From Taylor to Cramer to Vandermonde to Schur: A Closed Form Solution for Determining the Weights of Finite Difference Formulae for Derivatives on Random, One-Dimensional Grids

Vincent van Joolen *
Department of Mathematics
United States Naval Academy
121 Blake Road
Annapolis, MD 21402-5000, U.S.A.

May 25, 2007

*E-mail: vanjoole@usna.edu , Tel.: +1-410-293-6776, Fax: +1-410-293-4883.

Abstract

As nano-engineering emerges as a viable technology, it will soon be possible to scatter thousands, if not millions, of tiny sensors over a multi-dimensional region. If the purpose of these sensors is to determine rate-of-change data, there must be a capability to approximate derivatives using data from random sensor locations. This paper proposes a powerful, but simple algorithm to generate a finite difference formulae to approximate derivatives using a random, asymmetric, one-dimensional grid schemes. In a classic approach, Taylor series polynomials are used to obtain weights for the approximation formulae. Cramer's rule provides a procedure to solve the resulting system of equations. Patterns emerge that invoke properties of the Vandermonde matrix and Schur polynomials which are used to produce a closed form solution that is easily coded for computational purposes. The resulting equations are well suited for the 'scattered sensor' scenario and could rapidly adjust to changes in the grid resulting from the loss or addition of one or more sensors.

Key Words finite difference formulae, numerical differentiation, Taylor series, Vandermonde matrix, Schur polynomials, random one-dimensional grids..

1 Introduction

Numerical differentiation is useful for approximating rates of change for data where a function is unknown. The Taylor series provides a vehicle for generating finite difference formulae for derivatives. A sampling of such formulae on uniform one-dimensional grids is provided in CRC's Standard Mathematical Tables and Formulae [14]. Beyond these, pencil and paper derivations are possible. Though conceptually simple, this approach is time consuming. Several authors have developed algorithms for use in concert with computers that solve for such approximations using various uniform and non-uniform grid schemes. In an earlier work Keller and Pereyara present weight tables for derivatives of high order and accuracy on uniform grids [5]. Latter Fornberg noted isolated, systematic error's in this work and prescribed a scheme that utilizes simple recursion relations in which he reproduces corrected weight tables, again on uniform grids [3]. In both papers examples of one-sided, centered, and "half-way point" schemes are utilized. Fornberg advances his algorithms to non-uniform grids in [4]. Khan and Ohba present closed form expressions for finite difference approximations for use in digital differentiators [6, 7, 8, 9]. In these papers, data tables are produced that show derivative approximations for sample digital inputs, however, Khan and Ohba did not derive formulae to determine finite difference weights. None of the above mentioned papers provided error estimates for the approximations.

In this paper, a closed form expression is derived that can be used to approximate any order derivative to any order accuracy on uniform, non-uniform, or random one-dimensional grids. The resulting equation produces Taylor series weights as well as error estimates and works equally well for forward, backward, centered, and non-centered schemes. Approximation formulae can be obtained for random grid schemes with out additional computational complexity. In Sections 1 through 4, a four-point random grid is used to derive approximation formulae for the first, second and third derivatives. Sections 5 and 6 provide examples which use the prescribed procedure to generate approximations on a uniform and random grids respectively. In Section 7, the scheme is generalized for any number of grid points and to approximate derivatives of any order. Sections 8 through 10 provide further examples, using the general equation to determine derivative approximations on non-centered, half-point, and a priori grid schemes. Section 11 provides recommendations for future research with regards to random grid scenarios.

2 Finite Difference Formulae on Randomly Spaced One-Dimensional Grids up through the 3^{rd} Derivative

The derivation starts with a general four-term Taylor series expansion (with error term):

$$f_a = \frac{(ah)^0}{0!} f_{_0} + \frac{(ah)^1}{1!} f_{_0}^{^{(1)}} + \frac{(ah)^2}{2!} f_{_0}^{^{(2)}} + \frac{(ah)^3}{3!} f_{_0}^{^{(3)}} + \frac{(ah)^4}{4!} f^{^{(4)}}(\xi). \quad (1)$$

Here a is a real number that represents the relative position from the point of interest x_0 on a one-dimensional grid and h is the grid spacing. Since a randomly spaced one-dimensional grid is to be employed, we let $h = 1$. All information with regards to spacing be contained in a. Using this convention, equation (1) simplifies to:

$$f_a = \frac{a^0}{0!} f_{_0} + \frac{a^1}{1!} f_{_0}^{^{(1)}} + \frac{a^2}{2!} f_{_0}^{^{(2)}} + \frac{a^3}{3!} f_{_0}^{^{(3)}} + \frac{a^4}{4!} f^{^{(4)}}(\xi). \quad (2)$$

Note that if $a = 0$ and if we define $0^0 = 1$ then (2) correctly becomes $f_0 = f_0$.

