WALIA journal 36: 21-25, 2020
Available online at www.Waliaj.com
ISSN 1026-3861


# Redefining finite difference for derivatives in differential equations by polynomial technique

A. G. Shaikh [1,*], A. H. Sheikh [1], Asif Ali [2]

[1]*Department of Mathematics & Statistics, QUEST Nawabshah Pakistan*
[2]*Department of BS&RS, MUET Jamshoro Pakistan*

**Abstract:** The finite difference approximations are widely used to neutralize derivatives. This effort is made to generalize the concept and to understand the finite difference approximations using polynomials. The finite difference method is a way to obtain numerical solution of differential equation(s). It is working with discrete functions, value of unknown could be computed anywhere from the solution. In numerical solution, one cannot find the solution everywhere, but at prescribed locations, called nodes. The node spacing, in general, could be equal or unequal. The basic philosophy of the finite difference, what would we do is in the governing differential equation, we manipulate the governing differential equation directly. The finite difference approximation by polynomial technique method is used, and to understand what is happening, where this approximation is evaluating the derivative, what information it is using to calculate the function and its derivatives.

**Keywords:** Differential equations; Polynomial technique

## 1. Introduction

The finite difference approximations (Operto et al., 2007) is a tool to solve the differential equation(s) and system of differential equations . Their application to the Helmholtz type differential equations are studied in (A. Bayliss et al., 1983; Sheikh, 2014; Sheikh et al., 2016, 2011).

The basic philosophy of the finite difference, for the governing differential equation, is to manipulate the governing differential equation directly. The most important feature of the finite difference approximations is its simplicity. We can write the difference expressions from the derivatives. The finite difference method is a way of obtaining a numerical solution of differential equation(s) (Saad, 1996; Yserentant, 2005).

For the position of the points, from which the finite difference approximation is calculated, it is important that the points should be more closely spaced to improve accuracy, with more computational cost. The location of the point where the finite difference is being evaluated should be centered as possible for the best accuracy, and one of the disadvantage of finite difference approximation is we cannot generalized finite difference approximations. There are lots of methods for deriving finite difference approximations, but in this paper we are presenting polynomial technique (Bayona et al., 2019). It is simple, it builds on curve fitting polynomials to data. It can easily be implemented in MatLab.

* Corresponding Author.

## 2. Polynomial technique for finite difference approximations

The finite difference approximations imply that we do not need to have a continuous smooth function stored in memory, only the function at discrete points is manipulated (Atangana, 2019). The polynomial technique is powerful tool when employed along with the finite differences (Shaikh et al., 2019). Consider few random points as given in Figure 1, these points need not necessarily be distributed uniformly. Beside these points, where each point has a position $x$ and $y$ in two dimensions, then there will be function $f_1, f_2, f_3 \ldots f_7$, we have these random points. Further suppose that we want estimate function or its derivative at some point.

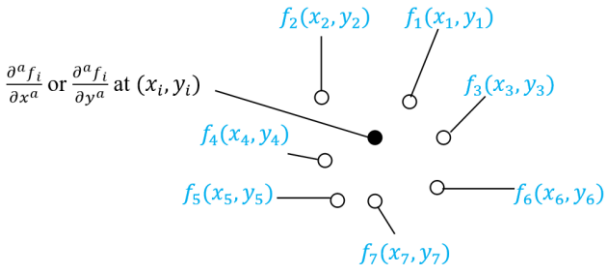


**Fig. 1:** Randomly generated points

We aim to estimate derivative at close circle point. The open circle points are the information, which will be used to find derivatives. Whether we are interpolating the function or any of its derivatives, it is always some weighted sum of the function values at each of those open circle points, so $f_i$ are known

$\frac{\partial^a f_i}{\partial x^a}$or$\frac{\partial^a f_i}{\partial y^a} = a_1 f_1 + a_2 f_2 + a_3 f_3 + a_4 f_4 + a_5 f_5 + a_6 f_6 + a_7 f_7,$

where $a_1, a_2, a_3, a_4, a_5, a_6, and\ a_7$ are called finite difference coefficients. Subsequently, problem of finding derivative reduces to find these co-efficients.

