These notes draw heavily on *Numerical Recipes*, a valuable resource for entry-level understanding of numerical methods relevant to physics.

1. Eigensystem Math

A hugely powerful set of methods in numerical linear algebra centers on the use of eigenvalues and eigenvectors. In fact, the SVD technique we just covered is very closely related to eigensystems.

What is the definition of eigenvalues and eigenvectors?

For an $N \times N$ matrix, its eigenvalues and eigenvectors are defined by:

$$\mathbf{A} \cdot \vec{x} = \lambda \vec{x} \tag{1}$$

Clearly this is only applicable to square matrices, because $\mathbf{A}$ has to map one vector to another in the same-dimensional space.

How many eigenvalues and eigenvectors are there?

Up to N distinct ones. For some matrices eigenvalues are "repeated." What we mean by this is clear from the proof.

The eigenvalues are defined by the equation:

$$\det |\mathbf{A} - \lambda \mathbf{I}| = 0 \tag{2}$$

which follows from the fact that $\mathbf{A} - \lambda \mathbf{I}$ always maps $\vec{x}$ to zero, and so is clearly not invertible.

The left side of the above equation is clearly an Nth-order polynomial in λ. This polynomial will have N roots (which can be repeated of course). Repeated eigenvalues are said to be "degenerate."

Each λ has an associated $\vec{x}$ that produces it. The norm of the eigenvector is arbitrary, actually.

Technically, these eigenvectors are *right eigenvectors*. There are *left eigenvectors* defined by the analogous equation:

$$\vec{x} \cdot \mathbf{A} = \lambda \vec{x} \tag{3}$$

How are the eigenvalues of left eigenvectors related to the eigenvalues of right eigenvectors?

We can construct the determinant, and the polynomial equation it produces is the same, so they are the *same* eigenvalues.

Now we get to perform a neat trick. Let's make matrices from the eigenvectors $\mathbf{X}_R$ and $\mathbf{X}_L$,

by using the eigenvectors as the columns of the matrices. Then:

$$\mathbf{A} \cdot \mathbf{X}_R = \mathbf{X}_R \cdot \begin{pmatrix} \lambda_0 & 0 & \dots & 0 \\ 0 & \lambda_1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \lambda_{N-1} \end{pmatrix} \tag{4}$$

and

$$\mathbf{X}_L \cdot \mathbf{A} = \begin{pmatrix} \lambda_0 & 0 & \dots & 0 \\ 0 & \lambda_1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \lambda_{N-1} \end{pmatrix} \cdot \mathbf{X}_L \tag{5}$$

Calling the diagonal matrix of eigenvalues $\mathbf{D}$, we can show:

$$\vec{X}_L \cdot \vec{X}_R \cdot \mathbf{D} = \mathbf{D} \cdot \vec{X}_L \cdot \vec{X}_R \tag{6}$$

But $\mathbf{Q} \cdot \mathbf{R} = (\mathbf{R} \cdot \mathbf{Q})^T$, so if one of these matrices is diagonal, to commute, the other has to be too (in detail this is only true if the diagonal elements are distinct).

This means that if the λ_i are all distinct, then each $\vec{x}_{R,i}$ is orthogonal to $\vec{x}_{L,i}$. You can of course define the norms of the eigenvectors to make them orthonormal.

If there are degenerate eigenvalues, then if you find the corresponding eigenvectors for each degenerate root, you can always linearly combine them into another eigenvector of the same root. So they define a linear subspace of more than one dimension. If there are M degenerate values, this is an M-dimensional subspace. You can choose M eigenvectors within this space. They don't have to be orthonormal, but you always are free to choose them to be, which is usually the right thing to do.

In other words, it can always be arranged that:

$$\mathbf{X}_L = (\mathbf{X}_R)^{-1} \tag{7}$$

Note that this implies that there is a similarity transform that diagonalizes any matrix:

$$\mathbf{X}_R^{-1} \cdot \mathbf{A} \cdot \mathbf{X}_R \tag{8}$$

An incredibly important case is that of real, symmetric matrices. In this case, the left and right eigenvectors are the same. Therefore: $\mathbf{X}_R^{-1} = \mathbf{X}_R^T$. This means the matrix of eigenvectors in this case can be thought of as a rotation. It is specifically a rotation of the matrix into a new coordinate system where the matrix is diagonal!

2. Principal Component Analysis

This brings us to a very common use case for eigenvectors, which is principal component analysis.

It is common for us to generate real, symmetric, and positive-definite matrices. An example is that of moments of inertia.

What is the definition of the tensor moment of inertia?

$$I_{ij} = \int \mathrm{d}^N \vec{x} \rho \left(x_i x_j \delta_{ij} - x_i x_j \right) \tag{9}$$

What are the principal axes of the moment of inertia?

