References

[1] L. N. Trefethen, Stability of Finite-Difference Models Containing Two Boundaries or Interfaces, *Math. Comp.* **45**, 172, 1985, pp. 279-300.

[2] M.H. Carpenter, D. Gottlieb, and S. Abarbanel, The Stability of Numerical Boundary Treatments for Compact High-Order Finite-Difference Schemes, NASA Contractor Report 187628, ICASE Report No. 91-71, Sept 1991. To appear in *JCP*.

[3] M.H. Carpenter, D. Gottlieb, and S. Abarbanel, Time-Stable Boundary Conditions for Finite-Difference Schemes Solving Hyperbolic Systems: Methodology and Application to High-Order Compact Schemes, NASA Contractor Report 191436, ICASE Report No. 93-9, March 1993. To appear in *JCP*.

[4] H.-O. Kreiss, and G. Scherer, Finite element and finite difference methods for hyperbolic partial differential equations, *Mathematical Aspects of Finite Elements in Partial Differential Equations*, (Academic Press, New York, 1974).

[5] B. Strand, Summation by Parts for Finite Difference Approximations for d/dx, Dept of Scientific Computing, Uppsala University, Uppsala, Sweden, Aug, 1991.

[6] M.H. Carpenter, D. Gottlieb, and S. Abarbanel, Stable and Accurate Boundary Treatments for Compact, High Order Finite Difference Schemes, *Applied Numer. Math.*, 12 (1993) pp. 55-87.

[7] P. Olsson, Summation by Parts, Projections, and Stability, RIACS Technical Report 93.04, June 1993.

[8] B. Gustafsson, The Convergence Rate for Difference Approximations to Mixed Initial Boundary Value Problems, *Math. Comp.* **29**, 130, 1975, pp. 396-406.

Rewriting equation (AI. 5) in terms of V and D one obtains:

$$
\begin{aligned}
(PV^1)_j &= (PV^n)_j + \frac{\lambda}{2}[(PDPV^n)_j + g(t)PD(e_0)_j] \qquad 1 \le j \le N \\
&= (PV^n)_j + \frac{\lambda}{2}(PDV^n)_j = P[(V^n) + \frac{\lambda}{2}(DV^n)]_j
\end{aligned}
\tag{AI. 9}
$$

It is clear that the reduced vector PV^1 is identical to the vector at time level *1* obtained from the conventional imposition of boundary conditions for $1 \le j \le N$ [see equation (5)]. Noting this equivalence, equation (AI. 6) can now be interpreted as

$$
\begin{aligned}
P(V^2)_j &= P(V^n)_j + \frac{\lambda}{2}[(PDPV^1)_j + g(t + \frac{\delta t}{2})PD(e_0)_j] \qquad 1 \le j \le N \\
&= P(V^n)_j + \frac{\lambda}{2}(PDV^1)_j = P[(V^n) + \frac{\lambda}{2}(DV^1)]_j]
\end{aligned}
\tag{AI. 10}
$$

The reduced vector PV^2 is identical to that obtained with the conventional boundary conditions for $1 \le j \le N$. This procedure can be used to show that each stage of the conventional boundary procedure is equivalent to that obtained from solving equations (AI. 5) - (AI. 8). Thus, the entire procedures are equivalent.

Appendix A

Once the spatial operator has been chosen, a P.D.E. becomes a system of O.D.E.'s, plus a boundary term. If the boundary term is treated as a source term, then entire system can then be treated by conventional techniques. We show that this technique suffers from the same loss of accuracy at the boundary as was discussed earlier.

We start with the governing equations

$$\begin{aligned} \frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} &= 0 \quad 0 \le x \le 1,\; t \ge 0 && \text{(AI. 1)} \\ u(0,t) &= g(t) && \text{(AI. 2)} \end{aligned}$$

The semi-discretized version of (AI. 1) - (AI. 2) is

$$\begin{aligned} \frac{dv_i}{dt} &= \frac{1}{h}(Dv)_i \quad i = 1, ..., N;\; t \ge 0 && \text{(AI. 3)} \\ v_0(t) &= g(t) && \text{(AI. 4)} \end{aligned}$$

where $V = {v_i}^T, i = 0, ..., N$. Define the reduced identity matrix P by

$$P = \begin{bmatrix} 0 & & & & \\ & 1 & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & \cdot \\ & & & & & 1 \end{bmatrix}$$

