

11-64  
1000

# Accurate Finite Difference Algorithms

John W. Goodrich  
*Lewis Research Center*  
*Cleveland, Ohio*

Prepared for the  
Barriers and Challenges in Computational Fluid Dynamics Workshop  
sponsored by the Institute for Computer Applications in Science  
and Engineering and NASA Langley Research Center  
Hampton, Virginia, August 5-7, 1996



National Aeronautics and  
Space Administration



# ACCURATE FINITE DIFFERENCE ALGORITHMS

JOHN W. GOODRICH  
*NASA Lewis Research Center*  
*Cleveland, OH 44135, USA*

## 1. Introduction

Clear examples of the difficulties associated with applying CFD techniques to apparently simple problems are provided by computational aeroacoustics and computational electromagnetics. These applications require accurate wave propagation over long distances for a wide range of frequencies, placing a severe demand on numerical algorithms, and raising issues related to efficiency, accuracy, compatible space and time treatments, high frequency data, propagation along characteristic surfaces, isotropy, stable and accurate artificial boundary treatments, and nonrestrictive stability bounds. This paper briefly presents two methods for the development of finite difference algorithms which are intended to address these issues. High order single step explicit algorithms are possible, and examples with up to eleventh order accuracy will be shown. High resolution algorithms in the sense of [9] are also possible, with amplification factor and relative phase change per time step which are virtually 1 for normal mode frequencies in  $[0, \pi]$  and CFL numbers in  $[0, 1]$ . If our most accurate algorithm is used to propagate an initial periodic sine wave, then after five periods with four grid points per wavelength, the maximum error is  $O[10^{-6}]$ , and after five hundred thousand periods with eight grid points per wavelength, the maximum error is  $O[10^{-4}]$ . High order algorithms are relatively more efficient [7], and their relative efficiency tends to increase as the error bound decreases, as the simulation time increases, and as the spatial dimension increases. Our algorithms show several orders of magnitude difference in the number of multiplications required to meet an error bound of  $O[10^{-4}]$  at five periods of propagation. Applied computations require boundary conditions. A new artificial boundary condition that has been developed by Hagstrom [5] will be shown for the linearized Euler equations. This boundary condition is local in time and does not require information about the distance or direction of either an assumed source or an expanding wave front. The boundary and propagation algorithms have a consistent derivation and similar properties [4]. Additional issues are raised by shocks, but are not addressed here.

## 2. Algorithms for the Linearized Euler Equations in 1D

As a first example we will consider the linearized Euler equations in one space dimension. The equations for the isentropic case can be written in nondimension-

alized form as the system

$$\begin{aligned}\frac{\partial u}{\partial t} + M \frac{\partial u}{\partial x} + \frac{\partial p}{\partial x} &= 0, \\ \frac{\partial p}{\partial t} + M \frac{\partial p}{\partial x} + \frac{\partial u}{\partial x} &= 0,\end{aligned}\tag{1}$$

where  $M$  is the constant mean convection Mach number,  $p$  is the disturbance pressure, and  $u$  is the disturbance velocity. System (1) can be diagonalized and solved by the method of characteristics to produce the general solution

$$\begin{aligned}u(x, t) &= \frac{1}{2}(ui(x - (M + 1)t) + pi(x - (M + 1)t)) \\ &\quad + \frac{1}{2}(ui(x - (M - 1)t) - pi(x - (M - 1)t)), \\ p(x, t) &= \frac{1}{2}(pi(x - (M + 1)t) + ui(x - (M + 1)t)) \\ &\quad + \frac{1}{2}(pi(x - (M - 1)t) - ui(x - (M - 1)t)),\end{aligned}\tag{2}$$

with initial data  $ui(x) = u(x, 0)$ , and  $pi(x) = p(x, 0)$ .

The first step in producing algorithms for solutions of (1) is to locally interpolate  $u$  and  $p$  at  $t_n$  with order  $D$  polynomials in  $x$ :

$$u(x_i + x, t_n) \approx ua(x) = \sum_{\alpha=0}^D u_{\alpha} x^{\alpha}, \quad p(x_i + x, t_n) \approx pa(x) = \sum_{\alpha=0}^D p_{\alpha} x^{\alpha}.\tag{3}$$

The expansion coefficients are obtained by the Method of Undetermined Coefficients using the known data on a given stencil. Note that there is no specification of a particular mesh or data type, and that separate derivative terms are not even considered. The use of single interpolation polynomials of order  $D$  is equivalent to simultaneously approximating derivatives up to the  $D^{\text{th}}$  order, with

$$u_{\alpha} = \frac{1}{\alpha!} \frac{\partial^{\alpha} ua}{\partial x^{\alpha}} \approx \frac{1}{\alpha!} \frac{\partial^{\alpha} u}{\partial x^{\alpha}}, \quad \text{and} \quad p_{\alpha} = \frac{1}{\alpha!} \frac{\partial^{\alpha} pa}{\partial x^{\alpha}} \approx \frac{1}{\alpha!} \frac{\partial^{\alpha} p}{\partial x^{\alpha}}.\tag{4}$$

The local interpolations (3) to  $u$  and  $p$  at time  $t_n$  are used as initial data for the exact solution (2), producing an approximate local solution

$$\begin{aligned}u(x_i + x, t_n + t) &\approx \frac{1}{2}(ua(x - (M + 1)t) + pa(x - (M + 1)t)) \\ &\quad + \frac{1}{2}(ua(x - (M - 1)t) - pa(x - (M - 1)t)), \\ p(x_i + x, t_n + t) &\approx \frac{1}{2}(pa(x - (M + 1)t) + ua(x - (M + 1)t)) \\ &\quad + \frac{1}{2}(pa(x - (M - 1)t) - ua(x - (M - 1)t)).\end{aligned}\tag{5}$$

