Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 391 (2022) 114576
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

An enhanced non-oscillatory BFECC algorithm for finite element solution of advective transport problems

Mohammad R. Hashemi*, Riccardo Rossi, Pavel B. Ryzhakov

Universitat Politècnica de Catalunya (UPC), 08034 Barcelona, Spain
Centre Internacional de Mètodes Numèrics en Enginyeria (CIMNE), 08034 Barcelona, Spain

Received 13 May 2021; received in revised form 23 December 2021; accepted 2 January 2022
Available online 24 January 2022

Abstract

In this paper, the so-called "back and forth error compensation correction (BFECC)" methodology is utilized to improve the solvers developed for the advection equation. Strict obedience to the so-called "discrete maximum principle" is enforced by incorporating a gradient-based limiter into the BFECC algorithm. The accuracy of the BFECC algorithm in capturing the steep-fronts in hyperbolic scalar-transport problems is improved by introducing a controlled anti-diffusivity. This is achieved at the cost of performing an additional backward sub-solution-step and modifying the formulation of the error compensation accordingly. The performance of the proposed methodology is assessed by solving a series of benchmarks utilizing different combinations of the BFECC algorithms and the underlying numerical schemes. Results are presented for both the structured and unstructured meshes.

Keywords: Convection-dominated transport; BFECC; Limiter; Monotonicity preservation; Discrete maximum principle

1. Introduction

In a wide range of fluid dynamic applications, an elemental step in the numerical simulations is to solve advective, or more generally, convection-dominated transport problems (for example see [1–3]). In this context, the main challenge addressed by the researchers presently is to accurately capture the steep fronts while suppressing the spurious oscillations. In other words, the numerical method should preserve the monotonicity property [4] of the problem while ensuring sufficient spatial accuracy [5]. This challenging requirement has made the numerical solution of convection-dominated transport problems an active topic for decades, and adopting the continuous finite element method, a vast variety of the approaches have so-far been developed [6–10].

Stemming from the streamline-upwind/Petrov–Galerkin (SUPG) method [11], a series of methods were developed by introducing a residual-based stabilization term [6]. Although stable for rather smooth cases, SUPG-like methods are not monotonicity-preserving and therefore, suffer from spurious oscillations in the vicinity of a steep

* Corresponding author at: Centre Internacional de Mètodes Numèrics en Enginyeria (CIMNE), 08034 Barcelona, Spain.
E-mail addresses: mhashemi@cimne.upc.edu (M.R. Hashemi), rrossi@cimne.upc.edu (R. Rossi), pryzhakov@cimne.upc.edu (P.B. Ryzhakov).

https://doi.org/10.1016/j.cma.2022.114576

gradient [12,13]. This causes the development of the so-called "spurious oscillations at layers diminishing (SOLD)" techniques [14], which need an extremely careful choice of parameters to provide a satisfactory result [15].

Taking into account that the mathematical description of the monotonicity-preservation can be rendered into the discrete maximum principle (DMP) [13], the necessary requirement for obtaining a non-oscillatory solution is that the solver embodies DMP. Successful methods have been developed based on introducing an artificial diffusion adjusted so that DMP is satisfied [8,16–18]. The class of algebraic flux correction schemes [1,19–23] is also developed by enforcing DMP at the level of the algebraic system of equations. Consistently with Godunov's statement [24], in order to retain both the spatial accuracy and monotonicity, almost all these methods rely on a nonlinear discretized equation, which in most cases, necessitates an iterative solution procedure.

As an alternative to such iterative methods, the back and forth error compensation correction (BFECC) algorithm creates a framework for improving the solution of any time-reversible problem [25,26]; applying BFECC to a first-order underlying scheme, a second-order numerical method is obtained [27]. The BFECC algorithm is based on three sub-solution-steps; first, advancing in-time using a first-order scheme, then, retreating in-time using the same scheme to evaluate the error, and finally, advancing the compensated field in-time using the same scheme. In this sense, if an explicit underlying scheme is used, the resulting method is fully explicit (with a fixed number of sub-steps). Assuming that the underlying scheme holds DMP, and considering the evaluated error as an anti-diffusivity term, BFECC can be categorized along with the predictor–corrector algorithms of the kind described in [28]. However, despite its great potential, there are only a few attempts to utilize, analyze, and enhance the BFECC algorithm. This is mainly due to the fact that the conventional (unlimited) BFECC algorithm deteriorates the capability of the underlying numerical scheme in terms of the prevention of the spurious oscillations. In order to circumvent this issue, limited BFECC algorithms were proposed; Selle et al. [27] proposed to detect and enforce the local bounds of the final solution following the characteristic line of the advection equation. General application of such limiter is not computationally justifiable unless the semi-Lagrangian CIR scheme [29] is used. In an alternative approach, Hu et al. [30] introduced a limiter based on the detection of the over/under-shoots in the final solution, which requires two additional sub-solution-steps that significantly increases the computational cost of the method.

Knowing that the BFECC algorithm violates DMP at the error compensation step, a shock detector (limiter) can be employed to retain the monotonicity-preserving property of the solver if the underlying scheme, itself, embodies DMP. In this way, no additional sub-solution-step is required and consequently, the efficiency of the BFECC algorithm is not affected. In the present work, a gradient-based [31] continuous nodal limiter [32] is incorporated to the BFECC algorithm, recovering the DMP of the resulting scheme. In addition to the methods based on the conventional BFECC, a modified algorithm is proposed permitting more accurate capturing of the steep fronts. This modified algorithm also results in a superior performance in the smooth cases. In order to highlight the versatility of the proposed BFECC algorithm, it is applied to DMP-preserving Eulerian and semi-Lagrangian underlying schemes.

In the following sections, first, the scalar transport equation and the low-order over-diffusive monotonicity-preserving solver are described. Then, the BFECC algorithms and the incorporation of the gradient-based limiter are discussed. In Section 4, an enhanced underlying scheme is briefly presented that partially compensates for the extra-diffusivity of the low-order underlying scheme. In the final section of the present paper, numerical tests are presented addressing the one- and two-dimensional advection problems on structured and unstructured meshes.

2. Scalar transport equation

2.1. Continuum formulation

As a frequently encountered example of hyperbolic problems, the conservation of scalar field $u(\mathbf{x}, t)$ is addressed in this work. This problem is governed by a time-reversible partial differential equation [27] formulated as

$$\frac{\partial u}{\partial t} + \nabla \cdot (\mathbf{v}u) = 0 \qquad \text{in } \Omega. \tag{1}$$

Assuming that velocity field $\mathbf{v}(\mathbf{x}, t)$ retains the incompressibility condition, $\nabla \cdot \mathbf{v} = 0$, Eq. (1) can be rewritten in advective form [33].

$$\frac{\partial u}{\partial t} + \mathbf{v} \cdot \nabla u = 0 \qquad \text{in } \Omega, \tag{2}$$

This equation is subject to the initial condition,

$$u(\mathbf{x}, 0) = u_0(\mathbf{x}) \qquad \text{in } \Omega, \tag{3}$$

and Dirichlet boundary condition

$$u = u_D \qquad \text{on } \partial\Omega_D, \tag{4}$$

providing that there is an inward flux at $\partial\Omega_D$, *i.e.* $\mathbf{v}\cdot\mathbf{n} < 0$ with $\mathbf{n}$ denoting the outward normal to boundary $\partial\Omega$.

2.2. Galerkin discretization

Using test-function $q \in \mathcal{L}_2(\Omega)$, Eq. (2) leads to the problem of finding u that satisfies

$$\int_\Omega q \left(\frac{\partial u}{\partial t} + \mathbf{v}\cdot\nabla u \right) = 0 \qquad \forall q. \tag{5}$$

The finite element solution to this problem is obtained by discretizing the computational domain into a set of elements, $\mathcal{E}$, and choosing both the test-function and the trial-function in the finite element space. In this way, u is approximated as $u_h = \sum_{\mathcal{N}^e} u_i \phi_i^e(\mathbf{x})$ and $q_h \in \phi_i; i \in \mathcal{N}^e \setminus \mathcal{N}_D$ within element e. Here, ϕ_i denotes the shape function associated with node i, and $\mathcal{N}^e$ and $\mathcal{N}_D$ are the sets of nodes associated with e and the Dirichlet boundary condition, respectively. By doing the substitutions, the discrete form of the problem reads

$$\mathcal{A}_{e\in\mathcal{E}} \left(\mathbb{M}_C^e \frac{d\mathbf{U}^e}{dt} + \mathbb{C}^e \mathbf{U}^e \right) = 0, \tag{6}$$

where $\mathbf{U}^e$ is the vector of nodal unknowns u_i with $i \in \mathcal{N}^e \setminus \mathcal{N}_D$. Here, operator $\mathcal{A}$ represents the assembly of the elemental system of equations, and $\mathcal{E}$ denotes the set of the elements in the computational domain. The entities of the elemental consistent mass and convection matrices are calculated as

$$m_{ij}^e = \int_{\Omega^e} \phi_i \phi_j d\Omega, \qquad i \in \mathcal{N}^e \setminus \mathcal{N}_D, \text{ and } j \in \mathcal{N}^e, \tag{7}$$

and

$$c_{ij}^e = \int_{\Omega^e} \phi_i \mathbf{v}\cdot\nabla\phi_j d\Omega, \qquad i \in \mathcal{N}^e \setminus \mathcal{N}_D, \text{ and } j \in \mathcal{N}^e, \tag{8}$$

respectively. Assembling the contributions of all the elements, the global linear system of equations is obtained as

$$\mathbb{M}_C \frac{d\mathbf{U}}{dt} + \mathbb{C}\mathbf{U} = 0. \tag{9}$$

Without loss of generality, the finite element space is constructed by shape functions of simplex elements in this work.

2.3. Stabilization

It is widely known that in its pure form (i.e. without introducing any diffusion), Eq. (9) is subject to severe numerical instabilities [12]. Starting from Eq. (6), an established practice [21,34,35] to achieve a stabilized numerical scheme is to substitute the consistent mass matrix with lumped mass matrix $\mathbb{M}_L^e$ and introduce artificial numerical diffusion $\mathbf{D}^e$, which gives

$$\mathcal{A}_{e\in\mathcal{E}} \left(\mathbb{M}_L^e \frac{d\mathbf{U}^e}{dt} + \mathbb{C}^e \mathbf{U}^e + \mathbf{D}^e \right) = 0. \tag{10}$$

The entities of $\mathbb{M}_L^e$ are obtained via row-sum lumping scheme, which reads

$$m_{L,ij}^e = \begin{cases} m_i^e = \int_{\Omega^e} \phi_i d\Omega & \textbf{if} \quad i = j \\ 0 & \textbf{if} \quad i \neq j \end{cases} \tag{11}$$

The numerical diffusion term can be calculated as

$$\mathbf{D}^e = \nu^e \left(\mathbb{M}_L^e - \mathbb{M}_C^e \right) \mathbf{U}^e, \tag{12}$$

to formulate the global system of equations as

$$\mathbb{M}_L \frac{d\mathbf{U}}{dt} = \mathbb{L}\mathbf{U}, \tag{13}$$

with $\mathbb{L}^e = \nu^e \left(\mathbb{M}_C^e - \mathbb{M}_L^e\right) - \mathbb{C}^e$. Equivalently, one can write

$$m_i \frac{du_i}{dt} = \sum_j l_{ij} u_j, \tag{14}$$

with $m_i = \sum_{e \in \mathcal{E}_i} m_i^e$ (for linear elements), where $\mathcal{E}_i$ denotes the set of elements that share node i. It is easy to show that, as a requirement for conservation, $\sum_j l_{ij} = 0$; therefore, the sufficient condition to abide with DMP and positivity of the result [21,28] is

$$l_{ij} \geq 0, \quad i \neq j. \tag{15}$$

This is the key to attain a stabilized monotonicity-preserving *low-order* scheme [1,19], and subsequently, prevent the spurious overshoots and undershoots in the result. Providing this condition, coefficient ν can be calculated for each element as

$$\nu^e = \max(\frac{c_{ij}^e}{m_{ij}^e}, 0) \qquad \forall i, j \in \mathcal{N}^e. \tag{16}$$

The resulting scheme is known to be non-oscillatory but strongly over-diffusive [35]. It must be noted that one can reduce the artificial diffusivity by calculating ν according to the DMP at the level of the assembled global system of equation. Nonetheless, the excessive diffusion of the stabilized scheme must be alleviated in order to obtain an accurate method. One possibility consists in applying the so-called "back and forth error compensation and correction (BFECC)" algorithm that is described below.

