FCNN: Five-point stencil CNN for solving reaction-diffusion equations

Yongho Kim\textsuperscript{a}, Yongho Choi\textsuperscript{b,∗}

\textsuperscript{a}Faculty of Mathematics, Otto-von-Guericke-Universität Magdeburg, Universitätsplatz 2, 39106, Magdeburg, Germany
\textsuperscript{b}Department of Mathematics and Big Data, Daegu University, Gyeongsan-si, Gyeongsangbuk-do 38443, Republic of Korea

Abstract

In this paper, we propose Five-point stencil CNN (FCNN) containing a five-point stencil kernel and a trainable approximation function. We consider reaction-diffusion type equations including heat, Fisher’s, Allen–Cahn equations, and reaction-diffusion equations with trigonometric functions. Our proposed FCNN is trained well using few data and then can predict reaction-diffusion evolutions with unseen initial conditions. Also, our FCNN is trained well in the case of using noisy train data. We present various simulation results to demonstrate that our proposed FCNN is working well.

Keywords: convolutional neural network, reaction-diffusion type equations, data-driven model

1. Introduction

To express diverse natural phenomena such as sound, heat, electrostatics, elasticity, thermodynamics, fluid dynamics, and quantum mechanics mathematically, various partial differential equations (PDEs) have been derived and numerical methods can be applied to solve these PDEs. Representative numerical methods for solving PDEs are the finite difference method, finite element

∗Corresponding author

Email addresses: ykim@mpi-magdeburg.mpg.de (Yongho Kim), yongho_choi@daegu.ac.kr (Yongho Choi)

URL: https://sites.google.com/view/yh-choi (Yongho Choi)
method, finite volume method, spectral method, etc. We focus on the finite difference method (FDM) which is to divide a given domain into finite grids and find an approximate solution using derivatives with finite differences \[1\]. This method uses each and its neighbor points to predict the corresponding point at the next time step. Likewise, in convolutional neural networks (CNNs) \[20\], convolution operators extract each pixel of an output by using the corresponding pixel and its neighbor pixels of an input. Also, the convolution operator is basically immutable. Hence, well-structured convolutional neural networks have a potential to solve partial differential equations numerically. Therefore, we propose Five-point stencil CNN (FCNN) containing a five-point stencil kernel and a trainable approximation function to obtain numerical solutions of the PDEs. Among the various PDEs representing natural phenomena, we deal with reaction-diffusion type equations. The reaction-diffusion model has been applied and used in various fields such as biology \[3, 4, 5\], chemistry \[6, 7, 8\], image segmentation \[9, 10, 11\], image inpainting \[12, 13, 14\], medical \[15, 16, 17\], and so on. In this paper, we use second order reaction-diffusion type equations: heat, Fisher’s, Allen–Cahn (AC) equation, and reaction-diffusion equations with trigonometric functions terms.

In recent years, neural networks have been widely applied to solve PDEs. Physics-informed neural networks (PINNs) \[22\] based on multi-layer perceptron (MLP) models approximate solutions by the optimization of a loss function consisting of given physics laws. The biggest benefit of PINNs is that solutions can be inferred without any iterative process such as a recurrence equation with respect to time. Furthermore, it is used for diverse applications such as Hidden Fluid Mechanics \[25\] that extracts hidden variables of a given equation using a PINN and observations. However, it is hard to optimize model parameters when we deal with complicated PDEs and their coefficients. In order to improve the training ability, combinations of PINNs and numerical methods have been developed, or other neural networks such as CNNs are selected. M. Raissi et al. \[22\] added Runge-Kutta methods to a PINN model for solving AC equation. Aditi et al. \[24\] proposed transfer learning and curriculum regularization which
start training PINNs on a specific safe domain and then transfer to a target domain. Hao Ma et al. [24] proposed a U-shape CNN so-called U-net [27] and they showed that the usage of target data in a loss function significantly improves optimization. Elie Bretin et al. [26] used convolutional neural networks derived from a semi-implicit approach to learn phase field mean curvature flows of the AC equation.

We propose data-driven models that approximate the solution of explicit finite difference scheme to solve second order reaction-diffusion type equations numerically. Our contributions are as follows:

1. We propose a five-point stencil convolution operator to solve reaction-diffusion type equations.
2. Our proposed model is trained using two consecutive snapshots to solve a given equation.
3. We demonstrate the robustness of our method using five reaction-diffusion type equations and noisy data.