The approximate local solution (5) is a function of  $x$  and  $t$ , and it is an exact solution to (1) with the local interpolants (3) as initial data, so that (5) is an exact local propagator for (1) and correctly incorporates the dynamics of (1). The fundamental viewpoint of this method is to approximate the solution of the system instead of particular terms in the governing equations. This fundamental idea is seen in the use of Riemann solvers for problems with shocks [2], and in the use of local Fourier decompositions and separation of variables to develop algorithms for a wide variety of problems [1]. The approximate local solution (5) is used to obtain a computational algorithm at the stencil center, with

$$\begin{aligned}
u(x_i, t_n + k) &\approx u_i^{n+1} = \frac{1}{2}(ua(-(M+1)k) + pa(-(M+1)k)) \\
&\quad + \frac{1}{2}(ua(-(M-1)k) - pa(-(M-1)k)), \\
p(x_i, t_n + k) &\approx p_i^{n+1} = \frac{1}{2}(pa(-(M+1)k) + ua(-(M+1)k)) \\
&\quad + \frac{1}{2}(pa(-(M-1)k) - ua(-(M-1)k)).
\end{aligned} \tag{6}$$

Algorithm (6) uses the exact local propagator (5) with approximate local data (3), so that the time evolution introduces no new error, but merely propagates what has been introduced by the interpolation. Note that finite difference forms are not specified, and that (6) represents a family of algorithms dependant upon the interpolant (3). If order  $D$  interpolants are used for a particular realization of (6), then the algorithm will have accuracy of order  $D$  in both space and time. There are several interpretations of algorithm (6). It may be viewed as an application of the Method of Characteristics, since in this case, the general solution form for the exact local propagator is obtained by this method. If the interpolation expansion coefficients are viewed as spatial derivatives (4), then the local solution in space and time (5) can be viewed as a Cauchy-Kowaleskaya expansion. The algorithm can also be reformulated as a truncated Taylor series expansion in the time step size  $k$ , with

$$\begin{aligned}
u_i^{n+1} &= u_0 - k(p_1 + Mu_1) + k^2(2Mp_2 + (M^2 + 1)u_2) \\
&\quad - k^3((1 + 3M^2)p_3 + M(3 + M^2)u_3) + \dots, \\
p_i^{n+1} &= p_0 - k(u_1 + Mp_1) + k^2(2Mu_2 + (M^2 + 1)p_2) \\
&\quad - k^3((1 + 3M^2)u_3 + M(3 + M^2)p_3) + \dots,
\end{aligned} \tag{7}$$

where the grid ratio  $\lambda = \frac{k}{h}$  is implicit in (7), since the coefficients  $u_\alpha$  and  $p_\alpha$  include the factor  $h^{-\alpha}$  for space step size  $h$ . A truncated Taylor expansion in time can be viewed as a generalized Lax-Wendroff method [8]. Algorithm (6) can also be reformulated as a conventional finite difference method, since the underlying spatial interpolations use local polynomials. Further details are in [3].

We will introduce four relatively conventional realizations of algorithm (6), with central stencils for the interpolants (3), and with values for  $u$  and  $p$  at each grid point. These algorithms have three, five, seven, and nine point central stencils, they are second, fourth, sixth, and eighth order accurate in both space and time, and they are referred to as the "c3o0ex," "c5o0ex," "c7o0ex," and "c9o0ex" methods, respectively. These four algorithms are all single step explicit methods with dispersive truncation errors, and each is stable for  $\frac{k}{h} \leq \frac{1}{1+M}$ . Grid refinement data for these algorithms is presented in Figure 1, confirming the order of accuracy of each of the four methods. This grid refinement data is discussed in more detail with data from an additional class of high resolution Hermitian methods that are introduced below.

Stable high order boundary treatments can be derived if these algorithms are viewed as Cauchy-Kowaleskaya expansions. The interpolant from the data on a single stencil next to a boundary is used as initial data for the approximation of the evolution of the solution over the interval from the stencil center to the boundary point. Values for  $u$  and  $p$  are computed at each grid point that is not on the boundary in this interval, and outgoing Riemann variables are computed at the boundary point. The boundary treatment for the c9o0ex method must be modified for stability, and it uses a truncated eight point boundary stencil with additional data at the boundary point. Table 1 presents results for a simulation with initial data  $u(x, 0) = 0$  and  $p(x, 0) = \text{Sin}(\pi x)$ , for  $-1 \leq x \leq 1$ , with  $\lambda = 0.8$  and  $M = 0$ . For the boundary treatment,  $u+p$  is computed at the  $x = 1$  boundary,  $u - p$  is computed at the  $x = -1$  boundary, and typical CFD boundary conditions are imposed, with  $u(-1, t)$  and  $p(1, t)$  given. The data in Table 1 shows that the propagation algorithms and boundary treatments are stable with from second to eighth order accuracy in both time and space. Further details are in [3].

TABLE 1: Data From Explicit Algorithms with Boundary Treatments  
Maximum Error in  $u$  or  $p$  at  $t = 10$

$\frac{2}{h}$	$n_{10}$	c3d0ex	c5d0ex	c7d0ex	c9d0ex
8	50	1.87D-01	1.17D-02	1.01D-02	8.65D-05
16	100	4.57D-02	9.98D-04	5.26D-05	8.50D-07
32	200	1.32D-02	8.11D-05	8.79D-07	4.69D-09
64	400	3.50D-03	5.54D-06	1.29D-08	2.09D-11
128	800	8.94D-04	3.58D-07	1.89D-10	3.75D-13
256	1600	2.25D-04	2.27D-08	3.35D-12	4.91D-13
512	3200	5.66D-05	1.43D-09	1.89D-12	
1024	6400	1.42D-05	9.12D-11		

