Estimation of the Saving Rate Function in the Spatial Solow-Swan Model by PINNs

Gabriel Schmökel[1], João P. Juchem Neto[2], Pedro H. A. Konzen[3]

[1]Instituto de Matemática e Estatística, Universidade Federal do Rio Grande do Sul, gabriel.schmokel@ufrgs.br
[2]Departamento de Economia e Relações Internacionais, Universidade Federal do Rio Grande do Sul, plinio.juchem@ufrgs.br
[3]Instituto de Matemática e Estatística, Universidade Federal do Rio Grande do Sul, pedro.konzen@ufrgs.br

Abstract. This study investigates the determination of the savings rate in the spatial Solow-Swan model using Physics-Informed Neural Networks (PINNs). The model, formulated as a reactive-diffusive partial differential equation, describes the evolution of capital density in a spatially distributed economy. We propose a PINN-based method to solve the inverse problem and estimate the savings rate from simulated data, employing two neural networks: one to approximate the solution of the equation and another to infer the savings rate. The results demonstrate low loss and reduced mean squared error, highlighting the viability of this approach for the economic model.

Keywords. Partial Differential Equations; Solow-Swan Spatial Model; Inverse Problem; PINNs (Physics-Informed Neural Networks); Savings Rate Function.

1. INTRODUCTION

The Solow-Swan model, developed in 1956 [8], [9], has been a reference for understanding economic growth, but it does not account for the spatial distribution of economic activities, limiting its applicability. To overcome this limitation, spatial dimensions were incorporated into the model [1], [4], enabling the study of the impact of location on growth. Recently, researchers have been exploring inverse problems, such as determining the convex-concave production function in the spatial model [2], [3], but to our knowledge, no one has investigated the determination of the savings rate function.

The machine learning technique known as Physics-Informed Neural Networks (PINNs) has been used to solve Partial Differential Equations (PDEs) [6], [7]. By incorporating the residual of the model PDEs, boundary conditions, and initial conditions into the loss function formulation, this approach significantly reduces the need for large amounts of data. Furthermore, the technique has proven to be highly effective in solving inverse problems, even with a limited data set [3].

In this context, this work seeks to estimate the savings rate function, dependent only on the spatial coordinate, by solving the inverse problem in the spatial Solow-Swan model using PINNs. We consider a model that describes the evolution of capital density in local economies distributed over the compact interval $\Omega = [0, l]$, with $0 < l < \infty$. At each point $x \in \Omega$, there exists a capital density $K(t, x) \geq 0$ and labor density $L(x) \geq 0$, which are used to produce an aggregate good through a Cobb-Douglas production function

$$f(K, L) = A(x)[K^{\phi}(t, x)L(x)^{1-\phi}]. \tag{1}$$

Here, $A(x) > 0$ represents the technological factor and $\phi \in (0,1)$ indicates the intensity of capital in production. The labor distribution is given by the exogenous function $L(x) \geq 0$. Thus, the evolution of the capital stock in the economy is governed by the following reactive-diffusive partial differential equation, with the corresponding initial and boundary conditions

$$K_t = s(x)A(x)\left[K^{\phi}L(x)^{1-\phi}\right] - \delta(x)K + d(x)K_{xx},\ x \in (0,l),\ t > 0, \tag{2a}$$

$$K(t,x) = K_0(x),\ x \in \Omega = [0,l],\ t = 0, \tag{2b}$$

$$K_x = 0,\ x \in \partial\Omega = \{0,l\},\ t > 0. \tag{2c}$$

where $s(x) \in (0,1)$ is the savings rate function we aim to estimate, $\delta(x) \in (0,1)$ is the capital depreciation rate, and $d(x) > 0$ is the capital diffusion coefficient, which represents the intensity of capital movement towards regions with lower available capital. Completing the model, the initial capital distribution, $K_0(x) \geq 0$, is given by Equation (2b), while the homogeneous Neumann boundary conditions, Equation (2c), ensure no transfer of capital and labor at the boundaries $\partial\Omega$, making the economy autarkic.

Next, we present the details of the proposed PINNs approach aimed at solving the inverse problem of estimating the savings rate function for the presented spatial Solow-Swan model. We then discuss the results of a test case and provide the final remarks.

