INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2009; **61**:864–887
Published online 17 December 2008 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1984

Insights on a sign-preserving numerical method for the advection–diffusion equation

E. Sousa*, †

CMUC, Department of Mathematics, University of Coimbra, 3001-454 Coimbra, Portugal

SUMMARY

In this paper we explore theoretically and numerically the application of the advection transport algorithm introduced by Smolarkiewicz to the one-dimensional unsteady advection–diffusion equation. The scheme consists of a sequence of upwind iterations, where the initial iteration is the first-order accurate upwind scheme, while the subsequent iterations are designed to compensate for the truncation error of the preceding step. Two versions of the method are discussed. One, the classical version of the method, regards the second-order terms of the truncation error and the other considers additionally the third-order terms. Stability and convergence are discussed and the theoretical considerations are illustrated through numerical tests. The numerical tests will also indicate in which situations are advantageous to use the numerical methods presented. Copyright © 2008 John Wiley & Sons, Ltd.

Received 10 January 2008; Revised 12 September 2008; Accepted 8 November 2008

KEY WORDS: advection–diffusion; non-oscillatory schemes; finite differences; MPDATA; sign preserving; computational fluid dynamics

1. INTRODUCTION

In numerical modeling of physical phenomena it is often necessary to solve the advective transport equation for positive-definite scalar functions, that is, we may wish to require that a non-negative variable remains non-negative under advection. Also, in some numerical simulation of fluids sign-preserving advection is often a necessary prerequisite of solution realizability [1, 2]. Numerical schemes of second- or higher-order accuracy can produce negative values in the solution due to the dispersive ripples. Lower-order schemes or higher-order schemes with zeroth-order diffusion added produce no ripples but suffer from excessive implicit diffusion.

Multi-dimensional positive-definite advection transport algorithm (Mpdata) was proposed in the early eighties as a simple positive-definite advection scheme for evaluating the advection of water

*Correspondence to: E. Sousa, CMUC, Department of Mathematics, University of Coimbra, 3001-454 Coimbra, Portugal.
†E-mail: ecs@mat.uc.pt

substance constituents in atmospheric cloud models [3] and further generalizations were presented in [4, 5]. Nowadays Mpdata has been used in different contexts and some review papers have been written about this method, see for instance [6, 7]. Several applications and comparisons with other methods can be found in the literature, for instance in [8–11].

Technically, Mpdata consists of a sequence of upwind iterations, where the initial iteration is the first-order accurate upwind scheme, and the subsequent iterations use a pseudo-velocity constructed from the leading truncation error of the preceding iteration. Thus the scheme itself refines the first order to a high order according to which terms of the leading truncation error we choose to construct the pseudo-velocities. In comparison to the total variation diminishing schemes [12] and essentially non-oscillatory schemes [13], the general iterative principle of the algorithm seems to be simple and can be easily developed from the Taylor series expansion applied to the classical upstream scheme. These type of schemes are particularly important for an advection problem, but also for an advection–diffusion problem [14–16] where a major objective can be to inhibit or prevent oscillations.

In this paper we study two versions of the Mpdata, the classical Mpdata [3] and the so-called third-order Mpdata [17] when applied to the one-dimensional unsteady advection–diffusion equation. For the classical Mpdata the pseudo-velocities of the iterations are constructed from the second-order leading truncation error and the third-order Mpdata consists of continuing the truncation error to identify the third-order error and then to build the pseudo-velocities with the additional terms. One of the purposes of the paper is to provide indicators of the situations in which we shall use the methods.

The paper is organized as follows. We start in the next section with a very brief review of the classical Mpdata method [3] and the third-order Mpdata [17]. In the third section we describe its application to the advection–diffusion equation. A stability region depending on the Courant number and the mesh Fourier number is displayed. We analyze the order of accuracy of the schemes and conclude that although for pure advection the third-order Mpdata reaches third order, for an advection–diffusion problem for the majority of cases, it only reaches second order. Nevertheless, the amplitude of the error diminishes considerably and it is always smaller than the amplitude of the error of the classical Mpdata scheme. The fact that the third-order Mpdata method is not necessarily of high order is also pointed out in [6] and [17], for variable advective fluid flows. In the end we present some test problems. We comment on the accuracy of the methods that depends on the Peclet number showing that both schemes present a less good performance for large mesh Peclet numbers.

2. THE CLASSICAL SIGN-PRESERVING NUMERICAL METHOD

The classical method was used for the advection equation. Consider the equation

$$\psi_t = -(u\psi)_x \tag{1}$$

Let Ψ_i^n be the numerical approximation of the solution of (1) defined in (x_i, t_n), where $t_n = n\Delta t$, $x_i = \mathrm{i}\Delta x$, and Δx is the space step and Δt is the time step.

The classical first-order upstream scheme can be written in the form

$$\Psi_i^{n+1} = \Psi_i^n - [F(\Psi_i^n, \Psi_{i+1}^n, v_{i+1}) - F(\Psi_{i-1}^n, \Psi_i^n, v_i)] \tag{2}$$

DOI: 10.1002/fld

where F is such that

$$F(\Psi_L, \Psi_C, v) = \tfrac{1}{2}(v+|v|)\Psi_L + \tfrac{1}{2}(v-|v|)\Psi_C \tag{3}$$

and

$$v_i = u_{i-1/2}\Delta t/\Delta x, \quad u_{i-1/2} = u(x_i - \Delta x/2)$$

For u constant, a sufficient and necessary condition for stability is

$$|u|\frac{\Delta t}{\Delta x} \leqslant 1 \tag{4}$$

For u not constant, this is the CFL condition that is a necessary condition for stability.

Let u be constant and $\psi \geqslant 0$. The truncation error for (2) is given by

$$\frac{(\Delta x)^2}{2\Delta t}(|v|-v^2)\psi_{x^2} - \frac{1}{6}\frac{\Delta x^3}{\Delta t}(v-3v|v|+2v^3)\psi_{x^3} + O\left(\frac{\Delta x^4}{\Delta t}\right) + O(\Delta t^3) \tag{5}$$

where $v = u\Delta t/\Delta x$ is the Courant number.

The truncation analysis shows that (2) more accurately approximates the advection–diffusion equation

$$\psi_t = -(u\psi)_x + (K\psi_x)_x \tag{6}$$

where

$$K = \frac{(\Delta x)^2}{2\Delta t}(|v|-v^2)$$

From (6) it can be seen that when Δt and Δx go to zero (6) approaches (1), but during a realistic computational process the scheme in (2) with finite Δx and Δt approximates more accurately an advection transport equation with additional diffusive terms rather than the original equation (1).

On the other hand, these implicit diffusion terms are important for the stability of the scheme and must not be explicitly subtracted from the scheme. An intuitive approach is to make the advection step using Equation (2) and then reverse the effect of the diffusion equation

$$\psi_t = (K\psi_x)_x \tag{7}$$

in the next corrective step.

Let

$$\text{error}^{(1)} = (K\psi_x)_x = \frac{\partial}{\partial x}(v^{(1)}\psi) \tag{8}$$

where

$$v^{(1)} = K\frac{1}{\psi}\frac{\partial \psi}{\partial x} \tag{9}$$

for $\psi > 0$ and $v^{(1)} = 0$ for $\psi = 0$. Therefore, we have the pseudo-velocity $v^{(1)}$ given by

$$v^{(1)} = \frac{(\Delta x)^2}{2\Delta t}(|v|-v^2)\frac{1}{\psi}\frac{\partial \psi}{\partial x} \tag{10}$$

DOI: 10.1002/fld

The superscript (1) shows that it is the first approximation to subtract the error.

Note that we can write (7) as

$$\psi_t = -(-v^{(1)}\psi)_x \tag{11}$$

and therefore the anti-diffusive velocities or pseudo-velocities are $v^{(1)}$.

In the approximation of (11) by (2) the factor in the numerator of $v^{(1)}$ defined in (10) will be represented using an upstream value, whereas the factor in the denominator will be approximated using a centered value. In this way, a nonlinearity is introduced and a higher-order approximation is found that still preserves positivity. Let $\Psi^{(1)}$ be the approximation given by the first iteration, which is determined using the right-hand side of (2). For

$$v^{(1)} = v^{(1)}\frac{\Delta t}{\Delta x} = \frac{1}{2}\Delta x(|v| - v^2)\frac{1}{\psi}\frac{\partial \psi}{\partial x}$$

a first-order accurate estimate is

$$v_i^{(1)} = (|v| - v^2)\frac{\Psi_i^{(1)} - \Psi_{i-1}^{(1)}}{\Psi_i^{(1)} + \Psi_{i-1}^{(1)}} \tag{12}$$

Sometimes, it is necessary to introduce a small value ε, e.g. $\varepsilon = 10^{-15}$ in

$$\psi \simeq \frac{2}{\Psi_i^{(1)} + \Psi_{i-1}^{(1)} + \varepsilon}$$

to ensure $v^{(1)} = 0$ when $\Psi_{i-1}^{(1)} = \Psi_i^{(1)} = 0$.

The corrective step is suggested in the form

$$\Psi_i^{n+1} = \Psi_i^{(1)} - [F(\Psi_i^{(1)}, \Psi_{i+1}^{(1)}, v_{i+1}^{(1)}) - F(\Psi_{i-1}^{(1)}, \Psi_i^{(1)}, v_i^{(1)})] \tag{13}$$

which estimates Ψ^{n+1} to the second order, while preserving the sign of Ψ and $\Psi^{(1)}$.

Mpdata works in the following way:

1. First we compute (2) to obtain $\Psi_i^{(1)}$.
2. Second, we calculate the pseudo-velocity $v_i^{(1)}$ using (12).
3. We determine the final value Ψ_i^{n+1} using (13).