### 3. Hermitian Algorithms

We will introduce a second family of algorithms for (1), which use the exact local propagator (5), but which are distinguished from the relatively conventional

algorithms introduced above by the use of Hermitian interpolants for (3). This particular family of algorithms uses and computes values for  $u$  and  $p$ , and for their spatial derivatives. Various orders of spatial derivative data are required at each grid point, depending upon the particular algorithm. If these algorithms are viewed as approximate local solutions for  $u$  and  $p$ , or as Cauchy-Kowaleskaya expansions that are locally defined as functions of  $x$  and  $t$ , then they can be differentiated in  $x$  to provide local solutions for the spatial derivatives that are consistent with the solutions for  $u$  and  $p$ . The local derivative solutions obtained by differentiating (5) in  $x$  are used to obtain computational forms for the spatial derivatives that are analogous to and consistent with the computational forms (6) for  $u$  and  $p$ . These forms may be expressed as Taylor series time expansions in  $k$ , with

$$\begin{aligned}
ux_i^{n+1} &= u_1 - 2k(p_2 + Mu_2) + 3k^2(2Mp_3 + (M^2 + 1)u_3) \\
&\quad - 4k^3((1 + 3M^2)p_4 + M(3 + M^2)u_4) + \dots, \\
px_i^{n+1} &= p_1 - 2k(u_2 + Mp_2) + 3k^2(2Mu_3 + (M^2 + 1)p_3) \\
&\quad - 4k^3((1 + 3M^2)u_4 + M(3 + M^2)p_4) + \dots,
\end{aligned} \tag{8}$$

and with similar forms for second and higher derivatives. The interpolation step for the Hermitian algorithms only produces interpolants (3) for the functions  $u$  and  $p$ , using local polynomial approximations and the Method of Undetermined Coefficients with all of the data at each grid point on a given stencil. Note that there is no separate consideration given to interpolation or propagation for the derivative data.

Hermitian algorithms with central stencils have stability problems if the maximum order of accuracy is obtained. Stable algorithms with exact propagators can be obtained on alternating grids offset by a half mesh width. We will introduce four algorithms which use staggered two point grids, and three algorithms which use staggered four point grids. On staggered two point grids, if just  $u$  and  $p$  are used, then a first order exact propagator algorithm is obtained, which we will denote by "c3o0s2." If up to first, second, or third derivatives are used in addition to  $u$  and  $p$ , then stable algorithms are obtained which are third, fifth, or seventh order accurate in space and time, referred to as the "c3o1s2," "c3o2s2," and "c3o3s2" methods, respectively. On staggered four point grids, if just  $u$  and  $p$  are used, then a third order exact propagator algorithm is obtained, which we will denote by "c5o0s2." If up to first or second derivatives are used in addition to  $u$  and  $p$ , then stable algorithms are obtained that are seventh or eleventh order accurate in space and time, referred to as the "c5o1s2," and "c5o2s2" methods, respectively. These seven algorithms are all single step explicit methods on staggered grids with diffusive truncation errors, and each is stable for  $\frac{k}{h} \leq \frac{1}{1+M}$ , where the staggered half step size is  $\frac{k}{2}$ . Note that two half time steps can be composed to produce single step algorithms with time step size  $k$ . The full time step algorithms resulting from staggered two point grids can be formulated as single step methods on a

central three point stencil, and the algorithms from staggered four point grids as single step methods on a central seven point stencil. Other combinations are possible, such as composing the c3o3s2 and c5o1s2 methods, to produce a single step explicit seventh order Hermitian method on a single central five point stencil. It will be shown below that some of these Hermitian algorithms have extraordinary resolution and accuracy.

#### 4. Numerical Experiments with the 1D Algorithms

Numerical experiments will be conducted with the eleven algorithms that have been introduced above, by computing solutions to (1) with initial data  $u(x, 0) = 0$  and  $p(x, 0) = \text{Sin}(\pi x)$  for  $-1 \leq x \leq 1$ , and with  $\lambda = 0.8$  and  $M = 0$ . Periodic boundaries will be imposed, with  $p(-1, t) = p(1, t)$  and  $u(-1, t) = u(1, t)$  for  $0 \leq t$ . Note that the initial data has wavelength and period 2. Figure 1 shows the  $\log_{10}$  of the maximum absolute error at  $t = 10$  in  $u$  or  $p$  versus the  $\log_2$  of the number of grid points per wavelength. This data corroborates the order of accuracy of each of the methods. Note that the first order c3o0s2 method is incapable of producing accurate results with any reasonable resolution, and that the second order c3o0ex or Lax-Wendroff method requires 64 grid points per wavelength to produce  $O[10^{-2}]$  accuracy after five periods. Note also that the c7o0ex and c9o0ex central methods show a greater sensitivity to roundoff errors than the high order Hermitian methods. An interesting feature of Figure 1 is the data at the coarsest resolution, with 4 grid points per wavelength. At this resolution, the simulations with the c3o2s2, c3o3s2, c5o1s2 and c5o2s2 algorithms have errors that range from  $O[10^{-2}]$  to  $O[10^{-6}]$ , in contrast with errors that range from  $O[1]$  to  $O[10^{-1}]$  for the other algorithms. An  $O[10^{-6}]$  error after five periods of propagation with four grid points per wavelength is exceptionally high resolution. Notice also the relative errors from methods which are of similar order. The c5o0s2 and c3o1s2 algorithms are both third order, but the c3o1s2 algorithm produces lower errors by about one order of magnitude at each grid resolution. The c5o1s2 and c3o3s2 algorithms are both seventh order, but the c3o3s2 algorithm produces lower errors by about two orders of magnitude at each grid resolution. The conventional sixth order c7o0ex and eighth order c9o0ex methods have larger errors at each grid level than the fifth order c3o2s2 and seventh order c5o1s2 Hermitian methods, respectively. In these comparisons of algorithms with similar orders of accuracy, the algorithm which produces lower errors has higher resolution from using more derivative information at each grid point. The accuracy of a numerical algorithm is determined by both its order of accuracy and its resolving power.

The periodic problem which produces Figure 1 is also used for propagation with 8 grid points per wavelength out to  $t = 10$ ,  $t = 1,000$ , and  $t = 100,000$ . The data is presented in Table 2, where  $O[1]$  errors are marked by asterisks, and where the algorithms are ranked by order. Note in Table 2 for  $t = 10$ , that the error data from the sixth order c7o0ex method is two orders of magnitude higher



than from the fifth order c3o2s2 method, and that the error from the eighth order c9o0ex method is two orders of magnitude higher than from the seventh order c5o1s2 method, and four orders of magnitude higher than from the seventh order c3o3s2 method. A similar comparison is also seen in Table 2 for the data from  $t = 1,000$  and  $t = 100,000$ . These comparisons once again show that both the order of accuracy and the resolution of a numerical algorithm determine its accuracy. Note that the only algorithms which produce errors that are not  $O[1]$  at  $t = 100,000$  are the high order and high resolution c5o1s2, c3o3s2 and c5o2s2 methods. These results also show that far field may be redefined by several orders of magnitude, and that efficient propagation to a truly far field requires methods which are both high order and high resolution.

