

# Quadratic approximation

 Google Classroom

---

*Quadratic approximations extend the notion of a local linearization, giving an even closer approximation of a function.*

---

## Background:

- [Local linearization](#)
- [Graphs](#)
- [Second partial derivatives](#)

## What we're building to

The goal, as with a local linearization, is to approximate a potentially complicated multivariable function  $f$  near some input, which I'll write as the vector  $\mathbf{x}_0$ . A quadratic approximation does this more tightly than a local linearization, using the information given by second partial derivatives.

### Non-vector form

In the specific case where the input of  $f$  is two dimensional, and you are approximating near a point  $(x_0, y_0)$ , you will see below that the quadratic approximation ends up looking like this:

$$Q_f(x, y) = f(x_0, y_0) +$$

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

$$\frac{1}{2} f_{xx}(x_0, y_0)(x - x_0)^2 +$$

$$f_{xy}(x_0, y_0)(x - x_0)(y - y_0) +$$

$$\frac{1}{2} f_{yy}(x_0, y_0)(y - y_0)^2$$

### Vector form:

The general form of this, for a scalar-valued function  $f$  with any kind of multidimensional input, here's what that approximation looks like:

$$Q_f(\mathbf{x}) = \underbrace{f(\mathbf{x}_0)}_{\text{Constant}} + \underbrace{\nabla f(\mathbf{x}_0) \cdot (\mathbf{x} - \mathbf{x}_0)}_{\text{Linear term}} + \underbrace{\frac{1}{2}(\mathbf{x} - \mathbf{x}_0)^T \mathbf{H}_f(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0)}_{\text{Quadratic term}}$$

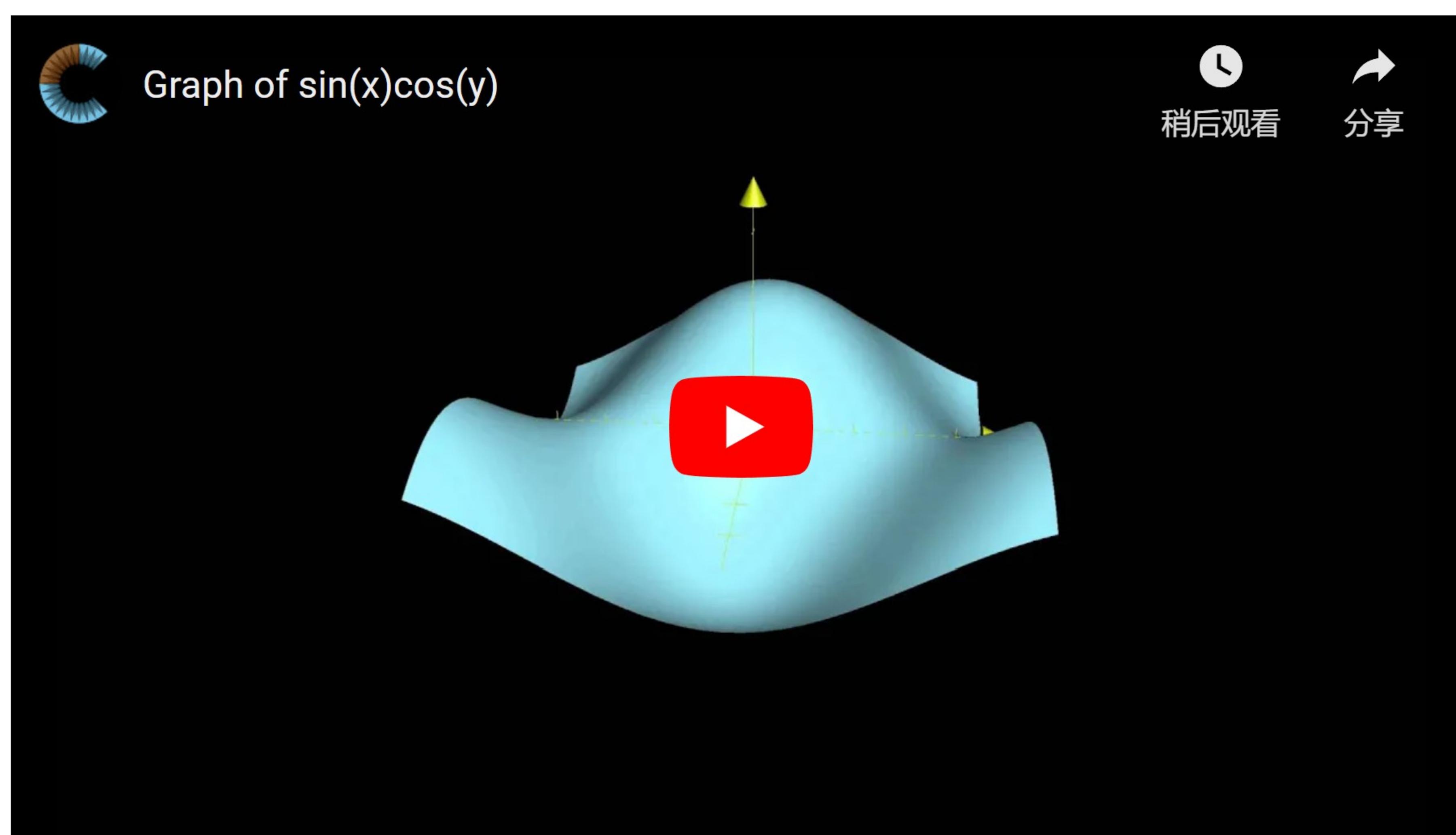
I know it looks a bit complicated, but I'll step through it piece by piece later on. Here's a brief outline of each term.

- $f$  is a function with multi-dimensional input and a scalar output.
- $\nabla f(\mathbf{x}_0)$  is the gradient of  $f$  evaluated at  $\mathbf{x}_0$ .
- $\mathbf{H}_f(\mathbf{x}_0)$  is the Hessian matrix of  $f$  evaluated at  $\mathbf{x}_0$ .
- The vector  $\mathbf{x}_0$  is a specific input, the one we are approximating near.
- The vector  $\mathbf{x}$  represents the variable input.
- The approximation function,  $Q_f$ , has the same value as  $f$  at the point  $\mathbf{x}_0$ , all its partial derivatives have the same value as those of  $f$  at this point, and all its *second* partial derivatives have the same value as those of  $f$  at this point.

## Tighter and tighter approximations

Imagine you are given some function  $f(x, y)$  with two inputs and one output, such as

$$f(x, y) = \sin(x) \cos(y)$$



The goal is to find a simpler function that approximates  $f(x, y)$  near some particular point  $(x_0, y_0)$ . For example,

$$(x_0, y_0) = \left(\frac{\pi}{3}, \frac{\pi}{6}\right)$$

## Zero-order approximation

The most naive approximation would be a constant function which equals the value of  $f$  at  $(x_0, y_0)$  everywhere. We call this a "0-order approximation".

