

A point-value enhanced finite volume method based on approximate delta functions



Li-Jun Xuan^{*}, Joseph Majdalani^{*}

Department of Aerospace Engineering, Auburn University, Auburn, AL, 36849, USA

ARTICLE INFO

Article history:

Received 28 April 2017

Received in revised form 24 October 2017

Accepted 31 October 2017

Available online 14 November 2017

Keywords:

Discontinuous Galerkin

Flux reconstruction

Correction procedure

Approximate delta function

ABSTRACT

We revisit the concept of an approximate delta function (ADF), introduced by Huynh (2011) [1], in the form of a finite-order polynomial that holds identical integral properties to the Dirac delta function when used in conjunction with a finite-order polynomial integrand over a finite domain. We show that the use of generic ADF polynomials can be effective at recovering and generalizing several high-order methods, including Taylor-based and nodal-based Discontinuous Galerkin methods, as well as the Correction Procedure via Reconstruction. Based on the ADF concept, we then proceed to formulate a Point-value enhanced Finite Volume (PFV) method, which stores and updates the cell-averaged values inside each element as well as the unknown quantities and, if needed, their derivatives on nodal points. The sharing of nodal information with surrounding elements saves the number of degrees of freedom compared to other compact methods at the same order. To ensure conservation, cell-averaged values are updated using an identical approach to that adopted in the finite volume method. Here, the updating of nodal values and their derivatives is achieved through an ADF concept that leverages all of the elements within the domain of integration that share the same nodal point. The resulting scheme is shown to be very stable at successively increasing orders. Both accuracy and stability of the PFV method are verified using a Fourier analysis and through applications to the linear wave and nonlinear Burgers' equations in one-dimensional space.

© 2017 Elsevier Inc. All rights reserved.

1. Introduction

The Dirac delta function represents a well-defined distribution that extends over a line of real numbers while possessing the unique property of vanishing everywhere except at the origin. Nonetheless, it still produces a unit value when integrated over the entire real line. Moreover, one of its most distinguishing properties stands, perhaps, in its ability to reproduce the values and derivatives of any function in integral form. In this paper, we show that the integral properties of the delta function may be useful in a number of computational settings as an alternative vehicle for evaluating functional values and derivatives over a finite domain. In numerical computations, however, the theoretical delta function suffers from singularities because of its sudden vanishing and infinite distribution. In 2011 and 2014, Huynh [1,2] introduced a very important concept, namely, that of an approximate delta function (ADF), which serves well to overcome these limitations. Accordingly, the ADF is defined as a finite-order polynomial that is capable of preserving the integral properties of the exact delta function in the evaluation of finite-order polynomials over finite domains. In this study, we extend the ADF concept

^{*} Corresponding authors.

E-mail addresses: LJ.Xuan@auburn.edu (L.-J. Xuan), joe.majdalani@auburn.edu (J. Majdalani).

by allowing the ADF polynomial to contain arbitrary coefficients and by defining ADF derivative weight functions that can be very effective in the development of a high-order numerical framework for solving partial differential equations.

It is well known that, in the field of computational fluid dynamics, low-order methods are often selected because of their simplicity and robustness, factors that jointly justify their recurrent use in engineering practice. Using similar CPU resources, however, high-order methods can provide more accurate solutions, albeit at the cost of increased complexity and reduced robustness. For this reason, numerous researchers have undertaken efforts to improve the manner by which high-order techniques may be constructed, with the aim of improving their accuracy while enhancing their stability and performance characteristics.

In this vein, the Discontinuous Galerkin (DG) method was developed because of its favorable attributes; these have led to its acceptance as one of the most widely relied upon high-order methods for solving the Navier–Stokes equations. The method itself was introduced in the context of the neutron transport problem by Reed and Hill [3], analyzed by LaSaint and Raviart [4] and then extended and popularized in the fluid dynamics community by Cockburn, Shu, Bassi, Rebay, and others (see [5–9], and the references therein).

One of the essential characteristics of the DG approach lies in its dependence on the Galerkin method to approximate a partial differential equation (PDE) that applies to a finite element. The corresponding PDE is subsequently converted into a series of ordinary differential equations (ODEs) that can be solved by standard methods.

Alternative approaches that seek to achieve high-order accuracy rely on differential forms. These may be exemplified by the pioneering work on the staggered-grid spectral method [10], as well as the spectral difference [11,12] and spectral volume approaches [13], which have been complemented by the elegant method of flux reconstruction [14,15,1] (FR), later evolving into the correction procedure via reconstruction [16–18] (CPR).

Among these high-order methods, different ways exist to appoint the degrees of freedom (DOFs) to each element at the cell-averaged or point-wise values, as well as their derivatives, which are later refreshed during the evaluation process. Although the Galerkin method and local reconstruction have been shown to provide formal avenues to derive the relevant ODEs in the context of the DG and differential approaches, the application of ADF to formulate the local ODEs will be used in this work as an alternative approach with particular benefits [1,2]. We further explore a generic ADF approach that contains arbitrary constants that can be specified in such a way to enhance the performance of the method to be reproduced. The characteristic attributes of this approach, such as simplicity, will constitute one of the main subjects of this article. In fact, one of the advantages of ADF implementation will be shown to be associated with its versatility in handling different DOF specifications.

It should also be noted that, in recent years, a well-developed constrained interpolation profile (CIP) with multi-moment finite volume (MFV) method has been developed (see Xie et al. [19] and the references therein). Apart from the cell-averaged value of a given element, MFV introduces additional DOFs on the element's edge and nodal points. The ability to share this supplementary information with neighboring elements transforms MFV into a more efficient scheme for saving the number of DOFs compared to other high-order methods of comparable accuracy. Pursuant to this approach, the sharing of additional DOFs within the context of continuity leads to the enhancement of the scheme's robustness. In fact, a similar concept may be attributed to the Active Flux (AF) method [20,21], where the unknown values at edge-based flux points are treated as independent DOFs and updated at every time step.

Because nodal points undergo the highest sharing rate, being shared by more elements than edges, it proves more efficient to increase the amount of information that is being communicated with a given element by placing all additional DOFs on the nodes only. As such, it is possible to augment the nodal information and extend the MFV and AF approaches by adding not only the unknown functional values at the nodal points, but also their derivatives. In this process, the updating of cell-averaged values may be accomplished in a manner that mirrors the traditional finite volume (FV) approach, thus guaranteeing the conservation of the scheme.

In practice, the manner by which additional information is updated on nodal points and edges constitutes the most distinguishing features in the MFV and AF schemes. We presently rely on an ADF procedure and set the integral domain to encompass all of the elements surrounding the point in question. This increases the radius of influence, as it were, that accompanies each update. Our nodal updating procedure may hence be likened to the case of an overlapped DG, where nodal values and derivatives can provide sufficient information for the high-order reconstruction of the unknown quantity in each element. As for the order of the “DG on the node,” it is no longer constrained by the DOFs on the nodal point itself. The nodal updating becomes comparable to the PnPm procedure [22]. Furthermore, since the precision of the method may be improved by increasing the amount of information that is assigned to the nodal points, we call this strategy a point-value enhanced finite volume method (PFV). As to the temporal updating, a conventional third-order total variation diminishing (TVD) Runge–Kutta scheme can be conveniently employed.

In this article, the approximate delta function is revisited and extended in Section 2 to comprise arbitrary constants. This is followed by applying the extended ADF to recover and generalize Taylor-based DG, nodal-based DG, and FR/CPR methods in Sections 3, 4, and 5, respectively. In Section 6, the ADF-based point value enhanced finite volume method is defined, implemented, and verified numerically. We retire in Section 7 with some conclusions and recommendations for future work.

2. Reviewing and extending the approximate delta function (ADF)

To set the stage, we recall that a delta function exhibits the following integral property:

$$\int_{-\infty}^{+\infty} f(y)\delta(y-x)dy = f(x), \quad (1)$$

where $f(x)$ stands for a continuous function of compact support. In 2011 and 2014, Huynh [1,2] introduced an approximate delta function (ADF) in the form of a finite-order polynomial that can mirror the integral property of a Dirac delta function over a finite domain, $x \in [-1, 1]$. In this work, we find it useful to change the domain boundaries to the generic interval $x \in [a, b]$ and write

$$\int_a^b P_N(y)\tilde{\delta}_N(y, z)dy = P_N(z), \quad (2)$$

where $P_N(y)$ is any N th-order polynomial and $\tilde{\delta}_N(x, z)$ represents an ADF polynomial of order N . The present analysis begins by reviewing the ADF formulation in [2] while extending it to the case of $x \in [a, b]$. We also consider generic ADF forms that have orders higher than N and that enable us to retrieve the derivatives of $P_N(y)$.

2.1. Revisiting Huynh's ADF concept

According to (3.4) in Ref. [2], a Legendre polynomial expansion may be used to express the ADF of order N explicitly as

$$\tilde{\delta}_N(x, z) = \frac{1}{h} \sum_{i=0}^N (2i+1)L_i(\xi)L_i(\eta), \quad \xi = \frac{x-x_c}{h/2}, \quad \eta = \frac{z-x_c}{h/2}, \quad (3)$$

where the present notation is used with $x_c = \frac{1}{2}(a+b)$ representing the domain center, $h = b-a$ denoting the domain width, and $L_i(\xi)$ standing for the Legendre polynomial of order i .

By assuming a solution interval that is bracketed by $[a, b] = [-1, 1]$ as in Ref. [2], and for $N \leq 4$, one may readily deduce from (3) that

$$\begin{aligned} \tilde{\delta}_0(x, z) &= \frac{1}{2}, \quad \tilde{\delta}_1(x, z) = \frac{1}{2} + \frac{3z}{2}x, \quad \tilde{\delta}_2(x, z) = \frac{9-15z^2}{8} + \frac{3z}{2}x + \frac{45z^2-15}{8}x^2, \\ \tilde{\delta}_3(x, z) &= \frac{9-15z^2}{8} + \frac{75z-105z^3}{8}x + \frac{45z^2-15}{8}x^2 + \frac{175z^3-105z}{8}x^3, \\ \tilde{\delta}_4(x, z) &= \frac{225-1050z^2+945z^4}{128} - \frac{105z^3-75z}{8}x - \frac{525-4410z^2+4725z^4}{64}x^2 \\ &\quad + \frac{175z^3-105z}{8}x^3 + \frac{945-9450z^2+11025z^4}{128}x^4. \end{aligned} \quad (4)$$

Note that at $z=0$, we recover four terms with two identical expressions,

$$\begin{aligned} \tilde{\delta}_0(x, 0) &= \tilde{\delta}_1(x, 0) = \frac{1}{2}, \quad \tilde{\delta}_2(x, 0) = \tilde{\delta}_3(x, 0) = \frac{9}{8} - \frac{15}{8}x^2, \\ \tilde{\delta}_4(x, 0) &= \frac{225}{128} - \frac{525}{64}x^2 + \frac{945}{128}x^4. \end{aligned} \quad (5)$$

Similarly at $z = \frac{1}{4}$, we extract,

$$\begin{aligned} \tilde{\delta}_0\left(x, \frac{1}{4}\right) &= \frac{1}{2}, \quad \tilde{\delta}_1\left(x, \frac{1}{4}\right) = \frac{1}{2} + \frac{3}{8}x, \quad \tilde{\delta}_2\left(x, \frac{1}{4}\right) = \frac{129}{128} + \frac{3}{8}x - \frac{195}{128}x^2, \\ \tilde{\delta}_3\left(x, \frac{1}{4}\right) &= \frac{129-195x^2}{128} + \frac{1095x-1505x^3}{512}, \\ \tilde{\delta}_4\left(x, \frac{1}{4}\right) &= \frac{41745+101745x^4}{32768} + \frac{1095x-1505x^3}{512} - \frac{68565x^2}{16384}. \end{aligned} \quad (6)$$

For the reader's convenience, the shapes of the approximate delta functions, $\tilde{\delta}_N(x, z)$ for $1 \leq N \leq 4$ are illustrated in Fig. 1 at both $z=0$ and $z = \frac{1}{4}$. It should be noted that identical ADF distributions may be found in Ref. [2] for $\tilde{\delta}_N(x, 0)$ ($N = 0, 1, \dots, 8$) and $\tilde{\delta}_N(x, 1)$ ($N = 4, 5, \dots, 8$).

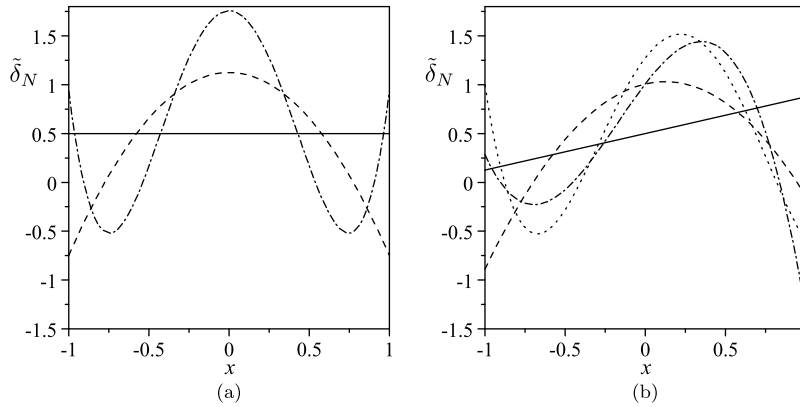


Fig. 1. The shape of the approximate delta function (ADF) for (a) $\tilde{\delta}_1(x, 0)$ (—), $\tilde{\delta}_2(x, 0)$ (---), and $\tilde{\delta}_4(x, 0)$ (- · -), as well as (b) $\tilde{\delta}_1(x, \frac{1}{4})$ (—), $\tilde{\delta}_2(x, \frac{1}{4})$ (---), $\tilde{\delta}_3(x, \frac{1}{4})$ (- · -), and $\tilde{\delta}_4(x, \frac{1}{4})$ (····).

