Flux correction for nonconservative convection-diffusion equation

Sergii Kivva

Received: date / Accepted: date

Abstract Our goal is to develop a flux limiter of the Flux-Corrected Transport method for a nonconservative convection-diffusion equation. For this, we consider a hybrid difference scheme that is a linear combination of a monotone scheme and a scheme of high-order accuracy. The flux limiter is computed as an approximate solution of a corresponding optimization problem with a linear objective function. The constraints for this optimization problem are derived from inequalities that are valid for the monotone scheme and apply to the hybrid scheme. Our numerical results with the flux limiters, which are exact and approximate solutions to the optimization problem, are in good agreement.

Keywords flux-corrected transport · nonconservative convection-diffusion equation · difference scheme · linear programming

Mathematics Subject Classification (2010) MSC 65M06 · MSC 65M08

1 Introduction

The objective of this paper is to develop a flux limiter for the flux-corrected transport (FCT) method for a nonconservative convection-diffusion equation. The numerical solution of such equations arises in a variety of applications such as hydrodynamics, heat, and mass transfer. To the best of our knowledge, we are not aware of any formulas for computing the FCT flux limiter for a nonconservative convection-diffusion equation.

On an interval \([a, b]\), we consider the initial boundary value problem (IBVP) for a nonconservative convection-diffusion equation

\[
\frac{\partial \rho}{\partial t} + u(x,t) \frac{\partial \rho}{\partial x} + \lambda(x,t) \rho = \frac{\partial}{\partial x} \left( D(x,t) \frac{\partial \rho}{\partial x} \right) + f(x,t), \quad t > 0 \quad (1.1)
\]

Sergii Kivva
Institute of Mathematical Machines and System Problems, National Academy of Sciences, Kyiv, Ukraine
E-mail: skivva@gmail.com
with initial condition
\[ \rho(x, 0) = \rho^0(x) \]
(1.2)
where \( 0 \leq D(x, t) \leq \mu = \text{const.} \).

For simplicity and without loss of generality, we assume that the Dirichlet boundary conditions are specified at the ends of the interval \([a, b]\)
\[ \rho(a, t) = \rho_a(t) \]
(1.3)
\[ \rho(b, t) = \rho_b(t) \]
(1.4)

The two-step FCT algorithm was firstly developed by Boris and Book [1] for solving a transient continuity equation. Within this approach, the flux at
the cell interface is computed as a convex combination of fluxes of a monotone low-order scheme and a high-order scheme. These two fluxes are combined by adding to one of them (basic flux) a limited flux that is the limited difference between the high-order and low-order fluxes at the cell interface. In the classical FCT method, the low-order flux is basic and the additional limited flux is antidiffusive. Kuzmin and his coworkers [4,5] consider the high-order flux as the basic with an additional dissipative flux. Such approach is now known as algebraic flux correction (AFC). The procedure of two-step flux correction consists of computing the time advanced low order solution in the first step and correcting the solution in the second step to produce accurate and monotone results. The basic idea is to switch between high-order scheme and positivity preserving low-order scheme to provide oscillation free good resolution in steep gradient areas, while at the same time preserve at least second-order accuracy in smooth regions. Later Zalesak [10,11] extended FCT to multidimensional explicit difference schemes. Since the 1970s, FCT has been widely used in the modeling of various physical processes. Many variations and generalizations of FCT and their applications are given in [8].

In this paper, we derive the flux correction formulas for the nonconservative convection-diffusion equation using the approach proposed in [3]. As in the classical FCT method, we use a hybrid difference scheme consisting of a convex combination of low-order monotone and high-order schemes. According to [3], finding the flux limiters we consider as a corresponding optimization problem with a linear objective function. The constraints for the optimization problem derive from the inequalities which are valid for the monotone scheme and apply to the hybrid scheme. The flux limiters are obtained as an approximate solution to the optimization problem. Numerical results show that these flux limiters produce numerical solutions that are in good agreement with the numerical solutions, the flux limiters of which are calculated from optimization problem and correspond to maximal antidiffusive fluxes.

The advantage of such approach is that the two-step classical FCT method is reduced to one-step. For flux corrections in the classical FCT method, it is necessary to know the low-order numerical solution at the current time step. In the proposed approach [3], it is sufficient to know only the numerical solution at the previous time step.
The paper is organized as follows. In Section 2, we discretize the IVBP (1.1)-(1.4) by a hybrid scheme. An analog of the discrete local maximum principle for the monotone scheme is given in Section 3. The optimization problem for finding flux limiters and the algorithm of its solving are described in Section 4. An approximate solution of the optimization problem is derived in Section 5. The results of numerical experiments are presented in Section 6. Concluding remarks are drawn in Section 7.

2 Hybrid difference scheme

In this section, we discretize the IBVP (1.1)-(1.4) using a hybrid difference scheme, which is a linear combination of a monotone scheme and a high-order scheme.

On the interval \([a, b]\), we introduce a nonuniform grid \(\Omega_h\)

\[
\Omega_h = \{ x_i : x_i = x_{i-1} + \Delta x_{i-1/2}, i = 1, N; x_0 = a, x_{N+1} = b \} \tag{2.1}
\]

Assuming that \(u(x, t)\) and \(\rho(x, t)\) are sufficiently smooth, we consider some approximations of the convective term in (1.1). For this, we integrate it on an interval \([x_{i-1/2}, x_{i+1/2}]\) and applying the rectangular approximation method at the point \(x_i\), as well as backward and forward differencing for the first-order derivative, we obtain the following upwind discretization

\[
\int_{x_{i-1/2}}^{x_{i+1/2}} u \frac{\partial \rho}{\partial x} dx = \Delta x_i \left[ \left( u^+ \frac{\partial \rho}{\partial x} \right)_i + \left( u^- \frac{\partial \rho}{\partial x} \right)_i \right] \tag{2.2}
\]

where \(\rho_i = \rho(x_i, t); \Delta x_i = (x_{i+1} - x_{i-1})/2\) is the spatial size of the \(i\)th cell; \(u^\pm = (u \pm |u|)/2\).