TABLE 2: Long Time Simulations For Each Algorithm  
Maximum Error in  $u$  or  $p$  at Various Times

Method	Order	$t = 10$	$t = 1,000$	$t = 100,000$
c3o0s2	1	*****	*****	*****
c3o0ex	2	6.76D-01	*****	*****
c5o0s2	3	2.42D-01	*****	*****
c3o1s2	3	3.97D-02	*****	*****
c5o0ex	4	9.14D-02	6.74D-01	*****
c3o2s2	5	2.27D-04	2.26D-02	*****
c7o0ex	6	1.11D-02	1.04D-01	*****
c5o1s2	7	4.52D-05	4.55D-03	3.75D-01
c3o3s2	7	6.74D-07	6.78D-05	6.75D-03
c9o0ex	8	1.39D-03	1.38D-02	*****
c5o2s2	11	9.33D-10	9.37D-08	9.37D-06

The superior accuracy of the high order and high resolution methods is obtained by using more complex algorithms, with more variables and equations, and with more operations per grid point per time step. It is natural to ask whether or not the simpler, more conventional algorithms are more or less efficient than the Hermitian algorithms. We will address this issue by replotting the data from Figure 1, this time with the  $\log_{10}$  of the maximum error at  $t = 10$  on the vertical axis, and the  $\log_{10}$  of the total number of multiplications required for the entire simulation out to  $t = 10$  on the horizontal axis. This data is presented in Figure 2. It has been shown in [7] that relative computational efficiency increases with order of accuracy, and that the relative efficiency increase as the error tolerance is lowered, and as the simulation time is increased. The data in Figure 2 from our methods clearly shows that for every level of error the efficiency in meeting that error tolerance increases with the order of accuracy of the algorithm, in spite of the fact that algorithms of radically different type are being used. In particular, the Hermitian algorithms not only use more degrees of freedom of data in order to

increase order of accuracy, but they also solve more equations for more variables, and yet they are still more efficient than the less accurate algorithms.

## 5. Normal Mode Analysis in 1D

The special properties of the full discretization Hermitian algorithms can be shown by a normal mode analysis. We will consider Hermitian algorithms applied to the linear first order wave equation

$$\frac{\partial u}{\partial t} + M \frac{\partial u}{\partial x} = 0. \quad (9)$$

The local exact propagator is defined by the Method of Characteristics. A normal mode for (9) can be written in local coordinates as

$$u(x, t) = \alpha \text{Exp}\left[i\theta \frac{(x - Mt)}{h}\right], \quad (10)$$

with amplitude  $\alpha$ , frequency  $\theta \in [0, \pi]$ , and with space mesh size  $h$ . The symbol  $\rho$  for the algorithm can be written in terms of the normal mode (10), with

$$\rho(\lambda, \theta) = \frac{u^{n+1}}{u^n}, \quad (11)$$

where  $u^{n+1}$  is obtained from the algorithm with the normal mode  $u^n$  as the known solution at  $t_n$ , and where  $\lambda = \frac{Mh}{k}$  is the CFL number. Spatial derivative data is needed by the Hermitian algorithms at each grid point, and since the normal mode (10) is perfectly known as a function of  $x$  at  $t_n$ , we will simply take and use its derivative values for this data. This heuristic procedure is intended to obtain qualitative insights about the algorithms, and is not intended as a rigorous stability analysis, which would treat the total system of  $u$  and its spatial derivatives with independent error modes expected in each. From (11) we obtain the norm

$$\|\rho(\lambda, \theta)\| = (\text{Re}[\rho]^2 + \text{Im}[\rho]^2)^{\frac{1}{2}}, \quad (12)$$

and the phase change per time step

$$pc(\lambda, \theta) = \text{Cos}^{-1}\left[\frac{\text{Re}[\rho]}{\|\rho(\lambda, \theta)\|}\right]. \quad (13)$$

Note that we have varied from [10] by using a definition of phase change in terms of  $\text{Cos}^{-1}$  rather than  $\text{Tan}^{-1}$ . For the phase speed properties of an algorithm, we will use the normalized relative phase change per time step

$$rpc(\lambda, \theta) = \frac{1}{\lambda\theta} \text{Cos}^{-1}\left[\frac{\text{Re}[\rho]}{\|\rho(\lambda, \theta)\|}\right], \quad (14)$$

rather than the phase change per time step (13).

All of the Hermitian methods with two point staggered grids were applied to the wave equation (9), and single step forms of the algorithms were obtained on three point central stencils. The norm of the amplification factor (12) and the relative phase change per time step (14) of these single step algorithm forms have been obtained numerically from an analytical expression. The most accurate of these four methods is the seventh order high resolution c3o3s2 algorithm, which uses  $u$  and its first through third spatial derivative at every grid point. The norm of the amplification factor (12) for the c3o3s2 algorithm is plotted in Figure 3(a) as a function of the wave number  $\theta \in [0, \pi]$  and the CFL number  $\lambda \in [0, 1]$ . Note that Figure 3(a) shows the norm of the amplification factor as less than or equal to 1 in the specified parameter range. The most dissipated behaviour is in the limit at  $\theta = \pi$ , and the norm of the amplification factor at  $\theta = \pi$  is plotted as a function of  $\lambda$  in Figure 3(b). Figure 3(b) shows the norm of the amplification factor varying between approximately 0.9988 and 1 at  $\theta = \pi$ . The relative phase change per time step (14) for the c3o3s2 algorithm is plotted in Figure 4(a), also as a function of the wave number  $\theta \in [0, \pi]$  and the CFL number  $\lambda \in [0, 1]$ . The most dispersed behaviour is in the limit at  $\theta = \pi$ , and the relative phase change at  $\theta = \pi$  is plotted as a function of  $\lambda$  in Figure 4(b). Figure 4(b) shows the relative phase change per time step varying between approximately 0.9997 and 1.0002 at  $\theta = \pi$ . The amplification factor and relative phase change per time step plots in Figures 3 and 4 are extraordinary, and show truly "spectral like" qualities for the c3o3s2 algorithm. Recall from the numerical experiments reported above, that the c5o2s2 algorithm is of even higher resolution than the c3o3s2 algorithm.

