A machine learning approach for detecting shocks with high-order hydrodynamic methods

Conference Paper · December 2019

DOI: 10.2514/6.2020-2024

4 authors, including:

Nathaniel Morgan
Los Alamos National Laboratory
117 PUBLICATIONS **980** CITATIONS
SEE PROFILE

Svetlana Tokareva
Los Alamos National Laboratory
41 PUBLICATIONS **530** CITATIONS
SEE PROFILE

Xiaodong Liu
Remcom
59 PUBLICATIONS **546** CITATIONS
SEE PROFILE

A machine learning approach for detecting shocks with high-order hydrodynamic methods

Nathaniel R. Morgan * and Svetlana Tokareva † and Xiaodong Liu ‡ and Andrew D. Morgan §

Artificial neural networks (ANNs) can be trained to recognize patterns within data. In this work, we train an ANN to detect cells that have a shock, so that it can be used as a troubled cell indicator with high-order hydrodynamic methods that must either apply limiters to polynomials or activate an artificial viscosity scheme near a shock depending on the type of numerical method. The overarching goal is to enable high-order solutions on smooth flows and to reduce the accuracy towards first-order near shocks. We train an ANN on a dataset consisting of shocks and smooth field variations. The ANN returns a scalar value in the range of zero to one, where the scalar value is very close to zero in a smooth region and is very close to one at a shock. The ANN shock detector is used with a high-order residual distribution (RD) Lagrangian hydrodynamic method to simulate multidimensional shock driven flows. The details on the ANN shock detector are presented along with simulation results from test problems.

Nomenclature

ANN	Artificial neural network
ML	Machine learning
w_{ij}	Weights in the ANN
F	Activation function
a_i	Inputs to a hidden layer
b_j	Outputs from a hidden layer
o	Output from the ANN
τ	A target value in a training set
(x, y)	Current coordinates
(X, Y)	Initial coordinates
Ω	Reference cell
(ξ, η)	Reference coordinates
$\mathbf{J}$	Jacobian matrix
j	Determinant of the Jacobian matrix
$\mathbf{u}$	Velocity
ρ	Density
$\boldsymbol{\sigma}$	Stress tensor
p	Pressure
e	Specific internal energy
c	Sound speed
μ	Shock impedance
λ	A shock detector
ω	User coefficient in the analytic shock detector
t	Time
DOF	Degree of freedom
$\mathbf{M}_{pq}$	Mass matrix
M_p	Lumped mass at kinematic node
M_k	Lumped mass at an internal energy node
R	Residual
β	Distribution coefficient for velocity
α	Distribution coefficient for specific internal energy
ϕ	Kinematic basis function
θ	Specific internal energy basis function
Superscript	
r	Iteration level
n	Time level
m	Sub-intervals in time
Q	The residual includes dissipation
Subscript	
h	Cell
p	An element vertex
g	A Gauss quadrature point
s	A value on the 4x4 stencil in a cell

*Scientist, X-Computational Physics Division. Corresponding author: nmorgan@lanl.gov
†Scientist, Theoretical Division
‡Post-doc research associate, Theoretical Division
§Student, Los Alamos Middle School

I. Introduction

A range of Lagrangian hydrodynamic methods exist for simulating gas and solid dynamics problems. The cell-centered finite volume (FV)[18] and modal discontinuous Galerkin (DG)[5,7–13,17,25] Lagrangian hydrodynamic methods represent the fields (e.g., velocity) with a Taylor-series polynomial that is discontinuous at the cell boundaries. Limiters must be applied near shocks to reduce the polynomials toward a piecewise constant field, which in turn, will create dissipation in the calculation through the Riemann solver on the cell boundary. A challenge is that the existing limiters can incorrectly treat smooth flow features like acoustic waves as shocks resulting in reduced accuracy. To resolve this issue, one can use a shock detector (also called a 'troubled cell' indicator) and only limit the solution near a shock. The Lagrangian staggered-grid FV and finite element (FE) hydrodynamic methods typically add dissipation to a calculation through an artificial viscosity term that can be activated using a shock detector. One can also use a Riemann solver and a limiter with FV and FE hydrodynamic methods to add dissipation to a calculation.[2,16] The Lagrangian residual distribution (RD) method[1] uses a low-order solution that has an artificial viscosity term, and then one performs a series of iterations to yield a higher-order solution if the solution is smooth; as a result, a shock detector is required with the RD method. Many researchers are pursuing shock detectors for use with Lagrange methods[2,3,13] to identify the cells that require, for instance, limiting or application of artificial viscosity. However, these existing shock detectors require tunable parameters that are problem dependent. To address this challenge, we propose a novel approach for detecting shocks using an artificial neural network (ANN) that is suitable for complex flows that have both shocks and smoothly varying fields.

