

Estimation of the Reynolds number in a Poiseuille flow using artificial neural networks

M. Carrillo¹, J.A. González² and U. Que³

Laboratorio de Inteligencia Artificial y Supercómputo, Instituto de Física y Matemáticas, Universidad Michoacana de San Nicolás de Hidalgo. Edificio C-3, Cd. Universitaria, 58040, Morelia, Michoacán, México.

Email: ¹mcarrillo@ifm.umich.mx, ²gonzalez@ifm.umich.mx,
³qsalinas@ifm.umich.mx

Abstract. In this work the estimation of Reynolds number in a 2-dimensional Poiseuille flow is explored employing artificial neural networks (ANNs). The velocity fields of the fluids were generated evaluating the Hage-Poiseuille equation for different Reynolds (Re) from 20 to 2000. The velocity profile obtained for each case is used as input data for the ANNs, which is then trained to predict the Re . The results show an accuracy of at least of 99.5% in all prediction cases. This analysis is the first step towards the construction of a Machine Learning algorithm capable of computing physical parameters in more general scenarios.

1. Motivation

The study of flows in incompressible fluids at moderate and low Reynolds (Re) number has applications in many relevant physical problems, such as laminar flows through channels and pipes, for which, extensive studies have been done in literature [1,2].

Nowadays, many of these applications are present in the field of engineering, involving viscous incompressible flows through ducts, filters and other devices. With that motivation and given its complexity, several studies have been performed in order to obtain accurate solutions to the Navier-Stokes equations describing the behavior of the fluid.

In complicated problems, where the solutions have to be computed using numerical methods, the estimation of some physical parameters is not an easy task. For instance, if we want to estimate the Reynolds number for a 2-dimensional flow around an obstacle, we need a method that using the vector fields obtained from the simulation (for example, the velocity field) can predict the Re associated to that experiment. As a first step to implement it, in this paper we present a simpler configuration: a Poiseuille flow and propose a method to estimate Reynolds numbers in the range from 20 to 2000.

To perform the estimation of the Reynolds number we use a Machine Learning method: artificial neural networks (ANNs), that have been very useful in pattern recognition, in particular they have been used to estimate and predict parameters in problems involving fluids. For example in [3] the authors compute the velocity and diffusion coefficient of a fluid that creates a flow around an obstacle in two dimensions using ANNs. In [4] an implementation that predicts the power ratio and torque of a vertical axis wind turbine training ANNs with experimental data were done. In [5] it is predicted the friction factor of an open channel flow. In [6] the flow patterns of different Reynolds numbers are obtained using ANNs, and in [7] the authors estimate the Reynolds number and the dimensions of the enclosure from the knowledge of the centreline velocity field using an inverse method. The structure of the paper is as follows: In section 2 we briefly, describe the Poiseuille flow problem, in section 3

the concepts of ANNs, in section 4 the methodology to estimate the Reynolds number and in section 5 we present the results and in section 6 we conclude.

2. Two dimensional Poiseuille flow

The Poiseuille equation, is a physical law that allows to describe the pressure drop in a Newtonian incompressible fluid with a laminar flow through a cylindrical pipe of constant cross section. It is assumed that the diameter of the cylinder is negligible compared with its length, and no acceleration of the fluid within the pipe is considered.

In classical hydrodynamics, the flow velocity at a given point in the space, can be obtained by solving the momentum equation of Navier-Stokes [1]. For a Poiseuille flow in two dimensions the analytical solution for the time-dependent behaviour is given by

$$v_x(y, t) = \frac{F}{2\nu} y(y-L) + \sum_{n=0}^{\infty} \frac{4FL^2}{\nu\pi^3(2n+1)^3} \sin\left(\frac{\pi y}{L}(2n+1)\right) \exp\left(-\frac{(2n+1)^2 \pi^2 \nu}{L^2} t\right), \quad (1)$$

where F is the magnitude of the body force in the direction of the flow, L is the diameter of the cylinder, and ν is the kinematic viscosity. The second term in the right hand side of equation (1) computes the transitional behavior of the solution. Thus, the stationary solution for the Poiseuille flow is

$$v_x(y, t) = \frac{F}{2\nu} y(y-L). \quad (2)$$

For a Poiseuille flow, the Reynolds number is

$$Re = \frac{v_{\max} L}{\nu}, \quad (3)$$

where v_{\max} , is the maximum velocity characteristic achieved in its steady state, related to the magnitude of the body force F by

$$v_{\max} = \frac{FL^2}{\nu}. \quad (4)$$

In this paper we propose a method to estimate Re as a function of the x component of the velocity of the y-axis, i.e., $v_x(y)$, using the ANNs machinery. In the next section, details about ANNs' concepts and operations are described.

