Numerical Experiments in Error Control for Sound Propagation Using a Damping Layer Boundary Treatment

John W. Goodrich*

NASA Glenn Research Center, Cleveland, Ohio, 44135, USA

This paper presents results from numerical experiments for controlling the error caused by a damping layer boundary treatment when simulating the propagation of an acoustic signal from a continuous pressure source. The computations are with the 2D Linearized Euler Equations (LEE) for both a uniform mean flow and a steady parallel jet. The numerical experiments are with algorithms that are third, fifth, seventh and ninth order accurate in space and time. The numerical domain is enclosed in a damping layer boundary treatment. The damping is implemented in a time accurate manner, with simple polynomial damping profiles of second, fourth, sixth and eighth power. At the outer boundaries of the damping layer the propagating solution is uniformly set to zero. The complete boundary treatment is remarkably simple and intrinsically independent from the dimension of the spatial domain. The reported results show the relative effect on the error from the boundary treatment by varying the damping layer width, damping profile power, damping amplitude, propagtion time, grid resolution and algorithm order. The issue that is being addressed is not the accuracy of the numerical solution when compared to a mathematical solution, but the effect of the complete boundary treatment on the numerical solution, and to what degree the error in the numerical solution from the complete boundary treatment can be controlled. We report maximum relative absolute errors from just the boundary treatment that range from $O[10^{-2}]$ to $O[10^{-7}]$.

I. Introduction

Aeroacoustics research has benefited from improved numerical methods and more powerful computers, so that computation is increasingly valuable as a source of insight and as a design tool.[3,19,24,30] Even with the best methods and computers, numerical simulations must have a finite computational domain. Closing and limiting a numerical domain with a boundary treatment is a critical element for Computational Aeroacoustics (CAA). A desirable boundary treatment limits the error in the numerical solution while using an affordable computational domain. Reviews of the ongoing work on this challenge include.[2,7,13,16,17,25,29] Artificial boundary treatments that use a bounding zone that surrounds a computational domain include Perfectly Matched Layers[4,14,18,21] and a Sponge Zone or Damping Layer.[1,20,22,23] Examples of widely used Artificial Boundary Conditions that do not require layers or zones are Hedstrom,[15] Thompson[26–28] and Giles.[5,6]

In this paper we present results from numerical experiments for propagating a continuous acoustic pressure signal in 2D with a uniform flow or with a steady parallel jet. Acoustic propagation is by means of the Linearized Euler Equations (LEE) in Cartesian coordinates. These computations use variations of Hermite/Cauchy-Kowaleskya/Taylor algorithms[8–12] to propagate the solution variables in the numerical domain and the damping layer. The algorithms in this paper use only data for the primitive variables, so

*Research Engineer, Acoustics Branch.

they are not strictly Hermite methods which use data for derivatives, and are more precisely Centered-Staggered/Cauchy-Kowaleskya/Taylor algorithms. We present results for the c4o0, c6o0, c8o0 and c10o0 methods on staggered two dimensional stencils that are four, six, eight and ten points wide, and that are third, fifth, seventh and ninth order accurate in both space and time, respectively. The results for propagating an acoustic signal with a steady parallel jet flow are only from the seventh order c8o0 algorithm. The boundary treatment has a damping layer around the numerical domain, and damping is implemented with the full accuracy of the time propagation by including damping terms in the governing equations which are used for the Cauchy-Kowaleskya recursion. The combination of the interior numerical domain and the surrounding damping layer is closed by setting all propagating variables to zero on the outer boundary of the damping layer.

Our concern here is with the effect on the accuracy of the numerical solution of the complete boundary treatment with both damping layers and outer boundary conditions. We present results of numerical experiments in order to validate this complete boundary treatment when used with Centered-Staggered/Cauchy-Kowaleskya/Taylor algorithms. The results reported here are comparable to what has been reported in[11] for the same acoustic simulation with similar grids, but with a third order Hermite/Cauchy-Kowaleskya/Taylor algorithm and a complex characteristic outer boundary condition. A particular question that arose from[11] is whether there is any reason to use a damping polynomial other than a quadratic. The numerical experiments that are reported here use damping profile polynomials of 2^{nd}, 4^{th}, 6^{th} and 8^{th} order, and the results show that any of the three higher order profiles will produce lower errors from the boundary treatment than the quadratic profile. Which of these three higher powers is optimal depends somewhat upon the algorithm, or its order, the damping layer width and the simulation time. In addition, the complexity of characteristic based outer boundary conditions has raised the question of whether a simpler boundary treatment could provide similar levels of accuracy. The results that we report here show that the essentially trivial outer boundary closer of the damping layer that we use actually provides comparable accuracy to the complex characteristic outer boundary condition as reported in.[11]

II. Governing Equations

We use Cartesian coordinates (x, y) in two space dimensions. As primitive variables, we use the specific volume $\sigma = 1/\rho$, velocity, and pressure, or

$$\vec{V} = (\sigma, u, v, p)^T. \tag{1}$$

We have nondimensionalized the governing compressible Euler Equations with the reference speed of sound a_R, density ρ_R, and length L_R. For consistency, the reference specific volume is $\sigma_R = 1/\rho_R$, and the reference time is $t_R = L_R/a_R$. We assume a perfect gas, and have taken the reference pressure as

$$p_R = a_R^2 \rho_R = \frac{a_R^2}{\sigma_R}, \tag{2}$$

so that for both the dimensional and nondimensional variables

$$a^2 = \gamma \frac{p}{\rho} = \gamma p \sigma. \tag{3}$$

As a consequence of this choice of reference quantities, the form of the compressible Euler equations is the same for both the dimensional and nondimensional variables. We assume that the ratio of specific heats $\gamma = 1.4$, as in a standard atmosphere.

The compressible Euler equations in two space dimensions linearized about the steady base flow solution $\vec{V_b} = (\sigma_b, u_b, v_b, p_b)^T$ are:

$$\frac{\partial \sigma}{\partial t} + u_b \frac{\partial \sigma}{\partial x} + v_b \frac{\partial \sigma}{\partial y} - \sigma_b \frac{\partial u}{\partial x} - \sigma_b \frac{\partial v}{\partial y} + u \frac{\partial \sigma_b}{\partial x} + v \frac{\partial \sigma_b}{\partial y} - \sigma \frac{\partial u_b}{\partial x} - \sigma \frac{\partial v_b}{\partial y} = 0, \tag{4}$$

$$\frac{\partial u}{\partial t} + u_b \frac{\partial u}{\partial x} + v_b \frac{\partial u}{\partial y} + \sigma_b \frac{\partial p}{\partial x} + u \frac{\partial u_b}{\partial x} + v \frac{\partial u_b}{\partial y} + \sigma \frac{\partial p_b}{\partial x} = 0, \tag{5}$$

$$\frac{\partial v}{\partial t} + u_b \frac{\partial v}{\partial x} + v_b \frac{\partial v}{\partial y} + \sigma_b \frac{\partial p}{\partial y} + u \frac{\partial v_b}{\partial x} + v \frac{\partial v_b}{\partial y} + \sigma \frac{\partial p_b}{\partial y} = 0, \tag{6}$$

$$\frac{\partial p}{\partial t} + u_b \frac{\partial p}{\partial x} + v_b \frac{\partial p}{\partial y} + \gamma p_b \frac{\partial u}{\partial x} + \gamma p_b \frac{\partial v}{\partial y} + u \frac{\partial p_b}{\partial x} + v \frac{\partial p_b}{\partial y} + \gamma p \frac{\partial u_b}{\partial x} + \gamma p \frac{\partial v_b}{\partial y} = p_S(x, y, t), \tag{7}$$

where we have included an unsteady pressure source p_S. These equations can be conveniently written as the succinct vector equation

$$\frac{\partial \vec{V}}{\partial t} + A_b \frac{\partial \vec{V}}{\partial x} + B_b \frac{\partial \vec{V}}{\partial y} + A \frac{\partial \vec{V_b}}{\partial x} + B \frac{\partial \vec{V_b}}{\partial y} = \vec{S}, \tag{8}$$

where the terms of the coefficient matrices A_b and B_b are from the steady base flow $\vec{V_b}$, while the terms of the coefficient matrices A and B are from the propagating acoustic signal $\vec{V}$, and where $\vec{S} = (0, 0, 0, p_S)^T$ is the source vector. The only actual constraint on $\vec{V_b}$ is that it is a solution to Euler's equations, and it can be constant, steady and dependent upon x and or y alone, or unsteady and dependent upon (x, y, t).

