

RESEARCH ARTICLE

Open Access



Physics-informed neural networks approach for 1D and 2D Gray-Scott systems

Fabio Giampaolo , Mariapia De Rosa, Pian Qi, Stefano Izzo and Salvatore Cuomo

*Correspondence:

fabio.giampaolo@unina.it

Dipartimento di Matematica e Applicazioni "R.Caccioppoli",
Università Degli Studi di Napoli
Federico II, Via Cupa Cinthia,
80126 Naples, Italy

Abstract

Nowadays, in the Scientific Machine Learning (SML) research field, the traditional machine learning (ML) tools and scientific computing approaches are fruitfully intersected for solving problems modelled by Partial Differential Equations (PDEs) in science and engineering applications. Challenging SML methodologies are the new computational paradigms named Physics-Informed Neural Networks (PINNs). PINN has revolutionized the classical adoption of ML in scientific computing, representing a novel class of promising algorithms where the learning process is constrained to satisfy known physical laws described by differential equations. In this paper, we propose a PINN-based computational study to deal with a non-linear partial differential equations system. In particular, using this approach, we solve the Gray-Scott model, a reaction–diffusion system that involves an irreversible chemical reaction between two reactants. In the unstable region of the model, we consider some a priori information related to dynamical behaviors, i. e. a supervised approach that relies on a finite difference method (FDM). Finally, simulation results show that PINNs can successfully provide an approximated Grey-Scott system solution, reproducing the characteristic Turing patterns for different parameter configurations.

Keywords: Physics-Informed Neural Networks, Scientific Machine Learning, Gray-Scott systems

Introduction

Over the last 10 years huge research efforts have been devoted to the study and application of reaction–diffusion models in real-world problems. The physical processes of reaction and diffusion are usually modelled by differential problems of the following type:

$$\begin{cases} -\operatorname{div}(\mu \nabla u) + \sigma u = f & \text{in } \Omega \\ u = 0 & \text{on } \partial \Omega \end{cases} \quad (1)$$

where Ω is a set in \mathbb{R}^n , $\mu \in L^\infty(\Omega)$ and $\sigma, f \in L^2(\Omega)$. This kind of models are widespread in many fields of physics, as electromagnetism [11], heat transfer [3] and biological sciences. Alan Turing, in a well-known manuscript [35] published in the 1952, assessed that reaction–diffusion models could be used to shape the chemical basis of morphogenesis.

This paper will deal with the Gray-Scott reaction–diffusion model. Such a system involves an irreversible chemical reaction between two generic substances U and V , whose concentration at a given point in space and time is modeled by the functions u and v . They react to each other and diffuse through the medium; therefore, the concentration of U and V at any given location changes with time and can differ from that at other locations. The Gray-Scott problem deals with two main chemical reactions:



where P is an inert product. The behavior of the system is described by the following coupled partial differential equations:

$$\begin{aligned} \frac{\partial u}{\partial t} &= -uv^2 + F(1 - u) + \Delta u \\ \frac{\partial v}{\partial t} &= uv^2 - (F + K)v + \Delta v. \end{aligned} \quad (3)$$

where F and K are two parameters called feed rate and kill rate, respectively, while D_u and D_v are diffusion's coefficients (for additional details see Section 3). A peculiarity of the Gray-Scott system is the high instability of the analytical problem due to diffusion terms: Turing [35] discovered that a stable stationary state tends to become unstable when a diffusive phenomenon occurs, which can lead to the reproduction of different configurations called *Turing patterns*.

In the literature, several numerical methods have been proposed to simulate Gray-Scott systems and generate the Turing patterns. In fact, this problem is generally solved with some state-of-the-art finite difference and Galerkin methods. Conversely, our work aims to study and apply a novel physics-informed neural networks (PINNs) methodology. In recent years, the field of machine learning has been greatly developed, from model improvement to algorithm optimization, and cross-application in other fields [17, 18, 33]. The PINN used in this paper is an emerging machine learning method. It adds constraints of physical conditions on the basis of traditional neural networks, making the predicted results more in line with natural laws of the addressed problem. Raissi et al. [29–31] introduced the concept of the physics-informed neural network to solve forward and inverse problems considering different types of PDEs, whose parameters involved in the governing equation are obtained from the training data. In particular PINNs approximate the solution of PDEs by defining a surrogate model for the differential problem by using artificial neural networks. Such a model is obtained by a learning procedure that minimizes a cost function, called *loss function*, that depends on the physics constraint, which acts as a penalizing term reducing the space of admissible solutions.

The primary goal of this research study is focused on the definition of a computational approach to solve a Gray-Scott system (3) by means of the physics-informed neural networks. The main contributions of this paper can be summarized as follows:

- (i) We have designed a physics-informed neural network strategy for 1D and 2D Gray-Scott systems;
- (ii) We have designed a suitable loss function taking into account some prior information related to dynamical behaviours;
- (iii) We have compared, in the one-dimensional case, the solutions predicted by the physics-informed neural network with the analytical ones;

- (iv) We have compared, in the two-dimensional case, the solutions predicted by physics-informed neural network with the state-of-the-art finite difference and Galerkin algorithms.

The rest of the paper is organized as follows: Section 2 presents a brief literature overview, Section 3 describes the parameters and the dynamics of the model, Section 4 shows the methodology used to address the problem with PINNs, Section 5 presents and discusses the obtained results. Finally, Section 6 concludes the paper.

Related works

As discussed in the previous section, Turing patterns can arise from the Gray-Scott model for particular configurations of the parameters of the problem. These kind of patterns are ubiquitous in nature, such as in fish patterns [2], leopard spots and zebra stripes [21]. Turing patterns can even describe the growth and distribution of vegetation in wasteland under the action of water and other influencing factors. Introduced in 1984 concerning the autocatalytic reaction in isothermal, continuous stirred tank reactors [8], Gray-Scott systems have known a growing interest on the part of scientific community: many experiments and numerical simulations have been performed, e.g. to explore the patterns, stability, and dynamics [1, 9, 22, 24]. Numerically, Rodrigo et al. [32] obtained the exact solutions of the reaction–diffusion equations by introducing ansatz to transform the original 1D Gray-Scott system into a new system. Mazin et al. [20] replicated and extended previous works and explored the pattern formation from a bifurcation analysis perspective. Korkmaz et al. [16] combined the implicit Rosenbrock method with the exponential B-spline configuration method to solve the numerical solution of the 1D autocatalytic system. Manaa et al. [19] numerically solved the 1D model using successive approximation and finite difference methods. A similar problem was also solved by Owolabi et al. [25]. Yadav et al. [36] proved the existence and uniqueness of the solution of the reaction-diffusion model by using Banach’s fixed point theorem and obtained the approximate solution of the 1D Gray-Scott problem by using a Galerkin finite element method. Pearson [27] performed a numerical simulation of a 2D Gray-Scott system using forward Euler integrals in the 2.5×2.5 domain, and found 12 different spatio-temporal patterns by varying the parameters. Chen et al. [4] investigated the stability and dynamics of localized speckle patterns in a 2D model. Similarly, Kolokolnikov et al. [15] discuss the problem of *zigzag* and fracture instability of stripes and rings in 2D models. Raei et al. [28] used the implicit differential stepping method to semi-discretize the Gray-Scott model in the time direction, and used the RBF-FD algorithm combined with the closest point method to solve the 3D problem. Numerical simulations of a multimodal coupled model of the Gray-Scott system were performed by Owolabi et al. [25].

