Journal of Scientific Computing, Vol. 18, No. 1, February 2003 (© 2003)

Strong Stability Preserving Properties of Runge–Kutta Time Discretization Methods for Linear Constant Coefficient Operators

Sigal Gottlieb[1] and Lee-Ad J. Gottlieb[1]

Received August 21, 2001; accepted (in revised form) January 11, 2002

Strong stability preserving (SSP) high order Runge–Kutta time discretizations were developed for use with semi-discrete method of lines approximations of hyperbolic partial differential equations, and have proven useful in many other applications. These high order time discretization methods preserve the strong stability properties of first order explicit Euler time stepping. In this paper we analyze the SSP properties of Runge Kutta methods for the ordinary differential equation $u_t = Lu$ where L is a linear operator. We present optimal SSP Runge–Kutta methods as well as a bound on the optimal timestep restriction. Furthermore, we extend the class of SSP Runge–Kutta methods for linear operators to include the case of time dependent boundary conditions, or a time dependent forcing term.

KEY WORDS: Strong stability preserving; Runge–Kutta methods; high order accuracy; time discretization.

1. INTRODUCTION

1.1. The History of SSP Methods

In solving time dependent hyperbolic Partial Differential Equations (PDEs) it is common practice to first discretize the spatial variables to obtain a semi-discrete method of lines scheme. This is then an Ordinary Differential Equation (ODE) system in the time variable which can be discretized by an

[1] Department of Mathematics, University of Massachusetts at Dartmouth, Dartmouth, Massachusetts 02747 and Division of Applied Mathematics, Brown University, Providence, Rhode Island 02912. E-mail: sgottlieb@umassd.edu

0885-7474/03/0200-0083/0

ODE solver. The simplest such ODE solver is the forward-Euler method and it is used widely for analysis of the nonlinear stability properties of the spatial discretization. The nonlinear stability properties are essential, since hyperbolic problems typically have discontinuous solutions and a stronger measure than linear stability is thus required. However, while forward-Euler is ideal for analysis of the stability properties of a given spatial discretization, it is only first order accurate. In practice, high order time discretizations which preserve all the stability properties of forward-Euler, are needed.

In [15, 14, 4 and 5] high order strong stability preserving (SSP) time discretization methods for the semi-discrete method of lines approximations of PDEs were developed. These methods are derived by assuming that the first order forward-Euler time discretization of the method of lines ODE is strongly stable under a certain norm, when the time step Δt is suitably restricted, and then finding higher order time discretizations that maintain strong stability for the same norm, perhaps under a different time step restriction.

SSP Runge Kutta methods were first considered for the solution of the hyperbolic equation

$$u_t + f(u)_x = 0 \tag{1.1}$$

where the spatial derivative, $f(u)_x$, is discretized by a TVD finite difference or finite element approximation denoted $-L(u)$ ([9, 12, 18, 3, 10] and [19]). In the process of discretizing, we have a spatial mesh made up of points denoted x_j and a temporal mesh of points denoted t^n. When discussing the fully discretized solution, we use the notation u_j^n to mean the approximation to the exact solution $u(x_j, t^n)$, and the corresponding vector u^n containing all the spatial information at a given time is given componentwise by $u^n = [u_j^n]$. The exact spatial discretization $L(u)$ above is irrelevant, as long as it has the property that when it is combined with the first order forward-Euler time discretization,

$$u^{n+1} = u^n + \Delta t \, L(u^n) \tag{1.2}$$

the Total Variation (TV) of the one-dimensional discrete solution u^n does not increase in time, i.e., the following, so called Total Variation Diminishing (TVD) property, holds

$$\mathrm{TV}(u^{n+1}) \leqslant \mathrm{TV}(u^n), \qquad \mathrm{TV}(u^n) := \sum_j |u_{j+1}^n - u_j^n| \tag{1.3}$$

for a sufficiently small time step Δt dictated by the CFL condition (see [1, 8]),

$$\Delta t \leqslant \Delta t_{\mathrm{FE}} \tag{1.4}$$

Here, Δt_{FE} is the largest allowable step size that will guarantee that the stability property above will hold for forward-Euler with the given PDE and spatial discretization (see [14]).

The objective of the high order SSP time discretization is to maintain the strong stability property (1.3) while achieving higher order accuracy in time, perhaps with a modified CFL restriction (measured here with a CFL coefficient, c)

$$\Delta t \leqslant c\, \Delta t_{\mathrm{FE}} \tag{1.5}$$

Numerical evidence presented in [4] demonstrated that oscillations may occur when using a linearly stable, high-order method which lacks the strong stability property, even if the same spatial discretization is TVD when combined with the first-order forward-Euler time-discretization. This illustrates that it is safer to use a SSP time discretization for solving hyperbolic problems. After all, SSP methods have the extra assurance of provable stability and in many cases do not increase the computational cost. In particular, SSP methods up to (and including) third order for ODEs with nonlinear operators L, and all SSP methods for ODEs with linear constant-coefficient operators do not require any additional stages or function evaluations [5].

In the initial development of these methods, ([15] and [14]) the relevant norm was the total variation norm: the forward-Euler time discretization of the method of lines ODE was assumed TVD, hence these methods were called TVD time discretizations. In fact, the essence of this class of high order time discretizations lies in its ability to maintain the strong stability in the same norm as the first order forward-Euler version, regardless of what this norm is, hence "strong stability preserving (SSP) time discretization" is a more suitable term which was first adopted in [5]. Additionally, since SSP methods (as we show below) are convex combinations of the first-order Euler method, any convex function satisfied by forward-Euler will be preserved by such high-order time discretizations. Thus, the SSP property is useful in a wide variety of applications. SSP Runge Kutta methods can be used whenever a method is needed which preserves any norm or convex-function property of forward-Euler. Also, although these methods were developed for use with nonlinear stability properties, they are equally useful in cases where the relevant operator is linear, and where linear norm properties are studied. In this paper we will study the properties of SSP Runge–Kutta methods for linear constant-coefficient operators.

1.2. SSP Runge–Kutta Methods

In [15], a general m stage Runge–Kutta method for

$$u_t = L(u) \tag{1.6}$$

is written in the form:

$$\begin{aligned} u^{(0)} &= u^n, \\ u^{(i)} &= \sum_{k=0}^{i-1} \left(\alpha_{i,k} u^{(k)} + \Delta t \, \beta_{i,k} L(u^{(k)})\right), \qquad \alpha_{i,k} \geqslant 0, \quad i = 1,..., m \\ u^{n+1} &= u^{(m)} \end{aligned} \tag{1.7}$$

This way of writing the method has become standard for SSP purposes (e.g., [14, 15, 4, 5]), and was shown ([15]) to be equivalent to the classical Runge–Kutta methods as written in [2]. The restriction on the coefficients $\alpha_{i,k}$ allows the SSP property to be achieved ([14]). Clearly, if all the coefficients $\beta_{i,k}$ are nonnegative ($\beta_{i,k} \geqslant 0$), and the consistency requirement

$$\sum_{k=0}^{i-1} \alpha_{i,k} = 1$$

is satisfied for any i, it follows that the intermediate stages in (1.7), $u^{(i)}$, amount to convex combinations of forward-Euler steps, with Δt replaced by $\frac{\beta_{i,k}}{\alpha_{i,k}} \Delta t$. We thus conclude

Lemma 1.1 [15]. If the forward-Euler method (1.2) is strongly stable under the CFL restriction (1.4), $\|u^n + \Delta t \, L(u^n)\| \leqslant \|u^n\|$, then the Runge–Kutta method (1.7) with $\beta_{i,k} \geqslant 0$, and $\beta_{i,k} = 0$ whenever $\alpha_{i,k} = 0$, is SSP, $\|u^{n+1}\| \leqslant \|u^n\|$, provided the following CFL restriction (1.5) is fulfilled,

$$\Delta t \leqslant c \, \Delta t_{\mathrm{FE}}, \qquad c = \min_{i,k} \frac{\alpha_{i,k}}{\beta_{i,k}} \qquad \forall \beta_{i,k} \neq 0 \tag{1.8}$$

If some of the $\beta_{i,k}$'s are negative, we need to introduce an associated operator $\tilde{L}$ corresponding to stepping *backward* in time. The requirement for $\tilde{L}$ is that it approximates the same spatial derivative(s) as L, but that the strong stability property holds $\|u^{n+1}\| \leqslant \|u^n\|$, (either with respect to the TV or another relevant norm), for the first order Euler scheme, solved backward in time, i.e.,

$$u^{n+1} = u^n - \Delta t\, \tilde{L}(u^n) \tag{1.9}$$

This can be achieved, for hyperbolic conservation laws, by solving the time-negative version of (1.1),

$$u_t = f(u)_x \tag{1.10}$$

Numerically, the only difference is the change of upwind direction. Clearly, $\tilde{L}$ can be computed with the same cost as that of computing L. We then have the following lemma.

