Pré-Publicações do Departamento de Matemática
Universidade de Coimbra
Preprint Number 08–10

# INSIGHTS ON A SIGN-PRESERVING NUMERICAL METHOD FOR THE ADVECTION-DIFFUSION EQUATION

ERCÍLIA SOUSA

ABSTRACT: 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 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 is advantageous to use the numerical methods presented.

KEYWORDS: advection-diffusion; non-oscillatory schemes; finite differences.

## 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. In many hydrodynamical systems, such as in the presence of turbulent diffusion, dealing with a shock or discontinuity is not as important as maintaining the positive definiteness of the evaluated scalar quantity. 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.

Multidimensional 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 substance constituents in atmospheric cloud models [3] and further generalisations were presented in [4], [5]. Nowadays Mpdata has been used in different contexts and some review

Received January 17, 2008.

papers have been written about this method, see for instance, [6], [7]. Several applications and comparison with other methods can be found in literature, for instance in [8],[9], [10], [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 an high-order according to which terms of the leading truncation error we choose to construct the pseudo-velocities. In comparison to the TVD (total variation diminishing) schemes [12] and ENO (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. This type of schemes are particularly important for an advection problem but also for an advection-diffusion problem [14], [15], [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 we shall use the methods.

The paper is organized as follow. 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 analyse 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. In the end we present some test problems. We comment 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 done for the advection equation. Consider the equation

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

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

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

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

where $F$ is such that

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

and

$$\nu_i = u_{i-1/2}\Delta t/\Delta x, \qquad u_{i-1/2} = u(x_i - \Delta x/2).$$

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

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

For $u$ not constant this is the CFL condition which is a necessary condition for stability.

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

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

where $\nu = u\Delta t/\Delta x$ 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

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

From (6) it can be seen that when $\Delta t$ and $\Delta x$ goes to zero (6) approaches (1), but during a realistic computational process the scheme in (2) with finite $\Delta x$ and $\Delta t$ 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 intuitively 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

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

where

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

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

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

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

Note that we can write (7) as

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

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

In the approximation of (11) by (2) the factor in the numerator of $\mathrm{v}^{(1)}$ 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

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

a first-order accurate estimate is

$$\nu_i^{(1)} \;=\; (|\nu| - \nu^2)\frac{\Psi_i^{(1)} - \Psi_{i-1}^{(1)}}{\Psi_i^{(1)} + \Psi_{i-1}^{(1)}}. \tag{12}$$

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

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

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

The corrective step is suggested in the form

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

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

Mpdata works in the following way:

1 - First we compute (2) to obtain $\Psi_i^{(1)}$.

2 - Secondly, we calculate the pseudo-velocity $\nu_i^{(1)}$ using (12).

3 - We determine the final value $\Psi_i^{n+1}$ using (13).

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

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

before calculate the final value $\Psi_i^{n+1}$ and where

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

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.

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

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

and therefore

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

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

$$\begin{aligned}\nu_i^{(k)} &= (|\nu_i^{(k-1)}| - (\nu_i^{(k-1)})^2)\frac{\Psi_i^{(k)} - \Psi_{i-1}^{(k)}}{\Psi_i^{(k)} + \Psi_{i-1}^{(k)}} \\ &\quad -\frac{4}{6}(\nu_i^{(k-1)} - 3(\nu_i^{(k-1)})^2 + 2(\nu_i^{(k-1)})^3)\frac{\Psi_{i-2}^{(k)} - 2\Psi_{i-1}^{(k)} + \Psi_i^{(k)}}{\Psi_{i-2}^{(k)} + 2\Psi_{i-1}^{(k)} + \Psi_i^{(k)}}.\end{aligned}$$

The factor in the denominator of the second term of $\nu_i^{(k)}$ 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 $|\nu_i^{(k)}| \leq 1$.

# 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

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

for $v = u + \omega$ where $\omega = \begin{cases} -\frac{(D\psi)_x}{\psi}, & \psi \neq 0 \\ 0, & \psi = 0. \end{cases}$

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

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

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

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

with

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

where $\mu$ 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 discretised $\alpha_i^{(1)}$, 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 $\Psi_i^{n+1}$ and where

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

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

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

The matricial 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 $\Psi^n$ and $\Psi^{(1)}$ the vectors of discretised values given by $\Psi^n = [\Psi_0^n, \ldots, \Psi_{N-1}^n]^T$ and $\Psi^{(1)} = [\Psi_0^{(1)}, \ldots, \Psi_{N-1}^{(1)}]^T$ respectively.

Secondly, 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 $\Psi^{(k)} = [\Psi_0^{(k)}, \ldots, \Psi_{N-1}^{(k)}]^T$. Therefore,

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

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

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

where

$$\begin{aligned} d_{0p}^{(k)} &= 1 - \frac{1}{2}(\alpha_{p+1}^{(k)} + |\alpha_{p+1}^{(k)}|) + \frac{1}{2}(\alpha_p^{(k)} - |\alpha_p^{(k)}|) \\ d_{-1p}^{(k)} &= \frac{1}{2}(\alpha_p^{(k)} + |\alpha_p^{(k)}|) \end{aligned}$$

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

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 $\Delta x$ is very small the iterative process is less effective for more iterations.