2.2. Defining a generic ADF expression

At this point, it may be helpful to specify the difference between ADFs at two distinct orders N and M by introducing the gap function, $\epsilon_M^N(x, z)$, where

$$\epsilon_M^N(x, z) = \tilde{\delta}_N(x, z) - \tilde{\delta}_M(x, z) = \frac{1}{h} \sum_{i=M+1}^{i=N} (2i+1)L_i(\xi)L_i(\eta); \quad N > M. \quad (7)$$

It is then possible to define a generic ADF, $\tilde{\delta}_{N|K}(x, z)$, as a polynomial of order $N+K$, namely, by superimposing a function, $\tilde{\delta}_N(x, z)$, of order N , and a sum of gap functions, $\epsilon_N^{N+i}(x, z)$, for $i = 1, 2, \dots, K$. This may be accomplished by taking

$$\tilde{\delta}_{N|K}(x, z) = \tilde{\delta}_N(x, z) + c_1 \epsilon_N^{N+1}(x, z) + c_2 \epsilon_N^{N+2}(x, z) + \dots + c_K \epsilon_N^{N+K}(x, z), \quad (8)$$

where the arbitrary coefficients are given by $c_i \in \mathbb{R}$, $i = 1, 2, 3, \dots, K$. In the foregoing, the indices $N|K$ denote a generic ADF polynomial of order $(N+K)$. By virtue of the ADF integral property, one may readily substitute $\tilde{\delta}_{N|K}(x, z)$ back into (2) to show that

$$\int_a^b P_N(y) \tilde{\delta}_{N|K}(y, z) dy = P_N(z), \quad (9)$$

where not only the order of the polynomial in $\tilde{\delta}_{N|K}(x, z)$ is raised to $N+K$, but also K arbitrary coefficients are introduced. Equation (9) can also be viewed as an integral-form definition of ADF other than the one given by (8). It can thus be seen that the function $\tilde{\delta}_M(x, z)$ not only satisfies

$$\int_a^b P_N(y) \tilde{\delta}_M(y, z) dy = P_N(z) \quad (10)$$

but also possesses an infinite number of solutions, when $M > N$. Furthermore, one can write $\tilde{\delta}_M(y, z) = \tilde{\delta}_{N|M-N}(x, z)$ when $M = N$, thus leading to $\tilde{\delta}_{N|0}(x, z) = \tilde{\delta}_N(x, z)$.

2.3. Defining ADF polynomial derivative weight functions

Besides the functions themselves, it is possible to define $(N+K)$ th-order ADF weight functions to generate the actual derivatives of an N th-order polynomial. This may be achieved by specifying $\tilde{\delta}'_{N|K}(y, z)$ and $\tilde{\delta}''_{N|K}(y, z)$ with the following integral properties:

$$\int_a^b P_N(y) \tilde{\delta}'_{N|K}(y, z) dy = \frac{dP_N(z)}{dz}, \quad \int_a^b P_N(y) \tilde{\delta}''_{N|K}(y, z) dy = \frac{d^2 P_N(z)}{dz^2}. \quad (11)$$

Then based on (9), one can deduce that

$$\tilde{\delta}'_{N|K}(x, z) = \frac{\partial \tilde{\delta}_{N|K}(x, z)}{\partial z}, \quad \tilde{\delta}''_{N|K}(x, z) = \frac{\partial^2 \tilde{\delta}_{N|K}(x, z)}{\partial z^2}. \quad (12)$$

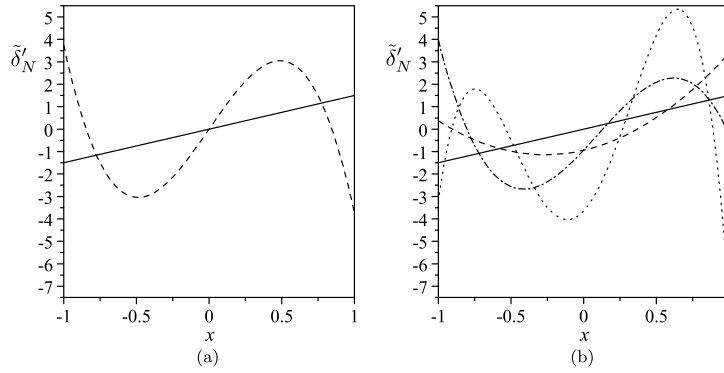


Fig. 2. The shape of the ADF derivative weight functions for (a) $\tilde{\delta}'_1(x, 0)$ (—) and $\tilde{\delta}'_3(x, 0)$ (---), as well as (b) $\tilde{\delta}'_1(x, \frac{1}{4})$ (—), $\tilde{\delta}'_2(x, \frac{1}{4})$ (---), $\tilde{\delta}'_3(x, \frac{1}{4})$ (- · -), and $\tilde{\delta}'_4(x, \frac{1}{4})$ (····).

More generally, for a given n th-order derivative $d^n P_N(z)/dz^n$, one may specify a corresponding ADF derivative weight function using

$$\tilde{\delta}_{N|K}^{(n)}(x, z) = \frac{\partial^n \tilde{\delta}_{N|K}(x, z)}{\partial z^n}. \quad (13)$$

For example, using the Legendre polynomial expression in (3), $\tilde{\delta}'_N(x, z)$ can be written as

$$\tilde{\delta}'_N(x, z) = \frac{\partial \tilde{\delta}_N(x, z)}{\partial z} = \frac{1}{2} \sum_{i=0}^N (2i+1) L_i(\xi) \frac{dL_i(\eta)}{d\eta}, \quad (14)$$

which, for the special case of $[a, b] = [-1, 1]$ and $N \leq 4$, yields the following sequence of $\tilde{\delta}'_N(x, z)$ ($N = 0, 1, \dots, 4$):

$$\begin{aligned} \tilde{\delta}'_0(x, z) &= 0, \quad \tilde{\delta}'_1(x, z) = \frac{3}{2}x, \quad \tilde{\delta}'_2(x, z) = -\frac{15}{4}z + \frac{3}{2}x + \frac{45}{4}zx^2, \\ \tilde{\delta}'_3(x, z) &= -\frac{15}{4}z + \frac{75 - 315z^2}{8}x + \frac{45z}{4}x^2 + \frac{525z^2 - 105}{8}x^3, \\ \tilde{\delta}'_4(x, z) &= \frac{945z^3 - 525z}{32} + \frac{75 - 315z^2}{8}x + \frac{2205z - 4725z^3}{16}x^2 \\ &\quad + \frac{525z^2 - 105}{8}x^3 + \frac{11025z^3 - 4725z}{32}x^4. \end{aligned} \quad (15)$$

Similarly, for $z = 0$ and $z = \frac{1}{4}$, we have

$$\begin{aligned} \tilde{\delta}'_1(x, 0) &= \tilde{\delta}'_2(x, 0) = \frac{3}{2}x, \\ \tilde{\delta}'_3(x, 0) &= \tilde{\delta}'_4(x, 0) = \frac{75x - 105x^3}{8}, \end{aligned} \quad (16)$$

and

$$\begin{aligned} \tilde{\delta}'_1\left(x, \frac{1}{4}\right) &= \frac{3}{2}x, \quad \tilde{\delta}'_2\left(x, \frac{1}{4}\right) = -\frac{15}{16} + \frac{3}{2}x + \frac{45}{16}x^2, \\ \tilde{\delta}'_3\left(x, \frac{1}{4}\right) &= -\frac{15}{16} + \frac{885}{128}x + \frac{45}{16}x^2 - \frac{1155}{128}x^3, \\ \tilde{\delta}'_4\left(x, \frac{1}{4}\right) &= -\frac{7455}{2048} + \frac{885}{128}x + \frac{30555}{1024}x^2 - \frac{1155}{128}x^3 - \frac{64575}{2048}x^4. \end{aligned} \quad (17)$$

The shapes of the ADF derivative weight functions, $\tilde{\delta}'_N(x, z)$ for $1 \leq N \leq 4$ are illustrated in Fig. 2 at both $z = 0$ and $z = \frac{1}{4}$. It should be noted that the scales and shapes of $\tilde{\delta}'_N(x, z)$ are different from those of the ADF introduced in Ref. [2] and illustrated in Fig. 1.

3. ADF in relation to the Taylor-based DG method for the scalar wave equation

3.1. Second- and third-order Taylor-based DG schemes for the one-dimensional wave equation

We begin by briefly reviewing the DG method in the context of the linear, one-dimensional, scalar wave equation. Using standard nomenclature, we consider:

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0; \quad f(u) = au, \quad a > 0, \quad x \in [0, 1]. \quad (18)$$

To proceed, we first subdivide the $[0, 1]$ interval uniformly into multiple elements $\{\Omega^e\}$, and then use Taylor expansion to discretize $u(x)$ over each element by taking

$$\begin{aligned} u^e(x, t) &= \bar{u}(t) + \frac{\partial u}{\partial x}(x - x_c) + \frac{\partial^2 u}{\partial x^2} \left[\frac{(x - x_c)^2}{2} - \frac{h^2}{24} \right] + \cdots \\ &= \bar{u}(t)B_0(x) + u_1(t)B_1(x) + u_2(t)B_2(x) + \cdots, \end{aligned} \quad (19)$$

where $\bar{u}$ denotes the cell-averaged u , x_c refers to the location of the cell-center, and h alludes to the width of the cell. Being a normal polynomial, the Taylor basis $(x - x_c)^i$ ($i = 1, 2, \dots$) can be modified through subtraction to produce

$$\int_{\Omega^e} B_i(x) dx = 0, \quad i > 1. \quad (20)$$

Next, (18) may be multiplied term by term by the Taylor bases

$$[B_0(x), B_1(x), B_2(x), \dots] = \left[1, x - x_c, \frac{(x - x_c)^2}{2} - \frac{h^2}{24}, \dots \right], \quad (21)$$

then integrated over the cell Ω^e to obtain the weak form

$$\int_{\Omega^e} \frac{\partial u^e}{\partial t} B_j dx + au^e B_j|_R - au^e B_j|_L - \int_{\Omega^e} au^e \frac{\partial B_j}{\partial x} dx = 0, \quad j = 0, 1, 2, \dots \quad (22)$$

where R and L refer to the adjacent right and left points. To produce a third-order scheme, the Taylor-series expansion in (19) may be inserted into the weak form (22). Then using the upwind flux with $a > 0$ at the cell interface, we can put

$$\begin{aligned} &\begin{bmatrix} h & 0 & 0 \\ 0 & \frac{h^3}{12} & 0 \\ 0 & 0 & \frac{h^5}{720} \end{bmatrix} \begin{bmatrix} \frac{d\bar{u}}{dt} \\ \frac{du_1}{dt} \\ \frac{du_2}{dt} \end{bmatrix}_i - a \begin{bmatrix} 0 & 0 & 0 \\ h & 0 & 0 \\ 0 & \frac{h^3}{12} & 0 \end{bmatrix} \begin{bmatrix} \bar{u} \\ u_1 \\ u_2 \end{bmatrix}_i + \\ &a(\bar{u} + \frac{1}{2}hu_1 + \frac{1}{12}h^2u_2)_i \begin{bmatrix} 1 \\ \frac{h}{2} \\ \frac{h^2}{12} \end{bmatrix} - a(\bar{u} + \frac{1}{2}hu_1 + \frac{1}{12}h^2u_2)_{i-1} \begin{bmatrix} 1 \\ -\frac{h}{2} \\ \frac{h^2}{12} \end{bmatrix} = 0. \end{aligned} \quad (23)$$

The resulting assortment of ordinary differential equations with respect to the variables $\bar{u}(t), u_1(t), u_2(t)$ may be handled using standard ODE solvers. In this context, the second and third-order schemes may be directly written as

$$\begin{cases} \frac{d\bar{u}_i}{dt} = \frac{a(\bar{u}_{i-1} - \bar{u}_i)}{h} + \frac{a(u_{1,i-1} - u_{1,i})}{2} \\ \frac{du_{1,i}}{dt} = \frac{6a(\bar{u}_i - \bar{u}_{i-1})}{h^2} - \frac{3a(u_{1,i-1} + u_{1,i})}{h} \end{cases}, \quad (24)$$

and

$$\begin{cases} \frac{d\bar{u}_i}{dt} = \frac{a(\bar{u}_{i-1} - \bar{u}_i)}{h} + \frac{a(u_{1,i-1} - u_{1,i})}{2} + \frac{ah(u_{2,i-1} - u_{2,i})}{12} \\ \frac{du_{1,i}}{dt} = \frac{6a(\bar{u}_i - \bar{u}_{i-1})}{h^2} - \frac{3a(u_{1,i-1} + u_{1,i})}{h} - \frac{a(u_{2,i-1} + u_{2,i})}{2} \\ \frac{du_{2,i}}{dt} = \frac{60a(\bar{u}_{i-1} - \bar{u}_i)}{h^3} + \frac{30a(u_{1,i-1} + u_{1,i})}{h^2} + \frac{5a(u_{2,i-1} - u_{2,i})}{h} \end{cases}. \quad (25)$$

