

Higher-Order Linear-Time Unconditionally Stable ADI Methods for Nonlinear Convection-Diffusion PDE Systems

Oscar P. Bruno*

Edwin Jimenez†

Abstract

We introduce a class of alternating direction implicit (ADI) methods, based on approximate factorizations of backward differentiation formulas (BDF) of order $p \geq 2$, for the numerical solution of two-dimensional, time-dependent, nonlinear, convection-diffusion PDE systems in Cartesian domains. The proposed algorithms, which do not require solution of nonlinear systems, additionally produce solutions with high-order accuracy in space through use of Chebyshev approximations. In particular, these methods give rise to reduced artificial dispersion and diffusion, and they therefore enable reductions on the sizes of the discretizations required to meet a prescribed error tolerance for a given problem. A variety of numerical results presented in this text demonstrate the claimed unconditional stability and high-order accuracy.

Keywords: Spectral method, unconditional stability, Fourier series, Fourier continuation, implicit solver, ADI, Burgers system.

1 Introduction

We introduce a class of alternating direction implicit (ADI) methods, based on approximate factorizations of backward differentiation formulas (BDF) of order $p \geq 2$, for the numerical solution of two-dimensional, time-dependent, nonlinear, convection-diffusion PDE systems in Cartesian domains. Like regular implicit time-marching methods, the algorithms proposed in this paper relax or altogether eliminate CFL stability constraints. Unlike previous implicit methods, however, the new approaches achieve unconditional stability without incurring the significant costs inherent in the nonlinear solves associated with the nonlinear convective terms. Additionally, they produce solutions with high-order accuracy in space and time. Thus, these methods, which do not require addition of numerical dissipation, give rise to reduced artificial dispersion and diffusion, and they therefore enable reductions on the sizes of the discretizations required to meet a prescribed error tolerance for a given problem. To our knowledge, these are the first spatially high-order algorithms in the literature for which unconditionally stability has been verified (if not rigorously proved) that exhibit, at the same time, high-order accuracy ($p = 2, 3$) in time. (The well known reference [2] presents an ADI algorithm of second-order of accuracy in time and space which, relying on use of numerical dissipation, enjoys unconditional stability.) Algorithms of even higher temporal accuracy ($p \geq 4$) with modest CFL constraints are also presented in this text, which could be of significant interest in certain contexts. All of these approaches are developed in conjunction with both, finite-difference and spectral spatial discretizations; in all cases the appropriate orders of temporal accuracy are verified, and unconditional stability ($p \leq 3$) is demonstrated.

(The use of fine spatial resolutions, which are often required to adequately represent complex domains with fine geometric features, boundary layers, turbulent solutions, etc, impose stringent numerical stability conditions for explicit time-marching methods; this effect is most pronounced for problems that include spatial diffusion. Implicit time-marching methods which, like the ones presented in this paper, can relax or altogether eliminate such numerical stability constraints, often do so at the expense of high computing costs. Indeed, a typical implicit step requires the inversion of a large generally nonlinear system of equations which, in multiple dimensions, can be very costly in terms of computation and memory requirements. The alternating direction implicit methods we use, in contrast, enjoy the enhanced stability inherent in regular implicit methods but they do so at reduced computing costs. Methods that can ensure high-order accuracy, both in time and in space, on the other hand, give rise to reduced artificial dispersion and diffusion, and they therefore enable reductions on the sizes of the discretizations required to meet a prescribed error tolerance.)

*Computing and Mathematical Sciences, Caltech, Pasadena, CA 91125, USA

†Computing and Mathematical Sciences, Caltech, Pasadena, CA 91125, USA

In contrast to other implicit methods, which must solve a multi-dimensional system of equations at every time step, an ADI algorithm evolves the solution of a multi-dimensional PDE one dimension at a time through the solution of a series of one-dimensional boundary-value problems. The present algorithms achieve high orders of temporal accuracy by means of novel factorizations of expressions resulting from BDF time discretizations. The ODE system solver framework we introduce for the one-dimensional ADI problems can be used in conjunction with various spatial approximations, including Chebyshev polynomials, Fourier continuation (FC) [1, 3, 5], and finite differences. The implementation of the proposed ADI schemes, further, is completely straightforward. For the sake of brevity, most of our examples concern the well-established Chebyshev spectral discretizations. Similar stability properties were observed in preliminary tests for spatial discretizations based on finite differences and the Fourier-continuation method [3, 5]. (Preliminary tests indicate that, while the FC-based solver requires somewhat finer discretizations for a given accuracy, it also gives rise to smaller numbers of GMRES iterations than the Chebyshev-based method in connection with certain necessary variable-coefficient ODE solves described in Section 4 below. The low-order finite-difference methods are, of course, much less efficient than either the Chebyshev- or FC-based algorithms.)

The derivation of our ADI schemes for a nonlinear PDE system relies on a few key observations. Most importantly, using the solution at time levels previous to $t = t^{n+1}$, the algorithm converts the nonlinear spatial operator into an implicit but linear operator with variable coefficients. The resulting approximately-factored equation is solved in “sweeps” along each of the Cartesian directions, including, as is common in ADI approaches, an intermediate “ $t^{n+1/2}$ ” step. All of the proposed algorithms are embodied in a single formula that includes BDF-based ADI methods of temporal orders $p = 1, \dots, 5$.

