

Finite volume methods and the equations of finite scale: A mimetic approach[†]

L. G. Margolin^{1,*},[†] and M. Shashkov²

¹*Applied Physics Division, Los Alamos National Laboratory, Los Alamos, NM 87545, U.S.A.*

²*Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM 87545, U.S.A.*

SUMMARY

After introducing the general concept of mimetic differencing, we focus on two specific methodologies, nonoscillatory methods and finite volume approximations. We provide a brief historical account of the development of these two mimetic strategies. We then describe the extension of these strategies to new techniques, a discrete operator calculus and implicit large eddy simulation. In each case, we provide illustrative examples. Further abstraction of these ideas leads to the concept of equations of finite scale, which we advocate as a more appropriate PDE model for constructing numerical algorithms. Published in 2007 by John Wiley & Sons, Ltd.

Received 22 March 2007; Revised 25 June 2007; Accepted 11 July 2007

KEY WORDS: mimetic approximation; finite volume; nonoscillatory methods

1. INTRODUCTION

After more than a half century of computer simulation, the construction of numerical algorithms remains more of an art form than a science. It appears that computational accuracy and stability are necessary but not sufficient strategies to design effective algorithms. In this paper, we will offer examples of an additional strategy, mimetic approximation, which often provides useful direction for creating and improving numerical algorithms.

The underlying idea of mimetic approximation is to build key physical properties of the PDE model exactly into the algorithm. As an example, consider fluid solvers based on Navier–Stokes equations, which exactly conserve momentum. Consistent discretizations will conserve momentum

*Correspondence to: L. G. Margolin, Applied Physics Division, Los Alamos National Laboratory, Los Alamos, NM, U.S.A.

[†]E-mail: len@lanl.gov

[‡]This article is a U.S. Government work and is in the public domain in the U.S.A.

Contract/grant sponsor: U.S. Department of Energy's NNSA; contract/grant number: DE-AC52-06NA25396

to the level of truncation error. However, algorithms based on finite volume discretizations will conserve momentum to the level of roundoff error. This is important because truncation terms, e.g. as measured by modified equation analysis, estimate the error made in one computational cycle. Although individually small, these errors can accumulate over the course of a simulation to an unacceptably high level.

There are many physical properties that are candidates for mimetic approximation. The importance of formulating the equations in conservation form was emphasized by Lax [1]. The preservation of physical symmetries has been considered in [2, 3]. Algorithms to embed time asymptotic behavior directly or through enslavement have been studied in [4, 5]. We refer the reader to [6–10] and the many references contained therein.

It is not the goal of this paper to provide a comprehensive review of the field. Rather, we will focus on two particular areas, that of finite volume methods and that of nonoscillatory approximations. Although the underlying physical principle of finite volume methods has been long recognized as conservation, it is only recently [11] that the connection between the numerical property of nonoscillatory approximation and the second law of thermodynamics has been appreciated. Also, the relation of artificial viscosity, as employed in Lagrangian/arbitrary Lagrangian Eulerian (ALE) simulations, to monotonicity- and sign-preserving methods now widely used in Eulerian simulations is very recently uncovered [12].

One common interpretation of finite volume methods is that one solves the equations of motion in integral form as opposed to differential form. For example, the Navier–Stokes equations are derived by first writing the conservation laws in integral form, and then taking the limit as the volume of integration tends to zero. However, in the case of discrete algorithms, Δx never tends to zero. This suggests that in deriving the governing equations for simulation, one should retain terms of order in Δx and higher. We refer to these as equations of finite scale (EFS) and propose that these are a more appropriate model for simulation. A technique for deriving the EFS based on renormalization is described in [12, 13].

The origin of nonoscillatory algorithms lies in the earliest days of computing with the introduction of artificial viscosity by von Neumann and Richtmyer [14]. The origins of finite volume approximations are more hazy, but date back before 1960 [15]. In the next section, we will recall some early results in these areas. In Section 3, we will describe the extension of finite volume ideas to develop a discrete operator calculus by enforcing integral relations among the fundamental operators: gradient, divergence and curl. In Section 4, we will explain how finite volume techniques and nonoscillatory approximation combine to produce a new technique for simulating high Reynolds' number turbulence—implicit large eddy simulation (ILES). Justification of this technique then leads naturally to the idea of EFS. In Section 5, we apply the ideas of EFS to the well-known process of convergence testing, with some unexpected results. We conclude the paper in Section 6 with a short discussion of overarching issues and current research.

