Local approximations to surfaces

Bro. David E. Brown, BYU–Idaho Dept. of Mathematics. All rights reserved.

Version 0.31, of May 9, 2013

## Contents

1 **Review**  
   1.1 Real-valued functions of one variable ............................................. 2  
   1.2 Level curves .......................................................... 3  
   1.3 Vector-valued functions .................................................. 3  
   1.4 Miscellany ............................................................ 3  

2 **Local approximations to surfaces**  
   2.1 Local linear approximations to surfaces ...................................... 5  
   2.2 Local quadratic approximations to surfaces ................................... 6  
   2.3 Isn’t there a better way to write this stuff? ............................... 7  

3 **Local approximations to level surfaces**  

4 **Local approximations to parameterized surfaces**  

5 **The technical stuff**  


1 Review

There’s a lot of talk in Calculus about approximations. Let’s keep talking! In this document, we’ll address local linear and local quadratic approximations to a function of two or three variables. Let’s begin with a little review.

1.1 Real-valued functions of one variable

Recall the following definitions.

Definitions 1.1. Let \( y = f(x) \) be twice differentiable at \( x = x_0 \).

1. The increment of \( x \) is \( \Delta x = x - x_0 \).
2. The increment of \( y \) is \( \Delta y = f(x) - f(x_0) \).
3. The differential of \( x \) is \( dx = \Delta x \).
4. The differential of \( y \) is \( dy = f'(x_0)dx \). Note that \( dy \) is sometimes called \( df \).
5. The local linear approximation to \( y = f(x) \) at \( x = x_0 \) is
   \[
   f(x) \approx f(x_0) + f'(x_0)(x - x_0),
   \]
   for \( x \) sufficiently close to \( x_0 \).
6. The local quadratic approximation to \( y = f(x) \) at \( x = x_0 \) is
   \[
   f(x) \approx f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2}f''(x_0)(x - x_0)^2,
   \]
   for \( x \) sufficiently close to \( x_0 \).

Note that the local linear approximation can also be thought of as \( f(x) \approx f(x_0) + df \).

Typically, we think of \( \Delta x \) as being “small.” Notice that the \( dx \) that’s defined above is NOT the same “\( dx \)” that appears in the symbol \( \frac{dy}{dx} \), because this symbol is defined by \( \frac{dy}{dx} = \lim_{\Delta x \to 0} \frac{\Delta y}{\Delta x} \), and after you’ve taken the limit, \( \Delta x \) has gone to zero. But when we write “\( dx \)” by itself, thinking of it as some small change in \( x \), we haven’t made \( dx \) go to 0 yet. So a loose “\( dx \)” by itself is just a small change in \( x \). In other words, it’s a \( \Delta x \), or even just \( x - x_0 \). It’s not the bottom half of a derivative.

Exercise 1.1.1. Convince yourself that the local linear and quadratic approximations to \( y = e^x \) at \( x = 0 \) are
   \[
   y \approx 1 + x \quad \text{and} \quad y \approx 1 + x + \frac{x^2}{2},
   \]
   respectively.

There is, of course, a theorem (called the increment theorem) to the effect that if \( f \) is differentiable at \( x_0 \), then for \( x \) sufficiently close to \( x_0 \), \( \Delta y \approx dy \), that is
   \[
   f(x) - f(x_0) \approx f'(x_0)dx,
   \]
which is equivalent to the local linear approximation. A similar theorem says that if \( f \) is twice differentiable at \( x_0 \), then for \( x \) sufficiently close to \( x_0 \),
   \[
   f(x) - f(x_0) \approx f'(x_0)dx + \frac{1}{2}f''(x_0)(x - x_0)^2,
   \]
\(^1\)This means that \( f''(x_0) \) exists.
which is equivalent to the local quadratic approximation.

Where does the quadratic term come from, in the local quadratic approximation? The simplest excuse for that term comes from thinking about how you might improve on the local linear approximation. You see, the local linear approximation takes the value \( f(x_0) \) and includes information about the “slope,” or instantaneous rate of change, of \( f \) near \( x_0 \). This gives a better approximation than just saying that \( f(x) \) is near \( f(x_0) \) when \( x \) is near \( x_0 \). The quadratic term contains information about the concavity of the function near \( x_0 \), so adding it to the local linear approximation adds yet more information about \( f \), and therefore makes for a better approximation. You see, if the function is concave up, then a parabola that opens up stands a chance of approximating the function better than a line would, in all its straightness. Likewise, if the function is concave down.