Applying the left and right rectangular rules for numerical integration and central differencing for the first-order derivative, we have another form of upwind discretization

\[
\int_{x_{i-1/2}}^{x_{i+1/2}} \frac{\partial \rho}{\partial x} dx = \int_{x_{i-1/2}}^{x_{i+1/2}} u^+ \frac{\partial \rho}{\partial x} dx + \int_{x_{i-1/2}}^{x_{i+1/2}} u^- \frac{\partial \rho}{\partial x} dx \tag{2.3}
\]

To obtain an approximation of a higher order, in the rectangular approximation rule at a point \(x_i\), we use central differencing for the first-order deriva-
\[ \int_{x_{i-1/2}}^{x_{i+1/2}} u \frac{\partial \rho}{\partial x} \, dx = \frac{\Delta x_i}{2} \left[ \rho_{i+1} - \rho_{i-1} \right] + M \Delta x_i \left( \Delta i_{i+1/2} \rho - \Delta i_{i-1/2} \rho \right) + O \left( \Delta x_i^3 \right) \]  

(2.4)

where \( M = \text{const.} \).

Applying the trapezoidal rule for numerical integration and central differencing for the first-order derivative, we obtain

\[ \int_{x_{i-1/2}}^{x_{i+1/2}} u \frac{\partial \rho}{\partial x} \, dx = \frac{\Delta x_i}{2} \left[ \frac{\rho_{i+1} - \rho_{i-1}}{\Delta i_{i+1/2} \rho} + \frac{\rho_{i+1} - \rho_i}{\Delta i_{i+1/2} \rho} \right] + O \left( \Delta x_i^3 \right) \]

(2.5)

Besides, we rewrite the convective term in (1.1) as follows:

\[ u \frac{\partial \rho}{\partial x} = \frac{\partial}{\partial x} (up) - \rho \frac{\partial u}{\partial x} \]

(2.6)

We discretize the terms on the right-hand side of (2.6) by the following difference relations

\[ \int_{x_{i-1/2}}^{x_{i+1/2}} \frac{\partial (up)}{\partial x} \, dx = u_{i+1/2}^{+} \rho_{i+1} + u_{i+1/2}^{-} \rho_{i+1+1} - u_{i-1/2}^{+} \rho_{i-1} + O \left( \Delta x_i \right) \]

(2.7)

\[ \int_{x_{i-1/2}}^{x_{i+1/2}} \frac{\partial (up)}{\partial x} \, dx = \frac{1}{2} u_{i+1/2} (\rho_{i+1} - \rho_{i+1}) - u_{i-1/2} (\rho_{i-1} + \rho_{i}) + O \left( \Delta x_i^2 \right) \]

(2.8)

\[ \int_{x_{i-1/2}}^{x_{i+1/2}} \rho \frac{\partial u}{\partial x} \, dx = \rho_i (u_{i+1/2} - u_{i-1/2}) + O \left( \Delta x_i^2 \right) \]

(2.9)

Using a convex combination of (2.7) and (2.8) to approximate the divergent term in (2.6), we discretize the convective term as

\[ \int_{x_{i-1/2}}^{x_{i+1/2}} u \frac{\partial \rho}{\partial x} \, dx = \left[ u_{i+1/2}^{+} \rho_{i+1} + u_{i+1/2}^{-} \rho_{i+1+1} + \beta_{i+1/2} \frac{u_{i+1/2}}{2} (\rho_{i+1} - \rho_i) \right. \]

\[ - u_{i-1/2}^{+} \rho_{i-1} - u_{i-1/2}^{-} \rho_{i-1+1} - \beta_{i-1/2} \frac{u_{i-1/2}}{2} (\rho_i - \rho_{i-1}) \] \[ - \rho_i (u_{i+1/2} - u_{i-1/2}) \]

(2.10)
Flux correction for nonconservative convection-diffusion equation

where $\beta_{i+1/2}$ is the flux limiter for the divergent part in square brackets of the convective flux. For a flux correction of the convective term in the divergent form, we refer to [3].

Below, to approximate the convective term in [1,1], we apply a convex combination of (2.3) and (2.5). Note that for $\sigma$ where $\beta_{i+1/2}$ is the flux limiter for the divergent part in square brackets of the convective flux. For a flux correction of the convective term in the divergent form, we refer to [3].

For a flux correction of the convective term in the divergent form, we refer to [3].

$\frac{1}{2} \left[ u_{i+1/2} \left( \rho_{i+1} - \rho_i \right) / \Delta x_{i+1/2} + u_{i-1/2} \left( \rho_i - \rho_{i-1} \right) / \Delta x_{i-1/2} \right] + u_{i-1/2} \left( \rho_i - \rho_{i-1} \right) / \Delta x_{i-1/2} / 2 = \frac{u_{i+1/2} \left( \rho_{i+1} - \rho_{i-1} \right) / \Delta x_{i+1/2}}{2} - \frac{u_{i-1/2} \left( \rho_i - \rho_{i-1} \right) / \Delta x_{i-1/2}}{2} \right) \right) (2.11)

The second term in square brackets on the right-hand side of (2.11) can be considered as an anti-diffusion.

