Richardson Extrapolation to compute the second derivative

Alexandra Mulholland 17336557 02/10/19

Aim and Abstract

The purpose of this investigation was to determine the accuracy of The Richardson Extrapolation- a numerical technique used to evaluate numerical derivatives while eliminating leading error terms- an issue which arose using Central difference method. The function to be differentiated was $f(x) = x \cdot e^x$ with its second derivative being $f"(x) = (x+2) \cdot e^x$. The analytical result for this was used for comparison and to determine relative errors. Estimations of the second derivative called $D_1(h),\ D_1(2h),\ D_2(h),\ D_3(h)$ and $D_4(h)$ (explained below) were found beginning with step size 0.4, evaluated at x=2. It was found that $D_4(0.05)$ was the most accurate estimation, most closely resembling the analytical result. This method was then extended in part 3 up to estimation value $D_8(h)$ under the same parameters. It was found that $D_5(0.025)$ produced the most accurate result, with the corresponding minimum relative error being $3.0411075132675167 \times 10^{-14}$. Part 4 involved analysing the change in accuracy for various estimations as h decreases. The scaling relationship was then determined, being 2^{2n}. Part 5 tested the change in rounding error when testing for smaller initial values of h, and the corresponding h value that leads to the highest accuracy, which was found to be 0.2.

Mathematical background

Previously, we had analysed the accuracy of the Central Difference approximation, with step size h. The Richardson extrapolation extends upon this method to better the estimations, by eliminating leading error terms.

We know the central difference approximation for the second derivative is:

$$f"(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + O(h^2)$$

$O(h^2)$ being an abbreviation for the leading error term ($\frac{h^2}{24} f^4(x)$).

For the purposes of this investigation, we will call this formula $D_1(h)$ and doubling the step size to 2h gives us $D_1(2h)$. We find that the leading error term is eight times larger when we apply this, however, the Richardson Extrapolation allows us to eliminate this error by subtracting $\frac{1}{8} D_1(2h)$ from $D_1(h)$.

$$f''(x) = \frac{f(x+2h) - 2f(x) + f(x-2h)}{4h^2} + O(h^2)$$

$O(h^2)$ in the case of $D_1(2h)$ being $\frac{h^2}{3} f^4(x)$.

We can then compute $D_2(h), D_3(h)$ etc. using the general formula below:

$$D_{n+1}(h) := D_n(h) - \frac{D_n(2h) - D_n(h)}{2^{2n} - 1} = \frac{2^{2n} D_n(h) - D_n(2h)}{2^{2n} - 1}$$

As n increases, these are considered improved estimates of the second derivative.

This is inputted into a computer scheme as a matrix, whereby the code decreases h in iterations and fills in cells to give values obtained for $D_1(h), D_2(h)$ etc. Beyond an estimation value of $D_8(h)$, rounding errors become too great.

Method

The function to be differentiated was $f(x) = x \cdot e^x$. The second derivative of this function is $f''(x) = (x+2) \cdot e^x$. This was computed analytically using "numpy" and plotted for x ranging from 0 to 2 in steps of size 0.0001.

The Richardson extrapolation was first applied with a constant step size, h, initialised to 0.4 and decreasing by half at each iteration. A code to evaluate the second derivative using estimations $D_1(h), D_1(2h), D_2(h), D_3(h)$ and $D_4(h)$ was inputted as a list. The results were tabulated, displaying the values obtained for the second derivative at the point x=2.

This list was then extended to compute estimations up to and including $D_8(h)$. The most accurate estimation for the second derivative computed at x=2 was then obtained, the value of h at which this occurred and the corresponding relative error. In both lists, any empty cells were coded to be zero, explained later under table 3.1.

The step size was then decreased, the corresponding increase in accuracy was analysed, and the scaling relationship determined. This was done by defining relative error for each estimation and plotting these values for various initial step sizes. The rounding error is then evident in the plots.