We consider the same distribution of points, but offset them. We will get a same distribution of the points with respect to their relative position. Again if we take the same smattering of open circle points, but we move the closed circle point, then we are evaluating our function or one of its derivatives at a different point, it turns out a different problem of finite difference coefficients.

## 3. Polynomial technique for deriving finite-difference approximations

The curve fitting to polynomials is called the polynomial technique for deriving the finite difference approximations. The general concept is defined as follows. Suppose we have to fit a $N^{th}$ degree polynomial to some set of points, we need $N + 1$ points, such that

$$\begin{aligned} f(x) &= a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \ldots + a_N x^N \\ f'(x) &= a_1 + 2a_2 x + 3a_3 x^2 + 4a_4 x^3 \ldots + N a_N x^{N-1} \\ f''(x) &= 2a_2 + 6a_3 x + +12 a_4 x^2 \ldots + N(N-1) a_N x^{N-2} \\ f'''(x) &= 6a_3 + 24 a_4 x + \ldots + N(N-1)(N-2) a_N x^{N-3} \\ &\vdots \end{aligned}$$

$f(0) = a_0, \quad f'(0) = a_1, \quad f''(0) = 2a_2, \quad f'''(0) = 6a_3 \ldots$

Shifting of coordinates is discussed in next section.

### 3.1. Shifted co-ordinates technique for finite-difference approximations

Suppose we wish to evaluate $f(x)$ or one of its derivatives at the general point $x = x_{fd}$. To do this, we shift our $x$-axis by $x_{fd}$ before fitting the polynomial. An offset will not affect values. Thus, we can rewrite our polynomial in terms of shifted coordinates as:

$$\begin{aligned} f(\tilde{x}_1) &= a_0 + a_1 \tilde{x}_1 + a_2 \tilde{x}_1^2 + a_3 \tilde{x}_1^3 + \ldots + a_N \tilde{x}_1 \\ f(\tilde{x}_2) &= a_0 + a_1 \tilde{x}_2 + a_2 \tilde{x}_2^2 + a_3 \tilde{x}_2^3 + \ldots + a_N \tilde{x}_2 \\ f(\tilde{x}_3) &= a_0 + a_1 \tilde{x}_3 + a_2 \tilde{x}_3^2 + a_3 \tilde{x}_3^3 + \ldots + a_N \tilde{x}_3 \\ &\vdots \\ \tilde{x}_n &= x_n - x_{fd} \\ f(\tilde{x}_{N-1}) &= a_0 + a_1 \tilde{x}_{N-1} + a_2 \tilde{x}_{N-1}^2 \\ &\quad + a_3 \tilde{x}_{N-1}^3 + \ldots + a_N \tilde{x}_{N-1} \end{aligned}$$

where $f(0) = a_0, \quad f'(0) = a_1, \ f''(0) = 2a_2, \ f'''(0) = 6a_3 \ldots$

Let us take the $x$-coordinates of the points from which approximate a derivative, store these $x$-coordinates in a column

$$[x] = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N+1} \end{bmatrix}$$

Shifted the function across the $x$-axis until $\tilde{x} = 0$, corresponds to the point where we wish to approximate the derivative.

$\frac{d^a}{dx^a} f(x = x_{fd}) = \frac{d^a}{dx^a} f(\tilde{x} = 0)$, next is to subtract $x_{fd}$ from the column vector $[x]$ to shift coordinates.

$$[\tilde{x}] = [\, x \,] - x_{fd} = \begin{bmatrix} x_1 - x_{fd} \\ x_2 - x_{fd} \\ \vdots \\ x_{N+1} - x_{fd} \end{bmatrix} = \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \vdots \\ \tilde{x}_{N+1} \end{bmatrix}$$

Using the column vector $[\tilde{x}]$ to build the matrix $[\tilde{X}]$.