In the example:

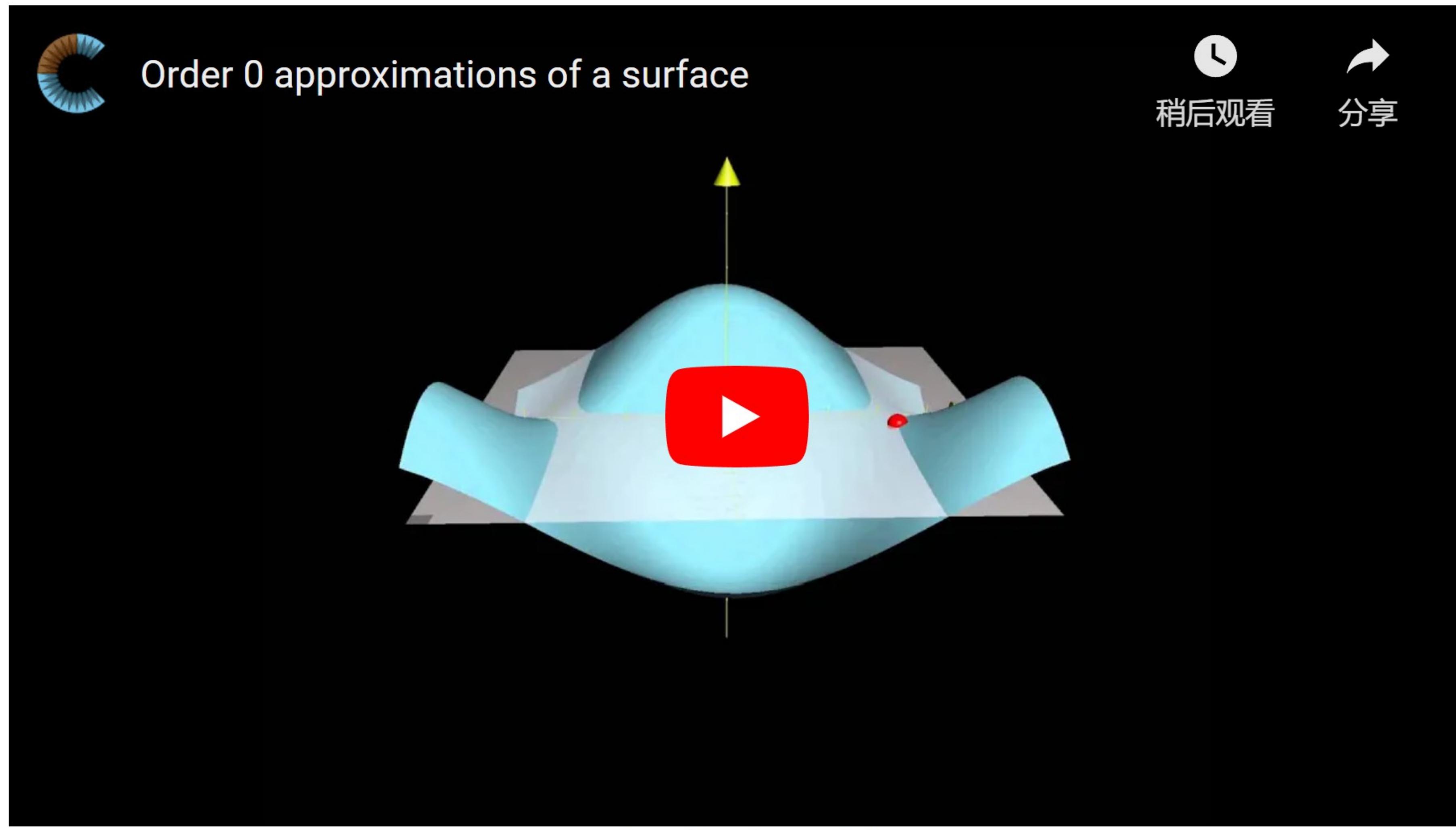
$$\begin{aligned} C(x, y) &= \sin\left(\frac{\pi}{3}\right) \cos\left(\frac{\pi}{6}\right) \\ &= \left(\frac{\sqrt{3}}{2}\right) \frac{\sqrt{3}}{2} \\ &= \frac{3}{4} \end{aligned}$$

Written in the abstract:

$$C(x, y) = f(x_0, y_0) \quad \leftarrow \text{Constant function}$$

Graphically:

The graph of this approximation function  $C(x, y)$  is a flat plane passing through the graph of our function at the point  $(x_0, y_0, f(x_0, y_0))$ . Below is a video showing how this approximation changes as we move the point  $(x_0, y_0)$  around.



[See video transcript](#)

The graph of  $f$  is pictured in blue, the graph of the approximation is white, and the point  $(x_0, y_0, f(x_0, y_0))$  is pictured as a red dot.

## First-order approximation

The constant function zero-order approximation is pretty lousy. Sure, it is guaranteed to equal  $f(x, y)$  at the point  $(x_0, y_0)$ , but that's about it. One step better is to use a [local linearization](#), also known as a "First-order approximation".

**In the example:**

$$L_f(x, y) = \frac{3}{4} + \frac{\sqrt{3}}{4} \left( x - \frac{\pi}{3} \right) + \frac{-\sqrt{3}}{4} \left( y - \frac{\pi}{6} \right)$$