Using (2) we now construct a set of four equations for four random grid points a_1 through a_4:

$$
\begin{aligned}
f_{a_1} &= \frac{a_1^0}{0!} f_{0} &+ \frac{a_1^1}{1!} f_{0}^{(1)} &+ \frac{a_1^2}{2!} f_{0}^{(2)} &+ \frac{a_1^3}{3!} f_{0}^{(3)} &+ \frac{a_1^4}{4!} f^{(4)}(\xi), \\
f_{a_2} &= \frac{a_2^0}{0!} f_{0} &+ \frac{a_2^1}{1!} f_{0}^{(1)} &+ \frac{a_2^2}{2!} f_{0}^{(2)} &+ \frac{a_2^3}{3!} f_{0}^{(3)} &+ \frac{a_2^4}{4!} f^{(4)}(\xi), \\
f_{a_3} &= \frac{a_3^0}{0!} f_{0} &+ \frac{a_3^1}{1!} f_{0}^{(1)} &+ \frac{a_3^2}{2!} f_{0}^{(2)} &+ \frac{a_3^3}{3!} f_{0}^{(3)} &+ \frac{a_3^4}{4!} f^{(4)}(\xi), \\
f_{a_4} &= \frac{a_4^0}{0!} f_{0} &+ \frac{a_4^1}{1!} f_{0}^{(1)} &+ \frac{a_4^2}{2!} f_{0}^{(2)} &+ \frac{a_4^3}{3!} f_{0}^{(3)} &+ \frac{a_4^4}{4!} f^{(4)}(\xi).
\end{aligned}
\tag{3}
$$

In the usual manner, obtain an approximation for a derivative by multiplying each line in (3) by a respective weight Ω_{a_i} and summing. This generates a new set of equations (remainder terms are excluded for the time being):

$$
\begin{aligned}
\Omega_{a_1} a_1^0 + \Omega_{a_2} a_2^0 + \Omega_{a_3} a_3^0 + \Omega_{a_4} a_4^0 &= 0!\delta_{0d}, \\
\Omega_{a_1} a_2^1 + \Omega_{a_2} a_2^1 + \Omega_{a_3} a_3^1 + \Omega_{a_4} a_4^1 &= 1!\delta_{1d}, \\
\Omega_{a_1} a_3^2 + \Omega_{a_2} a_2^2 + \Omega_{a_3} a_3^2 + \Omega_{a_4} a_4^2 &= 2!\delta_{2d}, \\
\Omega_{a_1} a_1^3 + \Omega_{a_2} a_2^3 + \Omega_{a_3} a_3^3 + \Omega_{a_4} a_4^3 &= 3!\delta_{3d}.
\end{aligned}
\tag{4}
$$

Here d is the order of the derivative that is to be approximated and δ_{ik} is the Kronecker delta. Equation (4) can be rewritten in matrix form:

$$
\begin{bmatrix}
a_1^0 & a_2^0 & a_3^0 & a_4^0 \\
a_1^1 & a_2^1 & a_3^1 & a_4^1 \\
a_1^2 & a_2^2 & a_3^2 & a_4^2 \\
a_1^3 & a_2^3 & a_3^3 & a_4^3
\end{bmatrix}
\begin{bmatrix}
\Omega_{a_1} \\ \Omega_{a_2} \\ \Omega_{a_3} \\ \Omega_{a_4}
\end{bmatrix}
=
\begin{bmatrix}
d!\delta_{0d} \\ d!\delta_{1d} \\ d!\delta_{2d} \\ d!\delta_{3d}
\end{bmatrix}.
\tag{5}
$$

Using weights calculated in (5), the combined remainder is given by:

$$
R \leq \begin{bmatrix} \Omega_{a_1} & \Omega_{a_2} & \Omega_{a_3} & \Omega_{a_4} \end{bmatrix}
\begin{bmatrix}
a_1^4 \\ a_2^4 \\ a_3^4 \\ a_4^4
\end{bmatrix}
\frac{f^{(4)}(\xi)}{4!},
\tag{6}
$$

where ξ can be any value within the range of the grid points. If $R = 0$, the next term in the Taylor series must be used to obtain the remainder:

$$R \leq \begin{bmatrix} \Omega_{a_1} & \Omega_{a_2} & \Omega_{a_3} & \Omega_{a_4} \end{bmatrix} \begin{bmatrix} a_1^5 \\ a_2^5 \\ a_3^5 \\ a_4^5 \end{bmatrix} \frac{f^{(5)}(\xi)}{5!}. \tag{7}$$

The linear system in (5) is well structured and is easily extended to incorporate additional grid points to determine approximation formulae for higher order derivatives or to provide greater accuracy. A method for solution follows.