## 6. Linearized Euler Equations in 2D

The algorithms that have been developed in one space dimension have all used the Method of Characteristics to obtain an exact solution form that propagates a local spatial interpolant. The Method of Characteristics cannot be used in multiple space dimensions for nondiagonalizable hyperbolic systems. We will consider the linearized Euler equations in two space dimensions in order to indicate an approach for developing exact local propagators in multiple space dimensions. Consider the linearized Euler equations for the nondimensionalized isentropic case,

$$\begin{aligned} \frac{\partial u}{\partial t} + U \frac{\partial u}{\partial x} + V \frac{\partial u}{\partial y} + \frac{\partial p}{\partial x} &= 0, \\ \frac{\partial v}{\partial t} + U \frac{\partial v}{\partial x} + V \frac{\partial v}{\partial y} + \frac{\partial p}{\partial y} &= 0, \\ \frac{\partial p}{\partial t} + U \frac{\partial p}{\partial x} + V \frac{\partial p}{\partial y} + \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} &= 0, \end{aligned} \tag{15}$$

where  $(U, V)$  is the constant mean convection velocity in Mach number,  $p$  is the

pressure disturbance, and  $(u, v)$  is the velocity disturbance. This two dimensional system is nondiagonalizable, with wave propagation along characteristic surfaces.

We will briefly describe the development of a second order explicit algorithm for (15) on a symmetric  $3 \times 3$  stencil. A second order local interpolant to  $u$  in  $x$  and  $y$  at  $t_n$  can be written as

$$\begin{aligned} u(x_i + x, y_j + y, t_n) \approx ua(x, y) = & u_{0,0} + u_{1,0}x + u_{2,0}x^2 \\ & + (u_{0,1} + u_{1,1}x + u_{2,1}x^2)y \\ & + (u_{0,2} + u_{1,2}x + u_{2,2}x^2)y^2, \end{aligned} \quad (16)$$

with similar interpolants for  $v$  and  $p$ . The expansion coefficients are simultaneously obtained by the Method of Undetermined Coefficients, and can be interpreted as spatial derivatives, with

$$u_{\alpha,\beta} = \frac{1}{\alpha!\beta!} \frac{\partial^{\alpha+\beta} ua}{\partial x^\alpha \partial y^\beta} \approx \frac{1}{\alpha!\beta!} \frac{\partial^{\alpha+\beta} u}{\partial x^\alpha \partial y^\beta}(x_i, y_j, t_n). \quad (17)$$

Notice that there are up to fourth order cross derivative terms in the interpolant (16). Exact polynomial solution forms for the linearized Euler equations (15) can be derived by substituting the expansion forms

$$\begin{aligned} u(x_i + x, y_j + y, t_n + t) \approx ua(x, y, t) &= \sum_{\alpha,\beta=0}^2 \sum_{\gamma=0}^4 u_{\alpha,\beta,\gamma} x^\alpha y^\beta t^\gamma, \\ v(x_i + x, y_j + y, t_n + t) \approx va(x, y, t) &= \sum_{\alpha,\beta=0}^2 \sum_{\gamma=0}^4 v_{\alpha,\beta,\gamma} x^\alpha y^\beta t^\gamma, \\ p(x_i + x, y_j + y, t_n + t) \approx pa(x, y, t) &= \sum_{\alpha,\beta=0}^2 \sum_{\gamma=0}^4 p_{\alpha,\beta,\gamma} x^\alpha y^\beta t^\gamma, \end{aligned} \quad (18)$$

into system (15), and obtaining all the terms with  $\gamma \neq 0$  by requiring system (15) to be satisfied for all  $x, y$  and  $t$ . Coefficients with  $\gamma \neq 0$  are equivalent to time derivatives, and the resulting polynomial solutions are expressed entirely in terms of the spatial expansion coefficients. Note that there will be third and fourth order time terms present in the exact propagator solution forms. This procedure for obtaining an exact solution form is equivalent to the Cauchy-Kowaleskaya procedure [6]. The exact polynomial solution forms from (18) give exact propagator algorithms with correct multidimensional wave dynamics for the local spatial interpolants (16) as initial data. Since a biquadratic interpolant is used, the resulting algorithm will have  $O[h^2 + k^2]$  truncation error, and it is dispersive. This procedure has been used with other symmetric stencils and interpolants to produce algorithms with fourth and sixth order accuracy in both space and time (see [3] for

details). Because a local exact propagator is used to develop these algorithms, the correct propagation of information along characteristic surfaces is automatically incorporated in the local approximation of the solution, so that this approach can be said to generalize the Method of Characteristics to nondiagonalizable systems in multiple space dimensions.

We have implemented new boundary conditions developed by Hagstrom [5] as algorithms that are compatible with our propagation methods for the linearized Euler equations in two space dimensions. Hagstrom actually provides a sequence of local approximations on the artificial boundary that are defined by a variable number of auxiliary functions. Each of the approximate problems are strongly well posed, and the approximate solutions converge exponentially to the exact solution on the open domain as the number of auxiliary functions increases [5]. As an example of these boundary conditions, consider the case of a subsonic mean flow in the positive  $x$  direction, with normal Riemann variables  $r = u + p$  and  $l = u - p$ . The function  $r$  is outgoing at an artificial boundary on the right, and is viewed as being defined on the interior and the surface, while the function  $l$  is incoming and is viewed as being defined only on the surface. The boundary surface values of  $r$  and  $v$  are obtained essentially in the same way as the outgoing Riemann variables in the one dimensional case considered above, with the algorithm forms being interpreted as Cauchy-Kowaleskaya expansions in space and time, and all relevant solution values obtained over a common boundary stencil. The function  $l$  is obtained with auxiliary functions  $f_j$  and  $g_j$ , from the system

$$\begin{aligned} \frac{\partial l}{\partial t} + V \frac{\partial l}{\partial y} - U \frac{\partial v}{\partial y} + \sum_{j=1}^m (f_j + g_j) &= 0, \\ \frac{\partial f_j}{\partial t} + \alpha_j \frac{\partial f_j}{\partial y} &= \frac{\beta_j}{2} \frac{\partial^2 (l - r)}{\partial y^2}, \\ \frac{\partial g_j}{\partial t} - \alpha_j \frac{\partial g_j}{\partial y} &= \frac{\beta_j}{2} \frac{\partial^2 (l - r)}{\partial y^2}, \end{aligned} \quad (19)$$

