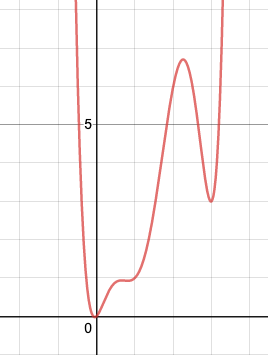
Single Variable Calculus Review:

1. Calculate the following derivatives:
2.  b.  c. 

d.  e.  f.  g. 

1. Find the minimum of using the first derivative test: .
2. Question: How would you find the absolute minimum of a function that had several relative minima?

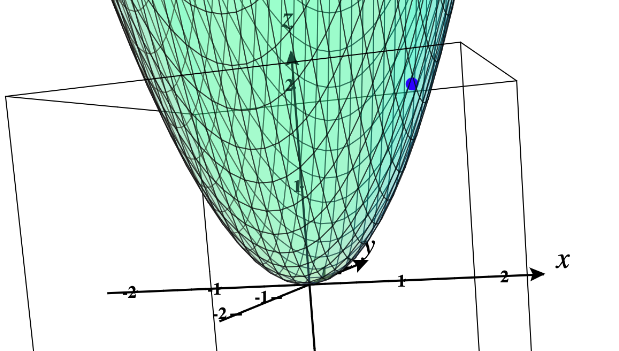


Notice that in #1f and #g, using the constant multiple rule, the constants just hung around and you multiplied by the derivative of the remaining x terms. We’ll do something similar with partial derivatives.

**Partial Derivatives:**

Partial derivatives are computed by taking the derivative with respect to one variable while you treat the other variables as constants. For the function given by , the partial derivative with respect to x is denoted by  and the partial derivative with respect to y is denoted by .

For example, consider the function given by . The partial derivatives are given by  and . In particular, if you wanted to evaluate the partial derivatives at the point (1,1), denoted with the blue dot below, you would get  and .



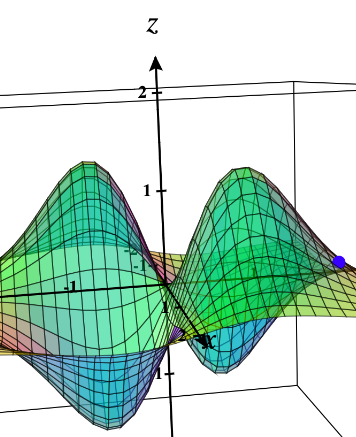
1. Evaluate the partial derivatives of the following functions:
2.  b. 
3.  d. 

Definition: Let z=f(x,y) be a function such that its partial derivatives exist. Then the **gradient** of f, denoted by , is the vector .

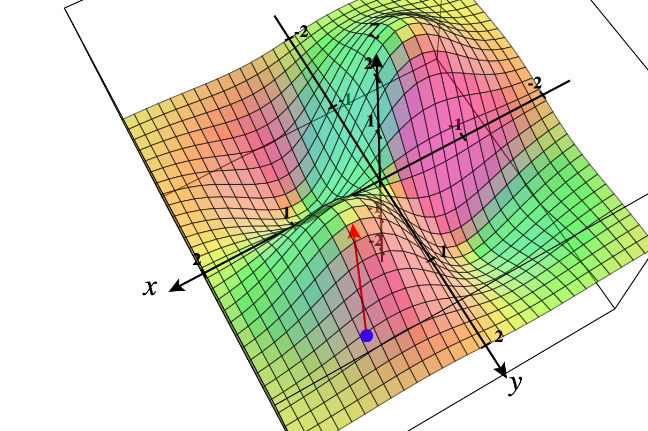
1. Compute the vector of the functions in #4a and #4b.

**Theorem:** Let f be differentiable at a point (x,y). Then the direction of maximum/minimum increase of f is given by  and , respectively.

Let’s look at a more complicated function of the form :



If we tilt our rotation, we see that at the point (1,1.5), the gradient is pointing in the steepest direction of ascent and the negative gradient is point towards the steepest descent:

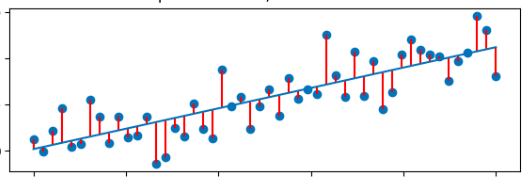


You can play around with more graphs here:

<https://www.monroecc.edu/faculty/paulseeburger/calcnsf/CalcPlot3D/>

**Linear Regression**

Going back to linear regression, recall that we’d like to minimize the sum of the squares of the residuals, where residuals are given by the difference between the actual and predicted y-values.



If our linear regression equation is given by , then we are trying to choose  and  so that we minimize the sum of the squares of the differences , where we have *m* points  in our training dataset.

