Intermediate Boundary Conditions for Runge-Kutta Time Integration of Initial-Boundary Value Problems

D. Pathria*

Abstract

Pseudospectral and high-order finite difference methods are well established for solving time-dependent partial differential equations by the method of lines. The use of high-order spatial discretizations has led in turn to a concomitant interest in high-order time stepping schemes, so that the temporal and spatial errors are of comparable magnitude. Explicit Runge-Kutta methods are widely used in practice, but a difficulty encountered with these is the loss of accuracy that results from wrong specifications of intermediate-stage boundary conditions. The best prescriptions to date can do no better than achieve third-order accuracy for general nonlinear problems. On the other hand, if these artificial boundary values are not explicitly prescribed but are computed by integrating the semi-discrete equations at the boundary, the maximum allowable time step is significantly reduced. The remedy proposed here is to prescribe analytically those values that would result from applying the Runge-Kutta solver at the boundaries, and hence maintain accuracy without incurring further step size restrictions. We describe in detail the implementation for hyperbolic equations, and present both scalar and vector examples.

Key words: initial boundary value problems, Runge-Kutta methods.

AMS subject classifications: 65L10, 65L06, 65M12.

*Division of Applied Mathematics, Brown University, Providence, RI 02912, U.S.A.

ICOSAHOM'95: Proceedings of the Third International Conference on Spectral and High Order Methods.

1 Introduction

A principal advantage of the method-of-lines approach for solving time-dependent partial differential equations is that it allows one to consider separately the issues related to the spatial and temporal discretizations. It is common practice to use standard ordinary differential equation (ODE) solvers for the time evolution, and among these, the one-step Runge-Kutta (RK) methods are a popular choice. A Runge-Kutta method constructs the numerical solution as linear combinations of approximations, usually of order lower than the scheme, computed at stages intermediate to the discrete time levels. Suppose we wish to solve

$$u_t = \mathcal{F}u, \tag{1}$$

where $u(t)$ may be a vector and where $\mathcal{F}$ is some operator acting on u. Given u^m, the numerical solution at time t_m, an explicit s-stage RK method forms intermediate values $u^{(1)}, u^{(2)}, \ldots, u^{(s)}$ according to

$$u^{(i)} = u^m + \Delta t \sum_{j=1}^{i-1} a_{ij} \mathcal{F} u^{(j)}, \tag{2}$$

from which the approximation at time level $t_{m+1} = t_m + \Delta t$ is assembled,

$$u^{m+1} = u^m + \Delta t \sum_{i=1}^{s} b_i \mathcal{F} u^{(i)}. \tag{3}$$

The order and stability properties of the method are determined by the particular values of the coefficients a_{ij} and b_i. Our interest here is in (stable) methods which, for sufficiently smooth problems and sufficiently small time steps, are of order $p > 3$ for general nonlinear $\mathcal{F}$. Such methods are particularly valuable when the ordinary differential equations originate in the semi-discretization of partial differential equations. High accuracy is needed to reduce phase errors in the computed solution, and especially important in long time integrations, so that time stepping

schemes of error comparable to the spatial discretization are used. In what follows, we assume that the dominant error is due to the time stepping, and that the spatial accuracy is of order at least as high as the Runge-Kutta method in question.

Suppose the problem (1) does come from a partial differential equation. Then u depends on other variables such as x, $\mathcal{F}$ involves spatial differential operators (or their discrete counterparts), and boundary conditions supplement the initial value problem. Given a time-dependent Dirichlet boundary condition,

$$u(x_0, t) = g(t), \tag{4}$$

it is usual to prescribe $u_0^{m+1} = g(t_{m+1})$, where the subscript denotes the numerical value at x_0. The question now arises of what boundary conditions, if any, to impose on the intermediate values. This problem is not so trivial as it may first appear, and there are two basic approaches [1, 2]. The first is to ignore the given boundary condition (4) and to compute $u_0^{(i)}$ by extending the inner Runge-Kutta scheme up to the boundary (one-sided stencils may be used with finite difference methods, whereas the discrete differentiation operator is automatically defined for pseudospectral discretizations). With this approach, the formal order of accuracy is preserved, but the step size one may take is significantly smaller than that allowed if intermediate boundary values are explicitly enforced [1]. The second approach is to explicitly prescribe intermediate stage boundary conditions, but it is not immediately obvious what these should be. A few specific formulations, known in the literature, are summarized below.

The conventional method views $u^{(i)}$ as an approximation to $u(t_m + c_i\Delta t)$ where $c_i = \sum_{j=1}^{i-1} a_{ij}$, and sets

$$u_0^{(i)} = g(t_m + c_i\Delta t) \tag{5}$$

at the end of each time increment. The connection between the ith stage and time $t_m + c_i\Delta t$ is somewhat vague, however, since an order p method does not necessarily have $u^{(i)} = u(t_m + c_i\Delta t) + O(\Delta t^{p+1})$. In fact, this strategy turns out to be a rather poor choice. For fixed CFL number[1] and general RK methods, the accuracy is reduced to second order across the domain [1]. A second strategy augments the ordinary differential equations at the interior nodes with

$$u_0'(t) = g'(t) \tag{6}$$

at the boundary. If the operator $\mathcal{F}$ were independent of Δt, the formal order of the RK solver would be retained. When operating with a fixed CFL number, however, the relationship between Δt and Δx results in a degradation of accuracy to third order, even for the case of constant coefficient scalar problems.