In recent years, a new numerical computing method called Physics-informed neural networks (PINNs) is gaining popularity among researchers in the fields of science and engineering [13]. The basic idea of the PINNs method [6] is to exploit the laws of physics in the form of differential equations to train neural networks. Unlike the data-driven neural network approach, the PINNs approach saves data costs while maximizing compliance with physical constraints of the problem. Many achievements in research in this field have been reached: Raissi et al. [31] used physical-informed neural networks to solve the forward and inverse problems of some classical nonlinear partial differential equations

under continuous and discrete-time conditions. Nascimento et al. [23] solved the fatigue crack propagation problem by integrating ordinary differential equations with recurrent neural networks. Pan et al. [26] learned the continuous-time Koopman operator through a machine learning method based on physical information and applied it to nonlinear dynamical systems in the field of fluid dynamics. Jagtap et al. [12] proposed a physical-informed neural network that obeys the hyperbolic conservation law in discrete domains, and applied it to the solution of forward and inverse problems. Tartakovsky et al. [34] proposed a PINNs method for estimating parameters and unknown physics (constitutive relations) in a PDE model and verified it in both linear and nonlinear diffusion equations.

Remarks on model stability

With reference to the differential problem (3), the Gray-Scott system describes the interaction of two chemical species U and V through the coupled dynamics of their concentrations u and v in the time-space domain. On the left-hand side of each equation the time derivative of one of these concentrations describes the rate at which it changes. The right-hand sides of both equations presents three separate terms: the reaction (2) takes place at a rate proportional to the concentration of U times the square of the concentration of V , so the first term uv^2 represents the *reaction rate*. Since the reaction uses up U and generates V , all of the chemical U will eventually get used up unless there is a way to replenish it. Then, the parameter F , called *feed rate*, manages the importance of the *replenishment* term $F(1 - u)$, while D_u is the *diffusion* term of U . The second equation, similarly to the first one, contains the term $(F + K)v$ which represents the *diminishment* factor and serves to limit the increase concentration V . The parameter K is called *kill rate*, it manages the rate at which V is converted to an inert product P and is multiplied by v since the substance V depends on its concentration.

Stability

Physical systems governed by partial differential equations generate interest in mathematics as regards, for example, the existence of equilibrium solutions and its dependence on the parameters of the problem. Since equilibrium may exist only for certain values of such parameters, a physical problem, to be well-posed (in addition to the existence and uniqueness of the solution and its continuous dependence on the data), needs a check about the stability of the solution. In this section we'll deal with the stability of the uniform steady state of the Gray-Scott model. In this perspective, the study about stability of a system allows to know, after a certain threshold, the evolution over time of the problem's solution. We use the following Theorem to analyze the stability.

Theorem 1 *Let (u, v) be a stationary point of the following system*

$$\begin{aligned} u_{n+1} &= f(u_n, v_n) \\ v_{n+1} &= g(u_n, v_n) \end{aligned} \tag{4}$$

and J be the Jacobian of (u, v) , then

- (i) *If all the eigenvalues λ_i of J , satisfy $|\lambda_i| < 1$ for $i = 1, 2$, then (u, v) is stable.*
- (ii) *If all the eigenvalues λ_i of J , satisfy $|\lambda_i| > 1$ for $i = 1, 2$, then (u, v) is unstable.*

In the stationary state, U and V do not depend on time. Moreover, for analysis purposes, also the diffusion terms have been set to zero:

$$\frac{\partial u}{\partial t} = \frac{\partial v}{\partial t} = \Delta u = \Delta v = 0.$$

Hence, the system (3) becomes

$$\begin{aligned} -uv^2 + F(1 - u) &= 0 \\ uv^2 - (F + K)v &= 0. \end{aligned} \quad (5)$$

Solving the system, the point $(u, v) = (1, 0)$ is a solution to the equations. As long as $4(F + K)^2 < F$, we obtain two further solutions:

$$(u, v) = \left(\frac{1}{2} \left[1 \pm \sqrt{1 - \frac{4(F + K)^2}{F}} \right], \frac{F}{2(F + K)} \left[1 \mp \sqrt{1 - \frac{4(F + K)^2}{F}} \right] \right),$$

while in the case $4(F + K)^2 = F$ these points are equal to

$$(u, v) = \left(\frac{1}{2}, 2(F + K) \right).$$

To determine the stability the Jacobian matrix of the system is computed as:

$$J(u, v) = \begin{pmatrix} -v^2 - F & -2uv \\ v^2 & 2uv - (F + K) \end{pmatrix},$$

resulting in

$$J(u, v) = \begin{pmatrix} -F & 0 \\ 0 & -(F + K) \end{pmatrix}. \quad (6)$$

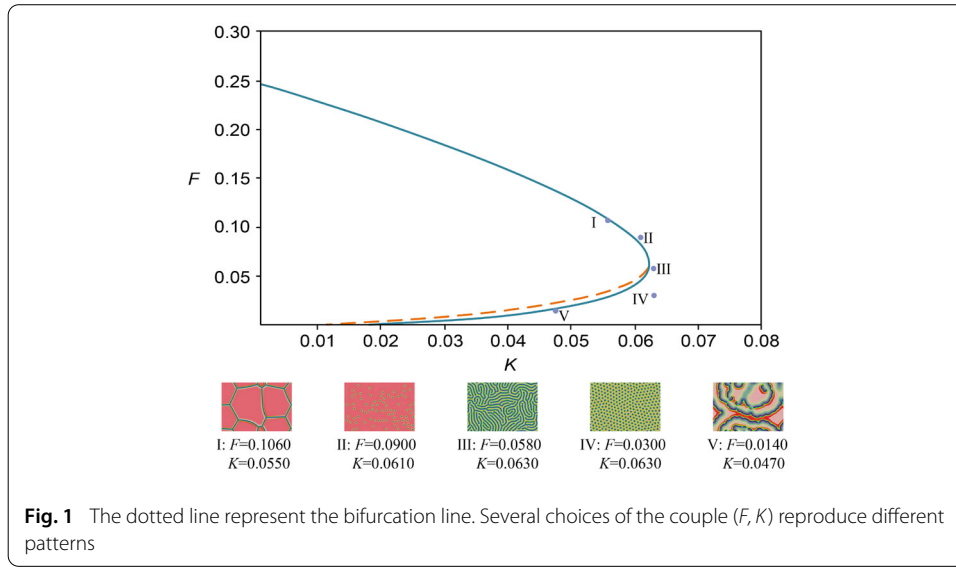
if evaluated in the point $(u, v) = (1, 0)$. Hence, for the Theorem 1, we observe that the point $(u, v) = (1, 0)$ is stable equilibrium since the Jacobian matrix presents two negative eigenvalues. As regards the other two stationary points, it can be observed that one of them presents always a stable manifold in one direction and an unstable one in the other, resulting in a saddle node. The other equilibrium exhibits a bifurcation line in the parameter space, in particular when $4(F + K)^2 = F$: these changes in the stability of fixed points are what create the patterns. We can observe in Fig. 1 how small fluctuations of parameters (F, K) lead to different patterns.

A physics-informed strategy

In the last few years, Neural Networks have been successfully adopted to solve nonlinear partial differential equations thanks to the introduction of a novel methodology, namely Physics-informed Neural Networks (PINNs). These are Artificial Intelligence approaches that takes into account physical constraints and *prior* information to deal with differential problems of the type:

$$\begin{aligned} \mathcal{F}(u(\mathbf{x}); \eta) &= f & \mathbf{x} \text{ in } \Omega, \\ \mathcal{B}(u(\mathbf{x})) &= g & \mathbf{x} \text{ in } \partial\Omega \end{aligned} \quad (7)$$

defined on a domain $\Omega \subset \mathbb{R}^d$ with boundary $\partial\Omega$. In Eq. 7, the function u represents the unknown solution, η are such parameters related to the physics, f and g are known functions; \mathcal{F} represent a non linear differential operator and \mathcal{B} denotes the initial and/or boundary conditions.