where the matrix P is dimension (N+1,N+1). Similarly, $e_0 = [1, 0, ..., 0]^T$, and has dimension (N+1). Next define the new variables $\tilde{V} = PV$, $\tilde{D} = PDP$ and $\tilde{q}_0 = PDe_0$. Now replace equation (AI. 1) by the fully discrete source equations:

$$\begin{aligned} \tilde{V}_j^1 &= \tilde{V}_j^n + \frac{\lambda}{2}[(\tilde{D}\tilde{V})_j^n + g(t)(\tilde{q}_0)_j] \qquad 1 \le j \le N && \text{(AI. 5)} \\ \tilde{V}_j^2 &= \tilde{V}_j^n + \frac{\lambda}{2}[(\tilde{D}\tilde{V})_j^1 + g(t + \frac{\delta t}{2})(\tilde{q}_0)_j] && \text{(AI. 6)} \\ \tilde{V}_j^3 &= \tilde{V}_j^n + \lambda[(\tilde{D}\tilde{V})_j^2 + g(t + \frac{\delta t}{2})(\tilde{q}_0)_j] && \text{(AI. 7)} \\ \tilde{V}_j^{n+1} &= \tilde{V}_j^n + \frac{\lambda}{6}[(\tilde{D}\tilde{V}^n)_j + 2(\tilde{D}\tilde{V}^1)_j + 2(\tilde{D}\tilde{V}^2)_j + (\tilde{D}\tilde{V}^3)_j] \\ &+ \frac{\lambda}{6}[\, g(t) + 4\, g(t + \frac{\delta t}{2}) + g(t + \delta t)](\tilde{q}_0)_j && \text{(AI. 8)} \end{aligned}$$

and in a similar vain,

$$
\begin{aligned}
\vec{G}^1 &= \vec{g}(t) + \frac{2\delta t}{3}\vec{g}'(t) + \frac{2(\delta t)^2}{9}\vec{g}''(t) - \{\, \vec{u}(t) - \frac{2\delta t}{3}\frac{\partial}{\partial x}[\vec{f}(\vec{u}(t) + \frac{\delta t}{3}\vec{f}_x(\vec{u})]\}_0 &(46)\\
&= \vec{g}(t) + \frac{2\delta t}{3}\vec{g}'(t) + \frac{2(\delta t)^2}{9}\vec{g}''(t) - \{\, \vec{u} - \frac{2\delta t}{3}\frac{\partial}{\partial x}[\vec{f}(\vec{u}(t)) + \frac{\delta t}{3}\vec{f}_{\vec{u}}\,\vec{u}_t + O([\delta t]^2)]\}_0 &(47)
\end{aligned}
$$

but $\vec{f}_{\vec{u}}\vec{u}_t = A(\vec{u})\vec{u}_t = \vec{f}_t$ and $\frac{\partial}{\partial x}[\vec{f}_t] = \frac{\partial}{\partial t}(\vec{f}_x) = \vec{u}_{tt}$. Thus

$$
\begin{aligned}
\vec{G}^1 &= \vec{g}(t) + \frac{2\delta t}{3}\vec{g}'(t) + \frac{2(\delta t)^2}{9}\vec{g}''(t) - \{\, \vec{u}^n - \frac{2\delta t}{3}\vec{u}_t + \frac{2(\delta t)^2}{9}\vec{u}_{tt}\}_0 + O([\delta t]^3)\\
\vec{G}^1 &= O([\delta t]^3) &(48)
\end{aligned}
$$

The consequence of equations (45) and (48), substituted into equation (40) with $\vec{v}^n \to \vec{u}(x,t)$ is that the error is proportional to

$$
\frac{|\vec{V}_i^{n+1} - \vec{u}_i(t + \delta t)|}{\delta t} = O([\delta t]^2); \quad i = 0, 1, ..., m \tag{49}
$$

near the boundary, (m finite), and proportional to $O([\delta t]^3)$ in the interior. In other words , the boundary conditions prescribed in equations (35) - (37) give us the required overall accuracy. Unfortunately, this procedure does not seem to generalize to RK integrations of order higher than three for the case of non-linear systems. The main reason is that beyond $\vec{u}_{tt}$, we will need to use the "Jacobian of the Jacobian" which can not be related simply to the temporal derivatives of the vector $\vec{u}(x,t)$.