A third strategy, proposed by Carpenter et al. in the context of hyperbolic problems, essentially amounts to replacing every application of $\mathcal{F}$ in the Runge-Kutta algorithm with a time differentiation at the boundary [1]. For example, the classical four-stage fourth-order method (RK4)

$$\begin{aligned}
u^{(1)} &= u^m \\
u^{(2)} &= u^m + \frac{1}{2}\Delta t\mathcal{F}u^{(1)} \\
u^{(3)} &= u^m + \frac{1}{2}\Delta t\mathcal{F}u^{(2)} \\
u^{(4)} &= u^m + \Delta t\mathcal{F}u^{(3)} \\
u^{m+1} &= u^m + \frac{\Delta t}{6}\left(\mathcal{F}u^{(1)} + 2\mathcal{F}u^{(2)} + 2\mathcal{F}u^{(3)} + \mathcal{F}u^{(4)}\right)
\end{aligned}$$

is supplemented by the intermediate boundary conditions

$$\begin{aligned}
u_0^{(2)} &= g(t_m) + \frac{1}{2}\Delta t g'(t_m) \\
u_0^{(3)} &= g(t_m) + \frac{1}{2}\Delta t g'(t_m) + \frac{1}{4}\Delta t^2 g''(t_m) \\
u_0^{(4)} &= g(t_m) + \Delta t g'(t_m) + \frac{\Delta t^2}{2} g''(t_m) + \frac{\Delta t^3}{4} g'''(t_m).
\end{aligned}$$

This procedure may be viewed as augmenting the inner equations with the boundary system

$$u_0'(t) = v(t), v'(t) = w(t), w'(t) = g'''(t).$$

In general, the strategy integrates $u_t = \mathcal{F}u$ at the boundary, and then replaces $\mathcal{F}^k u$ with $\dfrac{d^k g(t)}{d\,t^k}$.

When $\mathcal{F}$ has no time dependence, this is exactly what is needed and the Runge-Kutta method retains its formal order of accuracy. When $\mathcal{F}$ depends on time, however, either explicitly or implicitly through some nonlinearity, the recipe is not quite right, and one can generally get no better than third order out of it. There is a simple reason for this difference. Carpenter's strategy holds in the time-independent case because it is exactly the Runge-Kutta method of the interior applied at the boundary. For the time-dependent case, the agreement between the boundary and interior treatments is only approximate, and this inconsistency forms a barrier to attaining higher order. The conventional boundary treatment deviates still further from the Runge-Kutta algorithm and, as a result, its order is even lower.

The main point of this paper is that a Runge-Kutta integrator with a high order of accuracy for pure initial value

[1] roughly, the ratio of the time step to the grid spacing *at the boundary*. We shall make this more precise in the context of our numerical experiments.

problems will very likely lose this advantage unless the intermediate boundary data exhibits the same time errors as the intermediate stage values of the Runge-Kutta method are designed to do. A second point is that, for reasons of numerical stability, it is preferable to derive analytic formulas for these intermediate boundary values rather than simply apply the Runge-Kutta solver there. The paper is structured as follows: in section 2, we explain the basic principle and prove that the formal order is preserved if the boundary procedure follows the Runge-Kutta steps. Nonlinear conservation laws are discussed in Section 3, and we close in Section 4 with some general remarks.

2 The general recipe

A Runge-Kutta method of order p will not in general have $u^{(i)} = u(t_m + c_i\Delta t) + O(\Delta t^p)$. The algorithm relies on a systematic cancellation of lower order errors made at intermediate stages to recover the full accuracy at the discrete time levels. If the intermediate values are modified at order less than p, the necessary cancellations may fail to take place and the overall accuracy reduced. This is especially true when assigning values to the boundary nodes. The discrepancy between what is assigned and what would have been computed from the internal scheme (integrating $u_t = \mathcal{F}u$ at the boundary) is reflected in the numerically computed space derivatives and, with Δt being related to the grid spacing through a CFL-type relationship, there follows a global reduction in the space-time order. A detailed examination of this process for the advection equation discretized by finite difference schemes can be found in [1]. The correction proposed there turns out to be a simple example of the general remedy: to examine the boundary values as the Runge-Kutta method would compute when no intermediate-stage boundary conditions are prescribed (and full order is achieved), and to enforce these, to $O(\Delta t^p)$ at least, using the differential equation to obtain analytic expressions.

Consider the one-dimensional scalar equation,

$$u_t = f(u)u_x, \qquad u(0,t) = g(t),$$

with $f(u) < 0$ on $0 \le x \le 1$. The first step in the method of lines procedure is the spatial discretization which involves replacing the continuous variable x by the discrete variable $\underline{x} = [x_0 = 0, x_1, \ldots, x_{N-1}, x_N = 1]^T$. Here, and in what follows, underscores denote vectors in $\Re^{N+1}$, while subscripts indicate the grid location. Matrices, in $\Re^{(N+1)\times(N+1)}$, are capitalized and D represents the discrete differential operator defined on all $N+1$ grid values. If we step the semi-discrete system forward using the classical Runge-Kutta integrator RK4, without imposing intermediate boundary values, the fully discrete method is