2. METHODOLOGY

The inverse problem consists of estimating the savings rate function $s(x)$ from the solution of the problem (2) using PINNs with given data $\mathcal{D} = \left\{\left(\hat{t}_i, \hat{x}_i; \hat{K}_i\right)\right\}_{i=1}^{n_d}$, $n_d \geq 1$. The method involves training two neural networks: one to approximate the solution of the PDE problem (2) and another to estimate the savings rate function. Multilayer Perceptron (MLP) networks are employed for this purpose. The first network has inputs (t,x) and outputs the estimated capital density $\tilde{K}(t,x)$, i.e.

$$\tilde{K} = \mathcal{N}_K(t,x;\theta_K), \tag{3}$$

where θ_K denotes the network parameters (weights and biases). The second network has input x and outputs the estimated savings rate function $\tilde{s}(x)$, i.e.

$$\tilde{s}(x) = \mathcal{N}_s(x;\theta_s), \tag{4}$$

with network parameters θ_s.

The network architectures (number of hidden layers and neurons per layer) are chosen based on an exploratory study. The training of both networks is performed simultaneously using the Adam optimizer [5] with a learning rate of 10^{-3} to solve

$$\min_{\theta_K,\theta_s} \mathcal{L}(\theta_K,\theta_s), \tag{5}$$

where $\mathcal{L}(\theta_K, \theta_s)$ is the total loss function defined as

$$\mathcal{L}(\theta_K, \theta_s) = \omega_{PDE}\mathcal{L}_{PDE}(\theta_K, \theta_s) + \omega_{IC}\mathcal{L}_{IC}(\theta_K) + \omega_{BC}\mathcal{L}_{BC}(\theta_K) + \omega_{data}\mathcal{L}_{data}(\theta_K). \quad (6)$$

The first term, $\mathcal{L}_{PDE}(\theta_K, \theta_s)$, represents the loss associated with the residual of the PDE Equation (2a), as follows

$$\begin{aligned}\mathcal{L}_{PDE}(\theta_K, \theta_s) = \frac{1}{n_{PDE}} \sum_{s=1}^{n_{PDE}} \Bigg| \frac{\partial}{\partial t}\tilde{K}^{(s)}(t_i, x_i; \theta_K) - \tilde{s}(x_i)A(x_i)\left[\tilde{K}^{\phi}(t_i, x_i; \theta_K)L(x_i)^{1-\phi}\right] \\ +\delta(x_i)\tilde{K}(t_i, x_i; \theta_K) - d(x_i)\frac{\partial^2 \tilde{K}(t_i, x_i; \theta_K)}{\partial x^2}\Bigg|^2,\end{aligned} \quad (7)$$

where n_{PDE} is the number of points in the training set for the PDE. The sample points (t_i, x_i) are randomly selected from the domain $(0, l) \times (0, T)$, with final time $T > 0$. The second and third terms, $\mathcal{L}_{IC}(\theta_K)$ and $\mathcal{L}_{BC}(\theta_K)$ in Equation (6), denote the loss associated with the initial and boundary conditions, respectively. They are the mean squared error of the estimated and expected values of the capital density at $t = 0$ and its derivative on the boundaries of the spatial domain, respectively. The last term, $\mathcal{L}_{data}(\theta_s)$ in Equation (6), represents the loss associated with the observed data points $\mathcal{D}$ for the capital density. It is defined as

$$\mathcal{L}_{data}(\theta_K) = \frac{1}{n_d} \sum_{i=1}^{n_d} \left| \tilde{K}(\hat{t}_i, \hat{x}_i) - \hat{K}_i \right|^2. \quad (8)$$

The weights ω_{PDE}, ω_{IC}, ω_{BC}, and ω_{data} in Equation (6) are hyperparameters that control the influence of each term in the total loss function. The computations are performed using the PyTorch library in Python.

