exactly those of the usual 1D harmonic oscillator, but only the ones satisfying $\psi(0) = 0$, the so-called odd solutions. For the shooting method, we would normally pick a $\psi'(0) = \alpha$ and solve an IVP, and then try again. This is meaningless for Eq. 3, because the ODE part of our equation is linear, and it does not care about normalization (which is imposed afterwards). That is, integrating with $\psi'(0) = C\alpha$ gives us the same function, only scaled by C, not a new result! So changing $\psi'(0)$ is completely useless. If we cannot search for the eigenfunctions by changing $psi'(0)$, what do we do? We search by changing E! All solutions with different $\psi'(0)$ values for a given E are "the same", but we do not know which E value is the correct one. We lose $\psi'(0)$ as a free parameter, but we still have E, and that is the parameter we play with to obtain the eigenfunctions, i.e. the solutions with $\psi(\infty) = 0$.

To summarize, in addition to $\psi(0) = 0$, we can also fix $\psi'(0) = 1$ without loss of generality. Then we have apply the shooting method by changing E, which is a parameter in our ODE in Eq. 3. As we change E, $\psi_\infty \equiv \psi(\infty)$ will almost always diverge to $\pm\infty$,those special values of E where ψ_∞ changes from $+\infty$ to $-\infty$ are the eigenvalues, and the corresponding solutions to the IVPs are the eigenfunctions.

Note that the boundary is at infinity which cannot be reached numerically. Even if it could be reached, the function whose root we want to find, that is $\phi_\infty(E)$ is either $\pm\infty$ or 0, which is impossible to handle numerically. Your textbook wisely suggests that you should instead impose this condition at a finite but large value of $z = \zeta$. But how does this work exactly, and how large is large? The latter depends on the asymptotic behavior of the function which can usually be calculated analytically as above. Note that when we have a right value of ϵ, we will get a purely decaying function at infinity

$$\psi_n \sim e^{-z^2/2} \ .$$

So it will be almost 0 for $\zeta \gg 1$. Note that this is because the natural scale of z is 1, so being large is being large compared to 1. In general, the condition would be $z \gg \alpha^{-1/2}$, and the fact that α is of order unity is specific to this equation. For example, if we had not scaled our length, the meaning of a large distance would be $x \gg x_0$.

Let's return back to the question of how imposing the BC at a finite point exactly works. For anything other than an eigenvalue, we get

$$\psi \sim e^{-z^2/2} + Ce^{z^2/2} \ .$$

Then, you can see that imposing $\psi(\zeta) = 0$ necessarily gives you such a solution, i.e. not an eigenfunction. However $C \ll 1$, otherwise we could not satisfy the boundary condition due to the extremely large value of $e^{\zeta^2/2}$, so we get an almost-eigenfunction which is the best thing you can get in a numerical solution.

Solve the problem for various choices of ζ, and comment on your result. Even if your initial choice is not large enough for some reason, you should see that your solutions will not be changing significantly by increasing ζ after a while.

Ideally, you can pick ζ to be any large finite value, but the computational world is not ideal. Try consecutively larger values of ζ until you are not able to find solutions at all, and report your value. Why is there an upper limit to ζ? What can you change in your methods to be able to use higher ζ, which would *ideally* give better approximations to the true solution?

Your eigenvalues can be arbitrarily accurate using the finite-but-large boundary approach, but note that in any case your eigenfunctions will be only valid for $z \lesssim \zeta$ at best. One way to have a numerical

solution valid everywhere is using the asymptotics again. The numerical eigenfunctions might be completely off near ζ, but they are accurate some distance before it. So you can pick a point where you are in the asymptotic region ($z \gg 2$), but sill have accurate solutions ($z \lesssim \zeta$), then continue the solution artificially from this point on to infinity using the asymptotic behavior you know. The exact procedure calls for some care, you are not asked to perform this for this question.

(c) Find the first 4 eigenvalues and eigenfunctions using the *direct solution method* (Sec. 7.6 in the textbook). This method is not straightforward to implement either, you need to discretize the interval between the boundaries which is infinite. You might impose the boundary condition at a finite distance i.e. $\psi(\zeta) = 0$ similarly to the shooting method, but let us learn yet another method: *compactified coordinates*. Consider the alternative coordinate x_c defined as

$$z = \tan \frac{\pi z_c}{2} \ , \ z_c \in [0, 1] \ ,$$

and recast the differential equation as

$$\left[-\frac{2}{\pi^2} \left(1 + \tan^2 \frac{\pi z_c}{2} \right)^{-1} \frac{d}{dz_c} \left\{ \left(1 + \tan^2 \frac{\pi z_c}{2} \right)^{-1} \frac{d}{dz_c} \right\} + \frac{1}{2} \tan^2 \frac{\pi z_c}{2} \right] \psi = \epsilon \psi \qquad (4)$$

$$\implies \left[-\frac{2}{\pi^2} \left(1 + \tan^2 \frac{\pi z_c}{2} \right)^{-2} \frac{d^2}{dz_c^2} + \frac{2}{\pi} \left(1 + \tan^2 \frac{\pi z_c}{2} \right)^{-1} \tan \frac{\pi z_c}{2} \frac{d}{dz_c} + \frac{1}{2} \tan^2 \frac{\pi z_c}{2} \right] \psi = \epsilon \psi \qquad (5)$$

