A Conservative, Staggered-Grid Multidomain Method for the Euler Gas-Dynamics Equations

David A. Kopriva*

Abstract

We describe a new multidomain spectral collocation method for the solution of compressible flow problems that has features lacking in most current methods. Based on a staggered grid, it defines the solutions at the nodes of a Chebyshev Gauss quadrature rule and the fluxes at the nodes of a Chebyshev Gauss-Lobatto rule. The method is conservative and free-stream preserving. By way of numerical experiments, we show that it is exponentially accurate. A significant advantage of the method is that subdomain corners are not included in the approximation, making complex geometries easier to treat.

Key words: spectral methods, domain decomposition, compressible flows.

AMS subject classifications: 65P30, 76N99.

1 Introduction

In this paper, we describe a new multidomain spectral collocation method for the solution of inviscid compressible flow problems. The method is based on a staggered grid, analogous to fully staggered grids often used with finite difference methods. In our case, however, the solutions are defined at the nodes of the Chebyshev Gauss quadrature rule, while the fluxes are evaluated at the nodes of the Chebyshev Gauss-Lobatto rule. Staggered grid spectral approximations were first proposed for the solution of the incompressible Navier-Stokes equations (c.f. [1], pg. 234). Our grid will be identical to the fully staggered grid of Bernardi and Maday [2].

When applied to the Euler gas-dynamics equations, the staggered grid multidomain concept has many desirable features. First, like the cell averaged method of ref. [3], it is conservative. Thus, it should be possible to apply shock capturing techniques to the approximation. Only flux values are required at interfaces, not their derivatives, so discontinuous changes in the grids can occur across interfaces. The result is that subdomains can be defined independently of their neighbors, which makes the method geometrically flexible. Also, the interface conditions can be computed to the same temporal accuracy as the interior points, unlike the correction scheme methods [4]. Most important, in multiple space dimensions, the method does not include (the Gauss rules being open) the corners of subdomains. Thus the coding of the method does not require special cases at corners, and any number of subdomains can meet at a point without coding difficulty.

The paper is divided as follows. After the equations that we intend to solve are presented in Section 1, the approximation is described in Section 2 for problems in two space dimensions. In Section 3, we show that the method is both conservative and free-stream preserving. Section 4 provides three examples of the use of the method for two-dimensional problems. The first problem is that of a point source flow, for which there is an exact solution. We show that exponential accuracy is obtained for this problem. The next problem is a steady subsonic flow through an array of cylinders in a duct. We then solve a transonic flow in a two-dimensional converging-diverging nozzle and compare the results to experimental data. Concluding remarks are made in the final section.

2 The equations

In this paper, we describe the approximation of the Euler equations of gas-dynamics in conservative form,

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = 0, \tag{1}$$

*Supercomputer Computations Research Institute, The Florida State University, Tallahassee, FL 32306

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

where $\mathbf{Q}$ is the vector of solution unknowns and $\mathbf{F}(\mathbf{Q})$ and $\mathbf{G}(\mathbf{Q})$ are the advective flux vectors

$$\mathbf{Q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho e \end{bmatrix} \quad \mathbf{F} = \begin{bmatrix} \rho u \\ p + \rho u^2 \\ \rho uv \\ u(\rho e + p) \end{bmatrix} \quad \mathbf{G} = \begin{bmatrix} \rho v \\ \rho uv \\ p + \rho v^2 \\ v(\rho e + p) \end{bmatrix} . \tag{2}$$

We assume the gas is perfect so that $\rho e = p/(\gamma - 1) + \rho(u^2 + v^2)/2$ and $\gamma = 1.4$. For axisymmetric problems, such as the transonic flow in the converging-diverging nozzle discussed later, we interpret x as the axial coordinate and y as the radial coordinate. In that case we add the vector

$$\mathbf{H} = \frac{1}{y} \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 \\ v(\rho e + p) \end{bmatrix} \tag{3}$$

to the right hand side of equation (1).

3 The staggered grid approximation

3.1 Notation

The staggered grid approximation computes the solution values and advective fluxes on different grids. Unlike the common approximation [1], which uses only the nodes of the Chebyshev Gauss-Lobatto quadrature as collocation points, the new method uses both the Gauss and the Gauss-Lobatto points. We denote the Lobatto points by X_j and the Gauss points by $\overline{X}_{j+1/2}$, defined by

$$\begin{aligned} X_j &= \tfrac{1}{2}\left(1 - \cos\left(\tfrac{j\pi}{N}\right)\right) \quad j = 0, 1, \ldots, N \\ \overline{X}_{j+1/2} &= \tfrac{1}{2}\left(1 - \cos\left(\tfrac{2j+1}{2N+2}\pi\right)\right) \quad j = 0, 1, \ldots, N-1 \end{aligned} \tag{4}$$

In equation (4), we have mapped the usual collocation points defined on $[-1, 1]$ to the more convenient unit interval. The overbar and half point notation for the Gauss points is used only for its value as an analogy to staggered grid finite difference methods. It is understood that the Gauss points do not lie halfway between the Lobatto points [1].

We also define two polynomial approximations, one for the Gauss grid, and one for the Lobatto grid. Let the space of polynomials of degree less than or equal to N be denoted P_N. Let $\ell_j(\xi) \in \mathsf{P}_N$ be the Lagrange interpolating polynomial

$$\ell_j(\xi) = \prod_{\substack{i=0 \\ i \neq j}}^{N} \left(\frac{\xi - X_i}{X_j - X_i} \right) \tag{5a}$$

defined on the Lobatto grid. On the Gauss grid, we define $h_{j+1/2} \in \mathsf{P}_{N-1}$ to be the polynomial

$$h_{j+1/2}(\xi) = \prod_{\substack{i=0 \\ i \neq j}}^{N-1} \left(\frac{\xi - \overline{X}_{i+1/2}}{\overline{X}_{j+1/2} - \overline{X}_{i+1/2}} \right) \tag{5b}$$

Finally, let Q_j be a grid point value on the Lobatto grid and $\overline{Q}_{j+1/2}$ be a value defined on the Gauss grid. Then we write the polynomials that interpolate these values as

$$Q(X) = \sum_{j=0}^{N} Q_j \ell_j(X) \tag{6a}$$

$$\overline{Q}(X) = \sum_{j=0}^{N-1} \overline{Q}_{j+1/2}\, h_{j+1/2}(X) . \tag{6b}$$

3.2 Mapping of the subdomains

In two space dimensions, we subdivide the computational domain, Ω, into multiple quadrilateral subdomains, Ω^k. We make two assumptions about the subdivision in this paper. First, we restrict subdomains to intersect only at a point or along an entire side. Thus, we do not consider geometrically non-conforming approximations here. Second, we assume that the subdomain boundaries do not vary in time.