3. Back and forth error compensation and correction

The basic idea of the BFECC algorithm is to estimate and compensate for the error associated with any numerical underlying scheme utilized for solving a reversible differential equation [27]; this is done by reversing the solution of the numerical scheme and comparing the result with the starting state, which requires consecutive application of the underlying scheme in the forward and backward directions. For solving Eq. (2), the BFECC algorithm, as first proposed in [26], can be summarized in four steps:

1. starting from $u_n(\mathbf{x})$ and solving Eq. (2) forward in time to obtain $u_{n+1}^*(\mathbf{x})$.
2. starting from $u_{n+1}^*(\mathbf{x})$ and solving Eq. (2) backward in time (by reversing velocity vector $\mathbf{v}$) to obtain $u_n^*(\mathbf{x})$.
3. estimating the error as $e(\mathbf{x}) = [u_n^*(\mathbf{x}) - u_n(\mathbf{x})]/2$ and do the compensation as $\tilde{u}_n(\mathbf{x}) = u_n(\mathbf{x}) - e(\mathbf{x})$.
4. starting from $\tilde{u}_n(\mathbf{x})$ and solving Eq. (2) forward in time to obtain $u_{n+1}(\mathbf{x})$.

Here, subscript n denotes the solution at time $t = n\Delta t$. It should be noted that a variable time-step (Δt) can be used according to the requirement (CFL-like condition) of the underlying numerical scheme.

If the numerical scheme acquired to solve Eq. (2) can be formulated as

$$\mathbb{M}\frac{d\mathbf{U}}{dt} = \mathbb{L}\mathbf{U}, \tag{17}$$

employing the backward Euler scheme in time, the application of the BFECC algorithm reads

$$\left(\frac{1}{\Delta t}\mathbb{M} - \mathbb{L}_F\right)\mathbf{U}_{n+1}^* = \frac{1}{\Delta t}\mathbb{M}\mathbf{U}_n, \tag{18}$$

$$\left(\frac{1}{\Delta t}\mathbb{M} - \mathbb{L}_B\right)\mathbf{U}_n^* = \frac{1}{\Delta t}\mathbb{M}\mathbf{U}_{n+1}^*, \tag{19}$$

$$\mathbf{E} = \frac{1}{2}\left(\mathbf{U}_n^* - \mathbf{U}_n\right) = \frac{\Delta t}{2}\mathbb{M}^{-1}\left(\mathbb{L}_F\mathbf{U}_{n+1}^* + \mathbb{L}_B\mathbf{U}_n^*\right), \tag{20}$$

and finally,

$$\left(\frac{1}{\Delta t}\mathbb{M}-\mathbb{L}_F\right)\mathbf{U}_{n+1}=\frac{1}{\Delta t}\mathbb{M}\tilde{\mathbf{U}}_n \\ =\frac{1}{\Delta t}\mathbb{M}\mathbf{U}_n-\frac{1}{2}\left(\mathbb{L}_F\mathbf{U}^*_{n+1}+\mathbb{L}_B\mathbf{U}*_n\right). \tag{21}$$

Subscripts F and B, respectively, denote the forward and the backward advection of u. Here, it is assumed that velocity field $\mathbf{v}(\mathbf{x},t)$ is given and therefore, matrices $\mathbb{L}_F$ and $\mathbb{L}_B$ are constructed for mid-time-step velocity $\mathbf{v}_{n+1/2}=(\mathbf{v}_{n+1}+\mathbf{v}_n)/2$. For the simple one-dimensional case described in the following section, it is easy to show that

$$\mathbb{C}^e=\frac{1}{2}\left(\mathbb{L}^e_B-\mathbb{L}^e_F\right). \tag{22}$$

Therefore, the last term on the right-hand-side of Eq. (21) can be interpreted as an anti-diffusive term, which is introduced by application of the BFECC algorithm. This term partially compensates for numerical diffusion $\mathbf{D}$. This property of the BFECC algorithm leads to the dismissal of condition (15) and undermines the stability of the method by making it prone to spurious over-/undershoots in the result. The occurrence of such oscillations has been mentioned in the literature and was tackled by limiting the results [27,30]. In the following, this issue will be further discussed for a simple one-dimensional case.

3.1. Analysis of one-dimensional case

For the one-dimensional case with linear elements of length h, the elemental matrices associated with the algebraically stabilized scheme described in Section 2.3 are

$$\mathbb{M}^e_C=\begin{bmatrix}\frac{h}{3} & \frac{h}{6}\\ \frac{h}{6} & \frac{h}{3}\end{bmatrix}, \tag{23}$$

$$\mathbb{M}^e_L=\begin{bmatrix}\frac{h}{2} & 0\\ 0 & \frac{h}{2}\end{bmatrix}, \tag{24}$$

$$\mathbb{C}^e=\begin{bmatrix}-\frac{v}{2} & \frac{v}{2}\\ -\frac{v}{2} & \frac{v}{2}\end{bmatrix}, \tag{25}$$

$$\mathbb{L}^e_F=\begin{bmatrix}0 & 0\\ v & -v\end{bmatrix}, \tag{26}$$

and

$$\mathbb{L}^e_B=\begin{bmatrix}-v & v\\ 0 & 0\end{bmatrix}. \tag{27}$$

Upon assembling these matrices to obtain the global linear system of equations, one has

$$\frac{du_i}{dt}+\frac{v\,(u_i-u_{i-1})}{h}=0, \tag{28}$$

which is equivalent to the first-order upwind scheme. In this simple case, jth element is formed by nodes j and $j+1$.

Considering the forward Euler scheme for more simplicity, and applying the BFECC algorithm, the resulting method reads

$$u_{n+1,i}=u_{n,i}+\frac{1}{2}\left[\left(\lambda^3-\lambda^2\right)u_{n,i-2}+\left(-3\lambda^3+4\lambda^2+\lambda\right)u_{n,i-1} \right. \\ \left. +\left(3\lambda^3-5\lambda^2\right)u_{n,i}+\left(-\lambda^3+2\lambda^2-\lambda\right)u_{n,i+1}\right], \tag{29}$$

where $\lambda=v\Delta t/h$ denotes the Courant–Friedrichs–Levy (CFL) number. While the sum of the coefficients of nodal u on the right-hand-side of Eq. (29) is zero, condition (15) is not fulfilled and hence, DMP is not guaranteed. This explains the oscillatory results of the BFECC algorithm in the vicinity of steep fronts [30], regardless of the underlying scheme used for solving Eq. (2). In Section 3.3, this issue is resolved by introducing a limited BFECC algorithm with the monotonicity-preserving property.

3.1.1. Truncation error

The exact solution of Eq. (2) in one-dimension requires that

$$u(x,t+\Delta t)=u(x-\delta,t)=u(x,t)-\delta\frac{\partial u(x,t)}{\partial x}+\frac{1}{2}\delta^2\frac{\partial^2 u(x,t)}{\partial x^2} - \frac{1}{6}\delta^3\frac{\partial^3 u(x,t)}{\partial x^3}+O(\delta^4), \tag{30}$$

with $\delta = v\Delta t$. It is possible to perform the Taylor expansion for the discretized equations as well; the Galerkin scheme (9) can be expanded as

$$u_{n+1,i}=u_{n,i}-\delta\frac{u_{n,i+1}-u_{n,i-1}}{2h}=u_{n,i}-\delta\left(\frac{\partial u}{\partial x}+\frac{h^2}{6}\frac{\partial^3 u}{\partial x^3}+O(h^3)\right). \tag{31}$$

It must be noted that here, for the sake of simplicity, the mass matrix is considered to be lumped. Comparing Eqs. (30) and (31), the associated truncation error is

$$Tr_i=u(x_i,t_n+\Delta t)-u_{n+1,i}=\frac{\delta^2}{2}\frac{\partial^2 u}{\partial x^2}+\frac{\delta^3}{6}\left(\frac{h^2}{\delta^2}-1\right)\frac{\partial^3 u}{\partial x^3}+O(\delta^4). \tag{32}$$

Similarly, for the stabilized *low-order* underlying scheme (28) one obtains

$$Tr_i=\frac{\delta^2}{2}\left(1-\frac{h}{\delta}\right)\frac{\partial^2 u}{\partial x^2}+O(\delta^3). \tag{33}$$

Here, it is assumed that CFL number λ and consequently h/δ are set as constants. In this sense, factorizing δ appears to be logical.

Applying the same procedure to Eq. (29), for the BFECC algorithm using the stabilized *low-order* underlying scheme (28), one has

$$u_{n+1,i}=u_{n,i}-\delta\frac{\partial\phi}{\partial x}+\frac{\delta^2}{2}\frac{\partial^2\phi}{\partial x^2}-\delta^3\left(\frac{1}{2}+\frac{h}{2\delta}+\frac{h^2}{6\delta^2}\right)\frac{\partial^3\phi}{\partial x^3}+O(\delta^4). \tag{34}$$

The associated truncation error reads

$$Tr_i=\delta^3\left(\frac{1}{3}+\frac{h}{2\delta}+\frac{h^2}{6\delta^2}\right)\frac{\partial^3\phi}{\partial x^3}+O(\delta^4). \tag{35}$$

The truncation error shows a one-order improvement comparing to Eq. (33). It is evident that keeping the CFL number constant, element-size h and time-step Δt (or equivalently δ) are interchangeable.

It is worth noting that the positive coefficient of $\partial^2 u/\partial x^2$ in Eq. (32) shows the anti-diffusive (with severe spatial oscillations) characteristic of the Galerkin scheme. On the other hand, for $\lambda < 1$, the negative coefficient of the leading term in Eq. (33) reveals the diffusive nature of the stabilized *low-order* scheme, which is worsen by reducing the CFL number. Nonetheless, the absence of this leading term in Eq. (35), discloses the ability of the BFECC algorithm to compensate for the excessive diffusion of the solver. This section is closed by further proving the ability of the BFECC algorithm in removing the anti-diffusivity imposed by the Galerkin scheme; applying the BFECC algorithm to Eq. (31), one obtains

$$u_{n+1,i}=u_{n,i}+\frac{1}{16}\left[\lambda^3 u_{n,i-3}+2\lambda^2 u_{n,i-2}+\left(-3\lambda^3+8\lambda\right)u_{n,i-1} -4\lambda^2 u_{n,i}+\left(3\lambda^3-8\lambda\right)u_{n,i+1}+2\lambda^2 u_{n,i+2}-\lambda^3 u_{n,i+3}\right], \tag{36}$$

and consequently, has

$$u_{n+1,i}=u_{n,i}-\delta\frac{\partial u}{\partial x}+\frac{\delta^2}{2}\frac{\partial^2 u}{\partial x^2}+\delta^3\left(\frac{1}{4}-\frac{h^2}{12\delta^2}\right)\frac{\partial^3 u}{\partial x^3}+O(\delta^4). \tag{37}$$

Therefore, the associated truncation error is

$$Tr_i=\delta^3\left(-\frac{5}{12}+\frac{h^2}{12\delta^2}\right)\frac{\partial^3 u}{\partial x^3}+O(\delta^4). \tag{38}$$

The absence of $\partial^2 u/\partial x^2$ in Eq (38) asserts the compensation for the anti-diffusivity detected in Eq. (32).

3.2. Modified algorithm

In order to obtain further improvement, the BFECC algorithm can be modified as outlined in the following steps;

1. starting from $u_n(\mathbf{x})$ and solving Eq. (2) forward in time to obtain $u^*_{n+1}(\mathbf{x})$.
2. starting from $[u_n(\mathbf{x})+u^*_{n+1}(\mathbf{x})]/2$ and solving Eq. (2) *half-way* ($\Delta t/2$) backward in time (by reversing velocity vector $\mathbf{v}$) to obtain $u^*_n(\mathbf{x})$.
3. estimating the error as $e(\mathbf{x}) = u^*_n(\mathbf{x}) - u_n(\mathbf{x})$ and do the compensation as $\tilde{u}_n(\mathbf{x}) = u_n(\mathbf{x}) - e(\mathbf{x})$.
4. starting from $\tilde{u}_n(\mathbf{x})$ and solving Eq. (2) forward in time to obtain $u_{n+1}(\mathbf{x})$.