We can iterate $k = 2, \ldots,$ iter times so that

$$\Psi_i^{(k)} = \Psi_i^{(k-1)} - [F(\Psi_i^{(k-1)}, \Psi_{i+1}^{(k-1)}, v_{i+1}^{(k-1)}) - F(\Psi_{i-1}^{(k-1)}, \Psi_i^{(k-1)}, v_i^{(k-1)})] \tag{14}$$

before calculating the final value Ψ_i^{n+1} and where

$$v_i^{(k)} = (|v_i^{(k-1)}| - (v_i^{(k-1)})^2)\frac{\Psi_i^{(k)} - \Psi_{i-1}^{(k)}}{\Psi_i^{(k)} + \Psi_{i-1}^{(k)}} \tag{15}$$

Theoretically, iter can be any value, but as can be concluded from the performed tests, using iter>3 only negligibly improves the accuracy of the solution, while increasing the computational costs of the scheme.

DOI: 10.1002/fld

A third-order Mpdata method, presented in [17], which takes into consideration the third-order terms of the truncation error (5), consists in considering the pseudo-velocities as

$$v^{(1)}=\frac{1}{2}\frac{\Delta x^2}{\Delta t}(|v|-v^2)\frac{\psi_x}{\psi}-\frac{1}{6}\frac{\Delta x^3}{\Delta t}(v-3v|v|+2v^3)\frac{\psi_{x^2}}{\psi}$$

and therefore

$$v^{(1)}=v^{(1)}\frac{\Delta t}{\Delta x}=\frac{1}{2}\Delta x(|v|-v^2)\frac{\psi_x}{\psi}-\frac{1}{6}\Delta x^2(v-3v|v|+2v^3)\frac{\psi_{x^2}}{\psi}$$

In this case we consider the recursive $v_i^{(k)}$ given by

$$v_i^{(k)}=(|v_i^{(k-1)}|-(v_i^{(k-1)})^2)\frac{\Psi_i^{(k)}-\Psi_{i-1}^{(k)}}{\Psi_i^{(k)}+\Psi_{i-1}^{(k)}}-\frac{4}{6}(v_i^{(k-1)}-3(v_i^{(k-1)})^2+2(v_i^{(k-1)})^3)$$

$$\times\frac{\Psi_{i-2}^{(k)}-2\Psi_{i-1}^{(k)}+\Psi_i^{(k)}}{\Psi_{i-2}^{(k)}+2\Psi_{i-1}^{(k)}+\Psi_i^{(k)}}$$

The factor in the denominator of the second term of $v_i^{(k)}$ will be approximated by

$$\frac{4}{\Psi_{i-2}^{(k)}+2\Psi_{i-1}^{(k)}+\Psi_i^{(k)}}$$

in order to preserve the stability of the scheme, that is, to verify the CFL condition $|v_i^{(k)}|\leqslant 1$.

3. SIMULATION OF THE ADVECTION–DIFFUSION EQUATION

3.1. The numerical method

Consider the equation

$$\frac{\partial\psi}{\partial t}+\frac{\partial}{\partial x}(u\psi)=\frac{\partial^2}{\partial x^2}(D\psi) \tag{16}$$

where $D>0$ is the diffusion coefficient.

We can write

$$\frac{\partial\psi}{\partial t}+\frac{\partial}{\partial x}(u\psi-(D\psi)_x)=0 \tag{17}$$

or

$$\frac{\partial\psi}{\partial t}+\frac{\partial}{\partial x}(v\psi)=0 \tag{18}$$

DOI: 10.1002/fld

for $v=u+\omega$ where

$$\omega=\begin{cases}-\dfrac{(D\psi)_x}{\psi}, & \psi\neq 0\\ 0, & \psi=0\end{cases}$$

Therefore, we can simulate this equation using the advection equation (1). The first-order upstream scheme can be written as

$$\Psi_i^{n+1}=\Psi_i^n-[F(\Psi_i^n,\Psi_{i+1}^n,\alpha_{i+1})-F(\Psi_{i-1}^n,\Psi_i^n,\alpha_i)] \tag{19}$$

where F is defined in (3) and $\alpha_i=v_i\Delta t/\Delta x$ for $v_i=u_{i-1/2}+\omega_{i-1/2}$ with

$$u_{i-1/2}=u(x_i-\Delta x/2),\quad \omega_{i-1/2}=-D\frac{\Psi_i-\Psi_{i-1}}{\Delta x}\frac{2}{\Psi_i+\Psi_{i-1}}$$

The discretization of $\alpha=v\Delta t/\Delta x=(u+\omega)\Delta t/\Delta x$ is given by

$$\alpha_i=v_i-2\mu\xi_i \tag{20}$$

with

$$v_i=u_{i-1/2}\frac{\Delta t}{\Delta x},\quad \xi_i=\frac{\Psi_i-\Psi_{i-1}}{\Psi_i+\Psi_{i-1}},\quad \mu=D\frac{\Delta t}{\Delta x^2}$$

where μ is called the mesh Fourier number. The scheme is nonlinear even in the case of a uniform velocity field.

The implementation of the classical Mpdata method consists of first computing

$$\Psi_i^{(1)}=\Psi_i^n-[F(\Psi_i^n,\Psi_{i+1}^n,\alpha_{i+1})-F(\Psi_{i-1}^n,\Psi_i^n,\alpha_i)] \tag{21}$$

The pseudo-velocities are given by

$$v^{(1)}=\frac{1}{2}\frac{\Delta x^2}{\Delta t}(|\alpha|-(\alpha)^2)\frac{\psi_x}{\psi}$$

and the discretized $\alpha_i^{(1)}$ are computed by

$$\alpha_i^{(1)}=v^{(1)}\frac{\Delta t}{\Delta x}=(|\alpha_i|-(\alpha_i)^2)\frac{\Psi_i^{(1)}-\Psi_{i-1}^{(1)}}{\Psi_i^{(1)}+\Psi_{i-1}^{(1)}}$$

We can then iterate k times, $k=2,\ldots,$ iter

$$\Psi_i^{(k)}=\Psi_i^{(k-1)}-[F(\Psi_i^{(k-1)},\Psi_{i+1}^{(k-1)},\alpha_{i+1}^{(k-1)})-F(\Psi_{i-1}^{(k-1)},\Psi_i^{(k-1)},\alpha_i^{(k-1)})] \tag{22}$$

before computing the final value Ψ_i^{n+1} and where

$$\alpha_i^{(k)}=(|\alpha_i^{(k-1)}|-(\alpha_i^{(k-1)})^2)\frac{\Psi_i^{(k)}-\Psi_{i-1}^{(k)}}{\Psi_i^{(k)}+\Psi_{i-1}^{(k)}} \tag{23}$$

DOI: 10.1002/fld

Finally, we get

$$\Psi_i^{n+1} = \Psi_i^{(k)} - [F(\Psi_i^{(k)}, \Psi_{i+1}^{(k)}, \alpha_{i+1}^{(k)}) - F(\Psi_{i-1}^{(k)}, \Psi_i^{(k)}, \alpha_i^{(k)})] \tag{24}$$

The third-order Mpdata method consists in considering the pseudo-velocities as

$$v^{(1)} = \frac{1}{2}\frac{\Delta x^2}{\Delta t}(|\alpha| - \alpha^2)\frac{\psi_x}{\psi} - \frac{1}{6}\frac{\Delta x^3}{\Delta t}(\alpha - 3\alpha|\alpha| + 2\alpha^3)\frac{\psi_{x^2}}{\psi}$$

and

$$\alpha^{(1)} = v^{(1)}\frac{\Delta t}{\Delta x} = \frac{1}{2}\Delta x(|\alpha| - \alpha^2)\frac{\psi_x}{\psi} - \frac{1}{6}\Delta x^2(\alpha - 3\alpha|\alpha| + 2\alpha^3)\frac{\psi_{x^2}}{\psi}$$

In this case, we consider the recursive $\alpha_i^{(k)}$ given by

$$\begin{aligned}\alpha_i^{(k)} = (|\alpha_i^{(k-1)}| - (\alpha_i^{(k-1)})^2)\frac{\Psi_i^{(k)} - \Psi_{i-1}^{(k)}}{\Psi_i^{(k)} + \Psi_{i-1}^{(k)}} \\ -\frac{4}{6}(\alpha_i^{(k-1)} - 3(\alpha_i^{(k-1)})^2 + 2(\alpha_i^{(k-1)})^3)\frac{\Psi_{i-2}^{(k)} - 2\Psi_{i-1}^{(k)} + \Psi_i^{(k)}}{\Psi_{i-2}^{(k)} + 2\Psi_{i-1}^{(k)} + \Psi_i^{(k)}}\end{aligned} \tag{25}$$

The matrical form of the schemes can be described as follows. First we calculate

$$\Psi^{(1)} = A\Psi^n$$

where A is the iteration matrix coming out of the scheme (2) and Ψ^n and $\Psi^{(1)}$ the vectors of discretized values given by $\Psi^n = [\Psi_0^n, \ldots, \Psi_{N-1}^n]^{\mathrm{T}}$ and $\Psi^{(1)} = [\Psi_0^{(1)}, \ldots, \Psi_{N-1}^{(1)}]^{\mathrm{T}}$, respectively.