Once defined, the individual subdomains are mapped independently onto the unit square by an isoparametric transformation. Let the vector function $\mathbf{g}(s)$, $0 \leq s \leq 1$ define a parametric curve. Define also the polynomial of degree N,

$$\Gamma(s) = \sum_{j=0}^{N} \mathbf{g}(s_j)\, \ell_j(s) \tag{7}$$

that interpolates $\mathbf{g}$ at the Gauss-Lobatto points, defined in the first part of equation (4). For each subdomain, Ω^k, we define four such polynomial curves, $\Gamma_m(s)$, $m = 1, 2, 3, 4$ that bound the subdomain. Finally, we map each Ω^k onto the unit square by the linear blending formula

$$\begin{aligned} \mathbf{x}^N(X, Y) = {} & (1 - Y)\Gamma_1(X) + Y\Gamma_3(X) + (1 - X)\Gamma_4(Y) \\ & + X\Gamma_2(Y) - \mathbf{x}_1(1 - X)(1 - Y) \\ & - \mathbf{x}_2 X(1 - Y) - \mathbf{x}_3 XY - \mathbf{x}_4(1 - X)Y , \end{aligned} \tag{8}$$

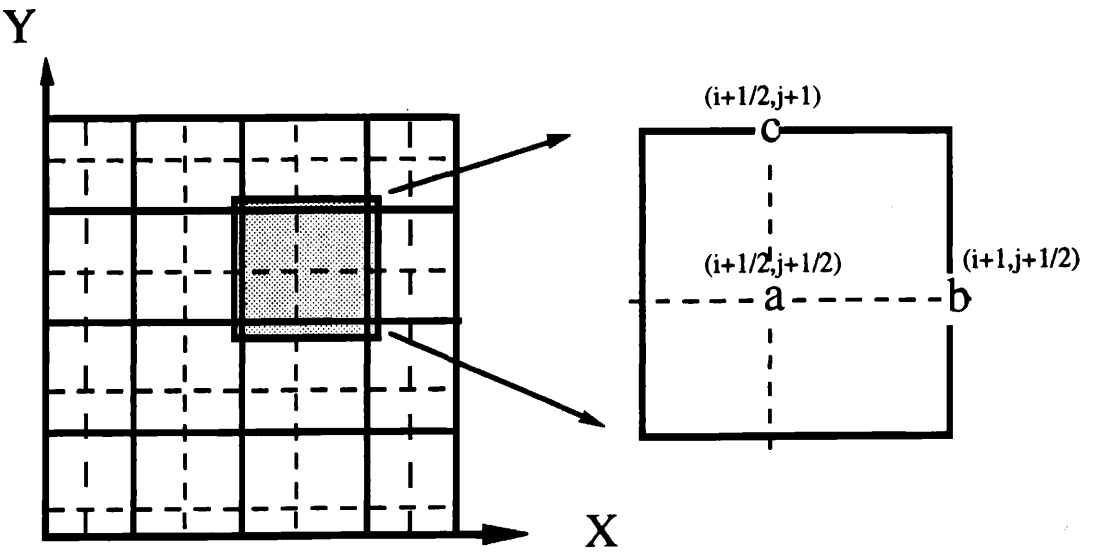

Figure 1: Diagram of the fully staggered grid in two space dimensions.

where the $\mathbf{x}_j$'s represent the locations of the corners of the subdomain. As an added restriction on the method, we will require that the polynomial orders on either side of an interface match, making the approximation functionally as well as geometrically conforming. This requirement means, essentially, that the grid points from either side coincide along an interface.

Under the mapping $\Omega^k \leftrightarrow [0,1] \times [0,1]$, the Euler equations (1) become

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{1}{J}\left[\frac{\partial \tilde{\mathbf{F}}}{\partial X} + \frac{\partial \tilde{\mathbf{G}}}{\partial Y}\right] \tag{9a}$$

where

$$\begin{aligned} &\tilde{\mathbf{F}} = y_Y^N \mathbf{F} - x_Y^N \mathbf{G} \quad \tilde{\mathbf{G}} = -y_X^N \mathbf{F} + x_X^N \mathbf{G} \\ &J(X,Y) = x_X^N y_Y^N - x_Y^N y_X^N \end{aligned} \tag{9b}$$

Since we assume that the subdomain boundaries do not move in time, we can write equation (9a) as

$$\frac{\partial \tilde{\mathbf{Q}}}{\partial t} + \frac{\partial \tilde{\mathbf{F}}(\mathbf{Q})}{\partial X} + \frac{\partial \tilde{\mathbf{G}}(\mathbf{Q})}{\partial Y} = 0 \tag{10}$$

where $\tilde{\mathbf{Q}} = J\mathbf{Q}$ and the fluxes are still defined in equation (9b).

3.3 Computation of the interior fluxes

A fully staggered grid is used in two space dimensions. A schematic of the grid on a single subdomain is shown in Figure 1. This grid is the same as the staggered grid proposed by Bernardi and Maday [2] for the solution of the incompressible Navier-Stokes equations. In what follows, we will ignore superscripts that denote which subdomain is being considered, unless necessary.

