Time Integration Schemes in Eulerian-Lagrangian Computations for Advection-Diffusion Reaction Problems

Mohammed Seaïd

Fachbereich Mathematik, TU Darmstadt, 64289 Darmstadt, Germany
seaid@mathematik.tu-darmstadt.de

Abstract

The accuracy and efficiency of several lower and higher order time integration schemes in Eulerian-Lagrangian computations are investigated for solution of advection diffusion problems with nonlinear reaction terms. The implementation of these schemes differ from their Eulerian counterparts in the fact that they are applied during each time step, along the characteristic curves rather than in the time direction. The major focus is to examine the computational characteristics of a class of implicit, explicit, and implicit-explicit time marching methods combined to Eulerian-Lagrangian procedure. The obtained results for several benchmark problems are considered to be representative, and might be helpful for a fair rating of solution schemes, particularly in long time computations.

Key words: Time Integration Schemes; Eulerian-Lagrangian Computations; Advection-Diffusion Reaction Problems; Runge-Kutta Methods

1 Introduction

A vast majority of phenomena in air pollution, fluid mechanics, atmospheric motions, ocean circulation, porous media or meteorology has been modelled by the transport-diffusion-reaction equations

$$\begin{aligned} \phi(\mathbf{x})\frac{\partial \mathbf{u}}{\partial t} + \mathbf{v}(t,\mathbf{x},\mathbf{u})\cdot\nabla\mathbf{u} - \nabla\cdot(\mathbf{D}(\mathbf{x})\nabla\mathbf{u}) &= \mathbf{f}(t,\mathbf{x},\mathbf{u}), \quad \text{in } (0,T)\times\Omega, \\ \alpha(\mathbf{x})\mathbf{u}(t,\mathbf{x}) + \beta(\mathbf{x})\mathbf{n}\cdot\nabla\mathbf{u} &= \mathbf{g}(t,\mathbf{x}), \quad \text{on } [0,T]\times\Gamma, \\ \mathbf{u}(\mathbf{x},0) &= \mathbf{u}_0(\mathbf{x}), \quad \text{in } \Omega, \end{aligned} \tag{1}$$

where Ω is an open bounded subdomain in $\mathbb{R}^d$ ($d = 1, 2$ or 3) with smooth boundary Γ, $\mathbf{n}$ denotes the outward normal on Γ, and $[0, T]$ is a time interval. Here ϕ denotes, for example porosity or density of transport medium, $\mathbf{u}(t, \mathbf{x})$ the concentration of some species, $\mathbf{v}\,(t, \mathbf{x}, \mathbf{u})$ the velocity field, $\mathbf{D}(\mathbf{x})$ the diffusion or dispersion tensor, and $\mathbf{f}(t, \mathbf{x}, \mathbf{u})$ the reaction term or force source. $\alpha(\mathbf{x})$, $\beta(\mathbf{x})$ and $\mathbf{g}(t, \mathbf{x})$ are given boundary functions, and $\mathbf{u}_0(\mathbf{x})$ is fixed initial condition. In the above and in what follows bold face types denote vector quantities. We assume that $\mathbf{v}$ is divergence-free

$$\nabla \cdot \mathbf{v} = 0\,. \tag{2}$$

Solving the equations (1) numerically is still a considerable task in the case of convection dominated problems; particularly when certain nondimensional parameters reach high values. As example of these parameters the Reynolds number for Navier-Stokes equations and Peclet number for convection-diffusion equations, this convective term is a source of computational difficulties and oscillations. It is well known that the solutions of the equations (1) present steep fronts and even shock discontinuities, which need to be resolved accurately in applications and often cause severe numerical difficulties [13].

Eulerian methods use fixed grids and incorporate some upstream weighting in their formulations to stabilize the schemes. Among the class of Eulerian methods are the Petrov-Galerkin methods, the streamline diffusion methods and also include many other methods such as the high resolution methods from computational fluid dynamics, in particular, the Godunov methods and the essentially non-oscillatory methods [6,20]. All of these Eulerian methods are easy to formulate and to implement. However, time truncation errors dominate their solutions that introduce numerical diffusion and are subject to the Courant-Friedrichs-Lewy (CFL) stability conditions that put a restriction on the size of the time steps taken in numerical simulations.

Eulerian-Lagrangian Methods (ELM), on the other hand, make use of the transport nature of the governing equations (1). They combine the fixed Eulerian grids with a particle tracking along the characteristic curves of the governing equations, compare [5,8,14,21,15,4]. The Lagrangian treatment in these methods greatly reduces the time truncation errors in the Eulerian methods. In addition, these methods alleviate the restrictions on the Courant number, thus allowing for large time steps in the simulations.

Monotonicity and mass conservation are properties that may be relevant for some problems and, therefore, their importance cannot be overlooked. In order to overcome the principal drawback of ELM, that is the failure to conserve mass, we consider in this paper an improved ELM with limiters to convert the methods to quasimonotone and mass-conservative at minor additional computational cost. Analysis of this method has been done in [7].

The emphasis in this work is on the time integration of the resultant system of ordinary differential equations (ODE) induced from discretization in space variable along the characteristic curves. The proposed schemes range from implicit procedures to explicit Runge-Kutta schemes and implicit-explicit splitting. These different methods lead to techniques all of which are occurring in Eulerian framework since years. Theoretical considerations can provide some ideas, concerning stability, convergence rates, restriction on time stepsizes, or qualitative behavior of the solution, but a complete quantitative analysis is not possible today. Therefore, the only way to make a judgment is to perform numerical tests, at least for some problems which seem to be representative. However, looking into the literature, it seems that there have not been many studies of this type which can give satisfactory answers.

In this paper, first, Lagrangian and Eulerian stages in ELM methods are introduced. Thereafter, time stepping techniques employed to integrate the semi-discrete problem are presented. After experiments with the different time integration schemes for a variety of advection-diffusion reaction benchmark problems, accuracy and efficiency of the time integration schemes are discussed.

2 Eulerian-Lagrangian Methods

The ELM methods we consider in this paper consist on two fractional steps. The first step is the Lagrangian interpretation for the advection part of (1) by the modified method of characteristics, while the second step uses the Eulerian coordinates for discretization of the reaction-diffusion part in (1).

2.1 Lagrangian Stage

Let first consider the homogeneous advective part of the problem (1)

$$\begin{aligned} \frac{D\mathbf{u}}{Dt} = \phi(\mathbf{x})\frac{\partial \mathbf{u}}{\partial t} + \mathbf{v}(t,\mathbf{x},\mathbf{u})\cdot\nabla\mathbf{u} = \mathbf{0}, \quad \text{in} \quad (0,T)\times\Omega\,, \\ \mathbf{u}(t,\mathbf{x}) = \mathbf{0}, \quad \text{on} \quad [0,T]\times\Gamma\,, \\ \mathbf{u}(0,\mathbf{x}) = \mathbf{u}_0(\mathbf{x}), \quad \text{in} \quad \Omega\,. \end{aligned} \tag{3}$$

Recall that $\frac{D\mathbf{w}}{Dt}$ measures the rate of change of the function $\mathbf{w}$ following the trajectories of the flow particles. The fundamental idea of ELM is to impose a regular grid at the new time level, and to backtrack the flow trajectories to the previous time level. At the old time level, the quantities that are needed are evaluated by interpolation from their known values on a regular grid.

Let the time interval $[0, T]$ be divided into N subintervals $[t_n, t_{n+1}]$ of length Δt such that $t_n = n\Delta t$ and $T = N\Delta t$. Following [15,8], the characteristic curves of the equation (3) are the solution of initial-value problem for ODE

$$\begin{aligned} \frac{\mathrm{d}\mathcal{X}(\tau; t_{n+1}, \mathbf{x})}{\mathrm{d}\tau} &= \mathbf{v}\left(\tau, \mathcal{X}(\tau; t_{n+1}, \mathbf{x}), \mathbf{u}\right), \quad \tau \in [t_n, t_{n+1}], \\ \mathcal{X}(t_{n+1}; t_{n+1}, \mathbf{x}) &= \mathbf{x}. \end{aligned} \tag{4}$$

Note that $\mathcal{X}(\tau; t_{n+1}, \mathbf{x})$ is the departure point at time τ of a particle that will arrive at $\mathbf{x}$ at time t_{n+1}. The ELM do not follow the flow particles forward in time, as the Lagrangian schemes do, instead their trace backwards the position at time t_n of particles that will reach the points of a fixed mesh at time t_{n+1}. By so doing, the ELM methods avoid the grid distortion difficulties that the conventional Lagrangian schemes have.