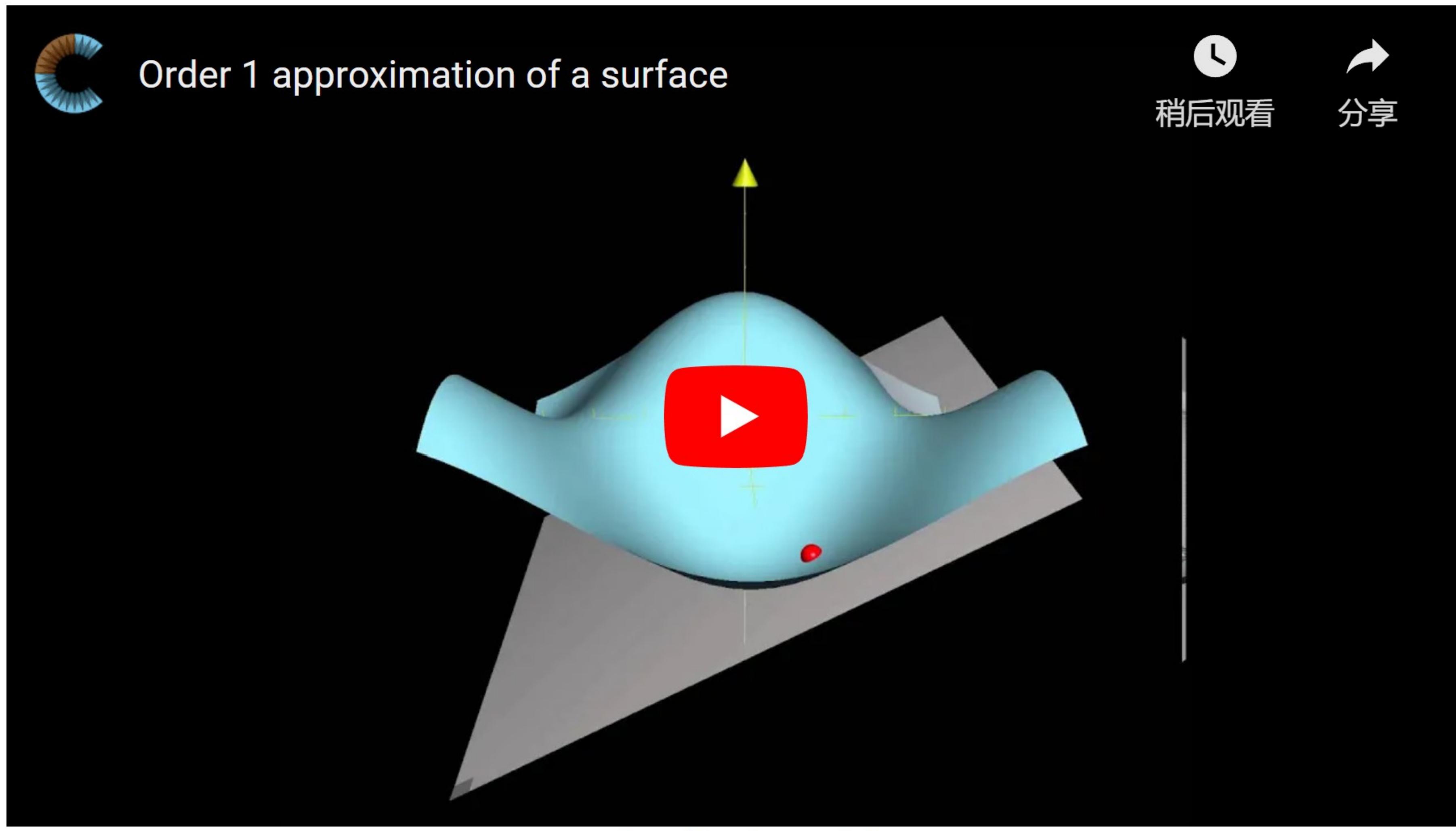
**Written in the abstract:**

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

Here,  $f_x$  and  $f_y$  denote the [partial derivatives](#) of  $f$ .

**Graphically:**

The graph of a local linearization is the plane **tangent** to the graph of  $f$  at the point  $(x_0, y_0, f(x_0, y_0))$ . Here is a video showing how this approximation changes as we move around the point  $(x_0, y_0)$ :

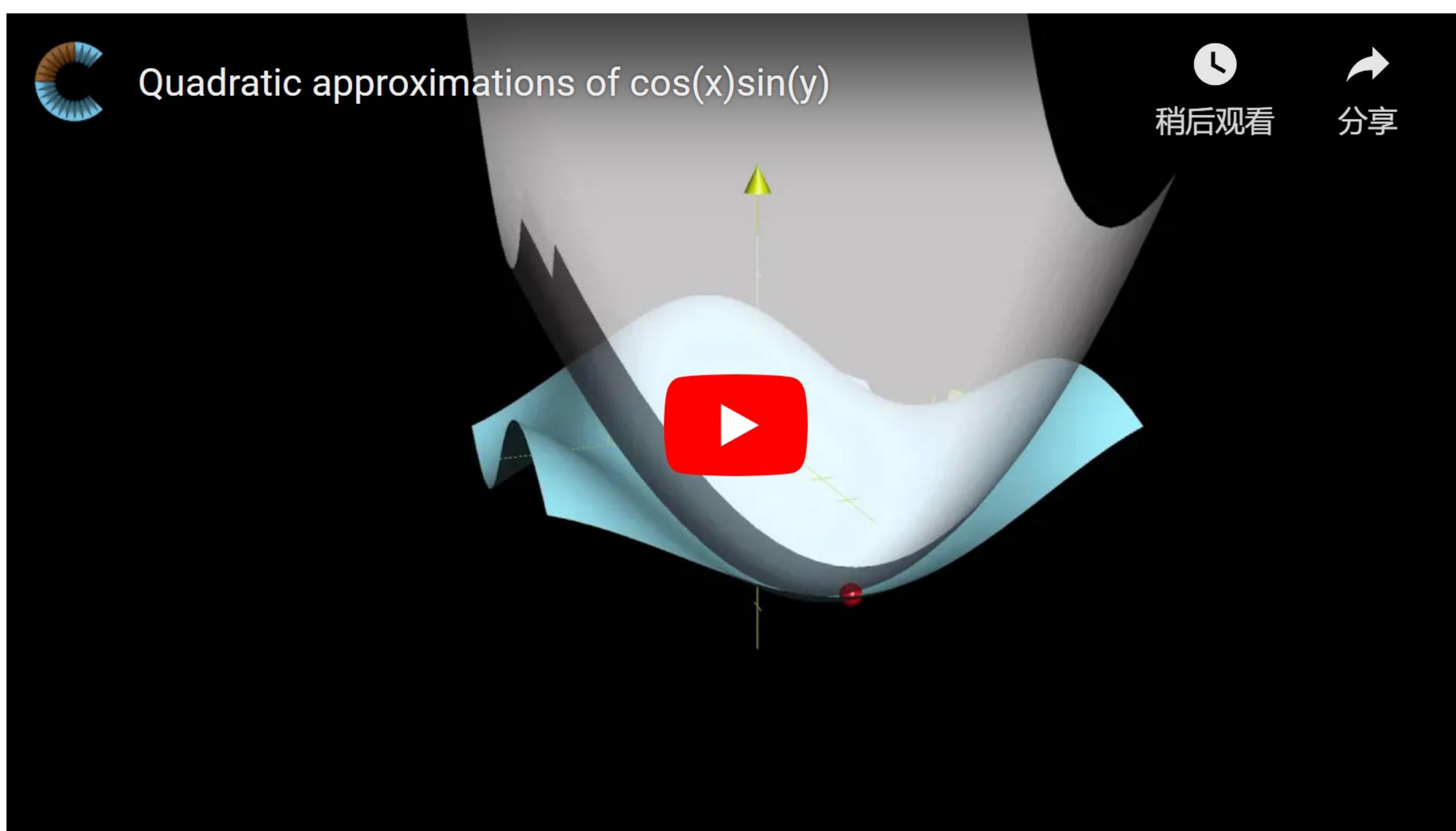


[See video transcript](#)

## Second-order approximation.

Better still is a **quadratic approximation**, also called a "second-order approximation".

The remainder of this article is devoted to finding and understanding the analytic form of such an approximation, but before diving in, let's see what such approximations look like graphically. You can think of these approximations as nestling into the curves the graph at the point  $(x_0, y_0, f(x_0, y_0))$ , giving it a sort of mathematical hug.