3. Artificial neural networks concepts

ANNs are a Machine Learning paradigm based on biological systems, commonly used in regression, pattern recognition and classification problems. ANNs are fed with some variables or data on a particular problem, and after a series of calculations made by the composition of non linear functions, give an outcome related with the information they were trained. The most common ANN is the Multilayer Perceptron (MLP), derived from the simple Perceptron [6], which consists of interconnected layers of nodes or neurons: an input layer, one or several middle or hidden layers and an output layer. In a MLP information goes from input to hidden layers and later from hidden to the output one. For neurons in hidden and output layers the connections are regulated by the weight coefficients, which are adjusted in the training phase.

In order to illustrate a MLP, or ANN in this case, in organization and operation, consider a 3 layer network with n inputs, m hidden neurons and l outputs. The relation between and input vector $\vec{x} = \{x_1, x_2, \dots, x_n\}$ and the k -the output neuron is determined by the expression

$$y_k = F\left(\tilde{\omega}_{0k} + \sum_{j=1}^m \tilde{\omega}_{jk} * G\left(\sum_{i=1}^n \omega_{ij} * x_i + \omega_{0j}\right)\right), \quad (5)$$

where ω_{ij} and $\tilde{\omega}_{jk}$ are the connection weights between the neurons in input-hidden and hidden-output layers respectively; ω_{0j} and $\tilde{\omega}_{0k}$ are some extra weights called bias, operating as thresholds; F and G are the named neurons' activations functions on each layer, commonly the logistic or hyperbolic tangent functions are employed.

In order to obtain the proper outcomes the weights must be adjusted properly. One way to do this is through a supervised training process, minimizing a cost function from a collection of examples with already known results or targets. Thus, the training phase consists of running a set of coupled inputs and targets $(\tilde{x}_i^p, \tilde{y}_k^p)$, with $i = 1, \dots, n$; $k = 1, \dots, l$; $p = 1, \dots, N$, where N is the number of example pairs, adjusting the weights such that minimize the cost function

$$E(\bar{\omega}) = \frac{1}{N} \sum_{p=1}^N \sum_{k=1}^l \frac{1}{2} (\tilde{y}_k^p - y_k^p)^2, \quad (6)$$

where $\bar{\omega}$ are the ANN's weights defined as before in a vector representation and y_k^p are the ANN's outcomes for the \tilde{x}_i^p inputs. In this work, the minimization of this cost function is made by a gradient descent algorithm called *backpropagation* [7]. This method searches the directions where the error diminishes more on each time step t , and updates the value of each ω_{ij} at the next time step $t+1$ by the rule

$$\omega_{ij}(t+1) = \omega_{ij}(t) - \gamma \frac{\partial E(t, \bar{\omega})}{\partial \omega_{ij}(t)} + \alpha (\omega_{ij}(t) - \omega_{ij}(t-1)), \quad (7)$$

with $0 < \gamma < 1$ the parameter known as the learning rate, determined by the user; $\alpha (\omega_{ij}(t) - \omega_{ij}(t-1))$ is called the momentum term, added for preventing that the minimization algorithm gets trapped in a local minimum, and $0 < \alpha < 1$ is another parameter to be adjusted by the user. The number of iterations for training depends on the learning parameter, the random weights initialization and the training set. The resulting weights after this process are used to compute the values in the prediction set.

4. Methodology

As mentioned in section 1, the main idea of this work is to use the velocity profile as input for the ANN, so it learns its associated Reynolds number giving some particular characteristics of the fluid and the cylinder. As a first step, we generate the velocity profiles using the analytic solution given by equation (2). It was assumed a fluid with a kinematic viscosity $\nu = 0.001 \text{ m}^2/\text{s}$ flowing through a cylinder of 0.5m, considering the magnitude of the body force F as the independent variable to construct 100 representative cases, in a range from 20 to 2000 Re with steps of $\Delta = 20$. On each case, 100 values of the velocity of the fluid $v_x(y)$ inside the cylinder have been extracted over a line transverse to the flow direction to be the inputs of the ANN. In figure 1, we present some of these profiles.

As the rule of thumb suggests, 80% of data should be used as training set, 10% as validation set and the remaining as the prediction set. The validation set is completely unknown to the ANN in the weight adjustment process, and is considered to check the performance of the ANN. In general, training is continued until the error in validation set gets into a local or global minimum, with the intention to avoid over-fitting on the training pairs. Sets were selected as follows:

- Prediction set: Values of the x component of the velocity, i.e., $v_x(y)$, for the simulation with $Re = 300, 600, 900, 1200, 1500, 1800, 1940, 1980, 2000$, some of them are shown in figure 1.
- Validation set: $v_x(t)$ values and their corresponding Re for 20, 40, 120, 280, 480, 1280, 1520, 1760, 1880 and 1920.

