ON THE QUASI-MONOTONE MODIFIED METHOD OF CHARACTERISTICS FOR TRANSPORT-DIFFUSION PROBLEMS WITH REACTIVE SOURCES

MOHAMMED SEAİD

Fachbereich Mathematik

TU Darmstadt, 64289 Darmstadt, Germany

E-mail: seaid@mathematik.tu-darmstadt.de

Abstract — This is an attempt to construct a strong numerical method for transport-diffusion equations with nonlinear reaction terms, which relies on the idea of the Modified Method of Characteristics that is explicit but stable and is second-order accurate in time. The method consists in convective-diffusive splitting of the equations along the characteristics. The convective stage of the splitting is straightforwardly treated by a quasi-monotone and conservative modified method of characteristics, while the diffusive-reactive stage can be approximated by an explicit scheme with an extended real stability interval. A numerical comparative study of the new method with Characteristics Crank-Nicholson and Classical Characteristics Runge-Kutta schemes, which are used in many transport-diffusion models, is carried out for several benchmark problems, whose solutions represent relevant transport-diffusion-reaction features. Experiments for transport-diffusion equations with linear and nonlinear reactive sources demonstrate the ability of our new algorithm to better maintain the shape of the solution in the presence of shocks and discontinuities.

2000 Mathematics Subject Classification: 76D50; 65M25; 65L06.

Keywords: convective and diffusive stages, modified method of characteristics, Runge-Kutta schemes.

1. Introduction

In this paper, given an open bounded domain $\Omega \subset \mathbb{R}^p$ ($p = 1, 2$ or 3) with a smooth boundary Γ and a time interval $[0, T]$, we are interested in solving approximately the transport-diffusion-reaction equations

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{u} - \operatorname{div}(\mathbf{D}(\mathbf{x}) \nabla \mathbf{u}) &= \mathbf{f}(\mathbf{u}, \mathbf{x}, t), \quad \text{in } \Omega \times (0, T), \\ \mathbf{u}(\mathbf{x}, t) &= 0, \quad \text{on } \Gamma \times [0, T], \\ \mathbf{u}(\mathbf{x}, 0) &= \mathbf{u}_0(\mathbf{x}), \quad \text{in } \Omega, \end{aligned} \tag{1.1}$$

where

$\mathbf{u}(\mathbf{x}, t)$ denotes, for example, the concentration of some chemical species;

$\mathbf{v}(\mathbf{x}, t)$ represents the velocity of a transport medium, such as water or air;

$\mathbf{D}(\mathbf{x})$ is the diffusion square, symmetric and positive definite matrix $(d_{ij}(\mathbf{x}))$;

∇ and $\mathbf{div}$ denote the gradient and divergence operators, respectively; and

$\mathbf{f}(\mathbf{u}, \mathbf{x}, t)$ gives a coupling between various concentrations and describes the linear or nonlinear reaction function together with emissions (sources) and depositions (sinks).

In the above and in what follows, the bold face denotes the vector quantities. We introduce the elliptic (diffusion) operator $\mathbf{A}$ defined by

$$\mathbf{A}\mathbf{w} := \mathbf{div}(\mathbf{D}(\mathbf{x})\nabla\mathbf{w}) \quad (1.2)$$

Problems of form (1.1) typically occur in air pollution [12, 42], fluid mechanics [35, 24], atmospheric motions [33, 23], and ocean circulation models [26, 32] or meteorology [36]. The existence and possible uniqueness of the solution to system (1.1) have been studied in a number of books and papers, among others, in [30, 15, 35, 7, 18]. Computation of its numerical solution is not trivial due to its nonlinearity, the presence of the convective terms $\mathbf{v} \cdot \nabla \mathbf{u}$ and the coupling of the equations through the reaction terms $\mathbf{f}(\mathbf{u}, \mathbf{x}, t)$.

Hence, in many transport-diffusion-reaction equations (1.1), the convective term is distinctly more important than the diffusive term; in particular, when certain nondimensional parameters reach high values. Among these parameters, the Reynolds number for the Navier–Stokes equations and the Peclet number for the diffusion-convection equations. This convective term is a source of computational difficulties and oscillations. It is well known that the solutions of equations (1.1) present steep fronts and even shock discontinuities, which need to be resolved accurately in applications and often cause severe numerical difficulties [18].

The Eulerian methods use fixed grids and incorporate some upstream weighting in their formulations to stabilize the schemes. Among the Eulerian methods are the Petrov–Galerkin methods, the streamline diffusion methods, and many other methods such as the high-resolution methods from computational fluid dynamics, in particular, the Godunov methods and the essentially non-oscillatory methods [8, 29]. All of these Eulerian methods are easy to formulate and implement. However, time truncation errors dominate their solutions and are subject to the Courant–Friedrichs–Lewy (CFL) stability conditions, which impose a restriction on the size of the time steps taken in numerical simulations.

The modified method of characteristics, on the other hand, makes use of the transport nature of the governing equations (1.1). They combine the fixed Eulerian grids with a particle tracking along the characteristic curves of the governing equations [6, 10, 24, 33, 34]. The Lagrangian treatment in these methods greatly reduces the time truncation errors in the Eulerian methods. In addition, these methods weaken the restrictions on the Courant number to allow for large time steps in the simulations.

Shape preservation and mass conservation are properties that may be relevant to some problems and, therefore, their importance cannot be overlooked. In order to avoid the principal drawback of the modified method of characteristics, that is the failure to conserve mass, we consider an improved modified method of characteristics with limiters to convert the method to quasi-monotone and conservative at a minor additional computational cost. A detailed analysis of this method and a similar procedure can be found in [4, 9].

The paper is arranged as follows. In section 2 the basic ideas of the quasi-monotone modified method of characteristics will be introduced. Section 3 is devoted to the combination

of the characteristics method with the Crank–Nicholson and classical Runge–Kutta schemes applied to the transport-diffusion-reaction problems. The new characteristics Runge–Kutta–Chebyshev scheme is set up in section 4. Several numerical experiments are presented in section 5. Section 6 contains the concluding remarks.

2. Quasi-Monotone Modified Method of Characteristics

In the Lagrangian frame of reference the evolution of the fluid in (1.1) is observed along the paths of imaginary fluid particles. There are some obvious disadvantages of evolving a set of particles along Lagrangian trajectories numerically. In particular, a grid that is initially uniform will in general become very irregular, often leading to a degradation of global accuracy. An alternative to avoid these disadvantages and other difficulties caused by the presence of convective terms in (1.1) is the Modified Method of Characteristics (MMC) or Semi-Lagrangian as it is known in meteorological community. The MMC has been developed to produce numerical methods that preserve the advantages of regular grids while simultaneously taking the advantage of the Lagrangian form of the equations. There is an extensive literature describing these methods. In particular, Staniforth *et al* [33] provide an excellent review of the application of the MMC to meteorological problems, Pironneau [24] and Süli [34] studied the application of the MMC to Navier-Stokes equations, Pudykiewicz *et al* [25] used the MMC to solve some ocean models, and Douglas and Russel [10] and Bermejo [3] formulated almost simultaneously the idea of combining the MMC and time implicit methods to construct an algorithm to solve the linear transport-diffusion equations.

In order to explain the MMC in more detail, we consider first the homogeneous advection equation (*i.e.* $\mathbf{D}(\mathbf{x}) \equiv 0$ and $\mathbf{f}(\mathbf{u}, \mathbf{x}, t) = 0$):

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{u} &= 0, \\ \mathbf{u}(\mathbf{x}, 0) &= u_0(\mathbf{x}). \end{aligned} \quad (2.1)$$

We introduce the total derivative of the function $\mathbf{w}$ as

$$\frac{D\mathbf{w}}{Dt} := \frac{\partial \mathbf{w}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{w}. \quad (2.2)$$

Recall that $\frac{D\mathbf{w}}{Dt}$ measures the rate of change in $\mathbf{w}$ following the trajectories of the fluid particles. The fundamental idea of the MMC is to impose a regular grid at a new time level and backtrack the fluid trajectories to the previous time level. At the old time level, the required quantities are evaluated by interpolation from their known values on a regular grid. In general, the velocity field $\mathbf{v}$ at the new time step is unknown, so the critical problem in this idea is the computation of the trajectory departure points.