[See video transcript](#)

**"Quadratic" means product of two variables**

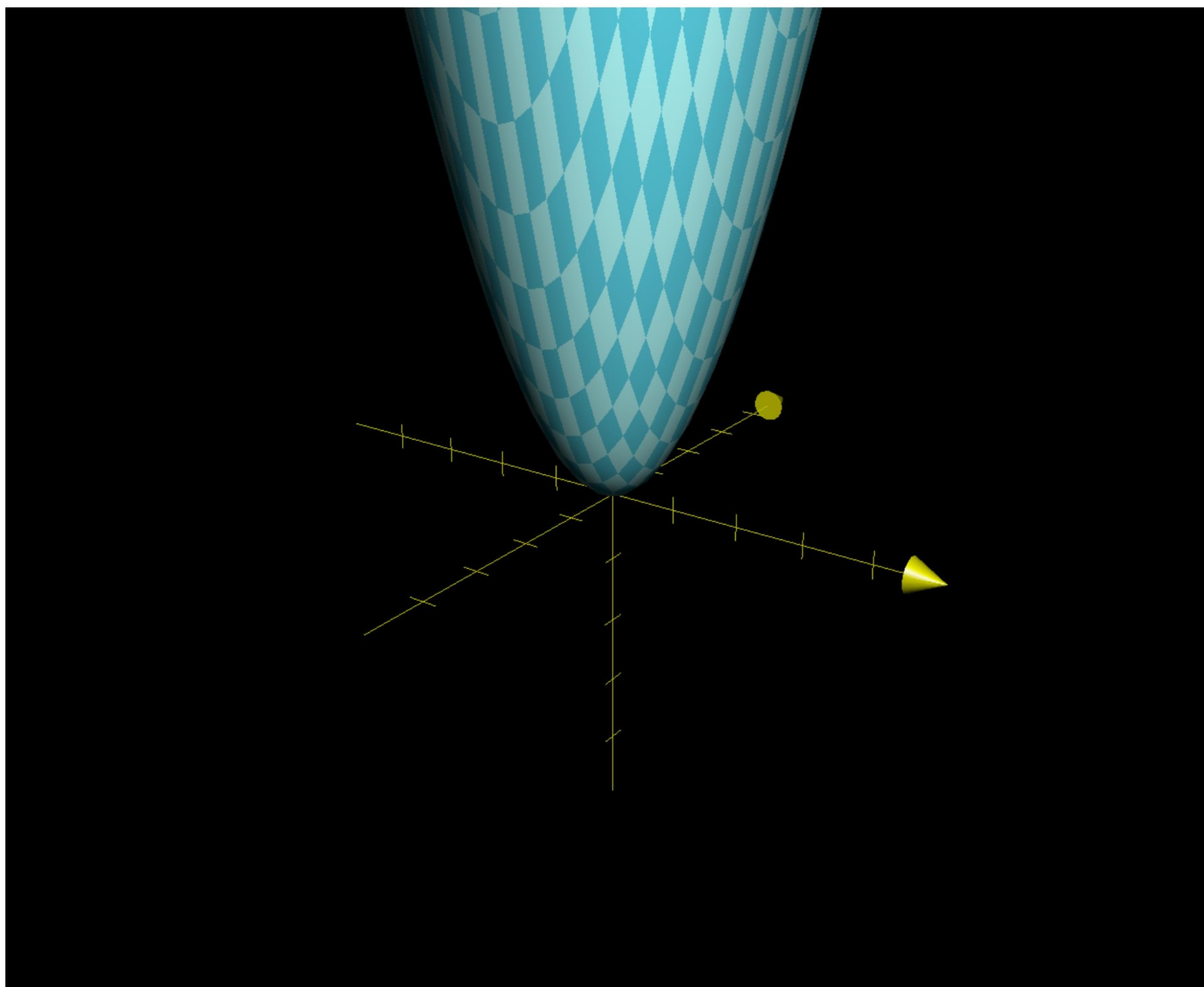
In single variable functions, the word "quadratic" refers to any situation where a variable is squared as in the term  $x^2$ . With multiple variables, "quadratic" refers not only to square terms, like  $x^2$  and  $y^2$ , but also terms that involve the product of two separate variables, such as  $xy$ .

In general, the "order" of a term which is the product of several things, such as  $3x^2y^3$ , is the total number of *variables* multiplied into that term. In this case, the order would be 5: Two  $x$ 's, three  $y$ 's, and the constant doesn't matter.

## Graphs of quadratic functions

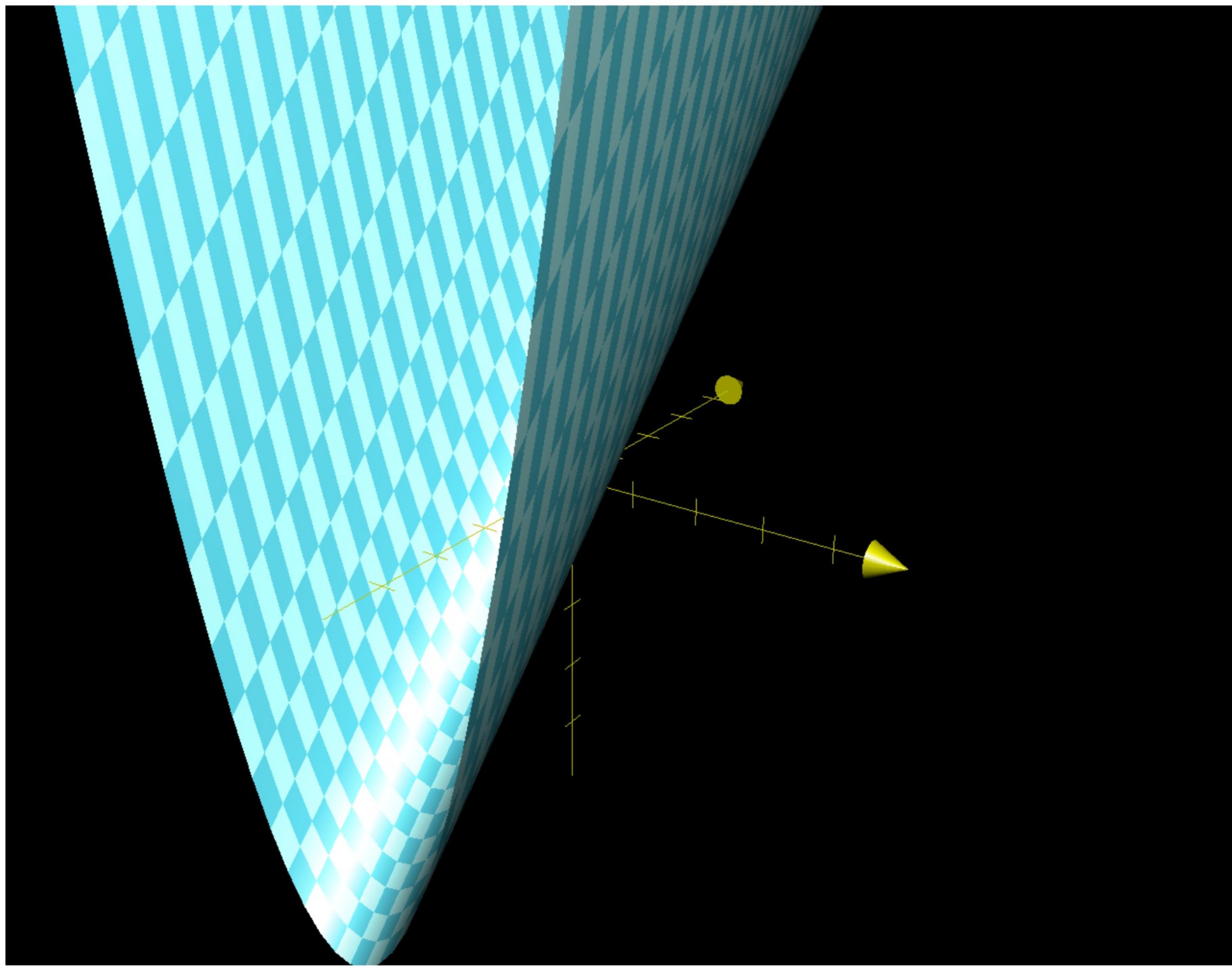
One way to think of quadratic functions is in terms of their **concavity**, which might depend on which direction you are moving in.

