

Local linearization

📖 Google Classroom

Learn how to generalize the idea of a tangent plane into a linear approximation of scalar-valued multivariable functions.

Background

- [The gradient](#)

What we're building to

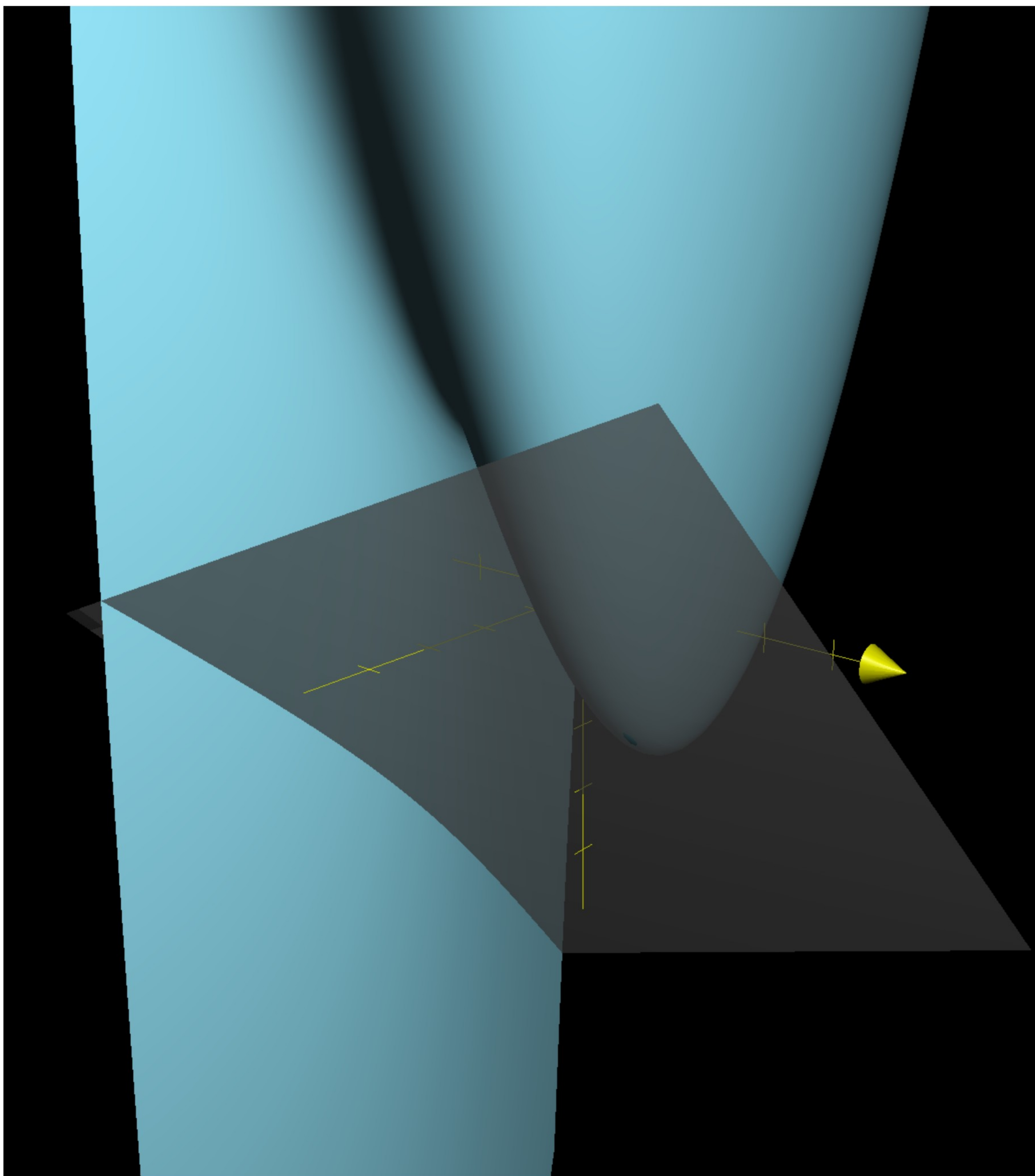
- Local linearization generalizes the idea of tangent planes to any multivariable function. Here, I will just talk about the case of scalar-valued multivariable functions.
- The idea is to approximate a function near one of its inputs with a simpler function that has the same value at that input, as well as the same partial derivative values.
- Written with vectors, here's what the approximation function looks like:

$$L_f(\mathbf{x}) = \underbrace{f(\mathbf{x}_0)}_{\text{Constant}} + \underbrace{\nabla f(\mathbf{x}_0)}_{\text{Constant vector}} \cdot \overbrace{(\mathbf{x} - \mathbf{x}_0)}^{\mathbf{x} \text{ is the variable}}$$

- This is called the local linearization of f near \mathbf{x}_0 .

Tangent planes as approximations

In the [previous article](#), I talked about finding the tangent plane to a two-variable function's graph.



The formula for the tangent plane ended up looking like this.

$$T(x, y) = f(x_0, y_0) + f_x(x_0, y_0)(x - x_0) + f_y(x_0, y_0)(y - y_0)$$

This function $T(x, y)$ often goes by a different name: The "**local linearization**" of f at the point (x_0, y_0) . You can think about this as the simplest function satisfying two properties:

1. It has the same value of f at the point (x_0, y_0) .
2. It has the same partial derivatives as f at the point (x_0, y_0) .

As always in multivariable calculus, it is healthy to contemplate a new concept without relying on graphical intuition. That's not to say you should not try to think visually. Maybe instead think purely about the input space, or think relevant transformation rather than the graph.

Fundamentally, a local linearization approximates one function near a point based on the information you can get from its derivative(s) at that point.

In the case of functions with a two-variable input and a scalar (i.e. non-vector) output, this can be visualized as a tangent plane. However, with higher dimensions we don't have this visual luxury, so we are left to think about it just as an approximation.

In real-world applications of multivariable calculus, you almost never care about an actual plane in space. Instead, you might have some complicated function, like, oh, I don't know, air resistance on a parachute as a function of speed and orientation. Dealing with the actual function may be tricky or computationally expensive, so it's helpful to approximate it with something simpler, like a linear function.

What do I mean by "Linear function"?

Consider a function with a multidimensional input.

$$f(x_1, x_2, \dots, x_n)$$

This function is called **linear** if in its definition, all the coordinates are just multiplied by constants, with nothing else happening to them. For example, it might look like this:

$$f(x_1, x_2, \dots, x_n) = 2x_1 + 3x_2 + \dots - 5x_n$$

The full story of linearity goes deeper (hence the existence of the field "Linear algebra"), but for now, this conception will do. Typically, instead of writing out all the variable like this, you would treat the input as a vector:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

And you would define the function using a dot product:

$$f(\mathbf{x}) = \begin{bmatrix} 2 \\ 3 \\ \vdots \\ -5 \end{bmatrix} \cdot \mathbf{x}$$

For the purposes of this article, and more generally when you talk about local linearization, you are allowed to add in a constant to this expression:

$$f(\mathbf{x}) = \underbrace{c}_{\text{Some constant}} + \underbrace{\mathbf{v}}_{\text{Some vector}} \cdot \mathbf{x}$$

If you wanted to be pedantic, this is no longer a linear function. It's what's called an "affine" function. But most people would say "whatever, it's basically linear".

Local linearization

Now, suppose your function $f(\mathbf{x})$ does *not* have the luxury of being linear. (The bolded " \mathbf{x} " still represents a multidimensional vector). It might be defined by some crazy expression way more wild than a dot product.

The idea of a local linearization is to approximate this function near some particular input value, \mathbf{x}_0 , with a function that *is* linear. Specifically, here's what that new function looks like:

$$L_f(\mathbf{x}) = \underbrace{f(\mathbf{x}_0)}_{\text{Constant}} + \underbrace{\nabla f(\mathbf{x}_0)}_{\text{Constant vector}} \cdot \underbrace{(\mathbf{x} - \mathbf{x}_0)}_{\mathbf{x} \text{ is the variable}}$$

- Notice, by plugging in $\mathbf{x} = \mathbf{x}_0$, you can see that both functions f and L_f will have the same value at the input \mathbf{x}_0 .
- The vector dotted against the variable \mathbf{x} is the gradient of f at the specified input, $\nabla f(\mathbf{x}_0)$. This ensures that both functions f and L_f will have the same *gradient* at the specified input. In other words, all their partial derivative information will be the same.