As done before for the conventional BFECC algorithm by employing the backward Euler scheme in time, the application of this modified BFECC algorithm to the scheme presented in Eq. (17) reads

$$\left(\frac{1}{\Delta t}\mathbb{M} - \mathbb{L}_F\right)\mathbf{U}^*_{n+1} = \frac{1}{\Delta t}\mathbb{M}\mathbf{U}_n, \tag{39}$$

as the first step, and

$$\left(\frac{2}{\Delta t}\mathbb{M} - \mathbb{L}_B\right)\mathbf{U}^*_{n} = \frac{1}{\Delta t}\mathbb{M}\left(\mathbf{U}^*_{n+1} + \mathbf{U}_n\right), \tag{40}$$

as the second step. Adding Eqs. (39) and (40), one obtains

$$\frac{2}{\Delta t}\mathbb{M}\mathbf{U}^*_n = \mathbb{L}_B\mathbf{U}^*_n + \mathbb{L}_F\mathbf{U}^*_{n+1} + \frac{2}{\Delta t}\mathbb{M}\mathbf{U}_n, \tag{41}$$

from which the third step of the modified BFECC algorithm leads to

$$\mathbf{E} = \mathbf{U}^*_n - \mathbf{U}_n = \frac{\Delta t}{2}\mathbb{M}^{-1}\left(\mathbb{L}_F\mathbf{U}^*_{n+1} + \mathbb{L}_B\mathbf{U}^*_n\right), \tag{42}$$

that is the same as the error calculated in Eq. (20) for the conventional BFECC algorithm. Therefore, it is readily seen that both the conventional and the modified BFECC algorithms are equivalent if applied to a solver formulated as Eq. (17) and discretized in time using the backward Euler scheme. Nevertheless, if an explicit (*e.g.* forward Euler) scheme is used, this modified algorithm is not equivalent to the conventional BFECC algorithm. In the following, it is shown that besides the conventional BFECC algorithm, the introduced modified BFECC algorithm can be acquired to add a controlled anti-diffusivity to the solution.

3.2.1. One-dimensional case

Similar to Section 3.1, application of the modified BFECC algorithm to the stabilized *low-order* scheme (28) with the forward Euler time discretization leads to

$$\begin{aligned} u_{n+1,i} = u_{n,i} + \frac{1}{4}\big[&\left(\lambda^3 - 2\lambda^2\right)u_{n,i-2} + \left(-3\lambda^3 + 7\lambda^2 + 2\lambda\right)u_{n,i-1} \\ &+ \left(3\lambda^3 - 8\lambda^2\right)u_{n,i} + \left(-\lambda^3 + 3\lambda^2 - 2\lambda\right)u_{n,i+1}\big]. \end{aligned} \tag{43}$$

This leads to

$$u_{n+1,i} = u_{n,i} - \delta\frac{\partial u}{\partial x} + \frac{\delta^2}{4}\frac{\partial^2 u}{\partial x^2} - \delta^3\left(\frac{1}{4} + \frac{h}{2\delta} + \frac{h^2}{6\delta^2}\right)\frac{\partial^3 u}{\partial x^3} + O(\delta^4), \tag{44}$$

from which, the truncation error is calculated as

$$Tr_i = \frac{\delta^2}{4}\frac{\partial^2 u}{\partial x^2} + \delta^3\left(\frac{1}{6} + \frac{h}{2\delta} + \frac{h^2}{6\delta^2}\right)\frac{\partial^3 u}{\partial x^3} + O(\delta^4). \tag{45}$$

Eq. (45) clearly shows that the modified BFECC algorithm adds half the amount of the anti-diffusivity of the Galerkin scheme (see Eq. (32)). Moreover, the modified algorithm neither improves nor impairs the order of the solver unlike the conventional BFECC algorithm which is proved to provide enhancement upon application to the first-order solvers. Nonetheless, in Section 3.4, a combined algorithm is introduced that benefits from the advantages of both the conventional and the modified BFECC algorithms.

3.3. Nodal limiter

As discussed above, the maximum principle and the positivity are no more guaranteed upon the application of (either the conventional or modified) BFECC algorithm. Therefor, in order to circumvent the associated instability issues in the present work, a continuous nodal limiter is utilized to control the application of the BFECC algorithm; the idea is to fully apply the error compensation according to the BFECC algorithm wherever the convected field is smooth while ignoring the correction in the vicinity of local extrema. In this way, upon the application of limiter function α, the third step of (either the standard or modified) BFECC algorithm reads $\tilde{u}_n(\mathbf{x}) = u_n(\mathbf{x}) - \alpha(\mathbf{x})e(\mathbf{x})$. It is worth mentioning that while DMP and the positivity condition are guaranteed for the solver underlying the fourth-step of the BFECC algorithm, preserving the monotonicity for $\tilde{u}_n$ is the sufficient condition for the BFECC algorithm to satisfy these essential requirements. Here, the continuity of the limiter function allows the partial application of the BFECC algorithm by quantifying the smoothness of the convected field.

The limiter utilized in the present work was originally proposed in [32] and further utilized in [18] to control over the artificial diffusion associated with the stabilization term introduced to a convection–diffusion equation. Later on, addressing its shortcoming for asymmetric meshes [31], a more general version of this limiter was introduced as [35]

$$\alpha_i = 1 - \left[\frac{\left| \sum_{j \in \mathcal{N}_i \setminus i} \beta_{ij} \left(u_i - u_j \right) \right|}{\sum_{j \in \mathcal{N}_i \setminus i} \beta_{ij} \left| u_i - u_j \right| + \varepsilon} \right]^{\zeta}, \tag{46}$$

where $\alpha_i = \alpha(\mathbf{x}_i)$ and $\mathcal{N}_i$ denotes the set of nodes, which share an edge with node i. In Eq. (46), $\varepsilon \sim O(10^{-15})$ is an extremely small constant that is introduced to prevent division by zero in cases of flat u, and power ζ characterizes the spatial variation of α by determining the acuteness of its decay rate nearby the location of a non-smooth convected field. In the present work, $\zeta = 2$ is set for limiting the BFECC algorithm. Coefficient β_{ij} is calculated based on the procedure introduced by Kuzmin et al. [35] in order to maintain the linearity-preservation in cases of an asymmetric mesh.

3.4. Combined algorithm

The outstanding characteristic of the conventional BFECC algorithm in enhancing the order of accuracy of the method begins to fade away as the limiter decreases from unity; this is an inevitable cost to preserve the monotonicity. The more acute the local change in the gradient is, the smaller the limiter becomes. On the other hand, the nodal limiter (46) can be employed as a shock detector [18], and consequently, a measure for determining the nodes that are subject to relatively large numerical diffusion. The basic idea here is to acquire the limited amount of anti-diffusivity introduced by the modified BFECC algorithm (see Eq. (45) and the discussion afterwards) to partially compensate for excessive numerical diffusion.

In this manner, the combined BFECC algorithm is proposed as

1. starting from $u_n(\mathbf{x})$ and solving Eq. (2) forward in time to obtain $u^*_{n+1}(\mathbf{x})$.
2. doing the backward steps:
   - 2.1 starting from $u^*_{n+1}(\mathbf{x})$ and solving Eq. (2) backward in time to obtain $u^*_n(\mathbf{x})$.
   - 2.2 starting from $[u_n(\mathbf{x}) + u^*_{n+1}(\mathbf{x})]/2$ and solving Eq. (2) half-way ($\Delta t/2$) backward in time to obtain $u^{**}_n(\mathbf{x})$.
3. do the compensation as $\tilde{u}_n(\mathbf{x}) = u_n(\mathbf{x}) - e(\mathbf{x})$ with error depending on α:

$$e(\mathbf{x}) = \begin{cases} \left[u^*_n(\mathbf{x}) - u_n(\mathbf{x}) \right]/2 & \textbf{if} \quad \alpha(\mathbf{x}) > \alpha_{th} \\ u^{**}_n(\mathbf{x}) - u_n(\mathbf{x}) & \textbf{if} \quad \alpha(\mathbf{x}) \leq \alpha_{th} \end{cases} \tag{47}$$

4. starting from $\tilde{u}_n(\mathbf{x})$ and solving Eq. (2) forward in time to obtain $u_{n+1}(\mathbf{x})$.

In this algorithm, α_{th} denotes the threshold, below which the conventional BFECC algorithm is substituted by the modified BFECC algorithm. Numerical tests show that the most desirable results can be obtained by $\alpha_{th} \approx 0.9$.

4. Enhanced scheme

In this section, a methodology is described that allows limiting the extra diffusivity of the stabilized *low-order* underlying scheme (10). The resulting scheme is called as the "*enhanced* scheme" throughout this paper. The improvement of the *low-order* stabilized scheme (10) is based on rolling back the stabilization procedure in the smooth area in order to minimize the artificial diffusion. In the meantime, the formulation remains intact in the vicinity of local extrema in order to hold DMP. Similar to the introduced limited BFECC algorithm, limiter α plays the key role in this formulation enhancement procedure.

Rewriting Eq. (10) and expanding the artificial diffusion term, $\mathbf{D}^e$, one has

$$\mathcal{A}_{e\in\mathcal{E}}\left(\mathbb{M}_L^e\frac{d\mathbf{U}^e}{dt}+\mathbb{C}^e\mathbf{U}^e+\nu^e\left(\mathbb{M}_L^e-\mathbb{M}_C^e\right)\mathbf{U}^e\right)=0. \tag{48}$$

Towards the minimization of the numerical diffusion, one can take two distinguished steps; bringing back the consistent mass-matrix and compensating for the artificial diffusion term. Incorporating the limiter, these two steps read

$$\mathcal{A}_{e\in\mathcal{E}}\left\{\left[\alpha^e\mathbb{M}_C^e+(1-\alpha^e)\mathbb{M}_L^e\right]\frac{d\mathbf{U}^e}{dt}+\mathbb{C}^e\mathbf{U}^e+\nu^e\left(\mathbb{M}_L^e-\mathbb{M}_C^e\right)\mathbf{U}^e-\alpha^e\hat{\mathbf{D}}^e\right\}=0, \tag{49}$$

where $\hat{\mathbf{D}}^e$ is an approximation of $\mathbf{D}^e$. For a simplex element, it can be shown that [35]

$$m_i^e u_i-\sum_{j\in\mathcal{N}^e}m_{ij}^e u_j=(1+d)\int_{\Omega^e}\phi_i(u_h-\bar{u}^e)d\Omega, \tag{50}$$

where d denotes the number of dimensions ($d=2$ in 2D) and elemental average $\bar{u}^e$ is calculated as

$$\bar{u}^e=\frac{\int_{\Omega^e}u_h d\Omega}{\int_{\Omega^e}d\Omega}. \tag{51}$$

Introducing $u_h(\mathbf{x})\approx\bar{u}^e+\mathbf{g}^e\cdot(\mathbf{x}-\bar{\mathbf{x}}^e)$ into Eq. (50), the entities of $\hat{\mathbf{D}}^e$ are calculated as

$$\hat{d}_i^e=\nu^e(1+d)\int_{\Omega^e}\phi_i\mathbf{g}^e\cdot(\mathbf{x}-\bar{\mathbf{x}}^e)d\Omega, \tag{52}$$

In this work, $\mathbf{g}^e$ is calculated as the elemental average of nodal gradients $\mathbf{g}_i$, which are obtained using lumped-mass projection of ∇u as

$$\mathbf{g}_i=\frac{1}{m_i}\int_{\Omega}\phi_i\sum_{j\in\mathcal{N}}\nabla\phi_j u_j d\Omega. \tag{53}$$

The elemental limiter is then the minimum of the associated nodal ones, *i.e.*

$$\alpha^e=\min_{i\in\mathcal{N}^e}\alpha_i^e. \tag{54}$$

In the computation of Eq. (54), α_i^e is calculated using Eq. (46) with $\zeta=4$. It must be noted that the presented scheme can be considered as an explicit variant of the method proposed by Kuzmin et al. [35], which has similarities in essence with the formulation introduced in [36]. It is also worth noting that for $\alpha^e\to 1$, Eq. (49) tends to the Galerkin scheme and therefore, a strong anti-diffusivity is expected. In Appendix B, the implementation of this *enhanced* scheme is further described. In the numerical tests, it is shown how the application of the proposed limited BFECC algorithm further improves the results by eliminating the extra anti-diffusivity of this underlying *enhanced* scheme. This property is likely to be related to the stabilization property of the BFECC algorithm, which was proved in [29]. In this sense, the application of the BFECC algorithms to the numerical schemes belonging to the class of the above described enhance scheme can be seen as a key for minimizing the imposed diffusivity by using more acute limiters.

