Received: 29 September 2022 | Accepted: 4 November 2022

DOI: 10.1002/pamm.202200044

An Inverse Neural Particle Method for Flow Field Reconstruction

Henning Wessels[1,*]

[1] Institute for Computational Modeling in Civil Engineering, Pockelsstr. 3, 38106 Braunschweig

Physics-Informed Neural Networks (PINNs) are a novel discretization scheme for the solution of partial differential equations (PDEs), where a neural network is chosen as global ansatz function. The problem of solving the PDE is then cast as an optimization problem and addressed by training the neural network. PINNs have been promoted to perform particularly well in inverse problems. This contribution presents recent advances of the Neural Particle Method, an updated Lagrangian Physics-Informed Neural Network, in the inverse problem of reconstructing flow fields from sparse data.

1 Introduction

Inverse problems have received increased attention over the past years, since they often provide a natural link between measurement data and physical models, which is a central task for digital twinning. The solution of inverse problems with numerical methods however remains challenging. Here, Physics-Informed Neural Networks (PINNs) have demonstrated their effectiveness in various publications, see e.g. [1, 2]. Recent work of our group focused on parameter identification using PINN [3], which can also be regarded as an inverse problem. The Neural Particle Method (NPM), which has first been proposed for inviscid free surface flows [4] and then been extended to viscous flows [5], employs an updated Lagrangian formulation of the governing equations. To the best of the author's knowledge, only forward problems have been considered with NPM so far. In this work, the NPM is extended towards inverse problems. In particular, the following question will be addressed: Given some information of the free surface profile of a fluid and boundary conditions, is it possible to reconstruct the velocity and pressure field inside the fluid?

2 The Neural Particle Method

The NPM has been proposed in [4] as an updated Lagrangian PINN for the solution of free-surface flows. Here, we revisit the NPM implementation for the inviscid Euler equations:

$$\begin{aligned} \mathbf{a} &= \frac{1}{\rho} \operatorname{grad} p + \mathbf{b} \\ \operatorname{div} \mathbf{v} &= \mathbf{0} \end{aligned} \tag{1}$$

Here, $\mathbf{a}$ denotes the acceleration, ρ the density, p the pressure, $\mathbf{b}$ accelerations related to body forces and $\mathbf{v}$ the velocity. The temporal discretization in NPM is based on Implicit Runge-Kutta (IRK) integration as proposed for discrete time PINNs in [1]. The extensions towards second order PDEs as in NPM is described in [4]. In IRK, the acceleration $\mathbf{a}$ is evaluated at stages $t^i = t\left(t_n + c_i \Delta t\right), \forall i \in [1, s]$, with s the total number of stages. The coefficients c^i are given by the Butcher tableau along with the coefficients a^{ji} and b^j appearing in (2) and (3). The update of the velocity follows:

$$\begin{aligned} \mathbf{v}^j &= \mathbf{v}_n + \Delta t \sum_{i=1}^{s} a^{ji}\, \mathbf{a}^i \left(\mathbf{v}^i,\, \mathbf{x}^i,\, t_n + c^i \Delta t\right) \\ \mathbf{v}_{n+1} &= \mathbf{v}_n + \Delta t \sum_{j=1}^{s} b^j \mathbf{a}^j \left(\mathbf{v}^j, \mathbf{x}^j,\, t_n + c^i \Delta t\right) \end{aligned} \tag{2}$$

and, accordingly the position update is subject to:

$$\begin{aligned} \mathbf{x}^j &= \mathbf{x}_n + \Delta t \sum_{i=1}^{s} a^{ji}\, \mathbf{v}^i \\ \mathbf{x}_{n+1} &= \mathbf{x}_n + \Delta t \sum_{j=1}^{s} b^j \mathbf{v}^j \end{aligned} \tag{3}$$

* Corresponding author: e-mail h.wessels@tu-braunschweig.de, phone +49 531 391 94530, fax +49 531 391 94511

