ISBN: 978-3-95450-247-9 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2024-MOPS72

SOLVING THE ORSZAG-TANG VORTEX MAGNETOHYDRODYNAMICS PROBLEM WITH PHYSICS-CONSTRAINED CONVOLUTIONAL NEURAL NETWORKS *

Christopher Leon †, Alexander Scheinker
Applied Electrodynamics Group, Los Alamos National Laboratory, Los Alamos, NM, USA
Ari Bormanis, University of Arizona, Tucson, Arizona , USA

Abstract

The 2D Orszag-Tang vortex magnetohydrodynamics (MHD) problem is studied through the use of physics-constrained convolutional neural networks (PCNNs). The density, ρ, and the magnetic field, $\mathbf{B}$, are forecasted, and we also predict $\mathbf{B}$ given the velocity field, $\mathbf{v}$, of the fluid. We examined the incorporation of various physics constraints into the PCNNs: absence of magnetic monopoles, non-negativity of ρ and use of only relevant variables. Translation equivariance was present from the convolutional architecture. The use of a residual architecture and data augmentation was found to increase performance greatly. The most accurate models were incorporated into the simulation, with reasonably accurate results. For the prediction task, the PCNNs were evaluated against a physics-informed neural network (PINN), which had the ideal MHD induction equation as a soft constraint. The use of PCNNs for MHD has the potential to produce physically consistent real-time simulations to serve as virtual diagnostics in cases where inferences must be made with limited observables.

INTRODUCTION

Magnetohydrodymaics (MHD) is a field that explores the behavior of electrically conductive fluids, an interesting intersection of fluid dynamics and electrodynamics. Phenomena that can be described in terms of MHD includes the iron core of the Earth and the corona of the Sun. The beam dynamics inside an accelerator shares many characteristics similar to MHD, thus can be of interest to the accelerator community.

Simulations of MHD can be computationally demanding. Although much work has been done in using surrogate machine learning (ML) models to speed up fluid and electrodynamics computations, relatively little has been done in applying these techniques to MHD. Here, we use techniques developed in a previous work to create physics-constrained convolutional neural networks (PCNNs) which is a novel recent method for incorporating hard physics constraints within neural networks for electrodynamics [1]. Unlike physics informed neural networks (PINNs), which do not guarantee any hard constraints, PCNNs directly build hard constraints into the ML architecture by generating vector and scalar potentials from which the electromagnetic fields are generated according to Maxwell's equations. In the PCNN approach for 3D electromagnetism rather than modeling the magnetic field with the NN and trying to impose Gauss's law for magnetism the magnetic vector potential is generated, $\mathbf{A}(\mathbf{r}, t)$. Then the physical constraint is automatically satisfied (i.e, $\nabla \cdot (\nabla \times \mathbf{A}) = 0$), up to numerical errors [1]. In this paper we implement the 2D version of that, with:

$$\mathbf{B}(\mathbf{r}, t) = \nabla_{\perp} A(\mathbf{r}, t), \tag{1}$$

where $A(\mathbf{r}, t)$ is a scalar field.

For ideal MHD, where the fluid has perfect conductivity and no viscosity, the equations of motions are:

$$\frac{D\rho}{Dt} = -\rho\nabla \cdot \mathbf{v}, \qquad \rho\frac{D\mathbf{v}}{Dt} = \nabla \cdot \left(-p\mathrm{I} + \mathbf{B}\mathbf{B}^T\right), \tag{2}$$

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \cdot \left(\mathbf{B}\mathbf{v}^T - \mathbf{v}\mathbf{B}^T\right), \qquad \nabla \cdot \mathbf{B} = 0, \tag{3}$$

where $\frac{D}{Dt} = \frac{\partial}{\partial t} + \mathbf{v} \cdot \nabla$. These correspond to conservation of mass, the force equation, the induction equation and Gauss's law for magnetism.

For computational MHD, the Orszag-Tang vortex [2] is a 2D problem commonly used to benchmark simulations[3–5]. It corresponds to the initial conditions of constant p and ρ, with initial velocity and magnetic fields of:

$$\mathbf{v}(x, y, t = 0) = \left(-\sin(2\pi y), \sin(2\pi x)\right), \tag{4}$$

$$\mathbf{B}(x, y, t = 0) = \nabla_{\perp} \frac{1}{2\pi\sqrt{4\pi}} \left(\frac{-\sin(4\pi x)}{2} + \cos(2\pi y)\right), \tag{5}$$

where $\nabla_{\perp} = \left(\frac{\partial}{\partial y}, -\frac{\partial}{\partial x}\right)$. These initial conditions end up producing a vortex.