5. Results

In this section, the performance of the proposed combined BFECC algorithm is investigated in three test-cases; in the first set of tests, different BFECC algorithms are applied to the *low-order* and *enhanced* underlying schemes

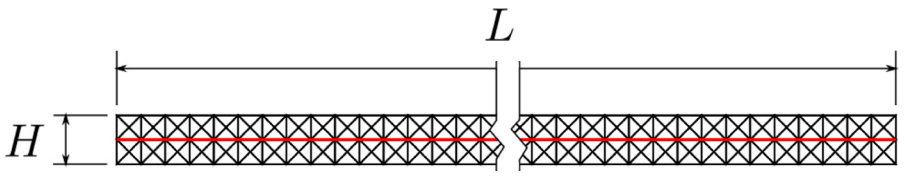


Fig. 1. Semi-1D mesh for advection of the square wave. Results are presented for the nodes lie on the center-line marked by a red solid line. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

and employed for the one-dimensional advection of both a non-smooth square-wave and a smooth sine-wave. The second test-case is the solid-body rotation of a notched cylinder, smooth hump and a cone [33], which is a well-established benchmark in this context. Here, the versatility of the proposed BFECC methodology is further analyzed by its application to the unconditionally stable (semi-Lagrangian) CIR scheme [27,29] (see Appendix A) and the SUPG scheme (with the cross-wind stabilization [12]). In the last test-case, the oblique in-flow of a scalar field is simulated in order to study the effect of the combined BFECC algorithm on the cross-stream and the stream-wise diffusion of the solver.

In the following, all the simulations are performed using the forward Euler scheme for discretizing the governing equations in time. Moreover, for the application of the combined BFECC algorithm, the switch between the algorithms is done according to the threshold of $\alpha_{th} = 0.9$ and 0.95 for the Eulerian schemes and the semi-Lagrangian approach, respectively. For these test-cases, L^1- and L^2-norm of the error are approximated as [28,31]

$$E_1 = \sum_{i \in \mathcal{N}} m_i \left| u(\mathbf{x}_i) - u_i \right|, \tag{55}$$

and

$$E_2 = \sqrt{\sum_{i \in \mathcal{N}} m_i \left[u(\mathbf{x}_i) - u_i \right]^2}, \tag{56}$$

respectively.

5.1. One dimensional advection

The test-cases addressed in this section consist of the one-dimensional[1] advection of a square-wave with an initially discontinuous field and a sine-wave, which corresponds to an initially smooth field; the associated initial conditions are defined, respectively, by

$$u_0(x) = \begin{cases} 1 & \textbf{if} \quad 0.1 \geq x \leq 0.31 \\ 0 & \textbf{else} \end{cases} \tag{57}$$

and

$$u_0(x) = \begin{cases} \frac{1}{2} - \frac{1}{2} \sin\left(10(x - 0.1) + \frac{\pi}{2}\right) & \textbf{if} \quad 0.1 \geq x \leq 0.3 \\ 0 & \textbf{else} \end{cases} \tag{58}$$

The former case is a well-established test for the assessment of the performance of the numerical methods [19,37] in the presence of a severe non-smoothness in the field. On the other hand, the sine-wave test is designed to reveal the ability of the numerical approach to minimize the unwanted side-effects of the compensatory anti-diffusivity. These test-cases are simulated on the semi-1D mesh shown in Fig. 1 with $L = 1$, $H = 0.02$, and $\mathbf{v} = \mathbf{e}_x$, where $\mathbf{e}_x$ is the unit vector in the x-direction. The associated mesh-size is calculated as $h = 1/DOF_{cl}$,[2] where DOF_{cl} denotes the number of degrees-of-freedom along the center-line of the domain shown in Fig. 1.

[1] It must be highlighted that the results are obtained on a two-dimensional mesh as shown in Fig. 1.

[2] It must be noted that due to the symmetry of the mesh (shown in Fig. 1), the effective mesh-size is smaller than $1/DOF_{cl}$.

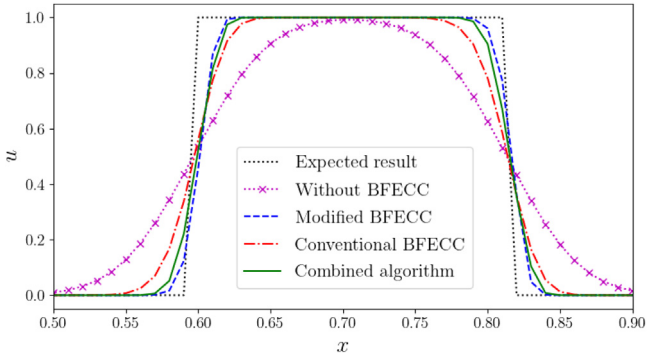


Fig. 2. Advection of a square-wave using different BFECC algorithms combined with the stabilized *low-order* underlying scheme.

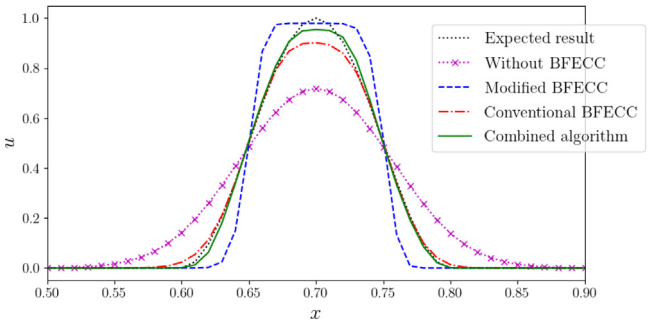


Fig. 3. Advection of a sine-wave with $dt = 0.004$ and $\theta = 0$.

5.1.1. Low-order stabilized Eulerian scheme

The first scheme to analyze in combination with the proposed limited BFECC algorithm is the stabilized *low-order* scheme described in Section 2.3. Here, the time-step is set to $dt = 0.004$ and $DOF_{cl} = 100$, which give $CFL = dt|\mathbf{v}|/h = 0.4$. The final ($t = 0.5$) distribution of u along the center-line is illustrated for the non-smooth and the smooth wave in Figs. 2 and 3, respectively.

As expected, without the application of an error compensation algorithm, the stabilized *low-order* scheme is too diffusive and consequently, leads to an undesirable solution in both cases; making a compensation for the extra diffusivity, the conventional BFECC algorithm dramatically improves the result. By further adding an extra anti-diffusion to the solution of the advection equation, the modified BFECC scheme provides a better result than the conventional BFECC algorithm in the non-smooth case. However, this extra anti-diffusion disturbs the solution for the smooth case. On the other hand, the combined BFECC algorithm although increases the computational cost by 30%, leads to a result that closely follows that of the modified BFECC algorithm in the non-smooth case while does not disturb the solution in the smooth case. Fig. 3 clearly shows the great advantage of using the combined BFECC algorithm for the advection of the smooth-wave; comparing to the conventional BFECC algorithm, the proposed BFECC algorithm provides a more accurate solution in the smooth case. A more critical assessment of the performance of different BFECC algorithms is possible by comparing L^1- and L^2-norm of the associated errors as presented in Tables 1 and 2.

So far, the results were reported for a single mesh with $DOF_{cl} = 100$. Here, the sine-wave test-case is further solved for $DOF_{cl} = 50$, 200, and 400, in order to assess the effect of different BFECC algorithms on the convergence of the solver, which is measured by the so-called "experimental order of convergence (EOC)" defined as [33,38]

$$EOC = \frac{\log(\frac{E(h_2)}{E(h_1)})}{\log(\frac{h_2}{h_1})}, \tag{59}$$

Table 1
Error associated with the advection of a square-wave using different BFECC algorithms combined with the stabilized *low-order* underlying scheme.

Algorithm	E_1	E_2
Without BFECC	1.301×10^{-3}	1.951×10^{-2}
Modified BFECC	2.995×10^{-4}	1.036×10^{-2}
Conventional BFECC	5.497×10^{-4}	1.260×10^{-2}
Combined algorithm	3.726×10^{-4}	1.076×10^{-2}

Table 2
Error associated with the advection of a sine-wave using different BFECC algorithms combined with the stabilized *low-order* underlying scheme.

Algorithm	E_1	E_2
Without BFECC	7.349×10^{-4}	1.117×10^{-2}
Modified BFECC	4.127×10^{-4}	8.293×10^{-3}
Conventional BFECC	1.015×10^{-4}	2.370×10^{-3}
Combined algorithm	6.947×10^{-5}	1.388×10^{-3}

Table 3
Convergence of the results of the advection of a sine-wave obtained using the stabilized *low-order* underlying scheme with and without the proposed combined BFECC algorithm for $dt = 0.001$.

Mesh-size	Without BFECC				Combined BFECC algorithm			
	E_1	EOC_1	E_2	EOC_2	E_1	EOC_1	E_2	EOC_2
1/50	0.0016		0.022		0.0014		0.021	
1/100	0.0011	0.55	0.016	0.46	0.00062	1.16	0.010	1.06
1/200	0.00062	0.83	0.0096	0.74	0.00013	2.22	0.0023	2.17
1/400	0.00024	1.40	0.0038	1.32	0.000019	2.80	0.00035	2.72

where $E(h)$ is the error associated with mesh-size h. The EOC values are presented for the stabilized *low-order* scheme with and without the proposed combined BFECC algorithm in Table 3. These set of data are obtained by setting the time-step to $dt = 0.001$.

It is clearly observed that in addition to the dramatic decrease in the magnitude of the error, the proposed combined BFECC algorithm improves the mesh-convergence; by applying the proposed algorithm, EOC is almost doubled. In the following, the same tests are administered for the alternative underlying scheme discussed in the present work, *i.e.* the *enhanced* method.

5.1.2. Enhanced scheme

Following the results presented for the *low-order* scheme, in this section, different BFECC algorithms are combined with the *enhanced* scheme (described in Section 4) and applied to the same one-dimensional test-cases. Considering that this enhanced underlying scheme is more sensitive to the time-step than the *low-order* scheme, here, $dt = 0.001$ is set for $DOF_{cl} = 100$. Results are presented in Figs. 4 and 5 for the non-smooth and the smooth test-cases, respectively.

Benefiting from limited corrective terms, it is expected that the *enhanced* scheme provides more accurate solutions without violating the positivity as well as the maximum principle; it is clearly seen by comparing the results presented in Fig. 4 with those presented in Figs. 2 for the non-smooth case. Nonetheless, for the smooth case, the application of the proposed combined BFECC algorithm to the stabilized *low-order* scheme provides a comparably accurate result (see Figs. 3 and 5).

Here, one should highlight the potential of the BFECC algorithm to adjust the extra anti-diffusivity together with its capability to compensate for the extra diffusivity of the schemes developed for the convection-dominated problems; it is evident in Fig. 5 that by applying either the conventional or the proposed combined BFECC algorithm, the anti-diffusivity of the *enhance* scheme is finely adjusted minimizing the associated error in the smooth case. In Tables 4 and 5, L^1- and L^2-norm of the error are presented for different approaches developed based on the *enhanced* scheme and applied to the one-dimensional advection of the square-wave and the sine-wave, respectively.

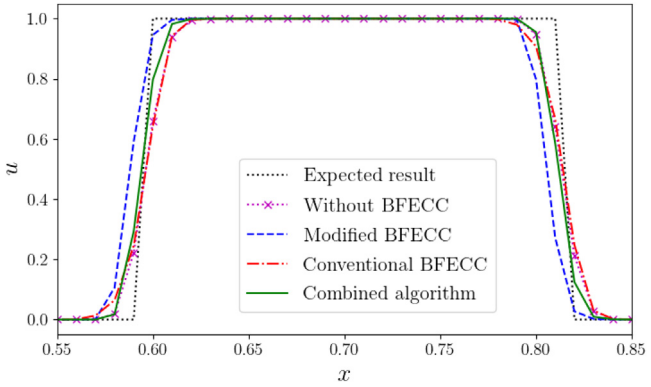


Fig. 4. Advection of a square-wave with improved stabilized scheme, $dt = 0.001$, and $\theta = 0$.