We approximate [1,1] by the following weighted difference scheme

\[
\frac{y_{i+1} - y_i}{\Delta t} + h_{i+1/2}^-(\sigma) + h_{i-1/2}^+(\sigma) + (\lambda y)_i^{(\sigma)} = f_i^{(\sigma)} \quad (2.12)
\]

where $y^n_i = y(x_i, t^n)$ is the grid function on $\Omega_h$; $\Delta t$ is the time step; $f_i^{(\sigma)} = \sigma f_i^{n+1} + (1 - \sigma) f_i^n, \sigma \in [0, 1]$. The numerical flux $h_{i+1/2}^{\pm,n}$ is written in the form

\[
h_{i+1/2}^{\pm,n} = 0.5 \left[ (u_{i+1}^n + d_i^{\pm,n} - a_i^{\pm,n} r_i^{\pm,n}) / \Delta x_{i+1/2} \right) \Delta t = 2 \rho^n_{i+1/2} - \rho^n_{i+1/2} \quad (2.13)
\]

where $a_i^{\pm,n} \in [0, 1]$ is the flux limiter; $\Delta x_{i+1/2} = y_{i+1} - y_i^{n}$; the coefficients $d_i^{\pm,n}$ and $r_i^{\pm,n}$ are computed as

\[
d_i^{\pm,n} = \pm \max \left( 0, \frac{D_i^{\pm,n}}{\Delta x_i} - \frac{\left| y_{i+1/2}^{n+1} - y_i^{n+1} \right|}{2} \right) \quad (2.14)
\]

\[
r_i^{\pm,n} = \mp \min \left( 0, \frac{D_i^{\pm,n}}{\Delta x_i} - \frac{\left| y_{i+1/2}^{n+1} - y_i^{n+1} \right|}{2} \right) \quad (2.15)
\]

Note that for $\sigma = 0$ scheme (2.12) is explicit and implicit for $\sigma > 0$. Let us denote by $y^n_i$ and $y^{n+1}_i$ the values of $\rho(x, t)$ at the left and right ends of the interval $[a, b]$ at time $t^n$.

We rewrite the difference scheme (2.12) in matrix form as

\[
\left[ E + \Delta t \sigma (A^{n+1} + A^n) \right] y^{n+1} = \Delta t \left[ (B^- \alpha^-)^{(\sigma)} + (B^+ \alpha^+)^{(\sigma)} \right] + \Delta t g^{(\sigma)}
\]

(2.16)

where $(B^\pm \alpha^\pm)^{(\sigma)} = \sigma B_{n+1}^\pm \alpha_{n+1}^\pm + (1 - \sigma) B_{n}^\pm \alpha_{n}^\pm; B_{n+1} = \text{diag} \{ h_i^n \}^N$ is the diagonal matrix; $E$ is the identity matrix of order $N$; $A = \{a_{ij}\}_{i,j=1}^N$ is tridiagonal square matrices of order $N$; $A = \text{diag}(\lambda_1, \ldots, \lambda_N)$ is the diagonal
matrix; \( \alpha^\pm = (\alpha^+_1, \ldots, \alpha^+_N)^T \in R^N \) are the numerical vectors of flux limiters; \( g = (g_1, \ldots, g_N)^T \) is the vector of boundary conditions and values of the function \( f \) at the points \( x_i \). Components of the vector \( g \) are given by

\[
g_1 = \frac{(u_1^+ + d_1^+)y_0}{\Delta_1/2x} + f_1; \quad g_1 = f_1; \quad g_N = \frac{-(u_{N+1}^- + d_N^-)y_{N+1}}{\Delta_{N+1}/2} + f_N \quad (2.17)
\]

Elements of the matrices \( A \) and \( B^\pm \) are calculated as

\[
a_{i-1} = \frac{-u_{i-1}^+ - d_i^-}{\Delta_{i-1}/2x}; \quad b_i^+ = \frac{y_i - y_{i-1}}{\Delta_{i-1}/2x} \\
\]

\[
a_{i+1} = \frac{-u_i^+ - d_i^-}{\Delta_{i+1}/2x}; \quad b_i^- = \frac{y_i - y_{i+1}}{\Delta_{i+1}/2x} \quad (2.18)
\]

3 Monotone difference scheme

We consider the system of equations (2.16) for \( \alpha^\pm, n, \alpha^{\pm, n+1} = 0 \)

\[
\begin{align*}
[E + \Delta t \sigma A^{n+1} + \Delta t \sigma A^{n+1}]y^{n+1} - \Delta t \sigma g^{n+1} \\
= [E - \Delta t (1 - \sigma)A^{n} - \Delta t (1 - \sigma)A^{n}] y^{n} + \Delta t (1 - \sigma)g^{n} \\
\end{align*} \quad (3.1)
\]

In this section, we obtain the monotonicity condition for the difference scheme (3.1) and derive for it an analog of the discrete local maximum principle, which plays a key role in the flux correction design.

**Definition 3.1** ([2]) A difference scheme

\[
y_i^{n+1} = H(y_{i-k}, y_{i-k+1}, \ldots, y_{i}, \ldots, y_{i+l}) \quad (3.2)
\]

is said to be monotone if \( H \) is a monotone increasing function of each of its arguments.

**Theorem 3.1** If \( \Delta t \) satisfies

\[
\Delta t \sigma \min_{1 \leq i \leq N} \lambda_{i}^{n+1} < 1 \quad (3.3)
\]

\[
\Delta t (1 - \sigma) \max_{1 \leq i \leq N} \left[ \frac{-u_{i-1}^{n} - d_i^{n}}{\Delta_{i-1}/2x} - \frac{u_i^{n} + d_i^{n}}{\Delta_{i+1}/2x} + \lambda_i^{n} \right] \leq 1 \quad (3.4)
\]

then the difference scheme (3.1) is monotone.

**Proof** If (3.3) holds, the matrix \( [E + \Delta t \sigma (A^{n+1} + A^{n+1})] \) is a strictly row diagonally dominant M-matrix. Then the inverse matrix \( [E + \Delta t \sigma (A^{n+1} + A^{n+1})]^{-1} \) is a matrix with nonnegative elements.