2 Alternating Direction Implicit Methods Based on Backward Differentiation Formulas

We derive a family of alternating direction implicit (ADI) methods of temporal orders as high as five for the numerical solution of time-dependent two-dimensional nonlinear convection-diffusion systems of partial differential equations (PDE) in Cartesian domains. Although our ADI methods are based on backward differentiation formulas (BDF), which are implicit methods for the numerical integration of ordinary differential equations, a similar strategy can in principle be used to derive ADI methods starting from other numerical ODE integration schemes. Our presentation begins with a brief review of BDF time-stepping methods for the numerical solution of systems of ordinary differential equations (ODE).

2.1 BDF Methods

Backward differentiation formulas (BDF) are implicit multi-step methods [6, p.492] for the numerical solution of initial-value problems of the form

$$\begin{cases} y'(t) = f(t, y(t)), & t \in (0, T] \\ y(0) = y_0, \end{cases} \quad (1)$$

where $f(t, y)$ is a given real-valued (scalar or vector) function in $C((0, T] \times \mathbb{R}^d)$, $T > 0$, and where d is a positive integer. Letting $\Delta t > 0$ and partitioning the integration interval $I := (0, T]$ into sub-intervals $I_n := (t^n, t^{n+1}]$ for $n = 0, \dots, N_{\Delta t}$, where $t^n = t^0 + n\Delta t$, a BDF method of order p approximates the value of $y'(t^{n+1})$ using the first derivative of the interpolating polynomial passing through $p + 1$ solution data-points at times $t^{n+1}, t^n, \dots, t^{n-p+1}$. The corresponding time-stepping algorithm takes the form

$$y^{n+1} = \sum_{m=0}^{p-1} a_m y^{n-m} + b \Delta t f^{n+1} + O(\Delta t^{p+1}). \quad (2)$$

Table 1 displays the BDF coefficients for $p = 1, \dots, 6$.

Table 1: Coefficients of BDF methods of order p for $p = 1, \dots, 6$.

p	a_0	a_1	a_2	a_3	a_4	a_5	b
1	1	0	0	0	0	0	1
2	$\frac{4}{3}$	$-\frac{1}{3}$	0	0	0	0	$\frac{2}{3}$
3	$\frac{18}{11}$	$-\frac{9}{11}$	$\frac{2}{11}$	0	0	0	$\frac{6}{11}$
4	$\frac{48}{25}$	$-\frac{36}{25}$	$\frac{16}{25}$	$-\frac{3}{25}$	0	0	$\frac{12}{25}$
5	$\frac{300}{137}$	$-\frac{300}{137}$	$\frac{200}{137}$	$-\frac{75}{137}$	$\frac{12}{137}$	0	$\frac{60}{137}$
6	$\frac{360}{147}$	$-\frac{450}{147}$	$\frac{400}{147}$	$-\frac{225}{147}$	$\frac{72}{147}$	$-\frac{10}{147}$	$\frac{60}{147}$

2.2 Two-dimensional Burgers System

This section introduces a class of BDF-based ADI methods for the nonlinear Burgers system

$$\begin{cases} u_t + 2uu_x + vu_y + uv_y = \nu \Delta u + f_u(x, y, t), & (x, y, t) \in U \times (0, T] \\ v_t + vu_x + uv_x + 2vv_y = \nu \Delta v + f_v(x, y, t), & (x, y, t) \in U \times (0, T] \\ u(x, y, t) = g_u(x, y, t), \ v(x, y, t) = g_v(x, y, t), & (x, y, t) \in \partial U \times (0, T] \\ u(x, y, 0) = u_0(x, y), \ v(x, y, 0) = v_0(x, y), & (x, y) \in U, \end{cases} \quad (3)$$

where $\nu > 0$ and $U \subset \mathbb{R}^2$ is a Cartesian domain. The final time $T > 0$, initial functions $\mathbf{u}_0(x, y) := (u_0(x, y), v_0(x, y))^T$, source functions $\mathbf{f}(x, y, t) := (f_u(x, y, t), f_v(x, y, t))^T$, and boundary-value functions $\mathbf{g}(x, y, t) := (g_u(x, y, t), g_v(x, y, t))^T$ are assumed to be given. For clarity, we re-express the system (3) in the vector form

$$\mathbf{u}_t + \mathcal{A}_1(\mathbf{u})\mathbf{u} + \mathcal{B}_1(\mathbf{u})\mathbf{u} = \nu \mathcal{A}_2\mathbf{u} + \nu \mathcal{B}_2\mathbf{u} + \mathbf{f}, \quad (4)$$

where $\mathbf{u} = (u, v)^T$ and where

$$\mathcal{A}_1(\mathbf{u}) = \begin{pmatrix} 2u\partial_x & 0 \\ v\partial_x & u\partial_x \end{pmatrix}, \quad \mathcal{B}_1(\mathbf{u}) = \begin{pmatrix} v\partial_y & u\partial_y \\ 0 & 2v\partial_y \end{pmatrix}, \quad (5a)$$

$$\mathcal{A}_2 = \begin{pmatrix} \partial_{xx} & 0 \\ 0 & \partial_{xx} \end{pmatrix}, \quad \mathcal{B}_2 = \begin{pmatrix} \partial_{yy} & 0 \\ 0 & \partial_{yy} \end{pmatrix}. \quad (5b)$$