The solution of (4) can be expressed as

$$\mathcal{X}(t_n; t_{n+1}, \mathbf{x}) = \mathbf{x} - \int_{t_n}^{t_{n+1}} \mathbf{v}\left(\tau, \mathcal{X}(\tau; t_{n+1}, \mathbf{x}), \mathbf{u}\right) d\tau. \tag{5}$$

To compute $\mathcal{X}(t_n; t_{n+1}, \mathbf{x})$ we used a method first proposed by [22] in the context of semi-Lagrangian schemes to integrate the weather prediction equations. Thus, we estimate the integral in (5) by a fourth order explicit Runge-Kutta scheme which is accurate enough to maintain a particle on its curved trajectory. The velocity is extrapolated at both, $t_{n+\frac{1}{2}}$ and t_{n+1}, and an iterative process is used to decouple the equations, see [18,19] for details.

Once the characteristics feet $\mathcal{X}(t_n; t_{n+1}, \mathbf{x})$ are known, the conventional ELM advect the solution of (3) at instant t_{n+1} as

$$\mathbf{u}(t_{n+1}, \mathbf{x}) = \bar{\mathbf{u}}(t_{n+1}, \mathbf{x}) = \mathbf{u}\left(t_n, \mathcal{X}(t_n; t_{n+1}, \mathbf{x})\right). \tag{6}$$

In general, the departure points $\mathcal{X}(t_n; t_{n+1}, \mathbf{x})$ do not coincide with the spatial position of a gridpoint. A requirement is then that the scheme to compute $\mathcal{X}(t_n; t_{n+1}, \mathbf{x})$ be provided with a search-locate algorithm to find the host element where such point is located. A general, efficient and easy to implement scheme to perform this step in arbitrary grids is presented in [2]. Assuming that a suitable approximation is made for $\mathcal{X}(t_n; t_{n+1}, \mathbf{x})$, then velocity $\mathbf{v}$ in (5) and function $\bar{\mathbf{u}}$ in (6) must be obtained by interpolation from known values at the grid points. The interpolation procedure we used in this paper is the bicubic Lagrange interpolations most commonly used in practice [22,5,18].

Note that the divergence-free condition (2) implies that

$$\int_\Omega \mathbf{v} \cdot \nabla \mathbf{u}\, \mathrm{d}\mathbf{x} = -\int_\Omega \mathbf{u} \nabla \cdot \mathbf{v}\, \mathrm{d}\mathbf{x} = 0,$$

and consequently leads for (3) to the conservation law

$$\int_\Omega \phi(\mathbf{x})\mathbf{u}(t,\mathbf{x})\,\mathrm{d}\mathbf{x} = \int_\Omega \phi(\mathbf{x})\mathbf{u}_0(\mathbf{x}), \quad \forall\, t \in [0,T]\,, \tag{7}$$

In many situations, the ELM methods do not preserve the condition (7) straightforwardly. However, techniques have been studied in literature to convert the conventional ELM methods to quasimonotone and mass conservation methods. For instance, authors in [5] used so-called monotone interpolation, this implies that a value obtained by interpolating in a grid cell, lies between the maximum and minimum value in the vertices of this grid cell. The analysis of stability and convergence of this method is detailed in [3]. Another way to preserve (7) is to consider an ELM with adjusted advection. Here we briefly describe the procedure and for further details and analysis we refer to [7]. Given $\mathbf{u}(t_n,\mathbf{x})$, the solution at the next time level $\mathbf{u}(t_{n+1},\mathbf{x})$ is obtained in the following steps:

Compute the departure point $\mathcal{X}(t_n;t_{n+1},\mathbf{x})$ from (5), identify the element of the mesh where such a point is located, and compute the approximation $\bar{\mathbf{u}}(t_{n+1},\mathbf{x}) = \mathbf{u}\left(t_n,\mathcal{X}(t_n;t_{n+1},\mathbf{x})\right)$ employing the bicubic Lagrange interpolation. Evaluate the conservation integrals

$$\mathbf{C}(t_n) = \int_\Omega \phi(\mathbf{x})\mathbf{u}(t_n,\mathbf{x})\,\mathrm{d}\mathbf{x}, \quad \text{and} \quad \bar{\mathbf{C}}(t_{n+1}) = \int_\Omega \phi(\mathbf{x})\bar{\mathbf{u}}(t_{n+1},\mathbf{x})\,\mathrm{d}\mathbf{x}\,.$$

Perturb the characteristics curves according to

$$\mathcal{X}^+ = \mathcal{X}(t_n;t_{n+1},\mathbf{x}) + \delta h, \quad \text{and} \quad \mathcal{X}^- = \mathcal{X}(t_n;t_{n+1},\mathbf{x}) - \delta h\,,$$

where h denotes the diameter elements to be introduced in the space discretization, and $\delta = \xi\dfrac{\mathbf{v}}{\phi}\Delta t$, with ξ is a fixed constant in $[0,1)$. Then, set

$$\tilde{\mathbf{u}}(t_{n+1},\mathbf{x}) = \begin{cases} \max(\mathbf{u}\left(t_n,\mathcal{X}^+\right),\mathbf{u}(t_n,\mathcal{X}^-))\,, & \text{if} \quad \mathbf{C}(t_n) > \bar{\mathbf{C}}(t_{n+1})\,, \\ \min(\mathbf{u}\left(t_n,\mathcal{X}^+\right),\mathbf{u}(t_n,\mathcal{X}^-))\,, & \text{if} \quad \mathbf{C}(t_n) \le \bar{\mathbf{C}}(t_{n+1})\,, \end{cases}$$

and calculate the corresponding conservation integral

$$\tilde{\mathbf{C}}(t_{n+1}) = \int_\Omega \phi(\mathbf{x})\tilde{\mathbf{u}}(t_{n+1},\mathbf{x})\,\mathrm{d}\mathbf{x}\,.$$

If $\bar{\mathbf{C}}(t_{n+1}) \neq \tilde{\mathbf{C}}(t_{n+1})$, calculate the limiting coefficient $\theta = \theta(t_n)$ in such a way that

$$\mathbf{C}(t_n) = \theta\bar{\mathbf{C}}(t_{n+1}) + (1-\theta)\tilde{\mathbf{C}}(t_{n+1})\,.$$

Update the new solution $\mathbf{u}(t_{n+1},\mathbf{x})$ by

$$\mathbf{u}(t_{n+1},\mathbf{x}) = \theta\bar{\mathbf{u}}(t_n,\mathbf{x}) + (1-\theta)\tilde{\mathbf{u}}(t_{n+1},\mathbf{x})\,. \tag{8}$$

It is easy to see that

$$\int_\Omega \phi(\mathbf{x})\mathbf{u}(t_{n+1},\mathbf{x})\,\mathrm{d}\mathbf{x} = \theta\bar{\mathbf{C}}(t_{n+1}) + (1-\theta)\tilde{\mathbf{C}}(t_{n+1}) = \mathbf{C}(t_n)\,,$$

or $\mathbf{C}(t_{n+1}) = \mathbf{C}(t_n)$, and the new adjusted solution (8) conserves the mass. This adjusted ELM which requires additional computational work can be easily implemented in an existing conventional ELM codes.