The first ANN troubled cell indicator was proposed by Ray and Hesthaven[21] and was used with a 1D Eulerian RKDG method to simulate 1D test cases. They trained an ANN to detect a troubled cell based on the cell averages of the neighboring cells, and the forward and backward differences at the face, so there are a total of 5 inputs in their ANN. Veiga and Abrall[24] investigated using a neural net to identify troubled cells with a 1D DG scheme. A recent project for the supercomputing challenge by Morgan[15] developed an ANN shock detector using 4 field values in 1D as the inputs. In that study, the ANN was very successful at identifying a 1D shock. In this work, we train and use an ANN to detect cells near a shock on 2D gas dynamics problems. A high-order field in a cell is calculated on a subcell stencil (e.g., 4x4) and is then scaled by the average. The scaled values on the subcell stencil are the inputs to the ANN, which returns a cell centered value in the range of zero to one. A value close to zero corresponds to a smooth variation in the field and a value close to one is a shock (i.e., a troubled cell). The ANN is trained using results from calculations that had an analytic shock detector[2,3] that was adjusted for each training test case to give the best possible answers. Once trained, the ANN shock detector can be applied to disparate types of simulations without any adjustments. A merit of the proposed ANN shock detector is that it is local to a cell so that it can be applied at mesh boundaries, such as a free surface or a contact surface in a Lagrangian hydrodynamic calculation where the neighboring cell-centered data is sparse.

The objective of this paper is to demonstrate the viability of using an ANN to identify the cells near a shock in complex flows to enable high-order solutions with a Lagrangian hydrodynamic method away from the shock. To accomplish this goal, the ANN is trained on several very challenging shock driven flows and then the ANN shock detector is used with a high-order Lagrangian RD hydrodynamic method on two very complex vortical flow problems. All calculations in this paper are performed using quadratic quadrilateral cells built from a tensor product of Bernstein polynomials. A low-order solution is used in the cells that are near a shock and a high-order solution is used everywhere else in the mesh. The results presented will show that the ANN shock detector is a promising technology for accurate and robust calculations with high-order hydrodynamic methods.

II. Artificial neural network

ANNs consist of a series of nodes (artificial neurons) that are connected together (Fig. 1). The nodes are arranged in layers, where the layers between the input and output layers are termed the hidden layers. The value at a node is calculated by adding together the inputs to this node multiplied by weights and then evaluating this sum in a function that is called an activation function. One activation function is the sigmoid function, which yields a value in the range of zero to one. The mathematical formula for obtaining a result at a node j in layer l of the ANN can be expressed as,

$$b_j^l = F\left(\sum_i w_{ij}^l a_i^{l-1}\right), \tag{1}$$

where a_i^{l-1} is an input from the $(l-1)^{th}$ layer, w_{ij}^l is the corresponding weight for the connection between node i on the previous layer and node j, F is the activation function, b_j is the result at node j, and the summation is over the nodes in the previous layer $l-1$. Every node in the ANN uses Eq. 1 to calculate its nodal value. The process of propagating the global inputs through the ANN is termed forward propagating. The weights w_{ij}^l in the ANN must be calculated to detect the difference between a smooth transition and a shock in a flow (*i.e.,* train the ANN). The next subsection will describe how the weights are calculated.

The ANN in this paper will have 16 inputs, which are the values from a 4x4 stencil inside a cell, and two hidden layers that taper down in size. The first hidden layer has 12 nodes and the second hidden layer has 8 nodes. The output from the ANN is a single value in the range of 0 to 1, and the output can be interpreted as the probability of a cell having a shock. The sigmoid function is used as the activation function in this work.

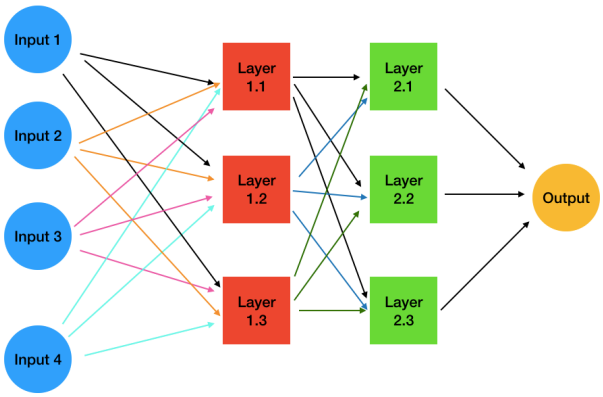


Figure 1: An ANN is shown above that has four inputs and a single output.[15] There are two hidden layers with three nodes in each hidden layer. In this work, the ANN for detecting a 2D shock will have 16 inputs (values from a 4x4 stencil inside a cell) and a single output. Two hidden layers that taper down in size are used in this work where the first hidden layer has 12 nodes and the second hidden layer has 8 nodes.