where  $\alpha_j = \sqrt{1 - U^2} \cos(\frac{j\pi}{2m+1}) + V$ , and  $\beta_j = -\frac{1-U^2}{2m+1} \sin^2(\frac{j\pi}{2m+1})$ . Note that system (19) is forced by the interior solution propagating across the artificial boundary in the form of the  $\frac{\partial v}{\partial y}$  term in the equation for  $l$ , and the  $\frac{\partial^2 r}{\partial y^2}$  terms in the equations for the auxiliary functions. Note also that this boundary system does not require assumptions about solution form or source location. In practice, we have used this condition with disturbance data entirely contained within the computational domain, and with boundary data initialized as 0. Similar systems are defined on the left hand artificial boundary (see [4], [5] for details). Boundary condition (19) and its left hand analog have been implemented in both second and fourth order algorithms, and numerical experiments with  $m = 2$  auxiliary functions have shown no visible evidence of reflection (see [4] for details).

## 7. Conclusions

Single step methods with high order accuracy in both space and time are shown and compared. A particular class of methods using Hermitian interpolation on alternating grids has shown both high order accuracy and spectral like high resolution. The high order and high resolution methods are more efficient than the less accurate methods by orders of magnitude, even though the high order and high resolution methods are considerably more complex. Calculation to the "far field" can be redefined as propagation to more than  $O[10^6]$  wavelengths or periods. Both high order accuracy and high resolution methods are required to compute to a true far field, with an example algorithm producing  $O[10^{-4}]$  errors after  $5 \times 10^5$  periods of propagation with eight grid points per wavelength. Algorithms for the linearized Euler equations in 2D are discussed, with propagation along characteristic surfaces, which generalizes the Method of Characteristics to nondiagonalizable Hyperbolic systems. Unobtrusive artificial boundary conditions are indicated.

## Bibliography

- [1] C-J Chen, H. Naseri-Neshat and P. Li, "The Finite Analytic Method," Iowa Institute of Hydrolic Research Report No 232, vol 1-4, The University of Iowa, (1980).
- [2] S. K. Godunov, "Finite Difference Method for Numerical Computation of Discontinuous Solutions of the Equation of Fluid Dynamics," *Mat. Sbornik.*, 47, 271 (1959).
- [3] 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 (September, 1995).
- [4] J. W. Goodrich, and T. Hagstrom, "Accurate Algorithms and Radiation Boundary Conditions for Linearized Euler Equations," AIAA paper 96-1660, (May, 1996).
- [5] T. Hagstrom, "On High Order Radiation Boundary Conditions," to appear in the proceedings of *The IMA Workshop on Computational Wave Propagation*, (Minneapolis, Minn., September, 1994).
- [6] A. Harten, B. Engquist, S. Osher and S. R. Chakravarthy, "Uniformly High Order Accurate Essentially Non-oscillatory Schemes, III," *J. Comput. Phys.*, 71, 231 (1987).
- [7] H. O. Kreiss and J. Olinger, "Comparison of accurate methods for the integration of hyperbolic equations," *Tellus*, 24, 199 (1972).
- [8] P. D. Lax and B. Wendroff, "Systems of Conservation Laws," *Comm. Pure Appl. Math*, 13, 217 (1960).
- [9] S. K. Lele, "Compact Finite Difference Schemes with Spectral like Resolution", *J. Comput. Phys.* 103, pp16-42 (1992).
- [10] R. Vichnevetsky, J. B. Bowels, *Fourier Analysis of Numerical Approximations of Hyperbolic Equations*, (SIAM, Philadelphia, 1982).

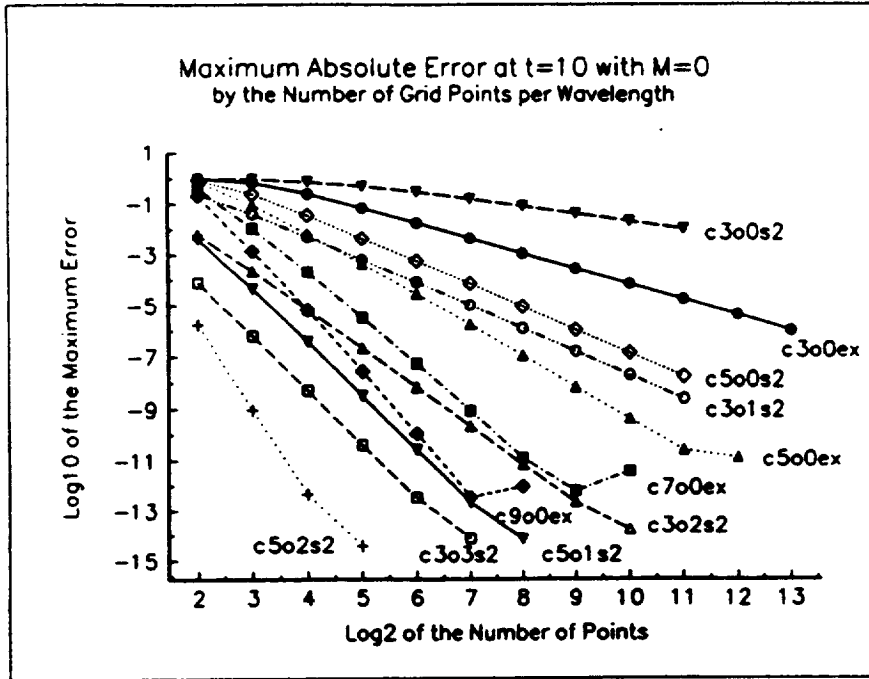


Figure 1: Maximum Error by the Number of Grid Points per Wavelength.

