

Numerical Differentiation⁻¹⁻

The **basic idea** of Numerical Differentiation is deceptively simple. To approximate the derivative of a function interpolate at suitably chosen points by a polynomial and differentiate the polynomial.

Here is **an example**. Let f be a given function. Let p be the quadratic polynomial that interpolates at x_0 , $x_1 = x_0 + h$, and $x_2 = x_0 + 2h$. Thus

$$p(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)}f(x_0) + \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)}f(x_1) + \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}f(x_2) \quad (1)$$

Differentiating once or twice, respectively, and evaluating at x_1 (to obtain any advantages of symmetry, and avoid artifacts of asymmetry) gives

$$f'(x_1) \approx p'(x_1) = \frac{f(x_2) - f(x_0)}{2h} + E_1 \quad (2)$$

and

$$f''(x_1) \approx p''(x_1) = \frac{f(x_2) - 2f(x_1) + f(x_0)}{h^2} + E_2 \quad (3)$$

These are the two most widely used numerical differentiation formulas in existence. In particular, they are ubiquitous in the numerical solution of differential equations. Of course it's clear that (2) just gives the slope of the secant through the points $(x_0, f(x_0))$ and $(x_2, f(x_2))$. The formula (3) can be derived by applying the secant idea twice:

$$\begin{aligned} f''(x_1) &\approx \frac{f'(x_1 + h/2) - f'(x_1 - h/2)}{h} \\ &\approx \frac{\frac{f(x_2) - f(x_1)}{h} - \frac{f(x_1) - f(x_0)}{h}}{h} \\ &= \frac{f(x_2) - 2f(x_1) + f(x_0)}{h^2} \end{aligned} \quad (4)$$

The truncation errors E_1 and E_2 in these approximations are easily obtained by expanding into a Taylor series. All evaluations are at x_1 , and to avoid clutter we omit the argument. Ignoring higher order terms we obtain

$$\begin{aligned} E_1 &= f'(x_1) - p'(x_1) = \\ &= -\frac{1}{2h} \left[\begin{array}{cccc} & f' & & \\ f & +hf' & +\frac{h^2}{2}f'' & +\frac{h^3}{6}f''' \\ -(f & -hf' & +\frac{h^2}{2}f'' & -\frac{h^3}{6}f''') \end{array} \right] \\ E_1 &= -\frac{h^2}{6}f''' \end{aligned} \quad (5)$$

and, similarly,

$$\begin{aligned}
E_2 &= f''(x_1) - p''(x_1) = \\
&= -\frac{1}{h^2} \left[\begin{array}{cccccc} & & f'' & & & \\ & f & +hf' & +\frac{h^2}{2}f'' & +\frac{h^3}{6}f''' & +\frac{h^4}{24}f'''' \\ & -2f & & & & \\ & f & -hf' & +\frac{h^2}{2}f'' & -\frac{h^3}{6}f''' & +\frac{h^4}{24}f'''' \end{array} \right] \\
E_2 &= -\frac{h^2}{12}f''''
\end{aligned} \tag{6}$$

So, in both cases, the error goes to zero as h^2 goes to zero, which is quite satisfactory.

This, however, is not what happens.

The ancient Fortran code listed in Table 1 approximates some derivatives of the exponential⁻²⁻

$$f(x) = e^x \tag{7}$$

at $x = 1$, using different formulas. The formulas are all taken from page 914 of Abramowitz/Stegun, Handbook of Mathematical Functions (the page in that book that I use the most).

The code produces the output shown in Table 2. The *roundoff unit*

$$\tau = 2.22 \times 10^{-16}$$

shown in the first line of that Table is the smallest positive number τ for which my computer (a Unix machine) recognizes that $1 + \tau > 1$. That number is computed in lines 2–6 of the code.

The columns of the output have the following meaning:

1 The line numbers 1–35 are not part of the output. They are provided by T_EX.

n A number running from 0 to 30.

h The discretization parameter,

$$h = \frac{1}{2^n} \tag{8}$$

Thus, going from one line to the next means that h is **halved**.

D1 The errors in the approximation of $f'(1)$ by the formula (2). The errors (in this and all remaining columns) are given as **base 10 logarithms**. Thus we would like the entries in that column to be negative and have large absolute values.