$[\tilde{X}] = [\ [\tilde{x}]^0 \quad [\tilde{x}]^1 \ldots \ [\tilde{x}]^N]$

Now we have to big matrix into Vandermonde matrix, where the first column is all ones, the second column are our values of $x$ then our values of $x^2$and all the way up to some value $N$, which is big square matrix given by

$$\begin{bmatrix} 1 & \tilde{x}_1 \ldots & \tilde{x}_1{}^N \\ 1 & \tilde{x}_2 \ldots & \tilde{x}_2{}^N \\ \vdots & \vdots & \vdots \\ 1 & \tilde{x}_{N+1} \ldots \ddots & (\tilde{x}_{N+1})^N \end{bmatrix} = \begin{bmatrix} 1 & x_1 - x_{fd} \ldots & (x_1 - x_{fd})^N \\ 1 & x_2 - x_{fd} \ldots & (x_2 - x_{fd})^N \\ \vdots & \vdots & \vdots \\ 1 & x_{N+1} - x_{fd} \ldots \ddots & (x_{N+1} - x_{fd})^N \end{bmatrix}$$

Once we have the Vandermonde matrix $[\tilde{X}]$ its inverse will be

$$[\tilde{Y}] = [\tilde{X}]^{-1} = \begin{bmatrix} \tilde{y}_{11} & \tilde{y}_{12} & \ldots & \tilde{y}_{1,N+1}{}^N \\ \tilde{y}_{21} & \tilde{y}_{22} & \ldots & \tilde{y}_{2,N+1}{}^N \\ \vdots & & \vdots & \vdots \\ \cdot & & \cdot & \cdot \\ \tilde{y}_{N+1,1} & \tilde{y}_{N+1,2} & \ldots & (\tilde{y}_{N+1\,N+1})^N \end{bmatrix}$$

We have $[\tilde{Y}]$, we have to solve now polynomial equation, calculate the polynomial coefficients

$$[a] = [\tilde{X}]^{-1}[f] = [\tilde{Y}][f]$$

$$\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{N+1} \end{bmatrix} = \begin{bmatrix} \tilde{y}_{11} & \tilde{y}_{12} & \ldots & \tilde{y}_{1,N+1}{}^N \\ \tilde{y}_{21} & \tilde{y}_{22} & \ldots & \tilde{y}_{2,N+1}{}^N \\ \vdots & & \vdots & \vdots \\ \cdot & & \cdot & \cdot \\ \tilde{y}_{N+1,1} & \tilde{y}_{N+1,2} & \ldots & (\tilde{y}_{N+1\,N+1})^N \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_{N+1} \end{bmatrix}$$

$$\begin{aligned} a_0 &= \tilde{y}_{11} f_1 + \tilde{y}_{12} f_2 + \tilde{y}_{13} f_3 + \ldots \tilde{y}_{1,N+1} f_{N+1} \\ a_0 &= \tilde{y}_{21} f_1 + \tilde{y}_{22} f_2 + \tilde{y}_{23} f_3 + \ldots \tilde{y}_{2,N+1} f_{N+1} \\ a_0 &= \tilde{y}_{31} f_1 + \tilde{y}_{32} f_2 + \tilde{y}_{33} f_3 + \ldots \tilde{y}_{3,N+1} f_{N+1} \\ &\vdots \\ a_0 &= \tilde{y}_{N+1,1} f_1 + \tilde{y}_{N+1,2} f_2 + \tilde{y}_{N+1,3} f_3 + \ldots \tilde{y}_{N+1,N+1} f_{N+1} \end{aligned}$$

Recall how we interpolate the function or one of its derivatives give the polynomial

$$f(\tilde{x}=0) = a_0 a_0$$
$$= \tilde{y}_{11}f_1 + \tilde{y}_{12}f_2 + \tilde{y}_{13}f_3 + \ldots \tilde{y}_{1,N+1}f_{N+1}$$
$$\frac{d}{dx}f(\tilde{x}=0) = a_0 a_1$$
$$= \tilde{y}_{21}f_1 + \tilde{y}_{22}f_2 + \tilde{y}_{23}f_3 + \ldots \tilde{y}_{2,N+1}f_{N+1}$$
$$\frac{d^2}{dx^2}f(\tilde{x}=0) = a_0 a_2 = \tilde{y}_{31}f_1 + \tilde{y}_{32}f_2 + \tilde{y}_{33}f_3 + \ldots \tilde{y}_{3,N+1}f_{N+1}$$

We can recognize that the rows of that $Y$ matrix are finite difference coefficients. We are driving real finite difference approximations, with the help of examples, using polynomial technique.

