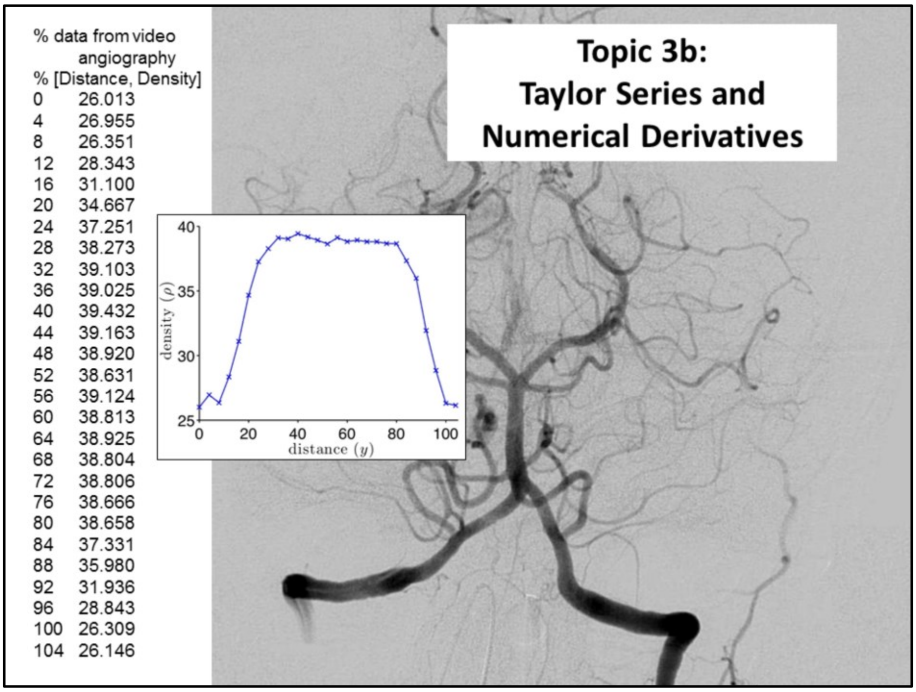
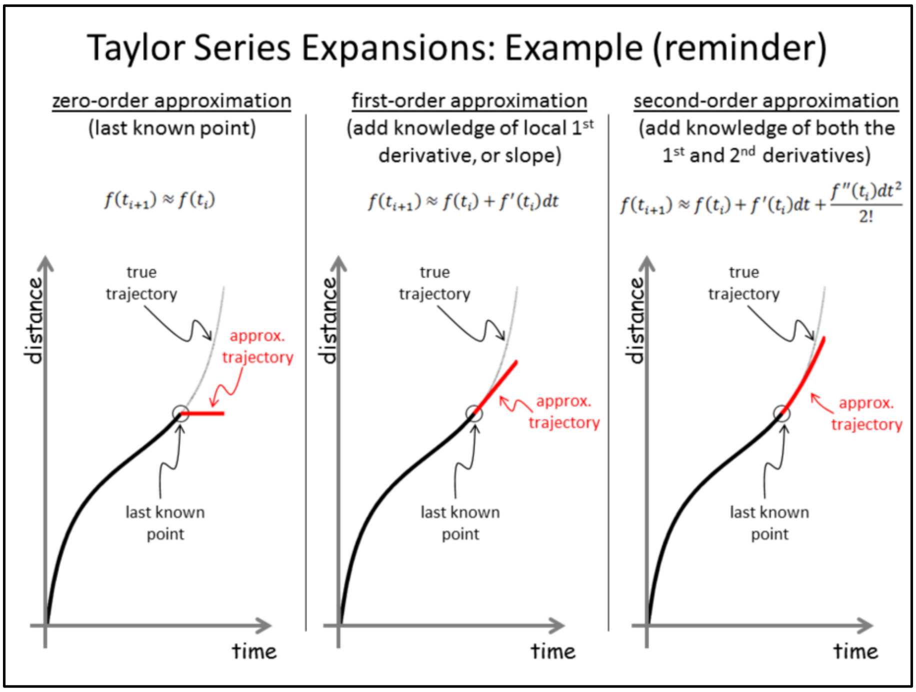


In the previous topic, we introduced the idea of Taylor series. In this discussion we extend our ideas about Taylor series and use them to define/derive numerical derivatives. In doing so we will discuss several ways to compute numerical derivatives (forward, centered, backward, and with different orders of accuracy), and we will discuss these here.



Just a reminder of what Taylor series expansions look – this was the second slide in the last topic also.

Taylor Series and Truncation Error

Nth order Taylor Series Expansion with a remainder term:

$$f(t_{i+1}) = f(t_i) + f'(t_i)dt + \frac{f''(t_i)}{2!}dt^2 + \frac{f^{(3)}(t_i)}{3!}dt^3 + \cdots + \frac{f^{(n)}(t_i)}{n!}dt^n + R_n$$

Truncated 1st Order Taylor Series:

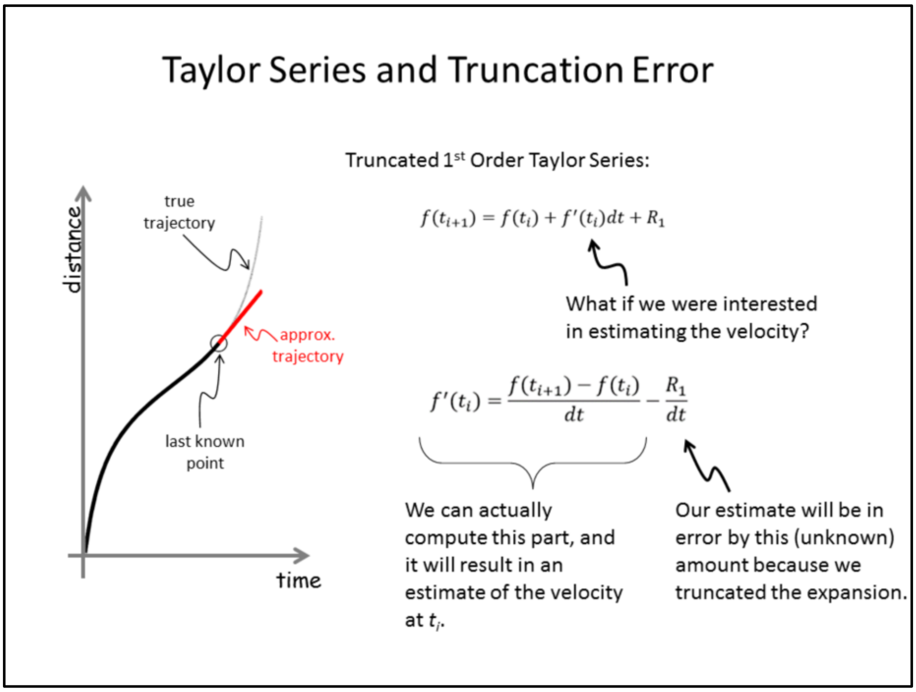
$$f(t_{i+1}) = f(t_i) + f'(t_i)dt + R_1$$

