[SQUEAKING]
[RUSTLING]
[CLICKING]
STEVEN G. JOHNSON:
OK, so last time
I talked about how in
order to define a gradient,
you need an inner product.
So that way, if you have a
scalar function of a vector,
the gradient is defined--
basically the
derivative has to be
a linear function that takes
a vector in and gives you
a scalar out.
So it turns out this has to
be-- if you have a dot product,
this has to be a dot product
of something with the x.
And we call the gradient.
So the gradient is the
thing with the same shape
as x that we take the
dot product with the x
to get the f.
So what I didn't mention is
that, in fact, not only did
we need a dot product
to define a gradient,
actually we swept something
under the rug earlier.
We actually need a norm in order
to even define a derivative
in the first place.
All right.
If you have a
vector space, a norm
is some measure of the
length of the vector
or a measure of distance.
A norm takes in a vector v
and gives you out a scalar.
And technically, to
qualify as a norm,
this map has to be non-negative.
It can't be negative.
It can only be 0 if v is 0.
If you multiply
the vector by 2, it

has to multiply the length by 2.
Or if you multiply the
length by negative 2,
it has to multiply the length by
2, basically the absolute value
of any scalar.
It has to satisfy something
called a triangle inequality.
So usually, most commonly
we're going to get a norm just
from an inner product.
So once you define
an inner product--
we talked about those last time.
You can even define inner
products of matrices--
you get a norm for free.
You can take the norm
is just a square root
of the inner
product with itself.
And this satisfies
all these properties
for any inner product.
So the reason I mention
this-- oh, and, by the way,
just cultural note.
So if you have a
continuous vector space
with an inner product, we
call that a Hilbert space.
If you have a continuous
vector space with a norm,
that's called a Banach space.
So it's a fancy-sounding
name, but it just
means you have a norm.
ALAN EDELMAN: You can
impress your friends
with your fancy mathematics,
but that's all it is.
STEVEN G. JOHNSON: Yes.
So the reason I
wanted to mention this
is that really the definition
of the derivative that we
used earlier implicitly
requires us to have a norm.
So it actually is both
the input and the output.

So it really only
applies to Banach spaces.
So the reason for that
is remember I define
the derivative to start with.
If you look at the
change in the output,
f of x plus delta
x minus f of x,
for not an infinitesimal but
a finite delta x that may be
small, remember that we defined
the derivative as the linear
part, as the linear operator
that gives the change to first
order, which means we
dropped any term that's--
we called it little o of
delta x-- any term that
goes to 0 faster than delta x.
So any term that's
small compared to this.
But in order to define
what it means to be small,
you need a norm.
If I have two vectors,
a column vector,
and I want to say is this column
vector smaller than that column
vector, how do I check it?
I check the length.
You need to map it
to a real number
to get a distance
or a smallness.
So formally, the definition
of this little o dx
is basically any function
such that the norm
of this over the norm
of delta x goes to 0,
as delta x goes to 0.
And in fact, even
to define a limit,
you need a norm of delta
x because if you've taken
[INAUDIBLE],, there's this
epsilon delta meaning
of a norm--
of a limit, sorry.

You can make this
arbitrarily small.
You can make this less
than or equal to epsilon
for all epsilon greater than 0.
And I'm not going to go
through the definition.
If you've seen the
definition of a limit,
there's some absolute values
in there that for vector spaces
have to turn into norms.
But basically it's just--
ALAN EDELMAN: My
experience is everyone's
seen the definition of
delta and epsilon limits,
and no one really
understands it.
STEVEN G. JOHNSON: Yeah.
ALAN EDELMAN: Is that fair?
Maybe some of you guys really
do, but most of us don't.
STEVEN G. JOHNSON: Yeah,
I mean, to be fair,
it took people 2,000
years to figure it out.
The concept of a limit
and an infinitesimal
was a big struggle
in mathematics
going back to the ancient
Greeks, Zeno's paradoxes
and so forth.
So it really took a
long time for people
to nail down what this meant.
But yeah, you need
to be able to have
a length, a norm of the
output, because this has
the same shape as the output.
These are the same shape as f.
To say that these terms
are small compared to delta
x, which you also need
a norm of delta x.
So just implicitly,
you always need
a norm of all of these