### 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 $\Delta t, \Delta x \to 0$ then $\alpha_i^{(k)} \to 0$ for all $k$. The latter implies that (22) for $k > 1$ converges to $\frac{\partial \psi}{\partial t} = 0$, 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 analising 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

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

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

$$\begin{aligned}\psi^{(2)} &= \psi^{(1)} - \int_{t_n}^{t_{n+\Delta t}} \frac{\partial}{\partial x}(v^{(1)}\psi^{(1)})dt + \frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_{n+\Delta t}} \frac{\partial}{\partial x}(K_\alpha^{(1)}\psi_x^{(1)})dt \\ &\quad + O\left(\frac{\Delta x^3}{\Delta t}\right)\Delta t \end{aligned} \quad (27)$$

for

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

We have

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

Similarly the $k$-iteration is given by

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

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 $(\Delta x^3/\Delta t)\Delta t$,

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

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

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

with the new term at the $k$ iteration

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

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

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

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

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

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

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

$$\psi^{n+1}=\psi^n-\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}(v\psi)dt+O\left(\Delta x^2\right)\Delta t. \tag{31}$$

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

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

$$\begin{aligned}\psi^{(1)} &= \psi^n-\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}(v\psi)dt+\frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}(K_\alpha\psi_x)dt\\ &\quad -\frac{\Delta x^3}{\Delta t}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}(B_\alpha\psi_{x^2})dt+O\left(\frac{\Delta x^4}{\Delta t}\right)\Delta t\end{aligned} \tag{32}$$

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

$$\begin{aligned}\psi^{(2)} &= \psi^{(1)}-\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}(v^{(1)}\psi^{(1)})dt+\frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}(K_\alpha^{(1)}\psi_x^{(1)})dt\\ &\quad -\frac{\Delta x^3}{\Delta t}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}(B_\alpha^{(1)}\psi_{x^2}^{(1)})dt+O\left(\frac{\Delta x^4}{\Delta t}\right)\Delta t\end{aligned} \tag{33}$$

for

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

$$K_\alpha^{(1)}=\frac{1}{2}(|\alpha^{(1)}|-(\alpha^{(1)})^2)\qquad B_\alpha^{(1)}=\frac{1}{6}(\alpha^{(1)}-3(\alpha^{(1)})^2+2(\alpha^{(1)})^3).$$

We get

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

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

Therefore, the $k$-iteration is given by

$$\begin{aligned}\psi^{(k)} &= \psi^{(k-1)} - \frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}[(K_\alpha^{(k-2)} - K_\alpha^{(k-1)})\psi_x^{(k-1)}]dt \\ &\quad +\frac{\Delta x^3}{\Delta t}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}((B_\alpha^{(k-2)} - B_\alpha^{(k-1)})\psi_{x^2}^{(k-1)})dt + O\left(\frac{\Delta x^4}{\Delta t}\right)\Delta t(35)\end{aligned}$$

where

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

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

We can write,

$$\begin{aligned}\psi^{n+1} &= \psi^{(k)} - \frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}[(K_\alpha^{(k-1)} - K_\alpha^{(k)})\psi_x^{(k)}]dt \\ &\quad +\frac{\Delta x^3}{\Delta t}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}[(B_\alpha^{(k-1)} - B_\alpha^{(k)})\psi_{x^2}^{(k)}]dt + O\left(\frac{\Delta x^4}{\Delta t}\right)\Delta t. \quad (36)\end{aligned}$$

Similarly 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

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

with the new terms at the $k$ iteration, respectively

$$\frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_{n+\Delta t}}K_\alpha^{(k)}\psi_x^{(k)}dt \qquad \frac{\Delta x^3}{\Delta t}\int_{t_n}^{t_{n+\Delta t}}B_\alpha^{(k)}\psi_{x^2}^{(k)}dt.$$

Taking in consideration the previous equalities we have

$$\begin{aligned}\psi^{n+1} &= \psi^{(1)} - \frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}(K_\alpha\psi_x^{(1)})dt + \frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}(K_\alpha^{(1)}\psi_x^{(1)})dt \\ &\quad +\frac{\Delta x^3}{\Delta t}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}(B_\alpha\psi_{x^2}^{(1)})dt - \frac{\Delta x^3}{\Delta t}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}(B_\alpha^{(1)}\psi_{x^2}^{(1)})dt\end{aligned}$$