Second, we iterate k times

$$\begin{aligned}\Psi^{(2)} &= A^{(1)}\Psi^{(1)} \\ \vdots &\quad \vdots \\ \Psi^{(k)} &= A^{(k-1)}\Psi^{(k-1)}\end{aligned}$$

where $A^{(k)}$ denotes the matrix iteration at k-iteration and $\Psi^{(k)} = [\Psi_0^{(k)}, \ldots, \Psi_{N-1}^{(k)}]^{\mathrm{T}}$. Therefore,

$$\Psi^{n+1} = A^{(k)}A^{(k-1)} \ \ldots \ A^{(1)}A\Psi^n$$

The matrices $A^{(k)}$ are given by

$$A^{(k)} = \begin{bmatrix} d_{00}^{(k)} & d_{10}^{(k)} & & d_{-10}^{(k)} \\ d_{-11}^{(k)} & d_{01}^{(k)} & d_{11}^{(k)} & \\ & & \ddots & \\ d_{1N-1}^{(k)} & \cdots & d_{-1N-1}^{(k)} & d_{0N-1}^{(k)} \end{bmatrix}$$

DOI: 10.1002/fld

where

$$d_{0p}^{(k)} = 1 - \tfrac{1}{2}(\alpha_{p+1}^{(k)} + |\alpha_{p+1}^{(k)}|) + \tfrac{1}{2}(\alpha_p^{(k)} - |\alpha_p^{(k)}|)$$

$$d_{-1p}^{(k)} = \tfrac{1}{2}(\alpha_p^{(k)} + |\alpha_p^{(k)}|)$$

$$d_{1p}^{(k)} = -\tfrac{1}{2}(\alpha_{p+1}^{(k)} - |\alpha_{p+1}^{(k)}|), \quad p = 0, \ldots, N-1$$

When $\alpha_i^{(k)} \to 0$ then $A^{(k)} \to I$ and to iterate is not relevant anymore. It is seen experimentally that after three iterations $\alpha_i^{(k)}$ becomes very small. Also, when Δx is very small the iterative process is less effective for more iterations.

The proposed scheme can also be combined with high-order schemes in time and maintain sign-preserving properties by following developments similar to the ones presented, for instance, in [7, 18–21].

3.2. Convergence and stability

3.2.1. Convergence. For $k=1$ the scheme is consistent. For $k>1$ the corrective iterations in Equation (22) do not affect the solution of the first iteration since from (24) it is easy to verify that when Δt, $\Delta x \to 0$ then $\alpha_i^{(k)} \to 0$ for all k. The latter implies that (22) for $k>1$ converges to $\partial\psi/\partial t = 0$, which means the scheme is consistent.

The algorithm order of accuracy can be determined by estimating the truncation error. It can be shown and confirmed by the results of tests that the schemes are at least second-order accurate as we shall discuss in this section. We start by analyzing the classical Mpdata scheme.

Assuming v is computed exactly, $v = u + \omega$, let $\psi^{(1)}$ be the approximation given by the first iteration of the scheme. It can be written in the form

$$\psi^{(1)} = \psi^n - \int_{t_n}^{t_{n+\Delta t}} \frac{\partial}{\partial x}(v\psi)\,\mathrm{d}t + \frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_{n+\Delta t}} \frac{\partial}{\partial x}(K_\alpha \psi_x)\,\mathrm{d}t + O\left(\frac{\Delta x^3}{\Delta t}\right)\Delta t \tag{26}$$

where $K_\alpha = (|\alpha| - \alpha^2)/2$ and $\alpha = v\Delta t/\Delta x$. The second iteration of the scheme is given by

$$\psi^{(2)} = \psi^{(1)} - \int_{t_n}^{t_{n+\Delta t}} \frac{\partial}{\partial x}(v^{(1)}\psi^{(1)})\,\mathrm{d}t + \frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_{n+\Delta t}} \frac{\partial}{\partial x}(K_\alpha^{(1)}\psi_x^{(1)})\,\mathrm{d}t + O\left(\frac{\Delta x^3}{\Delta t}\right)\Delta t \tag{27}$$

for

$$v^{(1)} = \frac{\Delta x^2}{\Delta t} K_\alpha \frac{\psi_x^{(1)}}{\psi^{(1)}} + O\left(\frac{\Delta x^3}{\Delta t}\right), \quad \alpha^{(1)} = v^{(1)}\frac{\Delta t}{\Delta x}, \quad K_\alpha^{(1)} = \frac{1}{2}(|\alpha^{(1)}| - (\alpha^{(1)})^2)$$

We have

$$\psi^{(2)} = \psi^{(1)} - \frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_{n+\Delta t}} \frac{\partial}{\partial x}[(K_\alpha - K_\alpha^{(1)})\psi_x^{(1)}]\,\mathrm{d}t + O\left(\frac{\Delta x^3}{\Delta t}\right)\Delta t \tag{28}$$

Similarly, the k-iteration is given by

$$\psi^{(k)} = \psi^{(k-1)} - \frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_{n+\Delta t}} \frac{\partial}{\partial x}[(K_\alpha^{(k-2)} - K_\alpha^{(k-1)})\psi_x^{(k-1)}]\,\mathrm{d}t + O\left(\frac{\Delta x^3}{\Delta t}\right)\Delta t \tag{29}$$

DOI: 10.1002/fld

where

$$K_\alpha^{(k)}=\frac{1}{2}(|\alpha^{(k)}|-(\alpha^{(k)})^2),\quad \alpha^{(k)}=v^{(k)}\frac{\Delta t}{\Delta x},\quad v^{(k)}=\frac{\Delta x^2}{\Delta t}K_\alpha^{(k-1)}\frac{\psi_x^{(k)}}{\psi^{(k)}}+O\left(\frac{\Delta x^4}{\Delta t}\right)$$

We obtain the integral of order $(\Delta x^3/\Delta t)\Delta t$

$$\psi^{n+1}=\psi^{(k)}-\frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}[(K_\alpha^{(k-1)}-K_\alpha^{(k)})\psi_x^{(k)}]\,\mathrm{d}t+O\left(\frac{\Delta x^3}{\Delta t}\right)\Delta t \tag{30}$$

We notice that the corrective step compensates the small diffusive term from the previous corrective step, that is, the idea behind each iteration is to cancel the term

$$\frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_{n+\Delta t}}K_\alpha^{(k-1)}\psi_x^{(k)}\,\mathrm{d}t$$

with the new term at the k-iteration

$$\frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_{n+\Delta t}}K_\alpha^{(k)}\psi_x^{(k)}\,\mathrm{d}t$$

We also observe that increasing the number of iterations does not necessarily increases the order of accuracy of the scheme even if it diminishes considerably the magnitude of the error. In the end of all iterations, we have

$$\begin{aligned}\psi^{n+1}=\psi^n&-\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}(v\psi)\,\mathrm{d}t+\frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}K_\alpha\frac{\partial}{\partial x}\left[\int_{t_n}^{t_{n+\Delta t}}(v\psi)\,\mathrm{d}t\right]\mathrm{d}\tau\\&+\frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}(K_\alpha^{(1)}\psi_x^{(1)})\,\mathrm{d}t-\frac{\Delta x^2}{\Delta t}\sum_{p=2}^{k}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}[(K_\alpha^{(p-1)}-K_\alpha^{(p)})\psi_x^{(p)}]\,\mathrm{d}t\\&+O\left(\frac{\Delta x^3}{\Delta t}\right)\Delta t\end{aligned}$$

and since the second-term integral is of order $(\Delta x^2/\Delta t)\Delta t^2$ then

$$\begin{aligned}\psi^{n+1}=\psi^n&-\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}(v\psi)\,\mathrm{d}t+\frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}(K_\alpha^{(1)}\psi_x^{(1)})\,\mathrm{d}t\\&-\frac{\Delta x^2}{\Delta t}\sum_{p=2}^{k}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}[(K_\alpha^{(p-1)}-K_\alpha^{(p)})\psi_x^{(p)}]\,\mathrm{d}t+O\left(\frac{\Delta x^3}{\Delta t}\right)\Delta t+O(\Delta x^2\Delta t)\Delta t\end{aligned}$$

If $\Delta t=c\Delta x$ we expect the method to be of second order since we have

$$\psi^{n+1}=\psi^n-\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}(v\psi)\,\mathrm{d}t+O(\Delta x^2)\Delta t \tag{31}$$

Therefore, we expect the method to start with first order and then with the iterations we expect to reach at least second order.

DOI: 10.1002/fld

For the third order Mpdata, we can do a similar study. Let $\psi^{(1)}$ be the approximation given by the first iteration of the scheme. It can be written in the form

$$\psi^{(1)} = \psi^n - \int_{t_n}^{t_{n+\Delta t}} \frac{\partial}{\partial x}(v\psi)\,\mathrm{d}t + \frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_{n+\Delta t}} \frac{\partial}{\partial x}(K_\alpha \psi_x)\,\mathrm{d}t$$

$$-\frac{\Delta x^3}{\Delta t}\int_{t_n}^{t_{n+\Delta t}} \frac{\partial}{\partial x}(B_\alpha \psi_{x^2})\,\mathrm{d}t + O\left(\frac{\Delta x^4}{\Delta t}\right)\Delta t \tag{32}$$

where $K_\alpha = (|\alpha| - \alpha^2)/2$, $B_\alpha = (\alpha - 3\alpha^2 + 2\alpha^3)/6$, and $\alpha = v\Delta t/\Delta x$. The second iteration of the scheme is given by

$$\psi^{(2)} = \psi^{(1)} - \int_{t_n}^{t_{n+\Delta t}} \frac{\partial}{\partial x}(v^{(1)}\psi^{(1)})\,\mathrm{d}t + \frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_{n+\Delta t}} \frac{\partial}{\partial x}(K_\alpha^{(1)}\psi_x^{(1)})\,\mathrm{d}t$$

$$-\frac{\Delta x^3}{\Delta t}\int_{t_n}^{t_{n+\Delta t}} \frac{\partial}{\partial x}(B_\alpha^{(1)}\psi_{x^2}^{(1)})\,\mathrm{d}t + O\left(\frac{\Delta x^4}{\Delta t}\right)\Delta t \tag{33}$$

for

$$v^{(1)} = \frac{\Delta x^2}{\Delta t}K_\alpha\frac{\psi_x^{(1)}}{\psi^{(1)}} - \frac{\Delta x^3}{\Delta t}B_\alpha\frac{\psi_{x^2}^{(1)}}{\psi^{(1)}} + O\left(\frac{\Delta x^3}{\Delta t}\right), \quad \alpha^{(1)} = v^{(1)}\frac{\Delta t}{\Delta x}$$

$$K_\alpha^{(1)} = \tfrac{1}{2}(|\alpha^{(1)}| - (\alpha^{(1)})^2), \quad B_\alpha^{(1)} = \tfrac{1}{6}(\alpha^{(1)} - 3(\alpha^{(1)})^2 + 2(\alpha^{(1)})^3)$$

