Multigrid solver with super-resolved interpolation

Anonymous authors
Paper under double-blind review

Abstract

The multigrid algorithm is an efficient numerical method for solving a variety of elliptic partial differential equations (PDEs). The method damps errors at progressively finer grid scales, resulting in faster convergence compared to standard iterative methods such as Gauss-Seidel. The prolongation, or coarse-to-fine interpolation operator within the multigrid algorithm lends itself to a data-driven treatment with ML super resolution, commonly used to increase the resolution of images. We (i) propose the novel integration of a super resolution generative adversarial network (GAN) model with the multigrid algorithm as the prolongation operator and (ii) show that the GAN-interpolation improves the convergence properties of the multigrid in comparison to cubic spline interpolation on a class of multiscale PDEs typically solved in physics and engineering simulations.

1 Introduction

Partial differential equations describe the behavior of many physical phenomena, from the smallest quantum scales to the largest scales in the universe. Efficient numerical methods are crucial to simulate these complex phenomena with current and foreseeable computational resources. The application of machine learning to accelerate and improve the accuracy of physical simulations is an increasingly more active area of research, leveraging the surge of advancements in machine learning architectures over the last decade (e.g. Wei et al., 2018; Peurifoy et al., 2018; Ranade et al., 2020; Sanchez-Gonzalez et al., 2020). While replacing conventional numerical solvers with machine learning models is promising, improving heuristic operators within existing mathematically-formal numerical methods can allow for interpretable computational gains, easier implementation and more rapid deployment.

2 Proposal

2.1 Multigrid method

Multigrid methods are particularly effective for solving multiscale differential equations relevant to physics and engineering (e.g. Trottenberg et al., 2001; Sidharth, 2020). Multigrid methods formally scale linearly with the number of unknowns in a discretized physical system, which make them attractive as high performance computing increasingly enables higher-fidelity and larger scale computations. Generally, the favorable scaling of multigrid methods is due to their ability to iteratively reduce solution errors at multiple scales by obtaining coarse solutions then utilizing the coarsened solutions to solve increasingly higher-fidelity solutions. The utilization of coarse fidelity solutions to accelerate finer fidelity solutions is enabled by a prolongation, or interpolation, operator.

Prolongation within a multigrid algorithm lends itself to a data-driven treatment due to the ability for machine learning techniques to perform data-informed interpolation, as seen frequently seen in super-resolution methods as used in image analysis (c.f. Farsiu et al., 2004; Park et al., 2003). In this work, we consider the applications of enhancing multigrid methods by using a data-driven prolongation operator akin to super-resolution operators used in image analysis. In particular we consider a common Poisson differential equation, frequently solved in engineering and physics calculations, written as

$$\nabla^2 p = \frac{\partial^2 p}{\partial^2 x} + \frac{\partial^2 p}{\partial^2 y} = f(x, y), \tag{1}$$

where the solution p is discretized on a grid. Figure 1a shows a schematic of the two-level multigrid, where the top horizontal level represents the highest resolution grid h_2. Applying the Poisson opera-
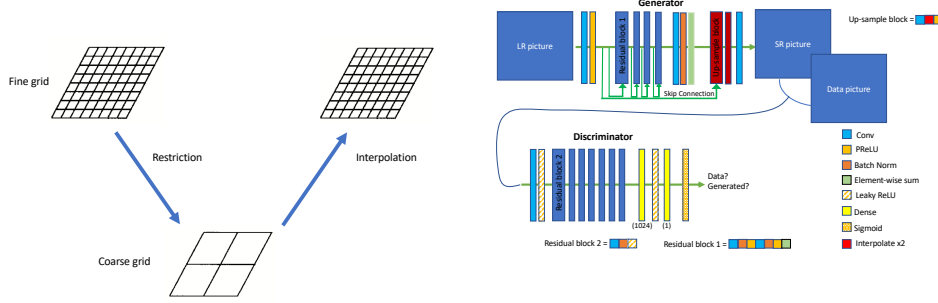
(a) Schematic of a two-level multigrid cycle. Adapted from Chen & Sun (2001).

(b) Structure of discriminator and generator we implement, modified from Ledig et al. (2017).