1.2 Level curves

(To be written. The gist is that the local linear approximation to a level curve is the same as for any other curve, but feels different because you don’t necessarily have an explicitly defined function you can differentiate to get the slope.)

1.3 Vector-valued functions

Similar results hold for vector-valued functions.

**Definitions 1.2.** Let \( \vec{r} = \vec{r}(t) \) be differentiable at \( t = t_0 \).

1. The increment of \( t \) is \( \Delta t = t - t_0 \).
2. The increment of \( \vec{r} \) is \( \Delta \vec{r} = \vec{r}(t) - \vec{r}(t_0) \).
3. The differential of \( t \) is \( dt = \Delta t \).
4. The differential of \( \vec{r} \) is \( d\vec{r} = f'(x_0)dt \).
5. The local linear approximation to \( \vec{r} = \vec{r}(t) \) at \( t = t_0 \) is
   \[ \vec{r}(t) \approx \vec{r}(t_0) + \vec{r}'(t_0)(t - t_0), \]
   for \( t \) sufficiently close to \( t_0 \).
6. The local quadratic approximation to \( y = f(x) \) at \( x = x_0 \) is
   \[ \vec{r}(t) \approx \vec{r}(t_0) + \vec{r}'(t_0)(t - t_0) + \frac{1}{2} \vec{r}''(t_0)(t - t_0)^2, \]
   for \( t \) sufficiently close to \( t_0 \).

**Exercise 1.3.1.** Calculate the local linear and quadratic approximations at \( t = 0 \) to the semicubic parabola \( \vec{r}(t) = [t \ t^2 \ t^3]^T \). (Answer: \( \vec{r}(t) \approx [t \ 0 \ 0]^T \) and \( \vec{r}(t) \approx [t \ t^2 \ 0]^T \), respectively. Fun, but not terribly surprising, when you think about it.)

1.4 Miscellany

Note that there are at least two more ways to write the local linear and quadratic approximations of real- or vector-valued functions of one variable. We used “\( \Delta x \)” and “\( \Delta t \)” in the foregoing. We could just as easily have used “\( \Delta x \)” and “\( dt \)” or “\( dx \)” and “\( dt \)” This gives the following:
The local linear approximation to \( y = f(x) \) at \( x = x_0 \) is

\[
f(x) \approx f(x_0) + f'(x_0)(x - x_0)
\]

or

\[
f(x) \approx f(x_0) + f'(x_0)\Delta x
\]

or

\[
f(x) \approx f(x_0) + f'(x_0)dx,
\]

for \( x \) sufficiently close to \( x_0 \).

In like manner, we can write the local quadratic approximation to \( y = f(x) \) at \( x = x_0 \) as

\[
f(x) \approx f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2}f''(x_0)(x - x_0)^2
\]

for \( x \) sufficiently close to \( x_0 \).

\[
f(x) \approx f(x_0) + f'(x_0)\Delta x + \frac{1}{2}f''(x_0)(\Delta x)^2
\]

or

\[
f(x) \approx f(x_0) + f'(x_0)dx + \frac{1}{2}f''(x_0)(dx)^2,
\]

for \( x \) sufficiently close to \( x_0 \). The local linear and quadratic approximations to \( f(t) = f(x) \) at \( x = x_0 \) can be treated in the same way.

**Exercise 1.4.1.** Rewrite the formula for the local linear approximation to \( f(t) \) at \( t = t_0 \) in two new ways.

**Exercise 1.4.2.** Repeat the previous exercise for the local quadratic approximation to \( f'(t) \) at \( t = t_0 \).

What does “for \( x \) sufficiently close to \( x_0 \)” mean? It means (in the linear case) that if you subtract \( f(x_0, y_0) \) from both sides (of the local linear approximation), divide through by \( \Delta x \), and let \( \Delta x \) go to 0, you get the definition of “derivative.” How does “for \( x \) sufficiently close to \( x_0 \)” get used? That depends on your discipline. If you’re an engineer, things like \( \Delta x \) and \( \Delta y \) are tolerances within which you must operate. If you’re a scientist, they’re related to “error bars,” which measure (in some sense) how much experimental error is involved in whatever you’re doing. If you’re a statistician, they’re deviations, sums of the squares of which you very likely want to minimize. And so on. Similar comments about \( t \) being sufficiently close to \( t_0 \) or about the quadratic case could be made.