3.2. ADF-based scheme underlying a Taylor-based solution

In what follows, we show how the derivatives may be updated using a generic ADF. The ensuing scheme may be viewed as a DG variant that implements the ADF concept instead of the Galerkin approach to derive the updating ODEs, thus leading to an ADF-DG method. On any interval $x \in [x_c - \frac{1}{2}h, x_c + \frac{1}{2}h]$, it is straightforward to evaluate generic ADF polynomials for the first-order derivatives from (13), namely,

$$\begin{aligned}\tilde{\delta}'_1(x, x_c) &= 12h^{-3}(x - x_c), \\ \tilde{\delta}'_{1|1}(x, x_c) &= \tilde{\delta}'_1(x, x_c) + 6C[(x - x_c)^2 - h^2/12]h^{-3},\end{aligned}\quad (26)$$

where C represents an arbitrary constant. In the interest of simplicity, because the only point of interest is x_c , all of the related $\tilde{\delta}_{N|M}^{(n)}(x, x_c)$ are abbreviated by $\tilde{\delta}_{N|M}^{(n)}(x)$. In general, $\tilde{\delta}'_{1|N}(x)$ will contain N arbitrary constants. In like fashion, we may retrieve $\tilde{\delta}'_2(x) = \tilde{\delta}'_1(x)$, and so,

$$\tilde{\delta}'_{2|1}(x) = \tilde{\delta}'_2(x) + 6C[20(x - x_c)^3 - 3h^2(x - x_c)]h^{-5}. \quad (27)$$

Along similar lines, the ADF weight functions for the second-order derivatives may be evaluated to be

$$\begin{aligned}\tilde{\delta}''_2(x) &= -30h^{-3} + 360h^{-5}(x - x_c)^2, \\ \tilde{\delta}''_{2|1}(x) &= \tilde{\delta}''_2(x) + 60C[20(x - x_c)^3 - 3h^2(x - x_c)]h^{-6}.\end{aligned}\quad (28)$$

In the implementation of the ADF-DG approach, the cell-averaged values can be updated in a DG manner, while rewriting the ODEs for the updating derivatives using

$$\frac{du_{1,i}}{dt} = \int_{\Omega^e} \frac{\partial u}{\partial t} \tilde{\delta}'_{N|1}(x) dx = \int_{\Omega^e} au \frac{\partial \tilde{\delta}'_{N|1}(x)}{\partial x} dx - (au^e \tilde{\delta}'_{N|1}|_R - au^e \tilde{\delta}'_{N|1}|_L). \quad (29)$$

Here $N = 1$ corresponds to the second-order scheme, where one may use $\tilde{\delta}'_{1|1}(x)$, reduce the Taylor expansion to a linear polynomial, and substitute the outcome into (29). This operation yields,

$$\frac{du_{1,i}}{dt} = \frac{6a(\bar{u}_i - \bar{u}_{i-1} - C[u]_i)}{h^2} - \frac{3a(u_{1,i-1} + u_{1,i})}{h}, \quad (30)$$

where the interface jump term may be readily determined from the expression $[u]_i = (u^+ - u^-)_{x_{i-1/2}} = \bar{u}_i - \frac{1}{2}hu_{1,i} - (\bar{u}_{i-1} + \frac{1}{2}hu_{1,i-1})$. In general, it is possible to employ a higher-order ADF derivative weight function, $\tilde{\delta}'_{1|M}$ with $M > 1$, thus leading to additional constants and, therefore, greater flexibility to enhance the scheme's capabilities. The ensuing analysis, however, falls beyond the scope of the present study and will be deferred to later work. We can see that when $C = 0$, the ADF-DG approach reproduces the second-order Taylor-based DG formulation identically.

To obtain a third-order scheme, the same procedure may be followed by first applying the ADF concept to derive the updating ODEs for the first and second-order derivatives using

$$\begin{aligned}\frac{du_{1,i}}{dt} &= \int_{\Omega^e} \frac{\partial u}{\partial t} \tilde{\delta}'_{2|1}(x) dx = \int_{\Omega^e} au \frac{\partial \tilde{\delta}'_{2|1}(x)}{\partial x} dx - (au^e \tilde{\delta}'_{2|1}|_R - au^e \tilde{\delta}'_{2|1}|_L), \\ \frac{du_{2,i}}{dt} &= \int_{\Omega^e} \frac{\partial u}{\partial t} \tilde{\delta}''_{2|1}(x) dx = \int_{\Omega^e} au \frac{\partial \tilde{\delta}''_{2|1}(x)}{\partial x} dx - (au^e \tilde{\delta}''_{2|1}|_R - au^e \tilde{\delta}''_{2|1}|_L),\end{aligned}\quad (31)$$

and then substituting a quadratic polynomial expansion of u to retrieve

$$\begin{cases} \frac{du_{1,i}}{dt} = \frac{6a(\bar{u}_i - \bar{u}_{i-1} + C_1[u]_i)}{h^2} - \frac{3a(u_{1,i-1} + u_{1,i})}{h} - \frac{a(u_{2,i-1} + u_{2,i})}{2} \\ \frac{du_{2,i}}{dt} = \frac{60a(\bar{u}_{i-1} - \bar{u}_i + C_2[u]_i)}{h^3} + \frac{30a(u_{1,i-1} + u_{1,i})}{h^2} + \frac{5a(u_{2,i-1} - u_{2,i})}{h} \end{cases}, \quad (32)$$

where the interface jump term may be identified as $[u]_i \equiv (u^+ - u^-)_{x_{i-1/2}} = \bar{u}_i - \frac{1}{2}hu_{1,i} + \frac{1}{12}h^2u_{2,i} - (\bar{u}_{i-1} + \frac{1}{2}hu_{1,i-1} + \frac{1}{12}h^2u_{2,i-1})$.

Here too, by setting $C_1 = C_2 = 0$, the third-order ADF-DG scheme returns the third-order Taylor-based DG expressions given by (25). Moreover, not only do the ADF-DG relations recover the Taylor-based DG approximations as special cases, their second- and third-order ADF-DG formulations contain arbitrary coefficients that may be judiciously adjusted to enhance the properties of the resulting scheme, specifically, by controlling dispersive and dissipative errors.

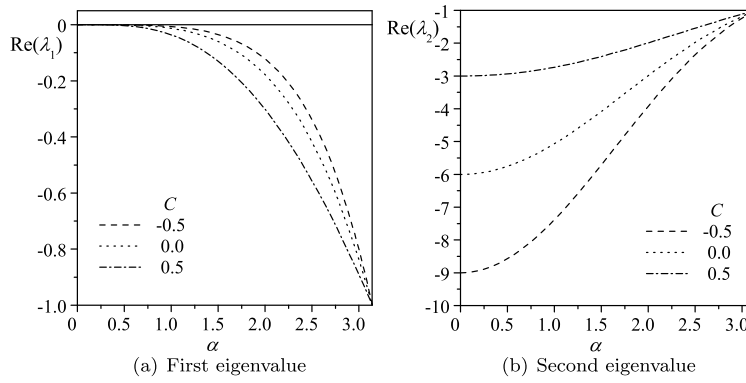


Fig. 3. The real parts of the two eigenvalues corresponding to the coefficient matrix $\mathbf{M}$ for a second-order Taylor-based ADF-DG using the constant coefficients of $C = -0.5, 0$, and 0.5 .

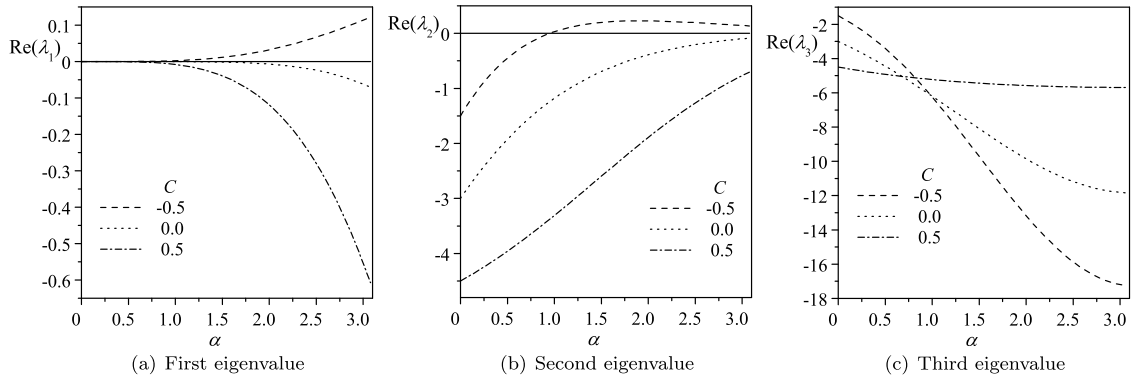


Fig. 4. The real parts of the three eigenvalues corresponding to the coefficient matrix $\mathbf{M}$ for a third-order Taylor-based ADF-DG using the constant coefficients of $C = -0.5, 0$, and 0.5 .

3.3. Fourier stability verification of the Taylor-based ADF-DG method

At this juncture, a conventional Fourier analysis is undertaken to assess the stability of a half-discretized ODE system of both second and third-order ADF-DG schemes. Letting $j = \sqrt{-1}$ and α refer to the imaginary unit and a wave number such that $\alpha \in [0, \pi]$, the solution of the ODE system using Fourier analysis may be written as

$$\mathbf{U}_i = e^{i\alpha j} \mathbf{A}, \quad (33)$$

where the integer i represents the cell index and $\mathbf{A}$ denotes the vector amplitude such that $\mathbf{U}_i = [\bar{u}_i, u_{1,i}, \dots]^T$ corresponds to the unknown vector stored in each element. By substituting (33) into the half-discretized, second-order ADF-DG system of ODEs, one arrives at

$$\frac{d\mathbf{A}}{dt} = \mathbf{M}\mathbf{A}, \quad (34)$$

where the coefficient matrix $\mathbf{M}$ consists of

$$\mathbf{M} = \begin{bmatrix} e^{-j\alpha} - 1 & \frac{1}{2}(e^{-j\alpha} - 1) \\ 6(C - 1)(e^{-j\alpha} - 1) & 3(C - 1)(e^{-j\alpha} + 1) \end{bmatrix}. \quad (35)$$

Fig. 3 displays the real parts of the first and second eigenvalues (λ_1, λ_2) of $\mathbf{M}$ using different coefficients, namely, $C = -0.5, 0$, and 0.5 . As one may infer graphically, the negative eigenvalues confirm the stability of the scheme for all three coefficients. Furthermore, the present analysis enables us to realize that the scheme tends to be more diffusive for the first eigenvalue when $C = 0.5$. The flexibility in selecting C can therefore be used to optimize the second-order Taylor-based DG, a task that can be relegated to a future study.

For the third-order ADF-DG, the problem may be simplified by taking $C_1 = C_2 = C$. Then following a similar procedure as before, the real parts of the first, second, and third eigenvalues, i.e. ($\lambda_1, \lambda_2, \lambda_3$), may be extracted and shown in Fig. 4. Forthwith, it may be immediately seen that the scheme remains stable for $C = 0$ and 0.5 , although it exhibits a slightly diffusive behavior for $C = 0.5$.

4. ADF in relation to the nodal DG method for the scalar, one-dimensional wave equation

4.1. Nodal discontinuous Galerkin method

For the second-order scheme, it is possible to introduce the shape functions

$$\lambda_1(x) = 1 - \frac{x - x_c + h/2}{h} \quad \text{and} \quad \lambda_2(x) = \frac{x - x_c + h/2}{h}, \quad (36)$$

thus prescribing the base functions, $B_i(x)$, $i = 1, 2$, where

$$[B_1(x), B_2(x)] = [\lambda_1(x), \lambda_2(x)]. \quad (37)$$

To make further headway, we follow a similar procedure to the one we pursued in the Taylor-based DG formulation. Here, the updating ODEs for the second-order nodal DG approach may be expressed as

$$\begin{cases} \frac{du_{1,i}}{dt} = \frac{a(4u_{2,i-1} - 3u_{1,i} - u_{2,i})}{h} \\ \frac{du_{2,i}}{dt} = \frac{a(3u_{1,i} - u_{2,i} - 2u_{2,i-1})}{h} \end{cases}. \quad (38)$$

The shape functions associated with a third-order scheme may be similarly compacted into

$$[B_1(x), B_2(x), B_3(x)] = \{2\lambda_1(x)[\lambda_1(x) - \frac{1}{2}], 4\lambda_1(x)\lambda_2(x), 2\lambda_2(x)[\lambda_2(x) - \frac{1}{2}]\}. \quad (39)$$

Lastly, the updating ODEs may be derived and re-arranged into

$$\begin{cases} \frac{du_{1,i}}{dt} = \frac{a(u_{3,i} - 6u_{1,i} + 9u_{3,i-1} - 4u_{2,i})}{h} \\ \frac{du_{2,i}}{dt} = \frac{a(5/2 u_{1,i} - u_{3,i} - 3/2 u_{3,i-1})}{h} \\ \frac{du_{3,i}}{dt} = \frac{a(3 u_{3,i-1} + 4u_{2,i} - 3 u_{3,i} - 4u_{1,i})}{h} \end{cases}. \quad (40)$$