Figure 1: Diagrams for two-level multigrid and GAN interpolation.

tor $\nabla^2 p - f$ to the grid h_2, using a relaxation method such as Gauss-Seidel, results in an residual r_f relative to the true solution. For these relaxation methods, high frequency errors (relative to the grid scale) are quickly reduced, while low spatial frequency errors remain. It can be shown (Trottenberg et al., 2001) that the residual also satisfies the Poisson equation, so we recursively solve the Poisson equation for this error. In order to efficiently suppress low frequency errors, we apply a restriction operator R to coarsen the grid, so that previously low spatial frequency errors on the fine grid become high frequency errors on the coarse grid. Once the coarsest level is reached, we apply a prolongation operator P to the solution in order to correct the solution on the fine grid. In a uniformly spaced grid, the restriction operator R, can be a simple as taking only the black points in red-black ordering. The prolongation operator requires more work, as it involves an interpolation scheme to create a higher resolution grid. For example, this interpolation scheme can be a two-dimensional mesh interpolation, such as the bivariate spline function in the SciPy Python library (Virtanen et al., 2020).

2.2 GAN AND INTEGRATION WITHIN MULTIGRID

Deep learning (LeCun & Hinton, 2015) methods have been able to capture and reproduce the structure of extremely complex physical systems (e.g. Portwood et al., 2021; Wang et al., 2020; He et al., 2019). Even without any prior knowledge of the system, the network can learn the behavior and patterns with adequate data, relatively quickly. Super resolution refers to the application of these networks to low resolution images in order to predict a high resolution version. A GAN (Goodfellow et al., 2014) is particularly effective as an accurate super resolution model. Here, we use the GAN architecture from Ledig et al. (2017), following the example in Birla, as shown in Figure 1b. Unlike Ledig et al. (2017) who use the pre-trained VGG loss designed for photographic image recognition, we use mean squared error (MSE) loss function for the generator.

The GAN takes $n_s \times n_s$ (here $n_s = 6$) point input tiles from the coarse solution for spatially-invariant training that generates more training samples from limited datasets. Using the windowed inputs, the GAN can operate on grids with different resolution without any change in input formats. Ghost cells are implemented by splitting the domain up into overlapping windows and utilize only a part of the any particular GAN's predictions for the high resolution grid.

The GAN prolongation operator works as follows:

- Find the max and min data of the grid $p_{\max}, p_{\min}$, and normalize the grid data using a symmetric log function from $[-\infty, \infty]$ to $[-1, 1]$, such that for positive data, $\log(p_{\max,\min}) = (1, 0)$ and for negative data, $\log p_{\max,\min} = (-1, 0)$.

- Divide the normalized coarse grid into overlapping $n_s \times n_s$ windows (i.e. assuming a n_s^2 kernel with stride = 2). Apply GAN interpolation to produce a set of high resolution n_l^2 window kernels. Here $n_l = 24$.
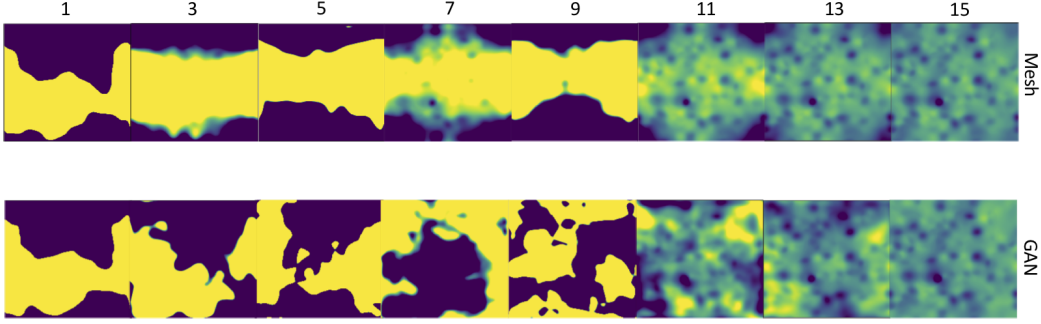
Figure 2: Comparison of the first 15 iterations of solving Eq. 1 with the multigrid solver starting from an initially random grid. Top: interpolation with traditional mesh interpolation. Bottom: interpolation with GAN super resolution.

- Construct the fine grid by assigning the central $(n_s + 2)^2$ fine window values to the fine grid. Any non-overlapping areas (i.e the edges) are assigned to the window covering that area.
- Transform and scale back to the original pressure data.

2.3 DATA AND TRAINING OF THE GAN

The pressure-Poisson formulation of the incompressible Navier-Stokes equations is a physical system with the form of Eq. 1. The pressure-Poisson equation is used to ensure mass conservation in incompressible flows and is derived from taking the divergence of the fluid momentum equation

$$\nabla^2 p = \nabla \cdot (\nu \nabla^2 u - (u \cdot \nabla) u) = f(x, y) \tag{2}$$

Here ν is the fluid viscosity, u is the fluid velocity, and p is the fluid pressure. Evolving the system in time involves calculating the source term $f(x, y)$ from the velocity field u, solving the Poisson equation for pressure (e.g. using a multigrid method), and then using that pressure to update the velocity field. We produce a training set of 200 pressure fields on a grid size 192^2 by evolving a biperiodic velocity field from a broadband initial condition using the incompressible Navier-Stokes.