Let the time interval $[0, T]$ be divided into N subintervals $[t_n, t_{n+1}]$ of length τ such that $t_n = n\tau$ and $T = N\tau$. Following Robert [26] and Douglas *et al* [10], the characteristic curves of the equation (2.1) are the solution of the system of ordinary differential equations

$$\begin{aligned} \frac{d\mathcal{X}(\mathbf{x}, t_{n+1}; t)}{dt} &= \mathbf{v}(\mathcal{X}(\mathbf{x}, t_{n+1}; t), t), \\ \mathcal{X}(\mathbf{x}, t_{n+1}; t_{n+1}) &= \mathbf{x}. \end{aligned} \quad (2.3)$$

Note that $\mathcal{X}(\mathbf{x}, t_{n+1}; t)$ is the departure point at time t of a particle which will arrive at $\mathbf{x}$ at time t_{n+1} . The MMC does not follow the fluid particles forward in time, as the Lagrangian

schemes do, instead it traces backward the position at time t_n of particles which will reach the points of a fixed mesh at time t_{n+1} . In so doing, the MMC avoids the grid distortion difficulties that the conventional Lagrangian schemes have.

The solution of (2.3) can be expressed as

$$\mathcal{X}(\mathbf{x}, t_{n+1}; t_n) = \mathbf{x} - \int_{t_n}^{t_{n+1}} \mathbf{v}(\mathcal{X}(\mathbf{x}, t_{n+1}; t), t) dt. \quad (2.4)$$

Some authors estimate $\mathcal{X}(\mathbf{x}, t_{n+1}; t)$ by the second-order explicit Runge–Kutta scheme, which the authors of [34, 3] have found to be not accurate enough to maintain a particle on its curved trajectory. A better choice is the fourth-order explicit Runge–Kutta scheme. However, it is inconvenient because the velocity should be extrapolated at both, $t_{n+\frac{1}{2}}$ and t_{n+1} . In this paper, we use the method first proposed by Temperton and Staniforth [36] in the context of Semi-Lagrangian schemes to integrate the weather prediction equations. Then, to compute $\mathcal{X}(\mathbf{x}, t_{n+1}; t_n)$ associated with the point $\mathbf{x}$, we define

$$\mathbf{a} := \mathbf{x} - \mathcal{X}(\mathbf{x}, t_{n+1}; t_n).$$

If we use the midpoint rule to approximate the integral in (2.4), we have

$$\mathbf{a} = \tau \mathbf{v}\left(\mathcal{X}(\mathbf{x}, t_{n+1}; t_{n+\frac{1}{2}}), t_{n+\frac{1}{2}}\right) + \mathcal{O}(\tau^3). \quad (2.5)$$

Using

$$\mathcal{X}(\mathbf{x}, t_{n+1}; t_{n+\frac{1}{2}}) = \mathbf{x} - \frac{1}{2}\mathbf{a} + \mathcal{O}(\tau^2),$$

and

$$\mathbf{v}\left(\mathcal{X}(\mathbf{x}, t_{n+1}; t_{n+\frac{1}{2}}), t_{n+\frac{1}{2}}\right) = \mathbf{v}\left(\mathbf{x} - \frac{1}{2}\mathbf{a}, t_{n+\frac{1}{2}}\right) + \mathcal{O}(\tau^2),$$

we write

$$\mathbf{a} = \tau \mathbf{v}\left(\mathbf{x} - \frac{1}{2}\mathbf{a}, t_{n+\frac{1}{2}}\right) + \mathcal{O}(\tau^3).$$

The velocity at $t_{n+\frac{1}{2}}$ can be defined by extrapolation from the two previous time levels by the formula

$$\mathbf{v}(\mathbf{x}, t_{n+\frac{1}{2}}) = \frac{3}{2}\mathbf{v}(\mathbf{x}, t_n) - \frac{1}{2}\mathbf{v}(\mathbf{x}, t_{n-1}) + \mathcal{O}(\tau^2)$$

recommended by Temperton and Staniforth [36], together with the following successive iterations to compute $\mathbf{a}$:

$$\begin{aligned} \mathbf{a}^{(0)} &= \tau \left[\frac{3}{2}\mathbf{v}(\mathbf{x}, t_n) - \frac{1}{2}\mathbf{v}(\mathbf{x}, t_{n-1}) \right], \\ \mathbf{a}^{(m)} &= \tau \left[\frac{3}{2}\mathbf{v}\left(\mathbf{x} - \frac{1}{2}\mathbf{a}^{(m-1)}, t_n\right) - \frac{1}{2}\mathbf{v}\left(\mathbf{x} - \frac{1}{2}\mathbf{a}^{(m-1)}, t_{n-1}\right) \right], \quad m = 1, 2, \dots \end{aligned}$$

It is also known [25] that

$$\|\mathbf{a} - \mathbf{a}^{(m+1)}\|_\infty \leq \frac{1}{4}\sigma\tau\|\mathbf{a} - \mathbf{a}^{(m)}\|_\infty, \quad m = 0, 1, \dots, \quad (2.6)$$

where $\|\cdot\|_\infty$ is the maximum norm and $\sigma = \|\nabla \mathbf{v}\|_\infty$. Hence, if $\sigma\tau$ is sufficiently small, we can conclude from (2.6) that a few iterations (3 or 4) are enough to approximate $\mathbf{a}$ up to $\mathcal{O}(\tau^3)$. A schematic representation of the quantities involved in computing the departure

Figure 1. A schematic diagram showing the main quantities used in the calculation of the departure points. The exact trajectory is represented by a solid line and the approximate trajectory with a dashed line.

points, in the one-dimensional case, is shown in Figure 1. Once $\mathbf{a}$ are known, the departure point values of the variables in our equations are defined as illustrated by

$$\mathbf{w}^*(\mathbf{x}, t_n) := \mathbf{w}(\mathcal{X}(\mathbf{x}, t_{n+1}; t_n), t_n), \quad (2.7)$$

and the usual solution of (2.1) by the MMC at instant t_n is simply

$$\mathbf{u}(\mathbf{x}, t_{n+1}) = \mathbf{u}(\mathcal{X}(\mathbf{x}, t_{n+1}; t_n), t_n) := \mathbf{u}^*(\mathbf{x}, t_n). \quad (2.8)$$

In general, $\mathcal{X}(\mathbf{x}, t_{n+1}; t_n)$ will not coincide with the spatial position of the grid point. The requirement is then that the scheme for computing $\mathcal{X}(\mathbf{x}, t_{n+1}; t_n)$ should be provided with a search-locate algorithm to find the host element where such a point is located. A general, efficient and easy-to-implement scheme for performing this step in arbitrary grids is presented in [1].

Assuming that a suitable approximation is made for $\mathbf{a}$, $\mathcal{X}$ will not generally lie at a grid point, so the velocity $\mathbf{v}$ and function $\mathbf{w}^*$ in (2.7) must be obtained by interpolation from the known values at the grid points. Thus interpolation plays a very important role and the accuracy of the interpolation method used has direct consequences for the accuracy of the MMC. The interpolation procedure used by us in this paper is the Biquadratic Lagrange interpolation previously used by other researchers, [36, 6]. However, a good reference to other interpolation procedures for the MMC can be found in [39, 11].

Naturally, the computed functions $\mathbf{u}$ in (2.1) should be nonnegative. Small negative values may cause large errors, because the physical equilibrium is disturbed. Most interpolation methods however do not preserve positivity, in which case after each interpolation step a certain filtering procedure should be carried out to make the function $\mathbf{u}$ nonnegative, for instance, with the use of the modified Ulrich–Ludes filter [19] and the Bartnicki filter [2]. Another way to prevent negative values is the use of the so-called monotone interpolation [6].