Points of type "a" in Figure 3 represent the Gauss/Gauss points $(\overline{X}_{i+1/2}, \overline{Y}_{j+1/2})$, $i = 0, 1, \ldots, N-1$, $j = 0, 1, \ldots, M-1$. The grid that results from these points is the tensor product of the one dimensional grid defined in equation (4). We approximate the solution and the transformation Jacobian at the Gauss/Gauss points, and denote them by $\overline{\mathbf{Q}}_{i+1/2,j+1/2}$ and $\overline{J}_{i+1/2,j+1/2} = J(\overline{X}_{i+1/2,j+1/2}, \overline{Y}_{i+1/2,j+1/2})$. From these, we compute the Gauss point values of $\tilde{\mathbf{Q}}_{i+1/2,j+1/2} = \overline{J}_{i+1/2,j+1/2}\overline{\mathbf{Q}}_{i+1/2,j+1/2}$. Finally, the interpolant of the solution through the Gauss points is a polynomial in $\mathsf{P}_{N-1,M-1} = \mathsf{P}_{N-1} \otimes \mathsf{P}_{M-1}$:

$$\tilde{\mathbf{Q}}(X,Y) = \sum_{i=0}^{N-1}\sum_{j=0}^{M-1} \tilde{\mathbf{Q}}_{i+1/2,j+1/2}\, h_{i+1/2}(X)\, h_{j+1/2}(Y) \tag{11}$$

The points of type "b" in Figure 1 form the Lobatto/Gauss grid whose points are written as $(X_i, \overline{Y}_{j+1/2})$, $i = 0, 1, \ldots, N$, $j = 0, 1, \ldots, M-1$. On this grid are evaluated the horizontal flux vector, $\tilde{\mathbf{F}}$ and the metric terms y_Y and x_Y. The metric terms are the analytical derivatives of the polynomial functions defined by equation (8) evaluated at these points. Then the horizon-

tal flux is the polynomial in $\mathsf{P}_{N,M-1}$ that passes through the grid point values

$$\begin{aligned}\tilde{\mathbf{F}}_{i,j+1/2} &= y_Y^N(X_i,\overline{Y}_{j+1/2})\,\mathbf{F}\left(\overline{Q}(X_i,\overline{Y}_{j+1/2})\right)\\ &\quad -x_Y^N(X_i,\overline{Y}_{j+1/2})\,\mathbf{G}\left(\overline{Q}(X_i,\overline{Y}_{j+1/2})\right)\,,\end{aligned}\tag{12}$$

In equation (12), $\overline{\mathbf{Q}}(X,Y)$ is the polynomial of the type (11) that passes through the points $\tilde{\mathbf{Q}}_{i+1/2,j+1/2}/\overline{J}_{i+1/2,j+1/2}$. Only the interior flux points are computed by equation (12). The boundary and interface flux definitions are described in the next sub-section.

The vertical flux and the derivatives y_X and x_X are defined on the Gauss/Lobatto grid, marked by "c" on Figure 1. The points on this grid are $(\overline{X}_{i+1/2},Y_j)$, $i=0,1,\ldots,N-1$, $j=0,1,\ldots,M$, and the vertical flux is the polynomial in $\mathsf{P}_{N-1,M}$ that passes through the points

$$\begin{aligned}\tilde{\mathbf{G}}_{i+1/2,j} &= -y_X^N(\overline{X}_{i+1/2},Y_j)\,\mathbf{F}\left(\overline{Q}(\overline{X}_{i+1/2},Y_j)\right)\\ &\quad +x_X^N(\overline{X}_{i+1/2},Y_j)\,\mathbf{G}\left(\overline{Q}(\overline{X}_{i+1/2},Y_j)\right)\,.\end{aligned}\tag{13}$$

Like equation (12), equation (13) is only applied to the calculation of interior point fluxes.

While it may appear that defining quantities on three different grids would lead to a significantly more complicated method than a single grid Lobatto approximation, this turns out not to be the case. First, the definition of the fluxes by equations (12) and (13) imply that the reconstruction procedure, i.e., the interpolation needed to compute the fluxes at the Lobatto points, is not a two-dimensional operation. Rather, it can be computed by a less expensive sequence of one-dimensional interpolations. The values of the solution vector that are required to compute the flux vectors are

$$\begin{aligned}\overline{\mathbf{Q}}(X_i,\overline{Y}_{j+1/2}) &= \sum_{i=0}^{N-1}\sum_{j=0}^{M-1}\overline{\mathbf{Q}}_{i+1/2,j+1/2}\\ &\quad\times\; h_{i+1/2}(X_i)h_{j+1/2}(\overline{Y}_{j+1/2})\\ &= \sum_{i=0}^{N-1}\overline{\mathbf{Q}}_{i+1/2,j+1/2}h_{i+1/2}(X_i)\end{aligned}\tag{14a}$$

and

$$\begin{aligned}\overline{\mathbf{Q}}(\overline{X}_{i+1/2},Y_j) &= \sum_{i=0}^{N-1}\sum_{j=0}^{M-1}\overline{\mathbf{Q}}_{i+1/2,j+1/2}\\ &\quad\times\; h_{i+1/2}(\overline{X}_{i+1/2})h_{j+1/2}(Y_j)\\ &= \sum_{j=0}^{M-1}\overline{\mathbf{Q}}_{i+1/2,j+1/2}h_{j+1/2}(Y_j)\end{aligned}\tag{14b}$$

since, by construction,

$$\begin{aligned}h_{m+1/2}(\overline{Y}_{j+1/2}) &= \delta_{m,j}\\ h_{n+1/2}(\overline{X}_{i+1/2}) &= \delta_{n,i}\end{aligned}\,,$$

where $\delta_{j,k}$ is the Kronecker delta function.

Next, when derivatives need to be evaluated at the Gauss/Gauss points, they can also be done as a sequence of one-dimensional derivatives. To compute the derivative approximations, the interpolants of the grid point values of the fluxes are differentiated, and the result is evaluated at the Gauss points. Computationally, this becomes

$$\begin{aligned}\left.\frac{\partial\tilde{\mathbf{F}}}{\partial X}\right|_{i+1/2,j+1/2} &= \sum_{n=0}^{N}\tilde{\mathbf{F}}_{n,j+1/2}\ell_n'(\overline{X}_{i+1/2})\\ \left.\frac{\partial\tilde{\mathbf{G}}}{\partial Y}\right|_{i+1/2,j+1/2} &= \sum_{m=0}^{N}\tilde{\mathbf{G}}_{i+1/2,m}\ell_m'(\overline{Y}_{j+1/2})\end{aligned}\tag{15}$$

The reconstruction and differentiation operations represented by equations (14) and (15) can both be computed by matrix multiplication.

The total work associated with the interpolation and differentiation operations in two space dimensions is twice that of a method that only uses the Lobatto grid. On the other hand, the method requires the same amount of work as the cell averaged method [3] in two space dimensions. The equivalence of the amount of work is due to the more complex nature of the reconstruction for the cell averaged method in two space dimensions.