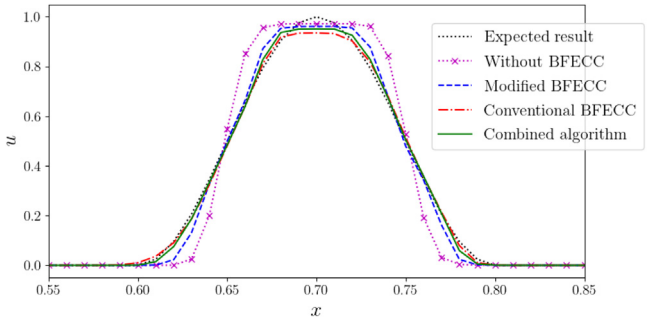


Fig. 5. Advection of a sine-wave with improved stabilized scheme, $dt = 0.001$, and $\theta = 0$.

Table 4
Error associated with the advection of a square-wave using different BFECC algorithms combined with the *enhanced* underlying scheme.

Algorithm	E_1	E_2
Without BFECC	2.729×10^{-4}	8.750×10^{-3}
Modified BFECC	3.502×10^{-4}	1.378×10^{-2}
Conventional BFECC	3.106×10^{-4}	9.223×10^{-3}
Combined algorithm	2.405×10^{-4}	8.598×10^{-3}

Similar to what is observed for the *low-order* scheme, the proposed combined BFECC algorithm shows an overall outperformance in the non-smooth and smooth cases; it improves the method in the non-smooth case while provides a slightly more accurate result than the conventional BFECC algorithm in the smooth case.

5.2. Solid-body rotation

In this section, the counter-clockwise rotation of a slotted disk,

$$u_0(x, y) = \begin{cases} 1 & \textbf{if} \begin{cases} \sqrt{(x-0.5)^2 + (y-0.75)^2} \leq 0.15 \quad \textbf{and} \\ |x - 0.5| \geq 0.025 \quad \textbf{or} \quad y \geq 0.85 \end{cases} \\ 0 & \textbf{else} \end{cases} \tag{60}$$

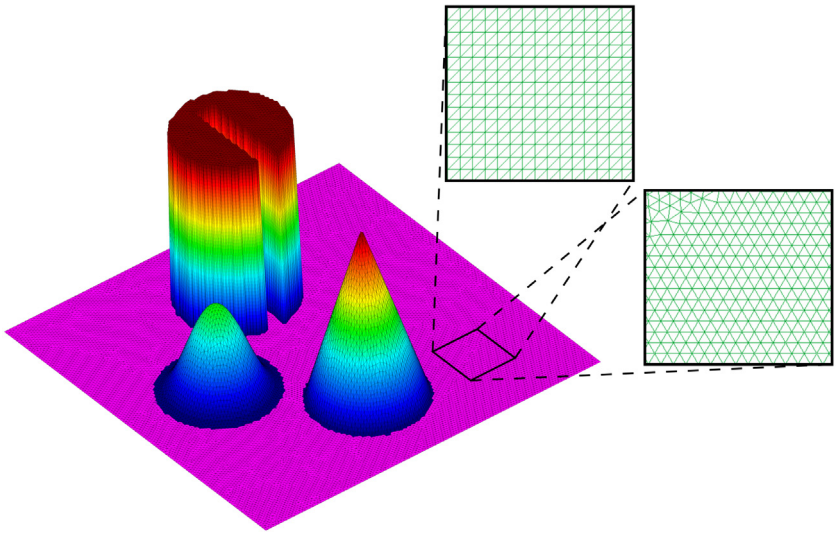


Fig. 6. Initial configuration of the solid-body rotation test-case. Structured and unstructured meshes are shown.

Table 5
Error associated with the advection of a sine-wave using different BFECC algorithms combined with the *enhanced* underlying scheme.

Algorithm	E_1	E_2
Without BFECC	3.658×10^{-4}	7.153×10^{-3}
Modified BFECC	1.390×10^{-4}	2.769×10^{-3}
Conventional BFECC	7.812×10^{-5}	1.536×10^{-3}
Combined algorithm	7.587×10^{-5}	1.446×10^{-3}

a non-smooth (sharp) cone,

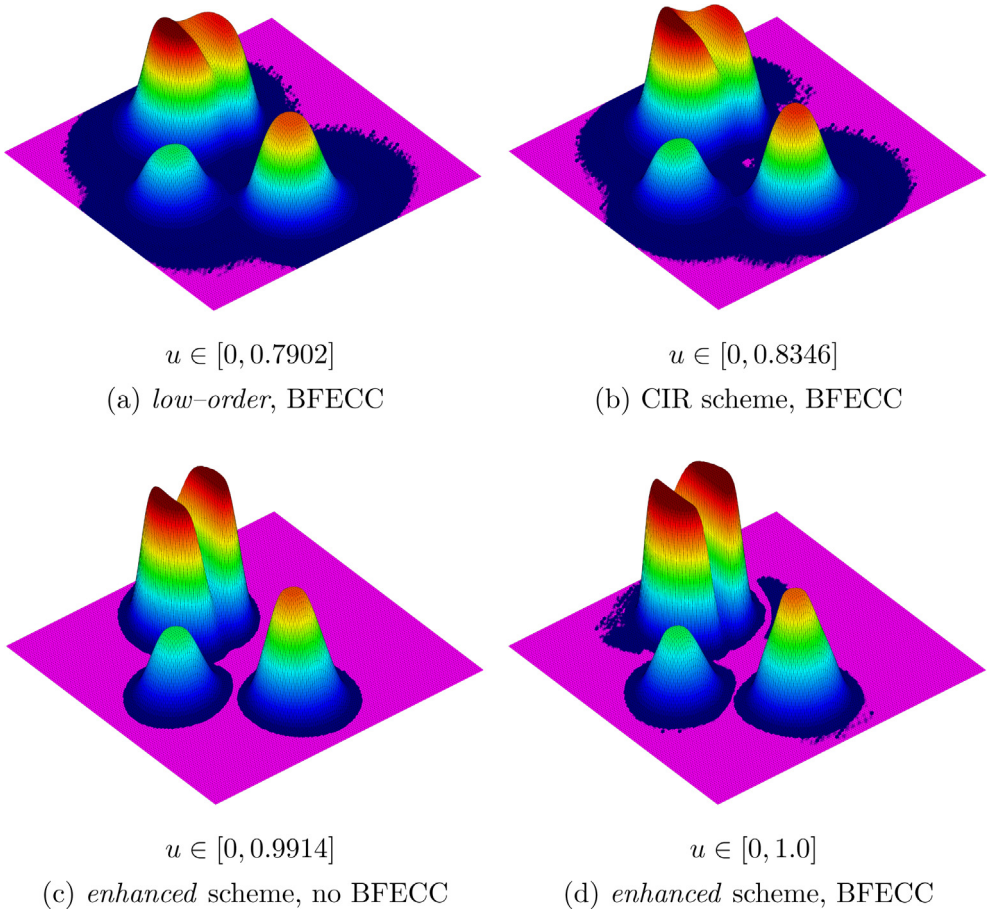
$$u_0(x, y) = \begin{cases} 1 - \frac{\sqrt{(x-0.5)^2+(y-0.25)^2}}{0.15} & \textbf{if} \quad \sqrt{(x-0.5)^2+(y-0.25)^2} \leq 0.15 \\ 0 & \textbf{else} \end{cases} \tag{61}$$

and a smooth hump,

$$u_0(x, y) = \begin{cases} \frac{1}{4} + \frac{1}{4}\cos\left(\frac{\pi\sqrt{(x-0.25)^2+(y-0.5)^2}}{0.15}\right) & \textbf{if} \quad \sqrt{(x-0.25)^2+(y-0.5)^2} \leq 0.15 \\ 0 & \textbf{else} \end{cases} \tag{62}$$

is simulated with $\mathbf{v}(x, y) = (0.5 - y, x - 0.5)$, in a square 1×1-domain centered at $(x, y) = (0.5, 0.5)$. The initial condition, u_0, is shown in Fig. 6. As first proposed in [33], this test has become a benchmark for the assessment of the performance of the numerical methods developed for convection-dominated problems [28,38–40]. Here, the time-step is set to $dt = 0.001$ unless otherwise mentioned, and the computational domain is discretized using both a structured mesh with 129^2 nodes and an unstructured mesh with the average mesh-size of $h = 1/128$ (see Fig. 6). In this section, all the results are presented after one complete rotation at $t = 6.28$.

Figs. 7 shows the results of the stabilized *low-order* and *enhanced* schemes with and without the proposed combined BFECC algorithm that are obtained using the structured mesh. Here, the result of the CIR underlying scheme is also included for the sake of its comparison with the *low-order* scheme. Without the BFECC algorithm, the *low-order* scheme (as well as the CIR scheme) brings about a highly diffused u-field and therefore, the corresponding results are not shown here. It is clearly seen that all the numerical schemes abide with the positivity and the maximum principle by keeping $0 \geq u \leq 1.0$. For a better assessment of the performance of the acquired numerical schemes, the results obtained using the *enhanced* scheme with and without the proposed combined BFECC algorithm on the unstructured mesh are also presented in Fig. 8.



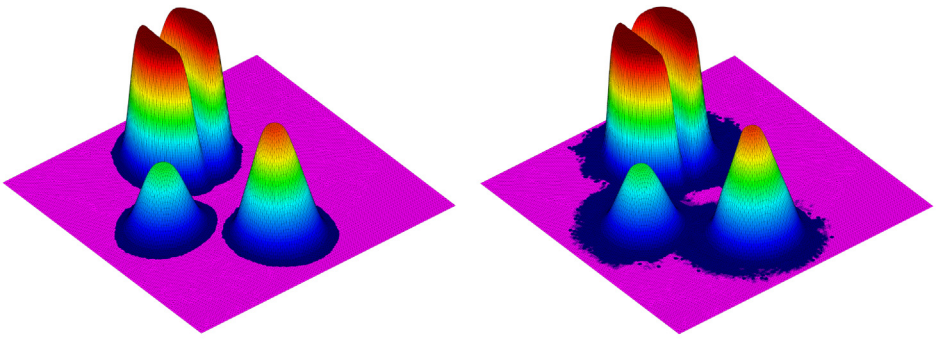
$u \in [0, 0.7902]$ (a) *low–order*, BFECC

$u \in [0, 0.8346]$ (b) CIR scheme, BFECC

$u \in [0, 0.9914]$ (c) *enhanced* scheme, no BFECC

$u \in [0, 1.0]$ (d) *enhanced* scheme, BFECC

Fig. 7. Solid-body rotation at $t = 6.28$. Results are obtained using different schemes on the structured mesh and presented as surface $z = u(x, y)$; (a) and (b) correspond to the *low-order* Eulerian scheme and the semi-Lagrangian underlying scheme with the application of the proposed combined BFECC algorithm, respectively. The results of the *enhanced* scheme without and with the combined BFECC algorithm is shown in (c) and (d), respectively.



$u \in [0, 1.0001]$ (a) Without BFECC

$u \in [0, 0.9995]$ (b) Combined BFECC

Fig. 8. Solid-body rotation at $t = 6.28$. Results are obtained for the unstructured mesh and presented as surface $z = u(x, y)$; (a) and (b) correspond to the *enhanced* underlying scheme without and with the combined BFECC algorithm, respectively.