PINNs aim in training a Neural Network to become a surrogate model of the PDE equation, or system, such that, given a space-time vector $\mathbf{x} := [x_1, \dots, x_n; t]$ and the dynamic related parameter η , the approximate solution $\hat{u}_\theta(\mathbf{x})$ is close to the actual $u(\mathbf{x})$: in this sense, this methodology can be used to solve differential problems in a *forward* setting. Conversely, if data are available (e.g. provided by sensors or numerical simulations), the same framework can be applied in a *inverse* approach, providing a parameter estimation driven by the coherence with physics and data constraints.

The architecture of the network chosen as well impacts on the surrogate model created through a PINN approach: exploiting the characteristics of different types of neurons analytical or physical properties of the differential problem can be embedded directly in the neural structure [31]. Following the most common approach in literature, in this work we design a feed-forward neural network (FF-DNN), also known as a multi-layer perceptron (MLP), to approach to the problem (3) in a PINN framework. FF-DNN is an architecture in which neurons in adjacent layers are connected, and neurons inside a single layer are not linked. More specifically, a neuron can be seen as a computational unit that combine a function (usually non-linear), named activation function, with a weighted sum of the neuron’s inputs, plus a bias factor.

Defined the layer function as:

$$h_i(\mathbf{x}_{i-1}; W_i, b_i) = \sigma_i(W_i \mathbf{x}_{i-1} + b_i)$$

where σ_i is a scalar (non-linear) activation function, W_i and b_i are the parameters defining the layer i , i.e. the weights of the links between the layer $i - 1$ and the layer i and the corresponding biases, the output $\hat{u}_\theta(\mathbf{x})$ of a FF-DNN can be written as a composition of functions

$$\hat{u}_\theta(\mathbf{x}) = \Gamma_L \circ \sigma \circ \Gamma_{L-1} \dots \circ \sigma \circ \Gamma_1(\mathbf{x}), \tag{8}$$

where, for any $1 \leq k \leq L$, it is defined

$$\Gamma_k(\mathbf{x}_{k-1}) = W_k \mathbf{x}_{k-1} + b_k \tag{9}$$

considering a NN composed of an input layer and L hidden layers, the last of whom is the output one, and the same activation function σ for each unit.

In general, the introduction of a physical constraint into the training process can be done in several manners: as mentioned above, the NN architecture can be designed to implicitly satisfy some properties of the faced problem; data related to the underlying physics can be collected or carefully crafted by numerical simulations and used in a supervised shape to allow the model to learn functions, vector fields, and mathematical operators; a suitable cost function can be chosen, usually built in the form of residual loss with respect to the physical laws underlying the problem and so expressed in the form of integral, differential equations or even fractional ones [13], that guarantees the convergence towards solutions of the model. In our case, as better discussed in Sec. , the last two methodologies have been applied according to the problem addressed.

However, in the general framework of PINNs, solving a differential problem of the type (7) means learning how to approximate the dynamics finding θ^* , the optimal NN parameters vector, by minimizing a loss function dependent on the differential equation $\mathcal{L}_{\mathcal{F}}$, the initial/boundary conditions $\mathcal{L}_{\mathcal{B}}$, and, if present, the known data \mathcal{L}_{data} , each of them adequately weighted:

$$\theta^* = \arg \min_{\theta} (\omega_{\mathcal{F}} \mathcal{L}_{\mathcal{F}}(\theta) + \omega_{\mathcal{B}} \mathcal{L}_{\mathcal{B}}(\theta) + \omega_{data} \mathcal{L}_{data}(\theta)).$$

It is worth underlining that, as regards the experiments performed in the following sections of this paper, especially in the case of 2D Gray-Scott systems, we set up a strategy which consists in adding a temporal collection of known data, i.e. solution samples computed through numerical methods, which is one of the core ideas of our work. In the dynamical evolution described by the Gray-Scott system, the complete morphology of associated Turing patterns is observed over a long period of time. However, too long “simulation times” may cause the system to evolve towards a local optimum: merely providing residual loss with initial/boundary conditions is not sufficient to guarantee the convergence to the actual solution of the problem and the rise of expected Turing patterns. If the predicted solutions of the system are not corrected in time using these known data, the final GS system may exhibit a state in which the reactants dissipate or uniformly cover the entire observation domain due to local optimal evolution. To solve this problem, we set up a time set containing n time points t_1, t_2, \dots, t_n . At each time point, we numerically obtain the correct value for the unknown solution, and then apply those known data to the surrogate model in the form of a loss function. It is worth noticing that known data from numerical methods inherently contain the physical information of the Gray-Scott model. So, in this case, \mathcal{L}_{data} can be written as:

$$\mathcal{L}_{data} = \sum_{i=1}^n \mathcal{L}_{data}(t_i)$$

where t_i is the time instance in which the known data are considered and the dependence on θ will be omitted now on for brevity. A schematic diagram the aforementioned procedure is shown in Fig. 2.

Experimental results

In this section, the performances and accuracy of the PINN approach are presented and discussed. In particular, one-dimensional and two-dimensional Gray-Scott systems are addressed. As regards the one-dimensional formulation (Sect.), the presented results, in both the cases considered, have been achieved by using PINNs with only boundary and

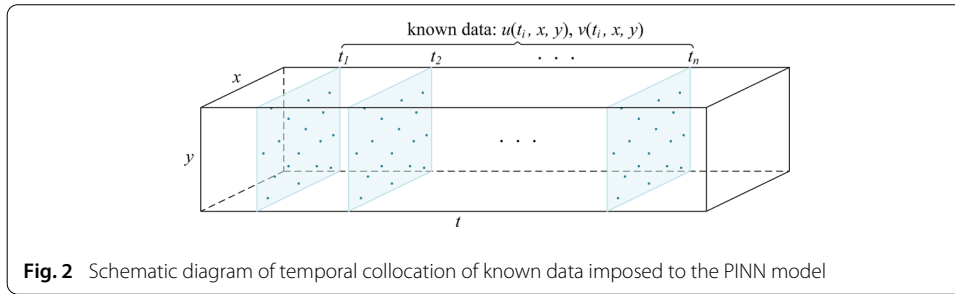


Fig. 2 Schematic diagram of temporal collocation of known data imposed to the PINN model

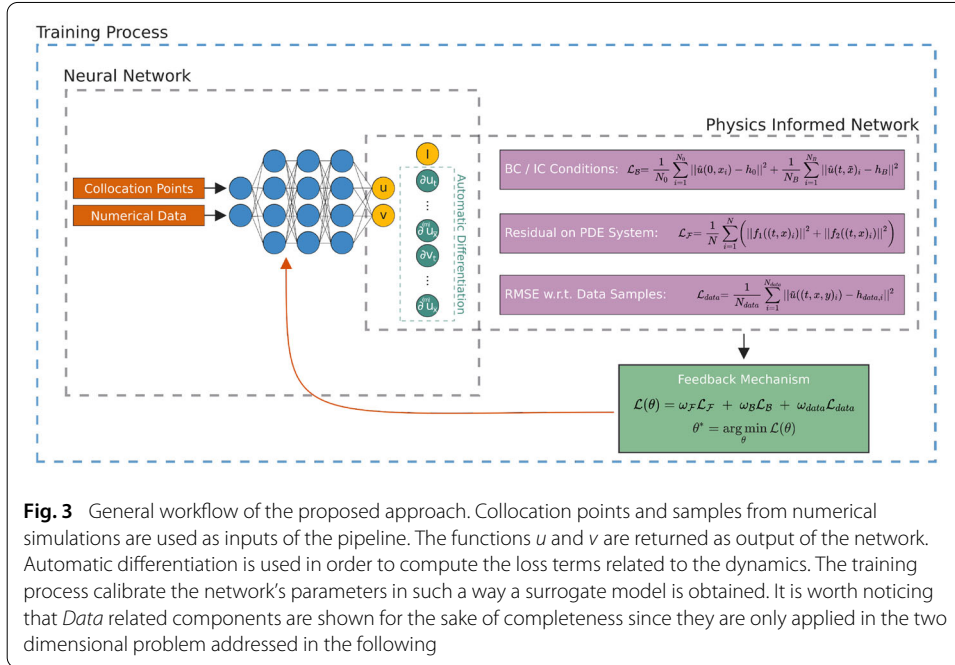


Fig. 3 General workflow of the proposed approach. Collocation points and samples from numerical simulations are used as inputs of the pipeline. The functions u and v are returned as output of the network. Automatic differentiation is used in order to compute the loss terms related to the dynamics. The training process calibrates the network's parameters in such a way a surrogate model is obtained. It is worth noticing that *Data* related components are shown for the sake of completeness since they are only applied in the two dimensional problem addressed in the following