The nonnegativity of the elements of matrix \( [E + \Delta t \sigma (A^{n+1} + A^{n+1})]^{-1} \times [E - \Delta t (1 - \sigma) (A^{n} + A^{n})] \) and, hence, the monotonicity of the scheme (3.1) follows from the nonnegativity of the elements \( [E - \Delta t (1 - \sigma) (A^{n} + A^{n})] \) for \( \Delta t \) satisfying (3.4).
**Theorem 3.2** If $\Delta t$ satisfies

\[
\Delta t(1 - \sigma) \max_{1 \leq i \leq N} \left[ \frac{u_{i+1/2}^n + d_{i+1}^n}{\Delta x} - \frac{u_{i-1/2}^n + d_{i}^n}{\Delta x} \right] \leq 1, \tag{3.5}
\]

then the numerical solution of the system of equations (3.1) satisfies the following inequalities

\[
\begin{align*}
\min_{k \in S_i} y_{ik}^n &= -\Delta t(1 - \sigma)\lambda_i^n y_{ik}^n + \Delta t(1 - \sigma)f_{ik}^n \\
&\leq y_{i+1}^n + \Delta t\sigma \sum_j a_{ij} \gamma_{ij}^{n+1} + \Delta t\sigma \lambda_i y_{i+1}^n + \Delta t\sigma g_{i+1}^n \\
&\leq \max_{k \in S_i} y_{ik}^n - \Delta t(1 - \sigma)\lambda_i^n y_{ik}^n + \Delta t(1 - \sigma)f_{ik}^n 
\end{align*} \tag{3.6}
\]

where $S_i$ is the stencil of the difference scheme (3.1) for an $i$th grid node.

**Proof** Let us prove the right-hand side of inequality (3.6). We rewrite the $i$th row of the system of equations (3.1) in the form

\[
y_{i+1}^n + \Delta t\sigma \sum_j a_{ij} \gamma_{ij}^{n+1} + \Delta t\sigma \lambda_i y_{i+1}^n + \Delta t\sigma g_{i+1}^n 
\]

Under condition (3.5), the first and third terms on the right-hand side of (3.7) are a convex linear combination, therefore

\[
y_{i+1}^n + \Delta t\sigma \sum_j a_{ij} \gamma_{ij}^{n+1} + \Delta t\sigma \lambda_i y_{i+1}^n + \Delta t\sigma g_{i+1}^n 
\]

(3.8)

The lower bound (3.6) is obtained in a similar way, which proves the theorem.

**Remark 3.1** Under condition (3.3), the matrix $G = [E + \Delta t \sigma(A^n + A^{n+1})]$ is a non-singular M-matrix, therefore $G^{-1}$ is a nonnegative and isotone matrix [9, p.52, 2.4.3], i.e. if $x \preceq y$, then $G^{-1}x \preceq G^{-1}y$. Here $\preceq$ denotes the natural (component-wise) partial ordering on $R^N$, i.e. $x \preceq y$ if and only if $x_i \leq y_i$ for all $i$. Thus, the change of the vector $y^{n+1}$ can be controlled by changing the right-hand side of the equation (3.1).

Inequalities (3.6) hold for the right-hand side of (3.1) and will be used to obtain restrictions on flux limiters in the scheme (2.16). We can consider (3.6) as an analogue of discrete local maximum principle for the scheme (3.1). Note that to obtain restrictions (3.6), it is sufficient for us to know the numerical solution of (3.1) at a previous time step.
4 Finding flux limiters

To find flux limiters for scheme (2.16), we implement the approach proposed in [3]. Our goal is to find maximal values of the flux limiters for which the solution of the difference scheme (2.16) is similar to the solution of the monotone difference scheme (3.1). For this, we require that the difference scheme (2.16) satisfies inequalities (3.6). Then finding the flux limiters can be considered as the following optimization problem

\[
\mathcal{I}(\alpha^{\pm,n}, \alpha^{\pm,n+1}) = \sum_{k=n}^{n+1} \sum_{i=1}^{N} \alpha^{\pm,k}_i + \sum_{k=n}^{n+1} \sum_{i=1}^{N} \alpha^{-k}_i \rightarrow \max_{\alpha^{\pm,n}, \alpha^{\pm,n+1} \in U_{ad}} \quad (4.1)
\]

subject to (2.16) and

\[
y^n + \Delta t(1-\sigma) f^n \leq [E - \Delta t(1-\sigma)A^n] y^n + \Delta t(B^+ \alpha^+ + B^- \alpha^-) \sigma + \Delta t(1-\sigma) g^n \leq \bar{y}^n + \Delta t(1-\sigma) f^n \quad (4.2)
\]

where \(y\) and \(\bar{y}\) are column vectors whose components are \(y_i = \min_{j} y_j\) and \(\bar{y}_i = \max_{j} y_j\). \(U_{ad}\) is the set of vectors \(\alpha^{\pm,n}, \alpha^{\pm,n+1}\), which is defined as the Cartesian product of \(N\)-vectors

\[
U_{ad} = \left\{ (\alpha^{\pm,n}, \alpha^{\pm,n+1}) \in (\mathbb{R}^N)^4 : \ 0 \leq \alpha^{\pm,k}_i \leq 1, \ k = n, n + 1 \right\} \quad (4.3)
\]

Note that for \(\sigma = 0\) the optimization problem (4.1)-(4.3) and (2.16) is a linear programming problem, and for \(\sigma > 0\) it is a nonlinear programming problem.

To solve the nonlinear optimization problem (4.1)-(4.3) and (2.16) in one time step, we use the following iterative process:

**Step 1.** Initialize positive numbers \(\delta, \varepsilon_1, \varepsilon_2 > 0\). Set \(p = 0\), \(y^{n+1,0} = y^n\), \(\alpha^{\pm,n,0}, \alpha^{\pm,n+1,0} = 0\).

**Step 2.** Find the solution \(\alpha^{\pm,n,p+1}, \alpha^{\pm,n+1,p+1}\) of the following linear programming problem

\[
\mathcal{I}(\alpha^{\pm,n,p+1}, \alpha^{\pm,n+1,p+1}) \rightarrow \max_{\alpha^{\pm,n,p+1}, \alpha^{\pm,n+1,p+1} \in U_{ad}} \quad (4.4)
\]