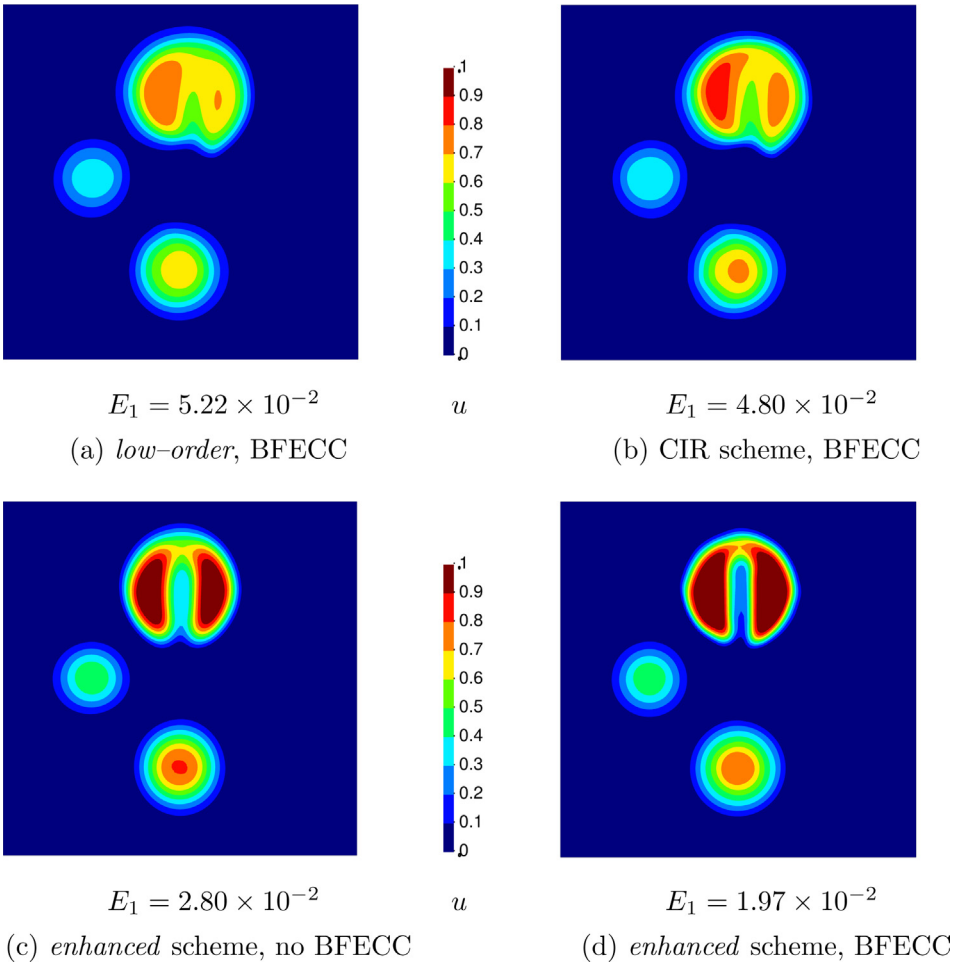


Fig. 9. Solid-body rotation at $t = 6.28$. Results are obtained using different schemes on the structured mesh and presented as contours of $u(x, y)$; (a) and (b) correspond to the *low-order* Eulerian scheme and the semi-Lagrangian approach with the underlying scheme of the proposed combined BFECC algorithm, respectively. The results of the *enhanced* scheme without and with the combined BFECC algorithm are shown in (c) and (d), respectively.

The slightly more accurate solution on the unstructured mesh is due to its slightly larger number of mesh-nodes comparing to the structured mesh.

In Figs. 9 and 10, the same set of results are presented as the contours of u at $t = 6.28$. The L^1-norm of the corresponding errors is also reported in these figures.

It is observable that, in an overall view, the semi-Lagrangian approach slightly outperforms the *low-order* Eulerian scheme while by applying these two schemes along with the BFECC algorithm, the symmetry of the slotted disk is disturbed after one complete rotation. In case of the *enhanced* scheme, the application of the proposed BFECC algorithm yields a considerable improvement in the advection of the slotted-disk.

In order to bring the effect of the proposed combined BFECC algorithm into sharp focus, the result of the *enhanced* scheme with and without the application of this algorithm are shown in Figs. 11 and 12 along different cut-lines passing through the domain. These figures correspond to the unstructured mesh.

In addition to the better representation of the slotted disk, the proposed BFECC algorithm remarkably improves the results for the advection of the smooth hump and the linear body of the cone, which is brought about by its capability to adjust the (anti-)diffusivity of the numerical schemes. In other words, using the presented *enhanced* scheme, due to an excessive anti-diffusivity, the result is subject to a difficulty denoted as "*terracing*" [21] that is majorly cured by utilizing the BFECC algorithm.

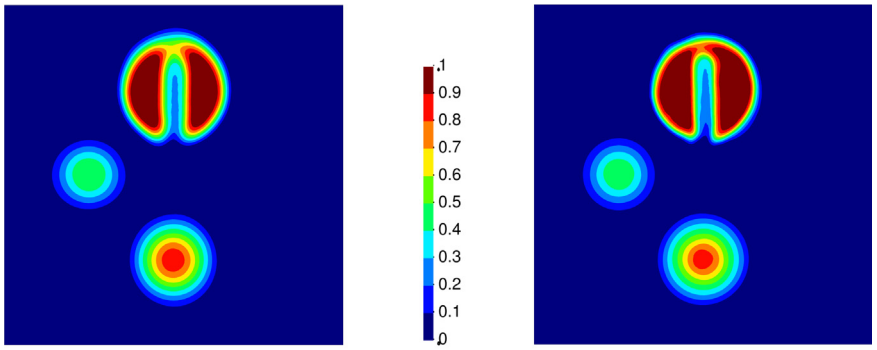


Fig. 10. Solid-body rotation at $t = 6.28$. Results are obtained for the unstructured mesh and presented as contours of $u(x, y)$; (a) and (b) correspond to the *enhanced* underlying scheme without and with the combined BFECC algorithm, respectively.

At the end of this section, it is worth to briefly investigate the performance of the proposed BFECC algorithm in combination with the SUPG-CWS scheme. Results are presented in Figs. 13 and 14 as the surface of $z = u(x, y)$ and contours of $u(x, y)$, respectively.

Upon the application of the proposed BFECC algorithm, the result of the SUPG-CWS scheme is dramatically improved. Therefore, the proposed BFECC can also be considered as a viable means to improve the class of SUPG-like methods.

5.3. Oblique inflow

This section aims at the investigation of the effect of the proposed BFECC algorithm on reducing the stream-wise as well as the cross-stream diffusion during the transport of a sharp layer. To this end, Eq. (2) is solved in a square 1×1-domain with constant velocity $\mathbf{v} = -0.8\mathbf{e}_x - 0.6\mathbf{e}_y$, $dt = 0.001$, and Dirichlet boundary condition

$$u_D(x, y) = \begin{cases} 1 & \textbf{if} \quad x \geq 0.8 \ \textbf{and} \ y = 1 \\ 0 & \textbf{else} \end{cases} \tag{63}$$

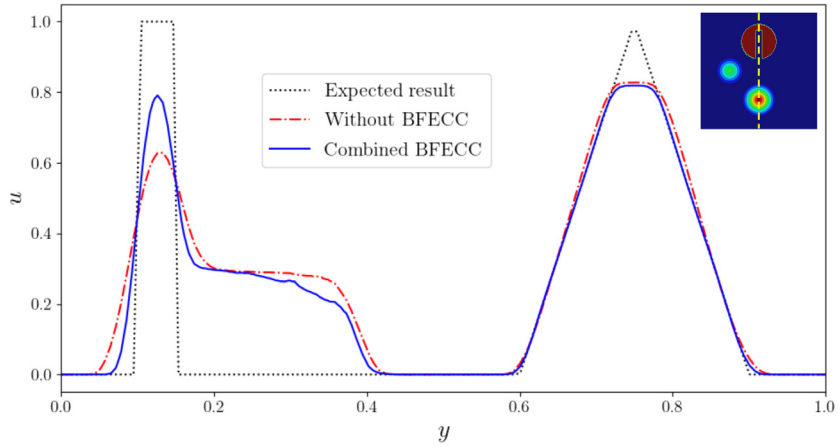
imposed on the inflow ($x = 1$ and $y = 1$) boundaries of the domain. Here, the time-step is set to $dt = 0.001$ and the results are obtained using the *enhanced* scheme with and without the proposed combined BFECC algorithm on the 129^2 structured mesh as shown in Fig. 15.

Figs. 16 and 17 present the results along a perpendicular to the stream and a parallel to the stream cut-line, respectively. It is clearly observable that the proposed combined BFECC algorithm effectively reduces the cross-stream diffusivity while it improves the capturing of the theoretically sharp stream-wise front.

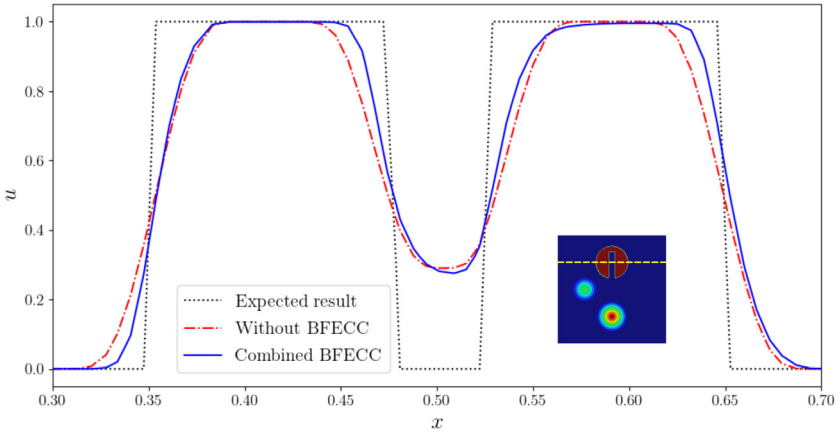
The L^1-norm of the error is $E_1 = 0.0203$ for the *enhanced* scheme without the BFECC algorithm. Upon the application of the proposed combined BFECC algorithm, the error is reduced to $E_1 = 0.0158$.

6. Conclusion

This work constituted a methodology to substantially improve the accuracy of the numerical solution of the advection equation by adjusting the diffusivity of the numerical schemes; this was achieved by enhancing the back and forth error compensation and correction (BFECC) algorithm. It was shown how a gradient-based limiter can be used to retain the monotonicity of the numerical method obtained as a combination of the BFECC algorithm and an originally monotonicity-preserving scheme. The proposed algorithm was combined with different stabilized schemes and the resulting solvers were applied to a series of advection test-cases. It was revealed that while the proposed algorithm possesses the capability of the conventional BFECC algorithm for adjusting both the extra diffusivity and anti-diffusivity of the underlying numerical scheme, it provides a considerable improvement to the result in



(a) $x = 0.5$

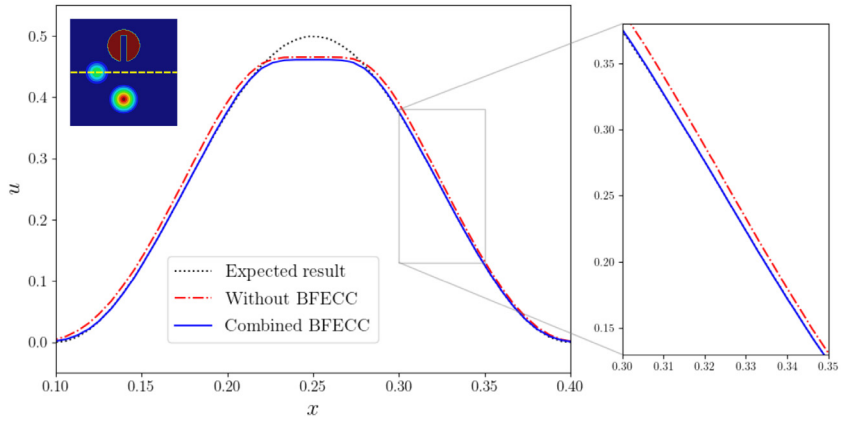


(b) $y = 0.75$

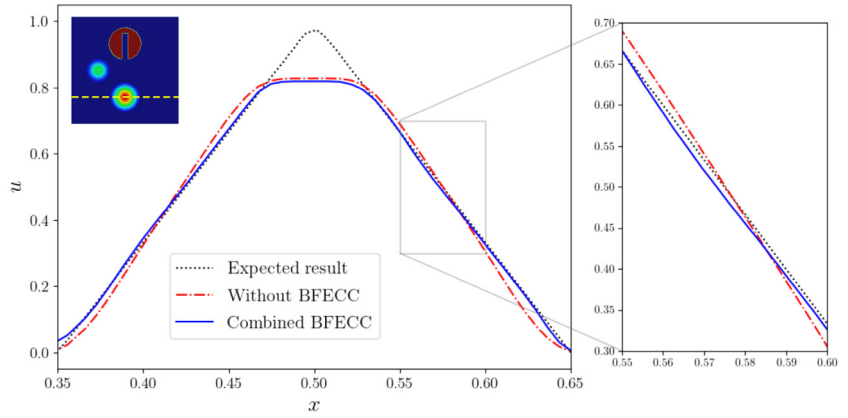
Fig. 11. Solid-body rotation at $t = 6.28$. Results are obtained using the *enhanced* underlying scheme on the unstructured mesh and presented for the nodes lie on (a) $x = 0.5$ and (b) $y = 0.75$ cut-lines.

the vicinity of the local extrema. In addition to a strong reduction in the error, it was proved that the proposed algorithm substantially increases the rate of mesh-convergence; it was almost doubled upon the application of the presented BFECC algorithm to a *low-order* scheme. In all cases, the compliance of the results with the positivity and maximum principle was observed.