- Training set: The $v_x(y)$ values for the remaining cases for Re starting from 60 to 1900. Leaving the cases with $Re = 1940, 1960, 1980, 2000$ as a test to explore the extrapolation capabilities of the algorithm.

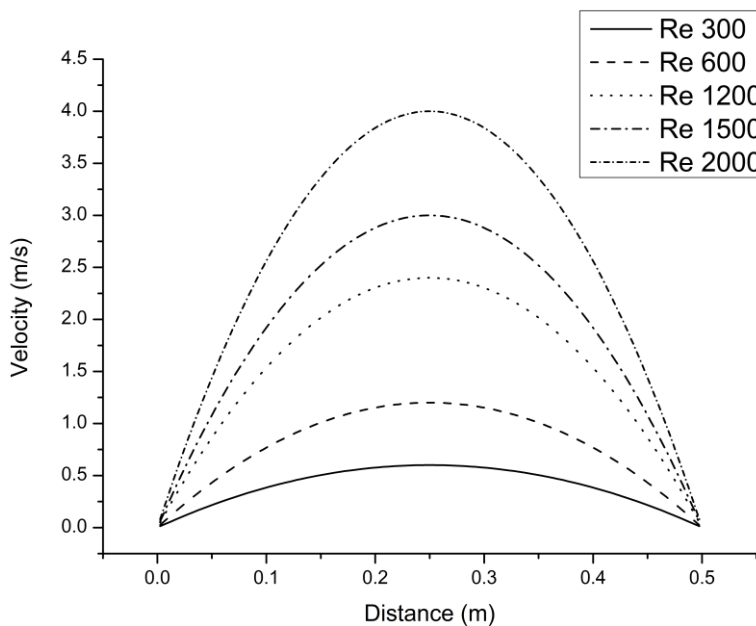


Figure 1. Velocity profiles inside the cylinder with a 0.5m diameter, for different Re . These curves were obtained modifying F in equation (2).

ANN's weights are randomly initialized. To show consistency on the results independently of their initial condition, ten ANNs have been constructed with the same structure, to calculate an average of the Re predicted. Every network has the following structure:

- Input layer with 100 inputs consisting of the x component of velocity, i.e., $v_x(y)$.
- Hidden layer with ten neurons, using the hyperbolic tangent as activation function.
- Output layer with a single neuron equipped with a linear function, giving the ANN's Re prediction.

Scaling data in ML is required in almost all cases. In this particular one, data seems to be in an appropriate range, however the number of inputs might overload the neurons' activation functions, in this case the hyperbolic tangent function is used in the hidden layer. We scaled the input values by a factor of 10 and the targets by a factor of 100, such that the ANNs produce the best results compared with other approaches such as max-min, mean-standard normalization in inputs; or logarithmic rescaling at the targets. The ANN's outcomes were rescaled back to obtain the physical results.

5. Results

Using a learning rate $\gamma = 0.05$ and $\alpha = 0.5$ together with backpropagation algorithm, the ANNs required in general from one thousand to five thousand iterations and a couple of minutes to reach a minimum on the validation set, time at which a signal to stop the training set was triggered. On figure 2, we observe one of the ANN's root mean square error (RMSE) on the training phase. The errors in training and validation decrease initially together, but in this case, after 1173 iterations the validation error starts growing slowly. This is the moment when the training algorithm was stopped saving those weights for the prediction phase.

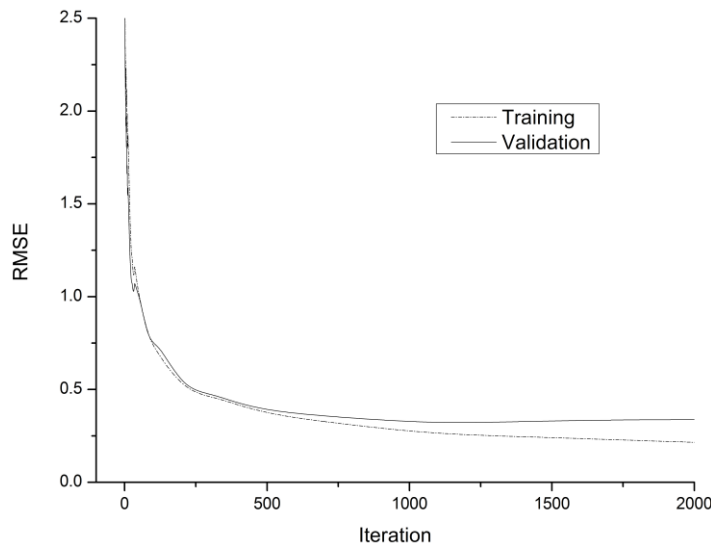


Figure 2. RMSE of training and validation phase. The ANN's training is stopped when the validation error gets a local or global minimum. In this case it was stop at iteration 1173, after which the error increases smoothly.