The remainder of this paper is organized as follows. In Section 2, we present how to create training data using explicit FDM, explain the FCNN concept, training process, and numerical solutions. In Section 3, we compare the prediction results using our proposed FCNN and the evolution of PDEs results using the FDM method and show the robustness of our FCNN. Finally conclusions are drawn in Section 4.

2. Methods and numerical solutions

FDM is to divide a given domain into finite grids and find an approximate solution using derivatives with finite differences [1]. We use explicit FDM to create training data with random initial conditions. We use only two consecutive FDM results, the initial and next time step results, as training data for each equation. To create training data, a computational domain is defined using a uniform grid of size $h = 1/N_x = 1/N_y$ and $\Omega_h = \{(x_i, y_j) = (a + (i-1.5)h, c + (j-1.5)h)\}$ for $1 \leq i \leq N_x + 2$ and $1 \leq j \leq N_y + 2$. Here, $N_x$ and $N_y$ are mesh
sizes on the computational domain \((a, b) \times (c, d)\). Let \(\phi_{ij}^n\) be approximations of \(\phi(x_i, y_j, n\Delta t)\) and \(\Delta t\) is temporal step size. The boundary condition is zero Neumann boundary condition. Laplacian of a function \(\phi\) is calculated using a five-point stencil method, the Laplacian \(\nabla^2 \phi\) can be approximated as follows:

\[
\nabla^2 \phi \approx \frac{\phi(x + h, y) + \phi(x - h, y) - 4\phi(x, y) + \phi(x, y + h) + \phi(x, y - h)}{h^2}.
\]

(1)

In this way, the first and second derivatives of \(\phi\) at each point \((x_i, y_j)\) (e.g., \(\phi_x, \phi_y, \phi_{xx}\) and \(\phi_{yy}\)) can be approximated within the \(3 \times 3\) local area centered \((x_i, y_j)\). This concept can be equivalent to \(3 \times 3\) convolution kernels. The \(3 \times 3\) kernels \(K\) following properties:

1. \(k_1 \oplus k_2 \in K\) for any \(k_1, k_2 \in K\) (element-wise summation)
2. \(k_1 \odot k_2 \in K\) for any \(k_1, k_2 \in K\) (element-wise multiplication)
3. \(k^{-1} \in K\) for any \(k \in K\) (element-wise division)
4. \(ak \in K\) for any \(k \in K\) and any real numbers \(a\)

The second-order PDEs can be solved numerically using combinations of \(3 \times 3\) kernels. Therefore, if we build a proper CNN as the form of recurrence Eq. (4), we can solve a PDE (2) numerically. A previous study about AC equation \([19]\) shows that FDM can be expressed as CNN.

\[
\frac{\partial \phi}{\partial t} = g(\phi(x, y, t)),
\]

(2)

\[
\frac{\phi^{n+1} - \phi^n}{\Delta t} = g(\phi^n),
\]

(3)

\[
\phi^{n+1} = \phi^n + \Delta tg^n.
\]

(4)

To solve second order reaction-diffusion type equations

\[
\phi_t = \alpha \nabla^2 \phi + \beta f(\phi),
\]

(5)

where \(\alpha\) is a diffusion coefficient, \(\beta\) is a reaction coefficient, and \(f\) is a smooth function to present reaction effect, we propose FCNN as a recurrence relation:

\[
\phi^{n+1} = \phi^n + \Delta t\alpha \nabla^2 \phi^n + \Delta t\beta f(\phi^n).
\]

(6)

As a CNN, \(F(\phi^n)\) containing a 5-point stencil filter and a pad satisfying given boundary conditions solves \(\Delta t\alpha \nabla \phi^n\). In order to approximate \(\phi^n + \Delta t\beta f(\phi^n)\)
terms, we define a trainable polynomial function $\epsilon(\phi^n)$ as follows:

$$
\epsilon(\phi^n) = a_0 + \sum_{k=1}^{N} a_i (\phi^n - b)^k,
$$

(7)

with model parameters $a_i$ for any $i \in \{0,1,\cdots,N\}$ and a real value $b$. Let $M(\phi^n) = F(\phi^n) + \epsilon(\phi^n)$ be a FCNN. Then, the inference is as follows:

$$
\phi^{n+1} = M_{\theta}(\phi^n),
$$

(8)

