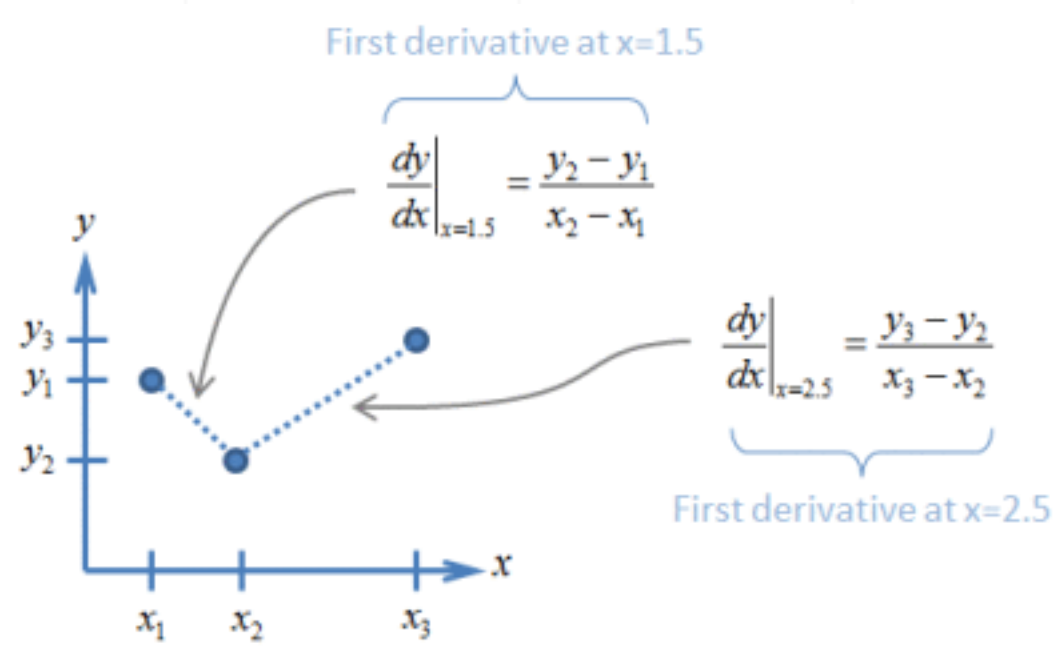
A Numerical Second Derivative from Three Points

January 12, 2013

3

What is the second derivative of three unevenly spaced points? Is there a formula for the numerical second derivative?

In practice, these points could represent measured data that we want to analyze like weather patterns or financial data. We will see that the second derivative is a linear combination of the three points.



We have three points, so we can find two first derivatives using those points.

Label the x and y coordinates for the three points and use the finite difference formula to calculate the first derivatives.

These derivatives are *not at the points*; they occur *between* pairs of points. For example, the derivative labeled x=1.5 is the derivative somewhere between x1 and x2. The number 1.5 suggests that it is precisely halfway between them, which is what we are assuming.

The red dots represent the first derivatives at x=1.5 and x=2.5. We can call these D1 and D2.

We use another finite difference formula to calculate the second derivative. The second derivative is the change in the first derivative divided by the distance between the points where they were evaluated. This is the same as “rise over run,” except that we replace the difference in y coordinates (the “rise”) with the difference in the first derivatives. In this example the difference between the first derivatives is D2 – D1. The distance between those derivatives (the “run”) is half of the distance between x1 and x3.

Now we plug in what we know about the first derivatives. This is the answer, but let’s take it a step further.

One thing we can do is to write the second derivative as a linear combination of the y coordinates for the three points. From this point forward we are doing algebra…

Now it is easy to see that the second derivative can be expressed as a linear combination of the y values.

Here is the Numerical Second Derivative Equation in Matrix Notation

Now for some examples

Example 1 – Evenly Spaced Points

Let's see how it works with evenly spaced points. The second derivative of $y = x^2$ is always 2, so this function is a good example. It turns out that the coefficients [1, -2, 1] work for *any* three points separated by 1 unit in x.

