SIAM J. SCI. COMPUT.
Vol. 32, No. 5, pp. 3130–3150

OPTIMAL EXPLICIT STRONG-STABILITY-PRESERVING GENERAL LINEAR METHODS*

EMIL M. CONSTANTINESCU† AND ADRIAN SANDU‡

Abstract. This paper constructs strong-stability-preserving general linear time-stepping methods that are well suited for hyperbolic PDEs discretized by the method of lines. These methods generalize both Runge–Kutta (RK) and linear multistep schemes. They have high stage orders and hence are less susceptible than RK methods to order reduction from source terms or nonhomogeneous boundary conditions. A global optimization strategy is used to find the most efficient schemes that have low storage requirements. Numerical results illustrate the theoretical findings.

Key words. general linear methods, method of lines, strong-stability-preserving, monotonicity

AMS subject classifications. 65M20, 65L06

DOI. 10.1137/090766206

1. Introduction.

The numerical solution of time-dependent partial differential equations and nonlinear hyperbolic conservation laws are of great practical importance as they model diverse physical phenomena that appear in areas such as mechanical and chemical engineering, aeronautics, astrophysics, meteorology and oceanography, financial modeling, and environmental sciences. Representative examples for nonlinear hyperbolic conservation laws include gas dynamics, shallow-water flow, groundwater flow, non-Newtonian flows, traffic flows, and advection and dispersion of contaminants.

In the method of lines approach, the temporal and spatial discretizations are independent. Traditionally Runge–Kutta (RK) and linear multistep (LM) methods have been used for the integration of ODEs, DAEs, and semidiscrete, time-dependent PDEs. General linear (GL) methods [4, 6, 18, 25, 47], under various names (e.g., hybrid methods, pseudo-Runge–Kutta) represent a natural generalization of both RK and LM methods that are aimed at improving their stability and accuracy while taking advantage of precomputed information. They use both internal stages such as RK methods and information from previous solution steps such as LM methods.

The development of GL methods is challenging because of the order and stability constraints. Moreover, the solutions to hyperbolic PDEs may not be smooth: shock waves or other discontinuous behavior can develop even from smooth initial data. In such cases strong-stability-preserving (SSP) numerical methods that satisfy nonlinear

*Received by the editors July 24, 2009; accepted for publication (in revised form) July 19, 2010; published electronically October 14, 2010. This work was supported by the National Science Foundation through award NSF CCF-0515170. This work was created by the University of Chicago as Operator of Argonne National Laboratory under contract DE-AC02-06CH11357 with the U.S. Department of Energy. The U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes.

http://www.siam.org/journals/sisc/32-5/76620.html

†Mathematics and Computer Science Division, Argonne National Laboratory, 9700 S. Cass Avenue, Argonne, IL 60439 (emconsta@mcs.anl.gov). The work of this author was supported in part by the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under contract DE-AC02-06CH11357.

‡Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061 (asandu@cs.vt.edu).

stability requirements are necessary to avoid nonphysical behavior (spurious oscillations, etc.) [21, 44]. This aspect is illustrated by one of our numerical examples in Figure 7.3(a) and explained later in this paper. The GL methods are very robust schemes with a large number of degrees of freedom; however, little work has been done in the context of SSP methods. Previous work includes GL methods for linear problems or simplified GL representations [21, 29, 31, 48].

GL methods preserve the linear invariants of the underlying system. They are thus well suited for consistent discretizations of conservation laws, for example, conservation of mass and momentum. However, the algebraic complexity of the order and stability conditions prevents one from analytically crafting effective high-order GL methods. Therefore, numerical searches are employed in practice. In this context it is desirable to find the global optimal solution as exemplified by the search for optimal, SSP fourth-order explicit RK methods [37, 50].

In this research we are concerned with the numerical solution of nonlinear time-dependent partial differential equations in the method of lines approach. In this framework, the discretization of spatial operators yields a set of coupled time-dependent ordinary differential equations:

$$y'(t) = f\left(t, y(t)\right), \quad t_0 \le t \le t_{\text{Final}}, \quad y(t_0) = y_0\,, \tag{1.1}$$

where $y \in \mathbb{R}^N$ is the semidiscrete state and f represents a discrete version of the spatial operators. System (1.1) is nonautonomous. For brevity, however, we skip the time argument of f, unless noted otherwise. In this work we do not consider the adjoint operator of f [43], that is, its downwind version.

The purpose of this work is twofold. First, we investigate the theoretical aspects of the SSP property for a class of GL methods that is most likely to be useful in practice. Second, we construct new optimal SSP time-stepping schemes that can be readily used in practice—schemes with multiple stages and multiple steps, that is, multistep-multistage (MM) methods. These methods are a subset of GL schemes that can be represented in Shu–Osher form [45], and therefore simplifies both the SSP analysis [22] and implementation, which resembles multistep-RK methods. Specifically, in this study (i) we develop a transformation that allows MM methods to be expressed as GL methods; (ii) for this class of methods we find the global optimal explicit schemes of orders 2, 3, and 4 with any combination of 2, 3, and 4 stages and steps; and (iii) we explore the construction of such methods with high stage orders. To the best of our knowledge these are the first explicit multistage methods with high stage orders.

The rest of this paper is organized as follows. In section 2 we present background theory on GL methods and SSP time-stepping schemes. The representation of the proposed SSP GL methods is given in section 3. In section 4 we introduce a transformation that converts the proposed representation to the standard GL framework, and in section 5 we present the formulation of the optimization problem for finding the coefficients of the methods. We discuss the proposed methods and present several schemes in more detail in section 6. Numerical results with several GL schemes are presented in section 7, and a summary discussion is given in section 8.

2. General linear methods. Various types of GL methods were introduced in the 1960s either as extensions of Runge–Kutta methods [23] to multistep methods or vice versa [5, 20]. The current representation of GL methods and their name were coined by Burrage and Butcher [4] in the following way. Denote the solution at the current step $(n-1)$ by an r-component vector $\mathbb{y}_{[n-1]} = [\mathbb{y}^{(1)}_{[n-1]}\, \mathbb{y}^{(2)}_{[n-1]} \cdots \mathbb{y}^{(r)}_{[n-1]}]^T$, which contains the available information in the form of numerical approximations to

the ODE (1.1) solutions and their derivatives at different time indices. The stage values (at step n) are denoted by $\mathbf{Y}^{(i)}$ and stage derivatives by $\mathbf{F}^{(i)} = f\left(\mathbf{Y}^{(i)}\right)$, $i = 1, 2, \ldots, s$, and can be compactly represented as

$$\mathbf{Y} = \left[\mathbf{Y}^{(1)}\, \mathbf{Y}^{(2)} \ldots \mathbf{Y}^{(s)}\right]^T, \quad \mathbf{F} = \left[\mathbf{F}^{(1)}\, \mathbf{F}^{(2)} \ldots \mathbf{F}^{(s)}\right]^T.$$

The r-value s-stage GL method is described by

$$\begin{aligned} \mathbf{Y}^{(i)} &= \sum_{j=1}^{s} \mathbf{a}^{(i,j)} \Delta t \mathbf{F}^{(j)} + \sum_{j=1}^{r} \mathbf{u}^{(i,j)} \mathbb{y}_{[n-1]}^{(j)}, \; i = 1, 2, \ldots, s, \\ \mathbb{y}_{[n]}^{(i)} &= \sum_{j=1}^{s} \mathbf{b}^{(i,j)} \Delta t \mathbf{F}^{(j)} + \sum_{j=1}^{r} \mathbf{v}^{(i,j)} \mathbb{y}_{[n-1]}^{(j)}, \; i = 1, 2, \ldots, r, \end{aligned} \tag{2.1}$$

where $\mathbf{A} = [\mathbf{a}^{(i,j)}]$, $\mathbf{B} = [\mathbf{b}^{(i,j)}]$, $\mathbf{U} = [\mathbf{u}^{(i,j)}]$, and $\mathbf{V} = [\mathbf{v}^{(i,j)}]$ are the coefficients that define each method, and Δt is the time discretization step. The coefficients $(\mathbf{A}, \mathbf{U}, \mathbf{B}, \mathbf{V})$ are grouped further into the GL matrix $\mathbb{M}$:

$$\begin{bmatrix} \mathbf{Y} \\ \mathbb{y}_{[n]} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{U} \\ \mathbf{B} & \mathbf{V} \end{bmatrix} \begin{bmatrix} \Delta t \mathbf{F} \\ \mathbb{y}_{[n-1]} \end{bmatrix} = \mathbb{M} \begin{bmatrix} \Delta t \mathbf{F} \\ \mathbb{y}_{[n-1]} \end{bmatrix}.$$

expression (2.1) is the most generic representation of GL methods [24, p. 434] and encompasses both RK methods ($r = 1$, $s > 1$) and LM methods ($r > 1$, $s = 1$) as particular cases. In this work we consider methods with both $r > 1$ and $s > 1$.

If method (2.1) is consistent (there exist vectors x_1, x_2 such that $\mathbf{V}x_1 = x_1$, $\mathbf{U}x_1 = \mathbb{1}$, and $\mathbf{B}\mathbb{1} + \mathbf{U}x_2 = x_1 + x_2$ [10, Def. 3.2 and 3.3]) and stable ($\|\mathbf{V}^n\|$ remains bounded, $\forall n \geq 1$ [10, Def. 3.1]), then the method (2.1) is convergent [10, Thm. 3.5], [11].

Preliminary work on the convergence of GL methods has been carried out in [6, 8, 18, 25, 47]. An in-depth description and survey material on GL methods can be found in [9, 10, 11, 24].

The initial input vector $\mathbb{y}_{[0]}$ can be generated through a "starting procedure," $\mathbb{S} = \left\{S_i : \mathbb{R}^N \to \mathbb{R}^N\right\}_{i=1\ldots r}$, represented by generalized RK methods [11, Chap. 53]:

$$S_i = \begin{array}{c|c} c^{(i)} & \mathcal{A}^{(i)} \\ \hline b_0^{(i)} & \left(b^{(i)}\right)^T \end{array}, \quad \begin{aligned} Y^{(i)} &= \mathbb{1}y(x_0) + \Delta t \mathcal{A}^{(i)} F^{(i)}, \\ S_i &= b_0^{(i)} y(x_0) + \Delta t \left(b^{(i)}\right)^T F^{(i)}, \end{aligned} \tag{2.2}$$

where $\mathbb{1}$ is a vector of ones, $(\mathcal{A}, b, c)$ represents a classical RK scheme, and b_0 is a switch that defines the output of the method to be either the solution or its derivative. The final solution is typically obtained by applying a "finishing procedure," $\mathbb{F} : \mathbb{R}^N \to \mathbb{R}^N$, to the last output vector. We denote by the GL process the GL method applied n times and described by $\mathbb{S}\mathbb{M}^n\mathbb{F}$; that is, $\mathbb{M}$ is applied n times on the vector provided by $\mathbb{S}$, and then $\mathbb{F}$ is used to extract the final solution.