Why does the quadratic term in the local quadratic approximation have a \( 1/2 \) in it: \( \frac{1}{2}f''(x_0)dx^2 \)? That’s for consistency with differentiation. To see what I mean, assume that \( f \) is twice differentiable and think about the local quadratic approximation to \( f \). It’s a parabola (because of the \((dx)^2 = (x-x_0)^2\) that it has in it), so its derivative ought to be a line. Which line? Since this parabola locally approximates \( f \), we might think the derivative of the local quadratic approximation also approximates \( f \). We’d be wrong. It actually approximates \( f' \), which you can see by differentiating both sides of the local quadratic approximation to \( f \):

\[
f(x) \approx f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2}f''(x_0)(x - x_0)^2
\]

differentiates to

\[
f'(x) \approx 0 + \frac{d}{dx}(f'(x_0)(x - x_0)) + \frac{1}{2} \frac{d}{dx}(f''(x_0)(x - x_0)^2)
\]

\[
= f'(x_0) + \frac{1}{2}f''(x_0)(x - x_0)^1,
\]

so

\[
f'(x) \approx f'(x_0) + f''(x_0)(x - x_0).
\]

Note carefully that if the \( 1/2 \) weren’t in front of the \( f'' \) term, it wouldn’t have canceled the \( 2 \) that “came down” from the \((x-x_0)^2\) part during the differentiation. So we see two things: First, the derivative of a local quadratic approximation is the local linear approximation of the derivative (which is a very pretty result) and, second: There needs to be a \( 1/2 \) in front of the quadratic term in any local quadratic approximation.

Back in Section 1.1, I gave one reason for the appearance of the quadratic term in the local quadratic approximation. Well, the previous paragraph points up another reason: The quadratic term helps approximate the change in the local linear approximation. You see, if you think about how \( f' \) changes as \( x \) varies near \( x_0 \), you’ll eventually end up writing down the differential of \( f' \), which is

\[
d(f') = (f')'(x_0)dx = f''(x_0)dx
\]

(assuming \( f'' \) exists, of course). The corresponding change in \( dy \) should be

\[
d(dy) = d(f'dx),
\]

evaluated at \( x_0 \). It would be mighty convenient if this were the same thing as \( d(f'(x_0))dx \). It is.\(^2\) This means we can plow on, rewriting \( d(f'dx) \) as

\[
d(f'(x_0))dx = ((f')'(x_0)dx)dx = f''(x_0)(dx)^2.
\]

\(^2\)As opposed to being something like \( d(f'(x_0))dx \) or \( f'(x_0)d(dx) \). Turns out that \( d(dx) = 0 \). If you want to know why, come ask me some time.
Attaching the factor of 1/2 we’ve already discussed gives the quadratic term

\[ \frac{1}{2} f''(x_0)(dx)^2, \]

as advertised.

Finally, think about the increment theorem. We now have LOTS of ways to write it. For example:

“If \( y = f(x) \) is differentiable at \( x = x_0 \) then \( dy \approx f'(x_0)\Delta x \).”

Exercise 1.4.3. Think of at least two more ways to write the increment theorem.

\section{Local approximations to surfaces}

If we want local approximations to surfaces, we should be able to get a fair amount of good out of thinking about local approximations to curves.

\subsection{Local linear approximations to surfaces}

Based on the foregoing, we can expect the local linear approximation to a surface to have first derivatives in it. Which ones? There \textit{are} infinitely many, after all. Well, \( \mathcal{L}(t) \) depends on \( t \), and we use \( d\mathcal{L}/dt \) in the local linear approximation of \( \mathcal{L}(t) \). And \( f(x) \) depends \( x \), and we use \( df/dx \) in the local linear approximation of \( f(x) \). Since \( f(x,y) \) depends on \( x \) and \( y \), let’s use \( \partial f/\partial x \) and \( \partial f/\partial y \) in the local linear approximation of \( f(x,y) \).

Fine. Now, the local linearizations in the previous section have things like \( \Delta z \) for \( (x,y) \) sufficiently close to \( (x_0,y_0) \). It is.