We get

$$\psi^{(2)} = \psi^{(1)} - \frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_{n+\Delta t}} \frac{\partial}{\partial x}[(K_\alpha - K_\alpha^{(1)})\psi_x^{(1)}]\,\mathrm{d}t$$

$$+\frac{\Delta x^3}{\Delta t}\int_{t_n}^{t_{n+\Delta t}} \frac{\partial}{\partial x}((B_\alpha - B_\alpha^{(1)})\psi_{x^2}^{(1)})\,\mathrm{d}t + O\left(\frac{\Delta x^4}{\Delta t}\right)\Delta t \tag{34}$$

Therefore, the k-iteration is given by

$$\psi^{(k)} = \psi^{(k-1)} - \frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_{n+\Delta t}} \frac{\partial}{\partial x}[(K_\alpha^{(k-2)} - K_\alpha^{(k-1)})\psi_x^{(k-1)}]\,\mathrm{d}t$$

$$+\frac{\Delta x^3}{\Delta t}\int_{t_n}^{t_{n+\Delta t}} \frac{\partial}{\partial x}((B_\alpha^{(k-2)} - B_\alpha^{(k-1)})\psi_{x^2}^{(k-1)})\,\mathrm{d}t + O\left(\frac{\Delta x^4}{\Delta t}\right)\Delta t \tag{35}$$

where

$$K_\alpha^{(k)} = \frac{1}{2}(|\alpha^{(k)}| - (\alpha^{(k)})^2), \quad \alpha^{(k)} = v^{(k)}\frac{\Delta t}{\Delta x}$$

$$v^{(k)} = \frac{\Delta x^2}{\Delta t}K_\alpha^{(k-1)}\frac{\psi_x^{(k)}}{\psi^{(k)}} - \frac{\Delta x^3}{\Delta t}B_\alpha^{(k-1)}\frac{\psi_{x^2}^{(k)}}{\psi^{(k)}} + O\left(\frac{\Delta x^4}{\Delta t}\right)$$

DOI: 10.1002/fld

We can write

$$\psi^{n+1} = \psi^{(k)} - \frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_{n+\Delta t}} \frac{\partial}{\partial x}[(K_\alpha^{(k-1)} - K_\alpha^{(k)})\psi_x^{(k)}]\,\mathrm{d}t$$

$$+\frac{\Delta x^3}{\Delta t}\int_{t_n}^{t_{n+\Delta t}} \frac{\partial}{\partial x}[(B_\alpha^{(k-1)} - B_\alpha^{(k)})\psi_{x^2}^{(k)}]\,\mathrm{d}t + O\left(\frac{\Delta x^4}{\Delta t}\right)\Delta t \tag{36}$$

Similar to the classical Mpdata, the corrective step for the third-order Mpdata compensates the small terms from the previous corrective step, that is, each iteration k pretends to cancel the terms

$$\frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_{n+\Delta t}} K_\alpha^{(k-1)}\psi_x^{(k)}\,\mathrm{d}t, \quad \frac{\Delta x^3}{\Delta t}\int_{t_n}^{t_{n+\Delta t}} B_\alpha^{(k-1)}\psi_{x^2}^{(k)}\,\mathrm{d}t$$

with the new terms at the k-iteration, respectively

$$\frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_{n+\Delta t}} K_\alpha^{(k)}\psi_x^{(k)}\,\mathrm{d}t, \quad \frac{\Delta x^3}{\Delta t}\int_{t_n}^{t_{n+\Delta t}} B_\alpha^{(k)}\psi_{x^2}^{(k)}\,\mathrm{d}t$$

Taking into consideration the previous equalities, we have

$$\psi^{n+1} = \psi^{(1)} - \frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_{n+\Delta t}} \frac{\partial}{\partial x}(K_\alpha\psi_x^{(1)})\,\mathrm{d}t + \frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_{n+\Delta t}} \frac{\partial}{\partial x}(K_\alpha^{(1)}\psi_x^{(1)})\,\mathrm{d}t$$

$$+\frac{\Delta x^3}{\Delta t}\int_{t_n}^{t_{n+\Delta t}} \frac{\partial}{\partial x}(B_\alpha\psi_{x^2}^{(1)})\,\mathrm{d}t - \frac{\Delta x^3}{\Delta t}\int_{t_n}^{t_{n+\Delta t}} \frac{\partial}{\partial x}(B_\alpha^{(1)}\psi_{x^2}^{(1)})\,\mathrm{d}t$$

$$-\frac{\Delta x^2}{\Delta t}\sum_{p=2}^{k}\int_{t_n}^{t_{n+\Delta t}} \frac{\partial}{\partial x}[(K_\alpha^{(p-1)} - K_\alpha^{(p)})\psi_x^{(p)}]\,\mathrm{d}t$$

$$+\frac{\Delta x^3}{\Delta t}\sum_{p=2}^{k}\int_{t_n}^{t_{n+\Delta t}} \frac{\partial}{\partial x}[(B_\alpha^{(p-1)} - B_\alpha^{(p)})\psi_{x^2}^{(p)}]\,\mathrm{d}t + O\left(\frac{\Delta x^4}{\Delta t}\right)\Delta t$$

and

$$\psi^{n+1} = \psi^n - \int_{t_n}^{t_{n+\Delta t}} (v\psi)_x\,\mathrm{d}t + \frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_{n+\Delta t}} \frac{\partial}{\partial x}(K_\alpha\psi_x)\,\mathrm{d}t - \frac{\Delta x^3}{\Delta t}\int_{t_n}^{t_{n+\Delta t}} \frac{\partial}{\partial x}(B_\alpha\psi_{x^2})\,\mathrm{d}t$$

$$-\frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_{n+\Delta t}} \frac{\partial}{\partial x}\left\{K_\alpha\left[\psi_x - \int_{t_n}^{t_{n+\Delta t}} \frac{\partial}{\partial x}(v\psi_x)\,\mathrm{d}t + \frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_{n+\Delta t}} \frac{\partial}{\partial x^2}(K_\alpha\psi_x)\,\mathrm{d}t\right.\right.$$

$$\left.\left.-\frac{\Delta x^3}{\Delta t}\int_{t_n}^{t_{n+\Delta t}} \frac{\partial}{\partial x^2}(B_\alpha\psi_{x^2})\,\mathrm{d}t\right]\right\}\mathrm{d}\tau$$

$$+\frac{\Delta x^3}{\Delta t}\int_{t_n}^{t_{n+\Delta t}} \frac{\partial}{\partial x}\left\{B_\alpha\left[\psi_{x^2} - \int_{t_n}^{t_{n+\Delta t}} \frac{\partial}{\partial x^2}(v\psi_x)\,\mathrm{d}t\right.\right.$$

DOI: 10.1002/fld

$$+\frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x^3}(K_\alpha\psi_x)\,\mathrm{d}t-\frac{\Delta x^3}{\Delta t}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x^3}(B_\alpha\psi_{x^2})\,\mathrm{d}t\Bigg]\Bigg\}\,\mathrm{d}\tau$$

$$+\frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}(K_\alpha^{(1)}\psi_x^{(1)})\,\mathrm{d}t-\frac{\Delta x^3}{\Delta t}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}(B_\alpha^{(1)}\psi_{x^2}^{(1)})\,\mathrm{d}t$$

$$-\frac{\Delta x^2}{\Delta t}\sum_{p=2}^{k}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}[(K_\alpha^{(p-1)}-K_\alpha^{(p)})\psi_x^{(p)}]\,\mathrm{d}t$$

$$+\frac{\Delta x^3}{\Delta t}\sum_{p=2}^{k}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}[(B_\alpha^{(p-1)}-B_\alpha^{(p)})\psi_{x^2}^{(p)}]\,\mathrm{d}t+O\left(\frac{\Delta x^4}{\Delta t}\right)\Delta t$$

Finally

$$\psi^{n+1}=\psi^n-\int_{t_n}^{t_{n+\Delta t}}(v\psi)_x\,\mathrm{d}t+\frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}\left[K_\alpha\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}(v\psi_x)\,\mathrm{d}t\right]\mathrm{d}\tau$$

$$-\frac{\Delta x^4}{\Delta t^2}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}K_\alpha\left[\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x^2}(K_\alpha\psi_x)\,\mathrm{d}t\right]\mathrm{d}\tau$$

$$-\frac{\Delta x^3}{\Delta t}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}B_\alpha\left[\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x^2}(v\psi_x)\,\mathrm{d}t\right]\mathrm{d}\tau+\frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}(K_\alpha^{(1)}\psi_x^{(1)})\,\mathrm{d}t$$

$$-\frac{\Delta x^3}{\Delta t}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}(B_\alpha^{(1)}\psi_{x^2}^{(1)})\,\mathrm{d}t-\frac{\Delta x^2}{\Delta t}\sum_{p=2}^{k}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}[(K_\alpha^{(p-1)}-K_\alpha^{(p)})\psi_x^{(p)}]\,\mathrm{d}t$$

$$+\frac{\Delta x^3}{\Delta t}\sum_{p=2}^{k}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}[(B_\alpha^{(p-1)}-B_\alpha^{(p)})\psi_x^{(p)}]\,\mathrm{d}t+O\left(\frac{\Delta x^4}{\Delta t}\right)\Delta t$$