2.2 Eulerian Stage

In order to differentiate in the characteristic direction $\mathbf{s} = \mathbf{s}(\mathbf{x})$ associated with the operator $\frac{D}{Dt}$, we first define

$$\gamma(\mathbf{x}) = \left[\phi(\mathbf{x})^2 + \|\mathbf{v}(t,\mathbf{x},\mathbf{u})\|^2\right]^{\frac{1}{2}}\,,$$

where $\|\cdot\|$ is the Euclidean norm. Then,

$$\frac{\partial}{\partial \mathbf{s}} = \frac{1}{\gamma(\mathbf{x})}\frac{D}{Dt}\,,$$

and the first equation in (1) is transformed to

$$\gamma(\mathbf{x})\frac{\partial \mathbf{u}}{\partial \mathbf{s}} - \nabla\cdot(\mathbf{D}(\mathbf{x})\nabla\mathbf{u}) = \mathbf{f}(t,\mathbf{x},\mathbf{u})\,.$$

Based on the approach presented in [8], the characteristic derivative is approximated by

$$\begin{aligned}\gamma(\mathbf{x})\frac{\partial \mathbf{u}}{\partial \mathbf{s}} &\approx \gamma(\mathbf{x})\frac{\mathbf{u}(t_{n+1},\mathbf{x}) - \bar{\mathbf{u}}(t_n,\mathbf{x})}{[\|\mathbf{x}-\mathcal{X}\|^2 + (\Delta t)^2]^{\frac{1}{2}}} \\ &= \phi(\mathbf{x})\frac{\mathbf{u}(t_{n+1},\mathbf{x}) - \bar{\mathbf{u}}(t_n,\mathbf{x})}{\Delta t}\,.\end{aligned}$$

The ELM methods can be applied in combination with finite element, finite difference or spectral methods. In particular, the application of ELM in the framework of finite element is known as the Characteristic-Galerkin method [4]. The first application to transport-diffusion problems is due to [8], the authors combined the finite element and finite difference methods with the method of characteristics to treat the one-dimensional form of the problem model (1) without reaction term (i.e. $\mathbf{f} = \mathbf{0}$).

Define a numerical mesh over Ω with grid-points $\mathbf{x}_h$ for spatial scale h. We use the notations $\mathbf{w}^n(\mathbf{x}) = \mathbf{w}(t_n,\mathbf{x})$, $w_h(t) = \mathbf{w}(t,\mathbf{x}_h)$ and $w_h^n = \mathbf{w}(t_n,\mathbf{x}_h)$ unless otherwise stated. Finite element or finite difference methods used for

the space discretization of the resulting problem from the Lagrangian stage, leads to the semi-discrete problem

$$\begin{aligned} \phi_h \frac{du_h}{dt} - \mathcal{D}_h u_h &= f_h(t, u_h), \quad t \in (0, T]\,, \\ u_h(0) &= u_0(\mathbf{x}_h)\,, \end{aligned} \tag{9}$$

where $\mathcal{D}_h$ denotes the space discretization of the diffusion operator with the boundary conditions included. The initial-value problem (9) can be rewritten in common ODE notation as

$$\frac{d\mathbf{U}}{dt} = \mathbf{F}(t, \mathbf{U})\,, \tag{10}$$

where $\mathbf{U}$ and $\mathbf{F}$ are vector-valued functions with the entries u_h and $\frac{1}{\phi_h}(\mathcal{D}_h u_h + f_h(t, u_h))$, respectively. Note that the well known method of lines applied directly to (1) results also in a system of ODE like (10), however, here the convective term which cause many difficulties in Eulerian semi-discretization has been moved from the right hand side in (10) using the Lagrangian step in ELM formulation.

3 Time Integration Schemes

In this section we briefly review some of the methods used to integrate the system of ODE (9) subject to the computed solution in the Lagrangian stage

$$\bar{u}_h^n = \mathbf{u}\left(t_n, \mathcal{X}_h(t_n; t_{n+1}, \mathbf{x}_h)\right)\,. \tag{11}$$

Difficulties often appears when the Jacobian of $\mathbf{F}$, $\partial\mathbf{F}/\partial\mathbf{U}$, has large eigenvalues. This may rise to stiffness. Thus, time integration schemes for ELM depend strongly on the diffusion coefficients and the grid refinements, and for these reasons it is preferable that these schemes have to be either implicit or explicit with large stability regions. On other hand, results on convergence analysis of ELM reported in [8,14,4] showed that the spatial error of ELM methods is $\mathcal{O}(h^\alpha)$ with $2 \leq \alpha \leq 3$. Therefore, we consider only second or higher order time discretization. We further demand that the methods proposed are not diffusive (do not introduce extra numerical diffusion) and avoid the order reduction. The following time integrators are selected:

3.1 Eulerian-Lagrangian Crank-Nicolson (ELM-CN)

The most popular time integrator for (9) in the ELM framework is the Crank-Nicolson method. Let $\bar{u}_h^n$ being computed as in (11), then the ELM-CN scheme solves for u_h^{n+1}

$$\phi_h \frac{u_h^{n+1} - \bar{u}_h^n}{\Delta t} - \frac{1}{2}\mathcal{D}_h u_h^{n+1} - \frac{1}{2}\mathcal{D}_h \bar{u}_h^n = \frac{1}{2} f_h^{n+1}(u_h^{n+1}) + \frac{1}{2} f_h^n(\bar{u}_h^n)\,, \tag{12}$$

Notice that the conventional Crank-Nicolson and ELM-CN schemes are unconditionally stable, so that the choice of Δt may be based on accuracy considerations. To find the solution $\mathbf{u}^{n+1}$ from (12) one has to solve, at each time level, a linear/nonlinear system of algebraic equations. For instance, in pure advection-diffusion problems (i.e. $\mathbf{f} = \mathbf{0}$) only linear systems arise, and Krylov subspace methods can be used as linear solvers.

3.2 Eulerian-Lagrangian Explicit Runge-Kutta (ELM-RK4)

The fourth order Runge-Kutta formulae [10] to solve (9) can be written as

$$\begin{aligned}
K_h^{(1)} &= \mathcal{D}_h \bar{u}_h^n + f_h\left(t_n, \bar{u}_h^n\right)\,, \\
K_h^{(2)} &= \mathcal{D}_h K_h^{(1)} + f_h\left(t_n + \frac{\Delta t}{2}, \bar{u}_h^n + \frac{1}{2}K_h^{(1)}\right)\,, \\
K_h^{(3)} &= \mathcal{D}_h K_h^{(2)} + f_h\left(t_n + \frac{\Delta t}{2}, \bar{u}_h^n + \frac{1}{2}K_h^{(2)}\right)\,, \\
K_h^{(4)} &= \mathcal{D}_h K_h^{(3)} + f_h\left(t_n + \Delta t, \bar{u}_h^n + K_h^{(3)}\right)\,, \\
u_h^{n+1} &= \bar{u}_h^n + \frac{\Delta t}{6\phi_h}\left(K_h^{(1)} + 2K_h^{(2)} + 2K_h^{(3)} + K_h^{(4)}\right)\,.
\end{aligned} \tag{13}$$

Because ELM-RK4 evaluate explicitly the right-hand side of the system (9), then it has to satisfy a stability criterium. Based on the analysis in [16] a stability criterium can be derived in such a way that the ELM-RK4 scheme is stable if the time stepsize Δt satisfies

$$\Delta t \leq \frac{h^2}{\sigma h^2 + 2^d \bar{\mathbf{d}}}\,, \tag{14}$$

where $\bar{\mathbf{d}}$ is the upper bound of the diffusion matrix $\mathbf{D}$, d is the dimension of the space domain Ω, and σ is the maximum of the Jacobian $|\partial \mathbf{f}/\partial \mathbf{u}|$.

3.3 Eulerian-Lagrangian Explicit Runge-Kutta Chebychev (ELM-RKC)

The Runge-Kutta Chebychev (RKC) method [23] has been designed for explicit time integration of systems of parabolic equations. The scheme possesses an extended real stability interval. To solve (9), the RKC scheme takes the form

$$\begin{aligned}
&K_h^{(0)} = \bar{u}_h^n\,,\\
&K_h^{(1)} = K_h^{(0)} + \tilde{\mu}_1 F_h^{(0)}\\
&K_h^{(j)} = \mu_j K_h^{(j-1)} + \nu_j K_h^{(j-2)} + (1-\mu_j-\nu_j)K_h^{(0)} + \\
&\qquad \tilde{\mu}_j F_h^{(j-1)} + \tilde{\gamma}_j F_h^{(0)}, \quad 2 \le j \le s\,,\\
&u_h^{n+1} = K_h^{(s)}\,.
\end{aligned} \tag{15}$$

The coefficients are define for arbitrary $s \ge 0$ as follows. Consider the Chebyshev polynomial of the first kind of degree j

$$T_j(z) = \cos(j\arccos z), \qquad -1 \le z \le 1\,.$$

Then

$$\epsilon = \frac{2}{13}, \quad q_0 = 1 + \frac{\epsilon}{s^2}, \quad q_1 = \frac{T_s'(q_0)}{T_s''(q_0)}\,,$$

$$b_j = \frac{T_s''(q_0)}{(T_s'(q_0))^2}, \quad (2 \le j \le s), \quad b_0 = b_2, \quad b_1 = b_2\,,$$

and

$$\tilde{\mu}_1 = b_1 q_1, \quad \mu_j = \frac{2b_j q_0}{b_{j-1}}, \quad \nu_j = \frac{-b_j}{b_j - 2}, \quad \tilde{\mu}_j = \frac{2b_j q_1}{b_{j-1}}\,,$$

$$\tilde{\gamma}_j = (1 - b_{j-1}T_{j-1}(q_0))\tilde{\mu}_j, \quad (2 \le j \le s)\,.$$