Example 2 – Unevenly Spaced Points

This is similar to the previous example, but the points are no longer evenly spaced. The third point is further away, and we see that the coefficients [2/3, -1, 1/3] place the smallest emphasis on the third point. That point is further away and for that reason its value could be very different from the values of the other two points. (In this case 25 is much larger than 4 or 9.) The small magnitude of the third coefficient helps to balance this out.

Example 3 – Nonquadratic Function

Now let's try something different. What if we use $y = x^3$ instead of $y = x^2$ to generate the points?

From calculus we know that the second derivative for $y = x^3$ is 6*x. At x = 3 this is 18… but the answer we got was 20. **What happened???**

The answer is: This is all being done numerically, not analytically. The numerical process does not have *a priori* knowledge about the function represented in those three points, so it uses the only model it knows how to use with three data points – a quadratic model.

Here's another way to look at it. This calculation process is automatically fitting a quadratic function through the three points and reporting the second derivative for that quadratic function.

If we had four points, we could derive a formula that works for four points. That formula would do the same thing, except that it would fit a cubic function instead of a quadratic function. We could do this for any number of points, but then the equations get complicated.

If we look at the cubic function and the fitted quadratic function, it is clear that the quadratic function is just an approximation in the neighborhood of x= 2, 3, and 5. However, it is a fairly good approximation in that neighborhood, so it makes sense that its first and second derivatives are also reasonably good approximations in that neighborhood. We can see from the x^2 coefficient (which is 10) that the quadratic function has a second derivative of 20. This agrees with the answer we got from the numerical model.

About these ads

Share this:

Twitter Facebook

Like

Be the first to like this.

Related

Introduction to Regularizing 3D Data – Part 2 of 2
In "Numerical Calculations"

RegularizeData2D, the Excel Spreadsheet Function for Regularizing 2D Data - Part 2 of 2
In "Numerical Calculations"

Introduction to Regularizing with 2D Data - Part 2 of 3
In "Numerical Calculations"

Introduction to Regularizing 3D Data – Part 2 of 2
In "Numerical Calculations"

RegularizeData2D, the Excel Spreadsheet Function for Regularizing 2D Data - Part 2 of 2
In "Numerical Calculations"

Introduction to Regularizing with 2D Data - Part 2 of 3
In "Numerical Calculations"

This entry was posted in Numerical Calculations and tagged Numerical Derivative, Second Derivative. Bookmark the permalink.

Bicubic Interpolation Excel Spreadsheet Function →

3 COMMENTS

lorfus | October 3, 2016 at 8:28 pm

Interesting, thanks!
How could we prove rigorously that the finite difference method gives the exact result for a quadratic function?

Reply

kintaar | October 3, 2016 at 10:00 pm

Hi Lorfus. Thanks for the question. You would have to fit a quadratic polynomial ax^2 + bx+ c to the three points and show that the second derivative from these equations equals 2*a from your fitted model. That would be sufficient to prove it.

On Monday, October 3, 2016, Math for Mere Mortals < comment-reply@wordpress.com > wrote:

>

Reply

Visal mahar | October 7, 2016 at 10:12 pm

How could we prove rigorously that the finite difference method gives the exact result for a quadratic function?

Reply

LEAVE A REPLY

Search …

RECENT POSTS

Cubic and Bicubic Interpolation Excel Functions

Minimize and MinimizeNonnegative Spreadsheet Optimization v6

Constrained Nonlinear Optimization in Excel *without* the Excel Solver

Excel Spreadsheet Function to Optimize Cells (without Using the Excel Solver or Goal Seek)

RegularizeData3D – the Excel Spreadsheet Function to Regularize 3D Data to a Smooth Surface

ARCHIVES

April 2014

November 2013

September 2013

July 2013

February 2013

January 2013

CATEGORIES

Interpolation

Numerical Calculations

Optimization

Regularizing

Uncategorized

VBA Spreadsheet Function

META

Register

Log in

Entries RSS

Comments RSS

WordPress.com

Blog at WordPress.com.