A. Training the ANN

The weights in the ANN must be calculated to best fit a set of cases covering smooth flows and flows with shocks. The output from the ANN for a smooth flow should be equal to 0 and the output from the ANN for a shock should be equal to 1. The goal is to find the weights that minimize the square of the output error for a training set $\{(\mathbf{s}_1, \tau_1), \ldots, (\mathbf{s}_n, \tau_n)\}$ where $\mathbf{s}_i$ is a vector of known inputs that create a corresponding output τ_1. For this application, there is a single output value τ_i, but for other applications, the output could be a vector. The error for the ANN over the training set is

$$E = \frac{1}{2}\sum_{i \in n} (o_i - \tau_i)^2, \tag{2}$$

where o_i is the output from the ANN, τ_i is the target (the output for training set i), and there are a total of n training sets. The backpropagation algorithm is used to minimize the error from the ANN.[22] We use the python keras package to calculate the weights.

B. Training data

The number of possible cases for training the ANN is potentially quite large. To help address this challenge, but not necessarily solve it, we scale the results on the 4x4 stencil by the average. In this work, we

calculate the density on the 4x4 stencil. The positions of the 4x4 stencil in the reference cell are defined as a tensor product of $[0, 1/3, 2/3, 1]$. The density at a point on the stencil ρ_s is calculated using strong mass conservation,[11]

$$\rho_s = \rho(t, \boldsymbol{\xi}_s) = \frac{\rho(t=0, \boldsymbol{\xi}_s)}{\det\left(\mathbf{J}(t, \boldsymbol{\xi}_s)\right)}, \tag{3}$$

where $\mathbf{J}$ is the Jacobian matrix, which is the gradient of the map $\mathbf{x} = \sum_p \phi_p(\boldsymbol{\xi})\mathbf{x}_p$ between the reference cell and the cell in the deformed physical coordinates. ϕ_p is a Bernstein basis function and $\mathbf{x}_p$ is the position of a vertex of the cell in the physical coordinates. The next section gives more details on the numerical method and the definitions of the variables.

The inputs to the ANN are the ρ_s values (16 of them) and the weights in the ANN are calculated to give 0 if the inputs are from a smooth flow or 1 if the inputs correspond to a shock. To create target data and input data for training the ANN, we use an existing analytic shock detector based on a discrete Mach number[2] that is dialed to give the best answers on a particular test problem. The analytic shock detector λ is the target value and given by,

$$\lambda = \min\left(1, \omega \frac{\hat{l}\,|\nabla \cdot \mathbf{u}|}{c}\right), \tag{4}$$

where ω is a tunable coefficient, c is the sound speed, $\hat{l}$ is the square root of the cell area, and $\nabla \cdot \mathbf{u}$ is the divergence of the velocity in the cell. In short, we train the ANN to match the calculated λ value for the 16 inputs ρ_s. The goal is for the ANN to match the test data and be able to robustly identify trouble cells on a *different set* of complex flows.

C. Lagrangian RD

The traditional high-order Lagrangian finite element method (FE) has a sparse global mass matrix. The Lagrangian RD method is a high-order matrix-free FE method. The RD method uses mass lumping combined with a special time stepping technique to deliver higher-order accuracy.

The FE method approximates the fields in a cell with a polynomial. The velocity field is staggered from the internal energy field. The velocity and specific internal energy polynomials are

$$\begin{aligned} \mathbf{u}(\boldsymbol{\xi}, t) &= \sum_{p}^{O(P_n)} \phi_p\left(\boldsymbol{\xi}\right) \cdot \mathbf{u}_p\left(t\right), \\ e(\boldsymbol{\xi}, t) &= \sum_{k}^{O(P_n-1)} \theta_k\left(\boldsymbol{\xi}\right) \cdot e_k\left(t\right), \end{aligned} \tag{5}$$

where the polynomial for the the specific internal energy field is one order lower than for the velocity field. The basis functions (Bernstein with the RD method) are denoted with ϕ_p (velocity) and θ_k (specific internal energy), and these basis functions are defined on a reference cell. The coefficients are $\mathbf{u}_p$, and e_k.

The Lagrangian FE method creates a system of equations to temporally evolve the nodal coefficients forward in time. The velocity evolution equation is

$$\sum_p \mathbf{M}_{qp} \frac{d\mathbf{u}_p}{dt} = -\sum_{h \ni q} \int_{\Omega_h} (\nabla_\xi \phi_q) \cdot j\mathbf{J}^{-1} \cdot \boldsymbol{\sigma} d\Omega, \tag{6}$$

where $h \ni q$ means the cells that contain q, $\mathbf{M}_{qp} = \sum_{h \ni q} \int_{\Omega_h} \rho \phi_q \phi_p j d\Omega$ is the global sparse mass matrix, $\mathbf{J}$ is the Jacobian matrix, and $j = \det(\mathbf{J})$. The integrals in Eq. 12 are over the reference cell Ω_h. A naive diagonalization (termed mass 'lumping') of the mass matrix will destroy the high-order accuracy. The lumped mass version of the velocity evolution equation is

$$M_p \frac{d\mathbf{u}_p}{dt} = -\sum_{h \ni p} \int_{\Omega_h} (\nabla_\xi \phi_p) \cdot j\mathbf{J}^{-1} \cdot \boldsymbol{\sigma} d\Omega, \tag{7}$$

where M_p is the nodal mass for the degree of freedom (DOF) p and

$$M_p = \sum_q \sum_{h \ni q} \int_{\Omega_h} \rho \phi_q \phi_p j d\Omega. \tag{8}$$