All the results presented in this work were obtained utilizing an explicit scheme (forward Euler discretization in time). Considering that the coefficient of unknowns incorporated only a combination of the consistent and the lumped mass matrix, the associated computational effort was rather low. Moreover, at each time-step, the proposed algorithm requires a fixed number of (four) sub-steps to estimate the error and do the correction. Therefore, in case the contribution of the consistent mass matrix is neglected, a fully explicit method would be obtained. Taking into account that by applying the proposed BFECC algorithm to the presented enhanced Eulerian scheme, the resulting error in the benchmark problem was comparable to that of the state-of-the-art numerical methods, this work provided an alternative to the nonlinear approaches developed to address convection-dominated transport problems. It must be noted that the application of the proposed algorithm is not limited to the underlying schemes presented in this work; in a wider view point, this algorithm can also be customized to be applied to numerical techniques other than the finite element method.

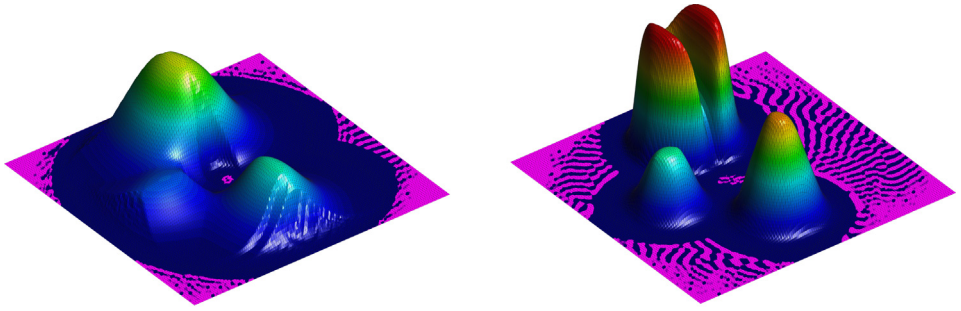


(a) $y = 0.5$



(b) $y = 0.25$

Fig. 12. Solid-body rotation at $t = 6.28$. Results are obtained using the *enhanced* underlying scheme on the unstructured mesh and presented for the nodes lie on (a) $x = 0.5$ and (b) $y = 0.75$ cut-lines.



$u \in [-6.6 \times 10^{-6}, 0.6267]$ $u \in [-1.1 \times 10^{-5}, 0.9307]$

(a) Without BFECC (b) Combined BFECC

Fig. 13. Solid-body rotation at $t = 6.28$. Results are obtained for the structured mesh and presented as surface $z = u(x, y)$; (a) and (b) correspond to the SUPG-CWS underlying scheme without and with the combined BFECC algorithm, respectively.

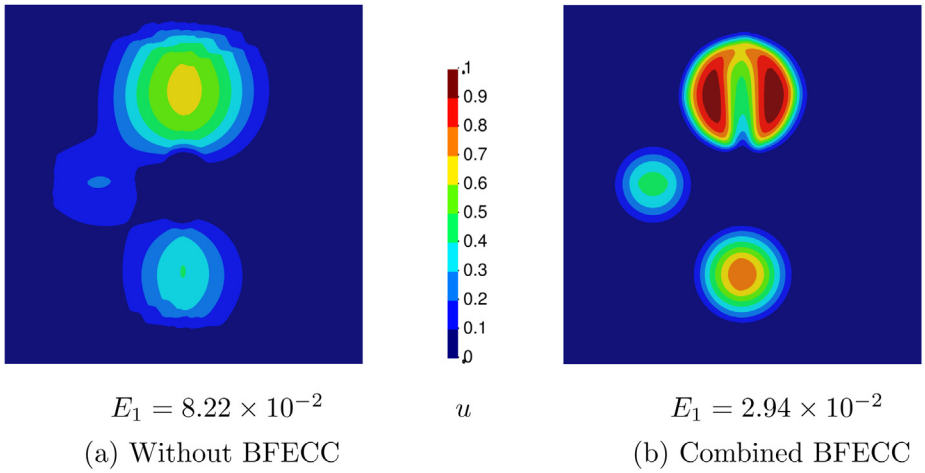


Fig. 14. Solid-body rotation at $t = 6.28$. Results are obtained for the structured mesh and presented as contours of $u(x, y)$; (a) and (b) correspond to the SUPG-CWS underlying scheme without and with the combined BFECC algorithm, respectively.

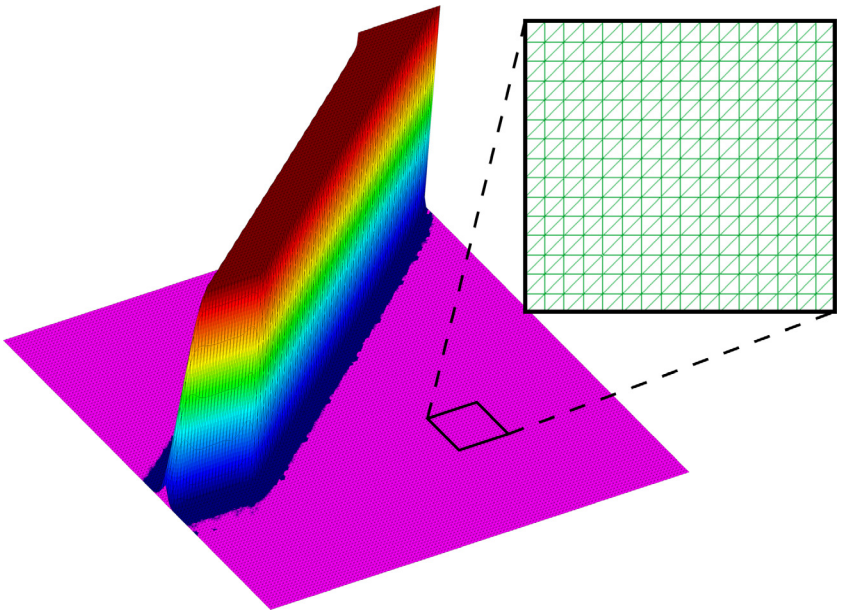


Fig. 15. Oblique inflow at $t = 1$, simulated using the *enhanced* underlying scheme with the proposed combined BFECC algorithm. The result is presented as surface $z = u(x, y)$.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Mohammad Reza Hashemi reports financial support was provided by Spain Ministry of Science and Innovation. Riccardo Rossi reports financial support was provided by Spain Ministry of Science and Innovation. Pavel Ryzhakov reports financial support was provided by Spain Ministry of Science and Innovation.

Acknowledgments

This work was performed within the framework of AMADEUS project (”Advanced Multi-scAle moDEling of coupled mass transport for improving water management in fUel cellS”, reference number PGC2018-101655-B-I00) supported by the *Ministerio de Ciencia, Innovacion e Universidades* of Spain. The authors also acknowledge financial support of the mentioned Ministry via the “Severo Ochoa Programme” for Centres of Excellence in R&D (referece: CEX2018-000797-S) given to the International Centre for Numerical Methods in Engineering (CIMNE).

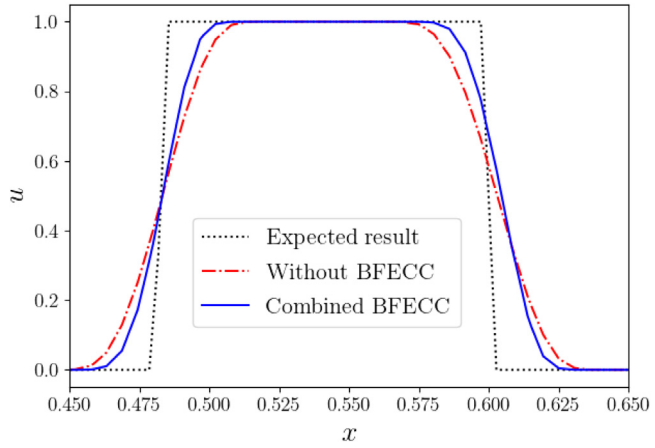


Fig. 16. Oblique inflow at $t = 1$, simulated using the *enhanced* underlying scheme with and without the proposed combined BFECC algorithm. Results are presented along a cut-line perpendicular to the stream ($y = 1 - 4x/3$).

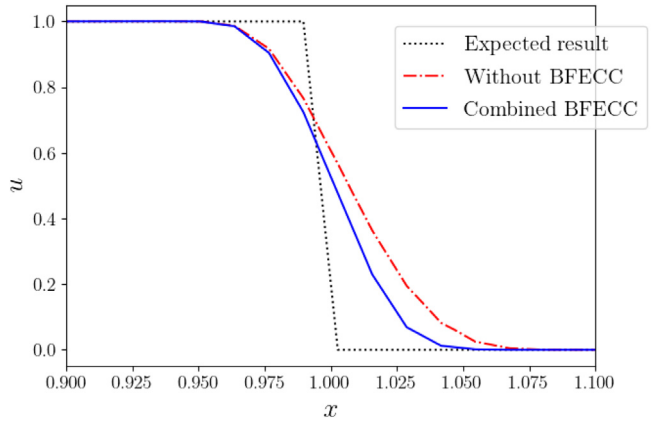


Fig. 17. Oblique inflow at $t = 1$, simulated using the *enhanced* underlying scheme with the BFECC algorithm. Results are presented along a cut-line parallel to the stream ($y = 0.325 + 3x/4$).

Appendix A. Semi-Lagrangian approach

The unconditionally stable CIR scheme [27,29,41,42], which is named after Courant, Isaacson, and Rees [43], depicts the constructive idea of the semi-Lagrangian approach for solving hyperbolic differential equations; the solution at $(\mathbf{x}, t)$ is obtained by following the corresponding characteristic line to reach $(\mathbf{x}', t - \Delta t)$ in the spatial–temporal space [44].

For Eq. (2) the CIR scheme reads

$$u(\mathbf{x}, t) = u(\mathbf{x} - \Delta t \mathbf{v}, t - \Delta t). \tag{A.1}$$

This scheme is temporally and spatially first-order [27]; nevertheless, it can be further enhanced to obtain a second-order solver [42] by acquiring non-linear interpolation schemes, which is beyond the scope of the present work. It should be noted that this scheme relies on the spatial search within the computational domain and consequently, in cases that the characteristic line points to the outside of the domain, the implementation of the solution algorithm is not straightforward. This issue specifically occurs in the vicinity of the inlet and curved boundaries.

Appendix B. Comment on enhanced scheme implementation

The *enhanced* scheme is based on the implementation of Eq. (49) that by using the forward Euler time discretization, reads

$$\frac{1}{dt}\mathbb{M}\mathbf{U}_{n+1} = \left(\frac{1}{dt}\mathbb{M} + \mathbb{C} + \mathbb{D}\right)\mathbf{U}_n - \mathbf{F}_n = 0, \tag{B.1}$$

where the elemental contributions are assembled as

$$\mathbb{M} = \mathcal{A}_{e\in\mathcal{E}}\left(\alpha^e\mathbb{M}_C^e + (1-\alpha^e)\mathbb{M}_L^e\right), \tag{B.2}$$

$$\mathbb{D} = \mathcal{A}_{e\in\mathcal{E}}\left(\nu^e\left[\mathbb{M}_L^e - \mathbb{M}_C^e\right]\right), \tag{B.3}$$

and

$$\mathbf{F} = \mathcal{A}_{e\in\mathcal{E}}\left(\alpha^e\hat{\mathbf{D}}^e\right). \tag{B.4}$$

In combination of the BFECC algorithm, Eq. (B.1) is solved in forward and backward convection steps, *i.e.* first, second, and fourth steps of the algorithms described in this paper.

References

[1] D. Kuzmin, S. Turek, High-resolution FEM-TVD schemes based on a fully multidimensional flux limiter, J. Comput. Phys. 198 (2004) 131–158, http://dx.doi.org/10.1016/j.jcp.2004.01.015, http://www.sciencedirect.com/science/article/pii/S0021999104000221.

[2] V. Le Chenadec, H. Pitsch, A monotonicity preserving conservative sharp interface flow solver for high density ratio two-phase flows, J. Comput. Phys. 249 (2013) 185–203, http://dx.doi.org/10.1016/j.jcp.2013.04.027, http://www.sciencedirect.com/science/article/pii/S0021999113002921.