things to define it.
And usually, we're
going to get it
because we're going to define--
in most cases, we'll
define an inner product,
as we'll want that
anyway because if we want
to take derivatives
of scalar functions,
we want to be able to
write down gradients.
But this is what
you really need.
So anyway, so I just wanted to--
this is something we swept
under the rug in the beginning.
But since we defined
Hilbert spaces,
so I thought I should
define a Banach space.
I mean, I'm still sweeping
some things under the rug.
I'm sweeping what
does it mean for it
to be continuous under the rug?
But yeah, I wanted to
throw that out there.
That's all I wanted to say.
ALAN EDELMAN: That's it?
STEVEN G. JOHNSON: Yeah.
Any questions about that?
ALAN EDELMAN: Questions?
By all means.
OK, good.
All right.
So this is just a
little notebook.
And if we really need--
this isn't the live version,
so I can't really do anything.
But I have a feeling that
this will be good enough.
But if we need the
live version, we
can just press a few buttons.
So if I understood
correctly, last time
you got the answer for
what is the gradient

of the determinant.
Is that right?
Did you derive this formula?
STEVEN G. JOHNSON:
I did not derive it.
I just gave the answer.
ALAN EDELMAN: You
gave the answer.
And there's a few
different formats.
Did you--
STEVEN G. JOHNSON: I
give it the first one.
Determinant A--
ALAN EDELMAN: Is the cofactor.
STEVEN G. JOHNSON:
--inverse transpose, yeah.
ALAN EDELMAN: Oh,
the gradient is
the determinant of A times--
that's the second one, right?
STEVEN G. JOHNSON: Yeah.
ALAN EDELMAN: So the
first one is the cofactor
of A, which is one of
those linear algebra terms
that you may or
may not remember.
This is the one.
And just another
term is the adjugate
of A transpose, which is the--
the adjugate of a
matrix is the inverse
of the matrix divided
by the determinant.
STEVEN G. JOHNSON: Multiplied
by the determinant.
ALAN EDELMAN: Multiplied
by the determinant, right.
Let's see.
Of course, if you have
the gradient, then--
did you write down this
version as well last time?
The d of the determinant.
STEVEN G. JOHNSON: Well, I did d
of any matrix function, so yes.
I defined the dot product
and matrix dot product.

ALAN EDELMAN: Right, right.
I see.
So d, the determinant, will be
the trace of whatever formula
you have over here,
this formula times dA.
dA.
STEVEN G. JOHNSON: Transposed.
Transposed times it, yeah.
ALAN EDELMAN: Yes, right, sorry.
STEVEN G. JOHNSON: [INAUDIBLE]
ALAN EDELMAN: Let
me clear my head.
Yes, the trace of this
thing transposed times dA,
which is the element-wise--
it's just like the dot product.
You all get that.
It's just like the
vector dot product.
You multiply corresponding
elements, and you take the sum.
Whenever you have the
trace of A transpose B,
as Steven is writing very nicely
over here, that's the A dot B.
So if we know the gradient, then
the d has to be this formula.
I'm just defining the
adjugate right here so
that I can have it handy.
As Steven was saying,
it's just the determinant
times the inverse.
This is just a definition.
And then there's the
cofactor matrix, which
is the adjugate of A transpose.
Once I've defined this
one, to define this one,
I just get to do the equality.
So this defines these functions.
And here I've sort of
written it every which way.
The inverse in terms of the
adjugate and the cofactor,
the adjugate in terms
of the determinant,
the inverse and the cofactor.
You get it.

All three possibilities
are written here.
So for 2-by-2 matrices,
here's the 2-by-2 matrix,
and here's the cofactor matrix.
Some of you will
recognize that when
you form the inverse
of a 2-by-2 matrix,
the determinant goes
in the denominator.
And the thing that goes
in the numerator-- right,
you're all good at
2-by-2 inverses.
Do you know that by heart?
Would you be able to
do it in your sleep?
You switch the a and the d,
and you negate the b and the c.
Well, let's see.
You negate the b and the c, but
I'm also doing the transpose.
So you negate the b
and c and transpose
because it's the cofactor.
For the adjugate, you
just take the minus.
And anyway, these
are all the formulas.
Here's the inverse.
So the inverse is the adjugate
divided by the determinant.
Doing all this
numerically just for fun.
So numerically, here's
a random matrix,
and here's a random
perturbation.
What we're going to do is
look at the determinant
of the perturbed A minus
the determinant of A.
So there's the numerical value.
And by the way, I know
Steven has recommended always
using things on the order
of square root of epsilon
to make the perturbations
10 to the minus eighth.