3 A Proposed Solution: From Taylor to Cramer to Vandermonde to Schur

We now use the system proposed in (5) to solve for Ω_{a_1} in an equation approximating the first derivative. Applying Cramer's rule yields:

$$\Omega_{a_1} = \frac{\begin{vmatrix} 0 & 1 & 1 & 1 \\ 1 & a_2 & a_3 & a_4 \\ 0 & a_2^2 & a_3^2 & a_4^2 \\ 0 & a_2^3 & a_3^3 & a_4^3 \end{vmatrix}}{\begin{vmatrix} 1 & 1 & 1 & 1 \\ a_1 & a_2 & a_3 & a_4 \\ a_1^2 & a_2^2 & a_3^2 & a_4^2 \\ a_1^3 & a_2^3 & a_3^3 & a_4^3 \end{vmatrix}}. \tag{8}$$

The denominator of (8) is the determinant of a Vandermonde matrix $\mathbf{V}$ which is known to be [1, 11]:

$$\prod_{1 \leq i < j \leq 4} (a_j - a_i). \tag{9}$$

In this example:

$$det(\mathbf{V}) = (a_4 - a_3)(a_4 - a_2)(a_4 - a_1)(a_3 - a_2)(a_3 - a_1)(a_2 - a_1). \tag{10}$$

Using cofactor expansion, the numerator reduces to:

$$-1 \begin{vmatrix} 1 & 1 & 1 \\ a_2^2 & a_3^2 & a_4^2 \\ a_2^3 & a_3^3 & a_4^3 \end{vmatrix} \tag{11}$$

This step generates an initial minor $\mathbf{M}_{(R,C)}$ of the original Vandermonde matrix $\mathbf{V}$ where R and C are the numerical values of the row and column removed from the matrix. Note that the $R = d + 1$ and C is tied to the index w of the weight being calculated (i.e. $w = 1$ for the first weight Ω_{a_1}, etc.). In this case we are calculating Ω_{a_1} for the first derivative. Therefore $R = 2$ and $C = 1$. In [2, 10, 12] it was shown that:

$$\det\left(\mathbf{M}_{(R,C)}\right) = s_\lambda(a_1, ..., a_{n\neq C}, ..., a_n)\det(\overline{\mathbf{V}}). \tag{12}$$

Here $\overline{\mathbf{V}}$ is a Vandermonde matrix the a_C, or in this case a_1, term removed. The expression s_λ is a Schur polynomial where $\lambda = r - d - 1$ and where r is the number of grid points use in the approximation. A Schur polynomial in n variables is expressed as follows:

$$\begin{aligned}
s_0(a_1, ..., a_n) &= 1, \\
s_1(a_1, ..., a_n) &= \sum_{i=1}^{n} a_i, \\
s_2(a_1, ..., a_n) &= \sum_{1\leq i<j\leq n} a_i a_j, \\
s_3(a_1, ..., a_n) &= \sum_{1\leq i<j<k\leq n} a_i a_j a_k, \\
\vdots \quad & \quad \vdots \quad \quad \vdots \\
s_n(a_1, ..., a_n) &= \prod_{i=1}^{n} a_i.
\end{aligned} \tag{13}$$

In this example $\lambda = 2$ and therefore,

$$s_2(a_2, a_3, a_4) = a_4a_3 + a_4a_2 + a_3a_2. \tag{14}$$

Additionally:

$$det(\overline{\mathbf{V}}) = (a_4 - a_3)(a_4 - a_2)(a_3 - a_2), \tag{15}$$

i.e. the determinant of the Vandermonde matrix with the a_1 term removed. Using the results from (10), (14), and (15) produces:

$$\Omega_{a_1} = \frac{a_4a_3 + a_4a_2 + a_3a_2}{(a_1 - a_4)(a_1 - a_3)(a_1 - a_2)}. \tag{16}$$

In a similar manner we derive an expression for the remaining weights:

$$\begin{aligned}
\Omega_{a_2} &= \frac{a_4a_3 + a_4a_1 + a_3a_1}{(a_2 - a_4)(a_2 - a_3)(a_2 - a_1)}, \\
\Omega_{a_3} &= \frac{a_4a_2 + a_4a_1 + a_2a_1}{(a_3 - a_4)(a_3 - a_2)(a_3 - a_1)}, \\
\Omega_{a_4} &= \frac{a_3a_2 + a_3a_1 + a_2a_1}{(a_4 - a_3)(a_4 - a_2)(a_4 - a_1)}.
\end{aligned} \tag{17}$$

These equations work equally well for uniform, non-uniform grids, or randomly generated grids. The procedure presented in this section is next used to determine weight equations for the second and third derivatives.