$$\begin{aligned}
\underline{u}^{(1)} &= \underline{u}^m \\
\underline{u}^{(2)} &= \underline{u}^{(1)} + \frac{\Delta t}{2}F(\underline{u}^{(1)})D\underline{u}^{(1)} \\
\underline{u}^{(3)} &= \underline{u}^{(1)} + \frac{\Delta t}{2}F(\underline{u}^{(2)})D\underline{u}^{(2)} \\
\underline{u}^{(4)} &= \underline{u}^{(1)} + \Delta t F(\underline{u}^{(3)})D\underline{u}^{(3)} \\
\underline{u}^{m+1} &= \underline{u}^{(1)} + \frac{\Delta t}{6}\Big(F(\underline{u}^{(1)})D\underline{u}^{(1)} + 2F(\underline{u}^{(2)})D\underline{u}^{(2)} \\
&\quad + 2F(\underline{u}^{(3)})D\underline{u}^{(3)} + F(\underline{u}^{(4)})D\underline{u}^{(4)}\Big) + \delta^{m+1}\underline{e}_0.
\end{aligned}$$

Here $F(\underline{u}) = \mathrm{diag}\,(f(u_0), f(u_1), \ldots, f(u_N))$, $\underline{e}_0 = [1,0,0,\ldots,0]^T$, and δ^{m+1} is defined so as to reset the physical boundary condition at the end of the time step. Since we assume that the spatial discretization is at least fourth order, for smooth solutions and sufficiently small time steps, the leading term in the local error due to the time stepping is $O(\Delta t^5)$.

Imposing intermediate-stage boundary conditions during the time increment results in a local modification of the above Runge-Kutta algorithm at each intermediate stage, viz.

$$\begin{aligned}
\underline{v}^{(1)} &= \underline{u}^m \\
\underline{v}^{(2)} &= \underline{v}^{(1)} + \frac{\Delta t}{2}F(\underline{v}^{(1)})D\underline{v}^{(1)} + \underline{\epsilon}^{(2)} \\
\underline{v}^{(3)} &= \underline{v}^{(1)} + \frac{\Delta t}{2}F(\underline{v}^{(2)})D\underline{v}^{(2)} + \underline{\epsilon}^{(3)} \\
\underline{v}^{(4)} &= \underline{v}^{(1)} + \Delta t F(\underline{v}^{(3)})D\underline{v}^{(3)} + \underline{\epsilon}^{(4)} \\
\underline{v}^{m+1} &= \underline{v}^{(1)} + \frac{\Delta t}{6}\Big(F(\underline{v}^{(1)})D\underline{v}^{(1)} + 2F(\underline{v}^{(2)})D\underline{v}^{(2)} \\
&\quad + 2F(\underline{v}^{(3)})D\underline{v}^{(3)} + F(\underline{v}^{(4)})D\underline{v}^{(4)}\Big) + \epsilon^{m+1}\underline{e}_0
\end{aligned}$$

where $\underline{\epsilon}^{(i)} = \epsilon^{(i)}\underline{e}_0$ are chosen to impose the desired artificial boundary values and ϵ^{m+1} sets $v_0(t_{m+1}) = g(t_{m+1})$. These modifications can be viewed as small input perturbations in the Runge-Kutta algorithm whose sum effect on the intermediate function values can be written

$$\underline{v}^{(i)} = \underline{u}^{(i)} + \tilde{\underline{\epsilon}}^{(i)}$$

where

$$\begin{aligned}
\tilde{\underline{\epsilon}}^{(1)} &= \underline{0} \\
\tilde{\underline{\epsilon}}^{(2)} &= \underline{\epsilon}^{(2)} \\
\tilde{\underline{\epsilon}}^{(3)} &= \underline{\epsilon}^{(3)} + \frac{\Delta t}{2}F(\underline{v}^{(2)})D\tilde{\underline{\epsilon}}^{(2)} + \\
&\quad \frac{\Delta t}{2}\tilde{E}^{(2)}F'(\underline{u}^{(2)} + \Theta^{(2)}\tilde{\underline{\epsilon}}^{(2)})D\underline{u}^{(2)}
\end{aligned}$$

$$\tilde{\underline{\epsilon}}^{(4)} = \underline{\epsilon}^{(4)} + \Delta t F(\underline{v}^{(3)}) D\tilde{\underline{\epsilon}}^{(3)} + \Delta t \tilde{E}^{(3)} F'(\underline{u}^{(3)} + \Theta^{(3)}\tilde{\underline{\epsilon}}^{(3)}) D\underline{u}^{(3)}.$$

Here, $\tilde{E}^{(i)} = \mathrm{diag}(\tilde{\epsilon}_0^{(i)}, \ldots, \tilde{\epsilon}_N^{(i)})$, $F'(\underline{u}) = \mathrm{diag}\,(f'(u_0), \ldots, f'(u_N))$, and $\Theta^{(i)} = \mathrm{diag}(\theta_0^{(i)}, \ldots, \theta_N^{(i)})$ with $0 \le \theta_j^{(i)} \le 1$. These expressions easily follow using Taylor's theorem under the assumption that f is continuously differentiable. If the numerical procedure is run with a fixed CFL ratio, that is, with $\Delta t/\Delta x$ constant for a finite difference method on a uniform grid or $\Delta t N^2$ constant for a pseudospectral approximation, then, near the boundary, the effects are of the order of the input perturbations,

$$\begin{aligned}
\tilde{\underline{\epsilon}}^{(1)} &= \underline{0} \\
\tilde{\underline{\epsilon}}^{(2)} &= O(\epsilon^{(2)}) \\
\tilde{\underline{\epsilon}}^{(3)} &= O(\epsilon^{(2)}, \epsilon^{(3)}) \\
\tilde{\underline{\epsilon}}^{(4)} &= O(\epsilon^{(2)}, \epsilon^{(3)}, \epsilon^{(4)}).
\end{aligned}$$

Away from the boundary, they are weaker: zero for finite difference schemes since D is band-limited, and smaller by a factor Δt for pseudospectral methods since the entries of D decay as N^{-2} off the diagonal.

