CSCI 2951-G: Computational Protein Folding Fall 2012

Lectures 3 and 4 — 09/12/2012 and 9/17/2012

Prof. Paul Valiant *Evgenios Kornaropoulos*

Vector Calculus Review

We will start with a review of some calculus. We start with one dimension to give intuition for higher dimensions.

Given a function from one variable to one variable, for example, $f(x) = x^2$, we can approximate it via its derivatives. The most trivial approximation is without derivatives: we can approximate $f(x)$ near c by just the constant $f(c)$. But if f is rapidly changing (has large derivative) then this estimate will become rapidly bad. Approximating $f(x) = x^2$ near $c = 1$ yields the approximation $x^2 \approx 1$ near 1. How bad is this approximation? If we evaluate the function at 1.1 then $1.1^2 = 1.21$, and our trivial approximation is off by 0.21. Why does this number make sense? Well, our constant approximation is ignoring the first derivative − the derivative of x^2 is $2x$, which is 2 when evaluated at our center, $c = 1$. Thus if we move 0.1 away from our center and ignore a derivative of 2, we would expect our estimate to be off by roughly 0.2. And indeed, 0.21 is pretty close to 0.2. This suggests a more sophisticated approximation: the constant approximation at 1, plus our difference from 1 multiplied by the derivative at 1. This is the first-order approximation around c: $f(x) \approx f(c) + f'(c) \cdot (x - c)$. In our case, for $f(x) = x^2$, the approximation around 1 is $x^2 \approx 1 + 2 \cdot (x - 1) = 2x - 1$. How accurate is this approximation? Well, for $x = 1.1$ the approximation yields 1.2 as compared to $x^2 = 1.21$; namely, we are off by 0.01, which is much better than before. Where does this 0.01 come from? The second derivative.

Analogously with what we just saw, where ignoring the 1st derivative gives an error proportional to the first derivative and proportional to the distance from the center c, ignoring the second derivative will yield error that is proportional to the second derivative and proportional to the distance from the center *squared*. The second-order approximation near c, analogously, will be $f(x) \approx f(c) + f'(c) \cdot (x-c) + \frac{1}{2} f''(c) \cdot (x-c)^2$. For the case of $f(x) = x^2$ the second derivative is 2, and thus the approximation around 1 is $1 + 2(x-1) + (x-1)^2$ which is *exactly* x^2. This makes sense since any errors in this approximation would come from the 3rd derivative, which is 0 everywhere for $f(x) = x^2$. (If you were wondering why we scale f'' by a half, this equality justifies why it cannot be any other way; in general, the nth derivative is scaled by $n!$, because the nth derivative corresponds to a term involving $(x - c)^n$, and its nth derivative is $n!$, which we need to cancel out.) Some of the approximations of this form which are particularly useful are that x^a near 1 is roughly $1 + a(x - 1)$, e^x near 0 is roughly $1 + x$ (and $\log x$ near 1 is roughly $x - 1$), $\sin(x)$ near 0 is roughly x, and $\cos(x)$ near 0 is roughly $1 - x^2/2$. If any of the above is unclear, draw diagrams with Matlab.

Two dimensions:

We now move to functions from two inputs to one output, for example $f(x, y) = x^2 + y^2$. We can plot this in Matlab as follows (see Figure 1):

```
[x,y]=meshgrid(-2:.1:2,-2:.1:2);
figure(1);clf;surf(x,y,x.^2+y.^2,'EdgeColor','none','FaceColor','interp');axis([-2 2 -2 2 0 12])
```

The `meshgrid` function creates two dimensional matrices x and y which specify the x and y coordinates of a grid of points with x and y coordinates specified by the two arguments respectively. (Remember that `-2:.1:2` constructs the vector that starts from -2 and goes in increments of 0.1 until it hits 2.) Play with this command on your own if you are confused.

The `surf` command plots a surface, with x, y, z coordinates specified by its first three arguments. The dot in `x.^2` tells Matlab to square *each* element of x; ordinarily, Matlab would interpret x as a matrix and take its matrix square, which is not what we want. The remaining arguments in `surf` are parameter-value pairs, specifying in this case that it should not draw edges on the graph, and that it should draw the faces with

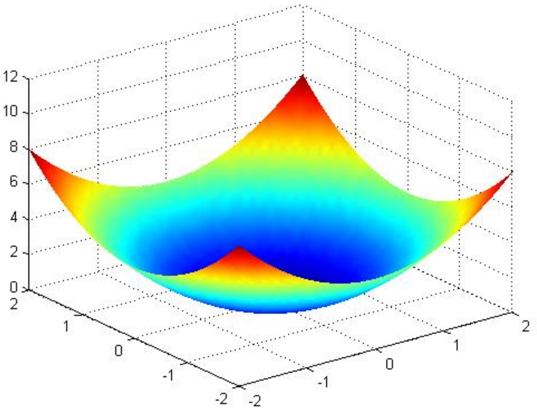


Figure 1: The function $f(x,y) = x^2 + y^2$

interpolated colors. The axis command specifies the axis limits, which Matlab usually sets automatically; here setting a high limit on the z axis gives the graph a better perspective.

Returning to our calculus thread: suppose we have the point $(x,y) = (1,2)$ and want to approximate values of $f(x,y) = x^2 + y^2$ near it. The constant approximation near $(1,2)$ is obvious: $f(x,y) \approx 1^2 + 2^2 = 5$. How inaccurate do we expect this to be as we move away from $(1,2)$? Well, the x derivative of $x^2 + y^2$ is $2x$, which is 2 for $x = 1$ which means that the function is changing at a rate of 2 in the x direction; correspondingly, it is changing at a rate of 4 in the y direction, since the y derivative is $2y$, evaluated at $y = 2$. Thus at the point $(1.1, 2.1)$ we might expect our estimate to be off by $0.1 \cdot 2 + 0.1 \cdot 4 = 0.6$ since our approximation ignores both of these rates of change. Viewed differently, a better approximation for $1.1^2 + 2.1^2$ would be to add 0.6 to our constant estimate, yielding 5.6. The actual value, of course, is 5.62. As you can guess, the general expression for the first-order approximation of $f(x,y)$ near (c,d) is $f(c,d) + f^{(c)}(c,d) \cdot (x-c) + f^{(d)}(c,d) \cdot (y-d)$. We can express this more compactly using vector notation: let $\nabla f(c,d)$ denote the vector of first derivatives (called the *gradient* of f, namely $[f^{(c)}(c,d) \;\; f^{(d)}(c,d)]$). Then, letting X denote the column vector $[x;y]$, and C denote the column vector $[c;d]$, we may express the first-order approximation of $f(x,y)$ near (c,d) as $\nabla f(C) \cdot (X - C)$ where the "$\cdot$" now denotes vector-vector multiplication (the dot product). This expression looks *very* similar to the expression from the one-dimensional case, but there is now a lot more going on. (Note, the square brackets notation is Matlab notation. $[a \; b]$ or $[a,b]$ is a row vector, but $[a;b]$ is a column vector.) See if you can derive approximations of two-dimensional functions based on second (or higher) derivatives.