***Example No. 1***

Suppose we have three points distributed uniformly, and we would like to calculate a finite difference approximation first order and second order derivatives at the midpoint, let we take shifted coordinates, we have distribution of three points, we want the $\tilde{X}=0$, where we evaluating finite difference in this case is the midpoint, so that $\tilde{x}$ value for midpoint is zero.

$f(x) = a_0 + a_1 x + a_2 x^2$

| $x$ | $y = x^2$ |
|---|---|
| -1 | 1 |
| 0 | 0 |
| 1 | 1 |

$$[\tilde{x}] = \begin{bmatrix} -h \\ 0 \\ h \end{bmatrix} \Rightarrow [\tilde{X}] = [[\tilde{x}]^0 \quad [\tilde{x}]^1 \quad [\tilde{x}]^3] = \begin{bmatrix} 1 & -h & h^2 \\ 1 & 0 & 0 \\ 1 & h & h^2 \end{bmatrix}$$
$$\Rightarrow [\tilde{Y}] = [\tilde{X}]^{-1} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{-1}{2h} & 0 & \frac{1}{2h} \\ \frac{1}{2h^2} & \frac{-1}{h^2} & \frac{1}{2h^2} \end{bmatrix}$$

We have to build the $[W]$ matrix, let us suppose $h=1$. Remember that the rows of this $Y$ matrix are essentially are finite difference coefficients

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{-1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & \frac{-1}{1^2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix}$$

$f(x_{fd}) \cong a_0 = f_2$, $\frac{df(x_{fd})}{dx} \cong a_1 = \frac{f_3 - f_1}{2}$, $\frac{d^2 f(x_{fd})}{dx^2} \cong a_2 = \frac{f_1 - 2f_2 + f_3}{2}$

$$f(x_2) = f_2 + \frac{f_3 - f_1}{2}x + a_2 \frac{f_1 - 2f_2 + f_3}{2}x^2$$
$$f(0) = f_2 + \frac{f_3 - f_1}{2}(0) + a_2 \frac{f_1 - 2f_2 + f_3}{2}(0)^2$$
$$f(0) = 1$$

***Example No. 2***

Suppose we have three points distributed uniformly, and we would like to calculate a finite difference approximation first order and second order derivatives at the midpoint, let we take shifted coordinates, we have distribution of three points, we want the $\tilde{X}=0$, where we evaluating finite difference in this case is the midpoint, so that $\tilde{x}$ value for midpoint is zero.

| $x$ | $y = x^2$ |
|---|---|
| -0.5 | 0.25 |
| 0 | 0 |
| 0.5 | 0.25 |

$f(x) = a_0 + a_1 x + a_2 x^2$

$$[\tilde{x}] = \begin{bmatrix} -h \\ 0 \\ h \end{bmatrix} \Rightarrow [\tilde{X}] = [[\tilde{x}]^0 \quad [\tilde{x}]^1 \quad [\tilde{x}]^3] = \begin{bmatrix} 1 & -h & h^2 \\ 1 & 0 & 0 \\ 1 & h & h^2 \end{bmatrix}$$
$$\Rightarrow [\tilde{Y}] = [\tilde{X}]^{-1} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{-1}{2h} & 0 & \frac{1}{2h} \\ \frac{1}{2h^2} & \frac{-1}{h^2} & \frac{1}{2h^2} \end{bmatrix}$$

We have to build the $[W]$ matrix, let us suppose $h=0.5$.