where $\theta$ is a set of model parameters. Figure 1 shows the computational graph of our explicit model FCNN containing model parameters $w_i$ for any $i \in \{0,1,2,3,4\}$ in a filter. Furthermore, $F$ represents the diffusion term on the uniform grid of $x$ and $y$ axes, so we set up $w_1 = w_3$ and $w_0 = w_4$ to cut down on training time. When the five-point stencil filter is used and $\epsilon$ is a $p$-th order polynomial function, the number of model parameters is only $p + 4$. Thus, the set-up enables to learn physical patterns from few data. In Algorithm 1, an initial image $\phi^0(=u^0)$ and the prediction $\phi^1$ at the next time $\Delta t$ are used with train data $u^0$ and $u^1$ to train a model $M_{\theta}$. The objective function $L(\phi^1,u^1)$ is the mean square error function as follows:

$$
L(\phi^1,u^1) = \frac{1}{N} \sum_{i=1}^{N} (\phi_i^1 - u_i^1)^2
$$

(9)
Algorithm 1: Training Procedure

Set an initial value $\phi^0 = u^0$, a small constant $\delta > 0$

Initialize $M_\theta(\phi^0) = F(\phi^0) + \epsilon(\phi^0)$ with model parameters $\theta$

while $\ell > \delta$ do

$\phi^1 \leftarrow M_\theta(\phi^0)$

Compute loss $\ell = L(\phi^1, u^1)$

Update $\theta$

end while

where $N$, $\phi^1$ and $u^1$ are the number of pixels in an output image, a prediction and its target respectively.

2.1. Reaction-diffusion type equations

To demonstrate the robustness of FCNN, we consider reaction-diffusion type equations: heat, Fisher’s, AC equation, reaction-diffusion equations with trigonometric functions. The reaction and diffusion coefficients used in each formula are arranged in Table 1. For the AC equation, $\beta = 1/\rho^2$ where $\rho$ is the thickness of the transition layer and which value is $\rho_0 \approx 0.012$ [19]. For the other equations, we select arbitrary coefficients. For all the following equations, the continuous equations and the discretized equations are described in turn, and the zero Neumann boundary condition is used.

|       | Heat | Fisher’s | AC  | Sine | Tanh |
|-------|------|----------|-----|------|------|
| $\alpha$ | 1    | 1        | 1   | 0.1  | 0.5  |
| $\beta$  | 0    | 20       | 6944| 40   | 10   |

Table 1: Diffusion ($\alpha$) and reaction ($\beta$) coefficients for the simulations.
2.1.1. Heat equation

\[ \phi_t = \alpha \Delta \phi, \quad (10) \]
\[ \phi^{n+1} = \text{conv}(\phi^n) + \phi^n. \quad (11) \]

2.1.2. Fisher’s equation

\[ \phi_t = \alpha \Delta \phi + \beta (\phi - \phi^2), \quad (12) \]
\[ \phi^{n+1} = \text{conv}(\phi^n) + \phi^n + \Delta t \beta (\phi^n - (\phi^n)^2). \quad (13) \]

2.1.3. AC equation

\[ \phi_t = \alpha \Delta \phi + \beta (\phi - \phi^3), \quad (14) \]
\[ \phi^{n+1} = \text{conv}(\phi^n) + \phi^n + \Delta t \beta (\phi^n - (\phi^n)^3). \quad (15) \]

2.1.4. Reaction-diffusion equation with trigonometric function: sin

\[ \phi_t = \alpha \Delta \phi + \beta \sin(\pi \phi), \quad (16) \]
\[ \phi^{n+1} = \text{conv}(\phi^n) + \phi^n + \Delta t \beta \sin(\pi \phi^n). \quad (17) \]

2.1.5. Reaction-diffusion equation with trigonometric function: tanh

\[ \phi_t = \alpha \Delta \phi + \beta \tanh(\phi), \quad (18) \]
\[ \phi^{n+1} = \text{conv}(\phi^n) + \phi^n + \Delta t \beta \tanh(\phi^n). \quad (19) \]

where \( \text{conv}(\phi^n) = \Delta t \alpha \Delta \phi^n \) and each discretized equation \((11), (13), (15), (17), (19) \) is implemented based on the model structure proposed in [19]. When \( \alpha = 1 \) and \( \beta = 1 \), all the equations show the almost similar evolution so we use different reaction coefficient \( \beta \) much bigger than diffusion coefficient \( \alpha \) to check diverse evolutions as shown in Table [1].
3. Simulation results

Assume that we observe a reaction-diffusion pattern and investigate the pattern rule under the constraint meaning that the observations and predictions follow the same PDE. Our proposed FCNN is trained using only two consecutive data which are the initial and next time step results for each equation. Then, we evaluate the model using diverse unseen initial values.

