

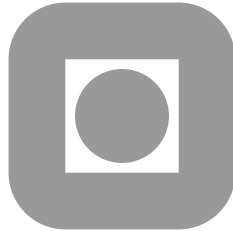
NORGES TEKNISK-NATURVITENSKAPELIGE
UNIVERSITET

**Semi-Lagrangian Runge–Kutta exponential
integrators for convection dominated problems**

by

E. Celledoni and B. K. Kometa

PREPRINT
NUMERICS NO. 9/2008



NORWEGIAN UNIVERSITY OF
SCIENCE AND TECHNOLOGY
TRONDHEIM, NORWAY

This report has URL

<http://www.math.ntnu.no/preprint/numerics/2008/N9-2008.pdf>

Address: Department of Mathematical Sciences, Norwegian University of Science and
Technology, N-7491 Trondheim, Norway.

Semi-Lagrangian Runge–Kutta exponential integrators for convection dominated problems

E. Celledoni and B. K. Kometa

September 23, 2008

We consider exponential integrators for convection-diffusion equations which can be implemented as semi-Lagrangian methods.

Semi-Lagrangian Runge-Kutta exponential integrators for convection dominated problems

Elena Celledoni and Bawfeh Kingsley Kometa

ABSTRACT. In this paper we consider the case of nonlinear convection-diffusion problems with a dominating convection term and we propose exponential integrators based on the composition of exact pure convection flows. The main reason for developing this type of methods is that as it turns out they can be applied to the numerical integration of the considered PDEs in a semi-Lagrangian fashion. Semi-Lagrangian methods perform well on convection dominated problems [Pi], [HE], [RM], [Ba].

In these methods linear convective terms are integrated exactly by first computing the characteristics corresponding to the gridpoints of the adopted discretization, and then producing the numerical approximation via an interpolation procedure.

Key words Additive Runge–Kutta methods, commutator-free methods, convection-diffusion equations, semi-Lagrangian methods

1. Introduction

The subject of this work is nonlinear convection-diffusion problems with a dominating convection term:

$$(1.1) \quad \frac{\partial}{\partial t} u(\mathbf{x}, t) + \mathbf{V} \cdot \nabla u(\mathbf{x}, t) = \nu \nabla^2 u + f(\mathbf{x}),$$

with $\mathbf{x} \in \Omega \subset \mathbb{R}^d$ and $\mathbf{V} : \mathbb{R}^d \times [0, T] \rightarrow \mathbb{R}^d$ is a given vector field, $u : \mathbb{R}^d \times [0, T] \rightarrow \mathbb{R}$, and $u(\mathbf{x}, 0) = u_0(\mathbf{x})$.

One of the popular models for the simulation of internal wave phenomena occurring between layers of stratified flow, consists of the Navier-Stokes equations with Boussinesq approximation. In this model the Navier-Stokes equations (which are themselves a suitable generalization of (1.1)) are coupled to a simpler convection-diffusion problem of the type (1.1), with u

1991 *Mathematics Subject Classification*. Primary 54C40, 14E20; Secondary 46E25, 20C20.

a conserved passive scalar, and $f(x) = 0$. The unknown u can represent temperature, or salt concentration in water, \mathbf{V} is the velocity field of the fluid coming from the numerical solution of the Navier-Stokes equations. In order to simulate properly the transport phenomena and the waves it is important to achieve the right balance between convection and diffusion, tuning appropriately ν over several experiments.

The case when the parameter ν is very small is particularly interesting and very challenging from the numerical point of view. In the Navier-Stokes equations this corresponds to the presence of high Reynolds numbers, as for example in the simulation of turbulent flows.

If we semidiscretize (1.1) in space with a finite element or finite difference method we obtain a system of ordinary differential equations of the type

$$(1.2) \quad y_t - C(y)y = Ay + f, \quad y(0) = y_0.$$

Here C is the discretized convection operator, A corresponds to the linear diffusion term, often negative definite.

A typical approach for solving numerically the semidiscretized equations is to treat convection and diffusion separately, the diffusion (as it might be stiff) with an implicit approach and the convection with an explicit integrator possibly of higher order, see for example [ARW], [CHQZ], [KC]. This approach has a big advantage, as most of the spatial discretizations of the diffusion operator give rise to matrices which are symmetric and negative definite, and implicit integration of the diffusion would require the solution of only symmetric positive definite linear algebraic systems.

When ν is small enough the diffusion term could in principle be treated explicitly as the CFL condition imposes the use of small time step-sizes anyway. In [XK] the authors point out that the use of low order semi-implicit methods in the direct numerical simulation of turbulent flows leads to prohibitive time step restrictions. In fact the time-step dictated by the CFL condition can be of several orders of magnitude smaller than the intrinsic temporal scale of the problem predicted by the theory. The CFL condition restrictions can be alleviated by the use of semi-Lagrangian schemes. In this case implicit integration for the diffusion term is usually adopted.

We present a new class of integration methods for convection-diffusion problems. These methods are exponential integrators and their peculiarity is that they allow for the computation of exponentials of the linearized convection term of (1.2). This means that terms of the form $W(h) \cdot g = \exp(\gamma h C(w)) \cdot g$, where w and g are fixed vectors in \mathbf{R}^n , and γ is a parameter of the integration method, appear as building blocks for the methods.

A simple example is the following first order integrator for (1.2),

$$(1.3) \quad y_{n+1} = \exp(hC(y_n))y_n + hAy_{n+1} + hf.$$

The goal here is to present and analyse higher order methods which share some of the features of (1.3). The methods in general need not be implicit in the diffusion part, but all the examples given in section 2 have this feature. The main reason for developing this type of methods is that as it turns out they can be applied to the numerical integration of the considered PDEs in a semi-Lagrangian fashion. In semi-Lagrangian methods [Pi], [HE], [RM], [Ba], linear convective terms are integrated *exactly* by first computing the characteristics corresponding to the gridpoints of the adopted discretization, and then producing the numerical approximation via an interpolation procedure.

The exponential $\exp(\gamma h C(w)) \cdot g$ is the solution of the semidiscretized equation

$$(1.4) \quad v' = C(w)v, \quad v(0) = g, \quad \text{in } [0, \gamma h],$$

which corresponds to the pure convection problem

$$(1.5) \quad \begin{aligned} \varphi_t + \mathbf{V} \cdot \nabla \varphi &= 0, & \varphi(x_i, 0) &= g_i, & \text{in } [0, \gamma h] \times \Omega, & \text{i.e.} \\ \frac{D\varphi}{Dt} &= 0, & \varphi(x_i, 0) &= g_i, & \text{in } [0, \gamma h] \times \Omega, \end{aligned}$$

where x_i are the nodes of the space discretization, $\mathbf{V} = \mathbf{V}(x_i) = w_i$, and $\frac{D}{Dt}$ is the total derivative. If we apply (1.3) directly to the problem (1.1) before discretizing in space, and we make the exponential $\exp(hC(y_n))y_n$ correspond to the exact integration of a pure convection problem, we obtain the following transport-diffusion algorithm studied in [Pi],

$$(1.6) \quad \begin{aligned} \frac{D\tilde{u}_{n+\frac{1}{2}}}{Dt} &= 0, & \tilde{u}_{n+\frac{1}{2}}(x, t_n) &= u_n(x), & \text{on } [t_n, t_n + h] \\ u_{n+\frac{1}{2}}(x) &:= \tilde{u}_{n+\frac{1}{2}}(x, t_n + h) \\ u_{n+1} &= u_{n+\frac{1}{2}} + h\nu\nabla^2 u_{n+\frac{1}{2}} + hf, \end{aligned}$$

the convecting vector field is $\mathbf{V}(x) = u_n(x)$. The accurate integration of the pure convection problem in (1.6) can be obtained by introducing characteristics, we get

$$(1.7) \quad \begin{aligned} u_{n+\frac{1}{2}}(x) &= \tilde{u}_{n+\frac{1}{2}}(x, t_n + h) = u_n(X(t_n)) \\ \frac{dX}{d\tau} &= u_n(X(\tau)), & X(t_n + h) &= x, \end{aligned}$$

and the equation for the characteristics $X(\tau)$ must be integrated backwards in time, either exactly or with a suitable numerical integrator. Now substituting the result in the second equation of (1.6) we have

$$(1.8) \quad \begin{aligned} \frac{dX}{d\tau} &= u_n(X(\tau)), & X(t_n + h) &= x, \\ u_{n+1} &= u_n(X(t_n)) + h\nu\nabla^2 u_{n+\frac{1}{2}} + hf. \end{aligned}$$

The outlined correspondence between (1.3) and (1.8) motivates the use of exponentials of C (the semidiscrete convection operator of (1.2)) in the integrators.