If the function has an upward concavity, as is the case, for example, with  $f(x, y) = x^2 + y^2$ , the graph will look something like this:

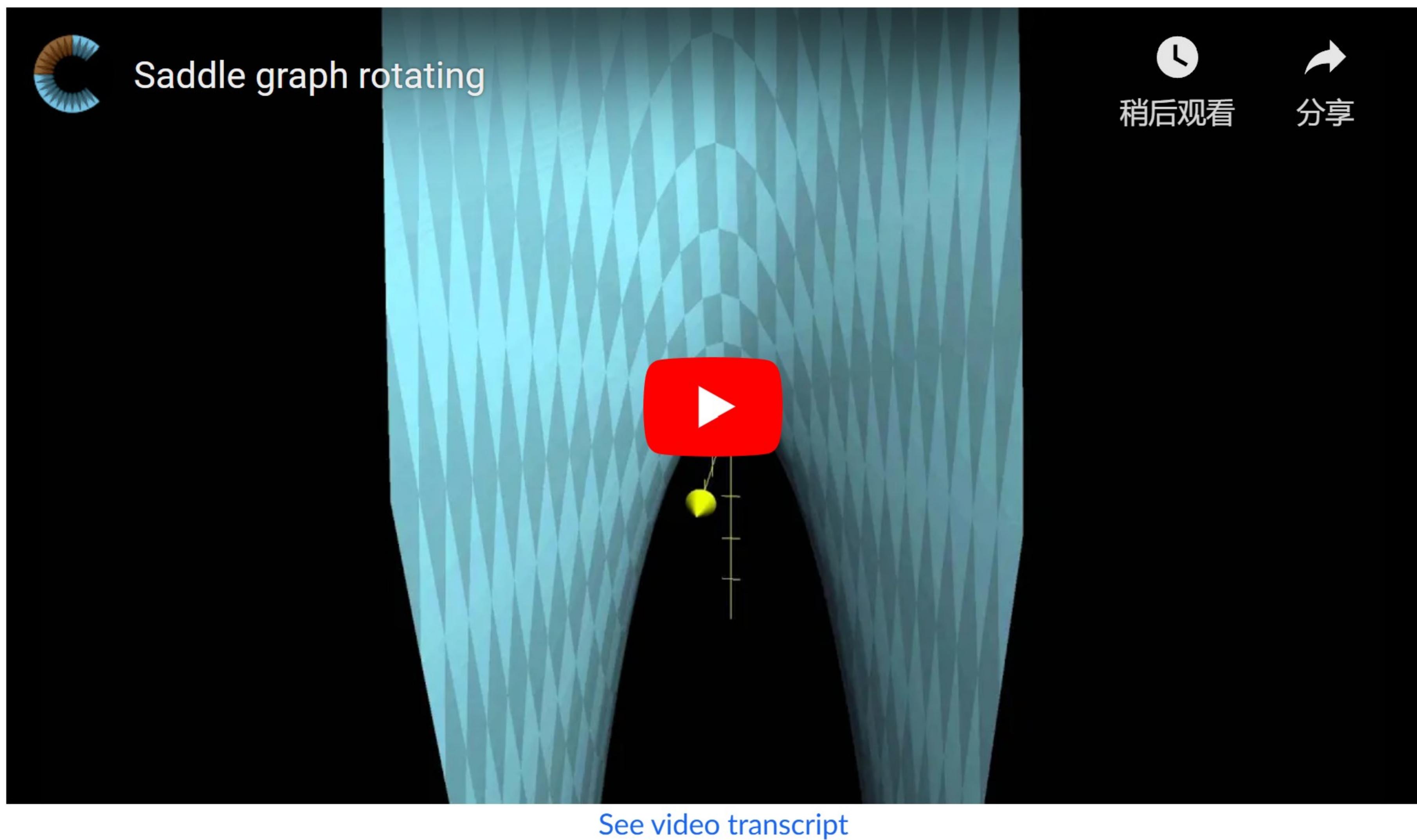


This shape, which is a three-dimensional parabola, goes by the name **paraboloid**.

If the function is concave up in one direction and linear in another, the graph looks like a parabolic curve has been dragged through space to trace out a surface. For example this happens in the case of  $f(x, y) = x^2 + y$ :



Finally, if the graph is concave up when traveling in one direction, but concave down when traveling in another direction, as is the case for  $f(x, y) = x^2 - y^2$ , the graph looks a bit like a saddle. Here's what such a graph looks like:



## Reminder on the local linearization recipe

To actually write down a quadratic approximation of a function  $f$  near the point  $(x_0, y_0)$ , we build up from the local linearization:

$$L_f(x, y) = \underbrace{f(x_0, y_0)}_{\text{Constant term}} + \underbrace{f_x(x_0, y_0)(x - x_0) + f_y(x_0, y_0)(y - y_0)}_{\text{Linear terms}}$$

It's worth walking through the recipe for finding the local linearization one more time since the recipe for finding a quadratic approximation is very similar.

- Start with the constant term  $f(x_0, y_0)$ , so that our approximation at least matches  $f$  at the point  $(x_0, y_0)$ .
- Add on linear terms  $f_x(x_0, y_0)(x - x_0)$  and  $f_y(x_0, y_0)(y - y_0)$ .
- Use the constants  $f_x(x_0, y_0)$  and  $f_y(x_0, y_0)$  to ensure that our approximation has the same partial derivatives as  $f$  at the point  $(x_0, y_0)$ .
- Use the terms  $(x - x_0)$  and  $(y - y_0)$  instead of simply  $x$  and  $y$  so that we don't mess up the fact that our approximation equals  $f(x_0, y_0)$  at the point  $(x_0, y_0)$ .