Remember that the rows of this $Y$ matrix are essentially are finite difference coefficients

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{-1}{2(0.5)} & 0 & \frac{1}{2(0.5)} \\ \frac{1}{2(0.5)^2} & \frac{-1}{(0.5)^2} & \frac{1}{2(0.5)^2} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix}$$
$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 1 \\ 4 & -4 & 4 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix}$$

$f(x_{fd}) \cong a_0 = f_2$, $\frac{df(x_{fd})}{dx} \cong a_1 = \frac{f_3 - f_1}{1}$, $\frac{d^2 f(x_{fd})}{dx^2} \cong a_2 = \frac{4f_1 - 4f_2 + 4f_3}{0.25}$

$$f(x_2) = f_2 + \frac{f_3 - f_1}{1}x + a_2 \frac{16f_1 - 16f_2 + 16f_3}{1}x^2$$
$$f(0) = 0.25 + \frac{f_3 - f_1}{2}(0) + a_2[16f_1 - 16f_2 + 16f_3](0)^2$$
$$f(0) = 0.25$$

These are our finite difference approximations, which are also forward differences.

***Example No. 3***

Calculate a finite difference approximation of same points, but now on first point, we want to see how it behave at the first point, let we take shifted coordinates

$$[\tilde{x}] = \begin{bmatrix} 0 \\ h \\ 2h \end{bmatrix} \Rightarrow [\tilde{X}] = [[\tilde{x}]^0 \quad [\tilde{x}]^1 \quad [\tilde{x}]^3]$$
$$= \begin{bmatrix} 1 & 0 & 0 \\ 1 & h & h^2 \\ 1 & 2h & (2h)^2 \end{bmatrix} \Rightarrow \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}$$
$$= \begin{bmatrix} 0 & 1 & 0 \\ \frac{-3}{2h} & \frac{1}{2h} & \frac{-1}{2h} \\ \frac{1}{2h^2} & \frac{-1}{h^2} & \frac{1}{2h^2} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix}$$

$f(x_{fd}) \cong a_0 = f_1$, $\frac{df(x_{fd})}{dx} \cong a_1 = \frac{1.5f_1 - 2f_2 - 0.5f_3}{h}$, $\frac{d^2 f(x_{fd})}{dx^2} \cong a_2 = \frac{f_1 - 2f_2 + f_3}{2h^2}$

These are our finite difference approximations, these forward difference also.

***Example No. 4***

We want to evaluate derivatives, at the midpoints of four points, so our column vector of the$x$ positions doesn't have a zero in it.

$$[\tilde{x}] = \begin{bmatrix} \frac{-3h}{2} & \frac{-h}{2} & \frac{h}{2} & \frac{3h}{2} \end{bmatrix}^T$$

$$[\tilde{x}] = \begin{bmatrix} \frac{-3h}{2} \\ \frac{-h}{2} \\ \frac{h}{2} \\ \frac{3h}{2} \end{bmatrix} \Longrightarrow [\tilde{x}] = \begin{bmatrix} [\tilde{x}]^0 & [\tilde{x}]^1 & [\tilde{x}]^2 & [\tilde{x}]^3 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & \frac{-3h}{2} & \frac{9h^2}{4} & \frac{-27h^3}{8} \\ 1 & \frac{-3h}{2} & \frac{h^2}{4} & \frac{-h^3}{8} \\ 1 & \frac{-3h}{2} & \frac{h^2}{4} & \frac{h^3}{8} \\ 1 & \frac{-3h}{2} & \frac{9h^2}{4} & \frac{27h^3}{8} \end{bmatrix}$$