D1B The errors in the alternative first order differentiation formula

$$f'(x_2) \approx \frac{2f(x_0) - 16f(x_1) + 16f(x_3) - 2f(x_4)}{24h} \tag{9}$$

which uses more points and has a truncation error that is $\mathcal{O}(h^4)$.

D2 The error in the second order derivative approximation is given by (3).

⁻²⁻ The exponential is particularly convenient for investigations like this since all of its derivatives are the same. Any error behavior involving derivatives of different degrees is not going to be an artifact of different derivatives having different values.

```

1      implicit double precision (a - h, o - z)
2      h = 1.0d0
3 10    continue
4      h = h/2.d0
5      if (1.0d0 + h .gt. 1.0d0) go to 10
6      h = h + h
7      write (20, 1020) h, dlog10(h)
8      e = dexp(1.0d0)
9      write (20, 1000)
10     do 20 i = 0, 30
11         h = 2.0d0**(- i)
12         dp1 = dexp(1.0d0 + h)
13         dm1 = dexp(1.0d0 - h)
14         dp2 = dexp(1.0d0 + 2.0d0*h)
15         dm2 = dexp(1.0d0 - 2.0d0*h)
16         d1 = dlog10(dabs(e - (dp1 - dm1)/(2.0d0*h)))
17         d1b = dlog10(dabs(e - (2.0d0*dm2 - 16.0d0*dm1 + 16.0d0*dp1 -
18 +           2.0d0*dp2)/(24.0d0*h)))
19         d2 = dlog10(dabs(e - (dp1 - 2.0d0*e + dm1)/(h**2)))
20         d2b = dlog10(dabs(e - (- dm2 + 16.0d0*dm1 - 30.0d0*e + 16.0d0*
21 +           dp1 - dp2)/(12.0d0*h**2)))
22         d3 = dlog10(dabs(e - (- 2.0d0*dm2 + 4.0d0*dm1 - 4.0d0*dp1 +
23 +           2.0d0*dp2)/(4.0d0*h**3)))
24         d4 = dlog10(dabs(e - (dm2 - 4.0d0*dm1 + 6.0d0*e - 4.0d0*dp1 +
25 +           dp2)/h**4))
26         write (20, 1010) i, h, d1, d1b, d2, d2b, d3, d4
27     20 continue
28     stop
29 1000 format ('  n  h=1/2**n ', 9x, 'D1', 3x, 'D1B', 4x, 'D2', 3x,
30 + 'D2B', 4x, 'D3', 4x, 'D4')
31 1010 format (i3, d12.4, 4x, 7f6.1)
32 1020 format (' ====> roundoff unit=', d12.4, ' LOG = ', f6.1, '/')
33     end

```

Table 1: Fortran Code

D2B The error in the alternative differentiation formula

$$f''(x_2) \approx \frac{-f(x_0) + 16f(x_1) - 30f(x_2) + 16f(x_3) - f(x_4)}{12h^2} \quad (10)$$

which has an error of $\mathcal{O}(h^4)$.

D3 The error in the third order differentiation formula

$$f'''(x_2) \approx \frac{-2f(x_0) + 4f(x_1) - 4f(x_3) + 2f(x_4)}{4h^3} \quad (11)$$

which has an error of $\mathcal{O}(h^2)$.

D4 The error in the fourth order differentiation formula

$$f''''(x_2) \approx \frac{f(x_0) - 4f(x_1) + 6f(x_2) - 4f(x_3) + f(x_4)}{h^4} \quad (12)$$

which has an error of $\mathcal{O}(h^2)$.

Notice the following points:

1	====> roundoff unit= 0.2220D-15 LOG = -15.7							
2								
3	n	h=1/2**n	D1	D1B	D2	D2B	D3	D4
4								
5	0	0.1000D+01	-0.3	-1.0	-0.6	-1.5	-0.1	-0.3
6	1	0.5000D+00	-0.9	-2.2	-1.2	-2.7	-0.8	-0.9
7	2	0.2500D+00	-1.5	-3.4	-1.8	-3.9	-1.4	-1.5
8	3	0.1250D+00	-2.1	-4.7	-2.5	-5.1	-2.0	-2.1
9	4	0.6250D-01	-2.8	-5.9	-3.1	-6.3	-2.6	-2.8
10	5	0.3125D-01	-3.4	-7.1	-3.7	-7.5	-3.2	-3.4
11	6	0.1562D-01	-4.0	-8.3	-4.3	-8.7	-3.8	-4.0
12	7	0.7812D-02	-4.6	-9.5	-4.9	-10.0	-4.4	-4.5
13	8	0.3906D-02	-5.2	-10.7	-5.5	-10.2	-5.0	-4.6
14	9	0.1953D-02	-5.8	-11.9	-6.1	-9.7	-5.6	-3.5
15	10	0.9766D-03	-6.4	-12.5	-6.7	-9.4	-6.1	-2.4
16	11	0.4883D-03	-7.0	-12.3	-7.3	-8.6	-5.6	-1.2
17	12	0.2441D-03	-7.6	-12.4	-8.2	-8.4	-4.7	-0.1
18	13	0.1221D-03	-8.2	-11.9	-7.3	-7.3	-4.7	1.3
19	14	0.6104D-04	-8.8	-11.8	-6.7	-6.7	-3.3	2.5
20	15	0.3052D-04	-9.4	-12.0	-6.3	-7.4	-2.1	3.6
21	16	0.1526D-04	-10.0	-10.7	-5.5	-5.5	-1.0	4.8
22	17	0.7629D-05	-10.7	-12.6	-4.9	-5.4	-0.7	6.1
23	18	0.3815D-05	-10.3	-10.2	-4.3	-4.5	0.8	7.3
24	19	0.1907D-05	-11.0	-10.0	-3.8	-4.0	1.5	8.4
25	20	0.9537D-06	-9.9	-10.3	-3.3	-4.2	2.4	9.5
26	21	0.4768D-06	-9.9	-9.9	-2.8	-2.8	0.4	10.7
27	22	0.2384D-06	-9.5	-9.5	-2.1	-2.2	4.2	11.9
28	23	0.1192D-06	-9.2	-8.7	-1.5	-1.7	5.1	13.2
29	24	0.5960D-07	-9.2	-10.5	-0.7	-0.8	0.4	14.6
30	25	0.2980D-07	-8.2	-8.1	-0.1	-0.2	7.2	15.6
31	26	0.1490D-07	-8.2	-7.9	0.4	-0.4	0.4	16.7
32	27	0.7451D-08	-8.2	-7.9	1.0	-0.1	0.4	18.1
33	28	0.3725D-08	-7.6	-7.5	1.5	1.0	9.6	19.2
34	29	0.1863D-08	-7.4	-7.4	2.1	1.4	10.5	20.3
35	30	0.9313D-09	-6.8	-6.5	2.7	1.9	11.4	21.5

Table 2: Numerical Results

- The system carries about 16 decimal digits.
- In the column D1, the error starts out as $10^{-0.3} \approx 0.5$. Halving h , and going to the next line gives an error $10^{-0.9} \approx 0.125$. This is consistent with the error being of $\mathcal{O}(h^2)$. Indeed, since the base 10 logarithm of $1/4$ is about -0.6, in going down that column, one expects that the base 10 logarithm of the error decreases by 0.6 in every line. That pattern holds until $n = 17$. Then, however, the error **increases** when going to the next line. Subsequently, the error fluctuates somewhat but never gets less than about 10^{-11} . This means our approximation of the first derivative at best is about 100,000 times worse than the accuracy with which that derivative can be represented on the computer!
- As column D1b shows, using more points does not greatly increase the accuracy. The smallest error is obtained for $h = 2^{-12}$ and is only about 30 times as small as the smallest error in the column D1.
- Columns D2 and D2B exhibit a similar pattern as columns D1 and D1B, except that the best accuracy of the second derivative is about 10,000 times lower than the best accuracy of the first derivative.
- Column D3 shows that the smallest error in the third derivative is about 10^{10} times the accuracy