The presented methods are of additive type. We derive methods up to and including order three. The convection part is treated by a commutator-free explicit method requiring the composition of exponentials of C [**CMO**], the linear diffusion with a Runge-Kutta (implicit) method. Additive Runge-Kutta methods can be cast into the class of partitioned Runge-Kutta method and their order conditions can be studied in this setting [**HW**], [**KC**]. In particular applying two Runge-Kutta methods of order p to two different parts of the vector field of the problem, does not guarantee order p overall, extra coupling conditions must be satisfied by the coefficients of the two methods.

In [**O**] the order theory for commutator free methods is derived: these methods are a generalization of Runge-Kutta methods based on composition of simplified exact flows, see also [**CG**], and require additional order conditions compared to the classical case. In principle one should expect that the combination of commutator-free methods and Runge-Kutta methods in an additive fashion could bring to extra (coupling) order conditions. Our preliminary analysis shows however that under some standard assumptions on the coefficients of the partitioned method, this is not the case for order one, two and three. For low order the conditions are a subset of the union of classical order conditions for partitioned Runge-Kutta methods and commutator-free methods. As a consequence it is quite easy to derive examples of methods in the new class by taking known IMEX methods as a starting point. New coupling conditions, involving the coefficients of the commutator free methods, do appear at order four, [**CK**].

The semi-Lagrangian methods here presented are based on Runge-Kutta one step strategies and are for this reason an alternative to the semi-Lagrangian multistep approaches proposed for example in [**XK**]. Previous preliminary work has also been presented in [**Ce**].

In section 2 we present the new methods, discuss the order conditions and give some concrete examples. In section 3 we illustrate the performance of the methods implementing them as semi-Lagrangian schemes.

2. Presentation of the new class of methods

We consider the semidiscretized ordinary differential equation

$$(2.1) \quad \dot{y} - C(y)y = Ay, \quad y(0) = y_0,$$

where for the sake of simplicity, but without loss of generality we have dropped the forcing term f of (1.2). Here A is a $n \times n$ matrix with constant coefficients and $C(y)$ is a matrix valued function, also $n \times n$.

The general format for the methods here studied is

for $i = 1 : s$ do

$$\begin{aligned} Y_i &= \varphi_i y_n + h \sum_{j=1}^s a_{i,j} \varphi_{i,j} A Y_j, \\ \varphi_i &= \exp(h \sum_k \alpha_{i,J}^k C(Y_k)) \cdots \exp(h \sum_k \alpha_{i,1}^k C(Y_k)), \\ \varphi_{i,j} &:= \varphi_i \varphi_j^{-1} \end{aligned}$$

end

$$\begin{aligned} y_{n+1} &= \varphi_{n+1} y_n + h \sum_{i=1}^s b_i \varphi_{n+1,i} A Y_i, \\ \varphi_{n+1} &= \exp(h \sum_k \beta_J^k C(Y_k)) \cdots \exp(h \sum_k \beta_1^k C(Y_k)), \\ \varphi_{n+1,i} &:= \varphi_{n+1} \varphi_i^{-1} \end{aligned}$$

In the study of the order conditions it is convenient to treat the numerical solution y_{n+1} as an extra stage value

$$Y_{s+1} := \varphi_{s+1} y_n + h \sum_{j=1}^s a_{s+1,j} \varphi_{s+1,j} A Y_j, \quad a_{s+1,j} := b_j,$$

with

$$\varphi_{s+1} = \exp(h \sum_k \alpha_{s+1,J}^k C(Y_k)) \cdots \exp(h \sum_k \alpha_{s+1,1}^k C(Y_k)), \quad \alpha_{s+1,J-l}^k := \beta_{J-l}^k,$$

and with $\varphi_{s+1,j} = \varphi_{s+1} \varphi_j^{-1}$.

For reasons of computational ease in the implementation of the methods, we are particularly interested in the case where the coefficients $a_{i,j}$ define a diagonally implicit method (DIRK), so that $a_{i,j} = 0$ when $j > i$. In this case the methods require the solution of only one linear system per stage. The discussion on order conditions is not affected by this choice.

The presented methods can be directly interpreted as transport-diffusion algorithms for convection-diffusion equations.

2.1. Order conditions. Let us define the Runge-Kutta coefficients

$$(2.2) \quad \hat{a}_{i,j} := \sum_{l=1}^J \alpha_{i,l}^j, \quad \hat{b}_j := \sum_{l=1}^J \beta_l^j,$$

for $i = 1, \dots, s$.

The new methods are then associated to the two Butcher tableaus,

$$(2.3) \quad \begin{array}{c|c} \mathbf{c} & \mathcal{A} \\ \hline & \mathbf{b} \end{array}, \quad \begin{array}{c|c} \hat{\mathbf{c}} & \hat{\mathcal{A}} \\ \hline & \hat{\mathbf{b}} \end{array},$$

the coefficients of the first tableau are used for the linear vector field Ay while the coefficients of the second tableau, via (2.2), are used for the nonlinear vector field $C(y)y$.

Assuming enough regularity of the solution of (2.1), expanding the exact and the numerical solution with Taylor series and matching the terms one obtains the order conditions reported in tables 4, 5.

In the tables we report the elementary differentials appearing in the Taylor expansions, and the corresponding bicoloured binary trees, related to each order condition. Black nodes (\bullet) refer to the vector field $C(y)y$ and white nodes (\circ) to Ay .

REMARK 2.1. Let us consider the following change of variables in (2.1)

$$(2.4) \quad y(t) = W(t) \cdot z(t),$$

with

$$(2.5) \quad W_t = C(Wz)W, \quad W(0) = I.$$

After differentiation and substituting for y_t in the right hand side of (2.1), we obtain the following system of equations equivalent to (2.1)

$$(2.6) \quad \begin{cases} z_t &= W^{-1}AWz, & z(0) = y_0, \\ W_t &= C(Wz)W, & W(0) = I. \end{cases}$$

We apply a partitioned Runge-Kutta approach to this system combining an implicit Runge-Kutta method for the equation for z and an explicit commutator-free method for the equation for W , to obtain

for $i = 1 : s$ do

$$Z_i = z_n + h \sum_{j=1}^i a_{i,j} Q_j^{-1} A Q_j Z_j$$

$$Q_i = \exp(\sum_k \alpha_{i,j}^k C(Q_k Z_k)) \cdots \exp(\sum_k \alpha_{i,1}^k C(Q_k Z_k)) \cdot W_n$$

end

$$z_{n+1} = z_n + \sum_{i=1}^s b_i W_i^{-1} A Q_i Z_i$$

$$W_{n+1} = \exp(\sum_k \beta_j^k C(Q_k Z_k)) \cdots \exp(\sum_k \beta_1^k C(Q_k Z_k)) \cdot W_n$$

Using the following transformation

$$(2.7) \quad Y_i := Q_i Z_i, \quad \varphi_i := Q_i W_n^{-1}, \quad y_{n+1} = W_{n+1} z_{n+1}, \quad \varphi_{n+1} = W_{n+1} W_n^{-1},$$

we obtain the original format of the methods. Since $Q_i Q_j^{-1} = \varphi_i \varphi_j^{-1}$ than $\varphi_{i,j} := \varphi_i \varphi_j^{-1}$ arises naturally in the expression for Y_i .