4 Expressions for a General Four-Point Approximation of the Second and Third Derivatives

Again, apply Cramer's rule to (5) to determine the first weight of the second derivative:

$$\Omega_{a_1} = \frac{\begin{vmatrix} 0 & 1 & 1 & 1 \\ 0 & a_2 & a_3 & a_4 \\ 2! & a_2^2 & a_3^2 & a_4^2 \\ 0 & a_2^3 & a_3^3 & a_4^3 \end{vmatrix}}{\begin{vmatrix} 1 & 1 & 1 & 1 \\ a_1 & a_2 & a_3 & a_4 \\ a_1^2 & a_2^2 & a_3^2 & a_4^2 \\ a_1^3 & a_2^3 & a_3^3 & a_4^3 \end{vmatrix}}. \tag{18}$$

Using cofactor expansion, the numerator equals:

$$2\begin{vmatrix} 1 & 1 & 1 \\ a_2 & a_3 & a_4 \\ a_2^3 & a_3^3 & a_4^3 \end{vmatrix}. \tag{19}$$

From (12) we have an expression of the determinant in (19):

$$s_1(a_2, a_3, a_4)\begin{vmatrix} 1 & 1 & 1 \\ a_2 & a_3 & a_4 \\ a_2^2 & a_3^2 & a_4^2 \end{vmatrix}. \tag{20}$$

Using (9) and (13) the first weight for the four-point approximation of the second derivative is:

$$\Omega_{a_1} = \frac{2(a_4 + a_3 + a_2)}{(a_4 - a_1)(a_3 - a_1)(a_2 - a_1)}. \tag{21}$$

In similar fashion the remaining weights are:

$$\begin{aligned} \Omega_{a_2} &= \frac{2(a_4 + a_3 + a_1)}{(a_4 - a_2)(a_3 - a_2)(a_1 - a_2)}, \\ \Omega_{a_3} &= \frac{2(a_4 + a_2 + a_1)}{(a_4 - a_3)(a_2 - a_3)(a_1 - a_3)}, \\ \Omega_{a_4} &= \frac{2(a_3 + a_2 + a_1)}{(a_3 - a_4)(a_2 - a_4)(a_1 - a_4)}. \end{aligned} \tag{22}$$

This example is completed by stating the easily obtained results for the weights for the four-point third derivative approximation:

$$\begin{aligned}
\Omega_{a_1} &= \frac{6}{(a_1 - a_4)(a_1 - a_3)(a_1 - a_2)}, \\
\Omega_{a_2} &= \frac{6}{(a_2 - a_4)(a_2 - a_3)(a_2 - a_1)}, \\
\Omega_{a_3} &= \frac{6}{(a_3 - a_4)(a_3 - a_2)(a_3 - a_1)}, \\
\Omega_{a_4} &= \frac{6}{(a_4 - a_3)(a_4 - a_2)(a_4 - a_1)}.
\end{aligned} \tag{23}$$

Numerical examples of these results are provided in the next section two sections.

5 A Classic Result: Four-Point Central-Difference Approximations for Derivatives on a Uniform Grid

In this example let $\{a_1, a_2, a_3, a_4\} = \{-2, -1, 1, 2\}$. Using (16) and (17) we have:

$$\begin{aligned}
\Omega_{a_1} &= \frac{(2)(1) + (2)(-1) + (1)(-1)}{(-2-2)(-2-1)(-2+1)} = \frac{1}{12}, \\
\Omega_{a_2} &= \frac{(2)(1) + (2)(-2) + (1)(-2)}{(-1-2)(-1-1)(-1+2)} = -\frac{2}{3}, \\
\Omega_{a_3} &= \frac{(2)(-1) + (2)(-2) + (-1)(-2)}{(1-2)(1+1)(1+2)} = \frac{2}{3}, \\
\Omega_{a_4} &= \frac{(1)(-1) + (1)(-2) + (-1)(-2)}{(2-1)(2+1)(2+2)} = -\frac{1}{12}.
\end{aligned} \tag{24}$$

When (6) is used to determine the error term, the result is $R = 0$, so we default to (7) which generates:

$$R = \begin{bmatrix} \frac{1}{12} & -\frac{2}{3} & \frac{2}{3} & -\frac{1}{12} \end{bmatrix} \begin{bmatrix} -32 \\ \\ -1 \\ \\ 1 \\ \\ 32 \end{bmatrix} \frac{f^{(5)}(\xi)}{5!} = -\frac{1}{30} f^{(5)}(\xi). \tag{25}$$

Thus the four-point approximation of the first derivative for this grid scheme is:

$$f^{'}(x_o) = \frac{1}{12} f_{-2} - \frac{2}{3} f_{-1} + \frac{2}{3} f_1 - \frac{1}{12} f_2 + \frac{1}{30} f^{(5)}(\xi), \tag{26}$$

which can be rewritten in the form found in [14]:

$$f^{'}(x_o) = \frac{1}{12}(f_{-2} - 8f_{-1} + 8f_1 - f_2) + \frac{1}{30} f^{(5)}(\xi). \tag{27}$$

Other approximation formulae that can be generated from this grid scheme are obtained using (21) through (23) are:

$$\begin{aligned} f^{''}(x_o) &= \frac{1}{3}(f_{-2} - f_{-1} - f_1 + f_2) - \frac{5}{12} f^{(4)}(\xi). \\ f^{'''}(x_o) &= \frac{1}{2}(-f_{-2} + 2f_{-1} - 2f_1 + f_2) - \frac{1}{4} f^{(5)}(\xi). \end{aligned} \tag{28}$$