Lemma 1.2 [15]. If the forward-Euler method combined with the spatial discretization L in (1.2) is strongly stable under the CFL restriction (1.4), $\|u^n + \Delta t\, L(u^n)\| \leqslant \|u^n\|$, and if Euler's method solved backward in time in combination with the spatial discretization $\tilde{L}$ in (1.9) is also strongly stable under the CFL restriction (1.4), $\|u^n - \Delta t\, \tilde{L}(u^n)\| \leqslant \|u^n\|$, then the Runge–Kutta method (1.7) is SSP, i.e., $\|u^{n+1}\| \leqslant \|u^n\|$, under the CFL restriction (1.5),

$$\Delta t \leqslant c\, \Delta t_{\mathrm{FE}}, \qquad c = \min_{i,k} \frac{\alpha_{i,k}}{|\beta_{i,k}|}, \qquad \forall \beta_{i,k} \neq 0 \tag{1.11}$$

provided $\beta_{i,k} = 0$ whenever $\alpha_{i,k} = 0$, and $\beta_{i,k} L$ is replaced by $\beta_{i,k} \tilde{L}$ whenever $\beta_{i,k}$ is negative.

Notice that, if for the same k, both $L(u^{(k)})$ and $\tilde{L}(u^{(k)})$ must be computed, the cost as well as storage requirement for this k is doubled. For this reason, we would like to avoid negative $\beta_{i,k}$ as much as possible. In [4] it was shown any four-stage fourth-order Runge–Kutta method for a nonlinear ODE will have at least one negative coefficient. In [17] it was shown that any Runge–Kutta method of fifth order or above for a nonlinear ODE will have at least one negative coefficient. Thus, we realize that for Runge–Kutta methods for nonlinear ODEs, negative coefficients would have to be considered. This is not, however, the case for Runge–Kutta methods for linear ODEs. In the linear constant coefficient case, we may have all nonnegative coefficients [5], and we proceed by discussing this case.

2. LINEAR CONSTANT COEFFICIENT SSP RUNGE–KUTTA METHODS OF ARBITRARY ORDER

Although SSP methods were developed for use with nonlinear stability properties, they are equally useful in cases where the relevant operator is linear, and where linear norm properties are studied. For example, SSP methods are useful where weighted L^2 higher order discretizations of spectral schemes are discussed ([7, 5, 11]). In [5] we found optimal N stage, Nth order SSP Runge–Kutta methods of arbitrary order of accuracy for *linear* ODEs suitable for solving PDEs with linear spatial discretizations. Such methods have optimal CFL number $c=1$. Raising the CFL number at the expense of adding another stage is an idea that was tried in [14]. In parallel to this work, S. Ruuth and R. Spiteri have studied this approach for nonlinear methods [16] and linear methods [17]. In this section, we consider linear SSP Runge–Kutta methods which have more stages than necessary for their order. This additional freedom allows for a higher CFL number. We present a bound on the optimal CFL number associated with an m stage, Nth order method. We then present some methods which are optimal in terms of the CFL restriction.

2.1. Useful Properties of the Linear SSP Runge–Kutta Method

In [5] we constructed a class of optimal (in the sense of CFL number) SSP Runge–Kutta methods of any order for the ODE (1.6) where L is *linear* and *time invariant*. With a linear L being realized as finite dimensional matrix we denote $L(u)=Lu$.

The method (1.7) may be rewritten as

$$u^{(i)}=\left(1+\sum_{k=0}^{i-1} A_{i,k}(\Delta t\, L)^{k+1}\right)u^{(0)}, \qquad i=1,\dots,m \tag{2.1}$$

where

$$A_{1,0}=\beta_{1,0}, \qquad A_{i,0}=\sum_{k=1}^{i-1}\alpha_{i,k}A_{k,0}+\sum_{k=0}^{i-1}\beta_{i,k}$$

$$A_{i,k}=\sum_{j=k+1}^{i-1}\alpha_{i,j}A_{j,k}+\sum_{j=k}^{i-1}\beta_{i,j}A_{j,k-1}, \qquad k=1,\dots,i-1$$

A method of this type will be SSP for a sufficiently small time step Δt, if all the coefficients $\alpha_{i,k}$ and $\beta_{i,k}$ are nonnegative. The CFL number (c in (1.8)) associated with this method can be written as $c=\frac{1}{\mu}$ where

$\mu = \max_{i,k} \frac{\beta_{i,k}}{\alpha_{i,k}}$. To facilitate the analysis of the optimal CFL number, we introduce $\mu_{i,k} = \frac{\beta_{i,k}}{\alpha_{i,k}}$. We can make this definition, since the SSP conditions above require that $\beta_{i,k} = 0$ whenever $\alpha_{i,k} = 0$. The following lemmas determine a bound on the relative size of each $A_{i,k}$, which depends on μ. These lemmas will later be used to get a bound on the optimal CFL number.

Lemma 2.1. For any method written in the form (2.1) above,

$$A_{M,0} \leqslant M\mu$$

for any $1 \leqslant M \leqslant m$, where $\mu = \max_{i,k} \frac{\beta_{i,k}}{\alpha_{i,k}}$.

Proof. Consider that

$$A_{1,0} = \beta_{1,0} = \frac{\beta_{1,0}}{\alpha_{1,0}} = \mu_{1,0} \leqslant \mu$$

Now proceed by induction: Assume $A_{j,0} \leqslant j\mu \; \forall j = 1,..., M-1$ then

$$\begin{aligned} A_{M,0} &= \sum_{j=1}^{M-1} \alpha_{M,j} A_{j,0} + \sum_{j=0}^{M-1} \beta_{M,j} \\ &\leqslant (M-1)\,\mu \sum_{j=1}^{M-1} \alpha_{M,j} + \mu \sum_{j=0}^{M-1} \alpha_{M,j} \\ &\leqslant M\mu \end{aligned}$$

□

Lemma 2.2. For any method written in the form (2.1) above,

$$A_{M,1} \leqslant \frac{M-1}{2}\,\mu A_{M,0}$$

for any $1 \leqslant M \leqslant m$, where $\mu = \max_{i,k} \frac{\beta_{i,k}}{\alpha_{i,k}}$.

Proof.

$$\begin{aligned} A_{2,1} = \beta_{2,1} A_{1,0} &= \tfrac{1}{2}(\beta_{2,1} A_{1,0} + \beta_{2,1}\beta_{1,0}) \\ &= \tfrac{1}{2}(\mu_{2,1}\alpha_{2,1} A_{1,0} + \beta_{2,1}\mu_{1,0}) \\ &\leqslant \tfrac{1}{2}(\mu_{2,1}\alpha_{2,1} A_{1,0} + \beta_{2,1}\mu_{1,0} + \mu\beta_{2,0}) \\ &\leqslant \tfrac{1}{2}\,\mu(\alpha_{2,1} A_{1,0} + \beta_{2,1} + \beta_{2,0}) \\ &= \tfrac{1}{2}\,\mu A_{2,0} \end{aligned}$$

Proceed by induction: assume

$$A_{j1} \leqslant \frac{j-1}{2} \mu A_{j,0} \qquad \text{for} \quad 2 \leqslant j \leqslant M-1$$

then

$$\begin{aligned} A_{M,1} &= \sum_{j=2}^{M-1} \alpha_{M,j} A_{j,1} + \sum_{j=1}^{M-1} \beta_{M,j} A_{j,0} \\ &\leqslant \sum_{j=2}^{M-1} \alpha_{M,j} \frac{j-1}{2} \mu A_{j,0} + \sum_{j=1}^{M-1} \mu_{M,j} \alpha_{M,j} A_{j,0} \\ &\leqslant \frac{M-2}{2} \mu \sum_{j=1}^{M-1} \alpha_{M,j} A_{j,0} + \sum_{j=1}^{M-1} \mu_{M,j} \alpha_{M,j} A_{j,0} \end{aligned}$$

where the zero term $\alpha_{M,j} \frac{j-1}{2} \mu A_{j,0}$ for $j = 1$ was added in the first summation.