In other words, the nodal mass is a summation over a row in the mass matrix.

The Lagrangian RD method uses a special time integration method that iteratively drives the residual toward zero, where the residual from cell h on DOF q is defined as

$$R^r_{q(h)} = \sum_p \mathbf{M}_{qp} \left(\mathbf{u}^r_p - \mathbf{u}^n_p \right) + \int_{t^n}^{t^m} \int_{\Omega_h} \left(\nabla_\xi \phi_q \right) \cdot j \mathbf{J}^{-1} \cdot \boldsymbol{\sigma} d\Omega dt. \tag{9}$$

The superscript m denotes a sub-interval time in $[t^n, t^{n+1}]$ and there are m equal sub-intervals in the time step for m^{th} order accuracy. The residual is driven toward zero using,

$$M_p \left(\mathbf{u}^{r+1}_p - \mathbf{u}^r_p \right) = \sum_{h \ni p} \beta^r_{p(h)} R^r_h. \tag{10}$$

where $r+1$ is the next temporal iteration level, M_p is the nodal lumped mass (Eq. 8), $\beta^r_{p(h)}$ is a distribution coefficient for cell h to node p, and R^r_h is the cell residual given by $R^r_h = \sum_{p \in h} R^{r,Q}_{p(h)}$. The term $R^{r,Q}_{p(h)}$ is the residual calculated using $R^{r,Q}_{p(h)} = R^r_{p(h)} + \mu_h \left(\mathbf{u}^r_p - \bar{\mathbf{u}}^r_h \right)$ where the bar denotes the cell average and μ_h is a shock impedance multiplied by an area and the time increment. In other words, $R^{r,Q}_{p(h)}$ has an artificial viscosity like term. The distribution coefficients $\beta^r_{p(h)}$ are calculated using,

$$\beta^r_{p(h)} = \frac{\max\left(\frac{R^{r,Q}_{p(h)}}{R^r_h},\ 0 \right)}{\sum_{p \in h} \max\left(\frac{R^{r,Q}_{p(h)}}{R^r_h},\ 0 \right)}. \tag{11}$$

These coefficients satisfy $\sum_{p \in h} \beta^r_{p(h)} = 1$. The temporal order of accuracy of the RD method is equal to the number of iterations, for instance, two iterations is second-order accurate.

The FE specific internal energy evolution equation is

$$\sum_k \hat{\mathbf{M}}_{qk} \frac{de_k}{dt} = \int_{\Omega_h} \theta_q \left(\nabla_\xi \mathbf{u} \right) \cdot j \mathbf{J}^{-1} : \boldsymbol{\sigma}\ d\Omega, \tag{12}$$

where $\hat{\mathbf{M}}_{qk} = \int_{\Omega_h} \rho \theta_q \theta_k j d\Omega$ is a local mass matrix to the cell. The residual with the Lagrangian RD method for an energy DOF is defined as,

$$\hat{R}^r_q = \sum_k \hat{\mathbf{M}}_{qk} \left(e^r_k - e^n_k \right) + \int_{t^n}^{t^m} \int_{\Omega_h} \theta_q \left(\nabla_\xi \mathbf{u} \right) \cdot j \mathbf{J}^{-1} \cdot \boldsymbol{\sigma} d\Omega dt. \tag{13}$$

The internal energy residual is driven toward zero using,

$$M_k \left(e^{r+1}_k - e^r_k \right) = \sum_{c \in p} \alpha^r_{k(h)} \hat{R}^r_h, \tag{14}$$

The lumped mass for the internal energy is $M_k = \sum_q \int_{\Omega_h} \rho \theta_q \theta_k j d\Omega$. The term $\hat{R}^r_h$ is the cell internal energy residual given by $\hat{R}^r_h = \sum_{k \in h} \hat{R}^{r,Q}_{k(h)}$. The coefficient $\alpha^r_{k(h)}$ distributes the residual for the cell to the k^{th} energy

DOF, and these coefficients satisfy $\sum_{k\in h} \alpha^r_{k(h)} = 1$. The internal energy distribution coefficients are calculated using,

$$\alpha^r_{k(h)} = \frac{\max\left(\frac{\hat{R}^{r,Q}_{k(h)}}{\hat{R}^r_h},\ 0\right)}{\sum_{k\in h} \max\left(\frac{\hat{R}^{r,Q}_{k(h)}}{\hat{R}^r_h},\ 0\right)}. \tag{15}$$

$\hat{R}^{r,Q}_{k(h)}$ is the energy residual calculated using $\hat{R}^{r,Q}_{k(h)} = \hat{R}^r_{k(h)} + \mu_h \left(e^r_k - \bar{e}^r_h\right)$ where the bar denotes the cell average and μ_h is a shock impedance multiplied by an area and the time increment.