3. RESULTS

We propose to obtain the savings rate function by solving the inverse problem given by Equation (2) using PINNs with two MLP networks for the respective case study: $l \equiv 10$, $T \equiv 10$, $\phi \equiv 1/3$, $\delta(x) \equiv 0.05$, $A(x) \equiv 1$, $d(x) \equiv 0.25$, and the functions $L(x) = 0.3x^2\left[1 - \cos\left(4\pi x/l\right)\right]$, $k_0(x) = \cos^4\left(\pi x/(2l) - \pi/2\right)$, and $s(x) = 0.2\cos^4\left(\pi x/(2l)\right)$. The first network solves the PDE in Equation (2), using t and x as inputs, while the second estimates the savings rate with input only from x. The training is carried out based on a set of $n_d = 110$ observed data points recorded for $K(x,t)$, where these points can be seen in Figure 1, these values of $K(x,t)$ were obtained through the solution of the forward problem using the Finite Element Method (FEM). The weights are set as $\omega_{PDE} = \omega_{IC} = \omega_{BC} = \omega_{data} = 1$.

The two MLP networks have the architectures $2-n_n^{(K)}\times n_l^{(K)}-1$ and $1-n_n^{(s)}\times n_l^{(s)}-1$, where n_n represents the number of neurons per hidden layer, n_l the number of hidden layers in the network, and the upper index identifies the corresponding network: the

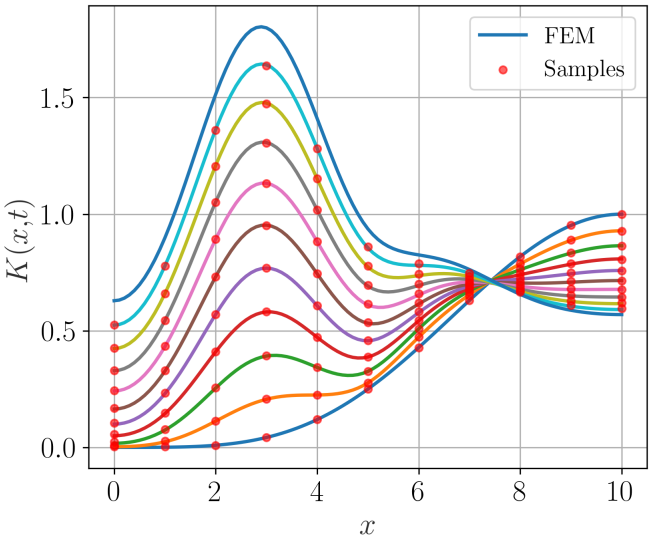
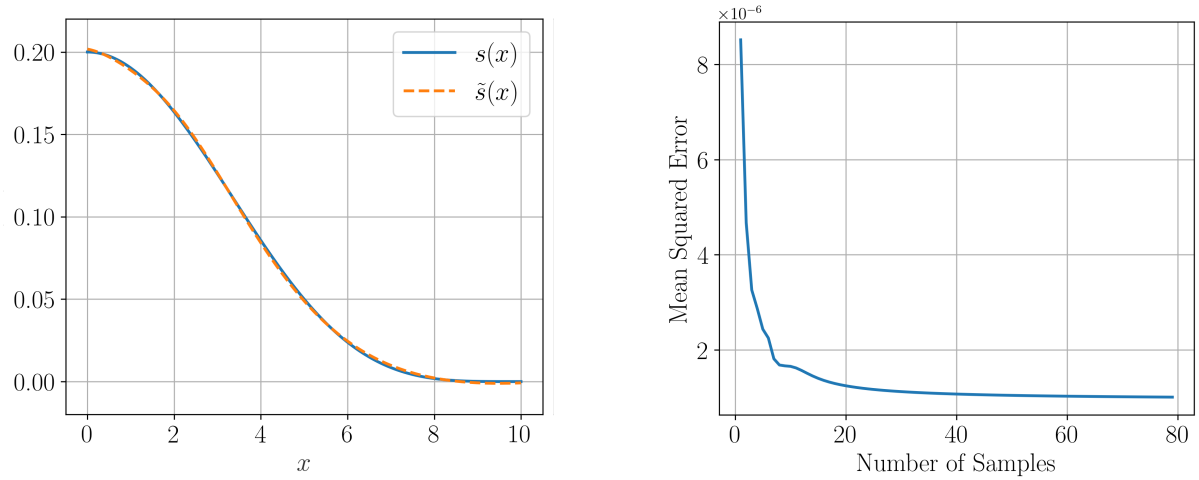