2.2. Derivation of the order conditions.

2.2.1. *Derivatives of the exact solution.* The derivatives of the exact solution can be written in the form

$$y^{(q)} = \sum_{k=0}^{q-1} \binom{q-1}{k} \left(\frac{d^{q-1-k}}{dh^{q-1-k}} (C(y) + A) \right) y^{(k)},$$

and therefore

$$\begin{aligned} y^{(1)}\Big|_{h=0} &= \dot{y}\Big|_{h=0} = C(y_0)y_0 + Ay_0, \\ y^{(2)}\Big|_{h=0} &= C'(y_0)(\dot{y}(0))y_0 + (C(y_0) + A)^2y_0. \end{aligned}$$

Here we have used the matrix $C'(y)(w)$ obtained by differentiating $C(y)$,

$$(C'(y)(w))_{i,j} := \sum_{k=1}^N \frac{\partial c_{i,j}}{\partial x_k}(y)w_k,$$

and $c_{i,j} := (C(y))_{i,j}$, and N is the number of components of y .

In short we will write $C, C'(\cdot)$ for $C(y_0), C'(y_0)(\cdot)$ respectively.

2.2.2. Derivatives of the numerical solution. We assume without loss of generality that $n = 0$.

We obtain

$$\dot{Y}_{s+1}\Big|_{h=0} = \dot{\varphi}_{s+1}\Big|_{h=0} y_0 + \sum_{j=1}^s a_{s+1,j} Ay_0$$

In general we have

$$\begin{aligned} (2.8) \quad Y_i^{(q)}\Big|_{h=0} &= \varphi_i^{(q)}\Big|_{h=0} y_0 + q \frac{d^{q-1}}{dh^{q-1}} \left(\sum_j a_{i,j} \varphi_i \varphi_j^{-1} AY_j \right)\Big|_{h=0} \\ &= \varphi_i^{(q)}\Big|_{h=0} y_0 + q \sum_j a_{i,j} \sum_{k=0}^{q-1} \binom{q-1}{k} \varphi_{i,j}^{(q-1-k)} AY_j^{(k)}\Big|_{h=0}, \end{aligned}$$

where $\varphi_{i,j} := \varphi_i \varphi_j^{-1}$, $i, j = 1, \dots, s$.

The equations

$$Y_{s+1}^{(q)}\Big|_{h=0} = y^{(q)}\Big|_{h=0}, \quad q = 0, 1, \dots, p$$

give the order conditions for order p , and for $p = 1, 2, 3$ are reported in table 4 and table 5.

The recursive computation of the derivatives of Y_i requires the use of φ_i and $\varphi_{i,j}$ and their derivatives which we will now discuss. Consider the following matrix-valued functions,

$$C_{i,J-l} := h \sum_k \alpha_{i,J-l}^k C(Y_k), \quad l = 0, \dots, J-1,$$

and

$$\tilde{C}_{i,J-l} := -h \sum_k \alpha_{i,l+1}^k C(Y_k), \quad l = 0, \dots, J-1.$$

We will denote with $B_{i,J-l}$, $l = 0, \dots, J-1$ either of $C_{i,J-l}$ or $\tilde{C}_{i,J-l}$, and consider

$$\psi_i := \exp(B_{i,J}) \cdot \exp(B_{i,J-1}) \cdots \exp(B_{i,1}).$$

For the two choiches of $B_{i,J-l}$ we have

$$\psi_i = \begin{cases} \varphi_i, & B_{i,J-l} = C_{i,J-l}, l = 0, \dots, J-1 \\ \text{and} & \\ \varphi_i^{-1} & B_{i,J-l} = \tilde{C}_{i,J-l}, l = 0, \dots, J-1, \end{cases}$$

respectively. We will also make use of

$$\psi_i^l := \exp(B_{i,J}) \cdot \exp(B_{i,J-1}) \cdots \exp(B_{i,J-l}).$$

We obtain¹

$$\dot{\psi}_i = \sum_{l=0}^{J-1} \text{Ad}_{\psi_i^l} \left(\text{dexp}_{-B_{i,J-l}}(\dot{B}_{i,J-l}) \right) \cdot \psi_i,$$

so

$$\dot{\psi}_i = S_i(h)\psi_i, \quad S_i(h) := \sum_{l=0}^{J-1} \text{Ad}_{\psi_i^l} \left(\text{dexp}_{-B_{i,J-l}}(\dot{B}_{i,J-l}) \right),$$

and as a direct consequence we have

$$(2.9) \quad \psi_i^{(r)} = \sum_{k=0}^{r-1} \binom{r-1}{k} \left(\frac{d^{r-1-k}}{dh^{r-1-k}} S_i(h) \right) \psi_i^{(k)}.$$

The following proposition will be used to find the derivatives of $S_i(h)$.

PROPOSITION 2.2. *Given $Z^0 = Z^0(h)$, $W = W(h)$ two matrix-valued differentiable functions then*

$$(2.10) \quad \frac{d^m}{dh^m} \text{Ad}_W Z^0 = \text{Ad}_W Z^r,$$

with

$$(2.11) \quad Z^r = [W^{-1}\dot{W}, Z^{r-1}] + \dot{Z}^{r-1}.$$

PROOF. By induction. □

From (2.11) by differentiation we obtain

$$(2.12) \quad \dot{Z}^r = [W^{-1}\dot{W} + W^{-1}\ddot{W}, Z^{r-1}] + [W^{-1}\dot{W}, \dot{Z}^{r-1}] + \ddot{Z}^{r-1},$$

¹We recall that $\text{dexp}_w(u) := \left. \frac{e^z - 1}{z} \right|_{z=\text{ad}_w} (u) = u + 1/2![w, u] + 1/3![w[w, u]] + \dots$ and $\text{ad}_w(u) := [w, u]$ (matrix commutator between w and u), $\text{Ad}_\psi(u) := \psi u \psi^{-1}$.

and using (2.11) and (2.12) and assuming $W(0) = I$, we obtain

$$(2.13) \quad \begin{aligned} \frac{d}{dh} \text{Ad}_W Z^0 \Big|_{h=0} &= Z^1(0) = [\dot{W}(0), Z^0(0)] + \dot{Z}^0(0) \\ &\quad \dot{Z}^1(0) = [-\dot{W}(0)^2 + \ddot{W}(0), Z^0(0)] + [\dot{W}(0), \dot{Z}^0(0)] + \ddot{Z}^0(0) \\ \frac{d^2}{dh^2} \text{Ad}_W Z^0 \Big|_{h=0} &= Z^2(0) = [\dot{W}(0), Z^1(0)] + \dot{Z}^1(0) \\ &\quad \dot{Z}^2(0) = [-\dot{W}(0)^2 + \ddot{W}(0), Z^1(0)] + [\dot{W}(0), \dot{Z}^1(0)] + \ddot{Z}^1(0) \\ \frac{d^3}{dh^3} \text{Ad}_W Z^0 \Big|_{h=0} &= Z^3(0) = [\dot{W}(0), Z^2(0)] + \dot{Z}^2(0) \\ &\quad \vdots \end{aligned}$$

Further assuming $Z^0 = \text{dexp}_{-B}(\dot{B})$ for some matrix-valued differentiable function $B = B(h)$, expanding the right hand side and differentiating we obtain

$$(2.14) \quad \begin{aligned} Z^0(0) &= \dot{B}(0) \\ \dot{Z}^0(0) &= \ddot{B}(0) \\ \ddot{Z}^0(0) &= \dddot{B}(0) - \frac{1}{2}[\dot{B}(0), \ddot{B}(0)] \\ \ddot{\ddot{Z}}^0(0) &= B^{IV}(0) - [\dot{B}(0), \ddot{\ddot{B}}(0)] + \frac{1}{2}[\dot{B}(0), [\dot{B}(0), \ddot{B}(0)]] \\ &\quad \vdots \end{aligned}$$