III. Test problems

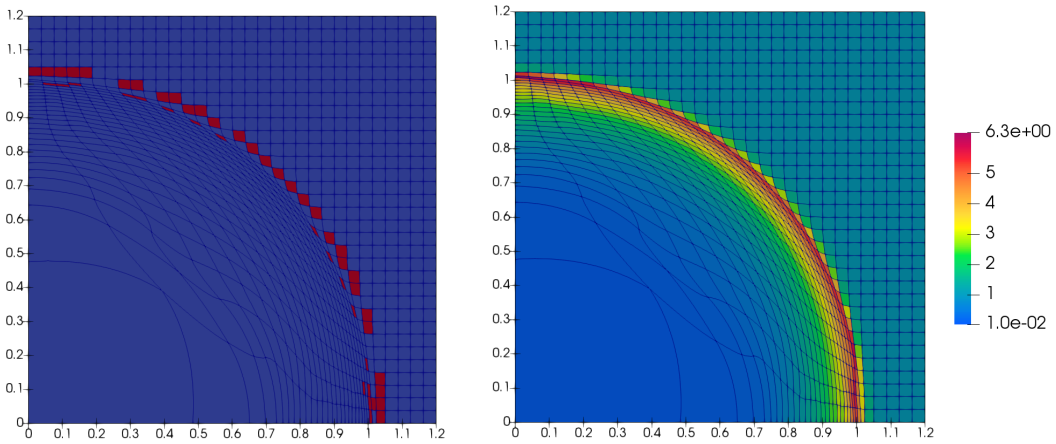
Several challenging multidimensional test problems are calculated to assess the merits of using an ANN to choose when to use a high-order Lagrangian RD method. All calculations use quadratic cells constructed from a tensor product of Bernstein polynomials. The 2D Sedov[20,23] and 2D Noh[19] test cases are calculated to generate training data for the ANN. Then, the 2D Taylor-Green vortex and 2D Triple-point vortex[2,4,6,14,16,26] test cases are calculated to validate the ANN at detecting cells near a shock. The 2D Taylor-Green vortex and 2D Triple-point vortex are vastly different than the training data set to help test the efficacy of using the ANN shock detector.

The weights in the ANN are calculated to match results from the Sedov and Noh test cases. The objective is for the ANN to match an existing shock detector based on a discrete Mach number[2,3] that was adjusted to give the best results for those two test cases. The training data set for Sedov was created using $\omega = 5$ and for Noh was created using $\omega = 10$ (see Eq. 4 for the definition of ω). The results for the Sedov problem are shown in Fig. 2 and the results for the Noh problem are shown in Fig. 3. For both test cases, the ANN shock detector successfully identified the cells near the shock. A low-order Lagrangian RD method (that has artificial viscosity) is used near the shock while a high-order Lagrangian RD method is used away from the shock. Majority of the cells in the mesh are high-order accurate so the results agree favorably with the analytic solution on a coarse mesh.

The trained ANN shock detector was used with the high-order Lagrangian RD hydrodynamic method to simulate the Taylor-Green vortex and triple-point vortex test cases. The goal is to access the efficacy of the ANN shock detector; as such, the ANN was not trained to fit these test cases. The Taylor-Green vortex is a smooth vortical flow and the results in Fig. 4 show that ANN shock detector correctly captures that every cell is a smooth field up to a very late time. The triple-point results in Fig. 5 demonstrate that the ANN shock detector can work exceptionally well on a problem with smooth flow regions, shocks, contact discontinuities, and significant vorticity. For this test case, the ANN shock detector tags the cells that have a strong shock.