For this method to have a superior order to the classical Mpdata method the following terms that seem to be of second order

$$\frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}K_\alpha\left[\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}(v\psi_x)\,\mathrm{d}t\right]\mathrm{d}\tau+\frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}(K_\alpha^{(1)}\psi_x^{(1)})\,\mathrm{d}t$$

$$-\frac{\Delta x^2}{\Delta t}\sum_{p=2}^{k}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}[(K_\alpha^{(p-1)}-K_\alpha^{(p)})\psi_x^{(p)}]\,\mathrm{d}t$$

should reach third order. If we sum these terms we have

$$\frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}P_\alpha\,\mathrm{d}t \tag{37}$$

DOI: 10.1002/fld

for

$$P_\alpha = K_\alpha \int_{t_n}^{t_{n+\Delta t}} \frac{\partial}{\partial x}(v\psi)_x \, \mathrm{d}t + K_\alpha^{(1)} \psi_x^{(1)} - \sum_{p=2}^{k} (K_\alpha^{(p-1)} - K_\alpha^{(p)}) \psi_x^{(p)} \tag{38}$$

and for (37), to reach third order we must have $P_\alpha = O(\Delta x^2) + O(\Delta t \Delta x)$. Note that

$$K_\alpha^{(k)} = \tfrac{1}{2}(|\alpha^{(k)}| - (\alpha^{(k)})^2)$$

for

$$\alpha^{(k)} = \frac{1}{2}\Delta x K_\alpha^{(k-1)} \frac{\psi_x^{(k)}}{\psi^{(k)}} - \Delta x^2 B_\alpha^{(k-1)} \frac{\psi_{x^2}^{(k)}}{\psi^{(k)}} + O(\Delta x^3)$$

and we have that $K_\alpha^{(k)} = O(\Delta x^2)$, for $k \geqslant 2$. Therefore, computing the derivative of (32) and substituting in (38) we have

$$P_\alpha = K_\alpha \int_{t_n}^{t_{n+\Delta t}} \frac{\partial}{\partial x}(v\psi)_x \, \mathrm{d}t + K_\alpha^{(1)} \psi_x - K_\alpha^{(1)} \int_{t_n}^{t_{n+\Delta t}} \frac{\partial}{\partial x}(v\psi)_x \, \mathrm{d}t - K_\alpha^{(1)} \psi_x^{(2)} + O(\Delta x^2)$$

We can write

$$P_\alpha = K_\alpha \int_{t_n}^{t_{n+\Delta t}} \frac{\partial}{\partial x}(v\psi)_x \, \mathrm{d}t + K_\alpha^{(1)} \psi_x - K_\alpha^{(1)} \psi_x^{(2)} + O(\Delta x^2) + O(\Delta x \Delta t)$$

From this results we can say that the third-order Mpdata does not necessarily increases the order of accuracy to third order but diminishes the magnitude of the error.

The analysis of this section indicates that the classical Mpdata is of second order and that the third-order Mpdata is not necessarily of third order. For both schemes and for $k \geqslant 3$, as we shall see in the next section, the error does not diminish considerably between iterations. This happens because $K_\alpha^{(k)}$ and $B_\alpha^{(k)}$ become significantly smaller.

3.2.2. Stability. The stability of the first step controls the stability of the subsequent iterations for both schemes, the classical Mpdata scheme and the third-order Mpdata scheme.

Let us assume that we have a uniform velocity field u. Then

$$\alpha = \left(u - D\frac{\psi_x}{\psi}\right)\frac{\Delta t}{\Delta x}$$

For

$$v = \frac{u\Delta t}{\Delta x}, \quad \mu = \frac{D\Delta t}{\Delta x^2}$$

the discretized values α_i are given by (20), that is

$$\alpha_i = v - 2\mu\xi_i$$

where ξ_i's are local approximations of ψ_x/ψ. The CFL stability condition is given by

$$|\alpha_i| = |v - 2\mu\xi_i| \leqslant 1$$

DOI: 10.1002/fld

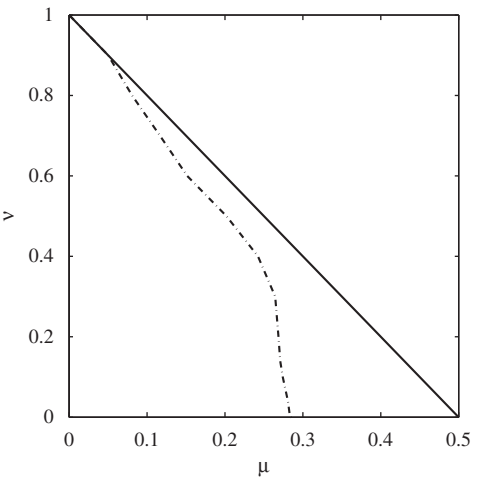

Figure 1. Stability region: analytical necessary condition (–); experimental condition (–·–).

Assuming $\psi \geqslant 0$ we have $|\xi_i| \leqslant 1$. Therefore, a stability condition is given by

$$v + 2\mu \leqslant 1$$

since

$$|v - 2\mu\xi_i| \leqslant |v| + 2\mu|\xi_i| \leqslant v + 2\mu$$

and if $v + 2\mu \leqslant 1$ we have $|v - 2\mu\xi_i| \leqslant 1$.

We plot this region and an experimental stability region in Figure 1, where we display the Courant number, v, versus the Fourier number, μ. The experimental stability region was computed by running calculations for different values of v and μ.

From (23) we have $\alpha_i^{(k)}$ given by

$$\alpha_i^{(k)} = (\alpha_i^{(k-1)} - (\alpha_i^{(k-1)})^2) \frac{\Psi_i^{(k)} - \Psi_{i-1}^{(k)}}{\Psi_i^{(k)} + \Psi_{i-1}^{(k)}}$$

Therefore, $|\alpha_i^{(k)}| \leqslant |\alpha_i^{(k-1)}|$ since

$$|\alpha_i^{(k)}| = |\alpha_i^{(k-1)}||(1 - \alpha_i^{(k-1)})| \left| \frac{\Psi_i^{(k)} - \Psi_{i-1}^{(k)}}{\Psi_i^{(k)} + \Psi_{i-1}^{(k)}} \right| \leqslant |\alpha_i^{(k-1)}|$$

and the stability of the first step controls the stability of the full algorithm with various iterations.

For the third-order Mpdata from (25), the $\alpha_i^{(k)}$ are given by

$$\alpha_i^{(k)} = |\alpha_i^{(k-1)}|(1 - \alpha_i^{(k-1)})\xi_i^{(k)} + \tfrac{4}{6}\alpha_i^{(k-1)}(1 - \alpha_i^{(k-1)})(2\alpha_i^{(k-1)} - 1)\eta_i^{(k)}$$

DOI: 10.1002/fld

where

$$\xi_i^{(k)} = \frac{\Psi_i^{(k)} - \Psi_{i-1}^{(k)}}{\Psi_i^{(k)} + \Psi_{i-1}^{(k)}} \quad \text{and} \quad \eta_i^{(k)} = \frac{\Psi_{i-2}^{(k)} - 2\Psi_{i-1}^{(k)} + \Psi_i^{(k)}}{\Psi_{i-2}^{(k)} + 2\Psi_{i-1}^{(k)} + \Psi_i^{(k)}}$$

If $0 \leqslant \alpha_i^{(k-1)} \leqslant 1$ then

$$\alpha_i^{(k)} = \alpha_i^{(k-1)}(1 - \alpha_i^{(k-1)})[\xi_i^{(k)} + \tfrac{4}{6}(2\alpha_i^{(k-1)} - 1)\eta_i^{(k)}]$$

and $|\alpha_i^{(k)}| \leqslant |\alpha_i^{(k-1)}|$ if $|\xi_i^{(k)}| + (\frac{2}{3})|\eta_i^{(k)}| \leqslant 1$.

If $-1 \leqslant \alpha_i^{(k-1)} \leqslant 0$ then

$$\alpha_i^{(k)} = \alpha_i^{(k-1)}(\alpha_i^{(k-1)} - 1)[\xi_i^{(k)} + \tfrac{4}{6}(2\alpha_i^{(k-1)} - 1)\eta_i^{(k)}]$$

and $|\alpha_i^{(k)}| \leqslant |\alpha_i^{(k-1)}|$ if $|\xi_i^{(k)}| + 2|\eta_i^{(k)}| \leqslant \frac{1}{2}$. The conditions imposed to $\xi_i^{(k)}$ and $\eta_i^{(k)}$ are in general satisfied.

4. TEST PROBLEMS

In this section we consider some numerical experiments in order to further compare the schemes discussed above and its different iterations. We first start to solve the one-dimensional linear advection equation with velocity u and periodic boundary conditions on the domain $0 \leqslant x \leqslant 1$ and second we solve the same problem for the advection–diffusion equation.

4.1. Test problem for $D = 0$

Consider the advection problem defined on the domain $0 \leqslant x \leqslant 1$ with initial condition

$$\psi(x, 0) = \mathrm{e}^{-(x-0.5)^2/L^2}$$

The exact solution is of the form

$$\psi(x, t) = \mathrm{e}^{-(x-0.5-ut)^2/L^2}$$

For our tests we assume $L = 0.02$ and in Figure 2 we display the initial solution and the exact solution at $t = 0.3$.

Consider the vectors $\psi^x = (\psi(x, t_1), \ldots, \psi(x, t_n))$ where ψ is the exact solution and $\Psi^x = (\Psi(x, t_1), \ldots, \Psi(x, t_n))$, where Ψ is the approximated solution. The error is defined by

$$\mathrm{Error}(x) = \|\psi^x - \Psi^x\|_\infty$$

where $\|\cdot\|_\infty$ is the l_∞ norm.