In(15) the stage $F_h^{(j)} = \frac{\Delta t}{\phi_h}\left(\mathcal{D}_h K_h^{(j)} + f_h(t_n + c_j\Delta t, K_h^{(j)})\right)$. The c_j's are

$$c_j = \frac{T_s'(q_0)}{T_s''(q_0)}\frac{T_s''(q_0)}{T_s'(q_0)} \approx \frac{j^2-1}{s^2-1} \quad (2 \le j \le s), \quad c_1 = \frac{c_2}{T_2'(q_0)} \approx \frac{c^2}{4}, \quad c_s = 1\,.$$

It is worth remarking that the number of stages s in ELM-RKC and the conventional RKC schemes varies with Δt such that (see e.g. [23])

$$s = 1 + entier\left[\left(1 + \frac{(\sigma h^2 + 2^d\bar{\mathbf{d}})\Delta t}{0.65h^2}\right)^{\frac{1}{2}}\right]\,. \tag{16}$$

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

3.4 Eulerian-Lagrangian Implicit Runge-Kutta (ELM-SDIRK3)

In [10], a third order Singly Diagonal Implicit Runge-Kutta (SDIRK3) scheme has been introduced. Applied to (9) ELM-SDIRK3 can be implemented as

$$\begin{aligned}
K_h^{(1)} &= \bar{u}_h^n + \gamma \frac{\Delta t}{\phi_h} \left(\mathcal{D}_h K_h^{(1)} + f_h \left(t_n + \gamma \Delta t, K_h^{(1)} \right) \right) , \\
K_h^{(2)} &= \bar{u}_h^n + (1-2\gamma) \frac{\Delta t}{\phi_h} \left(\mathcal{D}_h K_h^{(1)} + f_h (t_n + (1-\gamma)\Delta t, K_h^{(1)}) \right) + \\
&\qquad \gamma \frac{\Delta t}{\phi_h} \left(\mathcal{D}_h K_h^{(2)} + f_h (t_n + (1-\gamma)\Delta t, K_h^{(2)}) \right) , \\
u_h^{n+1} &= \bar{u}_h^n + \frac{\Delta t}{2\phi_h} \left(\mathcal{D}_h K_h^{(1)} + f_h (t_n, K_h^{(1)}) + \mathcal{D}_h K_h^{(2)} + f_h (t_n, K_h^{(2)}) \right) ,
\end{aligned} \tag{17}$$

where $\gamma = \frac{3-\sqrt{3}}{6}$. Note that the new solution u_h^{n+1} in (17) is independent of any explicit process within the integration step. At each time step the ELM-SDIRK3 involves the storage of two levels of the solution $\bar{u}_h^n$, and requires two linear/nonlinear equation solvers per step. For efficiency reasons it is recommendable that the resultant nonlinear algebraic equations are solved iteratively with multigrid techniques to speed up the computational work.

3.5 Eulerian-Lagrangian Rosenbrock Scheme (ELM-ROS3P)

The significant shortcomings that time integration schemes for (9) might suffer is the order reduction [17]. In order to avoid this disadvantage in the standard Rosenbrock methods, the authors in [12] proposed a third-order improved Rosenbrock method (ROS3P). Applied to the equations (10) the combined ELM to ROS3P (ELM-ROS3P) scheme reads

$$\begin{aligned}
\left(\frac{\mathbf{I}}{\gamma \Delta t} - \partial \mathbf{F}^n / \partial \mathbf{U} \right) \mathbf{K}^{(i)} &= \mathbf{F} \left(t_n + \alpha_i \Delta t, \mathbf{U}^n + \sum_{j=1}^{i-1} a_{ij} \mathbf{K}^{(j)} \right) + \\
&\qquad \sum_{j=1}^{i-1} \frac{c_{ij}}{\Delta t} \mathbf{K}^{(j)} + \gamma_i \Delta t \mathbf{F}(t_n, \mathbf{U}^n), \quad i = 1, 2, 3 , \\
\mathbf{U}^{n+1} &= \mathbf{U}^n + \sum_{i=1}^{3} m_i \mathbf{K}^{(i)} ,
\end{aligned} \tag{18}$$

where $\mathbf{I}$ is the identity matrix and $\partial \mathbf{F}^n / \partial \mathbf{U}$ is the matrix Jacobian of $\mathbf{F}(t, \mathbf{U})$ with respect to $\mathbf{U}$ evaluated at $(t_n, \mathbf{U}^n)$. The other parameters are taken from [12] such as:

$$\gamma = 7.886751345948129E-01$$

$$\begin{aligned} a_{21} &= 1.267949192431123E+00 \quad & c_{21} &= -1.607695154586736E+00 \\ a_{31} &= 1.267949192431123E+00 \quad & c_{31} &= -3.464101615137755E+00 \\ a_{32} &= 0.000000000000000E+00 \quad & c_{32} &= -1.732050807568877E+00 \end{aligned}$$

$$\begin{aligned} \alpha_1 &= 0.000000000000000E+00 \quad & \gamma_1 &= 7.886751345948129E-01 \\ \alpha_2 &= 1.000000000000000E+00 \quad & \gamma_2 &= -2.113248654051871E-01 \\ \alpha_3 &= 1.000000000000000E+00 \quad & \gamma_3 &= -1.077350269189626E+00 \end{aligned}$$

$$\begin{aligned} m_1 &= 2.000000000000000E+00 \\ m_2 &= 5.773502691896258E-01 \\ m_3 &= 4.226497308103742E-01 \end{aligned}$$

Notice that during the ELM-ROS3P integration loop no nonlinear equation solvers are needed. All what is needed, however, is the Jacobian matrix evaluated analytically or computed using forward difference formula. The linear solvers can be again a Krylov subspace method. See [12] for numerical tests with ROS3P in Eulerian form.

3.6 Eulerian-Lagrangian Implicit-Explicit Splitting (ELM-IMEX)

The last time integration scheme we present in this section is the diagonally Implicit-Explicit (IMEX) Runge-Kutta methods frequently used for time integration of partial differential systems which contain stiff and non stiff terms, we refer to [1] for more details and further references are therein. Let split the equation (10) as

$$\frac{d\mathbf{U}}{dt} = \mathbf{F}_{\text{exp}}(t, \mathbf{U}) + \mathbf{F}_{\text{imp}}(t, \mathbf{U}) \,, \tag{19}$$

where $\mathbf{F}_{\text{exp}}(t, \mathbf{U})$ and $\mathbf{F}_{\text{imp}}(t, \mathbf{U})$ are the parts of $\mathbf{F}$ to be treated explicitly and implicitly, respectively. In practice, in order to avoid nonlinear iterations, $\mathbf{F}_{\text{exp}}$ contains the nonlinear non stiff part (e.g. reaction terms), and $\mathbf{F}_{\text{imp}}$ includes the linear stiff part (e.g. diffusion terms). Hence, setting $\mathbf{F}_{\text{exp}} = \mathbf{f}$ and $\mathbf{F}_{\text{imp}} = \mathcal{D}_h$ in (19), the combined ELM to IMEX (ELM-IMEX) scheme we consider for the semi-discrete problem (9) consists of

$$
\begin{aligned}
K_h^{(i)} &= \bar{u}_h^n + \frac{\Delta t}{\phi_h}\sum_{j=1}^{i-1}\tilde{a}_{ij} f_h\left(t_n + \tilde{c}_i\Delta t, K_h^{(j)}\right) + \frac{\Delta t}{\phi_h}\sum_{j=1}^{s} a_{ij}\mathcal{D}_h K_h^{(j)}\,,\\
u_h^{n+1} &= \bar{u}_h^n + \frac{\Delta t}{\phi_h}\sum_{i=1}^{s}\tilde{w}_i f_h\left(t_n, K_h^{(i)}\right) + \frac{\Delta t}{\phi_h}\sum_{i=1}^{s} w_i\mathcal{D}_h K_h^{(i)}\,,
\end{aligned}
\tag{20}
$$

where s is the number of stage, a_{ij} are the stage weights, c_i the abscissas, and b_i are the main scheme weights. The $s \times s$ matrices $\tilde{A} = (\tilde{a}_{ij})$; $A = (a_{ij})$ and the s-vectors $\tilde{\mathbf{c}}$, $\mathbf{c}$; $\tilde{\mathbf{w}}$; $\mathbf{w}$ are the standard coefficients which characterize the IMEX s-stage Runge-Kutta scheme, and they are given by the usual double Butcher tables. In our numerical computation we consider the third order IMEX scheme developed in [1], the associated double Butcher tables can be represented as