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

and

$$\begin{aligned}\psi^{n+1} &= \psi^n-\int_{t_n}^{t_{n+\Delta t}}(v\psi)_x dt+\frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}(K_\alpha\psi_x)dt\\
&-\frac{\Delta x^3}{\Delta t}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}(B_\alpha\psi_{x^2})dt-\frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}\left\{K_\alpha\left[\psi_x\right.\right.\\
&-\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}(v\psi_x)dt+\frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x^2}(K_\alpha\psi_x)dt\\
&\left.\left.-\frac{\Delta x^3}{\Delta t}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x^2}(B_\alpha\psi_{x^2})dt\right]\right\}d\tau+\frac{\Delta x^3}{\Delta t}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}\left\{B_\alpha\left[\psi_{x^2}\right.\right.\\
&-\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x^2}(v\psi_x)dt+\frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x^3}(K_\alpha\psi_x)dt\\
&\left.\left.-\frac{\Delta x^3}{\Delta t}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x^3}(B_\alpha\psi_{x^2})dt\right]\right\}d\tau+\frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}(K_\alpha^{(1)}\psi_x^{(1)})dt\\
&-\frac{\Delta x^3}{\Delta t}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}(B_\alpha^{(1)}\psi_{x^2}^{(1)})dt\\
&-\frac{\Delta x^2}{\Delta t}\sum_{p=2}^{k}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}[(K_\alpha^{(p-1)}-K_\alpha^{(p)})\psi_x^{(p)}]dt\\
&+\frac{\Delta x^3}{\Delta t}\sum_{p=2}^{k}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}[(B_\alpha^{(p-1)}-B_\alpha^{(p)})\psi_{x^2}^{(p)}]dt+O\left(\frac{\Delta x^4}{\Delta t}\right)\Delta t.\end{aligned}$$

Finally,

$$\begin{aligned}\psi^{n+1} &= \psi^n-\int_{t_n}^{t_{n+\Delta t}}(v\psi)_x dt+\frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}\left[K_\alpha\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}(v\psi_x)dt\right]d\tau\\
&-\frac{\Delta x^4}{\Delta t^2}\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x}K_\alpha\left[\int_{t_n}^{t_{n+\Delta t}}\frac{\partial}{\partial x^2}(K_\alpha\psi_x)dt\right]d\tau\end{aligned}$$

$$-\frac{\Delta x^3}{\Delta t}\int_{t_n}^{t_n+\Delta t}\frac{\partial}{\partial x}B_\alpha\left[\int_{t_n}^{t_n+\Delta t}\frac{\partial}{\partial x^2}(v\psi_x)dt\right]d\tau$$
$$+\frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_n+\Delta t}\frac{\partial}{\partial x}(K_\alpha^{(1)}\psi_x^{(1)})dt-\frac{\Delta x^3}{\Delta t}\int_{t_n}^{t_n+\Delta t}\frac{\partial}{\partial x}(B_\alpha^{(1)}\psi_{x^2}^{(1)})dt$$
$$-\frac{\Delta x^2}{\Delta t}\sum_{p=2}^{k}\int_{t_n}^{t_n+\Delta t}\frac{\partial}{\partial x}[(K_\alpha^{(p-1)}-K_\alpha^{(p)})\psi_x^{(p)}]dt$$
$$+\frac{\Delta x^3}{\Delta t}\sum_{p=2}^{k}\int_{t_n}^{t_n+\Delta t}\frac{\partial}{\partial x}[(B_\alpha^{(p-1)}-B_\alpha^{(p)})\psi_x^{(p)}]dt+O\left(\frac{\Delta x^4}{\Delta t}\right)\Delta t$$

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

$$\frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_n+\Delta t}\frac{\partial}{\partial x}K_\alpha\left[\int_{t_n}^{t_n+\Delta t}\frac{\partial}{\partial x}(v\psi_x)dt\right]d\tau+\frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_n+\Delta t}\frac{\partial}{\partial x}(K_\alpha^{(1)}\psi_x^{(1)})dt$$
$$-\frac{\Delta x^2}{\Delta t}\sum_{p=2}^{k}\int_{t_n}^{t_n+\Delta t}\frac{\partial}{\partial x}[(K_\alpha^{(p-1)}-K_\alpha^{(p)})\psi_x^{(p)}]dt$$

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

$$\frac{\Delta x^2}{\Delta t}\int_{t_n}^{t_n+\Delta t}\frac{\partial}{\partial x}P_\alpha dt \tag{37}$$

for

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

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

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

for

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

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

$$\begin{aligned} P_\alpha &= K_\alpha \int_{t_n}^{t_{n+\Delta t}} \frac{\partial}{\partial x}(v\psi)_x dt + K_\alpha^{(1)}\psi_x - K_\alpha^{(1)} \int_{t_n}^{t_{n+\Delta t}} \frac{\partial}{\partial x}(v\psi)_x dt \\ &\quad - K_\alpha^{(1)}\psi_x^{(2)} + O(\Delta x^2) \end{aligned}$$

We can write

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

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 $k \geq 3$, 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 we have a uniform velocity field $u$. Then

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

For

$$\nu = \frac{u\Delta t}{\Delta x} \qquad \mu = \frac{D\Delta t}{\Delta x^2}$$

the discretised values $\alpha_i$ are given by (20), that is,

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

where $\xi_i'$s are local approximations of $\psi_x/\psi$. The CFL stability condition is given by

$$|\alpha_i| = |\nu - 2\mu\xi_i| \leq 1.$$

Assuming $\psi \geq 0$ we have $|\xi_i| \leq 1$. Therefore a stability condition is given by

$$\nu + 2\mu \leq 1,$$

since

$$|\nu - 2\mu\xi_i| \leq |\nu| + 2\mu|\xi_i| \leq \nu + 2\mu$$

and if $\nu + 2\mu \leq 1$ we have $|\nu - 2\mu\xi_i| \leq 1$.