$$\begin{aligned} &= \frac{M-1}{2} \mu \sum_{j=1}^{M-1} \alpha_{M,j} A_{j,0} + \sum_{j=1}^{M-1} \left(\mu_{M,j} - \frac{1}{2} \mu \right) \alpha_{M,j} A_{j,0} \\ &\leqslant \frac{M-1}{2} \mu \sum_{j=1}^{M-1} \alpha_{M,j} A_{j,0} + \sum_{j=1}^{M-1} \left(\mu_{M,j} - \frac{1}{2} \mu_{M,j} \right) \alpha_{M,j} A_{j,0} \\ &= \frac{M-1}{2} \mu \sum_{j=1}^{M-1} \alpha_{M,j} A_{j,0} + \sum_{j=1}^{M-1} \frac{1}{2} \mu_{M,j} \alpha_{M,j} A_{j,0} \\ &\leqslant \frac{M-1}{2} \mu \sum_{j=1}^{M-1} \alpha_{M,j} A_{j,0} + \frac{1}{2} (M-1) \mu \sum_{j=1}^{M-1} \mu_{M,j} \alpha_{M,j} \\ &= \frac{M-1}{2} \mu \sum_{j=1}^{M-1} \alpha_{M,j} A_{j,0} + \frac{1}{2} (M-1) \mu \sum_{j=1}^{M-1} \beta_{M,j} \\ &\leqslant \frac{M-1}{2} \mu \left(\sum_{j=1}^{M-1} \alpha_{M,j} A_{j,0} + \sum_{j=0}^{M-1} \beta_{M,j} \right) \end{aligned}$$

where we added the nonnegative quantity $\frac{M-1}{2} \mu \beta_{M,0}$ to the second summation

$$= \frac{M-1}{2} \mu A_{M,0}$$

□

Lemma 2.3. For any method written in the form (2.1) above,

$$A_{M,k} \leqslant \frac{M-k}{k+1} \mu A_{M,k-1}$$

for any $1 \leqslant M \leqslant m$, where $\mu = \max_{i,k} \frac{\beta_{i,k}}{\alpha_{i,k}}$.

Proof. Using Lemma 2.2 as the base case, we show that if

$$A_{p,l} \leqslant \frac{p-l}{l+1}\,\mu A_{p,l-1} \qquad \text{for} \quad 1 \leqslant p \leqslant M-1 \qquad \text{and} \qquad 1 \leqslant l \leqslant p-1$$

then for any $k < M-1$,

$$\begin{aligned}
A_{M,k} &= \sum_{j=k+1}^{M-1} \alpha_{M,j} A_{j,k} + \sum_{j=k}^{M-1} \beta_{M,j} A_{j,k-1} \\
&\leqslant \sum_{j=k+1}^{M-1} \alpha_{M,j} \frac{j-k}{k+1}\,\mu A_{j,k-1} + \sum_{j=k}^{M-1} \beta_{M,j} A_{j,k-1} \\
&= \sum_{j=k}^{M-1} \alpha_{M,j} \frac{j-k}{k+1}\,\mu A_{j,k-1} + \sum_{j=k}^{M-1} \beta_{M,j} A_{j,k-1} \\
&\qquad \text{where the zero term is added to the first summation} \\
&\leqslant \frac{M-1-k}{k+1}\,\mu \sum_{j=k}^{M-1} \alpha_{M,j} A_{j,k-1} + \sum_{j=k}^{M-1} \beta_{M,j} A_{j,k-1} \\
&= \frac{M-k}{k+1}\,\mu \sum_{j=k}^{M-1} \alpha_{M,j} A_{j,k-1} + \sum_{j=k}^{M-1} \left(\beta_{M,j} - \frac{1}{k+1}\,\mu\alpha_{M,j}\right) A_{j,k-1} \\
&= \frac{M-k}{k+1}\,\mu \sum_{j=k}^{M-1} \alpha_{M,j} A_{j,k-1} + \sum_{j=k}^{M-1} \left(\mu_{M,j} - \frac{1}{k+1}\,\mu\right) \alpha_{M,j} A_{j,k-1} \\
&\leqslant \frac{M-k}{k+1}\,\mu \sum_{j=k}^{M-1} \alpha_{M,j} A_{j,k-1} + \sum_{j=k}^{M-1} \left(\mu_{M,j} - \frac{1}{k+1}\,\mu_{M,j}\right) \alpha_{M,j} A_{j,k-1} \\
&\qquad \text{because } \mu_{M,j} \leqslant \mu \\
&= \frac{M-k}{k+1}\,\mu \sum_{j=k}^{M-1} \alpha_{M,j} A_{j,k-1} + \sum_{j=k}^{M-1} \left(1 - \frac{1}{k+1}\right) \beta_{M,j} A_{j,k-1} \\
&= \frac{M-k}{k+1}\,\mu \sum_{j=k}^{M-1} \alpha_{M,j} A_{j,k-1} + \sum_{j=k}^{M-1} \frac{k}{k+1}\,\beta_{M,j} A_{j,k-1} \\
&\leqslant \frac{M-k}{k+1}\,\mu \sum_{j=k}^{M-1} \alpha_{M,j} A_{j,k-1} + \sum_{j=k}^{M-1} \frac{k}{k+1}\,\beta_{M,j} \frac{j-k+1}{k}\,\mu A_{j,k-2} \\
&\leqslant \frac{M-k}{k+1}\,\mu \sum_{j=k}^{M-1} \alpha_{M,j} A_{j,k-1} + \frac{k}{k+1} \frac{M-k}{k}\,\mu \sum_{j=k-1}^{M-1} \beta_{M,j} A_{j,k-2}
\end{aligned}$$

by adding the nonnegative term $\beta_{M,k-1} A_{k-1,k-2}$ to the second summation

$$= \frac{M-k}{k+1} \mu \left(\sum_{j=k}^{M-1} \alpha_{M,j} A_{j,k-1} + \sum_{j=k-1}^{M-1} \beta_{M,j} A_{j,k-2} \right)$$

$$= \frac{M-k}{k+1} \mu A_{M,k-1}$$

Finally, for the case $k = M-1$,

$$\begin{aligned}
A_{M,M-1} &= \beta_{M,M-1} A_{M-1,M-2} \\
&= \mu_{M,M-1} \alpha_{M,M-1} A_{M-1,M-2} \\
&= \frac{1}{M} \mu \alpha_{M,M-1} A_{M-1,M-2} - \frac{1}{M} \mu \alpha_{M,M-1} A_{M-1,M-2} \\
&\quad + \mu_{M,M-1} \alpha_{M,M-1} A_{M-1,M-2} \\
&= \frac{1}{M} \mu \alpha_{M,M-1} A_{M-1,M-2} + \left(\mu_{M,M-1} - \frac{1}{M} \mu \right) \alpha_{M,M-1} A_{M-1,M-2} \\
&\leqslant \frac{1}{M} \mu \alpha_{M,M-1} A_{M-1,M-2} + \left(\mu_{M,M-1} - \frac{1}{M} \mu_{M,M-1} \right) \alpha_{M,M-1} A_{M-1,M-2} \\
&= \frac{1}{M} \mu \alpha_{M,M-1} A_{M-1,M-2} + \frac{M-1}{M} \mu_{M,M-1} \alpha_{M,M-1} A_{M-1,M-2} \\
&= \frac{1}{M} \mu \alpha_{M,M-1} A_{M-1,M-2} + \frac{M-1}{M} \beta_{M,M-1} A_{M-1,M-2} \\
&\leqslant \frac{1}{M} \mu \alpha_{M,M-1} A_{M-1,M-2} + \frac{M-1}{M} \beta_{M,M-1} \frac{M-1-M+2}{M-1} \mu A_{M-1,M-3} \\
&= \frac{1}{M} \mu \alpha_{M,M-1} A_{M-1,M-2} + \frac{1}{M} \mu \beta_{M,M-1} A_{M-1,M-3} \\
&\leqslant \frac{1}{M} \mu \alpha_{M,M-1} A_{M-1,M-2} + \frac{1}{M} \mu \beta_{M,M-1} A_{M-1,M-3} \\
&\quad + \frac{1}{M} \mu \beta_{M,M-2} A_{M-2,M-3} \\
&= \frac{1}{M} \mu A_{M,M-2}
\end{aligned}$$

□

2.2. Upper Bound for the Optimal CFL Number of a m Stage Nth Order Method

The lemmas in the preceeding section suggest a bound on the optimal size of the CFL number c, which depends on the number of stages m and the order N of the method.

