

## HIGH ORDER IMPLICIT-EXPLICIT GENERAL LINEAR METHODS WITH OPTIMIZED STABILITY REGIONS\*

HONG ZHANG<sup>†</sup>, ADRIAN SANDU<sup>†</sup>, AND SÉBASTIEN BLAISE<sup>‡</sup>

**Abstract.** In the numerical solution of partial differential equations using a method-of-lines approach, the availability of high order spatial discretization schemes motivates the development of sophisticated high order time integration methods. For multiphysics problems with both stiff and nonstiff terms implicit-explicit (IMEX) time stepping methods attempt to combine the lower cost advantage of explicit schemes with the favorable stability properties of implicit schemes. Existing high order IMEX Runge–Kutta or linear multistep methods, however, suffer from accuracy or stability limitations. This work shows that IMEX general linear methods (GLMs) are competitive alternatives to classic IMEX schemes for large problems arising in practice. High order IMEX-GLMs are constructed in the partitioned GLM framework developed earlier by the authors [*J. Sci. Comput.*, 61 (2014), pp. 119–144]. The stability regions of the new schemes are optimized numerically. The resulting IMEX-GLMs have similar stability properties as IMEX Runge–Kutta methods, but they do not suffer from order reduction and are superior in terms of accuracy and efficiency. The new IMEX-GLMs have considerably better stability properties than the IMEX linear multistep methods. Numerical experiments with two- and three-dimensional test problems illustrate the potential of the new schemes to speed up complex applications.

**Key words.** implicit-explicit integration, general linear methods, DIMSIM

**AMS subject classifications.** 65C20, 65M60, 86A10, 35L65

**DOI.** 10.1137/15M1018897

**1. Introduction.** Many problems in science and engineering are modeled by time-dependent systems of equations involving both stiff and nonstiff terms. Examples include advection-diffusion-reaction equations, fluid-structure interactions, and Navier–Stokes equations, and they arise in application areas such as mechanical and chemical engineering, astrophysics, meteorology and oceanography, and environmental science.

A method-of-lines approach is frequently employed to separate the spatial and temporal terms in the governing partial differential equations (PDEs). After the spatial terms are discretized by techniques such as finite differences, finite volumes, and finite elements, the resulting system of ordinary differential equations (ODEs) is integrated in time. Stiffness may result from different time scales involved (e.g., convective versus acoustic waves), from local processes such as chemical reactions, and from grids with complex geometry [29].

Explicit numerical integration schemes have maximum allowable time steps bounded by the fastest time scales in the system; for example, the time steps are restricted by the CFL stability condition. Implicit integration schemes can avoid the

---

\*Submitted to the journal's Methods and Algorithms for Scientific Computing section April 28, 2015; accepted for publication (in revised form) March 21, 2016; published electronically May 17, 2016. This work was supported through the awards NSF DMS-0915047, NSF DMS-1419003, NSF CCF-0916493, NSF OCI-0904397, NSF CMMI-1130667, NSF CCF-1218454, AFOSR FA9550-12-1-0293-DEF and AFOSR 12-2640-06 and by the Computational Science Laboratory at Virginia Tech.

<http://www.siam.org/journals/sisc/38-3/M101889.html>

<sup>†</sup>Computational Science Laboratory, Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061 (zhang@vt.edu, sandu@cs.vt.edu).

<sup>‡</sup>Institute of Mechanics, Materials and Civil Engineering, Université Catholique de Louvain, Louvain-la-Neuve, Belgium (sebastien.blaise@uclouvain.be).

step size restrictions but require the solution of large nonlinear systems at each step and are therefore computationally expensive. It is therefore of considerable interest to construct numerical integration schemes that avoid the time step restrictions while maintaining a high computational efficiency. In the implicit-explicit (IMEX) framework computational efficiency is achieved by performing an implicit integration only for the stiff components of the system.

IMEX methods treat the nonstiff term explicitly and the stiff term implicitly, therefore attempting to combine the low cost of explicit methods with the favorable stability properties of implicit methods. The development of IMEX linear multistep methods and IMEX Runge–Kutta (RK) methods has been reported in [4, 20, 27, 3, 12, 35, 40].

High order methods usually yield more accuracy and better efficiency than low order methods. Many modern PDE solvers are able to employ high order spatial discretizations, e.g., by using high degree polynomials in a discontinuous Galerkin (DG) approach. There is a need to develop high order time stepping formulas to be used in conjunction with high order spatial discretizations. This need motivates the current work.

Existing high order IMEX methods face challenges when applied to practical problems. High order IMEX linear multistep methods suffer from a marked reduction of the stability region with increasing order. IMEX-RK methods of order greater than two are known to suffer from possible order reduction for stiff problems, which reduces the efficiency of high order methods to that of low order methods. Common sources of stiffness in PDEs are due to small grid sizes, to the relaxation in hyperbolic systems, and to inhomogeneous boundary conditions. The order reduction due to stiff relaxation can be alleviated by incorporating additional order conditions [11]. A third order additive Runge–Kutta (ARK) scheme was introduced in [12] and demonstrated no order reduction when used for the time evolution of hyperbolic PDEs [10]. But methods of such type with order higher than 3 are not available. A considerable increase in the number of coupling conditions would make their construction difficult. Some possible remedies for applying RK methods to problems with inhomogeneous boundary conditions have also been proposed in [17]. However, these strategies require special treatment of boundaries, which brings in additional computational cost and complexity; moreover, they merely work for linear boundary conditions. So to the best of our knowledge, there is no effective and computationally efficient way for IMEX-RK methods to handle order reduction due to all types of stiffness.

Boscarino gave the theoretical global error estimates of several popular IMEX-RK methods for stiff systems in [9] and proved the connection between the stage order and these global error estimates (see Theorems 3.1 and 3.2 and Corollary 3.1 in [9]). The test results in [10] show that several third-order ARK schemes designed to have stage order two outperform classical ARK schemes, which has further confirmed the theory. Previous work clearly demonstrated that high stage order plays an important role in avoiding order reduction.

This work develops and tests new high order time stepping schemes with high stage order in the framework of IMEX general linear methods (IMEX-GLMs) that we have recently developed [42, 43]. The GLM family proposed by Butcher and Jackiewicz [13] generalizes both RK and linear multistep methods. The added complexity gives the flexibility to develop methods with better stability and accuracy properties. While RK and linear multistep methods are special cases of GLMs, the framework allows for the construction of many other methods as well. In [43, 42] we have developed second- and third-order IMEX-GLM schemes that showed considerable promise.

This study develops fourth- and fifth-order IMEX-GLMs with optimized stability properties. Numerical experiments confirm that these methods do not suffer from order reduction and are considerably more efficient than IMEX-RK methods on a suite of problems ranging from two-dimensional (2D) Allen–Cahn and Burgers equations to 3D compressible Euler equations.

The paper is organized as follows. Section 2 reviews the class of general linear methods. The construction of high order IMEX-GLMs with desired stability properties is discussed in section 3. This section first introduces desirable stability properties building upon existing stability theory for Runge–Kutta methods. Numerical results are reported in section 4. Conclusions are drawn in section 5.

**2. IMEX general linear methods.** IMEX time stepping methods are used to solve systems of ODEs of the form

$$(2.1) \quad y' = f(t, y) + g(t, y), \quad t_0 \leq t \leq t_F, \quad y(t_0) = y_0 \in \mathbb{R}^d,$$

where  $f$  is a nonstiff term, and  $g$  is a stiff term. Many systems of PDEs solved in the methods of lines framework lead to partitioned ODE systems (2.1) after semidiscretization in space. The nonstiff and stiff driving physical processes are captured by  $f$  and  $g$ , respectively.

Partitioned and IMEX general linear methods were developed in [43, 42]. An IMEX-GLM applied to (2.1) advances the solution for one step using

$$(2.2a) \quad Y_i = h \sum_{j=1}^{i-1} a_{i,j} f(Y_j) + h \sum_{j=1}^i \hat{a}_{i,j} g(Y_j) + \sum_{j=1}^r u_{i,j} y_j^{[n-1]}, \quad i = 1, \dots, s,$$

$$(2.2b) \quad y_i^{[n]} = h \sum_{j=1}^s \left( b_{i,j} f(Y_j) + \hat{b}_{i,j} g(Y_j) \right) + \sum_{j=1}^r v_{i,j} y_j^{[n-1]}, \quad i = 1, \dots, r.$$

Such a method is denoted IMEX-GLM( $p, q, s, r$ ) ( $p, q, s$ , and  $r$  stand for order, stage order, number of internal stages, and number of external stages, respectively) as defined in [28]. The implicit and the explicit components share the same abscissa vector  $\mathbf{c}$  and the same coefficients  $\mathbf{U}$  and  $\mathbf{V}$ . The IMEX-GLM (2.2) is represented compactly by the Butcher tableau

$$(2.3) \quad \begin{array}{c|cc|c} \mathbf{c} & \mathbf{A} & \hat{\mathbf{A}} & \mathbf{U} \\ \hline & \mathbf{B} & \hat{\mathbf{B}} & \mathbf{V} \end{array}.$$

To study the method (2.2) in [43, 42] the additively partitioned original system (2.1) is written in an equivalent component partitioned form [3]:

$$(2.4a) \quad y = x + z,$$

$$(2.4b) \quad x' = \tilde{f}(x, z) = f(x + z),$$

$$(2.4c) \quad z' = \tilde{g}(x, z) = g(x + z).$$

The external vector  $y_i^{[n-1]}$  is defined as a  $p$ th-order approximation of linear combinations of derivatives

$$(2.5) \quad y_i^{[n-1]} = \sum_{k=0}^r q_{i,k} h^k x^{(k)}(t_{n-1}) + \sum_{k=0}^r \hat{q}_{i,k} h^k z^{(k)}(t_{n-1}) + \mathcal{O}(h^{p+1}), \quad i = 1, \dots, r,$$

for some real parameters  $q_{i,k}$ ,  $i = 1, \dots, r$ ,  $k = 0, 1, \dots, p$ . Note that in (2.2)  $x_i^{[n]}$  and  $z_i^{[n]}$  need not be known individually once they are initialized in the first step. Only the combined external vector  $y_i^{[n]} = x_i^{[n]} + z_i^{[n]}$  is advanced at each step, similar to how regular GLMs proceed.

To initialize  $y_i^{[0]}$  the starting procedure developed in [43] advances the ODE solution by taking  $r - 1$  steps with a small step size  $\tau$  to obtain the solutions  $y_0, y_1^{\text{start}}, \dots, y_{r-1}^{\text{start}}$ . The derivative terms are approximated using only the function evaluations at these  $r$  points. The starting value for the external vector  $y_i^{[0]}$  is calculated via the formula

$$y_i^{[0]} = y_0 + q_{i,1} h f(y_0) + \hat{q}_{i,1} h g(y_0) + \sum_{k=2}^r \sum_{j=1}^r q_{i,k} h^k / \tau^{k-1} d_{k,j} f(y_j^{\text{start}}) + \sum_{k=2}^r \sum_{j=1}^r \hat{q}_{i,k} h^k / \tau^{k-1} d_{k,j} g(y_j^{\text{start}}).$$

In vector form it can be written as

$$(2.6) \quad y^{[0]} = \mathbf{1}_r \otimes y_0 + \tau (\mathbf{QD} \otimes \mathbf{I}_{d \times d}) ((\mathbf{R} \otimes \mathbf{I}_{d \times d}) F^{\text{start}}) + \tau (\hat{\mathbf{QD}} \otimes \mathbf{I}_{d \times d}) ((\mathbf{R} \otimes \mathbf{I}_{d \times d}) G^{\text{start}}),$$

where  $F^{\text{start}}$  and  $G^{\text{start}}$  consist of function values evaluated at the  $r$  starting points, e.g.,  $F^{\text{start}} = [f(y_0^{\text{start}}), f(y_1^{\text{start}}), \dots, f(y_{r-1}^{\text{start}})]^T$ .