$$[\tilde{Y}] = [\tilde{X}]^{-1} = \begin{bmatrix} \frac{-1}{16} & \frac{9}{16} & \frac{9}{16} & \frac{-1}{16} \\ \frac{1}{24h} & \frac{-9}{8h} & \frac{9}{8h} & \frac{-1}{24h} \\ \frac{1}{4h^2} & \frac{-1}{4h^2} & \frac{-1}{4h^2} & \frac{1}{4h^2} \\ \frac{-1}{6h^3} & \frac{1}{2h^3} & \frac{-1}{2h^3} & \frac{1}{6h^3} \end{bmatrix} \Rightarrow \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{-1}{16} & \frac{9}{16} & \frac{9}{16} & \frac{-1}{16} \\ \frac{1}{24h} & \frac{-9}{8h} & \frac{9}{8h} & \frac{-1}{24h} \\ \frac{1}{4h^2} & \frac{-1}{4h^2} & \frac{-1}{4h^2} & \frac{1}{4h^2} \\ \frac{-1}{6h^3} & \frac{1}{2h^3} & \frac{-1}{2h^3} & \frac{1}{6h^3} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix}$$

Solving above system, we get

$f(x_{2.5}) \cong a_0 = \frac{-f_1+9f_2+9f_3-f_4}{16}$,
$\frac{d}{dx}f(x_{2.5}) \cong a_1 = \frac{f_1-27f_2+27f_3-f_4}{24h}$, $\frac{d^2}{dx^2}f(x_{2.5}) \cong a_2 = \frac{f_1-f_2-f_3+f_4}{(2h)^2}$

and so on. If we look at these finite difference approximations, and then where the finite difference is being evaluated is too difficult to determine.

### 3.2. Implementing the polynomial technique in MatLab

Since for small matrices we have been using, above technique, we really don't need to use MatLab, but the reason MatLab is good to use is when matrix size increases or maybe we don't have uniform spacing grid(Yserentant, 2005), and we need to evaluate a completely different finite difference approximation for every point. We have so far derived finite difference approximations symbolically, but if we want to 4th order accurate finite differences. Recall our matrix equation representing polynomial written at each discrete point. It always had the following form where $w's$ were just numerical constants. The $h's$ were symbolic.

$$\begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_N \end{bmatrix} = \begin{bmatrix} 1 & & & & \\ & h & & & \\ & & h^2 & & \\ & & & \ddots & \\ & & & & h^N \end{bmatrix}^{-1} \begin{bmatrix} 1 & w_{12} & w_{13} & \dots & w_{1N} \\ 1 & w_{22} & w_{23} & \dots & w_{2N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & w_{N+1,2} & w_{N+1,3} & \dots & w_{N+1,N} \end{bmatrix}^{-1} \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_{N+1} \end{bmatrix}$$

The key aspect here is that $[w]$ will be completely numerical, so it is easily inverted using MatLab, it is also important both separately inverted. This accommodates large matrices (Brandt et al., 1984), and avoids symbolic manipulation, and letting

$$[V] = [W]^{-1}$$

$$\begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_N \end{bmatrix} = \begin{bmatrix} 1.v_{11} & 1.v_{12} & 1.v_{13} & \dots & 1.v_{1N} \\ \frac{1}{h}.v_{21} & \frac{1}{h}.v_{22} & \frac{1}{h}.v_{23} & \dots & \frac{1}{h}.v_{2N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{h^N}.v_{N+1,1} & \frac{1}{h^N}.v_{N+1,2} & \frac{1}{h^N}.v_{N+1,3} & \dots & \frac{1}{h^N}.v_{N+1,N} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_{N+1} \end{bmatrix}$$

### 3.3. Finite difference approximations

We will write finite difference approximations the polynomial coefficients.

$$f = a_0 = v_{11}f_1 + v_{12}f_2 + v_{13}f_3 + \cdots + v_{1N}f_{N+1}$$
$$\frac{df}{dx} = a_1 = \frac{v_{21}f_1 + v_{22}f_2 + v_{23}f_3 + \cdots + v_{2N}f_{N+1}}{h}$$
$$\frac{d^2f}{dx^2} = 2a_2 = \frac{v_{31}f_1 + v_{32}f_2 + v_{33}f_3 + \cdots + v_{3N}f_{N+1}}{h^2}$$
$$\vdots$$