Results and Discussion

Figure 1.1a

This figure shows the function of which the second derivative was found.

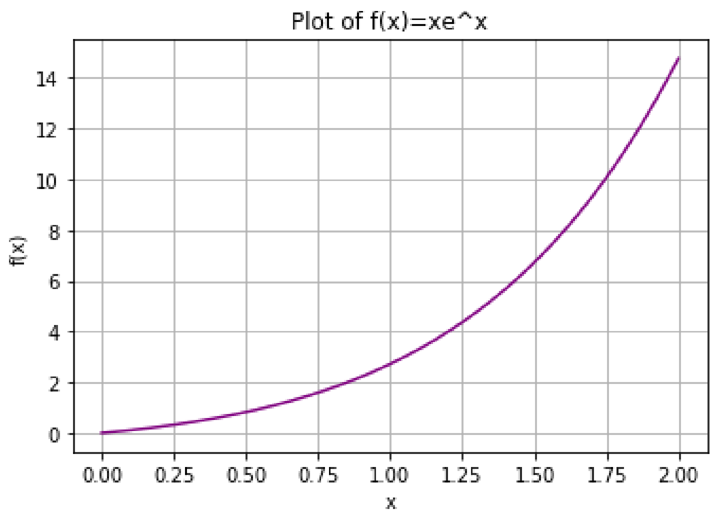


Figure 1.1b

This figure shows a plot of the analytical second derivative of f(x): $f''(x) = (x + 2) \cdot e^{x}$.