In Figures 3 and 4 we plot the error versus the mesh size, Δx, for $u = 0.5$ and $u = 0.75$, respectively. We observe that until two iterations we have considerable improvements in what concerns the accuracy of the method. For $k = 3$ we have improvements only in some situations. One of those situations is illustrated in Figure 4(b). Further onwards the improvements become meaningless. Therefore, for both methods it is not worth to go beyond three iterations, that is, $k = 3$.

DOI: 10.1002/fld

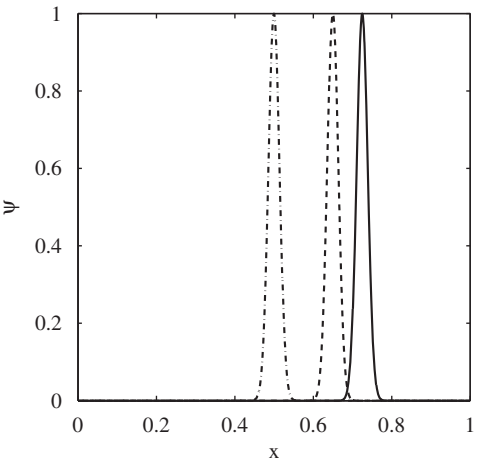

Figure 2. Initial solution (–·–). Exact solution at $t=0.3$ for $u=0.5$ (– –) and for $u=0.75$ (–).

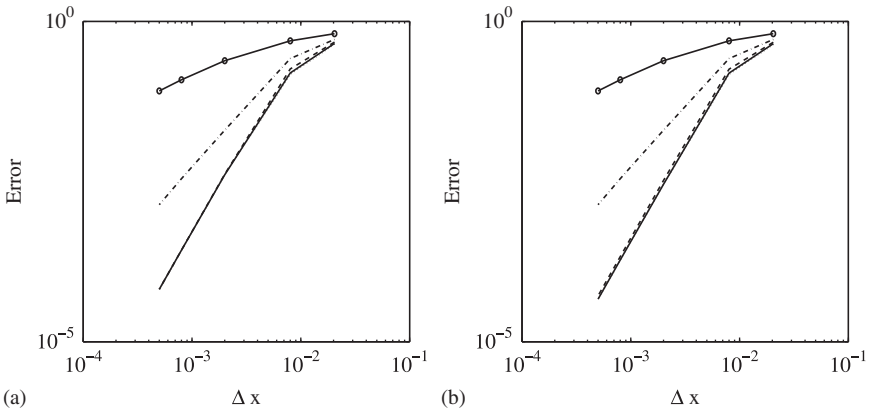

Figure 3. Error function as mesh is refined for: $k=0$, (-o-); $k=1$, (– · –); $k=2$ (– –); $k=3$ (–); $k=4$ (···). $u=0.5$: (a) classical Mpdata and (b) third-order Mpdata.

To further illustrate this point, in Figure 5 we plot the error versus the k-iterations for a fixed space step, namely $\Delta x=0.002$. It is clear that for $k>3$ there are no significant changes for the error values.

In Tables I–IV we show the order of accuracy of the classical Mpdata and the third-order Mpdata for $\Delta t=\Delta x$ and $u=0.1$, $u=0.25$, $u=0.5$, and $u=0.75$. We consider five iterations. We observe that the classical Mpdata is usually of second-order and the third-order Mpdata is of third order for $k\geqslant 2$.

Theoretically, as we have discussed in the previous section the third-order Mpdata can be only proved to be of second order. The fact that it reaches third order can be explained by assuming that the amplitude of the error diminishes in a way that the effect is a third-order convergence. It was also proved, in Section 3, that to increase the number of iterations does not necessarily increases

DOI: 10.1002/fld

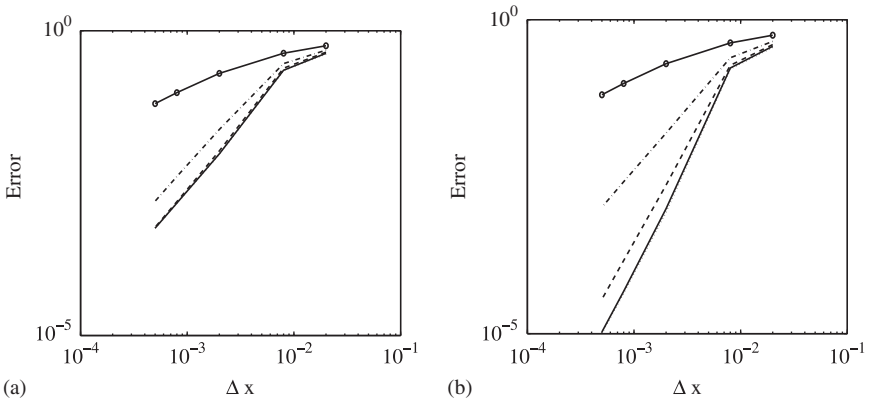

Figure 4. Error function as mesh is refined for: $k=0$, (-o-); $k=1$, (– · –); $k=2$ (– –); $k=3$ (–); $k=4$ (···). $u=0.75$: (a) classical Mpdata and (b) third-order Mpdata.

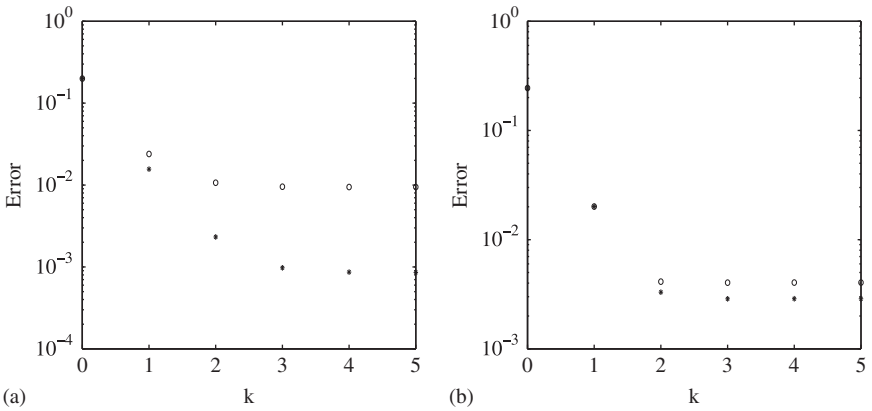

Figure 5. Error function as k increases: $\Delta x=0.002$ classical Mpdata (○); third-order Mpdata (∗): (a) $u=0.75$ and (b) $u=0.5$.

the order of accuracy for $k>1$ and this is highlighted in Tables I–IV, where we see that for $k\geqslant 2$ the order of accuracy is the same.

In Table III, for $u=0.5$, that is, $v=0.5$ since we are considering $\Delta t=\Delta x$, the classical Mpdata reaches an order of accuracy similar to the third-order Mpdata scheme. We note that K_v takes its maximum at $v=0.5$ and we have

$$B_v=\frac{|v|}{6}(1-3v+2v^2)=\frac{|v|}{6}(2v-1)(v-1)$$

with $B_v=0$ for $v=0.5$. Therefore, it is not surprising that for $v=0.5$, the classical Mpdata reaches smaller errors that are similar to the third-order Mpdata.

DOI: 10.1002/fld

Table I. Estimated convergence rate p for error assuming Error$\approx(\Delta x)^p$ at $u=0.1$ for the classical Mpdata and the third-order Mpdata.

Method	$k=0$	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
Classical	0.6	1.6	1.8	1.8	1.8	1.8
Third order	0.6	1.7	2.5	2.7	2.7	2.7

Table II. Estimated convergence rate p for error assuming Error$\approx(\Delta x)^p$ at $u=0.25$ for the classical Mpdata and the third-order Mpdata.

Method	$k=0$	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
Classical	0.6	1.5	1.8	1.8	1.8	1.8
Third order	0.6	1.7	2.5	2.5	2.5	2.5

Table III. Estimated convergence rate p for error assuming Error$\approx(\Delta x)^p$ at $u=0.5$ for the classical Mpdata and the third-order Mpdata.

Method	$k=0$	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
Classical	0.6	1.6	2.4	2.4	2.4	2.4
Third order	0.6	1.6	2.5	2.5	2.5	2.5

Table IV. Estimated convergence rate p for error assuming Error$\approx(\Delta x)^p$ at $u=0.75$ for the classical Mpdata and the third-order Mpdata.

Method	$k=0$	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
Classical	0.6	1.5	1.8	1.8	1.8	1.8
Third order	0.6	1.6	2.5	2.8	2.8	2.8

4.2. Test problem for $D>0$

Consider the advection–diffusion problem for $x\in[0,1]$ and the initial condition given by

$$\psi(x,0)=\mathrm{e}^{-(x-0.5)^2/L^2},\quad x\in[0,1] \tag{39}$$

The exact solution of this problem can be obtained from the eigenfunctions of the spatial differential operator, which are sines and cosines. Hence, the solution is given by means of a Fourier expansion

$$\psi(x,t)=\sum_{k=-\infty}^{+\infty} b_k \mathrm{e}^{-4\pi^2k^2Dt}\mathrm{e}^{2\pi k(x-Vt)\mathrm{i}} \tag{40}$$

with

$$b_k=\int_0^1 \mathrm{e}^{-(x-0.5)^2/L^2}\mathrm{e}^{2\pi kx\mathrm{i}}\,\mathrm{d}x$$

DOI: 10.1002/fld

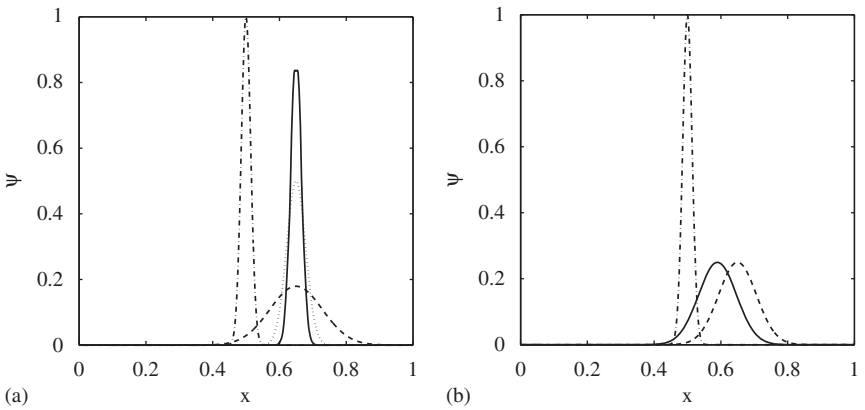

Figure 6. Initial solution (– · –). Exact solution at $t=0.3$ for: (a) $u=0.5$, $D=0.01$ (– –); $u=0.5$, $D=0.001$ (· · ·); $u=0.5$, $D=0.0001$ (–) and (b) $u=0.5$, $D=0.005$ (–); $u=0.3$, $D=0.005$ (– –).