We can now obtain the derivatives of S_i and ψ_i . By setting $W = \psi_i^l$ and $B = B_{i,J-l}$ we can calculate the derivatives of S_i using (2.10). We obtain

$$(2.15) \quad \begin{aligned} S_i(0) &= \sum_{l=0}^{J-1} \dot{B}_{i,J-l}, \\ \frac{d}{dh} S_i \Big|_{h=0} &= \sum_{l=0}^{J-1} \sum_{r=0}^{J-l-1} [\dot{B}_{i,J-r}(0), \dot{B}_{i,J-l}(0)] + \ddot{B}_{i,J-l}(0), \\ \frac{d^2}{dh^2} S_i \Big|_{h=0} &= 2 \sum_{l=0}^{J-1} \sum_{r=0}^{J-l-1} [\dot{B}_{i,J-r}(0), \ddot{B}_{i,J-l}(0)] + \\ &\quad \sum_{l=0}^{J-1} \sum_{r=0}^{J-l-1} [\ddot{B}_{i,J-r}(0), \dot{B}_{i,J-l}(0)] + \\ &\quad \sum_{l=0}^{J-1} \left(\ddot{\ddot{B}}_{i,J-l}(0) - \frac{1}{2}[\dot{B}_{i,J-l}(0), \ddot{\ddot{B}}_{i,J-l}(0)] \right), \end{aligned}$$

analogously the derivatives of $S_i^l = \sum_{r=0}^{J-l-1} \text{Ad}_{\psi_i^r}(\text{dexp}_{-B_{i,J-r}} \dot{B}_{i,J-r})$ are obtained as in the previous formulae but substituting $J-1$ as upper index in the external summation with $J-l-1$.

In table 1 we report the values of the derivatives of φ_i and φ_j^{-1} at $h=0$, these are obtained from (2.9) using (2.15). In table 2 we report the derivatives at 0 of $\varphi_{i,j}$, which are obtained using table 1 and

$$\varphi_{i,j}^{(m)} = \sum_{r=0}^m \binom{m}{r} \varphi_i^{(m-r)} (\varphi_j^{-1})^{(r)}.$$

The derivatives of Y_i are reported in table 3 and are obtained using tables 1 and 2 and the recursion formula (2.8).

q	$\varphi_i^{(q)}(0)$	$(\varphi_j^{-1})^{(q)}(0)$
0	I	I
1	$C\hat{c}_i$	$-C\hat{c}_j$
2	$2\sum_k \hat{a}_{i,k} C'(\hat{c}_k C y_0 + c_k A y_0) + \hat{c}_i^2 C^2$	$-2\sum_k \hat{a}_{j,k} C'(\hat{c}_k C y_0 + c_k A y_0) + \hat{c}_j^2 C^2$

TABLE 1. Derivatives of φ_i and its inverse.

q	$\varphi_{i,j}^{(q)}(0)$
0	I
1	$C(\hat{c}_i - \hat{c}_j)$
2	$2\sum_k (\hat{a}_{i,k} - \hat{a}_{j,k}) C'(\hat{c}_k C y_0 + c_k A y_0) + (\hat{c}_i - \hat{c}_j)^2 C^2$

TABLE 2. Derivatives of $\varphi_{i,j}$.

q	$Y_i^{(q)}(0)$
0	y_0
1	$(\sum_j \hat{a}_{i,j}) C y_0 + (\sum_j a_{i,j}) A y_0$
2	$2\sum_j \hat{a}_{i,j} C'(\hat{c}_j C + c_j A) + \hat{c}_i^2 C^2 + 2\hat{c}_i c_i C A - 2(\sum_j a_{i,j} \hat{c}_j) C A + 2\sum_j a_{i,j} A(\hat{c}_j C + c_j A)$

TABLE 3. Derivatives of Y_i at 0.

We here derive the first few rows of table 1, table 2 and table 3. From (2.9) we obtain that

$$(2.16) \quad \dot{\varphi}_i(0) = \sum_{l=0}^{J-1} \dot{B}_{i,J-l}(0) = \sum_{l=0}^{J-1} \dot{C}_{i,J-l}(0) = \sum_{l=0}^{J-1} \sum_k \alpha_{i,J-l}^k C = \left(\sum_k \hat{a}_{i,k} \right) C,$$

analogously one can compute $\varphi_j^{-1}(0)$. These expressions are reported in table 1 and can be used in (2.8) to obtain

$$(2.17) \quad \dot{Y}_i \Big|_{h=0} = \left(\sum_{k=1}^s \hat{a}_{i,k} \right) C y_0 + \left(\sum_{j=1}^s a_{i,j} \right) A y_0 = \hat{c}_i C y_0 + c_i A y_0,$$

see table 3. Imposing $\dot{Y}_{s+1} \Big|_{h=0} = \dot{y}(0)$, where $\dot{y}(0) = C y_0 + A y_0$ we obtain the following order conditions for order 1,

$$\sum_{k=1}^s \hat{a}_{s+1,k} = 1, \quad \sum_{j=1}^s a_{s+1,j} = 1.$$

These correspond to requiring consistency of the two Runge-Kutta methods (2.3).

For deriving the conditions for order two we use (2.8) and we get

$$\ddot{Y}_i \Big|_{h=0} = \varphi_i^{(2)} \Big|_{h=0} y_0 + 2 \sum_j a_{i,j} \left(\varphi_{i,j}^{(1)}(0) A Y_j(0) + \varphi_{i,j}(0) A \dot{Y}_j(0) \right)$$

with $\varphi_{i,j}(0) = I$ and $\varphi_{i,j}^{(1)}(0) = \dot{\varphi}_i(0) - \dot{\varphi}_j(0) = (\hat{c}_i - \hat{c}_j)C$. Using $\dot{Y}_j(0)$ from table 3 we obtain

$$(2.18) \quad \ddot{Y}_i \Big|_{h=0} = \varphi_i^{(2)} \Big|_{h=0} y_0 + 2 \sum_j a_{i,j} ((\hat{c}_i - \hat{c}_j) C A y_0 + \hat{c}_j A C y_0 + c_j A^2 y_0).$$

From (2.9) and (2.15) we obtain

$$(2.19) \quad \begin{aligned} \varphi_i^{(2)} \Big|_{h=0} &= \frac{d S_i(h)}{d h} \varphi_i(h) \Big|_{h=0} + (S_i(h)^2 \varphi_i(h)) \Big|_{h=0} \\ &= 2 \sum_k \hat{a}_{i,k} \hat{c}_k C' (C y_0) + 2 \sum_k \hat{a}_{i,k} c_k C' (A y_0) + \left(\sum_{j=1}^s \hat{a}_{i,j} \right)^2 C^2, \end{aligned}$$

which is reported in table 1. Substituting the results in (2.18) we obtain

$$(2.20) \quad \begin{aligned} \ddot{Y}_i \Big|_{h=0} &= (2 \sum_k \hat{a}_{i,k} \hat{c}_k C' (C y_0) + 2 \sum_k \hat{a}_{i,k} c_k C' (A y_0) + \hat{c}_i^2 C^2) y_0 + \\ &\quad 2 \sum_j a_{i,j} ((\hat{c}_i - \hat{c}_j) C A y_0 + \hat{c}_j A C y_0 + c_j A^2 y_0), \end{aligned}$$

and report it in table 3.

Taking $i = s+1$ and matching this result with $y^{(2)} \Big|_{h=0} = C'(\dot{y}(0))y_0 + (C + A)^2 y_0$, we obtain the four conditions for order two

$$\begin{aligned} \sum_j a_{s+1,j} \hat{c}_j &= \frac{1}{2}, & \sum_j \hat{a}_{s+1,j} c_j &= \frac{1}{2}, \\ \sum_j a_{s+1,j} c_j &= \frac{1}{2}, & \sum_j \hat{a}_{s+1,j} \hat{c}_j &= \frac{1}{2}. \end{aligned}$$

The conditions for order 3 are obtained in a similar way, see [CK].