The  $r \times r$  coefficient matrices  $\mathbf{Q}$ ,  $\mathbf{D}$ , and  $\mathbf{R}$  are computed as follows:

1.  $\mathbf{Q}$ ,  $\hat{\mathbf{Q}}$  are determined by the method coefficients  $\mathbf{A}$ ,  $\hat{\mathbf{A}}$  and the abscissa vector  $\mathbf{c}$ . These matrices can be computed columnwise via the order conditions [13]

$$(2.7) \quad q_0 = \mathbf{1}_s, \quad q_i = \frac{\mathbf{c}^i}{i!} - \frac{\mathbf{A} \mathbf{c}^{i-1}}{(i-1)!}; \quad \hat{q}_0 = \mathbf{1}_s, \quad \hat{q}_i = \frac{\mathbf{c}^i}{i!} - \frac{\hat{\mathbf{A}} \mathbf{c}^{i-1}}{(i-1)!}.$$

2. Starting with the approximation

$$(2.8) \quad \begin{bmatrix} \tau x'(t_0) \\ \tau^2 x''(t_0) \\ \vdots \\ \tau^r x^{(r)}(t_0) \end{bmatrix} = \tau \mathbf{D} \begin{bmatrix} x'(t_0) \\ x'(t_1) \\ \vdots \\ x'(t_{r-1}) \end{bmatrix} + \mathcal{O}(\tau^{r+1}),$$

expanding the right-hand side in Taylor series, and comparing the coefficients of each term allows us to identify each entry of  $\mathbf{D}$ .

3.  $\mathbf{R}$  is a diagonal rescaling matrix which has the form

$$(2.9) \quad \mathbf{R} = \text{diag}(h/\tau, h^2/\tau^2, \dots, h^r/\tau^r).$$

Note that this starting procedure enables us to compute the initial approximations with a smaller step size  $\tau \leq h$ . The initial approximations can be computed with a regular method of choice; the very small time steps ensure accurate initial solutions and also circumvent possible numerical stability issues with the auxiliary scheme. The starting procedure used for the experiments in this paper employs the IMEX-RK scheme. Considering the possible low accuracy caused by order reduction, in the starting procedure we use a step size half as large as the step size for the following integration. We point out that using the same step size typically works well based on our experience.

**3. Construction of high order IMEX-GLMs.** We now consider the construction of high order IMEX-GLMs. The partitioned GLM theory developed in [43] ensures that if the stage order is high, the IMEX-GLM method has the desired order without the need for coupling conditions. One imposes the order and stage order conditions independently on the implicit and on the explicit component GLMs.

The order conditions for constructing arbitrary GLMs are complicated. In this paper we choose the explicit and implicit components from a subclass of GLMs, named diagonally implicit multistage integration methods (DIMSIMs), for which the order conditions are more manageable. DIMSIMs are a subclass of GLMs characterized by the following properties [13]:

1.  $\mathbf{A}$  is lower triangular with the same element  $a_{i,i} = \lambda$  on the diagonal;
2.  $\mathbf{V}$  is a rank-1 matrix with the nonzero eigenvalue equal to one to guarantee preconsistency;
3. The order  $p$ , stage order  $q$ , number of external stages  $r$ , and number of internal stages  $s$  are related by  $q \in \{p-1, p\}$  and  $r \in \{s, s+1\}$ .

DIMSIMs can be categorized into four types according to [13]. Type 1 and type 2 methods have  $a_{i,j} = 0$  for  $j \geq i$  and are suitable for a sequential computing environment, while type 2 and type 3 methods have  $a_{i,j} = 0$  for  $j \neq i$  and are suitable for parallel computation. Methods of types 1 and 3 are explicit ( $a_{i,i} = 0$ ), while methods of types 2 and 4 are implicit ( $a_{i,i} = \lambda \neq 0$ ) and potentially useful for stiff systems.

Following [43] we are particularly interested in DIMSIMs with  $p = q = r = s$ ,  $\mathbf{U} = \mathbf{I}_{s \times s}$ , and  $\mathbf{V} = \mathbf{1}_s v^T$ , where  $v^T \mathbf{1}_s = 1$  [28]. The order conditions are satisfied if the coefficient matrix  $\mathbf{B}$  is computed from the relation

$$(3.1) \quad \mathbf{B} = \mathbf{B}_0 - \mathbf{A}\mathbf{B}_1 - \mathbf{V}\mathbf{B}_2 + \mathbf{V}\mathbf{A},$$

where the matrices  $\mathbf{B}_0, \mathbf{B}_1, \mathbf{B}_2 \in \mathbb{R}^{s \times s}$  have entries

$$(\mathbf{B}_0)_{i,j} = \frac{\int_0^{1+c_i} \phi_j(x) dx}{\phi_j(c_j)}, \quad (\mathbf{B}_1)_{i,j} = \frac{\phi_j(1+c_i)}{\phi_j(c_j)}, \quad (\mathbf{B}_2)_{i,j} = \frac{\int_0^{c_i} \phi_j(x) dx}{\phi_j(c_j)},$$

and  $\phi_i(x)$  are defined by  $\phi_i(x) = \prod_{j=1, j \neq i}^s (x - c_j)$  (cf. [13, Thm. 5.1], [28, Thm. 3.2.1]). Therefore to obtain high order DIMSIMs there is no need to solve complex nonlinear systems as one usually does in the construction of RK methods.

The important challenge that remains in the construction of IMEX-GLM methods is to achieve the desirable stability properties. This section first introduces desirable stability properties building upon existing stability theory for RK methods. A numerical optimization process used to maximize the IMEX stability regions is then discussed. Two new IMEX-DIMSIM methods of orders four and five are presented at the end.

### 3.1. Stability considerations.

*A-stability, L-stability, and inherited RK stability.* The classical linear stability theory [26] considers the scalar test problem whose solution decays to zero,

$$(3.2) \quad y' = \lambda y, \quad t \geq 0, \quad \operatorname{Re}(\lambda) \leq 0.$$

A numerical method is stable if when applied to solve the test problem (3.2) for one step of length  $h$  it generates a solution of nonincreasing size. A GLM  $(\mathbf{A}, \mathbf{B}, \mathbf{U}, \mathbf{V})$  (2.3) applied to the test problem gives a solution

$$(3.3) \quad y^{[n+1]} = \mathbf{M}(z) y^{[n]}, \quad \mathbf{M}(z) = \mathbf{V} + z \mathbf{B} (\mathbf{I}_{s \times s} - z \mathbf{A})^{-1} \mathbf{U}.$$

Here  $\mathbf{M}(z)$  is the stability matrix and has a corresponding stability function

$$(3.4) \quad p(w, z) = \det(w\mathbf{I}_{r \times r} - \mathbf{M}(z)),$$

where  $w, z \in \mathbb{C}$  and  $z = \lambda h$ .

A-stability requires that the method is unconditionally stable independent of the size of the time step  $h$ , i.e., the spectral radius of the stability matrix  $\rho(\mathbf{M}(z)) \leq 1$  for any  $z$ . L-stability further requires that  $\rho(\mathbf{M}(z)) \rightarrow 0$  when  $z \rightarrow \infty$  [26]. L-stable methods damp components of high frequencies and are particularly useful for stiff problems. Since IMEX-GLM schemes are designed to treat stiff parts of a given problem implicitly, we want the implicit component to be L-stable, or at least A-stable. Imposing L-stability directly on the GLM coefficients leads to a difficult analysis, with complexity increasing dramatically as the order increases.

The inherited RK stability property [41, 16] provides a practical way to achieve L-stability. This property requires that the stability function (3.4) has the form

$$(3.5) \quad p(w, z) = w^{s-1} (w - R(z)),$$

where  $R(z)$  is the stability function of an RK method of order  $p = s$ . When (3.5) holds the existing L-stability theory for RK methods can be applied to GLMs. Note that conditions (3.5) lead to additional nonlinear constraints on method coefficients; these constraints need to be solved accurately in practice.

*Stability analysis for IMEX-GLMs.* To study the linear stability of IMEX-GLM schemes we consider the following generalized linear test equation [43]:

$$(3.6) \quad y' = \xi y + \hat{\xi} y, \quad t \geq 0, \quad \operatorname{Re}(\xi), \operatorname{Re}(\hat{\xi}) \leq 0.$$

This test problem mimics the structure of (2.1). We consider  $\xi y$  to be the nonstiff term and  $\hat{\xi} y$  the stiff term and denote  $w = h\xi$  and  $\hat{w} = h\hat{\xi}$ .

Applying (2.2) to the test equation (3.6) and assuming  $\mathbf{I}_{s \times s} - w\mathbf{A} - \hat{w}\hat{\mathbf{A}}$  is nonsingular lead to

$$y^{[n]} = \mathbf{M}(w, \hat{w}) y^{[n-1]},$$

where the stability matrix is defined by [43]

$$(3.7) \quad \mathbf{M}(w, \hat{w}) = \mathbf{V} + \left( w\mathbf{B} + \hat{w}\hat{\mathbf{B}} \right) \left( \mathbf{I}_{s \times s} - w\mathbf{A} - \hat{w}\hat{\mathbf{A}} \right)^{-1} \mathbf{U}.$$

Let  $S \subset \mathbb{C}$  and  $\hat{S} \subset \mathbb{C}$  be the stability regions of the explicit GLM component and of the implicit GLM component, respectively. The *combined stability region* is defined by [43]

$$(3.8) \quad \mathcal{C} = \left\{ w \in S, \hat{w} \in \hat{S} : \rho(\mathbf{M}(w, \hat{w})) < 1 \right\} \subset S \times \hat{S} \subset \mathbb{C} \times \mathbb{C}.$$

For a practical analysis of stability we define a *desired stiff stability region*, e.g.,

$$\hat{S}_\alpha = \{ \hat{w} \in \hat{S} \cap \mathbb{C}^- : |\operatorname{Im}(\hat{w})| < \tan(\alpha) |\operatorname{Re}(\hat{w})| \},$$

and compute numerically the corresponding *constrained nonstiff stability region*:

$$(3.9) \quad \mathcal{S}_\alpha = \left\{ w \in S : \rho(\mathbf{M}(w, \hat{w})) < 1 \quad \forall \hat{w} \in \hat{S}_\alpha \right\}.$$

The IMEX-GLM method is stable if the constrained nonstiff stability region  $\mathcal{S}_\alpha$  is nontrivial (has a nonempty interior) and is sufficiently large for a prescribed (problem-dependent) value of  $\alpha$ , e.g.,  $\alpha = \pi/2$ .

### 3.2. Finding high order IMEX-DIMSims with large stability regions.

The implicit component of the IMEX-GLM is constructed first, and the desired L-stability property is imposed. L-stable GLMs existing in the literature can also be used as implicit components in the combined IMEX scheme.