***MatLab Example No. 1***

Here we need five points to calculate five polynomial coefficients, i.e

$$[\tilde{x}] = [-2h \quad -h \quad 0 \quad h \quad 2h]^T,$$

build the $[W]$, suppose

$$h = 1, [\tilde{x}] = [-2 \quad -1 \quad 0 \quad 1 \quad 2]^T$$
$$[W] = \begin{bmatrix} [\tilde{x}]^0 & [\tilde{x}]^1 & [\tilde{x}]^2 & [\tilde{x}]^3 & [\tilde{x}]^4 \end{bmatrix}$$
$$= \begin{bmatrix} 1 & -2 & 4 & -8 & 16 \\ 1 & -1 & 1 & -1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 & 16 \end{bmatrix}$$

$$[V] = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0.0833 & -0.6667 & 0 & 0.6667 & -0.0833 \\ -0.0417 & 0.6667 & -1.2500 & 0.6667 & -0.0417 \\ -0.0833 & 0.1667 & 0 & -0.1667 & 0.0833 \\ 0.0417 & -0.1667 & 0.2500 & -0.1667 & 0.0417 \end{bmatrix}$$

Here we write the finite difference approximations, to incorporate the symbolic $h's$ back in.

$$f \cong a_0 = \frac{0.f_1 + 0.f_2 + 1.f_3 + 0.f_4 + 0.f_5}{1}$$
$$\frac{\partial}{\partial x}f \cong a_1$$
$$= \frac{0.0833.f_1 - 0.6667.f_2 + 0.f_3 - 0.6667.f_4 - 0.0833.f_5}{h}$$

$$\frac{\partial^2}{\partial x^2} f \cong 2a_2$$
$$= \frac{-0.0417.f_1 + 0.6667.f_2 - 1.2500.f_3 + 0.1667.f_4 + 0.0417.f_5}{h^2}$$
$$\vdots$$

***MatLab Example No. 2***

Five points and five polynomial coefficients, i.e., build $[W]$, and, $h = 0.5$

$$[\tilde{x}] = [-2h \quad -h \quad 0 \quad h \quad 2h]^T, \text{build}[W], \text{and},[\tilde{x}] = [-1 \quad -0.5 \quad 0 \quad 0.5 \quad 1]^T$$
$$[W] = [[\tilde{x}]^0 \quad [\tilde{x}]^1 \quad [\tilde{x}]^2 \quad [\tilde{x}]^3 \quad [\tilde{x}]^4] = \begin{bmatrix} 1 & -1 & 1 & -1 & 1 \\ 1 & -0.5 & 0.25 & -0.1250 & 0.0625 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0.5 & 0.25 & 0.1250 & 0.0625 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$
$$[V] = [W]^{-1} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 1.1667 & -1.3333 & 0 & -.1667 & 1.3333 \\ -.1667 & 2.6667 & -5.0000 & -0.1667 & 2.6667 \\ -.6667 & 1.3333 & 0 & 0.6667 & -1.3333 \\ 0.6667 & -2.6667 & 4.0000 & 0.6667 & -2.6667 \end{bmatrix}$$
$$f \cong a_0 = \frac{0.f_1 + 0.f_2 + 1.f_3 + 0.f_4 + 0.f_5}{1}$$
$$\frac{\partial}{\partial x} f \cong a_1 = \frac{1.1667.f_1 - 1.3333.f_2 + 0.f_3 - 0.1667.f_4 + 1.3333.f_5}{h}$$
$$\frac{\partial^2}{\partial x^2} f \cong 2a_2 = \frac{-0.1667.f_1 + 2.6667.f_2 + 5.0000.f_3 - 0.1667.f_4 + 2.6667.f_5}{h^2}$$
$$\vdots$$