dynamic loss functions, i.e. without the loss component related to known data. In two-dimensional experiments (Sect.), we designed a suitable strategy by adding known data, generated through a numerical solver, to the loss functions of the PINN as additional constraint (Fig. 3).

1D Gray-Scott system

In the one-dimensional case, for testing the reliability of the PINNs, have been compared the predicted solution for the Gray-Scott system with the exact solution in the following (Case 1), and with numerical results with those obtained by the MATLAB solver (Case 2). Both the experiments assess the good accuracy of our approach when dealing with 1D Gray-Scott systems. Due to the slow convergence dictated by the PINNs, to have a good performance by minimizing the loss function, the duration of the training process has been set as 50, 000 epochs with 1000 epochs for the patience. In this way, if the best solution has already been achieved, the iteration stops; conversely, the algorithm continues its training until it reaches a better solution. The execution times in training is about 5 minutes on a GPU NVIDIA GeForce RTX 3080 with CPU intel core i9-9900k and 128 GB of RAM. In these cases, the loss functions is composed by two terms:

$$\mathcal{L} = \omega_B \mathcal{L}_B + \omega_F \mathcal{L}_F \tag{10}$$

where $\mathcal{L}_{\mathcal{F}}$ represents the loss component related to the dynamics and $\mathcal{L}_{\mathcal{B}}$ represents the loss component related to the initial/boundary conditions. The two terms can be written as follows:

$$\mathcal{L}_{\mathcal{B}} = \frac{1}{N_0} \sum_{i=1}^{N_0} \|\hat{u}(0, x_i) - h_0\|^2 + \frac{1}{N_B} \sum_{i=1}^{N_B} \|\hat{u}(t, \bar{x})_i - h_B\|^2, \quad (11)$$

$$\mathcal{L}_{\mathcal{F}} = \frac{1}{N} \sum_{i=1}^N \left(\|f_1((t, x)_i)\|^2 + \|f_2((t, x)_i)\|^2 \right), \quad (12)$$

where h_0 is the known initial condition, h_B is the known boundary condition, N_0 is the cardinality of the set $\{(t, x)_i | t = 0\}$ and N_B is the cardinality of the set $\{(t, \bar{x})_i\}$, in which \bar{x} represents the space points belonging to the boundary of the domain. Moreover the functions f_1 and f_2 are the residual with respect to the differential equations and they are defined as follows:

$$f_1 = \frac{\partial u}{\partial t} + uv^2 - F(1 - u) - \frac{\partial^2 u}{\partial x^2}$$

$$f_2 = \frac{\partial v}{\partial t} - uv^2 + (F + K)v - \frac{\partial^2 v}{\partial x^2}.$$

The accuracy of the trained method is assessed through the root mean square error (*RMSE*) of the exact value $u(t, x)_i$ and the predicted value $\hat{u}(t, x)_i$ inferred by the network.

$$RMSE = \sqrt{\frac{\sum_{i=1}^N |\hat{u}(t, x)_i - u(t, x)_i|^2}{N}}. \quad (13)$$

Case 1: The first case is represented by the one-dimensional Gray-Scott model discussed in [32]. Under the assumption $4(F + K)^2 < F$, we set $z = x - \beta t$ and (3) can be rewritten in terms of z , obtaining:

$$0 = \beta \frac{du}{dz} + uv^2 - F(1 - u) - \frac{d^2 u}{dz^2}$$

$$0 = \beta \frac{dv}{dz} - uv^2 + (F + K)v - \frac{d^2 v}{dz^2}, \quad (14)$$

where

$$\beta = \frac{\sqrt{2}}{4}(1 - 3\sqrt{\eta}), \quad \eta = 1 - 4F.$$

For this case, the boundary conditions are given as:

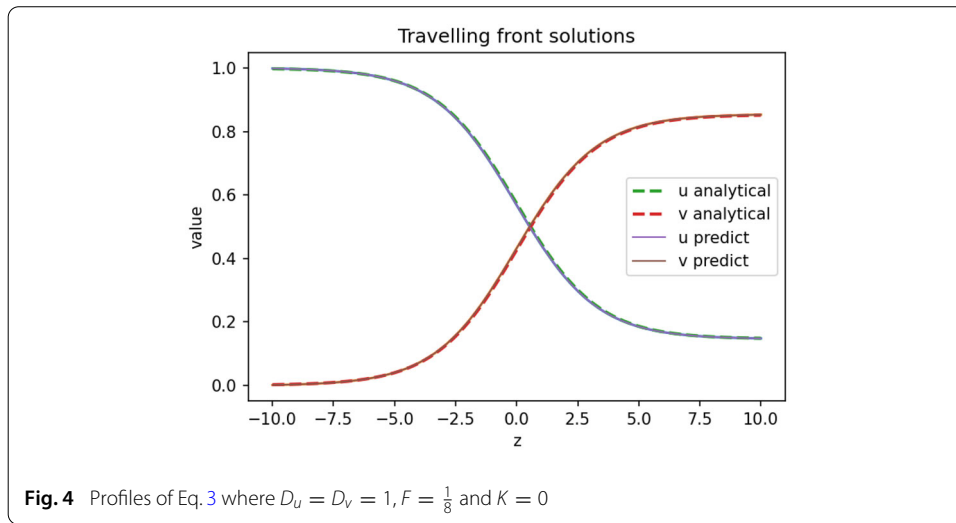
$$(u, v)(-10) = (1, 0) \quad (u, v)(10) = \left(\frac{2 - \sqrt{2}}{4}, \frac{2 + \sqrt{2}}{4} \right).$$

In this settings, some exact solutions to this problem can be provided as in the following:

$$u(z) = \frac{3 - \sqrt{\eta}}{4} - \frac{\sqrt{2\xi}}{4} \tanh\left(\frac{\sqrt{\xi}}{4}z\right)$$

$$v(z) = \frac{1 + \sqrt{\eta}}{4} + \frac{\sqrt{2\xi}}{4} \tanh\left(\frac{\sqrt{\xi}}{4}z\right),$$

where $\xi = 1 + \sqrt{\eta} - 2F$. The neural network architecture used for the PINN approach, in this case, consists of four hidden layers with 10 neurons in the first two layers and 20 neurons in the other ones. A sigmoid activation function for all the neurons and the Adam optimizer with a learning rate of 10^{-2} have been applied. As we can observe in Fig. 4, the predicted solutions, obtained on 600 domain points, overlaps the analytical one assessing



the PINNs reliability for this particular case of the one-dimensional Gray-Scott model. The *RMSE* between the predicted values of PINNs and exact solutions are as follows: $RMES_u = 1.7982 \times 10^{-3}, RMES_v = 1.7286 \times 10^{-3}$, where $N = 21$ calculation points are selected.