4.2. ADF-based scheme underlying the nodal DG solution

For the second-order representation, the solution on each cell may be reconstructed from

$$\tilde{u} = u_1 \lambda_1(x) + u_2 \lambda_2(x). \quad (41)$$

In this case, we may use $\tilde{\delta}_{1|1}(x, z)$ to obtain the updating differential relations. First, the ADF expressions may be determined at the left and right points using

$$\begin{aligned} \tilde{\delta}_{1|1,1} &= \frac{1}{h} - \frac{6\xi}{h} + \frac{6C}{h} \left(\xi^2 - \frac{1}{12} \right), \\ \tilde{\delta}_{1|1,2} &= \frac{1}{h} + \frac{6\xi}{h} + \frac{6C}{h} \left(\xi^2 - \frac{1}{12} \right), \end{aligned} \quad (42)$$

where $\xi = (x - x_c)/h$ and $\tilde{\delta}_{1|1,i} = \tilde{\delta}_{1|1}(x, x_i)$, $i = 1, 2$ represent the generic ADF polynomials on points 1 and 2, respectively. Then, by virtue of the integral property, we may write

$$\frac{du_{1,i}}{dt} = \int_{\Omega^e} \frac{\partial u}{\partial t} \tilde{\delta}_{1|1,1} dx = \int_{\Omega^e} au \frac{\partial \tilde{\delta}_{1|1,1}(x)}{\partial x} dx - (au^e \tilde{\delta}_{1|1,1}|_R - au^e \tilde{\delta}_{1|1,1}|_L), \quad (43)$$

and retrieve

$$\frac{du_{1,i}}{dt} = \frac{a[(4 + C_1)u_{2,i-1} - (3 + C_1)u_{1,i} - u_{2,i}]}{h}. \quad (44)$$

We may similarly multiply $\tilde{\delta}_{1|1,2}(x)$ by the linear wave equation (18) and perform integration by parts to deduce

$$\frac{du_{2,i}}{dt} = \frac{a[(3 - C_2)u_{1,i} - u_{2,i} - (2 - C_2)u_{2,i-1}]}{h}. \quad (45)$$

As usual, the jump term may be taken to be $[u]_i = u_{1,i} - u_{2,i-1}$ such that the ODEs for the second-order ADF-DG approach may be simplified into

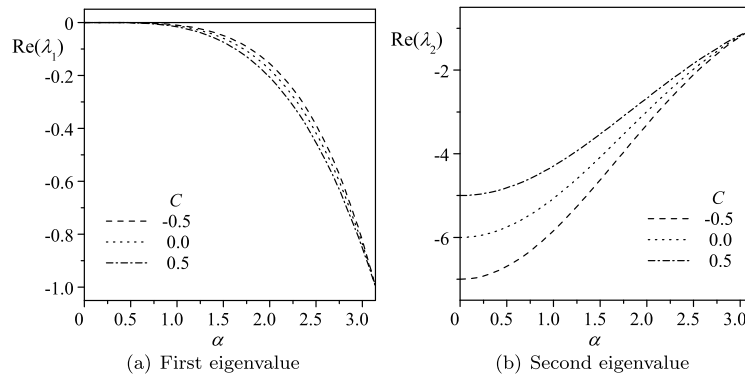


Fig. 5. The real parts of the two eigenvalues corresponding to the coefficient matrix $\mathbf{M}$ for a second-order nodal-based ADF-DG using the constant coefficients of $C = -0.5, 0$, and 0.5 .

$$\begin{cases} \frac{du_{1,i}}{dt} = \frac{a(4u_{2,i-1} - 3u_{1,i} - u_{2,i} + C_1[u]_i)}{h} \\ \frac{du_{2,i}}{dt} = \frac{a(3u_{1,i} - u_{2,i} - 2u_{2,i-1} - C_2[u]_i)}{h} \end{cases} \quad (46)$$

Here too, by taking $C_1 = C_2 = 0$, the second-order ADF-DG formulation may be seen to reproduce the second-order nodal DG expressions identically. Furthermore, the use of higher-order ADF representations of the type $\tilde{\delta}_{1|M}$, $M > 1$, will immediately give rise to additional coefficients and, hence, additional ways to improve the scheme.

For the third-order scheme, the solution on a cell may be synthesized from

$$\tilde{u} = 2\lambda_1(x)[\lambda_1(x) - \frac{1}{2}]u_1 + 4\lambda_1(x)\lambda_2(x)u_2 + 2\lambda_2(x)[\lambda_2(x) - \frac{1}{2}]u_3. \quad (47)$$

As before, the ADF polynomials corresponding to the left side, center, and right side of the element may be subsequently evaluated and consolidated into

$$\begin{aligned} \tilde{\delta}_{2|1,1} &= -\frac{3}{2h} - \frac{6\xi}{h} + \frac{30\xi^2}{h} + \frac{6C}{h} (20\xi^3 - 3\xi), \\ \tilde{\delta}_{2|1,2} &= \frac{9}{4h} - \frac{15\xi^2}{h} + \frac{6C}{h} (20\xi^3 - 3\xi), \\ \tilde{\delta}_{2|1,3} &= -\frac{3}{2h} + \frac{6\xi}{h} + \frac{30\xi^2}{h} + \frac{6C}{h} (20\xi^3 - 3\xi). \end{aligned} \quad (48)$$

These ADF relations enable us to express the updating ODEs as

$$\begin{cases} \frac{du_{1,i}}{dt} = \frac{a(u_{3,i} - 6u_{1,i} + 9u_{3,i-1} - 4u_{2,i} + C_1[u]_i)}{h} \\ \frac{du_{2,i}}{dt} = \frac{a(5/2u_{1,i} - u_{3,i} - 3/2u_{3,i-1} + C_2[u]_i)}{h} \\ \frac{du_{3,i}}{dt} = \frac{a(3u_{3,i-1} + 4u_{2,i} - 3u_{3,i} - 4u_{1,i} + C_3[u]_i)}{h} \end{cases}, \quad (49)$$

where the interface jump term is prescribed by $[u]_i = u_{1,i} - u_{3,i-1}$. As expected, by setting $C_1 = C_2 = C_3 = 0$, the third-order ADF-DG restores the third-order nodal DG formulation.

4.3. Fourier stability verification of the nodal-based ADF-DG method

Here too, the standard Fourier analysis may be carried out for the second-order ADF-based nodal DG using, for example, $C_1 = C_2 = C$. The real parts of the two eigenvalues for the cases of $C = -0.5, 0$, and 0.5 may be retrieved and plotted in Fig. 5. As one may observe, the stability of the scheme is confirmed for the three representative coefficients of C . Moreover, the ability to adjust C in a manner to control the stability properties of the scheme seems to offer an added benefit.

For the third-order ADF-based nodal DG approach, we may take, as before, $C_1 = C_2 = C_3 = C$. The real parts of three eigenvalues of the coefficient matrix $\mathbf{M}$ may be extracted and displayed in Fig. 6 for the three representative cases of $C = -0.5, 0$, and 0.5 . Clearly, the scheme remains stable for all three values of C , although it appears to be slightly diffusive for the first eigenvalue and $C = 0.5$.

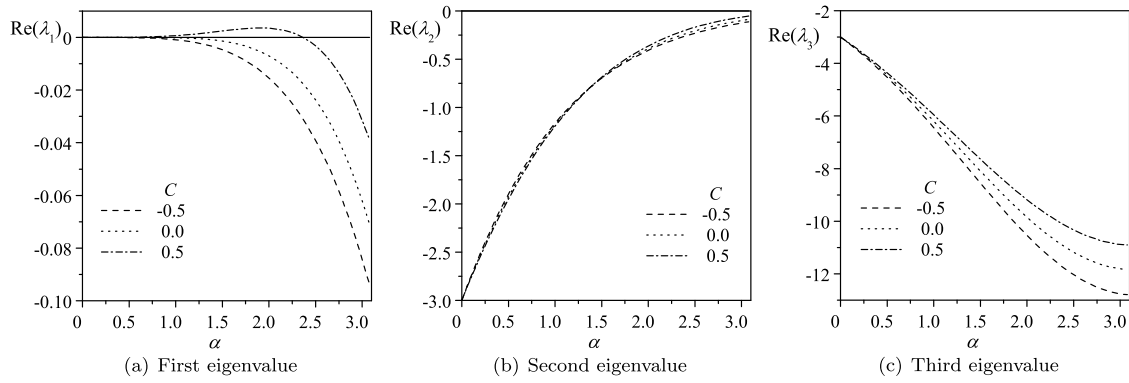


Fig. 6. The real parts of the three eigenvalues corresponding to the coefficient matrix $\mathbf{M}$ for a third-order nodal-based ADF-DG using the constant coefficients of $C = -0.5, 0$, and 0.5 .

5. ADF in relation to the one-dimensional Flux Reconstruction (FR) method

It is well recognized that the Flux-Reconstruction (FR) method and its variant Correction Procedure via Reconstruction (CPR) incorporate into their schemes the nodal DG approach as a special case. Furthermore, both approaches tend to be more efficient to implement than the integral DG form. For example, the main concept in the FR method is to store, on every cell i , the values at the solution points $u_{i,k}$, for $k = 1, \dots, N + 1$, and then reconstruct the flux function $F(x)$ using the flux values at both solution points and boundary interfaces. With this reconstructed $F(x)$, a differential form may be developed to obtain the right-hand side of the updating ODEs. Specifically in FR, the flux function may be reconstructed via

$$F_i(x) = f_i(x) + [\hat{F}_L - f_i(x_L)]g_{LB}(x) + [\hat{F}_R - f_i(x_R)]g_{RB}(x), \quad (50)$$

where $\hat{F}$ denotes the numerical flux at the interface (e.g., upwind-flux), while x_L and x_R refer to the locations of the left and right interfaces, respectively; as for g , it stands for the correction function with indices LB for ‘left boundary’ and RB for ‘right boundary.’ Here $f_i(x)$ may be reconstructed from $f_{i,k} = f(u_{i,k})$ at the solution points.

For the k th-solution point, the right-hand side of the updating ODEs may be formulated as

$$\left. \frac{dF_i}{dx} \right|_k = \left. \frac{df_i}{dx} \right|_k + [\hat{F}_L - f_i(x_L)]g'_{LB}(x_k) + [\hat{F}_R - f_i(x_R)]g'_{RB}(x_k). \quad (51)$$

Then the updating ODEs become simply

$$\frac{du_{i,k}}{dt} = - \left. \frac{dF_i}{dx} \right|_k; \quad \forall i, k. \quad (52)$$

Consequently, based on the ADF concept, the updating ODEs may be readily determined for the solution-point values viz.

$$\frac{du_{i,k}}{dt} = \int_{x_L}^{x_R} \frac{\partial u}{\partial t} \tilde{\delta}(x, x_k) dx = \int_{x_L}^{x_R} f(x) \frac{d\tilde{\delta}(x, x_k)}{dx} dx - \hat{F}_R \tilde{\delta}(x_R, x_k) + \hat{F}_L \tilde{\delta}(x_L, x_k). \quad (53)$$

The foregoing procedure may be viewed as an ADF-FR scheme, as it combines the generic ADF approach with a Flux Reconstruction base solution.

Assuming that on the right-hand side of (53) the reconstruction of $f(x)$ may be obtained from the flux at the solution points $f_{i,k}$, then from (50), it is possible to extract

$$f(x) = F_i(x) - [\hat{F}_L - f_i(x_L)]g_{LB}(x) - [\hat{F}_R - f_i(x_R)]g_{RB}(x). \quad (54)$$

Subsequently, we can put

$$\begin{aligned} \int_{x_L}^{x_R} f(x) \frac{d\tilde{\delta}(x, x_k)}{dx} dx &= \int_{x_L}^{x_R} F(x) \frac{d\tilde{\delta}(x, x_k)}{dx} dx \\ &- [\hat{F}_L - f_i(x_L)] \int_{x_L}^{x_R} g_{LB}(x) \frac{d\tilde{\delta}(x, x_k)}{dx} dx - [\hat{F}_R - f_i(x_R)] \int_{x_L}^{x_R} g_{RB}(x) \frac{d\tilde{\delta}(x, x_k)}{dx} dx. \end{aligned} \quad (55)$$

Then through the use of $F_i(x_L) = \hat{F}_L$, $F_i(x_R) = \hat{F}_R$, we can write

$$\int_{x_L}^{x_R} F(x) \frac{d\tilde{\delta}(x, x_k)}{dx} dx - \hat{F}_R \tilde{\delta}(x_R, x_k) + \hat{F}_L \tilde{\delta}(x_L, x_k) = - \int_{x_L}^{x_R} \frac{dF(x)}{dx} \tilde{\delta}(x, x_k) dx = - \left. \frac{dF(x)}{dx} \right|_{x_k}. \quad (56)$$

With these expressions in hand, the substitution of (55) into (53) leads to

$$\begin{aligned} \frac{du_{i,k}}{dt} &= \int_{x_L}^{x_R} f(x) \frac{d\tilde{\delta}(x, x_k)}{dx} dx - \hat{F}_R \tilde{\delta}(x_R, x_k) + \hat{F}_L \tilde{\delta}(x_L, x_k) \\ &= \int_{x_L}^{x_R} F(x) \frac{d\tilde{\delta}(x, x_k)}{dx} dx - \hat{F}_R \tilde{\delta}(x_R, x_k) + \hat{F}_L \tilde{\delta}(x_L, x_k) \\ &\quad - [\hat{F}_L - f_i(x_L)] \int_{x_L}^{x_R} g_{LB}(x) \frac{d\tilde{\delta}(x, x_k)}{dx} dx - [\hat{F}_R - f_i(x_R)] \int_{x_L}^{x_R} g_{RB}(x) \frac{d\tilde{\delta}(x, x_k)}{dx} dx \\ &= - \left. \frac{dF(x)}{dx} \right|_{x_k} - [\hat{F}_L - f_i(x_L)] \int_{x_L}^{x_R} g_{LB}(x) \frac{d\tilde{\delta}(x, x_k)}{dx} dx - [\hat{F}_R - f_i(x_R)] \int_{x_L}^{x_R} g_{RB}(x) \frac{d\tilde{\delta}(x, x_k)}{dx} dx. \end{aligned} \quad (57)$$

According to the FR method, in the case of $N+1$ solution points, $g_{LB}(x)$ becomes a polynomial of order $(N+1)$ that satisfies

$$g_{LB}(x_L) = 1 \quad \text{and} \quad g_{LB}(x_R) = 0. \quad (58)$$

In the context of the FR-DG approach, the $g_{LB}(x)$ polynomial is taken to be orthogonal to $P_{N-1}(x)$.