We plot this region and a experimental stability region in Figure 1, where we display the Courant number, $\nu$, versus the Fourier number, $\mu$. The experimental stability region was computed by running calculations for different values of $\nu$ and $\mu$.

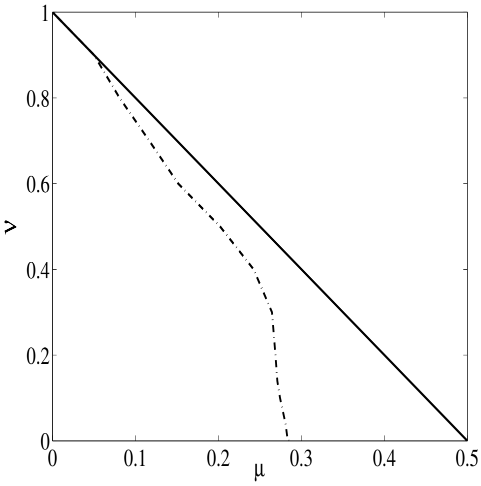


FIGURE 1. Stability region: Analytical necessary condition $(-)$; Experimental condition $(- \cdot -)$

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

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

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

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

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

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

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

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

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

and $|\alpha_i^{(k)}| \leq |\alpha_i^{(k-1)}|$ if $|\xi_i^{(k)}| + 2|\eta_i^{(k)}| \leq 1/2$. 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 $0 \leq x \leq 1$ and secondly 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 $0 \leq x \leq 1$ with initial condition

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

The exact solution is of the form

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

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

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

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

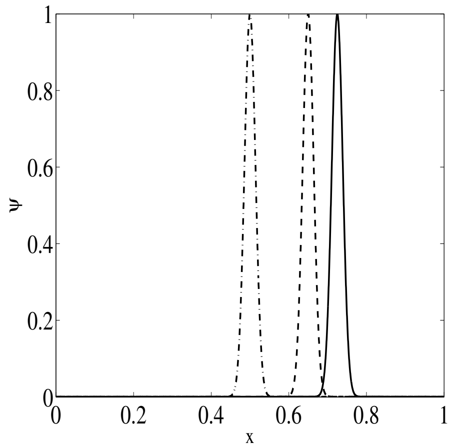


FIGURE 2. Initial solution $(- \cdot -)$. Exact solution at $t = 0.3$ for $u = 0.5$ $(--)$ and for $u = 0.75$ $(-)$

In Figures 3 and 4 we plot the error versus the mesh size, $\Delta x$, for $u = 0.5$ and $u = 0.75$ respectively. We observe that until two iterations we have considerable improvements in what concern 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). Onwards the improvements become meaningless. Therefore, for both methods is not worth to go further than three iterations, that is, $k = 3$. 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 Table 1 to Table 4 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 $k \geq 2$.

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 the order of accuracy for $k > 1$ and this is highlighted in Table 1 to Table 4, where we see that for $k \geq 2$ the order of accuracy is the same.

| 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 1. Estimated convergence rate $p$ for error assuming $Error \approx (\Delta x)^p$ at $\mathbf{u = 0.1}$ 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 | 1.5 | 1.8 | 1.8 | 1.8 | 1.8 |
| Third-order | 0.6 | 1.7 | 2.5 | 2.5 | 2.5 | 2.5 |

Table 2. Estimated convergence rate $p$ for error assuming $Error \approx (\Delta x)^p$ at $\mathbf{u = 0.25}$ 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 | 1.6 | 2.4 | 2.4 | 2.4 | 2.4 |
| Third-order | 0.6 | 1.6 | 2.5 | 2.5 | 2.5 | 2.5 |

Table 3. Estimated convergence rate $p$ for error assuming $Error \approx (\Delta x)^p$ at $\mathbf{u = 0.5}$ 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 | 1.5 | 1.8 | 1.8 | 1.8 | 1.8 |
| Third-order | 0.6 | 1.6 | 2.5 | 2.8 | 2.8 | 2.8 |

Table 4. Estimated convergence rate $p$ for error assuming $Error \approx (\Delta x)^p$ at $\mathbf{u = 0.75}$ for the classical Mpdata and the third-order Mpdata.

In Table 3, for $u = 0.5$, that is $\nu = 0.5$ 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 $K_\nu$ takes its maximum at $\nu = 0.5$ and we have

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

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

**4.2. Test problem for $D > 0$.** Consider the advection-diffusion problem for $x \in [0, 1]$ and the initial condition given by

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

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

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

with

$$b_k = \int_0^1 e^{-(x-0.5)^2/L^2} e^{2\pi kxi} dx.$$

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 Figure 7 and Figure 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 $\Delta x$ the error is the same for $k = 2$ and $k = 3$, although for large values of $\Delta x$ 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.