We construct the training set of 1000 pressure grids as follows:

- Select one of the 200 pressure fields randomly and restrict (downsample) the grid by a random power l of 2 on each grid length, such that the smallest grid size is $(n_s = 6)^2$. The result is a random grid of size $(192/2^l)^2$.
- From the grid above, pick a random 6^2 window. The total size of the space of such windows is $17,000$.
- Transform and rescale the window data range from $[-\infty, \infty]$ to $[-1, 1]$ in the same manner as described in the GAN prolongation method in 2.2, using the global min and max of all the pressure grids. We used $|p_{\min}| = 10^{-10}$ and $|p_{\max}| = 10^{-3}$.

3 PROPOSAL IMPLEMENTATION

We solve Eq. 2 using a test set source grid. Our implementation of the multigrid algorithm is characterized by four parameter choices: $N_{\mathrm{smooth,pre}}$ = number of smoothing operations on the initial grid; N_{smooth} = number of smoothing operations at the beginning of each iteration; N_{step} = number of times the the grid resolution is reduced by a factor of two on each dimension; $r_{\min}$ = coarsest resolution. Pre-smoothing is often necessary in order to ensure convergence. To study the efficacy of a GAN prolongation operator under controlled settings, we implement a two-level V-cycle multigrid algorithm (fine scale 192^2 grid to a coarse scale 12^2 grid, corresponding to N_{step} = 5 and $r_{\min} = 12$). A two-level multigrid method with a coarsening factor of 16 is expected to perform poorly, limited by prolongation and interpolation, and therefore serves as an ideal test bed
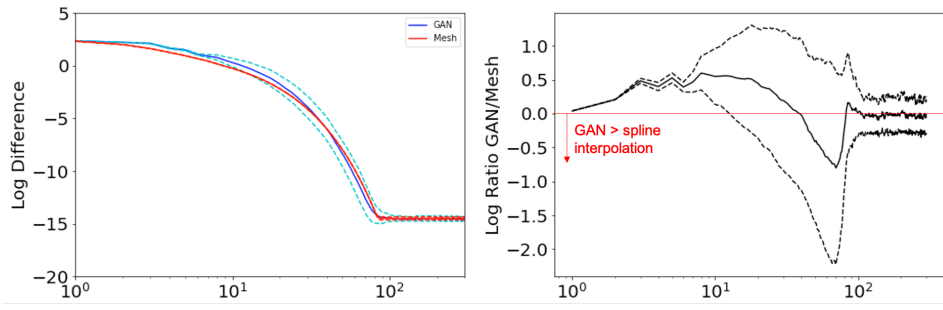
Figure 3: Averaged results of multigrid algorithm solving 100 different grids. Norm of the difference of grid between iterations as a function of iteration, shown for two choices of prolongation operator, two-dimensional spline and GAN super resolution interpolation. The multigrid parameters used are $N_{\text{smooth,pre}} = 10$, $N_{\text{smooth}} = 10$, $N_{\text{step}} = 4$, and $r_{\min} = 12$.
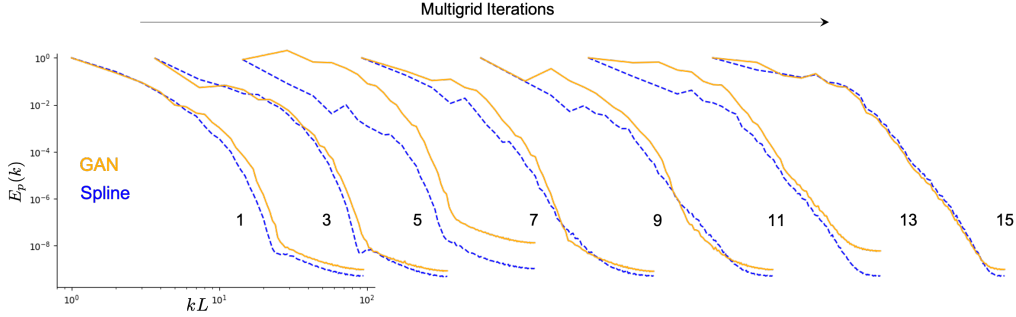

Figure 4: Convergence in p spectral density with increasing multigrid iterations.

for evaluating the gains made with a super-resolution GAN. We sample 100 pressure and source term grid pairs from our training set, following the method described in Section 2.3.