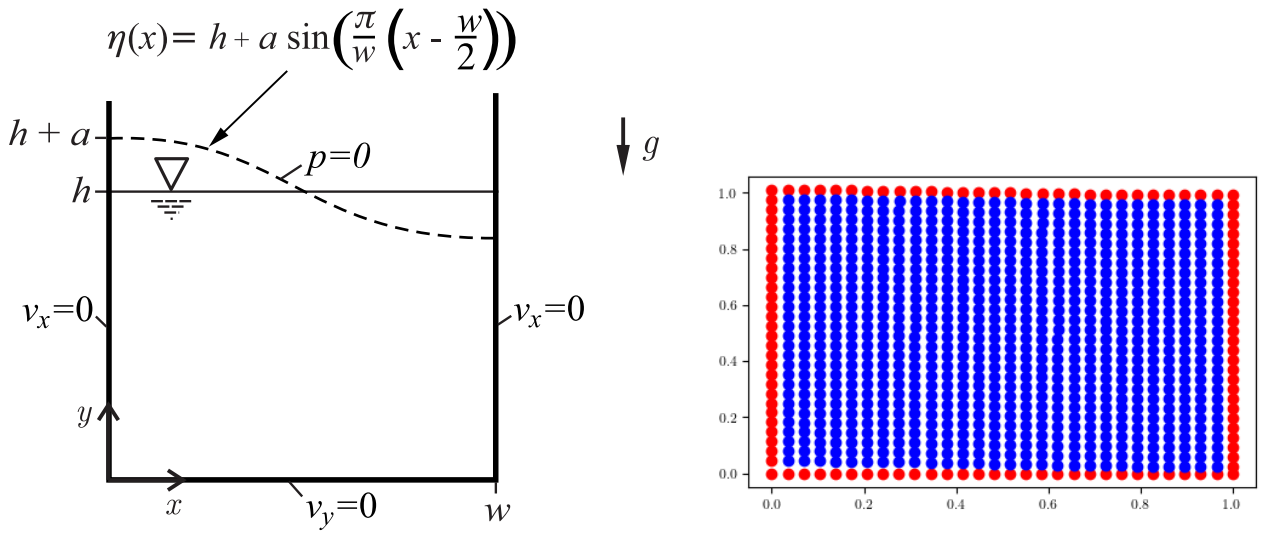

Fig. 1: Setup of the sloshing test case. In the present work, the geometry is defined by w = h = 1 m and a = 0.01 m. The density of the liquid is $\rho = 1$ kg/m^3 and the gravity acceleration g = 1 m/s^2. On the right, boundary and free surface nodes $\mathbf{v}_{n+1}(\mathbf{x}_{n+1}^{boun})$ are marked in red, while inner nodes $\mathbf{v}_{n+1}(\mathbf{x}_{n+1}^{inner})$ are blue.

The pressure stages are directly computed such that the incompressibility condition $(1)_2$ is fulfilled. Only the pressure of the new time step is interpolated from the IRK scheme:

$$p_{n+1} = \sum_{j=1}^{s} b^j p^j, \qquad \text{with} \quad \sum_{j=1}^{s} b^j = 1 \tag{4}$$

For the spatial discretization, a multi-output feed-forward neural network is chosen to approximate the velocity and the pressure at all stages t^i as well as the velocity at the new time step t_{n+1}. The concatenated ansatz for all primary variables writes:

$$\left[v_x^1, ..., v_x^s, v_{x\,n+1}, v_y^1, ..., v_y^s, v_{y\,n+1}, p^1, ..., p^s\right] = PINN\left(\mathbf{x}_{n+1}, \boldsymbol{\theta}\right) \tag{5}$$

Here, $\boldsymbol{\theta}$ denotes the vector of trainable network parameter.

3 A NPM formulation for flow field reconstruction

Unlike in forward problems, in the inverse setting the positions $\mathbf{x}_{n+1}$ at the current time step are (partially) known. Therefore, the Euler equations can be written in a purely Lagrangian formulation and the ansatz (5) becomes a function of the current position $\mathbf{x}_{n+1}$. See also table 1 for a comparison of the known and unknown quantities in the forward and inverse setting discussed in this paper.

