Diffusion model based synthetic data generation for partial differential equations

Rucha Apte [1] Sheel Nidhan [1] Rishikesh Ranade [1] Jay Pathak [1]

Abstract

In a preliminary attempt to address the problem of data scarcity in physics-based machine learning, we introduce a novel methodology for data generation in physics-based simulations. Our motivation is to overcome the limitations posed by the limited availability of numerical data. To achieve this, we leverage a diffusion model that allows us to generate synthetic data samples and test them for two canonical cases: (a) the steady 2-D Poisson equation, and (b) the forced unsteady 2-D Navier-Stokes (NS) vorticity-transport equation in a confined box. By comparing the generated data samples against outputs from classical solvers, we assess their accuracy and examine their adherence to the underlying physics laws. In this way, we emphasize the importance of not only satisfying visual and statistical comparisons with solver data but also ensuring the generated data's conformity to physics laws, thus enabling their effective utilization in downstream tasks.

1. Introduction

The application of machine learning (ML) and deep learning (DL) techniques in modeling partial differential equations (PDEs) has gained significant momentum over the past decade. These techniques have been employed to address various challenges in physics-based modeling, such as developing closure terms for large-eddy simulations and Reynolds-averaged Navier-Stokes equations in computational fluid dynamics (CFD) (Duraisamy, 2021; Maulik et al., 2019; Ling et al., 2016), enhancing computational efficiency for classical solvers (Bar-Sinai et al., 2019; Weymouth, 2022), and facilitating reduced-order modeling (Murata et al., 2020; Eivazi et al., 2022), among others. Additionally, the field of physics-based DL research has witnessed the emergence of numerous methodologies tailored for fast inference and generalizability across different classes of PDEs. Examples include physics-informed neural networks (PINNs) (Raissi et al., 2019), Fourier neural operators (FNOs) (Li et al., 2020), DeepONet (Lu et al., 2019), latent-space based local learning (Ranade et al., 2020; 2022) and more. Although these methods have demonstrated remarkable success in solving PDEs for a variety of applications, one of the main factors impacting their accuracy and generalizability is the scarcity of high-quality training data (Sun et al., 2017).

In recent years, denoising diffusion models have emerged as the leading technique for generative modeling (Sohl-Dickstein et al., 2015; Song & Ermon, 2020; Ho et al., 2020). These models follow a two-step process: a forward step, where noise is added in a Markovian manner, followed by a reverse denoising step learnt using a deep neural network. Once trained, the model can generate new samples by starting from various realizations drawn from a Gaussian noise distribution. Diffusion models have demonstrated tremendous success in various domains, including conditional and unconditional image generation (Dhariwal & Nichol, 2021; Rombach et al., 2022), speech generation (Chen et al., 2020), image super-resolution (Saharia et al., 2022), video generation (Ho et al., 2022), to name a few. However, in the physics domain, the utilization of diffusion models has been relatively limited. Shu et al. (2023) proposed a physics-inspired diffusion model for generating high-fidelity CFD data from low-fidelity/undersampled snapshots. Vlassis & Sun (2023) used denoising diffusion models for conditional gneration of micsrostructures, testing it on the mechanical MNIST dataset. Yang & Sommer (2023) used a diffusion-based model for temporal prediction of a chaotic fluid flow. Without encoding prior physics constraint, they found that the diffusion-based network had comparable performance to that of existing models. Lim et al. (2023) extended the use of diffusion models to map functional spaces. Trained on a single resolution, the authors demonstrated the generation of PDE solutions for a variety of resolutions.

In this study, we use denoising diffusion implicit models (DDIMs) for the unconditional generation of data for two distinct physical systems: 2-D Poisson system and 2-D Navier-Stokes flow. While our trained model lacks any prior physics encoding, we utilize physics-based constraints to

[1] Ansys, Inc., Canonsburg, PA , USA. Correspondence to: Rucha Apte <rucha.apte@ansys.com>.

Accepted after peer-review at the 1st workshop on Synergy of Scientific and Machine Learning Modeling, SynS & ML ICML, Honolulu, Hawaii, USA. July, 2023. Copyright 2023 by the author(s).

select snapshots that adhere to fundamental laws. We propose two approaches to apply the physics-based constraint: PDE residual calculation (for the 2-D Poisson equation) and comparison with solver output (for the 2-D NS flow). Our aim is that, in the future, the data generation paradigm based on diffusion models can partially alleviate the challenge of data scarcity for physics-based machine learning models.