It is easy to check, see for example [HW], that the conditions for order two coincide with the conditions of partitioned Runge-Kutta methods of the same order for the tableaux (2.3). Under the assumption that $c_k = \hat{c}_k$, the







condition	trees	elementary differential
order 1		
$\sum_i \hat{b}_i = 1$		C
$\sum_i b_i = 1$		A
order 2		
$2 \sum_i \hat{b}_i \hat{c}_i = 1$		$C'(C)$
$2 \sum_i \hat{b}_i c_i = 1$		$C'(A)$
$2 \sum_i b_i \hat{c}_i = 1$		AC
$2 \sum_i b_i c_i = 1$		A^2

TABLE 4. Conditions of order 1 and 2.

conditions for order three (see also [CK]) include a subset of the classical conditions of order 3 for partitioned RK methods for (2.3). Due to the linearity of A one of the classical elementary differentials does not appear. If we assume $c_k = \hat{c}_k$ the remaining conditions of order three (the first two in table 5) coincide with the conditions for commutator-free methods of order three, [CMO] and [O].

In the tables we have associated to each order condition its corresponding elementary differential and its corresponding bicolored rooted tree. We have used trees τ with $|\tau|+1$ nodes for representing the differentials of order $|\tau|$ in order to include the commutators between lower order differentials, as done in [O]¹.

REMARK 2.3. We consider the system (2.4). Xiu et al. in [XK] considered splittings of the Navier-Stokes equations into a convection and a diffusion problem and studied a semi-Lagrangian version of the partitioned combination: the second order Adams-Basforth and the Crank-Nicolson scheme (midpoint rule) (AB2-CN). This method does not achieve second order as a time integrator. Instead they proposed and proved numerically that combining a semi-Lagrangian scheme for the nonlinear convection term (linearized via a second order extrapolation scheme) together with the implicit midpoint rule for the remaining terms, would result in a scheme with overall second order accuracy in time. We have verified their observations in our framework by applying a CF version of the second order Adams-Basforth to the first equation of (2.4) (convection part) and the implicit trapezoidal rule to the second equation of (2.4) (diffusion part) of the nonlinear convection-diffusion problem. Assuming that W_{n-1}, W_n are known,

¹Recall the notation: if τ is a rooted tree $|\tau|$ is the number of nodes of the tree.

condition	trees	elementary differential
$4 \sum_{l=0}^{J-1} \sum_{r=0}^l \sum_k \beta_{J-r}^k \sum_m \beta_{J-l}^m \hat{c}_m +$ $-2 \sum_{l=0}^{J-1} \sum_{r=0}^l \sum_k \beta_{J-r}^k \sum_m \beta_{J-l}^m \hat{c}_k +$ $- \sum_{l=0}^{J-1} \sum_k \beta_{J-l}^k \sum_r \beta_{J-l}^r \hat{c}_r = 0$		$[C'(C), C]$
$4 \sum_{l=0}^{J-1} \sum_{r=0}^l \sum_k \beta_{J-r}^k \sum_m \beta_{J-l}^m c_m +$ $-2 \sum_{l=0}^{J-1} \sum_{r=0}^l \sum_k \beta_{J-r}^k \sum_m \beta_{J-l}^m c_k +$ $- \sum_{l=0}^{J-1} \sum_k \beta_{J-l}^k \sum_r \beta_{J-l}^r c_r = 0$		$[C'(A), C]$
$3 \sum_k \hat{b}_k \hat{c}_k^2 = 1$		$C''(C, C)$
$3 \sum_k \hat{b}_k \hat{c}_k c_k = 1$		$C''(C, A)$
$3 \sum_k \hat{b}_k c_k^2 = 1$		$C''(A, A)$
$6 \sum_k \hat{b}_k \sum_j \hat{a}_{k,j} \hat{c}_j = 1$		$C'(C'(C))$
$6 \sum_k \hat{b}_k \sum_j \hat{a}_{k,j} c_j = 1$		$C'(C'(A))$
$6 \sum_k \hat{b}_k \sum_j a_{k,j} \hat{c}_j = 1$		$C'(AC)$
$6 \sum_k \hat{b}_k \sum_j a_{k,j} c_j = 1$		$C'(A^2)$
$6 \sum_j b_j c_j - 6 \sum_j b_j \hat{c}_j c_j = 1$		CA^2
$6 \sum_j b_j \sum_m \hat{a}_{j,m} \hat{c}_m = 1$		$AC'(C)$
$6 \sum_j b_j \sum_m \hat{a}_{j,m} c_m = 1$		$AC'(A)$
$3 \sum_j b_j \hat{c}_j^2 = 1$		AC^2
$6 \sum_j b_j \sum_m a_{j,m} \hat{c}_m = 1$		A^2C
$6 \sum_j b_j \sum_m a_{j,m} c_m = 1$		A^3

TABLE 5. Conditions of order 3.

from the first equation of (2.4) we get

$$(2.21) \quad W_{n+1} = \exp\left(\frac{h}{2}(3C(y_n) - C(y_{n-1}))\right)W_n$$

which agrees with the semi-Lagrangian method used in [XK], but in place of the midpoint rule, we apply the implicit trapezoidal rule for the diffusion

term to obtain

$$(2.22) \quad z_{n+1} = z_n + \frac{h}{2}(W_n^{-1}AW_n z_n + W_{n+1}^{-1}AW_{n+1} z_{n+1}).$$

Transforming back to the variable $y = Wz$ and assuming y_{n-1} and y_n are known apriori we have the following scheme

$$(2.23) \quad y_{n+1} = E_n y_n + \frac{h}{2}(E_n A y_n + A y_{n+1}), \text{ where } E_n = \exp\left(\frac{h}{2}(3C(y_n) - C(y_{n-1}))\right)$$

which is tested numerically to be second order accurate in time. This has also been verified directly by Taylor expansion. A direct translation of the AB2-CN scheme into our semi-Lagrangian method results in

$$(2.24) \quad y_{n+1} = E_n y_n + \frac{h}{2}A(E_n y_n + y_{n+1})$$

which is tested to be only first order accurate in time. Further investigations on exponential integrators based on explicit multi-step formulae, combining the ideas of the present paper to the results of [CG] will give rise to a systematic order theory for methods like (2.23) and will be addressed elsewhere.

The incompressibility constraint of the Navier-Stokes equations leads to further complications in the design of high order numerical integrators, the reason being that after discretization in space these equations are not ordinary differential equations, but index 2 differential algebraic equations. For a discussion on these issues see for example [ZP].

2.3. Methods of order 2 and 3. The preliminary analysis of the order conditions reported in the previous section leads to the following conclusions:

- Any couple of classical RK methods of order 1 gives a new method of order 1.
- A couple of partitioned RK methods of order 2. gives a new method of order 2 .
- If we take a pair of PRK of order 3 (explicit + implicit) and construct a commutator-free method out of the explicit method in such a way that

$$(2.25) \quad \hat{b}_k = \sum_{l=0}^J \beta_{J,l}^k, \quad \hat{a}_{k,j} = \sum_{l=0}^J \alpha_{J,l}^j,$$

the resulting method satisfies the conditions for order 3 for the new class of methods.

We will use these observations to obtain methods up to and included order three in the new class of methods proposed in this paper. Our starting point are the IMEX methods of [ARW] and other partitioned Runge-Kutta

method (additive) methods reported in [HW]. In all these methods $b_i = \hat{b}_i$ and $c_i = \hat{c}_i$ for $i = 1, \dots, s$.

In particular it is advantageous to consider methods requiring a minimal number of exponentials per stage, i.e. minimal J , [CMO]. It is possible to obtain methods of order one and two with $J = 1$. For order three J must be at least 2. This implies that the methods of order one and two proposed in [ARW] can be used directly to obtain methods of the same order within the new class, while for order three we need to split up the coefficients of the explicit method according to (2.25) and also consider the extra (commutator-free) order condition from table 5 (first and second row, coinciding when we assume $c_k = \hat{c}_k$).