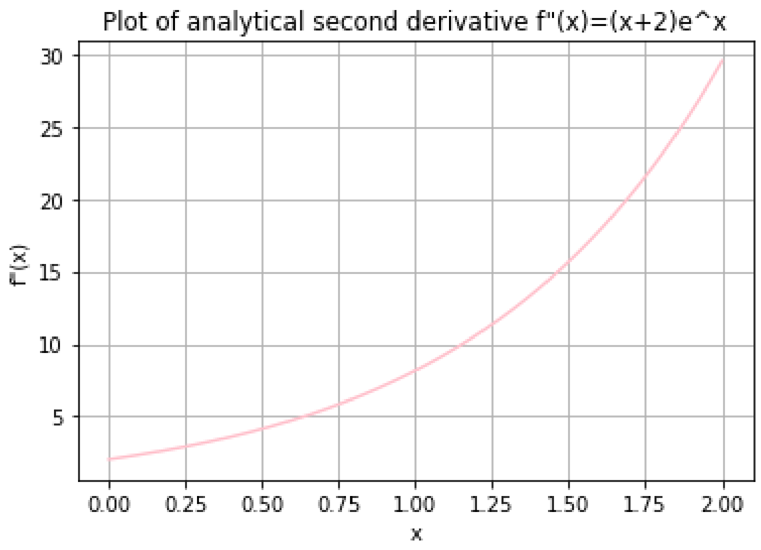


Figure 2.1a

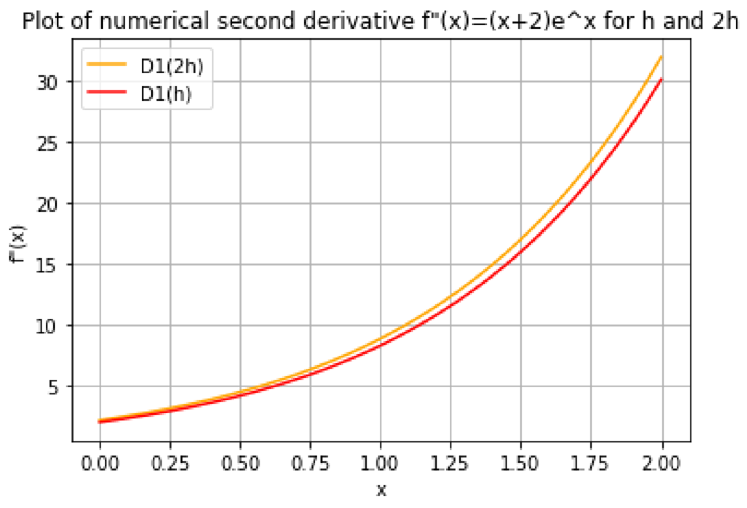


Figure 2.1a above shows the results for the central difference approximation (or $D_1(h)$) for the second derivative of the function. The red line is the result when using a step size, h, and the orange line that when using step size, 2h. When comparing to the analytical result, we can see in figure 2.1b that using step size h is closer to the accurate result. This is as expected, as doubling the step size in this way leads to an increased leading error by a factor of 8. This is eliminated when the estimation is improved ($D_2(h), D_3(h)$ etc.). Here, h was kept constant at 0.4.

Figure 2.1b

This figure is a magnified version of figure 1.2a whereby it was plotted for x in the range from 0 to 0.01, with the analytical plot included for comparison. We can see more clearly here that the central difference approximation for step size h is much more accurate than that for step size 2h.

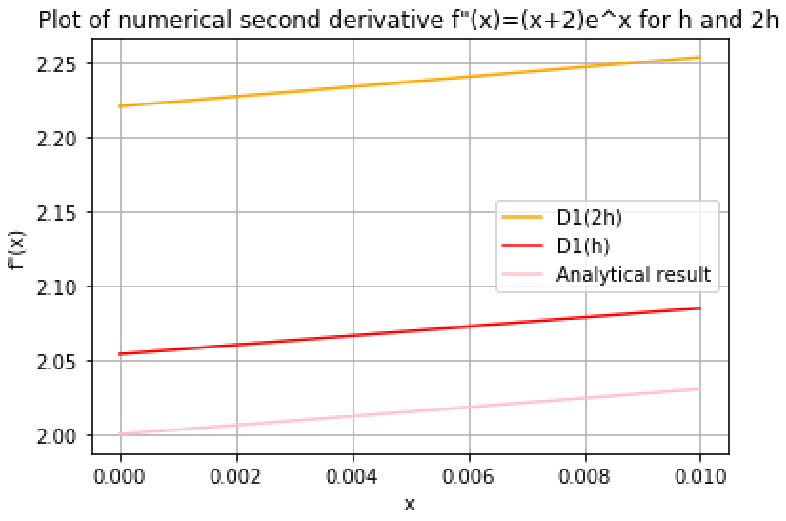


Table 2.1

Analytical: 29.5562243957226

H values	$D_1(h)$	$D_2(h)$	$D_3(h)$	$D_4(h)$
0.4	30.1515674803089			
0.2	29.70426847439435	29.555168805756168		
0.1	29.593186100007244	29.55615864187821	29.55622463095301	
0.05	29.56546174215901	29.55622028954293	29.556224399387247	29.5562243957116

The correct value for the second derivative evaluated at x=2 is the analytical result. The table shows the matrix produced by the code. $D_4(h)$ produces the most accurate estimation and occurs step size 0.05, being approximately 5.62×10^{-8} units off the true value.

Table 3.1

H values	$D_1(h)$	$D_2(h)$	$D_3(h)$	$D_4(h)$	$D_5(h)$	$D_6(h)$	$D_7(h)$	$D_8(h)$
0.4	30.1515674803089							
0.2	29.70426847439435	29.555168805756168						
0.1	29.593186100007244	29.55615864187821	29.55622463095301					
0.05	29.56546174215901	29.55622028954293	29.556224399387247	29.5562243957116				
0.025	29.558533539895386	29.55622413914084	29.556224395780703	29.556224395723454	29.5562243957235			
0.0125	29.55680166975298	29.556224379705508	29.55622439574315	29.556224395742554	29.55622439574263	29.556224395742646		
0.00625	29.556368713474505	29.556224394715013	29.55622439571565	29.556224395715212	29.556224395715102	29.556224395715073	29.556224395715066	
0.003125	29.556260474419098	29.55622439473396	29.556224394735224	29.556224394719663	29.55622439471576	29.556224394714782	29.55622439471454	29.55622439471448