A different way of arriving at the same answer is to imagine x and y as (smooth) functions of time, as $x(t)$ and $y(t)$, yielding that f is also a function of time: $f(x(t), y(t))$. We can now ask about the time derivative of f. This is provided by the chain rule as: $f^{(x)}(x,y)x'(t) + f^{(y)}(x,y)y'(t)$. Thus to approximate $f(a,b)$ near (c,d), we can imagine a particle that at time 0 is at position (c,d), and moves with velocity $(a-c, b-d)$ so that at time 1 it is at (a,b), and we can now treat $f(x(t), y(t))$ as a function of one variable, t, and approximate it with the one-dimensional methods above, yielding that $f(a,b) = f(x(1), y(1)) \approx f(x(0), y(0)) + 1 \cdot \frac{d}{dt} f(x(t), y(t))|_{t=0} = f(c,d) + f^{(x)}(c,d)(a-c) + f^{(y)}(c,d)(b-d)$, as above.

It is worth emphasizing that the expression, besides the constant $f(c,d)$ term, is a dot product, between the vector $[f^{(x)}(c,d) \;\; f^{(y)}(c,d)]$ which is the gradient, and the vector $[a-c; b-d]$ which is the displacement vector from the center, (c,d). This dot product means that, if you want to move "uphill" as fast as possible, you should head in the direction of the gradient; and if you want to keep the value of f constant then you should head perpendicular to the gradient; and if you want to head downhill as fast as possible, you should head in the direction opposite the gradient. If the gradient is the 0 vector, then it is possible we are at the minimum or maximum of the function; otherwise, we cannot possibly be at the minimum or maximum, as moving a tiny bit along the gradient in either direction will raise or lower the function respectively.

Lagrange Multipliers:

A slightly more subtle test for whether we are at a local extreme of the function is referred to as the method of *Lagrange multipliers.* This applies when, in addition to having a function $f(X)$ that we, say, are minimizing, we also have a *constraint*, for example, conservation of energy, which can be expressed as $h(X) = 0$ for some smooth function h. Near some certain point X such that satisfies the constraint $h(X) = 0$, we can ask, locally, what the set of points $X + \Delta$ for which $h(X + \Delta) = 0$ looks like. This can also be analyzed via the gradient, at least approximately, for small Δ: the linear approximation to h centered at X is just $h(X) + \nabla h(X) \cdot \Delta = \nabla h(X) \cdot \Delta$. Thus, near X, the set of directions Δ we can go in and still conserve $h(X + \Delta) = 0$ is the set of Δ that is *orthogonal* to the gradient of h. We can now ask, under what conditions might X be a local minimum of f, subject to the constraint $h(X) = 0$? And the answer is that if there exists a direction Δ that is orthogonal to ∇h, but not orthogonal to ∇f, then moving a tiny amount in direction Δ will (roughly) preserve $h(x) = 0$, but will linearly change $f(x)$, and hence X must not be a local minimum. (The image to have in mind is, if you have a fence on a hill, and you are standing at a point on the fence, trying to figure out if you are at the lowest point on the fence, then one test is the following: stare perpendicularly through the fence, and if you are not looking either *directly* downhill or uphill – if you are looking slantwise, somehow – then it must be that you can move left or right along the fence and get lower.) Conversely, if *every* direction Δ that is orthogonal to ∇h is also orthogonal to ∇f, then f is "flat" at X (subject to the constraint $h(X) = 0$) and thus a candidate for a local optimum. If we had a second contraint $g(X) = 0$, then the condition would be: for any Δ that is orthogonal to both $\nabla h(X)$ and $\nabla g(X)$, it is orthogonal to $\nabla f(X)$. Linear algebra tells us that this can only be the case if $\nabla f(X)$ can be expressed as a linear combination of $\nabla h(X)$ and $\nabla g(x)$. This is the famous Lagrange multipliers condition: X is a candidate for a minimum or maximum of f subject to constraints $h(X) = 0$ and $g(X) = 0$ provided that their exist multipliers λ_h, λ_g such that $\lambda_h \nabla h(X) + \lambda_g \nabla g(X) = \nabla f(X)$.

Symmetry:

The function $f(x, y) = x^2 + y^2$ is called *spherically symmetric* because its value depends only on the radius, $f(X) = |X|^2$, and hence its gradient is always in the radial direction, while movement in the transverse direction will preserve the radius and thus have "directional derivative" of 0. If we change coordinates from (x, y) to rotated coordinates such as $(z, w) = (\frac{x+y}{\sqrt{2}}, \frac{x-y}{\sqrt{2}})$ then the form of the function will not change: $x^2 + y^2 = f(x, y) = f(z, w) = z^2 + w^2$. In high dimensions, we can recognize a rotation by checking that it is a linear function, and then writing it out as a matrix: rotations correspons to *orthonormal* matrices, matrices where every column has length 1, and the dot product of any pair of different columns is 0.

Functions from many variables to many variables:

Functions from many variables to many variables can be analyzed analogously – instead of a single function $f(x, y)$, if there is a second function $g(x, y)$ then we can approximate both f and g near a center (c, d) of our choice, computing each approximation separately, without any more conceptual work. We work this out in a little more detail.

The pair of functions f and g together form a map from two variables (x, y) to two variables (f, g). As we considered the derivative of both f and g as a row vector, we may now consider the derivative of this map as a two by two matrix:

$$\begin{pmatrix} f^{(x)}(x, y) & f^{(y)}(x, y) \\ g^{(x)}(x, y) & g^{(y)}(x, y) \end{pmatrix}$$

Denoting the map as $F = [f; g]$ we can denote this matrix as ∇F. We can thus express the first-order approximation of F at $X = [x; y]$ near $C = [c; d]$ as $F(X) \approx F(C) + \nabla F \cdot (X - C)$, exactly as above, though with the notation now meaning something even more complicated, the "$\cdot$" now being a matrix-vector multiplication. This approximation is a linear map from two dimensions to two dimensions, and we can ask, as a standard linear algebra question, what this map does to *area*. The answer is that a linear map via the matrix M scales area via the determinant of M, and hence near (c, d), area is scaled as roughly $\det \nabla F$, a quantity which is called the *Jacobian*. Though we will likely not use multivariate integration in this course,

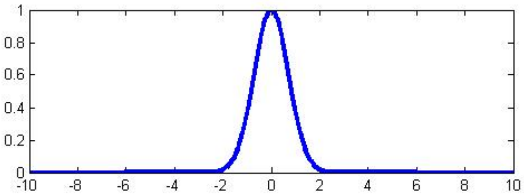


Figure 2: The bell curve, $f(x) = e^{-x^2}$

the *change of variables* formula for multivariate integration fundamentally uses the Jacobian, for the basic reason that if you change coordinates so that somehow a patch of area ϵ in the old coordinates has an area of, say, 5ϵ in the new coordinates, then we *must* divide by 5 after integrating in the new coordinates. Explicitly,

$$\int\int_{R} h(x,y)\,dx\,dy = \int\int_{F(R)} \frac{h(f,g)}{\det \nabla F}\,df\,dg$$

where R denotes the region of integration, and $F(R)$ denotes the result of mapping this region by F.