Recall that we can use the 'equal' sign because we've included the remainder time, which we admittedly don't know.

$$R_n = \frac{f^{(n+1)}(\xi)}{(n+1)!}dt^{n+1} \qquad R_1 = \frac{f^{(2)}(\xi)}{2!}dt^2$$

Again as a reminder, here we show an nth order Taylor series expansion that includes a remainder term. We also show a truncated 1st order Taylor Series, including the remainder term.

Including the remainder term allows us to use an 'equal' sign rather than an 'approximately equal' sign. This doesn't mean we actually know what R1 is, so this is a bit of trickery – don't forget that if we were actually evaluating the first order Taylor series, we are still making an approximation.



What if we were interested in evaluating the first derivative rather than the function? A simple re-arranging of our 1st order Taylor series allows us to do this. Note that if we carry the remainder term (the truncation error), we can keep the 'equal' sign.

To evaluate the numerical derivative (i.e., the first derivative of *f* with respect to *t*), we simply subtract the value of *f* at t_i from the value of t_{i+1} and divide the result by *dt*.

It is important to note the difference from what we've done before. We are no longer making a prediction of the function *f* at t_{i+1} based on our knowledge of the function and its derivatives at t_i . Instead, we are now assuming that we actually know the value of *f* at both t_{i+1} and t_i , and are using the knowledge to estimate the first derivative of *f*. In other words, we've turned the problem around.

It is also important to note in the formula given above for *f'* we have included the remainder term. Numerically, we can't actually calculate this because in practice we don't know what R_1 is. We are keeping it around for now, however, because it will help tell us how accurate our approximation for *f'* is.

Taylor Series and Truncation Error

Truncated 1st Order Taylor Series:

$$f(t_{i+1}) = f(t_i) + f'(t_i)dt + R_1$$

What if we were interested in estimating the velocity?

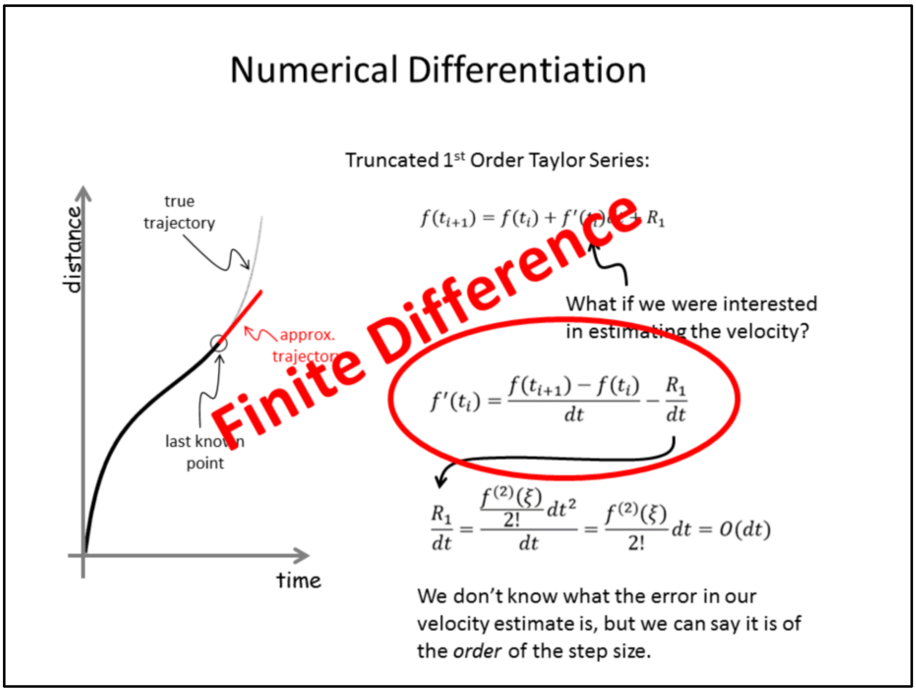
$$f'(t_i) = \frac{f(t_{i+1}) - f(t_i)}{dt} - \frac{R_1}{dt}$$

$$\frac{R_1}{dt} = \frac{\frac{f^{(2)}(\xi)}{2!}dt^2}{dt} = \frac{f^{(2)}(\xi)}{2!}dt = O(dt)$$

We don't know what the error in our velocity estimate is, but we can say it is of the *order* of the step size.

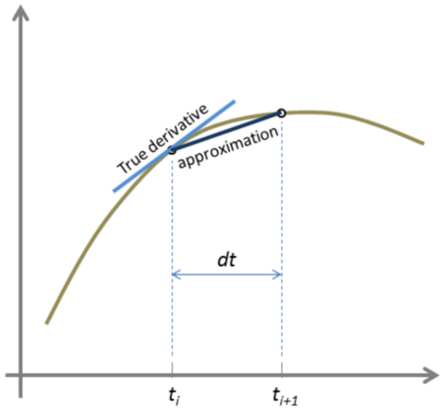
Keeping the remainder term allows us to figure out what the error in the finite difference term actually is. In this case, where we've started with a truncated 1st order Taylor series and have plugged in the formula for R_1 given on slide 3, we see that the error is proportional to the 2nd derivative of the original function (evaluated somewhere between the t_i and t_{i+1}).

It is also proportional to *dt* – so we say that the error in our velocity estimate is of order *dt*. If we cut step size in half, cwe also cut the error in the velocity estimate in half.