To test our ML models, we set forth the following three tasks:

1. ρ forecasting - given a density field, forecast it for the next time step:

$$\rho(t) \to \rho(t + \Delta t)$$

2. $\mathbf{B}$ forecasting - given a magnetic field, forecast it for the next time step:

$$\mathbf{B}(t) \to \mathbf{B}(t + \Delta t)$$

3. $\mathbf{B}$ predictions - given the velocity field, predict the magnetic field at the same time step:

$$\mathbf{v}(t) \to \mathbf{B}(t)$$

* This work was supported by the Los Alamos National Laboratory Laboratory Directed Research and Development (LDRD) Program Directed Research (DR) project number 20220074DR.
† cleon@lanl.gov

ISBN: 978-3-95450-247-9 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2024-MOPS72

Table 1: Physics Constraints with Neural Networks

Constraint	Implementation	Hard/Soft
Divergence-free	$\mathbf{B}(\mathbf{r},t) = \nabla_\perp A(\mathbf{r},t)$	Hard
Translation Equivariance	CNN Architecture	Hard
Non-negativity (e.g., $\rho \geq 0$)	Final Layer ReLU	Hard
Periodic BC's	Padding	Soft
Partial Differential Equation	Term in Cost Function	Soft

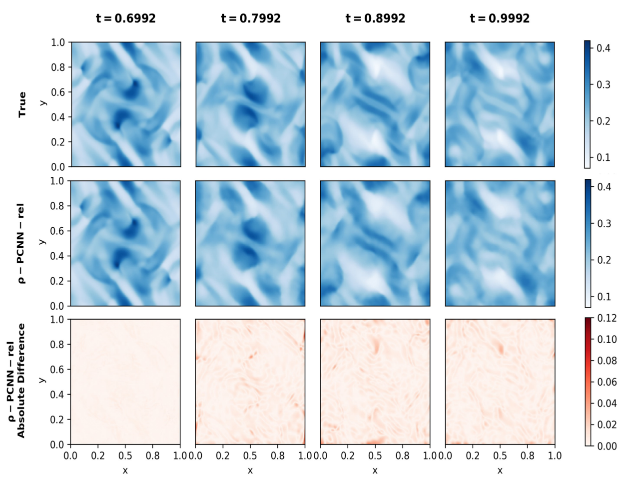

Figure 1: Predictions of ρ using a PCNN with only relevant variables as input, ρ-PCNN-rel. Figure reproduced from [7].

For each task, different models were tested and their performances compared. The models differed by architecture, the training procedure and the inclusion/absence of physics constraints (see Table 1). Note: a soft constraint is one where the model is encouraged to follow a certain constraint. A hard constraint is one that the network *must* follow. For tasks 1. and 2. the models were fed their own predictions for testing and compared to the actual evolution of ρ and $\mathbf{B}$.

DATA

The training and test data came from a conventional simulation based on constraint transport on a static mesh [4, 6]. We took the Orszag-Tang vortex from a time of $t = 0$ to $t = 1$, which consisted of 1,250 time steps. The training set went from $t = 0$ to $t = 0.7488$, while the test set was the rest of the simulation. This corresponded to a 75/25 train/test split. A spatial resolution of 128×128 with boundary conditions was used.

RESULTS

Unless stated otherwise, the NN models have an encoder-decoder convolutional neural networks architecture. All models were trained using the Adam optimizer, and Tensorflow was used for the creation and testing of the ML models.

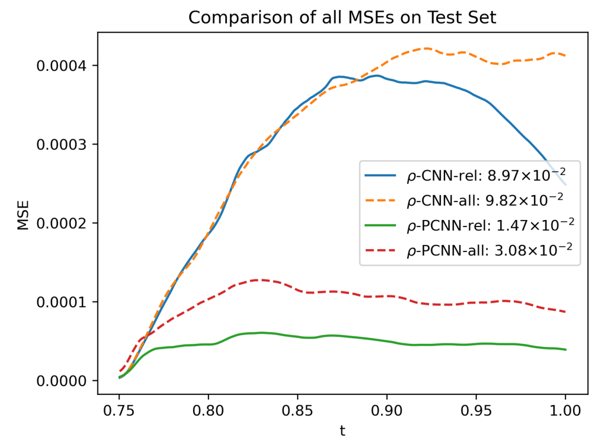

Figure 2: Predictions of ρ using a physics-constraint model with only relevant variables. Figure reproduced from [7].

ρ Forecasting

To enforce the non-negativity condition, $\rho \geq 0$, we included a $ReLU$ activation function in the final layer, where $ReLU(x) = \max(0, x)$. Models with (without) this feature were labelled PCNN (CNN).