I think the best way to understand this formula is to basically derive it for yourself in the context of a specific function.

Example 1: Finding a local linearization.

Problem: Have yourself a function:

$$f(x, y, z) = ze^{x^2 - y^3}$$

Find a linear function $L_f(x, y, z)$ such that the value of L_f and all its partial derivatives match those of f at the following point:

$$(x_0, y_0, z_0) = (8, 4, 3)$$

Step 1: Evaluate f at the chosen point

$$f(8, 4, 3) = \text{[input box]}$$

Check

[\[Hide explanation\]](#)

$$\begin{aligned} f(8, 4, 3) &= 3e^{8^2 - 4^3} \\ &= 3e^{64 - 64} \\ &= 3e^0 \\ &= 3 \end{aligned}$$

Step 2: Use this to start writing your function. Which of the following functions will be guaranteed to equal f at the input $(x, y, z) = (8, 4, 3)$?

Choose 1 answer:

☐ (A) $L_f(x, y, z) = 3 + 8ax + 4by + 3cz$

☐ (B) $L_f(x, y, z) = 3 + a(x - 8) + b(y - 4) + c(z - 3)$

For both of these, a , b and c are all arbitrary constants.

Check

[\[Hide explanation\]](#)

You might start writing the desired function L_f like this:

$$L_f(x, y, z) = 3 + \text{<something equal to 0 when } (x, y, z) = (8, 4, 3)$$

To make sure that the remainder of the equation is 0 at $(x, y, z) = (8, 4, 3)$ while keeping things linear, we only add constant multiples of the terms $(x - 8)$, $(y - 4)$, and $(z - 3)$, since these will all be 0 at the input.

$$L_f(x, y, z) = 3 + a(x - 8) + b(y - 4) + c(z - 3)$$

The partial derivatives of L_f , as you have written it so far, are precisely these constants a , b and c . So to force our function to have the same partial derivative information as f at the point $(8, 4, 3)$, we just need to set these constants equal to the corresponding partial derivatives of f at this point.

Step 3: Compute each partial derivative of $f(x, y, z) = ze^{x^2 - y^3}$

$$f_x(x, y, z) = \text{[input box]}$$

$$f_y(x, y, z) = \text{[input box]}$$

$$f_z(x, y, z) = \text{[input box]}$$

Check

[\[Hide explanation\]](#)

$$f_x(x, y, z) = \frac{\partial}{\partial x}(ze^{x^2 - y^3}) = ze^{x^2 - y^3}(2x) = 2xe^{x^2 - y^3}$$

$$f_y(x, y, z) = \frac{\partial}{\partial y}(ze^{x^2 - y^3}) = ze^{x^2 - y^3}(-3y^2) = -3y^2ze^{x^2 - y^3}$$

$$f_z(x, y, z) = \frac{\partial}{\partial z}(ze^{x^2 - y^3}) = e^{x^2 - y^3}$$

Now we evaluate each of these at $(8, 4, 3)$.

$$f_x(8, 4, 3) = \text{[input box]}$$

$$f_y(8, 4, 3) = \text{[input box]}$$

$$f_z(8, 4, 3) = \boxed{}$$

Check

[\[Hide explanation\]](#)

Luckily, our computations are made easier by the fact that $e^{8^2-4^3} = e^{64-64} = e^0 = 1$.

$$f_x(8, 4, 3) = 2(8)(3)e^{8^2-4^3} = 48$$

$$f_y(8, 4, 3) = -3(4^2)(3)e^{8^2-4^3} = -144$$

$$f_z(8, 4, 3) = e^{8^2-4^3} = 1$$

Step 4: Replacing the constants a , b and c in the expression of L_f with these partial derivative values, what do you get?

$$L_f(x, y, z) = \boxed{}$$

Check

[\[Hide explanation\]](#)

$$L_f(x, y, z) = 3 + 48(x - 8) - 144(y - 4) + 1(z - 3)$$

Now notice what this looks like if you write it with vector notation.