2. HISTORICAL NOTES

In fluid flows, Reynolds' number represents the ratio of the integral scales where inertial effects dominate to the smaller scales at which viscous dissipation occurs. High Reynolds' number flows, e.g. high-speed flows with shocks and most turbulent flows, present special challenges to simulation; because the physical mechanisms of dissipation cannot be resolved, they must be modeled. Von Neumann's artificial viscosity [14] was the first such model.

Artificial viscosity was introduced to suppress the unphysical oscillations that accompanied one-dimensional Lagrangian simulations of shocks; it is a constructive method to enforce the vanishing viscosity principle numerically. Over more than 50 years, generalizations to the original form have been proposed to steepen the representation of shocks, to extend the formulations to simulations in multiple dimensions and to more general meshes. Von Neumann's artificial viscosity also was the basis of the Smagorinsky model, the first subgrid scale model used for large eddy simulations (LES) of turbulence; see discussions in [12, 16].

The von Neumann viscosity in one dimension, usually denoted by q , is

$$q = c_0 \rho (\Delta x)^2 \left(\frac{\partial u}{\partial x} \right)^2 \quad (1)$$

Here, ρ is density, u is velocity, Δx is the mesh spacing and c_0 is a dimensionless constant of order 1. This form was suggested in [14] without derivation. Kuropatenko subsequently suggested a rationale for this form based on analytic solutions of the Riemann problem applied to the interface between two computational cells; see [17]. Even though no rigorous derivation of this form exists, two essential features of von Neumann's original formulation have survived all subsequent development: (1) the requirement for a mesh-dependent length scale and (2) the nonlinear dependence on the flow field velocity.

The origins of finite volume differencing lie in unpublished reports at the US National Laboratories—Los Alamos, Livermore and Sandia—as well as in complementary work in Russia. Some of the early U.S. history is recounted in [15] while that in Russia is described in [8]. The primary issues at that time concerned estimating a pressure gradient on an irregular mesh for two-dimensional Lagrangian simulations of compressible flow. The first implementation of finite volume approximation estimated the pressure gradient by surrounding each vertex with a control volume, and then converting the volume integral of the pressure gradient into a surface integral of pressure. In this formulation, momentum is exactly conserved by detailed balance *independent of the accuracy of the estimate*. Many choices of control volumes have been made and each choice leads to different approximations [18].

3. DISCRETE OPERATOR CALCULUS

The early development of a discrete operator calculus, also termed compatible differencing or the support operator method, proceeded independently in Russia and in the United States, but with very similar results. The underlying idea of the discrete operator calculus is that certain analytic relationships between the fundamental first-order differential operators divergence, gradient and curl, should be maintained between their discrete analogs. We will illustrate this idea with the formulation of the discrete divergence and discrete gradient operators, whose analytic counterparts are adjoint to each; this analytic relationship is instrumental in the demonstration of the conservation of energy.

In a Los Alamos report [19], and independently in a Russian journal article [20], a finite volume divergence operator DIV was constructed for a staggered mesh Lagrangian algorithm on a logically structured mesh by enforcing Reynolds' transport theorem at the discrete level. That is, the analytic

relation in two spatial dimensions (x, y)

$$\frac{dV}{dt} = \int_{V(t)} \operatorname{div} \mathbf{u} \, dV(x', y') \tag{2}$$

is translated directly into the definition

$$(\operatorname{DIV} \mathbf{u})_{i,j} \equiv \frac{1}{V_{i,j}} \sum_{\alpha \in A} \left[\frac{\partial V_{i,j}}{\partial x_\alpha} u_\alpha + \frac{\partial V_{i,j}}{\partial y_\alpha} v_\alpha \right] \tag{3}$$

Here, $V_{i,j}$ is the volume of the computational cell with logical indices i, j and (u, v) are the components of velocity, located at the vertices of the cell (denoted by the set A). Equation (3) is a very convenient form to write a discrete divergence because it is coordinate invariant, can be used for Cartesian, cylindrical, spherical, etc. coordinate systems. It is readily extended to more general mesh types and to three dimensions.

Shortly following, a discrete gradient operator was constructed by requiring adjointness to that divergence operator [21]. Although the original motivation for this work was to construct a positive-definite discrete diffusion operator, it was quickly realized that these operators also allowed the exact conservation of energy in Lagrangian simulations that did not utilize a total energy equation. To construct the discrete gradient $\operatorname{GRAD} = (G^x, G^y)$, we note the integral identity (for any scalar function p and vector \mathbf{u} and assuming a zero boundary integral)

$$\int_{V(t)} p(\operatorname{div} \mathbf{u}) \, dV(x', y') + \int_{V(t)} (\operatorname{grad} p) \cdot \mathbf{u} \, dV(x', y') = 0 \tag{4}$$