Finally, as an obvious corollary: a map is *area-preserving* if its Jacobian is everywhere 1.

The Gaussian Distribution

One of the most fundamental things in the areas of probability and statistics is the *Gaussian distribution*, often called the *normal distribution*, and whose shape is called the *bell curve*. This distribution is fundamental because it shows up exactly or approximately in many different places, and because it can be manipulated in many beautiful ways.

In one variable, the Gaussian distribution is specified by a center, and a width; changing the center just shifts the distribution; changing the width rescales the distribution in the natural way (if you make the distribution twice as thin, you also need to scale it by 2 so that its total integral remains 1 – probability distributions must have integral 1.) Ignoring all the constants and tweaks, the Gaussian distribution is defined by the function e^{-x^2}. This is a *probability density function*, which defines a probability distribution over the real numbers, defined via calculus: given a probability density function f, to compute the probability that a sample from f lies in the interval between 1 and 2, we take the *integral*: $\int_1^2 f(x)\,dx$. Thus for f to be a probability distribution over the reals, we need $\int_{-\infty}^{\infty} f(x)\,dx = 1$, meaning that the "total" probability is 1. If we try this for e^{-x^2}, we find that its integral is the slightly odd quantity $\sqrt{\pi}$, meaning that to make e^{-x^2} into a distribution, we should divide it by $\sqrt{\pi}$, to yield $\frac{1}{\sqrt{\pi}}e^{-x^2}$, which is now a legitimate probability density function. In general, the Gaussian distribution is defined as $\frac{1}{a\sqrt{\pi}}e^{-\left(\frac{x-b}{a}\right)^2}$, encompassing all possible shifts (b) and scalings (a).

Actually, how could we compute $\int_{-\infty}^{\infty} e^{-x^2}\,dx$? We can of course plot it in Matlab (see Figure 2):

```
x=-10:.001:10; plot(x,exp(-x.^2),'LineWidth',3)
```

and further, we can approximate the integral by just adding up the values of the function, every .001, and then dividing by 1000, as `x=-10:.001:10; sum(exp(-x.^2))/1000`, which yields 1.775. How close is this to $\sqrt{\pi}$? Well, we can ask Matlab: `x=-10:.001:10; sum(exp(-x.^2))/1000-sqrt(pi)` yields `1.5543e-015`, which is *very* small. In fact, when numerically approximating an integral like this, we would not usually expect to get accuracy down to 10^{-15} with only 1000 samples per unit. We can repeat the experiment with even fewer samples, for example, with samples spaced only every half unit: `x=-10:.5:10; sum(exp(-x.^2))*.5-sqrt(pi)` and we get `2.2204e-016` – the approximation is at the limit of machine precision, even with *very* coarsely spaced samples. This is just a hint at how amazingly well-behaved the Gaussian distribution is.

Back to our original thread, let us see how to rigorously compute this integral. One curious trick is to add a dimension: instead of the single dimensional Gaussian, we can consider the two-dimensional function

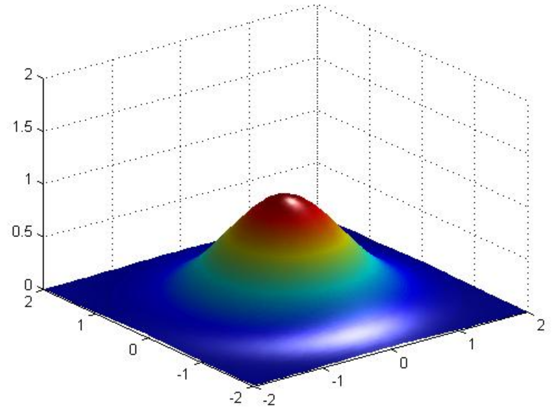


Figure 3: The two-dimensional Gaussian, $f(x,y) = e^{-(x^2+y^2)}$

$f(x,y) = e^{-x^2} \cdot e^{-y^2} = e^{-(x^2+y^2)}$. We can plot this in Matlab (see Figure 3) with:

```
[x,y]=meshgrid(-2:.1:2,-2:.1:2);surf(x,y,exp(-(x.^2+y.^2)),'EdgeColor','none',...
'FaceColor','interp','FaceLighting','Phong');axis([-2 2 -2 2 0 2]);light
```

(The "..." at the end of a line in Matlab tells Matlab you will continue entering the statement on the next line. When you type this in, you can omit the "..." and just type the statement all on one line. The "FaceLighting" parameter gives options for how "light" should interact with the faces of the surface when Matlab draws them – google "Phong lighting" if you are curious. Matlab's `light` actually creates a light, which will interact with the surface according to the "Phong" method.)

Because $f(x,y)$ can be expressed as the product of a Gaussian of x alone, and a Gaussian of y alone, the integral of their product (over all of space) is exactly the product of their integrals (over all of space), or just the square of the integral $\int_{-\infty}^{\infty} e^{-x^2}\,dx$. Keep this in mind as we evaluate $\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-(x^2+y^2)}\,dx\,dy$ a different way. Letting r be the radius of a point, $r = \sqrt{x^2+y^2}$, we note that $e^{-(x^2+y^2)} = e^{-r^2}$. For each r, there is a circle of points (x,y) at radius r, having circumference $2\pi r$, so thus we may alternatively evaluate our two-dimensional integral as just a one-dimensional integral over radii from 0 to infinity, as $\int_0^{\infty} e^{-r^2} \cdot 2\pi r\,dr$ (this change of coordinates could be justified more explicitly, but the basic point is that, for small ϵ, the set of points between radius r and radius $r+\epsilon$ has area roughly $2\pi r\epsilon$). This integral is surprisingly easy to compute, since the derivative of e^{-r^2} is $-2r \cdot e^{-r^2}$, yielding that this integral equals $-\pi e^{-r^2}\Big|_0^{\infty} = \pi$. Thus the two-dimensional integral of $e^{-(x^2+y^2)}$ is π, which implies that the one-dimensional integral of e^{-x^2} is $\sqrt{\pi}$, as we claimed above.