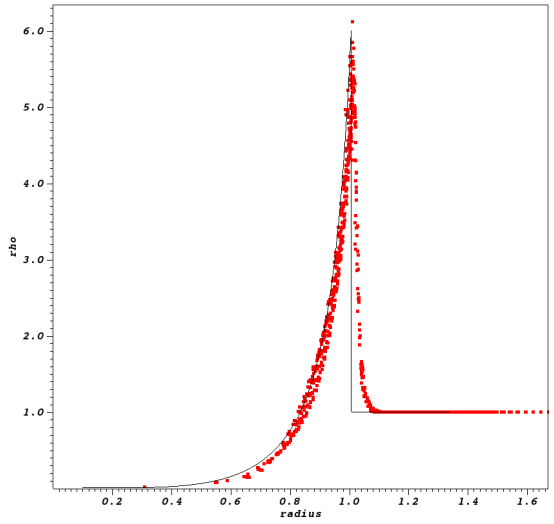
IV. Conclusion

Many researchers are pursuing shock detectors for use with hydrodynamic methods to identify the cells that require, for instance, limiting or application of artificial viscosity. However, these existing shock detectors require tunable parameters that are problem dependent. To address this challenge, we propose a novel approach for detecting shocks using an ANN that is suitable for complex flows that have both shocks and smoothly varying fields. A new ANN shock detector was presented and used with a high-order Lagrangian RD method to simulate shock driven flows. The Lagrangian RD method uses a low-order solution that has an artificial viscosity term, and then a series of iterations are performed to yield a higher-order solution if the flow is smooth. The ANN shock indictor is used to select the cells that have a high-order solution. The ANN was trained on several challenging shock driven flows using an existing analytic shock detector that was adjusted to give the best answers on each test problem. Then, several multidimensional vortical flow test cases were calculated to demonstrate the accuracy and robustness of the ANN at detecting shocks. For these test cases, the ANN shock detector selected the cells near the strong shock, which enabled high-order solutions over majority of the mesh. This work demonstrates the utility of using an ANN as a multidimensional troubled cell indicator. Future work will focus on training the ANN using a larger test suite.



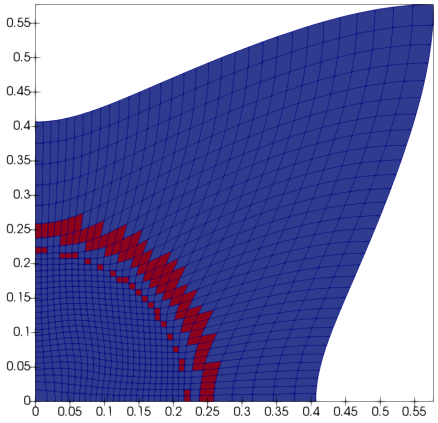
(a) ANN shock detector

(b) Density

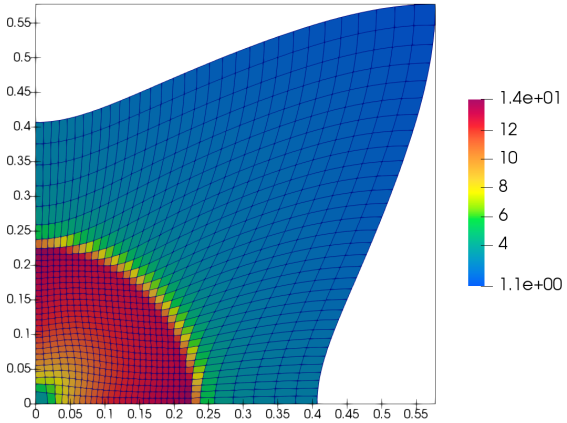


(c) Density scatter

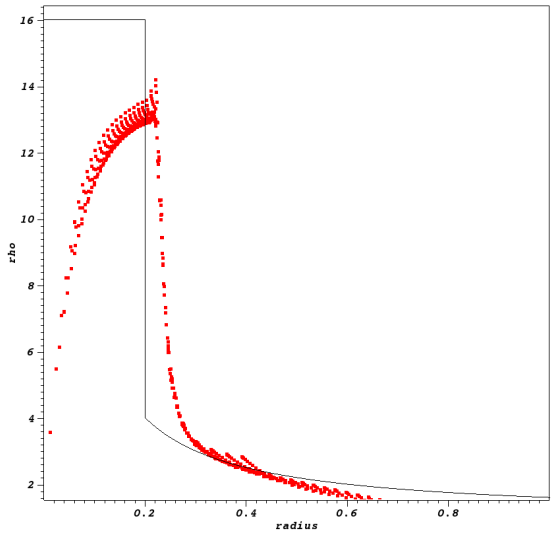
Figure 2: Results are shown for the 2D Sedov problem calculated using a high-order Lagrangian RD SGH method. The ANN shock detector is shown in (a) where the red cells are lower-order and the blue cells are high-order. The density in each cell is shown in (b). The density in every cell is compared to the exact solution as a function of the radius in (c). The mesh resolution is 32x32.