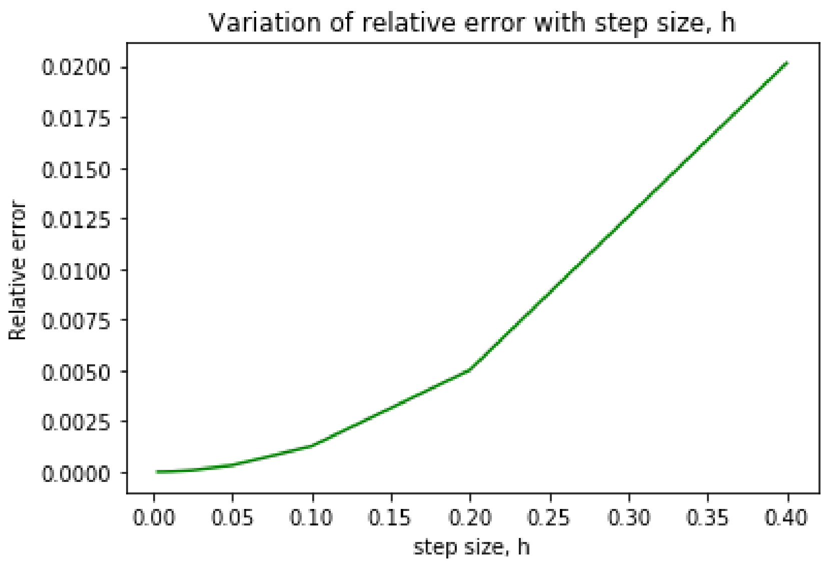
For part 3, we extended the range to which h would go, thus extending the approximations to $D_8(h)$. Beyond this, rounding errors begin to negatively affect the outcome. Table 3.1 shows the extended estimations evaluated at x=2. The code reads the blank spaces in the table to be zeros and uses the formula to generate an unwanted result. Therefore, for columns 1 to 4 where the column cell value was greater than the row value, for instance, $D_5(h)$ at h=0.05 (column 4, row 3), the cell value was coded to be zero.

$D_5(0.025)$ produces the most accurate value. The code to verify this is shown below, along with the results.

```
#D5 values, for h = 0.00625, 0.0125 and 0.025 respectively

print ("")
print (29.556224395715102-analytic2(x))  =   -7.499778575947857e-12
print (29.55622439574263-analytic2(x)) = 2.0026647007398424e-11
print (29.5562243957235-analytic2(x)) = 8.988365607365267e-13


#D6 Values
print ("")
print (29.556224395742646-analytic2(x)) =  2.0044410575792426e-11
print (29.556224395715073-analytic2(x))= -7.528200285378261e-12

#D7 value
print ("")
print (29.556224395715066-analytic2(x)) = -7.535305712735862e-12
```

With the smallest relative error at this value of h being $3.0411075132675167 \times 10^{-14}$.

Figure 4.1



This figure above shows that for the central difference approximation, the relative error increases with step size h, with a scale factor approximately equal to 2^{2n}.

Figure 4.2a

In these figures, the second derivative has again been evaluated at x=2, with various initial values of h. Figures 4.2a, b, c and d are all log log plots.

