2014

Numerical integration techniques for discontinuous manufactured solutions

Benjamin Grier
Clemson University, bgrier@clemson.edu

Edward Alyanak
Wright Patterson Air Force Base, Edward.Alyanak@wpafb.af.mil

Michael White
Wright Patterson Air Force Base, Michael.White.ctr@wpafb.af.mil

Jose Camberos
Wright Patterson Air Force Base, jose.camberos@wpafb.af.mil

Richard Figliola
Clemson University, fgliola@clemson.edu

Journal of Computational Physics 278 (2014) 193–203

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Numerical integration techniques for discontinuous manufactured solutions

Benjamin Grier [b,*], Edward Alyanak [a], Michael White [a], José Camberos [a], Richard Figliola [b]

[a] U.S. Air Force Research Laboratory, Wright Patterson Air Force Base, Dayton, OH 45433, USA
[b] Mechanical Engineering, Clemson University, Clemson, SC 29634, USA

ARTICLE INFO

Article history:
Received 14 January 2014
Received in revised form 14 August 2014
Accepted 23 August 2014
Available online 27 August 2014

Keywords:
Numerical integration
Method of manufactured solutions
Discontinuity

ABSTRACT

When applying the method of manufactured solutions (MMS) on computational fluid dynamic software, determining the exact solutions and source terms for finite volume codes where the stored value is an integrated average over the control volume is non-trivial and not frequently discussed. MMS with discontinuities further complicates the problem of determining these values. In an effort to adapt the standard MMS procedure to solutions that contain discontinuities we show that Newton–Cotes and Gauss quadrature numerical integration methods exhibit high error, first order limitations. We propose a new method for determining the exact solutions and source terms on a uniform structured grid containing shock discontinuities by performing linearly and quadratically exact transformations on split cells. Transformations are performed on triangular and quadrilateral elements of a systematically divided discontinuous cell. Using a quadratic transformation in conjunction with a nine point Gauss quadrature method, a minimum of 4th order accuracy is achieved for fully general solutions and shock shapes. A linear approximation of curved shocks is also experimentally shown to be 2nd order accurate. The numerical integration method is then applied to a CFD code using simple discontinuous manufactured solutions which return consistent 1st order convergence values. The result is an important step towards being able to use MMS to verify solutions with discontinuities. This work also highlights the use of higher order numerical integration techniques for continuous and discontinuous solutions that are required for MMS on higher order finite volume codes.

1. Introduction

The method of manufactured solutions (MMS) has been used to verify computational fluid dynamic (CFD) codes using continuous, open flow examples since it was first developed by Steinberg and Roache in 1985 [1]. MMS rigorously verifies if a code is solving the governing equations correctly by checking the observed order of accuracy of the global solution [2]. To apply MMS, solutions are manufactured to the chosen governing equations (ex. Euler, Navier–Stokes equations). As it is highly unlikely that the manufactured solutions are an exact solution to the problem at hand, source terms are generated for

* Corresponding author.
E-mail addresses: bgrier@clemson.edu (B. Grier), Edward.Alyanak@wpafb.af.mil (E. Alyanak), Michael.White.ctr@wpafb.af.mil (M. White), jose.camberos@wpafb.af.mil (J. Camberos), fgliola@clemson.edu (R. Figliola).

http://dx.doi.org/10.1016/j.jcp.2014.08.031
0021-9991/

the governing equations such that the manufactured solutions become exact. The numerical solution can then be compared to the manufactured solution to calculate the global discretization error and observed order of accuracy over multiple grid refinements.

An issue arises when performing MMS on finite volume codes since the stored value is an integrated average, as estimated by the fluxes around the control volume, and not the manufactured solution defined at a single point [3]. This is further complicated because manufactured solutions that are not analytically integrable are common. These solutions typically have many derivatives which helps ensure that the governing equations are fully exercised providing a more comprehensive test. Since source terms are derived from the exact solutions, allowing non-analytically integrable equations lifts a potentially difficult requirement. Trigonometric terms which are not analytically integrable and therefore do not have exact values, are frequently used to this end. Though usually not stated explicitly, current papers using MMS on finite volume codes, such as [2] and [4], likely use a midpoint approximation for determining the source terms and exact solutions. This 2nd order approximation of the integral is only appropriate for codes with order 2 or less. In Section 4 we show that it is simple to incorporate higher order methods that obtain a better first approximation and converge faster than the midpoint method. Using higher order approximations also extends the ability of MMS to higher order codes.

While MMS has been used primarily for simple continuous, open flow examples, using MMS with solutions containing discontinuities is still an open research issue [5]. Discontinuous MMS could be valuable for verifying code segments which may not activate for continuous flow solutions as well as to test the likely reduced order of a code when a shock appears; a potentially non-trivial case when dealing with linearly degenerate waves [6]. Another possible benefit is the testing of existing and new shock capturing schemes. Many of these schemes are tested using exact solutions to the Riemann problem consisting of simple waves separated by constant solutions and this form of testing may not be sufficient for obtaining the scheme's accuracy [7].

Determining the exact solutions and source terms for MMS with discontinuous solutions is a more difficult problem. It is shown in Section 3 that traditional Newton–Cotes methods have high error, first order limitations when cell faces are not shock aligned. Since discontinuous MMS has not yet been established in the literature, there are no guidelines for how a discontinuous problem should be defined. In [8] Roach suggests that grid convergence may be difficult to judge for a continuous solution with a sufficiently steep gradient. In this paper a piecewise solution technique is proposed to represent a shock and exactly integrable solutions are then used with varying shock shapes to test the accuracy of the integration approximations. In addition to CFD, this research could be of interest when dealing with any solution that contains a discontinuity. Discontinuities appear in many physical problems such as fluid–solid interfaces, dielectrics, and code to code couplings and are described generally in [9].