Proposition 2.1. Consider the family of m-stage, Nth order SSP Runge–Kutta methods (1.7) with nonnegative coefficients $\alpha_{i,k}$ and $\beta_{i,k}$. The CFL number c in (1.5) will be, at most, $c = m - N + 1$.

Proof. From the lemmas above, we see that for any $M \geqslant 1$

$$A_{M,k} \leqslant \frac{M-k}{k+1} \mu A_{M,k-1} \qquad \text{for} \quad 1 \leqslant k < M$$

and

$$A_{M,0} \leqslant M\mu$$

For an m-stage method to be Nth order, we must have [5]

$$A_{m,n} = \frac{1}{(n+1)!} \qquad \text{for} \quad n = 0, 1, \ldots, N-1 \tag{2.2}$$

so we have

$$\frac{1}{N!} = A_{m,N-1} \leqslant \frac{m-N+1}{N} \mu A_{m,N-2} = \frac{m+1-N}{N} \mu \frac{1}{(N-1)!}$$

Consequently,

$$\frac{1}{m+1-N} \leqslant \mu$$

The CFL number c would be, at most, $c = m + 1 - N$. □

This, however, is only a bound, and does not mean that such a CFL number can actually be obtained. As we will show, there are many cases in which this optimal CFL number is indeed attainable.

2.3. Some Optimal SSP Runge–Kutta Methods

In this section we construct some m-stage Nth order SSP methods with optimal CFL. We start with a first order, m stage method with CFL $c = m$:

Proposition 2.2. The m stage method given by

$$
\begin{aligned}
u^{(0)} &= u^n \\
u^{(i)} &= \left(1 + \frac{\Delta t}{m} L\right) u^{(i-1)}, \qquad i = 1,\ldots, m \\
u^{n+1} &= u^{(m)}
\end{aligned}
\tag{2.3}
$$

is first order accurate, with CFL number $c = m$.

Proof. Since for each nonzero $\alpha_{i,k}$ we have $\alpha_{i,k} = 1$ and $\beta_{i,k} = \frac{1}{m}$, and $\beta_{i,k} = 0$ whenever $\alpha_{i,k} = 0$, we have $\beta_{i,k} = \frac{1}{m}\alpha_{i,k}$ and the CFL number $c = m$ is clear. To check that the order, we notice that

$$
\begin{aligned}
u^{n+1} &= \left(1 + \Delta t \frac{1}{m} L\right)^m u^n \\
&= \left(1 + \Delta t\, L + \Delta t^2 \frac{m-1}{2m} L^2 + \cdots + \frac{1}{m^m} \Delta t^m L^m\right) u^n
\end{aligned}
$$

since $\frac{m-1}{2m} \neq \frac{1}{2}$ for any m, we have

$$
u^{n+1} = (1 + \Delta t\, L + O(\Delta t^2))\, u^n
$$

a first order method. □

We note that although this method has a significantly higher CFL than the standard first order method (which is, of course, the forward-Euler method), it has a correspondingly higher computational cost. Although the stepsize can be increased by a factor of m, the computational cost is also increased by the same factor. We need to look not only at the CFL number, but also at the number of steps needed. To reflect this, we define the *effective* CFL number $c_{\text{eff}} = s_r c$ where s_r is the ratio of the number of steps needed for the standard method to the number of steps needed for the current method. Thus, for the method (2.2) the effective CFL is, in fact, $c_{\text{eff}} = 1$. However, this method is useful as a stepping-stone for higher order methods.

For any desired integer optimal CFL number c, a first order ($N=1$) method of this CFL number is then given (as in Proposition 2.2 above) by the m-stage method:

$$\begin{aligned} u^{(0)} &= u^n \\ u^{(i)} &= (1+\Delta t\,\mu L)\,u^{(i-1)}, \qquad i=1,\dots,m \\ u^{n+1} &= u^{(m)} \end{aligned} \tag{2.4}$$

where $m=c$ and $\mu=\frac{1}{c}$. The next proposition shows how we can recursively build higher order methods. Starting with this method as a building block, we add one stage and increase the order to two, without changing the CFL number. Following the procedure detailed below, we can then build m stage, $N=m+1-c$ order methods with the optimal CFL number c chosen above. However, there in no guarantee that these methods will prove to be SSP.

Proposition 2.3. For any given CFL number $c=\frac{1}{\mu}$, where μ is chosen so that c is a positive integer, the class of m stage, $N=(m+1-c)$ order schemes of the form

$$\begin{aligned} u^{(0)} &= u^n \\ u^{(i)} &= u^{(i-1)}+\mu\,\Delta t\,Lu^{(i-1)}, \qquad i=1,\dots,m-1 \\ u^{(m)} &= \sum_{k=0}^{m-2} \alpha_{m,k}u^{(k)}+\alpha_{m,m-1}(u^{(m-1)}+\mu\,\Delta t\,Lu^{(m-1)}), \\ u^{n+1} &= u^{(m)} \end{aligned} \tag{2.5}$$

is given recursively by the coefficients:

$$\begin{aligned} \alpha_{m,k} &= \frac{1}{k\mu}\,\alpha_{m-1,k-1}, & k &= 1,\dots,m-2 \\ \alpha_{m,m-1} &= \frac{1}{m\mu}\,\alpha_{m-1,m-2}, & \alpha_{m,0} &= 1-\sum_{k=1}^{m-1}\alpha_{m,k} \end{aligned} \tag{2.6}$$

where the initial method is that given by the c-stage, first order method (2.4) above.

Proof. In (2.5), for each $1\leqslant i\leqslant m-1$

$$\begin{aligned} u^{(i)} &= u^{(i-1)}+\mu\,\Delta t\,Lu^{(i-1)} \\ &= (1+\mu\,\Delta t\,L)^i\,u^{(0)} \end{aligned}$$

We rewrite the method (2.5) above as

$$
\begin{aligned}
u^{(n+1)} &= \sum_{k=0}^{m-2} \alpha_{m,k}(1+\mu\,\Delta t\,L)^k\,u^{(0)} + \alpha_{m,m-1}(1+\mu\,\Delta t\,L)(1+\mu\,\Delta t\,L)^{m-1}\,u^{(0)} \\
&= \left(\sum_{k=0}^{m-2} \alpha_{m,k} \sum_{j=0}^{k} \frac{k!}{j!\,(k-j)!}\,\mu^j\,\Delta t^j\,L^j + \alpha_{m,m-1} \sum_{j=0}^{m} \frac{m!}{j!\,(m-j)!}\,\mu^j\,\Delta t^j\,L^j\right) u^{(0)} \\
&= \left(\sum_{j=0}^{m} \alpha_{m,j} + \left(\sum_{j=1}^{m-2} \alpha_{m,j}\,\frac{j!}{(j-1)!} + \alpha_{m,m-1}\,\frac{m!}{(m-1)!}\right)\mu\,\Delta t\,L \right.\\
&\quad + \left(\sum_{j=2}^{m-2} \alpha_{m,j}\,\frac{j!}{2!\,(j-2)!} + \alpha_{m,m-1}\,\frac{m!}{2!\,(m-2)!}\right)\mu^2\,\Delta t^2\,L^2 \\
&\quad + \left(\sum_{j=3}^{m-2} \alpha_{m,j}\,\frac{j!}{3!\,(j-3)!} + \alpha_{m,m-1}\,\frac{m!}{3!\,(m-3)!}\right)\mu^3\,\Delta t^3\,L^3 + \cdots \\
&\quad + \left(\sum_{j=m-2}^{m-2} \alpha_{m,m-2}\,\frac{j!}{(j-m+2)!\,(m-2)!} + \alpha_{m,m-1}\,\frac{m!}{(m-2)!\,2!}\right)\mu^{m-2}\,\Delta t^{m-2}\,L^{m-2} \\
&\quad \left. + \alpha_{m,m-1}\,\frac{m!}{(m-1)!}\,\mu^{m-1}\,\Delta t^{m-1}\,L^{m-1} + \alpha_{m,m-1}\mu^m\,\Delta t^m\,L^m\right) u^{(0)}
\end{aligned}
$$