We will sometimes simply write $\mathcal{A}_1$ and $\mathcal{B}_1$ instead of $\mathcal{A}_1(\mathbf{u})$ and $\mathcal{B}_1(\mathbf{u})$, respectively. Applying a BDF formula of order p to (4) we obtain the semi-discrete system

$$\mathbf{u}^{n+1} = \sum_{m=0}^{p-1} a_m \mathbf{u}^{n-m} + b\Delta t [-\mathcal{A}_1(\mathbf{u}^{n+1}) - \mathcal{B}_1(\mathbf{u}^{n+1}) + \nu \mathcal{A}_2 + \nu \mathcal{B}_2] \mathbf{u}^{n+1} + b\Delta t \mathbf{f}^{n+1} + O(\Delta t^{p+1})$$

i.e.,

$$\begin{aligned} [\mathcal{I} + b\Delta t(\mathcal{A}_1(\mathbf{u}^{n+1}) - \nu \mathcal{A}_2) + b\Delta t(\mathcal{B}_1(\mathbf{u}^{n+1}) - \nu \mathcal{B}_2)] \mathbf{u}^{n+1} = \\ \sum_{m=0}^{p-1} a_m \mathbf{u}^{n-m} + b\Delta t \mathbf{f}^{n+1} + O(\Delta t^{p+1}), \end{aligned} \quad (6)$$

where $\mathcal{I}$ denotes the identity operator.

To avoid the requirement of a nonlinear solve in our algorithms we use approximations, of certain orders of accuracy q , of the solution at time t^{n+1} that result from extrapolation of *known* solution values at previous time levels. The extrapolatory approximations we utilize for $\mathbf{u}^{n+1}$ are given by equations of the form

$$\tilde{\mathbf{u}}_q^{n+1} := \sum_{l=0}^{q-1} c_l \mathbf{u}^{n-l}, \quad (7)$$

Table 2: Extrapolation coefficients c_ℓ for $\mathbf{u}^{n+1} = \sum_{\ell=0}^{q-1} c_\ell \mathbf{u}^{n-\ell} + O(\Delta t^q)$.

q	c_0	c_1	c_2	c_3	c_4
2	2	-1	0	0	0
3	3	-3	1	0	0
4	4	-6	4	-1	0
5	5	-10	10	-5	1

where c_ℓ are coefficients that arise from consideration of a polynomial interpolating $\{\mathbf{u}^n, \dots, \mathbf{u}^{n-q+1}\}$ —so that, in particular, $\tilde{\mathbf{u}}_q^{n+1} = \mathbf{u}^{n+1} + O(\Delta t^q)$ as $\Delta t \rightarrow 0$. Table 2 contains the coefficients for an order q approximation of $\mathbf{u}^{n+1}$, for $q = 2, 3, 4, 5$.

Substituting $\mathbf{u}^{n+1}$ by $\tilde{\mathbf{u}}_p^{n+1}$ in the differential operators $\mathcal{A}_1$ and $\mathcal{B}_1$ in equations (5) we obtain the approximations

$$\tilde{\mathcal{A}}_1 := \mathcal{A}_1(\tilde{\mathbf{u}}_p^{n+1}) \approx \mathcal{A}_1(\mathbf{u}^{n+1}) \quad \text{and} \quad \tilde{\mathcal{B}}_1 := \mathcal{B}_1(\tilde{\mathbf{u}}_p^{n+1}) \approx \mathcal{B}_1(\mathbf{u}^{n+1})$$

in terms of *linear* differential operators $\tilde{\mathcal{A}}_1$ and $\tilde{\mathcal{B}}_1$ with variable coefficients. Using these approximations in equation (6) we obtain a linear problem for $\mathbf{u}^{n+1}$. Clearly, this substitution preserves the formal order of temporal accuracy inherent in the previous temporal discretization (6).

Next, we use the identity

$$\begin{aligned} \left[\mathcal{I} + b\Delta t(\tilde{\mathcal{A}}_1 - \nu\mathcal{A}_2) \right] \left[\mathcal{I} + b\Delta t(\tilde{\mathcal{B}}_1 - \nu\mathcal{B}_2) \right] \mathbf{u}^{n+1} &= \left[\mathcal{I} + b\Delta t(\tilde{\mathcal{A}}_1 - \nu\mathcal{A}_2) + b\Delta t(\tilde{\mathcal{B}}_1 - \nu\mathcal{B}_2) \right] \mathbf{u}^{n+1} \\ &\quad + (b\Delta t)^2 \left[\tilde{\mathcal{A}}_1 - \nu\mathcal{A}_2 \right] \left[\tilde{\mathcal{B}}_1 - \nu\mathcal{B}_2 \right] \mathbf{u}^{n+1} \end{aligned} \quad (8)$$

to obtain an approximate factorization of equation (6); note that, on account of the existing variable coefficients, the spatial operators in equation (8) in general do not commute. Adding $(b\Delta t)^2 [\tilde{\mathcal{A}}_1 - \nu\mathcal{A}_2][\tilde{\mathcal{B}}_1 - \nu\mathcal{B}_2]\mathbf{u}^{n+1}$ to both sides of (6) and using (8), we find

$$\begin{aligned} \left[\mathcal{I} + b\Delta t(\tilde{\mathcal{A}}_1 - \nu\mathcal{A}_2) \right] \left[\mathcal{I} + b\Delta t(\tilde{\mathcal{B}}_1 - \nu\mathcal{B}_2) \right] \mathbf{u}^{n+1} &= \sum_{m=0}^{p-1} a_m \mathbf{u}^{n-m} + b\Delta t \mathbf{f}^{n+1} + O(\Delta t^{p+1}) \\ &\quad + (b\Delta t)^2 \left[\tilde{\mathcal{A}}_1 - \nu\mathcal{A}_2 \right] \left[\tilde{\mathcal{B}}_1 - \nu\mathcal{B}_2 \right] \mathbf{u}^{n+1}. \end{aligned} \quad (9)$$