2. Methodology

2.1. Diffusion Model and Architecture

We utilize a diffusion model with a cosine scheduler that progressively degrades the data over 1000 steps (Sehwag, 2022). The reverse diffusion process is parametrized by a deep neural network based on the widely used U-Net architecture in the diffusion literature. To facilitate efficient reverse sampling from Gaussian noise, we employ the DDIM strategy described in Song et al. (2020) (Song et al., 2020) and use 500 sampling steps (instead of 1000) to accelerate the sampling speed. The input configuration of the U-Net architecture depends on the specific data to be modeled. For example, in case of the 2-D Poisson equation ($\nabla^2 u = f$), both variables u and f are provided as input to the U-Net through two separate channels. For problems involving temporal variation, different timesteps are provided as inputs to the U-Net architecture through separate channels. At the time of sampling, all the channels are initialized with Gaussian noise and DDIM is used to arrive at $T = 0$ from $T = 1000$, thus obtaining denoised channels. The utilization of the L_2 loss function resulted in greater model sample diversity, while the L_1 yielded less diverse outputs.

2.2. Physics-Based Constraints for Selection of Generated Data

For physics-based ML data, it is crucial to ensure that the generated data samples, whether obtained from a traditional solver or a machine learning approach, adhere to the underlying governing equations in addition to being visually and statistically accurate. To verify this, we propose two distinct approaches. In the first approach, we compute the MSE of the PDE residual over the grid, for example $\mathrm{MSE}(|\nabla^2 u - f|)$ in the case of a 2-D Poisson's equation. We selectively retain only those samples where this MSE is less than a particular threshold. This criterion ensures that the the generated solutions satisfy the governing equation. In an alternate approach, one can use a traditional solver to verify the quality of the generated data. In the case of steady state PDEs, we compare the MSE between the generated solution (u in Poisson's equation) with the u evaluated from a classical solver for the same generated f. Alternatively, for transient PDEs, the first channel of the generated data is provided as an input to a traditional solver and the MSE (averaged across the entire grid and remaining channels) of the diffusion-generated data and solver-generated data is evaluated. We retain only those sets of generated snapshots that exhibit an MSE lower than a certain threshold. This approach guarantees that the selected samples align with the underlying physics.

3. Experiments

We demonstrate our diffusion model based data generation technique for two distinct use cases outlined below.

3.1. 2-D Poisson Equation

In the case of the 2-D Poisson equation, $\nabla^2 u = f$, both u and f are passed as two separared channels to the U-Net architecture. The weights of the U-Net are optimized based on the loss function $L_{\text{poisson}} = \lambda \|\epsilon_u - \epsilon_u^{\text{pred}}\|^2 + \|\epsilon_f - \epsilon_f^{\text{pred}}\|^2$, where ϵ corresponds to the noise added in the forward diffusion. We empirically found that $\lambda = 2$ worked the best for 2-D Poisson equation. The network is trained on 10,000 pairs of $[f, u]$ generated on a 64×64 grid using a multigrid solver.

In this study, we aimed to address the challenge of generating f and u simultaneously using a diffusion model. Traditionally, data generation tasks often focus on generating one variable at a time, such as generating f or u independently. However, in our context, it was crucial to generate f and u together due to their inherent dependencies and interactions. This posed a more complex and challenging problem, as the diffusion model has to capture the joint distribution of f and u accurately.

3.2. 2-D Forced Navier Stokes Vorticity-Transport Equation

For the forced unsteady 2-D Navier-Stokes (NS) equation, we train the diffusion model on blocks of five consecutive vorticity (ω) fields, where separation between two consective fields in time $\Delta t = 1.6$s. The blocks size (five in this case) can be an arbitrary choice. These five consecutive vorticity fields are sent as five channels input to the U-Net network, and the loss in the noise prediction is calculated by summing over all the five channels, $L_{\text{NS}} = \sum_{c=1}^{5} \|\epsilon_c - \epsilon_c^{\text{pred}}\|^2$, where c is channel index. The network for 2-D NS equations is trained with 700 solutions, each starting with a different initial condition. The vorticity is evolved until $t = 320$s on 64×64 grid using a NS solver (Li et al., 2020). The viscosity is set at $\nu = 10^{-4}$ and the forcing function takes the form $f = 0.1\sin(4\pi(x+y)) + 0.1\cos(4\pi(x+y))$. To increase the amount of data for training, a sliding window approach with a stride of three was used. Hence, there is an overlap between two consecutive blocks of five snapshots.