Case 2: As a second case study, the parameters $D_u = D_v = 0.01, F = 0.09, K = -0.004$ are considered for the system (3). This case is also discussed in [19]. Here, the neural network architecture consists of four hidden layers with 20 neurons in each layer. The hyperbolic tangent activation function and Adam optimizer with a learning rate of 10^{-3} has been applied. The number of the points selected to train the model is $N = N_0 + N_b + N_r$, where $N_0 = 100$ are for the initial conditions, $N_b = 200$ are for boundary conditions and $N_r = 1024$ are collocation points. The numerical solutions obtained with the proposed approach are shown in Fig. 5a, b. In Fig. 5c, d the numerical solutions for the functions u and v obtained through a Galerkin method are presented as benchmark. The initial and boundary conditions are:

$$\begin{aligned}
 u(x, 0) &= u_s + 0.01 \sin\left(\frac{\pi x}{L}\right) & v(x, 0) &= v_s - 0.12 \sin\left(\frac{\pi x}{L}\right) \\
 u(0, t) = u(L, t) &= u_s & v(0, t) = v(L, t) &= v_s
 \end{aligned}$$

respectively, where $u_s = 1, v_s = 0, L = 2$.

In this case, the predictions of the PINN are compared with those obtained by the Galerkin method. Table 1 reports the *RMSE* obtained by matching the number of collocation points in the PINN (N_P) with the number of grid points in the Galerkin method (N_G). We set the same number of points for the two methodologies: $N_P = N_G = 176, 651, 15251$ respectively. As it can be observed, the *RMSE* are low in all the three settings, assessing that PINN approach provide comparable performance to Galerkin method. Table 2 reports the prediction errors for different number of collocation points on which the PINNs is trained compared to the high-precision Galerkin method solution (i.e. fixing $N_G = 15251$). As it can be observed, the accuracy of PINNs improves as the number of collocation points grows.

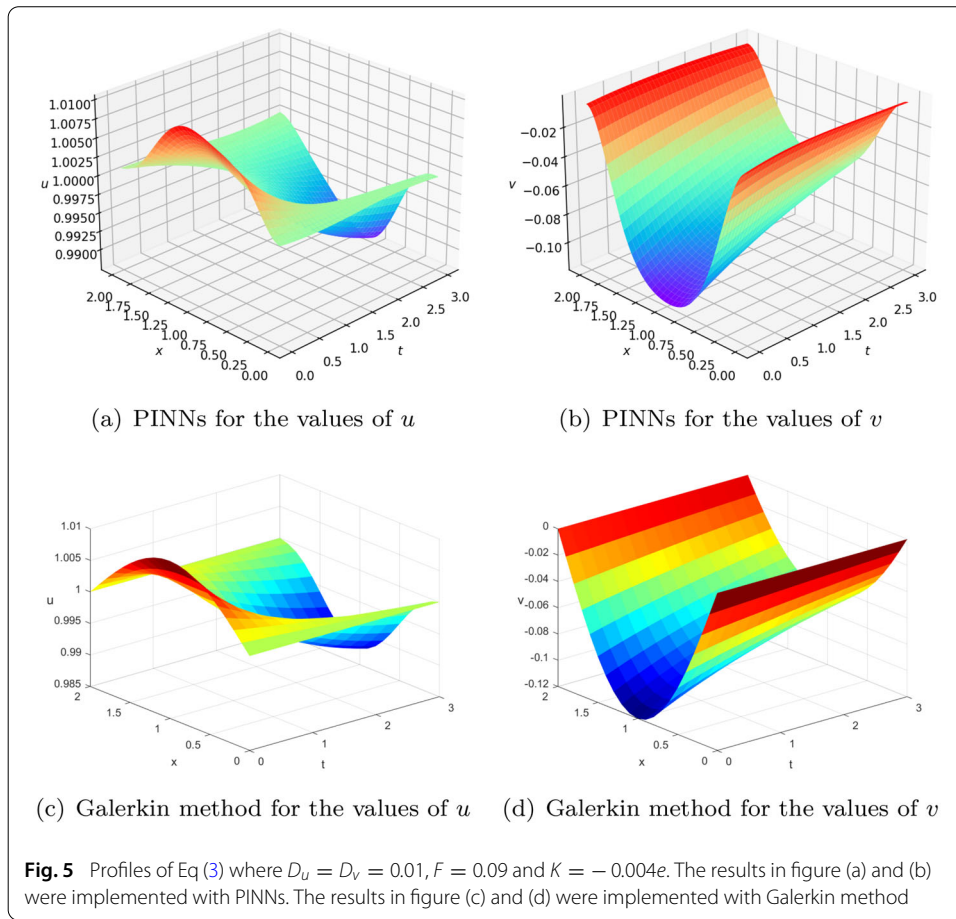


Table 1 The RMES for Case 2 by making $N_p = N_G$

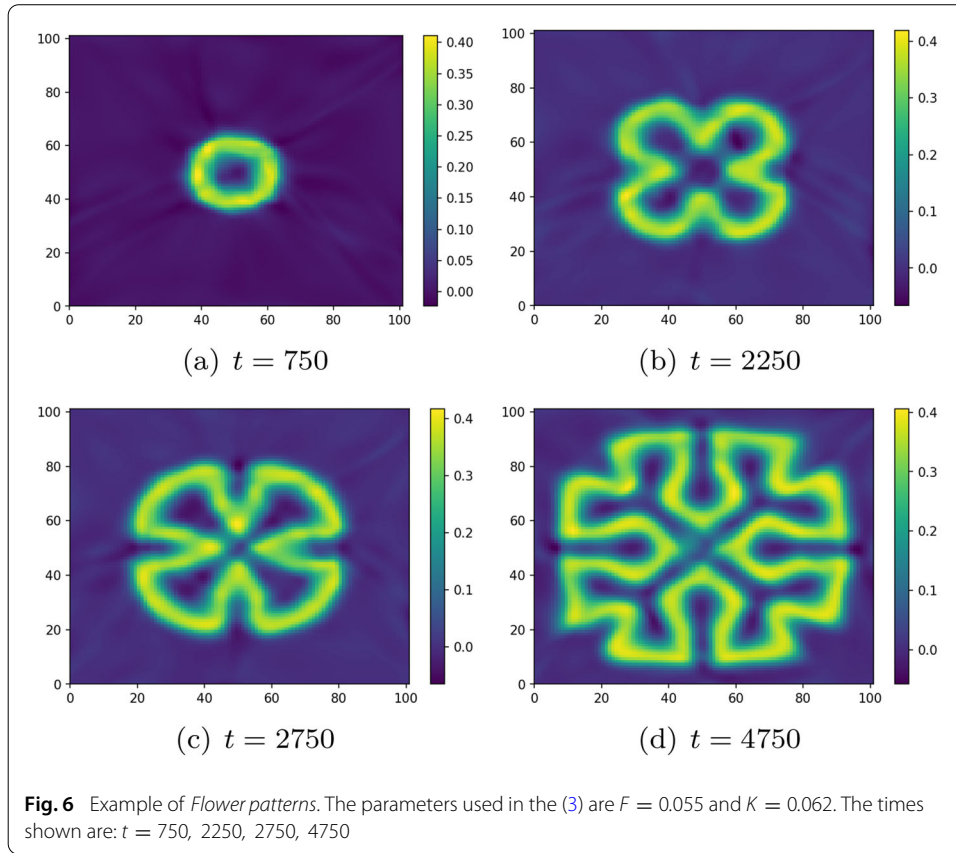
$N_p = N_G$	$RMSE_u$	$RMSE_v$
176	8.4760×10^{-4}	4.3799×10^{-3}
651	6.1477×10^{-4}	4.1715×10^{-3}
15251	5.8099×10^{-4}	3.8161×10^{-3}