Marginal distributions:

Given a two-dimensional distribution $f(x,y)$, we can ask what the *marginal* distribution of x is, which means, if we draw a random point (x,y) from the distribution specified by f, what is the distribution (probability density function) of x alone? If the distribution f is the standard two-variable Gaussian, $\frac{1}{\pi}e^{-(x^2+y^2)}$, then the answer is obvious: we designed this distribution so that the distribution of x is just the standard one-variable Gaussian. However, in general, we find the marginal by integrating over all possible values of y. (Think of this in analogy with the discrete version: if x and y have just two outcomes each, say 0 and 1, then if I give you the matrix f specifying a probability for each of the four outcomes (x is 0 and y is 0; x is 0 and y is 1; x is 1 and y is 0; x is 1 and y is 1), then you can compute the marginal probabilities of x alone by summing up over all possibilities for y – the probability that x is 0 is the probability that x is 0 and y is 0 *plus* the probability that x is 0 and y is 1. For the continuous case, we integrate instead of summing: $Pr_f[x] = \int_{-\infty}^{\infty} f(x,y)\,dy$. A distribution $f(x,y)$ that can be decomposed as the product of two distributions $g(x)$ and $h(y)$ will have $g(x)$ and $h(y)$ as its marginals of x and y respectively.

Instead of just asking about the probability distributions of x and y, we can also ask about, say, the distribution of $x+y$. Before we figure out what this is, we'll note one particularly interesting way of looking at this: let x be drawn randomly from a Gaussian distribution; let y be drawn *independently* from the same Gaussian distribution; what does the sum of these two independent random variables look like? It is a folklore fact that many quantities drawn from nature are distributed roughly like Gaussian distributions, for example, the height of a random human. If the histogram of heights of humans looks like a bell curve, what does the histogram of the *sum* of heights of pairs of random humans look like? We could assemble some messy integrals to tackle this problem, or instead go back to a familiar theme: spherical symmetry. Since the two-dimensional Gaussian is symmetric, we can rotate it any way we want without changing it. (See Figure 3 again to see just how symmetrical it is.)

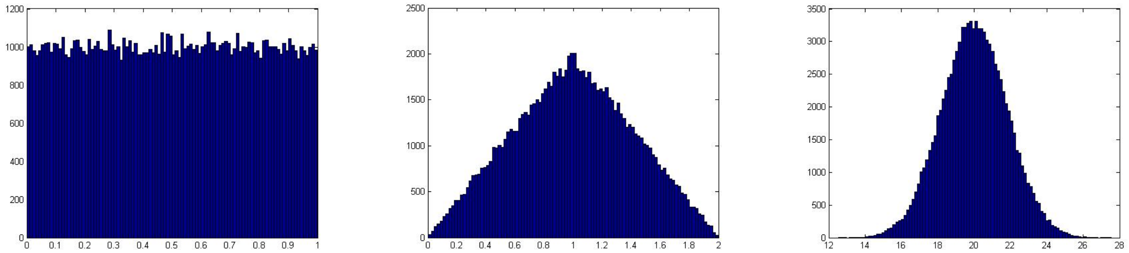
So the distribution of $x+y$ is the same as, say, $x+y$ rotated so that it lies on the x-axis. If we rotate $x+y$ to lie on the x axis, what will it be? Well expressing $x+y$ as the vector $(1,1)$, it has length $\sqrt{2}$, which if we rotated it to the x axis must be $\pm(\sqrt{2},0)$. Thus $x+y$ has the same distribution as $\sqrt{2}x$, under the Gaussian distribution, namely, it is a Gaussian that is "$\sqrt{2}$ times as fat", or more technically, drawn from the distribution $\frac{1}{\sqrt{2\pi}}e^{-x^2/2}$. Many deep facts about Gaussians can be derived by returning to this perspective.

It is worth noting that, if you draw (x,y) from the two-dimensional Gaussian $\frac{1}{\pi}e^{-(x^2+y^2)}$, then x and y will be independent, meaning they behave as though they were constructed independently. On the other hand, say, x and $x+y$ are *not* independent, for essentially obvious reasons (they both depend on x). These concepts become more intricate if the Gaussian is stretched in different directions: consider the process of drawing (x,y) from $\frac{1}{\pi}e^{-(x^2+y^2)}$ but then multiplying the vector (x,y) by a matrix M. Instead of points being distributed in a circular blob about the origin, they may now be distributed in a "stretched" blob. The simplest form of this is if we just scale x by some amount a, and y by some amount b. So now, instead of yielding a blob with size ≈ 1 in the x and y directions, the blob now has size a in the x direction and b in the y direction.

We can now ask, what is the probability density function of this (stretched) Gaussian? To compute this, we simply replace x by $\frac{x}{a}$ and y by $\frac{y}{a}$, and, finally, to make sure the total integral stays 1, we must divide by ab. (A different way of looking at this is that we transform (x,y) by the matrix $\begin{pmatrix} 1/a & 0 \\ 0 & 1/b \end{pmatrix}$, so an integral, under this transformation, is scaled by the determinant of this matrix, namely $\frac{1}{ab}$). Thus the probability density function is $\frac{1}{\pi ab}e^{-(x^2/a^2+y^2/b^2)}$. This can of course be generalized to n variables (with the π in the denominator becoming $\pi^{n/2}$).

The central limit theorem:

One of the main reasons why the Gaussian distribution shows up so often is that if you add up many independent random variables, the sum is often distributed close to a Gaussian. For example, the Matlab command `rand` returns a random real number between 0 and 1, and `rand(n,m)` returns an n by m matrix of random numbers between 0 and 1. This distribution is called the "uniform distribution between 0 and 1" because each possibility between 0 and 1 is uniformly likely. We can see this by drawing 100,000 random numbers and plotting their histogram: `hist(rand(1,100000),100)` – the 100 at the end specifies that the histogram has 100 bins (see Figure 4). This histogram is a way of approximately drawing the probability density function, and from this perspective, the "shape" of the uniform distribution looks like a rectangle. If we take *pairs* of samples from the uniform distribution and add them, we get something rather different: `hist(sum(rand(2,100000)),100)`, a distribution which now looks like a triangle. The more random numbers we sum up, the more it looks like a Gaussian: `hist(sum(rand(40,100000)),100)`. There are many ways of formally stating this fact, which is generally called the "central limit theorem" – there are many different conditions under which the distribution converges to Gaussian, and many different notions of what "close to Gaussian" means. In fact, it has been said that the modern history of statistics can be described by the history of central limit theorems. But for the moment, just remember that if you add up lots of independent random variables, their sum will be distributed essentially like a Gaussian, under a wide variety of conditions. (In particular, we have already derived the special case that if you add up two random numbers that are distributed as Gaussians, then their sum will be exactly Gaussian.)