III. Propagation Method

We propagate the solution variables in the numerical domain and the damping layer with Centered-Staggered/Cauchy-Kowaleskya/Taylor algorithms for the LEE that are variations of the Hermite/Cauchy-Kowaleskya/Taylor family of algorithms.[8–12] The Hermite methods use the primitive variables and some of their derivatives at each grid point, but the algorithms for this paper use only the primitive variables. The propagation algorithms use multidimensional spatial interpolation of the primitive variables at the cell centers of two staggered grid systems, local Cauchy-Kowaleskya recursion for computation of time derivatives, and Taylor series time advancement. We use a uniform spatial mesh with $\Delta x = \Delta y$ on each of the two staggered grids. The first or base grid is used for the initial data and the solution at the end of each full time step. The points of the second or staggered grid are for an intermediate solution at each half time step, and are at the centers of the cells from the first grid. The spatial interpolation is by dimensional recursion using all the data on a square stencil, effectively computing a local fully multidimensional data surface over the stencil. If the stencil width is n, then the square two dimensional stencil has n^2 data points, all of the spatial derivatives $\{(\partial_x^a, \partial_y^b) : 0 \le a, b \le n-1\}$ are computed by the interpolant, and the algorithm is order $n-1$ in space, even though the computed $(\partial_x^{n-1}, \partial_y^{n-1})$ derivative is of order $2(n-1)$. The Cauchy-Kowaleskya recursion uses the governing equations to recursively compute time derivatives from the interpolated space derivatives, with recursion on the order of the time derivative to as high an order as desired, where

$$\frac{\partial^{a+b+m+1} \vec{V}}{\partial x^a \partial y^b \partial t^{m+1}} = -\frac{\partial^{a+b+m+1} (A_b \vec{V} + A \vec{V_b})}{\partial x^{a+1} \partial y^b \partial t^m} - \frac{\partial^{a+b+m+1} (B_b \vec{V} + B \vec{V_b})}{\partial x^a \partial y^{b+1} \partial t^m} + \frac{\partial^{a+b+m} \vec{S}}{\partial x^a \partial y^b \partial t^m}, \tag{9}$$

and where A_b, B_b and $\vec{V_b}$ can depend upon x, y and t. In the codes for the simulations reported here, the base flows are steady and do not depend upon t. If the Cauchy-Kowaleskya recursion is carried out to order m in time, then the algorithm uses a Taylor series of order m for time advancement. We typically take $m = n - 1$, so that the algorithms are the same order in time as in space. A complete time step for the algorithm interpolates data on the cells of the first grid in order to compute an intermediate half time step solution at the points of the second grid, then interpolates data on the cells of the second grid in order to finish the whole time step by computing a new solution back on the first grid. We present results for the c4o0, c6o0, c8o0 and c10o0 methods, on staggered two dimensional stencils that are $n = 4$, 6, 8 and 10 points squared, and that are 3^{rd}, 5^{th}, 7^{th} and 9^{th} order accurate in both space and time, respectively. Note that the interpolant simultaneously calculates all of the spatial derivatives for each of the variables, so that

instead of separately and disparately calculating each of the spatial derivatives in the differential equation, the interpolant in effect computes a local initial data surface on each grid cell. The Cauchy-Kowaleskya recursion uses the full system of governing equations, so that the Taylor series time advance can be viewed as correctly evolving all of the local data surfaces locally in time, up to the algorithm order in time. This numerical process is essentially consistent with the differential equation, and up to the order of the recursion and time advancement, it inherently reproduces the fully multidimensional local propagation dynamics of the underlying system of partial differential equations. Note that for a constant coefficient system with a uniform mean flow, exact polynomial solutions are possible. In this case, if we take the order of time recursion to be $m = 2(n-1)$, then the algorithm in effect computes just such a local exact polynomial solution over each grid cell for each half time step, and the local multidimensional propagation would be exact. The complete algorithm is used in exactly the same way and performs similarly for computations in one, two and three space dimensions. Notice that since the algorithm orders are odd, the leading order error terms are diffusive.

IV. Damping Layers

We are interested in the accuracy of the propagated acoustic signal in the numerical domain

$$\Omega_N = [-x_L, x_R] \times [-y_B, y_T], \tag{10}$$

with the unsteady pressure source p_S centered at the origin in this domain. The numerical domain Ω_N is surrounded by damping layers of width w_R, w_L and w_Y on the right, the left and the top and bottom, respectively. The results for this paper all assume $w_R = w_L = w_Y$. The complete computational domain is

$$\Omega_C = [-(x_L + w_L), x_R + w_R] \times [-(y_B + w_Y), y_T + w_Y], \tag{11}$$

with the numerical domain $\Omega_N \subset \Omega_C$ as the inner core of interest where accuracy is desired. The damping domain $\Omega_D = \Omega_C \setminus \Omega_N$ is that part of the complete domain on all sides of the numerical domain where damping is done, and our concern with the damping domain is for how it effects the solution in the numerical domain Ω_N. Damping is done with N^{th} order polynomial damping profiles in x,

$$\begin{aligned} D_x(x) &= \delta_R((x - x_R)/w_R)^N, \quad \text{for } +x_R < x \le +x_R + w_R, \\ &= 0, \qquad\qquad\qquad\quad \text{for } -x_L \le x \le +x_R, \\ &= \delta_L((x - x_L)/w_L)^N, \quad \text{for } -(x_L + w_L) \le x < -x_L, \end{aligned} \tag{12}$$

and in y,

$$\begin{aligned} D_y(y) &= \delta_T((y - y_T)/w_Y)^N, \quad \text{for } +y_T < y \le +y_T + w_Y, \\ &= 0, \qquad\qquad\qquad\quad \text{for } -y_B \le y \le +y_T, \\ &= \delta_B((y - y_B)/w_Y)^N, \quad \text{for } -(y_B + w_Y) \le y < -y_B. \end{aligned} \tag{13}$$

In this paper we consider polynomials of order $N = 2$, 4, 6 and 8, and for simplicity we take

$$\delta_R = \delta_L = \delta_T = \delta_B = \delta. \tag{14}$$

Damping is done by modifying the governing equations to include the damping terms. The modified equations that are actually used for the Cauchy-Kowaleskya recursion throughout the computational domain $\Omega_C = \Omega_N \cup \Omega_D$ are

$$\frac{\partial \vec{V}}{\partial t} + A_b \frac{\partial \vec{V}}{\partial x} + B_b \frac{\partial \vec{V}}{\partial y} + A \frac{\partial \vec{V}_b}{\partial x} + B \frac{\partial \vec{V}_b}{\partial y} + (D_x(x) + D_y(y))\vec{V}(x, y, t) = \vec{S}. \tag{15}$$

Note that the damping terms are all zero in the numerical domain, and that the source terms are negligible in the damping domain. The damping profiles are both nonzero in the four corners where the damping layers

in x and y overlap, so that the damping is additive in the corners. The damping functions are differentiated in the Cauchy-Kowaleskya recursion process, so that they are differentially applied to all of the various terms computed in the recursion, and the damping is implemented with the time accuracy of the propagation.

The outer boundary conditions that close the damping layers are just

$$\vec{V}|_{\partial\Omega_C} = (\sigma, u, v, p)^T|_{\partial\Omega_C} = 0, \tag{16}$$

uniformly in time on all four outer boundaries. Since these boundary conditions are uniform in time on the outer boundaries, there is no computation of time evolution on the outer boundary. The propagating solution on the grid points next to the boundary does vary in time, so that the time evolution algorithm must be applied. There are two choices for implementation at these points, either interior biased interpolation can be used with one layer of boundary points, or centered interpolation can be used next to the boundary with enough layers of boundary points or "ghost points" to complete the stencil, depending upon the stencil width required for the interpolant order. For the sake of simplicity, we have chosen to add sufficient layers of boundary points so that there is no need for special interpolants near the outer boundary, and we can use the same interpolant everywhere in the combined numerical and damping domains. Note that each layer of boundary points where $\vec{V} = 0$ is equivalent to a boundary condition. By using one set of governing equations and one interpolant everywhere throughout the combined numerical and damping domain, the code structure is quite simple and very flexible. This code structure permits different interpolants to be swapped in and out to change the spatial order, and a simple change in the order of recursion and time advance easily changes the order of the algorithm as a whole. In addition, a simple replacement of the Cauchy-Kowaleskya recursion subroutine can adapt the algorithm to a different set of governing equations. The resulting code simplicity is particularly striking when compared with codes for characteristic based outer boundary treatments, such as were used in.[11] Characteristic based boundary conditions must be adapted to both the governing equations and the local flow conditions. The 2D characteristic based outer boundry conditions used in[11] require different boundary conditions on each outer boundary edge, plus combinations of two sets of boundary conditions in each corner, plus different interior biased interpolants along each edge and in each corner. In three space dimensions, characteristic based outer boundary conditions that are location dependent become significantly more complex. The results in[11] also used a Hermite algorithm that requires derivative data for all of the solution variables, with a concomitant requirement for additional boundary conditions.