Table 2 The RMSE of Case 2 by increasing N_p and fixing $N_G = 15251$

N_p	$RMSE_u$	$RMSE_v$
24	4.6145×10^{-3}	5.7044×10^{-2}
63	1.8322×10^{-3}	1.0239×10^{-2}
117	1.3297×10^{-3}	4.9414×10^{-3}
176	9.6979×10^{-4}	4.2570×10^{-3}
651	7.9761×10^{-4}	4.1751×10^{-3}
15251	5.8099×10^{-4}	3.8161×10^{-3}

2D Gray-Scott system

In this subsection, the two-dimensional Gray-Scott problem is addressed. The neural network architecture designed in the case of this study consists of four hidden layers with 20 neurons in each layer, a hyperbolic tangent activation function for the input, and a sigmoid activation function for the output. The set of collocation points is generated

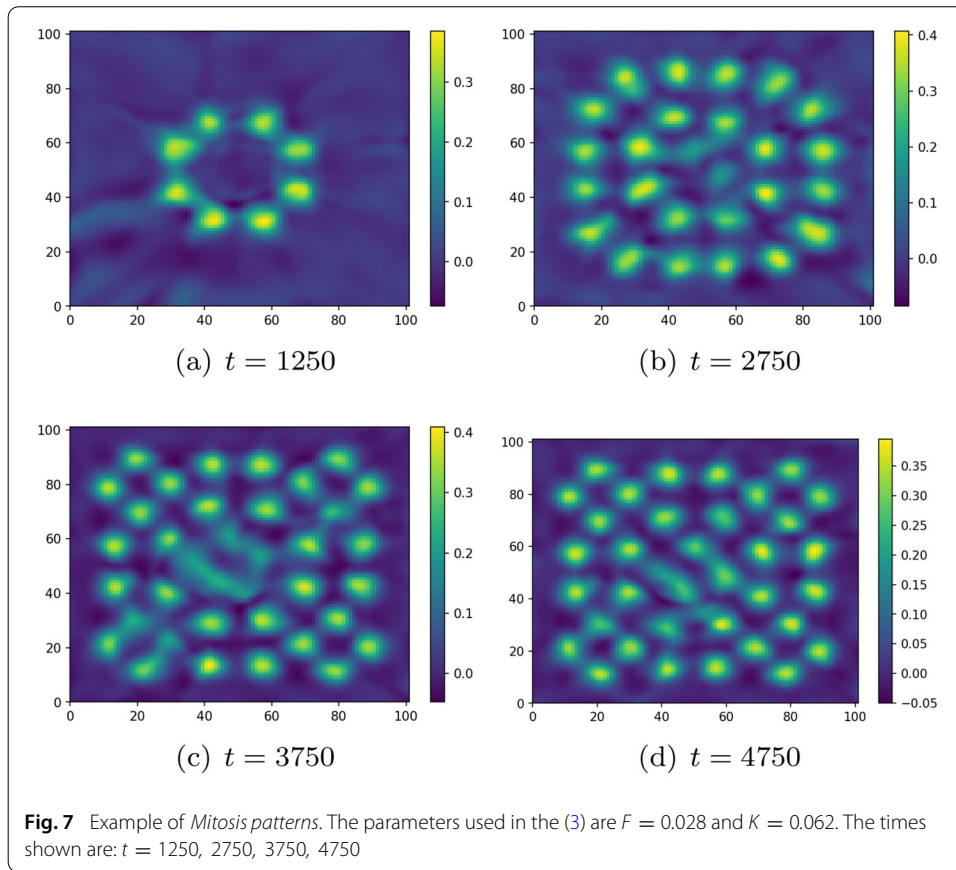


uniformly sampling the time-space domain in $N_r = 10,000$ points; the initial condition is provided to the model on a grid of $N_0 = 101 \times 101$ points; boundary conditions are imposed on $N_B = 4 \times 100$ points randomly sampled on the space-time surfaces related to the boundary of the domain, 100 points for side; finally, in $N_{data} = 10 \times 101 \times 101$ points, namely on a grid 101×101 in 10 time steps, the numerical solution is provided to the PINN model for the supervised component of the loss. To minimize the loss function the network has been run for 50,000 epochs using Adam optimizer [14] and a learning rate of 10^{-3} . The execution times in training is about 15 minutes on a GPU NVIDIA GeForce RTX 3080 with CPU intelcore i9-9900k and 128 GB of RAM. As regards the loss function, in the case under consideration it consists of three terms: the first one is related to the initial conditions and to the boundary conditions, the second one relates to the dynamics of the system, and the third computes the error with respect to known data points:

$$\mathcal{L} = \omega_B \mathcal{L}_B + \omega_{\mathcal{F}} \mathcal{L}_{\mathcal{F}} + \omega_{data} \mathcal{L}_{data}.$$

In particular, each of the aforementioned components can be written as:

$$\begin{aligned} \mathcal{L}_B &= \frac{1}{N_0} \sum_{i=1}^{N_0} \|\hat{u}(t, x, y)|_{t=0} - h_{0,i}\|^2 + \frac{1}{N_b} \sum_{i=1}^{N_b} \|\hat{u}(t, x, y)|_{x \in \partial\Omega, y \in \partial\Omega} - h_{B,i}\|^2, \\ \mathcal{L}_{\mathcal{F}} &= \frac{1}{N_f} \sum_{i=1}^{N_f} \left(\|f_1((t, x, y)_i)\|^2 + \|f_2((t, x, y)_i)\|^2 \right), \\ \mathcal{L}_{data} &= \frac{1}{N_{data}} \sum_{i=1}^{N_{data}} \|\tilde{u}((t, x, y)_i) - h_{data,i}\|^2, \end{aligned}$$



where, in each equation, the collocation point $(t, x, y)_i$ refers to the $i - th$ point of the set whose cardinality is N_0, N_B, N_{data} , respectively. While, the functions h_0^i, h_B^i and h_{data}^i are the initial, boundary and numerical constraints on each point $(t, x, y)_i$ considered. To provide the known data needed for the computation of \mathcal{L}_{data} , a finite difference method (FDM) second order accurate in space and first order accurate in time is used. The functions f_1 and f_2 are defined as follows:

$$\begin{aligned}
 f_1 &= \frac{\partial u}{\partial t} + uv^2 - F(1 - u) - D_u \Delta u \\
 f_2 &= \frac{\partial v}{\partial t} - uv^2 + (F + K)v - D_v \Delta v.
 \end{aligned}
 \tag{15}$$

In the following, experiments on different configurations of the Gray-Scott model’s parameters are shown, inspired to the work [27]. In particular, differences in parameters F, K and diffusion coefficients D_u and D_v , lead to different Turing patterns. For the different cases, we set the same space-time domain and initial/boundary conditions, i.e. the entire system was placed in the trivial state ($u = 1, v = 0$) at $t = 0$, with a small squared area located symmetrically about the center of the fields perturbed to $(u = \frac{1}{2}, v = \frac{1}{4})$. Periodic boundary conditions are imposed on the domain edges. Assuming constant $D_u = 2D_v = 2 \times 10^{-5}$ in all the following cases, F and K have been set in such a way that patterns well known in literature are produced in the temporal evolution of the system. The data for the supervised component of the loss function, i.e. \mathcal{L}_{data} , are provided as discussed before, and particularly in the time instances $t \in \{500, 1000, 1500, 2000, 2500, 3000, 3500, 4000, 4500, 5000\}$. The results obtained are the following (Figs. 6, 7, 8, 9):