We make an additional approximation of $\mathbf{u}^{n+1}$, this time of order $p-1$, which we denote by $\hat{\mathbf{u}}^{n+1}$ (not to be confused with $\tilde{\mathbf{u}}^{n+1}$), in the right-hand side of (9); we obtain

$$\begin{aligned} \left[\mathcal{I} + b\Delta t(\tilde{\mathcal{A}}_1 - \nu\mathcal{A}_2) \right] \left[\mathcal{I} + b\Delta t(\tilde{\mathcal{B}}_1 - \nu\mathcal{B}_2) \right] \mathbf{u}^{n+1} &= \sum_{m=0}^{p-1} a_m \mathbf{u}^{n-m} + b\Delta t \mathbf{f}^{n+1} + O(\Delta t^{p+1}) \\ &\quad + (b\Delta t)^2 \left[\tilde{\mathcal{A}}_1 - \nu\mathcal{A}_2 \right] \left[\tilde{\mathcal{B}}_1 - \nu\mathcal{B}_2 \right] \hat{\mathbf{u}}^{n+1}; \end{aligned} \quad (10)$$

note that the $(p-1)$ -order approximation we use for $\mathbf{u}^{n+1}$ is sufficient in this context to maintain the extant order of accuracy.

Dropping terms of order Δt^{p+1} and higher we obtain the implicit (factored) time-marching scheme

$$\begin{aligned} \left[\mathcal{I} + b\Delta t(\tilde{\mathcal{A}}_1 - \nu\mathcal{A}_2) \right] \left[\mathcal{I} + b\Delta t(\tilde{\mathcal{B}}_1 - \nu\mathcal{B}_2) \right] \mathbf{v}^{n+1} &= \sum_{m=0}^{p-1} a_m \mathbf{v}^{n-m} + b\Delta t \mathbf{f}^{n+1} \\ &\quad + (b\Delta t)^2 \left[\tilde{\mathcal{A}}_1 - \nu\mathcal{A}_2 \right] \left[\tilde{\mathcal{B}}_1 - \nu\mathcal{B}_2 \right] \hat{\mathbf{v}}^{n+1}, \end{aligned}$$

which we express in the ADI form

$$\begin{cases} \left[\mathcal{I} + b\Delta t(\tilde{\mathcal{A}}_1 - \nu\mathcal{A}_2) \right] \mathbf{v}^{n+1/2} &= \sum_{m=0}^{p-1} a_m \mathbf{v}^{n-m} + b\Delta t(-\tilde{\mathcal{B}}_1 + \nu\mathcal{B}_2) \hat{\mathbf{v}}^{n+1} + b\Delta t \mathbf{f}^{n+1} \\ \left[\mathcal{I} + b\Delta t(\tilde{\mathcal{B}}_1 - \nu\mathcal{B}_2) \right] \mathbf{v}^{n+1} &= \sum_{m=0}^{p-1} a_m \mathbf{v}^{n-m} + b\Delta t(-\tilde{\mathcal{A}}_1 + \nu\mathcal{A}_2) \mathbf{v}^{n+1/2} + b\Delta t \mathbf{f}^{n+1}. \end{cases} \quad (11)$$

This is our p -th order alternating-direction BDF algorithm, which we denote by **BDF(p)-ADI**. Note that each sweep in (11) requires the solution of a system of variable coefficient ODEs.

3 Boundary Conditions for ADI Schemes

The ADI schemes derived in Section 2 require the solution of a system of ODEs in each sweep which must be supplemented with a proper set of boundary conditions to complete a properly posed boundary-value problem (BVP). The prescription of boundary conditions for $\mathbf{v}^{n+1}$ in the second sweep of (11) does not present a problem: we have $\mathbf{v} = \mathbf{g}$ on the boundary of U . Although the solution at the intermediate step is commonly labeled $\mathbf{v}^{n+1/2}$, on the other hand, this quantity does not approximate, in general, the solution at time $t^{n+1/2} = t^n + \Delta t/2$ with the appropriate order of accuracy: using for $\mathbf{v}^{n+1/2}$ boundary values given by $\mathbf{g}(t^{n+1/2})$ would in general degrade the order of accuracy of the overall solver.

To obtain consistent boundary conditions for $\mathbf{v}^{n+1/2}$ we use the ADI scheme itself. Starting from (11), we cross add the left-hand side and right-hand side terms and simplify to obtain

$$\mathbf{v}^{n+1/2} = \mathbf{v}^{n+1} + b\Delta t \left[\tilde{\mathcal{B}}_1 - \nu \mathcal{B}_2 \right] (\mathbf{v}^{n+1} - \hat{\mathbf{v}}^{n+1}). \quad (12)$$

Along the domain boundary (12) becomes

$$\mathbf{v}^{n+1/2} = \mathbf{g}^{n+1} + b\Delta t \left[\tilde{\mathcal{B}}_1 - \nu \mathcal{B}_2 \right] (\mathbf{g}^{n+1} - \hat{\mathbf{g}}^{n+1}), \quad (13)$$

which simplifies further to $\mathbf{v}^{n+1/2} = \mathbf{g}$ if the boundary functions $\mathbf{g}$ are time-independent. Without proof we note that, even for time-dependent boundary values $\mathbf{g}$, these boundary conditions achieve the desired order of accuracy. This fact, which is clearly demonstrated in Figure 2, can be established by considering the boundary-layer character of the error—which, for Δt small enough, gives rise to an additional exponentially small factor in the error arising from the boundary condition. Convergence of order p for arbitrary values of Δt can be achieved, further, by using approximations of order p instead of order $p - 1$ in equation (10).