From the first two conditions of order three in table 5, setting $J = 2$ and using the conditions of order 1 and 2 one obtains

$$(2.26) \quad \sum_{k=1}^s (\beta_2^k + 2\beta_1^k c_k) = \frac{2}{3}.$$

EXAMPLE 2.4. The couple of methods with tableaus

$$\begin{array}{c|cc} 0 & & \\ \frac{1}{2} & 0 & \frac{1}{2} \\ \hline & 0 & 1 \end{array}, \quad \begin{array}{c|ccc} 0 & & & \\ \frac{1}{2} & \frac{1}{2} & 0 & \\ \hline & 0 & 1 & \end{array},$$

[ARW], gives raise to the new method of order two

$$\begin{array}{lcl} \varphi_{\frac{1}{2}} & = & \exp(\frac{h}{2}C(y_0)) \quad Y_{\frac{1}{2}} = \varphi_{\frac{1}{2}}y_0 + \frac{h}{2}AY_{\frac{1}{2}} \\ \varphi_1 & = & \exp(hC(Y_{\frac{1}{2}})) \quad y_1 = \varphi_1y_0 + h\varphi_1\varphi_{\frac{1}{2}}^{-1}AY_{\frac{1}{2}} \end{array}.$$

The corresponding transport-diffusion algorithm for convection-diffusion problems of type (1.1) has the form

$$\begin{aligned}
(2.27) \quad \frac{D\tilde{u}_0}{Dt} &= 0, \quad \tilde{u}_0(x, 0) = u_0(x), \quad \text{on } [0, \frac{h}{2}], \\
&\quad \mathbf{V}(x) = u_0(x), \\
&\quad \tilde{u}_0(x) = \tilde{u}_0(x, \frac{h}{2}), \\
u_{\frac{1}{2}} &= \tilde{u}_0 + \frac{h}{2}\nu\nabla^2 u_{\frac{1}{2}}, \\
\frac{D\tilde{u}_{\frac{1}{2}}}{Dt} &= 0, \quad \tilde{u}_{\frac{1}{2}}(x, 0) = \nu\nabla^2 u_{\frac{1}{2}}(x), \quad \text{on } [-\frac{h}{2}, 0], \\
&\quad \mathbf{V}(x) = u_0(x), \\
&\quad u_{\frac{1}{2}}(x) = \tilde{u}_{\frac{1}{2}}(x, -\frac{h}{2}), \\
\frac{D\tilde{u}_1}{Dt} &= 0, \quad \tilde{u}_1(x, 0) = u_0(x) + hu_{\frac{1}{2}}(x), \quad \text{on } [0, h], \\
&\quad \mathbf{V}(x) = u_{\frac{1}{2}}(x), \\
&\quad u_1(x) = \tilde{u}_1(x, h).
\end{aligned}$$

EXAMPLE 2.5. In the next example we consider a second order IMEX method from [ARW], we implement this method in the numerical experiments as an IMEX or a DIRK commutator-free method obtaining in both cases order 2:

$$\begin{array}{c|ccc} 0 & & & \\ \gamma & \gamma & & \\ 1 & \delta & 1-\delta & \\ \hline & 0 & 1-\gamma & \gamma \end{array}, \quad \begin{array}{c|ccc} 0 & 0 & & \\ \gamma & 0 & \gamma & \\ 1 & 0 & 1-\gamma & \gamma \\ \hline & 0 & 1-\gamma & \gamma \end{array}$$

with $\gamma = \frac{2-\sqrt{2}}{2}$ and $\delta = \frac{-2\sqrt{2}}{3}$. The implicit method is stiffly accurate and L-stable.

EXAMPLE 2.6. For obtaining order 3 we consider as starting point the partitioned Runge-Kutta methods proposed by Griepentrog [HW],

$$\begin{array}{c|ccc} 0 & & & \\ \frac{1}{2} & \frac{1}{2} & & \\ 1 & -1 & 2 & \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array}, \quad \begin{array}{c|ccc} 0 & 0 & & \\ \frac{1}{2} & -\frac{\beta}{2} & \frac{1+\beta}{2} & \\ 1 & \frac{3+5\beta}{2} & -(1+3\beta) & \frac{1+\beta}{2} \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array}.$$

The implicit method on the right is A-stable for $\beta = \sqrt{3}/3$. Starting from the explicit Runge-Kutta method on the left we can derive a commutator

free method of order 3 using (2.25) and (2.26), we obtain

$$\begin{array}{c|cc} 0 & & \\ \frac{1}{2} & \frac{1}{2} & \\ 1 & -1 & 2 \\ \hline & \frac{1}{12} & \frac{1}{3} & -\frac{1}{4} \\ \hline & \frac{1}{12} & \frac{1}{3} & \frac{5}{12} \end{array},$$

this method coupled with the implicit method gives approximations of order three for (2.6).

EXAMPLE 2.7. In the next example we have considered a third order IMEX method from [ARW] as a starting point and, using (2.25) and (2.26), we have obtained the following pair of Commutator-Free DIRK methods of order 3:

$$\begin{array}{c|cc} 0 & & \\ \gamma & \gamma & \\ 1-\gamma & \gamma-1 & 2(1-\gamma) \\ \hline & 0 & \frac{1}{2}-\varphi & \frac{1}{2}+\varphi \\ \hline & 0 & \varphi & -\varphi \end{array}, \quad \begin{array}{c|cc} 0 & 0 & \\ \gamma & 0 & \gamma \\ 1-\gamma & 0 & 1-2\gamma & \gamma \\ \hline & 0 & \frac{1}{2} & \frac{1}{2} \end{array}$$

with $\gamma = \frac{3+\sqrt{3}}{6}$ and $\varphi = \frac{1}{6(2\gamma-1)}$.

Alternatively one could have considered the following commutator-free method:

$$\begin{array}{c|ccc} 0 & & & \\ \gamma & \gamma & & \\ 1-\gamma & \gamma-1 & 2(1-\gamma) & 0 \\ \hline & \alpha & \beta & \sigma \\ \hline & -\alpha & \frac{1}{2}-\beta & \frac{1}{2}-\sigma \end{array}$$

where $\sigma = (\alpha + \beta(1 - 2\gamma) - \frac{1}{3})/(1 - 2\gamma)$, $\alpha = 1/2$, $\beta = 1/6$, and the same value of γ as above.

We have not found substantial difference in the numerical experiments between these two explicit commutator-free methods.

3. Numerical tests

3.1. Order test. We consider the KdV equation

$$u_t + uu_x = \nu u_{xxx}, \quad x \in [0, 2\pi], \quad t \in [0, 1]$$

with periodic boundary conditions, discretized by

$$U_t = D(U.^2) + AU$$

where D and A are obtained applying central differences for the first and third derivative in space on a uniform grid, with periodic boundary conditions, $U_i \approx u(x_i, t)$, $U.^2 = \text{diag}(U)U$ denotes the component-wise square of U . The operator $C(U) := D\text{diag}(U)$. The spatial discretization is such that $\Delta x = 2\pi/31$, and we consider a range of $h = 1/(2^4), \dots, 1/(2^9)$. In this experiment we verify that the order of the various methods predicted by the theory is observed also in practice, figure 1 (a). We consider $u_0 = 1.5 + \cos(x)$.

The methods considered are

DIRK-CF1	transport diffusion algorithm (by Pironneau)
IMEX2	based on the midpoint, example (2.4)
DIRK-CF2	based on the midpoint, example (2.4)
IMEX2L	IMEX based on an L-stable implicit DIRK method, example (2.5)
DIRK-CF2L	partitioned RK commutator-free method with L-stable DIRK method, example (2.5)
IMEX3G	partitioned RK method due to Griepentrog, example (2.6)
DIRK-CF3G	partitioned RK commutator-free method due to Griepentrog, example (2.6)
IMEX3	IMEX method, example (2.7)
DIRK-CF3	partitioned RK commutator-free method, example (2.7)

The numbers refer always to the order of the method. All the DIRK-CF method have a semi-Lagrangian counterpart which we denote with SL1, SL2, SL2L, SL3G and SL3 respectively. The exponentials of the DIRK-CF methods are simply computed by using the built-in Matlab function `expm`.