In the simulations, we use random initial value data with $100 \times 100$ mesh so that the size of the input data is $102 \times 102$ containing a pad as a boundary condition. Also, $N = 3$ (Heat, Fisher’s, AC) or 9 (Sine, Tanh) for $\epsilon(\phi^n)$ is fixed depending on given equations and a $3 \times 3$ convolutional filter is used with the stride of 1 in Eq. (7). Hence, the filter has $10,000 = ((100 + 2 – 3)/1 + 1)^2$ chances to learn the evolution of results images, so training a model using only two consecutive images are enough to optimize nine or thirteen model parameters ($w_0, \cdots, w_4, a_0, \cdots, a_3$). As an optimizer, ADAM [21] is used with a learning rate of 0.01 and without any regularization. Instead, we apply early stopping [28] based on a validation data to avoid overfitting. To demonstrate the approximation $\epsilon(\phi^n)$ for non-polynomial functions $f(\phi^n)$, we additionally consider sine and tanh functions besides heat, Fisher’s, and AC equations.

For the evaluation, we implement FCNN and FDM respectively and then measure the averaged relative $L_2$ error with 95% confidence interval over 100 novel random initial values as shown in Table 2.

Table 2: Relative $L_2$ error between FCNN and FDM. The $\pm$ shows 95% confidence intervals over 100 different random initial values.

| Equations | Relative $L_2$ error |
|-----------|-----------------------|
| Heat      | $8.4 \times 10^{-5} \pm 4 \times 10^{-6}$ |
| Fisher’s  | $4.0 \times 10^{-5} \pm 2 \times 10^{-6}$ |
| AC        | $1.3 \times 10^{-6} \pm 8 \times 10^{-7}$ |
| Sine      | $7.0 \times 10^{-5} \pm 5 \times 10^{-6}$ |
| Tanh      | $1.9 \times 10^{-4} \pm 4 \times 10^{-6}$ |
Furthermore, we validate the errors using different types of initial values for each equation as shown in Table 3. The initial conditions are described in the Appendix Section.

Table 3: Relative $L_2$ error between FCNN and FDM with diverse initial values.

| Initial shapes: | circle | star | circles | torus | maze |
|-----------------|--------|------|---------|-------|------|
| Heat            | $3.4 \times 10^{-5}$ | $4.4 \times 10^{-5}$ | $1.1 \times 10^{-5}$ | $1.1 \times 10^{-4}$ | $4.9 \times 10^{-5}$ |
| Fisher’s        | $8.7 \times 10^{-4}$ | $7.2 \times 10^{-4}$ | $1.9 \times 10^{-4}$ | $1.3 \times 10^{-4}$ | $3.7 \times 10^{-5}$ |
| AC              | $2.6 \times 10^{-7}$ | $2.7 \times 10^{-7}$ | $2.3 \times 10^{-7}$ | $2.0 \times 10^{-7}$ | $1.9 \times 10^{-7}$ |
| Sine            | $3.7 \times 10^{-4}$ | $2.2 \times 10^{-4}$ | $9.5 \times 10^{-5}$ | $7.5 \times 10^{-5}$ | $4.1 \times 10^{-5}$ |
| Tanh            | $1.8 \times 10^{-3}$ | $1.4 \times 10^{-3}$ | $6.3 \times 10^{-4}$ | $2.5 \times 10^{-5}$ | $2.9 \times 10^{-5}$ |

Figures 2–6 show the time evolution results when unseen initial shapes (circle, star, three circles, torus, and maze) are given after learning with two training data (random initial condition and next time step result with FDM). We compare the predicted results from pretrained models to the FDM results.

Table 4: Relative $L_2$ error with noise. The ± shows 95% confidence intervals over 100 different random initial values.

| $\sigma$ | Relative $L_2$ Error |
|----------|-----------------------|
| $10^{-6}$ | $9.1 \times 10^{-5} \pm 6 \times 10^{-4}$ |
| $10^{-4}$ | $3.3 \times 10^{-4} \pm 2 \times 10^{-4}$ |
| $10^{-2}$ | $1.4 \times 10^{-1} \pm 3 \times 10^{-2}$ |