When we compute a derivative in this way, we formally call the operation a finite difference. In this particular case we are looking at what we would call a forward difference. We call it this because we are trying to estimate the derivative of a function at some location *t* (or *x*, or whatever variable you are using) by looking forward to the value of the function at *t+h*, where *h* is the step size.

Numerical Differentiation



$$f'(t_i) \approx \frac{f(t_{i+1}) - f(t_i)}{dt}$$

This approximation for the first derivative of *f* is called a 'forward difference'.

Derivatives are ubiquitous in science and engineering!

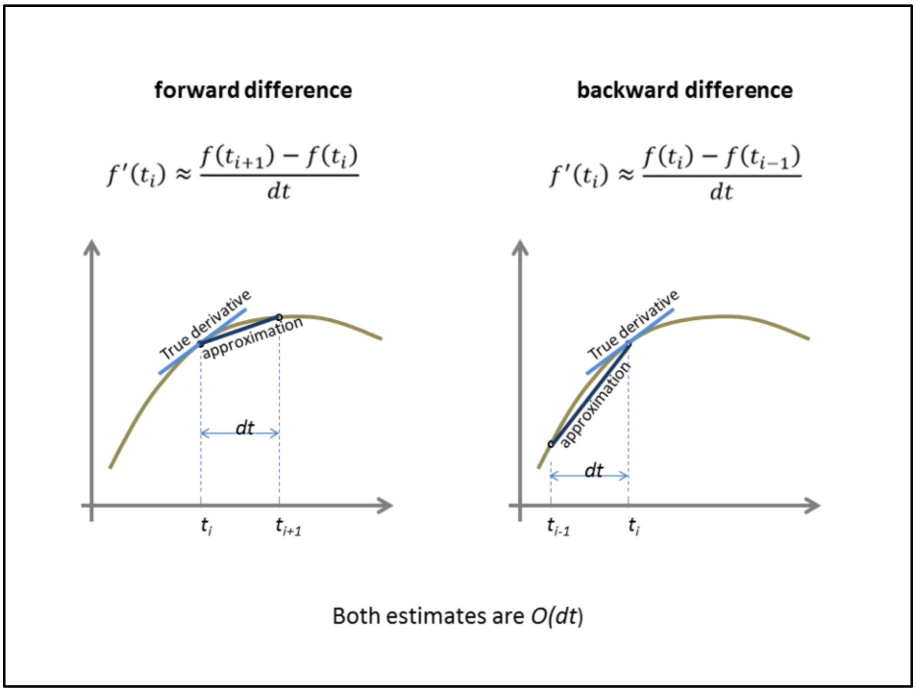
Newton's 2nd Law: $F = ma$

Newton's 2nd Law as expressed by Newton: $F = \frac{d(mv)}{dt}$

Graphically, the forward difference looks like this. We are trying to estimate the derivative of the function at $f(t_i)$, and to do so we are using knowledge of the value of the function at both t_i and t_{i+1}.

As you can see graphically, the forward difference looks biased – we are approximating the slope, but not in a particularly balanced way. As with the Taylor series approximations we used, this estimate for the first derivative becomes more and more accurate as we make *dt* smaller and smaller. (recall the formula for the error on the previous slide).

As it turns out, being able to compute a numerical derivative is extremely useful. We can use finite differences to help us evaluate Newton's second law, Fourier's law, Fick's law, Darcy's law, Ohm's law, Hooke's law, etc.



Note that if we can compute a forward difference, then we can quite likely calculate a backward difference. We can see this mathematically if we replace t_{i+1} with t_i and t_i with t_{i-1} in the equation for the finite difference – we're simply changing our perspective and looking backward instead of forward. Note that the backward difference also appears biased, but instead of biased toward later derivatives as is the case in the forward difference, the backward difference is biased toward earlier derivatives.

There are a couple of noteworthy items to discuss about the difference between forward and backward differences. One is that the backward difference is, in a sense, causal, and the forward difference is not. That is, if you were developing a tracking algorithm for a rocket that depended on real-time estimates of vertical velocity that could only be calculated from estimates of its altitude, then you would be hard-pressed to compute a forward difference without a time machine: you simply won't have the data in time to make the calculation because it doesn't yet exist. You can calculate a backward difference though, but your estimate will lag the true value because of the bias (this should make you want to keep the step size small!).

Another item worth mentioning is that when you compute these numerically from, say, an array of values for *f*, the operation on the right hand side of the equation looks very similar: you are taking values and subtracting the preceding value, and then dividing by the step size. The key difference, however, is what value of time the estimates of the derivative apply to….

A centered finite-difference seems like it would be more accurate….

Subtract the backward Taylor expansion from the forward Taylor expansion:

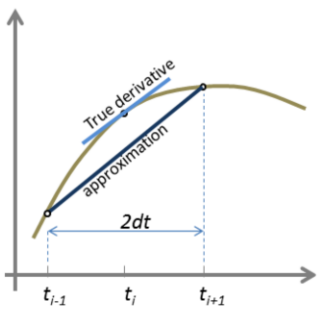
$$f(t_{i+1}) = f(t_i) + f'(t_i)dt + \frac{f''(t_i)dt^2}{2!} + \cdots$$