3.4 Interface and boundary fluxes

To describe how to compute the interface and boundary conditions using the staggered grid approximation, we will refer to Figure 2, which schematically represents four subdomains and the locations at which solution and flux values are computed. Only the grid points near the boundaries are marked. The circles represent the solution values, which are located on the Gauss/Gauss grid. The locations of the horizontal flux values, $\tilde{\mathbf{F}}_{i,j+1/2}$, are represented by solid squares. The locations of the vertical flux values, $\tilde{\mathbf{G}}_{i+1/2,j}$, are marked by hollow squares. From the diagram, we see that along the vertical interfaces between subdomains 1 and 2 and between subdomains 3 and 4, only the horizontal fluxes need to be computed. Along horizontal interfaces, like those between subdomains 1 and 3, only the vertical flux needs to be computed. Because the grid is fully staggered, the coupling is through subdomain faces only, not through corners.

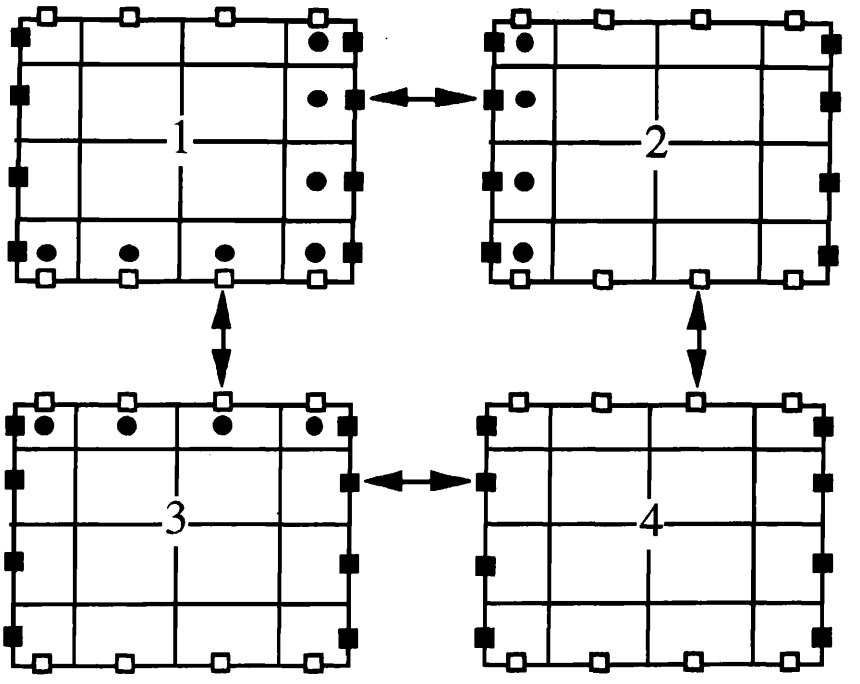

Figure 2: Diagram of four subdomains showing locations near interfaces where solutions and fluxes are computed. Symbols: circle – solution; solid square – $\tilde{\mathbf{F}}$; hollow square – $\tilde{\mathbf{G}}$.

Figure 2 indicates a significant advantage to the use of the fully staggered grid over the use of an unstaggered approximation, e.g. [4]. An unstaggered approximation requires special corner algorithms to be devised to compute corner solutions to ensure stable propagation of waves through the corners. The choice of bi-characteristics that determines the domains of dependence becomes more complex as the number of subdomains/boundaries that come together at a point increases. The staggered approximation does not include subdomain corners, so conditions do not have to be specified at corner points. Thus, any number of subdomains can come together at a point without the need for special point approximations and code.

The interpolation of the solution by equation (14) produces two solution values at an interface point, one from each of the two contributing subdomains. We do not expect these two values to coincide, except in the limit of infinite resolution. A single flux is calculated by solving an approximate Riemann problem that considers waves propagating normal to the interface. This normal wave approximation is common for finite difference approximations [5] and has been used for spectral approximations in [3]. Several solver choices are possible, but we have used Roe's [6] solver with the entropy fix. To illustrate the procedure, let us consider the situation along vertical interfaces shown in Figure 2. Formally, given the two states $\mathbf{Q}^L$ and $\mathbf{Q}^R$ that have been computed from the interpolation of the Gauss point values, Roe's approximate Riemann solver writes the horizontal flux as

$$\tilde{\mathbf{F}}(\mathbf{Q}^L, \mathbf{Q}^R) = \frac{1}{2}\left(\tilde{\mathbf{F}}(\mathbf{Q}^L) + \tilde{\mathbf{F}}(\mathbf{Q}^R)\right) - \frac{1}{2}\mathbf{R}|\Lambda|\mathbf{R}^{-1}(\mathbf{Q}^R - \mathbf{Q}^L) \tag{16}$$

where $\mathbf{R}$ is the matrix of the right eigenvectors of the Jacobian of $\tilde{\mathbf{F}}$, computed using the Roe-average of $\mathbf{Q}^L$ and $\mathbf{Q}^R$. The matrix Λ is the matrix of the eigenvalues of $\tilde{\mathbf{F}}$. This formula is modified to correct the entropy across sonic points. A detailed account can be found in [5].

Physical boundaries can be viewed as interfaces between the external flow and the computational region. Wall boundaries can be computed by imposing an opposing flow that enforces zero normal momentum flux across the interface. Subsonic inflow and outflow boundaries can be computed by replacing the solution that would have come from a neighboring subdomain by the free-stream values, if they are known. Thus, the flux at the boundary on the

left of Figure 2 would be computed by $\tilde{\mathbf{F}}(\mathbf{Q}_\infty, \mathbf{Q}^R)$. If the full state of the exterior flow is not known, an alternative procedure can be used such as that applied for the nozzle calculation below. Supersonic outflow boundaries require no extra conditions.