4. Conclusion

We have shown that the imposition of inflow boundary conditions at the intermediate steps of Runge-Kutta algorithms for hyperbolic P.D.E.'s has to be done in a counter-intuitive manner, if one is to preserve the overall accuracy of the scheme. The conventional (or "natural") way of assigning at level $n + \alpha_i$, $(0 < \alpha_i \leq 1)$ the value $g(t + \alpha_i \delta t)$ degrades the scheme to be of first order accuracy near the boundary and of second order accuracy overall. We presented ways to deal with the linear case of general order and with systems of conservation laws in the case of third order RK integration. Much work remains to be done for the non-linear case and linear problems with variable coefficients.

$$\begin{aligned}
\vec{v}_i^{\,1} &= \vec{v}_i^{\,n} + \frac{\lambda}{3}[D\vec{f}(\vec{v}^{\,n})]_i \qquad 1 \le i \le N && (35)\\
\vec{v}_0^{\,1} &= \vec{g}(t) + \frac{\delta t}{3}\vec{g}'(t) &&\\
\vec{v}_i^{\,2} &= \vec{v}_i^{\,n} + \frac{2\lambda}{3}[D\vec{f}(\vec{v}^{\,1})]_i \qquad 1 \le i \le N && (36)\\
\vec{v}_0^{\,2} &= \vec{g}(t) + \frac{2\delta t}{3}\vec{g}'(t) + \frac{2(\delta t)^2}{9}\vec{g}''(t) &&\\
\vec{v}_i^{\,n+1} &= \frac{1}{4}\vec{v}_i^{\,n} + \frac{3}{4}\vec{v}_i^{\,1} + \frac{3}{4}\lambda[D\vec{f}(\vec{v}^{\,2})]_i \qquad 1 \le i \le N && (37)\\
\vec{v}_0^{\,n+1} &= \vec{g}(t + \delta t) &&
\end{aligned}$$

Following the notation of equations (9) - (12) we have:

$$\vec{v}_i^{\,1} = \vec{v}_i^{\,n} + \frac{\lambda}{3}[D\vec{f}(\vec{v}^{\,n})]_i + \vec{G}^0\, e_0 \tag{38}$$

$$\vec{v}_i^{\,2} = \vec{v}_i^{\,n} + \frac{2\lambda}{3}[D\vec{f}(\vec{v}^{\,1})]_i + \vec{G}^1\, e_0 \tag{39}$$

$$\vec{v}_i^{\,n+1} = \frac{1}{4}\vec{v}_i^{\,n} + \frac{3}{4}\vec{v}_i^{\,1} + \frac{3\lambda}{4}[D\vec{f}(\vec{v}^{\,2})]_i + \vec{G}^2\, e_0 \tag{40}$$

where, it is clear from equation (35) - (37) that here:

$$\vec{G}^0 = \vec{g}(t) + \frac{\delta t}{3}\vec{g}'(t) - \vec{v}_0^{\,n} - \frac{\lambda}{3}[D\vec{f}(\vec{v}^{\,n})]_0 \tag{41}$$

$$\vec{G}^1 = \vec{g}(t) + \frac{2\delta t}{3}\vec{g}'(t) + \frac{2(\delta t)^2}{9}\vec{g}''(t) - \vec{v}_0^{\,n} - \frac{2\lambda}{3}[D\vec{f}(\vec{v}^{\,1})]_0 \tag{42}$$

$$\vec{G}^2 = \vec{g}(t+\delta t) - \frac{1}{4}\vec{v}_0^{\,n} - \frac{3}{4}\vec{v}_0^{\,1} - \frac{3}{4}\lambda[D\vec{f}(\vec{v}^{\,2})]_0 \tag{43}$$

As in the linear case described by equation (18), $\vec{G}^2$ will be given by a linear combination of $\vec{G}^0$ and $\vec{G}^1$, plus terms proportional to $(\delta t)^4$. We now show from equations (41) and (42) that $\vec{G}^0 = 0$, and $\vec{G}^1 = O([\delta t]^3)$, and thus the overall accuracy of $\vec{G}^2$ is $O([\delta t]^3)$.

As in section 2, in order to obtain the truncation error we substitute $\vec{u}(x_i, t)$ for $\vec{v}_i^{\,n}$. The vector $\vec{G}^0$ immediately follows as

$$\begin{aligned}
\vec{G}^0(\vec{u}) &= \vec{g}(t) + \frac{\delta t}{3}\vec{g}'(t) - (\vec{u}(t) + \frac{\delta t}{3}\vec{f}_x(\vec{u})\,)_0 && (44)\\
&= \vec{g}(t) + \frac{\delta t}{3}\vec{g}'(t) - (\vec{u}(t) + \frac{\delta t}{3}\vec{u}_t)_0 = 0 && (45)
\end{aligned}$$

Table (I.b) shows the results of a grid refinement study using the derivative form of the boundary condition.