L-stability indicates that  $\hat{w}$  in the nonstiff stability definition (3.9) can be any value on the negative half-plane. So the constrained region with  $\alpha = \pi/2$  is

$$\mathcal{S}_{\pi/2} = \left\{ w \in S : \rho(\mathbf{M}(w, re^{i\theta})) < 1 \quad \forall \theta \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right], \quad \forall r \in [0, -\infty) \right\}.$$

The corresponding explicit component is constructed next based on the following criteria: it shares the coefficients  $\mathbf{c}$ ,  $\hat{\mathbf{U}}$ ,  $\hat{\mathbf{V}}$  with the implicit component, it satisfies the desired order conditions, and it results in a large constrained stability region (3.9).

According to the order conditions in [43],  $\mathbf{B}$  depends on  $\mathbf{A}$  and  $\mathbf{c}$ . Thus the only free parameters in determining the explicit part are the  $s(s-1)/2$  elements of matrix  $\mathbf{A}$ . The problem of finding IMEX-DIMSims can be regarded as a numerical optimization problem to find the entries of  $\mathbf{A}$  such as to maximize the area of the constrained stability region  $\mathcal{S}_{\pi/2}$ .

We discretize the region  $\mathcal{S}_{\pi/2}$  using finite sets of points in polar coordinates

$$\mathcal{S}_{\pi/2} \approx \left\{ w \in S : \rho(\mathbf{M}(w, re^{i\theta})) < 1 \quad \forall \theta \in \Theta_f \subset \left[-\frac{\pi}{2}, \frac{\pi}{2}\right], \quad \forall r \in R_f \subset (-\infty, 0] \right\}.$$

For example,  $R_f = [0, -10^{-3}, -10^{-2}, \dots, -10^3]$  and  $\Theta_f$  are a set of equally spaced points between  $-\pi/2$  and  $\pi/2$ .

We next determine the boundary  $\partial\mathcal{S}_{\pi/2}$  of the constrained stability region. For this we consider the points of intersection of the boundary with vertical lines on the negative half-plane with abscissae  $x_k$ . An intersection point  $\tilde{w}_k = (x_k, y_k)$  should satisfy

$$(3.10) \quad \max_{r \in R_f, \theta \in \Theta_f} \rho(\mathbf{M}(\tilde{w}_k, re^{i\theta})) = 1.$$

Note that since the stability region is symmetric, we only need to consider the part above the real axis.

Starting with an initial point on the vertical line, e.g.,  $x_k + iy_*$ , where  $y_*$  is large enough to make the point outside the stability region, we apply the bisection Algorithm 1 to find the first point  $\tilde{w} = x_k + iy$  along the vertical line such that

$$(3.11) \quad \max_{r \in R_f, \theta \in \Theta_f} \rho(\mathbf{M}(\tilde{w}, re^{i\theta})) < 1.$$

A similar idea can be used to find the intersection of the stability region and the real axis, which is assumed to be the leftmost point of the stability region. Then we can determine the boundary with the above-mentioned algorithm. Algorithm 2 summarizes the procedure to approximate the area of the stability region.

As we can see, the objective function that approximates the area of the stability region is highly nonlinear and computationally expensive, especially for the construction of high order methods. The optimization problem is in general difficult to solve numerically. First we transform the maximization problem to a minimization problem by minimizing the negative of the objective function. Then we use a combination of the MATLAB genetic algorithm function `ga` and the MATLAB local minimizer `fminsearch`. We repeatedly apply the two optimization routines one after another using one's result as the starting point of the other. Each optimizer is run multiple

---

**Algorithm 1.** Bisection algorithm for finding the points of intersection.

---

```

Initialize  $y_{\text{top}} \leftarrow y_0$   $y_{\text{bot}} \leftarrow 0$ 
while  $y_{\text{top}} - y_{\text{bot}} > \text{tol}$  do
     $y_{\text{mid}} = (y_{\text{top}} + y_{\text{bot}})/2$ 
    if  $\tilde{w} \leftarrow c + i y_{\text{mid}}$  satisfies the condition (3.11) then
         $y_{\text{bot}} = y_{\text{mid}}$ 
    else
         $y_{\text{top}} = y_{\text{mid}}$ 
    end if
end while
return  $y_{\text{bot}}$ 

```

---



---

**Algorithm 2.** Algorithm for computing the area of constrained stability regions.

---

- 1: Find the point  $x_b$  of intersection of the stability region and the  $x$  axis using a bisection strategy similar to Algorithm 1
  - 2: Generate  $m$  vertical lines with abscissae  $x_k$  linearly spaced between  $x_b$  and 0
  - 3: Find the points of intersection of these lines and the stability region
  - 4: Approximate the area of the stability region using the trapezoidal method
- 

times until the results converge; each run is initialized with the previous result. We terminate the procedure when the result does not change across multiple runs for both optimizers.

**3.3. New IMEX-GLMs.** The construction of DIMSIMs starts with choosing the abscissa vector  $\mathbf{c}$  [14]. A natural choice is a vector of values equally spaced in the interval  $[0, 1]$ . For DIMSIMs of order  $p$  and stage order  $q = p$ , the last value  $\mathbf{c}_s = 1$  allows us to use the last stage value as the ODE solution at the next time step. This advantage also applies to IMEX-DIMSIM. Here we choose the common abscissae for the IMEX pairs equally spaced in  $[0, 1]$  and including 0 and 1. There is no evidence so far that other choices would lead to better schemes.

**3.3.1. A fourth-order IMEX-DIMSIM pair.** We start with the construction of the implicit part of the IMEX pair. Butcher and Jackiewicz [14] report a failed attempt to construct DIMSIMs with inherited RK stability,  $p = q = r = s = 4$ , and  $c = [0, 1/3, 2/3, 1]$ . Surprisingly we succeeded in solving the nonlinear system that comes from the stability constraints by using Mathematica software. For detailed information on the nonlinear system, we refer to [14]. The coefficients of the type 2 (implicit) DIMSIM we found are given in Figure 1. The choice of the diagonal element of  $\hat{A}$  equal to 0.572816062482135 ensures that the implicit method is  $L$ -stable, following the classic theory of RK methods [26]. We remark that this new implicit DIMSIM method can be used by itself due to its favorable stability properties.

The optimization problem formulated in section 3.2 for maximizing the constrained stability regions has six free variables, which are lower triangular entries of the coefficient matrix  $\mathbf{A}$ . The maximal area of the constrained stability region of the explicit method on the negative plane is approximately 1.34. Figure 2 shows the stability regions of the implicit component  $\hat{S}$ , of the explicit component  $S$ , as well as the constrained stability regions  $\hat{S}_\alpha$  for  $\alpha = \pi/2, \pi/3, \pi/4$ .

We will refer to the resulting method as IMEX-DIMSIM4. The coefficients of the explicit method to 15 accurate digits are given in Figure 1.



$$\begin{aligned}
A &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0.258897065974412 & 0 & 0 & 0 & 0 \\ 2.729801825357062 & -0.060004247312668 & 0 & 0 & 0 \\ 0.951308318232761 & 0.614160494289040 & 0.422498793609078 & 0 & 0 \\ 5.669708110906782 & -0.493235358869745 & 0.021475944586626 & 0.175951726795284 & 0 \\ 5.544708110906782 & 0.020653530019144 & -0.797968499857818 & 0.680943549709761 & 0 \\ 4.720814974705226 & 3.191226074825372 & -5.227438428178271 & 0.6861668900688894 & 0 \\ 4.848863779632135 & 2.337640759837926 & -3.218585217497575 & 0.418013495315584 & 0 \end{bmatrix} \\
B &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0.074436267358921 & 0.055555555555556 & 0.006172839506173 & 0 \\ 1 & -2.003130911377728 & 0.242223637993112 & 0.052716285344531 & 0.008600849263247 \\ 1 & -0.987967606130879 & 0.013613972830935 & 0.038658018404147 & 0.017011414548385 \\ 0.572816062482135 & 0 & 0 & 0 & 0 \\ 0.29478591621391 & 0.572816062482135 & 0 & 0 & 0 \\ 3.754531024312379 & -0.446626145372372 & 0.572816062482135 & 0 & 0 \\ 20.906355951077522 & -6.918033573971423 & 0.824272703722306 & 0.572816062482135 & 0 \end{bmatrix} \\
\hat{A} &= \begin{bmatrix} 2.818382755109841 & -0.107847984112942 & 1.213319973963157 & -0.548700992864529 & 0 \\ 3.266198817591976 & -1.885223345152593 & 3.830771904411522 & -1.797738883043436 & 0 \\ 3.774131970777119 & -3.469139895411032 & 5.100995462482731 & -4.672071998026633 & 0 \\ 1.800600620848989 & 6.203817506581311 & -13.407704583723200 & -5.034154872439978 & 0 \end{bmatrix} \\
\hat{B} &= \begin{bmatrix} 1 & -0.572816062482135 & 0 & 0 & 0 \\ 1 & -0.533961320770192 & -0.135383131938489 & -0.025650275076168 & -0.003021498328079 \\ 1 & -3.214054274755475 & -0.010779770975077 & -0.053097178648182 & -0.017299808772539 \\ 1 & -14.385411143310540 & 1.683679993026802 & 0.081422122041277 & -0.051803591005091 \end{bmatrix} \\
\hat{Q} &= \begin{bmatrix} 1 & -0.572816062482135 & 0 & 0 & 0 \\ 1 & -0.533961320770192 & -0.135383131938489 & -0.025650275076168 & -0.003021498328079 \\ 1 & -3.214054274755475 & -0.010779770975077 & -0.053097178648182 & -0.017299808772539 \\ 1 & -14.385411143310540 & 1.683679993026802 & 0.081422122041277 & -0.051803591005091 \end{bmatrix} \\
v &= [0.281364340879037 \quad -1.282889560784121 \quad 2.266595749735792 \quad -0.265070529830707 \quad ] \\
c &= [0 \quad 1/3 \quad 2/3 \quad 1 \quad ]
\end{aligned}$$

FIG. 1. Coefficients of the IMEX-DIMSIM-4.

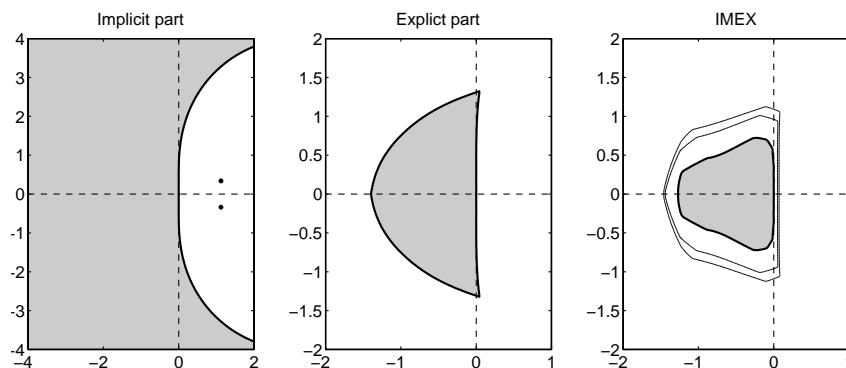


FIG. 2. Stability regions for the fourth-order IMEX-DIMSIM pair with  $p = q = r = s = 4$  and  $c = [0, 1/3, 2/3, 1]$ . From left to right are stability region  $\hat{S}$  of the implicit method, stability region  $S$  of the explicit method, and constrained stability regions  $\hat{S}_\alpha$  (with  $\alpha = \pi/2, \pi/3, \pi/4$  from interior toward exterior, respectively).