\[
\min_{j \in S_i} y_j^n - y_i^n + \Delta t(1-\sigma) \sum_{j \neq i} a^n_{ij} (y_j^n - y_i^n) \leq \Delta t(1-\sigma) \left( b^{n,\alpha^+ + n,p+1}_i a^{\alpha^+ + n,p+1}_i + b^{n,\alpha^- - n,p+1}_i a^{\alpha^- - n,p+1}_i \right) + \Delta t \sigma \left( b^{n,\alpha^+ + n+1,p+1}_i a^{\alpha^+ + n+1,p+1}_i + b^{n,\alpha^- - n+1,p+1}_i a^{\alpha^- - n+1,p+1}_i \right) \leq \max_{j \in S_i} y_j^n - y_i^n + \Delta t(1-\sigma) \sum_{j \neq i} a^n_{ij} (y_j^n - y_i^n) \quad (4.5)
\]
Step 3. For the $\alpha_{\pm,n,p+1}, \alpha_{\pm,n+1,p+1}$, find $y_{i}^{n+1,p+1}$ from the system of linear equations

$$
\begin{align*}
\left[ E + \Delta t \sigma \left( A_{n+1} + A_{n}^{(1)} \right) \right] y_{i}^{n+1,p+1} &= [E - \Delta t(1 - \sigma) (A_{n} + A_{n}^{(1)})] y_{i}^{n} \\
+ \Delta t \left[ \left( B^{+p} \alpha_{+,p+1}^{*} \right)^{(\sigma)} + \left( B^{-p} \alpha_{-,p+1}^{*} \right)^{(\sigma)} \right] + \Delta t g^{(\sigma)} \\
\end{align*}
$$

(4.6)

Step 4. Algorithm stop criterion

$$
\max_{i} \left| \frac{y_{i}^{n+1,p+1} - y_{i}^{n,p+1}}{\delta_{i}} \right| < \varepsilon_{1},
\max_{i} \left| \Im \left( \alpha_{\pm,n,p+1} \right) - \Im \left( \alpha_{\pm,n+1,p} \right) \right| < \varepsilon_{2}
$$

(4.7)

If conditions (4.7) hold, then $y_{i}^{n+1} = y_{i}^{n+1,p+1}$. Otherwise, set $p = p + 1$ and go to Step 2.

The solvability of the linear programming problem (4.4)-(4.5) is considered in the theorem below.

**Theorem 4.1** Assume that $\Delta t$ satisfies (3.3)-(3.5), then the linear programming problem (4.4)-(4.5) is solvable.

*Proof* To prove that problem (4.4)-(4.5) is solvable, it is sufficient to show that the objective function $\Im(\alpha_{\pm,n}^{*}, \alpha_{\pm,n+1}^{*})$ is bounded and the feasible set is non-empty. The boundedness of the function (4.4) follows from the boundedness of the vectors $\alpha_{\pm,n}$ and $\alpha_{\pm,n+1}$ whose coordinates vary from zero to one. On the other hand, if the hypothesis of the theorem is true, then the zero vectors $\alpha_{p,n}^{*}$ and $\alpha_{\pm,n+1}$ satisfy the system of inequalities (4.5).

This completes the proof of the theorem.

5 Flux limiter design

In the iterative process described in the previous section, the flux limiters are found by solving the linear programming problem (4.4)-(4.5). Solving a linear programming problem requires additional computational cost. Therefore, in the iterative process at Step 2 instead of (4.4)-(4.5), we use its approximate solution.

The purpose of this section is to find a nontrivial approximate solution to the linear programming problem (4.4)-(4.5). Nonzero $(\alpha_{\pm,n}^{*}, \alpha_{\pm,n+1}^{*}) \in U_{ad}$ satisfy the system of inequalities (4.5), and, omitting the iteration number, we rewrite the latter in the form

$$
(1 - \sigma) \left( b_{i}^{+n} \alpha_{i}^{+,n} + b_{i}^{-n} \alpha_{i}^{-,n} \right) + \sigma \left( b_{i}^{+n+1} \alpha_{i}^{+,n+1} + b_{i}^{-n+1} \alpha_{i}^{-,n+1} \right) \\
\leq \frac{1}{M} \left( \max_{j \in S_{i}} y_{j}^{n} - y_{i}^{n} \right) + (1 - \sigma) \sum_{j \neq i} a_{ij}^{n} \left( y_{j}^{n} - y_{i}^{n} \right)
$$

(5.1)
\[(1 - \sigma) \left( b_i^{\pm,n} a_i^{\pm,n} + b_i^{-,-n} a_i^{-,-n} \right) + \sigma \left( b_i^{\pm,n+1} a_i^{\pm,n+1} + b_i^{-,-n+1} a_i^{-,-n+1} \right) \]
\[\geq \frac{1}{M} \left( \min_{j \in S_i} y_j^n - y_i^n \right) + (1 - \sigma) \sum_{j \neq i} a_{ij}^n \left( y_j^n - y_i^n \right) \]  
\[0 \leq a_i^{\pm,n} \leq 1, \quad 0 \leq a_i^{\pm,n+1} \leq 1 \]  
\[(5.2)\]

For the left-hand sides of inequalities (5.1) and (5.2), the following estimates are valid
\[(1 - \sigma) \left( b_i^{\pm,n} a_i^{\pm,n} + b_i^{-,-n} a_i^{-,-n} \right) + \sigma \left( b_i^{\pm,n+1} a_i^{\pm,n+1} + b_i^{-,-n+1} a_i^{-,-n+1} \right) \]
\[\leq a_i^{\pm,max} \left[ (1 - \sigma) \left( \max(0, b_i^{\pm,n}) + \max(0, b_i^{-,-n}) \right) + \sigma \left( \max(0, b_i^{\pm,n+1}) + \max(0, b_i^{-,-n+1}) \right) \right] \]  
\[(5.4)\]
\[(1 - \sigma) \left( b_i^{\pm,n} a_i^{\pm,n} + b_i^{-,-n} a_i^{-,-n} \right) + \sigma \left( b_i^{\pm,n+1} a_i^{\pm,n+1} + b_i^{-,-n+1} a_i^{-,-n+1} \right) \]
\[\geq a_i^{-,-max} \left[ (1 - \sigma) \left( \min(0, b_i^{\pm,n}) + \min(0, b_i^{-,-n}) \right) + \sigma \left( \min(0, b_i^{\pm,n+1}) + \min(0, b_i^{-,-n+1}) \right) \right] \]  
\[(5.5)\]

where \(a_i^{\pm,max}\) and \(a_i^{-,-max}\) are the maximums of the components \(a_i^{\pm,n}\) and \(a_i^{-,-n+1}\) corresponding to the non-negative and non-positive coefficients \(b_i^{\pm}\) on the left-hand sides of (5.4) and (5.5), respectively.