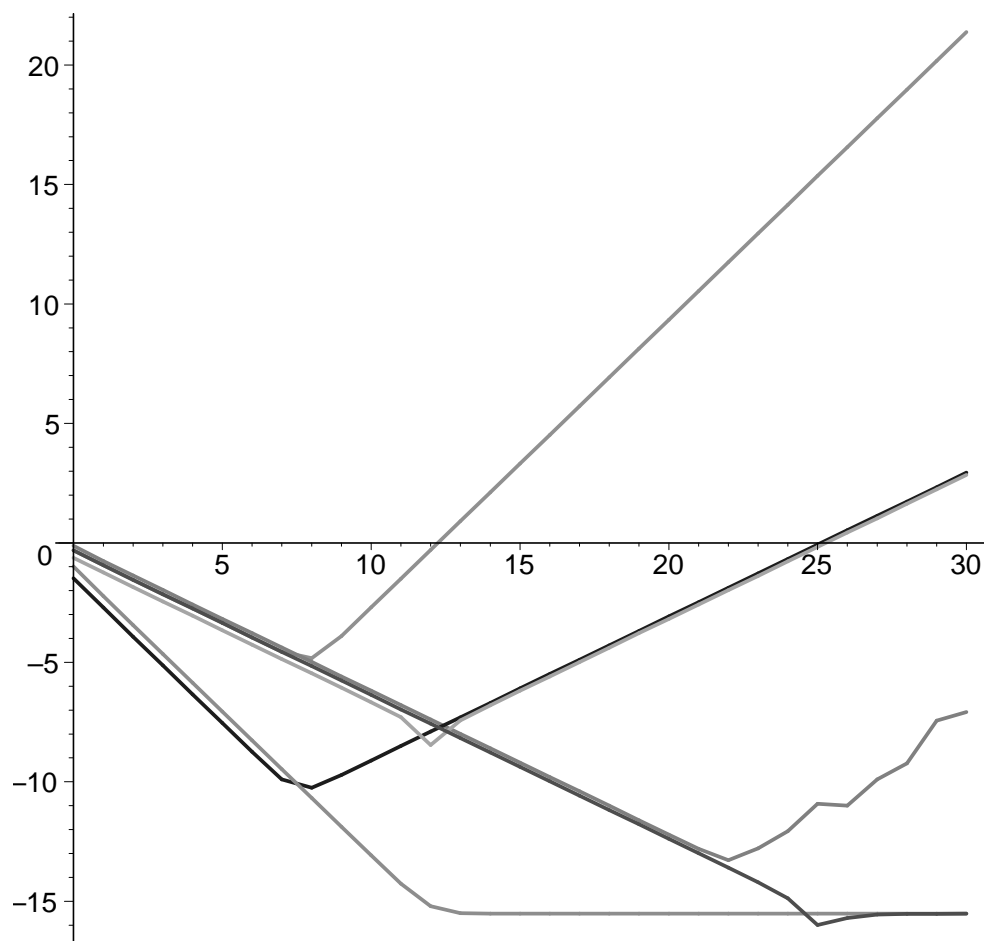


Figure 1. Accuracy of Derivatives, $f(x) = e^x$, $x = 1$.

of the computing system.

- The last column shows that the smallest error in the fourth derivative is about 10^{12} times the accuracy of the computing system.

The data in Table 2 are graphed in Figure 1. This is a log-log plot and so a monomial function $\phi(h) = h^p$ would show as a straight line. For all graphs, this is indeed the case for large values of h . However, the error becomes erratic when h is small enough for round-off errors to dominate, and it tends to increase as h decreases.

Things look grim. What's happening?

The common explanation of these difficulties is that when approximating derivatives we compute the differences of almost equal numbers. This leads to a cancellation of significant digits, and a loss of accuracy.

This begs the question: how about a formula that avoids those differences?

Actually, there is a fundamental reason why numerical differentiation is so difficult. Strictly speaking, it's impossible. The reason is that a function can be represented only within a certain error, and within that error, any derivative value at all is consistent with any available data. As illustrated in Figure 3, think of the graph of f as only being known to contained in a ribbon. Within