	(3,3-4-3,3)		$(5^2, 5^2-6-5^2, 5^2)$		(5,5,5,5,5-6-)	
Grid	CFL = 2.0 log L_2	Conv Rate	CFL = 1.4 log L_2	Conv Rate	CFL = 1.5 log L_2	Conv Rate
41	-2.394		-3.090		-2.490	
81	-3.613	4.05	-4.282	3.97	-3.767	4.24
161	-4.817	4.00	-5.486	3.99	-5.076	4.35
321	-6.019	3.99	-6.687	3.99	-6.377	4.32
641	-7.222	3.99	-7.891	4.00	-7.655	4.25
1281	-8.425	4.00	-9.099	4.01	-8.911	4.17

Table I.b: New physical boundary condition imposed as $\frac{d^3 u(0,t)}{dt^3} = g'''(t)$, for three high-order finite difference schemes.

Fourth-order temporal accuracy is recovered for all methods with this approach. In addition, the same maximum CFL was achieved in all cases, as was possible with the conventional boundary conditions. Similar results have been obtained for three stage third order and six stage fifth order RK schemes as well.

Section 3. Non-Linear Hyperbolic System

We have shown that the traditional imposition of time-dependent boundary conditions causes a degradation of the formal accuracy to first-order. We have also shown how to eliminate the problem, given the exact boundary condition and its derivatives on the boundary. We shall now show that boundary treatment similar to that resulting from the linear analysis of section 2, is also valid for third order RK schemes, even for the case of a system of non-linear conservation laws. The procedure does not obviously generalize to higher order RK integration, however. (The only technique available that does directly extend to non-linear hyperbolic systems is not imposing intermediate physical boundary conditions.)

Consider the system

$$\frac{\partial \vec{U}}{\partial t} = A(\vec{U})\,\frac{\partial \vec{U}}{\partial x} = \frac{\partial}{\partial x}\vec{f}(\vec{U}),\;\; 0 \le x \le 1; t \ge 0 \tag{33}$$

$$\vec{U}(0,t) = \vec{g}(t) \tag{34}$$

where $A(\vec{U})$ is the Jacobian of $\vec{f}(\vec{U})$ with respect to $\vec{U}$. We perform the integration using the third order RK scheme attributed to Heun. The boundary conditions at the intermediate time levels are obtained from solving the boundary equation $\frac{d^2 \vec{v}_0}{dt^2} = \vec{g}''(t)$ with Heun's method.

$$u(x,t) \;=\; \sin 2\pi(x-t),\;\; 0 \le x \le 1,\;\; t \ge 0 \tag{32}$$

In all cases the solution was advanced in time using the classical fourth-order RK scheme with boundary conditions imposed as in equations (5) - (8).

Upon grid refinement with a vanishingly small CFL, all schemes recover their theoretical accuracy of fourth-order, sixth-order and sixth-order, respectively. Table (I.a) shows that all schemes converge at a fourth-order rate for moderate resolutions, but gradually degrade to an asymptotic rate of 2.5. This trend is representative of high-order explicit or compact spatial operators advanced in time with a fourth-order RK scheme, with or without the SAT boundary procedure.

	(3,3-4-3,3)		$(5^2, 5^2-6-5^2, 5^2)$		(5,5,5,5,5-6-)	
Grid	CFL = 2.0 log L_2	Conv Rate	CFL = 1.4 log L_2	Conv Rate	CFL = 1.5 log L_2	Conv Rate
41	-2.371		-3.033		-1.537	
81	-3.570	3.98	-4.242	4.02	-2.221	2.27
161	-4.678	3.68	-5.450	4.02	-2.956	2.44
321	-5.593	3.04	-6.635	3.94	-3.704	2.48
641	-6.380	2.61	-7.711	3.57	-4.455	2.49
1281	-7.138	2.52	-8.586	2.91	-5.208	2.50

Table I.a: Conventional imposition of boundary condition $u(0,t) = g(t)$, for three high-order finite difference schemes.

Two possible remedies have been suggested to rectify the loss of accuracy for the linear problem: 1) do not impose intermediate physical boundary conditions, and 2) impose consistent intermediate boundary conditions derived from the physical boundary conditions and their derivatives. (Or alternatively, solve the related system of equations on the boundary).