Meaning, we would like to minimize our cost function J. Note that we see an m in the denominator to denote the average and we insert an extra 2 to make some derivatives prettier later on:



If we calculate our partial derivatives, we obtain:

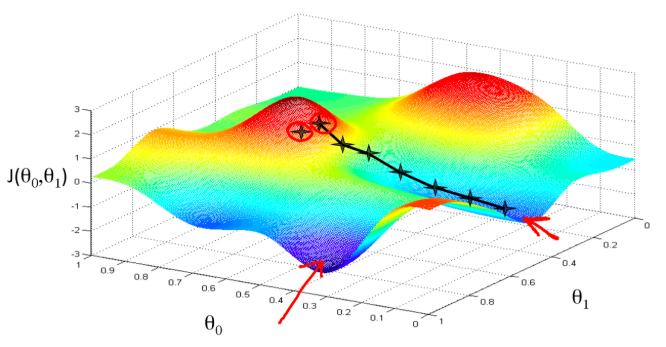
 and



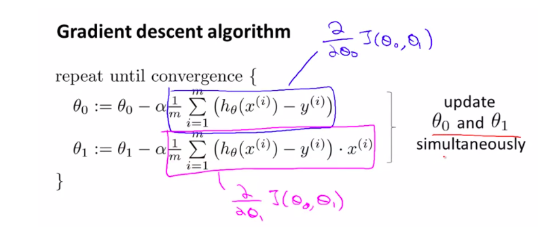
By our theorem above, if at any point  we head in the direction of steepest descent given by

, then we will wind up at a minimum, meaning, we will have chosen the ideal linear regression equation  that minimizes our error.

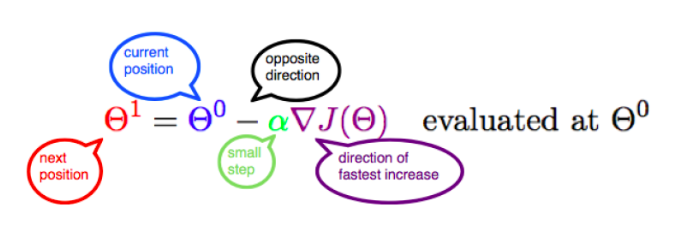
How can you be sure that you end up at the absolute min rather than a local min? You’ll want to randomly start at several different points:



Finally, the gradient descent algorithm for numerically obtaining the ideal parameters is given below. Basically, we will continue taking small steps in the direction of descent until we can’t get much lower and we converge to a minimum value (within a given tolerance):

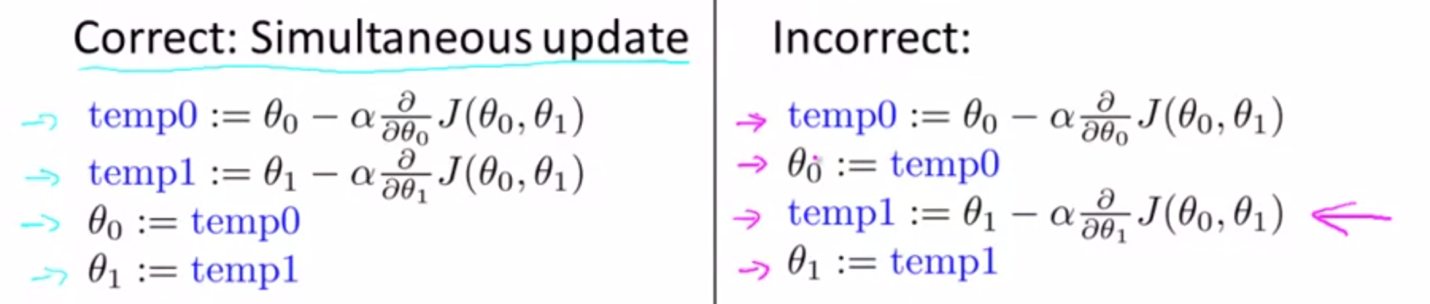


Where:



Note that alpha is the learning rate (which we have control.over) that lets us determine how big of a step size we would like to take as we descend.

Beware about updating our variables correctly:



Or in Python, you can simply use commas to update things simultaneously:



So many videos!

Visualizing Gradient

<https://www.youtube.com/watch?v=Ys_6XZQa-3g> (6:44)

Cost Functions

<https://www.coursera.org/learn/machine-learning/lecture/db3jS/model-representation>

<https://www.coursera.org/learn/machine-learning/lecture/rkTp3/cost-function> (8:12)

<https://www.coursera.org/learn/machine-learning/lecture/N09c6/cost-function-intuition-i> (11:09)

<https://www.coursera.org/learn/machine-learning/lecture/nwpe2/cost-function-intuition-ii> (8:48)

Gradient Descent

<https://www.coursera.org/learn/machine-learning/lecture/8SpIM/gradient-descent> (iteration details 11:30)

<https://www.coursera.org/learn/machine-learning/lecture/GFFPB/gradient-descent-intuition> (11:50)

<https://www.coursera.org/learn/machine-learning/lecture/kCvQc/gradient-descent-for-linear-regression>