3.5 Discretization of the equations

From equations (11)–(15), with boundary and interface fluxes defined by equation (16), we can now define the semi-discrete approximation for the solution unknowns within a subdomain:

$$\left.\frac{d\tilde{\mathbf{Q}}}{dt}\right|_{i+1/2,j+1/2} + \left[\frac{\partial \tilde{\mathbf{F}}}{\partial X} + \frac{\partial \tilde{\mathbf{G}}}{\partial Y}\right]_{i+1/2,j+1/2} = \mathbf{0}, \quad \begin{array}{l} i = 0, 1, \ldots, N-1 \\ j = 0, 1, \ldots, M-1 \end{array} \tag{17}$$

Equation (17) is a system of ordinary differential equations that must be integrated in time to get the approximate solution values at the Gauss points. In principle, any common integration procedure can be used. We have chosen to use low storage Runge-Kutta methods that require only 2-N storage locations. In this paper, we consider only the computation of steady-state problems, for which the time discretization is only an iterative procedure, so it is sufficient to use the low-order mid-point rule. For each subdomain, we compute

$$\begin{aligned} \tilde{\mathbf{Q}}^{n+1/2}_{i+1/2,j+1/2} &= \tilde{\mathbf{Q}}^{n}_{i+1/2,j+1/2} \\ &\quad - \tfrac{\Delta t}{2}\left[\frac{\partial \tilde{\mathbf{F}}^n}{\partial X} + \frac{\partial \tilde{\mathbf{G}}^n}{\partial Y}\right]_{i+1/2,j+1/2} \\ &\quad i = 0, 1, \ldots, N-1 \\ &\quad j = 0, 1, \ldots, M-1 \end{aligned} \tag{18}$$

$$\begin{aligned} \tilde{\mathbf{Q}}^{n+1}_{i+1/2,j+1/2} &= \tilde{\mathbf{Q}}^{n}_{i+1/2,j+1/2} \\ &\quad - \Delta t\left[\frac{\partial \tilde{\mathbf{F}}^{n+1/2}}{\partial X} + \frac{\partial \tilde{\mathbf{G}}^{n+1/2}}{\partial Y}\right]_{i+1/2,j+1/2} \\ &\quad i = 0, 1, \ldots, N-1 \\ &\quad j = 0, 1, \ldots, M-1 \end{aligned}$$

where the superscript n denotes the time level. For the staggered grid approximation, this method appears to have a good balance between the time step required and temporal damping introduced by the scheme. With additional knowledge of the eigenvalue structure of the differentiation matrices, other choices might include schemes optimized for rapid convergence to steady-state, such as those discussed in [7].

4 Properties of the staggered grid approximation

The staggered grid approximation is both conservative and free-stream preserving. A net gain or loss of $\tilde{\mathbf{Q}}$ is determined only by a net gain or loss through the exterior boundaries. Also, if the solution is constant in space, then the solution must remain constant in time also, even in the presence of a spatially varying mapping.

We first show that the staggered grid approximation is conservative. It is sufficient to consider four subdomains as shown in Figure 2. Let the quadrature weights $w_{i+1/2}, \eta_{j+1/2}$ be defined so that

$$\int_0^1 \int_0^1 P dX dY = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} P_{i+1/2,j+1/2} w_{i+1/2} \eta_{j+1/2} \quad \forall P \in \mathsf{P}_{N-1,M-1} \tag{19}$$

By the exactness of the quadrature, the sum of equation (17) times $w_{i+1/2}\eta_{j+1/2}$ over all the points within a subdomain is

$$\begin{aligned} &\sum_{i,j=0}^{N-1,M-1} \left.\frac{d\tilde{\mathbf{Q}}}{dt}\right|_{i+1/2,j+1/2} w_{i+1/2}\eta_{j+1/2} \\ &= \int_0^1 \frac{d\tilde{\mathbf{Q}}}{dt} dX dY \\ &= \sum_{i,j=0}^{N-1} \left[\frac{\partial \tilde{\mathbf{F}}}{\partial X} + \frac{\partial \tilde{\mathbf{G}}}{\partial Y}\right]_{i+1/2,j+1/2} w_{i+1/2}\eta_{j+1/2} \\ &= \int_0^1 \int_0^1 \left[\frac{\partial \tilde{\mathbf{F}}}{\partial X} + \frac{\partial \tilde{\mathbf{G}}}{\partial Y}\right] dX dY \end{aligned} \tag{20}$$

Thus, for each subdomain,

$$\begin{aligned} \frac{d}{dt}\int_0^1 \int_0^1 \tilde{\mathbf{Q}} dX dY = &- \int_0^1 \tilde{\mathbf{F}}(1, Y) dY + \int_0^1 \tilde{\mathbf{F}}(0, Y) dY \\ &- \int_0^1 \tilde{\mathbf{G}}(X, 1) dX + \int_0^1 \tilde{\mathbf{G}}(X, 0) dX \end{aligned} \tag{21}$$

When equation (21) is summed over all subdomains, the interior integrals cancel so that only the boundary contributions remain:

$$\frac{d}{dt}\sum_{k=1}^{4}\int_0^1\int_0^1 \tilde{\mathbf{Q}}^k dX dY = \int_0^1 \left(\tilde{\mathbf{F}}^1(0,Y)+\tilde{\mathbf{F}}^3(0,Y)\right)dY - \int_0^1 \left(\tilde{\mathbf{F}}^2(1,Y)+\tilde{\mathbf{F}}^4(1,Y)\right)dY + \int_0^1 \left(\tilde{\mathbf{G}}^3(X,0)+\tilde{\mathbf{G}}^4(X,0)\right)dX - \int_0^1 \left(\tilde{\mathbf{G}}^1(X,1)+\tilde{\mathbf{G}}^2(X,1)\right)dX \tag{22}$$

The staggered grid approximation is also free-stream preserving, which means that the isoparametric spatial mappings do not introduce false source terms. It is sufficient to consider the approximation within one subdomain, since all derivatives are computed locally by subdomain. If we take $\mathbf{F}(\mathbf{Q}) = \mathbf{G}(\mathbf{Q}) = \mathbf{1}$, then the approximation (17) becomes

$$\left.\frac{d\tilde{\mathbf{Q}}}{dt}\right|_{i+1/2,j+1/2} + \mathbf{1}\left[\frac{\partial}{\partial X}(y_Y^N - x_Y^N) + \frac{\partial}{\partial Y}(-y_X^N + x_X^N)\right]_{i+1/2,j+1/2} = \mathbf{0} \tag{23}$$

Since $\mathbf{x}^N \in \mathsf{P}_{N,N}$,

$$\frac{\partial}{\partial X}\left(\frac{\partial \mathbf{x}^N}{\partial Y}\right)_{i+1/2,j+1/2} = \sum_{k,l=0}^{N} \mathbf{x}_{k,l}^N \ell_k'(\overline{X}_{i+1/2})\ell_l'(\overline{Y}_{j+1/2}) = \frac{\partial}{\partial Y}\left(\frac{\partial \mathbf{x}^N}{\partial X}\right)_{i+1/2,j+1/2} \tag{24}$$

so that

$$\left.\frac{d\tilde{\mathbf{Q}}}{dt}\right|_{i+1/2,j+1/2} = \mathbf{0} \quad \begin{matrix} i = 0,1,\ldots,N-1 \\ j = 0,1,\ldots,M-1 \end{matrix}. \tag{25}$$

5 Examples

In this section, we use the staggered grid approximation to compute three steady flow problems. The first problem is subsonic flow from a point source. The flow is computed on a multiply connected geometry to show that the method is suitable for complex geometries. This problem has an exact, analytic solution, and we show that exponential convergence is obtained. The next problem is a steady subsonic flow through an array of cylinders in a duct. The final problem computes a transonic flow in an axisymmetric converging-diverging nozzle. The solution for the nozzle is compared to experimental data.