Now, for a function \( z = f(x,y) \), there are \textit{two} kinds of input values we can change: \( x \)-values and \( y \)-values. So the total change in output \( (dz) \) can be viewed as the-change-in-output-due-to-a-change-in-the-x-values PLUS the-change-in-output-due-to-a-change-in-the-y-values. So we can approximate the change in output \( (dz) \) by totaling (adding) the changes in output due to \( x \) \( (f_x dx) \) and those due to \( y \) \( (f_y dy) \). Well, that would mean

\[ dz = f_x(x_0,y_0)dx + f_y(x_0,y_0)dy. \]

This is the \textbf{total differential} of \( f(x,y) \). It is \textbf{not} the local linear approximation. You see, \( dz \) estimates only the change in \( z \) as you go from \( (x_0,y_0) \) to \( (x,y) \). It doesn’t include the change from \textit{what}. Saying “\( dz = 1 \)” is a little like saying, “You’ll go up one flight of stairs.” Well, if you don’t know which floor of the building you started on, you don’t know which one you’ll end up on, do you? So the formula for the local linearization needs to include the \( z \)-value we’re starting at, as well as \( dz \). That way, we know which floor we’re on at first and how many flights of stairs to go up or down, which should give us an idea of where we’ll end up.

Enough metaphors. Here’s the official version: If \( f(x,y) \) is differentiable at \( (x_0,y_0) \), the local linear approximation to \( f \) at \( (x_0,y_0) \) ought to be

\[ f(x,y) \approx f(x_0,y_0) + f_x(x_0,y_0)dx + f_y(x_0,y_0)dy, \]

for \( (x,y) \) sufficiently close to \( (x_0,y_0) \). It is.

\textbf{Example}. Try it: Write down the local linear approximation of \( z = -xy + \cos x \) at \((2,\pi/3)\).

Hmm… I suppose we ought to assemble all the necessary vocabulary for the local linear approximation. Try this:

\textbf{Definitions 2.1}. Let \( z = f(x,y) \) be differentiable at \( (x_0,y_0) \).

1. The \textit{increments} of \( x \) and \( y \) are \( \Delta x = x - x_0 \) and \( \Delta y = y - y_0 \), respectively.
2. The increment of \( z \) is \( \Delta z = f(x, y) - f(x_0, y_0) \).
3. The differentials of \( x \) and \( y \) are \( dx = \Delta x \) and \( dy = \Delta y \), respectively.
4. The total differential of \( z \) is \( dz = f_x(x_0, y_0)dx + f_y(x_0, y_0)dy \). Note that \( dz \) is sometimes called \( df \).
5. The local linear approximation to \( z = f(x, y) \) at \((x_0, y_0)\) is

   \[
   f(x, y) \approx f(x_0, y_0) + f_x(x_0, y_0)dx + f_y(x_0, y_0)dy,
   \]

   for \((x, y)\) near \((x_0, y_0)\).

   Note that the local linear approximation can also be thought of as \( f(x, y) \approx f(x_0, y_0) + df \), which is what the increment theorem for \( f(x, y) \) says. We state this theorem as:

**Theorem 2.1.1.** If \( f(x, y) \) is differentiable at \((x_0, y_0)\) then \( \Delta f \approx df \), or

   \[
   f(x_0, y_0) \approx f(x, y) - f(x_0, y_0) \approx f_x(x_0, y_0)dx + f_y(x_0, y_0)dy,
   \]

   for \((x, y)\) sufficiently close to \((x_0, y_0)\).

**Exercise 2.1.2.** Write down the local linear approximation to \( z = xy^2 - 3xy + 4x \) at \((1, 2)\).

Later on, we may want the convenience of writing the linear terms (the “\( f_x(x_0, y_0)dx + f_y(x_0, y_0)dy \)” part) as a dot product. To this end, make a column matrix (that is, a vector) out of \( dx \) and \( dy \) and call it \( d(x, y) \). That is, \( d(x, y) = \begin{bmatrix} dx \\ dy \end{bmatrix} \). Then

\[
\begin{align*}
  f_x(x_0, y_0)dx + f_y(x_0, y_0)dy &= \nabla f(x_0, y_0) \cdot d(x, y).
\end{align*}
\]

This turns the local linear approximation into

\[
  f(x, y) \approx f(x_0, y_0) + \nabla f \cdot d(x, y),
\]

for \((x, y)\) near \((x_0, y_0)\).

That “\( \nabla f \cdot d(x, y) \)” sure reminds me of a chain rule, but it’s not. Chain rules look like \( \nabla f \frac{\partial (x, y)}{\partial t} \), or something. The Jacobian matrix in the chain rule is a full-fledged derivative, not some differential.