[\[Hide explanation\]](#)

$$\begin{aligned} L_f \left(\begin{bmatrix} x \\ y \\ z \end{bmatrix} \right) &= 3 + \begin{bmatrix} 48 \\ -144 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} \\ &= f \left(\begin{bmatrix} 8 \\ 4 \\ 3 \end{bmatrix} \right) + \begin{bmatrix} f_x(8, 4, 3) \\ f_y(8, 4, 3) \\ f_z(8, 4, 3) \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} \\ &= f \left(\begin{bmatrix} 8 \\ 4 \\ 3 \end{bmatrix} \right) + \nabla f \left(\begin{bmatrix} 8 \\ 4 \\ 3 \end{bmatrix} \right) \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} \end{aligned}$$

It is just a specific form of the general formula shown above.

$$L_f(\mathbf{x}) = \underbrace{f(\mathbf{x}_0)}_{\text{Constant}} + \underbrace{\nabla f(\mathbf{x}_0)}_{\text{Constant vector}} \cdot \overbrace{(\mathbf{x} - \mathbf{x}_0)}^{\mathbf{x} \text{ is the variable}}$$

Example 2: Using local linearization for estimation

What follows is by no means a practical application, but working through it will help give a feel for what local linearization is doing.

Problem: Suppose you are on a desert island without a calculator, and you need to estimate $\sqrt{2.01 + \sqrt{0.99 + \sqrt{9.01}}}$. How would you do it?

Solution:

We can view this problem as evaluating a certain three-variable function at the point $(2.01, 0.99, 9.01)$, namely

$$f(x, y, z) = \sqrt{x + \sqrt{y + \sqrt{z}}}$$

I don't know about you, but I'm not sure how to evaluate square roots by hand. If only this function was linear! Then working it out by hand would only involve adding and multiplying numbers. What we could do is find the local linearization at a nearby point where evaluating f is easier. Then we can get close to the right answer by evaluating the linearization at the point $(2.01, 0.99, 9.01)$.

The point we care about is very close to the much simpler point $(2, 1, 9)$, so we find the local linearization of f near that point. As before, we must find

- $f(2, 1, 9)$
- All partial derivatives of f at $(2, 1, 9)$

The first of these is

$$\begin{aligned} f(2, 1, 9) &= \sqrt{2 + \sqrt{1 + \sqrt{9}}} \\ &= \sqrt{2 + \sqrt{1 + 3}} \\ &= \sqrt{2 + \sqrt{4}} \end{aligned}$$

$$\begin{aligned}
&= \sqrt{2+2} \\
&= \sqrt{4} \\
&= 2
\end{aligned}$$

Looks like someone chose a few convenient input values, eh?

On to the partial derivatives (heavy sigh). Since the square roots are abundant, let's write out for ourselves the derivative of \sqrt{x} .

$$\frac{d}{dx}\sqrt{x} = \frac{d}{dx}x^{\frac{1}{2}} = \frac{1}{2}x^{-\frac{1}{2}} = \frac{1}{2\sqrt{x}}$$

Okay, here we go. The simplest partial derivative is f_x

$$f_x = \frac{\partial}{\partial x} \sqrt{x + \sqrt{y + \sqrt{z}}} = \frac{1}{2\sqrt{x + \sqrt{y + \sqrt{z}}}}$$

Since y is nestled in there, f_y requires some chain rule action:

$$f_y = \frac{\partial}{\partial y} \sqrt{x + \sqrt{y + \sqrt{z}}} = \frac{1}{2\sqrt{x + \sqrt{y + \sqrt{z}}}} \cdot \frac{1}{2\sqrt{y + \sqrt{z}}}$$

Nestled even deeper, that tricky z will require two iterations of the chain rule:

$$f_z = \frac{\partial}{\partial z} \sqrt{x + \sqrt{y + \sqrt{z}}} = \frac{1}{2\sqrt{x + \sqrt{y + \sqrt{z}}}} \cdot \frac{1}{2\sqrt{y + \sqrt{z}}} \cdot \frac{1}{2\sqrt{z}}$$