(a) Uniform distribution (b) Sum of 2 uniform distributions (c) Sum of 40 uniform distributions

Figure 4: An illustration of convergence of the central limit theorem

Mean and Variance:

The two simplest ways of measuring a distribution are by its *mean* and *variance*. In one dimension, given a probability density function f, the mean is the expected value of the distribution; if you take many samples from the distribution and average them, the average will be close to the mean, and will converge to the mean as the number of samples goes to infinity. The formula for the mean is: $\mu = \int_{-\infty}^{\infty} x\, f(x)\, dx$, where μ (the Greek letter "mu") is commonly used to denote the mean. The second thing we generally want to know about a distribution is whether it is concentrated about its mean, or spread out. We measure this with the variance, which is the expected value of the *square* of the distance from the mean: $\sigma^2 = \int_{-\infty}^{\infty} (x-\mu)^2\, f(x)\, dx$, where the variance is denoted σ^2. The variable σ alone is the square root of the variance, which is called the *standard deviation*.

If you have two random variables x and y, then their means add: the mean of $x + y$ equals the mean of x plus the mean of y. (This is equivalent to what is referred to as "linearity of expectation".) If x and y are independent, then their variances will also add. The mean and variance characterize Gaussians in a nice way: for any (positive) mean and variance, the Gaussian of corresponding mean and variance is $f(x) = \frac{1}{\sqrt{\pi \cdot 2\sigma^2}} e^{-(x-\mu)^2/(2\sigma^2)}$. We have already seen that if we take x and y distributed according to a certain Gaussian, then their sum $x + y$ will be distributed according to a Gaussian of $\sqrt{2}$ times the width – where $\sqrt{2}$ times the width means twice the variance; this is an explicit example of variance adding. Note, however, that if x is distributed according to a Gaussian then $x + x$ will have *four* times its variance, not twice its variance, because x is *not* independent of x.

Simulating Physics

Armed with some basics of multivariate calculus and probability, we return to the topics of Lecture 3.

Energy Conservation

As we have seen in the previous lecture the energy of the system that we discussed is divided into kinetic energy $T(v) = \sum_i \frac{1}{2} m_i v_i^2$ and potential energy $U(p)$, where for the purposes of this lecture, we consider U to be an arbitrary smooth function of the vector of particle positions – explicitly, if we have n particles, then position is described by $3n$ coordinates, and U is a function from $3n$ variables to 1 variable (the potential energy). The force can be easily determined using the equation $F(p) = -\nabla U(p)$ (this is the gradient of the function U, as discussed in the first section; the gradient of U for a given p will be a vector also with $3n$ coordinates, as we would expect, given that the gradient describes the x, y, z components of the forces on each of the n particles).

The differential equations describing how this system will evolve, under the simplifying assumption that each particle has mass 1, will be (where a dot over a variable denotes the time derivative):

$$\dot{p} = v \tag{1}$$

$$\dot{v} = f = -\nabla U(p) \tag{2}$$

So all we need to have in order to study the physics of systems of particles (as in, atoms in a molecule) is the potential and kinetic energy functions. For this lecture, we will not write out the potential energy function explicitly – as we did in lecture 1 – as the analysis we will see today holds in general.

First, we show that evolving the system through differential equations 1 and 2 conserves energy. In order to do so, we have to study how the potential and the kinetic energy change. The potential energy depends only on position, so we have (by the chain rule, which for multivariate functions involves a dot product):

$$\frac{\partial}{\partial t} U(p) = \dot{p} \cdot \nabla U(p) = v \cdot \nabla U(p) \tag{3}$$

Explicitly, if U is a function of two variables with gradient $(-3, 2)$ at location (p_1, p_2), this means that if we increase p_1 by a tiny amount epsilon, then $U(p_1 + \epsilon, p_2)$ will be roughly $U(p_1, p_2) - 3\epsilon$; and if we increase p_2 by epsilon, then $U(p_1, p_2 + \epsilon)$ will be roughly $U(p_1, p_2) + 2\epsilon$. Thus if (p_1, p_2) is moving with velocity (v_1, v_2), then after epsilon time, U will be $U(p_1 + v_1\epsilon, p_2 + v_2\epsilon)$ which will roughly equal $U(p_1, p_2) + \epsilon(-3v_1 + 2v_2)$, namely, the time derivative of U will be the dot product between v and the gradient of U.

The kinetic energy changes as follows, using the product rule (recall that the "v^2" in the definition of kinetic energy really means *length* of v, squared; this can be alternatively expressed as the dot product of v with itself):

$$\frac{\partial}{\partial t} \frac{1}{2} v \cdot v = \frac{1}{2} \dot{v} \cdot v + \frac{1}{2} v \cdot \dot{v} = v \cdot \dot{v} = -v \cdot \nabla U(p)$$

The sum of this with Equation 3 equals zero, so the energy is conserved.

The Leapfrog Method

As we have seen in the previous lecture, if we evaluate and update the discretized forms of Equations 1 and 2 simultaneously, the results behave badly after a few iterations. Discretizing Equations 1 and 2 with a timestep of "dt" yields:

$$p \leftarrow p + v dt \tag{4}$$

$$v \leftarrow v - \nabla U(p) dt \tag{5}$$

We will now follow an *interleaved* approach to updating the values of p and v, computing a new value for p via Equation 4, and then using this *new* value of p to evaluate Equation 5, etc. Leapfrog integration is equivalent to updating positions p and velocities v at interleaved time points. This is illustrated in Figure 5: the red arrows on top correspond to updates via Equation 4, while the blue arrows on bottom correspond to updates via Equation 5. We can stop the evaluation at any point, but if we stop after updating p via Equation 4, then the most recently computed value of v will be effectively "half a timestep old"; conversely, if we stop after updating v via Equation 5, then the most recently computed value of p will be effectively "half a timestep old". This is the reason for the time labels in Figure 5.

Explicitly, in Figure 5 if we stop the process at $2dt$ after having computed "$p3$", the most recently computed value of v is "$v2$" which was computed for the time $\frac{3}{2}dt$, which is awkward, because we do not know p and v at the same time, as we might like to if we wanted to, say, estimate the total energy of the system at time $2dt$; even if we compute another iteration, we will just compute velocities "$v3$" for time $\frac{5}{2}dt$. If we want to estimate v at time $2dt$, we must interpolate, for example by averaging $\frac{1}{2}(v_{\frac{3}{2}dt} + v_{\frac{5}{2}dt})$.