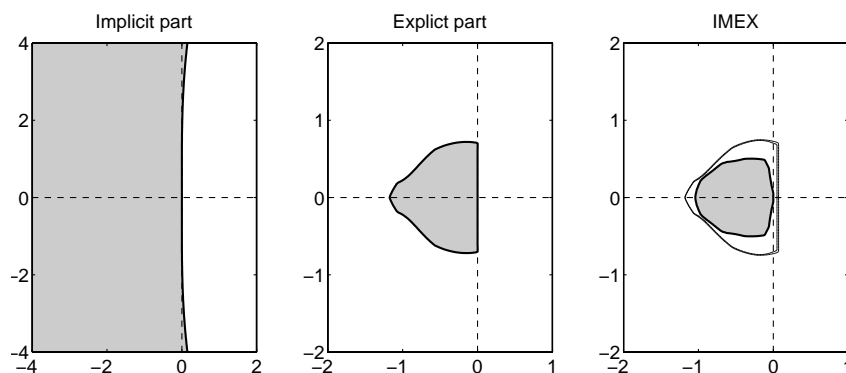


FIG. 3. Stability regions for the fifth-order IMEX-DIMSIM pair with  $p = q = r = s = 5$  and  $c = [0, 1/4, 1/2, 3/4, 1]$ . From left to right are stability region  $\hat{S}$  of the implicit method, stability region  $S$  of the explicit method, and constrained stability regions  $\hat{S}_\alpha$  (with  $\alpha = \pi/2, \pi/3, \pi/4$  from interior toward exterior, respectively).

**3.3.2. A fifth-order IMEX-DIMSIM pair.** An L-stable fifth-order type 2 (implicit) DIMSIM with  $p = q = r = s = 5$  and  $c = [0, 1/4, 1/2, 3/4, 1]$  was constructed by Butcher [15]. We have obtained its coefficients with improved accuracy from 6 to 15 decimal digits by solving the nonlinear conditions using the Levenberg–Marquardt algorithm implemented by the MATLAB routine `fsolve`.

The corresponding explicit component is obtained by the numerical optimization procedure described in section 3.2. The maximal area of the constrained stability region of the explicit method on the negative plane is approximately 0.83 and is smaller than the area of the fourth-order pair. Figure 3 shows the stability regions of the implicit component  $\hat{S}$ , the explicit component  $S$ , and the constrained stability regions  $\hat{S}_\alpha$  for  $\alpha = \pi/2, \pi/3, \pi/4$ .

We will refer to the resulting method as IMEX-DIMSIM5. The coefficients of the method to 15 accurate digits are given in Figure 4 (compare the implicit coefficients to [15]).

$$\begin{aligned}
A &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0.380631951399918 & 0 & 0 & 0 & 0 & 0 \\ -0.723344119927179 & 0.934338548518619 & 0 & 0 & 0 & 0 \\ -0.292421654731536 & 1.489386717103117 & 0.229042913082062 & 0 & 0 & 0 \\ 10.333193352608074 & 0.200217292186561 & 0.841800685401247 & -0.148918889975160 & 0 & 0 \end{bmatrix} \\
B &= \begin{bmatrix} -1.811278483713069 & 2.072219536433343 & 0.130011155311711 & 0.166279568600910 & 0.117403740739418 \\ -1.724125705935292 & 1.629858425322231 & 1.038344488645044 & -0.796914875843534 & 0.396841233783945 \\ -1.998394810009466 & 3.088356723470882 & -2.146707663207811 & 2.854109498231544 & -0.833722659704275 \\ -1.361504766226497 & 0.334933035918415 & 2.154212895587752 & 0.353113262914561 & -1.482126886275562 \\ 5.0910619244499312 & -29.458910962376240 & 55.143920860593482 & -43.440447985319850 & 3.112719239754878 \end{bmatrix} \\
Q &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -0.130631951399918 & 0.031250000000000 & 0.002604166666667 & 0.000162760416667 & 0 \\ 1 & 0.289005571408560 & -0.108584637129655 & -0.008364746307874 & 0.000170993363233 & 0 \\ 1 & -0.676007975453643 & -0.205618135816810 & -0.004861199044730 & 0.004533255151668 & 0 \\ 1 & -10.226292440220721 & 0.140734501734106 & 0.097068228416195 & 0.034078612640450 & 0 \end{bmatrix} \\
\hat{A} &= \begin{bmatrix} 0.278053841136452 & 0 & 0 & 0 & 0 & 0 \\ 0.220452276182580 & 0.278053841136452 & 0 & 0 & 0 & 0 \\ 2.294819895736366 & -0.602366708071285 & 0.278053841136452 & 0 & 0 & 0 \\ 5.054620901153854 & -1.529876218309763 & 0.097119141498823 & 0.278053841136452 & 0 & 0 \\ 9.345167780108133 & -1.412133513099773 & -1.883401998517870 & 0.782533955446870 & 0.278053841136452 & 0 \end{bmatrix} \\
\hat{B} &= \begin{bmatrix} 6.044855283302179 & -2.020000467205476 & 0.032934533641225 & 0.593578985923315 & -0.226664851205853 \\ 5.853954219943505 & -1.072092372634326 & -1.839270544389963 & 2.410922952843391 & -0.899263047489796 \\ 6.004175007913425 & -2.014097375842605 & 0.610845429880394 & -0.963490004887004 & -0.405182760273902 \\ 6.002703177071046 & -2.556003283230891 & 3.151551366098853 & -5.493514217893924 & 0.448102618067392 \\ 4.481882795290198 & 2.672564354868939 & -1.413660973235832 & -8.058154793746990 & 0.909905877341711 \end{bmatrix} \\
\hat{Q} &= \begin{bmatrix} 1 & -0.278053841136452 & 0 & 0 & 0 & 0 \\ 1 & -0.248506117319032 & -0.038263460284113 & -0.006085015868847 & -0.000561338127960 & 0 \\ 1 & -1.470507028801533 & 0.136564756449595 & 0.004900562818504 & -0.001619958388074 & -0.000365640421568 \\ 1 & -3.149917665479366 & 0.406619102975690 & 0.027778596315200 & -0.004406329750951 & -0.001692120959916 \\ 1 & -6.110220065073812 & 0.929780069812273 & 0.087106493228110 & -0.016782586272280 & -0.008434321001423 \end{bmatrix} \\
v &= [-0.079385465132435 \quad 0.554317572910577 \quad -1.569589549144155 \quad 2.332074592443682 \quad -0.237417151077669] \\
c &= [0 \quad 1/4 \quad 1/2 \quad 3/4 \quad 1]
\end{aligned}$$

FIG. 4. Coefficients of the IMEX-DIMSIM-5.

**4. Numerical tests.** We consider several test problems that are motivated by different application areas such as material science, fluid mechanics, and atmospheric modeling. All problems are governed by PDE and contain both stiff components and nonstiff components. The first two test cases are implemented in MATLAB using finite difference schemes for space discretization. The time integration is performed with the two high order IMEX general linear methods IMEX-DIMSIM4 and IMEX-DIMSIM5. We compare these methods against several classic IMEX schemes of the same order, including two IMEX additive RK methods, ARK4(3)6L[2]SA and ARK5(4)8L[2]SA, from Kennedy and Carpenter [33], and against two IMEX multistep methods from Hundsdorfer and Ruuth [27]. We also include the BHR553 IMEX RK method with stage order two from Boscarino [10], which satisfies a few additional coupling order conditions and exhibits no order reduction in the experiments with two stiff test problems [10]. Note that the order of the method is only three, and higher order methods of this type are not available at this time. All three chosen ARK methods have a stiffly accurate implicit component and their implicit part and explicit part share the same abscissa  $c = \hat{c}$  as our IMEX-DIMSIMs do.

We have also implemented the IMEX-DIMSIM schemes in the discontinuous Galerkin solver GMSH-DG [7] and applied them to the 3D compressible Euler equations coming from multiscale nonhydrostatic atmospheric simulations. The goal is to assess the performance of the high order IMEX-DIMSIMs and other IMEX methods including IMEX-RK methods and IMEX linear multistep methods on both 2D and 3D simulations. Besides the methods mentioned in the previous tests, we also include low order methods here for comparison in order to show the full advantage of high order methods. Specifically, we have considered a second-order ARK method (named IMEX-ARK2) developed particularly for the 3D compressible Euler equations in [24] and two classic methods of orders 2 and 3 from Ascher, Ruuth, and Spiteri (ARS) [3] that we denote by IMEX-ARS222 and IMEX-ARS343, respectively. The ARK schemes are constructed with L-stable, first-stage explicit, singly diagonal implicit RK (ESDIRK) methods and second stage order that alleviates the order reduction when applied to stiff problems. The ARS schemes also have L-stable implicit parts which can be cast into the form of ESDIRK by padding the Butcher table with zeros, but their stage order is lower. All the IMEX multistep methods we have used are IMEX-BDFs from Hundsdorfer and Ruuth [27]. With regard to conservation properties, all the IMEX-RK methods except ARS222 preserve linear invariants, whereas the IMEX-BDFs cannot preserve linear invariants [24]. It is noteworthy that some of the methods chosen here have been applied to a similar problem in [24], where continuous Galerkin methods are used for spatial discretization; the boundary conditions are simpler, leading to a less challenging test.

The first two experiments were performed on a workstation with four Intel Xeon E5-2630 processors. The GMSH-DG code was run on a workstation with 64 GB RAM and four AMD Opteron 6168 processors, each with 12 cores.

**4.1. Allen–Cahn equation.** We consider the 2D reaction-diffusion Allen–Cahn problem [18], which describes the process of phase transition in materials science.

$$(4.1) \quad \frac{\partial u}{\partial t} = \alpha \nabla^2 u + \beta(u - u^3) + f, \quad 0 \leq x, y \leq 1, \quad 0 \leq t \leq 0.5,$$

where the parameters are  $\alpha = 0.1$ ,  $\beta = 3$ , and  $f(t, x, y)$  is a source term that is consistent with the exact solution  $u(t, x, y) = 2 + \sin(2\pi(x - t)) \cos(3\pi(y - t))$ . Time varying Dirichlet boundary conditions (that represent the exact solution evaluated at

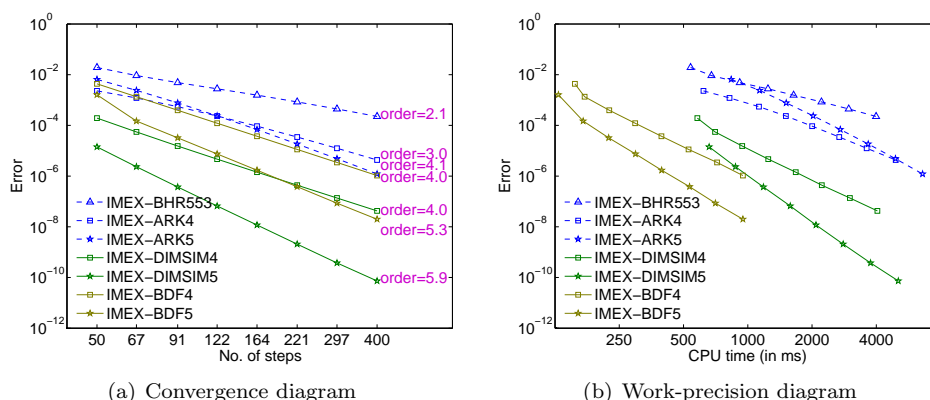


FIG. 5. Comparison of high order IMEX-DIMSIM, IMEX-ARK, and IMEX-BDF for the 2D Allen-Cahn equation (4.1). Shown are the temporal discretization errors corresponding to the solution at the final time  $t = 0.5$ .

the boundaries) are imposed. The spatial discretization is performed using a second-order central finite difference scheme on a uniform grid with  $\Delta x = \Delta y = 1/40$ .