0	0	0	0	0	0
$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0
$\frac{2}{3}$	$\frac{11}{18}$	$\frac{1}{18}$	0	0	0
$\frac{1}{2}$	$\frac{5}{6}$	$-\frac{5}{6}$	$\frac{1}{2}$	0	0
1	$\frac{1}{4}$	$\frac{7}{4}$	$\frac{3}{4}$	$-\frac{7}{4}$	0
	$\frac{1}{4}$	$\frac{7}{4}$	$\frac{3}{4}$	$-\frac{7}{4}$	0

0	0	0	0	0	0
$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	0
$\frac{2}{3}$	0	$\frac{1}{6}$	$\frac{1}{2}$	0	0
$\frac{1}{2}$	0	$-\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0
1	0	$\frac{3}{2}$	$-\frac{3}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
	0	$\frac{3}{2}$	$-\frac{3}{2}$	$\frac{1}{2}$	$\frac{1}{2}$

The left and right tables represent the explicit and the implicit Runge-Kutta methods. Note that in the splitting (20) only linear solvers are required.

4 Numerical Examples

In order to illustrate as well as to compare the performance and accuracy of the schemes introduced in the previous section, we have run some numerical tests for two-dimensional version of advection-diffusion reaction equation (1) i.e. $\mathbf{x} = (x, y)^T$ and $\mathbf{v} = (v_1, v_2)^T$. We assumed constant diffusion coefficient $\mathbf{D}(\mathbf{x}) = D$ and we set $\phi(\mathbf{x}) = 1$. We defined the CFL number as follows

$$
\mathrm{CFL}_x = \max|v_1|\frac{\Delta t}{\Delta x},\ \mathrm{CFL}_y = \max|v_2|\frac{\Delta t}{\Delta y},\ \mathrm{CFL} = \sqrt{\mathrm{CFL}_x^2 + \mathrm{CFL}_y^2}\,. \tag{21}
$$

For the two first examples the analytical solution is known, so that we can evaluate the error function $\mathbf{e}$ as

$$
\mathrm{e}_h^n = u_h^n - u(t_n, \mathbf{x}_h),
$$

where $u(t_n, \mathbf{x}_h)$ 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

$$\|\mathbf{e}\|_{L_t^\infty} = \max_{1\leq n\leq N} \|\mathbf{e}^n\|_{L_s^\infty},$$
$$\|\mathbf{e}\|_{L_t^1} = \Delta t \sum_{n=0}^{N} \|\mathbf{e}^n\|_{L_s^1},$$
$$\|\mathbf{e}\|_{L_t^2} = \left(\Delta t \sum_{n=0}^{N} \|\mathbf{e}^n\|_{L_s^2}^2\right)^{\frac{1}{2}}.$$

Here $\|.\|_{L_s^p} = \|.\|_{L^p(\Omega)}$ and $\|.\|_{L_t^p} = \|.\|_{L^p([0,T))}$ denote the discrete L^p-norms in the space domain Ω and time interval $[0, T)$, respectively. These error norms along with the elevation of 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 processors running Fortran codes under Linux 2.2. In all our computations, the spatial domain Ω is uniformly discretized with $\Delta x = \Delta y = h$, we used finite difference method for the space dicretization with a fourth-order centred difference for the the diffusion operator $\mathcal{D}_h$ using the nine stencil points. The resulting algebraic systems of linear equations were solved using the preconditioned conjugate gradient. We used the diagonal as preconditioner and a tolerance of 10^{-6} to stop the iterations.

4.1 Example 1

First we check the accuracy of the schemes in terms of temporal error norms. We consider the problem model (1) in the unit square with $v = (x, -y)^T$. The reaction term f, boundary and initial conditions are chosen in such a way

$$u(t, x, y) = \sin(\pi xy(1 + t))$$

is the analytical solution of the problem. In table 1 we summarize the relative error norms and the CPU time obtained at $t = 1$ for different time and space sizes keeping D fixed to 5×10^{-3}.

In terms of accuracy table 1 shows that only ELM-SDIRK3 and ELM-ROS3P have preserved the expected third order of accuracy, while the order of accuracy has been reduced in other schemes. For instance, the ELM-RK4 suffer seriously from the order reduction, it also goes unstable when the grid size is refined to $\frac{1}{128}$ (due to the stability condition (14)).

Regarding to the efficiency of the schemes, the CPU time in table 1 gives advantage to the ELM-RKC scheme. Needless to say that by decreasing the space step h the number of stages s in ELM-RKC increases according to the formula (16). On the other hand, we have noted that the main part of the CPU time in ELM-SDIRK3 and ELM-SDIRK3 was used for solving the associated linear systems using the preconditioned conjugate gradient subroutine.

Table 1
Results for the accuracy test problem with $D = 5 \times 10^{-3}$. Here — means instability of the scheme and the CPU time is given in seconds.

	Δt	h	L_t^∞-error	L_t^1-error	L_t^2-error	CPU
ELM-CN	10^{-2}	$\frac{1}{64}$	3.85E-04	3.61E-04	3.66E-04	1.96
	5×10^{-3}	$\frac{1}{64}$	9.99E-05	9.85E-05	9.89E-05	3.91
	5×10^{-3}	$\frac{1}{128}$	9.97E-05	9.59E-05	9.81E-05	9.57
ELM-RK4	10^{-2}	$\frac{1}{64}$	2.21E-04	2.03E-04	2.19E-04	2.56
	5×10^{-3}	$\frac{1}{64}$	5.07E-05	4.95E-05	4.97E-05	5.71
	5×10^{-3}	$\frac{1}{128}$	—	—	—	—
ELM-RKC	10^{-2}	$\frac{1}{64}$	3.85E-04	3.61E-04	3.67E-04	1.73
	5×10^{-3}	$\frac{1}{64}$	1.26E-05	1.14E-05	1.16E-05	3.02
	5×10^{-3}	$\frac{1}{128}$	1.17E-05	1.02E-05	1.07E-05	8.78
ELM-SDIRK3	10^{-2}	$\frac{1}{64}$	1.36E-04	1.29E-04	1.31E-04	2.97
	5×10^{-3}	$\frac{1}{64}$	1.81E-05	1.70E-05	1.72E-05	7.53
	5×10^{-3}	$\frac{1}{128}$	1.75E-05	1.61E-05	1.69E-05	19.71
ELM-ROS3P	10^{-2}	$\frac{1}{64}$	1.44E-04	1.40E-04	1.43E-04	2.71
	5×10^{-3}	$\frac{1}{64}$	2.00E-05	1.93E-05	1.94E-05	6.94
	5×10^{-3}	$\frac{1}{128}$	1.90E-05	1.72E-05	1.76E-05	18.13

4.2 **Example 2**

This example considers the advection-diffusion of rotating Gaussian pulse. The equations are of the form (1) with $f = 0$, $\mathbf{v} = (-4y, 4x)^T$, initial and boundary conditions are taken from the analytical solution

$$u(x, y, t) = \frac{\sigma^2}{\sigma^2 + 4Dt} \exp\left(-\frac{(\bar{x} - x_0)^2 + (\bar{y} - y_0)^2}{\sigma^2 + 4Dt)}\right),$$

where $\bar{x} = x\cos(4t) + y\sin(4t)$, $\bar{y} = -x\sin(4t) + y\cos(4t)$, $x_0 = -0.25$, $y_0 = 0$ and $\sigma^2 = 0.002$. The computational domain is $[-0.5, 0.5] \times [-0.5, 0.5]$ and the time period required for one complete rotation is $\frac{\pi}{2}$. We set $D = 10^{-4}$ and we display in table (2) the maximum, minimum, the spatial relative L^2-error norms and the CPU time after one complete revolution.