For this method to be Nth order, we must match this with the desired expansion

$$
u^{(n+1)} = \left(1 + \Delta t\,L + \frac{1}{2}\,\Delta t^2\,L^2 + \frac{1}{3!}\,\Delta t^3\,L^3 + \cdots + \frac{1}{N!}\,\Delta t^N\,L^N + O(\Delta t^{N+1})\right) u^{(0)}
$$

Clearly, for the m-stage method of the type (1.3) to be Nth order, the coefficients $\alpha_{i,k}$ must satisfy the order conditions:

$$
(\mu)^k \left(\sum_{j=k}^{m-2} \frac{j!}{(j-k)!}\,\alpha_{m,j} + \frac{(m)!}{(m-k)!}\,\alpha_{m,m-1}\right) = 1
$$

for $k = 0,\dots, N$. Correspondingly, the coefficients of a $(m+1)$-stage, $(N+1)$ order method must satisfy

$$
(\mu)^k \left(\sum_{j=k}^{m-1} \frac{j!}{(j-k)!}\,\alpha_{m+1,j} + \frac{(m+1)!}{(m+1-k)!}\,\alpha_{m+1,m}\right) = 1
$$

for $k = 0,\dots, N+1$.

Assume that we have a m stage N order method of the type (2.5). Using the recursive definition we obtain the coefficients of a $(m+1)$ stage method of the same type. The definition of $\alpha_{m+1,0}$ guarantees the correct $k=0$ order condition for the $(m+1)$ stage method. We proceed to show that the kth order condition for the m stage method together with the definition of the coefficients implies the $k+1$ order condition for the $m+1$ stage method:

$$
\begin{aligned}
1 &= (\mu)^k \left(\sum_{j=k}^{m-2} \frac{j!}{(j-k)!} \alpha_{m,j} + \frac{(m)!}{(m-k)!} \alpha_{m,m-1} \right) \\
&= (\mu)^k \left(\sum_{j=k}^{m-2} \frac{j!}{(j-k)!} (j+1)\, \mu\alpha_{m+1,j+1} + \frac{(m)!}{(m-k)!} (m+1)\, \mu\alpha_{m+1,m} \right) \\
&= (\mu)^{k+1} \left(\sum_{j=k+1}^{m-1} \frac{j!}{(j-(k+1))!} \alpha_{m+1,j} + \frac{(m+1)!}{((m+1)-(k+1))!} \alpha_{m+1,m} \right)
\end{aligned}
$$

The $k=0,..., N$ order conditions for the m stage method imply the $k=1,..., N+1$ order conditions for the $(m+1)$ stage method, and the $k=0$ order condition is true by definition. Thus, the order conditions for $k=0,..., N+1$ are satisfied and the $(m+1)$ stage method will be of order $(N+1)$. □

A scheme obtained in this way is SSP with CFL $c=\frac{1}{\mu}$ as long as the coefficients $\alpha_{i,k}$ are nonnegative. However, not all the methods generated in this way are SSP—most of them will have negative $\alpha_{i,k}$. Nevertheless, this method is useful for generating the following methods:

Method 1. The following are second order $(N=2)$ SSP methods with m stages and an optimal CFL number $c=m-1$:

$$
\begin{aligned}
u^{(0)} &= u^n \\
u^{(i)} &= \left(1+\frac{\Delta t}{m-1} L\right) u^{(i-1)}, \qquad i=1,..., m-1 \\
u^m &= \frac{1}{m} u^{(0)} + \frac{m-1}{m} \left(1+\frac{\Delta t}{m-1} L\right) u^{(m-1)} \\
u^{n+1} &= u^{(m)}
\end{aligned}
\tag{2.7}
$$

The CFL number of this method is clear by inspection. A quick verification of the order of this scheme follows:

$$u^{n+1} = \frac{1}{m}u^{(0)} + \frac{m-1}{m}\left(1+\frac{\Delta t}{m-1}L\right)^m u^{(0)}$$
$$= \left(\frac{1}{m}+\frac{m-1}{m}\left(1+m\frac{\Delta t}{m-1}L+\frac{m(m-1)}{2}\frac{\Delta t^2}{(m-1)^2}L^2+O(\Delta t^3)\right)\right)u^{(0)}$$
$$= \left(\frac{1}{m}+\frac{m-1}{m}+\Delta t\,L+\frac{1}{2}\Delta t^2 L^2+O(\Delta t^3)\right)u^{(0)}$$
$$= \left(1+\Delta t\,L+\frac{1}{2}\Delta t^2 L^2+O(\Delta t^3)\right)u^{(0)}$$

In fact, these methods are also nonlinearly second order [16]. Each such method uses m stages to attain the order usually obtained by a 2-stage method, but has CFL number $m-1$, thus the effective CFL number here is increased to $c_{\text{eff}} = \frac{2(m-1)}{m}$.

Method 2. Using the method in proposition (2.3) we generate methods of any order N with $m = N+1$ stages, which are SSP with CFL coefficient $c = 2$. Table I includes the coefficients of these methods. The effective CFL for these methods is also $c_{\text{eff}} = \frac{2N}{N+1} = \frac{2(m-1)}{m}$.

3. LINEAR CONSTANT COEFFICIENT OPERATORS WITH TIME DEPENDENT FORCING TERMS

As we have seen [5], SSP Runge Kutta methods suitable for a linear, constant coefficient ODE are easier to generate and have a higher CFL

Table I. Coefficients $\alpha_{m,j}$ of the m-Stage $N=(m-1)$ Order SSP Methods of the Form (2.5), Which Have CFL Number $c=2$

stages m	$\alpha_{m,0}$	$\alpha_{m,1}$	$\alpha_{m,2}$	$\alpha_{m,3}$	$\alpha_{m,4}$	$\alpha_{m,5}$	$\alpha_{m,6}$	$\alpha_{m,7}$	$\alpha_{m,8}$	$\alpha_{m,9}$
2	0	1								
3	$\frac{1}{3}$	0	$\frac{2}{3}$							
4	0	$\frac{2}{3}$	0	$\frac{1}{3}$						
5	$\frac{1}{5}$	0	$\frac{2}{3}$	0	$\frac{2}{15}$					
6	$\frac{1}{9}$	$\frac{2}{5}$	0	$\frac{4}{9}$	0	$\frac{2}{45}$				
7	$\frac{1}{7}$	$\frac{2}{9}$	$\frac{2}{5}$	0	$\frac{2}{9}$	0	$\frac{4}{315}$			
8	$\frac{2}{15}$	$\frac{2}{7}$	$\frac{2}{9}$	$\frac{4}{15}$	0	$\frac{4}{45}$	0	$\frac{1}{315}$		
9	$\frac{11}{81}$	$\frac{4}{15}$	$\frac{2}{7}$	$\frac{4}{27}$	$\frac{2}{15}$	0	$\frac{4}{135}$	0	$\frac{2}{2835}$	
10	$\frac{71}{525}$	$\frac{22}{81}$	$\frac{4}{15}$	$\frac{4}{21}$	$\frac{2}{27}$	$\frac{4}{75}$	0	$\frac{8}{945}$	0	$\frac{2}{14175}$

than SSP Runge Kutta methods for a nonlinear ODE. We wish to extend these nice results to the case of a constant linear operator with a time dependent forcing term. This is a case which also arises in linear PDEs with time dependent boundary conditions, and can be written as:

$$u_t = Lu + f(t) \tag{3.1}$$

where $u = [u_i]$ is a vector, $L = [L_{i,j}]$ is a constant matrix and $f(t) = [f_i(t)]$ is a vector of functions of t. This ODE is a linear time dependent ODE and as such, the Runge–Kutta methods derived above for a linear time-invariant ODE will not have the correct order. The problem is that the class of RK methods for linear, time dependent ODEs is not equivalent to those for linear time invariant ODEs [20]. However, if the functions $f(t)$ can be written in a suitable way, then we can convert the equation (3.1) to a linear constant-coefficient ODE.