V. The Computational Problem

The numerical experiments that we consider are either with the uniform 2D flow

$$\vec{V_b} = (\sigma, u, v, p)_b^T = \vec{V_1} = (1, 0.4, 0, 1/\gamma), \tag{17}$$

or with the steady parallel 2D jet

$$\vec{V_b} = (\sigma, u, v, p)_b^T = \vec{V_2} = (1, 0.4 + 0.4\exp[-25y^2], 0, 1/\gamma), \tag{18}$$

in either case nondimensionalized by the reference speed of sound a_R and specific volume $\sigma_R = 1/\rho_R$, with reference pressure $p_R = a_R^2/\sigma_R^2$. The initial state of the propagating variables is

$$\vec{V}(x, y, 0) = \vec{V_0} = (\sigma, u, v, p)_0^T = (0, 0, 0, 0)^T, \tag{19}$$

and the impulsively started pressure source is

$$\vec{S} = (0, 0, 0, p_S)^T = (0, 0, 0, 0.01\sin[2\pi t]\exp[-25(x^2 + y^2])^T. \tag{20}$$

The scale of this pressure source produces propagating pressure wave maximum values that are $O[10^{-4}]$. The numerical domain is normalized by the wavelength of the source, which is taken as the product of the

speed of sound of the base flow times the period of the signal source, and is

$$\Omega_N = [-x_L, x_R,] \times [-y_B, y_T] = \Omega_1 = [-3, 7] \times [-5, 5], \tag{21}$$

for the uniform flow with $\vec{V}_b = \vec{V}_1$, and

$$\Omega_N = [-x_L, x_R,] \times [-y_B, y_T] = \Omega_2 = [-3, 9] \times [-5, 5], \tag{22}$$

for the steady jet with $\vec{V}_b = \vec{V}_2$. Ω_N is the domain in which accuracy is desired. The calculations with the uniform flow use damping layer widths

$$w_R = w_L = w_T = w_B = w = 5, \quad \text{or} \quad w = 10, \tag{23}$$

and their computational domains are

$$\Omega_C = \Omega_N \cup \Omega_D = [-8, 12] \times [-10, 10], \quad \text{with} \quad w = 5, \tag{24}$$

or

$$\Omega_C = \Omega_N \cup \Omega_D = [-13, 17] \times [-15, 15], \quad \text{with} \quad w = 10. \tag{25}$$

The calculations with the parallel jet use damping layer width

$$w_R = w_L = w_T = w_B = w = 5, \tag{26}$$

with the computational domain

$$\Omega_C = \Omega_N \cup \Omega_D = [-8, 14] \times [-10, 10]. \tag{27}$$

The damping domain $\Omega_D = \Omega_C \setminus \Omega_N$ is part of the boundary treatment, and solution accuracy is not sought in it. The solution time is normalized by the source period, and we compute out to the final simulation times

$$T = 25, \quad 50, \quad 100, \quad 200, \quad \text{or} \quad 300. \tag{28}$$

Spatial grid resolution is generally $\Delta x = \Delta y = 1/h = 1/6$ with corresponding temporal resolutions $\Delta t = 1/k = 1/18$. The simulations with the third order c4o0 algorithm use $\Delta x = \Delta y = 1/h = 1/8$ and $\Delta t = 1/k = 1/24$. We use the seventh order c8o0 algorithm with $\Delta x = \Delta y = 1/h = 1/12$ and $\Delta t = 1/k = 1/36$ for comparing results for the uniform flow and the parallel jet. These grid resolutions fall well within the standard CFL constraint for explicit methods, which these methods satisfy, and they have been chosen so that the same grid resolution can be used for computations with mean flows that have faster peak velocities. The powers for the damping profile are $N = 2$, 4, 6 or 8, and the damping amplitude ranges from $\delta = 0$ to $\delta = 50$, simultaneously for both damping polynomials in either x or y.

Table 1. Size of the comparison solution domain Ω_C^* for $\vec{V}_b = \vec{V}_1$.

T	$w = 5$	$w = 10$
50	$[-24, 30] \times [-31, 31]$	$[-25, 33] \times [-36, 36]$
100	$[-45, 51] \times [-56, 56]$	$[-46, 54] \times [-58, 58]$
200	$[-87, 93] \times [-106, 106]$	$[-88, 96] \times [-108, 108]$
300	$[-129, 135] \times [-156, 156]$	$[-130, 138] \times [-158, 158]$

The issue that is being addressed is not the accuracy of the numerical solution when compared to a mathematical solution, which would be improved by grid refinement, but the effect of the boundary treatment

on the numerical solution, and to what degree the error from the boundary treatment can be controlled. In order to obtain a comparison solution $\vec{V}^*$ for computing errors from just the boundary treatment, we compute the comparison solution with the same code on a larger numerical domain Ω_N^*, with

$$\Omega_N^* = \Omega_C = \Omega_N \cup \Omega_D. \tag{29}$$

The computational domain for computing the comparison solution is

$$\Omega_C^* = \Omega_N^* \cup \Omega_D^* = (\Omega_N \cup \Omega_D) \cup \Omega_D^*, \tag{30}$$

where there is no damping in the auxiliary damping layer Ω_D^*, which is chosen large enough so that the comparison solution $\vec{V}^*$ is not disturbed on Ω_N^* in any way by any aspect of the boundary treatment. For the simulations with the uniform flow $\vec{V}_b = \vec{V}_1$, the sizes of the comparison solution computational domains for damping layers Ω_D with widths $w = 5$ and $w = 10$ are given in Table 1 for each of the simulation times. For the simulations with the parallel jet $\vec{V}_b = \vec{V}_2$, the sizes of the comparison solution computational domains are slightly larger on the right or downstream side. In either case, when we report an error, it is entirely due to the boundary treatment, since the numerical errors in Ω_N that are solely from the propagation method will have already been incorporated into the comparison solution $\vec{V}^*$, which is uncontaminated in Ω_N by any boundary treatment effect. The reported errors are the maximum absolute error relative to the maximum absolute solution

$$E_{R,\infty} = \frac{\max\{|\vec{V}(\vec{X}) - \vec{V}^*(\vec{X})| : \vec{X} \text{ in } \Omega_N\}}{\max\{|\vec{V}(\vec{X})| : \vec{X} \text{ in } \Omega_N\}} = \frac{\|\vec{V} - \vec{V}^*\|_{\Omega_N,\infty}}{\|\vec{V}\|_{\Omega_N,\infty}}, \tag{31}$$

where both maximums are obtained at the final simulation time T from data on the grid for the numerical domain Ω_N.

VI. Errors With No Explicit Damping

The computations discussed in this section are for the uniform base flow $\vec{V}_b = \vec{V}_1$. If $\delta = 0$, then there is no explicit damping, and any boundary treatment error is from the zero boundary condition on the outside of the damping domain as propagated with the implicit damping provided by the diffusive errors of the algorithm. Table 2 presents the error data $E_{R,\infty}$ for each of the four algorithms from simulations with the uniform base flow $\vec{V}_b = \vec{V}_1$, damping amplitude $\delta = 0$ and damping width $w = 5$, for simulation times $T = 25$, 50, 100, 200 and 300. Table 3 presents the data for each algorithm from simulations with the uniform base flow $\vec{V}_b = \vec{V}_1$, no damping and damping layer width $w = 10$, for simulation times $T = 50$, 100, 200 and 300. The simulations for the c4o0 algorithm are with $\Delta x = \Delta y = 1/8$ and $\Delta t = 1/24$, while the simulations for the other three algorithms are with $\Delta x = \Delta y = 1/6$ and $\Delta t = 1/18$. The slightly greater grid density for the c4o0 algorithm is because it is only third order. All of the errors are caused by uniformly setting the solution to $\vec{V} = 0$ on the outer boundary of the damping layer. This error from the outer boundary is propagated from all sides back and forth across the computational domain, and in particular, into the numerical domain Ω_N where $E_{R,\infty}$ is computed at the final simulation time T.

Note that the relative absolute error $E_{R,\infty}$ in both tables increases as the algorithm order is increased from the 3^{rd} order c4o0 algorithm to the 9^{th} order c10o0 algorithm, for each simulation time T, and with each damping zone width w. This is due to the greater diffusivity of the lower order algorithms, which is the other side of the greater resolution of the higher order algorithms. As the error from the outer boundary echoes across the domain it will be damped by the algorithmic diffusivity, so greater algorithmic diffusivity means lower relative error. It is quite remarkable that at this grid density and for $T \geq 50$, the 3^{rd} order c4o0 algorithm with no explicit damping has errors $E_{R,\infty} = O[10^{-5}]$ to $O[10^{-7}]$. On the other hand, the relatively low order of the c4o0 algorithm and the coarse resolution means that the solutions are poorly resolved in these simulations, so that while the relative error is low with this grid, the absolute error can be expected to be high. If the grid density were to be increased for this algorithm, so that the propagated solution were

Table 2. Maximum relative error $E_{R,\infty}$ with $w = 5$ and no damping.

T	$c4o0$	$c6o0$	$c8o0$	$c10o0$
25	1.4930D-03	3.1436D-03	8.9365D-03	1.8767D-02
50	6.6326D-05	2.7791D-04	7.3701D-03	2.4321D-02
100	3.1044D-05	2.6507D-04	7.3527D-03	2.4521D-02
200	2.3663D-05	2.2449D-04	7.2923D-03	2.4517D-02
300	2.2813D-05	2.2534D-04	7.2911D-03	2.4516D-02

Table 3. Maximum relative error $E_{R,\infty}$ with $w = 10$ and no damping.