Substituting (5.4) into (5.1), and (5.5) into (5.2) yields
\[a_i^{\pm,k} = \begin{cases} R_i^+ & b_i^{\pm,k} > 0 \\ R_i^- & b_i^{\pm,k} < 0 \end{cases} \quad k = n, n + 1 \]  
\[(5.6)\]

where
\[R_i^+ = \min(1, a_i^{\pm,max}) = \min(1, Q_i^+ / P_i^+) \]  
\[(5.7)\]
\[Q_i^+ = \frac{1}{M} \left( \max_{j \in S_i} y_j^n - y_i^n \right) + (1 - \sigma) \sum_{j \neq i} a_{ij}^n \left( y_j^n - y_i^n \right) \]  
\[(5.8)\]
\[Q_i^- = \frac{1}{M} \left( \min_{j \in S_i} y_j^n - y_i^n \right) + (1 - \sigma) \sum_{j \neq i} a_{ij}^n \left( y_j^n - y_i^n \right) \]  
\[(5.9)\]
\[P_i^+ = (1 - \sigma) \left( \max(0, b_i^{\pm,n}) + \max(0, b_i^{-,-n}) \right) + \sigma \left( \max(0, b_i^{\pm,n+1}) + \max(0, b_i^{-,-n+1}) \right) \]  
\[(5.10)\]
\[P_i^- = (1 - \sigma) \left( \min(0, b_i^{\pm,n}) + \min(0, b_i^{-,-n}) \right) + \sigma \left( \min(0, b_i^{\pm,n+1}) + \min(0, b_i^{-,-n+1}) \right) \]  
\[(5.11)\]

Remark 5.1 Note that similarly, the flux correction formulas can be obtained for the convex combination of (2.2) and (2.4), which approximates the convective term in equation (1.1). This approach is also applicable for schemes with a high-order approximation of the convective-diffusive flux. Moreover, this method and formulas (5.6)-(5.11) can be easily generalized to the multi-dimensional case.
6 Numerical Results

We conclude the paper with a number of numerical tests. The purpose of this section is to compare the results of the difference schemes considered in the paper. Below, we abbreviate by NDVL and NDVA the difference scheme (2.16), flux limiters of which are exact or approximate solutions of the linear programming problem (4.4)-(4.5). We also use DIV notation for the difference scheme, the flux correction of which is based on the divergent part of the convective flux (2.10).

In our calculations, we apply the GLPK (GNU Linear Programming Kit) v.4.65 set of routines for solving linear programming, mixed integer programming, and other related problem. GLPK is available at https://www.gnu.org/software/glpk/.

6.1 One-Dimensional Advection

We consider the one-dimensional advection test of Leonard et al. [6] on the uniform grid with $\Delta x = 0.01$ and constant velocity. The initial scalar profile consists of five different shapes: square wave, sine-squared, semi-ellipse, Gaussian, and triangle.

![Numerical results for the advection test](image)

**Fig. 1:** Numerical results for the advection test (6.1) with the NDVL scheme for various weights $\sigma$. Flux limiters are calculated using the linear programming problem (4.4)-(4.5).
Fig. 2: Comparison of the results for the advection test (6.1) with the NDVL and NDVA schemes for $\sigma = 0.5$.

Gaussian, and triangle. The initial profile is specified as

$$y(x_i) = \begin{cases} 
1 & \text{if } 0.05 \leq x_i \leq 0.25 \text{ (square wave)} \\
\sin^2 \left( \frac{\pi}{0.2} (x_i - 0.85) \right) & \text{if } 0.85 \leq x_i \leq 1.05 \text{ (sine-squared)} \\
\left(1 - \frac{1}{15\Delta x} (x_i - 1.75)^2 \right)^2 & \text{if } 1.6 \leq x_i \leq 1.9 \text{ (semi-ellipse)} \\
\exp \left[ -\frac{1}{2\gamma^2} (x_i - 2.65)^2 \right] & \text{if } 2.6 \leq x_i \leq 2.7 \text{ (Gaussian)} \\
10(x_i - 3.3) & \text{if } 3.3 \leq x_i \leq 3.4 \text{ (triangle)} \\
1.0 - 10(x_i - 3.4) & \text{if } 3.4 \leq x_i \leq 3.5 \\
0 & \text{otherwise} \\
\end{cases}$$

(6.1)

The standard deviation for the Gaussian profile is specified as $\gamma = 2.5$.

Numerical results with the NDVL scheme after 400 time steps at a Courant number of 0.2 are shown in Fig. 1. The flux limiters are calculated using the linear programming problem (4.4)-(4.5). At the right edge of the semi-ellipse for $\sigma = 0$ and $\sigma = 0.5$, we observe the well-known "terracing" phenomenon, which is a nonlinear effect of residual phase errors. It is shown in [8], [11] that high-order FCT methods (above fourth-order) significantly reduce phase errors and that selective adding diffusion can also reduce terracing. In the numerical solution of the implicit scheme, there is no terracing. The implicit scheme is more diffusive than the previous two, and its numerical solution is also more diffusive.

The Gaussian test problem has a single moving maximum and shows the effects of "clipping" the solution. This is because the flux limiter cannot account for the true peak of the Gaussian as it passes between the grid points. The maximum is clipped less as the order of the algorithm increases. The key to good performance here is the application of a more flexible limiter and a more accurate estimate of the allowable upper and lower bounds on the solution [10, 11].