4 Numerical Solution of ODE Systems with Variable Coefficients

Each one of the two equations in (11) requires the solution of a second-order variable-coefficient system of ODE of the form

$$\begin{cases} L\mathbf{u}(x) := P_2(x)\mathbf{u}''(x) + P_1(x)\mathbf{u}'(x) + P_0(x)\mathbf{u}(x) = \mathbf{r}(x), & -1 < x < 1, \\ \mathbf{u}(-1) = \mathbf{u}_l, \mathbf{u}(1) = \mathbf{u}_r, \end{cases} \quad (14)$$

for $\mathbf{u} := (u_1(x), u_2(x))^T$ over the interval $[-1, 1]$, where $\mathbf{u}_l := (u_{1,l}, u_{2,l})^T$ and $\mathbf{u}_r := (u_{1,r}, u_{2,r})^T$ are constant vectors. It is easy to check that the system (14) possesses a unique solution. In our algorithm, the solution to (14) is approximated by means of a Chebyshev grid $\{x_j\}_{j=1}^N \subset [-1, 1]$, where $x_j = -\cos(\pi(j-1)/(N-1))$.

Let V_N be a collection of discrete functions defined at the grid points x_j for $j = 1, \dots, N$. Letting $\mathbf{v}_N = (v_{1,N}, v_{2,N})^T \in V_N^2$, we will occasionally write $\mathbf{v}_j$ instead of $\mathbf{v}_N(x_j)$, for brevity. We note that the approximation of (14) can be succinctly re-stated as follows: find $\mathbf{v}_N \in V_N^2$ such that

$$\begin{cases} L_N \mathbf{v}_N(x_j) = \mathbf{r}(x_j), & j = 2, \dots, N-1, \\ \mathbf{v}_1 = \mathbf{u}_l, \mathbf{v}_N = \mathbf{u}_r, \end{cases} \quad (15)$$

where L_N , a discrete version of the operator L defined in (14), is defined as

$$L_N \mathbf{v}(x_j) := P_2(x_j) \mathbf{v}^{(2)}(x_j) + P_1(x_j) \mathbf{v}^{(1)}(x_j) + P_0(x_j) \mathbf{v}(x_j), \quad j = 2, \dots, N-1.$$

The approximate first and second derivatives $\mathbf{v}^{(1)}$ and $\mathbf{v}^{(2)}$ are evaluated as follows: letting $v_{1,N} = (v_{1,1}, \dots, v_{1,N})^T$ and $v_{2,N} = (v_{2,1}, \dots, v_{2,N})^T$, and letting $D = [d_{jk}]$, $j, k = 1, \dots, N$ denote the Chebyshev differentiation operator, then the m th derivative $v_{1,N}^{(m)}$ is computed as $v_{1,N}^{(m)} = D^m(v_{1,1}, \dots, v_{1,N})^T$ and similarly for $v_{2,N}$. Finally, the linear system (15) is vectorized and solved using GMRES. To accelerate the convergence of the GMRES solver, we use standard second-order finite difference approximations to (14) (that also satisfy the boundary conditions) as preconditioners.

5 Numerical Results

In this section we present examples that illustrate the accuracy and stability properties of the BDF-ADI methods developed in this text. In these examples spatial derivatives are evaluated by means of Chebyshev approximations, as described in the previous section, or, for comparison purposes, and to demonstrate the generality of the methodology proposed in this paper, by means of finite differences of order two. Denoting by $\mathcal{I}_N$, $\tilde{\mathcal{A}}_{1N}$, $\mathcal{A}_{2N}$, $\tilde{\mathcal{B}}_{1N}$, and $\mathcal{B}_{2N}$ the discrete approximations used for spatial differential operators $\mathcal{I}$, $\tilde{\mathcal{A}}_1$, $\mathcal{A}_2$, $\tilde{\mathcal{B}}_1$, and $\mathcal{B}_2$ respectively (resulting from e.g., Chebyshev differentiation, finite-differences, etc.), the boundary-value problems (11) take the fully discrete forms

$$\begin{cases} \left[\mathcal{I}_N + b\Delta t(\tilde{\mathcal{A}}_{1N} - \nu\mathcal{A}_{2N}) \right] \mathbf{v}_N^{n+1/2} &= \sum_{m=0}^{p-1} a_m \mathbf{v}_N^{n-m} + b\Delta t(-\tilde{\mathcal{B}}_{1N} + \nu\mathcal{B}_{2N}) \hat{\mathbf{v}}_N^{n+1} + b\Delta t \mathbf{f}^{n+1} \\ \left[\mathcal{I}_N + b\Delta t(\tilde{\mathcal{B}}_{1N} - \nu\mathcal{B}_{2N}) \right] \mathbf{v}_N^{n+1} &= \sum_{m=0}^{p-1} a_m \mathbf{v}_N^{n-m} + b\Delta t(-\tilde{\mathcal{A}}_{1N} + \nu\mathcal{A}_{2N}) \mathbf{v}_N^{n+1/2} + b\Delta t \mathbf{f}^{n+1}. \end{cases} \quad (16)$$

As mentioned in the previous section, these two boundary value problems (which correspond to the two discrete half-steps (11)) are solved by means of the iterative linear algebra solver GMRES.