Presently, for the N th-order $u(x)$, the appropriate ADF will be $\tilde{\delta}_{N|M}(M \geq 0)$. Then if we use $\tilde{\delta}_N(x, x_k) = \tilde{\delta}_{N|0}(x, x_k)$, which represents an N th-order polynomial, the derivative $d\tilde{\delta}_N(x, x_k)/dx$ becomes an $(N-1)$ th-order polynomial. This enables us to put

$$\int_{x_L}^{x_R} g_{LB}(x) \frac{d\tilde{\delta}_N(x, x_k)}{dx} dx = \int_{x_L}^{x_R} g_{RB}(x) \frac{d\tilde{\delta}_N(x, x_k)}{dx} dx = 0, \quad (59)$$

whence, based on (57), we have

$$\frac{du_{i,k}}{dt} = - \left. \frac{dF_i}{dx} \right|_k. \quad (60)$$

It can thus be seen that the present ADF-FR formulation gives rise to the same FR-DG expression.

When considering other cases where $g_{LB}(x)$ is no longer orthogonal to P_{N-1} , the flexibility of $\tilde{\delta}_{N|M}(x, x_k)$, which is enhanced by its inclusion of M arbitrary coefficients, enables us to solve for the coefficients from the linear integrals

$$\int_{x_L}^{x_R} g_{LB}(x) \frac{d\tilde{\delta}_{N|M}(x, x_k)}{dx} dx = 0, \quad \int_{x_L}^{x_R} g_{RB}(x) \frac{d\tilde{\delta}_{N|M}(x, x_k)}{dx} dx = 0. \quad (61)$$

This straightforward evaluation allows the complete determination of $\tilde{\delta}_{N|M}(x, x_k)$ in a manner that reproduces the corresponding FR formulation identically. However, by selecting $\tilde{\delta}_{N|M}(x, x_k)$ differently, it is possible to devise schemes that differ from the FR method, namely, in their ability to incorporate additional capabilities. Moreover, extensions of this ADF-FR method to two and three dimensions can be accomplished with similar ease, thus leading to identical conclusions to those realized here. In short, the ability of the ADF-FR method to extend the FR and CPR approaches is plausible.

6. Point-value enhanced finite volume method (PFV)

Based on the foregoing discussions, it may be safely argued that approximate delta functions may be viewed as effective tools for deriving the updating ODEs associated with different schemes through the use of a rigorous mathematical formalism. In this context, it is clear that the ADF approach offers a fresh and efficient strategy for updating the unknown information associated with a given element irrespective of the underlying scheme. Another advantage of this approach stands in its simplicity and flexibility in the placement of DOFs at any desired point in the element such as the nodal points and edges.

Since nodes undergo the highest sharing rate, the assignment of additional information at the nodes maximizes the number of neighbors with whom this information may be communicated. When information is added at one node, it is systematically shared by all of the surrounding nodes, thus leading to one of the most effective DOF strategies. For example,

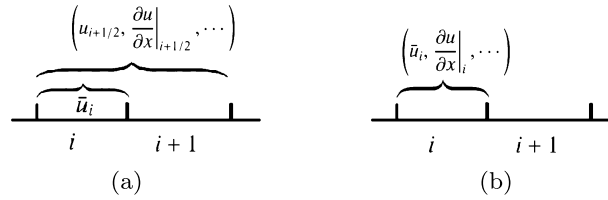


Fig. 7. Comparison between the DOF arrangement and integral domain of (a) the Point-value enhanced Finite Volume (PFV) method and (b) the Taylor-based DG method.

in the case of a triangular mesh, the addition of a single quantity at the nodes of one element enables us to immediately gain three values for this element. Furthermore, to ensure the scheme's conservation, the cell-averaged values are stored and updated identically to the manner used by the finite volume method.

Presently, both nodal values and derivatives are introduced to enhance the scheme's overall accuracy by achieving a high-order and compact representation of the unknown quantity in each element. To emphasize this property, we dub this scheme "Point-value enhanced Finite Volume (PFV)." The PFV method can, therefore, be compact, high-order, and stable, because of the continuity of the solution being enhanced by the sharing of nodal information.

Similar strategies may be seen in the multi-moment finite volume method (see [19] and the references therein), as well as the active flux method [20,21]. In these, however, the values, without the derivatives, are stored and updated on the edges and, in some cases, the nodes. The distinguishing characteristic of these approaches rests in the manner by which the updating of point information, values and derivatives can be performed. Unlike the differential form used by Xie et al. [19], the PFV relies on approximate delta functions to derive the updating ODEs for the additional DOFs at each point.

Simple sketches that compare the DOF arrangement and integral domain within the PFV to those associated with the Taylor-based DG method are furnished in Fig. 7. Although both PFV and Taylor-based DG methods seek to update the cell-averaged value in each element consistently with the finite volume method, the updating of derivatives is performed quite differently. In the PFV case, the weak integral form is implemented in the region $[x_{i-1/2}, x_{i+3/2}]$, while in DG method, the integration domain is restricted to $[x_{i-1/2}, x_{i+1/2}]$. The doubling of the domain width as well as the overlapping of PFV's integration interval enhances its stability relative to the DG scheme. As to the element-wise unknown reconstruction, the Taylor-based DG method possesses element-wise DOFs of the type $\{\bar{u}, \partial u/\partial x, \dots\}_i$. In contrast, every element within the PFV scheme carries multiple DOF quantities, namely, $\{u, \partial u/\partial x, \dots\}_{i-1/2}$, $\bar{u}_i$, and $\{u, \partial u/\partial x, \dots\}_{i+1/2}$. Another avenue for comparison is this. If we were to compare the DG scheme with $N \times$ DOFs in every element to a PFV formulation taken at the same order on the nodes, the PFV would possess $(2N + 1) \times$ DOFs for each element. It can hence be seen that for the same number (or order) of DOFs in one-dimensional space, the PFV strategy can double the order of the scheme compared to the DG formulation. In fact, the same argument not only holds but can be more beneficial when contemplating the extension of this approach to multiple dimensions.

In the interest of clarity, we refer to the PFV approach as P_0 FV when only the unknown value is stored at the nodal point. As such, the term P_n FV may be used when the coefficients of an n th-order polynomial are saved on each nodal point. Accordingly, the use of P_1 FV will imply that both the unknown quantity u and its first-order derivative are saved on the nodal point. Along similar lines, the term P_n FV $_m$ will be used to describe the approach in the presence of an m th-order polynomial reconstruction within the element. The m th-order polynomial in the element is typically reconstructed using the method of least squares and information that is stored on the cell and nodal points. At times, a weighted least-squares method may be implemented to ensure a suitable reconstruction while providing an additional avenue for controlling stability.

6.1. P_0 FV for the linear, one-dimensional, scalar wave equation

To illustrate the application of the P_0 FV, we begin by considering a classical example, namely, that of the linear wave equation in one-dimensional space as given by (18). To solve this hyperbolic equation, we construct a uniform mesh, store and update the cell-averaged values $\bar{u}_i$, as well as the point values $u_{i+1/2}$ at the interfaces. Based on the P_0 FV construct, the updating equations become

$$\begin{aligned} \frac{d\bar{u}_i}{dt} &= -\frac{\hat{F}_{i+1/2} - \hat{F}_{i-1/2}}{h}, \\ \frac{du_{i+1/2}}{dt} &= \int_{x_{i-1/2}}^{x_{i+3/2}} f(x) \frac{d\tilde{\delta}(x, x_{i+1/2})}{dx} dx - \hat{F}_{i+3/2} \tilde{\delta}(x_{i+3/2}, x_{i+1/2}) + \hat{F}_{i-1/2} \tilde{\delta}(x_{i-1/2}, x_{i+1/2}). \end{aligned} \quad (62)$$

6.1.1. Second-order P_0 FV $_1$ formulation

At every cell i , the solution may be expressed using a first-order polynomial, viz.

$$\bar{u}_i(x) = \bar{u}_i + \frac{u_{i+1/2} - u_{i-1/2}}{h}(x - x_i). \quad (63)$$

Then at the interface, $i + 1/2$, the left and right-hand side values may be computed using

$$u_{i+1/2}^L = \tilde{u}_i(x_{i+1/2}), \quad u_{i+1/2}^R = \tilde{u}_{i+1}(x_{i+1/2}). \quad (64)$$

Subsequently, the upwind numerical flux may be consolidated into

$$\hat{F}_{i+1/2} = \frac{1}{2}[au_{i+1/2}^L + au_{i+1/2}^R - |a|(u_{i+1/2}^R - u_{i+1/2}^L)], \quad (65)$$

and then inserted into (62). Because $\tilde{u}_i(x)$ is linear, we may use $\tilde{\delta}_1(x) = \frac{1}{2}h^{-1}$ for the updating of the point values. In this case, the domain integral in (62) vanishes by virtue of $d\tilde{\delta}_1/dx = 0$. As such, the updating expression for the point values simplifies into

$$\frac{du_{i+1/2}}{dt} = -\frac{\hat{F}_{i+3/2} - \hat{F}_{i-1/2}}{2h}. \quad (66)$$

6.1.2. Third-order P_0FV_2 formulation

For every cell i , the solution may be synthesized from $(\tilde{u}_{i-1}, \tilde{u}_i, \tilde{u}_{i+1})$ and $(u_{i-1/2}, u_{i+1/2})$, using the method of least squares. Accordingly, the solution in each cell may be written as

$$\tilde{u}_i(\xi) = \tilde{u}_i + a_1\xi + a_2\left(\frac{1}{2}\xi^2 - \frac{1}{24}\right). \quad (67)$$

Hence, the problem becomes that of minimizing the total deviation from the cell-averaged value and the values at the interface and neighboring cells. By taking

$$I_0 = [\tilde{u}_i(-1/2) - u_{i-1/2}]^2 + [\tilde{u}_i(1/2) - u_{i+1/2}]^2 + \left[\int_{-3/2}^{-1/2} \tilde{u}_i(\xi) d\xi - \tilde{u}_{i-1} \right]^2 + \left[\int_{1/2}^{3/2} \tilde{u}_i(\xi) d\xi - \tilde{u}_{i+1} \right]^2, \quad (68)$$

one obtains the linear system

$$\frac{\partial I_0}{\partial a_1} = 0, \quad \frac{\partial I_0}{\partial a_2} = 0. \quad (69)$$

Subsequently, by solving (69), the reconstructed second-order polynomial takes the form

$$\begin{aligned} \tilde{u}_i(\xi) = & \tilde{u}_i + \frac{u_{i+1/2} - u_{i-1/2} + 2(\tilde{u}_{i+1} - \tilde{u}_{i-1})}{5}\xi \\ & + \frac{36(\tilde{u}_{i+1} + \tilde{u}_{i-1}) + 6(u_{i+1/2} + u_{i-1/2}) - 84\tilde{u}_i}{37}\left(\frac{\xi^2}{2} - \frac{1}{24}\right), \end{aligned} \quad (70)$$

where $\xi = (x - x_i)/h$. Then at the interface, $i + 1/2$, the left and right-hand sides values may be evaluated from

$$u_{i+1/2}^L = \tilde{u}_i(x_{i+1/2}) \quad \text{and} \quad u_{i+1/2}^R = \tilde{u}_{i+1}(x_{i+1/2}). \quad (71)$$

The corresponding upwind-based numerical flux becomes

$$\hat{F}_{i+1/2} = \frac{1}{2}[au_{i+1/2}^L + au_{i+1/2}^R - |a|(u_{i+1/2}^R - u_{i+1/2}^L)]. \quad (72)$$

In this case, the ADF formulation yields

$$\tilde{\delta}_2(x, x_{i+1/2}) = \frac{9}{8h} - \frac{15}{8h^3}(x - x_{i+1/2})^2. \quad (73)$$