Data-driven models are sensitive to data noise. To investigate the noise effect, we inject Gaussian random noise $\eta \sim N(0, \sigma^2)$ to $u^1$ and then the model is trained using $u_0$ and $u_1 + \eta$ for the AC equation. Table 4 shows that the model can be trained under the noise condition. Figure 7 displays the results of the inference using contaminated models.
Figure 2: Time evolution of a circle shape of (a) Heat, (b) Fisher's, (c) AC, (d) Sine, and (e) Tanh equations.
Figure 3: Time evolution of a star shape of (a) Heat, (b) Fisher’s, (c) AC, (d) Sine, and (e) Tanh equations.
Figure 4: Time evolution of three circles of (a) Heat, (b) Fisher’s, (c) AC, (d) Sine, and (e) Tanh equations.
Figure 5: Time evolution of a torus shape of (a) Heat, (b) Fisher’s, (c) AC, (d) Sine, and (e) Tanh equations.
Figure 6: Time evolution of a maze shape of (a) Heat, (b) Fisher’s, (c) AC, (d) Sine, and (e) Tanh equations.
4. Conclusions

In this paper, we proposed Five-point stencil CNN (FCNN) containing a five-point stencil kernel and a trainable approximation function. We considered reaction-diffusion type equations including heat, Fisher’s, Allen–Cahn equations, and reaction-diffusion equations with trigonometric functions. We showed that our proposed FCNN can be trained well using few data (used only two consecutive data) and then can predict reaction-diffusion evolution with unseen diverse initial conditions. Also, we demonstrated the robustness of our FCNN under the noise condition.
Acknowledgments

The corresponding author Y. Choi was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (NRF-2020R1C1C1A0101153712).

Appendix

In this appendix session, we describe the initial conditions used in the simulation results session. A detailed description of these initial conditions can be found in our previous research paper [19].

1. The initial condition of a circle shape

\[
\phi(x, y, 0) = \tanh \left( \frac{R_0 - \sqrt{(x-0.5)^2 + (y-0.5)^2}}{\sqrt{2}\epsilon} \right), \quad (20)
\]

where \( R_0 \) is the initial radius of a circle.

2. The initial condition of a star shape

\[
\phi(x, y, 0) = \tanh \left( \frac{0.25 + 0.1 \cos(6\theta) - \sqrt{(x-0.5)^2 + (y-0.5)^2}}{\sqrt{2}\epsilon} \right), \quad (21)
\]

where

\[
\theta = \begin{cases} 
\tan^{-1} \left( \frac{y-0.5}{x-0.5} \right), & \text{if } (x > 0.5) \\
\pi + \tan^{-1} \left( \frac{y-0.5}{x-0.5} \right), & \text{otherwise}.
\end{cases}
\]

3. The initial condition of a torus shape

\[
\phi(x, y, 0) = -1 + \tanh \left( \frac{R_1 - \sqrt{XY}}{\sqrt{2}\epsilon} \right) - \tanh \left( \frac{R_2 - \sqrt{XY}}{\sqrt{2}\epsilon} \right), \quad (22)
\]

where \( R_1 \) and \( R_2 \) are the radius of major (outside) and minor (inside) circles, respectively. And, for simplicity of expression, \( XY = (x-0.5)^2 + (y-0.5)^2 \).

4. The initial condition of a maze shape
The initial condition of a maze shape is complicated to describe its equation, so refer to the codes which are available from the first author’s GitHub web page (https://github.com/kimy-de) and the corresponding author’s web page (https://sites.google.com/view/yh-choi/code).

(5) The initial condition of a random shape

\[ \phi(x, y, 0) = \text{rand}(x, y), \]

here the function \( \text{rand}(x, y) \) has a random value between \(-1\) and \(1\).

References

[1] P. Zhou, Numerical analysis of electromagnetic fields. Springer Science & Business Media, 2012.

[2] S Kondo, T Miura, "Reaction-diffusion model as a framework for understanding biological pattern formation." science 329.5999 (2010): 1616-1620.

[3] NF Britton, Reaction-diffusion equations and their applications to biology. Academic Press, 1986.

[4] P Broadbridge, BH Bradshaw-Hajek, "Exact solutions for logistic reaction–diffusion equations in biology.” Zeitschrift für angewandte Mathematik und Physik 67.4 (2016): 1-13.

[5] D Jeong, Y Li, Y Choi, M Yoo, D Kang, J Park, J Choi, J Kim "Numerical simulation of the zebra pattern formation on a three-dimensional model.” Physica A: Statistical Mechanics and its Applications 475 (2017): 106-116.

[6] BA Grzybowski, Chemistry in motion: reaction-diffusion systems for micro-and nanotechnology. John Wiley & Sons, 2009.