## Finding the quadratic approximation

For the quadratic approximation, we add on the quadratic terms  $(x - x_0)^2$ ,  $(x - x_0)(y - y_0)$ , and  $(y - y_0)^2$ , and for now we write their coefficients as the constants  $a$ ,  $b$  and  $c$  which we will solve for in a moment:

$$Q_f(x, y) = \underbrace{f(x_0, y_0)}_{\text{Order 0 part}} + \underbrace{f_x(x_0, y_0)(x - x_0) + f_y(x_0, y_0)(y - y_0)}_{\text{Order 1 part}} + \underbrace{a(x - x_0)^2 + b(x - x_0)(y - y_0) + c(y - y_0)^2}_{\text{Quadratic part}}$$

In the same way that we made sure that the local linearization has the same partial derivatives as  $f$  at  $(x_0, y_0)$ , we want the quadratic approximation to have the same second partial derivatives as  $f$  at this point.

The really nice thing about the way I wrote  $Q_f$  above is that the second partial derivative  $\frac{\partial^2 Q_f}{\partial x^2}$  depends only on the  $a(x - x_0)^2$  term.

- Try it! Take the second partial derivative with respect to  $x$  of every term in the expression of  $Q_f(x, y)$  above, and notice that they all go to zero except for the  $a(x - x_0)^2$  term.

Did you really try it? I'm serious, take a moment to reason through it. It really helps in understanding why  $Q_f$  is expressed the way it is.

This fact is nice because rather than taking the second partial derivative of the entire monstrous expression, you can view it like this:

$$\begin{aligned}\frac{\partial^2 Q_f}{\partial x^2}(x, y) &= (\text{A bunch of 0's}) + \frac{\partial^2}{\partial x^2} a(x - x_0)^2 + (\text{more 0's}) \\ &= \frac{\partial}{\partial x} 2a(x - x_0) \\ &= 2a\end{aligned}$$

Since the goal is for this to match  $f_{xx}(x, y)$  at the point  $(x_0, y_0)$ , you can solve for  $a$  like this:

$$a = \frac{1}{2} f_{xx}(x_0, y_0)$$

**Test yourself:** Use similar reasoning to figure out what the constants  $b$  and  $c$  should be.

[\[Hide explanation\]](#)

The mixed partial derivative  $\frac{\partial^2 Q_f}{\partial x \partial y}$  depends only on the  $b(x - x_0)(y - y_0)$  term (try it!), so we have

$$\begin{aligned}\frac{\partial^2 Q_f}{\partial x \partial y} &= (\text{A bunch of 0's}) + \frac{\partial^2}{\partial x \partial y} b(x - x_0)(y - y_0) + (\text{more 0's}) \\ &= \frac{\partial}{\partial x} b(x - x_0) \\ &= b\end{aligned}$$

Since we want this mixed partial derivative of our approximation to equal the mixed partial derivative of  $f$  at this point,  $f_{xy}(x_0, y_0)$ , we say

$$b = f_{xy}(x_0, y_0)$$

Nearly identical reasoning will lead us to the conclusion

$$\boxed{\mathbf{c} = \frac{1}{2} f_{yy}(x_0, y_0)}$$

We can now write our final quadratic approximation, with all six of its terms working in harmony to mimic the behavior of  $f$  at  $(x_0, y_0)$ :

$$Q_f(x, y) = f(x_0, y_0) +$$

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

$$\frac{1}{2} f_{xx}(x_0, y_0)(x - x_0)^2 +$$

$$f_{xy}(x_0, y_0)(x - x_0)(y - y_0) +$$

$$\frac{1}{2} f_{yy}(x_0, y_0)(y - y_0)^2$$

## Example: Approximating $\sin(x) \cos(y)$

To see this beast in action, let's try it out on the function from the introduction.

---

**Problem:** Find the quadratic approximation of

$$f(x, y) = \sin(x) \cos(y)$$

about the point  $(x, y) = \left(\frac{\pi}{3}, \frac{\pi}{6}\right)$ .

---

**Solution:**

To collect all the necessary information, you need to evaluate  $f(x, y) = \sin(x) \cos(y)$  and all of its partial derivatives and all of its second partial derivatives at the point  $\left(\frac{\pi}{3}, \frac{\pi}{6}\right)$ .

$$f\left(\frac{\pi}{3}, \frac{\pi}{6}\right) = \boxed{\phantom{00}}$$

**Check**

[\[Hide explanation\]](#)

$$f\left(\frac{\pi}{3}, \frac{\pi}{6}\right) = \sin\left(\frac{\pi}{3}\right) \cos\left(\frac{\pi}{6}\right)$$

$$= \left(\frac{\sqrt{3}}{2}\right) \frac{\sqrt{3}}{2}$$

$$= \frac{3}{4}$$

$$f_x(x, y) = \boxed{\phantom{00}}$$

$$f_x\left(\frac{\pi}{3}, \frac{\pi}{6}\right) = \boxed{\phantom{00}}$$

**Check**

[\[Hide explanation\]](#)

$$f_x(x, y) = \frac{\partial}{\partial \textcolor{teal}{x}} \sin(\textcolor{teal}{x}) \cos(y)$$

$$= \cos(\textcolor{teal}{x}) \cos(y)$$

$$f_x\left(\frac{\pi}{3}, \frac{\pi}{6}\right) = \cos\left(\frac{\pi}{3}\right) \cos\left(\frac{\pi}{6}\right)$$

$$= \left(\frac{1}{2}\right) \frac{\sqrt{3}}{2}$$

$$= \frac{\sqrt{3}}{4}$$

$$f_y(x, y) = \boxed{\phantom{00}}$$

$$f_y\left(\frac{\pi}{3}, \frac{\pi}{6}\right) = \boxed{\phantom{00}}$$

**Check**

[\[Hide explanation\]](#)

$$f_y(x, y) = \frac{\partial}{\partial y} \sin(x) \cos(\textcolor{red}{y})$$

$$= -\sin(x) \sin(\textcolor{red}{y})$$

$$f_y\left(\frac{\pi}{3}, \frac{\pi}{6}\right) = -\sin\left(\frac{\pi}{3}\right) \sin\left(\frac{\pi}{6}\right)$$

$$= -\left(\frac{\sqrt{3}}{2}\right) \frac{1}{2}$$

$$= \frac{-\sqrt{3}}{4}$$

$$f_{xx}(x, y) = \boxed{\phantom{00}}$$

$$f_{xx}\left(\frac{\pi}{3}, \frac{\pi}{6}\right) = \boxed{\phantom{00}}$$

[Check](#)