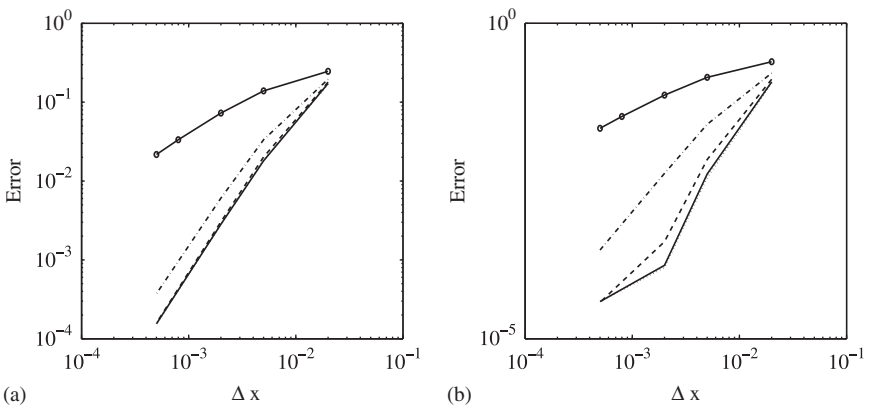

Figure 7. Error function as mesh is refined for: $k=0$, (-o-); $k=1$, (– · –); $k=2$ (– –); $k=3$ (–); $k=4$ (· · ·). $u=0.5$, $D=0.001$. (a) classical Mpdata and (b) third-order Mpdata. Order of convergence in Table VI.

We are assuming periodic boundary conditions. In Figure 6 we plot the exact solutions for the five cases we have chosen to run the experiments.

In Figures 7 and 8 we plot the error versus the space step for the classical Mpdata and the third-order Mpdata and for different iterations. It is highlighted by these figures that the corrective iteration $k=2$ is the most effective. Further increase in iteration number has very little effect. As we refine the mesh the iterative process can be less advantageous. This is illustrated in Figure 7(b), where for smaller Δx the error is the same for $k=2$ and $k=3$, although for large values of Δx three iterations lead to significant smaller errors.

From Figure 5 of the advection problem and Figure 9 of the advection–diffusion problem it can be observed that the effect of increasing the k-iterations is similar for the advection equation and the advection–diffusion equation, that is, it is not worth to iterate more than three times.

DOI: 10.1002/fld

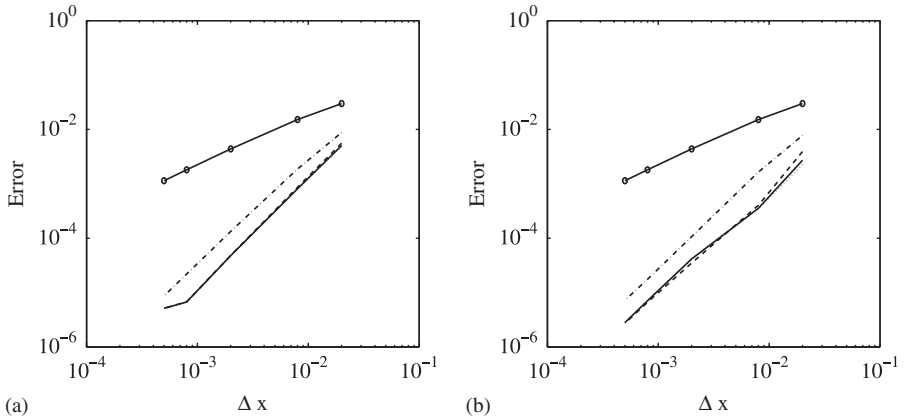

Figure 8. Error function as mesh is refined for: $k=0$, (-o-); $k=1$, (– · –); $k=2$ (– –); $k=3$ (–); $k=4$ (· · ·). $u=0.5$, $D=0.01$. (a) classical Mpdata and (b) third-order Mpdata. Order of convergence in Table V.

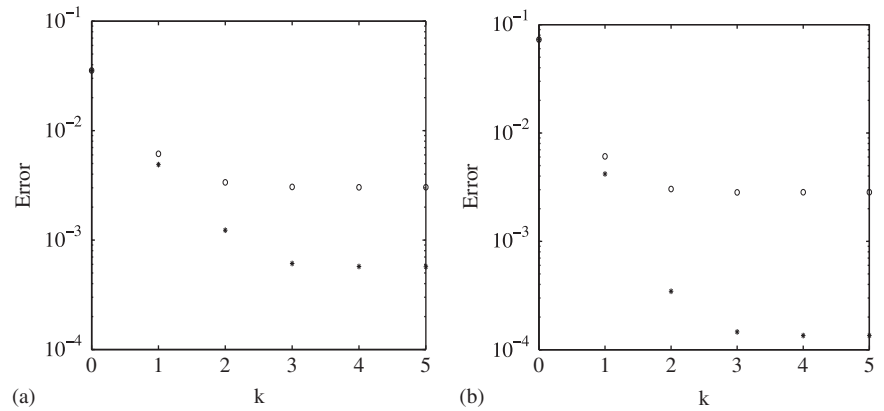

Figure 9. Error function as k increases: Mpdata classical (○); Mpdata third order (∗): (a) $u=0.5$, $D=0.005$, $\Delta x=0.008$ and (b) $u=0.5$, $D=0.001$, $\Delta x=0.002$.

Tables V–IX show the order of accuracy of the classical Mpdata and the third-order Mpdata. Contrary to what happened in the previous section, for $D=0$, we observe that the third-order Mpdata in general does not reach third order. This is consistent with the theoretical analysis we have done in Section 3. It was also found that the order of accuracy depends on the values v, μ, that is, on the Peclet number

$$\mathbf{Pe}=\frac{v}{\mu}$$

For large Peclet numbers the methods seem not to perform well as shown in Table VII. We note that this behavior is initially associated with the classical first-order upstream scheme, that is, $k=0$. Then, we observe that to iterate for this case does not improve the order of accuracy and does not

DOI: 10.1002/fld

Table V. Estimated convergence rate p for error assuming Error$\approx(\Delta x)^p$ for $u=0.5$, $D=0.01$, $t=0.3$, $v=0.001$, $\mathbf{Pe}=50\Delta x$.

Method	$k=0$	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
Classical	0.9	1.9	1.9	1.9	1.9	1.9
Third order	0.9	1.9	2.0	1.9	1.8	1.8

Table VI. Estimated convergence rate p for error assuming Error$\approx(\Delta x)^p$ for $u=0.5$, $D=0.001$, $t=0.3$, $v=0.01$, $\mathbf{Pe}=500\Delta x$.

Method	$k=0$	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
Classical	0.7	1.7	1.9	1.9	1.9	1.9
Third order	0.7	1.8	2.2	2.2	2.2	2.2

Table VII. Estimated convergence rate p for error assuming Error$\approx(\Delta x)^p$ for $u=0.5$, $D=0.0001$, $t=0.3$, $v=0.01$, $\mathbf{Pe}=5000\Delta x$.

Method	$k=0$	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
Classical	0.4	0.7	0.7	0.6	0.6	0.6
Third order	0.4	0.6	0.6	0.6	0.6	0.6

Table VIII. Estimated convergence rate p for error assuming Error$\approx(\Delta x)^p$ for $u=0.3$, $D=0.005$, $t=0.3$, $v=0.001$, $\mathbf{Pe}=60\Delta x$.

Method	$k=0$	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
Classical	0.9	1.9	2.1	2.1	2.0	2.0
Third order	0.9	1.9	2.0	1.9	1.8	1.8

Table IX. Estimated convergence rate p for error assuming Error$\approx(\Delta x)^p$ for $u=0.5$, $D=0.005$, $t=0.3$, $v=0.001$, $\mathbf{Pe}=100\Delta x$.

	$k=0$	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
Classical	0.8	1.9	2.1	2.0	2.0	2.0
Third order	0.8	1.9	2.1	1.9	1.9	1.9

diminish the error significantly. Both methods seem to have second-order accuracy although for some cases the third-order Mpdata seems of higher order as highlighted in Table X.

The purpose of Table X is to illustrate that for some intervals of Δx the order of accuracy of the third-order Mpdata can be around third order. In Table X we present the error values for a certain interval of Δx where the third-order scheme reaches order 3.5 if we assume $k=3$. This feature is also visible in Figure 7(b). This order of accuracy seems to be the result of the diminishing

DOI: 10.1002/fld

Table X. Error results for $u=0.5$, $D=0.001$. For some intervals of the space step the order of convergence of the third-order Mpdata can be around three although in general it is around second order.