Explicit time stepping methods have a maximal allowable time step  $h \propto \Delta x^2$  due to the CFL condition related to diffusion. To overcome this limitation we treat the stiff diffusion term implicitly and the remaining terms explicitly. Since the discrete diffusion term is linear we perform a single LU factorization of the matrix  $\mathbf{I} - h\gamma\mathbf{J}$  and reuse it throughout the simulation; here  $\gamma$  is a method coefficient and  $\mathbf{J}$  is the Jacobian of the stiff diffusion.

The reference solution  $u_{\text{ref}}$  is obtained using the MATLAB routine `ode15s` with very tight tolerances  $\text{AbsTol} = \text{RelTol} = 3 \times 10^{-14}$ . The absolute solution error magnitude is measured in the  $L_2$  norm:

$$(4.2) \quad \mathbf{E} = \|u - u^{\text{ref}}\|_2.$$

Figure 5(a) shows the errors at the final time for solutions computed using different numbers of steps. All three IMEX-ARK methods show mild order reduction—one order lower than the theoretical order. There is no order reduction for the IMEX-DIMSIM schemes; IMEX-DIMSIM4 displays the theoretical order while IMEX-DIMSIM5 shows a higher convergence than the theoretical order. The IMEX-DIMSIMs give considerably more accurate results than the IMEX-ARK methods. Surprisingly BHR553 is unable to avoid order reduction, although it has stage order of two. Therefore, higher stage order may still be demanded, but improving stage order beyond two for ARKs, if possible, would be difficult considering the complexity brought by additional coupling conditions. IMEX-BDFs do not lead to order reduction but generally produce larger errors than IMEX-DIMSIMs given the same step size.

The corresponding work-precision diagrams of errors versus CPU time are shown in Figure 5(b) and reveal a sizable gap in efficiency between IMEX-DIMSIMs and IMEX-ARKs. This can be expected because of the gap in accuracy observed in Figure 5(a) and the fact that IMEX-DIMSIMs have fewer stages than the IMEX-ARK methods of the same order and therefore require fewer function evaluations and linear solves per step. IMEX-BDFs solve the problem most efficiently with a low to medium level of accuracy mainly because only one linear solve, which usually takes most of the computation time, is needed at each time step. Figure 6 shows the spatial

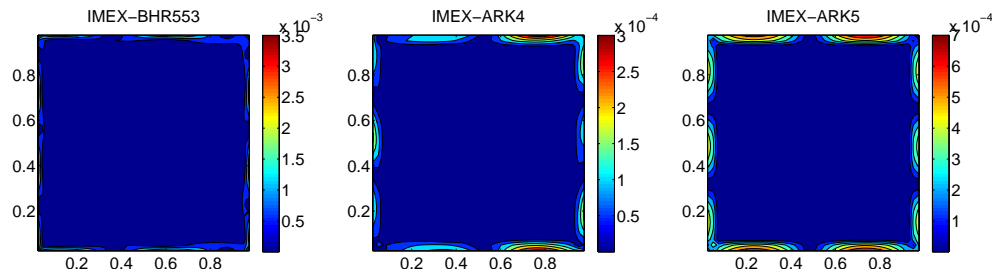


FIG. 6. Absolute temporal errors at the final time  $t = 0.5$  for various IMEX schemes on the 2D Allen-Cahn equation (4.1). A fixed time step of size  $h = 1/50$  is used. IMEX-ARK methods show large errors originating near boundaries and evenly distributed small errors in the interior of the domain.

distribution of the absolute errors  $|u_{\text{numerical}} - u_{\text{reference}}|$  at final time; this is only the temporal discretization error as we compare against a reference solution that uses the same spatial discretization. IMEX-ARK methods give large errors near boundaries and relatively smaller errors in the interior of the domain are evenly distributed. The order reduction phenomenon of IMEX-ARK methods originates with errors at the boundaries but plague the whole domain as the time evolves.

**4.2. Burgers equation.** The 2D viscous Burgers equation [5]

$$(4.3) \quad \frac{\partial u}{\partial t} + \frac{1}{2} \nabla(u \cdot u) = \nu \nabla^2 u, \quad \nu = 0.1, \quad 0 \leq x, y \leq 1, \quad 0 \leq t \leq 1,$$

is a simplification of the 2D Navier-Stokes equations which admits the analytic solution

$$u^{\text{analytic}}(t, x, y) = \left(1 + e^{\frac{x+y-t}{2\nu}}\right)^{-1}.$$

The initial conditions and the Dirichlet boundary values correspond to the analytic solution. Spatial derivatives are discretized with second-order central finite differences on a uniform grid with resolution  $\Delta x = \Delta y = 1/50$ .

The application of the IMEX integration treats the diffusion term implicitly and the convective term explicitly. We compare the numerical solutions against a reference solution computed with MATLAB routine *ode15s* with tolerances  $AbsTol = RelTol = 3 \times 10^{-14}$  that uses the same spatial discretization. Therefore the errors (4.2) reported here are only due to the temporal discretization.

Figure 7 compares the performance of various IMEX schemes. The convergence diagram in Figure 7(a) shows that the three IMEX-ARKs exhibit order reduction to order two. The IMEX-DIMSims and IMEX-BDF4 converge with their theoretical orders. IMEX-BDF5 clearly suffers from poor stability for large step sizes, which is indicated by the outlier points on the plot. If step size becomes too small, round-off errors may dominate. So the range of useful step sizes for IMEX-BDF5 is limited, making it difficult to show its full order in this convergence test. The efficiency diagram in Figure 7(b) illustrates again a gap in performance between IMEX-DIMSims and IMEX-ARKs, with IMEX-DIMSims demonstrating a considerably better efficiency than IMEX-ARKs. IMEX-BDFs are the most efficient—for step sizes where they are stable.

Figure 8 shows the spatial distribution of absolute errors at the final time. The boundary errors dominate the accuracy of the results for all schemes. The boundary

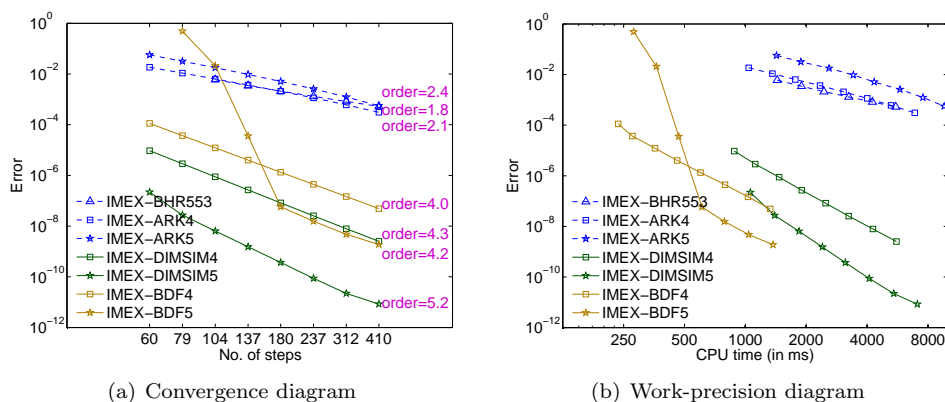


FIG. 7. Comparison of high order IMEX-DIMSIM and IMEX-ARK results for the 2D viscous Burgers equation (4.3). The integration time interval is  $[0, 1]$ . Shown are the temporal discretization errors corresponding to the solution at the final time  $t = 1$ .

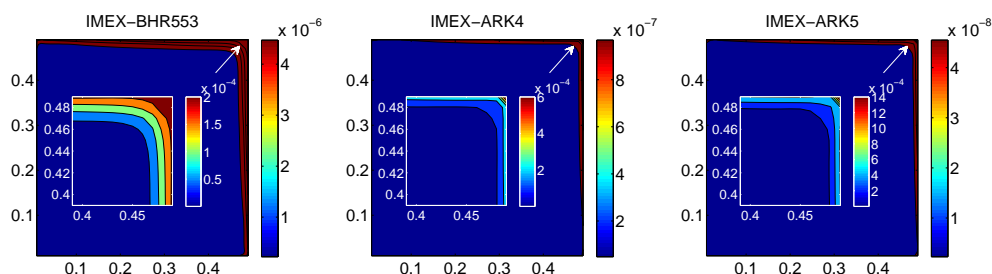


FIG. 8. Absolute temporal errors at the final time  $t = 1$  for various IMEX ARK schemes on the 2D viscous Burgers equation (4.3). A fixed time step of size  $h = 1/180$  is used. All methods show larger errors concentrating near the top and right boundaries.

conditions for this PDE may be more challenging than the previous one since they affects both spatial derivative terms in (4.3). Nevertheless, the error magnitude is much smaller for the IMEX-DIMSIM solutions.

### 4.3. Application to atmospheric simulations.

**4.3.1. Compressible Euler equations.** The dynamics of nonhydrostatic atmospheric processes can be described by the compressible Euler equations [25]:

$$\begin{aligned}
 (4.4a) \quad & \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \\
 & \frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \mathbf{u} + p \mathbf{I}) = -\rho g \hat{\mathbf{e}}_z, \\
 & \frac{\partial \rho \theta}{\partial t} + \nabla \cdot (\rho \theta \mathbf{u}) = 0,
 \end{aligned}$$

where  $\rho$  is the density,  $\mathbf{u} = (u, v, w)^T$  is the velocity vector,  $w$  being used in 3D case,  $\theta$  is the potential temperature, and  $\mathbf{I}$  is the identity matrix. The gravitational acceleration is denoted by  $g$ , while  $\hat{\mathbf{e}}_z$  is a unit vector pointing upward. The prognostic

variables are  $\rho$ ,  $\rho\mathbf{u}$ , and  $\rho\theta$ . The pressure  $p$  in the momentum equation is computed by the equation of state

$$(4.4b) \quad p = p_0 \left( \frac{\rho\theta R_d}{p_0} \right)^{\frac{c_p}{c_v}},$$

where  $p_0 = 10^5$  Pa is the surface pressure,  $R_d$  is the ideal gas constant, and  $c_p$  and  $c_v$  are the specific heat of the air for constant pressure and volume. To better maintain the hydrostatic state we follow the splitting introduced by Giraldo and Restelli [25],

$$\begin{aligned} \rho(\mathbf{x}, t) &= \bar{\rho}(z) + \rho'(\mathbf{x}, t), \\ (\rho\theta)(\mathbf{x}, t) &= \overline{(\rho\theta)}(z) + (\rho\theta)'(\mathbf{x}, t), \\ p(\mathbf{x}, t) &= \bar{p}(z) + p'(\mathbf{x}, t), \end{aligned}$$

where the overlined values are in hydrostatic balance. The governing equation (4.4) can then be rewritten as

$$(4.5a) \quad \begin{aligned} \frac{\partial \rho'}{\partial t} &= -\nabla \cdot (\rho\mathbf{u}), \\ \frac{\partial \rho\mathbf{u}}{\partial t} &= -\nabla \cdot (\rho\mathbf{u}\mathbf{u} + p'\mathbf{I}) - \rho'g\hat{\mathbf{e}}_z, \\ \frac{\partial (\rho\theta)'}{\partial t} &= -\nabla \cdot (\rho\theta\mathbf{u}) \end{aligned}$$

and closed with

$$(4.5b) \quad p' = p_0 \left( \frac{\rho\theta R_d}{p_0} \right)^{\frac{c_p}{c_v}} - \bar{p}.$$

The equations are discretized in space using the discontinuous Galerkin method, whose usage for geophysical simulations is gaining popularity, e.g., [19, 8, 34, 39, 25, 36, 22, 1, 32]. The model, based upon the mesh database of the GMSH mesh generator code [21], has been used to solve several PDEs, in the domains of geophysics [37, 31] and engineering [38, 30]. For more information about the space discretization, refer to [7].