We discretize the z_c rather than z

$$z_{c,i} = i \overbrace{\frac{1}{N}}^{h} \quad , \quad i = 0, 1, \ldots, N \quad .$$

The values at the endpoints are fixed, $\psi(z_{c,0}) = \psi(z_{c,N}) = 0$, so we need to solve an eigenvalue problem for $H\psi = \epsilon\psi$, where ψ is the vector whose components are the wavefunction evaluated at the discrete interior points $[\psi_1 \ldots \psi_{N-1}]$ and H becomes a matrix representing a discrete Hamiltonian. The direct method has turned into calculating the spectra of a linear operator, H, by discretization (Sec. 7.8 of the textbook). All elements of H are finite in the interior, and note that the potential term $\tan^2 \frac{\pi z_c}{2}$ diverges as h^{-2} at most (for $H_{N-1,N-1}$) as number of sample points increase, which is no faster than the divergence of the discrete second derivative. It might be easier to work with $h^2 H$ as before.

The tangent function is a popular choice for compactification, but you can easily see that there are infinitely many options. However, some care might be due in our choice, especially in terms of how fast the compactification function is growing at its singularity (e.g. $\tan \frac{\pi(1-h)}{2} \sim h^{-1}$). Opportunities of compactification show up quite often in differential equations, and you should use it with caution. It might lead to many problems that are not apparent to the inexperienced.

As a final note, if you try to discretize the Hamiltonian in Eq. 5 using the most straightforward approach, the resulting matrix will not be symmetric. There is a way to obtain a Hermitian form using Eq. 4, but you can ignore the Hermiticity issue if you want.

(d) Let et us consider a non-linear Schrödinger equation on $x \in (-\infty, \infty)$ to see a true application of the direct solution method:

$$-\frac{\hbar^2}{2m} \frac{d^2}{dx^2} \psi = -U_0 (|\psi|^2 - |\psi_0^2|^2) \psi \ ,$$

where $U_0 > 0$. This is the *Gross-Pitaevskii equation* applied to a simplified Bose-Einstein condensate with attractive interactions. Scale x and ψ to simplify the equations, and show that one solution is

$$\psi(z) = \sqrt{2} \psi_0 \ \text{sech} \ z \ .$$

Now discretize the equation in light of what we have seen, and find the numerical solution for the ground state by root finding. You can use compactification or imposing the boundary conditions at large finite points, one method is sufficient. Try different values of N, compare the solutions to the above exact result, and comment.

Problem 23 (Bonus)

(a) Consider a second order linear ODE without a first order derivative term:

$$\frac{d^2y}{dx^2} = -g(x)y(x) + s(x) \ .$$

Show that the integrating scheme

$$y_{n+1}\left(1 + \frac{h^2}{12}g_{n+1}\right) = 2y_n\left(1 - \frac{5h^2}{12}g_n\right) - y_{n-1}\left(1 + \frac{h^2}{12}g_{n-1}\right) + \frac{h^2}{12}(s_{n+1} + 10s_n + s_{n-1}) \quad (6)$$

has $\mathcal{O}(h^6)$ accuracy in a single step. This is Numerov's method. It is a 2-step method, but is much more accurate than other examples we have seen such as Adams methods.

(b) Show that the cumulative error in Numerov's method is $\mathcal{O}(h^4)$. If we followed our logic from the RK family, we would naively conclude that the cumulative error was $\mathcal{O}(h^5)$.[8]

(c) Numerov's method is clearly superior to AB2 and RK2, and theoretically seems to be on par with RK4 in many respects while requiring fewer evaluations of the RHS. Implement Numerov's method and compare its speed and accuracy to RK4 for a simple example. You do not have to implement RK4 yourself, use any source you trust. Note that the default method in `solve_ivp` is not RK4, but a more accurate and efficient RK5(4). While at it, compare Numerov to this method as well.

If you love computational physics, you might want to solve the previous problem using Eq 6.[9] It is clear how to use this technique in the shooting method, but it can be viewed as a discretization scheme, and used to solve for eigenvalues as well! The last link is way over our level, I did not use the word "love" in vain.

[10]

[8] It seems this is a common error, even in some respectable textbooks. I learnt these facts from Peter Young's website for his computational physics course.

[9] No grade points for doing this, love does not seek compensation. However, feel free to submit your work as part of the solution, I am not someone to say no to free code.

[10] Is this my personal record for the number and total length of footnotes?