Not imposing intermediate physical boundary conditions results in a fully discrete scheme which is formally fourth order accurate. (By definition a fourth order scheme in space and time will remain fourth order in the absence of boundary conditions). A problem with this remedy, however, is that the stability of the scheme is greatly reduced. When using the RK4 scheme with various finite difference operators, at least of a factor of two (and often much more) decrease in CFL was observed.

The alternative remedy is to impose consistent intermediate boundary conditions, derived from the physical boundary conditions and its derivatives. For the scalar advection defined by equations (29) - (30) it is sufficient to solve the derivative boundary conditions described by the system of equations (27).

on the boundary provides a general technique for imposing the correct boundary conditions on the linear problem.

At this point we comment on why the above predicted phenomena has not been observed previously by the practitioners of higher order methods. From equation (21) we see that the leading error coefficient in the expression is proportional to $(\lambda)^4\delta x$. For example, if D represents a fourth order central difference operator with fourth order boundary closures, and $u = e^{i(x-t)}$ then the error at the point next to the boundary becomes

$$u_t + u_x = \frac{19\lambda^4\delta x^1}{165888} + \frac{i(184320 - 1520\lambda)\lambda^4\delta x^2}{79626240} + \frac{(+172800 + 23040\lambda - 190\lambda^2)\lambda^4\delta x^3}{79626240} + \ldots \tag{28}$$

Thus if $\lambda << 1$ for a given δx it is possible that $\lambda^4\delta x \leq (\delta t)^4$. However, if one refines the grid, or conversely, if one runs the computation at the allowable stable CFL ($\lambda = O(1)$), then the first order error becomes apparent. A detailed study of this effect is now presented.

We begin by showing the the results from a grid refinement study using three high-order central difference schemes; 1) (3,3-4-3,3): a fourth order explicit interior with two third-order explicit boundary points at each end of the domain, 2) $(5^2, 5^2 - 6 - 5^2, 5^2)$[2]: a sixth-order compact interior, with two fifth-order boundary closures at each end of the domain, and 3) (5,5,5,5,5,5-6-5,5,5,5,5,5)[6]: a sixth-order explicit, with six fifth-order boundary points at each end of the domain. (See Carpenter et al[2] for detail of the high-order nomenclature). All schemes are GKS and time stable for the scalar hyperbolic equation. In addition, when used in conjunction with the SAT [3] boundary procedure, the semi-discrete form of scheme 3) can be shown to be time stable for the constant coefficient hyperbolic system. An explicit, a compact implicit, and a scheme which satisfies the summation by parts energy norm were chosen to illustrate the generality of the problem, as well as the remedy.

The model problem used to test the schemes is the scalar hyperbolic equation

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0, \;\; 0 \leq x \leq 1, \; t \geq 0 \tag{29}$$

$$u(0,t) = \sin 2\pi(-t), \;\; t \geq 0 \tag{30}$$

$$u(x,0) = \sin 2\pi(x), \;\; 0 \leq x \leq 1, \tag{31}$$

The exact solution is

Rearranging equation (20) we have

$$\frac{V^{n+1} - u(t+\delta t)}{\delta t} = \delta x\, \frac{(\lambda)^4}{96}(D^3 e_0) g'' \;+\; O([\delta x]^2) \;=\; O(\delta x) \tag{21}$$

which means that the RK approximation is locally (near the boundary) only first order accurate. The locality is assured by the finite support of D for explicit schemes, and by geometrically decaying coefficients in space for compact schemes. However, the overall accuracy in the $L_{\inf}$ norm cannot exceed second order (see Gustafsson [8]).

The remedy for the dilemma posed by equation (21) suggests itself when one examines equations (17) and (18). We note that if we set $G^0 = G^1 = G^2 = 0$, then $V^{n+1} = u(t+\delta t) + O([\delta t]^5) \;+\; e_0 G^3$. But, with $G^0 = G^1 = G^2 = 0$, then $G^3 = O([\delta t]^5)$, and we have the correct order for V^{n+1}. To achieve these identities we specifically use in equations (5)-(8) the following expressions for the intermediate boundary conditions:

$$v_0^1 = g(t) + \frac{\delta t}{2} g'(t) \tag{22}$$

$$v_0^2 = g(t) + \frac{(\delta t)}{2} g'(t) + \frac{(\delta t)^2}{4} g''(t) \tag{23}$$

$$v_0^3 = g(t) + (\delta t) g'(t) + \frac{(\delta t)^2}{2} g''(t) + \frac{(\delta t)^3}{4} g'''(t) \tag{24}$$