In Section 3 we prove in one dimensional space that the midpoint approximation typically used for MMS with finite volume codes results in a reduction of formal order when any discontinuity is present in the system. This same trend is also demonstrated experimentally in two dimensions for Simpson's and Gauss quadrature approximations. Due to both the low order restrictions and high uncertainty associated with using Newton–Cotes or Gauss quadrature methods in the presence of a discontinuity, a new method is explored in Section 4. Here we apply a cell transformation combined with a 6th order Gauss quadrature approximation. Each cell containing a discontinuity is systematically divided into triangular and quadrilateral elements and each element is then transformed using linearly and quadratically exact transformations. The 6th order Gauss quadrature method can then be used to obtain high order of accuracy in the estimation of both the exact solutions as well as the source terms.

In Section 5 the developed integration method is implemented into a CFD code. The code used is a cell centered, finite volume, 1st order, Eulerian scheme within the software AVUS (Air Vehicles Unstructured Solver). AVUS is used as a representative testing platform with the convenience of accessible source code. In CFD the manufactured solutions on either side of a discontinuity cannot be completely arbitrary and must be linked by physical equations. This is due to the nature of jump conditions which do not typically contain source terms as are normally used with MMS. Work completed by [10] and [11] have proposed solutions for dealing with source terms within the Riemann solver, but the code used in this paper contains no such modifications. We show that discontinuous manufactured solutions can be linked through Rankine–Hugoniot jump conditions. A simple discontinuous manufactured solution is then shown to converge with 1st order accuracy in the L_1 domain.

2. High order approximations

When performing MMS it is usually assumed that the analytical values to the exact solutions and source terms are known. This is not guaranteed to be true with finite volume codes, and approximations to the integral must be used. Despite this issue, MMS has been used to verify finite volume codes in the past and examples are given in [12] and [2]. Typically, the cell average is calculated using a midpoint approximation which is formally 2nd order accurate. While the midpoint approximation may be adequate for the code at hand, it should be noted that it is advantageous to institute a higher order scheme. Integration methods are frequently chosen based on where data already exists, but having the analytic functions means we are free to choose any method that best suits our needs. Two Newton–Cotes methods, the midpoint rule and Simpson's rule, are experimentally validated for a simulated manufactured solution, Eq. (1), in Table 1.

$$f(x, y) = e^y - \cos x \tag{1}$$

Table 1
Numerical integration error and observed orders of convergence with no discontinuity.

Grid size	Midpoint	Observed order	Simpson's	Observed order	Gauss quadrature	Observed order
2×2	3.30e0		3.17e−2		2.26e−6	
4×4	7.60e−1	2.00	2.12e−3	3.90	3.53e−8	6.00
8×8	1.90e−1	2.00	1.35e−4	3.98	5.52e−10	6.00
16×16	4.75e−2	2.00	8.47e−6	3.99	8.67e−12	5.99

While Newton–Cotes methods can be constructed for any order of accuracy, they can become unstable due to Runge's phenomenon [13]. Gauss quadrature methods are typically more stable and a 9 point stencil with a formal order of accuracy of $O(h_x^6, h_y^6)$ was also tested in Table 1. Each numerical integration method was tested for a cell with nodes $(0,0)$, $(0,1)$, $(1,1)$, and $(1,0)$. The cell was then discretized into an $n \times n$ grid where n was initialized at 2 and then doubled for each run. The orders of convergences and integration errors are displayed in Table 1.

Using continuous solutions, all methods behaved as expected. The Gauss quadrature method begins to break down in subsequent iterations since it quickly reaches the 16 bit precision of the calculated analytical value. It should be noted that, although the midpoint method may match the formal order of accuracy of the code being verified, the initial exactness of the solution may be inappropriate for the chosen solution. This may lead to slower convergences or possible oscillation around the predicted value. Considering that manufactured solutions provide the user with information over the entire domain, it is the authors' opinion that an integration method of higher order than the formal spacial order should always be implemented with finite volume codes to ensure there are no complications due to inaccurate integration approximations. This also shows that MMS is indeed applicable to finite volume codes with formal orders higher than two.

3. Numerical integration with discontinuities

An important step in the development of MMS with discontinuities is determining the exact solution and source terms for cells which are divided through their interior. Since numerical integration methods, like those used in Section 2, are defined for continuous solutions, their formal order on split cells is not obvious. We prove the formal order of the midpoint approximation for a one dimensional solution containing a discontinuity using a 1 panel method in the proof below. Consider an approximation of the solution on either side of the discontinuity using a Taylor's series,

$$f(x) = \begin{cases} f_c + b_1(x - x_c) + c_1(x - x_c)^2 + \cdots & x_j \le x < x_d \\ f_d + b_2(x - x_c) + c_2(x - x_c)^2 + \cdots & x_d < x \le x_{j+1}, \end{cases} \tag{2}$$

where the cell average is defined by

$$x_c \equiv \frac{1}{2}(x_j + x_{j+1}). \tag{3}$$

The location of the discontinuity is defined by

$$x_d \equiv x_c + \frac{\alpha\Delta}{2}, \quad \{\alpha \mid 0 < \alpha \le 1\}, \tag{4}$$

where Δ is

$$\Delta \equiv x_{j+1} - x_j. \tag{5}$$

Rearranging the bounds of Eq. (2) yields

$$f(x) = \begin{cases} f_c + b_1(x - x_c) + c_1(x - x_c)^2 + \cdots & x_c - \frac{\Delta}{2} \le x < x_c + \frac{\alpha\Delta}{2} \\ f_d + b_2(x - x_c) + c_2(x - x_c)^2 + \cdots & x_c + \frac{\alpha\Delta}{2} < x \le x_c + \frac{\Delta}{2}, \end{cases} \tag{6}$$

and integrating with respect to x gives

$$\int_{x_c - \frac{\Delta}{2}}^{x_c + \frac{\Delta}{2}} f(x)\,dx = \Delta\left(\frac{1+\alpha}{2} f_c + \frac{1-\alpha}{2} f_d\right) + \Delta^2\left(\frac{1-\alpha^2}{8}(b_2 - b_1)\right) + \mathcal{O}(\Delta^4). \tag{7}$$