At the end of the time step, $\underline{v}^{m+1} = \underline{u}^{m+1} + \tilde{\underline{\epsilon}}^{m+1}$, where the elements of $\tilde{\underline{\epsilon}}^{m+1}$ are $O(\epsilon^{(2)}, \epsilon^{(3)}, \epsilon^{(4)}, \epsilon^{m+1})$ in the neighbourhood of $x_0 = 0$, and zero (finite differences) or one order higher (pseudospectral) away from it. If $\epsilon^{(2)}$, $\epsilon^{(3)}$, and $\epsilon^{(4)}$ are $O(\Delta t^q)$, then for $q \ge 5$, fourth order accuracy is retained. For $1 \le q \le 4$, the local truncation error is $O(\Delta t^q)$ near x_0, and $O(\Delta t^{q-1})$ (pseudospectral) or zero (finite differences) away from it. In both cases, however, the boundary error restricts the global error to $O(\Delta t^q)$ [5]. Consequently, the order to which the prescribed boundary values deviate from those of the free Runge-Kutta method determines the order of the fully discrete scheme. To retain fourth order accuracy using RK4, the imposed values should agree to $O(\Delta t^4)$.

In terms of formal order, it would be enough to use a Runge-Kutta method without imposing intermediate boundary values. However, the stability of the scheme is noticeably reduced since $\mathcal{F}$ involves (high-order) spatial differentiation operators, and the inclusion of the boundary points is enough to produce a significant reduction in the maximum allowable time step. Consequently, while artificially imposed boundary values should agree with the numerical ones that would be obtained through recursive computations $\mathcal{F}\mathcal{F}\cdots\mathcal{F}u$, the spatial derivatives of u should be expressed as far as possible in terms of $g(t)$ and its time derivatives. If $\mathcal{F}$ is time-independent, each iterate $u^{(i)}$ is some linear combination of u, u_t, u_{tt}, and simply replacing every application of $\mathcal{F}$ by a time differentiation would be enough. This is the generalization of Carpenter's scheme, and we shall refer to it as the "linearly consistent" strategy. For time-dependent operators, $\mathcal{F}u^{(i)}$ no longer neatly corresponds to $u_t^{(i)}$, and its specification in terms of $g(t)$ becomes more complicated. In what follows, we consider the definition of "fully consistent" strategies for the solution of conservation laws.

3 Conservation laws

3.1 The scalar case

If we solve the scalar equation

$$u_t = f(u)u_x$$

using a Runge-Kutta method, the computations are easily traced by using the inverse relation

$$u_x = \frac{1}{f(u)} u_t \tag{7}$$

to express higher-order space-time derivatives and intermediate values as algebraic functions of u and its time derivatives. The classical fourth-order Runge-Kutta method, for example, becomes

$$\begin{aligned}
u^{(2)} &= u + \frac{\Delta t}{2} u_t \\
u^{(3)} &= u + \frac{\Delta t}{2} f^{(2)} u_x^{(2)} \\
&= u + \frac{\Delta t}{2}\frac{f^{(2)}}{f} u_t + \frac{\Delta t^2}{4}\frac{f^{(2)}}{f^2}\left(f u_{tt} - f_u u_t^2\right) \\
u^{(4)} &= u + \Delta t f^{(3)} u_x^{(3)} \\
&= u + \Delta t \frac{f^{(3)}}{f} u_t \\
&\quad + \frac{\Delta t^2}{2}\frac{f^{(3)}}{f^3}\left(f^{(2)} f u_t^2 - 2 f^{(2)} f_u u_t^2 + f^{(2)} f u_{tt}\right) \\
&\quad + \frac{\Delta t^3}{4}\frac{f^{(3)}}{f^4}\left(-2 f_u^{(2)} f f_u u_t^3 + 2 f_u^{(2)} f^2 u_t u_{tt}\right. \\
&\qquad -2 f^{(2)} f f_u u_t^3 - 6 f^{(2)} f f_u u_t u_{tt} \\
&\qquad \left. + f^{(2)} f^2 u_{ttt} + 6 f^{(2)} f_u^2 u_t^3\right) \\
&\quad + \frac{\Delta t^4}{8}\frac{f^{(3)} f_u^{(2)}}{f^4}\left(f_u^2 u_t^4 - 2 f f_u u_t^2 u_{tt} + f^2 u_{tt}^2\right)
\end{aligned}$$

where f denotes $f(u(t))$, $f^{(2)}$ denotes $f(u^{(2)})$, $f^{(3)}$ denotes $f(u^{(3)})$, and so on.