This identity is used to prove the conservation of total energy for the (analytic) Navier–Stokes equations. We write a discrete analog for an unstructured mesh, estimating the first integral as a sum over all cells \mathcal{B} and the second as a sum over all vertices \mathcal{A} .

$$\sum_{b \in \mathcal{B}} p_b (\operatorname{DIV} \mathbf{u})_b V_b + \sum_{a \in \mathcal{A}} [(G^x p)_a u_a + (G^y p)_a v_a] V_a = 0 \tag{5}$$

Here, V_a are the control volumes associated with the vertices, which we assume cover the mesh,

$$\sum_{b \in \mathcal{B}} V_b = \sum_{a \in \mathcal{A}} V_a$$

By comparing the coefficients of u_a in the first and second sums, we obtain explicit expressions for the components of the gradient of a scalar function p

$$(G^x p)_a = -\frac{1}{V_a} \sum_{b \in B_a} \frac{\partial V_b}{\partial x_b} p_b \tag{6}$$

$$(G^y p)_a = -\frac{1}{V_a} \sum_{b \in B_a} \frac{\partial V_b}{\partial y_b} p_b \tag{7}$$

Here, B_a is the set of cells that share the vertex a . As is the case of divergence, the coefficients in the above expressions depend only on the geometry of the mesh and are generally applicable to any cell-centered scalar field. More details of these specific calculations and their import to the conservation of energy can be found in [7, 15]. Equivalent results are found at even earlier dates in the Russian literature in [20].

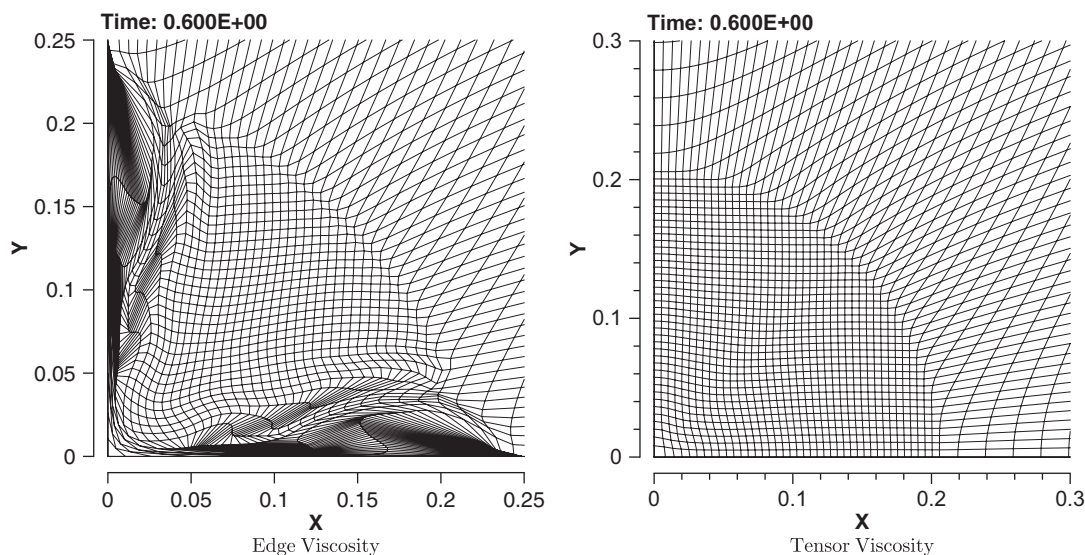


Figure 1. Two simulations of the Noh problem on a cartesian mesh (slightly different scales).

We will illustrate the effectiveness of the discrete operator calculus with an example related to the construction of a tensor artificial viscosity. One of the continuing issues in formulating an artificial viscosity for multidimensional Lagrangian calculations is to minimize the dependence of the solution on the relation of the grid to the flow symmetries, a problem termed mesh imprinting. In [22], a tensor viscosity is formulated by systematic construction of the discrete divergence of a tensor and the compatible discrete gradient of a vector. The derivation is algebraically complex requiring the use of local coordinate systems and a metric tensor, but otherwise follows the same ideas as described earlier in this section. For details of this derivation, the reader is referred to [22]. Figure 1 compares the mesh resulting from simulations of the Noh problem [23] using a standard edge viscosity (left panel) with that using the new tensor viscosity (right panel). The Noh problem consists of a converging shock with point symmetry and is a standard for testing a code's ability to preserve symmetry and for evaluating the effects of wall heating. Here, we use a cartesian grid that does not mirror the expected symmetries of the flow. In the left panel, jets have formed along the coordinate axes, resulting in a highly deformed grid. In the right panel, the new viscosity has a circular shock of the correct radius and a smooth grid behind the shock.