$$f(t_{i-1}) = f(t_i) - f'(t_i)dt + \frac{f''(t_i)dt^2}{2!} - \cdots$$

$$f(t_{i+1}) - f(t_{i-1}) = 2f'(t_i)dt + 2\frac{f^{(3)}(t_i)dt^3}{3!} + \cdots$$

… solve for $f'(t_i)$

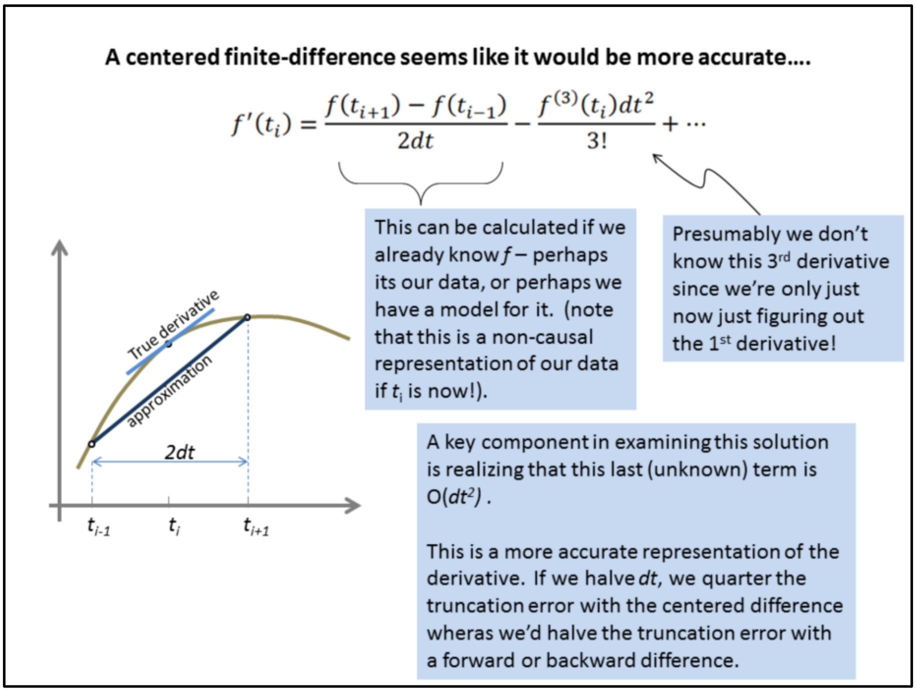
$$f'(t_i) = \frac{f(t_{i+1}) - f(t_{i-1})}{2dt} - \frac{f^{(3)}(t_i)dt^2}{3!} + \cdots$$



It seems somewhat intuitive to remove the bias in the forward and backward differences by straddling the point of interest. In other words, you could compute the finite difference at t_i by subtracting the value of the function at t_{i-1} and then dividing by *twice* the difference. This would be a good intuition on your part. What is harder to intuit, however, is what the accuracy of this approximation for the derivative is (in general, it is not equal to the true value of the derivative, although it looks quite close in the figure shown here).

We would call your intuitive method for calculating the finite difference in such a manner a 'centered difference'. We can more formally derive the centered difference (don't let your intuition take offences – it was correct, after all) by writing out a forward Taylor series (top line) and a backward Taylor series (2^{nd} line). We haven't actually formally discussed backward Taylor series, but they are quite similar to the forward Taylor series but use *-dt* instead of *+dt*. And if you re-arrange the terms in the backward Taylor series shown here, you should get the formula for the backward difference shown on the previous page.

If we then subtract the backward Taylor series from the forward Taylor series, we end up with the third equation down from the top. Note that several terms cancel when we do this, but not all of them. We simply re-arrange the terms, and solve for *f'* as shown on the bottom equation on this slide.

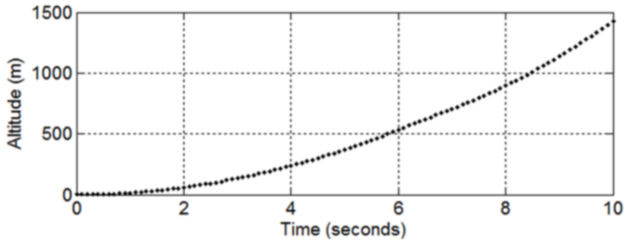


So lets look a little more closely at what our result was. When we did the difference, the even order terms (e.g., the second derivative, the fourth derivative) cancel – this happens because of the alternating signs in the backward Taylor series. The first term on the right hand side matches what we intuitively feel will be a better approximation than either the forward or backward difference. Note that this is a 'non-causal' way to do a derivative: if t_i is now, then we would have no knowledge of $f(t_{i+1})$. So, if we want to implement this in a real-time system, we have to be content with always being one time step behind.

The second term on the right hand side depends on the third derivative of the function, and is the order of dt^2. Presumably we don't know what this third derivative is (especially if we are just figuring out what the 1st derivative is!), so this represents the error in our approximation of the first derivative. Note that the error in the centered finite difference is order dt^2, but the error in the forward and backward differnces was only dt (this is a result of stopping the Taylor series after the first order term). So, for the centered finite difference, if we halve the step size dt then we quarter the truncation error.

Example

Rocket Altitude Data



Raw data

(hmmm.....what do you think about the precision of the altitude data?)

Time (sec)	Altitude (m)
0	0.002943
0.1	0.194238
0.2	0.679833
0.3	1.459728
0.4	2.533923
0.5	3.902418
0.6	5.565213
0.7	7.522308
0.8	9.773703
0.9	12.319398
1	15.159393
1.1	18.293688

At 0.2 seconds:

backward difference

$$\frac{0.679833 - 0.194238}{0.1} = 4.9 \text{ m/s}$$

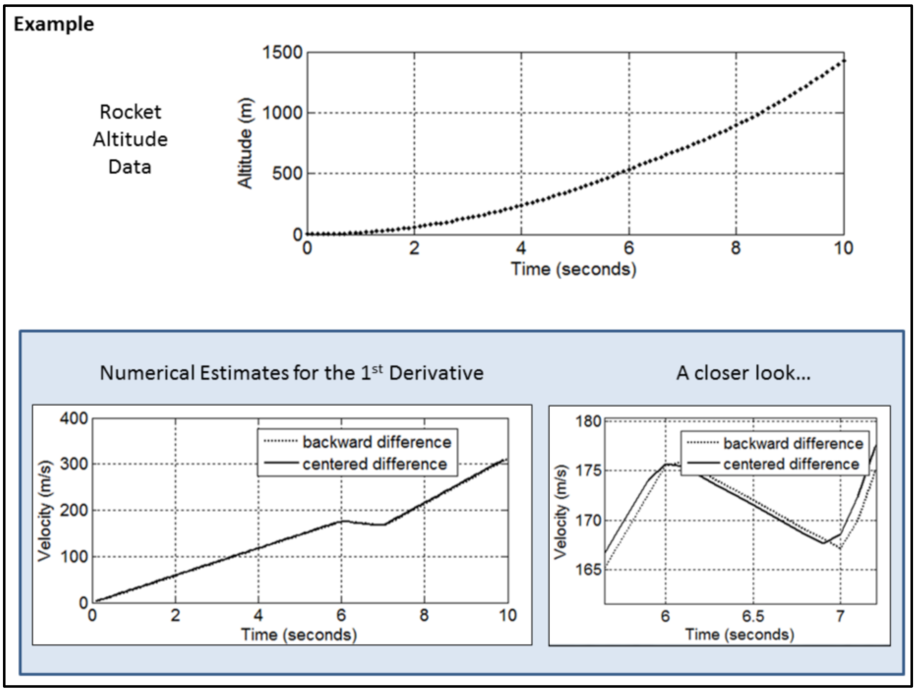
centered difference

$$\frac{1.459728 - 0.194238}{0.2} = 6.3 \text{ m/s}$$

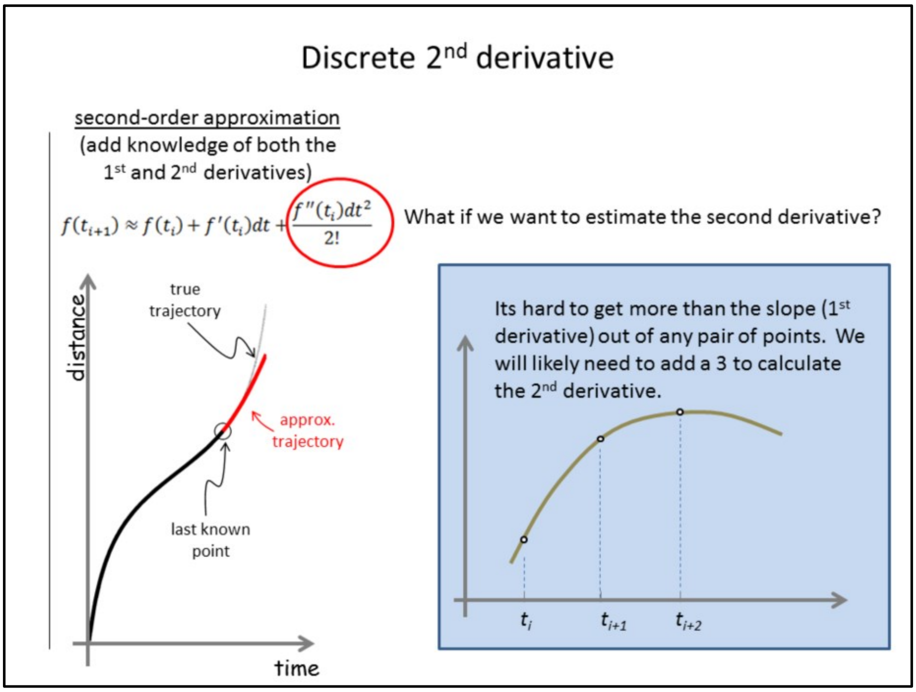
So, lets look at an example. Suppose we are measuring the altitude of a rocket as it is launched. The altitude (in meters) is shown as a function of time (in seconds) in the graph above, between 0 and 10 seconds, and in the table for times between 0 and 1.1 seconds in step sizes of 0.1 seconds. Presumably, we are *sampling* the altitude data at a rate of 10 Hz. If we examine the plot, the slope (and hence the velocity) is changing, and so the rocket must be accelerating.

If we are interested in the velocity at 0.2 seconds, we have a few ways to examine it. We could compute a backward difference, as shown here., resulting in an estimate of 4.9 m/s. The accuracy of this estimate is proportional to the step size, 0.1, and also to the second derivative of the data (the acceleration, in this case) evaluated somewhere between 0.1 and 0.2 seconds, all divided by 2 (see slides 4-6). In this case, it turns out that the acceleration (which is difficult to figure out from the problem statement, but I happen to omnisciently know) is about 30 m/s^2 over this range of times, and so the error is about -[30 m/s^2]*[0.1 s]/2 = -1.5 m/s. This is a fairly large error given that the estimated velocity is 4.9 m/s.

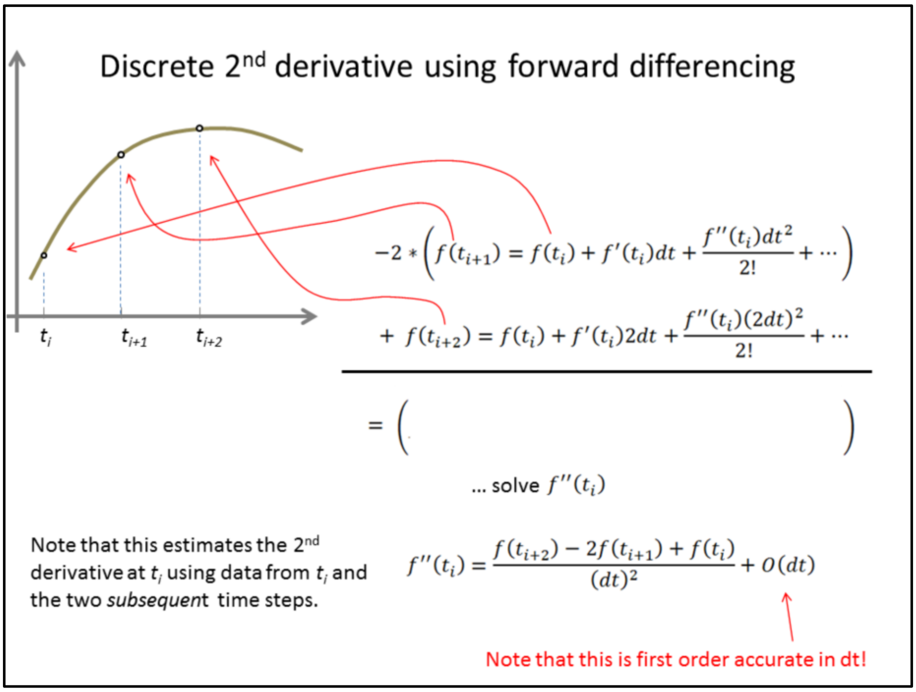
If we use a centered difference, we get a velocity estimate of 6.3 m/s, which is much more accurate. In fact, if the acceleration is constant over this time window, then the third derivative of the altitude is zero and the resulting error is zero! Note that this is not always the case, however.