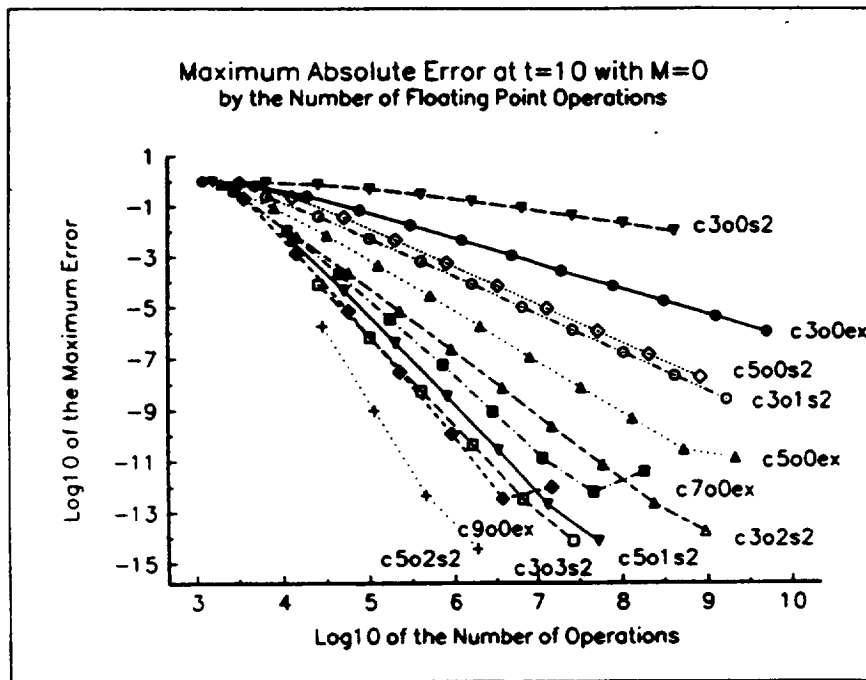


Figure 2: Maximum Error by the Total Number of Multiplications.

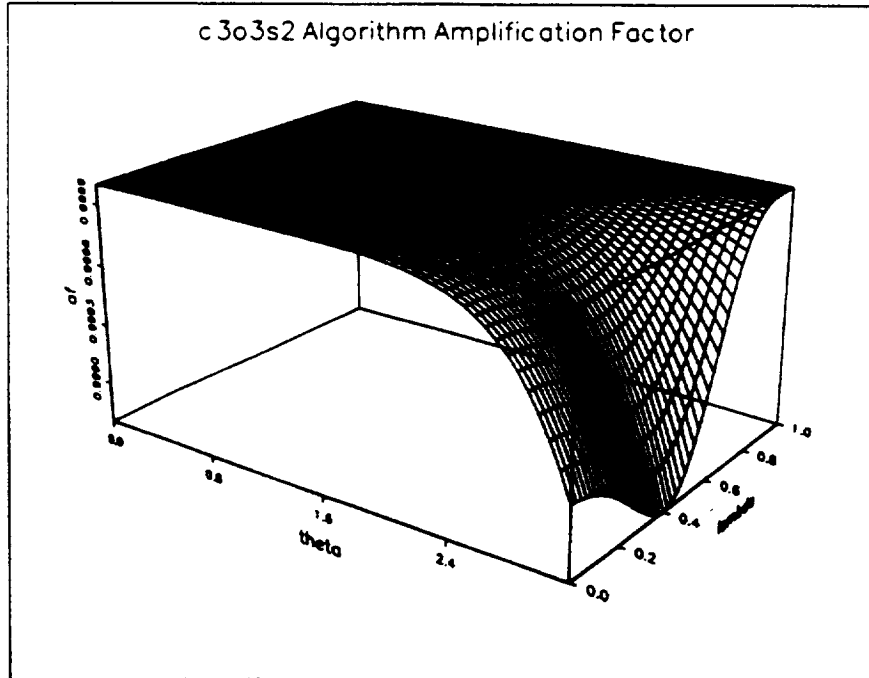


Figure 3(a): Amplification Factor for the c3o3s2 Algorithm.

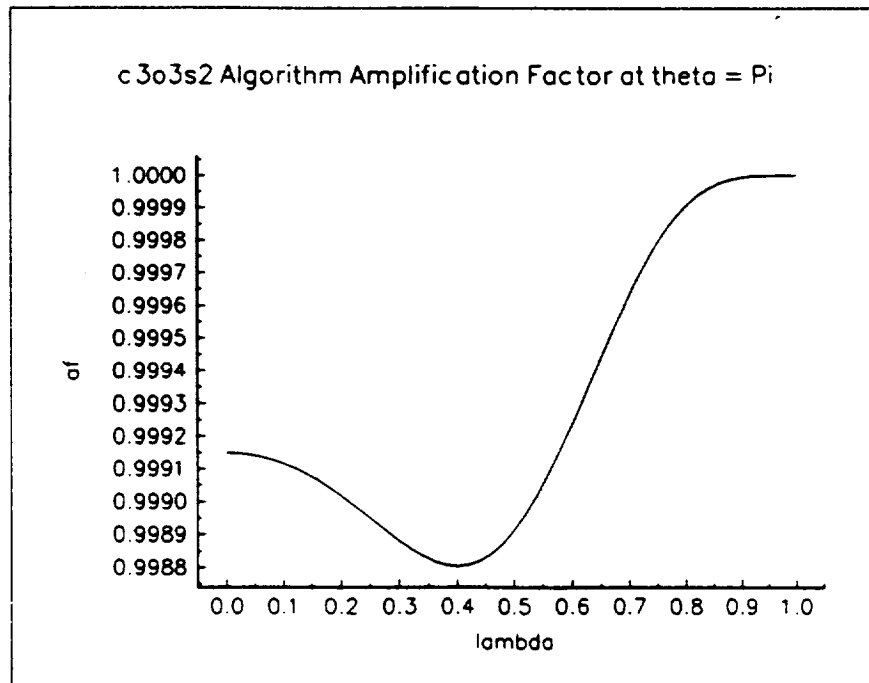


Figure 3(b): Amplification Factor at  $\theta = \pi$  for the c3o3s2 Algorithm.