5.1 Subsonic point source flow

As our first example, we consider the flow of a steady, irrotational gas exiting from a point. This flow can be solved exactly by a hodograph transformation [8]. The streamlines are radial, and level curves of the Mach number, pressure and density are circles centered on the source.

We solve the point source flow on the grid shown in Figure 3. The geometry, a square with five circles cut out of its interior, was chosen to show that the method can be used to compute a flow in a complex, multiply connected region. Twenty four subdomains were used to cover the computational domain. Shown on the figure are the grid lines of the Lobatto grid. The solutions themselves are actually defined interior to each "cell" bounded by the grid lines.

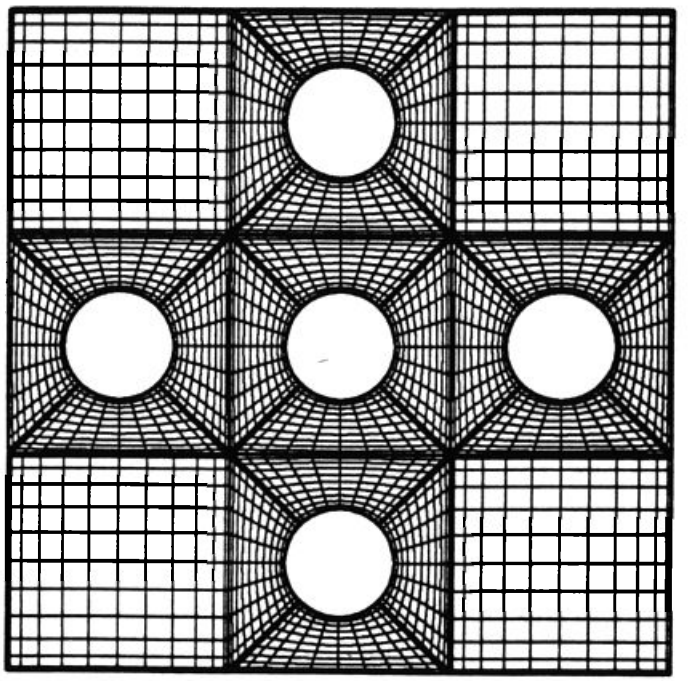

Figure 3: Grid for point source problem.

The boundary conditions were chosen so that the exact steady solution is radial flow with a point source at the center of the middle circle of Figure 3. The center cutout circle was specified as an inflow boundary, with the conditions chosen so that the Mach number of the incoming flow was $M = 0.6$. The boundary conditions along the remaining cutout circles were either inflow or outflow, depending on the direction of the normal velocity. The square outer

boundary of the problem was an outflow boundary. For all inflow/outflow boundaries, the exact solution was used to provide the external flow values required by the Riemann solver.

In Figure 4, we plot the exact and computed Mach number contours for the solution of the point source flow. The contour lines of the exact solution, which are plotted with dashed lines, are coincident with the solid contour lines of the computed solution. In this figure, and in those following, contours are plotted using solution values interpolated from the Gauss points to the Lobatto points. The interpolation is done for display reasons, since a plot using the Gauss points would show gaps between the subdomains, representing the fact that the solution is not defined on the interfaces. On the other hand, plotting the interpolant does give some indication of the size of the jumps in the solution at the interfaces.

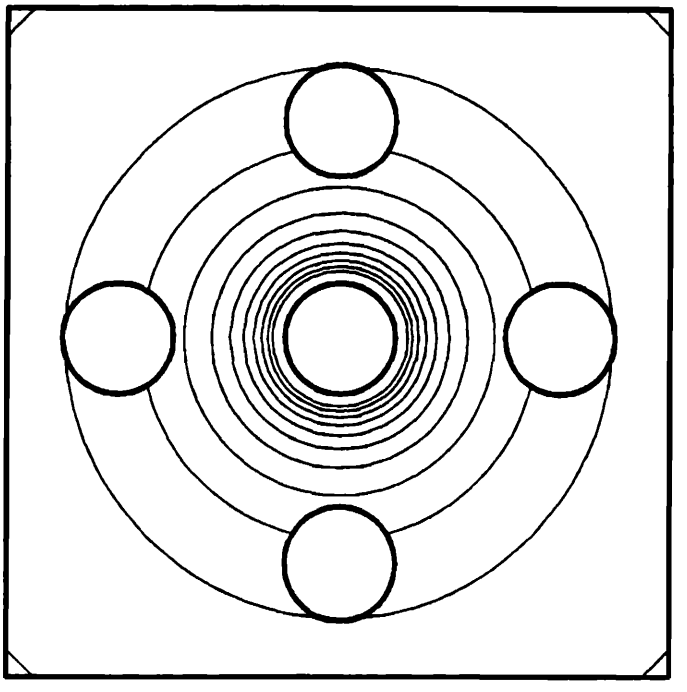

Figure 4: Solution of the point source flow for the geometry shown in Fig. 3. The exact solution is plotted with dashed lines, the computed with solid lines.

Figure 5 shows the maximum error in the density as a function of the number of points per subdomain and indicates exponential convergence of the solution. We see that doubling the number of points per subdomain causes the error to decay by approximately two orders of magnitude.