### 2.2 Local quadratic approximations to surfaces

The local quadratic approximation is missing from the above list, because it needs additional attention, being more complicated than any local approximation we’ve seen, so far. We expect from our previous experience that the local quadratic approximation to a surface ought to have a 1/2 in front of its quadratic term. And it ought to involve second derivatives of \( f(x, y) \). But which ones? Both \( f_x \) and \( f_y \) have infinitely many first derivatives. We worked around this before by just differentiating in both the \( x \)- and \( y \)-directions. This suggests differentiating both \( f_x \) and \( f_y \) with respect to both \( x \) and \( y \), which is what we’ll do.

Indeed, if we were to think about how the first derivatives of \( f \) change as \((x, y)\) varies near \((x_0, y_0)\), we’d eventually write down the differential of \( f_x \), or

\[
  d(f_x) \approx (f_x)_x(x_0, y_0)dx + (f_x)_y(x_0, y_0)dy = f_{xx}(x_0, y_0)dx + f_{xy}(x_0, y_0)dy.
\]

We’d do the same for \( f_y \), writing

\[
  d(f_y) \approx (f_y)_x(x_0, y_0)dx + (f_y)_y(x_0, y_0)dy = f_{yx}(x_0, y_0)dx + f_{yy}(x_0, y_0)dy.
\]

So what?
So think about a change in $dz$. That should be

$$d(dz) = d(f_z(x_0, y_0)dx + f_y(x_0, y_0)dy) = d(f_z(x_0, y_0)dx) + d(f_y(x_0, y_0)dy).$$

It would sure be convenient if $d(f_z(x_0, y_0)dx)$ were the same thing as $d(f_z)(x_0, y_0)dx$. It is.\(^3\) This means we can keep forging ahead, to get that $d(dz)$ ought to be

$$d(f_z)(x_0, y_0)dx + d(f_x)(x_0, y_0)dx = (f_{xx}(x_0, y_0)dx + f_{xy}(x_0, y_0)dy)dx + (f_{yx}(x_0, y_0)dx + f_{yy}(x_0, y_0)dy)dy$$

$$= f_{xx}(x_0, y_0)(dx)^2 + f_{xy}(x_0, y_0)dydx + f_{yx}(x_0, y_0)dxdy + f_{yy}(x_0, y_0)(dy)^2.$$

If the second partial derivatives of $f$ are continuous at $(x_0, y_0)$, then $f_{xy}(x_0, y_0) = f_{yx}(x_0, y_0)$. This means we can rewrite the foregoing as

$$f_{xx}(x_0, y_0)(dx)^2 + 2f_{xy}(x_0, y_0)dxdy + f_{yy}(x_0, y_0)(dy)^2.$$

Multiplying this by 1/2 will give us the quadratic term in the local quadratic approximation to $z = f(x, y)$.

Well, we already know what the local linear approximation to $z = f(x, y)$ is. Let’s tack on the quadratic term and see what we get:

**Definition 2.2.1.** If $z = f(x, y)$ is twice differentiable at $(x_0, y_0)$ then the local quadratic approximation to $f$ at $(x_0, y_0)$ is

$$f(x, y) \approx f(x_0, y_0) + f_x(x_0, y_0)dx + f_y(x_0, y_0)dy + \frac{1}{2} f_{xx}(x_0, y_0)(dx)^2 + f_{xy}(x_0, y_0)dxdy + \frac{1}{2} f_{yy}(x_0, y_0)(dy)^2,$$

for $(x, y)$ near $(x_0, y_0)$.

**Exercise 2.2.2.** Write down the local quadratic approximation to $z = xy^2 - 3xy + 4x$ at $(1, 2)$.

### 2.3 Isn’t there a better way to write this stuff?

If you want a better way to write this stuff, think about matrix multiplication. You know, like when we were discussing chain rules. For example, if $f(x, y) = 3x^2y - 2\cos(x - y)$, we’d say something like “Let $u = x^2y$ and $v = x - y$, so that $f = 3u - 2\cos v$.” Then we’d calculate

$$\vec{\nabla} f \bigg|_{u=x^2y,v=x-y} = \begin{bmatrix} 3 & 2\sin v \end{bmatrix} \bigg|_{u=x^2y,v=x-y} = \begin{bmatrix} 3 & 2\sin(x-y) \end{bmatrix}$$