The order conditions for a RK method are derived, without loss of generality [2], for autonomous system $y'(x) = g(y(x))$. The reason for the "no loss of generality" is that any system of the form

$$u'(x) = h(x, u(x))$$

can be converted to an autonomous system by setting

$$y(x) = \begin{pmatrix} x \\ u(x) \end{pmatrix}$$

and then

$$y'(x) = g(y(x)) = g\begin{pmatrix} x \\ u(x) \end{pmatrix} = \begin{pmatrix} 1 \\ h(x, u) \end{pmatrix}$$

In many cases, we can convert equation (3.1) to a linear, constant coefficient ODE using a similar transformation. We first write (or approximate, if necessary) $f(t)$ as

$$f_i(t) = \sum_{j=0}^{n} a_j^i q_j(t) = Aq(t)$$

where $A = [A_{i,j}] = [a_j^i]$ is a constant matrix and $q(t) = [q_j(t)]$ are a set of functions which have the property that $q'(t) = Dq(t)$, where D is a constant matrix. Once the approximation to $f(t)$ is obtained, the ODE (3.1) can be converted into the linear, constant coefficient ODE

$$y_t = My(t) \tag{3.2}$$

where

$$y(t) = \begin{pmatrix} q(t) \\ u(t) \end{pmatrix}$$

and

$$M = \begin{pmatrix} D & 0 \\ A & L \end{pmatrix}$$

Thus, an equation of the form (3.1) can be approximated (or given exactly) by an linear constant coefficient ODE, and the SSP Runge–Kutta methods derived in Sec. 2.3 can be applied to this case.

Remark. We stress that we are talking about preserving the stability properties of forward-Euler as applied to the equation $y_t = My$. It is possible (indeed, expected) that forward-Euler applied to $u_t = Lu$ may satisfy properties not satisfied when applied to $u_t = Lu + f(t)$. It is also possible that some properties satisfied by forward-Euler when applied to the exact equation $u_t = Lu + f(t)$ may not be satisfied once $f(t)$ is approximated.

Remark. To approximate the functions $f(t)$ we can use the polynomials $q_j(t) = t^j$. In this case, the differentiation matrix D is given by

$$D = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 2 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 3 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & n & 0 \end{pmatrix}$$

A better approximation can be obtained using the Chebyshev polynomials. For these polynomials the relationship between the polynomial and its derivative is given by $T'_n(x) = \sum_{j=0}^{n} b_j T_j(x)$ where (see [6]),

$$b_j = \begin{cases} n & \text{for} \quad j = 0, \quad \text{if } n \text{ is odd} \\ 2n & \text{for} \quad j > 0, \quad \text{if } j + n \text{ is odd} \end{cases}$$

In other words, the derivative matrix D takes the form:

$$D = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 3 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 8 & 0 & 8 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 5 & 0 & 10 & 0 & 10 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 12 & 0 & 12 & 0 & 12 & 0 & 0 & \cdots & 0 & 0 \\ 7 & 0 & 14 & 0 & 14 & 0 & 14 & 0 & \cdots & 0 & 0 \\ . & . & . & . & . & . & . & . & \cdots & . & . \\ n & 0 & 2n & 0 & 2n & 0 & 2n & 0 & \cdots & 2n & 0 \end{pmatrix} \quad \text{if } n \text{ is odd}$$

or

$$D = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 3 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 8 & 0 & 8 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 5 & 0 & 10 & 0 & 10 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 12 & 0 & 12 & 0 & 12 & 0 & 0 & \cdots & 0 & 0 \\ 7 & 0 & 14 & 0 & 14 & 0 & 14 & 0 & \cdots & 0 & 0 \\ . & . & . & . & . & . & . & . & \cdots & . & . \\ 0 & 2n & 0 & 2n & 0 & 2n & 0 & 2n & \cdots & 2n & 0 \end{pmatrix} \quad \text{if } n \text{ is even}$$

4. NUMERICAL RESULTS

We approximate the solution to the equation

$$u_t = u_{xx} + 4t^3 \qquad 0 \leqslant x \leqslant \pi \tag{4.1}$$

with initial condition

$$u(x, 0) = \sin(x)$$

and boundary conditions

$$u(0, t) = u(\pi, t) = t^4$$

This equation has the exact solution

$$u(x, t) = t^4 + e^{-t} \sin(x)$$

We employ the second order centered difference spatial discretization

$$u_{xx} \approx \frac{u_{j+1} - 2u_j + u_{j-1}}{\Delta x^2}$$

which gives us the linear operator

$$L = \frac{1}{\Delta x^2} \begin{pmatrix} -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 \\ . & . & . & . & . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -2 \end{pmatrix}$$

in

$$u_t = Lu + 4t^3 \tag{4.2}$$

To incorporate the time-dependent boundary conditions as well as the time dependent forcing term, we define

$$y = \begin{pmatrix} 1 \\ t \\ t^2 \\ t^3 \\ t^4 \\ u \end{pmatrix}$$

and the ODE becomes

$$
y_t = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\
0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\
0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\
0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\
0 & 0 & 0 & 4 & \frac{1}{\Delta x^2} & \frac{-2}{\Delta x^2} & \frac{1}{\Delta x^2} & 0 & 0 & \cdots & 0 & 0 \\
0 & 0 & 0 & 4 & 0 & \frac{1}{\Delta x^2} & \frac{-2}{\Delta x^2} & \frac{1}{\Delta x^2} & 0 & \cdots & 0 & 0 \\
0 & 0 & 0 & 4 & 0 & 0 & \frac{1}{\Delta x^2} & \frac{-2}{\Delta x^2} & \frac{1}{\Delta x^2} & \cdots & 0 & 0 \\
. & . & . & . & . & . & . & . & . & . & . & . \\
0 & 0 & 0 & 4 & 0 & 0 & \cdots & 0 & 0 & \frac{1}{\Delta x^2} & \frac{-2}{\Delta x^2} & \frac{1}{\Delta x^2} \\
0 & 0 & 0 & 4 & \frac{1}{\Delta x^2} & 0 & \cdots & 0 & 0 & 0 & \frac{1}{\Delta x^2} & \frac{-2}{\Delta x^2}
\end{pmatrix} y
$$

In all these numerical experiments we use $\Delta x = \frac{1}{101}$. The following time discretizations were used:

1. The first order forward-Euler discretization:

$$y^{n+1} = (1 + \Delta t\, L)\, y^n$$

2. The 6-stage ($m = 6$), 5th order ($N = 5$) method with CFL number $c = 2$, given in Table I:

$$
\begin{aligned}
u^{(0)} &= u^n \\
u^{(i)} &= \left(1 + \frac{\Delta t}{2} L\right) u^{(i-1)}, \qquad i = 1,..., 5 \\
u^{(6)} &= \frac{1}{9} u^{(0)} + \frac{2}{5} u^{(1)} + \frac{4}{9} u^{(3)} + \frac{2}{45} \left(1 + \frac{\Delta t}{2} L\right) u^{(5)} \\
u^{n+1} &= u^{(6)}
\end{aligned}
$$

The high order Runge–Kutta method was compared to the forward-Euler method. As predicted, the maximal time-step Δt allowed was doubled for method (2) compared to the forward-Euler method. In Figs. 1 and 2 we see the effects of instability in the forward-Euler method when the time-step Δt is too high. This method is stable when $\Delta t \leqslant \frac{1}{2}\Delta x^2 = \Delta t_1$, however once Δt is increased to $\Delta t = 0.51\Delta x^2$, the method becomes unstable in 400 iterations (Fig. 1). If we increase Δt to $\Delta t = \frac{3}{4}\Delta x^2$, the method becomes unstable in 20 iterations, and when $\Delta t = \Delta x^2$, the instability has destroyed the solution completely by 15 iterations (Fig. 2). The 6-stage, 5th order method is stable as long as $\Delta t \leqslant \Delta x^2$. Figure 3 shows that for $\Delta t = \Delta x^2$, the method is stable even at final time $t = 0.010195$, however, when the time step is raised to $\Delta t = 1.15\Delta x^2$, the wiggles characteristic of instability are apparent at time $t = 0.010146$, or 90 iterations. As expected, the time step allowed doubled.