***MatLab Example No. 3***

Five points and five polynomial coefficients, i.e., build$[W]$, and,$h = 0.25$,

$$[\tilde{x}] = [-2h \quad -h \quad 0 \quad h \quad 2h]^T [\tilde{x}] = [-0.5 \quad -0.25 \quad 0 \quad 0.25 \quad 0.5]^T$$
$$[V] = [W]^{-1} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 1.1667 & -1.3333 & 0 & -.1667 & 1.3333 \\ -.1667 & 2.6667 & -5.0000 & -0.1667 & 2.6667 \\ -.6667 & 1.3333 & 0 & 0.6667 & -1.3333 \\ 0.6667 & -2.6667 & 4.0000 & 0.6667 & -2.6667 \end{bmatrix}$$
$$f \approx a_0 = \frac{0.f_1 + 0.f_2 + 1.f_3 + 0.f_4 + 0.f_5}{1}$$
$$\frac{\partial}{\partial x} f \approx a_1 = \frac{1.1667.f_1 - 1.3333.f_2 + 0.f_3 - 0.1667.f_4 + 1.3333.f_5}{h}$$
$$\frac{\partial^2}{\partial x^2} f \approx 2a_2 = \frac{-0.1667.f_1 + 2.6667.f_2 + 5.0000.f_3 - 0.1667.f_4 + 2.6667.f_5}{h^2}$$
$$\vdots$$

## 4. Conclusion

There are many methods for deriving finite difference approximations, but among all methods the polynomial technique is better performant which is presented in this work. Because of simplicity, this is based on curve fitting. Implementation is also simple and costs less comparatively. Analysis of finite difference approximation is presented by polynomial technique. Few examples endorse claims. The results show the polynomial technique is simple and can be implemented in MatLab which makes it easier, which validate our claim.

## References


A. Bayliss, Goldstein, C.I., Turkel, E., 1983. An iterative method for the Helmholtz equation. Journal of Computational Physics 49, 443–457. https://doi.org/DOI: 10.1016/0021-9991(83)90139-0

Atangana, A., 2019. A New Numerical Approximation of Fractional Differentiation: Upwind Discretization for Riemann-Liouville and Caputo Derivatives, in: Taş, K., Baleanu, D., Machado, J.A.T. (Eds.), Mathematical Methods in Engineering. Springer International Publishing, Cham, pp. 193–212. https://doi.org/10.1007/978-3-319-90972-1_13

Bayona, V., Flyer, N., Fornberg, B., 2019. On the role of polynomials in RBF-FD approximations: III. Behavior near domain boundaries. Journal of Computational Physics 380, 378–399. https://doi.org/10.1016/j.jcp.2018.12.013

Brandt, A., McCormick, S.F., Ruge, J.W., 1984. Algebraic multigrid (AMG) for sparse matrix equations, in: Evans, D.J. (Ed.), Sparsity and Its Applications. Cambridge University Press, Cambridge.

Operto, S., Virieux, J., Amestoy, P., L'Excellent, J., Giraud, L., Ali, H., 2007. 3D finite-difference frequency-domain modeling of visco-acoustic wave propagation using a massively parallel direct solver: A feasibility study. Geophysics 72, SM195–SM211. https://doi.org/10.1190/1.2759835

Saad, Y., 1996. Iterative Methods for Linear system. PWS Publishing Company.

Shaikh, A.G., Sheikh, A.H., Ali, A., Zeb, S., 2019. Critical Review of Preconditioners for Helmholtz Equation and their Spectral Analysis. Indian Journal of Science and Technology 12, 8.

Sheikh, A.H., 2014. Development of Helmholtz Solver Based on Shifted Laplace Preconditioner and a Multigrid Deflation Technique (PhD Thesis). Delft University of Technology, The Netherlands.

Sheikh, A.H., Lahaye, D., Garcia Ramos, L., Nabben, R., Vuik, C., 2016. Accelerating the Shifted Laplace Preconditioner for the Helmholtz Equation by Multilevel Deflation. J. Comput. Phys. 322, 473–490. https://doi.org/10.1016/j.jcp.2016.06.025

Sheikh, A.H., Vuik, C., Lahaye, D., 2011. A scalable Helmholtz solver combining the shifted Laplace preconditioner with Multigrid deflation (No. 11– 01). DIAM, TU Delft Netherlands.

Yserentant, H., 2005. Sparse grid spaces for the numerical solution of the electronic Schrödinger equation. Numerische Mathematik 101, 381–389.