Figure 1. Observed data points for $K(x,t)$ obtained using FEM. Each curve corresponds to a time instant $t \in \{0, 1, \ldots, 10\}$. Source: Author.

first network estimates $\tilde{K}(t, x)$, while the second the $\tilde{s}(x)$. To determine the appropriate values of n_n and n_l, we conducted separate studies on the ideal architecture for both approximating the savings rate function $\tilde{s}(x)$ and solving the forward problem defined by Equation (2) using PINNs. Due to the stochastic nature of the procedure (including initialization and the training algorithm), each study was repeated three times. In both cases, we adopted a stopping criterion of $\mathcal{L} < 10^{-5}$, where $\mathcal{L}$ represents the total loss of the MLP in approximating $\tilde{s}(x)$, calculated based on the function provided by the case study, and the total loss in estimating $\tilde{K}(x,t)$, where the capital value $\tilde{K}(x,t)$ was compared with the numerical solution obtained by the finite element method. In all tests performed, the stopping criterion was achievable within 50000 epochs (training iterations).

The final architecture adopted to estimate $\tilde{s}(x)$ was $1 - 40 \times 1 - 1$. The process of defining the network architecture for the savings rate function involved analyzing different configurations, varying the number of hidden layers $n_l^{(s)}$ from 1 to 6, and the number of neurons per layer $n_n^{(s)}$ from 10 to 100, with increments of 10. The average value of the smallest number of training iterations, calculated from the stopping criterion, for the best architecture was $\overline{n}_e^{(s)} = 250$, where $n_e^{(s)}$ indicates the total number of iterations until the stopping criterion was reached. The result of the estimated curve for $\tilde{s}(x)$ is shown in Figure 2a. To evaluate the error in the approximation, we used the mean squared error of the estimated and expected $s(x)$, and the results are shown in Figure 2b, which demonstrates that, starting from 40 points, the mean squared error converges to a value close to 10^{-6}.

A preliminary study was conducted to identify the most promising architectures for solving the PDE in Equation (2) using PINNs. In this study, different numbers of

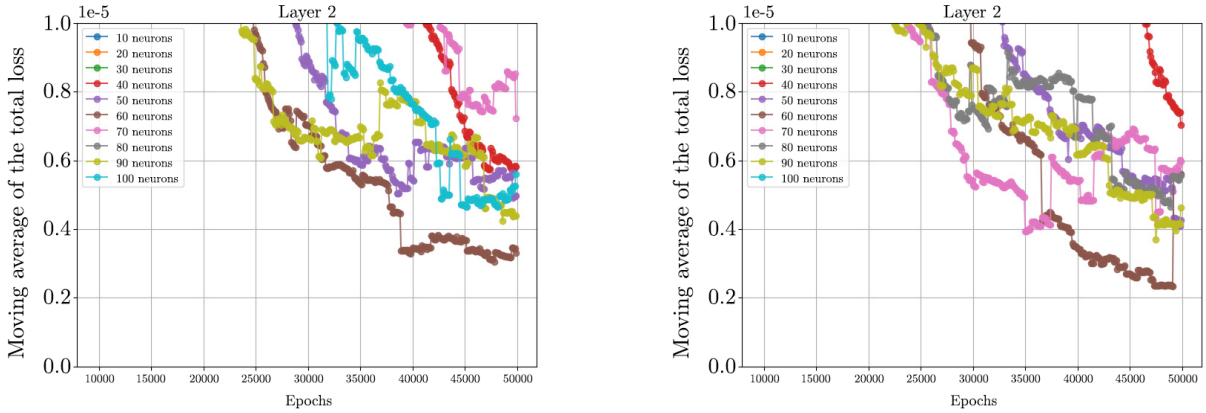


(a) Estimated *vs.* expected $s(x)$.

(b) Mean squared error of the estimated $s(x)$.