Using Eq. 2, we know that $\rho(t + \Delta t)$ should depend on $\rho(t)$ and $\mathbf{v}(t)$, the only relevant variables. Models that used for input relevant (all) variables were labelled -rel(-all)

The model that performed best was ρ-PCNN-rel, which included the non-negativity constraint and only used relevant variables. How its predictions compared to to actual test data can be seen in Fig. 1 and the performance of the four models evaluated by MSE can be seen in Fig. 2.

***B** Forecasting*

To enforce the $\nabla \cdot \mathbf{B} = 0$ constraint, we had the network predict the a potential and then used the 2D version of $\mathbf{B} = \nabla \times \mathbf{A}$, which is: :

$$\mathbf{B} = \nabla_\perp A \quad (6)$$

where A is a scalar field, and $\nabla \cdot (\nabla_\perp A) = 0$ automatically follows. Thus, this is a hard constraint.

For a single time step, we expect the magnetic field to change by an amount, $\Delta\mathbf{B}(t, \Delta t)$:

$$\mathbf{B}(t + \Delta t) = \mathbf{B}(t) + \Delta\mathbf{B}(t, \Delta t). \quad (7)$$

Given a small time step Δt, we expect that in general $|\Delta\mathbf{B}(t, \Delta t)| \ll |\mathbf{B}(t)|$. Thus, having the network predict $\Delta\mathbf{B}(t, \Delta t)$ is easier than predicting $\mathbf{B}(t + \Delta t)$. Alternatively, you can view this as initiating the prediction of $\mathbf{B}(t+\Delta t)$ with $\mathbf{B}(t)$, which is a good starting point since the field changes only minorly in between successive time steps. Given this, we turned to a Residual Network (ResNet) architecture, illustrated in Fig. 3. We found this greatly improved results.

ISBN: 978-3-95450-247-9 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2024-MOPS72

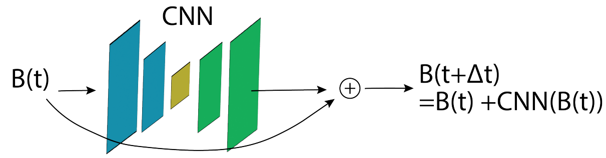

Figure 3: Using a ResNet architecture to predict the next time step for **B**.

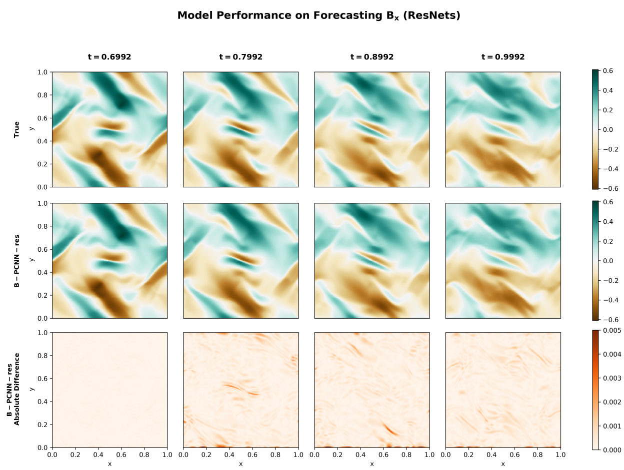

Figure 4: Predictions of B_x using a physics-constraint ResNet model. Figure reproduced from [7].

The predictions using both physics constraints and a ResNet architecture can be seen in Fig. 4

To further improve predictions, we used data augmentation. Specifically, some NN's were trained on 50 MHD simulations beforehand. These simulations differed from the Orszag-Tang Vortex by having random initial conditions for the p and ρ fields. These were produced first by taking random Gaussian noise and then smoothing it with a convolution with of a Gaussian. The inclusion of this data, in addition to the Ozsang-Tang training data, greatly improved results.

In addition to accurately reproducing the data, we tested to see how accurate the divergeneless condition held using the metric of relative divergence:

$$\epsilon_{rd} = \frac{|\nabla \cdot \mathbf{B}|\Delta x}{\sqrt{2p}} \tag{8}$$

where Δx is the spatial spacing in the simulation. It was found that generall PCNN's had lower average ϵ_{rd}

We also tested replacing the magnetic field prediction part of our constraint transport simulation with a neural network. The result closely followed the full simulation.