(a) ANN shock detector

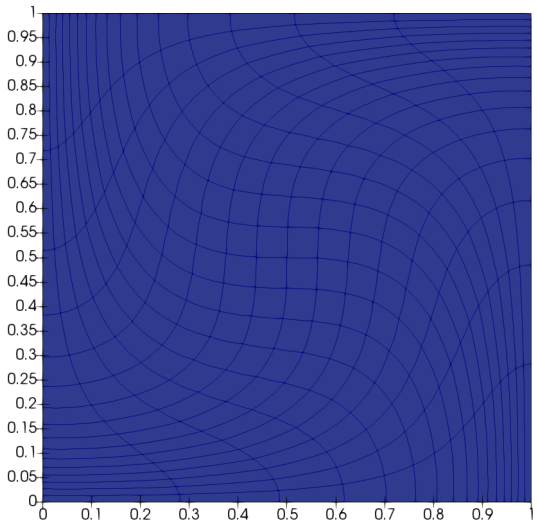


(b) Density

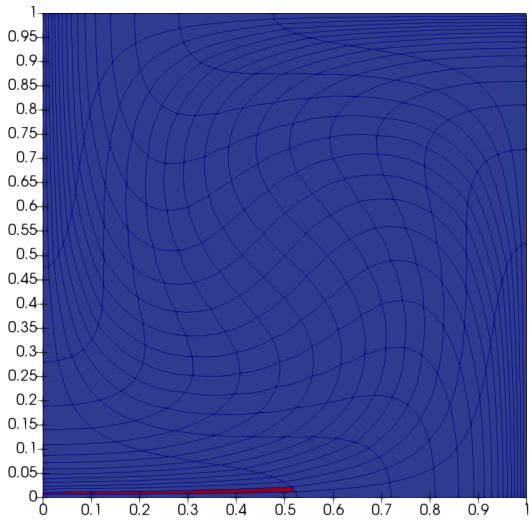


(c) Density scatter

Figure 3: Results are shown for the 2D Noh problem calculated using a high-order Lagrangian RD hydrodynamic method. The ANN shock detector is shown in (a) where the red cells are lower-order and the blue cells are high-order. The density in each cell is shown in (b). The density in every cell is compared to the exact solution as a function of the radius in (c). The mesh resolution is 32x32.

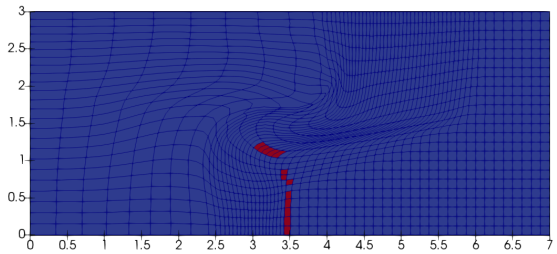


(a) ANN shock detector, t=0.5

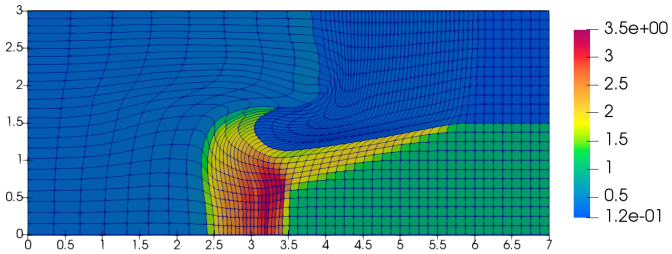


(b) ANN shock detector, t=0.76

Figure 4: Results at t=0.5 (a) and t=0.76 (b) are shown for the Taylor-Green vortex problem calculated using a high-order Lagrangian RD hydrodynamic method. The ANN shock detector correctly identifies that every cell has a smooth field (blue color) at t=0.5 so a high-order solution is obtained on every cell in this test case at that time. At t=0.76, a cell is tagged incorrectly as being troubled.



(a) ANN shock detector



(b) Density

Figure 5: Results at t=3 are shown for the triple point vortex problem calculated using a high-order Lagrangian RD hydrodynamic method. The ANN shock detector is shown in (a) where the red cells are lower-order and the blue cells are high-order. The density in each cell is shown in (b). The ANN shock detector permits a high-order solution over majority of the mesh.

V. Acknowledgments

We gratefully acknowledge the support of the NNSA through the Laboratory Directed Research and Development (LDRD) program at Los Alamos National Laboratory. The Los Alamos unlimited release number is LA-UR-19-25154.

References

[1]R. Abgrall, N. Morgan K. Lipnikov, and S. Tokareva. Multidimensional staggered grid residual distribution scheme for Lagrangian hydrodynamics. *SIAM Journal of Scientific Computing*, 2019.

[2]V. Chiravalle and N. Morgan. A 3D finite element ALE method using an approximate Riemann solution. *International Journal for Numerical Methods in Fluids*, 83:642–663, 2016.

[3]V. Chiravalle and N. Morgan. A 3D Lagrangian cell-centered hydrodynamic method with higher-order reconstructions for gas and solid dynamics. *Computers & Fluids*, 83:642–663, 2018.

[4]S. Galera, P-H. Maire, and J. Breil. A two-dimensional unstructured cell-centered multi-material ALE scheme using VOF interface reconstruction. *Journal of Computational Physics*, 229:5755–5787, 2010.