Table 2
Results for the Gaussian pulse after one revolution with $D = 10^{-4}$ and different CFL numbers. The CPU times are listed in minutes.

	h	CFL	$u_{\max}$	$u_{\min}$	L_s^2-error	CPU
Analytical	$\frac{1}{64}$		0.8642	0.0		
ELM-CN	$\frac{1}{64}$	2.5	0.8357	0.0	1.52E-03	1.50
	$\frac{1}{64}$	1.25	0.8279	0.0	1.51E-03	2.87
	$\frac{1}{128}$	2.5	0.8491	0.0	4.08E-04	6.19
ELM-RK4	$\frac{1}{64}$	2.5	0.7810	-0.0001	1.03E-03	1.91
	$\frac{1}{64}$	1.25	0.7323	-0.0001	1.02E-03	4.30
	$\frac{1}{128}$	2.5	0.8114	0.0	2.64E-04	9.16
ELM-RKC	$\frac{1}{64}$	2.5	0.7701	-0.0002	1.54E-03	1.02
	$\frac{1}{64}$	1.25	0.7319	-0.0005	1.52E-03	2.41
	$\frac{1}{128}$	2.5	0.7921	0.0	4.22E-04	5.33
ELM-SDIRK3	$\frac{1}{64}$	2.5	0.8639	0.0	3.63E-04	3.02
	$\frac{1}{64}$	1.25	0.8633	0.0	3.61E-04	5.93
	$\frac{1}{128}$	2.5	0.8641	0.0	8.39E-05	13.17
ELM-ROS3P	$\frac{1}{64}$	2.5	0.8640	0.0	3.60E-04	3.11
	$\frac{1}{64}$	1.25	0.8634	0.0	3.59E-04	6.07
	$\frac{1}{128}$	2.5	0.8641	0.0	8.81E-05	14.28

Table (2) reflects the expected spatial order of accuracy in ELM schemes and the quasi-monotonicity of the ELM when it combined with implicit time integration schemes (compare for instance the minimum and the maximum in ELM-SDIRK3 or ELM-ROS3P). In figure (1) we plot the results obtained for two different values of D, $h = \frac{1}{64}$ and CFL $= 2.5$ using the ELM-SDIRK3.

4.3 Example 3

This example consists to test the ability of ELM methods to compute the steady state solution of incompressible Navier-Stokes problem. Here we con-

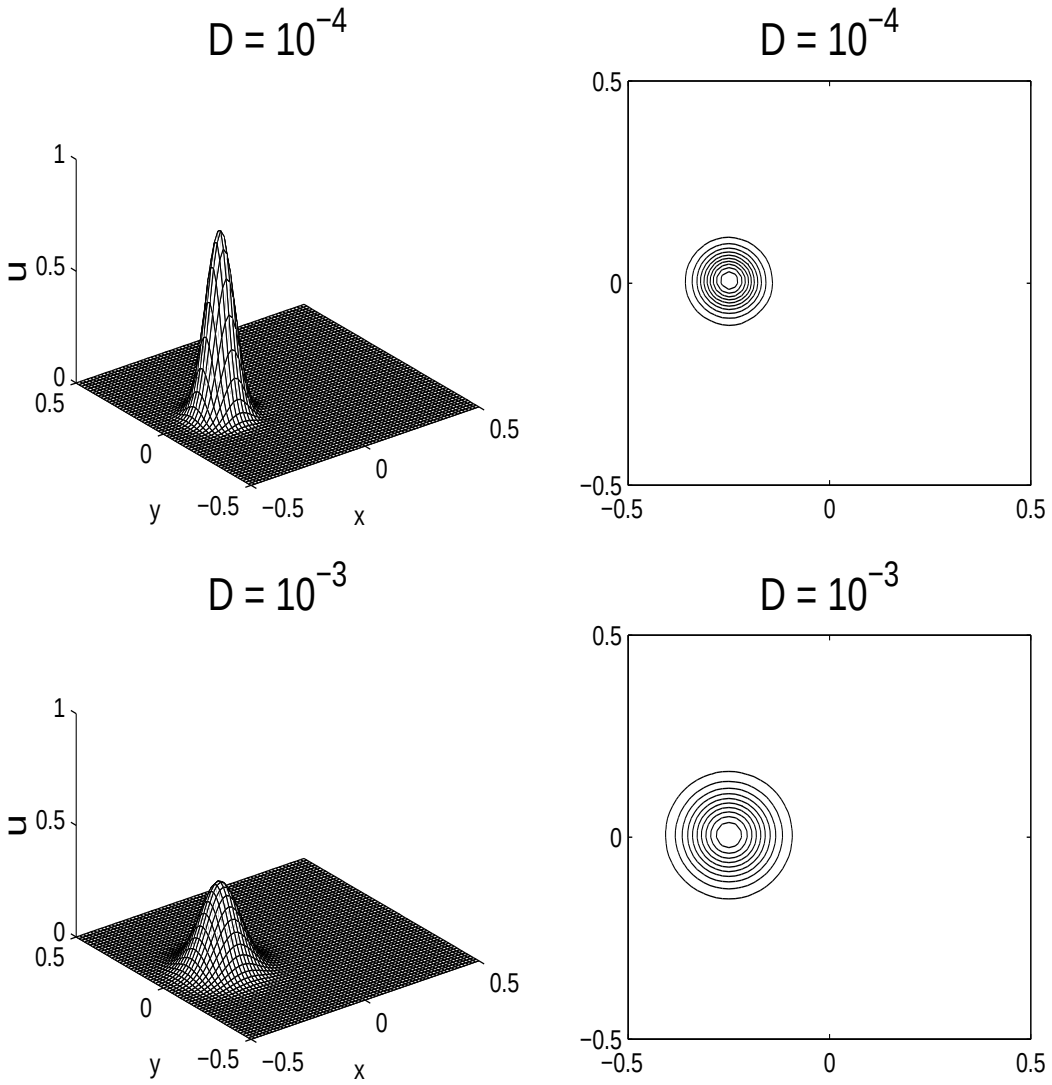

Fig. 1. The plots of u after one revolution using ELM-SDIRK3 scheme.

sider the canonical two-dimensional lid-driven cavity flow from [9]. We used the same geometry and boundary condition as in [9]. The time loop in our computations was terminated when the difference between two consecutive computed solutions in L^∞-norm is less than 10^{-5}.

In table 3 we compare the results for different Reynolds numbers and grid sizes. We have observed that ELM-ROS3P scheme gives exactly the same results as [9]. Furthermore, the plots of stream function, computed using ELM-ROS3P scheme, in figure 2 are in good agreement with those presented in [9].

Table 3. Results for the Lid-driven cavity flow problem, $\Delta t = 5 \times 10^{-3}$. ψ_c is the value of stream function at the center of the primary vortex located at (x_c, y_c) for different Reynolds numbers and grid refinements. The CPU times are given in hours.

		$Re = 10^2$			$Re = 10^3$			$Re = 10^4$		
	h	ψ_c	(x_c, y_c)	CPU	ψ_c	(x_c, y_c)	CPU	ψ_c	(x_c, y_c)	CPU
ELM-CN	$\frac{1}{64}$	-0.059	$(0.5879, 0.8612)$	1.64	-0.076	$(0.4033, 0.7710)$	2.79	-0.057	$(0.4933, 0.8132)$	5.91
	$\frac{1}{128}$	-0.073	$(0.6031, 0.8039)$	3.94	-0.084	$(0.4917, 0.6932)$	6.12	-0.041	$(0.4710, 0.7603)$	11.32
	$\frac{1}{256}$	-0.081	$(0.6099, 0.7681)$	7.23	-0.099	$(0.5002, 0.6076)$	13.91	-0.027	$(0.4669, 0.6663)$	29.15
ELM-RKC	$\frac{1}{64}$	-0.052	$(0.5879, 0.8612)$	1.21	-0.076	$(0.4033, 0.7710)$	2.34	-0.054	$(0.4933, 0.8132)$	5.75
	$\frac{1}{128}$	-0.071	$(0.6033, 0.8039)$	2.94	-0.083	$(0.4919, 0.6932)$	5.89	-0.039	$(0.4715, 0.7603)$	12.31
	$\frac{1}{256}$	-0.081	$(0.6099, 0.7681)$	6.15	-0.099	$(0.5002, 0.6076)$	13.39	-0.026	$(0.4669, 0.6663)$	31.29
ELM-ROSP3	$\frac{1}{64}$	-0.111	$(0.6158, 0.7408)$	1.67	-0.120	$(0.5269, 0.6280)$	2.01	-0.143	$(0.5184, 0.7494)$	2.88
	$\frac{1}{128}$	-0.108	$(0.6167, 0.7379)$	3.54	-0.119	$(0.5302, 0.5896)$	3.86	-0.130	$(0.5113, 0.6104)$	7.52
	$\frac{1}{256}$	-0.103	$(0.6172, 0.7344)$	7.10	-0.117	$(0.5313, 0.5625)$	7.60	-0.118	$(0.5117, 0.5332)$	16.79
ELM-SDIRK	$\frac{1}{64}$	-0.112	$(0.6158, 0.7409)$	1.99	-0.123	$(0.5269, 0.6279)$	1.86	-0.147	$(0.5186, 0.7497)$	2.71
	$\frac{1}{128}$	-0.109	$(0.6167, 0.7379)$	3.62	-0.119	$(0.5302, 0.5896)$	3.77	-0.131	$(0.5112, 0.6106)$	7.22
	$\frac{1}{256}$	-0.103	$(0.6172, 0.7344)$	7.03	-0.117	$(0.5313, 0.5625)$	7.51	-0.119	$(0.5117, 0.5333)$	16.01