At an inflow boundary, given $u(x_0, t) = g(t)$, every term on the right hand side is known exactly so the analytical expressions can be explicitly imposed at intermediate

stages. In doing so, there is no need to express $f^{(2)}$, $f^{(3)}$, or $f_u^{(2)}$ in terms of u, u_t, u_{tt}, ..., since they may be evaluated without degrading the stability properties of the scheme (that is done by the iterated x-differentiation). Nevertheless, it is instructive to do so here and examine the Taylor series expansions,

$$\begin{aligned}
u^{(2)} &= u + \frac{\Delta t}{2} u_t \\
u^{(3)} &= u + \frac{\Delta t}{2} u_t + \frac{\Delta t^2}{4} u_{tt} \\
&\quad + \frac{\Delta t^3}{16 f^2} \left(f f_{uu} u_t^2 - 2 f_u^2 u_t^2 + 2 f f_u u_{tt} \right) u_t \\
&\quad + \frac{\Delta t^4}{96 f^2} \left(f f_{uuu} u_t^2 - 3 f_u f_{uu} u_t^2 + 3 f f_{uu} u_{tt} \right) u_t^2 \\
&\quad + O(\Delta t^5) \\
u^{(4)} &= u + \Delta t u_t + \frac{\Delta t^2}{2} u_{tt} + \frac{\Delta t^3}{8 f^2} \left(-2 f f_u u_t u_{tt} \right. \\
&\quad \left. + 2 f^2 u_{ttt} - f f_{uu} u_t^3 + 2 f_u^2 u_t^3 \right) \\
&\quad + \frac{\Delta t^4}{24 f^3} \left(6 f^2 f_u u_{tt}^2 + 24 f_u^3 u_t^4 - 18 f f_u f_{uu} u_t^4 \right. \\
&\quad \left. + 12 f^2 f_{uu} u_t^2 u_{tt} - 33 f f_u^2 u_t^2 u_{tt} + 2 f^2 f_{uuu} u_t^4 \right. \\
&\quad \left. + 6 f^2 f_u u_t u_{ttt} \right) + O(\Delta t^5).
\end{aligned}$$

Note that the first intermediate value of the conventional treatment differs from the one above in the $O(\Delta t^2)$ term. Accordingly, the resulting scheme is expected to have order two. If $u'(x_0, t) = g'(t)$ is adjoined to the equations of the interior, the difference occurs at the $O(\Delta t^3)$ term, for a third-order approximation. Further examination shows that Carpenter's scheme is exactly the Runge-Kutta sequence problems when f does not depend on u. For nonlinear problems, however, the second and third intermediate boundary values differ at the $O(\Delta t^3)$ term, and so third order accuracy is anticipated.

Example: We solve the nonlinear equation $u_t = (u^2)_x$ on the domain $0 \le x, t \le 1$, for the exact solution

$$u(x,t) = \frac{x}{2(2-t)}.$$

A Legendre pseudospectral approximation on $N+1$ Gauss-Lobatto-Legendre collocation points is used to approximate space derivatives. The classical fourth order Runge-Kutta integrator is used for the time stepping, subject to conventional, $u'(1,t) = g'(t)$, linearly consistent, fully consistent, and no intermediate boundary treatment at the inflow boundary $x = 1$. At $x = 0$, the solution is identically zero, and causes no interference with our results. With a CFL number fixed at $\Delta t = N^{-2}$, the errors at time $t = 1$ very nicely demonstrate the predicted convergence rates of 2, 3, 3, 4, and 4 respectively (Fig. 1). □

When the CFL ratio is increased to 6.5, the instability

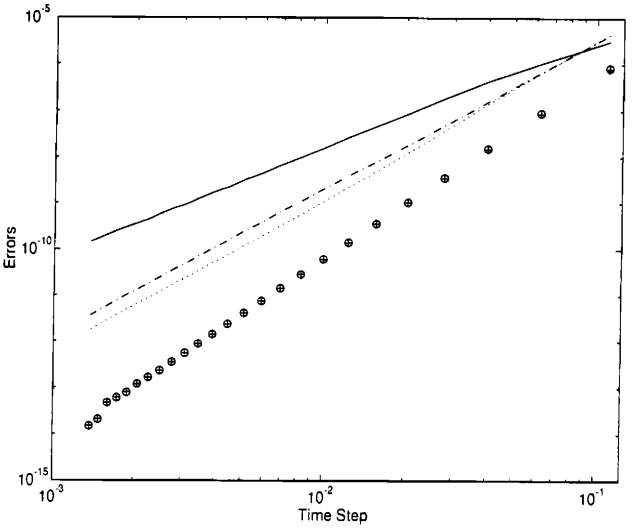

Figure 1: Log-log plot of the L_2 errors at time $t = 1$ for the nonlinear equation $u_t = (u^2)_x$, computed using conventional (–), $u'(1,t) = g'(t)$ (-.),a linearly consistent ($\cdots$), fully consistent (o), and no (+) intermediate conditions at $x = 1$. The discretization was Legendre pseudospectral in space, with $N = 3, \ldots, 27$, marched forward with RK4 and a fixed CFL ratio $\Delta t = 1/N^2$.

associated with free intermediate boundary conditions becomes apparent. Table 1 lists the CFL number (integer) at which each algorithm exhibited arithmetic overflow with polynomial degree $N = 25$. A four-fold gain is achieved by imposing *some* intermediate boundary conditions.

We make one note here, that the vanishing of $f(u)$ at the point $(1, t_m)$ causes no real difficulty. A simple remedy is to take a few very small time steps without imposing any boundary conditions at the intermediate stages.