T	$c4o0$	$c6o0$	$c8o0$	$c10o0$
50	5.2331D-04	9.1656D-04	2.3565D-03	1.0331D-02
100	2.6468D-05	4.4113D-05	1.9533D-03	1.0577D-02
200	1.9987D-06	1.2023D-05	1.9407D-03	1.0589D-02
300	7.9275D-07	1.0818D-05	1.9408D-03	******

more accurately resolved, then the reflected error would also be more accurately resolved, and the resulting relative error $E_{R,\infty}$ would increase. The errors for the 9^{th} order c10o0 algorithm are all $O[10^{-2}]$, since the solution and the reflected error are both more accurately propagated by that algorithm at this grid density, with less dissipation from algorithmic damping. In general, if the solution is more accurately computed, either with a higher powered algorithm or with a greater grid density, then the outer boundary error will also be propagated with less error, and $E_{R,\infty}$ will be greater without explicit damping.

Note also that for each algorithm, and for each damping layer width w, the observed errors $E_{R,\infty}$ generally decrease with simulation time T, contrary to the ordinary pattern of error accumulation with simulation time. Recall that the pressure source p_S is impulsively started, so a steep leading wave front is created and propagated. This impulsively started wave front can be expected to create a relatively large error when reflected by the outer boundary condition $\vec{V}|_{\partial\Omega_C} = 0$. These initial errors, like all errors, are damped by algorithmic diffusivity, and are convected downstream and out of the numerical domain by the mean velocity field. These simulations can be viewed as dissipitive dynamical systems with a periodic driving force, so that a transient from an impulsive start will decay, with overall convergence to a periodic solution. As the initial transient and its reverberations decay in these simulations, the remaining error can be attributed solely to the outer boundary treatment, and could be expected to decrease. Notice in both tables that the errors decrease more slowly with simulation time, and actually increase slightly in a couple of cases from $T = 200$ to $T = 300$. With $w = 5$, the initial error transients appear to have substantially decayed by $T = 100$ for the c4o0 algorithm, and by $T = 50$ for the other three. With $w = 10$, the initial error transients for the c4o0 and c6o0 algorithms have decayed by $T = 200$ or $T = 300$, and by $T = 100$ for the c8o0 and c10o0 algorithms. With no explicit damping in the damping layer, the lower order algorithms with greater dissipativity will end up with lower relative errors, and therefore more simulation time would be needed for the decay of the initial error transient to the levels of the persisting errors from the outer boundary.

VII. Damping With A Uniform Flow

Data for computations with explicit damping to simulation times $T = 100$ with $w = 5$ is reported in Tables 4, 6, 8 and 10 for the c4o0, c6o0, c8o0 and c10o0 algorithms, respectively. Data with explicit damping

Table 4. Maximum Relative Error $E_{R,\infty}$ at $T = 100$ for $w = 5$ using c4o0.

δ	$N = 2$	$N = 4$	$N = 6$	$N = 8$
0.0	3.1044D-05	3.1044D-05	3.1044D-05	3.1044D-05
1	2.1854D-05	9.8552D-06	1.2769D-05	1.5717D-05
2	3.5519D-05	6.4663D-06	7.4203D-06	9.0186D-06
3	5.0448D-05	6.5299D-06	6.5877D-06	6.4306D-06
4	6.5526D-05	6.6041D-06	6.7090D-06	6.2108D-06
5	8.0741D-05	6.6785D-06	6.5709D-06	6.3990D-06
6	9.6105D-05	6.6963D-06	6.6822D-06	6.7506D-06
7	1.1161D-04	6.6822D-06	6.8064D-06	7.0471D-06
8	1.2726D-04	6.6889D-06	6.8462D-06	7.2336D-06
9	1.4304D-04	6.6960D-06	6.7981D-06	7.3070D-06
10	1.5895D-04	6.6856D-06	6.6977D-06	7.2933D-06
20	3.2627D-04	6.9890D-06	6.9380D-06	7.3876D-06
30	5.2126D-04	7.5286D-06	6.8520D-06	6.9523D-06
40	7.6793D-04	8.3641D-06	6.9546D-06	6.9281D-06
50	1.0623D-03	9.0760D-06	7.0151D-06	7.1633D-06

Table 5. Maximum Relative Error $E_{R,\infty}$ at $T = 300$ for $w = 5$ using c4o0.

δ	$N = 2$	$N = 4$	$N = 6$	$N = 8$
0	2.2813D-05	2.2813D-05	2.2813D-05	2.2813D-05
1	1.6771D-05	4.9506D-06	7.8680D-06	1.0240D-05
2	3.0437D-05	1.3319D-06	2.7185D-06	4.3412D-06
3	4.5394D-05	8.9120D-07	1.1890D-06	1.9081D-06
4	6.0522D-05	9.0588D-07	8.1313D-07	1.0823D-06
5	7.5769D-05	9.5816D-07	7.2542D-07	8.2815D-07
6	9.1136D-05	1.0146D-06	7.2504D-07	7.4933D-07
7	1.0663D-04	1.0715D-06	7.2492D-07	7.2539D-07
8	1.2224D-04	1.1285D-06	7.2492D-07	7.2503D-07
9	1.3799D-04	1.1853D-06	7.2494D-07	7.2492D-07
10	1.5387D-04	1.2418D-06	7.2493D-07	7.2515D-07
20	3.2114D-04	1.8102D-06	7.2952D-07	9.1251D-07
30	5.1616D-04	2.3235D-06	9.2508D-07	1.0485D-06
40	7.6285D-04	3.3041D-06	1.0299D-06	1.1391D-06
50	1.0573D-03	2.9688D-06	1.1368D-06	1.6830D-06

Table 6. Maximum Relative Error $E_{R,\infty}$ at $T = 100$ for $w = 5$ using c6o0.

δ	$N = 2$	$N = 4$	$N = 6$	$N = 8$
0	2.6507D-04	2.6507D-04	2.6507D-04	2.6507D-04
1	3.1538D-05	3.9657D-05	6.4839D-05	9.0944D-05
2	4.8068D-05	1.1514D-05	2.2698D-05	3.6056D-05
3	6.9087D-05	7.1338D-06	1.0735D-05	1.6319D-05
4	9.0537D-05	6.5983D-06	7.4464D-06	9.8990D-06
5	1.1220D-04	6.6778D-06	6.4781D-06	7.8039D-06
6	1.3408D-04	6.7723D-06	6.6277D-06	7.1990D-06
7	1.5617D-04	6.8238D-06	6.8227D-06	7.0549D-06
8	1.7845D-04	6.8574D-06	6.8537D-06	7.0472D-06
9	2.0094D-04	6.8890D-06	6.7839D-06	7.1885D-06
10	2.2362D-04	6.9267D-06	6.6703D-06	7.1432D-06
20	4.6209D-04	7.5140D-06	6.8065D-06	7.5220D-06
30	7.3692D-04	7.8639D-06	6.7329D-06	7.4406D-06
40	1.0910D-03	8.6263D-06	6.8165D-06	7.5824D-06
50	1.5335D-03	1.0061D-05	7.0670D-06	7.9677D-06

Table 7. Maximum Relative Error $E_{R,\infty}$ at $T = 300$ for $w = 5$ using c6o0.

δ	$N = 2$	$N = 4$	$N = 6$	$N = 8$
0	2.2534D-04	2.2534D-04	2.2534D-04	2.2534D-04
1	2.6601D-05	3.4425D-05	5.9696D-05	8.2217D-05
2	4.3131D-05	6.2235D-06	1.7624D-05	3.2172D-05
3	6.4181D-05	1.5625D-06	5.1648D-06	1.2089D-05
4	8.5683D-05	9.9872D-07	1.9351D-06	4.8042D-06
5	1.0738D-04	1.0130D-06	1.0280D-06	2.2214D-06
6	1.2926D-04	1.0751D-06	7.8671D-07	1.2806D-06
7	1.5132D-04	1.1439D-06	7.4186D-07	9.2236D-07
8	1.7358D-04	1.2138D-06	7.3196D-07	8.1127D-07
9	1.9603D-04	1.2842D-06	7.3267D-07	8.4795D-07
10	2.1868D-04	1.3546D-06	7.3028D-07	8.8452D-07
20	4.5711D-04	2.0802D-06	8.0357D-07	1.6324D-06
30	7.3196D-04	2.9816D-06	1.4154D-06	2.2618D-06
40	1.0861D-03	4.7811D-06	1.8385D-06	2.5747D-06
50	1.5286D-03	5.0829D-06	2.0886D-06	3.0584D-06

Table 8. Maximum Relative Error $E_{R,\infty}$ at $T = 100$ for $w = 5$ using c8o0.