This equation is checked using the case of $\alpha = 1$, no discontinuity, which returns Eq. (8), the standard third order 1-panel midpoint method.

$$\int_{x_c - \frac{\Delta}{2}}^{x_c + \frac{\Delta}{2}} f(x)\,dx = \Delta f_c + \frac{\Delta^3}{12} c_1 + \mathcal{O}(\Delta^4) \tag{8}$$

If α is defined on any section between 0 and 1 the midpoint rule can be extracted from Eq. (7) resulting in

$$\int_{x_c-\frac{\Delta}{2}}^{x_c+\frac{\Delta}{2}} f(x)\,dx = \Delta f_c + \Delta\frac{1-\alpha}{2}(f_d - f_c), \tag{9}$$

which gives a leading truncation error of

$$R = \Delta\frac{1-\alpha}{2}(f_d - f_c). \tag{10}$$

This new remainder term on the 1 panel method returns a first order approximation. This can also be generalized for an N point panel method assuming an equally spaced domain (A, B) with N cells such that

$$\Delta = \frac{B-A}{N}. \tag{11}$$

The integral over the entire domain is given as

$$\int_A^B f(x)\,dx = \int_A^{x_j} f(x) + \int_{x_j}^{x_{j+1}} f(x) + \int_{x_{j+1}}^B f(x). \tag{12}$$

Labeling the cell with the discontinuity N_d the outside integrals are completed using Eqs. (13) and (14).

$$\int_A^{x_j} f(x)\,dx \approx \sum_{n=1}^{N_d-1}\left(\Delta f_{c,n} + \frac{\Delta^3}{12}c_{1,n}\right) \tag{13}$$

$$\int_{x_{j+1}}^B f(x)\,dx \approx \sum_{n=N_d+1}^{N}\left(\Delta f_{d,n} + \frac{\Delta^3}{12}c_{2,n}\right) \tag{14}$$

The evaluation of the middle integral, derived in the same manner as above, is given as

$$\int_{x_j}^{x_{j+1}} f(x)\,dx = \Delta\left(\frac{1+\alpha}{2}f_{c,N_d} + \frac{1-\alpha}{2}f_{d,N_d}\right) + \Delta^2\left(\frac{1-\alpha^2}{8}(b_{2,N_d} - b_{1,N_d})\right) + \mathcal{O}(\Delta^4). \tag{15}$$

Comparing to the midpoint rule, the leading error term comes from Eq. (15) and is once again given with first order accuracy as

$$R = \Delta\frac{1-\alpha}{2}(f_{d,N_d} - f_{c,N_d}). \tag{16}$$

The consequences of using the midpoint method with a discontinuity becomes quite clear. No matter the refinement of the cells on the global domain, the error on the cell with the discontinuity will dominate, at best, with first order accuracy. The validation of this in two dimensions for the midpoint, Simpson's and Gauss quadrature rules is shown experimentally below. To simulate a discontinuity, a piecewise, analytically integrable solution, Eq. (17), and oblique shock equation, Eq. (18), were chosen. The contour plot is shown in Fig. 1(a).

$$f(x, y) = \begin{cases} \sin x + \cos y & \text{for } shock(x, y) > 0 \\ e^y - \cos x & \text{for } shock(x, y) < 0 \end{cases} \tag{17}$$

$$shock = -x + \frac{y}{5} - \frac{1}{3} \tag{18}$$

Due to the ubiquity of discontinuities in physics, the term shock in this paper is used in terms of a general Riemann problem [9]. It should then be noted that Eq. (17) may not be appropriate for MMS within a CFD solver, but is presented here simply to test the numerical integration.

The shock is defined by Eq. (18) such that the domain is split with neither solution dominating. Despite the expectation of 1st order convergence in one dimension, it is clearly seen in Table 2 that the convergence values vary significantly, even going to negative values in places. While the average of all orders are all contained between $\mathcal{O}(h)$ and $\mathcal{O}(h^2)$ this tendency is disconcerting for MMS code verification which is known for being extremely sensitive to coding mistakes [2]. Discontinuous CFD solutions do not converge faster than $\mathcal{O}(h)$ [14,6,15], yet it is still conceivable these inconsistent approximations could skew an order of accuracy test. In the following sections we develop techniques for overcoming these restrictions.