When (73) is substituted into the domain integral in (62), it may be evaluated separately on each cell by taking

$$\int_{x_{i-1/2}}^{x_{i+3/2}} f(x) \frac{d\tilde{\delta}_2}{dx} dx = \int_{x_{i-1/2}}^{x_{i+1/2}} f(x) \frac{d\tilde{\delta}_2}{dx} dx + \int_{x_{i+1/2}}^{x_{i+3/2}} f(x) \frac{d\tilde{\delta}_2}{dx} dx, \quad (74)$$

which can be readily computed using a two-point Gaussian–Legendre quadrature on each cell. In this exercise, the two Gaussian points consist of $\xi_{1,2} = \pm \frac{\sqrt{3}}{6}$, and the corresponding weight equals $\frac{1}{2}h$. For each cell i , the Gaussian point corresponds to $x_k = \xi_k h + x_i$, and so the derivatives and integrals over the cell domain may be evaluated sequentially using

$$\left. \frac{d\tilde{\delta}_2}{dx} \right|_{i,k} = -\frac{15}{4h^2}(\xi_k - 1/2), \quad \left. \frac{d\tilde{\delta}_2}{dx} \right|_{i+1,k} = -\frac{15}{4h^2}(\xi_k + 1/2), \quad (75)$$

and so

$$\int_{x_{i-1/2}}^{x_{i+3/2}} f(x) \frac{d\tilde{\delta}_2}{dx} dx = -\frac{15}{8h} [(\xi_1 - 1/2)f_{i,1} + (\xi_2 - 1/2)f_{i,2} + (\xi_1 + 1/2)f_{i+1,1} + (\xi_2 + 1/2)f_{i+1,2}], \quad (76)$$

where $f_{i,1} = f[\tilde{u}_i(\xi_1)]$, and $f_{i,2} = f[\tilde{u}_i(\xi_2)]$ represent the fluxes at the Gaussian points in element i . Forthwith, the updating equation for $u_{i+1/2}$ reduces to

$$\frac{du_{i+1/2}}{dt} = -\frac{15}{8h} [(\xi_1 - 1/2)f_{i,1} + (\xi_2 - 1/2)f_{i,2} + (\xi_1 + 1/2)f_{i+1,1} + (\xi_2 + 1/2)f_{i+1,2}] + \frac{3}{4h} (\hat{F}_{i+3/2} - \hat{F}_{i-1/2}), \quad (77)$$

6.1.3. Fourth-order P_0FV_3 formulation

At every cell i , the solution may be deduced, as usual, from $(\bar{u}_{i-1}, \bar{u}_i, \bar{u}_{i+1})$ and $(u_{i-1/2}, u_{i+1/2})$, using the method of least squares. As before, by writing the solution in each cell as

$$\tilde{u}_i(\xi) = \bar{u}_i + a_1\xi + a_2\left(\frac{1}{2}\xi^2 - \frac{1}{24}\right) + a_3\xi^3, \quad (78)$$

minimizing the total deviation between the cell value and the values at the interface and neighboring cells can be captured through

$$I_0 = [\tilde{u}_i(-1/2) - u_{i-1/2}]^2 + [\tilde{u}_i(1/2) - u_{i+1/2}]^2 + \left[\int_{-3/2}^{-1/2} \tilde{u}_i(\xi) d\xi - \bar{u}_{i-1} \right]^2 + \left[\int_{1/2}^{3/2} \tilde{u}_i(\xi) d\xi - \bar{u}_{i+1} \right]^2. \quad (79)$$

Minimizing the deviation will hence require setting

$$\frac{\partial I_0}{\partial a_1} = 0, \quad \frac{\partial I_0}{\partial a_2} = 0, \quad \frac{\partial I_0}{\partial a_3} = 0. \quad (80)$$

By solving the linear system given by (80), one arrives at a third-order polynomial of the form

$$\begin{aligned} \tilde{u}_i(\xi) = & \bar{u}_i - \frac{\bar{u}_{i+1} - \bar{u}_{i-1} - 10(u_{i+1/2} - u_{i-1/2})}{8} \xi \\ & + \frac{36(\bar{u}_{i+1} + \bar{u}_{i-1}) + 6(u_{i+1/2} + u_{i-1/2}) - 84\bar{u}_i}{37} \left(\frac{\xi^2}{2} - \frac{1}{24} \right) \\ & + \frac{3(\bar{u}_{i+1} - \bar{u}_{i-1}) - 2(u_{i+1/2} - u_{i-1/2})}{6} \xi^3. \end{aligned} \quad (81)$$

As for the flux at the interface, its computation may be accomplished using a procedure that mirrors our evaluation at previous orders. At this particular order, however, two ADFs, namely, $\tilde{\delta}_2(x, x_{i+1/2}) = \tilde{\delta}_3(x, x_{i+1/2})$, must be used in conjunction with three Gaussian points in order to suitably resolve the domain integrals. To proceed, these quantities may be substituted into the domain integrand of (62) and evaluated term-by-term to produce

$$\int_{x_{i-1/2}}^{x_{i+3/2}} f(x) \frac{d\tilde{\delta}_3}{dx} dx = -\frac{15}{8h} \left[\sum_{k=1}^3 w_k(\xi_k - 1/2)f_{i,k} + \sum_{k=1}^3 w_k(\xi_k + 1/2)f_{i+1,k} \right], \quad (82)$$

where the Gaussian weights consist of $w_1 = w_3 = \frac{5}{9}$, and $w_2 = \frac{8}{9}$, while the Gaussian points comprise $\xi_{1,3} = \pm \frac{\sqrt{15}}{10}$, and $\xi_2 = 0$. At this fourth order, the updating equation for $u_{i+1/2}$ may be rearranged into

$$\frac{du_{i+1/2}}{dt} = -\frac{15}{8h} \left[\sum_{k=1}^3 w_k(\xi_k - 1/2)f_{i,k} + \sum_{k=1}^3 w_k(\xi_k + 1/2)f_{i+1,k} \right] + \frac{3}{4h} (\hat{F}_{i+3/2} - \hat{F}_{i-1/2}). \quad (83)$$

6.2. P_1FV for the linear, one-dimensional, scalar wave equation

In this case, apart from the two equations that arise in (62), a third relation will be necessary in order to adequately update the first-order derivative on each node. This expression is

$$\frac{dv_{i+1/2}}{dt} = \int_{x_{i-1/2}}^{x_{i+3/2}} hf(x) \frac{d\tilde{\delta}'(x, x_{i+1/2})}{dx} dx - h\hat{F}_{i+3/2}\tilde{\delta}'(x_{i+3/2}, x_{i+1/2}) + h\hat{F}_{i-1/2}\tilde{\delta}'(x_{i-1/2}, x_{i+1/2}), \quad (84)$$

where

$$v_{i+1/2} = h \left. \frac{du}{dx} \right|_{i+1/2}. \quad (85)$$

6.2.1. Third-order P_1FV_2 formulation

As we gradually increase the order, the solution at every cell can be reconstructed from the quantities and derivatives, $\bar{u}_i$ and $(u_{i-1/2}, u_{i+1/2}, v_{i-1/2}, v_{i+1/2})$, retrieved from the method of weighted least squares. In this case, the solution in each cell may be expanded into

$$\tilde{u}_i(\xi) = \bar{u}_i + a_1\xi + a_2\left(\frac{1}{2}\xi^2 - \frac{1}{24}\right), \quad (86)$$

and so the distance to the values and derivatives at interfaces may be expressed as

$$I_0 = s[\tilde{u}_i(-1/2) - u_{i-1/2}]^2 + s[\tilde{u}_i(1/2) - u_{i+1/2}]^2 + \left[\frac{d\tilde{u}_i}{d\xi}(-1/2) - v_{i-1/2}\right]^2 + \left[\frac{d\tilde{u}_i}{d\xi}(1/2) - v_{i+1/2}\right]^2. \quad (87)$$

Naturally, minimizing the total deviation requires taking

$$\frac{\partial I_0}{\partial a_1} = 0, \quad \frac{\partial I_0}{\partial a_2} = 0. \quad (88)$$

Thus by solving (88), the reconstructed second-order polynomial becomes

$$\begin{aligned} \tilde{u}_i(\xi) = & \bar{u}_i + \frac{s(u_{i+1/2} - u_{i-1/2}) + 2(v_{i+1/2} + v_{i-1/2})}{4 + s}\xi \\ & + \frac{6[su_{i+1/2} + su_{i-1/2} - 2s\bar{u}_i + 6(v_{i+1/2} - v_{i-1/2})]}{36 + s}\left(\frac{\xi^2}{2} - \frac{1}{24}\right), \end{aligned} \quad (89)$$

where $s = 10^{-6}$ denotes the least-squares weight associated with the nodal values. The remaining procedure to obtain the updating ODE for $u_{i+1/2}$ proves to be identical to that already developed for P_0FV_2 . To illustrate the manner by which the updating of $v_{i+1/2}$ may be accomplished, we consider the ADF polynomial for the derivative weight function $\tilde{\delta}'_2(x)$, namely,

$$\tilde{\delta}'_2(x) = \frac{3x}{2h^3}. \quad (90)$$

In this case, two Gaussian points will be necessary in the evaluation of the domain integral. After some algebra, we retrieve the updating equation for $v_{i+1/2}$, namely,

$$\frac{dv_{i+1/2}}{dt} = \frac{3}{4h} \left[\sum_{k=1}^2 w_k(\xi_k - 1/2)f_{i,k} + \sum_{k=1}^2 w_k(\xi_k + 1/2)f_{i+1,k} \right] - \frac{3}{2h}(\hat{F}_{i+3/2} + \hat{F}_{i-1/2}), \quad (91)$$

where the Gaussian weights may be set at unity with $w_1 = w_2 = 1$.

6.2.2. Fourth-order P_1FV_3 formulation

The highest order that we will describe in this study consists of using the method of weighted least-squares at every cell i for the values of $(\bar{u}_{i-1}, \bar{u}_i, \bar{u}_{i+1})$ and $(u_{i-1/2}, u_{i+1/2}, v_{i-1/2}, v_{i+1/2})$ to arrive at a fourth-order formulation. For the solution in each cell

$$\tilde{u}_i(\xi) = \bar{u}_i + a_1\xi + a_2\left(\frac{1}{2}\xi^2 - \frac{1}{24}\right) + a_3\xi^3, \quad (92)$$

the total deviation that must be minimized can be estimated from

$$I_0 = s[\tilde{u}_i(-1/2) - u_{i-1/2}]^2 + s[\tilde{u}_i(1/2) - u_{i+1/2}]^2 + \left[\frac{d\tilde{u}_i}{d\xi}(-1/2) - v_{i-1/2}\right]^2 + \left[\frac{d\tilde{u}_i}{d\xi}(1/2) - v_{i+1/2}\right]^2, \quad (93)$$

and so, its extrema may be found by taking

$$\frac{\partial I_0}{\partial a_1} = 0, \quad \frac{\partial I_0}{\partial a_2} = 0, \quad \frac{\partial I_0}{\partial a_3} = 0. \quad (94)$$

After solving (94), the reconstructed third-order polynomial can be written as

$$\begin{aligned} \tilde{u}_i(\xi) = & \bar{u}_i + \frac{6(u_{i+1/2} - u_{i-1/2}) - (v_{i+1/2} + v_{i-1/2})}{4}\xi \\ & + \frac{6[su_{i+1/2} + su_{i-1/2} - 2s\bar{u}_i + 6(v_{i+1/2} - v_{i-1/2})]}{36 + s}\left(\frac{\xi^2}{2} - \frac{1}{24}\right) \\ & + \frac{3(\bar{u}_{i+1} - \bar{u}_{i-1}) - 2(u_{i+1/2} - u_{i-1/2})}{6}\xi^3, \end{aligned} \quad (95)$$

where $s = 10^{-6}$ stands for the least-squares weight ascribed to the nodal values. We replace a_3 from (81), which makes scheme stable. Here too, three Gaussian points may be effectively used following the procedure that we developed for the P_0FV_3 formulation. At the outset, the updating equation for $v_{i+1/2}$ may be extracted and compacted into

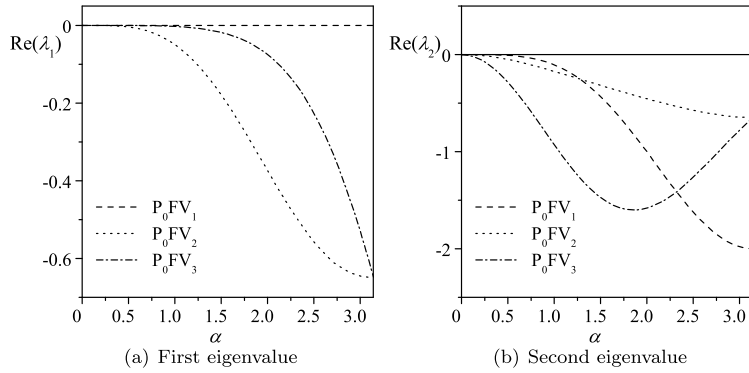


Fig. 8. The real parts of the two eigenvalues of the coefficient matrix $\mathbf{M}$ corresponding to P_0FV_n ($n = 1, 2, 3$).

$$\frac{dv_{i+1/2}}{dt} = \frac{15}{16h} \left[\sum_{k=1}^3 w_k [5 - 21(\xi_k - 1/2)^2] f_{i,k} + \sum_{k=1}^3 w_k [5 - 21(\xi_k + 1/2)^2] f_{i+1,k} \right] + \frac{15}{4h} (\hat{F}_{i+3/2} + \hat{F}_{i-1/2}). \quad (96)$$

6.3. Fourier stability verification of the PFV method

To further confirm the viability of the PFV method, a Fourier stability analysis is carried out as in the case of ADF-DG method. Here the unknown vectors are specified as $\mathbf{U}_j = [\bar{u}_j, u_{j+1/2}]^T$ and $\mathbf{U}_j = [\bar{u}_j, u_{j+1/2}, v_{j+1/2}]^T$ for the P_0FV_n and P_1FV_n , respectively. As before, the coefficient matrices $\mathbf{M}$ at different orders may be readily evaluated along with the real parts of the corresponding eigenvalues. In what follows, we use Fig. 8 to display the real parts of the two eigenvalues for P_0FV_n , and Fig. 9 to provide the real parts of the three eigenvalues for P_1FV_n . In all cases considered, it may be seen that the P_0FV_n scheme remains stable in its half-discretized form. The stability of the P_1FV_2 solution is also confirmed, although P_1FV_3 exhibits a small region of α with weakly positive eigenvalues. This instability may be readily suppressed by the TVD Runge–Kutta time-matching technique, which is known for its effectiveness in stabilizing the scheme. The ensuing behavior will be illustrated in the forthcoming numerical examples.