The numerical results for which the flux limiters are calculated using exact and approximate solutions of the linear programming problem (4.4)-(4.5)
Table 1: $L^1$-norm of errors and the maximum values of the numerical results for the advection test (6.1) with the DIV, NDVL and NDVA schemes.

| $\sigma$ | DIV $L^1$ error | $y_{max}$ | NDVL $L^1$ error | $y_{max}$ | NDVA $L^1$ error | $y_{max}$ |
|----------|-----------------|-----------|-----------------|-----------|-----------------|-----------|
| wav      |                 |           |                 |           |                 |           |
| 0.0      | $2.1811 \times 10^{-2}$ | 1.0000 | $8.1136 \times 10^{-2}$ | 1.0000 | $8.1182 \times 10^{-2}$ | 1.0000 |
| 0.5      | $4.3933 \times 10^{-2}$ | 0.9997 | $6.5511 \times 10^{-2}$ | 0.9976 | $6.5527 \times 10^{-2}$ | 0.9973 |
| 1.0      | $6.9477 \times 10^{-2}$ | 0.9843 | $7.6861 \times 10^{-2}$ | 0.9653 | $7.6774 \times 10^{-2}$ | 0.9650 |
| sine     |                 |           |                 |           |                 |           |
| 0.0      | $1.6883 \times 10^{-2}$ | 0.9938 | $4.6661 \times 10^{-2}$ | 0.9913 | $4.7052 \times 10^{-2}$ | 0.9766 |
| 0.5      | $1.6423 \times 10^{-2}$ | 0.8895 | $3.2650 \times 10^{-2}$ | 0.7917 | $3.2759 \times 10^{-2}$ | 0.7899 |
| 1.0      | $3.9029 \times 10^{-2}$ | 0.7043 | $4.5601 \times 10^{-2}$ | 0.6300 | $4.5694 \times 10^{-2}$ | 0.6286 |
| elp      |                 |           |                 |           |                 |           |
| 0.0      | $1.7926 \times 10^{-2}$ | 0.973 | $4.9044 \times 10^{-2}$ | 0.9774 | $4.8959 \times 10^{-2}$ | 0.9775 |
| 0.5      | $1.7913 \times 10^{-2}$ | 0.9810 | $2.8675 \times 10^{-2}$ | 0.9526 | $2.8660 \times 10^{-2}$ | 0.9524 |
| 1.0      | $3.6078 \times 10^{-2}$ | 0.9601 | $3.9624 \times 10^{-2}$ | 0.9421 | $3.9603 \times 10^{-2}$ | 0.9422 |
| gau      |                 |           |                 |           |                 |           |
| 0.0      | $1.3639 \times 10^{-2}$ | 0.9764 | $6.9049 \times 10^{-2}$ | 0.8991 | $6.8116 \times 10^{-2}$ | 0.8661 |
| 0.5      | $2.7592 \times 10^{-2}$ | 0.6629 | $4.5363 \times 10^{-2}$ | 0.5438 | $4.5337 \times 10^{-2}$ | 0.5417 |
| 1.0      | $4.3681 \times 10^{-2}$ | 0.4828 | $4.8852 \times 10^{-2}$ | 0.3965 | $4.8882 \times 10^{-2}$ | 0.3949 |
| tri      |                 |           |                 |           |                 |           |
| 0.0      | $2.3205 \times 10^{-2}$ | 0.9389 | $4.8921 \times 10^{-2}$ | 0.8555 | $4.8870 \times 10^{-2}$ | 0.8517 |
| 0.5      | $1.3843 \times 10^{-2}$ | 0.8214 | $2.6126 \times 10^{-2}$ | 0.7404 | $2.6180 \times 10^{-2}$ | 0.7391 |
| 1.0      | $3.1245 \times 10^{-2}$ | 0.6655 | $3.7023 \times 10^{-2}$ | 0.6006 | $3.7123 \times 10^{-2}$ | 0.5991 |

wav = Square wave; sine = Sine-squared; elp = Semi-ellipse; gau = Gaussian; tri = Triangle.

are slightly different. Their $L^1$-norm of errors and the maximum values are

![Graph](image.png)

Fig. 3: Comparison of the numerical results for the advection test (6.1) with the DIV and NDVL schemes for $\sigma = 0.5$. 
presented in Table 1. The comparison of the NDVL and NDVA results with
\( \sigma = 0.5 \) is given in Fig. 2.

In Fig. 2, the solutions computed by the NDVL scheme are compared with
the DIV scheme. Their \( L^1 \)-norm of errors and the maximum values are pre-
sented in Table 1. Notice, that both the maximum values and the errors of
the DIV scheme are better than the corresponding maximum values and errors of
the NDVL scheme.

6.2 Solid Body Rotations

In this section, we consider the rotation of solid bodies \[7,8,10\] under an
incompressible flow that is described by the linear equation

\[
\frac{\partial \rho}{\partial t} + \mathbf{u} \cdot \nabla \rho = 0 \quad \text{in} \quad \Omega = (0,1) \times (0,1)
\]

with zero boundary conditions. The initial condition includes a slotted cylin-
der, a cone and a smooth hump (Fig. 4). The slotted cylinder of radius 0.15
and height 1 is centered at the point (0.5,0.75) and

\[
\rho(x,y,0) = \begin{cases} 
1 & \text{if } |x - 0.5| \geq 0.025 \text{ or } y \geq 0.85 \\
0 & \text{otherwise}
\end{cases}
\]

The cone of also radius \( r_0 = 0.15 \) and height 1 is centered at point \( (x_0, y_0) = (0.25, 0.5) \) and

\[
\rho(x,y,0) = 1 - r(x,y)
\]

where

\[
r(x,y) = \frac{\min(\sqrt{(x-x_0)^2 + (y-y_0)^2}, r_0)}{r_0}
\]

Fig. 4: Initial data and exact solution at the final time for the solid body
rotation test
The hump is given by

$$\rho(x, y, 0) = \frac{1}{4} (1 + \cos(\pi r(x, y)))$$

where \((x_0, y_0) = (0.5, 0.25)\) and \(r_0 = 0.1\).