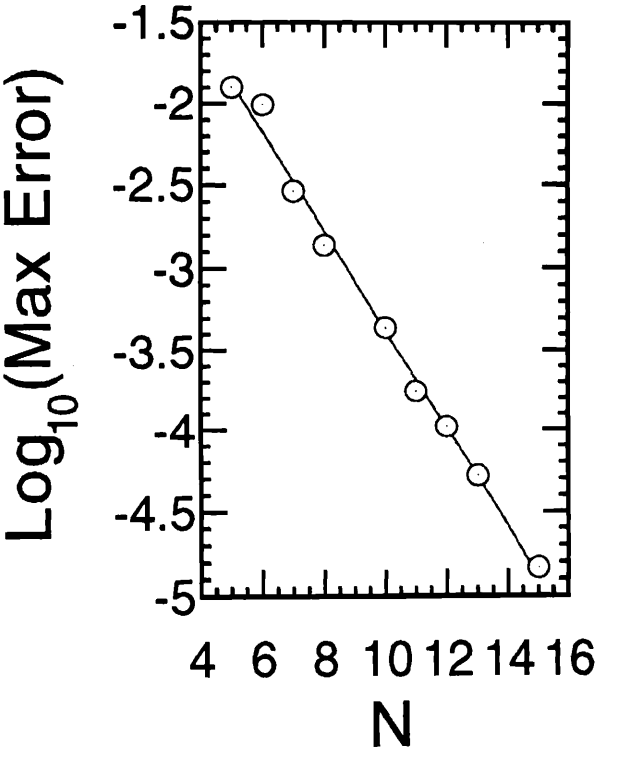

Figure 5: Convergence of the density for the solution shown in Fig. 4.

5.2 Subsonic flow through an array of cylinders in a duct

Our second example is the solution of a flow through an array of cylinders inside a straight duct (Figure 6). Wall boundary conditions were enforced on the upper and lower boundaries, in addition to the cylinder surfaces. The boundary conditions at the left and right enforced a horizontal, Mach 0.25 free stream. The initial conditions specified the uniform flow everywhere. Solution contours for the Mach number are shown in Figure 6b.

5.3 Transonic flow in a converging-diverging nozzle

To show that the method is applicable to transonic problems, we compute the flow in an axisymmetric converging-diverging nozzle. We have chosen the nozzle used in the experimental investigation of Cuffel et al. [9], which was designed to show significant two dimensional effects. The

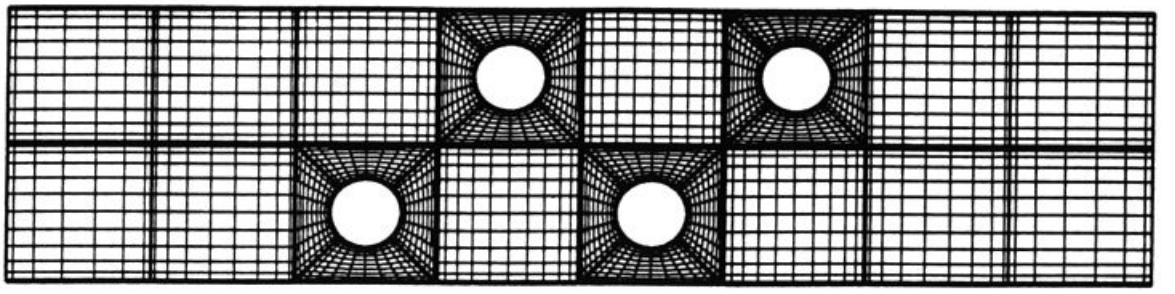

Figure 6a: Grid for flow in a duct.

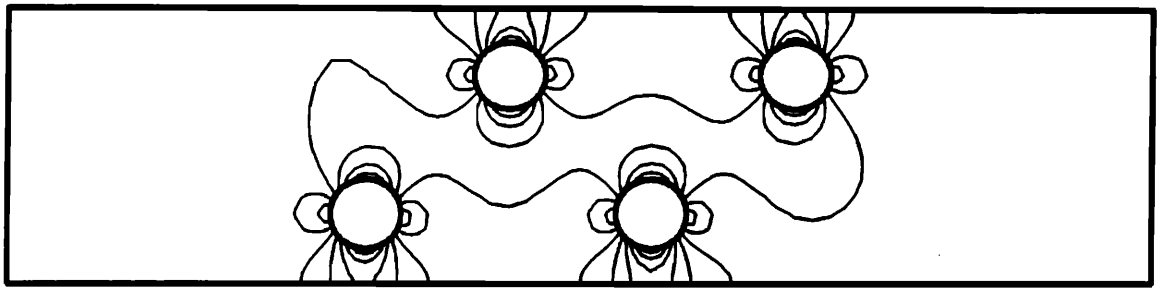

Figure 6b: Mach contours for the flow in the duct.

nozzle consists of a converging section with half angle of 45° and a diverging section with half angle of 15°. The experimental tests were done in air with a stagnation temperature of 300K and stagnation pressure of 1.01×10^{-2} Pa. Figure 7 shows the nozzle geometry and the grid used in our computations.

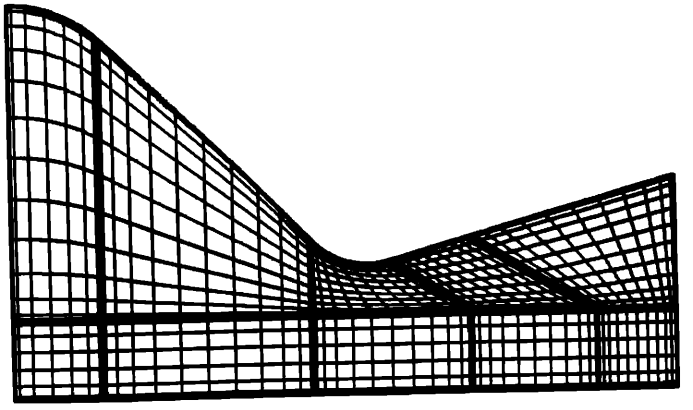

Figure 7: Grid for the 45°–15° converging-diverging nozzle.

To match the experimental conditions, we scaled the equations (1)–(3) by $\rho = \rho^*/\rho_{tot}, p = p^*/p_{tot}$, where the * represents the dimensional quantity. Under this scaling, the temperature and entropy become $T = T^*/T_{tot}, S_{tot} = 0$. The initial condition for the computation was the exact solution of the quasi-one-dimensional nozzle having the same area as the two-dimensional nozzle. For the inflow condition at the left boundary, we specify the tangential velocity to be zero, the entropy to be zero, and the temperature to be equal the total temperature specified by the experiment. At the right boundary, the outflow is supersonic, so no boundary condition is necessary there.