This implies that a value obtained by interpolating in a grid cell, lies between the maximum and minimum values in the vertices of this grid cell. In this case, only nonnegative values can arise, since the initial condition field is nonnegative, and we obtain a non-oscillatory MMC algorithm that possesses good preservation of the advected field shape in the vicinity of strong gradients and maintains the order of convergence in regions where the solution is sufficiently smooth. This method was first introduced by Bermejo and Staniforth [6] to convert the MMC to a quasi-monotone method, while a detailed analysis for the new method has been recently made in [4].

The resulting Quasi-monotone Modified Method of Characteristics (QMMC) can be realized by the simple four-step marching procedure:

Algorithm 1. The QMMC algorithm

Step 1. Compute the departure point $\mathcal{X}(\mathbf{x}, t_{n+1}; t_n)$ and identify the element of the grid where such a point is located.

Step 2. Compute the approximation $\mathbf{u}^*(\mathbf{x}, t_{n+1}) := \mathbf{u}(\mathcal{X}(\mathbf{x}, t_{n+1}; t_n), t_{n+1})$ employing the Biquadratic Lagrange interpolation.

Step 3. Evaluate $\mathbf{u}^-$ and $\mathbf{u}^+$ as the minimum and maximum value in the vertices of the grid cell, respectively.

Step 4. Set

$$\begin{aligned} \mathbf{u}(\mathbf{x}, t_{n+1}) &= \mathbf{u}^-, & \text{if } \mathbf{u}^*(\mathbf{x}, t_{n+1}) < \mathbf{u}^-, \\ \mathbf{u}(\mathbf{x}, t_{n+1}) &= \mathbf{u}^+, & \text{if } \mathbf{u}^*(\mathbf{x}, t_{n+1}) > \mathbf{u}^+, \\ \mathbf{u}(\mathbf{x}, t_{n+1}) &= \mathbf{u}^*(\mathbf{x}, t_{n+1}), & \text{otherwise.} \end{aligned}$$

3. Application to Transport-Diffusion-Reaction Problems

The QMMC can be used in combination with finite elements, finite differences or spectral methods. In particular, the application of the MMC within the framework of finite elements is known as the Characteristic–Galerkin method. The first application to transport-diffusion problems is due to Douglas and Russel [10], who combined the finite element and finite difference methods with the method of characteristics to treat the one-dimensional form of problem (1.1). Since the MMC takes care of the convective terms in (1.1), we focus on the reaction-diffusion terms. In this section we consider two methods to solve the reaction-diffusion problem. Both methods are second-order in time, but the first one is implicit and the second one is explicit.

3.1. Space semi-discretization

For the sake of simplicity in the presentation we shall use the central finite difference method for the space discretization. Define a regular grid over Ω with grid points $\mathbf{x}_h$ for spatial scale h . We use the notation $\mathbf{w}_n(\mathbf{x}) := \mathbf{w}(\mathbf{x}, t_n)$, $W_h(t) := \mathbf{w}(\mathbf{x}_h, t)$, and $W_{h,n} := \mathbf{w}(\mathbf{x}_h, t_n)$ unless otherwise specified. According to Douglas and Russel [10], the physical time term and the convection terms in (1.1) are combined to give the characteristic direction as follows.

Let

$$\psi_h(t) = \left[1 + |\mathbf{v}(\mathcal{X}_h(\mathbf{x}_h, t_{n+1}; t), t)|^2 \right]^{\frac{1}{2}}. \quad (3.1)$$

The derivative along the characteristic direction $s_h = s_h(t)$ is given by

$$\frac{\partial}{\partial s_h} = \frac{1}{\psi_h(t)} \frac{D}{Dt}. \quad (3.2)$$

Using (3.1) and (3.2), the semidiscretization version of the equation (1.1) can be rewritten as

$$\psi_h(t) \frac{dU_h}{ds} - A_h U_h = F_h(U_h, t), \quad (3.3)$$

where A_h denotes the central finite difference discretization of the diffusion operator $\mathbf{A}$ given by (1.2).

Since

$$ds = [dt^2 + d\mathcal{X}^2]^{\frac{1}{2}} = \psi_h(t) dt,$$

we approximate (3.3) by the semi-discrete system of ordinary differential equations

$$\begin{aligned} \frac{dU_h}{dt} - A_h U_h &= F_h(U_h, t), \\ U_h(0) &= \mathbf{u}_0(\mathbf{x}_h). \end{aligned} \quad (3.4)$$

The computation of the total derivative by the MMC leads to the solution of the difference reaction-diffusion problem (3.4). In this section we consider two procedures to deal with problem (3.4).

3.2. Implicit characteristic Crank–Nicholson scheme

Once $\mathcal{X}_h(\mathbf{x}_h, t_{n+1}; t_n)$ is known, the departure point values of the variables in our equations are defined as illustrated by

$$W_{h,n}^* := \mathbf{w}(\mathcal{X}_h(\mathbf{x}_h, t_{n+1}; t_n), t_n).$$

Thus, the complete discretization of (1.1) along the trajectories $\mathcal{X}_h(\mathbf{x}, t_{n+1}; t)$ in each interval $[t_n, t_{n+1}]$ can be formulated as

$$\begin{aligned} \frac{U_{h,n+1} - U_{h,n}^*}{\tau} - (1 - \theta) A_h U_{h,n+1} - \theta A_h U_{h,n}^* &= \\ (1 - \theta) F_{h,n+1}(U_{h,n+1}) + \theta F_{h,n}(U_{h,n}^*), \quad U_{h,0} &= \mathbf{u}_0(\mathbf{x}_h). \end{aligned} \quad (3.5)$$

The parameter θ determines the time stepping scheme, for example, at $\theta = 0$ equations (3.5) are the first-order Characteristic Backward Euler scheme (CBE), while at $\theta = \frac{1}{2}$ equations (3.5) represent the second-order Characteristic Crank–Nicholson scheme (CCN). All these schemes have already been used by other authors for numerical solution of (1.1). For instance, the CBE scheme is formulated and analyzed in [24, 13] and the CCN one is proposed in [3, 25].

Note that the CCN scheme is unconditionally stable for linear problems, so that the choice of τ may be based on accuracy considerations. To find the solution U_{n+1} from (3.5), one has to solve, at each time level, a linear (nonlinear) system of algebraic equations.

The CCN algorithm for solving the linear case of (1.1) is executed in the following steps. Given $U_{h,n}$, $U_{h,n+1}$ is computed by

Algorithm 2. The CCN algorithm

Step 1 – Step 4 are similar to the QMMC Algorithm for obtaining the solution

$$\hat{U}_{h,n} := \mathbf{u}(\mathbf{x}_h, t_{n+1}).$$

Step 5. Compute the diffusion term $A_h \hat{U}_{h,n}$ by the central finite difference method.

Step 6. Solve the linear system of algebraic equations to obtain the solution $U_{h,n+1}$ from (3.5).

3.3. The case of nonlinear reactive terms

In this section we formulate the CCN scheme within the monotone iterative framework to deal with the nonlinear reaction terms $\mathbf{f}$ in (1.1). Note that (3.5) is a sequence of generalized stationary reaction-diffusion problems. Therefore, to solve (1.1) using the CCN scheme, one has to solve a diffusion problem at each time step. To do this, we propose the monotone iterative method frequently used by Pao [22] for nonlinear parabolic equations. The scheme we present here is an adaptation of this method to our reformulation of the CCN scheme. It consists of the following Picard iterations:

$$\begin{aligned} U_{h,0} &= \mathbf{u}_0(x_h), \quad U_{h,n}^{(0)} = U_{h,n-1}, \quad U_{h,n}^{*(0)} = U_{n-1}(\mathcal{X}_h(x_h, t_{n+1}; t_n), t_n), \\ \frac{U_{h,n+1}^{(m)} - U_{h,n}^{*(m)}}{\tau} - (1 - \theta)A_h U_{h,n+1}^{(m)} - \theta A_h U_{h,n}^{*(m)} + \gamma_n U_{h,n+1}^{(m)} &= \gamma_n U_{h,n+1}^{(m-1)} + \\ (1 - \theta)f_{h,n+1}(U_{h,n+1}^{(m-1)}) + \theta f_{h,n}(U_{h,n}^{*(m-1)}), \quad m &= 1, 2, \dots, \end{aligned} \quad (3.6)$$

where γ_n is a nonnegative monotone term chosen at each instant t_n such that

$$\gamma_n = \max_h \left\{ -\frac{\partial f}{\partial u}(U_{h,n}^*) \right\}. \quad (3.7)$$

Then the CCN algorithm for (1.1) is reformulated as shown in Algorithm 3.