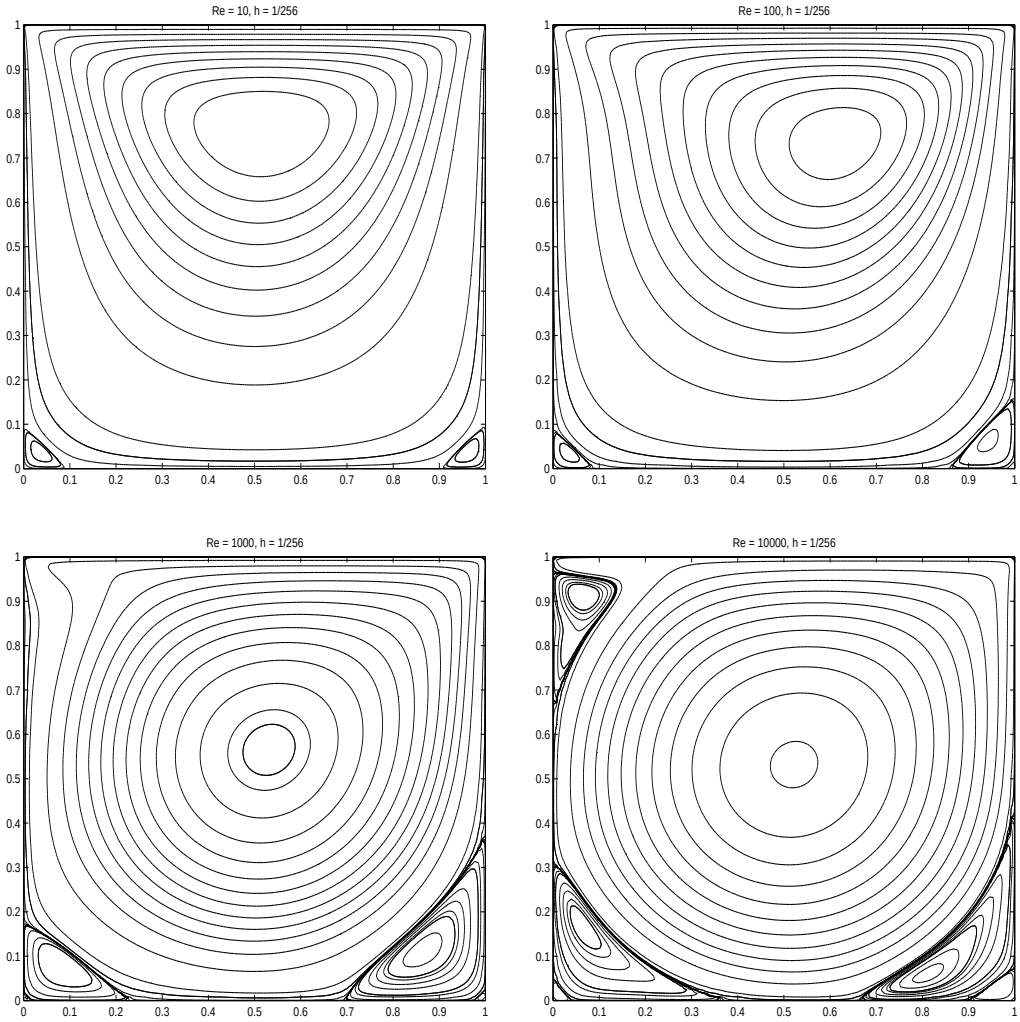

Fig. 2. The stream function plots using the ELM-ROS3P scheme.

4.4 Example 4

Our final test consists of applying the ELM-IMEX scheme to an advection-diffusion problem with nonlinear reaction source. The problem statement adds the convection-diffusion term to the standard Brusselator problem. This non-physical problem is much harder than the previous tests, both because the coupling equations through the reaction terms, and because the nonlinearity of this reaction terms. The problem without convection was solved in [23] using the conventional RKC. The governing system is

$$\begin{aligned} \frac{\partial u_1}{\partial t} + \mathbf{v}.\nabla u_1 - D\Delta u_1 &= 1 + u_1^2 u_2 - 4.4u_1 \,, \\ \frac{\partial u_2}{\partial t} + \mathbf{v}.\nabla u_2 - D\Delta u_2 &= 3.4u_1 - u_1^2 u_2 \,, \end{aligned} \tag{22}$$

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

$$u_1(0,x,y) = 1.5 + y, \qquad u_2(0,x,y) = 1 + 5x.$$

The diffusion coefficient D used is 2×10^{-3} and the velocity field $\mathbf{v}$ is given by

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

This problem was run on the unit square. In figure 3 we plot the behavior of the u_2 component using the ELM-IMEX scheme with $h = \frac{1}{64}$ and $\Delta t = 10^{-2}$. Some other accuracy results are listed in table 4. It is apparent that the mass is conserved in the ELM schemes presented in this paper.

Table 4
Results for the Brusselator with convection-diffusion at $t = 1.5$ with $D = 2 \times 10^{-3}$ and $\Delta t = 10^{-2}$. The CPU time is listed in minutes

	h	$u_{1,\min}$	$u_{1,\max}$	Relative mass	CPU
ELM-RK4	$\frac{1}{64}$	0.268	4.866	0.978	1.29
ELM-RKC	$\frac{1}{64}$	0.267	4.866	0.997	0.73
ELM-ROS3P	$\frac{1}{64}$	0.268	4.867	1.0	2.15
ELM-IMEX	$\frac{1}{64}$	0.268	4.867	1.0	6.27
ELM-RK4	$\frac{1}{128}$	—	—	—	—
ELM-RKC	$\frac{1}{128}$	0.266	4.863	0.999	2.93
ELM-ROS3P	$\frac{1}{128}$	0.267	4.864	1.0	8.31
ELM-IMEX	$\frac{1}{128}$	0.267	4.864	1.0	23.39

5 Concluding Remarks

In this paper we have selected several time integration schemes to be combined with a modified method of characteristics for solutions of advection-diffusion reaction problems. These schemes included fully explicit, fully implicit and implicit-explicit approaches which have been widely used in the method of lines to integrate the space semi-discretized partial differential equations. The main difference between method of lines and the Eulerian-Lagrangian methods