The U-Net architecture used for 2-D Poisson and NS equation data consists of $6M$ parameters with adaptive group normalization. The codebase for this work is built on (Sehwag, 2022). The U-Net network is conditioned with the diffusion time t through feature vector embedding. Both networks are trained for 250 epochs.

4. Results

4.1. 2-D Poisson Equation

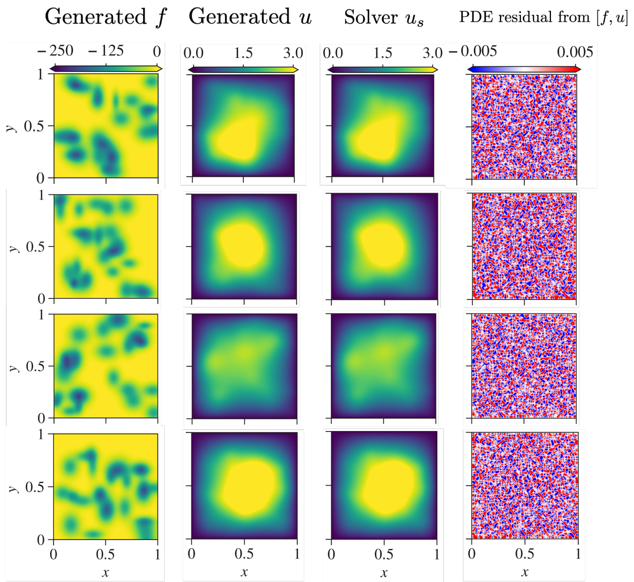

Figure 1. Four $[f_{\text{generated}}, u_{\text{generated}}]$ pairs generated from diffusion model (first two columns from left), u_{solver} generated from the corresponding f (third column), PDE residual calculated using generated $[f, u]$ pairs (fourth column). MSE of PDE residuals for all these four generated samples $< 6 \times 10^{-6}$.

In the context of 2-D Poisson, we conducted a comprehensive analysis of the generated $[f, u]$ pairs, focusing on their visual quality and adherence to the underlying governing equations. Figure 1 showcases four such pairs (left two columns). The contours of the synthetically generated $[f, u]$ exhibit smooth boundaries, and their values fall within the expected range. However, it is important to note that visual appearance alone does not guarantee adherence to the underlying physics equations.

To ensure the physical validity of the generated images, we examine the (PDE) residual, $\nabla^2 u_{\text{generated}} - f_{\text{generated}}$, associated with each generated pair (last column in Figure 1). All the samples in the figure correspond to MSE of PDE residual $< 6 \times 10^{-6}$. When comparing them to the solver-generated $u_{\text{generated}}$ (third column), obtained by passing $f_{\text{generated}}$ through a finite-difference solver, we observe that all the generated samples exhibit a very close resemblance between $u_{\text{generated}}$ and u_{solver}.

Table 1. Relative L_2 error between the mean and standard deviation of the solver-generated data distribution and the diffusion-generated data distribution for 2-D Poisson equation. We condition the diffusion statistics for 2-D Poisson on MSE of residual $< 6 \times 10^{-6}$.

DATA	MEAN	STANDARD DEVIATION
2-D POISSON, f	0.007	0.007
2-D POISSON, u	0.004	0.01

Table 1 shows the relative L_2 error between the statistics of synthetic and solver generated f and u. For both variables, the percentage error is quite small , indicating the efficacy of a diffusion-based model in recovering the statistics of the data distribution. For the threshold of MSE of PDE residual $< 6 \times 10^{-6}$, 99962 pairs out of a total generated $100,000$ pairs were admissible and can be used for other downstream tasks.