The aim of adding the term γ_n in (3.6) is to preserve the monotonicity of the iterating solution $U_{h,n+1}^{(m)}$, and therefore to ensure the fast convergence of the iterative process. We refer to [22] for more details.

3.4. Explicit characteristic Runge–Kutta scheme

To formulate the Characteristic Runge–Kutta (CRK) scheme for the approximation of (1.1), we briefly recall the conventional explicit, optimal second-order Runge–Kutta method used to integrate the system of ordinary differential equations

$$\begin{aligned} \frac{d\mathbf{w}}{dt} &= \mathbf{g}(t, \mathbf{w}), \quad t \in (0, T], \\ \mathbf{w}(0) &= \mathbf{w}_0, \end{aligned} \quad (3.8)$$

where $\mathbf{w}(t) = (w_1(t), \dots, w_N(t))^T$ and $\mathbf{g}(t, \mathbf{w}) = (g_1(t, \mathbf{w}), \dots, g_N(t, \mathbf{w}))^T$. According, for example, to the textbook [16], we formulate the second-order Runge–Kutta scheme to solve

Algorithm 3. The CCN algorithm for the nonlinear case

Step 1 – Step 4 are similar to the QMMC Algorithm for obtaining the solution $\hat{U}_{h,n} := \mathbf{u}(\mathbf{x}_h, t_{n+1})$.

Step 5. Compute the monotone terms γ_n in (3.7).

Step 6. Set $U_{h,n}^{(0)} = U_{h,n-1}$ and $\bar{U}_{h,n}^{(0)} = \hat{U}_{h,n-1}$. For $m = 1, 2, \dots$

- Set $\bar{U}_{h,n}^{(m)} = \hat{U}_{h,n-1}$
- Compute the diffusion term of (3.6) $\mathcal{D}_h = (I_h - (1-\theta)\tau A_h + \tau\gamma_n I_h)$, A_h is the central finite difference of (1.2) and I_h is the identity matrix
- Compute the right-hand side $\mathcal{B}_{h,n+1}$ of (3.6) as

$$\mathcal{B}_h = \bar{U}_{h,n}^{(m)} + \theta\tau A_h \bar{U}_{h,n}^{(m)} + \tau\gamma_n U_{h,n+1}^{(m-1)} + (1-\theta)\tau f_{h,n+1}(U_{h,n+1}^{(m-1)}) + \theta\tau f_h(\bar{U}_{h,n}^{(m)})$$

- Compute $U_{h,n+1}^{(m)}$ by solving the linear system $\mathcal{D}_h U_{h,n+1}^{(m)} = \mathcal{B}_h$
- The iterative process is terminated as soon as the inequality

$$\frac{\|\mathbf{U}_{n+1}^{(m)} - \mathbf{U}_{n+1}^{(m-1)}\|_2}{\|\mathbf{U}_{n+1}^{(m)}\|_2} \leq \text{tol}$$

is satisfied. Here $\|\cdot\|_2$ denotes the discrete L^2 -norm and “tol” is the given tolerance.

(3.8) as

$$\begin{aligned} \mathbf{K}_1 &= \tau \mathbf{g}(t_n, \mathbf{w}_n), \\ \mathbf{K}_2 &= \tau \mathbf{g}(t_n + \frac{\tau}{2}, \mathbf{w}_n + \frac{1}{2}\mathbf{K}_1), \\ \mathbf{w}_{n+1} &= \mathbf{w}_n + \mathbf{K}_2. \end{aligned}$$

If we identify the entries $w_i(t)$ and $g_i(t, \mathbf{w})$ in (3.8) with U_h and $A_h U_h + F_h(U_h, t)$ of (3.4), respectively, then we can formulate the CRK scheme to integrate (1.1) as follows. Given $U_{h,n}$, $U_{h,n+1}$ is computed by Algorithm 4.

Algorithm 4. The CRK algorithm

Step 1 – Step 4 are similar to the QMMC Algorithm for obtaining the solution $\hat{U}_{h,n} := \mathbf{u}(\mathbf{x}_h, t_{n+1})$.

Step 5. Compute $K_{1h} = \tau A_h \hat{U}_{h,n} + \tau F_{h,n}(\hat{U}_{h,n})$.

Set $\bar{U}_{h,n} = \hat{U}_{h,n} + \frac{1}{2}K_{1h}$ and evaluate $K_{2h} = \tau A_h \bar{U}_{h,n} + \tau F_{h,n+\frac{1}{2}}(\bar{U}_{h,n})$.

Step 6. Then, compute $U_{h,n+1} = \hat{U}_{h,n} + K_{2h}$.

Because the CRK scheme explicitly evaluates the right-hand side of system (3.4), it should satisfy the stability criterion. This stability criterion can be attained based on the

McDonald's analysis [20] for the linear case. Hence, the CRK scheme is stable if

$$\frac{\bar{d}\tau}{h^2} \leq \frac{1}{2^p}, \quad (3.9)$$

where $\bar{d} = \max_{ij} d_{ij}$ and p is the dimension of the space domain Ω .

Note that (3.9) is the CFL stability condition for the transport-diffusion problem (1.1). On the other hand, when the reaction terms $\mathbf{f}$ in (1.1) are nonlinear we can prove, on the basis of the results of [27], that the CRK scheme is stable under the restriction

$$\tau \leq \frac{h^2}{Lh^2 + 2^p \bar{d}}, \quad (3.10)$$

where L is the maximum of the Jacobian $\left| \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right|$.

4. New Characteristic Runge–Kutta–Chebyshev Scheme

Using the CCN scheme to integrate equations (1.1), either nonlinear or linear algebraic equations should be solved at every time step of the computational process. The finding of solutions to such systems requires much computation. On the other hand, in integrating equation (1.1) with the use of the CRK scheme, the algebraic problem does not arise, however, the time step should be very small because of the stability constraints (3.9) or (3.10).

When the size of the problem is too large the CPU time and the number of evaluations of the associated right-hand side become critical and may limit the resolution of the numerical model. In this paper, we present an alternative approach that relies on the idea of the MMC that is explicit but unconditionally stable and second-order in τ . To be more specific, we combine the explicit Runge–Kutta–Chebyshev (RKC) method proposed by Van der Houwen *et al* [38, 37] with the MMC to solve (1.1). The RKC method is an s -stage Runge–Kutta method designed for the explicit integration of systems of the form of (3.4). The method possesses an extended real stability interval with a length $\beta(s)$ proportional to s^2 and can be used with arbitrarily large s .

4.1. Background of the RKC method

The RKC formulas are given in [40], and a comprehensive linear stability and convergence analysis of these formulas is found in [41]. The formulas are also studied in the review article [31] along with the Fortran RKC code.

According to the ordinary differential system (3.8), the RKC formulas are of the form

$$\begin{aligned} \mathbf{Y}_0 &= \mathbf{w}_n, \\ \mathbf{Y}_1 &= \mathbf{Y}_0 + \tilde{\mu}_1 \mathbf{G}_0, \\ \mathbf{Y}_j &= \mu_j \mathbf{Y}_{j-1} + \nu_j \mathbf{Y}_{j-2} + (1 - \mu_j - \nu_j) \mathbf{Y}_0 + \tilde{\mu}_j \mathbf{G}_{j-1} + \tilde{\gamma}_j \mathbf{G}_0, \quad 2 \leq j \leq s, \\ \mathbf{w}_{n+1} &= \mathbf{Y}_s, \quad n = 0, 1, \dots \end{aligned} \quad (4.1)$$

All the coefficients are available in analytical form for arbitrary $s \geq 0$. They are defined as follows.