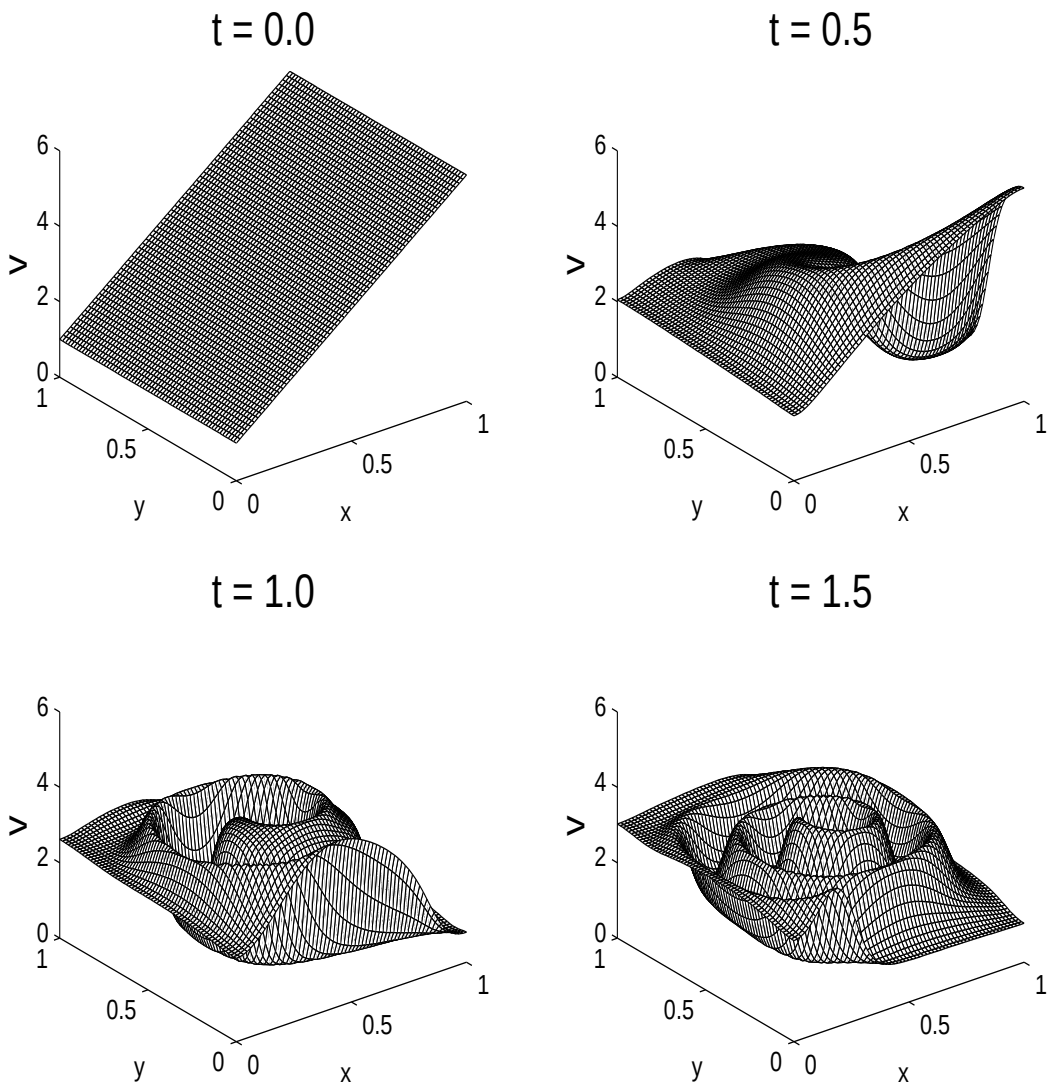

Fig. 3. The plots of the component u_2 obtained by ELM-IMEX scheme.

presented here is that the convective term that has to be treated carefully in method of lines has been moved from ELM methods by using the Lagrangian interpretation of the transport nature of the equations. This fact makes the ELM very attractive for the convection dominated flow problems where the steady state computations are needed and the time stepsize is restricted by the CFL conditions.

In order to show clearly the effects of time integration schemes for ELM computations we have modified the conventional ELM to decrease for instance, interpolation effects and spatial errors. Hence, we have used an ELM with adjusted advection to ensure conservation of mass at each time step, we also used high order (fourth order Runge-Kutta) scheme to calculate the departure points and high order interpolation (bicubic Lagrange) procedures. Furthermore, we have used high order space discretization (fourth order centred

difference). All these ingredients together are combined and turned into accurate time marching ELM scheme to tackle some difficulties in the numerical integration of the advection-diffusion reaction problems.

Numerical experiments have been carried out on two-dimensional advection-diffusion problems with linear and nonlinear reaction terms. Using different diffusion regimes and grid refinements, we have compared the accuracy and the efficiency of the schemes in terms of relative errors and CPU times.

Finally, we want to point out that due to the use of the Lagrangian coordinates, the ELM methods require more implementation work than Eulerian methods which are relatively easy to formulate and to implement.

Acknowledgements. The author would like to thank J. Lang for many stimulating discussions about the implementation of ROS3P scheme.

References

[1] U. Ascher, S. Ruuth, R. Spiteri, Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations. Appl. Numer. Math. **25** (1997) 151–167.

[2] A. Allievi, R. Bermejo, A Generalized Particle Search-Locate Algorithm for Arbitrary Grids, J. Comp. Physics, **132** (1992) 157–166.

[3] R. Bermejo, Analysis of a Class of Quasi-Monotone and Conservative Semi-Lagrangian Advection Schemes, Numer. Math. **87** (2001) 579–623.

[4] R. Bermejo, A Galerkin-Characteristic Algorithm for Transport- Diffusion equations, SAIM J. Numer. Anal. **32** (1995) 425–454.

[5] R. Bermejo, A. Staniforth, The Conversion of Semi-Lagrangian Advection Schemes to Quasi-Monotone Schemes, Month. Wea. Rev. **120** (1992) 2622–2632.

[6] C.N. Dawson, Godunov-Mixed Methods for Advective Flow Problems in One Space Dimension, SIAM J. Numer. Anal. **28** (1991) 1282–1309.

[7] J. Douglas, C. Huang, F. Pereira, The Modified Method of Characteristics with Adjusted Advection, Numer. Math. **83** (1999) 353–369.

[8] J. Douglas, 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) 871–885.

[9] U. Ghia, K.N. Ghia, C.T. Shin, High-Re Solutions for Incompressible Flow using the Navier-Stokes Equations and Multigrid Method, J. Comp. Phys. **48** (1982) 387–411.

[10] E. Hairer, S.P. Nørsett, G. Wanner, Solving Ordinary Differential Equations I, Nonstiff Problems, Springer Series in Computational Mathematics, Vol. 8, Berlin-Heidelberg-New York, 1987.

[11] S.V. Krisnamachari, L.J. Hayes, 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) 1462–1473.

[12] J. Lang, J. Verwer, ROS3P An Accurate Third-Order Rosenbrock Solver Designed for Parabolic Problems, BIT **41** (2001) 731–738.

[13] K.W. Morton, Numerical Solution of Convection-Diffusion Problems, Chapman & Hall, London, 1996.

[14] O. Pironneau, On the Transport-Diffusion Algorithm and its Applications to the Navier-Stokes Equations, Numer. Math. **38** (1982) 309–332.

[15] A. Robert, A stable Numerical Integration Scheme for the Primitive Meteorological Equations, Atmos. Ocean **19** (1981) 35–46.

[16] J.M. Sanz-Serna, A.M. Stuart, A Note Uniform in Time Error Estimates for Approximations to Reaction-Diffusion Equations, IMA J. Numer. Anal. **12** (1992) 457–462.

[17] J.M. Sanz-Serna, J.G. Verwer, W.H. Hundsdorfer, Convergence and Order Reduction of Runge-Kutta Schemes Applied to Evolutionary Problems in Partial Differential Equations, Numer. Math. **50** (1986) 405–418.

[18] M. Seaïd, On the Quasi-Monotone Modified Method of Characteristics for Transport-Diffusion Problems with Reactive Sources, Comp. Methods in App. Math. **2** (2002) 186–210.

[19] M. Seaïd, Semi-Lagrangian Integration Schemes for Viscous Incompressible Flows, Comp. Methods in App. Math. **2** (2002) 392–409.

[20] C. Shu, S. Osher, Efficient Implementation of Essentially Non-Oscillatory Shock Capturing Schemes, J. Comp. Physics **77** (1988) 439–471.

[21] A. Staniforth, J. Côté, Semi-Lagrangian Integration Schemes for the Atmospheric Models: A Review, Wea. Rev. **119** (1991) 2206–2223.

[22] C. Temperton, A. Staniforth, An Efficient Two-Time-Level Semi-Lagrangian Semi-Implicit Integration Scheme, Quart. J. Roy. Meteor. Soc. **113** (1987) 1025–1039.

[23] J.G. Verwer, W.H. Hundsdorfer, B.P. Sommeijer, Convergence Properties of the Runge-Kutta-Chebychev Method, Numer. Math. **57** (1990) 157–178.