6.4. Numerical verification

In order to test the accuracy and stability of the PFV scheme, two benchmark cases will be considered: the linear wave equation given by (18) with $a = 1$ as well as the nonlinear Burgers' equation. In this process, periodic boundary conditions will be imposed at both ends of the computational domain, which will be bracketed over the interval $[0, 1]$. Two different initial conditions will be tested in the context of the linear wave equation by specifying two forms of the initial function, $u_0(x)$. For simplicity, a uniform mesh is considered. Subsequently, to verify the successive orders of the PFV scheme in the context of the one-dimensional wave equation, the number of grid points will be taken to be 8, 16, 32, and 64, respectively. In analyzing Burgers' equation, an additional stencil of 128 points is considered to fully confirm the convergence order. To quantify the error, we define the residual to be $R_i = |u_{\text{exact}} - u_{\text{numeric}}|$ and use both the $L_1 = \sum R_i dx$ and $L_\infty = \max R_i$ error measurements of the cell-averaged values to quantify the error. Furthermore, a third-order TVD-Runge–Kutta method is relied upon for time matching.

6.5. Linear wave equation with an initial function of $u_0(x) = \sin(2\pi x)$

Using $u_0(x) = \sin(2\pi x)$, the problem is run for one period of time, until $t = 1$. The errors entailed in the PFV scheme at different orders are cataloged and compared in Tables 1, 2, and 3. Everywhere, h denotes the grid step size.

We refer the reader to Fig. 10 where the left and right subsets provide a useful comparison of the base 10 logarithms, $\lg(L_1)$ and $\lg(L_\infty)$, at different PFV orders. Based on these findings, it may be ascertained that the intended orders are secured. Nonetheless, P_0FV_3 appears to slightly under-perform its projected fourth order, unlike the P_1FV_2 result which displays an accelerated convergence rate. These findings demonstrate that, when taken at the same order n , the P_1FV_n formulation outperforms the P_0FV_n result in the attainable degree of precision.

In these numerical experiments, the critical CFL, which represents the maximum CFL number to keep the scheme stable, is computed numerically, although it can be alternatively determined using Fourier analysis. Specifically for the case of 16 grid points, the code is executed in time up to $t = 1000$. In principle, the scheme is deemed stable when the maximum value of u over the entire domain, $x \in [0, 1]$, continues to depreciate with the passage of time. This test also enables us to determine the critical CFL, which is evaluated and posted in Table 4. Based on these results, it may be seen that the PFV scheme is highly stable. Another characteristic that may be inferred from these results concerns the critical CFL, which does not decrease monotonically with successive increases in the order of the solver, contrarily to its variation within most other methods.

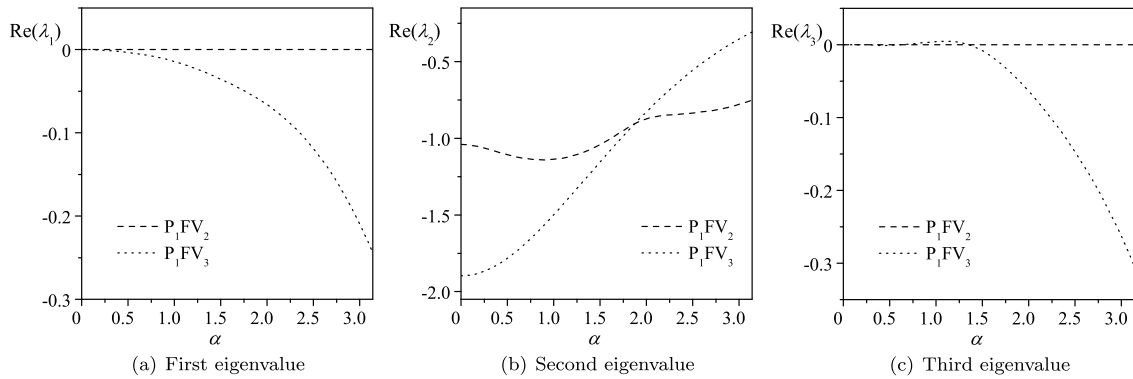


Fig. 9. The real parts of the three eigenvalues of the coefficient matrix $\mathbf{M}$ corresponding to P_1FV_n ($n = 2, 3$).

Table 1

Characteristics of the linear wave equation with $a = 1$, $u_0 = \sin(2\pi x)$, and $t \in [0, 1]$. The decimal logarithms, $\lg(L_1)$ and $\lg(L_\infty)$, refer to the errors entailed in the second-order P_0FV_1 scheme.

$\lg(h)$	$\lg(L_1)$	Order	$\lg(L_\infty)$	Order
−0.903	−0.685	–	−0.480	–
−1.204	−1.278	1.98	−1.076	1.98
−1.505	−1.886	2.02	−1.689	2.04
−1.806	−2.492	2.01	−2.295	2.01

Table 2

Characteristics of the linear wave equation with $a = 1$, $u_0 = \sin(2\pi x)$, and $t \in [0, 1]$. Here $\lg(L_1)$ and $\lg(L_\infty)$ refer to the third-order P_0FV_2 and P_1FV_2 errors.

$\lg(h)$	P_0FV_2		P_1FV_2		P_1FV_2		P_1FV_2	
	$\lg(L_1)$	Order	$\lg(L_\infty)$	Order	$\lg(L_1)$	Order	$\lg(L_\infty)$	Order
−0.903	−0.980	–	−0.771	–	−1.153	–	−0.952	–
−1.204	−1.785	2.68	−1.595	2.90	−2.271	3.71	−2.077	3.74
−1.505	−2.665	2.92	−2.468	2.90	−3.452	3.92	−3.258	3.92
−1.806	−3.539	2.90	−3.343	2.90	−4.648	3.98	−4.452	3.97

Table 3

Characteristics of the linear wave equation with $a = 1$, $u_0 = \sin(2\pi x)$, and $t \in [0, 1]$. Here $\lg(L_1)$ and $\lg(L_\infty)$ refer to the errors in the fourth-order P_0FV_3 and P_1FV_3 schemes.

$\lg(h)$	P_0FV_3		P_1FV_3		P_1FV_3		P_1FV_3	
	$\lg(L_1)$	Order	$\lg(L_\infty)$	Order	$\lg(L_1)$	Order	$\lg(L_\infty)$	Order
−0.903	−2.485	–	−2.304	–	−2.777	–	−2.606	–
−1.204	−3.556	3.56	−3.366	3.53	−4.099	4.39	−3.908	4.33
−1.505	−4.614	3.51	−4.419	3.50	−5.356	4.18	−5.161	4.16
−1.806	−5.714	3.65	−5.518	3.65	−6.579	4.06	−6.383	4.06

6.6. Linear wave equation with an initial function of $u_0(x) = \sin^2(2\pi x)$

In comparison to the previous case, using an initial condition of $u_0(x) = \sin^2(2\pi x)$ leads to a more complex distribution, namely, one that engenders more extrema. By running the problem over one period of time, $t = 1$, the errors at different PFV orders are collected and compared in Tables 5, 6, and 7. Furthermore, the left and right subsets of Fig. 11 compare the logarithmic $\lg(L_1)$ and $\lg(L_\infty)$ variations at successive PFV orders.

Upon close examination of the tabular and graphical results, it may be confidently ascertained that the expected orders have been very closely reproduced. Two exceptions may be noted, as the P_0FV_3 solution appears at a slightly lower order than the expected fourth order, whereas the P_1FV_2 solution converges more rapidly than expected. Here too, for the same order scheme, the P_1FV_n approximation displays a better degree of precision than its P_0FV_n counterpart.

6.7. Burgers equation with an initial function of $u_0(x) = \frac{1}{2} + \sin(2\pi x)$

To further verify the consistent accuracy of the PFV approach in the solution of nonlinear equations, the standard Burgers' equation is considered with an initial function of $u_0(x) = \frac{1}{2} + \sin(2\pi x)$. This hyperbolic PDE may be written as

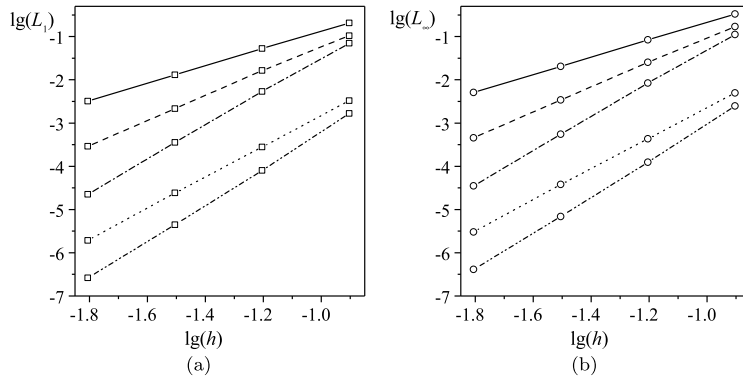


Fig. 10. Convergence behavior of the PFV errors at differently specified orders. Results correspond to the linear wave equation with $a = 1$ and $u_0 = \sin(2\pi x)$ at $t = 1$. Here we show (a) $\lg(L_1)$ ($\square$) and (b) $\lg(L_\infty)$ ($\circ$) for the errors in the P_nFV_m scheme at successive levels of accuracy: P_0FV_1 (—), P_0FV_2 (---), P_1FV_2 (- · -), P_0FV_3 (·····), and P_1FV_3 (- - - -).

Table 4

Critical CFL using the PFV scheme at different orders in conjunction with the linear wave equation with $a = 1$, $u_0 = \sin(2\pi x)$, and a time resolution leading up to $t = 1000$.

	P_0FV_1	P_0FV_2	P_0FV_3	P_1FV_2	P_1FV_3
CFL	1.20	1.61	0.88	1.67	0.74

Table 5

Characteristics of the linear wave equation with $a = 1$, $u_0 = \sin^2(2\pi x)$, and $t \in [0, 1]$. Here $\lg(L_1)$ and $\lg(L_\infty)$ correspond to the errors in the second-order P_0FV_1 scheme.

$\lg(h)$	$\lg(L_1)$	Order	$\lg(L_\infty)$	Order
-0.903	-0.456	-	-0.379	-
-1.204	-0.767	1.03	-0.549	0.56
-1.505	-1.285	1.72	-1.087	1.49
-1.806	-1.887	2.00	-1.690	2.00

Table 6

Characteristics of the linear wave equation with $a = 1$, $u_0 = \sin^2(2\pi x)$, and $t \in [0, 1]$. Here $\lg(L_1)$ and $\lg(L_\infty)$ refer to the third-order P_0FV_2 and P_1FV_2 errors.

$\lg(h)$	P_0FV_2				P_1FV_2			
	$\lg(L_1)$	Order	$\lg(L_\infty)$	Order	$\lg(L_1)$	Order	$\lg(L_\infty)$	Order
-0.903	-0.489	-	-0.412	-	-0.427	-	-0.362	-
-1.204	-1.020	1.76	-0.804	1.30	-1.148	2.40	-0.954	1.97
-1.505	-1.790	2.56	-1.600	2.64	-2.265	3.71	-2.070	3.70
-1.806	-2.666	2.91	-2.470	2.89	-3.449	3.93	-3.253	3.93

Table 7

Characteristics of the linear wave equation with $a = 1$, $u_0 = \sin^2(2\pi x)$, and $t \in [0, 1]$. Here $\lg(L_1)$ and $\lg(L_\infty)$ correspond to the errors in the fourth-order P_0FV_3 and P_1FV_3 schemes.

$\lg(h)$	P_0FV_3				P_1FV_3			
	$\lg(L_1)$	Order	$\lg(L_\infty)$	Order	$\lg(L_1)$	Order	$\lg(L_\infty)$	Order
-0.903	-1.301	-	-1.187	-	-1.440	-	-1.438	-
-1.204	-2.417	3.71	-2.266	3.59	-2.775	4.43	-2.601	3.86
-1.505	-3.490	3.56	-3.299	3.43	-4.101	4.40	-3.911	4.35
-1.806	-4.564	3.57	-4.367	3.55	-5.358	4.18	-5.164	4.16

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} \left(\frac{u^2}{2} \right) = 0; \quad x \in [0, 1], \quad (97)$$

with periodic boundary conditions on both sides of the computational domain. To promote a smooth solution, the numerical effort is carried out up to time $t = \frac{1}{4\pi}$. In this case, our reference solution is obtained straightforwardly using the P_1FV_3 formulation in conjunction with a benchmark mesh of 256 grid points. Subsequently, the reference cell-averaged

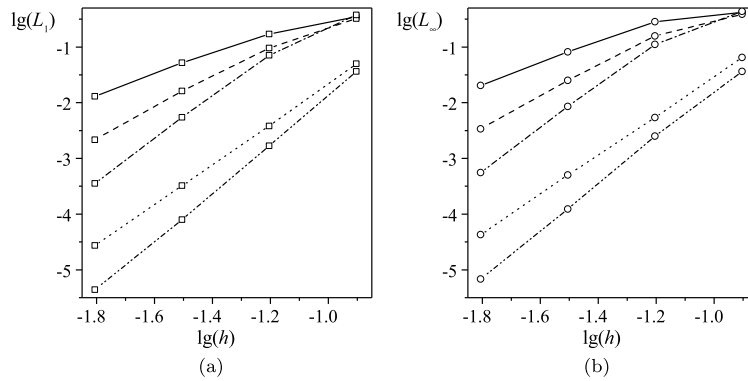


Fig. 11. Convergence behavior of the PFV errors at differently specified orders. Results correspond to the linear wave equation with $a = 1$ and $u_0 = \sin^2(2\pi x)$ at $t = 1$. Here we show (a) $\lg(L_1)$ ($\square$) and (b) $\lg(L_\infty)$ ($\circ$) for the errors in the P_nFV_m scheme at successive levels of accuracy: P_0FV_1 (—), P_0FV_2 (---), P_1FV_2 (- · - ·), P_0FV_3 (·····), and P_1FV_3 (- - - -).