In table 1, we present the averaged values computed by the 10 ANNs and the relative error for their predictions. The network reached an accuracy of 100%, considering an error less than 0.5% over all sets, this means that the network was capable to classify all unknown elements with an error less than 0.5%. In the extrapolation regime ($Re = 1940, 1960, 1980, 2000$), the relative errors are bigger than in the interpolation regime since the ANNs were trained with $Re = 1900$ as the top training case, figure 3. The best results are on the middle region of data, for Re from 600 to 1800, those samples lie within the interpolation regimen and nearly other samples in the validation set. This shows the sensitivity of the ANN to the selection of the training and validation sets.

6. Conclusions

In this work, we presented a method to predict the Reynolds number in a two dimensional Poiseuille flow with an accuracy of 99.5 % for unknown flow patterns, this performance is reached based on the simplicity of the problem. The paper shows the first step towards the construction of a numerical method capable to predict physical constants analyzing only a snapshot of a vector field (in this case the x component of the velocity). Considering the good performance shown in here we are planning to consider more complicated scenarios. In particular, we would like to implement an ANN that predicts the Reynolds number of an incompressible fluid flowing around a cylindrical obstacle, expecting that the complexity of the problem could decrease the ANN performance. These results were obtained evaluating a formula. However, in more complicated scenarios, the inputs for the ANN cannot be generated using analytic solutions, and it will be necessary to use data directly from experiments or obtained using numerical simulations, nevertheless once trained the results can be obtain faster than other approaches.

Acknowledgments

This research is supported by grant CIC-UMSNH 4.23

Table 1. Predictions and errors made by the ANNs in the prediction set. All error are less than 0.5%. In the cases out of the training range ($Re = 1940, 1960, 1980, 2000$) the error increases as expected.

Reynolds	Prediction	Relative Error (%)
300	300.414±0.355	0.138
600	600.107±0.192	0.018
900	900.232±0.295	0.026
1200	1200.028±0.124	0.002

1500	1499.702±0.348	0.020
1800	1800.922±0.101	0.051
1940	1937.841±0.596	0.111
1960	1956.434±0.842	0.182
1980	1974.664±1.140	0.270
2000	1992.493±1.490	0.375

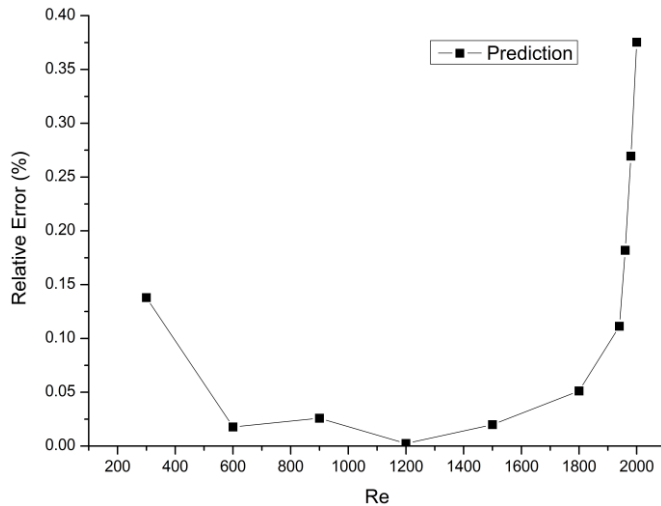


Figure 3. Relative errors between the real Re and the ANNs' averaged outcomes in the prediction set. All errors are less than 0.5%, however those outside the interpolation regime increase.

References

- [1] Batchelor G K 1997 *An Introduction to Fluid Dynamics* (Cambridge: London Cambridge University Press)
- [2] Landau L D and Lifshitz E M 1987 *Fluids Mechanics* (Oxford: Pergamon Press)
- [3] Lauret P, Heymes F, Aprin L, Johannet A and Slagen P 2015 *Chemical Engineering Transactions* **43** 1621-26
- [4] Sargozaei J and Kianifar A 2009 *Simulation Modelling Practice and Theory* **17** 1290-98
- [5] Yuhong Z and Wenxin H 2009 *Commun Nonlinear Sci Numer Simulat* **14** 2373-78
- [6] Zhang L, Akiyama M, Huang K, Sugiyama H and Ninomiya N 1996 *Estimation of Flow Patterns by Applying Artificial Neural Networks* **2** 1358-63
- [7] Das R 2012 *A simulated annealing-based inverse computational fluid dynamics model for unknown parameter estimation in fluid flow problem* **26** 499-13
- [8] Rosenblatt F 1961 *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms* (Washington DC: Spartan Books)
- [9] Rumelhart D E, Hinton G E and Williams R J *Nature* **323** 533-36