[7] I Sgura, B Bozzini, D Lacitignola, "Numerical approximation of oscillating Turing patterns in a reaction-diffusion model for electrochemical material
growth.” AIP Conference Proceedings. Vol. 1493. No. 1. American Institute of Physics, 2012.

[8] G Hariharan, R Rajaraman, "A new coupled wavelet-based method applied to the nonlinear reaction–diffusion equation arising in mathematical chemistry." Journal of Mathematical Chemistry 51.9 (2013): 2386-2400.

[9] H Tek, BB Kimia, "Image segmentation by reaction-diffusion bubbles.” Proceedings of IEEE International Conference on Computer Vision. IEEE, 1995.

[10] S Esedog, YHR Tsai, "Threshold dynamics for the piecewise constant Mumford–Shah functional.” Journal of Computational Physics 211.1 (2006): 367-384.

[11] Z Zhang, YM Xie, Q Li, S Zhou, "A reaction–diffusion based level set method for image segmentation in three dimensions.” Engineering Applications of Artificial Intelligence 96 (2020): 103998.

[12] M Bertalmio et al. "Image inpainting.” Proceedings of the 27th annual conference on Computer graphics and interactive techniques. 2000.

[13] Y Li, D Jeong, J Choi, S Lee, J Kim, "Fast local image inpainting based on the Allen–Cahn model.” Digital Signal Processing 37 (2015): 65-74.

[14] J Yu, J Ye, S Zhou, "Reaction-diffusion system with additional source term applied to image restoration.” International Journal of Computer Applications 975 (2016): 8887.

[15] E Özuğurlu, ”A note on the numerical approach for the reaction–diffusion problem to model the density of the tumor growth dynamics.” Computers & Mathematics with Applications 69.12 (2015): 1504-1517.

[16] HG Lee, Y Kim, J Kim, ”Mathematical model and its fast numerical method for the tumor growth.” Mathematical Biosciences & Engineering 12.6 (2015): 1173.
[17] M Yousefnezhad, CY Kao, SA Mohammadi, "Optimal Chemotherapy for Brain Tumor Growth in a Reaction-Diffusion Model." SIAM Journal on Applied Mathematics 81.3 (2021): 1077-1097.

[18] S.M. Allen, J.W. Cahn, “A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening.” Acta metallurgica 27.6 (1979): 1085–1095.

[19] Yongho Kim, Gihnam Ryu, and Yongho Choi (2021) Fast and Accurate Numerical Solution of Allen-Cahn Equation, Mathematical Problems in Engineering, vol. 2021, Article ID 5263989, 12 pages, 2021. https://doi.org/10.1155/2021/5263989

[20] LeCun, Yann and Boser, Bernhard and Denker, John and Henderson, Donnie and Howard, R. and Hubbard, Wayne and Jackel, Lawrence, (1990), Handwritten Digit Recognition with a Back-Propagation Network, Advances in Neural Information Processing Systems, Vol 2.

[21] Diederik P. Kingma and Jimmy Ba, (2017). Adam: A Method for Stochastic Optimization, arXiv:1412.6980.

[22] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis, (2018). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, Journal of Computational Physics 378 (2019): 686-707.

[23] Aditi S. Krishnapriyan, Amir Gholami, Shandian Zhe, Robert M. Kirby, and Michael W. Mahoney, (2021). Characterizing Possible Failure Modes in Physics-Informed Neural Networks, Neural Information Processing Systems (NeurIPS) 2021, arXiv:2109.01050.

[24] Hao Ma, Yuxuan Zhang, Nils Thuerey, Xiangyu Hu, Oskar J. Haidn, (2021). Physics-driven Learning of the Steady Navier-Stokes Equations using Deep Convolutional Neural Networks, arXiv:2106.09301.
[25] Maziar Raissi, Alireza Yazdani, and George Em Karniadakis. (2020) Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations, Science 367.6481 (2020): 1026-1030.

[26] Elie Bretin, Roland Denis, Simon Masnou, and Garry Terii, (2021). Learning Phase Field Mean Curvature Flows With Neural Networks, arXiv:2112.07343.

[27] Olaf Ronneberger, Philipp Fischer and Thomas Brox, (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation, Medical Image Computing and Computer-Assisted Intervention (MICCAI) 2015, Lecture Notes in Computer Science, vol 9351. Springer, Cham.

[28] Lutz Prechelt, (1998). Early Stopping - but when?, In: Orr G.B., Müller KR. (eds) Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science, vol 1524.