Table 5 to Table 9 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 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 $\nu$, $\mu$, that is, on the Peclet number,

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

For large Peclet numbers the methods seem not to perform well as shown in Table 7. To iterate for this case does not improve the order of accuracy and does not 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 10.

| 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 5. Estimated convergence rate $p$ for error assuming $Error \approx (\Delta x)^p$ for $u = 0.5$, $D = 0.01$, $t = 0.3$, $\nu = 0.001$, $\mathbf{Pe} = 50\Delta x$

| 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 6. Estimated convergence rate $p$ for error assuming $Error \approx (\Delta x)^p$ for $u = 0.5$, $D = 0.001$, $t = 0.3$, $\nu = 0.01$, $\mathbf{Pe} = 500\Delta x$

| 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 7. Estimated convergence rate $p$ for error assuming $Error \approx (\Delta x)^p$ for $u = 0.5$, $D = 0.0001$, $t = 0.3$, $\nu = 0.01$, $\mathbf{Pe} = 5000\Delta x$

| 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 8. Estimated convergence rate $p$ for error assuming $Error \approx (\Delta x)^p$ for $u = 0.3$, $D = 0.005$, $t = 0.3$, $\nu = 0.001$, $\mathbf{Pe} = 60\Delta x$

| | $k=0$ | $k=1$ | $k=2$ | $k=3$ | $k=4$ | $k=5$ |
|---|---|---|---|---|---|---|
| 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 |

Table 9. Estimated convergence rate $p$ for error assuming $Error \approx (\Delta x)^p$ for $u = 0.5$, $D = 0.005$, $t = 0.3$, $\nu = 0.001$, $\mathbf{Pe} = 100\Delta x$

| 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 |

Table 10. 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 is around second order.

The purpose of Table 10 is to illustrate that for some intervals of $\Delta x$ the order of accuracy of the third-order Mpdata can be around third-order. In Table 10 we present the error values for a certain interval of $\Delta x$ 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 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.

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.

## 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 improvement 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 advantages for using these methods in comparison with other methods in 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 avoid the need of implementing numerical boundary conditions and its disadvantages, see for instance, [18]. In situations where is fundamental to avoid oscillations, this seems to be very adequate, specially if we are interested in problems with small Peclet numbers.

# 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 *Intern. Journal of Num. Methods in Fluids*, 2006, **50**: 1123-1144.

[8] Hecht MW, Bryan FO and 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, *Intern. Journal of Num. Methods in Fluids*, 2006, **50**: 1159-1173.

[10] Margolin LG, Shashkov M, MPDATA: Gauge transformations, limiters and monotonicity,*Intern. Journal of Num. 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 J. Numer. Anal.*, 1987, **24**: 279-309.
[14] Hernandez JA, High order finite volume schemes for the advection-diffusion equation, *Int. J. Numer Methods Engrg* 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, *Int. j. Numer Methods Engrg* 1990 **30**: 729-766.
[16] Morton KW, Numerical solution for the convection-diffusion equation, Chapman & Hall, 1996.
[17] Margolin LG, Smolarkiewicz PK, Antidiffusive velocities for multipass donor cell advection, *SIAM J. Scientific Computing* , 1998, **20**: 907-929.
[18] Sousa E, Higher order schemes and numerical boundary conditions, *Comput. Methods Appl. Mech. Engrg.*, 2007, **196**: 4444-4457.

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

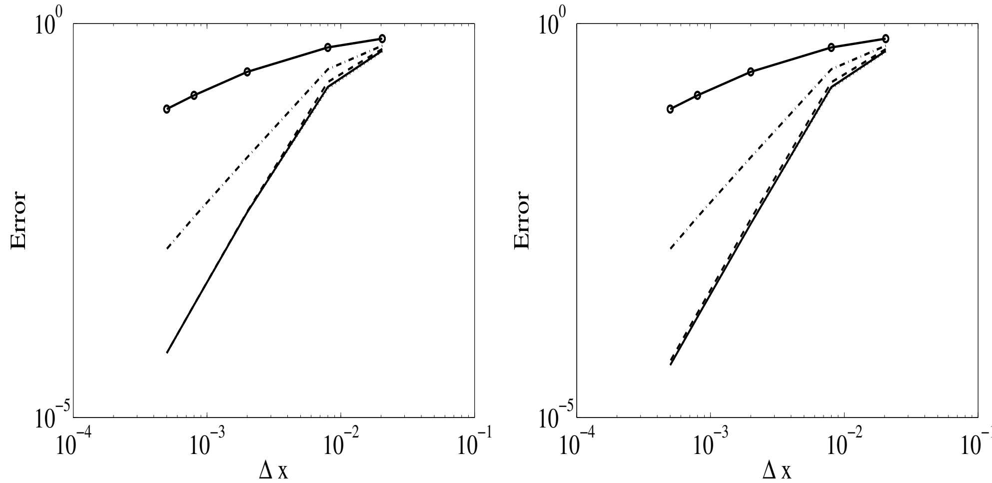


FIGURE 3. Error function as mesh is refined for: $k = 0$, $(-o-)$; $k = 1$, $(-- \cdot -)$; $k = 2$ $(--)$; $k = 3$ $(-)$; $k = 4$ $(\cdots)$ . $u = 0.5$ (a) Classical Mpdata (b)Third-order Mpdata