Throughout this section we take the number of points along each dimension such that $N_x = N_y =: N$. In the Chebyshev case, the approximate solution $\mathbf{v}_N(x_j, y_k) = \mathbf{v}_{jk} \approx \mathbf{u}_{jk}$ is computed over (appropriately scaled versions of) the Chebyshev grids $x_j = -\cos(\pi(j-1)/(N-1))$ and $y_k = -\cos(\pi(k-1)/(N-1))$. For the finite-difference examples, in turn, we utilize the uniform mesh $\{(x_j, y_k)\}_{j=1, k=1}^{N_x, N_y} \subset U$, where $x_j = (j-1)\Delta x$, $y_k = (k-1)\Delta y$ and $\Delta x = (l_2 - l_1)/(N_x - 1)$, $\Delta y = (m_2 - m_1)/(N_y - 1)$.

In our first example we consider the system (3) over a Cartesian domain $U := [-3, 3]^2$. For the initial condition and boundary functions we use $\mathbf{u}_0(x, y) = (u_0(x, y), v_0(x, y))^T \equiv 0$ and $\mathbf{g}(x, y, t) = (g_u(x, y, t), g_v(x, y, t))^T \equiv 0$; the source terms, in turn, are set to $\mathbf{f} = (f_u(x, y, t), f_v(x, y, t))^T$ with

$$f_u(x, y, t) = Ae^{-r/\sigma^2} (y \cos(\theta) + x \sin(\theta)) \quad \text{and} \quad f_v(x, y, t) = Ae^{-r/\sigma^2} (y \sin(\theta) - x \cos(\theta)),$$

where $r = \sqrt{x^2 + y^2}$ and $\theta = t\pi/2$, and where $\sigma^2 = 0.4$ and $A = 100$. The solution of this problem is the rotating vortex depicted in Figure 1. This solution was obtained using $N = 100$ and $\Delta t = 10^{-2}$, for various times t in the interval $t \in (0, 5]$. The solution was computed by means of the BDF(3)-ADI method in conjunction with Chebyshev spatial differentiation operators. These images demonstrate the stability of the algorithm, in spite of the extremely small minimum Chebyshev mesh-size (which is of the order of 10^{-4}): for an explicit method the CFL constraint (time-step of the order of the square of the minimum mesh-size) requires $\Delta t \lesssim 10^{-8}$. In the present method, however we see that stability (with high-order accuracy) result from use of the time-step $\Delta t = 10^{-2}$ —without recourse to solution of a challenging nonlinear systems which are associated with classical implicit solvers.

In order to easily quantify errors, in our second example we consider a problem involving a manufactured solution $\mathbf{u} = (u, v)^T$ of (3) over $U = [0, 1]^2$ given by

$$u(x, y, t) = \sin(2\pi k_x(x+t)) \sin(2\pi k_y(y+t)) \quad (17a)$$

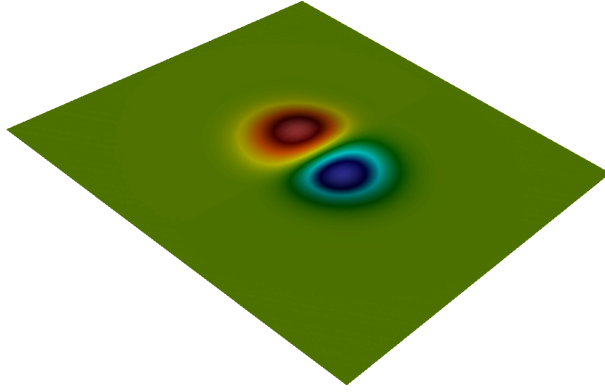
$$v(x, y, t) = \sin(2\pi k_x(x+t)) + \sin(2\pi k_y(y+t)) \quad (17b)$$

together with corresponding source terms $\mathbf{f}$ and Dirichlet boundary conditions. To verify the expected temporal order of accuracy in Figure 2 we present the maximum error

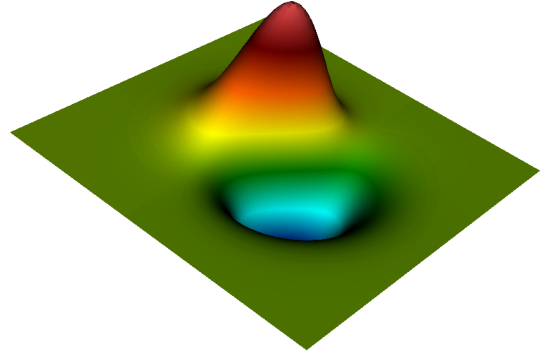
$$\|\mathbf{u} - \mathbf{v}\|_{\max} = \max_{0 \leq i, j \leq N} \{|u_{ij} - v_{ij}|\}$$

at the final time $T = 0.01$ versus a range of time step sizes Δt and several values of N . Derivatives and ODE systems are approximated using centered second-order finite differences (FD-2) and Chebyshev approximations. The results indicate that BDF(2)-ADI and BDF(3)-ADI achieve second-order and third-order temporal accuracy, as expected.