[\[Hide explanation\]](#)

$$f_{xx}(x, y) = \frac{\partial}{\partial \textcolor{teal}{x}} \frac{\partial}{\partial \textcolor{teal}{x}} (\sin(\textcolor{teal}{x}) \cos(y))$$

$$= \frac{\partial}{\partial \textcolor{teal}{x}} \cos(\textcolor{teal}{x}) \cos(y)$$

$$= -\sin(\textcolor{teal}{x}) \cos(y)$$

$$f_{xx}\left(\frac{\pi}{3}, \frac{\pi}{6}\right) = -\sin\left(\frac{\pi}{3}\right) \cos\left(\frac{\pi}{6}\right)$$

$$= -\left(\frac{\sqrt{3}}{2}\right) \frac{\sqrt{3}}{2}$$

$$= \frac{-3}{4}$$

$$f_{xy}(x, y) = \boxed{\phantom{00}}$$

$$f_{xy}\left(\frac{\pi}{3}, \frac{\pi}{6}\right) = \boxed{\phantom{00}}$$

[Check](#)

[\[Hide explanation\]](#)

$$f_{xy}(x, y) = \frac{\partial}{\partial \textcolor{teal}{x}} \frac{\partial}{\partial \textcolor{red}{y}} (\sin(\textcolor{teal}{x}) \cos(\textcolor{red}{y}))$$

$$= \frac{\partial}{\partial \textcolor{teal}{x}} (\sin(\textcolor{teal}{x})(-\sin(\textcolor{red}{y})))$$

$$= -\cos(\textcolor{teal}{x}) \sin(\textcolor{red}{y})$$

$$f_{xy}\left(\frac{\pi}{3}, \frac{\pi}{6}\right) = -\cos\left(\frac{\pi}{3}\right) \sin\left(\frac{\pi}{6}\right)$$

$$= -\left(\frac{1}{2}\right) \frac{1}{2}$$

$$= \frac{-1}{4}$$

$$f_{yy}(x, y) = \boxed{\phantom{00}}$$

$$f_{yy}\left(\frac{\pi}{3}, \frac{\pi}{6}\right) = \boxed{\phantom{00}}$$

[Check](#)

[\[Hide explanation\]](#)

$$f_{yy}(x, y) = \frac{\partial}{\partial \textcolor{red}{y}} \frac{\partial}{\partial \textcolor{red}{y}} (\sin(x) \cos(\textcolor{red}{y}))$$

$$= \frac{\partial}{\partial \textcolor{red}{y}} (-\sin(x) \sin(\textcolor{red}{y}))$$

$$= -\sin(x) \cos(\textcolor{red}{y})$$

$$f_{yy}\left(\frac{\pi}{3}, \frac{\pi}{6}\right) = -\sin\left(\frac{\pi}{3}\right) \cos\left(\frac{\pi}{6}\right)$$

$$= -\left(\frac{\sqrt{3}}{2}\right) \frac{\sqrt{3}}{2}$$

$$= -\frac{3}{4}$$

Almost there! As a final step, apply all these values to the formula for a quadratic approximation.

[\[Hide explanation\]](#)

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

$$\frac{1}{2}f_{xx}(x_0, y_0)(x - x_0)^2 +$$

$$f_{xy}(x_0, y_0)(x - x_0)(y - y_0) +$$

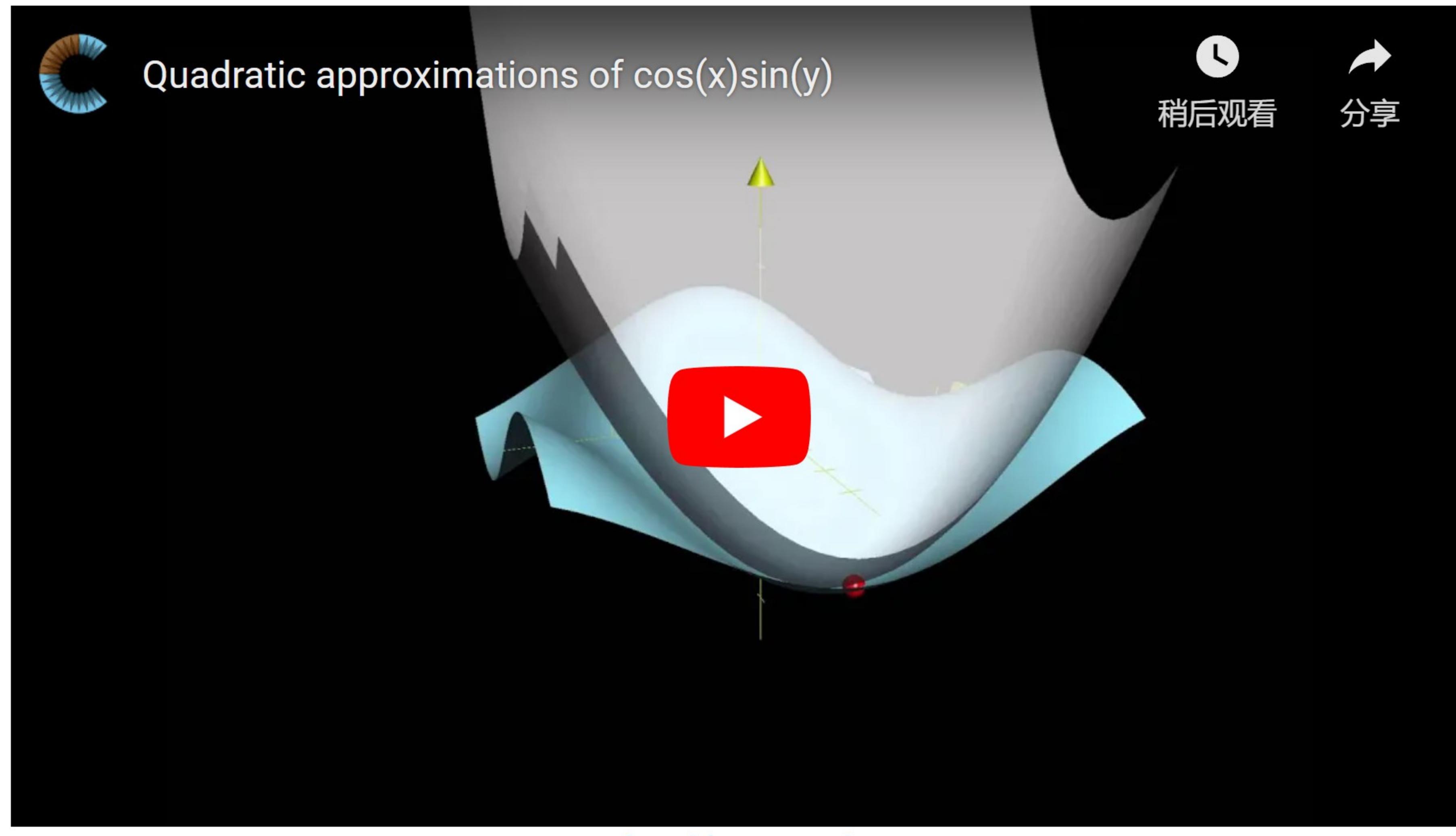
$$\frac{1}{2}f_{yy}(x_0, y_0)(y - y_0)^2$$

↓

$$= \frac{3}{4} + \frac{\sqrt{3}}{4} \left( x - \frac{\pi}{3} \right) + \frac{-\sqrt{3}}{4} \left( y - \frac{\pi}{6} \right) +$$

$$\frac{1}{2} \cdot \frac{-3}{4} \left( x - \frac{\pi}{3} \right)^2 + \frac{-1}{4} \left( x - \frac{\pi}{3} \right) \left( y - \frac{\pi}{6} \right) + \frac{1}{2} \cdot \frac{3}{4} \left( y - \frac{\pi}{6} \right)^2$$

So for example, to generate the animation of quadratic approximations, this is the formula I had to plug into the graphing software.