Consider the Chebyshev polynomial of the first kind of degree j

$$T_j(z) = \cos(j \arccos z), \quad -1 \leq z \leq 1.$$

Then

$$\begin{aligned} \epsilon &= \frac{2}{13}, \quad q_0 = 1 + \frac{\epsilon}{s^2}, \quad q_1 = \frac{T'_s(q_0)}{T''_s(q_0)}, \\ b_j &= \frac{T''_s(q_0)}{(T'_s(q_0))^2}, \quad (2 \leq j \leq s), \quad b_0 = b_2, \quad b_1 = b_2, \end{aligned}$$

and

$$\begin{aligned} \tilde{\mu}_1 &= b_1 q_1, \quad \mu_j = \frac{2b_j q_0}{b_{j-1}}, \quad \nu_j = \frac{-b_j}{b_j - 2}, \quad \tilde{\mu}_j = \frac{2b_j q_1}{b_{j-1}}, \\ \tilde{\gamma}_j &= (1 - b_{j-1} T_{j-1}(q_0)) \tilde{\mu}_j, \quad (2 \leq j \leq s). \end{aligned}$$

In (4.1), the stage $\mathbf{G}_j = \mathbf{g}(t_n + c_j \tau, Y_j)$. The c_j value is

$$c_j = \frac{T'_s(q_0)}{T''_s(q_0)} \frac{T''_s(q_0)}{T'_s(q_0)} \approx \frac{j^2 - 1}{s^2 - 1} \quad (2 \leq j \leq s), \quad c_1 = \frac{c_2}{T'_2(q_0)} \approx \frac{c^2}{4}, \quad c_s = 1.$$

For the reaction-diffusion problems (i.e. $\mathbf{v} = 0$) with a small diffusion the stiffness is modest, and the RKC method offers a very attractive alternative to standard explicit integration schemes and unconditionally stable, implicit ones. For numerical illustrations see, for instance, [41, 5].

4.2. Formulation and implementation

We are now in a position to formulate our new Characteristic Runge–Kutta–Chebyshev (CRKC) scheme based on the combination of the QMMC scheme for the convective term and the RKC scheme for the diffusive-reactive term, as we did with the CCN and CRK schemes in the above sections. The implementation of the CRKC scheme to approximate the solution of (1.1) is carried out in the following steps. Given $U_{h,n}$, $U_{h,n+1}$ is computed by Algorithm 5.

It is worth noting that the number of stages s in the CRKC and conventional RKC schemes varies with τ such that (see e.g. [31])

$$s = 1 + \text{int} \left[\left(1 + \frac{8\bar{d}\tau}{0.65h^2} \right)^{\frac{1}{2}} \right]. \quad (4.2)$$

However, the CRKC schemes differs from the RKC one in that the CRKC scheme is applied during each subinterval $[t_n, t_{n+1}]$ along the characteristic curves rather than in the time direction.

5. Numerical Examples

To illustrate the performance and accuracy of our CRKC scheme introduced in the previous section and to compare the CRKC results with those obtained by the CCN and CRK schemes, we have run a series of numerical tests for the transport-diffusion-reaction equation (1.1). The first series of numerical examples consists of two-dimensional passive advection-diffusion

Algorithm 5. The CRKC algorithm

Step 1 – Step 4 are similar as the QMMC Algorithm for obtaining the solution $\hat{U}_{h,n} := \mathbf{u}(\mathbf{x}_h, t_{n+1})$.

Step 5. Compute

$$\begin{aligned} Y_{0,h} &= \hat{U}_{h,n}, \\ Y_{1,h} &= Y_{0,h} + \tilde{\mu}_1 \tau A_h \hat{U}_{h,n} + \tilde{\mu}_1 \tau F_{h,n}(\hat{U}_{h,n}). \end{aligned}$$

For $j = 2, 3, \dots, s$ evaluate

$$\begin{aligned} Y_{j,h} &= (1 - \mu_j - \nu_j) Y_{0,h} + \mu_j Y_{j-1,h} + \nu_j Y_{j-2,h} + \tilde{\mu}_j \tau A_h Y_{j-1,h} + \\ &\quad \tilde{\mu}_j \tau F_{h,n+c_j}(Y_{j-1,h}) + \tilde{\gamma}_j \tau A_h \hat{U}_{h,n} + \tilde{\gamma}_j \tau F_{h,n}(\hat{U}_{h,n}). \end{aligned}$$

Step 6. Then, set $U_{h,n+1} = Y_{s,h}$.

of some initial conditions with known analytical solutions, so that we can evaluate the relative error function **Err** as

$$\text{Err}_h = U_{h,n} - u(\mathbf{x}_h, t_n), \quad (5.1)$$

where $u(\mathbf{x}_h, t_n)$ and $U_{h,n}$ are the exact and numerical solutions, respectively, at grid points $\mathbf{x}_h$ and time t_n . The following error norms are defined:

$$\|\mathbf{w}\|_\infty = \max_h |w_h|, \quad \|\mathbf{w}\|_2 = h \left(\sum_h |w_h|^2 \right)^{\frac{1}{2}}, \quad \text{Mass}(\mathbf{w}) = h^2 \sum_h w_h.$$

These error norms, along with the elevation of the contour plots, give a good idea of the accuracy of the procedure. The efficiency of the solvers is compared in the CPU time context on a PC with AMD-K6 200 processor. Furthermore, the algebraic system of linear equations resulting in the CCN scheme were solved using the Preconditioned Conjugate Gradient (PCG). We use the diagonal as a preconditioner and a tolerance of 10^{-6} to stop the iterations.

5.1. The cylinder rotation test

The purpose of this first test is to compare the QMMC results with those computed with the standard MMC. This test consists of one or more rotations of a cylinder in the plane, which can mathematically be formulated by

$$\frac{\partial u}{\partial t} + v_1 \frac{\partial u}{\partial x} + v_2 \frac{\partial u}{\partial y} = 0, \quad \text{in } \Omega \times (0, T],$$

where $\Omega = [0, 1] \times [0, 1]$ covered by uniformly spaced grids with $h = \frac{1}{80}$, the velocities are

$$v_1 = 2\pi(y - \frac{1}{2}), \quad v_2 = -2\pi(x - \frac{1}{2}), \quad (5.2)$$

The analytical solution of this problem is given as

$$u(x, y, t) = \begin{cases} 1, & \text{if } (x - r(t))^2 + (y - s(t))^2 \leq 0.15^2 \\ 0, & \text{otherwise,} \end{cases}$$

where $r(t) = \frac{2+\sin(2\pi t)}{4}$ and $s(t) = \frac{2+\cos(2\pi t)}{4}$.

From (5.2) it follows that the characteristics are circles in the xy -plane with the centre in $(\frac{1}{2}, \frac{1}{2})$. The CFL condition used is 3.5. From the expressions

$$CFL_x = \max |v_1| \frac{\tau}{h}, \quad CFL_y = \max |v_2| \frac{\tau}{h}, \quad CFL = \sqrt{CFL_x^2 + CFL_y^2}, \quad (5.3)$$

it follows that for this example the CFL condition is given as $CFL = 80\pi\sqrt{2}\tau$.

Figure 2 reports the results obtained by the MMC and QMMC schemes after one rotation. We see that the MMC scheme exhibits nonphysical oscillations and substantially greater distortions, especially at the foot and the upper face of the cylinder where the gradient is sharper. From the same figure we see that these oscillations are completely absent from the QMMC results. The cross-section at $y = x + 10h$ of the solutions after one and three rotations in Figure 3 clearly points to these facts.

Table 1 gives the maximum and minimum error norms for the numerical solutions. On terms of $\|\cdot\|_\infty$ and $\|\cdot\|_2$ the QMMC results are computed more accurately. From the values of “max” and “min” in Table 1, as well as from Figure 3 we observe high and negative values for the MMC that are absent from the QMMC results. Concerning the mass conservation, Table 1 shows that the MMC scheme has lost 13% of the initial mass after the three rotations whereas the QMMC scheme is mass conserving.