Figure 2. Approximation of the savings rate function and evaluation of the associated mean squared error. Source: Author.

hidden layers, $n_l^{(K)}$, ranging from 1 to 6, and different numbers of neurons per layer, $n_n^{(K)}$, ranging from 10 to 100 with increments of 10, were evaluated. To minimize the influence of randomness, the same tests were repeated twice. The study involved evaluating the moving average of the total loss, $\mathcal{L}$, calculated every 100 iterations, over a total of 50000 iterations. The configuration with $n_l^{(K)} = 2$ hidden layers produced the best results. The curves generated by this configuration are presented in Figure 3. By analyzing Figure 3a and Figure 3b, we observe that the configuration with approximately $n_n^{(K)} = 60$ neurons achieved the best performance.



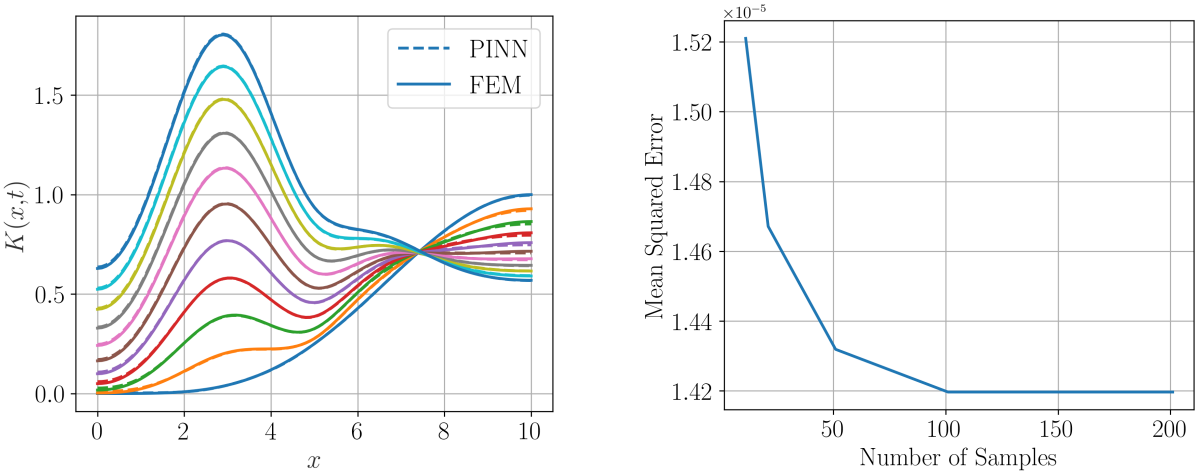
(a) Moving average of the total loss of the network for the first test.

(b) Moving average of the total loss of the network for the second test.

Figure 3. Moving average of the total loss of the network over 100 iterations when estimating $\tilde{K}(x,t)$ using the direct PINNs method. Source: Author.

Before defining the architecture $1 - 60 \times 2 - 1$ as the most suitable for estimating $\tilde{K}(x,t)$, we investigated the influence of the number of samples on the network to determine the optimal input set for $n_x \times n_t$, where n_x is the number of spatial input points and n_t

is the number of temporal input points. To this end, we explored different sample sizes, ranging from 25^2, 50^2, 75^2, 100^2, to 125^2. In this analysis, in addition to considering the configuration with 60 neurons, we also examined its neighboring architectures, testing configurations with 50 and 70 neurons. As a result of this analysis, the best architecture using the stopping criterion was $1-70\times2-1$, with $n_x \times n_t = 50^2$. The total average number of iterations until the stopping criterion was met was $\overline{n}_e^{(K)} = 14 \times 10^3$. The results of $\tilde{K}(x,t)$ are shown in Figure 4a for the time instants $t \in \{0, 1, 2, \ldots, 10\}$, and are compared with numerical finite element solutions. To assess the accuracy of the approximation, we calculated the mean squared error. The tests were conducted for different sample sizes, varying between 11, 21, 101, and 201 points. Figure 4b presents the results, indicating that, starting from 100, the mean squared error converges to 1.42×10^{-5}.



(a) Approximation of $\tilde{K}(x,t)$ by solving the forward problem using PINNs.