δ	$N = 2$	$N = 4$	$N = 6$	$N = 8$
0	7.3527D-03	7.3527D-03	7.3527D-03	7.3527D-03
1	1.9038D-04	7.9686D-03	1.5576D-03	2.2787D-03
2	5.4334D-05	1.0131D-04	3.5209D-04	7.1974D-04
3	8.0721D-05	1.8508D-05	8.7676D-05	2.3945D-04
4	1.0547D-04	8.7586D-06	2.5207D-05	8.0350D-05
5	1.3012D-04	8.2486D-06	1.0214D-05	3.0347D-05
6	1.5471D-04	8.5352D-06	7.1038D-06	1.4426D-05
7	1.7923D-04	8.9024D-06	6.5961D-06	8.8113D-06
8	2.0370D-04	9.2699D-06	6.4436D-06	7.2109D-06
9	2.2815D-04	9.6386D-06	6.3638D-06	7.8851D-06
10	2.5258D-04	1.0015D-05	6.3005D-06	8.2652D-06
20	4.9948D-04	1.3979D-05	6.8437D-06	9.7028D-06
30	7.6173D-04	1.7375D-05	7.7713D-06	1.2908D-05
40	1.0610D-03	2.1254D-05	8.2338D-06	1.4921D-05
50	1.4017D-03	2.6457D-05	9.7707D-06	1.8382D-05

Table 9. Maximum Relative Error $E_{R,\infty}$ at $T = 300$ for $w = 5$ using c8o0.

δ	$N = 2$	$N = 4$	$N = 6$	$N = 8$
0	7.2911D-03	7.2911D-03	7.2911D-03	7.2911D-03
1	1.8496D-04	7.9443D-04	1.5518D-03	2.2700D-03
2	4.8991D-05	9.9695D-05	3.5632D-04	7.2380D-04
3	7.5424D-05	1.3411D-05	8.4608D-05	2.4246D-04
4	1.0019D-04	3.3746D-06	1.9917D-05	7.7878D-05
5	1.2483D-04	2.7558D-06	5.1706D-06	2.5725D-05
6	1.4940D-04	3.0026D-06	1.7534D-06	9.1077D-06
7	1.7390D-04	3.3729D-06	9.5728D-07	3.2450D-06
8	1.9837D-04	3.7611D-06	7.5744D-07	1.3917D-06
9	2.2281D-04	4.1525D-06	7.4484D-07	1.8547D-06
10	2.4724D-04	4.5456D-06	7.8893D-07	2.2613D-06
20	4.9419D-04	8.4585D-06	1.2840D-06	4.4937D-06
30	7.5649D-04	1.2002D-05	2.0897D-06	8.1192D-06
40	1.0558D-03	1.5993D-05	2.8867D-06	9.7104D-06
50	1.3965D-03	2.1187D-05	4.4037D-06	1.3040D-05

Table 10. Maximum Relative Error $E_{R,\infty}$ at $T = 100$ for $w = 5$ using c10o0.

δ	$N = 2$	$N = 4$	$N = 6$	$N = 8$
0	2.4521D-02	2.4521D-02	2.4521D-02	2.4521D-02
1	5.2590D-04	2.1753D-03	4.2816D-03	6.2632D-03
2	1.3842D-04	2.4306D-04	8.8103D-04	1.8390D-03
3	1.8732D-04	3.3439D-05	1.9733D-04	5.6376D-04
4	2.4585D-04	1.1145D-05	4.7289D-05	1.7304D-04
5	3.0385D-04	1.0321D-05	1.4513D-05	5.7531D-05
6	3.6107D-04	1.1155D-05	7.6035D-06	2.2205D-05
7	4.1753D-04	1.2055D-05	6.3145D-06	1.0417D-05
8	4.7327D-04	1.2953D-05	6.0015D-06	7.4360D-06
9	5.2832D-04	1.3854D-05	5.9470D-06	9.0071D-06
10	5.8272D-04	1.4761D-05	5.9463D-06	9.5066D-06
20	1.0982D-03	2.4112D-05	7.3354D-06	1.4003D-05
30	1.5798D-03	3.3038D-05	8.6303D-06	1.4011D-05
40	2.0544D-03	4.2769D-05	1.0075D-05	2.3476D-05
50	2.5572D-03	5.1748D-05	1.9478D-05	3.2302D-05

Table 11. Maximum Relative Error $E_{R,\infty}$ at $T = 200$ for $w = 5$ using c10o0.

δ	$N = 2$	$N = 4$	$N = 6$	$N = 8$
0	2.4517D-02	2.4517D-02	2.4517D-02	2.4517D-02
1	5.2169D-04	2.1715D-03	4.2772D-03	6.2560D-03
2	1.3421D-04	2.3891D-04	8.7763D-04	1.8354D-03
3	1.8315D-04	2.9196D-05	1.9314D-04	5.6147D-04
4	2.4171D-04	6.9691D-06	4.2957D-05	1.7013D-04
5	2.9970D-04	6.0329D-06	1.0366D-05	5.3718D-05
6	3.5690D-04	3.2921D-06	3.2921D-06	1.7761D-05
7	4.1334D-04	7.7315D-06	2.0362D-06	6.1763D-06
8	4.6907D-04	8.6504D-06	1.7333D-06	2.6527D-06
9	5.2412D-04	9.5734D-06	1.8554D-06	4.0073D-06
10	5.7852D-04	1.0498D-05	2.0365D-06	4.6216D-06
20	1.0940D-03	1.9794D-05	2.9482D-06	9.8345D-06
30	1.5756D-03	2.8852D-05	4.0356D-06	1.5304D-05
40	2.0503D-03	3.8692D-05	5.7254D-06	1.9260D-05
50	2.5531D-03	4.7668D-05	9.7533D-06	2.8188D-05

computed to $T = 300$ with $w = 5$ is reported in Tables 5, 7 and 9 for the c4o0, c6o0 and c8o0 algorithms, respectively, and to $T = 200$ with $w = 5$ in Table 11 for the c10o0 algorithm. All of these computations are on the numerical domain $\Omega_N = [-3, 7] \times [-5, 5]$ with the base flow $\vec{V}_b = \vec{V}_1 = (1, 0.4, 0, 1/\gamma)$. The grid resolution for the c4o0 computations is $\Delta x = \Delta y = 1/8$ and $\Delta t = 1/24$, while the grid resolution for the other three algorithms is $\Delta x = \Delta y = 1/6$ and $\Delta t = 1/18$. The maximum damping amplitude δ is in the first column of these tables, and in the second through fifth columns are the maximum relative errors $E_{R,\infty}$ in Ω_N for the damping profile powers $N = 2$, 4, 6 and 8, respectively. The data in these four pairs of tables reflects the effect of damping profile power, damping amplitude, simulation time, algorithm and grid resolution, while the damping layer width is the same throughout. The first row in each of these tables is for $\delta = 0$ with no active damping, as discussed above, and is included here for ease of reference. The remaining rows in these tables are from simulations with active damping $\delta > 0$.

Tables 4 and 5 are from computations with the c4o0 algorithm for simulation times $T = 100$ and 300, respectively. For every N in both Tables, the error levels for damping with $\delta = 1$ are all very close to the error levels for no damping with $\delta = 0$, differing by less than a factor of 5. For $N = 2$, the errors grow with damping amplitude from $\delta = 1$ to $\delta = 50$ until they are two orders of magnitude larger than the errors with no damping at $\delta = 0$. The rightmost three columns in this pair of tables are for damping profile powers $N = 4$, 6 or 8. For each $N \neq 2$, the largest errors for $\delta > 0$ are reported at $\delta = 1$, and the error levels for $\delta \geq 2$ are all relatively stable at one or two orders of magnitude less than for $\delta = 0$ with no damping. Note that the errors for $T = 300$ are smaller than the corresponding errors for $T = 100$, by up to as much as an order of magnitude, and relatively smaller for larger N.

Tables 6 and 7 are from computations with the c6o0 algorithm for simulation times $T = 100$ and 300, respectively. The pattern and levels of error variation in these tables for the c6o0 algorithm are close to the pattern and levels in Tables 4 and 5 for the c4o0 algorithm. In both Tables, for every N, the error levels for damping with $\delta = 1$ are all within an order of magnitude of the error levels for no damping with $\delta = 0$. For $N = 2$, the error minimum at $\delta = 1$ is a little less than one order of magnitude smaller then the error with no damping, and the error grows with δ until it is two orders of magnitude greater than the error minimum. For $N \neq 2$ in Table 6 at $T = 100$, the error declines with δ to a minimum that is about an order of magnitude smaller than the error at $\delta = 1$, and then the error grows with δ, but it never gets larger than the error at $\delta = 1$. For $N \neq 2$ in Table 7 at $T = 300$, the error declines with δ to a minimum that is up to two orders of magnitude smaller than the error at $\delta = 1$, and then the error grows with δ, but at $\delta = 50$ it is only up to about an order of magnitude smaller than the error at $\delta = 1$. Notice that the errors from the c4o0 and c6o0 algorithms have a similar range for $\delta \geq 1$, but that the undamped errors from the c4o0 algorithm are about an order of magnitude smaller than from the c6o0 algorithm. Damping has a greater effect relative to no damping for the c6o0 algorithm than for the c4o0 algorithm, because the errors are more accurately propagated with the 5^{th} order c6o0 algorithm.