3.2 The vector case

Now suppose that $\vec{u} \in \Re^n$, and we have the nonlinear hyperbolic system

$$\vec{u}_t = F(\vec{u}) \vec{u}_x$$

where $F(\vec{u}) \in \Re^{n \times n}$. The intermediate values for the classical Runge-Kutta method are then

$$\vec{u}^{(2)} = \vec{u} + \frac{\Delta t}{2} \vec{u}_t$$

Intermediate Boundary Treatment	CFL=1		CFL=6.5		Overflow At CFL #
	L_2	L_∞	L_2	L_∞	(N=25)
Conventional	2.3279	2.1662	2.0402	1.8399	28
Solve $u'(1,t) = g'(t)$	3.1646	2.8873	2.8841	2.6075	31
Linearly Consistent	3.3426	3.1312	3.0544	2.8499	31
Fully Consistent	4.0029	4.0011	3.7895	3.7873	34
None	3.9929	3.9923	.0030	-0.2490	7

Table 1: Convergence rates for the Legendre–RK4 Solution of $u_t = (u^2)_x$, based on linear least squares fit to the log-log errors at time $t = 1$, when using polynomials of degrees $N = 3, \cdots, 27$. Results are presented for CFL numbers $N^2\Delta t = 1$ and $N^2\Delta t = 6.5$. The table also lists the CFL ratio (integer value) at which the numerical procedure with $N = 25$ resulted in arithmetic overflow.

$$\vec{u}^{(3)} = \vec{u} + \frac{\Delta t}{2}F^{(2)}\vec{u}_x + \frac{\Delta t^2}{4}F^{(2)}\vec{u}_{xt}$$

$$\vec{u}^{(4)} = \vec{u} + \Delta t F^{(3)}\vec{u}_x + \frac{\Delta t^2}{2}F^{(3)}\left(F^{(2)}_{\vec{u}}\vec{u}^{(2)}_x\vec{u}_x + F^{(2)}\vec{u}_{xx}\right) + \frac{\Delta t^3}{4}F^{(3)}\left(F^{(2)}_{\vec{u}}\vec{u}^{(2)}_x\vec{u}_{xt} + F^{(2)}\vec{u}_{xxt}\right)$$

and all terms on the right can be written without reference to space derivatives. Letting $B(\vec{u}) = F^{-1}(\vec{u})$, we may form in sequence:

$$\begin{aligned}
\vec{u}_x &= B\vec{u}_t \\
B_x &= B_{\vec{u}}\vec{u}_x \\
B_t &= B_{\vec{u}}\vec{u}_t \\
\vec{u}_{xt} &= B_t\vec{u}_t + B\vec{u}_{tt} \\
B_{xt} &= B_{\vec{u}\vec{u}}\vec{u}_x\vec{u}_t + B_{\vec{u}}\vec{u}_{xt} \\
B_{tt} &= B_{\vec{u}\vec{u}}\vec{u}_t\vec{u}_t + B_{\vec{u}}\vec{u}_{tt} \\
\vec{u}_{xtt} &= B_{tt}\vec{u}_t + 2B_t\vec{u}_{tt} + B\vec{u}_{ttt} \\
\vec{u}_{xx} &= B_x\vec{u}_t + B\vec{u}_{xt} \\
\vec{u}_{xxt} &= B_{xt}\vec{u}_t + B_x\vec{u}_{tt} + B_t\vec{u}_{xt} + B\vec{u}_{xtt}
\end{aligned}$$

which, together with the relation $\vec{u}^{(2)}_x = \vec{u}_x + 1/2\Delta t\vec{u}_{xt}$, can be inserted into the expressions for $\vec{u}^{(3)}$ and $\vec{u}^{(4)}$ above to obtain the correct intermediate behaviour at the boundary.

This behaviour is explicitly known only when all characteristics at x_0 are flowing into the domain. More typically, the given boundary conditions will be fewer in number than the unknowns. Obtaining exact expressions for the intermediate values is as hard as solving the original problem, so numerical values of unspecified components of $\vec{u}$ and their time derivatives must be used. We can accept an $O(\Delta t^p)$ error in the intermediate function values; this is entirely consistent with the error in the numerical approximations. Temporal derivatives are needed at lower orders; specifically, an $O(\Delta t^{p-k})$ approximation of the kth time derivative is required. These may be calculated from stored values at previous time levels together with the current numerical value of $\vec{u}$. Weights for one-sided difference approximations are easily computed using Fornberg's algorithm [3].

Example: The shallow water wave equations for the

height of a fluid $h(x,t)$ and its velocity $u(x,t)$,

$$\begin{pmatrix} h \\ u \end{pmatrix}_t + \begin{pmatrix} u & h \\ G & u \end{pmatrix} \begin{pmatrix} h \\ u \end{pmatrix}_x = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \tag{8}$$

where G is the gravitational constant, have an exact solution

$$\begin{aligned} h(x,t) &= \xi^2 \\ u(x,t) &= 2\sqrt{G}\xi - 2\sqrt{GH} \end{aligned}$$

where $\xi = \dfrac{x + 2\sqrt{GH}\,t}{1 + 3\sqrt{G}\,t}$ and H represents the equilibrium height. Defining $\phi = u + 2\sqrt{Gh}$ and $\psi = u - 2\sqrt{Gh}$ converts the system to characteristic form,

$$\begin{pmatrix} \phi \\ \psi \end{pmatrix}_t + \frac{1}{4} \begin{pmatrix} 3\phi + \psi & 0 \\ 0 & \phi + 3\psi \end{pmatrix} \begin{pmatrix} \phi \\ \psi \end{pmatrix}_x = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