Figure 1 and table 1 clearly demonstrate that generated data with low PDE residuals demonstrate a very close agreement with the underlying physics equations and exhibit a greater visual and statistical resemblance to solver solutions, making them more suitable for subsequent analysis and utilization. Conversely, generated data with high PDE residuals should be approached with caution, as they may deviate significantly from the desired physical behavior.

4.2. 2-D Forced Navier Stokes Vorticity-Transport Equation

Figure 5 (first and third rows) shows the evolution of diffusion-generated vorticity. Corresponding to the first and third rows, we present the solver-generated vorticity in the second and fourth rows, respectively. The colorbar highlights the magnitudes, confirming that they fall within the expected range. Unlike the analysis performed on 2-D Poisson data, which involved evaluating PDE residuals, we adopt a different approach to validate the physical correctness of the data generated on 2-D NS vorticity-transport equation. The first snapshot is passed as an initial condition to a traditional solver. Hence, only the next four snapshots are compared in figure 5. The top two rows correspond to $\text{MSE} \approx 8 \times 10^{-3}$ between diffusion-generated and solver-generated snapshots. We find that the diffusion-generated snapshots qualitatively capture the flow dynamics – note the roll-up of vortex around $[x, y] = [0.5, 0.5]$ in both diffusion- and solver- generated snapshots (top two rows). For bottom two rows, although the flow dynamics between diffusion and solver generated snapshots look qualitatively consistent, MSE (calculated over the grid and the four snapshots) is one order of magnitude higher, at 0.045.

Figure 3 shows the mean distribution of solver-generated

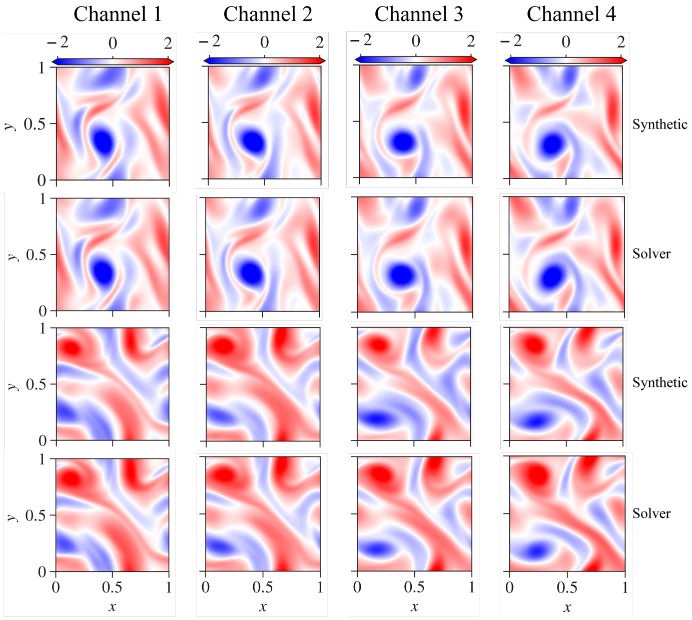

Figure 2. Diffusion- and solver-generated vorticity snapshots for 2-D forced NS system. Top two rows correspond to $\mathrm{MSE} \approx 8 \times 10^{-3}$ between solver-generated and diffusion-generated snapshots. Bottom two rows correspond to $\mathrm{MSE} \approx 0.045$.

vorticity field and diffusion-generated vorticity field. The solver-generated mean field (figure 3b) show inclined patches of alternate sign vorticity, reminiscent of the forcing function that is being applied. We find that the diffusion model is able to qualitatively reproduce the inclined patches, corresponding to the forcing pattern qualitatively (figure 3a). However, it is important to note that the distribution of positive vorticity patches appears to be more dominant in the generated data compared to the solver-generated distribution.

Finally, in addition to quantitative evaluation metrics, we conducted a visual inspection of the generated samples to assess their diversity for both datasets (attached in appendix). It was evident that the generated data samples exhibited a wide range of variations and distinct features for both 2-D Poisson and 2-D NS equations. The visually diverse nature of the generated samples indicates the effectiveness of diffusion model model in producing novel and unique outputs.