2.1. Order conditions for GL methods. Butcher [7] introduced an abstract representation of derivatives occurring in the Taylor expansion of the exact solution of (1.1). The derivatives are represented by *rooted tree* structures [7, 26], which are then used to algebraically characterize the order conditions for GL methods. Let $\mathbb{T}$ denote the set of rooted trees, and consider mappings of type $\Phi : \mathbb{T} \to \mathbb{R}$, which are called *elementary weight functions* and associate a scalar to each element of $\mathbb{T}$.

Let $\mathcal{T} \in \mathbb{T}$. Then $r(\mathcal{T})$ denotes the *order* of $\mathcal{T}$ and $\gamma(\mathcal{T})$ the *density* of $\mathcal{T}$. It is also useful to consider $E^{(\theta)} : \mathbb{T} \to \mathbb{R}$, the "exact solution operator" of differential equation (1.1), which represents the *elementary weights for the exact solution* at $\theta\Delta t$. If $\theta = 1$, then $E^{(1)}(\mathcal{T}) = E(\mathcal{T}) = 1/\gamma(\mathcal{T})$, and, in general, $E^{(\theta)}(\mathcal{T}) = \theta^{r(\mathcal{T})}/\gamma(\mathcal{T})$. The order can be analyzed algebraically by introducing a mapping $\xi_i : \mathbb{T} \to \mathbb{R}$: $\xi_i(\phi) = b_0^{(i)}$, $\xi_i(\mathcal{T}) = \Phi^{(i)}(\mathcal{T})$, where $\Phi^{(i)}(\mathcal{T})$, $i = 1, \ldots, r$, results from (2.2) and ϕ represents the "empty tree." Then for the general linear method $(\mathbf{A}, \mathbf{U}, \mathbf{B}, \mathbf{V})$, one has

$$\eta(\mathcal{T}) = \mathbf{A}\eta D(\mathcal{T}) + \mathbf{U}\xi(\mathcal{T}), \qquad \widehat{\xi}(\mathcal{T}) = \mathbf{B}\eta D(\mathcal{T}) + \mathbf{U}\xi(\mathcal{T}), \tag{2.3}$$

where η, ηD are mappings from $\mathbb{T}$ to scalars that correspond to the internal stages and stage derivatives, and $\widehat{\xi}$ represents the output vector. The exact weights are obtained from $[E\xi](\mathcal{T})$. The order of the GL method can be determined by a direct comparison between $\widehat{\xi}(\mathcal{T})$ and $[E\xi](\mathcal{T})$. The algebraic procedure described above is presented in more detail in [11], and a criterion for order p is given for a GL method described by $\mathbb{M}$ and $\mathbb{S}$. By using this general criterion, it is difficult to construct practical SSP GL and initializing methods because of the strong requirements placed on the starting procedure.

In this work we consider the GL methods started with the approximations of the exact solution up to order p for each step and order q for each stage, indicated by $\mathbb{S}[p, q]$. To this end we use appropriate SSP starting procedures [22, 37, 49] and a modified criterion for order conditions that considers the entire GL process: $\mathbb{S}\mathbb{M}^n\mathbb{F}$. This criterion is introduced in [16]. In this approach the order analysis is focused on the outcome of the GL process and has weaker constraints. Given a starting procedure $\mathbb{S}[p, q]$, an order p GL method with stage order q results from the direct comparison of elementary wights of $[\mathbb{S}\mathbb{M}^n\mathbb{F}](\mathcal{T}_p) = [E^n\xi](\mathcal{T}_p)$ $\forall \mathcal{T}_p$, $r(\mathcal{T}_p) \leq p$ and $[\eta_i](\mathcal{T}_q) = [E^{(\theta_i)}](\mathcal{T}_q)$ $\forall \mathcal{T}_q$, $r(\mathcal{T}_q) \leq q$, where θ_i is the time corresponding to stage i. This criterion is a direct consequence of [16, Def. 3 and Prop. 1]. In other words, the internal stages are approximated to the lowest possible order as long as the output selected by the finalizing procedure has the required order. The somewhat related but distinct concept of effective order [11] can lead to a higher order of accuracy.

2.2. Linear stability of GL methods. The linear stability analysis of method (2.1) is performed on a linear scalar test problem: $y'(t) = ay(t)$, $a \in \mathbb{C}$. Applying (2.1) to the test problem yields a solution of form $y^{n+1} = R(z)\, y^n$,

$$R(z) = \mathbf{V} + z\mathbf{B}\left(I - z\mathbf{A}\right)^{-1}\mathbf{U}, \tag{2.4}$$

where $z = a\Delta t$ and $R(z)$ is referred to as the stability matrix of the scheme.

For given z, method (2.1) is linearly stable if the spectral radius of $R(z)$ is contained by the complex unit disk. The stability region is defined as the set $\mathcal{S} = \{z \in \mathbb{C} : |R(z)| \leq 1\}$. The linear stability region provides valuable insight for the method's behavior with nonlinear systems. A similar approach to that for LM methods is used to compute the stability region for GL methods. Additional details can be found in [11].

2.3. Strong-stability-preserving time discretizations. Strong-stability-preserving integrators are high-order time-stepping schemes that preserve the stability properties of the spatial discretization used with explicit Euler time stepping. Spurious oscillations (nonlinear instabilities) can occur in a numerical solution that obeys

the classical linear stability conditions (von Neumann analysis) [22]. In PDEs with hyperbolic components an appropriate spatial discretization combined with an SSP time-stepping method yields a numerical solution that does not exhibit nonlinear instabilities. A nonlinear example shown in Figure 7.3(a) illustrates this behavior. In this section we review some background material on SSP methods.

DEFINITION 2.1 (strong stability [22, 38, 45]). *A sequence* $\{y_{[n]}\}$ *is said to be strongly stable in a given norm or seminorm* $||\cdot||$ *if* $||y_{[n]}|| \leq ||y_{[n-1]}|| \forall n \geq 1$.

The favorable properties of SSP schemes derive from convexity arguments. In particular, if the PDE semidiscretization with forward Euler method is strongly stable for any time step smaller than Δt_{FE} (i.e., $||y + \Delta t f(y)|| \leq ||y|| \forall \Delta t \leq \Delta t_{\text{FE}}$), then higher-order methods can be constructed as convex combinations of forward Euler steps with various step sizes [45]. For example an explicit s-stage RK method can be represented in Euler steps (also known as the Shu–Osher representation):

$$y^{(1)}_{[n-1]} = y_{[n-1]}\,, \tag{2.5a}$$

$$y^{(i)}_{[n-1]} = \sum_{j=1}^{i-1} \left[\alpha^{(i,j)} y^{(j)}_{[n-1]} + \beta^{(i,j)} \Delta t F^{(j)}_{[n-1]}\right] \;;\; i = 2, 3, \ldots, s, s+1\,, \tag{2.5b}$$

$$y_{[n]} = y^{(s+1)}_{[n-1]}\,. \tag{2.5c}$$

SSP methods preserve the strong stability of the forward Euler scheme for bounded time steps $\Delta t \leq \mathcal{C} \cdot \Delta t_{\text{FE}}$, where $\mathcal{C}$ is referred to as the CFL coefficient for the SSP property [22] or simply SSP coefficient [31].

THEOREM 2.2 (strong stability preserving for RK methods [22, 45]). *If the forward Euler method is strongly stable under the CFL restriction* $\Delta t \leq \Delta t_{\text{FE}}$*, then the RK method* (2.5) *with* $\alpha^{(i,j)}\,,\ \beta^{(i,j)} \geq 0$ *is SSP provided that* $\Delta t \leq \mathcal{C} \Delta t_{\text{FE}}$*, where* $\mathcal{C} = \min\left\{\, \alpha^{(\mathrm{i,j})}/\beta^{(\mathrm{i,j})} : 1 \leq \mathrm{i} \leq \mathrm{s},\, 1 \leq \mathrm{j} \leq \mathrm{i}-1,\, \beta^{(\mathrm{i,j})} \neq 0 \right\}$.

Methods with $\beta^{(i,j)} \leq 0$ are possible by using the adjoint operator of f (i.e., the downwind-biased spatial discretization of f) [22, 28, 43]; however, they are not addressed in this study.

The equivalence between the CFL coefficient and the radius of absolute monotonicity for multistep, multistage, and GL methods is discussed in [28, 31, 48]. In order to compare the efficiency of different methods, the scaled or effective CFL (or SSP [31]) coefficient, $\widehat{\mathcal{C}}$, is considered as the CFL coefficient divided by the number of function evaluations for one time step. Methods with high $\widehat{\mathcal{C}}$ allow large time steps and hence are more efficient.

Ketcheson [31] explores the limits for SSP GL methods with linear operators, which also represent efficiency barriers for methods with nonlinear operators. It is also noteworthy that the scaled CFL coefficient for an explicit GL method cannot exceed one. Huang [29] explores a type of hybrid method based on LM methods with one stage evaluation. In this work we extend the SSP concept to general linear methods (2.1) applied to nonlinear problems, and we search for the most efficient (i.e., the largest $\widehat{\mathcal{C}}$) GL schemes.

3. Multistep-multistage monotonic methods. We consider the following explicit k-step s-stage multistep-multistage method to compute the numerical solution

of (1.1) with time step Δt. The solution at step n, $y_{[n]} \approx y(t_{[n]}) = y(n\Delta t)$ is given by

$$y^{(1)}_{[n-1]} = y_{[n-1]}\,, \tag{3.1a}$$

$$\begin{aligned} y^{(i)}_{[n-1]} = &\sum_{\ell=2}^{k}\sum_{j=1}^{s}\left(\alpha^{(i,j)}_{[n-\ell]}y^{(j)}_{[n-\ell]} + \beta^{(i,j)}_{[n-\ell]}\Delta t F^{(j)}_{[n-\ell]}\right) \\ &+ \sum_{j=1}^{i-1}\left(\alpha^{(i,j)}_{[n-1]}y^{(j)}_{[n-1]} + \beta^{(i,j)}_{[n-1]}\Delta t F^{(j)}_{[n-1]}\right)\ ;\ i = 2,\,3,\ldots,\,s,s+1\,, \end{aligned} \tag{3.1b}$$

$$y_{[n]} = y^{(s+1)}_{[n-1]}\,, \tag{3.1c}$$

where $F^{(i)}_{[n-\ell]} = f(y^{(i)}_{[n-\ell]})$. We refer to $y^{(i)}_{[n-\ell]}$, $i = 1,\ldots,s$, $\ell = 0,\ldots,k$, as the stage i value at step $n-\ell$, and to $F^{(i)}_{[n-\ell]}$ as the corresponding stage derivative. The first sum in (3.1b) represents linear combinations of stage values and derivatives evaluated at previous steps, whereas the second sum describes the internal stages of the current step evaluation. Each stage value $y^{(i)}_{[n-\ell]}$ is an approximation to $y\left(t_{[n-\ell]} + c^{(i)}\Delta t\right)$. The *abscissae*, $c = [c^{(1)} = 0,\, c^{(2)},\,\ldots,\,c^{(s)},\,c^{(s+1)} = 1]^T$, can be shown to satisfy

$$c^{(1)} = 0\,,\ c^{(s+1)} = 1\,, \tag{3.2a}$$

$$\begin{aligned} c^{(i)} = 1 + &\sum_{\ell=2}^{k}\sum_{j=1}^{s}\left(\alpha^{(i,j)}_{[n-\ell]}\left(c^{(j)} - \ell\right) + \beta^{(i,j)}_{[n-\ell]}\right) \\ &+ \sum_{j=1}^{i-1}\left(\alpha^{(i,j)}_{[n-1]}\left(c^{(j)} - 1\right) + \beta^{(i,j)}_{[n-1]}\right)\ ;\ i = 2,\,3,\ldots,\,s\,. \end{aligned} \tag{3.2b}$$

Representation (3.1) can be seen as a generalization of the Shu–Osher representation [45] to MM methods.