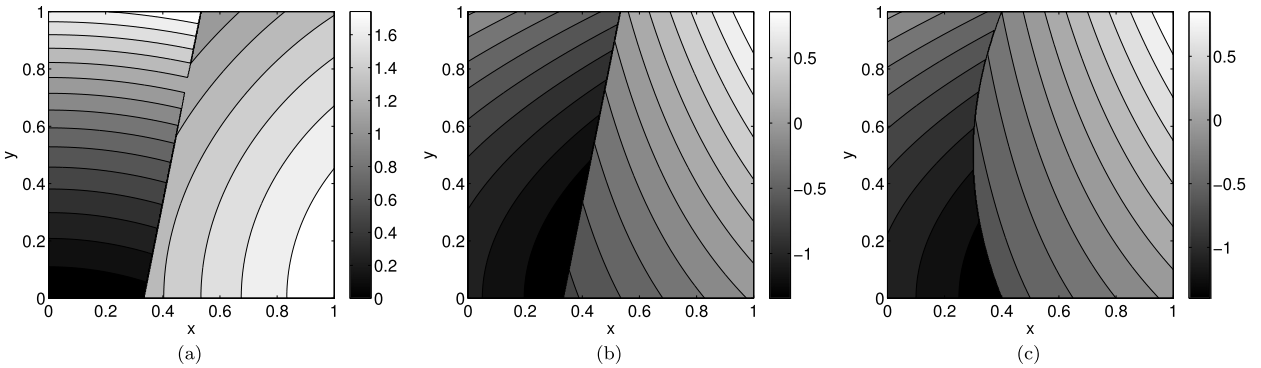

Fig. 1. Contour plots for Eq. (17) with an oblique shock (a) and Eq. (28) with an oblique shock (b) and a bow shock (c).

Table 2
Numerical integration error and observed orders of convergence with an oblique discontinuity.

Grid size	Midpoint	Observed order	Simpson's	Observed order	Gauss quadrature	Observed order
2×2	5.99e0		1.40e1		2.33e−1	
4×4	3.75e−1	4.00	1.23e0	3.51	2.09e−1	0.16
8×8	3.79e−1	−0.02	2.76e−1	2.15	8.31e−2	1.33
16×16	4.61e−2	3.04	5.70e−2	2.28	2.02e−2	2.04
32×32	3.26e−2	0.50	5.88e−2	−0.05	8.95e−3	1.17
64×64	2.12e−2	0.62	1.52e−2	1.95	4.37e−3	1.04

4. Numerical integration of cells divided by discontinuities

The following section develops techniques for overcoming the first order restrictions presented in Section 3. As is sometimes accomplished with codes that have mixed time and spatial orders [5], it is theoretically possible to approximate the source terms and exact solutions based on larger grid refinements. Provided that this sub-discretization was doubled for every subsequent refinement, a standard Newton–Cotes method would theoretically double its accuracy and behave as a 2nd order method. This is unfortunately in lieu of the highly inconsistent convergence values shown in Table 2, and it is up to the user to verify that this works for each individual manufactured solution due to the high variability in answers. It is also worth noting that this can drastically increase computational requirements, possibly reducing the feasible number of refinements.

A higher order method for dealing with cells containing discontinuities is to perform a transformation on the divided parts such that a Gauss quadrature method can be used without requiring weighted points on both sides of the discontinuity. A quadrilateral and triangular transformation are developed here and are used to integrate each segment of the divided cell. Assuming a uniform, structured grid there are two typical scenarios, totaling six possible cases illustrated in Fig. 2. There are also two special scenarios denoted by a *, totaling an additional ten cases for which the shock could break up a cell.

1. Shock passes through opposite faces of a bilateral cell.
2. Shock passes through adjacent faces of a bilateral cell.
3. *Shock passes through a corner and a face.
4. *Shock pass through opposite corners.

The left two cases in Fig. 2 require only quadrilateral transformations as each division has four points, but cases 3–6 require additional manipulation as each cell is initially split into a three and five node elements. The five node elements can be further split up into two quadrilateral elements that can be solved using two transformations, as demonstrated by the dotted line. This implies that cases 3–6 in Fig. 2 will utilize three separate transformations and Gauss quadrature integrations as shown in Fig. 3. Special scenarios 3 and 4 can similarly be subdivided with a combination of triangular and quadrilateral transformations but are omitted here as each case can be avoided with a careful selection of grid refinements and placement of the discontinuity.

In summary, a combination triangular and quadrilateral transformations are needed for all cell integrations. Since Newton–Cotes methods require equally-spaced points and are typically of lower order, a Gauss quadrature method is chosen for exclusive use in the following sections. Writing the quadrature rule for use in transformed space $\{\xi, \eta\}$, gives

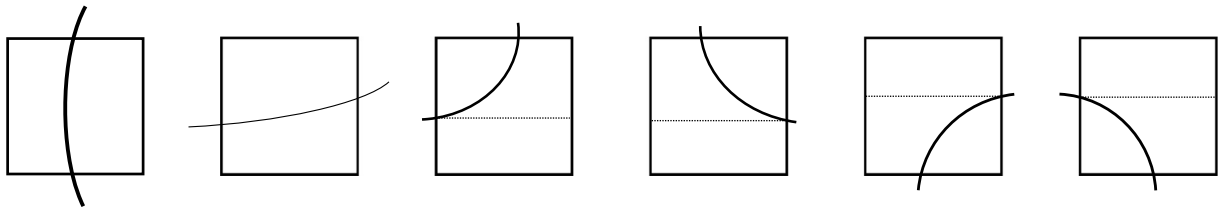

Fig. 2. Six typical cases for a shock passing through a cell.

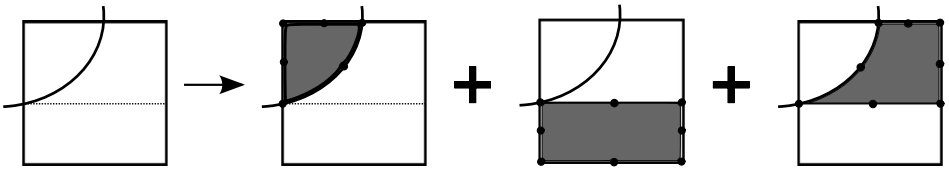

Fig. 3. The different transformations required to split a sample cell with a shock passing through the upper and left hand faces.

$$I = \sum_{i=1}^{N} J_i w_i f\left(x[\xi_i, \eta_i, x_i], y[\xi_i, \eta_i, y_i]\right), \tag{19}$$

where J_i is the Jacobian. Function positions and weights are given in [13]. The following sections outline the required transformations.