The problem could in fact be solved for the characteristic variables but, for the sake of generality, we continue to solve the system (8) for the primitive variables h and u. Boundary conditions are, however, imposed through the characteristics for reasons of stability [4]. With $G = H = 1$, $\phi(x,t)$ flows to the right for $1 \le x \le 3/2$, while $\psi(x,t)$ flows to the left. so that one boundary condition is required at each endpoint. For illustration, we take a common sort of specification

$$h(1,t) = g_1(t), \qquad u(3/2,t) = g_2(t)$$

for which the characteristic boundary conditions become

$$\phi(1,t) = 4\sqrt{Gg_1(t)} + \psi(1,t), \quad \psi(3/2,t) = 2g_2(t) - \phi(3/2,t).$$

These are imposed as follows. Consider the point $x_0 = 1$ (the point $x_N = 3/2$ is treated similarly). Having calculated u_0^m and h_0^m numerically by applying the Runge-Kutta method at all points, the boundary values are adjusted as

$$\begin{aligned} \psi_0^m &\leftarrow u_0^m - 2\sqrt{Gh_0^m} \\ \phi_0^m &\leftarrow 4\sqrt{Gg_1(t_m)} + \psi_0^m \\ h_0^m &\leftarrow \frac{1}{16G}(\phi_0^m - \psi_0^m)^2 \\ u_0^m &\leftarrow \frac{1}{2}(\phi_0^m + \psi_0^m). \end{aligned}$$

Exactly the same sequence is followed at intermediate stages, using the numerically computed values of $u_0^{(i)}$ and $h_0^{(i)}$.

The linearly and fully consistent schemes require approximations to the first three time derivatives. The derivatives of ψ_0 are computed using $\psi_0^{m-4}, \psi_0^{m-3}, \ldots, \psi_0^m$; from these those of ϕ_0 may be formed,

$$\frac{\partial^k \phi_0^m}{\partial t^k} \leftarrow 4\frac{d^k \sqrt{Gg_1(t_m)}}{dt^k} + \frac{\partial^k \psi_0^m}{\partial t^k},$$

and subsequently the approximations to the derivatives of h_0^m and u_0^m. Starting data is generated by taking a few very small steps without imposing intermediate boundary conditions. For the conventional treatment and integrating the boundary conditions, which need exact values at intermediate times, the boundary conditions are imposed directly on the primitive variables h and u. There is no exact value, not even an $O(\Delta t^p)$ approximation, of $u(1,t)$ at time $t_{m+1/2}$ say, so that $\psi(1, t_m + \Delta t/2)$ cannot be computed (although it could be extrapolated from stored values).

Figure 2 shows the error decay when the various boundary treatments are applied at CFL numbers 0.2, 2, and 3, while Table 2 lists the convergence rates as found by linear least-squares fitting. Fourth-order accuracy is obtained by the fully consistent treatment at all CFL numbers, and the method allows a time step far greater than that permitted when no intermediate boundary conditions are imposed. □