With a harmless abuse of notation to avoid the Kronecker products, method (3.1) can be represented compactly by

$$\mathcal{Y}_{[n-1]} = e_1\, y_{[n-1]} + \sum_{\ell=1}^{k}\left(\Lambda_{[n-\ell]}\mathcal{Y}_{[n-\ell]} + \Gamma_{[n-\ell]}\Delta t F\left(\mathcal{Y}_{[n-\ell]}\right)\right)\,, \tag{3.3}$$

where $\mathcal{Y}_{[n-\ell]} = [y^{(1)}_{[n-\ell]}\, y^{(2)}_{[n-\ell]}\,\ldots\, y^{(s+1)}_{[n-\ell]}]^T$, $\Lambda_{[n-\ell]} = [\alpha^{(i,j)}_{[n-\ell]}]$, $\Gamma_{[n-\ell]} = [\beta^{(i,j)}_{[n-\ell]}]$, $1 \le i,j \le s+1$, and $e_1 = [1,\,0,\,\ldots,\,0]^T$. Schemes of type (3.3) with $k = 1$ are equivalent to the ones investigated by Shu and Osher [45], Gottlieb, Shu, and Tadmor [22], and Higueras [27, 28].

Because of the quantities that are transferred from one step to the next in (3.1), the concepts of method order and stage order are more difficult to define than for multistep or multistage methods. We introduce the following definition.

Definition 3.1 (consistency order for (3.1)). *Consider method* (3.1) *with the following properties:*

$$\begin{cases} y_{[m-k+\ell]} = y\left(t_{[m-k+\ell]}\right) + \mathcal{O}\left(\Delta t^{p+1}\right)\,, \\ y^{(i)}_{[m-k+\ell]} = y\left(t_{[m-k+\ell]} + c^{(i)}\Delta t\right) + \mathcal{O}\left(\Delta t^{q+1}\right)\,, \\ \quad 1 \le \ell \le k\,,\ 2 \le i \le s\,,\ n-k-1 \le m \le n-1. \end{cases} \tag{3.4a}$$

The k-step s-stage MM method (3.1) *with* (3.4a) *is said to be (at least) of order* p *and stage order* q *if the following expression holds for* $2 \le i \le s$, $n = 1, 2, \ldots$:

$$\text{(3.4b)} \qquad \begin{cases} y_{[n]} = y\left(t_{[n]}\right) + \mathcal{O}\left(\Delta t^{p+1}\right), \\ y^{(i)}_{[n-1]} = y\left(t_{[n-1]} + c^{(i)}\Delta t\right) + \mathcal{O}\left(\Delta t^{q+1}\right). \end{cases}$$

Remark. Expression (3.4a) for $m = 1$ is equivalent to the concept of the starting procedure for GL methods.

THEOREM 3.2 (strong stability preserving for MM methods). *If the forward Euler method is strongly stable under the CFL restriction* $\Delta t \le \Delta t_{\mathrm{FE}}$, *then the general linear method* (3.1) *with* $\alpha^{(i,j)}_{[n-\ell]}, \beta^{(i,j)}_{[n-\ell]} \ge 0$ *is SSP provided that* $\Delta t \le \mathcal{C}\Delta t_{\mathrm{FE}}$, *where*

$$\mathcal{C} = \min\left\{\alpha^{(i,j)}_{[n-\ell]}/\beta^{(i,j)}_{[n-\ell]} : 1 \le i \le \mathrm{s},\, 1 \le j \le i-1,\, 1 \le \ell \le k,\, \beta^{(\mathrm{i,j})}_{[\mathrm{n}-\ell]} \ne 0\right\}.$$

Proof. By consistency one has that $\sum_{j\ell} \alpha^{i,j}_{[n-\ell]} = 1$, $i = 1, \ldots, s+1$. The rest of the proof follows immediately from [22, 45]. ∎

We mentioned that GL and MM methods generalize both RK and LM methods. In what follows we present two examples of classical explicit schemes represented as MM (3.1) methods. We consider two-step linear multistep (Adams–Bashforth) method given by $s = 1$, $k = 2$ with $p = 2$ and RK methods with $s = 2$ (and $k = 1$) in Butcher tableau representation:

$$\text{(3.5)} \qquad y_{[n]} = y_{[n-1]} + \frac{3}{2}hF_{[n-1]} - \frac{1}{2}hF_{[n-2]},$$

$$\text{(3.6)} \qquad \begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}.$$

Their corresponding representation in form (3.1), (3.3) is given by the following coefficients:

$$\text{for (3.5)} \qquad \alpha^{(2,1)}_{[n-1]} = 1, \quad \beta^{(2,1)}_{[n-1]} = \frac{3}{2}, \quad \beta^{(2,1)}_{[n-2]} = -\frac{1}{2},$$

$$\text{for (3.6)} \qquad \alpha^{(2,1)}_{[n-1]} = \beta^{(2,1)}_{[n-1]} = 1,\ \alpha^{(3,1)}_{[n-1]} = \alpha^{(3,2)}_{[n-1]} = \beta^{(3,2)}_{[n-1]} = \frac{1}{2}.$$

4. Representation of multistep-multistage schemes as general linear methods. The convergence theory for GL methods has been developed for more than three decades. In order to take advantage of this body of work, the MM methods (3.1) that provide direct access to the SSP conditions need to be transformed into GL representation (2.1). We begin with (3.3) and consider

$$\mathcal{Y}_{[n-1]} = \sum_{\ell=2}^{k}\left(\Lambda_{[n-\ell]}\mathcal{Y}_{[n-\ell]} + \Gamma_{[n-\ell]}\Delta t F\left(\mathcal{Y}_{[n-\ell]}\right)\right) + e_1 y_{[n-1]} + \Lambda_{[n-1]}\mathcal{Y}_{[n-1]} + \Delta t\Gamma_{[n-1]}F\left(\mathcal{Y}_{[n-1]}\right).$$

The determinant of $\left(I - \Lambda_{[n-1]}\right)$ is one, and thus (3.1) can be expressed as

$$\mathcal{Y}_{[n-1]} = \sum_{\ell=2}^{k} \left(\overline{\Lambda}_{[n-\ell]}\mathcal{Y}_{[n-\ell]} + \Delta t\overline{\Gamma}_{[n-\ell]}F\left(\mathcal{Y}_{[n-\ell]}\right)\right) + \overline{e}y^{(1)}_{[n-1]} + \Delta t\mathbb{A}F\left(\mathcal{Y}_{[n-1]}\right),$$

where

$$\begin{aligned}
\overline{e} &= \left(I - \Lambda_{[n-1]}\right)^{-1} e_1\,,\\
\overline{\Lambda}_{[n-\ell]} &= \left(I - \Lambda_{[n-1]}\right)^{-1} \Lambda_{[n-\ell]}\,, \quad 2 \le \ell \le k\,,\\
\overline{\Gamma}_{[n-\ell]} &= \left(I - \Lambda_{[n-1]}\right)^{-1} \Gamma_{[n-\ell]}\,, \quad 2 \le \ell \le k\,, \text{ and}\\
\mathbb{A} &= \left(I - \Lambda_{[n-1]}\right)^{-1} \Gamma_{[n-1]} = \left[\mathcal{A}\, b^T\right]^T.
\end{aligned}$$

It follows that method (3.1) can be expressed as a GL scheme of form (2.1):

$$\mathbb{y}_{[n]} = \left[y_{[n]}, \mathcal{Y}^T_{[n-k+1]}, \ldots, \mathcal{Y}^T_{[n-1]}, \Delta t f(\mathcal{Y}_{[n-k+1]})^T, \ldots, \Delta t f(\mathcal{Y}_{[n-1]})^T\right]^T,$$

$$\mathbb{M} = \begin{bmatrix}
\mathbb{A} & \overline{\Lambda}_{[n-k]} & \overline{\Lambda}_{[n-k+1]} & \cdots & \cdots & \overline{\Lambda}_{[n-2]} & \overline{\Gamma}_{[n-k]} & \overline{\Gamma}_{[n-k+1]} & \cdots & \cdots & \overline{\Gamma}_{[n-2]}\\
0 & 0 & I & & & & & & & & \\
0 & 0 & 0 & \ddots & & & & & & & \\
\vdots & \vdots & \vdots & & I & & & & & & \\
0 & 0 & 0 & \cdots & 0 & I & & & & & \\
\mathcal{A} & \widehat{\overline{\Lambda}}_{[n-k]} & \widehat{\overline{\Lambda}}_{[n-k+1]} & \cdots & \cdots & \widehat{\overline{\Lambda}}_{[n-2]} & \widehat{\overline{\Gamma}}_{[n-k]} & \widehat{\overline{\Gamma}}_{[n-k+1]} & \cdots & \cdots & \widehat{\overline{\Gamma}}_{[n-2]}\\
0 & 0 & 0 & 0 & 0 & 0 & I & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \ddots & & & \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & & I & & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & I & \\
I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix},$$

where $\widehat{\overline{\Lambda}}$ and $\widehat{\overline{\Gamma}}$ are the first s rows of $\overline{\Lambda}$ and $\overline{\Gamma}$, respectively.

The order conditions and linear stability properties for the methods of type (3.1) considered in this study are analyzed in the GL method framework described in section 2.

We explore five types of methods that are differentiated by the amount of previous information used and implicitly by the search space for optimality. The input/output vectors—and hence the method use of past information—are illustrated in Table 1. GL methods of type one retain all available past information within s stages and k steps, type two retain only the past stage values and step derivatives, and type three keep only the past stage values. Type four use only the past step values—the least amount of information and memory. Type five is a hybrid of RK and LM methods. By using these five method types we implicitly restrict the amount of memory required by the GL schemes.

5. The optimization problem. The maximum coefficient $\mathcal{C}$, formed by α, β ratios, provides the optimal MM method. For a given MM scheme with specific s stages, k steps, and type, we search for an SSP GL method of order p and stage order q.

The most efficient SSP GL method is then given by the argument that maximizes a polynomial constrained mathematical programming problem described below. For

TABLE 1

The input/output vector components for different method types and a description of their use of past information ($k \geq 1$, $s \geq 1$, $1 \leq \ell \leq k$). The retained stage values and derivatives are represented by "□" and "⋆" symbols, respectively. Type one retains all available information within s stages and k steps, whereas type four uses the least (i.e., only the past step values). Type five is a hybrid method that resembles RK and LM methods.