4.1. Linear transformations

The quadrilateral transformation maps all x and y coordinates to the ξ and η axes where the point $\xi = \eta = 0$ is the new transformed element's center bounded by $\xi = \pm 1$ and $\eta = \pm 1$. The coordinates are transformed using Eq. (20).

$$\begin{Bmatrix} x \\ y \end{Bmatrix} = \begin{Bmatrix} \sum N_i x_i \\ \sum N_i y_i \end{Bmatrix} \tag{20}$$

Individual shape functions are developed using Eq. (21).

$$[N] = [X][A]^{-1} \tag{21}$$

While X, Eq. (23), can contain any basis functions, the first three terms of Pascal's triangle, Eq. (22), and an additional cross product are typically used for a linear interpolation of a quadrilateral element.

$$\begin{array}{cccc} & & 1 & \\ & \xi & & \eta \\ \xi^2 & & \xi\eta & & \eta^2 \\ \xi^3 & & \xi^2\eta & & \xi\eta^2 & & \eta^3 \end{array} \tag{22}$$

$$X = [\,1 \quad \xi \quad \eta \quad \xi\eta\,] \tag{23}$$

Each row of A corresponds to a node location evaluated at each X index. Using the first 4 positions of Fig. 4(a), A equates to

$$A = \begin{bmatrix} 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}. \tag{24}$$

The shape functions are evaluated as

$$\begin{aligned} N_1 &= (1-\xi)(1-\eta)/4, \qquad N_2 = (1+\xi)(1-\eta)/4, \\ N_3 &= (1+\xi)(1+\eta)/4, \qquad N_4 = (1-\xi)(1+\eta)/4. \end{aligned} \tag{25}$$

The Jacobian, required for determining the partial cell area, is

$$\mathbf{J} = \det \begin{Bmatrix} \sum N_{i,\xi} x_i & \sum N_{i,\xi} y_i \\ \sum N_{i,\eta} x_i & \sum N_{i,\eta} y_i \end{Bmatrix}, \tag{26}$$

where the second subscript of N designates the respective derivative. The full evaluation of Eq. (26) is not included for brevity. More information on transformations and shape functions can be found in Ref. [16]. The triangular element is

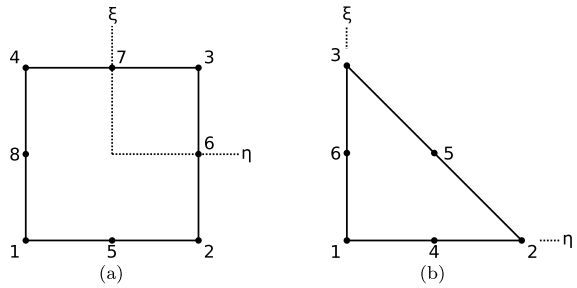

Fig. 4. Required node order when generating the A matrix for quadrilateral (a) and triangular (b) transformations.

Table 3
Percent integration error and observed orders of convergence for a low order solution (Eq. (28)), subdivided by an oblique (Eq. (18)) and curved shock (Eq. (27)) and solved via a linear transformation.

Grid size	Oblique error	Observed order	Quadratic error	Observed order
2×2	3.31e−14		3.16e0	
4×4	0.00	undef	7.62e−1	2.05
8×8	3.31e−14	undef	1.54e−1	2.31
16×16	8.29e−14	−1.32	4.34e−2	1.82
32×32	3.65e−13	−2.14	1.05e−2	2.05
64×64	1.47e−12	−2.02	2.65e−3	1.98

constructed using the same derivation, but using only the first three terms of Pascal's triangle, Eq. (22), and the first three nodes shown in Fig. 4(b). The locations and weights for the seven point, 6th order Gauss quadrature method for triangles are given with double precision in [17].

The described transformations above assume linear interpolation between the nodes. To test how well this assumption approximates a curved shock, the oblique and curved shocks are compared when using polynomial manufactured solutions that the Gauss quadrature solves exactly. An N^2 Gauss quadrature is exact for polynomials of $2N-1$ and the 9 point stencils used in this study are therefore exact for 5th order polynomials [13]. The oblique solution is retained as Eq. (18) and the curved shock solution is given below in Eq. (27).

$$shock = -x + 0.4(y-0.5)^2 + 0.3 \tag{27}$$

The chosen polynomials for integration are

$$f(x,y) = \begin{cases} xy + x - 1 & \text{for } shock(x,y) > 0 \\ y^2 - x - 1 & \text{for } shock(x,y) < 0 \end{cases} \tag{28}$$

and are shown in Fig. 1(b) and (c) for each shock shape. It should be noted that the solutions proposed in Eq. (28) are chosen such that the solution remains 5th order underneath the integrand. While the Gauss quadrature numerical scheme does not use analytic integration, it has been shown experimentally that the function must remain sub 5th order for the entirety of the operation shown in Eq. (29)

$$I_{\text{exact}} = \int_a^b \int_c^{shock(0,y)} f(x,y)_{\text{left}} \partial x \partial y + \int_a^b \int_{shock(0,y)}^{c} f(x,y)_{\text{right}} \partial x \partial y, \tag{29}$$

where a, b, and c are constants.