6 A Not-so-Classic Result: Four-Point Approximations for Derivatives on a Random Asymmetric Grid

This section demonstrates the versatility the procedure by extending it to non-uniform, random, asymmetric grids. Consider the grid-points $\{a_1, a_2, a_3, a_4\} = \{-.149, .051, .323, .410\}$. The equations in the previous sections are so simply applied that the approximation formulae for the first three derivatives here stated without derivation:

$$\begin{aligned} f_0^{'} &= -3.22 f_{a_1} + 1.89 f_{a_2} + 4.28 f_{a_3} - 2.25 f_{a_4} + 7.73 * 10^{-4} f^{(4)}(\xi), \\ f_0^{''} &= 29.7 f_{a_1} - 59.8 f_{a_2} + 55.9 f_{a_3} - 25.8 f_{a_4} + 4.42 * 10^{-3} f^{(4)}(\xi), \\ f_0^{'''} &= -114 f_{a_1} + 307 f_{a_2} - 537 f_{a_3} + 344 f_{a_4} - .159 f^{(4)}(\xi). \end{aligned} \tag{29}$$

Here $f_0^{(i)} = f^{(i)}(x_0)$ and $f_{a_n} = f(x_0 + a_n)$. To provide a numerical example, the first three derivatives at $x = 1$ are approximated for:

$$f(x) = ln\left(\frac{1}{1+x^2}\right). \tag{30}$$

The results are summarized in the table below:

Table 1
Numerical Example
Approximation of 1st 3 Derivatives for $ln\left(\frac{1}{1+x^2}\right)$
(Evaluated at $x = 1$)

Grid Points (Relative): {-.149, .051, .323, .410}
Grid Points (Absolute): {.851, 1.051, 1.323, 1.410}

	Approximation	Actual	|Difference|	|Remainder|
$f^{(1)}(1)$	-.998	-1.00	.00181	.00296
$f^{(2)}(1)$	.00838	0.00	.00838	.0169
$f^{(3)}(1)$	.623	1.00	.377	.608

In this table, we see that the difference between actual and approximated values falls within the bounds of the Taylor series remainder. Greater accuracy or higher order derivatives are possible by increasing the number of grid points. The following section generalizes the algorithm for any number of points.

7 A General Case

Consider a general case for a non-uniform grid. We obtain an approximation $f^{(d)}(x_0)$ using r grid points designated by the integers a_1 through a_r. The necessary Taylor series expressions (without the remainder terms) are:

$$\begin{array}{cccccccc}
f_{a_1} & \approx & \frac{a_1^0}{0!}f_0 & + \ldots + & \frac{a_1^d}{d!}f_0^{(i)} & + \ldots + & \frac{a_1^{(r-1)}}{(r-1)!}f_0^{(r-1)} \\
\vdots & & \vdots & \ddots & \vdots & \ddots & \vdots \\
f_{a_d} & \approx & \frac{a_d^0}{0!}f_0 & + \ldots + & \frac{a_d^d}{d!}f_0^{(i)} & + \ldots + & \frac{a_d^{(r-1)}}{(r-1)!}f_0^{(r-1)} \\
\vdots & & \vdots & \ddots & \vdots & \ddots & \vdots \\
f_{a_r} & \approx & \frac{a_r^0}{0!}f_0 & + \ldots + & \frac{a_r^d}{d!}f_0^{(i)} & + \ldots + & \frac{a_r^{(r-1)}}{(r-1)!}f_0^{(r-1)}
\end{array} \tag{31}$$

After multiplying each expression in (31) by the respective weight Ω_{a_n} and summing components, the following result is generated:

$$\begin{bmatrix} a_1^0 & \dots & a_d^0 & \dots & a_r^0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_1^d & \dots & a_d^d & \dots & a_r^d \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_1^{r-1} & \dots & a_d^{r-1} & \dots & a_r^{r-1} \end{bmatrix} \begin{bmatrix} \Omega_{a_1} \\ \vdots \\ \Omega_{a_d} \\ \vdots \\ \Omega_{a_r} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ d! \\ \vdots \\ 0 \end{bmatrix}, \qquad (32)$$

Applying Cramer's rule to solve for Ω_{a_1} in (32) yields:

$$\Omega_{a_1} = \frac{\begin{vmatrix} 0 & \dots & a_d^0 & \dots & a_r^0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ d! & \dots & a_d^d & \dots & a_r^i \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & a_d^{r-1} & \dots & a_r^{r-1} \end{vmatrix}}{\begin{vmatrix} a_1^0 & \dots & a_d^0 & \dots & a_r^0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_1^i & \dots & a_d^d & \dots & a_r^i \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_1^{r-1} & \dots & a_d^{r-1} & \dots & a_r^{r-1} \end{vmatrix}}. \qquad (33)$$