Lets have a closer look at these numerical estimates for 1st derivative. At first glance, the backward and centered differences give quite close results (see the figure on the lower left). Note, however, that the vertical scale on the lower-left figure extends from 0 to 400 m/s, and we calculated an error of 'only' 1.5 m/s on the previous slide. If we zoom in to have a closer look (figure on the lower right), we see that there is a bias – the velocity estimate computed with the backward difference consistently lags the centered difference. The difference between these to estimate depends on the rate of change of velocity (the acceleration) - higher accelerations increase this bias. The reason for this bias is based in the mechanics of calculating the numerical derivitives. The centered finite difference used points that are balanced around a particular time, and the backward difference is restricted only to looking at a time history (albeit, a short time history).



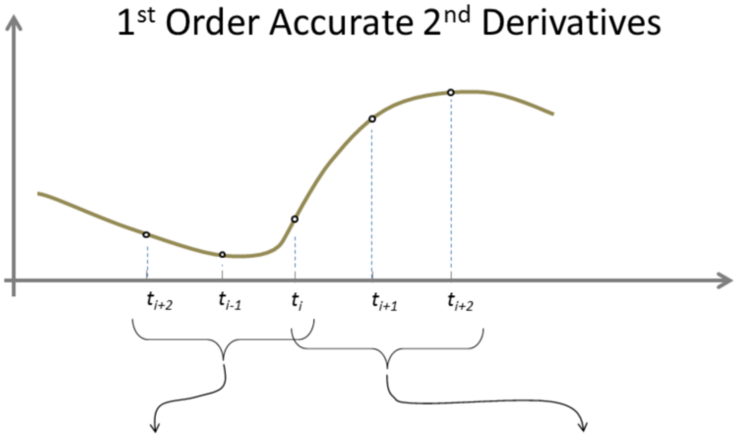
So what if we wanted to calculate a second derivative of our function – perhaps we our rocket control system requires knowledge of the acceleration as well (aside: we probably have accelerometers on our rocket, but in practice we might like to control the long-term drift of these accelerometers with a more stable and possibly less-accurate measurement). Its hard to get anything more out of two individual points than a slope (velocity), so we will likely need to use at least three pointes to calculate the 2nd derivative.



One way to get the three points we presumably need to estimate a 2nd derivative is to combine two different Taylor series. We could, for example, combine Taylor series approximations for a step size of *dt*, which uses points at $f(t_i)$ and $f(t_{i+1})$, and also for a step size of 2*dt*, which again uses a information at $f(t_i)$ but now also includes $f(t_{i+2})$. We can write these down and combine them to remove the dependence on *f'*, which seems like a useful thing to do, and then algebraically solve the result for *f''*. This results in this estimate (using forward differencing) shown above.

Note that when we do this, the result is of the order of *dt*. This might seem odd at first glance – after all, we are using Taylor series approximations that have terms up to order dt^2 with remainder terms that are of the order of dt^3. Note that we are re-arranging this combination of Taylor series approximations to solve for *f''*, which involves a step where we divide the entire equation by dt^2.

1^{st} Order Accurate 2^{nd} Derivatives



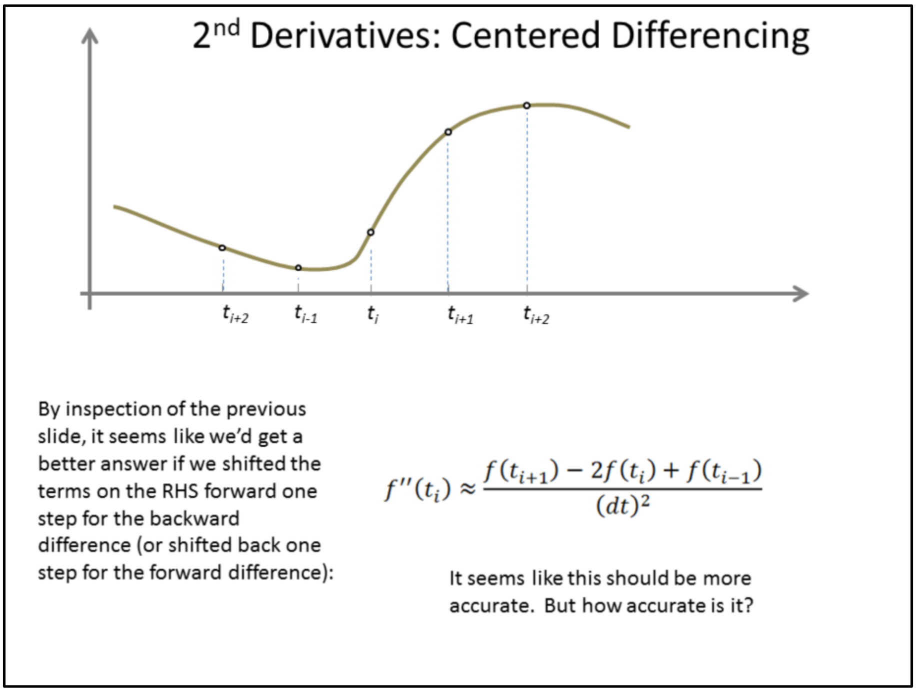
$$f''(t_i) = \frac{f(t_i) - 2f(t_{i-1}) + f(t_{i-2})}{(dt)^2} + O(dt)$$

backwards differencing

$$f''(t_i) = \frac{f(t_{i+2}) - 2f(t_{i+1}) + f(t_i)}{(dt)^2} + O(dt)$$

forward differencing

We could also develop a formula for calculating the 2^{nd} derivative using a backwards difference (which might be more realistic for, say, a real-time control system). This is something you should try doing yourself. Note that the result for both the backwards and forward differences is proportional to *dt*.



Again, by inspection it seems like we would get a better (less biased) answer by using a more balanced approach. Either we could shift the terms on the right hand side of the backward difference forward one step, or we could shift the terms on the right hand side of the forward difference backward one step. Either way, we would end up with the formula above which looks quite similar to our previous estimates. It seems like this will work better – but is it actually more accurate? And if so, how accurate is it?