and

$$\frac{\partial(u, v)}{\partial(x, y)} = \begin{bmatrix} 2xy & x^2 \\ 1 & -1 \end{bmatrix}$$

and then

$$\frac{\partial f}{\partial(x, y)} = \left( \vec{\nabla} f \bigg|_{u=x^2y,v=x-y} \right) \frac{\partial(u, v)}{\partial(x, y)} = \begin{bmatrix} 3 & 2\sin(x-y) \end{bmatrix} \begin{bmatrix} 2xy & x^2 \\ 1 & -1 \end{bmatrix}, \ldots$$

(Sigh. Good times.) But here’s the magic moment: The multiplication of $\begin{bmatrix} 3 & 2\sin(x-y) \end{bmatrix}$ by $\begin{bmatrix} 2xy & x^2 \\ 1 & -1 \end{bmatrix}$ consists of taking the dot product of the row matrix (that is, of $\vec{\nabla} f$) with each column of the Jacobian matrix. We’d get

$$\begin{bmatrix} 6xy + 2\sin(x-y) & 3x^2 - 2\sin(x-y) \end{bmatrix}.$$ 

Point is, to multiply two matrices together, take the dot product of each row of the left-hand matrix with each column of the right-hand matrix.

---

\(^3\)As opposed to being something like $d(f_z)(x_0, y_0)dx + f_z(x_0, y_0)d(dx)$. Actually, it turns out that $d(dx) = 0$. If you want to know why, read the section on the technical stuff, at the end of this document.
Example 2.3.1. \[
\begin{bmatrix}
1 & 2 & 3 \\
4 & 5 & 6
\end{bmatrix}
\begin{bmatrix}
a & r & x \\
b & s & y \\
c & t & z
\end{bmatrix}
= \begin{bmatrix}
a + 2b + 3c & r + 2s + 3t & x + 2y + 3z \\
a + 5b + 6c & 4r + 5s + 6t & 4x + 5y + 6z
\end{bmatrix}
\]

Example 2.3.2. \[
\begin{bmatrix}
-2 & 1 \\
3 & 0 \\
5 & -7
\end{bmatrix}
\begin{bmatrix}
3 & -4 & 1 \\
0 & 12 & -1
\end{bmatrix}
= \begin{bmatrix}
(-2)(3) + (1)(0) & (-2)(-4) + (1)(12) & (-2)(1) + (1)(-1) \\
(3)(3) + (0)(0) & (3)(-4) + (0)(12) & (3)(1) + (0)(-1) \\
(5)(3) + (-7)(0) & (5)(-4) + (-7)(12) & (5)(1) + (-7)(-1)
\end{bmatrix}
= \begin{bmatrix}
-15 & 4 & 9 \\
-6 & -5 & 20 \\
9 & -12 & 3
\end{bmatrix}
\]

You can multiply more than two matrices together. Just start at the left and work your way right:

Example 2.3.3. \[
\begin{bmatrix}
a & b \\
1 & 3 \\
2 & 4
\end{bmatrix}
\begin{bmatrix}
a + 3b & 2a + 4b
\end{bmatrix}
= \begin{bmatrix}
(a + 3b)a + (2a + 4b)b
\end{bmatrix}
= a^2 + 5ab + 4b^2
\]

Here’s how matrix multiplication helps. Suppose \( f = f(x, y) \) has continuous second partial derivatives. Define the Hessian matrix of \( f \) to be

\[
H_f = \begin{bmatrix}
f_{xx} & f_{xy} \\
f_{yx} & f_{yy}
\end{bmatrix}.
\]

(It’s the matrix of second partial derivatives of \( f \), just as the gradient is the vector of first partial derivatives of \( f \).) Recall that \( d(x, y) = \begin{bmatrix} dx \\ dy \end{bmatrix} \). Transpose this column to make a row out of it,\(^4\) and call the result \( d(x, y)^T \); that is, \( d(x, y)^T = \begin{bmatrix} dx \\ dy \end{bmatrix} \).