[5]Z. Jia and S. Zhang. A new high-order discontinuous Galerkin spectral finite element method for Lagrangian gas dynamics in two-dimensions. *Journal of Computational Physics*, 230:2496–2522, 2011.

[6]M. Kucharik, R. Garimella, S. Schofield, and M. Shashkov. A comparative study of interface reconstruction methods for multi-material ALE simulations. *Journal of Computational Physics*, 229:2432–2452, 2010.

[7]Z. Li, X. Yu, and Z. Jia. The cell-centered discontinuous Galerkin method for Lagrangian compressible Euler equations in two-dimensions. *Computers & Fluids*, 96:152–164, 2014.

[8]E. Lieberman, X. Liu, N. Morgan, D. Luscher, and D. Burton. A higher-order Lagrangian discontinuous Galerkin hydrodynamic method for solid dynamics. *Computer Methods in Applied Mechanics and Engineering*, 353:467–490, 2019.

[9]E. Lieberman, N. Morgan, D. Luscher, and D. Burton. A higher-order Lagrangian discontinuous Galerkin hydrodynamic method for elastic-plastic flows. *Computers & Fluids*, 78:318–334, 2019.

[10]X. Liu, N. Morgan, and D. Burton. A Lagrangian cell-centered discontinuous Galerkin hydrodynamic method for 2D Cartesian and RZ axisymmetric coordinates. 2018 AIAA Aerospace Sciences Meeting, AIAA-2018-1562, Kissimmee, Florida, 2018.

[11]X. Liu, N. Morgan, and D. Burton. A Lagrangian discontinuous Galerkin hydrodynamic method. *Computers & Fluids*, 163:68–85, 2018.

[12]X. Liu, N. Morgan, and D. Burton. Lagrangian discontinuous Galerkin hydrodynamic methods in axisymmetric coordinates. *Journal of Computational Physics*, 373:253–283, 2018.

[13]X. Liu, N. Morgan, and D. Burton. A high-order Lagrangian discontinuous Galerkin hydrodynamic method for quadratic cells using a subcell mesh stabilization scheme. *Journal of Computational Physics*, 386:110–157, 2019.

[14]R. Loubère, P.-H. Maire, M. Shashkov, J. Breil, and S. Galera. ReALE: a reconnection-based arbitrary Lagrangian Eulerian method. *Journal of Computational Physics*, 229:4724–4761, 2010.

[15]A. Morgan. Detecting shock waves with artificial intelligence. NM supercomputing challenge, March 2019. Student report.

[16]N. Morgan, K. Lipnikov, D. Burton, and M. Kenamond. A Lagrangian staggered grid Godunov-like approach for hydrodynamics. *Journal of Computational Physics*, 259:568–597, 2014.

[17]N. Morgan, X. Liu, and D. Burton. A Lagrangian discontinuous Galerkin hydrodynamic method for higher-order triangular elements. 2018 AIAA Aerospace Sciences Meeting, AIAA-2018-1092, Kissimmee, Florida, 2018.

[18]N. Morgan, X. Liu, and D. Burton. Reducing spurious mesh motion in Lagrangian finite volume and discontinuous Galerkin hydrodynamic methods. *Journal of Computational Physics*, 372:35–61, 2018.

[19]W. Noh. Errors for calculations of strong shocks using an artificial viscosity and an artificial heat flux. *Journal of Applied Physics*, 72:78–120, 1987.

[20]C. Pederson, B. Brown, and N. Morgan. The Sedov blast wave as a radial piston verification test. *Journal of Verification, Validation and Uncertainty Quantification*, 1:1–9, 2016.

[21]D. Ray and J. Hesthaven. An artificial neural network as a troubled-cell indicator. *Journal of Computational Physics*, 367:166–191, 2018.

[22]D. Rumelhar, G. Hinton, and R. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

[23]L. Sedov. *Similarity and Dimensional Methods in Mechanics*. Academic Press, 1959.

[24]M. Veiga and R. Abrall. Towards a general stabilization method for conservation laws using a multilayer perceptron neural network: 1D scalar and system of equations. In ECCM-ECFD 2018 6th European Conference on Computational Mechanics (Solids, Structures and Coupled Problems) 7th European Conference on Computational Fluid Dynamics, Glasgow, United Kingdom, June 2018.

[25]F. Vilar, P-H. Maire, and R. Abgrall. A discontinuous Galerkin discretization for solving the two-dimensional gas dynamics equations written under total Lagrangian formulation on general unstructured grids. *Journal of Computational Physics*, 276:188–234, 2014.

[26]J. Waltz, N. Morgan, T Canfield, M. Charest, and J. Wohlbier. A three-dimensional finite element arbitrary Lagrangian-Eulerian method for shock hydrodynamics on unstructured grids. *Computers & Fluids*, 92:172–187, 2013.