Return to the Taylor series expansions:

$$f(t_{i+1}) = f(t_i) + f'(t_i)dt + \frac{f''(t_i)dt^2}{2!} + \frac{f^{(3)}(t_i)dt^3}{3!} + \frac{f^{(4)}(t_i)dt^4}{4!} + \cdots$$

$$+ \quad f(t_{i-1}) = f(t_i) - f'(t_i)dt + \frac{f''(t_i)dt^2}{2!} - \frac{f^{(3)}(t_i)dt^3}{3!} + \frac{f^{(4)}(t_i)dt^4}{4!} - \cdots$$

$$f(t_{i+1}) + f(t_{i-1}) = 2f(t_i) + 2\frac{f''(t_i)dt^2}{2!} + 2\frac{f^{(4)}(t_i)dt^4}{4!} + \cdots$$

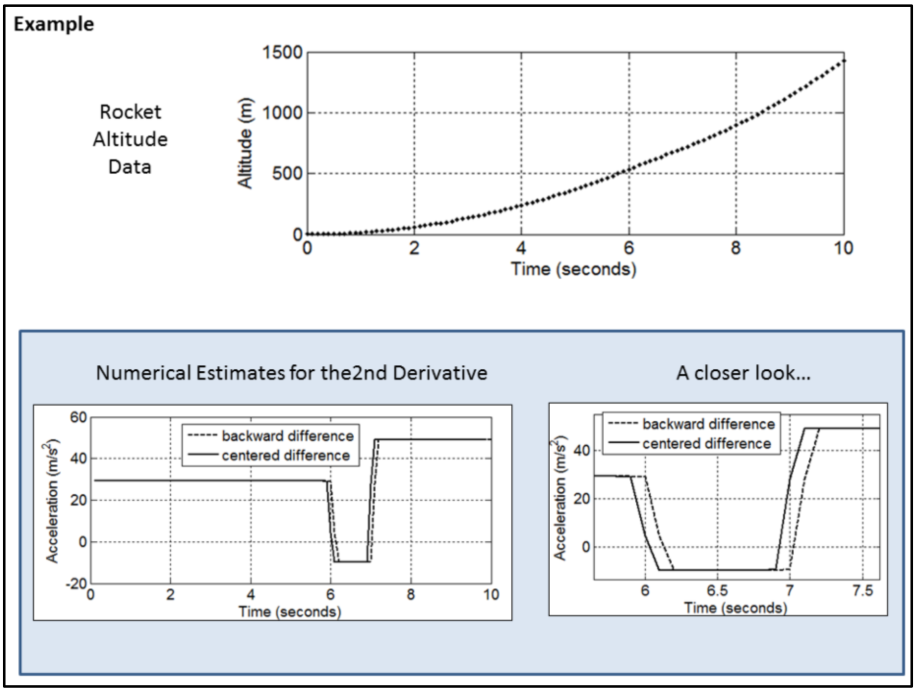
Now solve for the 2nd derivative:

$$f''(t_i) = \frac{f(t_{i+1}) - 2f(t_i) + f(t_{i-1})}{dt^2} - \frac{f^{(4)}(t_i)dt^2}{12} + \cdots$$

This term gets truncated, and is *O*(dt^2)

In similar fashion to what we did for the estimate of the first derivative using a centered finite difference, we begin by expanding the Taylor series out several terms. We do this for both a forward difference (top equation) and a backward difference (second from top). Once again, note that to compute the backward difference we use a step size of -*dt*, which results in alternating signs in the Taylor series expansion for the backward difference. If we simply sum these two series,several of the terms cancel and we get the third equation down, and we can re-arrange this equation to solve for *f''*.

After solving for *f''*, we get a term that is identical to the term we had on the previous slide, plus an additional term that depends on the fourth derivative of *f* and is proportional to dt^2. There will be other terms as well that are not shown, but these should all be smaller than the second term shown on the right hand side. Thus, we would say that this approximation for the 2nd derivative is of the order dt^2. We could also more formally define an error term – it will depend on the fourth derivative of the function evaluated between t_{i-1} and t_{i+1}.



Lets return to our example with the rocket altitude data, and now calculate the acceleration of the rocket (i.e., the second derivative). Here, we are using the formula's for the backward and centered differences, and comparing the results. As before, the backward difference is lagging the centered defference.

Interestingly, the rocket appears to be accelerating at something close to 30 m/s^2 for the first 6 seconds or so, and then is *decelerating* for a second, and then continues to accelerate again at something close to 50 m/s^2. Presumably, the rocket has dropped a booster at 6 seconds, and then continued on its way.

Note that we don’t have to stop here:

$$f(t_{i-1}) = f(t_i) - f'(t_i)dt + \frac{f''(t_i)dt^2}{2!} + O(dt^3)$$

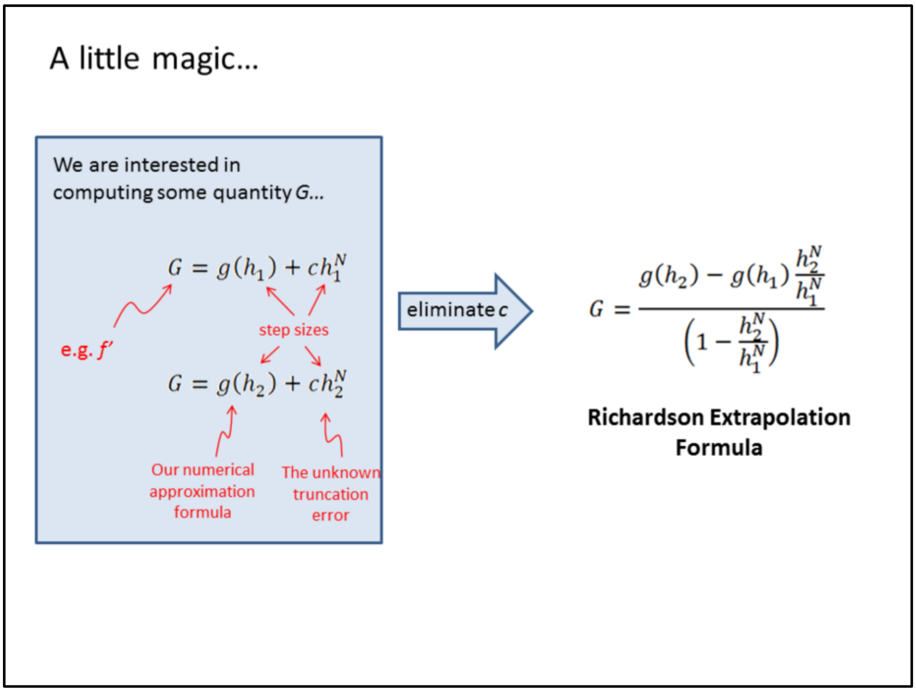
$$f''(t_i) = \frac{f(t_i) - 2f(t_{i-1}) + f(t_{i-2})}{(dt)^2} + O(dt)$$

Substitute the second derivative and solve for *f’*:

$$f'(t_i) = \frac{3f(t_i) - 4f(t_{i-1}) + f(t_{i-2})}{2dt} + O(dt^2)$$

A second order accurate, causal numerical estimate for the 1st derivative.