## Vector notation using the Hessian

Perhaps it goes without saying that the expression for the quadratic

approximation is long. Now imagine if  $f$  had three inputs,  $x$ ,  $y$  and  $z$ . In principle you can imagine how this might go, adding terms involving  $f_z$ ,  $f_{xz}$ ,  $f_{zz}$ , on and on with all 3 partial derivatives and all 9 second partial derivative. But this would be a total nightmare!

Now imagine you were writing a program to find the quadratic approximation of a function with 100 inputs. Madness!

It actually doesn't have to be that bad. When something is not that complicated in principle, it shouldn't be that complicated in notation. Quadratic approximations are a *little* complicated, sure, but they're not absurd.

Using vectors and matrices, specifically the [gradient](#) and [Hessian](#) of  $f$ , we can write the quadratic approximation  $Q_f$  as follows:

$$Q_f(\mathbf{x}) = \underbrace{f(\mathbf{x}_0)}_{\text{Constant}} + \underbrace{\nabla f(\mathbf{x}_0) \cdot (\mathbf{x} - \mathbf{x}_0)}_{\text{Linear term}} + \underbrace{\frac{1}{2}(\mathbf{x} - \mathbf{x}_0)^T \mathbf{H}_f(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0)}_{\text{Quadratic term}}$$

Let's break this down:

- The boldfaced  $\mathbf{x}$  represents the input variable(s) as a vector,

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ \vdots \end{bmatrix}$$

Moreover,  $\mathbf{x}_0$  is a particular vector in the input space. If this has two components, this formula for  $Q_f$  is just a different way to write the one we derived before, but it could also represent a vector with any other dimension.

- The dot product  $\nabla f(\mathbf{x}_0) \cdot (\mathbf{x} - \mathbf{x}_0)$  will expand into the sum of all terms of the form  $f_x(\mathbf{x}_0)(x - x_0)$ ,  $f_y(\mathbf{x}_0)(y - y_0)$ , etc. if this is not familiar from the vector notation for local linearization, work it out for yourself in the case of 2-dimensions to see!

[\[Hide explanation\]](#)

$$\begin{aligned}\nabla f(\mathbf{x}_0) \cdot (\mathbf{x} - \mathbf{x}_0) &= \begin{bmatrix} f_x(x_0, y_0) \\ f_y(x_0, y_0) \end{bmatrix} \cdot \left( \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \right) \\ &= \begin{bmatrix} f_x(x_0, y_0) \\ f_y(x_0, y_0) \end{bmatrix} \cdot \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix} \\ &= f_x(x_0, y_0)(x - x_0) + f_y(x_0, y_0)(y - y_0)\end{aligned}$$

- The little superscript  $T$  in the expression  $(\mathbf{x} - \mathbf{x}_0)^T$  indicates "transpose". This means you take the initial vector  $(\mathbf{x} - \mathbf{x}_0)$ , which looks something like this:

$$(\mathbf{x} - \mathbf{x}_0) = \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix}$$

Then you flip it, to get something like this:

$$(\mathbf{x} - \mathbf{x}_0)^T = \begin{bmatrix} x - x_0 & y - y_0 \end{bmatrix}$$

- $\mathbf{H}_f(\mathbf{x}_0)$  is the [Hessian](#) of  $f$ .
- The expression  $(\mathbf{x} - \mathbf{x}_0)^T \mathbf{H}_f(\mathbf{x}_0) (\mathbf{x} - \mathbf{x}_0)$  might seem complicated if you have never come across something like it before. This way of expressing quadratic terms is actually quite common in vector-calculus and vector-algebra, so it's worth expanding an expression like this at least a few times in your life. For example, try working it out in the case where  $\mathbf{x}$  is two-dimensional to see what it looks like.

[\[Hide explanation\]](#)

$$(\mathbf{x} - \mathbf{x}_0)^T \mathbf{H}_f(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0)$$

$$\begin{aligned}
&= \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix}^T \begin{bmatrix} f_{xx}(x_0, y_0) & f_{xy}(x_0, y_0) \\ f_{yx}(x_0, y_0) & f_{yy}(x_0, y_0) \end{bmatrix} \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix} \\
&= [(x - x_0) \quad (y - y_0)] \begin{bmatrix} f_{xx}(x_0, y_0) & f_{xy}(x_0, y_0) \\ f_{yx}(x_0, y_0) & f_{yy}(x_0, y_0) \end{bmatrix} \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix} \\
&= [(x - x_0) \quad (y - y_0)] \begin{bmatrix} f_{xx}(x_0, y_0)(x - x_0) + f_{xy}(x_0, y_0)(y - y_0) \\ f_{yx}(x_0, y_0)(x - x_0) + f_{yy}(x_0, y_0)(y - y_0) \end{bmatrix} \\
&= f_{xx}(x_0, y_0)(x - x_0)^2 + f_{xy}(x_0, y_0)(y - y_0)(x - x_0) \\
&\quad + f_{yx}(x_0, y_0)(x - x_0)(y - y_0) + f_{yy}(x_0, y_0)(y - y_0)^2 \\
&= f_{xx}(x_0, y_0)(x - x_0)^2 + 2f_{xy}(x_0, y_0)(y - y_0)(x - x_0) + f_{yy}(x_0, y_0)(y - y_0)^2
\end{aligned}$$

You should find that it is exactly 2 times the quadratic portion of the non-vectorized formula we derived above.

## What's the point?

In truth, it is a real pain to compute a quadratic approximation by hand, and it requires staying very organized to do so without making a little mistake. In practice, people rarely work through a quadratic approximation like the example above, but knowing how they work is useful for at least two broad reasons:

- **Computation:** Even if you never have to write out a quadratic approximation, you may one day need to program a computer to do it for a particular function. Or even if you are relying on someone else's program, you may need to analyze how and why the approximation is failing in some circumstance.
- **Theory:** Being able to reference a second-order approximation helps us to reason about the behavior of general functions near a point. This will be useful later in figuring out if a point is a local maximum or minimum.