Table 2: \(L^1\)-norm of errors and the maximum values of the numerical solutions for the solid body rotation test with the DIV, NDVL, and NDVA schemes

| \(
\sigma
\) | \(L^1\) error | \(y_{\text{max}}\) | \(L^1\) error | \(y_{\text{max}}\) | \(L^1\) error | \(y_{\text{max}}\) |
|---|---|---|---|---|---|---|
| 0.0 | 2.5900\times10^{-2} | 1.0000 | 4.189\times10^{-2} | 0.9959 | 4.4337\times10^{-2} | 0.9946 |
| Cyl | 0.5 | 2.0221\times10^{-2} | 0.9912 | 4.052\times10^{-2} | 0.9548 | 4.0256\times10^{-2} | 0.9547 |
| 1.0 | 3.9557\times10^{-2} | 0.9681 | 3.9751\times10^{-2} | 0.9141 | 3.9749\times10^{-2} | 0.9139 |
| Cn | 0.0 | 2.9773\times10^{-3} | 0.8709 | 3.419\times10^{-3} | 0.8144 | 3.4419\times10^{-3} | 0.8143 |
| 0.5 | 2.1664\times10^{-3} | 0.8434 | 2.6798\times10^{-3} | 0.8094 | 2.6799\times10^{-3} | 0.8092 |
| 1.0 | 2.4633\times10^{-3} | 0.8190 | 2.8654\times10^{-3} | 0.7905 | 2.8655\times10^{-3} | 0.7905 |
| Hm | 0.0 | 1.2495\times10^{-3} | 0.4947 | 2.1282\times10^{-3} | 0.4808 | 2.1283\times10^{-3} | 0.4804 |
| 0.5 | 1.2132\times10^{-3} | 0.4645 | 1.7634\times10^{-3} | 0.4248 | 1.7636\times10^{-3} | 0.4247 |
| 1.0 | 1.4077\times10^{-3} | 0.4247 | 1.7701\times10^{-3} | 0.3869 | 1.7703\times10^{-3} | 0.3868 |

Cyl = Slotted Cylinder; Cn = Cone; Hm = Hump.
The flow velocity is calculated by \( u(x, y) = (-2\pi(y - 0.5), 2\pi(x - 0.5)) \) and in result of which the counterclockwise rotation takes place about domain point \((0.5, 0.5)\). The computational grid consists of uniform \(128 \times 128\) cells. The exact solution of Eq. \((6.2)\) reproduces by the initial state after each full revolution.

The numerical results produced with the NDVL, NDVA, and DIV schemes after one full revolution (5000 time steps) with different weights \(\sigma\) are presented in Fig. 5. The \(L^1\)-norm of errors and the maximum values of the numerical results are given in Table 2. As in the above advection test, we also note a good agreement between the numerical results obtained with the NDVL and NDVA schemes. Again, the solution obtained by the DIV scheme is more accurate than the solutions computed by the NDVL and NDVA schemes.

7 Conclusions

In this paper, we derive the formulas for calculating flux limiters for the FCT method for a nonconservative convection-diffusion equation. The flux limiter is computed as an approximate solution of the optimization problem that can be considered as a background of the FCT approach.

Following FCT, we consider a hybrid scheme which is a linear combination of monotone and high-order schemes. The difference between high-order flux and low-order flux is considered as an antidiffusive flux. The finding maximal flux limiters for the antidiffusive fluxes is treated as an optimization problem with a linear objective function. Constraints for the optimization problem are inequalities that are valid for the monotone scheme and applied to the hybrid scheme. This approach allows us to reduce classical two-step FCT to a one-step method for explicit difference schemes and design flux limiters with desired properties.

Numerical experiments show the best results are obtained for the flux correction for the divergent part of the convective flux of a nonconservative convection-diffusion equation. We also note a good agreement between the numerical results for which the flux limiters are computed using exact and approximate solutions of optimization problem.

References

1. Boris, J.P., Book, D.L.: Flux-corrected transport. I. SHASTA, a fluid transport algorithm that works, J. Comput. Phys. 11, 38–69 (1973) https://doi.org/10.1016/0021-9991(73)90147-2
2. Harten, A., Hyman, J.M., Lax, P.D., Keyfitz, B.: On finite-difference approximations and entropy conditions for shocks, Comm. Pure Appl. Math. 29, 297–322 (1976) https://doi.org/10.1002/cpa.3160290305
3. Kivva, S.: Flux-corrected transport for scalar hyperbolic conservation laws and convection-diffusion equations by using linear programming. Journal of Computational Physics, 109874, (2020) In Press, https://doi.org/10.1016/j.jcp.2020.109874
4. Kuzmin, D.: Explicit and implicit FEM-FCT algorithms with flux linearization, J. Comput. Phys. 228, 2517–2534 (2009) https://doi.org/10.1016/j.jcp.2008.12.011
5. Kuzmin, D., Möller, M.: Algebraic Flux Correction I. Scalar Conservation Laws, in: Flux-Corrected Transp., 155–206 (2006) [https://doi.org/10.1007/3-540-27206-2_6]
6. Leonard, B.P., Lock, A.P., Macvean, M.K.: The nirvana scheme applied to one-dimensional advection, Int. J. Numer. Methods Heat Fluid Flow. 5, 341–377 (1995) [https://doi.org/10.1108/09615539510095170]
7. Leveque, R.J.: High-resolution conservative algorithms for advection in incompressible flow, SIAM J. Numer. Anal. 33, 627–665 (1996) [https://doi.org/10.1137/0733033]
8. Oran, E.S., Boris, J.P.: Numerical Simulation of Reactive Flow. Second Edition, Cambridge University Press (2001) [https://doi.org/10.1017/CBO9780511574474]
9. Ortega, J.M., Rheinboldt, W.C.: Iterative Solution of Nonlinear Equations in Several Variables, New York: Academic Press (1970)
10. Zalesak, S.T.: Fully multidimensional flux-corrected transport algorithms for fluids, J. Comput. Phys. 31, 335–362 (1979) [https://doi.org/10.1016/0021-9991(79)90051-2]
11. Zalesak, S.T.: The Design of Flux-Corrected Transport (FCT) Algorithms For Structured Grids, in: Flux-Corrected Transp., 29–78 (2006) [https://doi.org/10.1007/3-540-27206-2_3]