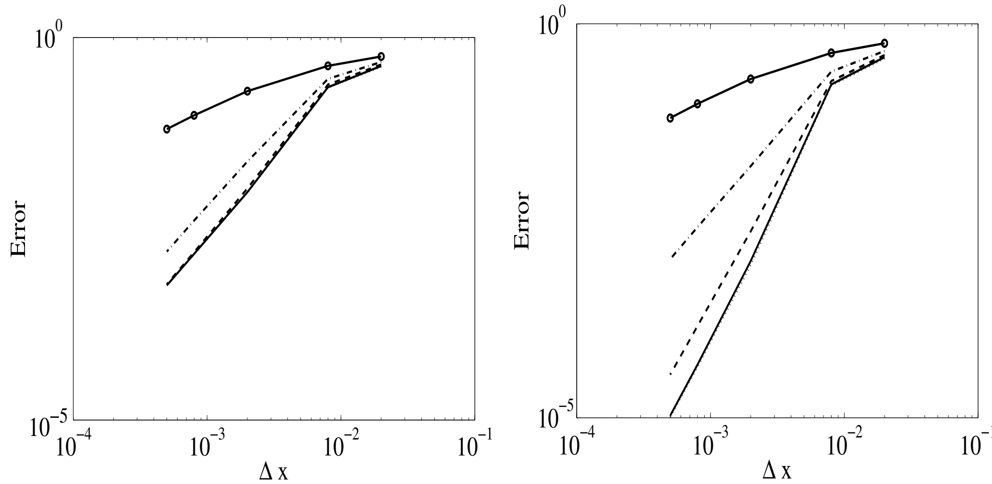


FIGURE 4. Error function as mesh is refined for: $k = 0$, $(-o-)$; $k = 1$, $(-- \cdot -)$; $k = 2$ $(--)$; $k = 3$ $(-)$; $k = 4$ $(\cdots)$ . $u = 0.75$ (a) Classical Mpdata (b) Third-order Mpdata

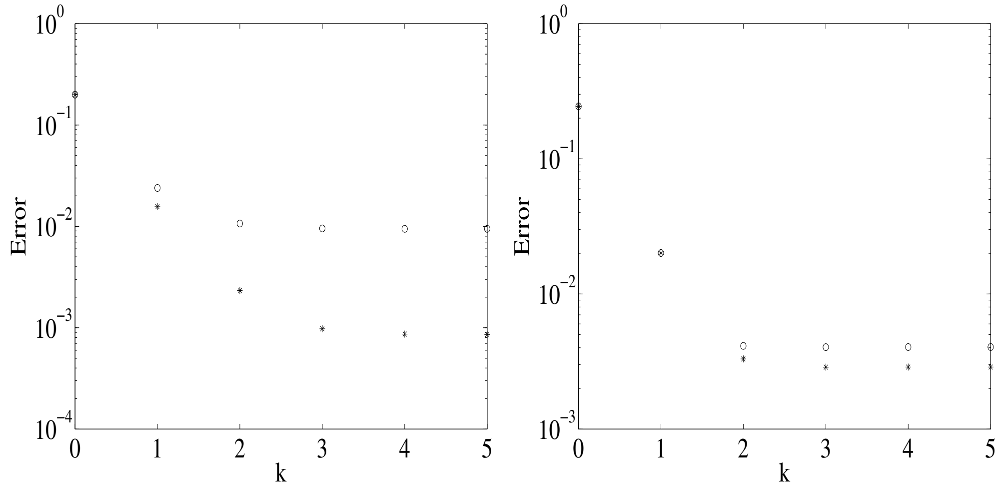


FIGURE 5. Error function as $k$ increases: $\Delta x = 0.002$ Classical Mpdata ($o$); Third-order Mpdata ($*$) (a) $u = 0.75$ (b)$u = 0.5$

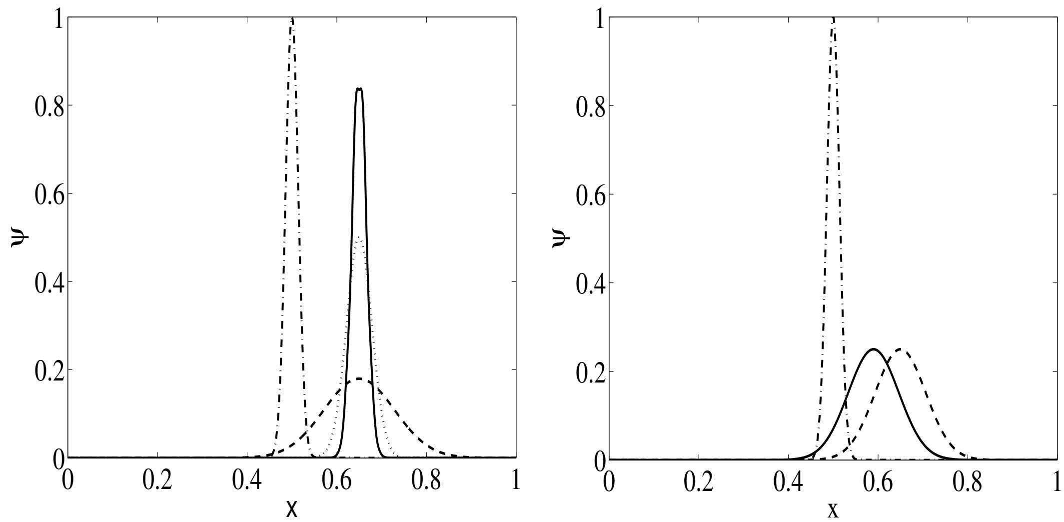


FIGURE 6. Initial solution $(- \cdot -)$. Exact solution at $t = 0.3$ for: (a) $u = 0.5$ $D = 0.01$ $(--)$; $u = 0.5$ $D = 0.001$ $(\cdots)$; $u = 0.5$ $D = 0.0001$ $(-)$. (b) $u = 0.5$ $D = 0.005$ $(-)$; $u = 0.3$ $D = 0.005$ $(--)$