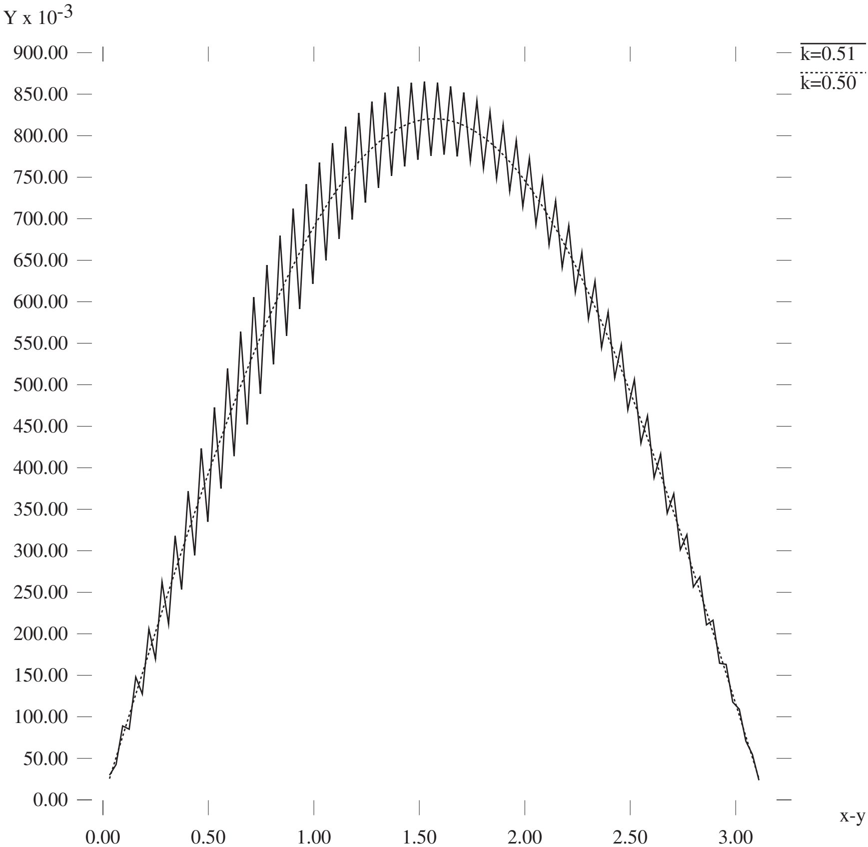

Fig. 1. Forward-Euler applied to the test problem. In each case, $\Delta t = k\,\Delta x^2$, where $\Delta x = \frac{1}{101}$. When $k \leqslant 0.5$ the method is stable. The numerical solution is shown for $k = 0.5$ after 408 time steps and for $k = 0.51$ after 400 time steps (final time$=$0.020047).

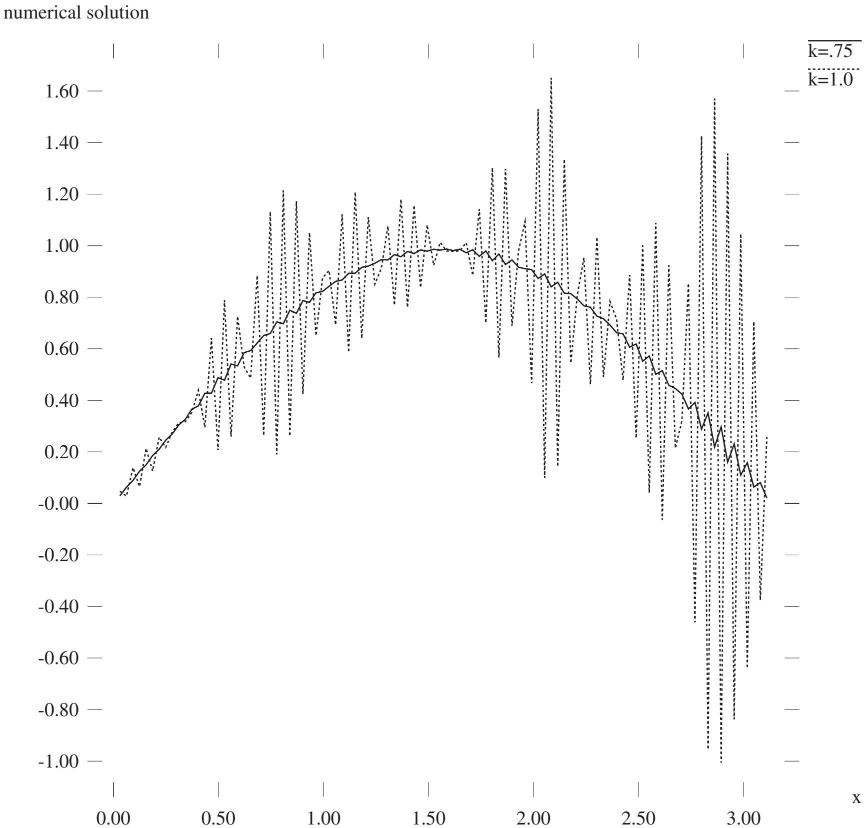

Fig. 2. Forward-Euler applied to the test problem. Once again, $\Delta t = k\,\Delta x^2$, where $\Delta x = \frac{1}{101}$. The method becomes unstable after very few time steps. Results shown are for $k = 0.75$ after 20 time steps and for $k = 1.0$ after 15 time steps. The instability apparent in the $k = 0.75$ case is worsened in the $k = 1.0$ case. We notice that the extent of instability in this example, for a fixed final time, depends not on the number of time steps, but mainly on the size of Δt.

An interesting point which arised from the technique used in Sec. 3 is that the time accuracy of the method is now important from the point of view of the first few elements in the new vector y. Since the time dependent boundary conditions or forcing is now not given explicitly, but by its differential equation $q_t = Dq$, the time-stepping method must also solve this ODE. If the time stepping method is not of a high enough order, the boundary conditions or forcing may not be resolved properly. In Fig. 4 we see the effect of numerically solving the ODE above on the term t^4. A method of fourth order or above will solve this exactly. We see that the

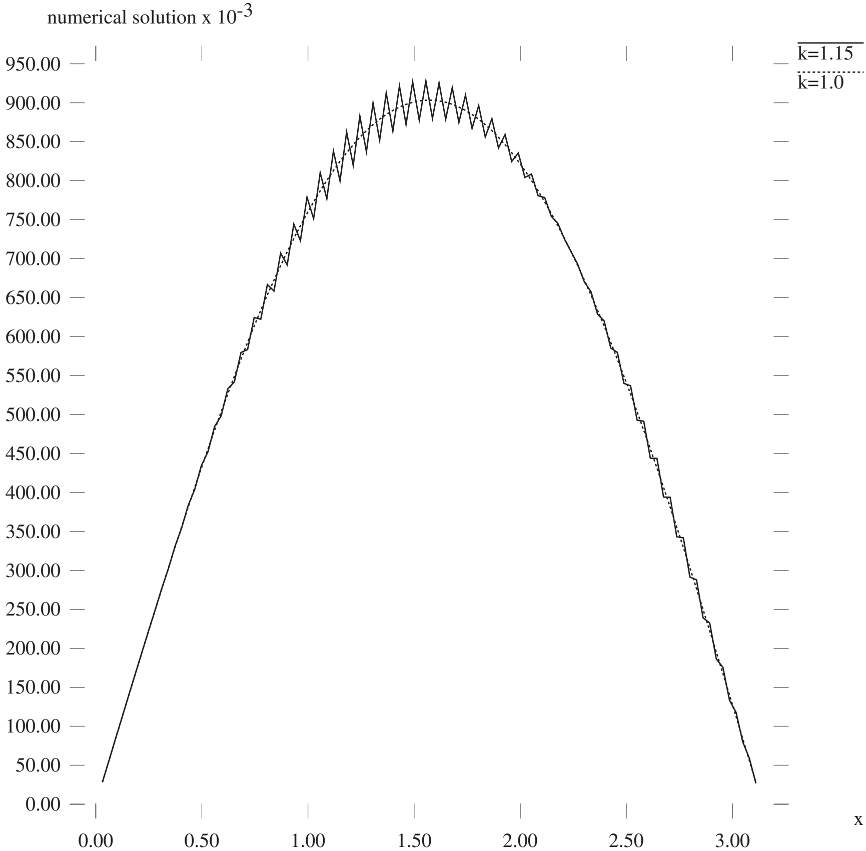

Fig. 3. The 5th order, 6 stage SSP Runge–Kutta method with CFL number $c = 2$ is applied to the test problem. Once again, $\Delta t = k\,\Delta x^2$, where $\Delta x = \frac{1}{101}$. The CFL of the SSP method guarantees that this method will be stable with double the maximal allowed step-size for the forward-Euler method. These results illustrate that the method is stable for $\Delta t \leqslant \Delta x^2$, and becomes unstable shortly after. The two curves shown are the numerical solutions for $k = 1.00$ after 104 time steps (final time$=$0.010195) and for $k = 1.15$ after 90 time steps (final time$=$0.010146). The instability is apparent in the $k = 1.15$ case.

fifth order method (2) solves it exactly, but a first order method (1) does not.