In this experiment the IMEX methods of order 2 and 3 produce a bigger error compared to the corresponding DIRK-CF methods. The advantage of the DIRK-CF3 on the IMEX3 is clearly seen in figure 1 (b), where the two numerical solutions are compared to the reference solution (denoted *exact* in the figure). The reference solution is computed with the same discretization in space ($\Delta x = 2\pi/63$), and the time-integration is carried out by the Matlab function `ode15s` (with tolerances 10^{-10}). The time interval is $[0, 8]$ and the time-step is $h = 0.0312$ for both IMEX3 and DIRK-CF3. Doubling the size of the time-step the IMEX3 method becomes unstable.

3.2. 1D Viscous Burgers' equation. In figure 2, and 3 we consider

$$u_t + uu_x = \nu \nabla^2 u,$$

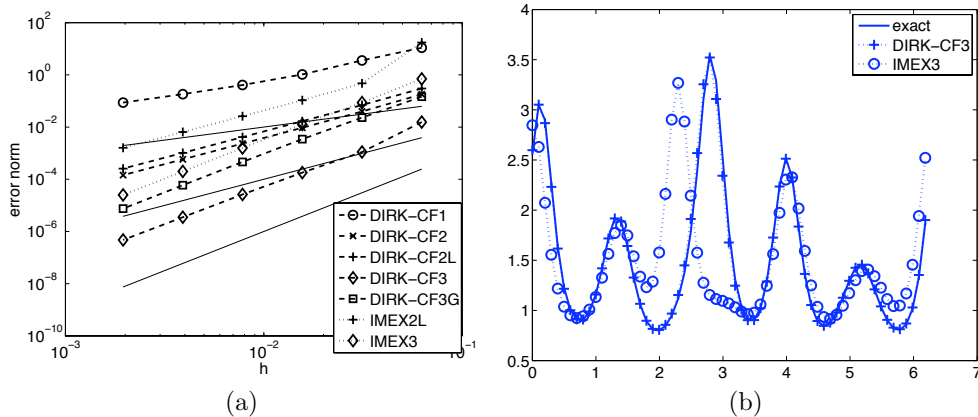


FIGURE 1. KdV equation: space discretization by central differences. (a) Order test: global error at $t = 1$, for different values of the time step h . $\nu = 0.05$, $x \in [0, 2\pi]$, $\Delta x = \frac{2\pi}{31}$. (b) Comparison of IMEX3 and DIRK-CF3 for the KdV equation: numerical solution at $t = 8$ as a function of $x \in [0, 2\pi]$, $\nu = 0.01$, $\Delta x = \frac{2\pi}{63}$, $h = \frac{1}{2^5}$.

on $[0, 1]$ with homogeneous Dirichlet BCs, integrated on $[0, 1]$. We plot the relative error in the ∞ norm versus the viscosity. The error is computed as follows:

$$\frac{\|U - U_{ref}\|_{\infty}}{\|U_{ref}\|_{\infty}}.$$

We discretize in space with a step $\Delta x = 1/81$ and in time with $h = 1.8\Delta x$. The reference solution, U_{ref} , is obtained discretizing in space with $\tilde{\Delta x} = 1/2592$ while the time integration of the discrete problem is performed in Matlab by the built-in function `ode15s` with absolute and relative tolerances 10^{-8} . For small viscosity values the semi-Lagrangian methods perform better than their DIRK-CF and IMEX counterparts. Some of the Eulerian methods perform very unstably for low viscosities and some break down. Figure 2 considers cubic spline interpolation for the convecting vector field and the evaluation of the initial solution at the departure points of the characteristics, while in figure 3 we used the built in Matlab function `interp1` performing piecewise cubic, monotonic interpolation. The choice of the interpolation method in the implementation of semi-Lagrangian methods is crucial for optimizing the performance at low viscosities, but this issue will not be discussed in detail in this paper.

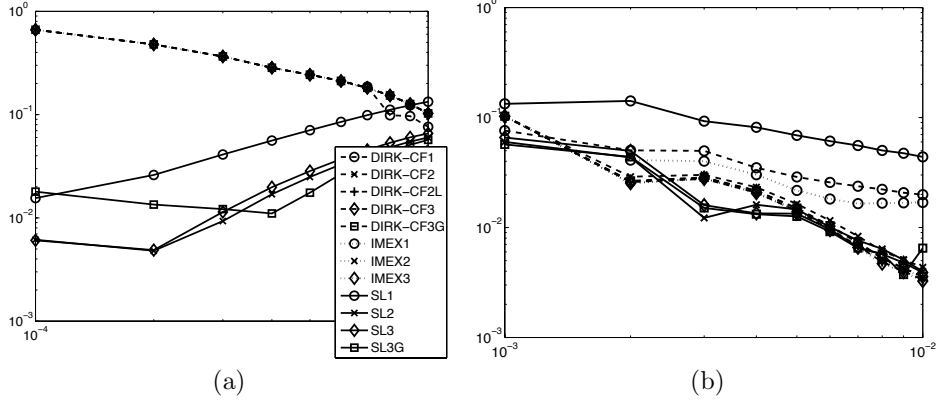


FIGURE 2. $u(x,0) = \sin(\pi x)$, $\Delta x = 1/81$, $t = 2$, $h = 1.8\Delta x$: relative error against viscosity for IMEX, DIRK-CF and semi-Lagrangian methods with cubic spline interpolation, (a) $\nu \in [0.0001, 0.001]$ and (b) $\nu \in [0.001, 0.01]$.

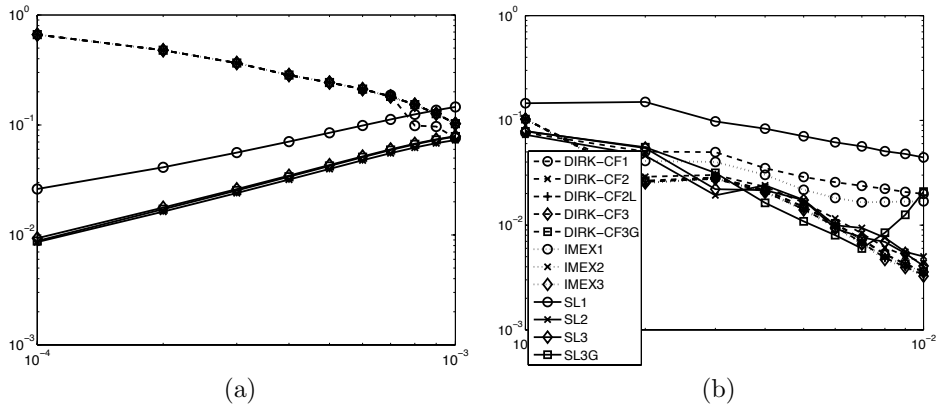


FIGURE 3. $u(x,0) = \sin(\pi x)$, $\Delta x = 1/81$, $t = 2$, $h = 1.8\Delta x$: relative error against viscosity for IMEX, DIRK-CF and semi-Lagrangian methods with piecewise cubic monotonic interpolation, (a) $\nu \in [0.0001, 0.001]$ and (b) $\nu \in [0.001, 0.01]$.

3.3. Linear convection and convection-diffusion in 2D. We firstly consider a pure convection problem on the square with Dirichlet boundary conditions and with a convecting vector field tangent to the boundary of the square, $\mathbf{V} = \begin{bmatrix} \pi \sin(\pi x) \cos(\pi y) \\ -\pi \cos(\pi x) \sin(\pi y) \end{bmatrix}$. The initial condition is

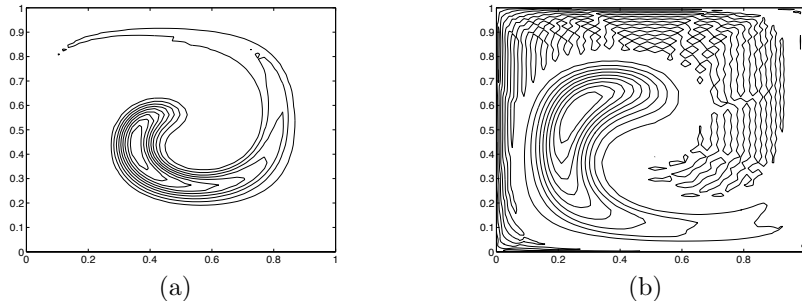


FIGURE 4. Pure convection with spectral methods in two dimensions. Semi-Lagrangian versus Eulerian method: (a) approximation of the characteristics trajectories with an accurate time-stepping procedure and interpolation of the initial condition; (b) semidiscretization in space and accurate solution of the linear system of odes in time.