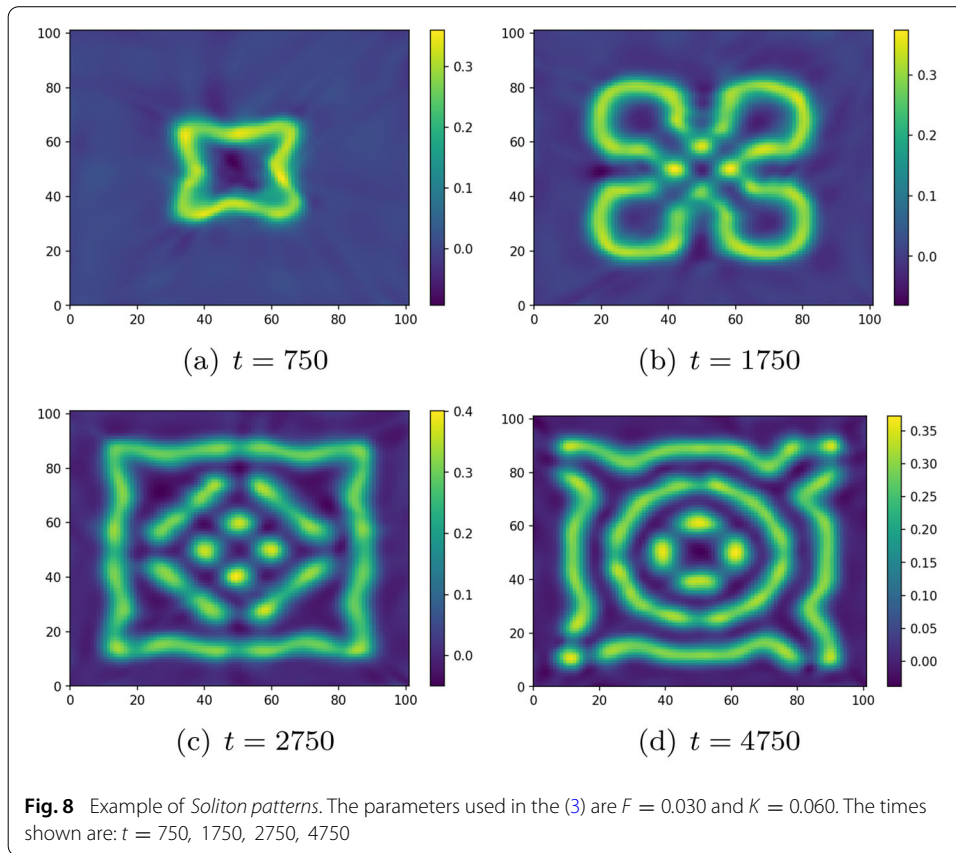
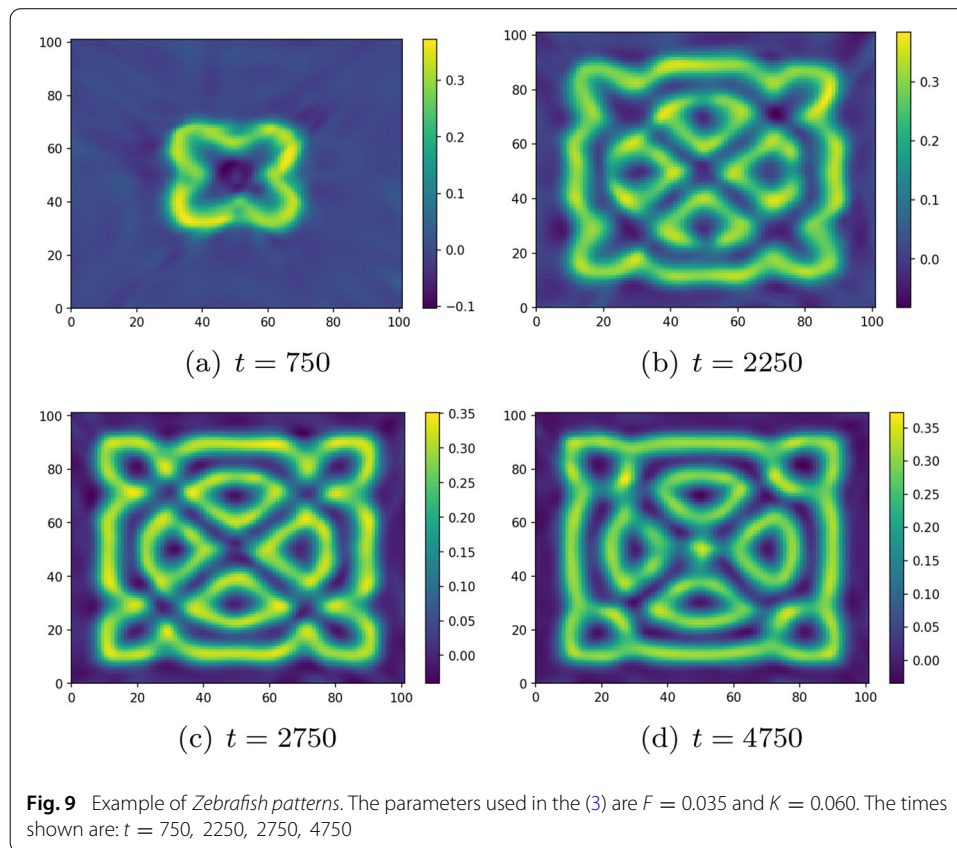


Table 3 The *MAE* and *RMSE* for the four experiments. Errors, in terms of *RMSE* and *MAE*, of the PINN provided solution with respect to numerical solution of bidimensional Gray-Scott system. Experiments are conducted for different configurations of the parameter couple (F,K) which give four types of Turing patterns during the temporal evolution

<i>PATTERNS</i>	<i>MAE_u</i>	<i>MAE_v</i>	<i>RMSE_u</i>	<i>RMSE_v</i>
<i>FLOWER</i>	0.22	0.10	0.30	0.16
<i>MITOSIS</i>	0.19	0.07	0.26	0.11
<i>SOLITON</i>	0.22	0.09	0.29	0.13
<i>ZEBRAFISH</i>	0.23	0.10	0.29	0.14

In Table 3 the errors, in terms of *RMSE* with respect to the functions u and v , are reported in the cases different patterns (for several values of the parameters F and K) are exhibited in the temporal evolution of the Gray-Scott system. It is worth recalling that, since in this case no analytical solution is providable, the approximated solutions given by the PINN approach have been compared with numerical simulations of the system. As can be noticed, magnitude of errors remain constant despite very different spatio-temporal configurations can create. This suggest that approximation errors due to the methodology remains stable. More in detail, the learning process or the expressiveness of the network which models the differential problem, guarantee the applicability of the proposed approach on a wide range of problems modelled by Gray-Scott system.



Conclusions and discussions

In this paper, a physics-informed neural networks methodology for solving Gray-Scott systems has been proposed. Such problem has been addressed for 1D and 2D settings respectively. In the first case, we compare the approximated solutions obtained through the PINN approach with the exact one, if it exists, and also with solutions provided by numerical solvers. In 2D numerical experiments, by considering four (F, K) problem configurations. The proposed approach have proven to be able to provide approximated solutions that mimic the typical patterns exhibited by Gray-Scott models in the instability regions of the parameter space. The comparison results well assess the performance of the PINN solver for Gray-Scott systems. To the best of our knowledge, this work represents the first attempt about the design of a Physics-Informed Neural Network for the above discussed topic. The main issue to be addressed during the design of a Physics-Informed Neural Network is to find a suitable learning approach that can produce an approximation of the system's solution. In this context, if the loss function is not properly designed, the neural network fitting model could disregard the mutual dependence of u and v in the system equations. In particular, as regards the 2D experiments, the PINN approach tends to converge towards local optima of the loss function, namely the constant solution $(u, v) = (1, 0)$ rather than converging to a sharp solution of the problem then related to the Turing patterns. To overcome this issue, in this paper we have proposed an approach that takes into account a-priori knowledge on the system in order to force the PINN to convergence to the actual solution of the system according to the values of the parameters F and K . In particular, we have re-designed the data loss to take into account some snapshots of

the provided numerical solution; in this way, reliable results have been obtained also in the case two spatial dimensions are considered. It is important to mention that there are still some issues to be addressed as future research directions: for example, solving a 3D Gray-Scott system by using PINNs represents a challenging task that surely will be addressed in future works. All these considerations pave the way to further studies on the optimization of possible constraints in the residual PDEs to correct misbehaviors of the network and also to save computational costs. In such a way a reliable methodology based on Physics Informed Neural Networks can be provided as regards the general, and interesting, framework of Gray-Scott systems.