5. CONCLUDING REMARKS

While the development of SSP Runge–Kutta methods was primarily geared toward nonlinear operators, the wide applicability of these methods

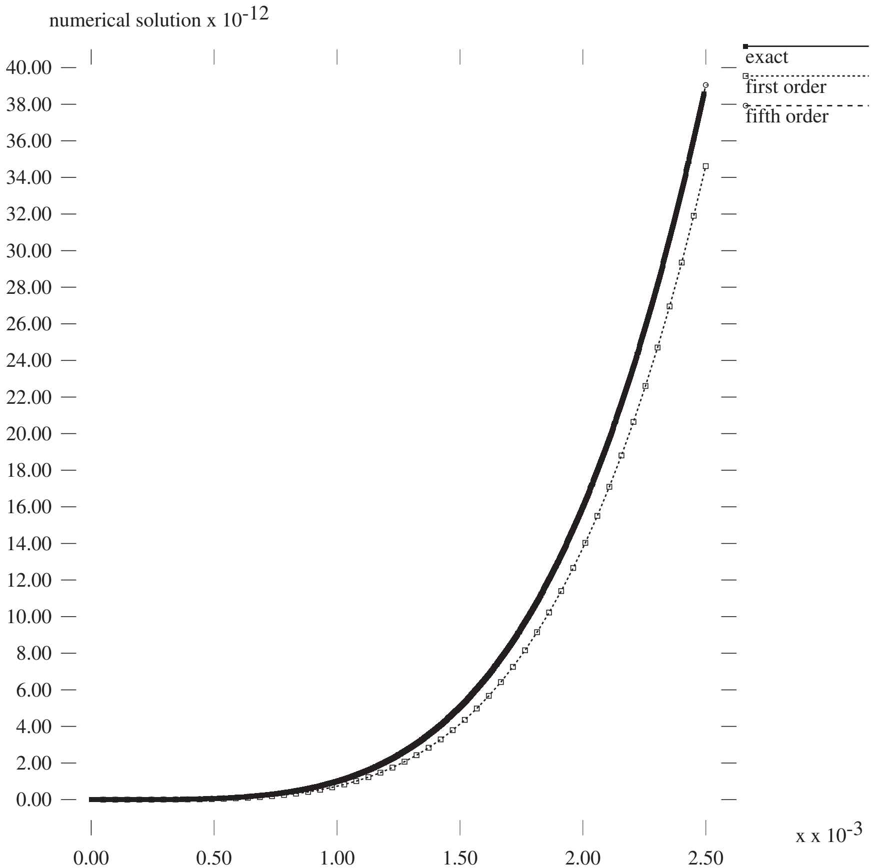

Fig. 4. The technique of Sec. 3 involves rewriting the forcing term $4t^3$ and the boundary conditions t^4 as the differential equation that governs them. This figure shows how t^4 is approximated by the fifth order method and the first order forward-Euler. As expected, the fifth order method captures the curve exactly while the first order method does not.

have motivated us to consider SSP methods for linear, time invariant operators. In [5] we presented a class of linear SSP Runge Kutta methods with m stages and of order m, which had optimal CFL number $c = 1$. Here we present a class of first order m stage methods with CFL $c = m$, a class of second order m stage methods with CFL $c = m-1$ and a class of $m-1$ order, m stage methods with CFL $c = 2$. We show that these methods are optimal, and that the optimal CFL for a Nth order m stage method is, at most, $c = m-N+1$. Although these results are not,in general, extendable to ODEs with time-dependent linear operators, we extend it to a special case of this class, which proves useful for linear PDEs with time dependent forcing or boundary conditions.

ACKNOWLEDGMENTS

Research supported by NSF Grant DMS-0106743.

REFERENCES

1. Allen, M. B., and Isaacson, E. L. (1998). *Numerical Analysis for Applied Science*, John Wiley, New York.
2. Butcher, J. C. (1987). *The Numerical Analysis of Ordinary Differential Equations: Runge–Kutta and General Linear Methods*, John Wiley, New York,.
3. Cockburn, B., and Shu, C.-W. (1989). TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws II: General framework. *Math. Comp.* **52**, 411–435.
4. Gottlieb, S., and Shu, C.-W. (1998). Total variation diminishing Runge–Kutta schemes. *Math. Comp.* **67**, 73–85.
5. Gottlieb, S., Shu, C.-W., and Tadmor, E. (2001). Strong stability preserving high order time discretization methods. *SIAM Rev.* **43**(1), 89–112.
6. Gottlieb, D., and Orszag, S. A. (1977). *Numerical Analysis of Spectral Methods: Theory and Applications*, Society for Industrial and Applied Mathematics, Philadelphia.
7. Gottlieb, D., and Tadmor, E. (1991). The CFL condition for spectral approximations to hyperbolic initial-boundary value problems. *Math. Comp.* **56**, 565–588.
8. Gustafsson, B., Kreiss, H. O., and Oliger, J. (1995). *Time Dependent Problems and Difference Methods*, John Wiley, New York.
9. Harten, A. (1983). High resolution schemes for hyperbolic conservation laws. *J. Comput. Phys.* **49**, 357–393.
10. Kurganov, A., and Tadmor, E. (1999). New high-resolution schemes for nonlinear conservation laws and related convection-diffusion equations, UCLA CAM Report No. 99-16.
11. Levy, D., and Tadmor, E. (1998). From semi-discrete to fully discrete: Stability of Runge–Kutta schemes by the energy method. *SIAM Rev.* **40**, 40–73.
12. Osher, S., and Chakravarthy, S. (1984). High resolution schemes and the entropy condition. *SIAM J. Numer. Anal.* **21**, 955–984.
13. Osher, S., and Tadmor, E. (1988). On the convergence of difference approximations to scalar conservation laws. *Math. Comp.* **50**, 19–51.
14. Shu, C.-W. (1988). Total-variation-diminishing time discretizations. *SIAM J. Sci. Stat. Comput.* **9**, 1073–1084.
15. Shu, C.-W., and Osher, S. (1988). Efficient implementation of essentially non-oscillatory shock-capturing schemes. *J. Comput. Phys.* **77**, 439–471.
16. Ruuth, S. J., and Spiteri, R. J. (2001). *A New Class of Optimal High-Order Strong-Stability-Preserving Methods*, Unpublished manuscript (under review).
17. Ruuth, S. J., and Spiteri, R. J. (2001). *Two Barriers on Strong Stability Preserving Time Discretization Methods*, Unpublished manuscript (under review).
18. Sweby, P. K. (1984). High resolution schemes using flux limiters for hyperbolic conservation laws. *SIAM J. Numer. Anal.* **21**, 995–1011.

19. Tadmor, E. (1997). Approximate solutions of nonlinear conservation laws. In Quarteroni, A. (ed.), *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*, Lectures Notes from CIME Course Cetraro, Italy, Lecture Notes in Mathematics 1697, Springer-Verlag, 1998, pp. 1–150.
20. Verner, J. H. (1996). High-order explicit Runge–Kutta pairs with low stage order. *Appl. Numer. Methods* **22**, 345–357.