Note that equations (22 - 24) are precisely the intermediate boundary conditions obtained by numerically integrating $\frac{d^3 u}{dt^3} = g'''(t)$ with the classical fourth order RK scheme. Specifically, replacing the third order equation in u with the system of first order equations for the unknown functions V_0, vt_0 and vtt_0,

$$\frac{d(V)_0}{dt} = (vt)_0 \tag{25}$$

$$\frac{d(vt)_0}{dt} = (vtt)_0 \tag{26}$$

$$\frac{d(vtt)_0}{dt} = g'''(t) \tag{27}$$

on the boundary, and integrating with the standard fourth order RK scheme, yields precisely the intermediate boundary conditions proposed in equation (24). Solving the third order o.d.e (27) on the boundary, is a remedy that can be applied to any four stage fourth order RK scheme, and thus provides a simple and general means of implementing the correct intermediate boundary conditions. For three stage third order RK schemes, solving $\frac{d^2 u}{dt^2} = g''(t)$

To determine the formal accuracy of (17) we substitute the exact solution $u(x_i,t)$, (projected at the points x_i), for v_i^n. Under the assumption on the order of the differentiation matrix Q, it is clear that D^kV^n in equation (17) can be replaced, up to fourth-order accuracy, by $h^k[(-1)^k\frac{\partial^k u(x,t)}{\partial x^k}]$ at the points (x_i,t). Equation (17) then becomes

$$\begin{aligned}
V^{n+1} &= u(t\ +\ \delta t) + O([\delta t]^5,[\delta x]^5) \\
&+ G^0[\frac{\lambda}{3}D\ +\ \frac{\lambda^2}{6}D^2\ +\ \frac{\lambda^3}{12}D^3]e^0 \\
&+ G^1[\frac{\lambda}{3}D\ +\ \frac{\lambda^2}{6}D^2]e^0 \\
&+ G^2[\frac{\lambda}{6}D]e^0 \\
&+ G^3e^0
\end{aligned} \tag{18}$$

Applying the same procedures to equations (13) - (16) we get:

$$\begin{aligned}
G^0 &= g(t\ +\ \frac{\delta t}{2}) - g(t) - \frac{\delta t}{2}g'(t) \\
G^1 &= g(t\ +\ \frac{\delta t}{2}) - g(t) - \frac{(\delta t)}{2}g'(t) - \frac{(\delta t)^2}{4}g''(t) - \frac{\lambda}{2}d^1_{00}\,G^0 \\
G^2 &= g(t\ +\ \delta t) - g(t) - (\delta t)g'(t) - \frac{(\delta t)^2}{2}g''(t) - \frac{(\delta t)^3}{4}g'''(t) - \lambda d^1_{00}\,G^1 - \frac{(\lambda)^2}{2}d^2_{00}\,G^0 \\
G^3 &= g(t\ +\ \delta t) - g(t) - (\delta t)g'(t) - \frac{(\delta t)^2}{2!}g''(t) - \frac{(\delta t)^3}{3!}g'''(t) - \frac{(\delta t)^4}{4!}g''''(t) \\
&- [\frac{\lambda}{3}d^1_{00}\ +\ \frac{(\lambda)^2}{6}d^2_{00} + \frac{(\lambda)^3}{12}d^3_{00}]G^0 \\
&- [\frac{\lambda}{3}d^1_{00}\ +\ \frac{(\lambda)^2}{6}d^2_{00}]\,G^1 \\
&- \frac{\lambda}{6}d^1_{00}\,G^2
\end{aligned} \tag{19}$$

where $d^1_{00} = (De_0)_0$, $d^2_{00} = (D^2e_0)_0$, $d^3_{00} = (D^3e_0)_0$, and $(D^ke_0)_0$ is the first element of the vector D^ke_0; $1 \le k \le 3$. A Taylor series expansion of equation (19) clearly shows that $G^0 = O([\delta t]^2)$; as it does for G^1, G^2 and G^3, for arbitrary λ. In addition, it can be shown that the vectors D^ke_0 $(1 \le k \le 3)$ are linearly independent. Substituting equation (19) into equation (18) yields the expression

$$V^{n+1} - u(t+\delta t) = [\delta t]^2\frac{(\lambda)^3}{96}(D^3e_0)g'' + O([\delta t]^3) \tag{20}$$

$$
\begin{aligned}
v_0^3 &= g(t + \delta t) \\
v_i^{n+1} &= v_i^n + \frac{\lambda}{6}[(DV^n)_i + 2(DV^1)_i + 2(DV^2)_i + (DV^3)_i] \quad 1 \le i \le N \qquad (8) \\
v_0^{n+1} &= g(t + \delta t)
\end{aligned}
$$

where $\lambda = \frac{\delta t}{h}$.