The set of equations (4.5) applied to atmospheric flows is a good candidate for an IMEX time discretization, because of the different temporal scales involved. In usual atmospheric configurations, the acoustic waves are the fastest phenomena, with a propagation speed of about  $340 \text{ ms}^{-1}$ . This high celerity restricts the explicit time step to a small value due to the CFL stability condition. However, acoustic waves are generally not important for the modeler who is more interested in advective time scales. The IMEX method allows one to circumvent the CFL condition by treating the linear acoustic waves implicitly, while the remaining terms are explicit. According to Giraldo, Restelli, and Läuter [23], the right-hand side of (4.5a) is additively split into a linear part responsible for the acoustic waves and a nonlinear part. The linear term

$$(4.6) \quad - \begin{bmatrix} \nabla \cdot (\rho\mathbf{u}) \\ \nabla \cdot (p'\mathbf{I}) + \rho'g\hat{\mathbf{e}}_z \\ \nabla \cdot (\rho\theta\mathbf{u}) \end{bmatrix}$$

with the pressure linearized as

$$p' = \frac{c_p \bar{p}}{c_v \rho \bar{\theta}} (\rho\theta)'$$

is treated implicitly, while the remaining (nonlinear) terms are treated explicitly.



**4.3.2. Test cases.** In this paper we consider 2D and 3D rising thermal bubble test cases slightly modified from the ones introduced in [25].

*2D case.* The motion of the air is driven by a time varying potential temperature perturbation from the bottom boundary,

$$(4.7) \quad \theta' = \begin{cases} 0 & \text{for } r > r_c, \\ \frac{\theta_c}{2} \left( 1 + \cos\left(\frac{\pi r}{r_c}\right) \right) \sin^2\left(\frac{\pi t}{50}\right) & \text{for } r \leq r_c, \end{cases}$$

where  $\theta_c = 5^\circ\text{C}$ ,  $r = \sqrt{(x - x_c)^2}$ ,  $r_c = 250$  m, and  $(x, z) \in [0, 1000]^2$  with  $t \in [0, 200]$  s and  $x_c = 500$  m. No-flux boundaries are used for the other three boundaries. The computational domain is a 2D uniform mesh with actual resolution of  $15^2$  elements. Fourth-order polynomials are used on each element, yielding a grid resolution of about 13.3 m and an ODE system that contains  $\sim 2.3 \times 10^4$  variables.

*3D dimensional case.* The bottom boundary is also imposed as (4.7) with  $r = \sqrt{(x - x_c)^2 + (y - y_c)^2}$ ,  $r_c = 250$  m,  $(x, y, z) \in [200, 800]^2 \times [0, 600]$  m, and  $(x_c, y_c) = (500, 500)$  m. The integration time interval is  $[0, 150]$  s. No-flux boundaries are used for all the other boundaries. A 3D uniform mesh grid with actual resolution of  $8^3$  elements each with fourth-order polynomials is used. So the grid resolution is about 15 m, and the resulting ODE system has  $3.2 \times 10^5$  degrees of freedom. Diffusion terms

$$(4.8) \quad \begin{bmatrix} \nabla \cdot (\mu \nabla \rho') \\ \nabla \cdot (\mu \nabla (\rho \mathbf{u})) \\ \nabla \cdot (\mu \nabla (\rho \theta')) \end{bmatrix}$$

with  $\mu = 0.4 \text{ m}^2 \text{ s}^{-1}$  are added to the right-hand side of (4.5a) to limit the oscillations resulting from a high order spatial discretization of a complex flow on a coarse grid.

Figure 9 shows the reference solutions at the final time for 2D and 3D cases.

The 2D simulation is run in serial while the 3D simulation uses 16 MPI (message-passing interface) processes due to the much larger scale. The linear system arising in the 2D case is solved with the direct LU solver in PETSc [6]. For efficiency, the linear system from the 3D case is solved using the direct MUMPS [2] solver through PETSc. We have chosen a direct solver rather than an iterative solver for the following reasons:

- Integration with a fixed time step using singly diagonally schemes allows us to perform LU decomposition only once in the beginning of the simulation and reuse the result across all the time steps. This is particularly beneficial for long-time simulations.
- Direct solvers do not require one to consider issues such as initial guess, stopping criterion, and preconditioners, which, however, are critical for the performance of iterative solvers.
- Direct solvers could provide accurate solution so that the error from solving the linear system would influence less on the convergence study of time stepping methods.

**4.3.3. Numerical results.** The relative  $L_2$  errors for each of the prognostic variables

$$(4.9) \quad \mathbf{E}(q) = \sqrt{\frac{\int_{\Omega} (q^{\text{numerical}} - q^{\text{reference}})^2 d\Omega}{\int_{\Omega} (q^{\text{reference}})^2 d\Omega}}$$

are measured against a reference solution obtained by applying the classic fourth-order explicit RK method to solve the original (nonsplit) model with a very small

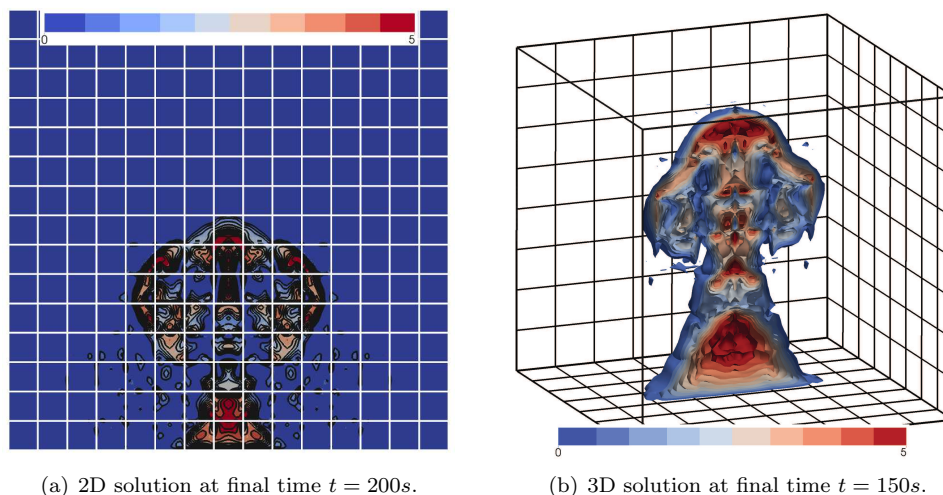


FIG. 9. Perturbation of potential temperature (in  $^{\circ}C$ ) from the simulation of thermal rising bubble governed by (4.5). The background mesh is displayed in wireframe.

time step  $h = 0.005s$ . Since the time varying boundary conditions imposed on the temperature  $\theta$  only affect the momentum equations of (4.5), we discuss the results for the variables  $\rho \mathbf{u}$ . For the 3D case, the  $u$  and  $w$  components in  $\mathbf{u}$  are considered due to symmetry of  $u$  and  $v$ .

Figure 10 compares the convergence results and efficiency for various IMEX methods for the 2D simulations. Each method is tested with nine different time step sizes, generating nine points if there is no stability issue. Any missing point on the plots, typically corresponding to a large time step, indicates that the solution explodes due to the stability restriction. This experiment leads to the following conclusions:

- As expected, all IMEX-DIMSIMs display their theoretical orders of accuracy.
- However, all IMEX-RK schemes of theoretical order higher than two show a clear order reduction, which translates into a loss of computational efficiency.
- The second-order IMEX-RK methods preserve their second order. Among them, ARS222 [3] yields poor accuracy and achieves only the level of  $10^{-4}$  even when the time step becomes as small as  $200/10248 \sim 0.02$  s.
- ARK2 [24] and BHR553 [10] have limited stability (as illustrated by the first four points missing from its curve).
- We have also tested IMEX-BDFs of orders two, three, and four [27]. Only BDF2 can solve this problem correctly; the others are unstable for all the selected time steps. This is not surprising since the Jacobian eigenvalues have large imaginary parts, as seen in Figure 11, and the implicit parts of higher order BDF methods are not A-stable. Among the three second-order methods, BDF2 is slightly more efficient than ARS222 but obviously less efficient than DIMSIM2.
- We note that high order methods are more efficient than low order methods; the new fourth-order method DIMSIM4 performs the best overall.
- We have also tested large step sizes and found that the maximal allowable step sizes for ARK4 and DIMSIM4 are both approximately equal to 1.0 s.

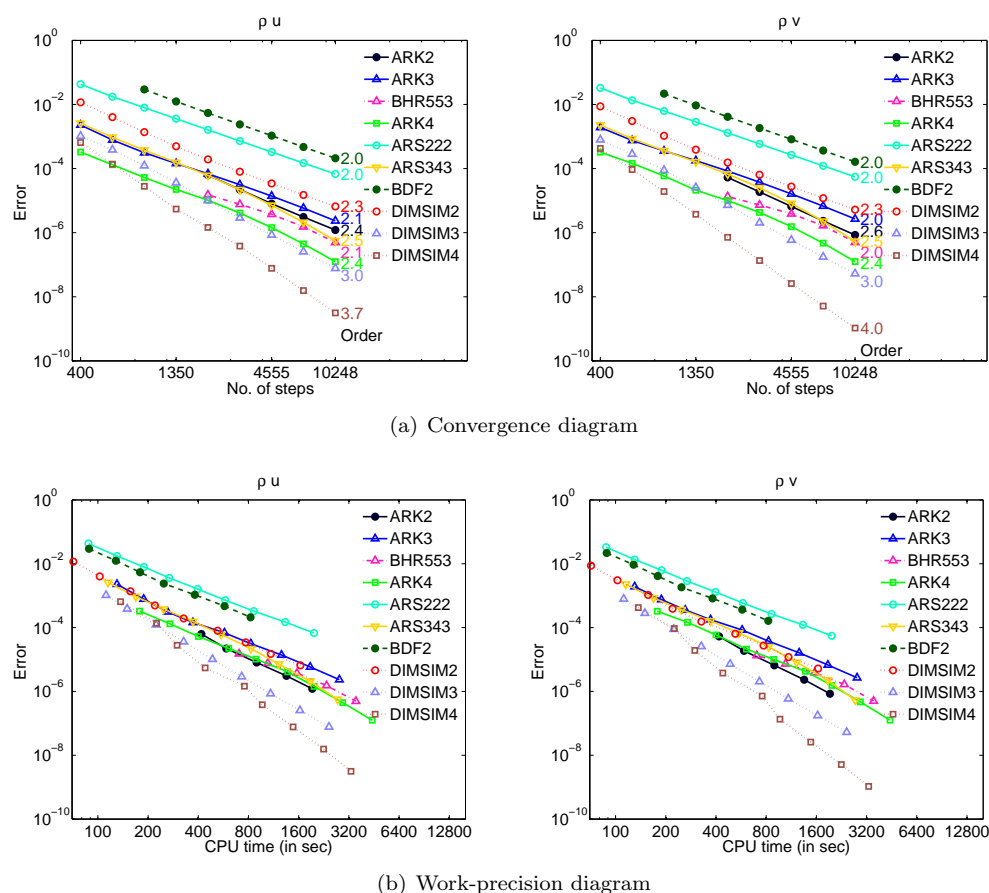


FIG. 10. Comparison of various IMEX time integration methods for the 2D rising bubble simulation (4.5). The integration time interval is  $[0, 200]$  s and is divided into 400, 600, 900, 1350, 2025, 3037, 4555, 6832, and 10,248 equal time steps to obtain the points in the diagrams. Temporal errors for all the variables (4.9) are computed for the solution at the final time.