The denominator of (33) is:

$$\prod_{1 \le i < j \le r} (a_j - a_i), \qquad (34)$$

where $\{a_i\} = \{a_1, a_2, a_3, ..., a_r\}$. Using cofactor expansion, the numerator reduces to:

$$(-1)^{(C+R)} d! \begin{vmatrix} a_2^0 & \dots & a_d^0 & \dots & a_r^0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_2^{d-1} & \dots & a_d^{d-1} & \dots & a_r^{d-1} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_2^{r-1} & \dots & a_d^{r-1} & \dots & a_r^{r-1} \end{vmatrix}, \qquad (35)$$

where $R = d + 1$, $C = w$, and $\lambda = r - d - 1$. From (12) we know that the determinant of this matrix is:

$$\det\left(\mathbf{M}_{(d+1,w)}\right) = s_{(r-d-1)}(a_1, ..., a_{n \neq w}, ..., a_r) \prod_{\substack{1 \le i < j \le r \\ i,j \neq w}} (a_j - a_i). \qquad (36)$$

Using (34) through (36) generates:

$$\Omega_w = (-1)^{(d+w+1)} d! \left(\frac{\prod\limits_{\substack{1 \leq i < j \leq r \\ i,j \neq w}} (a_j - a_i)}{\prod\limits_{1 \leq i < j \leq r} (a_j - a_i)} \right) s_{(r-d-1)}(a_1, ..., a_{n \neq w}, ..., a_r). \tag{37}$$

After cancelation, a closed form expression for the finite difference weight results:

$$\Omega_w = (-1)^{(d+w+1)} d! \left(\frac{s_{(r-d-1)}(a_1, ..., a_{n \neq w}, ..., a_r)}{\prod\limits_{1 \leq i \neq w \leq r} (a_w - a_i)} \right). \tag{38}$$

Note that if a uniform grid is used in the derivation with grid spacing h where $f_{a_i} = f(x_0 + a_i h)$ equation (38) becomes:

$$\Omega_w = (-1)^{(d+w+1)} \frac{d!}{h^d} \left(\frac{s_{(r-d-1)}(a_1, ..., a_{n \neq w}, ..., a_r)}{\prod\limits_{1 \leq i \neq w \leq r} (a_w - a_i)} \right). \tag{39}$$

In this equation, the values for a_i are not required to be sequential integers, so the grid my still have a random character. In the following examples we use (39) to generate additional "not-so-classical" results.

8 Numerical Example: An Asymmetric Grid with Forward and Backward Elements

Consider a-six term approximation for the 4^{th} derivative with four backward grid points and one forward grid point. The following system is set up using (32):

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ -4 & -3 & -2 & -1 & 0 & 1 \\ (-4)^2 & (-3)^2 & (-2)^2 & (-1)^2 & 0^2 & 1^2 \\ (-4)^3 & (-3)^3 & (-2)^3 & (-1)^3 & 0^3 & 1^3 \\ (-4)^4 & (-3)^4 & (-2)^4 & (-1)^4 & 0^4 & 1^4 \\ (-4)^5 & (-3)^5 & (-2)^5 & (-1)^5 & 0^5 & 1^5 \end{bmatrix} \begin{bmatrix} \Omega_{-4} \\ \Omega_{-3} \\ \Omega_{-2} \\ \Omega_{-1} \\ \Omega_0 \\ \Omega_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{4!}{h^4} \\ 0 \end{bmatrix}. \quad (40)$$

Six grid points are utilized ($r = 6$) to determine the 4^{th} derivative ($d = 4$), therefore $\lambda = r - d - 1 = 1$. The grid point vector is given by $a = [-4, -3, -2, -1, 0, 1]$. We solve for 3^{rd} weight ($w = 3$) corresponding to the $a_3 = -2$ grid point. The mesh size is h. Substituting these values in (39) gives:

$$\Omega_3 = (-1)^{(3+4+1)} \frac{4!}{h^4} \left(\frac{s_1(-4, -3, -1, 0, 1)}{\prod_{1 \leq i \neq 3 \leq r} (-2 - a_i)} \right) = -\frac{14}{h^4}. \quad (41)$$

The other weights are obtained in a similar fashion and the error term is calculated as before resulting in:

$$f_0^{(4)} \approx \frac{1}{h^4}(-f_{-4} + 6f_{-3} - 14f_{-2} + 16f_{-1} - 9f_0 + 2f_1) + \frac{5h^2}{6} f^{(6)}(\xi). \quad (42)$$

The grid points in this example can also be used to calculate weights for up through the 5^{th} derivative. A summary of weights and remainders is presented in Table 2.