It can be seen in Table 3 that the errors associated with the oblique shock solution are instantly machine error. This was expected since the quadrature and transformation are both exact for the chosen solution rendering the early convergence values meaningless. The increase in error with grid size likely results from an increasing number of summation procedures. When using the curved shock solution the initial errors are quite high, but the observed order is a significant improvement over the results shown in Table 2. Not only are the results much more consistent, but it is also shown in Table 3 that the Gauss quadrature integration combined with a linear transformation is a second order approximation of a solution containing a curved shock.

A consideration when using this method, especially at small grid sizes, is that the solution is not a true grid doubling. Recalling Figs. 2 and 3 the number of Gauss quadrature calculations can go from 1, no shock, to 3, a shock that passes through adjacent faces. Fig. 5 visualizes how this affects the total number of Gauss calculations as the grid is refined. Each refinement in Fig. 5 performs 2, 6, and 20 quadrature calculations for (a), (b), and (c) respectively. The ratio between each of these is 3 : 1, and 3.33 : 1, a significantly lower margin than the required 4 : 1 for grid doubling. This issue is mitigated as $N \to \infty$ where the ratio approaches 4 due to a smaller percentage of the cells containing a discontinuity. It is unlikely

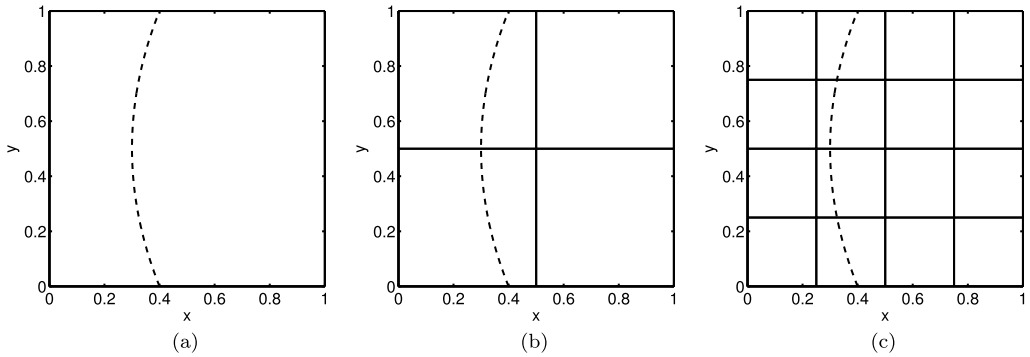

Fig. 5. Grid refinement affected by the number of Gauss quadrature calculations: (a) 2, (b) 6, (c) 20.

Table 4
Convergence of integration error for a general solution equation (17), subdivided by an oblique shock and solved via a linear transformation.

Grid size	% Error	Observed order
2×2	6.49e−7	
4×4	1.98e−8	5.04
8×8	3.55e−10	5.80
16×16	6.05e−12	5.88

that this is an issue when implementing into CFD codes due to a practical lower cell count limit for which solutions are solvable.

Errors for the linear transformation method, especially at small grid sizes, may seem higher than acceptable for an MMS problem where the exact solution is known. Provided that the integration method maintains a higher convergence than the code, the asymptotic observed order is guaranteed to be independent of the integration approximation. It can also be shown using a more complicated solution set, described by Eq. (17) and the oblique shock, Eq. (18), that the transformation does retain the sixth order accuracy of the Gauss quadrature technique when an oblique discontinuity is used. This is a significant advantage over the results presented in Section 3. This exemplifies how important the transformation process is for maintaining accuracy and a higher order method is therefore explored in the following section.

4.2. Quadratic transformations

In the previous section it was shown that four node transformation methods worked extremely well provided that the shock was linear, while the initial errors on a quadratic shock were significantly higher. In this section an 8 and 6 point quadratic transformation of the cell is developed to improve the accuracy and order of convergence when using an arbitrary curve. The procedure is nearly identical to the previous section while using all node locations from Fig. 4. The additional midpoint value allows for the inclusion of higher order basis functions and the new X vector is defined by

$$X = \begin{bmatrix} 1 & \xi & \eta & \xi\eta & \xi^2 & \eta^2 & \xi^2\eta & \eta^2\xi \end{bmatrix} \tag{30}$$

for the quadrilateral element and

$$X = \begin{bmatrix} 1 & \xi & \eta & \xi\eta & \xi^2 & \eta^2 \end{bmatrix} \tag{31}$$

for the triangular. The A matrix, shape functions, and Jacobian are generated as before, but are left out for brevity.

Referring back to Fig. 4, a possible confusion arises when choosing points 5–8 and 4–6 respectively for curved shocks. Since each of these points is a midpoint along a two dimensional line, it is not always obvious where the midpoint should be taken. Assume an element which is bounded on one side by the equation $y = x^2$. A midpoint could be chosen for either axis. If the y midpoint $(\sqrt{.5}, .5)$ is selected, the transformation tries to model the function $x = \sqrt{y}$ out of the basis functions available in Eq. (31). This is impossible to complete exactly whereas $y = x^2$ is obviously contained. While a general shock function may not be modeled exactly anyway, it remains important to be consistent across all cells. Since all shock functions used in this paper are of the form $x = f(y)$ the half-way points used are all based on the vertical axis.

Using the new quadratically exact transformation and Eq. (28) the curved results presented in Table 3 are now solved exactly in Table 5. As with the linear transformation and solution set shown in Table 4, if the basis functions contained in Eq. (30) capture the shock exactly, the 6th order of the Gauss quadrature method is still retained for any piecewise function. Also shown in Table 5 is the error associated with a general solution, Eq. (17), with a new more complicated shock shape described by Eq. (32).

$$shock = -x + 0.6(y - 0.7)^3 + 0.4 \tag{32}$$

Table 5
Integration error for a polynomial solution divided by a quadratic shock and a general solution divided by a cubic shock and solved via a quadratically exact transformation.