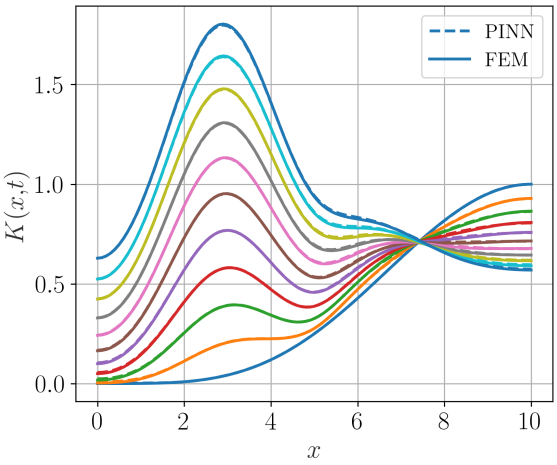
(b) Mean squared error of the approximation of $\tilde{K}(x,t)$ as a function of the number of points.

Figure 4. Results of the approximation of $\tilde{K}(x,t)$ and mean squared error for different sample sizes. Source: author.

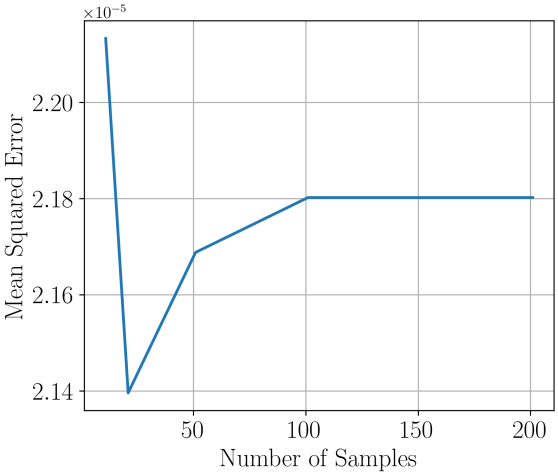
After conducting both studies, we concluded that the ideal architectures to solve the inverse problem described by Equation (2) are $2-70\times2-1$ and $1-40\times1-1$, both using an input set for the samples with $n_{PDE} = 50^2$. The first network is responsible for estimating the capital $\tilde{K}(x,t)$, while the second approximates the savings rate function $\tilde{s}(x)$. With this architecture, 30100 training iterations were sufficient to meet the stopping criterion. The results obtained for $\tilde{K}(x,t)$ are presented in Figure 5a, along with numerical finite element solutions, corresponding to the time instants $t \in \{0, 1, 2, \ldots, 10\}$. As for the forward problem, we calculated the mean squared error of $\tilde{K}(x,t)$. Figure 5b shows that, for samples with 100 points or more, the mean squared error converges to a value close to 2.18×10^{-5}.

The Figure 6a presents the result of the estimated curve for $\tilde{s}(x)$ for the adopted architecture. We used the mean squared error to assess the quality of the estimations,

for different numbers of input points for x. The results are presented in Figure 6b, which shows that, from 40 points onward, the mean squared error converges to a value close to 2.2×10^{-5}.

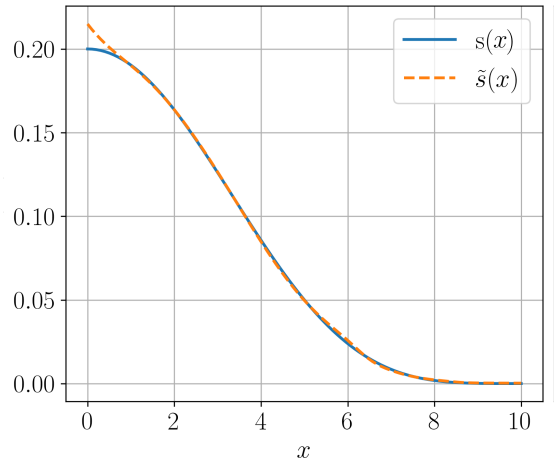


(a) Approximation of $\tilde{K}(x,t)$ by solving the inverse problem using PINNs.