There are many other examples of the benefits of the discrete operator calculus; e.g. applications to Maxwell's equations [24], to diffusion equations [25], etc. Another related issue concerns the null space of discrete operators. Because of the finite size of a computational cell, the null space of the discrete gradient may be larger than that of the analytic gradient. On quadrilateral Lagrangian meshes, this enables a mesh distortion known as hourglassing. A method for mitigating hourglass patterns is described in [26]. The importance of preserving vector identities was also observed in [27, 28]. The unifying theme of the results of this section is the formulation of the targeted mimetic property in integral form.

4. EQUATIONS OF FINITE SCALE

In the early 1990s, an unexpected result arose from the community doing LES of turbulent flow. As noted in the previous section, the dissipative scales in most turbulent flows of practical interest cannot be resolved in numerical simulation, which then requires modification of the governing PDE to include subgrid models that account for the effects of the unresolved scales of motion. The unexpected result, documented independently by several authors, was that simulations based on a particular class of fluid solvers could produce excellent results with no need for any explicit subgrid scale model. The results encompassed a variety of algorithms applied to a diverse spectrum of problems. The commonality is that all solvers were based on nonoscillatory finite volume (NFV) schemes. Details of this early work and many examples of applications can be found in [29] and the references therein.

Despite the many successes and the computational advantages of this approach, the lack of a formal mathematical basis has impeded the acceptance of ILES by the mainstream of the turbulence modeling community, even though the underlying NFV schemes are widely used in computational fluid dynamics. In 2000, a rationale for ILES was put forward [13]. Noting that the dependent variables in finite volume codes are averages of quantities over the computational cells, e.g.

$$\bar{u} \equiv \frac{1}{\Delta x} \int_{x-\Delta x/2}^{x+\Delta x/2} u(x') dx' \quad (8)$$

and so \bar{u} will explicitly depend on Δx for any nonlinear solution u , the authors proposed:

- that the Navier–Stokes equations are not the best choice of PDE model upon which to base the algorithms, and that a more appropriate model should depend explicitly on the size of the cells. The proposed model was termed an EFS.
- that NFV methods were successful *because they accurately solved the EFS*.

The body of [13] contains two new results. First, the authors derive the EFS for a one-dimensional fluid whose every point obeys Burgers' equation. Second, the authors derive the modified equation for a particular NFV solver, MPDATA [30] applied to Burgers' equation. The EFS, keeping only space-averaged terms, is

$$\frac{\partial \bar{u}}{\partial t} = -\bar{u} \frac{\partial \bar{u}}{\partial x} + \nu \frac{\partial^2 \bar{u}}{\partial x^2} - \frac{\partial Q}{\partial x} \quad (9)$$

where L is the averaging length, ν is the viscous coefficient and

$$Q \equiv \frac{1}{6} \left(\frac{L}{2} \right)^2 \left(\frac{\partial \bar{u}}{\partial x} \right)^2 \quad (10)$$

That is, the EFS is Burgers' equation augmented by one additional term of $\mathcal{O}(L^2)$. Note the congruence of Q with q in Equation (1), if L is identified with the cell size Δx .

The modified equation analysis in [13] of the MPDATA algorithm reveals three types of truncation terms that are of order in Δx^2 . First, there is a term proportional to $\partial Q / \partial x$, which arises directly from the finite volume character of MPDATA. Second, there is a similar term that arises from the nonoscillatory property of MPDATA, but that is absolutely dissipative on a *cell-by-cell* basis. Third, there is a dispersive term that represents numerical error. The similarity

of the EFS and the MPDATA modified equation forms the basis of the authors' rationale for ILES.

We now offer some general observations about the EFS and their relation to NFV methods.

- The additional terms of the EFS arise directly from the nonlinearity of the advective terms. Thus, similar terms will appear in the EFS of both the incompressible and the compressible Navier–Stokes equations. The EFS for two-dimensional incompressible Navier–Stokes has been derived in [12].
- The derivation of the EFS is an analytic result, valid for flows of any Reynolds' number whether laminar or turbulent, and independent of numerical issues.
- The length scale L in Equation (10) should be associated with an observer when comparing with experimental results, or with the mesh size when constructing numerical algorithms. That the solution of the EFS depends on L mirrors the fact that different observers will see different results. This observation, though somewhat philosophical, has very practical implications for convergence testing that we will discuss in the next section.
- The appearance of the $\partial Q/\partial x$ term is not unique to MPDATA, but is generic to all NFV schemes; this is a direct consequence of finite volume approximation. In other words, *all finite volume schemes solve EFS that depend explicitly on Δx* .
- The local dissipative property of NFV schemes is a sufficient *but not necessary* condition to insure that simulations obey the second law of thermodynamics. ILES simulations based on different NFV algorithms are essentially similar, but not identical. One might hypothesize that NFV algorithms whose dissipation is order Δx^3 or higher would exhibit some advantages. This hypothesis is confirmed in [12].