Table 1. Error norms for the cylinder rotation test after one and three rotations, $CFL = 3.5$

		$\ \mathbf{Err}\ _\infty$	$\ \mathbf{Err}\ _2$	Mass	min	max
1 Rot	Exact			1.000	0.000	0.070
	QMMC	0.525	0.0603	1.000	0.000	0.069
	MMC	0.621	0.0719	1.113	-0.090	0.000
3 Rot	Exact			0.070	0.000	1.000
	QMMC	0.631	0.078	0.069	0.000	1.000
	MMC	0.642	0.083	0.061	-1.106	1.127

Figure 2. Results for the cylinder rotation (left) and the corresponding contour plots (right) after one revolution for $CFL = 3.5$

Figure 3. Cross-section of the results in Figure 2 at $y = x + 10h$ after one revolution (left) and three revolutions (right)

5.2. Gaussian hill test

Our second example is the diffusion of the Gaussian hill in a uniformly rotating flow field. This problem was presented in [33]. The problem statement is given by

$$\begin{aligned} \frac{\partial u}{\partial t} + v_1 \frac{\partial u}{\partial x} + v_2 \frac{\partial u}{\partial y} - \nu \Delta u &= 0, \quad \text{in } \Omega \times (0, T], \\ u(x, y, 0) &= 100 \exp\left(-\frac{(x - x_0)^2 + (y - y_0)^2}{2\sigma^2}\right), \quad \text{in } \Omega, \end{aligned}$$

where Ω is a $3200km \times 3200km$ domain, $v_1 = -\omega y$ and $v_2 = \omega x$, with $\omega = 10^{-5}/s$ being the angular velocity. To obtain a well-defined problem, the initial and boundary conditions should be consistent with the following analytical solution:

$$u(x, y, t) = \frac{100}{1 + \frac{2\nu t}{\sigma^2}} \exp\left(-\frac{\bar{x}^2 + \bar{y}^2}{2(\sigma^2 + 2\nu t)}\right),$$

where $\bar{x} = x - x_0 \cos \omega t + y_0 \sin \omega t$, $\bar{y} = y - x_0 \sin \omega t - y_0 \cos \omega t$, $(x_0, y_0) = (-800km, 0)$ is the centre of the initial Gaussian hill and $\sigma = 2 \cdot 10^4 km^2$. The parameters of this experiment are those used in [25]. Thus, the initial grid has 32×32 points with $h = 100km$ in the two directions, the diffusion coefficient ν is allowed to take the values $10^4 m^2/s$ and $5 \cdot 10^4 m^2/s$. The CFL numbers used are 2 and 10. From expressions (5.3) it follows that for this test the CFL number is given as $CFL = 16\sqrt{2}\omega\tau$.

We define the relative L^2 -norm and the relative Mass as

$$\frac{\|\mathbf{U} - \mathbf{u}\|_2}{\|\mathbf{u}\|_2} \quad \text{and} \quad \frac{\text{Mass}(\mathbf{U})}{\text{Mass}(\mathbf{u})},$$

respectively. Table 2 and Table 3 show these relative norms obtained by the three schemes for two different values of ν and CFL after one rotation. From both tables we see that all the schemes give roughly similar results with some differences in the maximum value of the numerical solutions, especially when $CFL = 10$. However, with increasing the time step τ

or decreasing the space step h , the CRKC results are slightly more accurate and efficient than the CRK or CCN results, respectively.

With increasing τ the CRK scheme becomes unstable, and with decreasing h the linear system solver in the CCN scheme becomes slow. For instance, for $CFL = 2$ the number of iterations in the PCG solver is 3 and the number of stages in the CRKC scheme is 2, whereas for $CFL = 10$ the number of iterations in the CCN scheme becomes 5 and the number of stages in the CRKC one is 3 for either value of ν .

Table 2. Relative error norms and maximum values for the Gaussian hill test for $\nu = 5 \cdot 10^4$ after one revolution

	Schemes	L ² -norm	Mass	max
$CFL = 2$	CCN	0.01236	0.987	20.853
	CRKC	0.01237	0.987	20.852
	CRK	0.01237	0.986	20.852
$CFL = 10$	CCN	0.03704	1.003	23.914
	CRKC	0.03689	1.003	23.972
	CRK	0.03679	1.001	23.998

Table 3. Relative error norms and maximum values for the Gaussian hill test for $\nu = 10^4$ after one revolution

	Schemes	L ² -norm	Mass	max
$CFL = 2$	CCN	0.1138732	0.933	37.329
	CRKC	0.1138736	0.933	37.329
	CRK	0.1138736	0.933	37.329
$CFL = 10$	CCN	0.0863497	0.996	53.906
	CRKC	0.0863285	0.996	53.911
	CRK	0.0863285	0.995	53.911

Figure 4 and Figure 5 show the analytical and numerical results obtained by the CRKC scheme together with the corresponding contour plots for $\nu = 10^4$ and $\nu = 5 \cdot 10^4$, respectively; the CFL is fixed at 10. The CCN and CRK results are not presented here because are overlapped with the CRKC results, as clearly seen in the cross-section at $y = x + 4h$ given in Figure 6 for all the schemes.

Figure 4. Analytical and numerical solutions obtained by the CRKC scheme for $\nu = 10^4$ and $CFL = 10$ after one rotation

Figure 5. Analytical and numerical solutions obtained by the CRKC scheme for $\nu = 5 \cdot 10^4$ and $CFL = 10$ after one rotation

Figure 6. Cross-section of the results obtained by the three schemes at $y = x + 4h$ after one revolution. $CFL = 10$

5.3. Two travel fronts test

This test problem is a convection-diffusion equation where the velocity field varies with time, while two separate fronts travel at different speeds and eventually coalesce into one front. The problem has been previously solved in [14, 28] by the finite element method. The problem we consider here is

$$\begin{aligned} \frac{\partial u}{\partial t} + v_1 \frac{\partial u}{\partial x} + v_2 \frac{\partial u}{\partial y} &= \nu \Delta u \quad \text{in } \Omega \times (0, T], \\ u(x, y, 0) &= v_1(x, 0)v_2(y, 0) \quad \text{in } \Omega, \\ u(x, y, t) &= v_1(x, t)v_2(y, t) \quad \text{on } \partial\Omega \times (0, T], \end{aligned} \quad (5.4)$$

where

$$v_i(\alpha, t) = \frac{0.1e^{-A} + 0.5e^{-B} + e^{-C}}{e^{-A} + e^{-B} + e^{-C}},$$

with $\alpha = x$ when $i = 1$ and $\alpha = y$ when $i = 2$, with $A = \frac{0.05}{\nu}(\alpha - 0.5 + 4.95t)$, $B = \frac{0.25}{\nu}(\alpha - 0.5 + 0.75t)$, $C = \frac{0.50}{\nu}(\alpha - 0.375)$, and $u(x, y, t) = v_1(x, t)v_2(y, t)$ is the analytical solution of (5.4). $\Omega = [0, 1] \times [0, 1]$ discretized into 50 grid points in each direction, the diffusion coefficient ν takes the values $4 \cdot 10^{-3}$ and $3 \cdot 10^{-4}$. The time step τ is fixed at 0.005.

In the first runs for this example, we fixed ν and computed the solutions using the three schemes for different times. Table 4 shows the L^2 -error norms of these runs together with the CPU time used by each scheme. As is seen, there are no excessive differences in terms of the L^2 -error norms. However, the computational cost as to the CPU time can give a relevant comparison between the three schemes. Moreover, as the time step τ was increased from 0.005 to 0.01, holding constant h with $\nu = 4 \cdot 10^{-3}$, the CRK scheme lost stability, while the CCN and CRKC schemes remain stable for all values of τ but with a difference on the efficiency. For example, for the latter values, the CPU time of the CCN scheme is about twice that of the CPU time of the CRKC one. This fact can be a disadvantage of

Figure 7. CRKC results for (5.4) at different times with $\nu = 4 \cdot 10^{-3}$ and $\tau = 0.005$

Figure 8. Principal diagonal cross-section for the three schemes at $t = 0.4$ with $\tau = 0.005$

the CCN scheme when we have to integrate the transport-diffusion problems for large time computation using refined grids.