5. Conclusions

In this work, we introduced a data generation methodology based on diffusion models and validated it for two canonical physical systems: 2-D Poisson and 2-D forced Navier-Stokes vorticity-transport equation. Our findings

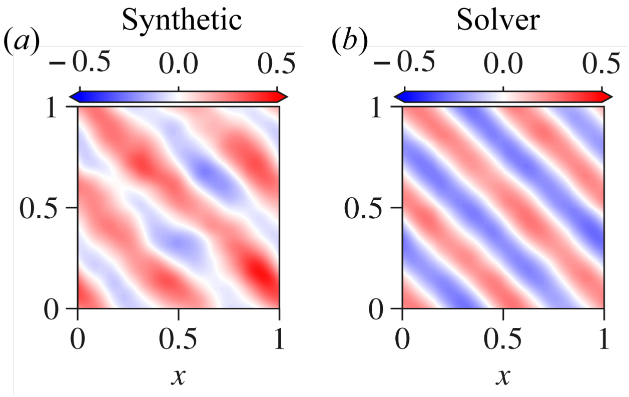

Figure 3. Two-dimensional contours of the mean vorticity field from the (a) diffusion-generated data distribution and (b) solver-generated distribution. For statistics on the diffusion-generated vorticity field, we condition on MSE between diffusion-generated and solver-generated snapshots $< 2 \times 10^{-2}$.

demonstrate that the diffusion model can effectively generate visually and statistically consistent samples. To leverage these samples for subsequent downstream tasks such as training physics-based machine learning algorithms, one can employ PDE-based residuals or solver-based filtering methods to select physically consistent samples. This approach ensures that the generated data adheres to the underlying physics and can be reliably used in further analyses.

It is important to note that this work is ongoing, and there are several future directions to explore. One future direction involves incorporating physics-based losses in the training and sampling algorithm of the diffusion model itself. In our current model, data generation is limited to a specific resolution. However, we are concurrently working on implementing super-resolution techniques for diffusion models. This will enable interpolation between resolutions and the generation of high-fidelity images. We are also exploring the extension of this method to solutions represented on unstructured meshes. Additionally, we plan to utilize the governing parameters of the data to condition the model in the future, enhancing its general-purpose capabilities.

Broader Impact

This research direction can have significant impact on various fields important to society, which also face the issue of data scarcity, e.g., climate science, material science, etc. Diffusion models can contribute to these fields by providing a means to generate physically accurate data for training and validating machine learning algorithms. However, the reliance on machine learning techniques for physics-based simulations may raise concerns about the interpretability and explainability of the models.

References

Bar-Sinai, Y., Hoyer, S., Hickey, J., and Brenner, M. P. Learning data-driven discretizations for partial differential equations. *Proceedings of the National Academy of Sciences*, 116(31):15344–15349, 2019.

Chen, N., Zhang, Y., Zen, H., Weiss, R. J., Norouzi, M., and Chan, W. Wavegrad: Estimating gradients for waveform generation. *arXiv preprint arXiv:2009.00713*, 2020.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

Duraisamy, K. Perspectives on machine learning-augmented reynolds-averaged and large eddy simulation models of turbulence. *Physical Review Fluids*, 6(5):050504, 2021.

Eivazi, H., Le Clainche, S., Hoyas, S., and Vinuesa, R. Towards extraction of orthogonal and parsimonious nonlinear modes from turbulent flows. *Expert Systems with Applications*, 202:117038, 2022.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., and Fleet, D. J. Video diffusion models. *arXiv preprint arXiv:2204.03458*, 2022.

Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., and Anandkumar, A. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

Lim, J. H., Kovachki, N. B., Baptista, R., Beckham, C., Azizzadenesheli, K., Kossaifi, J., Voleti, V., Song, J., Kreis, K., Kautz, J., et al. Score-based diffusion models in function space. *arXiv preprint arXiv:2302.07400*, 2023.

Ling, J., Kurzawski, A., and Templeton, J. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *Journal of Fluid Mechanics*, 807:155–166, 2016.

Lu, L., Jin, P., and Karniadakis, G. E. Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. *arXiv preprint arXiv:1910.03193*, 2019.

Maulik, R., San, O., Rasheed, A., and Vedula, P. Subgrid modelling for two-dimensional turbulence using neural networks. *Journal of Fluid Mechanics*, 858:122–144, 2019.