Table 8

Characteristics of the nonlinear Burgers' equation with $u_0(x) = \frac{1}{2} + \sin(2\pi x)$, and $t \in [0, \frac{1}{4}]$. Here $\lg(L_1)$ and $\lg(L_\infty)$ refer to the second-order P_0FV_1 errors.

$\lg(h)$	$\lg(L_1)$	Order	$\lg(L_\infty)$	Order
-0.903	-2.028	-	-1.763	-
-1.204	-2.488	1.53	-1.955	0.64
-1.505	-2.963	1.58	-2.283	1.09
-1.806	-3.516	1.84	-2.768	1.61
-2.107	-4.098	1.93	-3.328	1.86

Table 9

Characteristics of the nonlinear Burgers' equation with $u_0(x) = \frac{1}{2} + \sin(2\pi x)$, and $t \in [0, \frac{1}{4}]$. Here $\lg(L_1)$ and $\lg(L_\infty)$ refer to the third-order P_0FV_2 and P_1FV_2 errors.

lg(h)	P ₀ FV ₂				P ₁ FV ₂			
	lg(L ₁)	Order	lg(L _∞)	Order	lg(L ₁)	Order	lg(L _∞)	Order
−0.903	−2.318	–	−1.786	–	−2.215	–	−1.711	–
−1.204	−3.039	2.39	−2.408	2.07	−2.903	2.28	−2.365	2.17
−1.505	−3.861	2.73	−2.995	1.95	−3.822	3.05	−2.881	1.71
−1.806	−4.738	2.90	−3.832	2.78	−4.789	3.21	−3.811	3.09
−2.107	−5.637	2.99	−4.710	2.92	−5.805	3.38	−4.804	3.30

Table 10

Characteristics of the nonlinear Burgers' equation with $u_0(x) = \frac{1}{2} + \sin(2\pi x)$, and $t \in [0, \frac{1}{4}]$. Here $\lg(L_1)$ and $\lg(L_\infty)$ refer to the fourth-order P_0FV_3 and P_1FV_3 errors.

$\lg(h)$	P_0FV_3				P_1FV_3			
	$\lg(L_1)$	Order	$\lg(L_\infty)$	Order	$\lg(L_1)$	Order	$\lg(L_\infty)$	Order
-0.903	-3.125	-	-2.599	-	-4.043	-	-3.691	-
-1.204	-3.999	2.90	-3.252	2.17	-4.557	1.71	-3.771	0.26
-1.505	-4.797	2.56	-3.904	2.17	-5.474	3.04	-4.417	2.14
-1.806	-5.848	3.50	-4.809	3.01	-6.608	3.77	-5.492	3.57
-2.107	-6.991	3.80	-5.882	3.56	-7.812	4.00	-6.689	3.98

values on coarser meshes are calculated using the benchmark solution. Here the numerical flux at the interface is of the local-Lax-Friedrich (LLF) type, namely,

$$\hat{F}_{i+1/2} = \frac{1}{2} \left[f^L + f^R - \max(|u_{i+1/2}^L|, |u_{i+1/2}^R|) (u_{i+1/2}^R - u_{i+1/2}^L) \right], \quad (98)$$

where $f^L = \frac{1}{2} (u_{i+1/2}^L)^2$ and $f^R = \frac{1}{2} (u_{i+1/2}^R)^2$.

After several numerical runs, the errors accrued in the PFV formulations at varying orders are collected and displayed in [Tables 8, 9, and 10](#). The logarithmic behavior of the errors are also provided in the two subsets of [Fig. 12](#), where $\lg(L_1)$ and $\lg(L_\infty)$ are characterized at progressive PFV orders.

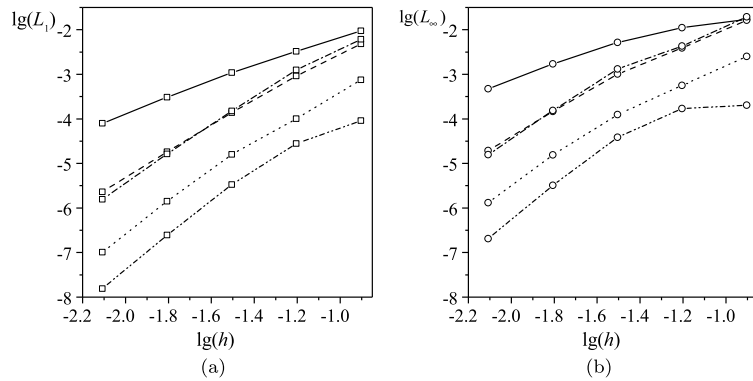


Fig. 12. Convergence of the PFV errors at differently specified orders. Results correspond to the nonlinear Burgers' equation with $u_0 = \frac{1}{2} + \sin(2\pi x)$ at $\pi t = \frac{1}{4}$. Here we show (a) $\lg(L_1)$ ($\square$) and (b) $\lg(L_\infty)$ ($\circ$) for the errors in the P_nFV_m scheme at different orders: P_0FV_1 (—), P_0FV_2 (---), P_1FV_2 (-·-·-), P_0FV_3 (·····), and P_1FV_3 (- - - -).

Despite the increased complexity of Burgers' nonlinear equation, it is gratifying that the projected orders are achieved with a fair degree of precision. As before, two exceptions are noted and these include the P_0FV_3 representation, which evolves at a slightly slower rate than the expected fourth order and, conversely, the P_1FV_2 formulation, which seems to converge more rapidly than its anticipated rate. It may thus be speculated, although not formally proven, that the P_1FV_3 will outperform the P_0FV_3 by exhibiting a smaller error and a higher convergence rate. In fact, this trend becomes more noticeable at higher orders because, at the third order, the P_1FV_2 solution may be viewed as being only slightly more accurate than its P_0FV_2 counterpart when comparing error magnitudes.

7. Conclusion

This work revisits and extends Huynh's [1,2] concept of an approximate delta function (ADF), which can be expressed in the form of a finite-order polynomial with such a distinct integral property over a finite domain that it can be used to complement and extend the capabilities of existing computational methods. This is accomplished by providing the means to incorporate additional coefficients that can directly enhance the properties of the scheme under consideration. ADF polynomials enable the user to experiment with different arrangements of DOFs, and this feature can lead to the reconstruction of high-order methods with favorable properties. Despite the development of this work independently of Huynh's [1,2,23], it shares similar characteristics.

In this study, we first show that generic ADF polynomials exhibit useful properties: an ADF polynomial of order $(N + K)$, which we label here as $\tilde{\delta}_{N|K}$, reproduces the integral property of a delta function for an arbitrary polynomial of order N , while providing K arbitrary constants that can be judiciously specified. At the outset, an ADF polynomial can be used to derive the updating ODEs associated with high-order numerical schemes.

To illustrate the versatility of ADF polynomials, we show that the ADF procedure can be effectively used to reconstruct the updating ODEs of the Taylor-based, nodal DG, and Flux Reconstruction methods identically. In this process, the ADF technique provides the means to extend these techniques by introducing extra coefficients and functionalities that can be optimally specified. Then using Fourier analysis, the Taylor and nodal-based ADF-DG methods are shown to be stable and that their stability is enhanced using auxiliary coefficients.

Subsequently, by leveraging the ADF tool to handle different DOF settings, a point-value enhanced finite volume (PFV) method is introduced, which stores and updates the cell-averaged values along with the values and derivatives of unknown quantities at all nodal points. Within the PFV framework, the cell-averaged values are updated in the same manner as in the finite volume method to ensure conservation. Furthermore, the nodal information is updated through ADF integration over the entire collection of elements surrounding a given point. The updating of nodal quantities on multiple elements leads to a stable algorithm, as confirmed using a Fourier stability analysis.

By way of verification, the PFV technique is vetted by investigating its performance in treating the linear, one-dimensional, wave equation as well as the nonlinear, one-dimensional Burgers' equation. In both cases considered, a careful analysis of the logarithmic errors confirms the projected orders along with the convergence rate of error residuals. In this process, the improved stability of the PFV method is ascertained, thus demonstrating the ability of the present approach to enhance both stability and accuracy hand-in-hand. In future work, we hope to extend the PFV method to two and three spatial dimensions both with and without the incorporation of viscous effects.

Acknowledgements

This work is supported partly by Adaptive Delta Analytics, LLC, and partly by the Hugh and Loeda Francis Chair of Excellence, Department of Aerospace Engineering, Auburn University.

References

- [1] H.T. Huynh, High-order methods including discontinuous Galerkin by reconstructions on triangular meshes, in: 49th AIAA Aerospace Sciences Meeting, Orlando, FL, 2011, AIAA Paper 2011-44.
- [2] H.T. Huynh, On Formulations of Discontinuous Galerkin and Related Methods for Conservation Laws, NASA/TM-2014-218135, 2014.
- [3] W.H. Reed, T.R. Hill, Triangular Mesh Methods for the Neutron Transport Equation, Los Alamos Scientific Laboratory Report, LA-UR-73-479 LA-UR-73, 1973, p. 479.
- [4] P. LaSaint, P.A. Raviart, On a finite element method for solving the neutron transport equation, in: Mathematical Aspects of Finite Elements in Partial Differential Equations, Academic Press, 1974, pp. 89–145.
- [5] B. Cockburn, G. Karniadakis, C.W. Shu (Eds.), Discontinuous Galerkin Methods: Theory, Computation, and Application, Springer, 2000.
- [6] F. Bassi, S. Rebay, A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations, J. Comput. Phys. 131 (1997) 267–279.
- [7] F. Bassi, S. Rebay, High-order accurate discontinuous finite element solution for the 2D Euler equations, J. Comput. Phys. 138 (1997) 251–285.
- [8] B. Cockburn, C.-W. Shu, The local discontinuous Galerkin methods for time-dependent convection diffusion systems, SIAM J. Numer. Anal. 35 (1998) 2440–2463.
- [9] C.-W. Shu, Discontinuous Galerkin Method for Time Dependent Problems: Survey and Recent Developments, Springer, 2012.
- [10] D.A. Kopriva, J.H. Kolas, A conservative staggered-grid Chebyshev multidomain method for compressible flows, J. Comput. Phys. 125 (1996) 244–261.
- [11] Y. Liu, M. Vinokur, Z.J. Wang, Discontinuous spectral difference method for conservation laws on unstructured grids, J. Comput. Phys. 216 (2006) 780–801.
- [12] Z.J. Wang, Y. Liu, G. May, Spectral difference method for unstructured grids II: extension to the Euler equations, J. Sci. Comput. 32 (2007) 45–71.
- [13] Z.J. Wang, L. Zhang, Y. Liu, Spectral (finite) volume method for conservation laws on unstructured grids IV: extension to two-dimensional Euler equations, J. Comput. Phys. 194 (2) (2004) 716–741.
- [14] H.T. Huynh, A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods, in: 18th AIAA Computational Fluid Dynamics Conference, Miami, FL, 2007, AIAA Paper 2007-4079.
- [15] H.T. Huynh, A reconstruction approach to high-order schemes including discontinuous Galerkin for diffusion, in: 47th AIAA Aerospace Sciences Meeting, Orlando, FL, 2009, AIAA Paper 2009-4303.
- [16] Z.J. Wang, H. Gao, A unifying lifting collocation penalty formulation including the discontinuous Galerkin, spectral volume/difference methods for conservation laws on mixed grids, J. Comput. Phys. 228 (2) (2009) 8161–8186.
- [17] Z.J. Wang, H. Gao, A high-order lifting collocation penalty formulation for the Navier–Stokes equations on 2-D mixed grids, in: 19th AIAA Computational Fluid Dynamics Conference, San Antonio, TX, 2009, AIAA Paper 2009-3784.
- [18] H. Gao, Z.J. Wang, H.T. Huynh, Differential formulation of discontinuous Galerkin and related methods for the Navier–Stokes equations, Commun. Comput. Phys. 13 (4) (2013) 1013–1044.
- [19] B. Xie, S. Li, A. Ikebata, F. Xiao, A multi-moment finite volume method for incompressible Navier–Stokes equations on unstructured grids: volume-average/point-value formulation, J. Comput. Phys. 277 (2014) 138–162.
- [20] T.A. Eymann, Active Flux Schemes, Ph.D. thesis, University of Michigan, 2013.
- [21] T.A. Eymann, P.L. Roe, Multidimensional active flux schemes, in: 21st AIAA Computational Fluid Dynamics Conference, San Diego, CA, 2013, AIAA Paper 2013-2940.
- [22] M. Dumbser, Arbitrary high order $P_N P_M$ schemes on unstructured meshes for the compressible Navier–Stokes equations, Comput. Fluids 39 (1) (2010) 60–76.
- [23] H.T. Huynh, On high-order upwind methods for advection, in: 23rd AIAA Computational Fluid Dynamics Conference, Denver, CO, 2017, AIAA Paper 2009-4303.