In the case where $\tau = 0.005$ and $\nu = 4 \cdot 10^{-3}$, the number of stages in the CRKC scheme is 2. This number becomes 5 when τ is increased to 0.05 and the CRKC scheme consumes only 73% of the CPU time used by the CCN one. No oscillations are detected in the CRKC scheme for this case.

Figure 7 presents some results for the CRKC scheme with $\nu = 4 \cdot 10^{-3}$ and $\tau = 0.005$ at different time intervals. The CCN and CRK results are not given here because, at least for this test, they are the same. The cross-section along the principal diagonal for the three schemes is presented in Figure 8. As mentioned above, there is no difference between the schemes and they do not diffuse the fronts or give oscillations near the steep gradients.

Table 4. L²-norm for the two travel fronts test with $\tau = 0.005$. The number between brackets refers to CPU time (in seconds)

	Schemes	$t = 0.2$	$t = 0.3$	$t = 0.5$	$t = 0.6$
$\nu = 3 \cdot 10^{-4}$	CCN	0.0097 (6.4)	0.030 (9.5)	0.025 (15.7)	0.017 (18.7)
	CRKC	0.0099 (6.3)	0.031 (9.4)	0.026 (15.5)	0.018 (18.5)
	CRK	0.0099 (6.3)	0.031 (9.4)	0.026 (15.5)	0.018 (18.5)
$\nu = 4 \cdot 10^{-3}$	CCN	0.0031 (7.4)	0.004 (11.1)	0.006 (18.4)	0.007 (21.1)
	CRKC	0.0031 (7.3)	0.004 (10.8)	0.006 (17.6)	0.007 (20.9)
	CRK	0.0031 (7.3)	0.004 (10.8)	0.006 (17.6)	0.007 (20.8)

5.4. Brusselator with a diffusion-convection test

Our final test consists in adding the convection-diffusion term to the standard Brusselator problem. This nonphysical problem is much more difficult than the previous tests due to both the coupling equations through the reaction terms and the nonlinearity of these reaction terms. The problem without convection was solved in [41] using the conventional RKC scheme. The governing system is

$$\begin{aligned} \frac{\partial u}{\partial t} + \mathbf{w} \cdot \nabla u - \nu \Delta u &= 1 + u^2 v - 4.4u, \\ \frac{\partial v}{\partial t} + \mathbf{w} \cdot \nabla v - \nu \Delta v &= 3.4u - u^2 v, \end{aligned}$$

subject to the homogeneous Neumann boundary conditions and the following initial conditions:

$$u(x, y, 0) = 1.5 + y, \quad v(x, y, 0) = 1 + 5x.$$

The diffusion coefficients ν used are 10^{-4} and $2 \cdot 10^{-3}$ and the velocity field $\mathbf{w}$ is given by

$$\mathbf{w} = (1 - e^{-t}) \left(-\frac{\partial \psi}{\partial y}, \frac{\partial \psi}{\partial x} \right)^T, \quad \text{where } \psi(x, y) = \sin^2(\pi x) \sin^2(\pi y).$$

Figure 9. Evolution of the component v obtained by the CRKC scheme for $h = \frac{1}{64}$, $\nu = 0.002$ and $\tau = 0.01$

This problem was run on the unit square, with a uniform mesh of step h in all spatial directions.

Table 5. Results for the component u in the Brusselator at $t = 1$ with $\tau = 0.01$ and $\nu = 10^{-4}$

	Schemes	min	max	Mass	CPU(in seconds)
$h = \frac{1}{64}$	CCN	0.260	5.128	1.621	50.360
	CRKC	0.260	5.127	1.621	42.110
	CRK	0.260	5.126	1.621	42.000
$h = \frac{1}{128}$	CCN	0.260	5.117	1.585	253.669
	CRKC	0.260	5.115	1.585	207.819
	CRK	0.260	5.116	1.585	173.930
$h = \frac{1}{256}$	CCN	0.260	5.115	1.570	1141.46
	CRKC	0.260	5.113	1.570	902.320
	CRK	0.260	5.114	1.570	764.779

Table 6. Results for the component u in the Brusselator at $t = 1$ with $\tau = 0.01$ and $\nu = 2 \cdot 10^{-3}$

	Schemes	min	max	Mass	CPU(in seconds)
$h = \frac{1}{64}$	CCN	0.267	4.864	1.710	50.740
	CRKC	0.267	4.866	1.710	42.100
	CRK	0.267	4.863	1.710	41.970
$h = \frac{1}{128}$	CCN	0.267	4.864	1.684	256.790
	CRKC	0.267	4.863	1.684	175.750
	CRK	—	—	—	—
$h = \frac{1}{256}$	CCN	0.267	4.864	1.671	1195.239
	CRKC	0.267	4.863	1.671	789.019
	CRK	—	—	—	—

The purpose of this example is to test the efficiency of the schemes when the grids are refined and the time steps are fixed only for accuracy considerations. In such situations, the resulting linear system to be solve in the CCN scheme becomes very huge so that the scheme loses efficiency. On the other hand, refining the grids keeping constant the time step, the CRK scheme gives spurious solutions and becomes unstable at the end.

On the contrary, these difficulties cannot arise in the CRKC scheme since it is explicit and stable under the condition on the number of stages to use. These behaviours are displayed in Table 5 and Table 6 where the efficiency of the CRKC scheme can be examined in the CPU time and the accuracy for the maximum, minimum, and mass of the numerical solution for the component u . (In Table 6 “—” means instability of the CRK scheme). Needless to say that when $\nu = 2 \cdot 10^{-3}$, with decreasing space step h the number of stages s in the CRKC scheme increases, for instance, $s = 3$ when $h = \frac{1}{128}$ and $s = 5$ when $h = \frac{1}{256}$.

The iterations in the CCN scheme was stopped with a tolerance of 10^{-6} and for both values of ν in Table 5 and Table 6 the mean number of iterations m to reach this tolerance was 7. However, the major portion of the CPU time was used by the PCG solver depending on the mesh size in space h and the viscosity ν .

Figure 9 illustrates the evolution of the solution v obtained by the CRKC scheme on the time interval $[0, 4]$ for $h = \frac{1}{64}$ and $\tau = 0.01$. The CRKC scheme is very satisfactory for this nonlinear coupled problem.

6. Concluding Remarks

In conclusion note that the modified method of characteristics and the second-order explicit Runge–Kutta–Chebyshev scheme are accurately connected and turned into an efficient CRKC algorithm to resolve some difficulties in the numerical integration of the transport-diffusion problems with reactive sources. We have illustrated by a number of numerical examples the ability of the new CRKC scheme to perform very well in the presence of strong gradients and discontinuities without oscillations and numerical diffusion. We have also compared the results of some transport-diffusion problems obtained by the CCN and CRK schemes, which are very often used in transport-diffusion modelling, with the results computed by the CRKC scheme. The principal conclusions drawn through this comparison are the following:

- i. For transport-diffusion-reaction models with large diffusion (or viscosity) coefficients, the stability criterion (3.9) or (3.10), that the CRK scheme should satisfy, may sometimes require an unreasonably small step size τ and the CRK scheme becomes unable to approximate the numerical solution for a long-time computation, whereas the CRKC scheme can be a good choice for the numerical approximation of such a situation. Needless to say that with increasing τ the number of stages in the CRKC scheme increases so that it remains stable.
- ii. When the size of problem (1.1) is too large, particularly when the reaction terms $\mathbf{f}$ in (1.1) are coupled nonlinear functions, the CCN scheme becomes quite slow due to the CPU time and the storage for solving the associated linear system. By contrast, the CRKC scheme is a suitable candidate because it is both explicit and unconditionally stable.

Finally, all the schemes presented in this paper are second-order in time. However, using the idea of the DUMKA methods developed by Lebedev [17] and Medovikov [21], and following the arguments of section 4, it is possible to construct high-order characteristic explicit schemes with large stability intervals. Results on these schemes will be reported in the near future.