Note that we don’t have to stop here . For example, we could use what we’ve already figured out in the preceding slids to make more accurate estimates for the first derivatives using either centered, forward, or backward differences. We could also extend what we’ve done for the second derivative in order to calculate the third, fourth, etc. derivatives.



Before we leave this topic, we should examine something called Richardson extrapolation which will allow us to more accurately make estimates of our numerical derivatives without increasing the complexity of the formula's we are using.

The basic idea is that we are interested in estimating some quantity *G*, which in this case could be our first derivative. We do so by evaluating a formula (say, a first order accurate backward finite difference) that is a function of the step size, h1, and has an (unknown) error given by ch_1^N (if we keep with our 1st order accurate example, then N = 1). If we do so, we get some answer. But we should be able to get a similar answer if we did the exact same calculation but with a different step size, h2. In both cases, we know the form of the error and its dependency on the step size, but we don't know what c is. We will assume that the 'c' in both cases is approximately the same (or, rather, we will assume they are identical), and then combine both formula's to eliminate 'c'. If we do so, we are left with a formaul that depends only on our original estimtes, which are g(h1) and g(h2), the two step sizes h1 and h2, and the order of accuracy N. It turns out that this is a significantly more accurate estimate than either of the original estimates alone.

Richardson Extrapolation Example:

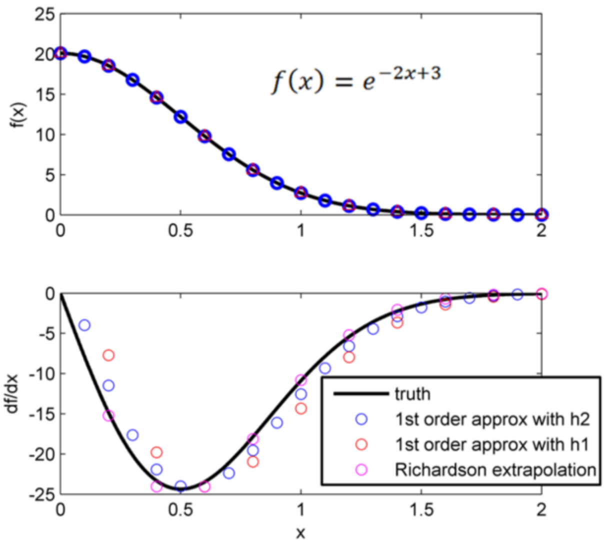
Here we are examining *f(x)* numerically with two step sizes:

h1 = 0.2

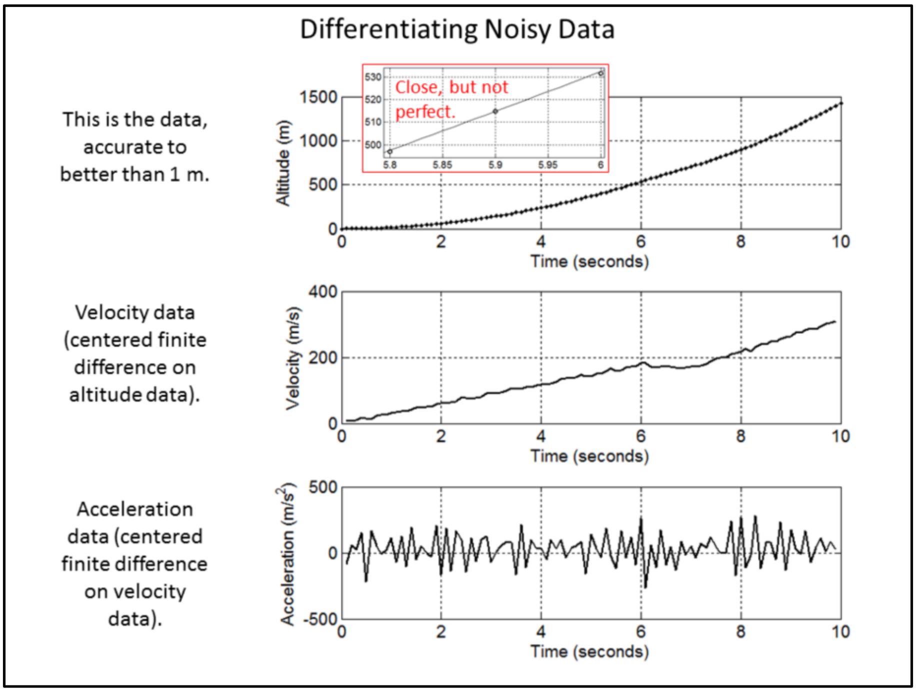
h2 = 0.1

Here we are estimating *df/dx* three different ways: 1st order accurate backward finite differencing for the sequences with h1 and h2, and by Richardson extrapolation.

$$f'(x_i) \approx \frac{f(x_i) - f(x_{i-1})}{h}$$



An example where we use Richardson extrapolation to estimate df/dx. Note the different step sizes in each of the approximations.



If we are trying to take derivatives of real data, we often run into problems caused by noise. As it turns out, derivatives *amplify* the noise in our data, and the order derivative we use the more trouble we can run into. Sometimes this is combatted by taking the original data and fitting it to polynomials, splines, etc. – this is a subject we will return to later on in the semester.

Take-home messages

- We've extended our ideas about Taylor series to first and second derivatives
  - Forward, backward, centered
- We also examined the remainder term so that we could estimate the order of accuracy for these finite differences
- We have introduced the idea of Richardson extrapolation for increasing the accuracy of our models
- Derivatives *amplify* the noise in our data