And he's right.
I never do that, but you
should listen to him.
I just start typing three
or four 0's and a 1.
And actually, it's been good
enough for my purposes just
to check things.
I mean, Steven's is more--
it's the best possible one,
a square root of epsilon.
But with a quick
and dirty test, I
don't have the time to type
all those 0's, and I never
remember to type 1e minus 8.
So I just type these four
or five 0's or three, four,
or five 0's.
But in any event, here's
what the finite difference.
Here's the trace of
the adjugate times.
We see that they're correct to
enough digits to believe it.
STEVEN G. JOHNSON: How
come the adjugate is not
transposed there?
There's something missing here.
Oh, no, it's the
adjugate-- yeah, OK, right.
The determinant is the
transpose of the adjugate.
Never mind.
Never mind.
Adjugate is the great--
ALAN EDELMAN: I have to go
back and look at these formulas
to answer your--
STEVEN G. JOHNSON:
It's the transpose
of the gradient, so yes.
ALAN EDELMAN: Let me
say, yes, what you just
said, that the adjugate
of the transpose
is the thing that you want.
And so the trace needs
to transpose it twice,
so it's left non-transposed.

Yeah.
You got it.
The gradient is
the one transposed,
and this has to be the
transpose of the gradient.
STEVEN G. JOHNSON: Yeah.
This is gradient of determinant.
ALAN EDELMAN: Right,
like a double negative
makes a positive, a double
transpose makes for a no op.
STEVEN G. JOHNSON: This
is our dot product.
ALAN EDELMAN: Yep.
That's right.
OK.
So to actually see
the gradient, we
can rely on Julia's
internal forward difference
mode, for example, which is--
forward differentiation.
It's not forward difference.
It's automatic differentiation.
It's different from
forward differencing.
It's the forward
mode automatic--
I see this, and I think forward
differences, but it's not.
I think in this
lecture, if I get
a chance in just
a little bit, I'll
tell you about how this
forward mode works.
Steven kind of gave
you one view of it.
I'll give you
another view today.
But you just say, give me the
derivative or the gradient
of the determinant function,
and Julia will happily do it.
And of course, I
can compare that
with the adjugate
of A transpose.
And you guys know me by now.

When I see these things matching, it looks like to all the digits, it makes me happy. It makes me think, wow, this formula for the derivative is correct. Right, so this, for sure, is the derivative of the gradient. OK. I don't know. Maybe I tried to say this before, but I'm just going to repeat it, if you've heard me say it. But just philosophically, I find it remarkable that you could think of a limit of a finite difference, and the great mathematical gods let us have a formula. I mean, you've all done it in calculus, like the difference of a-- you take a sine and a little bit more sine, you get a cosine. Or the log, you get 1 over x. Or x squared, you get 2x. But I don't know. Could you guys imagine a universe where the mathematical gods weren't kind enough? Not every integral could be written as a formula. I mean, as you know, lots of integrals can't be written in terms of elementary functions. But derivatives, you could always do it. And you could do it for scalar calculus. That's why calculus is so easy to teach and is a beginning subject. You could do it for vector calculus, and you could do it for these complicated matrix functions.

I don't know.
Do you ever stop and think
about that being remarkable,
or you just take it as a given
and move on with your lives?
I think it's amazing
that we could have
a formula for this difference.
I just do.
And a simple formula, in effect.
But maybe you guys just
take it as a granted given,
but I don't know.
I think formulas are
gifts from the gods,
and I don't take
them for granted.
All right.
So this is really
just to show you
how to do reverse mode in Julia.
So it's not much different.
I'm just calling
Zygote, which is
one of the big, popular
packages in Julia
to do reverse mode autodiff.
And you see it's
not much different.
This is the ForwardDiff
package dot gradient,
and this is the
Zygote dot gradient.
Under the hood, it's
getting the answer
in a completely different
way, by going reverse mode.
But it actually gives
the same answer.
Though I wouldn't be surprised--
maybe Steven knows better--
I wouldn't be surprised if
this is just built-in formulas
in both cases.
I don't know.
Let's see.
We could do it symbolically,
but let's get to the proof.
So there are a couple of ways
to prove this mathematically.

So one relatively simple proof is to remember the Laplace expansion of determinants. So I suspect you all remember that if you want to calculate the determinant of a big matrix, usually people take maybe the first row. But in fact, you could take any row. But do you remember? You take the top-left entry times the determinant of what happens if you cross out the first row and first column. Then the second entry, and you cross out that row and column with a minus sign. Plus, minus, plus, minus. You remember that rule? So that's the Laplace expansion. And the key fact is that if-- for example, if I'm working with Ai1. Let's say I'm starting with the i-th row. Then Ai1 is not inside any of these Cs. Ai1 only appears here. Everything else you see depends on other elements of the matrix, but it doesn't depend on Ai1. Similarly, if you look at Ai2, Ci2, and every other term only depends on Ai2. To make that point very clear, here what I did was I took this matrix, and I made this matrix-- this 3-by-3 matrix you'll see, it's almost completely numerical, but I put one symbol in the bottom right. And if you take the determinant, you see that this is an--