They are defined by the eigenvectors that define the coordinate system in which the moment of inertia is diagonal.

Similar games can be played with any real, symmetric matrix, and it is often convenient and useful to find the coordinate system in which it is diagonal.

Another common example is an example of dimensionality reduction known as Principal Component Analysis. If you have some set of data points $\vec{q}_i$, with mean $\vec{q}_0$, you can define the residual:

$$\vec{r}_i = \vec{q}_i - \vec{q}_0 \tag{10}$$

and its covariance:

$$\mathbf{C} = \sum_{ijk} r_{ij} r_{ik} \hat{e}_j \hat{e}_k \tag{11}$$

This covariance matrix has eigenvalues and vectors. Can you tell me what they are for the distribution I draw on the board?

The eigenvalues and vectors tell you the coordinate system that diagonalizes this covariance matrix, and what the diagonalized matrix is.

For the distribution on the board, which eigenvector has the larger eigenvalue?

The one corresponding to the larger variance.

Because the eigenvector matrix $\mathbf{X}$ is a rotation matrix, we can apply it to a vector. This is equivalent to projecting the vector onto each eigenvector, so it gives the components of the data in the new space.

This is a very generally useful operation. Especially for high-dimensional data sets, looking at only the first few eigencomponents of the data can often be a good way to approximate the data with a low-dimensional model.

3. Finite Difference and Waves on a String

Eigensystems are also useful in the analysis of many dynamical problems. One particular example is waves on a string. I take the description of this example from notes of Peter Arbenz of ETH Zürich.

For small displacements, the differential equation for the 1-D problem of a wave on a string is:

$$-\rho(x)\frac{\partial^2 u}{\partial t^2} + \frac{\partial}{\partial x}\left(p(x)\frac{\partial u}{\partial x}\right) + q(x)u(x,t) = 0 \tag{12}$$

with u being the displacement at position x, t being time, ρ being the mass density, p being related to the elasticity of the string, and q being the spring-like force. We assume $u = 0$ at the boundaries ($x = 0$ and $x = 1$).

We can separate variables:

$$u(x,t) = v(t)w(x) \tag{13}$$

and find:

$$\rho(x)\ddot{v}(t)w(x) - v(t)\left(p(x)w'(x)\right)' - q(x)v(t)w(x) = 0 \tag{14}$$

and we can rearrange to find:

$$\frac{\ddot{v}(t)}{v(t)} = \frac{1}{\rho(x)w(x)}\left(p(x)w'(x)\right)' + \frac{q(x)}{\rho(x)} \tag{15}$$

Since this holds always for all x and t, it must be a constant, which we will call λ.

This leads to a simple answer for $v(t)$:

$$\ddot{v} = -\lambda v \tag{16}$$

where, because of some foresight regarding solutions to the right-hand side that can satisfy the boundary conditions, we assume $\lambda > 0$.

What are solutions for v?

$$v = a\cos(\sqrt{\lambda}t) + b\sin(\sqrt{\lambda}t) \tag{17}$$

The consequence of this is that whatever pattern is established for $w(x)$ corresponding to λ, this will oscillate in time. Because the equation doesn't have any first order term in time, and no nonlinear terms, there is no dissipation so in this model it oscillates forever.

There in general will be an infinite number of possible λ values, each with a specific pattern $w(x)$ it corresponds to, and as λ grows, the period decreases. This is what you are familiar with regarding waves.

The spatial equation is:

$$\frac{1}{\rho(x)}\left[\big(p(x)w'(x)\big)' + q(x)w(x)\right] = -\lambda w(x) \tag{18}$$

This can be thought of as "linear differential operator on $w(x)$ equals scalar times $w(x)$." Sounds a tiny bit like an eigenvalue problem, and in fact it is, formally. Remember function space is a vector space. So linear operators are the same as linear mappings, but in an infinite dimensional space. Just like eigenvectors form a new basis set associated with their matrix, there are eigenfunctions that form a new basis set of function space associated with each linear operator. In the context of numerical solutions to this problem of the sort I'm about to describe, this analogy is explicit: we will literally construct a finite matrix $\mathbf{A}$ to approximate the linear operator.

If we determine the eigenfunctions, then a solution becomes:

$$u(x,t) = \sum_k w_k(x)\left[a_k\,\cos(\sqrt{\lambda_k}t) + b_k\,\sin(\sqrt{\lambda}t)\right] \tag{19}$$

where a_k and b_k are determined by the initial conditions ($u(x,t=0)$ and $\dot{u}(x,t=0)$).

For constant ρ, p, and q, we can simplify the equation. For simplicity I'll use units where $\rho = p = 1$.

$$-w'' + qw = \lambda w \tag{20}$$

or

$$w'' = (q-\lambda)w \tag{21}$$

which yields:

$$w = A\exp\left[\sqrt{(q-\lambda)}x\right] \tag{22}$$

What happens for $\lambda < q$ and $\lambda > q$?