Next, evaluate each one of these at $(2, 1, 9)$. This might seem like a lot, but they are all made up of the same three basic components:

$$\begin{aligned}
\frac{1}{2\sqrt{x + \sqrt{y + \sqrt{z}}}} &= \frac{1}{2\sqrt{2 + \sqrt{1 + \sqrt{9}}}} = \frac{1}{2\sqrt{2+2}} = \frac{1}{4} \\
\frac{1}{2\sqrt{y + \sqrt{z}}} &= \frac{1}{2\sqrt{1 + \sqrt{9}}} = \frac{1}{2\sqrt{4}} = \frac{1}{4} \\
\frac{1}{2\sqrt{z}} &= \frac{1}{2\sqrt{9}} = \frac{1}{6}
\end{aligned}$$

Plugging these values into our expressions for the partial derivatives, we have

$$f_x(2, 1, 9) = \frac{1}{4}$$

$$f_y(2, 1, 9) = \frac{1}{4} \cdot \frac{1}{4} = \frac{1}{16}$$

$$f_z(2, 1, 9) = \frac{1}{4} \cdot \frac{1}{4} \cdot \frac{1}{6} = \frac{1}{96}$$

Unraveling the formula for local linearization, we get

$$\begin{aligned} L_f(\mathbf{x}) &= f(\mathbf{x}_0) + \nabla f(\mathbf{x}_0) \cdot (\mathbf{x} - \mathbf{x}_0) \\ &= f(\mathbf{x}_0) + f_x(\mathbf{x}_0)(x - x_0) + f_y(\mathbf{x}_0)(y - y_0) + f_z(\mathbf{x}_0)(z - z_0) \\ &= \boxed{2 + \frac{1}{4}(x - 2) + \frac{1}{16}(y - 1) + \frac{1}{96}(z - 9)} \end{aligned}$$

Finally, after all this work, we can plug in $(x, y, z) = (2.01, 0.99, 9.01)$ to compute our approximation

$$\begin{aligned} &2 + \frac{1}{4}(2.01 - 2) + \frac{1}{16}(0.99 - 1) + \frac{1}{96}(9.01 - 9) \\ &= 2 + \frac{0.01}{4} + \frac{-0.01}{16} + \frac{0.01}{96} \end{aligned}$$

Calculating this by hand still isn't *easy*, but at least it's doable. When you work it out, the final answer is

$$\boxed{2.001979}$$

Had we just used a calculator, the answer is

$$\sqrt{2.01 + \sqrt{0.99 + \sqrt{9.01}}} \approx \boxed{2.001978}$$

So our approximation is pretty good!

Why do we care?

Although it is not common to find yourself estimating square roots on a desert island (at least where I'm from), what *is* common in the contexts of math and engineering is wrangling with complicated but differentiable functions. The phrase "just linearize it" is tossed around so much that not knowing what it means could be awkward.

Remember, a **local linearization approximates one function near a point based on the information you can get from its derivative(s) at that point**. Even though you can use a computer to evaluate functions, that's not always enough.

- You might need to evaluate it many thousands of times per second, and working it out in full takes too long.
- Maybe you don't even have the function explicitly written out, and you just have a few measurements near a point which you wish to extrapolate.
- Sometimes what you care about is the inverse function, which can be hard or even impossible to find for the function as a whole, whereas inverting linear functions is relatively straight-forward.

Summary

- Local linearization generalizes the idea of tangent planes to any multivariable function.
- The idea is to approximate a function near one of its inputs with a simpler function that has the same value at that input, as well as the same partial derivative values.
- Written with vectors, here's what the approximation function looks like:

$$L_f(\mathbf{x}) = \underbrace{f(\mathbf{x}_0)}_{\text{Constant}} + \underbrace{\nabla f(\mathbf{x}_0)}_{\text{Constant vector}} \cdot \overbrace{(\mathbf{x} - \mathbf{x}_0)}^{\mathbf{x} \text{ is the variable}}$$

- This is called the local linearization of f near \mathbf{x}_0 .