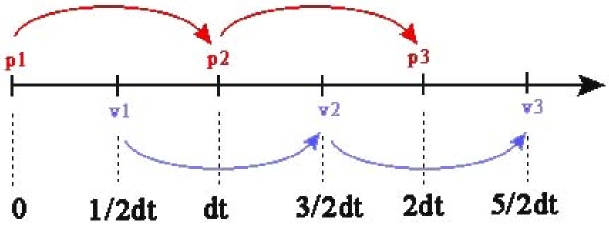


Figure 5: An illustration of the Leapfrog Method

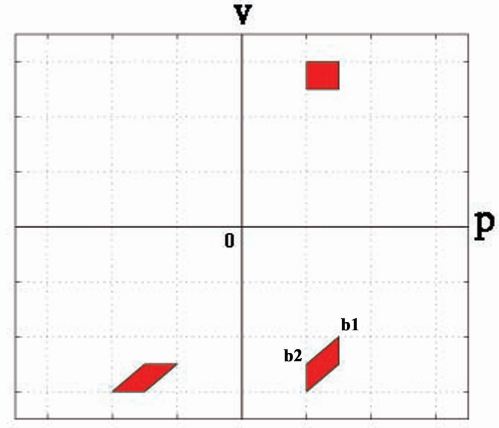


Figure 6: An illustration of the area conservation property

Symplectic Simulations

One of the key benefits of the leapfrog scheme, which is worth analyzing in some detail, is that it is "area preserving", or in more technical language, *symplectic*. Namely, if we consider a region of (p, v)-space of a certain area, and then apply several steps of the leapfrog method, the claim is that the resulting region will have the *same* area.

We will present this notion with an illustrative example. Suppose that variables v and p are in one dimension. According to the Leapfrog method, if we evaluate, say, Equation 5, then variable v is going to change while p will remain the same. So the question here is, why does this transformation preserves the area? Take a small region (the box in the upper right) draw it in the p-v plane as in Figure 6. Then see if the area remains the same after each step of the leapfrog method.

After an iteration of Equation 5, the box will be translated vertically (in the v direction), and might also be *skewed*. The amount that a point will be translated vertically by Equation 5 depends *only* on the value of p, and not on its vertical position v. Thus the right edge of the box might be translated down to a value $b1$, and the left edge translated down to a different value $b2$, which yields a parallelogram. Since the amount of vertical translation does not depend on the vertical position, each vertical line in the box may only be translated vertically, and may not be stretched or displaced horizontally. Thus the image of this box under the map of Equation 5 will be (roughly) a parallelogram of the same width and height as the original box, and hence have the same area.

This intuition can be formalized using calculus: in one dimension we can approximate a function locally by its value and derivative at a point – $f(x)$ near c can be approximated as $f(c) + (x - c)\, f'(c)$, and the map f rescales length by $f'(c)$. In two dimensions, the generalization of this says that $f(X)$ near C can be approximated locally as $f(C) + (X - C) \cdot \nabla f$, where ∇f is the 2 by 2 *matrix* of first derivatives; further, the map f rescales area by the *determinant* of the matrix ∇f. (This determinant is called the *Jacobian* of the map; the matrix itself is sometimes called the Jacobian matrix, or just the matrix of first derivatives.)

We can thus compute the matrix of first derivatives of the map of Equation 5: the derivative of the new value of p with respect to p is just 1, since Equation 5 does not modify p; thus the top left entry of the Jacobian matrix is 1. The derivative of the new value of p with respect to v is 0, which is the bottom left entry. The derivative of the new value of v with respect to v is $\frac{d}{dv}v = 1$, which is the bottom right entry. The derivative of the new value of v with respect to p is $\frac{d}{dp}(-\nabla U(p)dt)$ which is just negative dt times the second derivative of U, for the case where p is one dimensional. Thus the Jacobian matrix is:

$$A_1 = \begin{pmatrix} 1 & -U''(p)dt \\ 0 & 1 \end{pmatrix} \tag{6}$$

Taking the determinant of this (the product of the top left and bottom right entries, minus the product of the top right and bottom left entries) tells us that the map of Equation 5 exactly preserves area:

$$\det(A_1) = 1^2 + 0 \cdot U''(p)dt = 1$$

The corresponding analysis of Equation 4, with the roles of p and v swapped, yields an analogous result, with the Jacobian matrix being

$$B_1 = \begin{pmatrix} 1 & 0 \\ dt & 1 \end{pmatrix} \tag{7}$$

where the element in the bottom left is the derivative of the new value of v with respect to v, which is $\frac{d}{dv}(vdt) = dt$.

The Jacobian of B_1 is thus also 1, indicating that Equation 4 also preserves the area:

$$\det(B_1) = 1^2 - 0 \cdot dt = 1$$

Using the fact that the area is preserved we can claim that our system will never present the dilation effect that we observed in the previous lecture when simulating a spring with the simultaneous update rule (simultaneously updating via Equations 4 and 5).

Notice that if we apply both transformations simultaneously, the Jacobian matrix has both the top right and bottom left entries filled in, as

$$\begin{pmatrix} 1 & -U''(p) \cdot dt \\ dt & 1 \end{pmatrix}$$

which has determinant $1 + U''(p) \cdot dt^2$, which in general is *not* one, and can lead to rapid blowup as we saw in the last lectue. One thing to note: the amount this differs from 1 is controlled by dt^2, and thus this method is called a "second-order method".

While it is clear that symplectic integration schemes rule out certain kinds of blow-up, there is a very important and much more general theorem that shows that such schemes are very well behaved in an unexpected sense:

Theorem. *For any symplectic (area-preserving) scheme, such as the leapfrog method, for small enough dt there is a conserved quantity which is close to the energy E, and converges to E as dt approaches 0.*

We will get another sense of the power of symplectic integrators by considering the following natural question: suppose we have a symplectic system, if you let the system evolve enough, what is the long-term distribution of states?

For concreteness, consider the possible configurations of a simple harmonic oscillator (a spring), as we saw in the last lecture, where we plot the position and velocity on two axes (see Figure 7). Since the energy of the spring is proportional to the square of the distance of a configuration from the origin, the system will be restricted to evolve along a certain circle.