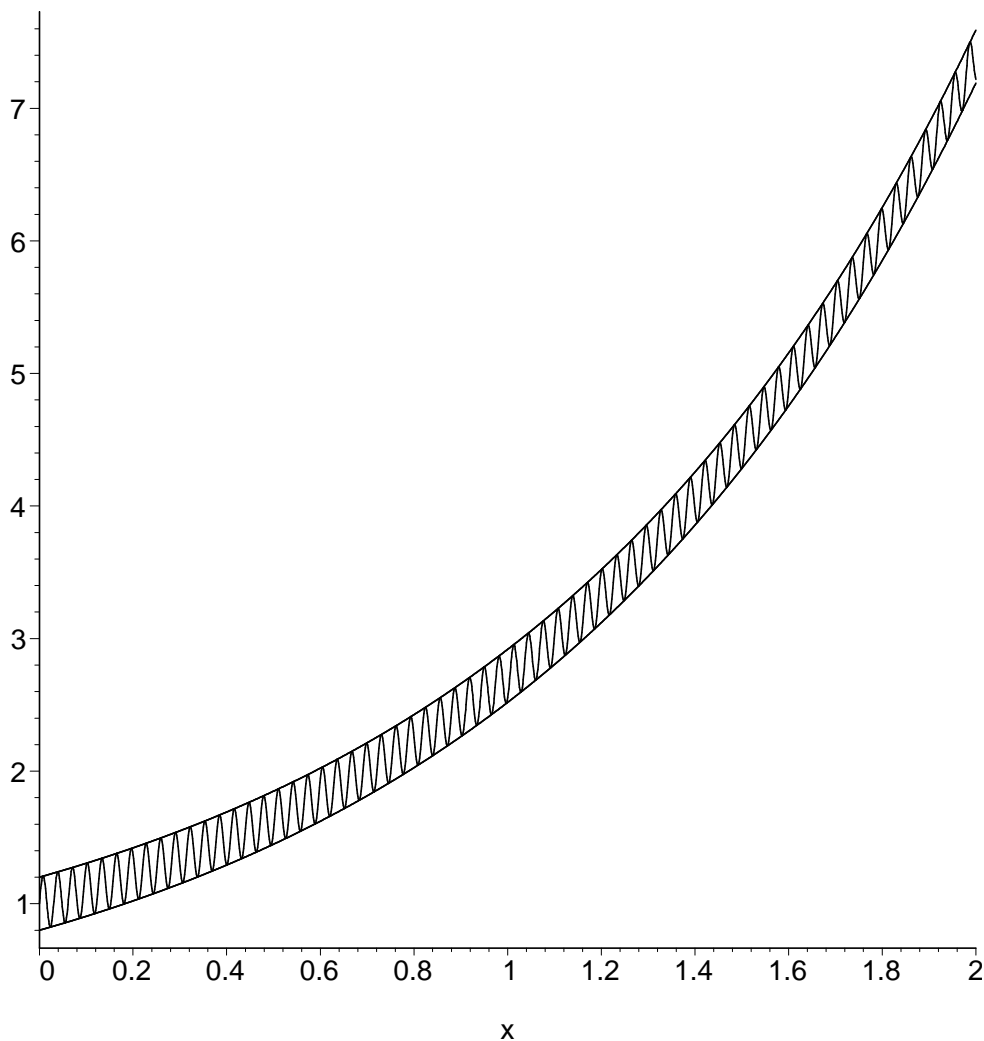


Figure 2. Function Accuracy, $f(x) = e^x$, $\epsilon = 0.2$.

that ribbon it exhibit oscillations that could cause it to have any (first or higher order) derivative of any value at all.

Practically speaking one needs to strike a balance between the truncation error and the round-off error. There is an optimal value of h . However, it is not easy in general to compute that value of h . It varies widely and depends on the function, the degree of the derivative, and the formula. In many algorithms, for example, for the minimization of functions of several variables, numerical differentiations occur deep inside sets of nested loops, and need to be carried out automatically, without human assistance. The algorithms may be very sensitive to the quality of those derivative estimates.

For a detailed discussion of numerical differentiation, see, for example, Section 8.6 of Gill, Murray, Wright, Practical Optimization, Academic Press, 1981.

Here is an example of the kind of calculation one can do.

Suppose we can evaluate f within a tolerance ϵ . Consider our first order differentiation formula

(2). We obtain

$$\begin{aligned}
f'(x) &= \frac{f(x+h) + \epsilon_+ - f(x-h) - \epsilon_-}{2h} \quad \text{where } |\epsilon_+|, |\epsilon_-| \leq \epsilon \\
&= \frac{f(x+h) - f(x-h)}{2h} + \frac{\epsilon_+ - \epsilon_-}{2h} \\
&= f'(x) - \frac{h^2}{6} f''' + \frac{\epsilon_+ - \epsilon_-}{2h}
\end{aligned} \tag{13}$$

The basic idea whenever one has two competing error terms is to balance them by making them, or bounds on them, approximately equal. The second term can be bounded:

$$\left| \frac{\epsilon_+ - \epsilon_-}{2h} \right| \leq \frac{2\epsilon}{2h} = \frac{\epsilon}{h}. \tag{14}$$