$\mathbf{u}(x, y, 0) = e^{(-45(y-\frac{3}{4})^2 - 15(x-\frac{1}{2})^2)}$. We discretize in space with a spectral Galerkin method based on the Gauss-Lobatto-Legendre (GLL) points. We seek the solution at time $t = 1.2$ and we use a restarted Krylov subspace method for the approximation of the matrix exponential, [CM], to solve accurately the linear system of ODEs arising from the space discretization. This approach is compared to the use of an integration method for computing fairly accurately the characteristics (the `ode23` Matlab routine with tolerances set at 10^{-3}) and the interpolation on the GLL nodes of the initial condition on the arrival points of the characteristics. The results are shown in figure 4. The semi-Lagrangian method does not produce any substantial numerical dispersion compared to the Eulerian approach.

In the next experiment we consider linear convection-diffusion on the square $\Omega = [-1, 1]^2$ with viscosity $\nu = 10^{-4}$. We compare the performance of the Eulerian DIRK-CF3 method and of its semi-Lagrangian version SL3 integrating on the time interval $[0, 2\pi]$ using 40 time-steps (i.e. with step-size $h = 0.1571$). The same test case has been considered in [XK].

The initial data is

$$u(x, y, 0) = e^{-[(x-x_0)^2 + (y-y_0)^2]/2\lambda^2},$$

with $(x_0, y_0) = (-0.5, 0)$, $\lambda = 1/8$. The solution of this PDE is

$$u(x, y, t) = \frac{\lambda^2}{\lambda^2 + 2\nu t} e^{[\hat{x}^2 + \hat{y}^2]/2(\lambda^2 + 2\nu t)},$$

where $\hat{x} = x - x_0 \cos t - y_0 \sin t$, $\hat{y} = y + x_0 \sin t - y_0 \cos t$.

We use a spectral Galerkin discretization in space based on polynomials of degree $p = 16$ on Gauss-Lobatto-Legendre nodes. In figure 5 one can

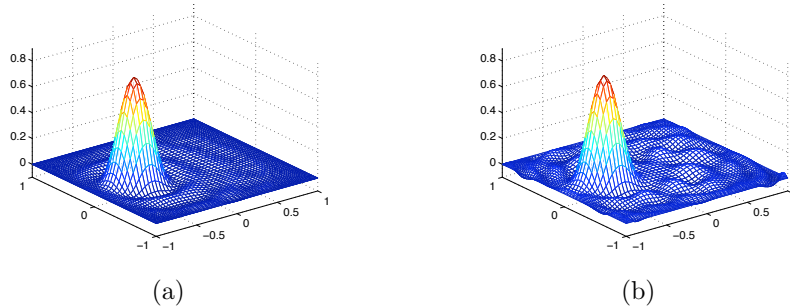


FIGURE 5. Linear convection-diffusion with spectral methods in two dimensions. Semi-Lagrangian versus Eulerian method: (a) SL3 integration in time, the diffusion is discretized on the GLL grid and the pure convection problems are approximated by the exact computation of the characteristics trajectories and the interpolation on the GLL nodes of the initial condition; (b) semi-discretization in space with a spectral Galerkin method and time integration performed with the DIRK-CF3 method, the exponentials of the convection operator are computed using the Matlab built in function `expm`.

observe that the Eulerian approach has a tendency of introducing spurious oscillations in the numerical solution compared to the semi-Lagrangian method. The values of the relative error computed in the L_2 norm are of comparable size for the two approaches. We have an error of 0.0124 for the semi-Lagrangian method and of 0.0263 for its Eulerian counterpart. The maximum and minimum values of the numerical approximation are 0.8164 and -0.0239 for the SL3 and 0.8324 and -0.0500 for the DIRK-CF3. The maximum and minimum values of the exact solution are 0.9225 and 0.

4. Conclusions

We presented a new class of integrators for convection dominated problems with a nonlinear convection term. The methods have good properties when applied in a semi-Lagrangian fashion. They require the solution of only one linear system per stage. The methods have been tested preliminarily on the viscous Burgers' equation, and the KdV equation and with a linear two dimensional convection-diffusion equation. The results are encouraging. A preliminary study of the order conditions has also been presented.

A more general setting for the methods would consider

$$\varphi_{i,j} := \exp\left(h \sum_k \alpha_{i,j,j}^k C(Y_k)\right) \cdots \exp\left(h \sum_k \alpha_{i,j,1}^k C(Y_k)\right),$$

in the formulae of section 2. We have not fully explored this possibility yet, but we believe that in such setting one should be able to design methods involving a smaller number of exponentials per stage compared to the present case.

References

- [ARW] U. M. Ascher, S. J. Ruuth and B. T. R. Wetton, *Implicit-explicit methods for time-dependent partial differential equations*, SIAM J. Num. Anal., **32** (1995), 797–823.
- [Ba] M.J. Baines, *Moving finite elements*, Monographs on Numerical Analysis, Clarendon Press, Oxford.
- [CHQZ] C. Canuto, M. Y. Hussaini, A. Quarteroni and T. A. Zang, *Spectral methods in Fluid Dynamics*, Springer-Verlag, Berlin, (1988).
- [CG] P. E. Crouch and R. Grossman *Numerical integration of ordinary differential equations on manifolds*, Journal of Nonlinear Science, *3* (1993),1–33.
- [CMO] E. Celledoni, A. Marthinsen and B. Owren, *Commutator-free Lie group methods*, FCGS, *19* (2003), 341–352.
- [Ce] E. Celledoni and B. K. Kometa, *Eulerian and semi-Lagrangian commutator-free exponential integrators*, CRM Proceedings vol. 39, *19* (2004).
- [CK] E. Celledoni, *Order theory for semi-Lagrangian exponential integrators*, in preparation.
- [CM] E. Celledoni and I. Moret *A Krylov projection method for systems of ODEs*, Appl. Num. Math., **24** (1997), 365–378.
- [HW] E. Hairer and G. Wanner *Solving Ordinary Differential Equations II*, Springer series in Computational Mathematics, Springer, (1996), second edition.
- [HE] R.W. Hockney and J.W. Eastwood *Computer simulations using particles*, McGraw-Hill, New York, 1981.
- [KC] C. A. Kennedy and M. H. Carpenter *Additive Runge–Kutta schemes for convection-diffusion-reaction equations*, App. Num. Math. **44** (2003), 139–181.
- [O] B. Owren, *Order conditions for commutator-free Lie group methods*, J. Phys. A, **39** (2006), pp.5585–5599.
- [Pi] O. Pirroneau, *On the transport-diffusion algorithm and its applications to the Navier-Stokes equations*, Numer. Math. *38*, 309–332 (1982).
- [ZP] Z. Zheng and L. Petzold, *Runge-Kutta-Chebyshev projection method*, J. Comput. Phys. *219* (2006), no. 2, 976–991
- [RM] M.D. Rees and K.W. Morton, *Moving point, particle and free Lagrange methods for convection-diffusion equations*, SIAM J. Sci. Statist. Comput. *12*(3), 547–572.
- [XK] D. Xiu and G.E. Karniadakis, *A semi-Lagrangian high-order method for Navier-Stokes equation*, J. of Comput. Phys. *172*, 658–684 (2001).

INSTITUTT FOR MATEMATISKE FAG, NTNU, 7049 TRONDHEIM

E-mail address: elenac@math.ntnu.no

E-mail address: kometa@math.ntnu.no