	Type 1	Type 2	Type 3	Type 4	Type 5
Assump.		$\beta_{[n-\ell]}^{(i,j=2\ldots s)} = 0$	$\beta_{[n-\ell]}^{(i,j)} = 0$	$\alpha_{[n-\ell]}^{(i,j=2\ldots s)} = 0$ $\beta_{[n-\ell]}^{(i,j)} = 0$	$\alpha_{[n-\ell]}^{(i,j=2\ldots s)} = 0$ $\beta_{[n-\ell]}^{(i,j=2\ldots s)} = 0$
$y_{[n-k+\ell]}$ $= y_{[n-k+\ell]}^{(1)}$	□ ⋆	□ ⋆	□ –	□ –	□ ⋆
$y_{[n-k+\ell]}^{(2)}$	□ ⋆	□ –	□ –	– –	– –
⋮	⋮ ⋮	⋮ ⋮	⋮ ⋮	⋮ ⋮	⋮ ⋮
$y_{[n-k+\ell]}^{(s)}$	□ ⋆	□ –	□ –	– –	– –

an order p, stage order q MM method, with s stages and k steps, consider the triplet of indices $\Omega = \{(i,j,\ell) : 1 \leq i \leq s+1,\ 1 \leq j \leq s,\ 1 \leq \ell \leq k\}$. Then the optimization problem becomes

$$\mathcal{C} = \max \left(\min \left(\frac{\alpha_{[n-\ell]}^{(i,j)}}{\beta_{[n-\ell]}^{(i,j)}} \right)_{(i,j,\ell)\in\overline{\Omega}} \right), \quad \overline{\Omega} = \Omega / \{\beta_{[n-\ell]}^{(i,j)} = 0\} \tag{5.1a}$$

$$\text{subject to } [\mathbb{SM}^n\mathbb{F}](\mathcal{T}_p) = [E^n\xi](\mathcal{T}_p)\ \forall \mathcal{T}_p \in \mathbb{T},\ r(\mathcal{T}_p) \leq p\,, \tag{5.1b}$$

$$[\eta_i](\mathcal{T}_q) = [E^{(c^{(i)})}](\mathcal{T}_q)\ \forall \mathcal{T}_q \in \mathbb{T},\ r(\mathcal{T}_q) \leq q\,, \tag{5.1c}$$

$$0 \leq \alpha_{[n-\ell]}^{(i,j)} \leq 1\,,\ 0 \leq \beta_{[n-\ell]}^{(i,j)} \leq U_\beta\,. \tag{5.1d}$$

The values of β do not have an upper bound; however, the maximizer typically does not exceed the unit range for practical methods, and hence β values can be constrained to have an upper bound close to one without losing the global optimality of the solution. We also know that the maximum scaled CFL maximizer is less than or equal to one [31]. We therefore set $U_\beta = 5$, which guarantees $\widehat{\mathcal{C}} \geq \frac{1}{5s}$. The order conditions explained earlier [11, 16] are imposed in (5.1b) and (5.1c), and the SSP conditions are established in (5.1a) and (5.1d).

The setup of the numerical optimization problem is similar to the one described in [38]. GAMS (general algebraic modeling system) [3] is used to preprocess the problem. BARON [40] is then used to find the global maximizer with the default setting and extra parameters ConTol = 10^{-12}, EpsA = 10^{-10}, and EpsR = 10^{-5}. Within these limits, BARON guarantees global optimality and provides a maximizer that satisfies the equality and inequality constraints to at least 12 decimals—the maximum limit for BARON. More details regarding the mathematical problem setup can be found in [38]. A limit of 48 hours is imposed on the computational time. If the global solution is not found, the best feasible solution, which may not be globally optimal, is returned. These solutions are referred to as suboptimal.

To obtain practical method coefficients, a local search using *NLPSolve* in Maple is used to increase the solution precision to 15 decimals. Because of the problem complexity, there are instances in which the refinement approach fails to give a better solution in a fixed number of iterations. In these cases we show the partial solution,

which indicates a local or global maximum with a precision of 12 decimals.

6. SSP GL methods. In this section we present the SSP GL schemes obtained through the procedure described in this paper. We explore methods of orders $p = 2, 3, 4$ with $s, k = 2, 3, 4$ stages and steps and stage order $q = 1, \ldots, p$. Furthermore, we present several optimal SSP GL methods in more detail that are of orders 2, 3, and 4 with stage orders $q = p$ and $q = p - 1$. Complete results include $q = 1, \ldots, p$ and can be found in the unabridged technical report [17]. All methods have positive values for β with a fixed upper bound. The initial solution as produced by the SSP starting procedure $\mathbb{S}[p, q]$ provides the solution with an order of consistency at least equal to the orders p and q of the GL method under consideration for the initial steps and stages, respectively.

The proposed SSP GL schemes are denoted by GLpPqQsSkK, where P indicates the method order, Q the stage order, S the number of stages, and K the number of steps of the equivalent MM representation.

6.1. Second-order methods. The scaled CFL coefficient ($\widehat{\mathcal{C}}$) for the GL methods of order $p = 2$, $q = 1$ and $q = 2$ are summarized in Table 2. For each s and k configuration, where global convergence is achieved, all method types lead to the same result. It follows that methods of type four are a good representation for optimal second-order SSP GL methods. For a few second-order methods as well as for the other ones, global optimality was not achieved for all s, k configurations due to restrictions on the computational time. In these cases we provide the best solutions found.

The most efficient RK schemes with stage order one and $s = 2, 3, 4$ (Table 2, $k = 1$) have $\widehat{\mathcal{C}} = 0.50$, $\widehat{\mathcal{C}} = 0.67$, and $\widehat{\mathcal{C}} = 0.75$, respectively. The second-order GL methods presented in this work are more efficient than the existing SSP RK methods with the same number of stages (see Table 2 for $q = 1$). To the best of our knowledge there are no high-stage-order explicit SSP RK schemes. In this sense, in addition to the higher CFL coefficient, the GL methods with $q = 2$ summarized in Table 2 are superior to the classical SSP RK schemes.

Optimal LM schemes for up to 50 steps are presented in [31]. The optimal GL method with $k = 4$ ($s = 4$) has $\widehat{\mathcal{C}} = 0.93$ and is more efficient than the optimal LM scheme with $k = 4$, which has $\widehat{\mathcal{C}} = 0.66$. LM schemes with 15 steps are required to equal the efficiency of the proposed GLp2q2s4k4.

We next consider the optimal GL method with three stages and steps, stage order two, GLp2q2s3k3 (6.1), $\mathcal{C} = 2.57$ ($\widehat{\mathcal{C}} = 0.86$), which is described by the coefficients given below and requires five memory registers.

$$
\begin{array}{l}
\alpha^{(2,1)}_{[n-1]} = 0.973398050642691, \quad \beta^{(2,1)}_{[n-1]} = 0.379405979378177, \\
\alpha^{(3,2)}_{[n-1]} = 0.979404360713112, \quad \beta^{(3,2)}_{[n-1]} = 0.381747087369108, \\
\alpha^{(4,3)}_{[n-1]} = 0.983666449265926, \quad \beta^{(4,3)}_{[n-1]} = 0.383408341858481 \\
\hline
\alpha^{(2,1)}_{[n-3]} = 0.026601949357309, \\
\alpha^{(3,1)}_{[n-3]} = 0.020595639286888, \\
\alpha^{(4,1)}_{[n-3]} = 0.016333550734074 \\
\hline
c = [0,\, 0.326202080663559,\, 0.660039549070913\,,\, 1]^T .
\end{array}
\tag{6.1}
$$

TABLE 2

The scaled CFL coefficient, $\widehat{\mathcal{C}}$, for the optimal SSP GL methods with $p = 2$, $q = 1$, and $q = 2$. The superscript represents the number of memory registers required by each method. In most cases all method types lead to the same result. The subscripts indicate the method type if the results differ. Suboptimal results are denoted by light font face.

$q = 1$

$k\backslash s$	2	3	4
1	$\mathbf{0.50}^3$	$\mathbf{0.67}^3$	$\mathbf{0.75}^3$
2	$\mathbf{0.71}^4$	$\mathbf{0.82}^4$	$\mathbf{0.87}^4$
3	$\mathbf{0.81}^5$	$\mathbf{0.88}^5$	$\mathbf{0.91}^5$
4	$\mathbf{0.86}^6$	$\mathbf{0.91}^6$	$\mathbf{0.93}^6$,0.68_1^{15}

$q = 2$

$k\backslash s$	2	3	4
1	-	-	-
2	$\mathbf{0.59}^4$	$\mathbf{0.74}^4$	$\mathbf{0.81}^4$
3	$\mathbf{0.78}^5$	$\mathbf{0.86}^5$	$\mathbf{0.89}^5_{4,5}$,0.84_1^{21},$0.89^5_{2,3}$
4	$\mathbf{0.85}^6$	$\mathbf{0.90}^6$	$\mathbf{0.93}^6_{4,5}$,0.83_1^{14},$0.93^6_{2,3}$

For this case we illustrate the complete method as well as the quantities (enclosed in "[]") that need to be stored in memory after each step:

$$
\begin{aligned}
y^{(1)}_{[n-1]} &= y_{[n-1]}\,, && \big[y_{[n-1]}, y_{[n-2]}, y_{[n-3]}\big],\\
y^{(2)}_{[n-1]} &= \alpha^{(2,1)}_{[n-1]} y^{(1)}_{[n-1]} + \beta^{(2,1)}_{[n-1]} \Delta t F^{(1)}_{[n-1]} + \alpha^{(2,1)}_{[n-3]} y^{(1)}_{[n-3]}\,, && \big[y^{(2)}_{[n-1]}, F^{(2)}_{[n-1]}, y_{[n-1]}, y_{[n-2]}, y_{[n-3]}\big],\\
y^{(3)}_{[n-1]} &= \alpha^{(3,2)}_{[n-1]} y^{(2)}_{[n-1]} + \beta^{(3,2)}_{[n-1]} \Delta t F^{(2)}_{[n-1]} + \alpha^{(3,1)}_{[n-3]} y^{(1)}_{[n-3]}\,, && \big[y^{(3)}_{[n-1]}, F^{(3)}_{[n-1]}, y_{[n-1]}, y_{[n-2]}, y_{[n-3]}\big],\\
y^{(4)}_{[n-1]} &= \alpha^{(4,3)}_{[n-1]} y^{(3)}_{[n-1]} + \beta^{(4,3)}_{[n-1]} \Delta t F^{(3)}_{[n-1]} + \alpha^{(4,1)}_{[n-3]} y^{(1)}_{[n-3]}\,, && \big[y^{(4)}_{[n-1]}, y_{[n-1]}, y_{[n-2]}\big],\\
y_{[n]} &= y^{(4)}_{[n-1]}\,. && \big[y_{[n]}, y_{[n-1]}, y_{[n-2]}\big],
\end{aligned}
$$

The most efficient LM method with three steps ($k = 3$) has $\widehat{\mathcal{C}} = 0.5$. LM schemes require at least nine steps ($k = 9$) [31] to equal the same efficiency as GLp2q2s3k3.