So we consider the equation

$$\frac{h^2}{6} |f'''| = \frac{\epsilon}{h} \tag{15}$$

which has the solution

$$h = \sqrt[3]{\frac{6\epsilon}{f'''}}. \tag{16}$$

Of course, it's not very satisfactory that computing a first derivative involves using the third derivative. However, one may be able to use a bound on the third derivative or an estimate. We can also check the validity of (16) for our example. Letting $\epsilon = 10^{-16}$ (the round-off unit) and $f''' = e$ (the value of the exponential at $x = 1$), we obtain

$$h = \sqrt[3]{\frac{6 \times 10^{-16}}{e}} \approx 2^{-17.33}. \tag{17}$$

Thus the best accuracy in column D1 of Table 2 should occur at or near $n = 17$, which is consistent with the Table.

Numerical differentiation is hard. Luckily, differentiation is fairly easy analytically. By comparison, integration is hard analytically, but easy numerically.

So, ponder this table:

	Integration	Differentiation
Analytical	Hard	<i>Easy</i>
Numerical	<i>Easy</i>	Hard

The human species would be in trouble if both words **Hard** were to appear in one column. They don't. Is this a matter of luck, or is there some fundamental mathematical reason? Let me know what you think!

COEFFICIENTS FOR DIFFERENTIATION

Differentiation Formula: $\frac{dkf(x)}{dx^k} \Big|_{x=x_j} = \frac{k!}{m!h^k} \sum_{i=0}^m A_i f(x_i)$