The quadratic term in the local quadratic approximation to \( f \) can now be written as

\[
\frac{1}{2} d(x, y)^T H_f d(x, y),
\]

when this matrix product is evaluated at \((x_0, y_0)\). (Note the 1/2, inherited from local quadratic approximations to curves.) Here's the calculation that proves that the above really is the quadratic term:

\[
\frac{1}{2} d(x, y)^T H_f d(x, y) = \frac{1}{2} \begin{bmatrix} dx \\ dy \end{bmatrix} \begin{bmatrix}
f_{xx} & f_{xy} \\
f_{yx} & f_{yy}
\end{bmatrix} \begin{bmatrix} dx \\ dy \end{bmatrix}
= \frac{1}{2} \begin{bmatrix} f_{xx}dx + f_{yx}dy & f_{xy}dx + f_{yy}dy
\end{bmatrix} \begin{bmatrix} dx \\ dy \end{bmatrix}
= \frac{1}{2} \left( (f_{xx}dx + f_{yx}dy)dx + (f_{xy}dx + f_{yy}dy)dy \right)
= \frac{1}{2} \left( f_{xx}(dx)^2 + 2f_{xy}dxdy + f_{yy}(dy)^2 \right).
\]

When evaluated at \((x_0, y_0)\), this becomes

\[
\frac{1}{2} f_{xx}(x_0, y_0)(dx)^2 + f_{xy}(x_0, y_0)dxdy + \frac{1}{2} f_{yy}(x_0, y_0)(dy)^2,
\]
as advertised.

So, the compact way of writing the local quadratic approximation to \( f \) at \((x_0, y_0)\) is

\[
f(x, y) \approx f(x_0, y_0) + \nabla f(x_0, y_0) \cdot d(x, y) + \frac{1}{2} d(x, y)^T H_f(x_0, y_0) d(x, y),
\]

\(^4\)Transposing a column turns it into a row; transposing a row gives you a column. Or, if your matrix has multiple rows or columns, then transposing it turns all the rows into columns and all the columns into rows: \[
\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}^T = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}.
\]
for \((x, y)\) near \((x_0, y_0)\). (Some people like to write it as
\[
f(x, y) \approx f(x_0, y_0) + \nabla f(x_0, y_0) \cdot d(x, y) + \frac{1}{2} H f(x_0, y_0) \cdot d(x, y)^2,
\]
maybe to make it look a bit more like the one-variable version. But I think this is cheating just a little.\(^5\) In its full glory, the local quadratic approximation is
\[
f(x, y) \approx f(x_0, y_0) + f_x(x_0, y_0) dx + f_y(x_0, y_0) dy + \frac{1}{2} f_{xx}(x_0, y_0)(dx)^2 + f_{xy}(x_0, y_0)dxdy + \frac{1}{2} f_{yy}(x_0, y_0)(dy)^2
\]
\[= f(x_0, y_0) + f_x(x_0, y_0)(x - x_0) + f_y(x_0, y_0)(y - y_0)
\]
\[+ \frac{1}{2} f_{xx}(x_0, y_0)(x - x_0)^2 + f_{xy}(x_0, y_0)(x - x_0)(y - y_0) + \frac{1}{2} f_{yy}(x_0, y_0)(y - y_0)^2,
\]
for \((x, y)\) near \((x_0, y_0)\). Knowing how lazy I am, I bet you can guess how I prefer writing it!

3 Local approximations to level surfaces

(To be written. The gist is that the nature of the local approximations stays the same, but you have to have a different recipe for writing them down, because you don’t necessarily have an explicitly defined function you can differentiate.)

4 Local approximations to parameterized surfaces

(To be written. Again, the upshot is that the fact that the surface is parameterized makes you need a slightly different recipe for writing down the local approximations.)

5 The technical stuff

Sorry, but I have to finish the rest of this document before I write this section.

\(^5\)A more elegant way to write the local quadratic approximation comes from realizing that the gradient is the first derivative of \(f\) and the Hessian is the second derivative. Then you can write \(f(x, y) \approx f(x_0, y_0) + Df(x_0, y_0)d(x, y) + \frac{1}{2} d(x, y)^T D^2 f(x_0, y_0) d(x, y)\), or in its “cheating version,” \(f(x, y) \approx f(x_0, y_0) + Df(x_0, y_0)d(x, y) + \frac{1}{2} D^2 f(x_0, y_0) d(x, y)^2\). Stripped down a bit more, to suppress the dependence on \((x, y)\) and \((x_0, y_0)\), we have \(z \approx z_0 + Dz d(x, y) + \frac{1}{2} D^2 z d(x, y)^2\). There are other ways to write it, yet, but this is enough for one document.