6.2. Third-order methods. The scaled CFL coefficient for the SSP GL methods $p = 3$ and $q = 1$ are summarized in Table 3. The most efficient RK schemes with stage order one and $s = 3, 4$ (Table 3, $k = 1$) have $\widehat{\mathcal{C}} = 0.33$ and $\widehat{\mathcal{C}} = 0.50$, respectively. The proposed SSP GL methods with $\widehat{\mathcal{C}}$ ranging from 0.55 to 0.58 for $s = 3$ and 0.58 for $s = 4$ are more efficient than the aforementioned classical SSP RK methods. We also note that the optimal GL methods with $s = 2$, $q = 2$ of type five have also been discovered by Spijker [48].

Methods of order $p = 3$, $q = 2$ are summarized in Table 4 and methods with $q = 3$ in Table 5. The most efficient third-order SSP LM scheme with $k = 4$ has $\widehat{\mathcal{C}} = 0.33$; there are no third-order LM methods with less than four steps. The maximum CFL coefficient attained by an LM method is 0.58 (for $k \geq 6$ [31]). The proposed SSP GL methods reach this efficiency in four steps; furthermore, SSP GL schemes with less than four steps are possible.

Not all GL methods are globally optimal; therefore, it is possible to find more efficient GL schemes. The most efficient third-order schemes found within the allocated time frame are shown in Tables 3–5.

We select two methods, GLp3q2s3k2 and GLp3q3s2k3, and investigate their properties in more detail. The optimal SSP GL method with stage order two, three stages, and two steps, GLp3q2s3k2 (6.2), has $\mathcal{C} = 1.65$ ($\widehat{\mathcal{C}} = 0.55$), and requires six memory

TABLE 3

The scaled CFL coefficient, $\widehat{\mathcal{C}}$, for the optimal (bold face) and suboptimal (light face) SSP GL methods with $p = 3$, $q = 1$. The superscript represents the number of memory registers required by each method and the subscript the type of method.

$k\backslash s$	2	3	4
1	-	$\mathbf{0.33}^{3}$	$\mathbf{0.50}^{3}$
2	$\mathbf{0.37}^{6}_{1,2,5}$,$\mathbf{0.12}^{5}_{3,4}$	$\mathbf{0.55}^{6}_{1,2,5}$,$\mathbf{0.42}^{6}_{3,4}$	0.58^{6}_{1},$0.58^{7}_{2,5}$,0.53^{5}_{3},$\mathbf{0.53}^{5}_{4}$
3	$\mathbf{0.56}^{8}_{1,2,5}$,$\mathbf{0.20}^{6}_{3,4}$	$\mathbf{0.58}^{8}_{1,2,5}$,$\mathbf{0.42}^{6}_{3,4}$	0.58^{6}_{1},$0.58^{7}_{2,5}$,0.53^{5}_{3},0.53^{5}_{4}
4	$\mathbf{0.57}^{10}_{1,2,5}$,$\mathbf{0.20}^{6}_{3,4}$	$0.58^{8}_{1,2,5}$,$\mathbf{0.42}^{6}_{3,4}$,$\mathbf{0.58}^{8}_{5}$	$0.58^{7}_{1,2,5}$,$0.53^{5}_{3,4}$

TABLE 4

The scaled CFL coefficient, $\widehat{\mathcal{C}}$, for the optimal (bold face) and suboptimal (light face) SSP GL methods with $p = 3$, $q = 2$. The results for methods that are not refined to full double precision are underlined. The superscript represents the number of memory registers required by each method and the subscript the type of method.

$k\backslash s$	2	3	4
1	-	-	-
2	$\mathbf{0.37}^{6}_{1,2,5}$,$\mathbf{0.13}^{5}_{3,4}$	$\mathbf{0.55}^{6}_{1,2,5}$,$\mathbf{0.39}^{5}_{3,4}$	$\underline{0.55}^{-6}_{1,2,5}$,$\underline{\mathbf{0.52}}^{6}_{3,4}$
3	$\mathbf{0.56}^{8}_{1,2,5}$,$\mathbf{0.20}^{6}_{3,4}$	$\underline{0.58}^{8}_{1,2,5}$,$\mathbf{0.41}^{6}_{3,4}$	$\underline{0.56}^{8}_{2,5}$,$\underline{0.52}^{6}_{3,4}$
4	$\mathbf{0.57}^{10}_{1,2,5}$,$\mathbf{0.20}^{6}_{3,4}$	$\underline{0.58}^{8}_{2,5}$,$\mathbf{0.41}^{6}_{3,4}$	$\underline{0.56}^{8}_{2,5}$,$0.52^{6}_{3,4}$

registers:

$$
\begin{array}{l}
\alpha^{(2,1)}_{[n-1]} = 0.857663370271785, \quad \beta^{(2,1)}_{[n-1]} = 0.519611900224726, \\
\alpha^{(3,2)}_{[n-1]} = 0.770413480757674, \quad \beta^{(3,2)}_{[n-1]} = 0.466751905900312, \\
\alpha^{(4,3)}_{[n-1]} = 0.841153332326449, \quad \beta^{(4,3)}_{[n-1]} = 0.509609360199215 \\
\hline
\alpha^{(2,1)}_{[n-2]} = 0.142336629728215, \\
\alpha^{(3,1)}_{[n-2]} = 0.229586519242326, \quad \beta^{(3,1)}_{[n-2]} = 0.129608154625262, \\
\alpha^{(4,1)}_{[n-2]} = 0.158846667673551, \quad \beta^{(4,1)}_{[n-2]} = 0.096236614148583 \\
\hline
c = [0,\, 0.377275270496511,\, 0.657431495630257\,,\, 1]^T .
\end{array}
\tag{6.2}
$$

The optimal SSP GL method with stage order three, two stages, and three steps, GLp3q3s2k3 (6.3), has $\mathcal{C} = 1.10$ ($\widehat{\mathcal{C}} = 0.55$) and requires eight memory registers:

$$
\begin{array}{l}
\alpha^{(2,1)}_{[n-1]} = 0.803084592008657, \quad \beta^{(2,1)}_{[n-1]} = 0.729588628543267, \\
\alpha^{(3,2)}_{[n-1]} = 0.846696784194569, \quad \beta^{(3,2)}_{[n-1]} = 0.769209559888867 \\
\hline
\alpha^{(2,1)}_{[n-3]} = 0.196915407991343, \quad \beta^{(2,1)}_{[n-3]} = 0.140265790357552, \\
\alpha^{(3,1)}_{[n-3]} = 0.153303215805431, \quad \beta^{(3,1)}_{[n-3]} = 0.134349217930499 \\
\hline
c = [0,\, 0.476023602918134\,,\, 1]^T .
\end{array}
\tag{6.3}
$$

6.3. Fourth-order methods. The proposed fourth-order SSP GL methods with $q = 1, 2, 3, 4$ are summarized in Tables 6–9. There are no classical SSP RK methods with $s \leq 4$ and positive β values [39]. Ruuth [37] studied fourth-order explicit SSP RK methods that implicitly have stage order one. The optimal method

TABLE 5

The scaled CFL coefficient, $\widehat{\mathcal{C}}$, for the optimal (bold face) and suboptimal (light face) SSP GL methods with $p = 3$, $q = 3$. The results for methods that are not refined to full double precision are underlined. The superscript represents the number of memory registers required by each method and the subscript the type of method.

$k\backslash s$	2	3	4
1	-	-	-
2	$\mathbf{0.17}^{6}_{1,2,5}$	$\mathbf{0.48}^{6}_{1,2,5}$	$0.52^{6}_{1,2,5}$
3	$\mathbf{0.55}^{8}_{1,2,5}$	0.58^{18}_{1},$\mathbf{0.55}^{8}_{2,5}$	$\underline{0.47}^{24}_{1}$,$0.47^{18}_{2}$, $\underline{0.52}^{8}_{5}$
4	$\mathbf{0.58}^{14}_{1}$,$\mathbf{0.57}^{10}_{2,5}$	0.57^{24}_{1},$\mathbf{0.55}^{8}_{2,5}$	$\underline{0.46}^{19}_{2}$, $\underline{0.52}^{8}_{5}$

TABLE 6

The scaled CFL coefficient, $\widehat{\mathcal{C}}$, for the optimal (bold face) and suboptimal (light face) SSP GL methods with $p = 4$, $q = 1$. The results for methods that are not refined to full double precision are underlined. The superscript represents the number of memory registers required by each method and the subscript the type of method.

$k\backslash s$	2	3	4
1	-	-	-
2	-	$\mathbf{0.29}^{6}_{1,2,5}$,$\mathbf{0.11}^{6}_{3,4}$	$0.40^{8}_{1,2,5}$,$0.28^{5}_{3,4}$
3	$\mathbf{0.25}^{8}_{1,2,5}$	$\underline{0.39}^{7}_{1,2,5}$,$\mathbf{0.11}^{6}_{3,4}$	0.38^{9}_{1},$0.46^{7}_{2,5}$,$0.28^{5}_{3,4}$
4	$\mathbf{0.34}^{10}_{1,2,5}$	$\underline{0.46}^{9}_{1,2,5}$,$0.08^{7}_{3,4}$	0.13^{22}_{2},0.27^{7}_{3},0.28^{5}_{4},0.48^{9}_{5}

with five stages has $\widehat{\mathcal{C}} = 0.30$. The most efficient GL method, although not optimal, has a $\widehat{\mathcal{C}} = 0.46$.

Fourth-order SSP LM schemes have at least five steps. The five-step LM scheme has $\widehat{\mathcal{C}} = 0.02$, for six steps the $\widehat{\mathcal{C}} = 0.16$. The proposed SSP GL methods attain $\widehat{\mathcal{C}} = 0.39$ (for GLp4q4s3k4, see Table 9). The optimal LM methods need nine steps to achieve this efficiency. More efficient SSP GL with lower stage orders summarized in Tables 6–8 are possible.

We next present two methods. The optimal SSP GL method with stage order three, three stages, and three steps, GLp4q3s3k3 (6.4), has $\mathcal{C} = 1.07$ ($\widehat{\mathcal{C}} = 0.36$) and requires eight memory registers:

$$
\begin{array}{ll}
\alpha^{(2,1)}_{[n-1]} = 0.79779687008967, & \beta^{(2,1)}_{[n-1]} = 0.742235840146894, \\
\alpha^{(3,2)}_{[n-1]} = 0.685074051305928, & \beta^{(3,2)}_{[n-1]} = 0.637363385465199, \\
\alpha^{(4,1)}_{[n-1]} = 0.39703332125451, & \beta^{(4,1)}_{[n-1]} = 0.369382698548981, \\
\alpha^{(4,3)}_{[n-1]} = 0.409097066488626, & \beta^{(4,3)}_{[n-1]} = 0.380606287428385 \\
\hline
\alpha^{(3,1)}_{[n-2]} = 0.267934431946272, & \beta^{(3,1)}_{[n-2]} = 0.249274653304665, \\
\alpha^{(4,1)}_{[n-2]} = 0.149202105282063, & \beta^{(4,1)}_{[n-2]} = 0.138811211371724 \\
\hline
\alpha^{(2,1)}_{[n-3]} = 0.20220312991033, & \beta^{(2,1)}_{[n-3]} = 0.144131507391754, \\
\alpha^{(3,1)}_{[n-3]} = 0.0469915167478, & \\
\alpha^{(4,1)}_{[n-3]} = 0.044667506974801 &
\end{array}
\tag{6.4}
$$

$$
c = [0,\, 0.481961087717987,\, 0.854899608262766\,,\, 1]^T .
$$

As in the previous cases, the past information is evaluated only at previous steps, and no previous stages are involved in the computation of the current step; i.e.,

TABLE 7

The scaled CFL coefficient, $\widehat{\mathcal{C}}$, for the optimal (bold face) and suboptimal (light face) SSP GL methods with $p = 4$, $q = 2$. The results for methods that are not refined to full double precision are underlined. The superscript represents the number of memory registers required by each method and the subscript the type of method.