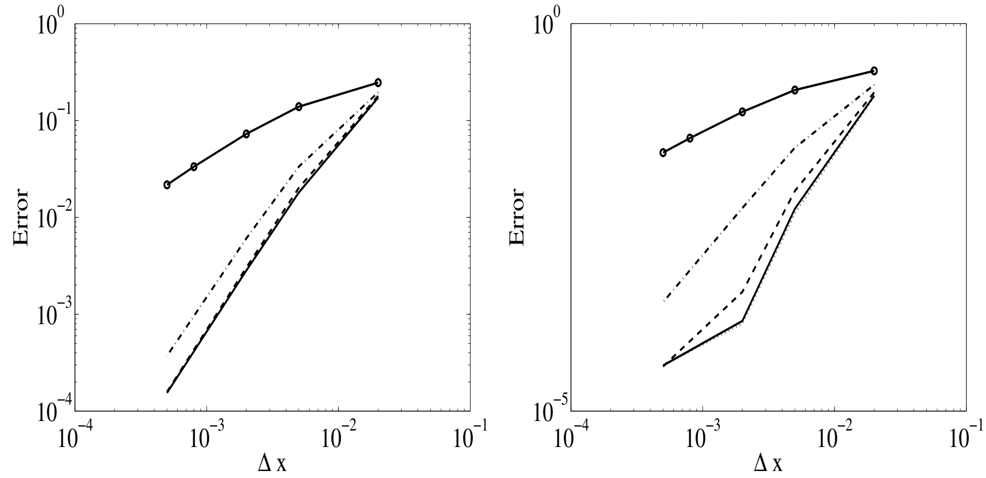


FIGURE 7. Error function as mesh is refined for: $k = 0$, $(-o-)$; $k = 1$, $(--\cdot-)$; $k = 2$ $(--)$; $k = 3$ $(-)$; $k = 4$ $(\cdots)$ . $u = 0.5$, $D = 0.001$ (a) Classical Mpdata (b) Third-order Mpdata. Order of convergence in Table 6.

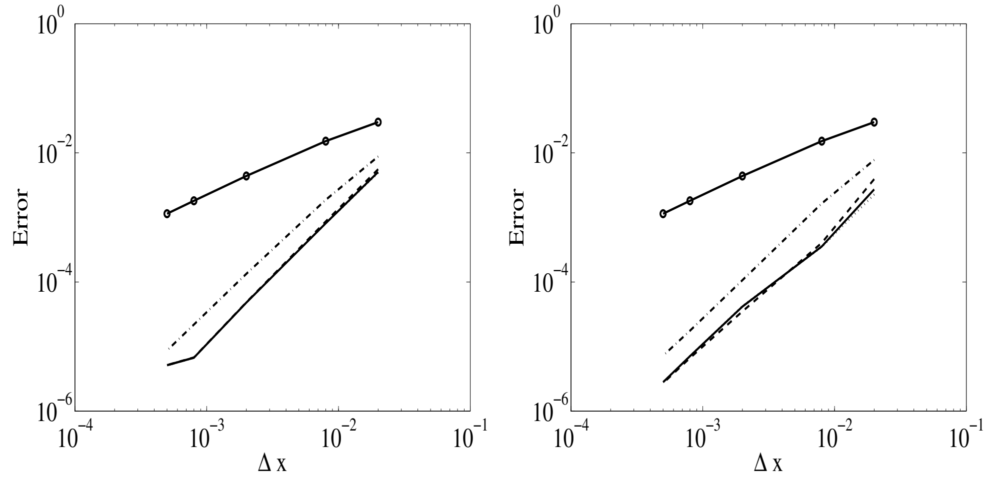


FIGURE 8. Error function as mesh is refined for: $k = 0$, $(-o-)$; $k = 1$, $(--\cdot-)$; $k = 2$ $(--)$; $k = 3$ $(-)$; $k = 4$ $(\cdots)$ . $u = 0.5$, $D = 0.01$ (a) Classical Mpdata (b) Third-order Mpdata. Order of convergence in Table 5.