We consider the problem of reconstructing the pressure field at t_{n+1} given the velocity field at t_n and t_{n+1} as well as the position $\mathbf{x}_{n+1}$. Following the notation in [4], the associated loss function is composed of four terms:

$$\mathcal{L} = \underbrace{SSE_{IRK} + SSE_{\mathrm{div}\,\mathbf{v}}}_{\text{PDE loss terms}} + \underbrace{SSE_{\bar{p}} + SSE_{\mathbf{v}_{n+1}}}_{\text{data loss terms}} \tag{6}$$

with the PDE loss terms

$$\begin{aligned} SSE_{IRK} &= \sum_{j=1}^{s} \left|\mathbf{v}_n^j\left(\mathbf{x}_{n+1}^{boun}\right) - \mathbf{v}_n\right|^2 + \left|\mathbf{v}_n^{s+1}\left(\mathbf{x}_{n+1}^{boun}\right) - \mathbf{v}_n\right|^2 \\ SSE_{\mathrm{div}\,\mathbf{v}} &= \sum_{i=1}^{s} \left|\mathrm{div}\,\mathbf{v}^i\left(\mathbf{x}_{n+1}\right)\right|^2 + \left|\mathrm{div}\,\mathbf{v}_{n+1}\left(\mathbf{x}_{n+1}\right)\right|^2 \end{aligned} \tag{7}$$

related to the IRK integration of the balance of momentum and to the balance of mass. The data loss terms

$$\begin{aligned} SSE_{\bar{p}} &= \sum_{i=1}^{s} \left|p^i\left(\bar{\mathbf{x}}_p\right) - \bar{p}\right|^2 \\ SSE_{\mathbf{v}_{n+1}} &= \sum_{i=1}^{s} \left|\mathbf{v}_{n+1}\left(\mathbf{x}_{n+1}^{boun}\right) - \bar{\mathbf{v}}_{n+1}\right|^2 \end{aligned} \tag{8}$$

	known	unkown
forward	$\mathbf{x}_n, \mathbf{v}_n$	$\mathbf{x}_{n+1}, \mathbf{v}_{n+1}, \mathbf{p}_{n+1}$
inverse	$\mathbf{x}_{n+1}, \mathbf{v}_n(\mathbf{x}_{n+1}^{boun}), \mathbf{v}_{n+1}(\mathbf{x}_{n+1}^{boun})$	$\mathbf{v}_{n+1}(\mathbf{x}_{n+1}^{inner}), \mathbf{p}_{n+1}$

Table 1: List of known and unknown quantities in the forward and inverse setting in NPM. In the inverse setting, the velocity of the old and new time step are known only at the boundary and free surface ($\mathbf{x}_{n+1}^{boun}$)

are related to the zero pressure boundary condition at the free surface and the velocity data. As described in [4], velocity boundary conditions are strictly imposed by an extended ansatz, the so-called *hard boundary conditions*. In the inverse setting, the data-fit term $SSE_{\mathbf{v}_{n+1}}$ becomes an additional loss term that enforces the ansatz (5) to approximate the velocity field at t_{n+1}.

4 Results

As a test case, sloshing in a cylinder is considered. The geometry of the test case is illustrated in figure 1, and the position of inner and boundary nodes ($\mathbf{x}_{n+1}^{inner}$ and $\mathbf{x}_{n+1}^{boun}$, respectively) is graphically illustrated.

Figure 2 illustrates the numerical results. The error in the velocity field is expressed as absolute deviation $(\mathbf{v}_{n+1} - \bar{\mathbf{v}}_{n+1})$ from the prescribed velocity field $\bar{\mathbf{v}}_{n+1}$. The pressure is accurately reconstructed everywhere inside the domain with a relative error below 2%. Since pressure is enforced at the free surface, the free surface is excluded from the error computation.