We are going to try to determine the long-term distribution of configurations along the circle. (The fact that we are dealing with a spring, with configurations along a circle, will not affect the analysis at all – it is

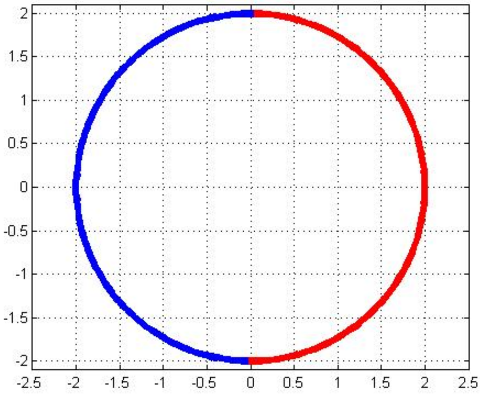


Figure 7: Distribution of the potential positions in a Harmonic oscillator.

completely general; we use this simple example so that we can easily draw a diagram.) The kind of situation that we want to rule out is the following: suppose that the particle has twice as high a probability of being at each point in the left semicircle (blue) as being in each point in the right (red). Let us now consider what happens as we evaluate the leapfrog method. We split the analysis into cases:

Case 1a: Physics sometimes moves from the red region to the blue region.
Case 1b: Physics sometimes moves from the blue region to the red region.
Case 2: Physics always "conserves color".

By assumption, the probability density of each blue point is twice as high as the probability density of the red; however, if Case 1a holds, then there is a region of red, of length, say, ℓ which physics will map to the blue region after some amount of time. Since physics is symplectic, the length of this region is preserved, and thus the blue region that gets mapped to must also have length ℓ. However, the *probability* must also be preserved, and since the probability of being in a region of length ℓ is twice as high for red regions as blue regions – if d is the probability density of red then $2d$ is the probability density of blue, and the probability of being in a red region of length ℓ is $d\ell$ while the probability of being in a blue region of length ℓ is $2d\ell$ – thus Case 1a contradicts conservation of probability. Similarly, Case 1b is ruled out.

Thus the only way red and blue regions can have different probability densities is if there is a *new* conservation law, in this case what we call "conservation of color". This result can be stated slightly more formally as:

Theorem (Liouville's theorem)**.** *At equilibrium (that is, after enough time), every configuration of the system that is allowed by the conservation laws has the same probability density (is equally likely).*

This fundamental result underlies statistical mechanics, and underlies the derivation of the final result of this lecture, the Boltzmann distribution, which we derive from two different perspectives.

The Boltzmann distribution

Multi-Dimensional Harmonic Oscillators

We first derive a useful special case of the Boltzmann distribution that involves multivariate Gaussians and makes use of the central limit theorem.

Consider n one-dimensional harmonic oscillators, namely n particles where the ith particle has a one-dimensional position p_i and velocity v_i; potential energy $\sum_i \frac{1}{2}p_i^2$ and kinetic energy $\sum_i \frac{1}{2}v_i^2$. If we consider the system as parameterized by these $2n$ numbers, as a vector, then the total energy of the system is just half the square of the length of this vector. Thus energy conservation yields that the system will evolve on the surface of a $2n$-sphere, of radius the square root of twice whatever energy the system started with.

Further, if we assume that energy is the only conserved quantity (technically this requires adding an arbitrarily small amount of interaction between the harmonic oscillators so that energy can transfer between them, but we will ignore this arbitrarily small correction here), then Liouville's theorem yields that, at equilibrium, the probability distribution of configurations of our system will be distributed uniformly over the surface of this sphere.

Given that the $2n$-element vector consisting of (p, v) for each of the n coordinates of position and velocity is distributed uniformly over the surface of a $2n$-sphere, we may now ask, what is the marginal distribution of an individual coordinate? We could calculate this explicitly, but it will be more informative to use an approximation that converges as the number of particles increases: "high-dimensional Gaussians stay close to the surface of a sphere".

How might we prove something like this? We have already seen that the Gaussian distribution is spherically symmetric – its probability density depends only on radius and not on the direction from the origin. All that remains to show is that the radii of samples from a high-dimensional Gaussian will be tightly clustered around some value away from 0, which will imply that the Gaussian behaves like a slightly blurry version of the uniform distribution over the surface of a sphere.

Suppose we have samples $(v_1, \ldots, v_n, p_1, \ldots, p_n)$ from a $2n$-dimensional Gaussian. Thus each of p_i and and v_i will behave like independent samples from a one-dimensional Gaussian. Further, while it is hard to work with the radius $\sqrt{\sum_i p_i^2 + v_i^2}$ directly, the square of the radius is very well behaved: $\sum_i p_i^2 + v_i^2$ is just the sum of $2n$ independent samples from a certain distribution, the distribution of the result of taking a Gaussian sample and squaring it. Let us call this the Gaussian-squared distribution.

We could explicitly figure out some properties of the Gaussian-squared distribution: it has a mean μ, and a standard deviation σ^2, which are some positive constants, but fortunately we do not have to. Instead we simply appeal to the central limit theorem: the distribution of $\sum_i p_i^2 + v_i^2$ is the sum of $2n$ independent samples from the Gaussian-squared distribution; since means and variances add for independent distributions, the mean of this distribution will be $2n\mu$ and the variance will be $2n\sigma^2$, and as n increases, the central limit theorem implies that this distribution will look increasingly like a simple *Gaussian* with mean $2n\mu$ and variance $2n\sigma^2$.

The crucial thing to note here is that the "width" of a Gaussian is its standard deviation, the square root of its variance, which in this case is $\sqrt{2n\sigma^2}$. So whatever constants μ and σ^2 happen to be, as n increases, the mean will eventually dwarf the standard deviation because n will eventually dwarf $\sqrt{n}$. For example, for $n = 10,000$ we have n is 100 times as big as $\sqrt{n}$, which would imply, roughly, that the square of the radius of our multivariate Gaussian is constant to within 1%, or, taking square roots, that our multivariate Gaussian stays on a spherical shell, to within 1%. For this reason, we consider a multivariate Gaussian as being a good model of the equilibrium distribution of a harmonic oscillator, when n is large.