Tables 8 and 9 are from computations with the c8o0 algorithm for simulation times $T = 100$ and 300, respectively. The data for the c8o0 algorithm show a more gradual transition with damping amplitude from $\delta = 0$ than was observed in the data for the c4o0 and c6o0 algorithms. In Tables 8 and 9 for both $T = 100$ and 300, the errors for $\delta = 1$ are less than an order of magnitude smaller than for $\delta = 0$ for $N = 4$, 6 and 8, while the errors for $\delta = 2$ are greater than an order of magnitude smaller than for $\delta = 0$ for all N. In these two tables from the c8o0 algorithm the general error patterns for $\delta \geq 2$ are like the error patterns for $\delta \geq 1$ from the c4o0 and c6o0 algorithms. For each $N \neq 2$, the largest errors for $\delta > 1$ are reported at $\delta = 2$, and they are about an order of magnitude greater than for the c4o0 and c6o0 algorithms, but the smallest errors are within an order of magnitude of the smallest errors for the c6o0 algorithm. For $N = 2$, the largest errors for $\delta > 0$ are reported at $\delta = 50$.

Tables 10 and 11 are from computations with the c10o0 algorithm for simulation times $T = 100$ and 200, respectively. Note that the longest simulation time for the c10o0 algorithm is $T = 200$ instead of $T = 300$ for the other three algorithms, simply due to the time required to compute the comparison solution for this algorithm at $T = 300$. The pattern in the data for the c10o0 algorithm is similar to the pattern for the c8o0 algorithm, with the general pattern of reported error levels becoming evident at about $\delta = 2$ for the c10o0

algorithm instead of at $\delta = 1$ for the more diffusive and lower order c4o0 and c6o0 algorithms. For each $N \neq 2$, the largest errors for $\delta > 1$ are reported at $\delta = 2$, and they are as much as two orders of magnitude smaller than for the errors with no damping. For $N = 2$, the largest errors for $\delta > 0$ are reported at $\delta = 50$. In both Tables, for $N = 2$, the error levels for damping with $\delta = 1$ are about two orders of magnitude smaller than the error levels for no damping with $\delta = 0$, while for $N \neq 2$, the error levels with $\delta = 1$ are about an order of magnitude smaller than the error levels for no damping with $\delta = 0$. The pattern and levels of error variation for $\delta \geq 2$ in these tables for the c10o0 algorithm are very close to the pattern and levels for $\delta \geq 2$ in Tables 8 and 9 for the c8o0 algorithm.

Table 12. Smallest observed error $E_{R,\infty}$ with $w = 5$ at $T = 100$.

algorithm	$N = 2$	$N = 4$	$N = 6$	$N = 8$
c4o0	2.1854D-05	6.4663D-06	6.5709D-06	6.2108D-06
c6o0	3.1538D-05	6.5983D-06	6.4781D-06	7.0472D-06
c8o0	5.4334D-05	8.2486D-06	6.3638D-06	7.2109D-06
c10o0	1.3842D-04	1.0321D-05	5.9463D-06	7.4360D-06

Table 13. Smallest observed error $E_{R,\infty}$ with $w = 5$ at $T = 300$.

algorithm	$N = 2$	$N = 4$	$N = 6$	$N = 8$
c4o0	1.6771D-05	8.9120D-07	7.2492D-07	7.2492D-07
c6o0	2.6601D-05	9.9872D-07	7.3028D-07	8.1127D-07
c8o0	4.8991D-05	2.7558D-06	7.4484D-07	1.3917D-06
c10o0 $(T = 200)$	1.3421D-04	3.2921D-06	1.7333D-06	2.6527D-06

Consider the difference in the error variation between the data for $N = 2$ and the data for $N = 4$, 6 and 8 in Tables 4 through 11. For $N = 2$, the errors increase with damping amplitude from $\delta = 1$ to $\delta = 50$, for $\delta \geq 2$ they are all greater than the error with $\delta = 0$, and for $\delta = 50$ they are two orders of magnitude greater than with no damping. On the other hand, for each $N = 4$, 6 and 8, and for both $T = 100$ and $T = 300$, for $\delta \geq 1$ the errors initially decrease with damping amplitude until a local error minimum is observed for some value of $\delta \leq 50$. For each $N \neq 2$, the error levels for $\delta \geq 2$ are all one or two orders of magnitude less than for no damping with $\delta = 0$. In these computations, damping with a quadratic polynomial is not nearly as accurate as damping with higher order polynomial profiles. A quadratic damping profile with $N = 2$ does not produce the smallest errors, but it does produce the largest, and damping with a quadratic polynomial can make the errors grow considerably larger than not damping at all. For a quadratic damping profile, there is a second derivative discontinuity at the interface between the damping layer and the numerical domain. Higher order damping profiles have weaker discontinuities at this interface. In addition, when compared to higher order damping profiles, quadratic damping more rapidly creates a distortion from the damping in the propagating solution surfaces as they enter the damping layer. Active damping can make a significant difference, but it can either help or hurt, depending upon how it is done.

In order to get a general idea of the data in Tables 4 through 11, Tables 12 and 13 give the smallest observed error for each of the four algorithms and for each damping profile power N. Each row in these tables is from one of Tables 4 through 11, and is indexed by the algorithm in the far lefthand column. The entries in each row are ordered by the damping profile power N, and give the least error in each column from the originating table headed by that damping profile power. Both tables are for damping layer width $w = 5$. Table 12 presents the error data $E_{R,\infty}$ from each of the four algorithms with damping width $w = 5$ for simulation time $T = 100$. Table 13 presents the data from the three lower order algorithms with damping

layer width $w = 5$ for simulation time $T = 300$, and for $T = 200$ from the c10o0 algorithm. Notice in these summary tables, that across the rows for each algorithm, damping with $N = 2$ always has the largest least error. Notice also that the upper right triangular section in each table generally has similar error levels, and that these are the lowest levels in each table. In Table 12 the least errors are $O[10^{-6}]$ at $T = 100$, and in Table 13 the least errors are $O[10^{-7}]$ at $T = 300$. For each algorithm, the lowest errors with the high values of N at $T = 300$ are about an order of magnitude lower than the comparable errors at $T = 100$. In order to reach these lowest error levels in each table, a higher order damping profile power N is used with the higher order algorithms. On the other hand, the lower left entries in these tables show that if a lower order damping profile power N is used with a higher order algorithm, then this combination will produce a smallest observed error that is larger by as much as two orders of magnitude than for the same higher order algorithm with higher order damping profiles. In other words, even though higher order damping profiles work better with all of the algorithms, the relative improvement with damping profile power is greater for the higher order algorithms, and is even needed by them to achieve the best performance possible. A damping profile with power N will have an N^{th} derivative discontinuity at the interface between the numerical domain Ω_N and the damping layer Ω_D, and since the damping terms are included in the governing equations everywhare in the computational domain Ω_C, this will create an N^{th} order error at the interface. The data pattern in these two tables suggest the rule of thumb that the damping profile power should be at least as large as the algorithm order. It appears from these tables that the 9^{th} order c10o0 algorithm might need either a damping polynomial with $N = 10$ to reach the lowest error levels, or to be run longer than the simulation time $T = 200$, or both. Note that quadratic damping appears to be the least accurate with all of the algorithms.

VIII. The Effect Of Damping Layer Width

Tables 14 and 15 present data from simulations with the 7^{th} order c8o0 algorithm for $w = 10$ at $T = 100$ and $T = 300$, respectively, and with the uniform mean flow $\vec{V}_1$ and the grid density $\Delta x = \Delta y = 1/6$ and $\Delta t = 1/18$. The comparable c8o0 simulations that produced Tables 8 and 9 are for $w = 5$, but besides this, the simulations are exactly the same. As a summary comparison, Tables 16 and 17 present the smallest observed errors by damping profile power N for each width w at $T = 100$ and 300, respectively. The left most column in these tables gives the number of the Table from which the data in that row is obtained, the next left column gives the damping layer width w, and the rest of each row gives the smallest observed errors from the table for that width by the damping profile power N. At both simulation times we observe that the reduction in the smallest observed error by doubling the damping layer width w is greater for the lower order damping polynomial powers N. For $N = 2$, this error reduction ranges from a factor of 4.3 at $T = 100$ to 6.6 at $T = 300$, and for $N = 8$, the error reduction is 1.2 at $T = 100$ and 2.2 at $T = 300$. Similarly, for both damping layer widths, we observe that the reduction in the smallest observed error by tripling the simulation time T is greater for the higher order damping polynomial power N. For $N = 2$, this error reduction is 1.1 for $w = 5$ and 1.7 for $w = 10$, for $N = 6$ it is 8.5 for $w = 5$ and 10.4 for $w = 10$, and for $N = 8$ it is 5.2 for $w = 5$ and 9.9 for $w = 10$. Including the damping layer, the complete computational domain for $w = 5$ is $[-8, 12] \times [-10, 10]$, or 400 square units of area, while the complete domain for $w = 10$ is $[-13, 17] \times [-15, 15]$, or 900 square units. The computational effort is directly proportional to the area times the time. Increasing the damping layer width increases the computational area and the effort at both times by a factor of $9/4 = 2.25$, while increasing the time increases the effort by a factor 3 for either damping layer width. For $N = 2$, the error is decreased with more efficiency by widening the damping layer than by lengthening the simulation time. For $N \neq 2$, error decrease is more efficient by increasing the simulation time. The most effective error decrease is for $N \neq 2$ with both the wider damping layer $w = 10$ and the longer simulation time $T = 300$, with effort increasing by 6.75 while error decreases by factors of 8.4, 10.4 and 9.9 for $N = 4$, 6 and 8, respectively. For $N = 2$ the only efficient strategy is to increase the simulation time, and even then, the relative error is largest for quadratic damping if all else is the same.