$d\backslash w$	-4	-3	-2	-1	0	1	$\times$	Remainder
1	$\big(\frac{1}{20}$	$-\frac{1}{3}$	1	-2	$\frac{13}{12}$	$\frac{1}{5}\big)$	$\frac{1}{h}$	$\frac{1}{30}h^5f^{(6)}(\xi)$
2	$\big(\frac{1}{12}$	$-\frac{1}{2}$	$\frac{7}{6}$	$-\frac{1}{3}$	$-\frac{5}{4}$	$\frac{5}{6}\big)$	$\frac{1}{h^2}$	$\frac{13}{180}h^4f^{(6)}(\xi)$
3	$\big(-\frac{1}{4}$	$\frac{7}{4}$	$-\frac{11}{2}$	$\frac{17}{2}$	$-\frac{25}{4}$	$\frac{7}{4}\big)$	$\frac{1}{h^3}$	$-\frac{1}{8}h^3f^{(6)}(\xi)$
4	$(-1$	6	-14	16	-9	2)	$\frac{1}{h^4}$	$-\frac{5}{6}h^2f^{(6)}(\xi)$
5	$(-1$	5	-10	10	-5	1)	$\frac{1}{h^5}$	$-\frac{3}{2}hf^{(6)}(\xi)$

Table 2
Summary of Weights for First Five Derivatives using a Non-Central Grid Scheme

9 Numerical Example - Approximations Using Midpoints of Uniform Grids

Here we use the midpoints of a uniform grid. Consider the grid points $\{-3h, -2h, -1h, 0, 1h, 2h, 3h\}$. We derive an approximation for the first derivative at $\{-\frac{5}{2}h, -\frac{3}{2}h, -\frac{1}{2}h, \frac{1}{2}h, \frac{3}{2}h, \frac{5}{2}h\}$, i.e. $a = \{-\frac{5}{2}, -\frac{3}{2}, -\frac{1}{2}, \frac{1}{2}, \frac{3}{2}, \frac{5}{2}\}$, $r = 6$, $d = 1$ and $\lambda = 4$. Using (38) ω_1 is calculated:

$$\Omega_1 = -\frac{1}{h}\left(\frac{s_4\left(-\frac{3}{2}, -\frac{1}{2}, \frac{1}{2}, \frac{3}{2}, \frac{5}{2}\right)}{\prod\limits_{1<i\leq 6}\left(a_i + \frac{5}{2}\right)}\right) = -\frac{3}{640h}. \tag{43}$$

The remaining weights and errors are summarized in Table 3 which includes data for approximations up through the 5^{th} derivative. Note that the 0^{th} order derivative in this example is equivalent to interpolation.

Table 3 Summary of Weights for First Five Derivatives Using Midpoints of Uniform Grids								
$d \backslash w$	$-\frac{5}{2}$	$-\frac{3}{2}$	$-\frac{1}{2}$	$\frac{1}{2}$	$\frac{3}{2}$	$\frac{5}{2}$	$\times$	Remainder
0	$\big(\frac{3}{256}$	$-\frac{25}{256}$	$\frac{75}{128}$	$\frac{75}{128}$	$-\frac{25}{256}$	$\frac{3}{256}\big)$	1	$\frac{5}{1024}h^6 f^{(6)}(\xi)$
1	$\big(-\frac{3}{640}$	$\frac{25}{384}$	$-\frac{75}{64}$	$\frac{75}{64}$	$-\frac{25}{384}$	$\frac{3}{640}\big)$	$\frac{1}{h}$	$\frac{5}{7168}h^6 f^{(7)}(\xi)$
2	$\big(-\frac{5}{48}$	$\frac{13}{16}$	$-\frac{17}{24}$	$-\frac{17}{24}$	$-\frac{13}{16}$	$\frac{-5}{48}\big)$	$\frac{1}{h^2}$	$-\frac{259}{5760}h^4 f^{(6)}(\xi)$
3	$\big(\frac{1}{8}$	$-\frac{13}{8}$	$\frac{17}{4}$	$-\frac{17}{4}$	$\frac{13}{8}$	$-\frac{1}{8}\big)$	$\frac{1}{h^3}$	$-\frac{37}{1920}h^4 f^{(7)}(\xi)$
4	$\big(\frac{1}{2}$	$-\frac{3}{2}$	1	1	$-\frac{3}{2}$	$\frac{1}{2}\big)$	$\frac{1}{h^4}$	$\frac{7}{24}h^2 f^{(6)}(\xi)$
5	$(-1$	5	-10	10	-5	$1)$	$\frac{1}{h^5}$	$\frac{5}{24}h^2 f^{(7)}(\xi)$

10 Numerical Example - A Priori Grid Point Selection

Now we arbitrarily select grid points. Physically, this might occur in a sensor network where a number of sensors have become nonfunctional. For example, consider the 1^{st}, 3^{rd}, and 5^{th} grid points in the backwards direction and the 2^{nd} and 4^{th} grid points in the forward direction. Do not include the central point, i.e. $a = \{-5, -3, -1, 2, 4\}$ and $r = 5$. Use a to approximate $f_0^{(3)}$, i.e. $d = 3$ and $\lambda = 1$. From (39) Ω_1 is calculated:

$$\Omega_1 = -\frac{3!}{h^4}\left(\frac{s_1(-3,-1,2,4)}{\prod\limits_{1<i\leq 5}(a_i+5)}\right) = -\frac{1}{42h^3}. \quad (44)$$