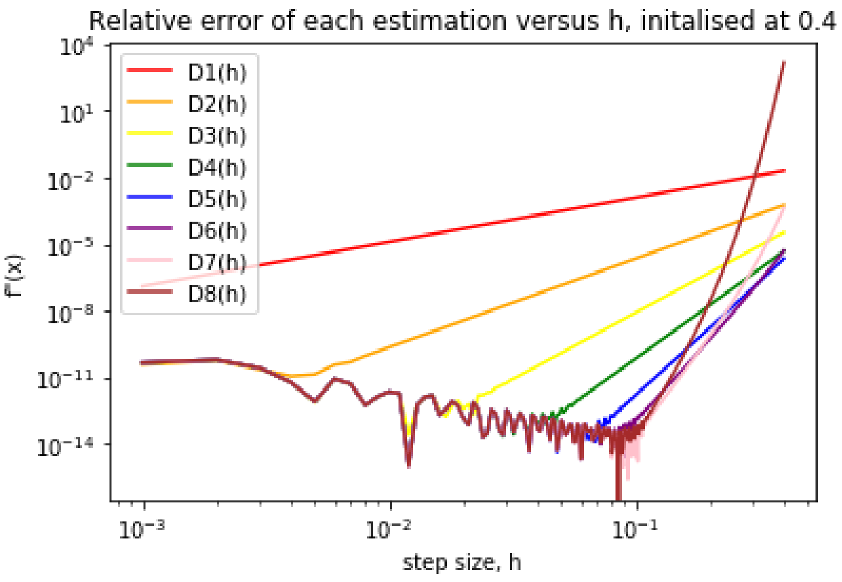


Figure 4.2b

All the figures show a rapid increase in the value of the second derivative for certain initial values of h. $D_1(h)$ is the quickest to deviate from the expected values, and $D_8(h)$ the latest, indicating it is the most accurate. However, it is more suspect to rounding errors.

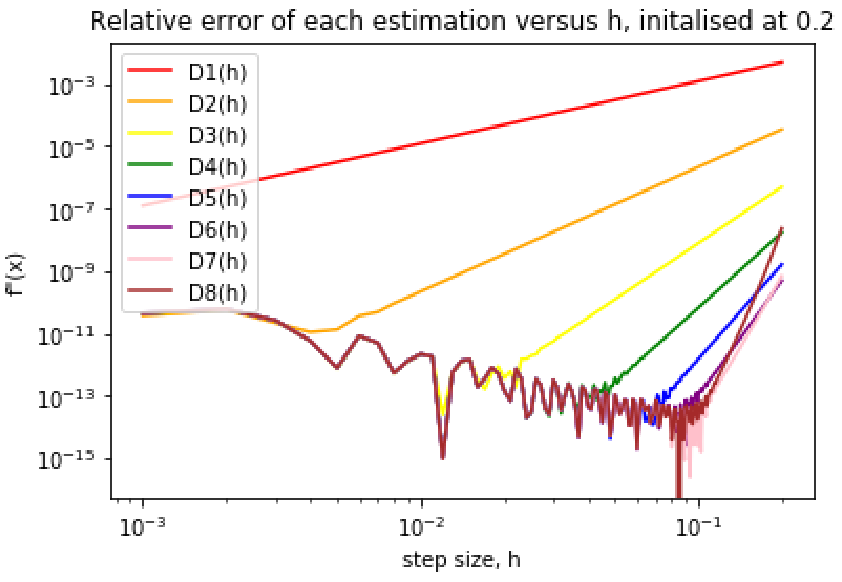


Figure 4.2c

We can see the rounding error increases with decreased h initial values.