Thus, assuming we are satisfied with this approximation, we can now say that the marginal distribution of, say, v_{17}, the velocity of the 17th particle, is approximately the marginal of the multivariate Gaussian, which is simply a one-dimensional Gaussian centered at 0, namely $\frac{1}{w\sqrt{\pi}} e^{-v_{17}^2/w^2}$, for some width w. Indeed, instead of just looking at the marginal of v_{17}, we could look at the 200-dimensional marginal of, say, the positions and velocities of the first 100 particles. As long as $2n$ is enough larger than 200, by the same argument, the distribution will be very close to the 200-dimensional Gaussian $\frac{1}{(w\sqrt{\pi})^{200}} e^{-(1/w^2)\sum_{i=1}^{100} v_i^2 + p_i^2}$. The startling thing here is that the quantity in the exponent exactly negative the total energy of the first 100 particles divided by $\frac{w^2}{2}$. The proportionality constant $\frac{w^2}{2}$ remains to be specified, and in fact we will dodge this issue entirely by *defining* temperature T to be that thing for which the proportionality constant $\frac{w^2}{2}$ equals $k_B \cdot T$ where k_B is a physical constant known as Boltzmann's constant. The result we have just shown for harmonic oscillators is stated more generally as:

Theorem (Boltzmann distribution)**.** *Given a system in thermal equilibrium with a large enough collection of particles at temperature T, the probability of configuration S is proportional to $e^{-E(S)/(k_B T)}$ where k_B is Boltzmann's constant and $E(S)$ is the energy of the configuration S.*

(It is worth noting one interesting distinction: n harmonic oscillators being at constant energy means the configuration is on the surface of a sphere; being at constant temperature means being drawn from a multivariate Gaussian that hugs the surface of the sphere. The distinction between constant energy and constant temperature is subtle, and good to be aware of, and something corresponding is true for systems beyond n harmonic oscillators.)

In general, the total energy of the system will not just be half the sums of the squares of all the positions and velocities – when we simulate molecular dynamics, the potential energy function will be a much more complicated function of the positions of all the atoms. However, the kinetic energy function will always be just $\sum_i \frac{1}{2}m_i(v_{ix}^2+v_{iy}^2+v_{iz}^2)$, and this will be the only way in which the velocities enter the energy function. Hence the only dependence on velocity in the Boltzmann distribution will be a factor of $e^{-\sum_i \frac{1}{2}m_i(v_{ix}^2+v_{iy}^2+v_{iz}^2)/(k_BT)}$, meaning that the probability distribution of the *velocities* of our atoms will be a (squashed) multivariate Gaussian, no matter how complicated our potential energy function is. This is essentially the equipartition theorem:

Theorem (Equipartition theorem). *Given a system of particles in thermal equilibrium, the energy in each coordinate of each particle is $\frac{1}{2}k_BT$ on average.*

(How could you derive the $\frac{1}{2}k_BT$ from the Boltzmann distribution?)

The equipartition theorem is actually somewhat more general than what we have stated here, as you might guess from the spherical symmetry of the multivariate Gaussian – it's not as though the x-coordinate of v_1 enjoys any special properties that a rotated coordinate system would not. In particular, if you consider an arbitrary linear combination of the velocities, for example twice the x-velocity of the 1st particle minus three times the z-velocity of the 8th particle, this quantity too is distributed like a single-variable Gaussian, and has average energy $\frac{1}{2}k_BT$, provided you figure out the appropriate way to define the "mass" that you would multiply by half the square of this odd quantity to yield its energy. Explicitly, the expected square of the x velocity of the first particle will be proportional to the inverse of its mass, $\frac{1}{m_1}$, and the expected square of the z velocity of the 8th particle will be proportional to $\frac{1}{m_8}$, and since these velocities are distributed as independent Gaussians, the expected square of our quantity $2v_{1x} - 3v_{8z}$ will be proportional to $\frac{4}{m_1} + \frac{9}{m_8}$, so thus, we call the inverse of this the *effective mass* of the particle: $\frac{1}{\frac{4}{m_1}+\frac{9}{m_8}}$. We have thus shown that a "fictitious particle" with velocity $2v_{1x} - 3v_{8z}$ and mass $\frac{1}{\frac{4}{m_1}+\frac{9}{m_8}}$ will obey the equipartition theorem just like any other particle. Work through an intuitive example yourself to get a sense for what is going on here. The second and third problems in Homework 1 were examples of this.

Standard derivation of the Boltzmann distribution

What follows is the standard textbook derivation of the Boltzmann distribution, which might be slightly less intuitive than the multivariate Gaussian derivation above, and also abuses math notation, but applies to more general systems.[1]

Suppose we discretize the system, by considering all the possible states the system can be in; we denote the energy of the ith state by ϵ_i. We consider *many* copies of the system, and let n_i denote the number of *copies of the system* that are in state i. The two conservation laws we use here are that energy is conserved:

$$E = \sum_i n_i \epsilon_i$$

and that the number of particles is conserved:

$$N = \sum_i n_i$$

By Liouville's theorem, any combination of states that is allowed by the conservation laws is equally likely. Given a certain vector $(n_1, n_2, \ldots)$ describing how many of the N copies of the system are in each state, how

[1] This derivation is taken from `http://bouman.chem.georgetown.edu/S98/boltzmann/boltzmann.htm`

many ways are there to assign this combination of states to the N copies of the system? Just the standard formula for choosing multiple subsets at once:

$$W = \frac{N!}{n_1!n_2!\dots}$$

Our goal here is to find the vector $(n_1, n_2, \dots)$ that maximizes W subject to the constraints on energy and particle number. We invoke the standard Stirling approximation, that for large k, $\log k! \approx k \log k - k$, which leads to

$$\log W \approx N \log N - \sum_i (n_i \log n_i) - \left(N - \sum_i n_i \right),$$

where the last term equals 0 by conservation of particle number, and the first term is constant. Hence we must optimize $\sum_i (n_i \log n_i)$ subject to our constraints. We take a slightly sketchy math step here and assume that for large enough numbers of particles, we can treat everything as continuous instead of discrete, and will use the Lagrange multipliers condition (discussed above) to tell us how to optimize this constrained function. The gradient of $\sum_i (n_i \log n_i)$ is the vector

$$\nabla \log W \approx (1 + \log n_1, 1 + \log n_2, \dots),$$

and the gradients of the two constraints are $\nabla E = (\epsilon_1, \epsilon_2, \dots)$ and $\nabla N = (1, 1, 1, \dots)$. The Lagrange multipliers condition says that $(n_1, n_2, \dots)$ can only be an optimum of $\log W$ subject to the constraints on energy and particle number if there exist multipliers λ_1, λ_2 such that

$$\nabla \log W = \lambda_1 \nabla E + \lambda_2 \nabla N,$$

which for the ith coordinate yields the condition $1 + \log n_i = \lambda_1 \epsilon_1 + \lambda_2$. Moving the 1 to the other side and taking the exponent of both sides yields $n_i = e^{\lambda_1 \epsilon 1} \cdot e^{\lambda_2 - 1}$, namely that given many copies of our system at thermal equilibrium at a certain temperature, the probability that a given copy of the system is in the ith possible state is proportional to the exponential of some constant λ_1 times the energy of the ith state. We define temperature T so that $\lambda_1 = -k_B T$, yielding the Boltzmann distribution, as desired.