One of the primary functions of explicit subgrid scale models is to dissipate kinetic energy at a physically correct rate. In the following calculation, we demonstrate that an ILES simulation of decaying turbulence also dissipates energy at the physically correct rate. Kolmogorov derived an exact relation between the third-order global moment of the longitudinal velocity and the rate of energy dissipation [31]. We have analyzed a simulation of decaying turbulence in a three-dimensional box with periodic boundary condition. The simulation is inviscid—that is,

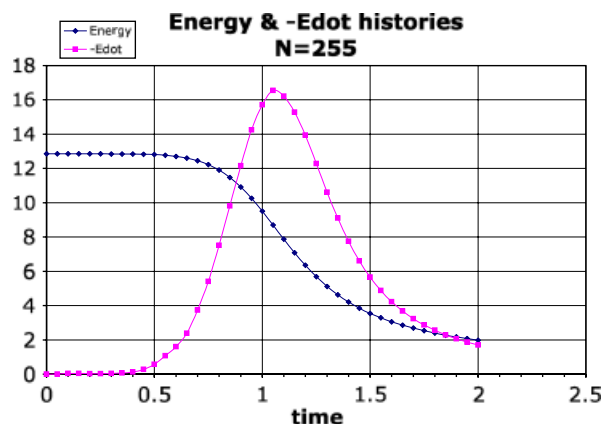


Figure 2. Time history of the total energy E_{tot} and energy dissipation rate $-\text{Edot}$ for $N = 255$.

Table I. Ratio of the left and right-hand-sides of Equation (11) versus time, verifying Kolmogorov's 4/5th law.

Time	1.00	1.25	1.50	1.75	2.00
$-15\langle u_x^3 \rangle \delta x^2 / (4\varepsilon)$	0.785	0.933	1.028	1.054	1.019

the physical viscosity $\nu = 0$. The mesh consists of 255^3 cells with uniform spacing Δx in all directions. The time history of energy and energy dissipation are shown in Figure 2. More description of the simulation can be found in [32]. Keeping in mind that we used uniform spacing Δx , and that the turbulence is assumed (in the law) to be isotropic, the 4/5th law can be written as

$$3\langle u_x^3 \rangle = -\frac{4}{5} \frac{\varepsilon}{\Delta x^2} \quad (11)$$

where $\varepsilon = \partial/\partial t(E_{\text{tot}})$ can be found in Figure 2. In Table I, we show the ratio of the left- and right-hand sides of Equation (11). At time $t = 1.0$, the ratio is less than unity, probably indicating that the flow is not yet sufficiently isotropic. At later times, the agreement is excellent. The satisfaction of the 4/5th law demonstrates that energy dissipation rate is independent of the viscosity and is controlled in simulation, as it is in theory, by the large scales of the flow.

5. CONVERGENCE TESTING

Convergence testing is a strategy commonly used to verify fluid codes. One begins by choosing a problem to which the exact solution—which we will term *truth*—is known. One next simulates that problem on a sequence of grids of increasing resolution. Comparison of the simulation results with *truth* produces a set of errors as a function of resolution. If the error decreases as the resolution increases, one says the code is convergent for that test problem. In the case of a convergent code, one can quantify the result by plotting the error as a function cell size Δx . If the graph can be fit by the model error $\sim (\Delta x)^p$, then one says the code is in an asymptotic regime for that problem, and has an order of convergence p .

In the previous section, we pointed out that for finite volume simulations, *truth* should depend on the resolution. This suggests two modifications to the standard convergence testing process. First, to calculate error, one should integrate the exact solution over the volume of the computation cell [in analogy with Equation (8)] before comparing with the calculated value. Since the difference between a point value, say at the center of a cell, and the average over the cell is of order Δx^2 , the importance of this modification seems clear. Our second suggestion is less obvious. We propose that the results of simulations at all resolutions be moved to the coarsest mesh before evaluating the error. This has both philosophic and practical consequences.

Consider that, from the point of view of the test problem, there is a unique specification of the discrete problem for each resolution. However, from the point of view of the coarsest resolution, there are many equivalent problems at the finer resolutions, each corresponding to a different choice of the initial conditions of the scales unresolved on the coarse mesh. Thus, the error of the simulation *on the coarse mesh* is the sum of two terms—discretization error and uncertainty due to unknown initial conditions. By moving each of the simulation results to the coarsest mesh, we average out the smaller scales and eliminate them as a source of error in studying convergence, thus isolating the discretization error.