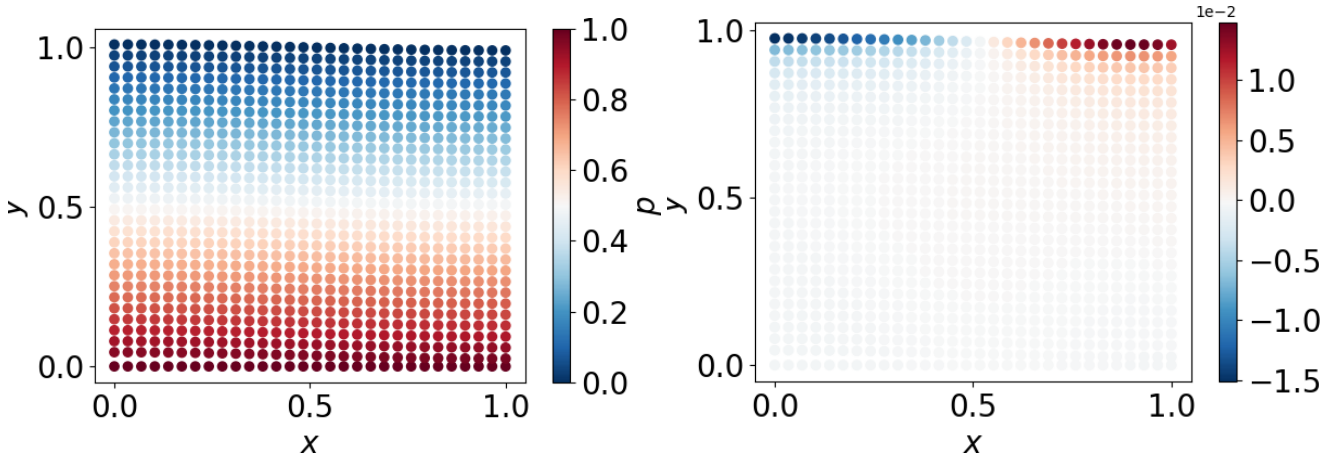

Fig. 2: Pressure: Reconstructed pressure field $\mathbf{p}_{n+1}$ (left) and relative error w.r.t. to reference NPM solution (right) obtained in the forward setting. Since zero pressure is enforced at the free surface, the free surface has been excluded from the error computation.

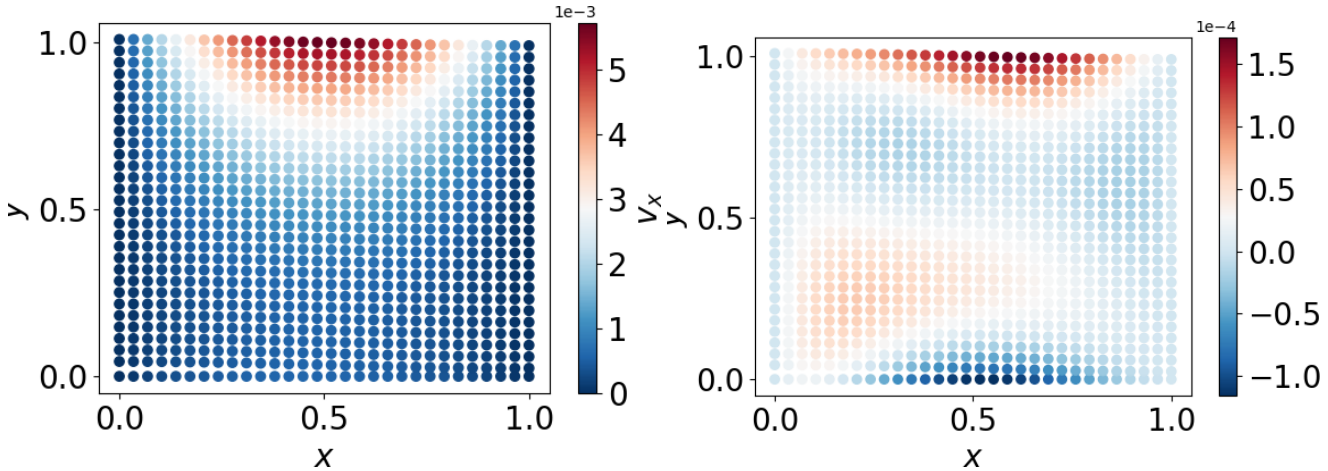

Fig. 3: Velocity (x-direction): Reconstructed velocity field $\mathbf{v}_{x\,n+1}$ (left) and difference of actual and inferred velocity field (right).