I don't know whether
to call this a linear
or an affine function.
STEVEN G. JOHNSON:
It would be affine.
ALAN EDELMAN: Affine
for this class.
Some people would
actually say it's
linear in the sense of
linear, quadratic, cubic.
It's first-degree polynomial.
But let's call it affine for
the purposes of this class
because it's not 13a plus 0.
But whatever it is, it's
a first-degree polynomial
is what it is.
And the fact of the
matter is the coefficient
of a is exactly
this determinant.
It's 4 times 4 minus 3.
It's 16 minus 3.
It's 13.
And so the
coefficient of every--
if you make any
element a symbol,
the coefficient that's in
front of it is just this minor.
And so taking derivatives of
first-degree functions is easy.
It's just the slope.
The derivative of
this determinant
with respect to this element
is clearly the number 13.
And so the way to
say this, in general,
is if I want to take the
derivative of determinant
with respect to any
Aij element, the slope
is the thing that multiplies it.
So it's Cij.
And so that is one
immediate way to conclude
that the gradient
of the determinant

is the cofactor matrix.
It's that simple.
There's another proof
that is sort of--
I mean, this proof
is pretty simple.
I think it's easy to agree that
this is a nice, simple proof.
There's another proof that
might seem a slightly harder,
in one way, but in a
way, it's sort of--
mathematicians like
this kind of proof.
And so you get to take
your pick which one you
like best, but let me just
show you an alternative proof.
So in this alternative
proof, what we're going to do
is we're going to figure out the
right answer near the identity.
And then we're going to--
and then we're going to use
that to bootstrap ourselves
to any other matrix.
You know how to expand--
you're all familiar,
if I ask you
to compute the characteristic
polynomial of a matrix,
let's call it M. If
I need to do the--
here, I'll do what Steven
does, and I'll do it like this.
So if anybody asked
you to calculate
the characteristic polynomial--
and I'm using the mouse, which
means it's really sloppy.
Not that my handwriting is so
great, but it's not this bad.
All right.
So the characteristic
polynomial of any matrix M
is usually written like this.
And there's lots of factors.
There's lambda to the
n and all the way down
to the determinant, plus or

minus the determinant of M. So you remember that. And if you want, you can-- if you want, you can make this a plus sign, and then you get plus signs in this whole formula. And so this is not much different. Here if lambda was 1, if you just took lambda equals 1, you'd have determinant of I-- well, let's just see it this way. Determinant of I plus a matrix would be 1 plus. And then there would be the terms that you would get. There would be the next terms. This thing here, as you all know, is the trace of the matrix. So maybe I should have put that in. You get lambda to the n plus lambda to the n minus 1 times the trace of the matrix. So if you make a tiny, little perturbation, the determinant of I plus dA-- I guess I should have made this 1 plus the trace of dA, to be honest. Let's fix that right now. So the determinant of I plus dA would be 1. That would be 1 to the n. Plus 1 to the n minus 1 times the trace of the matrix. And then there's the lower-order terms. So that's one way to think of it. And so there we immediately get the answer around the identity. And now if we want to get this anywhere, all we have to do is recognize that if we want to go to the determinant of A plus dA, then we just go A times A inverse over here, and that's just the identity. But then we can use the properties of determinants to pull out the determinant of A. And you just get I plus A inverse dA. And this basically here, we just think of this A inverse dA as the trace formula. And therefore, we get our answer, the very answer we're looking for. In a way, this is more complicated, but mathematicians like this one better than-- I don't know why. They're both valid. You get to take your pick. There's something I like about this, though it is a little bit more complicated. But in any event, we get the same answer. So what do we have here? So application to the derivative of the characteristic polynomial. So once again, there's the simple proof. The characteristic polynomial of a matrix is the product of x minus the eigenvalues. Probably a different sign from what I have here. You take the derivative of this product. You get the sum of these products, n minus 1 at a time, which you could rewrite like this. But you can also directly do-- with our technology,

you can do this and get
basically the same answer
as the direct proof.
And then I have some
numerical checks.
Let's see.
And the derivative of
the log determinant.
Log determinant comes up a lot.
Logs have lots of
functions come up a lot.
For example, Steven,
I don't know,
a few lectures ago talked
about this f over f prime.
It's what shows up whenever you
do anybody's Newton's method.
And of course, this
could be written as 1
over the log f prime.
So basically, the logarithmic
derivative and its reciprocal
come up all over mathematics.
So the derivative of the
log of the determinant
is simply the trace of
the inverse times the dA.
This you've seen, A inverse.
And that's it.
Any questions?
That basically covers the
gradient of the determinant.
Any questions about that?
So maybe a few words
about determinant.
Interestingly
enough, people often
tell you that you should
never compute a determinant.
Or hardly ever might
be a fair term.
So determinants are real.
It's a real favorite of
elementary linear algebra
classes.
Determinants are great for
telling you in exact arithmetic
whether a matrix
is singular or not.
So a matrix has determinant