Murata, T., Fukami, K., and Fukagata, K. Nonlinear mode decomposition with convolutional neural networks for fluid dynamics. *Journal of Fluid Mechanics*, 882:A13, 2020.

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

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

Ranade, R., Hill, C., Ghule, L., and Pathak, J. A composable machine-learning approach for steady-state simulations on high-resolution grids. *arXiv preprint arXiv:2210.05837*, 2022.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

Saharia, C., Ho, J., Chan, W., Salimans, T., Fleet, D. J., and Norouzi, M. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Sehwag, V. minimal-diffusion. https://github.com/VSehwag/minimal-diffusion, 2022.

Shu, D., Li, Z., and Farimani, A. B. A physics-informed diffusion model for high-fidelity flow field reconstruction. *Journal of Computational Physics*, 478:111972, 2023.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pp. 2256–2265. PMLR, 2015.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Song, Y. and Ermon, S. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33:12438–12448, 2020.

Sun, C., Shrivastava, A., Singh, S., and Gupta, A. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pp. 843–852, 2017.

Vlassis, N. N. and Sun, W. Denoising diffusion algorithm for inverse design of microstructures with fine-tuned nonlinear material properties. *arXiv preprint arXiv:2302.12881*, 2023.

Weymouth, G. D. Data-driven multi-grid solver for accelerated pressure projection. *Computers & Fluids*, 246: 105620, 2022.

Yang, G. and Sommer, S. A denoising diffusion model for fluid field prediction. *arXiv e-prints*, pp. arXiv–2301, 2023.

A. Appendix - Diversity of generated data for both 2-D Poisson and 2-D NS

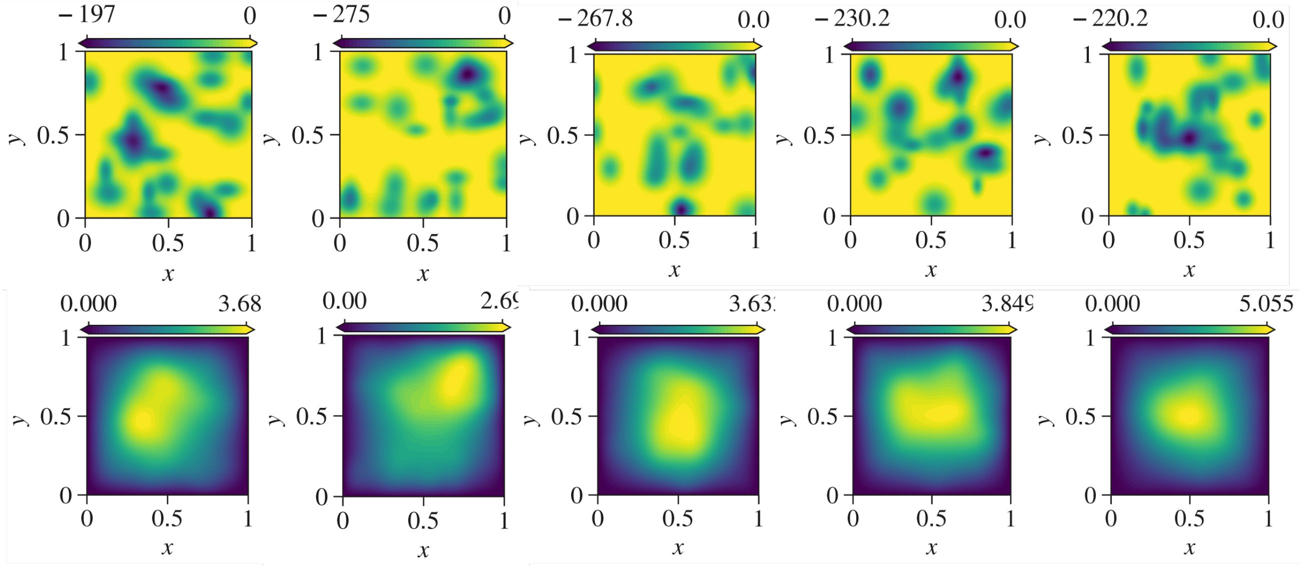

Figure 4. Example set of five randomly chosen $[f, u]$ pairs – f in top row and corresponding u in bottom row for 2-D Poisson equation.