This agrees with the prediction of the stability analysis in section 3.1 which shows that the IMEX-DIMSIM has a good stability property.

The 3D results given in Figure 12 lead to the following conclusions:

- IMEX-BDFs of orders three and four cannot solve this problem. This is due to the Jacobian eigenvalues having large imaginary parts. The efficiency of BDF2 is comparable to DIMSIM2 at a low accuracy level but becomes worse at a high accuracy level.
- ARK2 and BHR553 also fail for all the selected time steps, including the smallest value of  $150/1200 = 0.125$  s. This is likely due to their poor stability properties, as already indicated by the 2D results.
- The relative efficiency of different methods are the same as the 2D results. DIMSIM4 is still the winner. Some methods such as BDF2, ARS222, ARS343, and DIMSIM4 have a leftmost point which is obviously off track. This is because the methods are getting close to instability. It also affects the work-precision diagram in Figure 12. At these points, the linear system is becoming close to singular.
- Neither ARK5 nor DIMSIM5 is a suitable solver for this test problem because

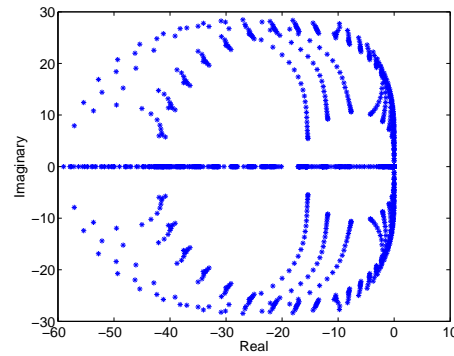


FIG. 11. Plot of eigenvalues of the Jacobian for 2D rising bubble test problem.

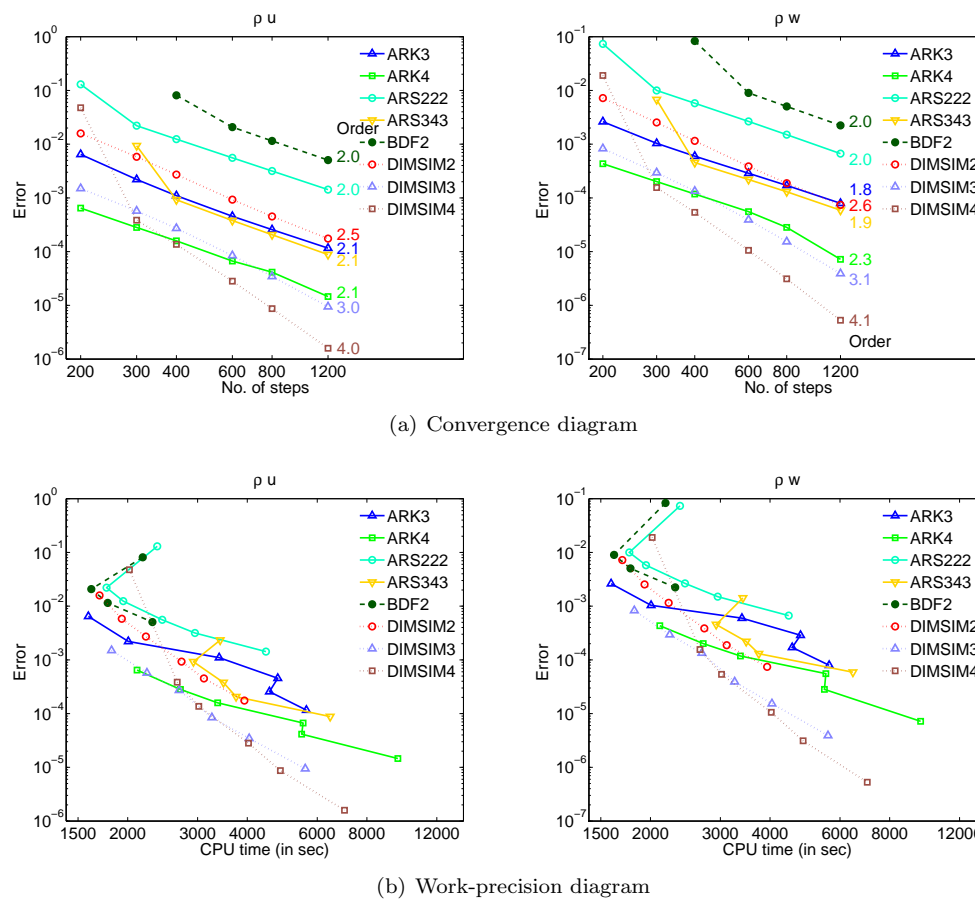


FIG. 12. Comparison of various IMEX time integration methods for the 3D rising bubble (4.5). The integration time interval is  $[0, 150]$  s and is divided into 150, 200, 250, 300, 350, 400, 600, 900, and 1350 equal time steps to obtain the points in the diagrams. Temporal errors for all the variables (4.9) are computed for the solution at the final time. BHR533 and ARK2 fail for any tested time steps due to instability and are therefore not present on this plot.

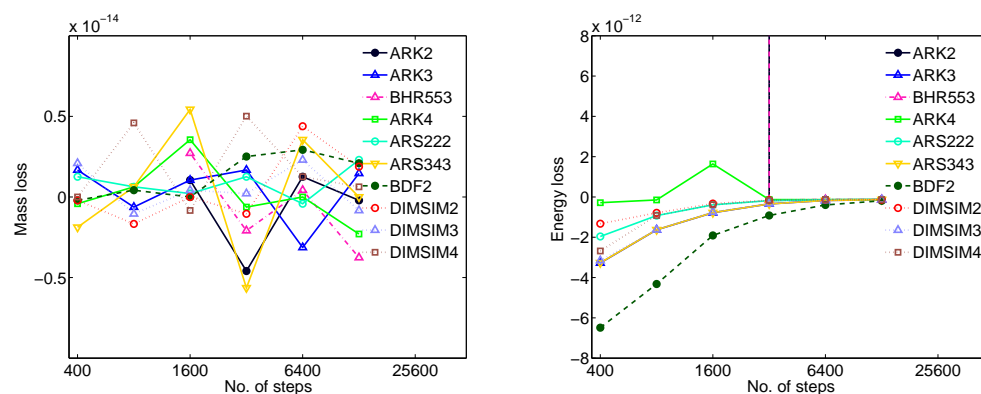


FIG. 13. Conservation properties of various IMEX time integration methods with different step sizes for the 2D rising bubble simulation (4.5). The integration time interval is  $[0, 200]$  s. and is divided into 400, 800, 1600, 3200, 6400, and 12,800 equal time steps to obtain the data points in the diagrams.

the maximal step sizes for them are restricted to values that are too small to make them competitive. For efficiency, a stability region covering a large part of the imaginary axis is required, since there are many eigenvalues of the Jacobian close to the imaginary axis.

Finally, we investigate the ability of the time integration methods to conserve mass and energy. Because the system has mass and energy exchanges at the bottom boundary, the above test problem is not a good one for testing the conservation properties. Alternatively we use the test case in [25], which is slightly different from ours; in that test case, the initial condition is

$$(4.10) \quad \theta' = \begin{cases} 0 & \text{for } r > r_c, \\ \frac{\theta_c}{2} \left( 1 + \cos\left(\frac{\pi r}{r_c}\right) \right) & \text{for } r \leq r_c, \end{cases}$$

and no-flux boundaries are used at all sides. The mass and energy loss are defined as

$$\frac{\text{Mass}(t_f) - \text{Mass}(t_0)}{\text{Mass}(t_0)}, \quad \text{Mass}(t) = \int_{\Omega} \rho \, d\Omega,$$

and

$$\frac{\text{Energy}(t_f) - \text{Energy}(t_0)}{\text{Energy}(t_0)}, \quad \text{Energy}(t) = \int_{\Omega} \rho e \, d\Omega,$$

respectively. The total energy is defined as the sum of internal, kinetic, and potential energies  $e(t) = c_v T(t) + (\mathbf{u} \cdot \mathbf{u})/2 + gz$ . The governing equations for this test conserve both mass and energy.

Figure 13 shows the mass and energy loss evaluated at the end of the simulation for all the IMEX methods with different step sizes. All the time integration methods can conserve mass up to machine precision. All of the methods except IMEX-BDF2 also conserve energy quite well, though not to machine precision. The exact mass conservation can be expected, because the continuity equation is treated implicitly in the IMEX splitting and the implicit parts of all the tested IMEX methods can preserve linear invariants. Furthermore, we can actually expect that if the mass is included partially in the implicit part and partially in the explicit part of the IMEX settings, ARS222 and IMEX-BDFs will not conserve mass because they do not satisfy the mass conservation conditions as explained in [24]. All proposed IMEX-DIMSims still conserve mass.

**5. Conclusions and future work.** Multiscale problems in science and engineering are modeled by time-dependent systems of equations involving both stiff and nonstiff terms. IMEX time stepping schemes perform an implicit integration only for the stiff components of the system and thus combine the low cost of explicit methods with the favorable stability properties of implicit methods.

Many modern PDE solvers use high order spatial discretization schemes, e.g., the discontinuous Galerkin approach with high degree polynomials. Often the high order of spatial discretization is paired with a low order traditional time stepping scheme. It is therefore of considerable importance to develop high order time stepping algorithms that match the accuracy of the spatial discretization.

This paper addresses the need for high order IMEX temporal discretizations in large-scale applications. We construct new fourth- and fifth-order IMEX DIMSIM schemes based on L-stable implicit components and with the explicit components optimized such as to maximize the constrained stability regions. The new methods have good stability properties and can take large step sizes for stiff problems.