Since not all of the external flow values are known at the left boundary, particularly the inflow velocity, it is not convenient to use the Riemann solver to impose the boundary condition. Instead, we use the following characteristic-like method that allows us to specify the known parameters. The fact that the inflow condition enforces $v = 0$ means that the flow is essentially one dimensional there. In terms of the Mach number, M, and the sound speed, a, the left-going Riemann invariant for the one dimensional flow will satisfy

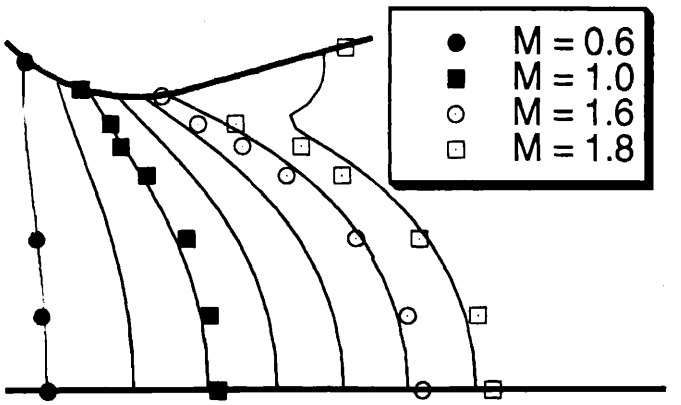

Figure 8: Comparison of computed and measured Mach number near the nozzle throat.

$$a\left(M - \frac{2}{\gamma - 1}\right) = R^-_{computed} = a_{comp.}\left(M_{comp.} - \frac{2}{\gamma - 1}\right) \tag{26}$$

where the computed quantities represent values computed at the boundary by the reconstruction procedure. Equation (26), plus the relationship between the Mach number, total sound speed and sound speed gives

$$\frac{\sqrt{\gamma}}{\sqrt{1 + \frac{\gamma-1}{2}M^2}}\left(M - \frac{2}{\gamma - 1}\right) = R^-_{computed}. \tag{27}$$

Equation (27) can be written as a quadratic equation in the Mach number and solved directly. Once the inflow Mach number is known, the sound speed can be computed. From the Mach number, the sound speed, tangential velocity and the entropy, all remaining variables and the boundary flux can be computed.

Some results computed for the nozzle are shown in Figures 8 and 9. First a comparison of the computed Mach contours and the measured Mach number in the neighborhood of the nozzle throat is shown in Figure 8. We see good agreement between the computed Mach contours and the measured values up to about $M = 1.6$. We note that the discrepancies are consistent with the discrepancies observed with other inviscid flow solvers reported in [9]. A comparison between the computed and measured values of the pressure along the upper wall of the nozzle is shown in Figure 9.

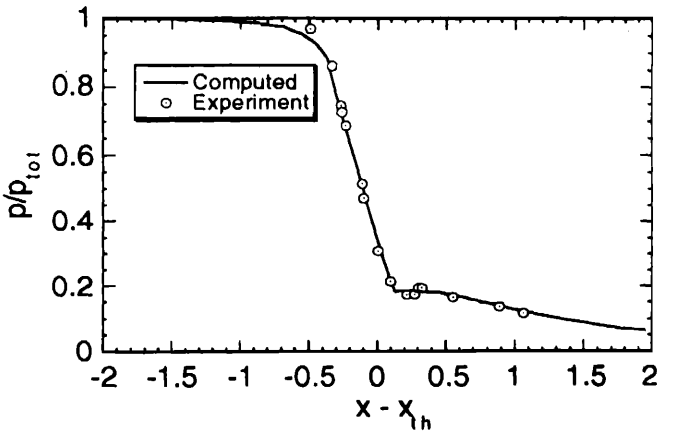

Figure 9: Comparison of computed and measured upper wall pressure as a function of distance from the nozzle throat.

6 Concluding remarks

We have described a new, staggered-grid Chebyshev spectral multidomain method for the solution of inviscid compressible flow problems. In this method, the solutions are defined at the nodes of a Gauss quadrature rule, while the fluxes are evaluated at the nodes of a Gauss-Lobatto rule. An approximate Riemann solver is used to determine the characteristic decomposition needed to advect waves through a subdomain interface. We have presented applications here to two dimensional problems, but the method is also applicable to one dimensional problems, and should extend directly to three dimensions. The method should also be applicable to the solution of any hyperbolic system in which the flux vector itself can be decomposed into wave components. The staggered grid multidomain concept has many desirable features, including conservation, free-stream preservation, geometric flexibility and programming simplicity.

Acknowledgments

The author would like to thank Prof. D. Gottlieb, Prof. P.L. Roe and Prof. C.-W. Shu for many helpful discussions. This research was supported in part by the U.S. Department of Energy through Contract # DE-FC05-85ER250000, by the National Science Foundation through Grant DMS-9404322.

References

[1] C. Canuto, M.Y. Hussaini, A. Quarteroni and T.A.

Zang, Spectral Methods in Fluid Mechanics, Springer-Verlag, New York, 1987.

[2] C. Bernardi and Y. Maday, *A collocation method over staggered grids for the stokes problem*, Int. J. Num Meth. Fluids, 8 (1988), pp. 537–557.

[3] J. Giannakouros and G. E. Karniadakis, *A spectral element-FCT method for the compressible Euler equations*, J. Comp. Phys, 115 (1994), pp. 65–88.

[4] D.A. Kopriva, *Multidomain spectral solution of the Euler gas-dynamics equations*, J. Comp. Phys., 96 (1991), pp. 428–450.

[5] C. Hirsch, Numerical Computation of Internal and External Flows, Vol. II, John Wiley and Sons, Chichester, England (1990).

[6] P.L. Roe, *Approximate Riemann solvers, parameter vectors, and difference schemes*, J. Comp. Phys., 43 (1981), pp. 357–372.

[7] M-Q Chen and C. Chiu, *Optimal m-stage Runge-Kutta schemes for steady-state solutions of hyperbolic equations*, Num. Meth. for P.D.E's, 9 (1993), pp. 643–666.

[8] R. Courant and K.O. Friedrichs, Supersonic Flow and Shock Waves, Springer-Verlag, New York, 1976.

[9] R.F. Cuffel, L.F. Back and P. F. Massier, *Transonic flowfield in a supersonic nozzle with small throat radius of curvature*, AIAA J., 7 (1969), pp. 1364–1366.