References

- [1] A. Allievi and R. Bermejo, *A generalized particle search-locate algorithm for arbitrary grids*, J. Comp. Physics, **132** (1997), pp. 157–166.
- [2] J. Bartnicki, *An efficient positive definite method for the numerical solution of the advection equation*, Tech. Rep. WP-86-35, International Institute for Applied Systems Analysis, A-2361, Laxenburg, Austria, 1986.
- [3] R. Bermejo, *A Galerkin-characteristic algorithm for transport-diffusion equations*, SAIM J. Numer. Anal., **32** (1995), pp. 425–454.
- [4] R. Bermejo, *Analysis of a class of quasi-monotone and conservative Semi-Lagrangian advection schemes*, Numer. Math., **87** (2001), pp. 579–623.
- [5] R. Bermejo and M. Seaid, *A comparative study of some explicit schemes for a nonlinear biochemical reaction-diffusion system*, in: *Numerical Methods for Differential Equations* (P. Oliveira, F. Oliveira, F. Patrício, J. Ferreira, and A. Araújo, eds.), Coimbra University, 1998, pp. 380–389.
- [6] R. Bermejo and A. Staniforth, *The conversion of Semi-Lagrangian advection schemes to quasi-monotone schemes*, Month. Wea. Rev., **120** (1992), pp. 2622–2632.
- [7] C. Carasso, P. Charrier, B. Hamouzet, and J. L. Joly, *Nonlinear Hyperbolic Problems*, vol. 1402 of *Lecture Notes in Mathematics*, 1989.
- [8] C. N. Dawson, *Godunov-mixed methods for advective flow problems in one space dimension*, SIAM J. Numer. Anal., **28** (1991), pp. 1282–1309.
- [9] J. Douglas, C. Huang, and F. Pereira, *The modified method of characteristics with adjusted advection*, Numer. Math., **83** (1999), pp. 353–369.
- [10] J. Douglas and T. F. Russell, *Numerical methods for convection dominated diffusion problems based on combining the method of characteristics with finite elements or finite differences*, SIAM J. Numer. Anal., **19** (1982), pp. 871–885.
- [11] J. K. Dukowicz and J. W. Kodis, *Accurate conservative remapping (rezoning) for arbitrary Lagrangian-Eulerian computations*, SAIM J. Sci. Stat. Comput., **8** (1987), pp. 172–215.
- [12] How, R. Berkowicz, A. Eliassen, and A. Prahm, *Comparison of numerical techniques for use in air pollution models with nonlinear chemical reactions*, Atmospheric Environment, **23** (1991), pp. 967–983.
- [13] G. Jakimow, E. Yakimiw, and A. Robert, *An implicit formulation for the horizontal diffusion in grid-point models*, Mon. Wea. Rev., **120** (1992), pp. 124–130.
- [14] S. V. Krisnamachari, L. J. Hayes, and T. F. Russel, *A finite element alternating-direction method combined with a modified method of characteristics for convection-diffusion problems*, SIAM J. Numer. Anal., **26** (1989), pp. 1462–1473.
- [15] O. Ladyzhenskaya, *The Mathematical Analysis of the Navier-Stokes Equations*, North-Holland, Amsterdam, 1985.
- [16] J. D. Lambert, *Numerical Methods for Ordinary Differential Systems*, John Wiley & Sons, New-York, 1991.
- [17] V. I. Lebedev, *How to solve stiff systems of differential equations by explicit methods*, in: *Numerical Methods and Applications* (G. I. Marchuk, ed.), 1994, pp. 45–80.
- [18] J. LeVeque Randall, *Numerical Methods for Conservation Laws*, Lectures in Mathematics ETH Zürich, 2nd edn., 1992.
- [19] G. Ludes and W. Ulrich, *Numerical simulation of plume-advection over a hill with coupled models. Part I: Numerical solution of the transport equation*, Meteor. Atmos. Phys., **39** (1988), pp. 97–107.
- [20] A. McDonald, *Accuracy of multiple-upstream Semi-Lagrangian advective schemes*, Mon. Wea. Rev., **112** (1984), pp. 1267–1275.
- [21] A. A. Medovikov, *High order explicit methods for parabolic equations*, BIT, **38** (1998), pp. 372–390.

- [22] C. V. Pao, *Numerical methods for coupled systems of nonlinear parabolic boundary value problems*, J. Math. Anal., **151** (1990), pp. 581–608.
- [23] J. Pedlosky, *Geophysical Fluid Dynamics*, 2nd edn., Springer-Verlag, 1987.
- [24] O. Pironneau, *On the transport-diffusion algorithm and its applications to the Navier-Stokes equations*, Numer. Math., **38** (1982), pp. 309–332.
- [25] J. Pudykiewicz and A. Staniforth, *Some properties and comparative performance of the Semi-Lagrangian method of Robert in the solution of advection-diffusion equation*, Atmos. Ocean, **22** (1984), pp. 283–308.
- [26] A. Robert, *A stable numerical integration scheme for the primitive meteorological equations*, Atmos. Ocean, **19** (1981), pp. 35–46.
- [27] J. M. Sanz-Serna and A. M. Stuart, *A note uniform in time error estimates for approximations to reaction-diffusion equations*, IMA J. Numer. Anal., **12** (1992), pp. 457–462.
- [28] M. Seaïd and M. El-Amrani, *New characteristic-finite element explicit schemes for the incompressible Navier-Stokes equations*, in: *Advanced Computational Methods in Engineering* (R. Van Keer, B. Verbeeghe, M. Nogge, and E. Noldus, eds.), 1998, pp. 425–432.
- [29] C. Shu and S. Osher, *Efficient implementation of essentially non-oscillatory shock capturing schemes*, J. Comp. Physics, **77** (1988), pp. 439–471.
- [30] J. Smoller, *Shock Waves and Reaction-Diffusion Equations*, Springer-Verlag, New York, Heidelberg, Berlin, 1982.
- [31] B. P. Sommeijer, L. F. Shampine, and J. G. Verwer, *RKC: an explicit solver for parabolic PDEs*, Tech. Rep. Mas-R9715, Mathematisch Centrum, Amsterdam, 1997.
- [32] Y. Son and D. Haidvogel, *A semi-implicit ocean circulation model using a generalized topography following coordinate system*, J. Comp. Physics, **115** (1994), pp. 228–244.
- [33] A. Staniforth and J. Côté, *Semi-Lagrangian integration schemes for the atmospheric models: A review*, We. Rev., **119** (1991), pp. 2206–2223.
- [34] E. Süli, *Convergence and stability of the Lagrange-Galerkin method for the Navier-Stokes equations*, Numer. Math., **53** (1988), pp. 459–483.
- [35] R. Temam, *Navier-Stokes Equations: Theory and Numerical Analysis*, North-Holland, Amsterdam, 1984.
- [36] C. Temperton and A. Staniforth, *An efficient two-time-level Semi-Lagrangian semi-implicit integration scheme*, Quart. J. Roy. Meteor. Soc., **113** (1987), pp. 1025–1039.
- [37] P. J. Van der Houwen, *Explicit Runge—Kutta formulas with increased stability boundaries*, Numer. Math., **20** (1972), pp. 149–164.
- [38] P. J. Van der Houwen and B. P. Sommeijer, *On the internal stability of explicit, m-stage Runge—Kutta methods for large m-values*, Z. Angew. Math. Mech., **60** (1980), pp. 479–485.
- [39] M. Van Loon, *Tests on Semi-Lagrangian transport and interpolation*, Tech. Rep. NM-R9217, Mathematisch Centrum, Amsterdam, 1992.
- [40] J. G. Verwer, *Explicit Runge—Kutta methods for parabolic partial differential equations*, NM-R96026, Amsterdam, 1996.
- [41] J. G. Verwer, W. H. Hundsdorfer, and B. P. Sommeijer, *Convergence properties of the Runge—Kutta—Chebyshev method*, Numer. Math., **57** (1990), pp. 157–178.
- [42] Z. Zlatev, *Computer Treatment of Large Air Pollution Models*, Kluwer, Dordrecht, 1995.

Received 2 Nov. 2001

Revised 7 Jun. 2002