Equations (5) - (8) take the semi-discrete variable $v_i(t)$, from the time level n, to $v_i(t+\delta t)$ at time level $n + 1$.

For the purpose of analysis, the above system is rewritten in the following form, again with $V = [v_0, ...v_N]^T$:

$$
\begin{aligned}
V^1 &= V^n + \frac{\lambda}{2}DV^n + G^0 e_0 \qquad (9) \\
V^2 &= V^n + \frac{\lambda}{2}DV^1 + G^1 e_0 \qquad (10) \\
V^3 &= V^n + \lambda DV^2 + G^2 e_0 \qquad (11) \\
V^{n+1} &= V^n + \frac{\lambda}{6}[DV^0 + 2DV^1 + 2DV^2 + DV^3] + G^3 e_0 \qquad (12)
\end{aligned}
$$

where $e_0 = [1, 0, ...0]^T$, and

$$
\begin{aligned}
G^0 &= g(t + \frac{\delta t}{2}) - v_0^n - \frac{\lambda}{2}(DV^n)_0 \qquad (13) \\
G^1 &= g(t + \frac{\delta t}{2}) - v_0^n - \frac{\lambda}{2}(DV^1)_0 \qquad (14) \\
G^2 &= g(t + \delta t) - v_0^n - \lambda(DV^2)_0 \qquad (15) \\
G^3 &= g(t + \delta t) - v_0^n - \frac{\lambda}{6}[DV^0 + 2DV^1 + 2DV^2 + DV^3]_0 \qquad (16)
\end{aligned}
$$

Substitution of (9) into (10) and the result into (11), etc. leads to the following expression for V^{n+1}:

$$
\begin{aligned}
V^{n+1} &= [I + \frac{\lambda}{1!}D + \frac{\lambda^2}{2!}D^2 + \frac{\lambda^3}{3!}D^3 + \frac{\lambda^4}{4!}D^4]V^n \\
&+ G^0[\frac{\lambda}{3}D + \frac{\lambda^2}{6}D^2 + \frac{\lambda^3}{12}D^3]e^0 \\
&+ G^1[\frac{\lambda}{3}D + \frac{\lambda^2}{6}D^2]e^0 \\
&+ G^2[\frac{\lambda}{6}D]e^0 \\
&+ G^3e^0 \qquad (17)
\end{aligned}
$$

2. The linear case

To illustrate the phenomenon of loss of accuracy due to the conventional imposition of inflow boundary conditions, we consider the following problem:

$$\begin{aligned} \frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} &= 0 \quad 0 \leq x \leq 1, \ t \geq 0 && (1) \\ u(0,t) &= g(t) && (2) \end{aligned}$$

The semi-discretized version of equations (1) - (2) is

$$\begin{aligned} \frac{dv_i}{dt} &= (QV(t))_i \quad i = 1, ..., N; \, t \geq 0 && (3) \\ v_0(t) &= g(t), && (4) \end{aligned}$$

where $V = {v_i}^T = [v_0, ...v_N]^T$ is the semi-discrete approximation which converges to $u(x_i, t)$ at the spatial grid points x_i (for stable discretizations); and Q is the differentiation matrix representation of the derivative operator $\frac{\partial}{\partial x}$. The specific form of Q depends on the algorithm used and in particular on the order of accuracy. For all finite difference operators on uniform grids (which suffices for the present purpose of illustration) we may write $Q = \frac{1}{h}D$, where h is the mesh spacing.

The demonstration of accuracy deterioration will be shown for the four-stage "classical" RK scheme, which is one of the most common RK time advancing schemes. For the analysis to make sense we assume that the differentiation matrix Q, is at least of fourth-order accuracy up to the boundary. It should be noted, however, that this illustration could be carried out for any RK algorithm.