B Predictions

Here, we tested CNN models against a physics-informed neural network (PINN) [8]. This was a dense neural network with input of spaces and time: x, y, t.

$$x, y, t \rightarrow \mathbf{B}(x, y, t) \tag{9}$$

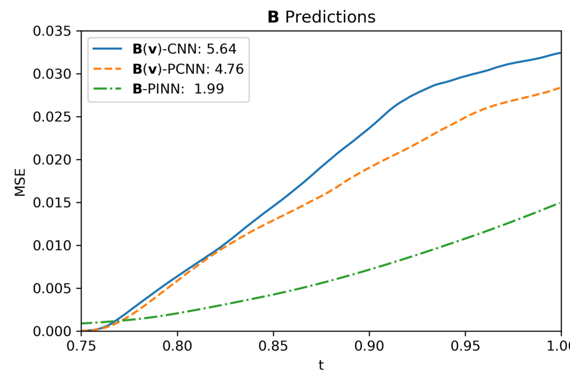

Figure 5: MSE error of **B** prediction. The total MSE error summed up is next to the model name in the legend. Figure reproduced from [7].

The advantage of this architecture is that through automatic differentiation, fast and accurate derivatives can be taken with respect to space and time. This can be used to include a term in the loss function that encourages the network to satisfy a partial differential equation, for here specifically the induction equation, Eq. (3):

$$L_{PDE} = \sum_i \left| \frac{\partial \mathbf{B}_i}{\partial t} - \nabla \cdot \left(\mathbf{B}_i \mathbf{v}_i^T - \mathbf{v}_i \mathbf{B}_i^T \right) \right|^2 \tag{10}$$

where the sum is over all points in the training data. Unlike CNN models, though, these type of models lack the hard constraint of spatial translation equivariance.

Figure 5 shows the MSE of a CNN model with physics constraints, a physics model without and the PINN.

CONCLUSION

The models that performed best generally contained physics constraints. ML techniques like using a ResNet architecture and data augmentation were also useful.

This work can be useful in speeding up computational MHD, either as a supplement to computationally expensive conventional simulations or through the hybridization of conventional simulation and an ML surrogate model. Also, it can be applied to inverse problems, which are often ill-posed and traditional methods struggle.

One of the benefits of using ML to speed up otherwise computationally expensive slow simulations is to provide real-time virtual diagnostics, as in [9], which can be used for real-time adaptive beam control. For example, in [10] the first approach to adaptive ML was demonstrated combining a deep learning with adaptive feedback [11] for automatic control of the longitudinal phase space of the LCLS FEL electron beam.

REFERENCES

[1] A. Scheinker and R. Pokharel, "Physics-constrained 3d convolutional neural networks for electrodynamics," *APL Machine Learning*, vol. 1, no. 2, 2023.

ISBN: 978-3-95450-247-9 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2024-MOPS72

[2] S. A. Orszag and C.-M. Tang, "Small-scale structure of two-dimensional magnetohydrodynamic turbulence," *Journal of Fluid Mechanics*, vol. 90, no. 1, pp. 129–143, 1979.

[3] J. M. Picone and R. B. Dahlburg, "Evolution of the orszag–tang vortex system in a compressible medium. ii. supersonic flow," *Physics of Fluids B: Plasma Physics*, vol. 3, no. 1, pp. 29–44, 1991.

[4] P. Mocz, M. Vogelsberger, and L. Hernquist, "A constrained transport scheme for mhd on unstructured static and moving meshes," *Monthly Notices of the Royal Astronomical Society*, vol. 442, no. 1, pp. 43–55, 2014.

[5] B. Snow, A. Hillier, G. Murtas, G. J. Botha, and S. Camera, "Shock identification and classification in 2d magnetohydrodynamiccompressible turbulence—orszag–tang vortex," *Experimental Results*, vol. 2, e35, 2021.

[6] C. R. Evans and J. F. Hawley, "Simulation of magnetohydrodynamic flows-a constrained transport method," *The Astrophysical Journal*, vol. 332, pp. 659–677, 1988.

[7] A. Bormanis, C. A. Leon, and A. Scheinker, "Solving the orszag–tang vortex magnetohydrodynamics problem with physics-constrained convolutional neural networks," *Physics of Plasmas*, vol. 31, no. 1, 2024.

[8] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational physics*, vol. 378, pp. 686–707, 2019.

[9] A. Scheinker, F. Cropp, and D. Filippetto, "Adaptive autoencoder latent space tuning for more robust machine learning beyond the training set for six-dimensional phase space diagnostics of a time-varying ultrafast electron-diffraction compact accelerator," *Physical Review E*, vol. 107, no. 4, p. 045 302, 2023.

[10] A. Scheinker, A. Edelen, D. Bohler, C. Emma, and A. Lutman, "Demonstration of model-independent control of the longitudinal phase space of electron beams in the linac-coherent light source with femtosecond resolution," *Physical review letters*, vol. 121, no. 4, p. 044 801, 2018.

[11] A. Scheinker and M. Krstić, "Minimum-seeking for clfs: Universal semiglobally stabilizing feedback under unknown control directions," *IEEE Transactions on Automatic Control*, vol. 58, no. 5, pp. 1107–1122, 2012.