Table 14. Maximum Relative Error $E_{R,\infty}$ at $T = 100$ for $w = 10$ using c8o0.

δ	$N = 2$	$N = 4$	$N = 6$	$N = 8$
0	1.9533D-03	1.9533D-03	1.9533D-03	1.9533D-03
1	1.2738D-05	3.4119D-05	1.0968D-04	2.1452D-04
2	1.8377D-05	6.6795D-06	1.0993D-05	2.9307D-05
3	2.4691D-05	6.7165D-06	6.5974D-06	1.0374D-05
4	3.0987D-05	6.9389D-06	6.8708D-06	7.8906D-06
5	3.7252D-05	6.8000D-06	6.9328D-06	7.0134D-06
6	4.3493D-05	6.6660D-06	7.0733D-06	6.2657D-06
7	4.9716D-05	6.5253D-06	7.2468D-06	6.4581D-06
8	5.5929D-05	6.3871D-06	7.4256D-06	6.8782D-06
9	6.2134D-05	6.2452D-06	7.5994D-06	7.3661D-06
10	6.9332D-05	6.1160D-06	7.7649D-06	7.7671D-06
15	9.9256D-05	5.8975D-06	8.4488D-06	8.3154D-06
20	1.3008D-04	6.1427D-06	8.8546D-06	8.5523D-06
30	1.9143D-04	6.5064D-06	7.9234D-06	9.0853D-06
40	2.5254D-04	6.7191D-06	7.4800D-06	9.5495D-06
50	3.1367D-04	7.0092D-06	7.2186D-06	9.9713D-06

Table 15. Maximum Relative Error $E_{R,\infty}$ at $T = 300$ for $w = 10$ using c8o0.

δ	$N = 2$	$N = 4$	$N = 6$	$N = 8$
0	1.9408D-03	1.9408D-03	1.9408D-03	1.9408D-03
1	7.4695D-06	2.9620D-05	1.0337D-04	2.0929D-04
2	1.3187D-05	1.1026D-06	6.0255D-06	2.1829D-05
3	1.9450D-05	7.0294D-07	9.3140D-07	2.9113D-06
4	2.5705D-05	7.3337D-07	6.5116D-07	8.7703D-07
5	3.1952D-05	7.5825D-07	6.3653D-07	6.6043D-07
6	3.8192D-05	7.8306D-07	6.3603D-07	6.3836D-07
7	4.4424D-05	8.0786D-07	6.3610D-07	6.3606D-07
8	5.0648D-05	8.3266D-07	6.3619D-07	6.3581D-07
9	5.6866D-05	8.5746D-07	6.3627D-07	6.3578D-07
10	6.3077D-05	8.8225D-07	6.3636D-07	6.3579D-07
15	9.4035D-05	1.0062D-06	6.3728D-07	6.3576D-07
20	1.2485D-04	1.1301D-06	6.3826D-07	6.3579D-07
30	1.8616D-04	1.3776D-06	6.4021D-07	6.3599D-07
40	2.4726D-04	1.6250D-06	6.4217D-07	6.3586D-07
50	3.0840D-04	1.8720D-06	6.4408D-07	6.3654D-07

Table 16. Smallest observed error $E_{R,\infty}$ from c8o0 at $T = 100$.

Table	w	$N = 2$	$N = 4$	$N = 6$	$N = 8$
8	5	5.4334D-05	8.2486D-06	6.3005D-06	7.2109D-06
14	10	1.2738D-05	5.8781D-06	6.5974D-06	6.2657D-06

Table 17. Smallest observed error $E_{R,\infty}$ from c8o0 at $T = 300$.

Table	w	$N = 2$	$N = 4$	$N = 6$	$N = 8$
9	5	4.8991D-05	2.7558D-06	7.4484D-07	1.3917D-06
15	10	7.4695D-06	7.0294D-07	6.3603D-07	6.3576D-07

IX. Damping With A Parallel Jet

Table 18. $E_{R,\infty}$ at $T = 50$ using c8o0 with a uniform flow.

δ	$N = 2$	$N = 4$	$N = 6$	$N = 8$
0	2.5577D-02	2.5577D-02	2.5577D-02	2.5577D-02
1	5.3790D-04	2.4647D-03	4.8330D-03	7.0131D-03
5	1.2924D-04	3.1870D-05	4.7355D-05	7.3903D-05
10	2.3880D-04	3.9404D-05	4.9990D-05	4.5910D-05
15	3.5122D-04	3.8657D-05	4.6522D-05	5.1614D-05
20	4.6372D-04	3.9515D-05	4.4855D-05	5.5577D-05
30	6.8830D-04	3.9135D-05	4.5565D-05	6.0124D-05
40	9.1741D-04	3.9343D-05	5.4631D-05	6.1003D-05
50	1.1813D-04	4.0562D-05	5.7788D-05	6.2183D-05

Table 18 presents data for simulations with the uniform mean flow

$$\vec{V_b} = (\sigma, u, v, p)_b^T = \vec{V_1} = (1, 0.4, 0, 1/\gamma), \tag{32}$$

on the numerical domain $\Omega_N = [-3, 7] \times [-5, 5]$. Table 19 presents data for simulations with the steady parallel jet

$$\vec{V_b} = (\sigma, u, v, p)_b^T = \vec{V_2} = (1, 0.4 + 0.4\exp[-25y^2], 0, 1/\gamma), \tag{33}$$

on the numerical domain $\Omega_N = [-3, 9] \times [-5, 5]$, which is extended to the right or downstream because of the faster peak convection velocity of the parallel jet. All of the simulations that produced Tables 18 and 19 are with the governing equations (15) where the pressure source is (20) where the damping profiles D_x and D_y in (12) and (13) are as discussed above, and where the terms in the coefficient matrices A_b and B_b are from the mean flow $\vec{V_b}$ in each case. Both cases have been nondimensionalized by the reference speed of sound a_R and specific volume $\sigma_R = 1/\rho_R$, with reference pressure $p_R = a_R^2/\sigma_R^2$. The uniform mean flow simulations have constant coefficient governing equations, while the steady parallel jet flow simulation coefficients are steady in time but variable in space. All of the simulations that created the error data in Tables 18 and 19

Table 19. $E_{R,\infty}$ at $T = 50$ using c8o0 with a parallel jet.

δ	$N = 2$	$N = 4$	$N = 6$	$N = 8$
0	3.4352D-02	3.4352D-02	3.4352D-02	3.4352D-02
1	4.8408D-04	2.1640D-03	4.2952D-03	6.3682D-03
5	2.4912D-04	4.3282D-05	5.0270D-05	7.2488D-05
10	4.6223D-04	4.7095D-05	6.0419D-05	6.4522D-05
15	7.0335D-04	4.4325D-05	6.9968D-05	7.4832D-05
20	9.9035D-04	4.3823D-05	7.6220D-05	8.8411D-05
30	1.8007D-03	5.0259D-05	8.2466D-05	1.0463D-04
40	2.9059D-03	6.5535D-05	8.7784D-05	1.1321D-04
50	4.1414D-03	6.7744D-05	9.4313D-05	1.1388D-04

have grid sizes

$$\Delta x = \Delta y = 1/12, \quad \text{and} \quad \Delta t = 1/36, \tag{34}$$

they are with damping layer width $w = 5$, and they are for simulation time $T = 50$. When comparing the errors in Tables 18 and 19 from the uniform and jet flow calculations, it is seen that they differ by a factor of up to approximately 2, indicating that the complete boundary treatment performs very much the same for both a uniform base flow and a steady parallel jet. Note that this data is for the relatively short simulation time $T = 50$ when compared to the prior data for $T = 100$ and $T = 300$, and recall that the boundary treatment errors are still relatively large for $T \leq 100$. Note also that the grid resolution here is double what was used for the prior simulations, and that the previous data has suggested that errors from the boundary treatment will be larger if the solution is better resolved. In light of these considerations, the errors with the uniform flow reported in Table 18 are reasonably close to those reported for the c8o0 algorithm in Table 8 at $T = 100$ with half the grid resolution, differing by up to a factor of approximately 5. Consequently, this data indicates that all of the previous discussion of the effects of damping profile power, damping layer width, simulation time and algorithm order should apply to variable coefficient parallel jet simulations with these algorithms just as they did to constant coefficient uniform flow simulations.