$k\backslash s$	2	3	4
1	-	-	
2	-	$\mathbf{0.29}^{6}_{1,2,5}$,$\mathbf{0.11}^{7}_{3,4}$	$\underline{0.39}^{8}_{1,2,5}$,$\mathbf{0.28}^{\mathbf{5}}_{\mathbf{3,4}}$
3	$\mathbf{0.25}^{8}_{1,2,5}$	$\mathbf{0.39}^{7}_{1,2,5}$,$0.07^{6}_{3,4}$	$\underline{0.16}^{17}_{1}$, $\underline{0.46}^{8}_{2}$,$\underline{0.27}^{8}_{3}$, 0.28^{10}_{4},$\underline{0.45}^{10}_{5}$
4	$\mathbf{0.34}^{10}_{1,2,5}$	$\underline{0.45}^{11}_{1}$,$0.46^{9}_{2}$,$0.08^{7}_{3,4}$,$\underline{0.45}^{9}_{5}$	$\underline{0.24}^{9}_{3}$,0.28^{11}_{4},$\underline{0.48}^{12}_{5}$

TABLE 8

The scaled CFL coefficient, $\widehat{\mathcal{C}}$, for the optimal (bold face) and suboptimal (light face) SSP GL methods with $p = 4$, $q = 3$. The results for methods that are not refined to full double precision are underlined. The superscript represents the number of memory registers required by each method and the subscript the type of method.

$k\backslash s$	2	3	4
1	-	-	-
2	-	$\mathbf{0.22}^{6}_{1,2,5}$	$\underline{0.35}^{10}_{1}$,$\underline{0.37}^{8}_{2}$,$\underline{0.38}^{6}_{5}$
3	$0.22^{8}_{1,2,5}$	$\mathbf{0.36}^{8}_{1,2,5}$	$\underline{0.45}^{8}_{5}$
4	$0.32^{9}_{1,2,5}$	$\underline{0.43}^{9}_{1,2,5}$	0.44^{12}_{5}

$\alpha^{i,j=2\ldots s+1}_{[n-\ell]} = \beta^{i,j=2\ldots s+1}_{[n-\ell]} = 0$, $\ell \geq 2$. This is a desirable outcome because the storage requirements become less; however, there are several instances in which previous stages are also required (e.g., GLp4q4s2k4, type 1 [17]), and hence this aspect cannot be generalized.

The optimal method with stage order four, three stages, and three steps, GLp4q4s3k3 (6.5), has $\mathcal{C} = 0.88$ ($\widehat{\mathcal{C}} = 0.29$) and requires seven memory registers:

$$
\begin{array}{l}
\alpha^{(2,1)}_{[n-1]} = 0.501452936754328, \quad \beta^{(2,1)}_{[n-1]} = 0.570650194053946, \\
\alpha^{(3,2)}_{[n-1]} = 0.571621756632096, \quad \beta^{(3,2)}_{[n-1]} = 0.65050185658275, \\
\alpha^{(4,1)}_{[n-1]} = 0.104408345813576, \quad \beta^{(4,1)}_{[n-1]} = 0.118816021270125, \\
\alpha^{(4,3)}_{[n-1]} = 0.555337610608053, \quad \beta^{(4,3)}_{[n-1]} = 0.631970603881811 \\
\hline
\alpha^{(2,1)}_{[n-2]} = 0.461766417377124, \quad \beta^{(2,1)}_{[n-2]} = 0.260645867579256, \\
\alpha^{(3,1)}_{[n-2]} = 0.365441633624919, \quad \beta^{(3,1)}_{[n-2]} = 0.31755158184828, \\
\alpha^{(4,1)}_{[n-2]} = 0.267081022184514, \quad \beta^{(4,1)}_{[n-2]} = 0.303936473329277 \\
\hline
\alpha^{(2,1)}_{[n-3]} = 0.036780645868547, \\
\alpha^{(3,1)}_{[n-3]} = 0.062936609742985, \\
\alpha^{(4,1)}_{[n-3]} = 0.073173021393856 \\
\hline
c = [0,\, 0.295968352518983,\, 0.645920534894549\,,\, 1]^T\,.
\end{array}
\tag{6.5}
$$

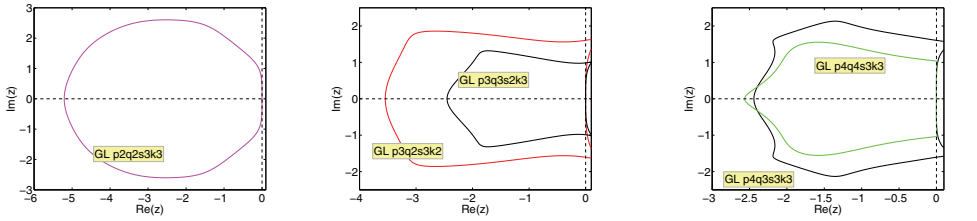
These two methods are used in our numerical experiments.

7. Numerical investigation. In this section we investigate numerically the linear stability, monotonicity, and order of several SSP GL methods presented in more detail in the previous sections. We begin with the linear stability analysis.

TABLE 9

The scaled CFL coefficient, $\widehat{\mathcal{C}}$, for the optimal (bold face) and suboptimal (light face) SSP GL methods with $p = 4$, $q = 4$. The results for methods that are not refined to full double precision are underlined. The superscript represents the number of memory registers required by each method and the subscript the type of method.

$k\backslash s$	2	3	4
1	-	-	-
2	-	-	$\underline{0.14}_1^{14}$
3	-	$\underline{0.33}_1^{18}$,0.29_2^{10},$\mathbf{0.29}_5^{7}$	$\underline{0.22}_1^{22}$, $\underline{0.13}_2^{16}$,$\underline{0.26}_5^{11}$
4	0.32_1^{13},$0.27_{2,5}^{9}$	$\underline{0.39}_1^{20}$,$\underline{0.39}_2^{9}$, $\underline{\mathbf{0.39}}_5^{9}$	$\underline{0.01}_1^{30}$, $\underline{0.10}_2^{18}$,$\underline{0.31}_5^{12}$



(a) GLp2q2s3k3 (b) GLp3q2s3k2, GLp3q3s2k3 (c) GLp4q3s3k3, GLp4q4s3k3

FIG. 7.1. *Linear stability regions for the selected SSP GL methods:* (a) *GLp2q2s3k3 (magenta),* (b) *GLp3q2s3k2 (blue), GL p3q3s2k3 (red), and* (c) *GLp4q3s3k3 (green), GLp4q4s3k3 (black). The stability region is represented by the bounded set enclosed by each curve.*

7.1. The linear stability of the selected methods. In this section we explore the stability regions for the selected methods presented in section 6 by using the procedure described in section 2.2.

In Figure 7.1 we show the linear stability regions for the following methods: GLp2q2s3k3, GLp3q2s3k2, GLp3q3s2k3, GLp4q3s3k3, and GLp4q4s3k3.

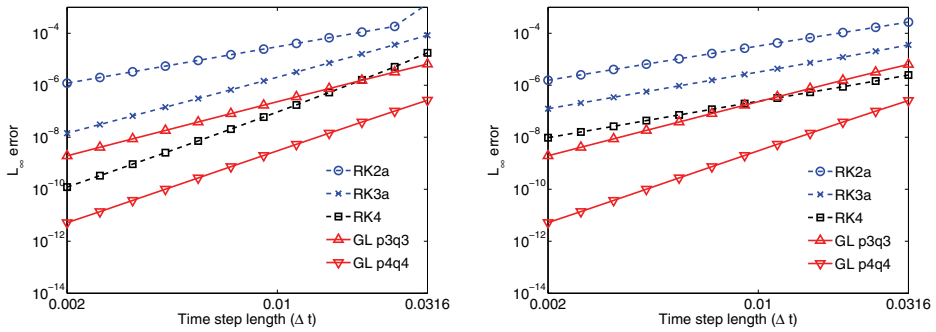
We remark that the stability regions contain a segment of the imaginary axis, which is a desirable property when solving PDEs via the method of lines with certain spatial discretizations [30]. A stability region with similar properties can be found for the other methods not shown here.

7.2. Validation for order preservation. We illustrate the boundary/source order reduction phenomenon and consider a classical initial value test problem with a nonlinear source described in [42]:

$$\frac{\partial y(t,x)}{\partial t} = -\frac{\partial y(t,x)}{\partial x} + b(t,x)\,, \quad \begin{array}{ll} 0 \le x \le 1, & y(t,0) = b(t,0), \\ 0 \le t \le 1, & y(0,x) = y_0(x), \end{array} \tag{7.1}$$

with the initial condition $y_0(x) = 1 + x$ and (left) boundary and source term defined by $b(t,x) = (t-x)/(1+t)^2$. The exact solution given by $y(t,x) = (1+x)/(1+t)$ is linear in space, allowing us to use first-order upwind space discretization without introducing discretization errors. For the time integration the SSP RK methods of orders 2, 3, and the classical RK method ($p = 4$) are employed. All explicit RK methods have the stage order equal to one. Sanz-Serna, Verwer, and Hundsdorfer [42] show that explicit RK methods with $p \ge 3$ suffer from order reduction on problems with nonhomogeneous boundary conditions or nonzero source terms such as (7.1).

For problem (7.1) we distinguish two cases, one that illustrates the order reduction phenomenon, and, for validation purposes, one that does not. Specifically, if the spatial and temporal grids are refined simultaneously, one notices that low stage-order



(a) Classical order retention (b) Order reduction

FIG. 7.2. *Order analysis for the classical SSP RK methods of orders two and three (blue), the fourth-order RK method, and SSP GL methods GLp3q3* (6.3) *and GLp4q4* (6.5) *(red). The GL methods preserve their corresponding orders, whereas the classical RK methods suffer from order reduction.*

methods suffer from order reduction ($p \leq 2$) [42]. If the space grid is maintained fixed—the ODE problem is fixed—then the (classical) order of consistency is preserved.

Figure 7.2 shows the discretization error versus the time step without order reduction (Figure 7.2(a)) and with order reduction (Figure 7.2(b)). In the former case, the order of the RK methods is preserved, whereas in the later case the order clearly drops to two for all RK methods. A special boundary/source treatment can be used to alleviate this problem, but with great effort and limited success [12, 41, 42]. This discussion also applies to implicit RK methods with low stage orders such as DIRK [34].