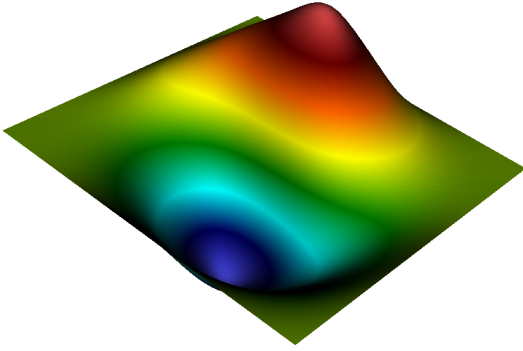
To conclude this section we use the solution (17) over $[0, 1]^2$ to demonstrate the unconditional stability of the proposed BDF-ADI schemes. We fix $\nu = 1$, the final time $T = 1000$, and the number of points $N = 20$ and solve the system (16) for time step sizes $\Delta t = 1, 10^{-1}, 10^{-2}$ with BDF-ADI(2) and BDF-ADI(3). Chebyshev approximations are used in all cases. Figure 3 shows that while the solution is in some cases inaccurate (as it should be, in view of the extremely coarse time-steps used), the maximum error remains bounded. Note that a typical explicit time-marching scheme coupled with a Chebyshev approximation would impose a stability constraint proportional to $1/N^4$ so that with $N = 20$, $\Delta t \approx 10^{-5}$ would be necessary for stability. The results presented in Figure 3 demonstrate that the proposed algorithms remain stable for values of Δt that are orders of magnitude beyond the stability limit required by explicit methods.



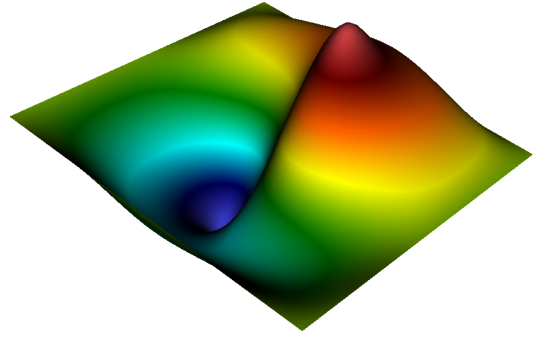
(a) $t = 0.03$



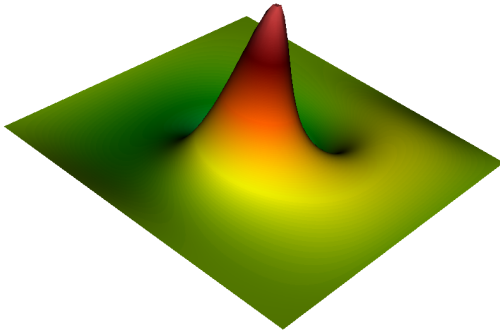
(b) $t = 0.3$



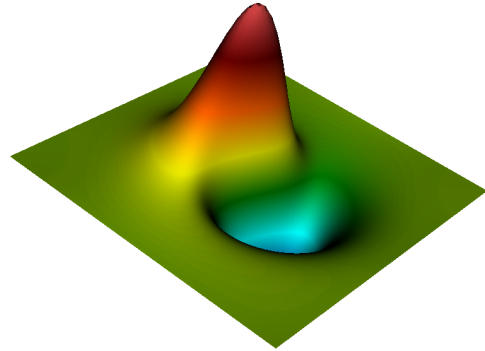
(c) $t = 1.0$



(d) $t = 2.0$



(e) $t = 3.2$



(f) $t = 4.0$

Figure 1: Burgers' system solution in a Cartesian domain using BDF(3)-ADI and Chebyshev approximations.

6 Conclusions

We have presented a class of alternating direction implicit methods based on approximate factorizations of backward differentiation formulas of order p for the numerical solution of a two-dimensional nonlinear PDE system of Burgers equations in a Cartesian domain. Our ADI schemes can be coupled with various spatial approximations such as standard or compact finite differences, Fourier continuation, and Chebyshev approximations. Thus, by combining different BDF(p)-ADI schemes and spatial approximations, an overall algorithm can be devised that is high-order in

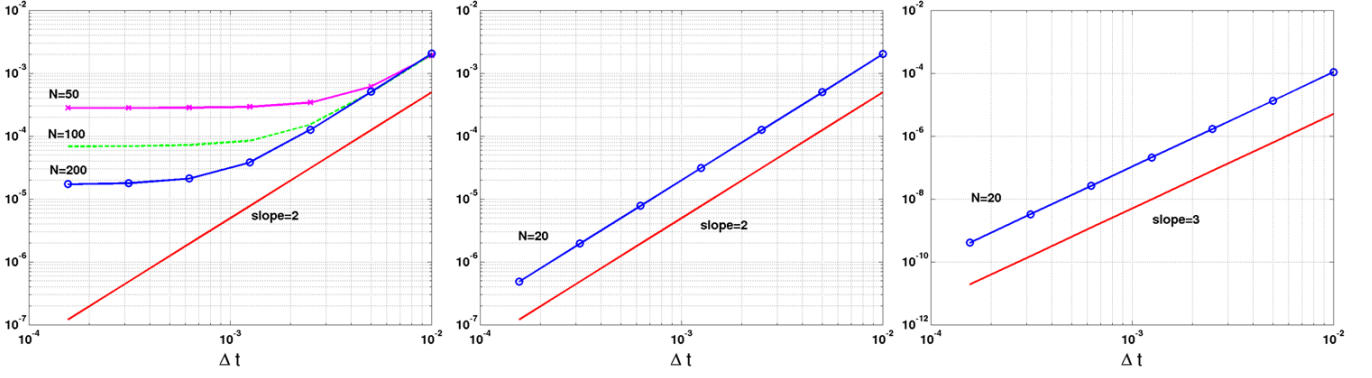


Figure 2: Temporal convergence as $\Delta t \rightarrow 0$, using various spatial resolutions ($N_x = N_y = N$), of the approximate solution to the system (3) over $[0, 1]^2$. Maximum errors versus time step size Δt obtained using (*left*) second-order finite differences ($N = 50, 100, 200$) and Chebyshev approximation with $N = 20$ points with (*center*) BDF(2)-ADI and (*right*) BDF(3)-ADI.