Acknowledgements

SC is a member of GNCS-INdAM, UMI-TAA and UMI-AI Italian research groups.

Authors' contributions

Every co-author participated to derive the methodology, the algorithm and the results, and wrote the manuscript. Every co-author participated to fruitful discussions. All authors read and approved the final manuscript.

Funding

The paper was financially supported by the DAMM project (ex-numeracy), University of Naples Federico II, Department of Mathematics and Applications "R. Caccioppoli", Via Cinthia, 80126, Naples, Italy.

Declarations

Competing interests

The authors declare that they have no competing interests.

Received: 25 March 2022 Accepted: 26 April 2022

Published online: 25 May 2022

References

1. Adamatzky A. Generative complexity of gray-scott model. *Comm Nonlinear Sci Numer Simul.* 2018;56:457–66.
2. Barrio R, Varea C, Aragón J, Maini P. A two-dimensional numerical study of spatial pattern formation in interacting Turing systems. *Bull Math Biol.* 1999;61(3):483–505.
3. H. Carslaw, J. Jaeger. *Conduction of heat in solids.* New York: Oxford. 1959.
4. Chen W, Ward MJ. The stability and dynamics of localized spot patterns in the two-dimensional gray-scott model. *SIAM J Appl Dyn Syst.* 2011;10(2):582–666.
5. Crank J. *The mathematics of diffusion.* Oxford: Oxford University Press; 1979.
6. Cuomo S, Di Cola VS, Giampaolo F, Rozza G, Raissi M, Piccialli F. Scientific machine learning through physics-informed neural networks: Where we are and what's next. *arXiv preprint arXiv:2201.05624*, 2022.
7. Doelman A, Kaper TJ, Zegeling PA. Pattern formation in the one-dimensional gray-scott model. *Nonlinearity.* 1997;10(2):523.
8. Gray P, Scott SK. Autocatalytic reactions in the isothermal, continuous stirred tank reactor: Oscillations and instabilities in the system $a + 2b \rightarrow 3b$; $b + c \rightarrow c$. *Chem Eng Sci.* 1984;39(6):1087–97.
9. Har-Shemesh O, Quax R, Hoekstra AG, Sloot PM. Information geometric analysis of phase transitions in complex patterns: the case of the gray-scott reaction-diffusion model. *J Stat Mech.* 2016;2016(4):043301.
10. Hasnain S, Bashir S, Linker P, Saqib M. Efficiency of numerical schemes for two dimensional gray scott model. *AIP Advan.* 2019;9(10):105023.
11. Jackson JD. *Classical electrodynamics*, 1999.
12. Jagtap AD, Kharazmi E, Karniadakis GE. Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. *Computer Methods Appl Mech Eng.* 2020;365:113028.
13. Karniadakis GE, Kevrekidis IG, Lu L, Perdikaris P, Wang S, Yang L. Physics-informed machine learning. *Nature Reviews. Physics.* 2021;3(6):422–40.
14. Kingma DP, Ba J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
15. Kolokolnikov T, Ward MJ, Wei J. Zigzag and breakup instabilities of stripes and rings in the two-dimensional gray-scott model. *Stud Appl Math.* 2006;116(1):35–95.
16. Korkmaz A, Ersoy O, Dag I. Motion of patterns modeled by the gray-scott autocatalysis system in one dimension. *arXiv preprint arXiv:1605.09712*, 2016.
17. Lin JC-W, Djenouri Y, Srivastava G. Efficient closed high-utility pattern fusion model in large-scale databases. *Inform Fusion.* 2021;76:122–32.
18. Lin JC-W, Djenouri Y, Srivastava G, Yun U, Fournier-Viger P. A predictive ga-based model for closed high-utility itemset mining. *Applied Soft Computing.* 2021;108: 107422.
19. Manaa SA, Rasheed J. Successive and finite difference method for gray scott model. *Sci J Univer Zakh.* 2013;1(2):862–73.
20. Mazin W, Rasmussen K, Mosekilde E, Borckmans P, Dewel G. Pattern formation in the bistable gray-scott model. *Math Computers Simul.* 1996;40(3–4):371–96.

21. McGough JS, Riley K. Pattern formation in the gray-scott model. *Nonlinear Anal.* 2004;5(1):105–21.
22. Muratov CB, Osipov V. Stability of the static spike autosolitons in the gray-scott model. *SIAM J Appl Math.* 2002;62(5):1463–87.
23. Nascimento RG, Fricke K, Viana FA. A tutorial on solving ordinary differential equations using python and hybrid physics-informed neural network. *Eng Appl Artif Intell.* 2020;96:103996.
24. Nishiura Y, Ueyama D. Spatio-temporal chaos for the gray-scott model. *Physica D Nonlinear Phenomena.* 2001;150(3–4):137–62.
25. Owolabi KM, Patidar KC. Numerical solution of singular patterns in one-dimensional gray-scott-like models. *Int J Nonlinear Sci Numer Simul.* 2014;15(7–8):437–62.
26. Pan S, Duraisamy K. Physics-informed probabilistic learning of linear embeddings of nonlinear dynamics with guaranteed stability. *SIAM J Appl Dyn Syst.* 2020;19(1):480–509.
27. Pearson JE. Complex patterns in a simple system. *Science.* 1993;261(5118):189–92.
28. Raei M, Cuomo S, Colecchia G, Severino G. Solving 3-d gray-scott systems with variable diffusion coefficients on surfaces by closest point method with rbf-fd. *Mathematics.* 2021;9(9):924.
29. Raissi M, Perdikaris P, Karniadakis GE. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10561.* 2017.
30. Raissi M, Perdikaris P, Karniadakis GE. Physics informed deep learning (part ii): Data-driven discovery of nonlinear partial differential equations. 2017.
31. Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J Comput Phys.* 2019;378:686–707.
32. Rodrigo M, Mimura M. Exact solutions of reaction-diffusion systems and nonlinear wave equations. *Japan J Ind Appl Math.* 2001;18(3):657–96.
33. Shao Y, Lin JC-W, Srivastava G, Guo D, Zhang H, Yi H, Jolfaei A. Multi-objective neural evolutionary algorithm for combinatorial optimization problems. *IEEE Transactions on Neural Networks and Learning Systems,* 2021.
34. Tartakovsky AM, Marrero CO, Perdikaris P, Tartakovsky GD, Barajas-Solano D. Physics-informed deep neural networks for learning parameters and constitutive relationships in subsurface flow problems. *Water Resources Research.* 2020;56(5):e2019WR026731.
35. Turing AM. The chemical basis of morphogenesis. *Bull Math Biol.* 1990;52(1):153–97.
36. Yadav OP, Jiwari R. A finite element approach for analysis and computational modelling of coupled reaction diffusion models. *Numer Methods Partial Differ Equ.* 2019;35(2):830–50.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)