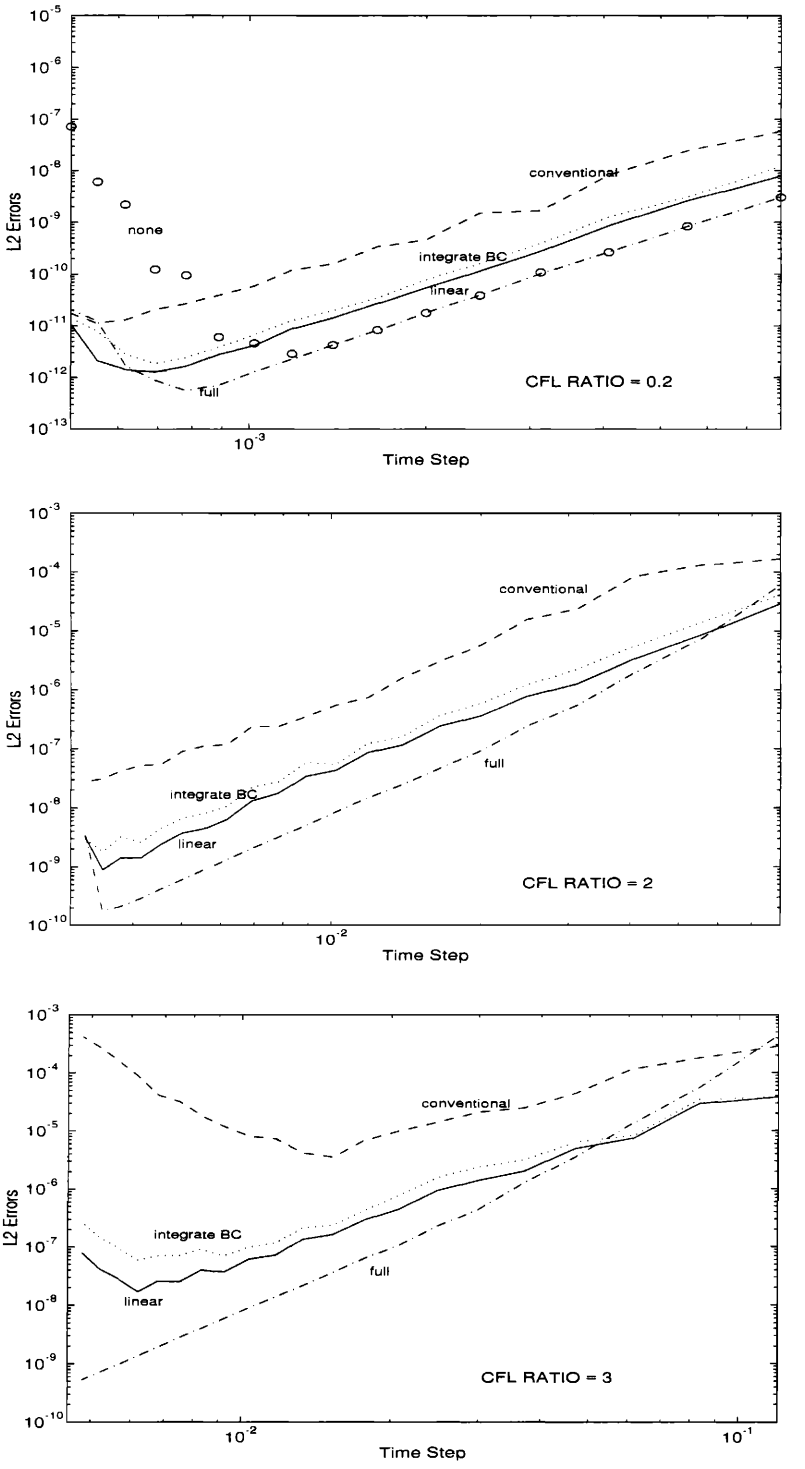

Figure 2: Log-log plot of the L_2 errors of the Legendre-RK4 solution of the shallow water wave equations at time $t = 1$. Fixed CFL ratios of 0.2, 2, and 3 were used for $N = 5, \cdots, 25$ in each case. The convergence rates for the different boundary conditions are listed in Table 2.

Intermediate Boundary Treatment	CFL=0.2		CFL=2		CFL=3		Overflow at CFL #
	L_2	L_∞	L_2	L_∞	L_2	L_∞	(N=25)
Conventional	3.3027	3.4850	2.9913	2.9101	2.1078	2.0313	3.196
$h'(1,t) = g_1'(t), \quad u'(3/2,t) = g_2'(t)$	3.6434	3.4850	3.2166	3.0047	2.4779	2.4089	3.264
Linearly Consistent	3.6665	3.5594	3.2959	3.0961	2.7039	2.5564	3.230
Fully Consistent	3.7887	3.6679	3.9146	3.9475	4.5357	4.5308	3.400
None	3.7253	3.6014	*****	*****	*****	*****	0.272

Table 2: Convergence rates for the Legendre–RK4 Solution of the shallow water wave equations, with CFL numbers 0.2, 2 and 3. In each case, the rate was calculated by a linear least-squares fit over those points where the L_2 error was strictly decreasing as Δt was reduced. The last column lists the CFL ratios $\Delta t N^2$ at which arithmetic overflow occured when solving the shallow water wave equations on an $N = 25$ grid up to time $t = 1$.

4 Conclusions

In this paper, we have presented a logical and consistent principle for formulating intermediate boundary conditions for Runge-Kutta methods, and have demonstrated its usefulness for the solution of nonlinear conservation laws. The basic idea is that prescribed boundary values at intermediate stages should exhibit the same time errors as the Runge-Kutta method would compute when no intermediate boundary conditions are imposed. The importance of matching interior and boundary errors has appeared before, notably in connection with operator-splitting methods [6], but appears not to be widely recognized in the Runge-Kutta context. For stability reasons, it is preferable to derive analytic formulas for the error terms than to form them using numerical space derivatives. Much remains to be done, however, to adapt this idea to problems in higher space dimension and for more general operators.

We have confined our attention here to the temporal accuracy of the scheme, but similar arguments can be made regarding the spatial approximation. Our analysis at the boundary has completely ignored the presence of discretization errors in the interior. Artificial boundary conditions should ideally take this into account, so that the numerical solution over the full domain is as smooth as possible. The treatment presented here for the time errors should be complemented in space, possibly through the use of penalty methods, and remains the subject of future work.

References

[1] M. H. Carpenter, D. Gottlieb, S. Abarbanel, and W.-S. Don. The Theoretical Accuracy of Runge-Kutta Time Discretizations for the Initial Boundary Value Problem: A Careful Study of the Boundary Error. *SIAM Sci. Comput.*, 16(8):1241–1245, 1995.

[2] C. M. de Sterke, K. R. Jackson, and B. D. Robert. Nonlinear coupled-mode equations on a finite interval: a numerical procedure. *J. Opt. Soc. Am. B*, 8(2):403–412, 1991.

[3] B. Fornberg. Generation of Finite Difference Formulas for Arbitrarily Spaced Grids. *Math. Comp.*, 51:699–706, 1988.

[4] D. Gottlieb, M. Gunzburger, and E. Tadmor. On Numerical Boundary Treatment of Hyperbolic Systems for Finite Difference and Finite Element Methods. *SIAM J. Num. Anal.*, 19(4):671–682, 1982.

[5] B. Gustafsson. The Convergence Rate for Difference Approximations to Mixed Initial Boundary Value Problems. *Math. Comp.*, 29(130):396–406, 1975.

[6] R. J. LeVeque. Intermediate Boundary Conditions for Time-Split Methods Applied to Hyperbolic Partial Differential Equations. *Math. Comp.*, 47(175):37–54, 1986.