For $\lambda < q$, the solution is exponential, and to meet the boundary conditions it must be zero. So if your string is springy, it can't support low frequency modes.

For $\lambda > q$, this is oscillatory, with shorter wavelengths for higher λ, as you expect.

What is the dispersion relation for this system?

The angular frequency squared is $\omega^2 = (2\pi)^2\lambda$, and the wavenumber squared is $k^2 = (2\pi)^2(\lambda - q)$. So we can write:

$$k^2 = \omega^2 - (2\pi)^2 q \tag{23}$$

If $q = 0$, this is just the ordinary dispersion relation for waves.

The reason to do this numerically is if $q(x)$, $p(x)$, or $\rho(x)$ are not constant. How do we go about this numerically?

First note that we can without punishment change the definitions of q and p to allow us to set $\rho = 1$.

Here we will use finite differences. We choose a series of N points $x_i = ih$ for $i = 1 \ldots N$, with $h = 1/(N+1)$ (we don't need $x = 0$ or $x = 1$ because at these $w = 0$). We approximate derivatives as:

$$\frac{\mathrm{d}g}{\mathrm{d}x} \approx \frac{g(x_{i+1/2}) - g(x_{i-1/2})}{h} \tag{24}$$

For $g = pw'$ can evaluate:

$$p(x_{i+1/2}) \left[\frac{\mathrm{d}w}{\mathrm{d}x}\right]_{x_{i+1/2}} = p(x_{i+1/2}) \frac{w(x_{i+1}) - w(x_i)}{h} \tag{25}$$

and then we can write:

$$\begin{aligned} \left[\frac{\mathrm{d}}{\mathrm{d}x}\left(p\frac{\mathrm{d}w}{\mathrm{d}x}\right)\right]_{x_i} &= \frac{1}{h}\left[g(x_{i+1/2}) - g(x_{i-1/2})\right] \\ &= \frac{1}{h}\left[p(x_{i+1/2})\frac{w(x_{i+1}) - w(x_i)}{h} - p(x_{i-1/2})\frac{w(x_i) - w(x_{i-1})}{h}\right] \\ &= \frac{1}{h^2}\left[-p(x_{i+1/2})w(x_{i-1}) + \left(p(x_{i-1/2}) + p(x_{i+1/2})\right) w(x_i) - p(x_{i+1/2})w(x_{i+1})\right] \end{aligned} \tag{26}$$

and

If we call $\vec{w} = \{w(x_1), w(x_2), \ldots, w(x_N)\}$, and call $\vec{r} = \{q(x_1)/\rho(x_1), \ldots, q(x_N)/\rho(x_N)$, this leads to the following matrix equation:

$$\mathbf{M} \cdot \vec{w} + \vec{r} \cdot \mathbf{I}\vec{w} = \lambda \vec{w} \tag{27}$$

where:

$$\mathbf{M} = \frac{1}{h^2}\begin{pmatrix} \frac{p_{1/2}+p_{3/2}}{\rho_1} & -\frac{p_{3/2}}{\rho_1} & 0 & \ldots & \ldots & 0 \\ \frac{-p_{3/2}}{\rho_2} & \frac{p_{3/2}+p_{5/2}}{\rho_2} & -\frac{p_{5/2}}{\rho_2} & 0 & \ldots & 0 \\ 0 & \frac{-p_{5/2}}{\rho_3} & \frac{p_{5/2}+p_{7/2}}{\rho_3} & -\frac{p_{7/2}}{\rho_3} & 0 & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & \ldots & 0 & 0 & \frac{-p_{N-1/2}}{\rho_N} & \frac{p_{N-1/2}+p_{N+1/2}}{\rho_N} \end{pmatrix} \tag{28}$$

Thus, we have converted this problem to a matrix eigenproblem! Where we need to find the eigenvectors and eigenvalues of:

$$\mathbf{A} = \mathbf{M} + \vec{q} \cdot \mathbf{I} \tag{29}$$

Note that while the real problem has an infinite number of eigenvalues, obviously this problem only has N eigenvalues. Also note that in this formulation of the problem the matrix $\mathbf{A}$ is tridiagonal. If we used higher order approximations for the derivative, it would be band-diagonal. (There is a finite-element expression of the problem which makes the matrix filled, but also requires lower N for the same precision).

An important lesson here is that we did *not* decide to integrate the differential equation directly in both time and space. Instead we analyzed the problem before adopting a numerical approach.

This turned a two-dimensional problem into a one-dimensional problem and allowed us to generate all of the classes of solutions, not just integrate one set of initial conditions. (Note that even more analysis would lead us to even better methods than the finite difference method we used here — but that's a different story).