Grid size	Quadratic error	Observed order	Cubic error	Observed order
2×2	1.40e−13		1.65e−2	
4×4	1.20e−13	0.22	2.78e−3	2.57
8×8	1.20e−13	0.00	1.49e−4	4.23
16×16	1.81e−13	−0.58	6.43e−6	4.53
32×32	2.21e−13	−0.29	3.92e−7	4.04

This results in a method which is 4th order accurate for all general solution sets not captured exactly by the numerical integration and all shock shapes not captured exactly by the transformation.

As is shown in [12], a code is verified based on the asymptotic convergence values observed over multiple grid refinements and not a single value which can lead to both over and underpredicted orders. Retaining an approximation order of a high magnitude should provide high confidence in the validity of the verification, and may reduce the computational requirements needed to reach the asymptotic limit.

5. Implementation of piecewise manufactured solutions

Manufactured solutions which simulate a physically correct discontinuity are needed for performing MMS within a CFD code. Continuing with the piecewise technique from the previous sections, this requires two sets of manufactured solutions on either side of the discontinuity for all primitives and all generated source terms. For the shock jump to remain physically correct the piecewise equations must be linked by a jump condition. In the testing platform AVUS, the Riemann problem solved by Gottlieb and Groth [18] reduces to the Rankine–Hugoniot jump conditions, Eqs. (33)–(35), for strong shocks.

$$\rho_1 u_1 \sin\beta = \rho_2 u_2 \sin(\beta-\delta) \tag{33}$$

$$P_1 - P_2 = \rho_2\big(u_2\sin(\beta-\delta)\big)^2 - \rho_1(u_1\sin\beta)^2 \tag{34}$$

$$7\left(\frac{P_2}{\rho_2}-\frac{P_1}{\rho_1}\right) = (u_1\sin\beta)^2 - \big(u_2\sin(\beta-\delta)\big)^2 \tag{35}$$

where β is the shock angle and δ is the turn angle. In this example the manufactured solutions are first defined on the upwind side of the discontinuity and the downwind side is then solved for. Solving for the right state primitives yields

$$\rho_2 = \frac{\rho_1 u_1 \sin\beta}{u_2\sin(\beta-\delta)} \tag{36}$$

$$P_2 = P_1 + \rho_1 u_1^2 \sin^2\beta - \rho_1 u_1 u_2 \sin\beta\sin(\beta-\delta) \tag{37}$$

$$6u_2^2\sin^2(\beta-\delta) - u_2\left(\frac{7P_1\sin(\beta-\delta)}{\rho_1 u_1\sin\beta} + 7u_1\sin\beta\sin(\beta-\delta)\right) - \frac{7P_1}{\rho_1} - u_1^2\sin^2 = 0. \tag{38}$$

A set of left state manufactured solutions are chosen as

$$u_1 = 600 + 100y, \qquad v_1 = 0, \qquad \rho_1 = 0.8, \qquad p_1 = 150\,000, \tag{39}$$

with a shock boundary defined by Eq. (18). Despite the left state's simple nature, the right state equations are not given due to their lengthy formulation. While it is possible to pass more general symbolic expressions through the Rankine–Hugoniot jump conditions, the downwind solutions quickly become computationally taxing with small increases in upwind and shock boundary complexity.

The solution was computed with first order accuracy and a visualization of the u-component velocity and its relative error are shown in Fig. 6 for a 50×50 grid. The remaining primitive variables and their errors exhibit a similar pattern and are not shown. It is unsurprising to see that the maximum error lies near the shock and requires multiple cells for the information to propagate through. Table 6 displays the convergence values for all primitive variables using L_1 norms.

The orders shown above approach the expected 1st order convergence typically associated with shocks [6]. A similar procedure could also be used to empirically test the results shown in [6], where the formal order of linear discontinuities reduces to $p_f/(p_f+1)$ where p_f is the formal order for smooth problems. While the orders shown in Table 6 are a proof of concept for the integration method presented throughout this paper, the simplicity of the manufactured solutions does not lend to a strong verification test. Increasing manufactured solution and shock boundary complexity while retaining computational feasibility remains an open research issue.

6. Conclusion

A discussion on how to determine the exact solutions and source terms when performing the method of manufactured solutions on finite volume codes was presented. We showed that it is simple and advantageous to use a higher order

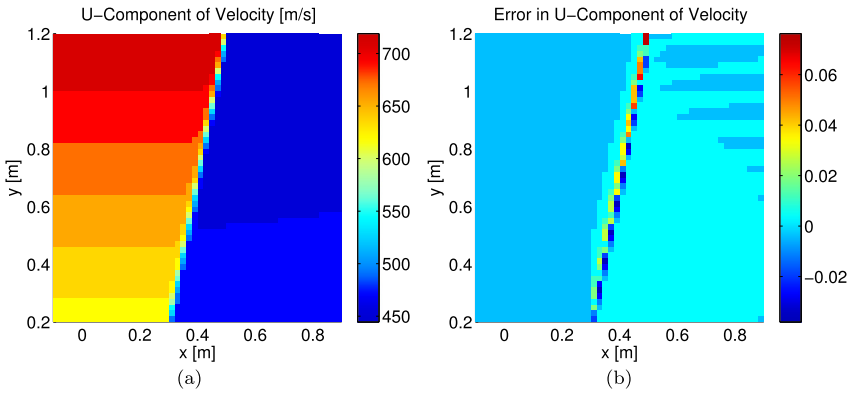

Fig. 6. U-component of velocity (a) and its relative error (b) for an oblique shock.

Table 6
Convergence in the L_1 domain using Eq. (39) combined with an oblique shock.