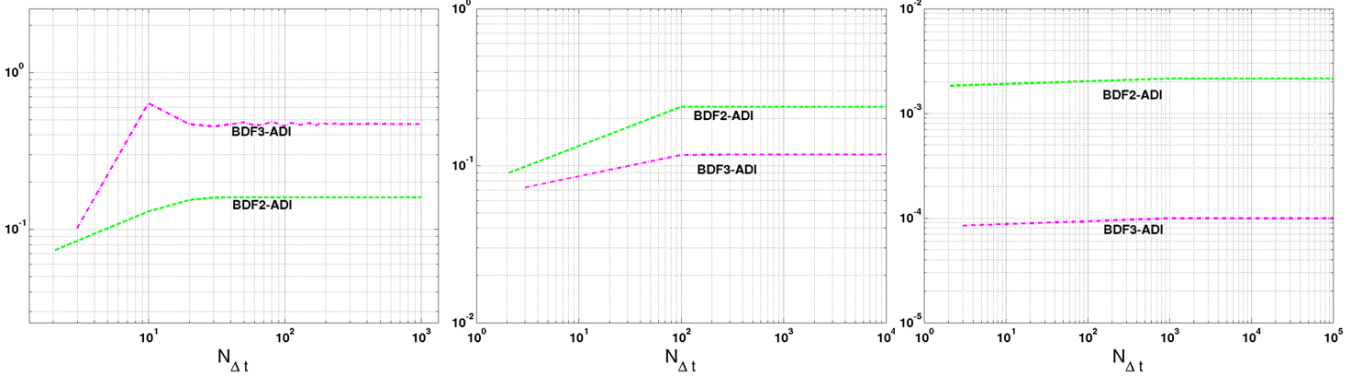


Figure 3: Stability of BDF(2)-ADI and BDF(3)-ADI coupled with Chebyshev approximations demonstrated by computing the maximum error at a final time $T = 1000$ with a fixed resolution $N = 20$ and time step sizes (*left*) $\Delta t = 1$, (*center*) $\Delta t = 10^{-1}$, and (*right*) $\Delta t = 10^{-2}$.

both time and space or even spectral in space if Chebyshev approximations are used. Clearly, suitable modifications of the proposed approaches could be used to enable solution of a variety of linear and nonlinear PDEs—including, for example, the Navier-Stokes equations.

As in the original linear Peaceman-Rachford method, our ADI schemes evolve the solution from one time-level to the next by means of the solution of a sequence of one-dimensional boundary-value problems. Unlike some recent ADI methods for *linear* problems [4], which require multiple fractional steps to achieve a high temporal order, our BDF(p)-ADI schemes utilize a single one-dimensional BVP solve per dimension, and they do not require Richardson’s extrapolation. The ODE system solver framework we presented, whose key ingredient is a preconditioned GMRES solver in the case of the dense matrices that result from Fourier continuation or Chebyshev approximations, or a banded linear system solver in the finite difference case, can be implemented in a straightforward manner.

Extensive numerical experiments indicate that the schemes BDF(2)-ADI and BDF(3)-ADI exhibit unconditional stability. On the other hand, while methods such as BDF(4)-ADI and BDF(5)-ADI appear to be subject to a stability constraint similar to a CFL condition, they nevertheless remain stable with time step sizes that are orders of magnitude larger than the stability limit imposed by explicit time-marching schemes for equations with second-order derivatives. Of course, only rigorous analysis of the underlying schemes can establish conclusively the true numerical stability limits of the methods presented; research in these regards is presently ongoing. While the ADI methods presented here were illustrated with initial-boundary value problems defined over Cartesian domains, preliminary results suggest that these approaches can also be applied in complex curvilinear domains; such extensions, however, have been left for future work.

Acknowledgments

The authors gratefully acknowledge support by the Air Force Office of Scientific Research and the National Science Foundation.

References

- [1] N. Albin and O. P. Bruno. A spectral FC solver for the compressible Navier–Stokes equations in general domains I: Explicit time-stepping. *Journal of Computational Physics*, 230(16):6248–6270, 2011.
- [2] R. M. Beam and R. F. Warming. An implicit factored scheme for the compressible Navier-Stokes equations. *AIAA Journal*, 16(4):393–402, April 1978.
- [3] O. P. Bruno and M. Lyon. High-order unconditionally stable FC-AD solvers for general smooth domains I. basic elements. *Journal of Computational Physics*, 229:2009–2033, 2010.
- [4] J. Lee and B. Fornberg. Some unconditionally stable time stepping methods for the 3d maxwell’s equations. *Journal of Computational and Applied Mathematics*, 166(2):497–523, 2004.
- [5] M. Lyon and O. P. Bruno. High-order unconditionally stable FC-AD solvers for general smooth domains II. elliptic, parabolic and hyperbolic PDEs; theoretical considerations. *Journal of Computational Physics*, 229:3358–3381, 2010.
- [6] A. Quarteroni, R. Sacco, and F. Saleri. *Numerical Mathematics*. Texts in Applied Mathematics. Springer, Paris, FR, 2000.