Iterations	Method	$\Delta x=0.008$	$\Delta x=0.002$	p
$k=0$	Both	1.8009e−01	7.2765e−02	0.7
$k=1$	Classical	7.0004e−02	6.0779e−03	1.8
	Third order	5.6337e−02	4.1845e−03	1.9
$k=2$	Classical	4.9920e−02	3.0399e−03	2.0
	Third order	2.5185e−02	3.4568e−04	3.1
$k=3$	Classical	4.4139e−02	2.8317e−03	2.0
	Third order	1.7920e−02	1.4598e−04	3.5

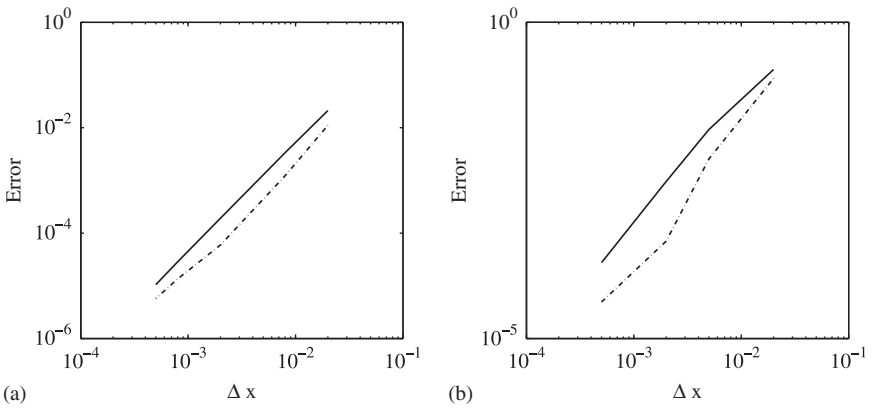

Figure 10. Error versus space step for classical Mpdata (–) and third-order Mpdata (– · –) and $k=2$: (a) $u=0.5$, $D=0.005$. Same order of accuracy but smaller error for the third-order Mpdata. (b) $u=0.5$, $D=0.001$. High-order accuracy for the third-order Mpdata.

amplitude of the error as we already pointed out previously since theoretically it is only proved that the method is second order as it happens in general.

Figure 10 shows the error versus the space step for two iterations, that is, $k=2$. We display in Figure 10(a) an example of the classical Mpdata and the third-order Mpdata which order of accuracy is the same, although the error is smaller for the third order Mpdata. Figure 10(b) shows an example where the third order Mpdata is of higher order.

In Figures 7 and 8 some of the presented convergence curves may reveal a finite precision effect. This may be due to the fact that the iterative procedure implicitly involves the numerical computation of increasingly high-order derivatives, which tends to be a numerically unstable operation. This behavior seems to appear only for short space steps Δx, and mainly for $k\geqslant 3$. Therefore, in practical applications when running $k=3$ or $k=4$ iterations we should not use a very refined mesh or if we use a more refined mesh we are advised to run $k=1$ or $k=2$ iterations.

Finally, we would like to illustrate in Figure 11 one of the important advantages of using these schemes which is the fact that they do not present oscillations. It is also visible in this figure that for $k=2$ we have a good approximated solution.

DOI: 10.1002/fld

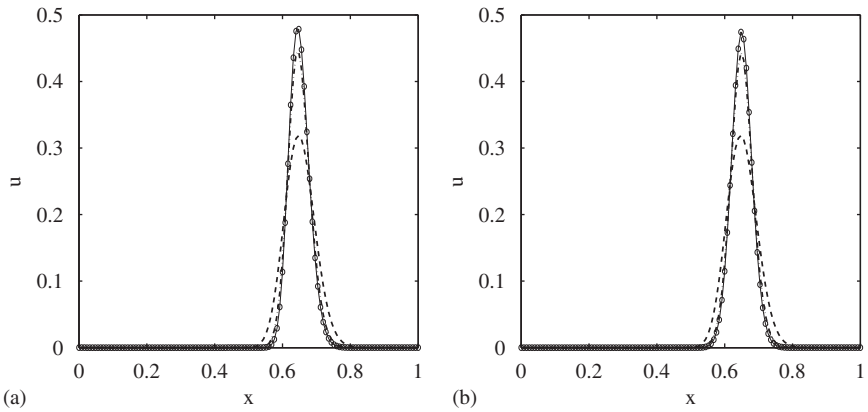

Figure 11. Approximated solution for $u=0.5$, $D=0.001$, and $\Delta x=0.008$ $k=0$ (– –); $k=1$ (– · –); $k=2$ (–○–): (a) classical Mpdata and (b) third-order Mpdata.

5. FINAL REMARKS

In this paper we present results for the classical Mpdata and the third-order Mpdata schemes applied to the one-dimensional unsteady advection–diffusion equation. The order of convergence of both methods in general increases for one and two iterations, but usually does not increase for more than two iterations, although the magnitude of the error can diminish. We can say that after the third iteration there are no significant improvements for both schemes and the best choice seems to be two iterations. Furthermore, the third-order Mpdata is not, in some situations, of a higher order, although the magnitude of the error is still smaller than the magnitude of the error of the classical Mpdata method. Additionally, both schemes perform better for small Peclet numbers.

Considering the advantages of using these methods in comparison with other methods in the literature, we can point out that these methods are easy to implement and for a small number of iterations, namely two iterations, the schemes reach a good order of accuracy and the error seems to diminish considerably. In problems where we need to deal with boundary conditions, the fact that the scheme only uses three discrete points can be very helpful since it avoids the need for implementing numerical boundary conditions and its disadvantages, see for instance, [22]. In situations where it is fundamental to avoid oscillations, this seems to be very adequate, specially if we are interested in problems with small Peclet numbers.

The extension of the results presented from one dimension to two dimensions is not straightforward in the sense that the recursive velocities contain cross derivatives that appear in the truncation error analysis. However, the extension from two dimensions to three is straightforward. The presence of cross derivative terms in the truncation error gives us a degree of freedom, when we write the pseudo-velocities, not present in the one-dimensional analysis. Additionally, in one dimension the sign of the pseudo-velocities did not change through the recursion process due to the stability condition. In two dimensions, generalization of this condition applies. In terms of accuracy and number of required deferred corrections, although we did not run any experiments,

DOI: 10.1002/fld

in some situations, it is expected that we may need one or two more iterations to have similar accuracy results.

REFERENCES

1. Prusa JM, Smolarkiewicz PK, Garcia RR. On the propagation and breaking at high altitudes of gravity waves excited by tropospheric forcing. *Journal of the Atmospheric Sciences* 1996; **53**:2186–2216.
2. Margolin LG, Reisner JM, Smolarkiewicz PK. Application of the volume of fluid method to the advection-condensation problem. *Monthly Weather Review* 1997; **125**:2265–2273.
3. Smolarkiewicz PK. A simple positive definite advection scheme with small implicit diffusion. *Monthly Weather Review* 1983; **111**:479–486.
4. Smolarkiewicz PK. A fully multidimensional positive definite advection transport algorithm with small implicit diffusion. *Journal of Computational Physics* 1984; **54**:325–362.
5. Smolarkiewicz PK, Clark TL. The multidimensional positive definite advection transport algorithm: further development and applications. *Journal of Computational Physics* 1986; **67**:396–438.
6. Smolarkiewicz PK, Margolin LG. MPDATA: a finite difference solver for geophysical flows. *Journal of Computational Physics* 1998; **140**:459–480.
7. Smolarkiewicz PK. Multidimensional positive definite advection transport algorithm: an overview. *International Journal for Numerical Methods in Fluids* 2006; **50**:1123–1144.
8. Hecht MW, Bryan FO, Holland WR. A consideration of tracer advection schemes in a primitive equation ocean model. *Journal of Geophysical Research* 1998; **103**:3301–3321.
9. Hecht MW. Forward-in-time upwind-weighted methods in ocean modelling. *International Journal for Numerical Methods in Fluids* 2006; **50**:1159–1173.
10. Margolin LG, Shashkov M. MPDATA: gauge transformations, limiters and monotonicity. *International Journal for Numerical Methods in Fluids* 2006; **50**:1193–1206.
11. Smolarkiewicz PK, Szmelter J. MPDATA: an edge-based unstructured grid formulation. *Journal of Computational Physics* 2005; **206**:624–649.
12. Harten A. High resolution schemes for hyperbolic conservation laws. *Journal of Computational Physics* 1983; **49**:357–393.
13. Harten A, Osher S. Uniformly high-order accurate non-oscillatory schemes. *SIAM Journal on Numerical Analysis* 1987; **24**:279–309.
14. Hernandez JA. High order finite volume schemes for the advection–diffusion equation. *International Journal for Numerical Methods in Engineering* 2001; **53**:1211–1234.
15. Leonard BP, Mokhtari S. Beyond first-order upwinding the ultra-sharp alternative for non-oscillatory steady-state simulation of convection. *International Journal for Numerical Methods in Engineering* 1990; **30**:729–766.
16. Morton KW. *Numerical Solution for the Convection–Diffusion Equation*. Chapman & Hall: Berlin, 1996.
17. Margolin LG, Smolarkiewicz PK. Antidiffusive velocities for multipass donor cell advection. *SIAM Journal on Scientific Computing* 1998; **20**:907–929.
18. Smolarkiewicz PK. On forward-in-time differencing for fluids. *Monthly Weather Review* 1991; **119**:2505–2510.
19. Smolarkiewicz PK, Margolin LG. On forward-in-time differencing for fluids: extension to a curvilinear framework. *Monthly Weather Review* 1993; **119**:1847–1859.
20. Stevens DE, Bretherton S. A forward-in-time advection scheme and adaptative multilevel flow solver for nearly incompressible atmospheric flow. *Journal of Computational Physics* 1996; **129**:284–295.
21. Ahmad NN, Bacon DP, Hall MS. Application of the multidimensional positive definite advection transport algorithm (MPDATA) to environmental modelling on adaptive unstructured grids. *International Journal for Numerical Methods in Fluids* 2006; **50**:1247–1268.
22. Sousa E. Higher order schemes and numerical boundary conditions. *Computer Methods in Applied Mechanics and Engineering* 2007; **196**:4444–4457.

DOI: 10.1002/fld