[3] C.-T. Ha, J.H. Lee, A modified monotonicity-preserving high-order scheme with application to computation of multi-phase flows, Comput. & Fluids 197 (2020) 104345, http://dx.doi.org/10.1016/j.compfluid.2019.104345, https://www.sciencedirect.com/science/article/pii/S0045793019303044.

[4] V.S. Borisov, On discrete maximum principles for linear equation systems and monotonicity of difference schemes, SIAM J. Matrix Anal. Appl. 24 (2003) 1110–1135, http://dx.doi.org/10.1137/S0895479802409687, https://epubs-siam-org.recursos.biblioteca.upc.edu/doi/abs/10.1137/S0895479802409687, publisher: Society for Industrial and Applied Mathematics.

[5] J.-L. Guermond, M. Nazarov, B. Popov, Y. Yang, A second-order maximum principle preserving Lagrange finite element technique for nonlinear scalar conservation equations, SIAM J. Numer. Anal. 52 (2014) 2163–2182, http://dx.doi.org/10.1137/130950240, https://epubs.siam.org/doi/10.1137/130950240, publisher: Society for Industrial and Applied Mathematics.

[6] R. Codina, Comparison of some finite element methods for solving the diffusion-convection-reaction equation, Comput. Methods Appl. Mech. Engrg. 156 (1998) 185–210, http://dx.doi.org/10.1016/S0045-7825(97)00206-5, http://www.sciencedirect.com/science/article/pii/S0045782597002065.

[7] V. John, E. Schmeyer, Finite element methods for time-dependent convection–diffusion–reaction equations with small diffusion, Comput. Methods Appl. Mech. Engrg. 198 (2008) 475–494, http://dx.doi.org/10.1016/j.cma.2008.08.016, http://www.sciencedirect.com/science/article/pii/S0045782508003150.

[8] S. Badia, A. Hierro, On monotonicity-preserving stabilized finite element approximations of transport problems, SIAM J. Sci. Comput. 36 (2014) A2673–A2697, http://dx.doi.org/10.1137/130927206, https://epubs.siam.org/doi/10.1137/130927206, publisher: Society for Industrial and Applied Mathematics.

[9] D. Kuzmin, Monolithic convex limiting for continuous finite element discretizations of hyperbolic conservation laws, Comput. Methods Appl. Mech. Engrg. 361 (2020) 112804, http://dx.doi.org/10.1016/j.cma.2019.112804, http://www.sciencedirect.com/science/article/pii/S0045782519306966.

[10] J. Bonilla, S. Badia, Monotonicity-preserving finite element schemes with adaptive mesh refinement for hyperbolic problems, J. Comput. Phys. 416 (2020) 109522, http://dx.doi.org/10.1016/j.jcp.2020.109522, https://www.sciencedirect.com/science/article/pii/S0021999120302965.

[11] A.N. Brooks, T.J.R. Hughes, Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations, Comput. Methods Appl. Mech. Engrg. 32 (1982) 199–259, http://dx.doi.org/10.1016/0045-7825(82)90071-8, https://www.sciencedirect.com/science/article/pii/0045782582900718.

[12] R. Codina, A discontinuity-capturing crosswind-dissipation for the finite element solution of the convection–diffusion equation, Comput. Methods Appl. Mech. Engrg. 110 (1993) 325–342, http://dx.doi.org/10.1016/0045-7825(93)90213-H, http://www.sciencedirect.com/science/article/pii/004578259390213H.

[13] E. Burman, A. Ern, Nonlinear diffusion and discrete maximum principle for stabilized Galerkin approximations of the convection–diffusion-reaction equation, Comput. Methods Appl. Mech. Engrg. 191 (2002) 3833–3855, http://dx.doi.org/10.1016/S0045-7825(02)00318-3, https://www.sciencedirect.com/science/article/pii/S0045782502003183.

[14] V. John, P. Knobloch, On spurious oscillations at layers diminishing (SOLD) methods for convection–diffusion equations: Part I – A review, Comput. Methods Appl. Mech. Engrg. 196 (2007) 2197–2215, http://dx.doi.org/10.1016/j.cma.2006.11.013, https://www.sciencedirect.com/science/article/pii/S0045782506003926.
[15] V. John, P. Knobloch, On spurious oscillations at layers diminishing (SOLD) methods for convection–diffusion equations: Part II – analysis for P1 and Q1 finite elements, Comput. Methods Appl. Mech. Engrg. 197 (2008) 1997–2014, http://dx.doi.org/10.1016/j.cma.2007.12.019, https://www.sciencedirect.com/science/article/pii/S0045782508000182.
[16] E. Burman, On nonlinear artificial viscosity, discrete maximum principle and hyperbolic conservation laws, BIT Numer. Math. 47 (2007) 715–733, http://dx.doi.org/10.1007/s10543-007-0147-7.
[17] J.-L. Guermond, M. Nazarov, A maximum-principle preserving C0 finite element method for scalar conservation equations, Comput. Methods Appl. Mech. Engrg. 272 (2014) 198–213, http://dx.doi.org/10.1016/j.cma.2013.12.015.
[18] S. Badia, J. Bonilla, Monotonicity-preserving finite element schemes based on differentiable nonlinear stabilization, Comput. Methods Appl. Mech. Engrg. 313 (2017) 133–158, http://dx.doi.org/10.1016/j.cma.2016.09.035, http://www.sciencedirect.com/science/article/pii/S0045782516306405.
[19] D. Kuzmin, S. Turek, Flux correction tools for finite elements, J. Comput. Phys. 175 (2002) 525–558, http://dx.doi.org/10.1006/jcph.2001.6955, http://www.sciencedirect.com/science/article/pii/S0021999101969554.
[20] D. Kuzmin, Algebraic Flux Correction I, Dordrecht, 2012, pp. 145–192, http://dx.doi.org/10.1007/978-94-007-4038-9_6.
[21] D. Kuzmin, R. Lö, S. Turek, Flux-Corrected Transport: Principles, Algorithms, and Applications, Springer, 2012.
[22] G.R. Barrenechea, V. John, P. Knobloch, Analysis of algebraic flux correction schemes, SIAM J. Numer. Anal. 54 (2016) 2427–2451, http://dx.doi.org/10.1137/15M1018216, https://epubs.siam.org/doi/abs/10.1137/15M1018216, publisher: Society for Industrial and Applied Mathematics.
[23] G.R. Barrenechea, V. John, P. Knobloch, R. Rankin, A unified analysis of algebraic flux correction schemes for convection–diffusion equations, SeMA J. 75 (2018) 655–685, http://dx.doi.org/10.1007/s40324-018-0160-6.
[24] S. Godunov, I. Bohachevsky, Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics, Matematičeskij Sbornik 89 (1959) 271–306.
[25] L. Hu, Numerical Algorithms Based on the Back and Forth Error Compensation and Correction (Ph.D. thesis), Publisher: Georgia Institute of Technology, 2014, https://smartech.gatech.edu/handle/1853/54839, accepted: 2016-05-27T13:08:58Z.
[26] T.F. Dupont, Y. Liu, Back and forth error compensation and correction methods for removing errors induced by uneven gradients of the level set function, J. Comput. Phys. 190 (2003) 311–324, Publisher: Elsevier..
[27] A. Selle, R. Fedkiw, B. Kim, Y. Liu, J. Rossignac, An unconditionally stable MacCormack method, J. Sci. Comput. 35 (2008) 350–371, http://dx.doi.org/10.1007/s10915-007-9166-4.
[28] D. Kuzmin, Explicit and implicit FEM-FCT algorithms with flux linearization, J. Comput. Phys. 228 (2009) 2517–2534, http://dx.doi.org/10.1016/j.jcp.2008.12.011, http://www.sciencedirect.com/science/article/pii/S0021999108006475.
[29] T.F. Dupont, Y. Liu, Back and forth error compensation and correction methods for semi-Lagrangian schemes with application to level set interface computations, Math. Comp. (2007) 647–668, Publisher: JSTOR.
[30] L. Hu, Y. Li, Y. Liu, A limiting strategy for the back and forth error compensation and correction method for solving advection equations, Math. Comp. 85 (2016) 1263–1280, http://dx.doi.org/10.1090/mcom/3026, https://www.ams.org/mcom/2016-85-299/S0025-5718-2016-03026-4/.
[31] D. Kuzmin, J.N. Shadid, Gradient-based nodal limiters for artificial diffusion operators in finite element schemes for transport equations, Internat. J. Numer. Methods Fluids 84 (2017) 675–695, http://dx.doi.org/10.1002/fld.4365, http://onlinelibrary.wiley.com/doi/abs/10.1002/fld.4365, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/fld.4365.
[32] G.R. Barrenechea, E. Burman, F. Karakatsani, Edge-based nonlinear diffusion for finite element approximations of convection–diffusion equations and its relation to algebraic flux-correction schemes, Numer. Math. 135 (2017) 521–545, http://dx.doi.org/10.1007/s00211-016-0808-z.
[33] R.J. LeVeque, High-resolution conservative algorithms for advection in incompressible flow, SIAM J. Numer. Anal. 33 (1996) 627–665, http://dx.doi.org/10.1137/0733033, https://epubs-siam-org.recursos.biblioteca.upc.edu/doi/abs/10.1137/0733033, publisher: Society for Industrial and Applied Mathematics.
[34] J. Donea, A. Huerta, Finite Element Methods for Flow Problems, John Wiley & Sons, 2003.
[35] D. Kuzmin, S. Basting, J.N. Shadid, Linearity-preserving monotone local projection stabilization schemes for continuous finite elements, Comput. Methods Appl. Mech. Engrg. 322 (2017) 23–41, http://dx.doi.org/10.1016/j.cma.2017.04.030, http://www.sciencedirect.com/science/article/pii/S0045782516302262.
[36] V. John, S. Kaya, W. Layton, A two-level variational multiscale method for convection-dominated convection–diffusion equations, Comput. Methods Appl. Mech. Engrg. 195 (2006) 4594–4603, http://dx.doi.org/10.1016/j.cma.2005.10.006, https://www.sciencedirect.com/science/article/pii/S0045782505004457.
[37] D. Kuzmin, On the design of general-purpose flux limiters for finite element schemes. i. scalar convection, J. Comput. Phys. 219 (2006) 513–531, http://dx.doi.org/10.1016/j.jcp.2006.03.034, http://www.sciencedirect.com/science/article/pii/S0021999106001902.
[38] C. Lohmann, D. Kuzmin, J.N. Shadid, S. Mabuza, Flux-corrected transport algorithms for continuous Galerkin methods based on high order Bernstein finite elements, J. Comput. Phys. 344 (2017) 151–186, http://dx.doi.org/10.1016/j.jcp.2017.04.059, http://www.sciencedirect.com/science/article/pii/S0021999117303388.
[39] J. Bonilla, S. Badia, Maximum-principle preserving space–time isogeometric analysis, Comput. Methods Appl. Mech. Engrg. 354 (2019) 422–440, http://dx.doi.org/10.1016/j.cma.2019.05.042, https://www.sciencedirect.com/science/article/pii/S0045782519303123.
[40] D. Kuzmin, M. Quezada de Luna, Subcell flux limiting for high-order Bernstein finite element discretizations of scalar hyperbolic conservation laws, J. Comput. Phys. 411 (2020) 109411, http://dx.doi.org/10.1016/j.jcp.2020.109411, http://www.sciencedirect.com/science/article/pii/S0021999120301856.

[41] J. Strain, A fast modular semi-Lagrangian method for moving interfaces, J. Comput. Phys. 161 (2000) 512–536, http://dx.doi.org/10.1006/jcph.2000.6508, https://www.sciencedirect.com/science/article/pii/S0021999100965082.

[42] C. Min, F. Gibou, A second order accurate projection method for the incompressible Navier–Stokes equations on non-graded adaptive grids, J. Comput. Phys. 219 (2006) 912–929, http://dx.doi.org/10.1016/j.jcp.2006.07.019, https://www.sciencedirect.com/science/article/pii/S0021999106003366.

[43] R. Courant, E. Isaacson, M. Rees, On the solution of nonlinear hyperbolic differential equations by finite differences, Comm. Pure Appl. Math. 5 (1952) 243–255, http://dx.doi.org/10.1002/cpa.3160050303, http://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.3160050303, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpa.3160050303.

[44] J. Strain, Semi-Lagrangian methods for level set equations, J. Comput. Phys. 151 (1999) 498–533, http://dx.doi.org/10.1006/jcph.1999.6194, http://www.sciencedirect.com/science/article/pii/S0021999199961946.