FIRST DERIVATIVE (k=1)								THIRD DERIVATIVE (k=3)								
* j	A ₀	A ₁	A ₂	A ₃	A ₄	A ₅	$\frac{h^k}{k!}$ Error	j	A ₀	A ₁	A ₂	A ₃	A ₄	A ₅	$\frac{h^k}{k!}$ Error	*
Three Point (m=2)								Four Point (m=3)								
0	-3	4	-1				$\frac{1}{3}$	0	-1	3	-3	1			$-\frac{1}{4}$	
1	-1	0	1				$-\frac{1}{6} h^3 f^{(3)}$	1	-1	3	-3	1			$-\frac{1}{12} h^4 f^{(4)}$	
2	1	-4	3				$\frac{1}{3}$	2	-1	3	-3	1			$\frac{1}{12} h^4 f^{(4)}$	
Four Point (m=3)								Five Point (m=4)								
0	-11	18	-9	2			$-\frac{1}{4}$	0	-10	36	-48	28	-6		$\frac{7}{24}$	
1	-2	-3	6	-1			$\frac{1}{12}$	1	-6	20	-24	12	-2		$\frac{1}{24}$	
2	1	-6	3	2			$-\frac{1}{12} h^4 f^{(4)}$	2	-2	4	0	-4	2		$-\frac{1}{24} h^5 f^{(5)}$	
3	-2	9	-18	11			$\frac{1}{4}$	3	2	-12	24	-20	6		$\frac{1}{24}$	
Five Point (m=4)								Six Point (m=5)								
0	-50	96	-72	32	-6		$\frac{1}{5}$	0	-85	355	-590	490	-205	35	$-\frac{5}{16}$	
1	-6	-20	36	-12	2		$-\frac{1}{20}$	1	-35	125	-170	110	-35	5	$-\frac{1}{48}$	
2	2	-16	0	16	-2		$\frac{1}{30} h^5 f^{(5)}$	2	-5	-5	50	-70	35	-5	$\frac{1}{48} h^6 f^{(6)}$	
3	-2	12	-36	20	6		$-\frac{1}{20}$	3	5	-35	70	-50	5	5	$-\frac{1}{48} h^6 f^{(6)}$	
4	6	-32	72	-96	50		$\frac{1}{5}$	4	-5	35	-110	170	-125	35	$\frac{1}{48}$	
Six Point (m=5)								FOURTH DERIVATIVE (k=4)								
0	-274	600	-600	400	-150	24	$-\frac{1}{6}$	0	1	-4	6	-4	1		$-\frac{1}{12} h^5 f^{(5)}$	
1	-24	-130	240	-120	40	-6	$\frac{1}{30}$	1	1	-4	6	-4	1		$-\frac{1}{24} h^5 f^{(5)}$	
2	6	-60	-40	120	-30	4	$-\frac{1}{60} h^6 f^{(6)}$	2	1	-4	6	-4	1		$-\frac{1}{144} h^6 f^{(6)}$	
3	-4	30	-120	40	60	-6	$\frac{1}{60} h^6 f^{(6)}$	3	1	-4	6	-4	1		$\frac{1}{24} h^5 f^{(5)}$	
4	6	-40	120	-240	130	24	$-\frac{1}{30}$	4	1	-4	6	-4	1		$\frac{1}{12} h^5 f^{(5)}$	
5	-24	150	-400	600	-600	274	$\frac{1}{6}$	5	-35	205	-490	590	-355	85	$\frac{5}{16}$	
SECOND DERIVATIVE (k=2)								Fifth Point (m=4)								
* j	A ₀	A ₁	A ₂	A ₃	A ₄	A ₅	$\frac{h^k}{k!}$ Error	j	A ₀	A ₁	A ₂	A ₃	A ₄	A ₅	$\frac{h^k}{k!}$ Error	*
Three Point (m=2)								Six Point (m=5)								
0	1	-2	1				$-\frac{1}{2} h^3 f^{(3)}$	0	15	-70	130	-120	55	-10	$\frac{17}{144}$	
1	1	-2	1				$-\frac{1}{24} h^4 f^{(4)}$	1	10	-45	80	-70	30	-5	$\frac{5}{144}$	
2	1	-2	1				$\frac{1}{2} h^3 f^{(3)}$	2	5	-20	30	-20	5	0	$-\frac{1}{144} h^6 f^{(6)}$	
Four Point (m=3)								Fifth Point (m=4)								
0	6	-15	12	-3			$\frac{11}{24}$	3	0	5	-20	30	-20	5	$-\frac{1}{144} h^6 f^{(6)}$	
1	3	-6	3	0			$-\frac{1}{24} h^4 f^{(4)}$	4	-5	30	-70	80	-45	10	$\frac{5}{144}$	
2	0	3	-6	3			$\frac{11}{24}$	5	-10	55	-120	130	-70	15	$\frac{17}{144}$	
3	-3	12	-15	6												
Five Point (m=4)								FIFTH DERIVATIVE (k=5)								
0	35	-104	114	-56	11		$-\frac{5}{12} h^5 f^{(5)}$	j <th>A₀</th> <th>A₁</th> <th>A₂</th> <th>A₃</th> <th>A₄</th> <th>A₅</th> <th>$\frac{h^k}{k!}$ Error</th> <th>*</th>	A ₀	A ₁	A ₂	A ₃	A ₄	A ₅	$\frac{h^k}{k!}$ Error	*
1	11	-20	6	4	-1		$\frac{1}{24}$	Six Point (m=5)								
2	-1	16	-30	16	-1		$\frac{1}{180} h^6 f^{(6)}$	0	-1	5	-10	10	-5	1	$-\frac{1}{48}$	
3	-1	4	6	-20	11		$-\frac{1}{24} h^5 f^{(5)}$	1	-1	5	-10	10	-5	1	$-\frac{1}{80}$	
4	11	-56	114	-104	35		$\frac{5}{12}$	2	-1	5	-10	10	-5	1	$-\frac{1}{240} h^6 f^{(6)}$	
Six Point (m=5)								3	-1	5	-10	10	-5	1	$\frac{1}{80}$	
0	225	-770	1070	-780	305	-50	$\frac{137}{360}$	4	-1	5	-10	10	-5	1	$\frac{1}{48}$	
1	50	-75	-20	70	-30	5	$-\frac{13}{360}$	5	-1	5	-10	10	-5	1		
2	-5	80	-150	80	-5	0	$\frac{1}{180}$									
3	0	-5	80	-150	80	-5	$\frac{1}{180} h^6 f^{(6)}$									
4	5	-30	70	-20	-75	50	$-\frac{13}{360}$									
5	-50	305	-780	1070	-770	225	$\frac{137}{360}$									

Compiled from W. G. Bickley, Formulae for numerical differentiation, Math. Gaz. 25, 19-27, 1941 (with permission).

*See page 11.

Fifth Derivative, central difference, 7-point

$-1 \quad 4 \quad -5 \quad 0 \quad 5 \quad -4 \quad 1$
 $2h^5$
 $\frac{h^2}{3} (v)$