0, it's singular.
If the determinant
is not 0, it's not.
And that sounds like
a really good idea,
to have something like that.
But it turns out that when
you're doing computations
in finite precision, if
you're doing it on a computer,
the determinant turns out
to be not so meaningful.
It gets to be hard to
compute accurately.
There are a lot of issues with
calculating the determinant.
It turns out that while the
pure mathematicians live
in a binary world where a matrix
is singular or non-singular,
the truth of the matter is
is that the world of matrices
is not so binary.
It's a bit more of
a spectrum where
matrices are singular or nearly
singular or a little bit bad
or not at all bad.
And probably you've all heard
the word that I'm referring to.
The word that we use
in numerical analysis
is conditioning.
So ill conditioned means a
matrix is nearly singular.
And well conditioned means
that it's very non-singular.
Too many double
negatives there, but it's
sort of the good
side of singular
when we say it's
well conditioned.
And so the determinant
doesn't really give--
the determinant it's
not a really good--
give a good job of talking
about how nearly singular
matrices are.

The condition number, which is related to singular values-- I'm not going to talk about that today-- is a much better way of talking about matrices being singular or not. So you learned it all in a course, like 18.06 or elementary linear algebra. You learned about determinants. And then later on, when you compute, people tell you to forget about determinants mostly. There are times, but mostly. And the other thing we tell people to do is forget about the characteristic polynomial as well. That's not how we calculate eigenvalues either. We don't take roots of polynomials. Anybody happen to know how we compute eigenvalues in the real world? We don't do characteristic polynomials. Anybody know the magic two letters that happen when you type eigenvalues? How many of you just thought it was the characteristic polynomial? You take the roots. How many of you had any idea how roots got taken-- eigenvalues got taken on the computer? So do you have any idea what the algorithm is being used?

AUDIENCE: [INAUDIBLE]

ALAN EDELMAN: The power method.

AUDIENCE: Yeah.

ALAN EDELMAN: OK, so you probably

didn't hear the student
saying that, well, in 18.06,
I learned about something like
the power method, which gives
you the dominant eigenvalue.
Yeah.
And then nobody else in
the room has any idea
how eigenvalues get calculated?
Just a little bit
of culture here.
So people don't know.
I see.
I kind of feel like I
ruined the question then.
I should have just asked how
are eigenvalues computed?
Because I imagine many
of you would have said,
isn't it the
characteristic polynomial?
You get the roots.
Because every one
of you have formed
the characteristic polynomials
of 2-by-2 matrices.
I know you have.
You got that quadratic equation,
and you solve for the roots.
You remember?
Who remembers doing that?
Quadratics, you get the roots.
If you had a mean teacher, maybe
they forced you to do a cubic,
but I bet they didn't.
Anyone ever do it for a cubic?
Maybe it was rigged
to be easy though.
So right, so none of that
happens on the computer.
I'm not going to tell you
in detail how it's done,
but I will mention just
the fact that it's not
the characteristic
polynomial is half
of what I want you to know.
And the other half is there's
something called the QR
algorithm for eigenvalue.

So in general, a QR for
a matrix factors a matrix
to orthogonal times
upper triangular.
And a funny thing happens.
If you factor a matrix into QR
and then reverse it and get RQ,
and if you do that again,
factor that new matrix into QR
and reverse it to RQ,
and you keep doing that,
essentially the eigenvalues
magically appear.
And there are some details.
If the matrix is
symmetric, the matrix
will actually become more
and more diagonal as you go.
If it's not symmetric, but
it has real eigenvalues,
it will become triangular.
And you'll see the eigenvalues
on the diagonal eventually.
And if it's complex, you'll
get these little 2-by-2 pieces
which are easy to get
the eigenvalues from.
So there are a bunch of
tricks to accelerate all this,
but the basic idea
is QR, RQ, QR, RQ.
You might actually try it in
Julia or Python or whatever.
MATLAB, whatever you
like to do one day.
You'll see it just works.
STEVEN G. JOHNSON:
But it is related
to the power method under
the hood, if you dig deep.
ALAN EDELMAN: Oh,
dig really deep.
That's right.
It's sort of like a block power
method in multiple dimensions
all at once.
It's crazy.
Yeah.
OK.
All right.

So that's that one.