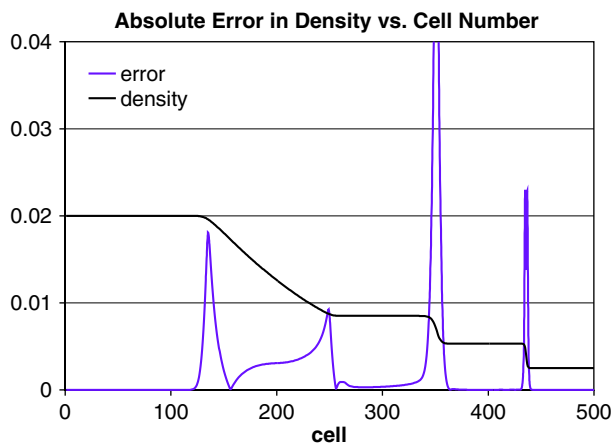


Figure 3. Showing the spatial distribution of absolute value of error for the $N = 512$ simulation. A notional profile of density is superposed, illustrating error is largest at the contact and the shock.

To illustrate these ideas, we present results from a convergence study of simulations of a simple shock tube. This flow, with parameters described in [33], is a popular test problem as it combines three fundamental fluid structures—shocks, contacts and rarefactions. Our simulations employed an Eulerian framework, realized as a Lagrangian integration plus a total rezone of the grid. As we do not consider the errors in detail, nor compare with other code results, we omit further details of the method.

Our convergence study consists of a sequence of seven runs of the Sod problem, with the coarsest mesh having $N = 128$ cells, then refined successively by a factor of 2; the finest mesh is $N = 8192$. Because the solution is self-similar, it is sufficient to chose a problem time long enough that initial conditions are forgotten and short enough that the flow at the boundaries is undisturbed. A notional profile of density (i.e. with arbitrary scale) is shown in Figure 3, with the absolute value of error superposed. The figure shows that the largest errors occur at the contact (as would be expected in an Eulerian simulation), and the shock. However, there is significant error within the rarefaction as well. We assess error in an \mathcal{L}^2 norm using the density field in three ways:

- Compute the difference of the numerical result ρ_i with the cell-centered value of *truth*.
- Compute the difference of ρ_i with *truth* averaged over the cell.
- Move each of the finer resolutions to the coarsest mesh while conserving mass, and then compute the difference of this averaged numerical result with *truth* averaged over the coarsest mesh.

The results of these three measures of error are shown in the convergence plot in Figure 4. The first two measures of error are essentially identical; they indicate an order of convergence as $p \approx 0.58$. This is in contrast to the coarsened measure of error, which indicates $p \approx 1.12$. That is, our suggested procedure for convergence studies indicates an order of convergence *twice as large* as the more conventional procedure.

We remark that while averaging filters the smaller scales of motion from the solution, it does not remove the dynamical effects of those scales, i.e. the backscatter. The source of these effects is the nonlinearity of the equations, e.g. as manifested in the advective terms. The strength of these

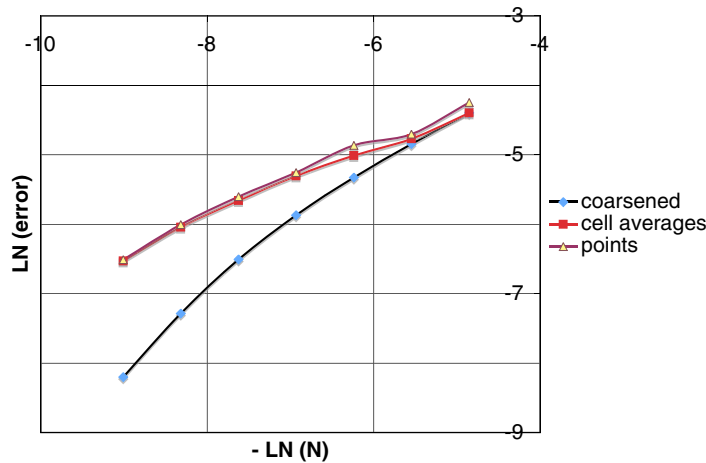


Figure 4. Convergence diagram for the Sod problem for three different \mathcal{L}^2 measures of error.

effects then will be most important in high Reynolds' number flows. Ultimately, in the limit of turbulent flow, it is necessary to extend the averaging process to the integral scales of the problem and to consider the convergence of global moments of the flow.