Grid sizes	ρ	Order	u	Order	v	Order	p	Order
12×12	6.44e−03		3.84e+00		2.11e+00		1.93e+03	
25×25	3.66e−03	0.813	2.26e+00	0.764	1.09e+00	0.948	1.10e+03	0.816
50×50	2.03e−03	0.850	1.26e+00	0.849	5.87e−01	0.897	6.06e+02	0.859
100×100	1.08e−03	0.912	6.65e−01	0.917	3.08e−01	0.932	3.20e+02	0.918
200×200	5.65e−04	0.935	3.46e−01	0.941	1.56e−01	0.984	1.68e+02	0.935
400×400	2.88e−04	0.972	1.78e−01	0.961	7.84e−02	0.988	8.45e+01	0.989

method and 6th order Gauss quadrature results were shown to work at a much higher accuracy and order of convergence than traditional midpoint approximations.

The topic of discontinuous MMS was presented and the midpoint approximation was explored further using cells containing discontinuities. We showed that the formal order of the midpoint approximation in one dimension reduces to $\mathcal{O}(h)$ for single and N panel methods when the cell is divided at any point in its interior. An experimental verification was conducted in two dimensions for the midpoint rule, Simpson's rule, and a nine point Gauss quadrature. The results were not acceptable for MMS application due to their high inconsistency and low order of accuracy.

A new method was presented for determining the exact solutions and source terms by combining a Gauss quadrature approximation alongside a transformation of the split cell. By carefully subdividing the cell at strategic locations, the numerical integration was performed using quadrilateral and triangular elements. It was shown that fully general solutions and shock shapes resulted in a consistent 2nd order method for a linear transformation. A quadratic transformation was also developed and resulted in a 4th order method for solutions which are not captured exactly by either the numerical integration or transformation. Shock shapes which were captured exactly by the transformation basis functions resulted in a method which retained the full order of accuracy of the Gauss quadrature integration.

Lastly, the numerical integration method was implemented into a CFD code and tested using a simple discontinuous solution combined with an oblique shock. This resulted in consistent first order convergence using L_1 norms. The simplicity of the manufactured solutions and shock did not lend to a robust verification test of fully general discontinuous solutions but serves as a proof of concept for piecewise MMS.

Acknowledgements

The authors gratefully acknowledge the Air Force STAR-DP program for their support of this work. The authors would also like to thank Dr. Christopher Roy, Virginia Tech, and Dr. Larry Lambe, MSSRC Inc., for their invaluable discussions on the topic.

References

[1] S. Steinberg, P.J. Roache, Symbolic manipulation and computational fluid dynamics, J. Comput. Phys. 57 (2) (1985) 251–284, http://dx.doi.org/10.1016/0021-9991(85)90045-2.
[2] K. Salari, P. Knupp, Code verification by the method of manufactured solutions, Tech. rep., Sandia National Laboratories, 2000.
[3] J. Blazek, Computational Fluid Dynamics: Principles and Applications, Elsevier Science, 2005.
[4] C.J. Roy, Review of code and solution verification procedures for computational simulation, J. Comput. Phys. 205 (2005) 131–156.
[5] W. Oberkampf, C.J. Roy, Verification and Validation in Scientific Computing, 2010, Cambridge.
[6] J. Banks, T. Aslam, W. Rider, On sub-linear convergence for linearly degenerate waves in capturing schemes, J. Comput. Phys. 227 (14) (2008) 6985–7002, http://dx.doi.org/10.1016/j.jcp.2008.04.002.
[7] V. Ostapenko, On convergence of high order shock capturing difference schemes, AIP Conf. Proc. 1301 (1) (2010) 413–425, http://dx.doi.org/10.1063/1.3526641.

[8] P.J. Roache, Verification and Validation in Computational Science and Engineering, Hermosa, 1998.
[9] R. Courant, D. Hilbert, Methods of Mathematical Physics, vol. II, Interscience Publishers, 1966.
[10] P. Jenny, B. Müller, Rankine–Hugoniot–Riemann solver considering source terms and multidimensional effects, J. Comput. Phys. 145 (2) (1998) 575–610.
[11] B.D. Rogers, A.G. Borthwick, P.H. Taylor, Mathematical balancing of flux gradient and source terms prior to using Roe's approximate Riemann solver, J. Comput. Phys. 192 (2) (2003) 422–451.
[12] C. Roy, C. Ober, T. Smith, Verification of a Compressible CFD Code Using the Method of Manufactured Solutions, American Institute of Aeronautics and Astronautics, 2002, http://dx.doi.org/10.2514/6.2002-3110.
[13] C. Pozrikidis, Numerical Computation in Science and Engineering, Oxford University Press, 1998.
[14] M.H. Carpenter, J.H. Casper, Accuracy of shock capturing in two spatial dimensions, AIAA J. 37 (9) (1999) 1072–1079, http://dx.doi.org/10.2514/2.835.
[15] S. Godunov, A difference scheme for numerical computation of discontinuous solution of hydrodynamic equations, Math. Sb. 47 (1959) 271–306 (in Russian), translated US Joint Publ. Res. Service, JPRS 7226.
[16] R. Cook, Concepts and Applications of Finite Element Analysis, Wiley, 2001.
[17] G.R. Cowper, Gaussian quadrature formulas for triangles, Int. J. Numer. Methods Eng. 7 (3) (1973) 405–408, http://dx.doi.org/10.1002/nme.1620070316.
[18] J. Gottlieb, C. Groth, Assessment of Riemann solvers for unsteady one-dimensional inviscid flows of perfect gases, J. Comput. Phys. 78 (2) (1988) 437–458, http://dx.doi.org/10.1016/0021-9991(88)90059-9.