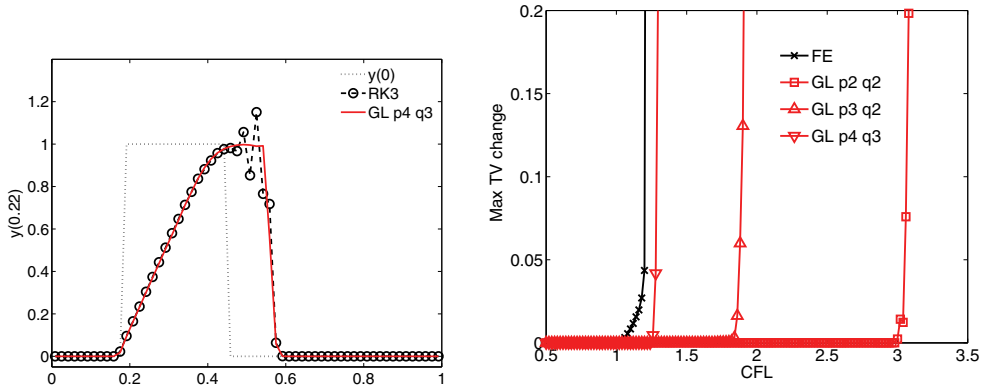
We next consider two of the proposed SSP GL methods of orders three and four, GLp3q3s2k3 (6.3) and GLp4q4s3k3 (6.5), to solve problem (7.1). They are initialized with SSP methods of corresponding orders. We remark that the starting procedures typically use low stage order methods, and therefore, error can be accumulated in the first k steps. This effect can be alleviated by using a smaller time step for method initialization. In Figure 7.2 we show that GL methods retain their corresponding orders of consistency. Moreover, a visual inspection of Figure 7.2(a) reveals that the truncation error coefficient of the GL methods appears to be smaller than that of the classical methods (with stage order one) under consideration.

7.3. Monotonicity validation. We now investigate the monotonicity preservation for a nonlinear PDE. The inviscid Burgers equation is

$$\frac{\partial y(t,x)}{\partial t} + \frac{\partial}{\partial x}\left(\frac{1}{2}y(t,x)^2\right) = 0\,, \quad 0 \leq x \leq 1\,,\ 0 \leq t \leq t_{\text{Final}}\,. \tag{7.2}$$

The spatial discretization uses an m-point equidistant grid, $\Delta x = 1/m$, $x_i = (i - 1/2)\Delta x$, $i = 1, \ldots, m$, with periodic boundary conditions. A third-order upwind-biased flux limited scheme based on the work of Osher and Chakravarthy [14, 35, 36] is used to obtain the spatial discretization operator. The algorithms can be found in [15, 32]. This method is SSP with forward Euler steps and, hence, with the proposed GL methods described in this work.

The initial solution for (7.2) is represented by a step function that produces a shock and a rarefaction (expansion) wave [32]. The GL methods are initialized by using the appropriate starting procedures discussed earlier. Spurious oscillations can



(a) CFL # $= 1.32 \times \max(y)$ (b) TV vs. CFL coefficient

FIG. 7.3. (a) *Solution of the Burgers equation using the third-order RK method* ($s = 3$, $\mathcal{C} = 1$) *and fourth-order GLp4q3s3k3* (6.4) ($\mathcal{C} = 1.07$). *In* (b) *we show the maximum change of the solution TV with forward Euler (FE)* $\mathcal{C} = 1$ *and GL methods of orders two* (6.1) $\mathcal{C} = 2.57$, *three* (6.2) $\mathcal{C} = 1.65$, *and four* (6.4) $\mathcal{C} = 1.07$.

occur in the solution if the time step used by the method violates the CFL condition. In this case the CFL condition with forward Euler steps is one, and therefore, the SSP condition of the GL methods is satisfied if the CFL coefficient of the method $\mathcal{C}$ is smaller than the CFL number of the problem: $\mathcal{C} \leq$ problem CFL number $= \max(y)\Delta t/\Delta x$.

In Figure 7.3(a) we show the solution of the Burgers equation integrated with RK3 ($s = 3$, $\mathcal{C} = 1$) and GLp4q3s3k3 (6.4) ($\mathcal{C} = 1.07$) at $t = 0.23$. The problem CFL number is 1.32. Spurious oscillations are generated by the RK scheme. The GL scheme has a larger $\mathcal{C}$ than does RK3 and remains stable.

Next we investigate the SSP property when using the total variation (TV) seminorm:

$$\mathrm{TV}(y(t,x)) = \sum |y(t,x_i) - y(t,x_{i-1})|\,, \quad i = 1,\ldots,m\,.$$

The preservation of the strong stability requires that the TV norm be nonincreasing from one step to the next. It follows that the maximum total variation change is

$$\max\left(\mathrm{TV}(y(t_{[i]},x)) - \mathrm{TV}(y(t_{[i-1]},x))\right) \leq 0\,, \quad i = 1,\ldots,n\,, \quad n = t_{\mathrm{Final}}/\Delta t\,.$$

In Figure 7.3(b) we show the maximum TV change for the solution of (7.2) by using the forward Euler scheme and GL methods of orders two (6.1), three (6.2), and four (6.4) in time. For this example, the upwind method in space was used to avoid the limiter artifacts. The solution is evolved to time 0.5 by using increasing $\mathcal{C}$ values. The theoretical SSP bounds are clearly illustrated in these numerical experiments.

7.4. High dimensional PDE example. We present a two-dimensional problem that uses the proposed GL methods and other LM and RK SSP and non-SSP time integration schemes. The problem solved in this section is a variation of Molenkamp [51]: a nonautonomous linear advection equation in a velocity field that describes a rigid body rotation:

(7.3)
$$\frac{\partial c}{\partial t} + u\frac{\partial c}{\partial x_1} + v\frac{\partial c}{\partial x_2} = S(t,x_1,x_2)\,, \quad 0 \leq t \leq T\,, \quad c(0,x_1,x_2) = \sin(h_0(x_1,x_2))\,,$$

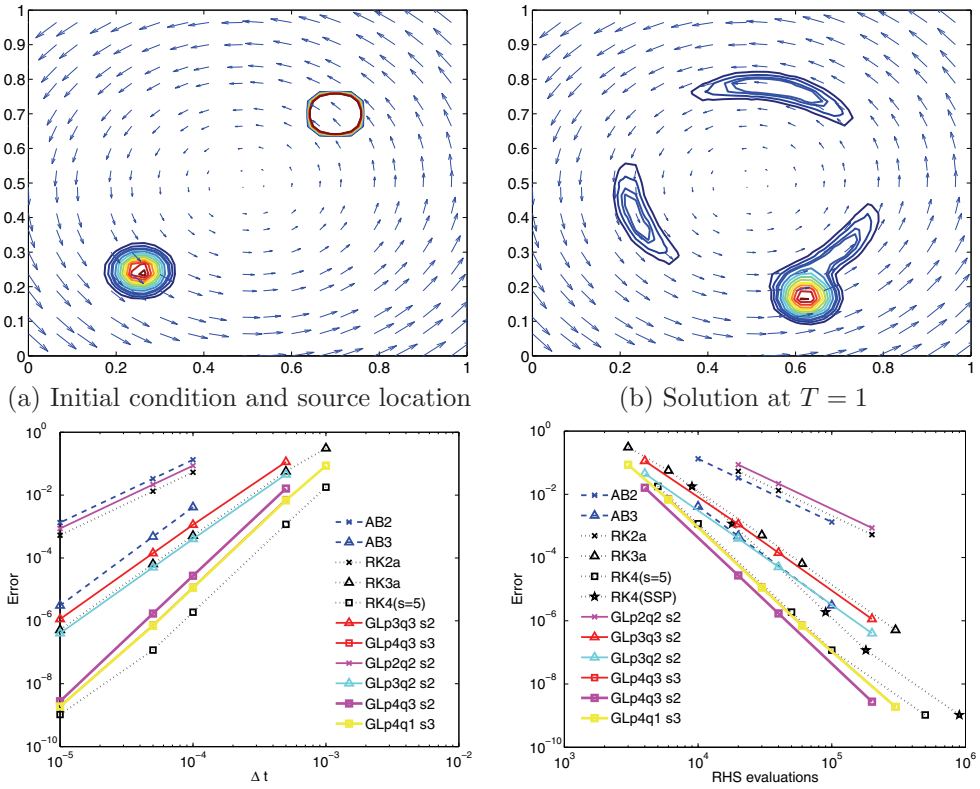


FIG. 7.4. *Solution of the two-dimensional problem* (7.3). *The initial condition (lower-left) and source location (upper-right) are shown in* (a). *The solution at* $t = T = 1$ *is illustrated in* (b). *The* L_2 *convergence of selected time-stepping methods is shown in* (c) *and the computational cost represented by the number of right-hand side (RHS) evaluations associated with each method is assessed in* (d).

$$u = -2\pi x_2\,, \quad v = 2\pi x_1\,, \quad S(t, x_1, x_2) = 10\sin(10\pi t)^4 \sin(\pi h_1(x_1)) \sin(\pi h_2(x_2))\,,$$

where $0 \le x_1, x_2 \le 1$ represent the space dimensions, and $c = c(t, x_1, x_2)$. Functions h_i, $i = 1, 2, 3$ are spatial restriction operators, which place the initial and source (S) profiles in the lower left and upper right quadrants, respectively (see Figure 7.4(a)). The solution consists in a rotating sinusoidal profile around the origin (0,0) and a time-dependent pulsating source. A high-order continuous Galerkin method [19] is used for the discretization of the spatial derivatives in (7.3) on a 10×10 element quadrilateral mesh, each with five nodes.

The time integration is performed by using the following explicit methods: $p = 2, 3$ Adams–Bashforth LM methods, SSP RK 2a ($s = 2$), RK 3a ($s = 3$), and non-SSP RK4 ($s = 5$) [13], and several SSP GL methods introduced in this work and explicitly defined in [17]. The LM and RK schemes were chosen based on their pervasiveness in numerical simulation codes [1, 2, 33, 46].

In Table 10 we show the relative efficiency of several time-stepping methods. The method indicated as RK4(SSP) is an SSP implementation of RK4 ($s = 5$), which requires four additional function evaluations due to the fact that the internal Euler

TABLE 10

Relative efficiency of selected time-stepping methods. The values represent the additional time or compute effort (in percentage) for each method to obtain a given absolute solution accuracy at $t = T = 1$. For instance, if the desired accuracy is 10^{-6}, then non-SSP RK4 is expected to take 28.16% more time to run than GLp4q3s2.

Error	AB3	RK3a	RK4(s=5)	GLp3q3 s2	GLp4q3 s3	RK4(SSP)	GLp3q2 s2	GLp4q3 s2	GLp4q1 s3
10^{-2}	36.62	-	-	-	0.00	-	-	-	-
10^{-3}	95.54	-	29.10	158.48	21.76	132.37	81.78	0.00	20.36
10^{-4}	128.48	258.46	28.51	211.93	21.31	131.32	120.65	0.00	20.08
10^{-5}	167.23	333.19	28.19	276.93	21.10	130.74	166.65	0.00	20.04
10^{-6}	213.28	424.67	28.16	356.54	21.03	130.69	222.97	0.00	19.98
10^{-7}	266.83	534.74	29.89	452.31	25.93	133.81	290.72	0.00	25.09
10^{-8}	329.54	667.89	59.90	568.16	31.92	187.81	372.69	0.00	31.47

steps are taken in the negative time direction. We note the superiority of high-order low stage-number methods, in particular GLp4q3 s2. The GL methods analyzed here have high stage orders as well, and, therefore, are expected to perform better in the presence of stiff boundary conditions. The SSP implementation of RK4 performs poorly when compared with the SSP methods, and is expected to be even less robust with SSP-compliant spatial discretizations in the SSP metric, as noted in the previous sections.