The previous paragraph suggests that one might deliberately perturb the initial conditions on the more highly resolved meshes to assess the sensitivity of the coarse solution to unresolved scales, leading to a constructive process for estimating the uncertainty of simulation.

6. DISCUSSION

In this paper, we have offered two general strategies for constructing numerical algorithms. First, we described mimetic methods that build selected physical properties of the model equations directly into the numerical approximations. Second, we argued that the model equations themselves should be written in integral form employing no assumptions about the smallness of the volumes. We termed these the EFS. To illustrate these ideas, we focused on two examples of mimetic differencing—finite volume methods and nonoscillatory approximation. For each of these, we provided a short historical context, and then followed their evolution into more modern applications. We included computational examples to illustrate their effectiveness.

The development of a discrete operator calculus described in Section 3 made use of the fundamental definitions of the first-order differential operators in integral form. Most of these results were derived to facilitate spatial differencing on irregular meshes that result from Lagrangian and ALE codes. While the original concept of finite volumes in Section 2 employed detailed balance to ensure exact conservation of momentum, our extensions invoked adjointness to ensure the discrete analog of integration by parts, leading to the conservation of total energy. We note that exact conservation in either case does not imply accuracy. In particular, artificial viscosity is required to damp unphysical oscillations in Lagrangian simulations.

The extension of these results to non-Lagrangian frameworks, and in particular to the terms representing material advection, involved the additional constraint of nonoscillatory differencing.

Modified equation analysis reveals that such algorithms implicitly contain terms analogous to artificial viscosity. This realization led to the idea that the origin of these terms lay in the model PDEs themselves, when appropriate modifications are made to account for the definition of finite volume variables, e.g. Equation (8). We have termed these the EFS.

Current efforts to improve the mimetic framework for algorithm development continue in several directions. In Section 5, we advocated several modifications to the standard process of convergence testing with the goal of separating the discretization error (an issue of numerics) from the uncertainty of unknown initial conditions (an issue of physics). We also indicated how the modified procedures could lead to a constructive approach to estimating simulation uncertainty.

A second direction concerns the derivation of the EFS for compressible flow. One immediate result here is the appearance of an artificial heat conduction, arising from the advective terms of the energy equation. Such a term was advocated by Noh [23], but is not yet commonly used in modern Lagrangian algorithms. However, this research is complicated by the presence of the equation of state, whose finite volume extension requires input from nonequilibrium thermodynamics. For example, kinetic energy dissipated by the subgrid scale models of the EFS (i.e. terms depending on Δx) may require a finite time to show up as internal energy where it affects the macroscopic pressure.

We end our discussion with the following quote from Phil Browne, one of the pioneers of finite volume differencing [34].

By integrated gradients, we mean taking a small element and applying the physics to it to get the difference equation directly. This seems like a good approach because we leave out the step of shrinking quantities to zero (which often drops out terms that are important as the difference equations are applied over and over again in time-dependent problems).

ACKNOWLEDGEMENTS

We appreciate the help of P. J. Blewett, E. J. Caramana, and P. P. Whalen in reconstructing the early history of finite volume differencing. This work was performed under the auspices of the U.S. Department of Energy's NNSA by the Los Alamos National Laboratory operated by Los Alamos National Security, LLC under contract number DE-AC52-06NA25396.

REFERENCES

1. Lax PD. Weak solutions of nonlinear hyperbolic equations and their numerical computation. *Communications on Pure and Applied Mathematics* 1954; **7**:159–193.
2. Caramana EJ, Whalen PP. Numerical preservation of symmetry properties of continuum problems. *Journal of Computational Physics* 1998; **146**:174–198.
3. Margolin L, Shashkov M. Using a curvilinear grid to construct symmetry-preserving discretizations for Lagrangian gas dynamics. *Journal of Computational Physics* 1999; **149**:389–417.
4. Jin S, Levermore CD. Numerical schemes for hyperbolic conservation laws with stiff relaxation terms. *Journal of Computational Physics* 1996; **126**:449–467.
5. Jones DA, Margolin LG, Poje AC. Accuracy and nonoscillatory properties of enslaved difference schemes. *Journal of Computational Physics* 2002; **181**:705–728.
6. Shashkov M. *Conservative Finite-difference Methods on General Grids*. CRC Press: New York, 1996.
7. Margolin LG, Shashkov M, Smolarkiewicz PK. Discrete operator calculus for finite difference approximations. *Computer Methods in Applied Mechanics and Engineering* 2000; **187**:365–383.
8. Samarskii AA. *The Theory of Difference Schemes*. Marcel Dekker, Inc: New York, Basel, 2001.