The remaining weights and error term are summarized in Table 4 along with weights/errors for approximations through the 4^{th} derivative:

Table 4 Summary of Weights for a Priori Grid Scheme							
$i\backslash w$	-5	-3	-1	2	4	$\times$	Remainder
0	$\Big(\frac{1}{21}$	$-\frac{2}{7}$	1	$\frac{2}{7}$	$-\frac{1}{21}\Big)$	1	$-h^5 f^{(5)}(\xi)$
1	$\Big(\frac{1}{36}$	$-\frac{9}{70}$	$-\frac{13}{60}$	$\frac{11}{30}$	$-\frac{31}{630}\Big)$	$\frac{1}{h}$	$-\frac{47}{60}h^4 f^{(5)}(\xi)$
2	$\Big(-\frac{13}{252}$	$\frac{23}{70}$	$-\frac{5}{12}$	$\frac{13}{105}$	$\frac{1}{63}\Big)$	$\frac{1}{h^2}$	$\frac{17}{20}h^3 f^{(5)}(\xi)$
3	$\Big(-\frac{1}{42}$	0	$\frac{1}{10}$	$-\frac{1}{7}$	$\frac{1}{15}\Big)$	$\frac{1}{h^3}$	$\frac{23}{20}h^2 f^{(5)}(\xi)$
4	$\Big(\frac{1}{21}$	$-\frac{6}{35}$	$\frac{1}{5}$	$-\frac{4}{35}$	$\frac{4}{105}\Big)$	$\frac{1}{h^4}$	$-\frac{3}{5}h f^{(5)}(\xi)$

11 Future Research

In the context of a scattered sensor scenario, it is highly improbable that a large number of sensors will fall in a linear pattern to allow the construction of a one dimensional grid. However, there is a liklihood that several sensors may fall on approximately the same line. Further investigation is recommended to determine if the data provide from these sensors might by used to construct finite difference approximations. Specifically, how might corrections be applied and how would error terms be affected in such an 'almost' one-dimensional scheme. It is also recommended that this algorithm be extended to two- and three-dimensional grid schemes, but more specifically, to schemes were grid point line ups are not orthogonal to each other.

Acknowledgments This author acknowledges Dr. George Nakos, United States Naval Academy, Annapolis, MD who encouraged him to submit these results for publication. He also acknowledges the support from the United States Naval Academy for providing the time and resources to research this topic.

References

[1] A. Bjorck and V. Pereyra, Solutions of Vandermonde Systems of Equations, *Mathematics of Computation*, **24** (1970) 893–903.

[2] T. Ernst, Generalized Vandermonde determinants, *U.U.D.M Tech Report*, **6** (2000).

[3] B. Fornberg, Generation of Finite Difference Formulas on Arbitrarily Spaced Grid, *Mathematics of Computation*, **51** (1988) 699–706.

[4] B. Fornberg, Calculations of Weights in Finite Difference Formulas, *SIAM Rev.*, **40** (1998) 685–691.

[5] H. B. Keller and V. Pereyra, Symbolic Generation of Finite Difference Formulas, *Mathematics of Computation*, **32** (1978) 955–971.

[6] I. R. Khan and R. Ohba, Closed-form expressions for the finite difference approximations of first and higher derivatives based on Taylor series, *J. Comp. Appl. Math.*, **107** (1999) 179–193.

[7] I. R. Khan and R. Ohba, New finite difference formulas for numerical differentiation, *J. Comp. Appl. Math.*, **126** (2000) 269–276.

[8] I. R. Khan, R. Ohba, and N. Hozumi, Mathematical proof of closed form expressions for finite difference approximations based on Taylor series, *J. Comp. Appl. Math.*, **150** (2003) 303–309.

[9] I. R. Khan and R. Ohba, Taylor series based finite difference approximation of higher-degree derivatives, *J. Comp. Appl. Math.*, **154** (2003) 115–124.

[10] I.G. Macdonald, *Symmetric Functions and Hall Polynomials* , Oxford Mathematical Monographs, Second Edition, 1995.

[11] N. Macon and A. Spitzbart, Inverses of Vandermonde Matrices, *The American Mathematical Monthly*, **65** (1958), 95–100.

[12] S. D. Mara, Polynomials arising in factoring generalized Vandermonde determinants: an algorithm for computing their coefficients, *The Mathematical and Computer Modeling*, **34** (2001) 271–281.

[13] J. H. Mathews, Computer Derivations of Numerical Differential Formulae, *J. of Math. Education in Sci. and Tech.*, **34**(2003) 280–287.

[14] D. Zwillinger, *Standard Mathematical Tables and Formulae*, 30th ed., CRC Press, New York, 1996.