8. Discussion. This paper brings an important contribution to the area of SSP numerical methods. We design schemes with the SSP property based on a new class of methods that represent a generalization of both RK and LM schemes. Several schemes of practical importance are presented in this paper. Additional methods are presented in a technical report [17].

The importance of the SSP property has been discussed in section 2.3 and illustrated numerically in section 7.3. Methods with a larger CFL coefficient ($\mathcal{C}$) are more efficient because they allow larger time steps. We employ a global search procedure to identify the best possible SSP GL methods.

The proposed GL methods can attain high stage orders, a property that alleviates the order reduction phenomenon encountered in the classical explicit RK schemes due to nonhomogeneous boundary/source terms.

The numerical scheme storage requirements are also important, especially in large-scale applications. We explore methods that carry a decreasing amount of information from one step to the next in order to reduce the memory requirements. We remark that in several cases the most robust methods also produce low storage schemes.

We have explored schemes with positive values for the β coefficients. It is also possible to consider negative ones; however, in this case the adjoint (downwind) discretization of f is required, and this is not always easy to obtain. We do not address the issue of changing the time step, which currently requires restarting the problem. Better procedures need to be identified for an efficient implementation.

Acknowledgments. We are grateful to Todd Munson and Mihai Anitescu for their helpful suggestions in setting up GAMS and BARON. We would also like to thank Paul Fischer for providing us with the 2D continuous Galerkin code.

REFERENCES

[1] S. BALAY, K. BUSCHELMAN, V. EIJKHOUT, W. D. GROPP, D. KAUSHIK, M. G. KNEPLEY, L. C. MCINNES, B. F. SMITH, AND H. ZHANG, *PETSc Users Manual*, Technical Report ANL-95/11 - Revision 3.0.0, Argonne National Laboratory, Argonne, IL, 2008.

[2] S. Balay, K. Buschelman, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang, *PETSc Web page*, 2009, available online at http://www.mcs.anl.gov/petsc.
[3] A. Brooke, D. Kendrick, A. Meeraus, and R. Rosenthal, *GAMS - A User's Guide*, The Scientific Press, San Francisco, 1988.
[4] K. Burrage and J. C. Butcher, *Nonlinear stability of a general class of differential equation methods*, BIT, 20 (1980), pp. 185–203.
[5] J. C. Butcher, *A modified multistep method for the numerical integration of ordinary differential equations*, J. Assoc. Comput. Mach., 12 (1965), pp. 124–135.
[6] J. C. Butcher, *On the convergence of numerical solutions to ordinary differential equations*, Math. Comp., 20 (1966), pp. 1–10.
[7] J. C. Butcher, *An algebraic theory of integration methods*, Math. Comp., 26 (1972), pp. 79–106.
[8] J. C. Butcher, *Linear and nonlinear stability for general linear methods*, BIT, 27 (1987), pp. 182–189.
[9] J. C. Butcher, *General linear methods for stiff differential equations*, BIT, 41 (2001), pp. 240–264.
[10] J. C. Butcher, *General linear methods*, Acta Numer., 15 (2006), pp. 157–256.
[11] J. C. Butcher, *Numerical Methods for Ordinary Differential Equations*, 2nd ed., John Wiley & Sons, Chichester, UK, 2008.
[12] M. Carpenter, D. Gottlieb, S. Abarbanel, and W.-S. Don, *The theoretical accuracy of Runge-Kutta time discretizations for the initial boundary value problem: A study of the boundary error*, SIAM J. Sci. Comput., 16 (1995), pp. 1241–1252.
[13] M. Carpenter and C. Kennedy, *Fourth-Order 2N-Storage Runge-Kutta Schemes*, Technical report 109112, NASA, Washington, DC, 1994.
[14] S. R. Chakravarthy and S. Osher, *Numerical experiments with the Osher upwind scheme for the Euler equations*, AIAA J., 21 (1983), pp. 1241–1248.
[15] S. Chakravarthy and S. Osher, *Computing with high-resolution upwind schemes for hyperbolic equations*, in Large-Scale Computations in Fluid Mechanics, Lectures in Appl. Math., 22 (1985), pp. 57–86.
[16] E. Constantinescu, *On the order of general linear methods*, Appl. Math. Lett., 22 (2009), pp. 1425–1428.
[17] E. Constantinescu and A. Sandu, *Optimal Explicit Strong-Stability-Preserving General Linear Methods: Complete Results*, Technical report ANL/MCS-TM-304, Argonne National Laboratory, Mathematics and Computer Science Division Technical Memorandum, Argonne, Argonne, IL, available online at http://www.mcs.anl.gov/uploads/cels/papers/TM304A.pdf, Jan. 2009.
[18] G. J. Cooper, *The order of convergence of general linear methods for ordinary differential equations*, SIAM J. Numer. Anal., 15 (1978), pp. 643–661.
[19] M. O. Deville, P. F. Fischer, and E. H. Mund, *High-Order Methods for Incompressible Fluid Flow*, Cambridge University Press, Cambridge, UK, 2002.
[20] C. W. Gear, *Hybrid methods for initial value problems in ordinary differential equations*, SIAM J. Numer. Anal., 2 (1965), pp. 69–86.
[21] S. Gottlieb, *On high order strong stability preserving Runge–Kutta and multi step time discretizations*, J. Sci. Comput., 25 (2005), pp. 105–128.
[22] S. Gottlieb, C.-W. Shu, and E. Tadmor, *Strong stability-preserving high-order time discretization methods*, SIAM Rev., 43 (2001), pp. 89–112.
[23] W. B. Gragg and H. J. Stetter, *Generalized multistep predictor-corrector methods*, J. Assoc. Comput. Mach., 11 (1964), pp. 188–209.
[24] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations* I*: Nonstiff Problems*, Springer-Verlag, Berlin, 1993.
[25] E. Hairer and G. Wanner, *Multistep-multistage-multiderivative methods for ordinary differential equations*, Computing, 11 (1973), pp. 287–303.
[26] E. Hairer and G. Wanner, *On the Butcher group and general multi-value methods*, Computing, 13 (1974), pp. 1–15.
[27] I. Higueras, *On strong stability preserving time discretization methods*, J. Sci. Comput., 21 (2004), pp. 193–223.
[28] I. Higueras, *Representations of Runge-Kutta methods and strong stability preserving methods*, SIAM J. Numer. Anal., 43 (2005), pp. 924–948.
[29] C. Huang, *Strong stability preserving hybrid methods*, Appl. Numer. Math., 59 (2009), pp. 891–904.

[30] W. Hundsdorfer and J. Verwer, *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*, Springer Ser. Comput. Math. 33, Springer-Verlag, Berlin, 2003.
[31] D. I. Ketcheson, *Computation of optimal monotonicity preserving general linear methods*, Math. Comp., 78 (2009), pp. 1497–1513.
[32] C. B. Laney, *Computational Gasdynamics*, Cambridge University Press, Cambridge, UK, 1998.
[33] R. LeVeque, *CLAWPACK Web page*, 2009, available online at http://www.amath.washington.edu/claw.
[34] C. B. Macdonald, S. Gottlieb, and S. J. Ruuth, *A numerical study of diagonally split Runge–Kutta methods for PDEs with discontinuities*, J. Sci. Comput., 36 (2008), pp. 89–112.
[35] S. Osher and S. Chakravarthy, *High resolution schemes and the entropy condition*, SIAM J. Numer. Anal., 21 (1984), pp. 955–984.
[36] S. Osher and S. Chakravarthy, *Very high order accurate TVD schemes*, in Oscillation Theory, Computation, and Methods of Compensated Compactness, IMA Vol. Math. Appl. 2, Springer, New York, 1986, pp. 229–274.
[37] S. J. Ruuth, *Global optimization of explicit strong-stability-preserving Runge-Kutta methods*, Math. Comp., 75 (2006), pp. 183–207.
[38] S. J. Ruuth and W. Hundsdorfer, *High-order linear multistep methods with general monotonicity and boundedness properties*, J. Comput. Phys., 209 (2005), pp. 226–248.
[39] S. J. Ruuth and R. Spiteri, *Two barriers on strong-stability-preserving time discretization methods*, J. Sci. Comput., 17 (2002), pp. 211–220.
[40] N. Sahinidis and M. Tawarmalani, *BARON* 7.2.5*: Global optimization of mixed-integer nonlinear programs*, User's Manual, 2005, available online at http://archimedes.cheme.cmu.edu/baron/baron.html.
[41] J. M. Sanz-Serna and J. G. Verwer, *Stability and convergence at the PDE/stiff ODE interface*, Appl. Numer. Math., 5 (1989), pp. 117–132.
[42] J. M. Sanz-Serna, J. G. Verwer, and W. H. Hundsdorfer, *Convergence and order reduction of Runge-Kutta schemes applied to evolutionary problems in partial differential equations*, Numer. Math., 50 (1987), pp. 405–418.
[43] C.-W. Shu, *Total-variation-diminishing time discretizations*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 1073–1084.
[44] C.-W. Shu, *A survey of strong stability preserving high order time discretizations*, in Collected Lectures on the Preservation Of Stability Under Discretization, D. Estep and T. Tavener, eds., SIAM, Philadelphia, 2002, pp. 51–65.
[45] C.-W. Shu and S. Osher, *Efficient implementation of essentially nonoscillatory shock-capturing schemes*, J. Comput. Phys., 77 (1988), pp. 439–471.
[46] W. Skamarock, J. Klemp, J. Dudhia, D. Gill, D. Barker, M. Duda, X.-Y. Huang, W. Wang, and J. Powers, *A Description of the Advanced Research WRF Version* 3, Technical report, Tech Notes-475+ STR, NCAR, Boulder, CO, 2008.
[47] R. Skeel, *Analysis of fixed-stepsize methods*, SIAM J. Numer. Anal., 13 (1976), pp. 664–685.
[48] M. N. Spijker, *Stepsize conditions for general monotonicity in numerical initial value problems*, SIAM J. Numer. Anal., 45 (2007), pp. 1226–1245.
[49] R. J. Spiteri and S. J. Ruuth, *A new class of optimal high-order strong-stability-preserving time discretization methods*, SIAM J. Numer. Anal., 40 (2002), pp. 469–491.
[50] R. J. Spiteri and S. J. Ruuth, *Non-linear evolution using optimal fourth-order strong-stability-preserving Runge-Kutta methods*, Math. Comput. Simulation, 62 (2003), pp. 125–135.
[51] C. Vreugdenhil and B. Koren, *Numerical Methods for Advection–Diffusion Problems*, vol. 45, Vieweg, Braunschweig, Germany, 1993.