9. Hyman JM, Shashkov M. Natural discretizations for the divergence, gradient and curl on logically rectangular grids. *International Journal of Applied Numerical Mathematics* 1997; **33**:81–104.
10. Hyman JM, Shashkov M. The adjoint operators for the natural discretizations of the divergence, gradient and curl on logically rectangular grids. *Applied Numerical Mathematics* 1997; **25**:413–442.
11. Merriam ML. Smoothing and the second law. *Computer Methods in Applied Mechanics and Engineering* 1987; **64**:177–193.
12. Margolin LG, Rider WJ, Grinstein FF. Modeling turbulent flow with implicit LES. *Journal of Turbulence* 2006; **7**:1–27.
13. Margolin LG, Rider WJ. A rationale for implicit turbulence modeling. *International Journal for Numerical Methods in Fluids* 2002; **39**:821–841.
14. von Neumann J, Richtmyer RD. A method for the numerical calculation of hydrodynamic shocks. *Journal of Applied Physics* 1950; **21**:232–237.
15. Bauer AL, Burton DE, Caramana EJ, Loubere R, Shashkov MJ, Whalen PP. The internal consistency, stability, and accuracy of the discrete, compatible formulation of Lagrangian hydrodynamics. *Journal of Computational Physics* 2006; **218**:572–593.
16. Smagorinsky J. The beginnings of numerical weather prediction and general circulation modeling: early recollections. *Advances in Geophysics* 1983; **25**:3–37.
17. Wilkins ML. Use of artificial viscosity in multidimensional fluid dynamic calculations. *Journal of Computational Physics* 1980; **36**:281–303.
18. Margolin LG, Nichols BD. Fundamental considerations in the integration of the momentum equations in finite difference codes. *Proceedings of the 3rd International Conference on Numerical Methods in Laminar and Turbulent Flow*, University of Washington, Seattle, WA, 1983; 411–421.
19. Margolin LG, Adams TF. Spatial differencing for finite difference codes. *Report LA-10249*, Los Alamos Scientific Laboratory, 1985.
20. Samarskii A, Tishkin V, Favorskii A, Shashkov M. Operational finite difference methods. *Differential Equations* 1981; **17**:854–862.
21. Margolin LG, Tarwater AE. A diffusion operator for Lagrangian codes. In *Numerical Methods for Heat Transfer*, Lewis RW, Morgan K, Habashi WG (eds). Pineridge Press: Swansea, 1987; 1252–1260.
22. Campbell JC, Shashkov MJ. A tensor artificial viscosity using a mimetic finite difference algorithm. *Journal of Computational Physics* 2001; **172**:739–765.
23. Noh WF. Errors for calculations of strong shocks using an artificial viscosity and an artificial heat flux. *Journal of Computational Physics* 1978; **72**:78–120.
24. Hyman JM, Shashkov M. Mimetic discretizations for Maxwell's equations. *Journal of Computational Physics* 1999; **151**:881–909.
25. Lipnikov K, Morel J, Shashkov M. Mimetic finite difference methods for diffusion equations on non-orthogonal non-conformal meshes. *Journal of Computational Physics* 2004; **199**:589–597.
26. Margolin LG, Pyun JJ. A method for treating hourglass patterns. In *Numerical Methods in Laminar and Turbulent Flows*, Taylor C, Hafey MM (eds). Pineridge Press: Swansea, 1988; 149–160.
27. Chorin AJ. Numerical solution of Navier–Stokes equations for an incompressible fluid. *Bulletin of American Mathematical Society* 1967; **73**:928.
28. Brackbill JU, Barnes DC. The effect of nonzero-del.B on the numerical solution of the magneto-hydrodynamic equations. *Journal of Computational Physics* 1980; **35**:426–430.
29. Grinstein FF, Margolin LG, Rider WJ. *Implicit Large Eddy Simulation*. Cambridge University Press: Cambridge, 2007.
30. Smolarkiewicz PK, Margolin LG. MPDATA: a finite difference solver for geophysical flows. *Journal of Computational Physics* 1998 **140**:459–480.
31. Frisch U. *Turbulence*. Cambridge University Press: Cambridge, U.K., 1998; 76–77.
32. Margolin LG, Smolarkiewicz PK, Wyszogradzki AA. Dissipation in implicit turbulence models: a computational study. *Journal of Applied Mechanics* 2006; **73**:469–473.
33. Sod GA. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *Journal of Computational Physics* 1978; **27**:1–31.
34. Browne PL. A derivation of some difference forms for the equation of motion for compressible flow in two-dimensional Lagrangian hydrodynamics using integration of pressures over surfaces. *Report LA-10587-MS*, Los Alamos Scientific Laboratory, 1986.