Figure 3 shows a proof-of-concept test of our multigrid solver on 100 grids. We plot the mean profiles and $1 - \sigma$ bounds of the norm difference between the pressure grids at the current and previous iteration. The multigrid solver converges in 100 iterations (compared to 10^4 iterations for a simple iterative solver). Statistically, we find that using the super resolution prolongation operator results in faster convergence to the 10^{-6} numerical error range, compared to bivariate spline operator; the large $1 - \sigma$ lines shows that the performance is distributed. For some fields, super resolution exhibits faster convergence, but for other field types, the spline interpolation can fare better. These tests demonstrate strongly that there is potential for super-resolution to improve the efficiency of the multigrid algorithm by data-driven prolongation operators. Figure 2 shows a sequence of the first 15 iterations of a particular field. While both solvers converge to machine zero, with the conventional spline-based interpolation, the early iterations are unable to capture mid to high-frequency information. With the GAN-based interpolation however, the solver feeds in higher frequency information in the early iterations. This is confirmed via the pressure field spectra shown in Figure 4.

4   CONCLUSION

In this research, we consider novel integration of data-driven interpolation into the robust multigrid differential equation solution technique as applied to a common differential system utilized in many physics and engineering domains. We have demonstrated the ability for super-resolution techniques, GANs in particular, to serve as a prolongation operator which can improve solution time when compared to high order spline interpolation techniques. As we continue our investigations for this approach, we propose that the integration of data-driven methods into formally-derived numerical solution procedures is a promising approach for computational acceleration of simulations.

REFERENCES

D. Birla. URL `https://github.com/deepak112/Keras-SRGAN`.

Shu-hua Chen and Wen-yih Sun. Application of the multigrid method and a flexible hybrid coordinate in a nonhydrostatic model. *Monthly weather review*, 129(11):2660–2676, 2001.

Sina Farsiu, Dirk Robinson, Michael Elad, and Peyman Milanfar. Advances and challenges in super-resolution. *International Journal of Imaging Systems and Technology*, 14(2):47–57, 2004.

I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Ward-Farley, S. Ozair, A. Courville, and Y. Bengio. *Adv. Neural Procs. Sys.*, 27:2672 – 2680, 2014.

Siyu He, Yin Li, Yu Feng, Shirley Ho, Siamak Ravanbakhsh, Wei Chen, and Barnabás Póczos. Learning to predict the cosmological structure formation. *Proceedings of the National Academy of Sciences*, 116(28):13825–13832, 2019.

Y. LeCun and G. Hinton. Deep learning. *Nature*, 521:436 – 444, 2015.

Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. pp. 4681–4690, 2017.

Sung Cheol Park, Min Kyu Park, and Moon Gi Kang. Super-resolution image reconstruction: a technical overview. *IEEE signal processing magazine*, 20(3):21–36, 2003.

John Peurifoy, Yichen Shen, Li Jing, Yi Yang, Fidel Cano-Renteria, Brendan G DeLacy, John D Joannopoulos, Max Tegmark, and Marin Soljačić. Nanophotonic particle simulation and inverse design using artificial neural networks. *Science advances*, 4(6):eaar4206, 2018.

Gavin D Portwood, Balasubramanya T Nadiga, Juan A Saenz, and Daniel Livescu. Interpreting neural network models of residual scalar flux. *Journal of Fluid Mechanics*, 907:A23, 2021.

Rishikesh Ranade, Chris Hill, and Jay Pathak. Discretizationnet: A machine-learning based solver for navier-stokes equations using finite volume discretization. *arXiv preprint arXiv:2005.08357*, 2020.

Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *International Conference on Machine Learning*, pp. 8459–8468. PMLR, 2020.

G S Sidharth. A multiscale subgrid decomposition. *AIAA Scitech 2020 Forum*, pp. 0820, 2020.

U. Trottenberg, C. W. Oosterlee, and A. Schuller. *Multigrid*. 2001.

Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272, 2020.

Rui Wang, Karthik Kashinath, Mustafa Mustafa, Adrian Albert, and Rose Yu. Towards physics-informed deep learning for turbulent flow prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1457–1466, 2020.

Qianshi Wei, Ying Jiang, and Jeff ZY Chen. Machine-learning solver for modified diffusion equations. *Physical Review E*, 98(5):053304, 2018.