The above mentioned four-stage integration, together with the conventionally imposed boundary conditions, is implemented as follows:

$$\begin{aligned} v_i^1 &= v_i^n + \frac{\lambda}{2}(DV^n)_i && 1 \leq i \leq N && (5) \\ v_0^1 &= g(t + \frac{\delta t}{2}) \\ v_i^2 &= v_i^n + \frac{\lambda}{2}(DV^1)_i && 1 \leq i \leq N && (6) \\ v_0^2 &= g(t + \frac{\delta t}{2}) \\ v_i^3 &= v_i^n + \lambda(DV^2)_i && 1 \leq i \leq N && (7) \end{aligned}$$

Introduction

The recent interest in long time integration is due to the need to tackle problems in areas such as aero-acoustics, electro-magnetics and others. This in turn necessitates working with higher order accurate spatial differencing operators. In many cases the time-stepping algorithm of choice is a multi-stage Runge-Kutta (RK) of temporal order of accuracy comparable to the spatial one.

Several bothersome issues arise when using RK methods for long time integration. A principle concern is the imposition of numerical boundary conditions which retain the formal accuracy of the numerical method and guarantee numerical stability of the solution. For example (see Trefethen [1], or Carpenter et. al [2], [3]) Lax and GKS stability are not sufficient to assure that the there is no exponentially growing temporal error when using realistic grids. To alleviate this "anomaly" the use of spatial operators which have specific semi-discrete energy norms has been proposed [4], [5], [6], [7]. These papers have primarily focused on the semi-discrete form of the equations.

The present paper, however, deals with the loss of accuracy due to the imposition of time dependent boundary conditions $g(t)$, dictated by the physics of the problem. The conventional way of dealing with the uncertainty of what is happening at the intermediate stages of the RK time advancement is to impose at $t+\alpha_\nu \delta t$, the boundary value $g(t+\alpha_\nu \delta t)$, where α_ν is the coefficient appropriate to the particular ν^{th} stage of the given RK algorithm.

It will be shown that this conventional boundary condition imposition leads to a numerical scheme which is only first order accurate in the neighborhood of the boundary, *leading to a global accuracy of second order only.* Another approach is to treat the time-dependent boundary condition, $g(t)$, as a source term in the governing partial differential equation (p.d.e), thereby avoiding the need to $\underline{formally\ specify\ intermediate}$ boundary conditions. However, it can be shown that procedure is equivalent to the conventional method with its attended problems (see Appendix A). A third natural procedure is indeed $\underline{not\ to\ specify}$ any intermediate boundary condition, but to obtain them from the inner scheme. This method retains the accuracy of the spatial operator, but significantly reduces the allowable time step for stability, rendering the scheme less attractive.

In section 2) we analyze and pinpoint the reasons for the deterioration of the accuracy and provide a simple recipe' for restoring the accuracy in the case of linear, constant coefficient hyperbolic system of p.d.e.'s.

In section 3) we deal with a non-linear hyperbolic system of conservation laws. The remedies provided in Sections 2) and 3) can not be generalized to non-linear systems integrated by RK schemes of arbitrary order of accuracy. We show that for the RK of third order, the remedy of section 2) is effective.

THE THEORETICAL ACCURACY OF RUNGE-KUTTA TIME DISCRETIZATIONS FOR THE INITIAL BOUNDARY VALUE PROBLEM: A CAREFUL STUDY OF THE BOUNDARY ERROR[1]

Mark H. Carpenter
Theoretical Flow Physics Branch
Fluid Mechanics Division
NASA Langley Research Center
Hampton, VA. 23681-0001

David Gottlieb[1]
Division of Applied Mathematics
Brown University
Providence, RI 02912

Saul Abarbanel[1]
Department of Mathematical Sciences
Division of Applied Mathematics
Tel-Aviv University
Tel-Aviv, ISRAEL

Wai-Sun Don[1]
Division of Applied Mathematics
Brown University
Providence, RI 02912

ABSTRACT

The conventional method of imposing time dependent boundary conditions for Runge-Kutta (RK) time advancement reduces the formal accuracy of the space-time method to first order locally, and second order globally, independently of the spatial operator. This counter intuitive result is analyzed in this paper.

Two methods of eliminating this problem are proposed for the linear constant coefficient case: 1) impose the exact boundary condition only at the end of the complete RK cycle, 2) impose consistent intermediate boundary conditions derived from the physical boundary condition and its derivatives. The first method, while retaining the RK accuracy in *all cases*, results in a scheme with much reduced CFL condition, rendering the RK scheme less attractive. The second method retains the same allowable time step as the periodic problem. However it is a general remedy only for the linear case. For non-linear hyperbolic equations the second method is effective only for for RK schemes of third order accuracy or less. Numerical studies are presented to verify the efficacy of each approach.

[1]This research was supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1-19480 while the authors were in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA 23681.