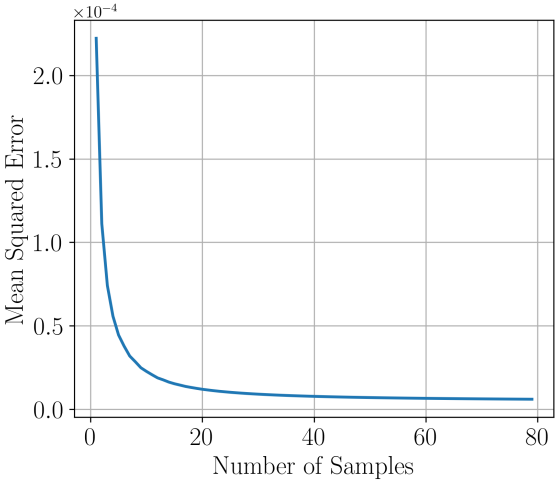


(b) Mean squared error of the approximation of $\tilde{K}(x,t)$.

Figure 5. Results of the approximation of $\tilde{K}(x,t)$ and mean squared error for different sample sizes. Source: author.



(a) Estimation of $\tilde{s}(x)$ by PINNs when solving the inverse problem.



(b) Mean squared error of the estimation of $\tilde{s}(x)$ by PINNs when solving the inverse problem.

Figure 6. Estimation of the savings rate function and evaluation of the associated mean squared error by PINNs for the inverse problem. Source: Author.

4. CONCLUSIONS

We presented the application of PINNs to solve the inverse problem of the Solow-Swan spatial model, aiming to estimate the savings rate function. The method uses MLP neural networks to simultaneously estimate both the solution to the PDE model and the unknown savings rate function. For this purpose, two neural networks are trained: the

first models the evolution of capital density, while the second estimates the savings rate as a function of the spatial coordinate.

The training, based on a set of just 110 observational data points computed for $K(x,t)$, proved to be sufficient. The results indicate that the proposed approach is effective in estimating the savings rate function, with a mean squared error on the order of 10^{-5} when compared to the expected function. These results confirm the efficiency of the PINNs technique in solving the inverse problem of the Solow-Swan spatial model, providing a good approximation for the savings rate function. As the next step, we plan to test the methodology on new case studies and apply it to real data provided by IBGE.

ACKNOWLEDGMENTS

The authors thank CAPES for granting the Master's Scholarship, which made this work possible.

REFERENCES

[1] C. Camacho and B. Zou, "The Spatial Solow Model", *Economics Bulletin*, no. 2, pp. 1–11, 2004.

[2] R. Engbers, "Inverse problems in geographical economics: parameter identification in the spatial Solow model", *Economic Record*, no. 32, pp. 334–361, 2014. DOI: 10.1098//rsta.2013.0402.

[3] W. Hu, "A new method to solve the forward and inverse problems for the spatial Solow model by using Physics Informed Neural Networks (PINNs)", *Engineering Analysis with Boundary Elements*, p. 106 013, 2024. DOI: https://doi.org/10.1016/j.enganabound.2024.106013.

[4] J. Juchem Neto, J. Claeyssen, and S. Pôrto Júnior, "Returns to scale in a spatial Solow–Swan economic growth model", *Physica A: Statistical Mechanics and its Applications*, p. 122 055, 2019. DOI: https://doi.org/10.1016/j.physa.2019.122055.

[5] D. P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, 2017. arXiv: 1412.6980 [cs.LG]. [Online]. Available: https://arxiv.org/abs/1412.6980.

[6] L. Lu, R. Pestourie, W. Yao, Z. Wang, F. Verdugo, and S. G. Johnson, "Physics-Informed Neural Networks with Hard Constraints for Inverse Design", *SIAM Journal on Scientific Computing*, vol. 43, no. 6, B1105–B1132, 2021. DOI: 10.1137/21M1397908.

[7] M. Raissi, P. Perdikaris, and G. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations", *Journal of Computational Physics*, pp. 686–707, 2019. DOI: https://doi.org/10.1016/j.jcp.2018.10.045.

[8] R. Solow, "A Contribution to the Theory of Economic Growth", *Quarterly Journal of Economics*, pp. 65–94, 1956. DOI: https://doi.org/10.2307/1884513.

[9] T. W. Swan, "Economic Growth and Capital Accumulation", *Economic Record*, no. 32, pp. 334–361, 1956. DOI: https://doi.org/10.1111/j.1475-4932.1956.tb00434.x.