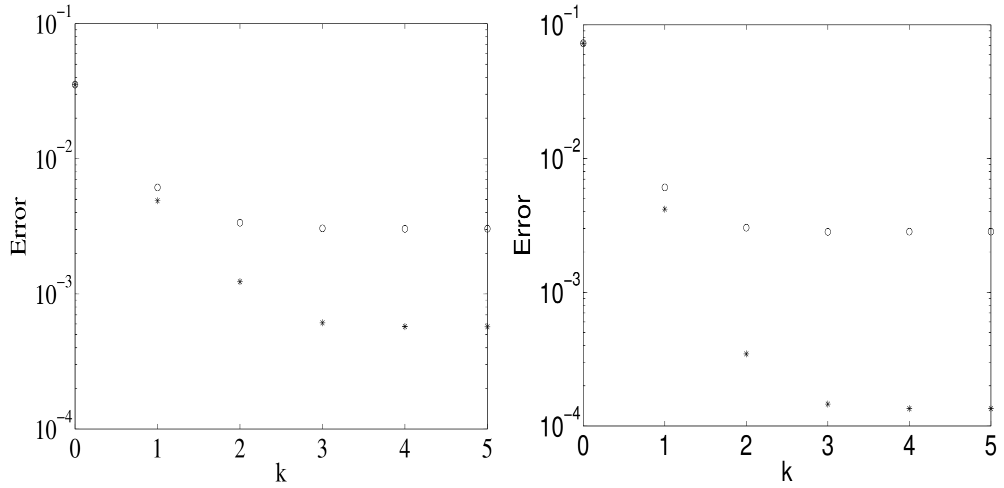


FIGURE 9. Error function as $k$ increases: Mpdata classical ($o$); Mpdata third-order ($*$) (a) $u = 0.5$ $D = 0.005$ $\Delta x = 0.008$; (b) $u = 0.5$ $D = 0.001$ $\Delta x = 0.002$

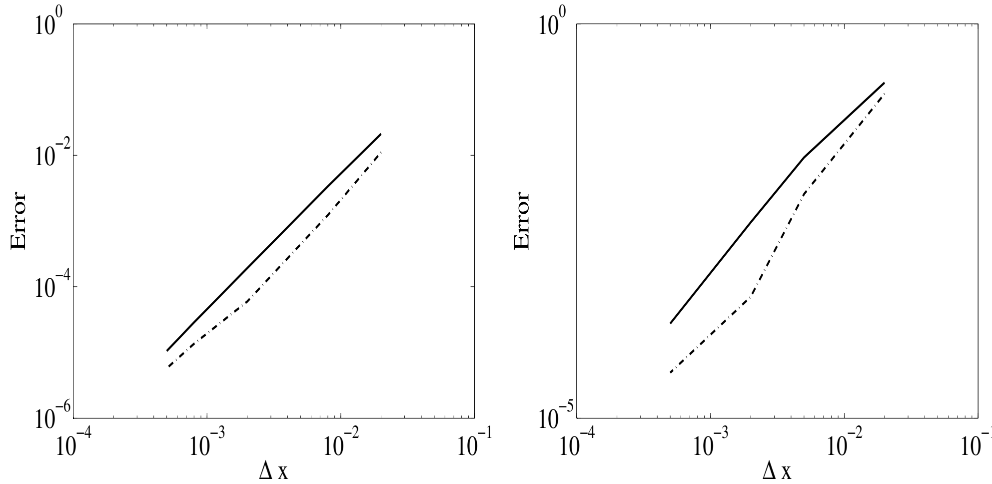


FIGURE 10. Error versus space step for Classical Mpdata ($-$) and Third-order Mpdata ($- \cdot -$) and $k = 2$. (a) $u = 0.5$, $D = 0.005$. Same order of accuracy but smaller error for the Third-order Mpdata. (b) $u = 0.5$, $D = 0.001$. High-order accuracy for the Third-order Mpdata

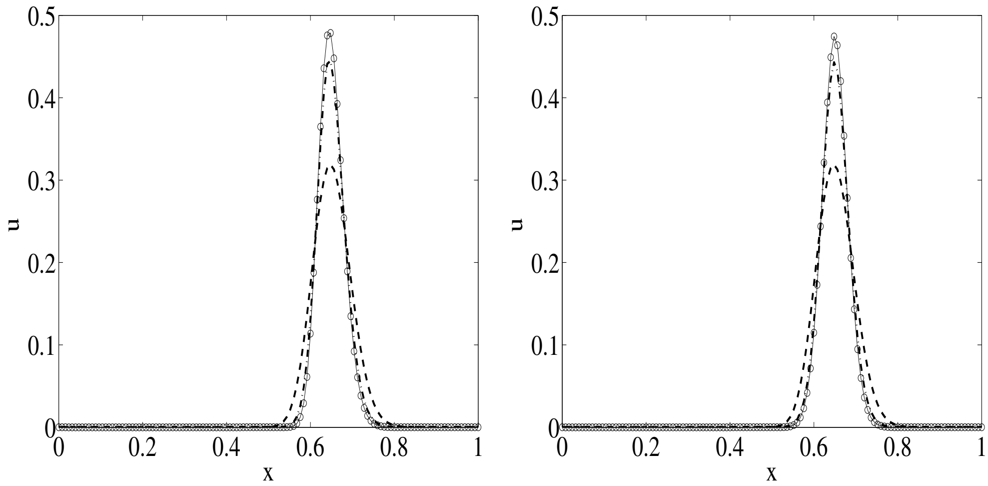


FIGURE 11. Approximated solution for $u = 0.5$, $D = 0.001$ and $\Delta x = 0.008$ $k = 0$ $(--)$; $k = 1$ $(- \cdot -)$; $k = 2$ $(-o-)$ (a)Classical Mpdata (b)Third-order Mpdata