Several test problems from different application areas that can benefit from IMEX integration are considered. These problems are the 2D Allen–Cahn and Burgers equations with finite difference spatial discretizations, and 2D and 3D compressible Euler equations with discontinuous Galerkin space discretizations. The performance of the new fourth- and fifth-order IMEX-DIMSIs is compared against existing IMEX schemes including multistage methods and multistep methods. While the IMEX BDF methods work best on the simpler test problems, IMEX DIMSIM methods show the best performance for the complex flow simulations. In all cases the IMEX-DIMSIs can use large step sizes—similar to those taken by traditional IMEX-RK methods. However, the high stage order enables our methods to avoid the order reduction that plagues classic IMEX-RK methods when applied to stiff systems or to problems with complex boundary conditions. Their stability properties allow them to solve problems where high order IMEX BDFs fail. In all cases IMEX-DIMSIs are considerably more efficient than the other methods. In addition, IMEX-DIMSIs show mass conservation within roundoff errors and show good behavior of the energy conservation even if the methods have not been specifically designed for this.

Typically multiscale flow simulations are carried out using fixed, predefined time steps. This is the approach taken in this paper as well. Ongoing work by the first two authors focuses on the development of adaptive stepsize IMEX-GLM schemes.

The high order IMEX-GLM schemes proposed herein are of interest not only to multiscale nonhydrostatic atmospheric simulations but also in many other fields where large-scale multiscale simulations are carried out with high order spatial discretizations. IMEX-GLMs can prove especially useful in situations where IMEX-RK methods suffer from order reduction; specific examples include stiff systems of singular perturbation type or problems with challenging time-dependent boundary conditions.

#### REFERENCES

- [1] V. AIZINGER AND C. DAWSON, *A discontinuous Aalerkin method for two-dimensional flow and transport in shallow water*, Adv. Water Resources, 25 (2002), pp. 67–84.
- [2] P. R. AMESTOY, I. S. DUFF, J. KOSTER, AND J.-Y. L'EXCELLENT, *A fully asynchronous multi-frontal solver using distributed dynamic scheduling*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 15–41.
- [3] U. ASCHER, S. RUUTH, AND R. SPITERI, *Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations*, Appl. Numer. Math., 25 (1997), pp. 151–167.

- [4] U. ASCHER, S. RUUTH, AND B. WETTON, *Implicit-explicit methods for time-dependent partial differential equations*, SIAM J. Numer. Anal., 32 (1995), pp. 797–823.
- [5] A. R. BAHADIR, *A fully implicit finite-difference scheme for two-dimensional burgers' equations*, Appl. Math. Comput., 137 (2003), pp. 131–137.
- [6] S. BALAY, S. ABHYANKAR, M. F. ADAMS, J. BROWN, P. BRUNE, K. BUSCHELMAN, V. EIJKHOUT, W. D. GROPP, D. KAUSHIK, M. G. KNEPLEY, L. C. MCINNES, K. RUPP, B. F. SMITH, AND H. ZHANG, PETSc Web page, <http://www.mcs.anl.gov/petsc> (2014).
- [7] S. BLAISE, J. LAMBRECHTS, AND E. DELEERSNIJDER, *A stabilization for three-dimensional discontinuous Galerkin discretizations applied to nonhydrostatic atmospheric simulations*, Internat. J. Numer. Methods Fluids (2015), doi: 10.1002/fld.4197.
- [8] S. BLAISE AND A. ST-CYR, *A dynamic hp-adaptive discontinuous Galerkin method for shallow-water flows on the sphere with application to a global tsunami simulation*, Monthly Weather Rev., 140 (2012), pp. 978–996.
- [9] S. BOSCARINO, *Error analysis of IMEX Runge–Kutta methods derived from differential-algebraic systems*, SIAM J. Numer. Anal., 45 (2007), pp. 1600–1621.
- [10] S. BOSCARINO, *On an accurate third order implicit-explicit Runge–Kutta method for stiff problems*, Appl. Numer. Math., 59 (2009), pp. 1515–1528.
- [11] S. BOSCARINO, L. PARESCHI, AND G. RUSSO, *Implicit-explicit Runge–Kutta schemes for hyperbolic systems and kinetic equations in the diffusion limit*, SIAM J. Sci. Comput., 35 (2013), pp. A22–A51.
- [12] S. BOSCARINO AND G. RUSSO, *On a class of uniformly accurate IMEX Runge–Kutta schemes and applications to hyperbolic systems with relaxation*, SIAM J. Sci. Comput., 31 (2009), pp. 1926–1945.
- [13] J. C. BUTCHER AND Z. JACKIEWICZ, *Diagonally implicit general linear methods for ordinary differential equations*, BIT, 33 (1993), pp. 452–472.
- [14] J. C. BUTCHER AND Z. JACKIEWICZ, *Construction of diagonally implicit general linear methods of type 1 and 2 for ordinary differential equations*, Appl. Numer. Math., 21 (1996), pp. 385–415.
- [15] J. C. BUTCHER AND Z. JACKIEWICZ, *Construction of high order diagonally implicit multistage integration methods for ordinary differential equations*, Appl. Numer. Math., 27 (1998), pp. 1–12.
- [16] J. C. BUTCHER AND W. WRIGHT, *The construction of practical general linear methods*, BIT, 43 (2003), pp. 695–721.
- [17] M. CALVO AND C. PALENCIA, *Avoiding the order reduction of Runge–Kutta methods for linear initial boundary value problems*, Math. Comp., 71 (2002), pp. 1529–1543.
- [18] X. CHEN, *Generation, propagation, and annihilation of metastable patterns*, J. Differential Equations, 206 (2004), pp. 399–437.
- [19] R. COMBLEN, S. BLAISE, V. LEGAT, J. REMACLE, E. DELEERSNIJDER, AND J. LAMBRECHTS, *A discontinuous finite element baroclinic marine model on unstructured prismatic meshes. Part I: space discretization*, Ocean Dynamics, 60 (2010), pp. 1371–1393.
- [20] J. FRANK, W. HUNSDORFER, AND J. VERWER, *On the stability of implicit-explicit linear multistep methods*, Appl. Numer. Math., 25 (1997), pp. 193–205.
- [21] C. GEUZAINÉ AND J. REMACLE, *GMSH: A 3-D finite element mesh generator with built-in pre- and post-processing facilities*, Internat. J. Numer. Methods Engrg., 79 (2009), pp. 1309–1331.
- [22] F. GIRALDO, J. HESTHAVEN, AND T. WARBURTON, *Nodal high-order discontinuous Galerkin methods for the spherical shallow water equations*, J. Comput. Phys., 181 (2002), pp. 499–525.
- [23] F. GIRALDO, M. RESTELLI, AND M. LÄUTER, *Semi-implicit formulations of the Navier–Stokes equations: Application to nonhydrostatic atmospheric modeling*, SIAM J. Sci. Comput., 32 (2010), pp. 3394–3425.
- [24] F. X. GIRALDO, J. F. KELLY, AND E. M. CONSTANTINESCU, *Implicit-explicit formulations of a three-dimensional nonhydrostatic unified model of the atmosphere (numa)*, SIAM J. Sci. Comput., 35 (2013), pp. B1162–B1194.
- [25] F. X. GIRALDO AND M. RESTELLI, *A study of spectral element and discontinuous Galerkin methods for the Navier–Stokes equations in nonhydrostatic mesoscale atmospheric modeling: Equation sets and test cases*, J. Comput. Phys., 227 (2008), pp. 3849–3877.
- [26] E. HAIRER, S. NORSETT, AND G. WANNER, *Solving Ordinary Differential Equations I. Nonstiff Problems*, Springer-Verlag, Berlin, 1993.
- [27] W. HUNSDORFER AND S. J. RUUTH, *IMEX extensions of linear multistep methods with general monotonicity and boundedness properties*, J. Comput. Phys., 225 (2007), pp. 2016–2042.
- [28] Z. JACKIEWICZ, *General Linear Methods for ODE*, Wiley, New York, 2009.

- [29] S. JEBENS, O. KNOTH, AND R. WEINER, *Partially implicit peer methods for the compressible Euler equations*, J. Comput. Phys., 230 (2011), pp. 4955–4974.
- [30] A. KAMENI, J. LAMBRECHTS, J. REMACLE, S. MEZANI, F. BOUILLAUT, AND C. GEUZAIN, *Discontinuous Galerkin method for computing induced fields in superconducting materials*, IEEE Trans. Magnetics, 48 (2012), pp. 591–594.
- [31] T. KARNA, V. LEGAT, AND E. DELEERSNIJDER, *A baroclinic discontinuous Galerkin finite element model for coastal flows*, Ocean Model., 61 (2013), pp. 1–20.
- [32] J. F. KELLY AND F. X. GIRALDO, *Continuous and discontinuous Galerkin methods for a scalable three-dimensional nonhydrostatic atmospheric model: Limited-area mode*, J. Comput. Phys., 231 (2012), pp. 7988–8008.
- [33] C. A. KENNEDY AND M. H. CARPENTER, *Additive Runge-Kutta schemes for convection-diffusion-reaction equations*, Appl. Numer. Math., 44 (2003), pp. 139–181.
- [34] R. NAIR, S. THOMAS, AND R. LOFT, *A discontinuous Galerkin global shallow water model*, Monthly Weather Rev., 133 (2005), pp. 876–888.
- [35] L. PARESCHI AND G. RUSSO, *Implicit-explicit Runge-Kutta schemes and applications to hyperbolic systems with relaxation*, J. Sci. Comput., 3 (2000), pp. 269–287.
- [36] D. SCHWANENBERG AND J. KÖNGETER, *A discontinuous galerkin method for the shallow water equations with source terms*, in Discontinuous Galerkin Methods, Lect. Notes Comput. Sci. Eng. 10, Springer, Berlin, 2000, pp. 419–424.
- [37] B. SENY, J. LAMBRECHTS, R. COMBLIN, V. LEGAT, AND J.-F. REMACLE, *Multirate time stepping for accelerating explicit discontinuous Galerkin computations with application to geophysical flows*, Internat. J. Numer. Methods Fluids, 71 (2013), pp. 41–64.
- [38] B. SENY, J. LAMBRECHTS, T. TOULORGE, V. LEGAT, AND J. REMACLE, *An efficient parallel implementation of explicit multirate Runge-Kutta schemes for discontinuous Galerkin computations*, J. Comput. Phys., 256 (2014), pp. 135–160.
- [39] A. ST-CYR AND D. NECKELS, *A fully implicit Jacobian-free high-order discontinuous Galerkin mesoscale flow solver*, in Computational Science—ICCS 2009, Lecture Notes in Comput. Sci. 5545, Springer, Berlin, 2009, pp. 243–252.
- [40] J. G. VERWER, B. P. SOMMEIJER, AND W. HUNSDORFER, *RKC time-stepping for advection-diffusion-reaction problems*, J. Comput. Phys., 201 (2004), pp. 61–79.
- [41] W. WRIGHT, *General Linear Methods with Inherent Runge-Kutta Stability*, Ph.D. thesis, University of Auckland, 2002.
- [42] H. ZHANG AND A. SANDU, *A second-order diagonally-implicit-explicit multi-stage integration method*, Procedia CS, 9 (2012), pp. 1039–1046.
- [43] H. ZHANG, A. SANDU, AND S. BLAISE, *Partitioned and implicit-explicit general linear methods for ordinary differential equations*, J. Sci. Comput., 61 (2014), pp. 119–144.