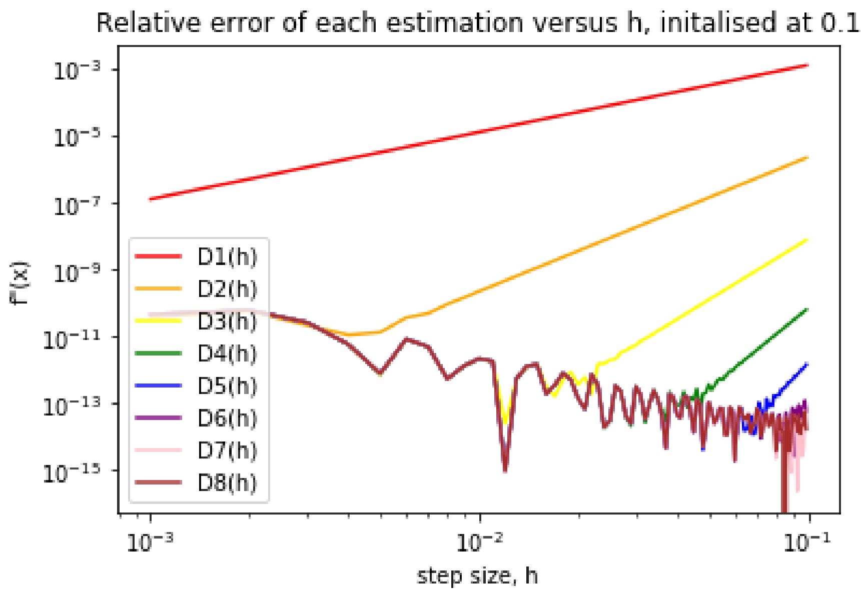
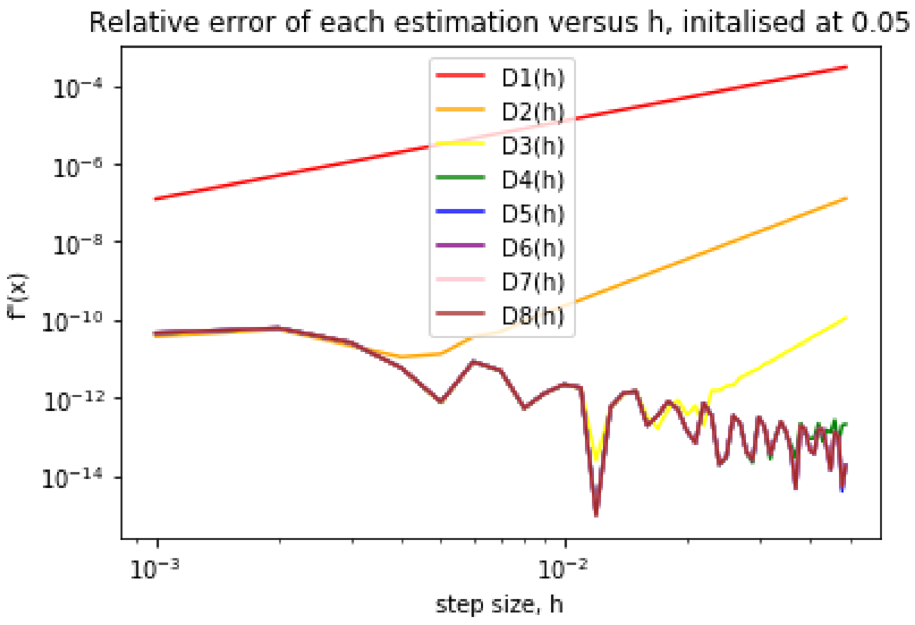


Figure 4.2d

The jagged line pattern we see in each figure represent the various rounding errors which occur at different step sizes for each estimation. Increasing the sample size will result in straighter trends.



Conclusion

Part 1 of the investigation went as expected, the analytical result being used as a reference and for relative errors, in order to better see how the estimations improved. For part 2, the most accurate estimation occurred for $D_4(h)$. Here, for h=0.05, the value for the second derivative at x=2 was 29.5562243957116. This estimation approximately 5.62×10^{-8}, closer than previous estimations.

Part 3 extended on these iterations and it was found that the most accurate estimation was $D_5(h)$, evaluated for step size 0.025, producing a value of 29.5562243957235, just 8.988365607365267$\times$ 10^{-13} units off the expected value. The resultant relative error was 3.0411075132675167 $\times$ 10^{-14}.

Parts 4 and 5 show that the relative error increases with h with scale factor 2^{2n}. It can be seen the relationship may become linear beyond h around 0.2. In part 5, it was deduced that the rounding error increases as the initial value of h is decreased. And the most accurate value occurs at h=0.2.

One can conclude that the Richardson Extrapolation method is effective in accurately estimating a value for the second derivative of $f(x) = x \cdot e^x$.