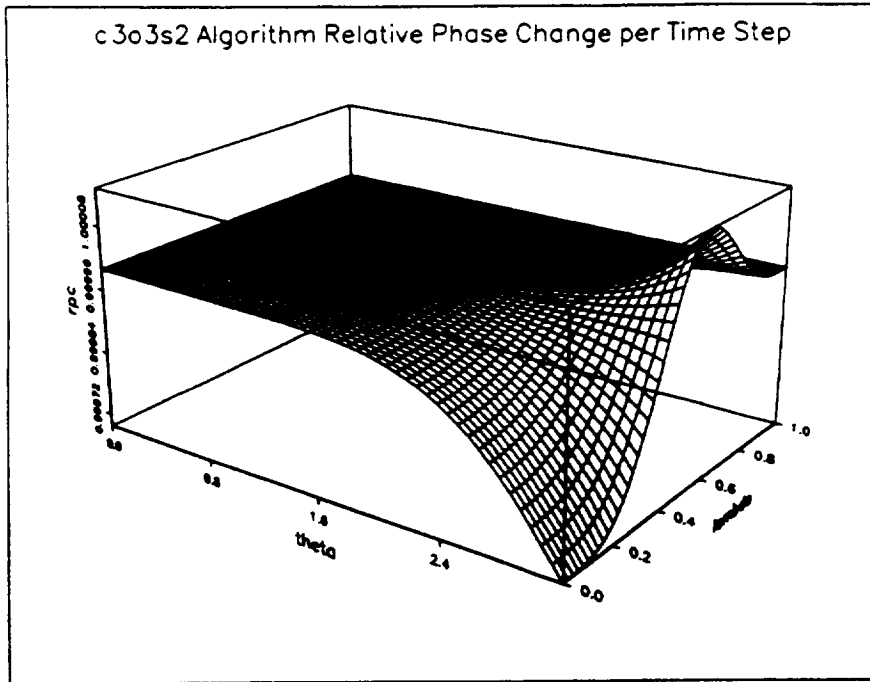


Figure 4(a): Relative Phase Change for the c3o3s2 Algorithm.

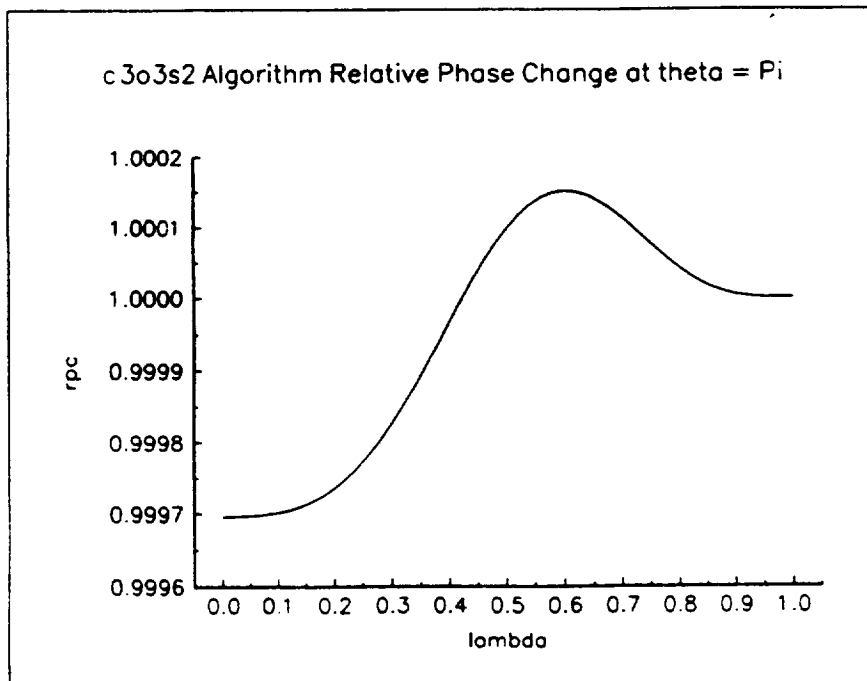


Figure 4(b): Relative Phase Change at  $\theta = \pi$  for the c3o3s2 Algorithm.

# REPORT DOCUMENTATION PAGE

Form Approved  
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

<b>1. AGENCY USE ONLY (Leave blank)</b>	<b>2. REPORT DATE</b> December 1996	<b>3. REPORT TYPE AND DATES COVERED</b> Technical Memorandum	
<b>4. TITLE AND SUBTITLE</b>  Accurate Finite Difference Algorithms		<b>5. FUNDING NUMBERS</b>  WU-538-03-11	
<b>6. AUTHOR(S)</b>  John W. Goodrich		<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b>  National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135-3191	
<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>  E-10564		<b>9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b>  National Aeronautics and Space Administration Washington, DC 20546-0001	
<b>10. SPONSORING/MONITORING AGENCY REPORT NUMBER</b>  NASA TM-107377		<b>11. SUPPLEMENTARY NOTES</b> Prepared for the Barriers and Challenges in Computational Fluid Dynamics Workshop sponsored by the Institute for Computer Applications in Science and Engineering and NASA Langley Research Center, Hampton, Virginia, August 5-7, 1996. Responsible person, John W. Goodrich, organization code 5940, (216) 433-5922.	
<b>12a. DISTRIBUTION/AVAILABILITY STATEMENT</b>  Unclassified - Unlimited Subject Categories 64, 71, and 07  This publication is available from the NASA Center for AeroSpace Information, (301) 621-0390.		<b>12b. DISTRIBUTION CODE</b>	
<b>13. ABSTRACT (Maximum 200 words)</b>  Two families of finite difference algorithms for computational aeroacoustics are presented and compared. All of the algorithms are single step explicit methods, they have the same order of accuracy in both space and time, with examples up to eleventh order, and they have multidimensional extensions. One of the algorithm families has spectral like high resolution. Propagation with high order and high resolution algorithms can produce accurate results after $O[10^6]$ periods of propagation with eight grid points per wavelength.			
<b>14. SUBJECT TERMS</b>  High resolution; High accuracy; Finite difference algorithms; Computational aeroacoustics		<b>15. NUMBER OF PAGES</b> 17	
		<b>16. PRICE CODE</b> A03	
<b>17. SECURITY CLASSIFICATION OF REPORT</b> Unclassified	<b>18. SECURITY CLASSIFICATION OF THIS PAGE</b> Unclassified	<b>19. SECURITY CLASSIFICATION OF ABSTRACT</b> Unclassified	<b>20. LIMITATION OF ABSTRACT</b>