X. Conclusion

This paper reports numerical experiments with the Linearized Euler Equations in two space dimensions for propagating a continuous acoustic signal through a steady mean flow, primarily with a uniform flow, but a parallel jet has been considered. We have used the specific volume $\sigma = 1/\rho$, the Cartesian velocity components (u, v), and the pressure p as primitive variables. The acoustic signal is from an impulsively started continuous pressure source. Explicit and local Centered-Staggered/Cauchy-Kowaleskya/Taylor algorithms have been used for these computational experiments, with 3^{rd}, 5^{th}, 7^{th} and 10^{th} order accuracy in time and space. The main purpose of these numerical experiments has been to gain insight into how to control the simulation error when using a boundary treatment that combines a damping layer with an outer boundary condition. For these numerical experiments we have foregone a location dependent characteristic based outer boundary condition, but have chosen the very simple expedient of setting the data uniformly to zero on the outer boundary of the damping layer. We observe relative absolute errors that range from $O[10^{-2}]$ to $O[10^{-7}]$. The relative effect on simulation error of algorithm order, grid resolution, simulation time, damping layer width, damping profile power and damping amplitude is considered, and the boundary treatment with a uniform flow and with a parallel jet is compared. The damping profiles that have been

used are polynomials of order $N = 2$, 4, 6 and 8.

- Errors from just a damping layer with no explicit damping originate with the zero boundary condition on the outer boundary of the damping layer. Errors with no explicit damping from the dissipative 3^{rd} order c4o0 algorithm are $O[10^{-5}]$ or less for simulation times $T \geq 50$, while errors with no damping from the 9^{th} order c10o0 algorithm are all $O[10^{-2}]$.
- For explicit damping, a quadratic damping profile with $N = 2$ does not produce the smallest errors, but it does produce the largest. Damping with a quadratic polynomial can make the errors grow considerably larger than not damping at all. Active damping can make a significant difference, but it can either help or hurt, depending upon how it is done.
- Higher order damping profiles work better with all of the algorithms, but the relative improvement with damping profile power is greater for the higher order algorithms, and is even needed by them to achieve the best performance possible. As a rule of thumb, the damping profile power should be at least as large as the algorithm order.
- In general, if the solution is more accurately computed, either with a higher order algorithm or with a greater grid density, then the error from the outer boundary will also be propagated more accurately, and the reported boundary treatment error $E_{R,\infty}$ will be greater, irrespective of explicit damping.
- For each algorithm, and for each damping layer width w, the observed errors decrease with simulation time T.
- Error tends to decrease as the damping layer width increases.
- Error tends to decrease as the damping profile exponent increases.
- The relative decrease in error from increasing damping layer width is greater for smaller profile powers N, and the relative decrease in error from increasing simulation time is greater for larger N.
- Damping with a parallel jet behaves very much like damping with a uniform base flow.

One set of governing equations with both source and damping terms has been used throughout the complete simulation domain. Simple changes of spatial interpolant, the order of recursion and the order of time advance easily changes the order of the algorithm as a whole. These algorithms generally have the same order in space and time, and results in 2D have been shown from algorithms that are 3^{rd} to 9^{th} order accurate in both space and time. Except for the multidimensional interpolants, which use dimensional recursion, the algorithms are intrinsically independent of dimension, and they capture the multidimensional propagation of the governing equations up to the order of the algorithm. Compared to characteristic based outer boundary treatments, the resulting code simplicity and flexibility is particularly striking. The code simplicity and flexibility of these algorithms with this boundary treatment is particularly suitable for 3D simulations. A simple replacement of the Cauchy-Kowaleskya recursion subroutine can adapt the algorithm to a different set of governing equations.

Acknowledgments

The Commercial Supersonics Technology Project under NASA Advanced Air Vehicles Program supported this work.

References

[1]Daniel J. Bodony, "Analysis of sponge zones for computational fluid dynamics," *J. Comp. Phys.* **212** (2006) pp681-702.

[2]Tim Colonius, "Modeling Artificial Boundary Conditions for Compressible Flow," *Annu. Rev. Fluid Mech.* **36** (2004) pp315-345.

[3]Tim Colonius, Sanjiva K. Lele, "Computational aerocoustics: progress on nonlinear problems of sound generation," *Prog. Aerospace Sciences* **40** (2004) pp345-416.

[4]Maria Fröling, *A Numerical Investigation of a Perfectly Matched Layer for Computational Aero Acoustics*, TRITA-NA-E04041, Master's Thesis, Royal Institute of Technology, Stockholm Sweden (2004).

[5]Giles, M. B., "Non-reflecting Boundary Conditions for the Euler Equations," Computational Fluid Dynamics Lab., Massachusetts Inst. of Technology, Cambridge, MA, CFDL-TR 88-1, Feb., 1988.

[6]Giles, Michael B., "Nonreflecting Boundary Conditions for Euler Equation Calculations," *AIAA Journal* **28** (1990), pp2050-2058.

[7]Dan Givoli, "High-order local non-reflecting boundary conditions: a review," *Wave Motion* **39** (2004), pp319-326.

[8]J. W. Goodrich, "An Approach to the Development of Numerical Algorithms for first Order Linear Hyperbolic Systems in Multiple Space Dimensions: The Constant Coefficient Case," NASA TM 106928 (1995).

[9]J. W. Goodrich, "Accurate Finite Difference Algorithms," NASA TM 107377 (1996).

[10]J. W. Goodrich, "A comparison of Numerical Methods for Computational Aeroacoustics," AIAA 99-1943, *The 5th AIAA/CEAS Aeroacoustics Conference*, (Bellevue, WA, May, 1999).

[11]J. W. Goodrich, "Experiments in numerical error control for sound propagation in a 2D jet using a damping layer and nonreflecting outer boundaries," *Int. J. Aeroacoustics* **14** (2015), pp489-520.

[12]J. W. Goodrich, T. Hagstrom, J. Lorenz, "Hermite Methods for Hyperbolic Initial-Boundary Value Problems," *Math. Comp.* **75** (January, 2006) pp595-630.

[13]Thomas Hagstrom, "Radiation Boundary Conditions for the Numerical Simulation of Waves," *Acta Numerica* **8** (1999) pp47-106.

[14]Thomas Hagstrom, *A New Construction of Perfectly Matched Layers for Hyperbolic Systems with Applications to the Linearized Euler Equations*, in "Proceedings of the Sixth International Conference on Mathematical and Numerical Aspects of Wave Propagation," Finland, E. Heikkola, P. Neittaanmki, G. C. Cohen, P. Joly, eds., pp. 125-129, Springer Verlag (2003).

[15]Hedstrom, G. W., *J. Comp. Phys.* **30** (1979) pp222.

[16]Fang Q. Hu, "Absorbing Boundary Conditions," *Int. J. Comp. Fluid Dyn.* **18** (2004) pp513-522.

[17]Moshe Israel, Steven A. Orszag, "Approximation of Radiation Boundary Conditions," *J. Comp. Phys.* **41** (1981) pp115-135.

[18]Smadar Karni, "Far-Field Filtering Operators for Suppression of Reflections from Artificial Boundaries," *SIAM J. Numer. Anal.* **33** (1996) No.3, pp1014-1047.

[19]Sanjiva K. Lele, "Computational Aeroacoustics - A review," AIAA-97-0018 (1997).

[20]Ali Mani, "Analysis and optimization of numerical sponge layers as a nonreflective boundary treatment," *J. Comp. Phys.* **231** (2012) pp704-716.

[21]M. Motamed, *PML Methods for Aero Acoustics Computations*, TRITA-NA-E03108, Master's Thesis, Royal Institute of Technology, Stockholm Sweden (2003).

[22]E. Romenski, V. A. Titarev, and E. F. Toro, "Perfectly Matched Layers with High Rate Damping for Hyperbolic Systems," *CFD J.* **15** (2006) No. 2, pp240-246.

[23]James A. Sochacki, *Absorbing Boundary Conditions Using Regions with Exponential Decay*, in "Mathematical and Numerical Aspects of Wave Propagation Phenomena," Proceedings of the First International Conference on Mathematical and Numerical Aspects of Wave Propagation, (Strasbourg, France, April 23-26, 1991). Finland, Gary Cohen, Laurence Haloern, Patrick Joly, eds., pp. 293-302, SIAM (1991).

[24]Christopher K. W. Tam, "Computational Aeroacoustics: Issues and Methods," AIAA-97-1774 (1997).

[25]Christopher K. W. Tam, "Advances in Numerical Boundary Conditions for Computational Aeroacoustics," *AIAA J.* **33** (1995) No. 10, pp1788-1796.

[26]Thompson, Kevin W., *Lecture Series in Computational Fluid Dynamics*, NASA TM100010, 1987.

[27]Kevin W. Thompson, "Time Dependent Boundary Conditions for Hyperbolic Systems," *J. Comp. Phys.* **68** (1987) pp1-24.

[28]Kevin W. Thompson, "Time Dependent Boundary Conditions for Hyperbolic Systems, II" *J. Comp. Phys.* **89** (1990) pp439-461.

[29]Semyon V. Tsynkov, "Numerical solution of problems on unbounded domains. A review.," *Applied Num. Math.* **27** (1998) pp465-532.

[30]Meng Wang, Jonathan B. Freund, and Sanjiva K. Lele, "Computational Prediction of Flow-Generated Sound," *Annu. Rev. Fluid Mech.* **38** (2006) pp483-512.