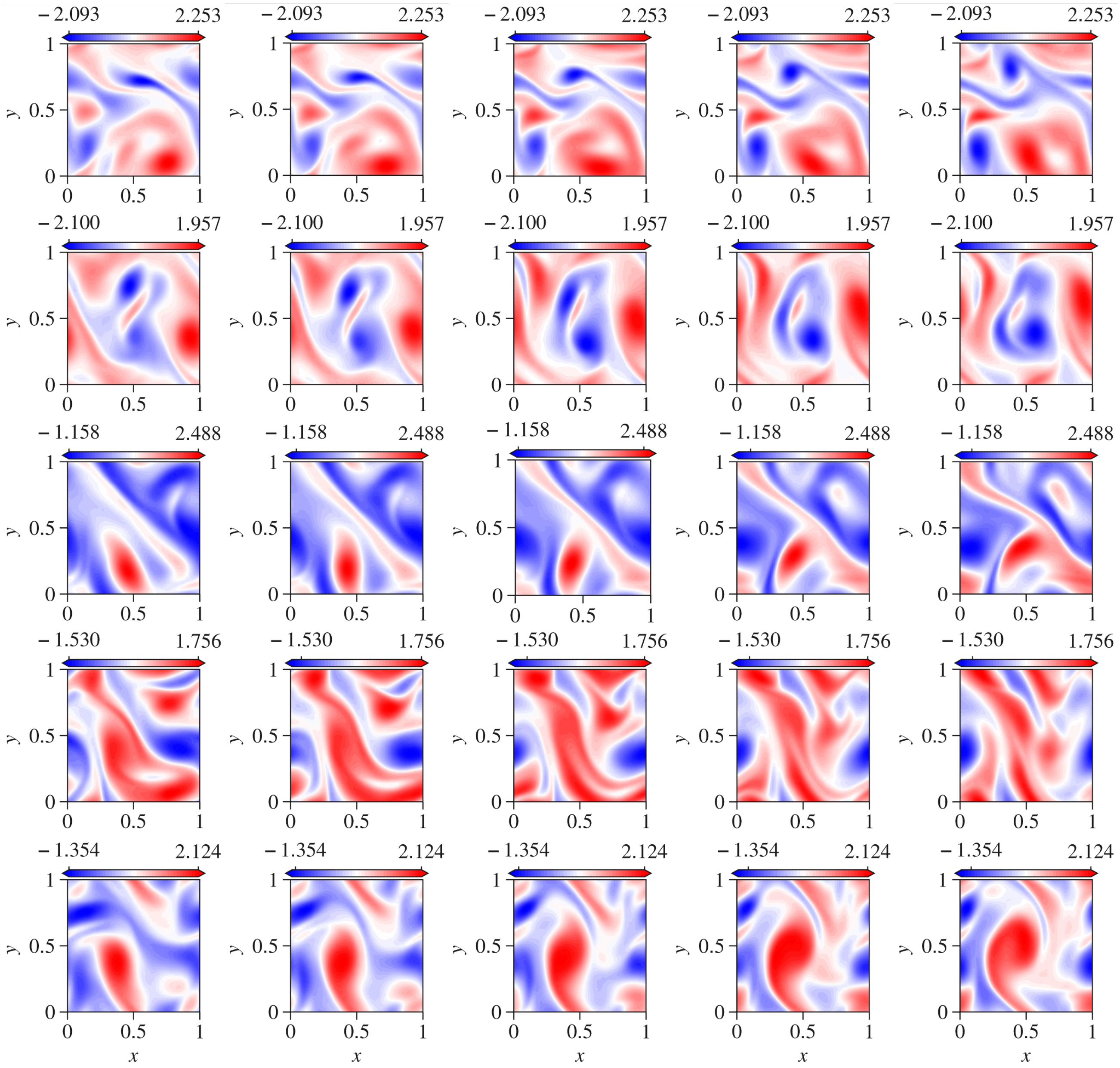

Figure 5. Example set of five randomly chosen vorticity fields for force 2-D NS equation. Snapshot 1 to 5 from leftmost to rightmost columns.