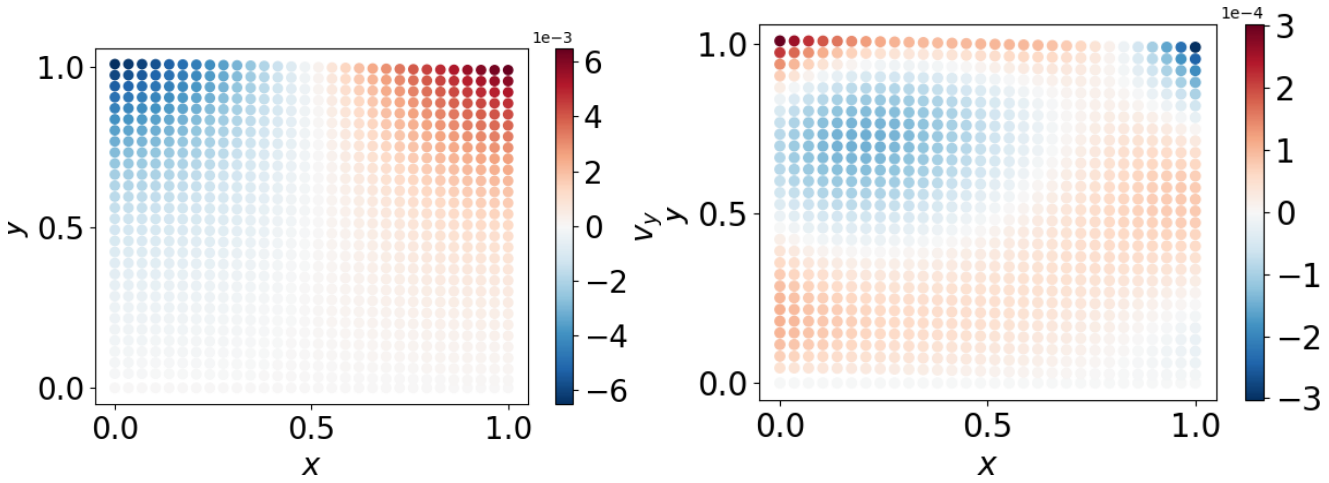

Fig. 4: Velocity (y-direction): Reconstructed velocity field $\mathbf{v}_{y\,n+1}$ (left) and difference of actual and inferred velocity field (right).

5 Conclusion and outlook

An inverse formulation of the invisicid neural particle method has been presented. The numerical results show promising results of flow field reconstruction, which give rise to a variety of relevant applications in the future. Thus, the effectiveness of this approach in advanced test cases is investigated in ongoing work.

Acknowledgements The support of the German Research Foundation is gratefully acknowledged in the following projects:

- DFG GRK2075-2: *Modelling the constitutional evolution of building materials and structures with respect to aging*.
- DFG 501798687: *Monitoring data driven life cycle management with AR based on adaptive, AI-supported corrosion prediction for reinforced concrete structures under combined impacts*. Subproject of SPP 2388: *Hundred plus - Extending the Lifetime of Complex Engineering Structures through Intelligent Digitalization*.

In both projects, the solution of inverse problems is a key enabler to link measurement data and physical models. Open access funding enabled and organized by Projekt DEAL.

References

[1] M. Raissi, P. Perdikaris, and G.E. Karniadakis, JCP **378**, (2019).
[2] M. Raissi, M. Yazdani, and G.E. Karniadakis, Science **367(6481)**, (2020).
[3] D. Anton, and H. Wessels, arXiv **arXiv:2212.07723 (cs)**, (2022).
[4] H. Wessels, C. Weißenfels, and P. Wriggers, CMAME **368**, (2020).
[5] J. Bai, Y. Zhou, Y. Ma, H. Jeong, H. Zhan, C. Rathnayaka, E. Sauret, and Y. Gu, CMAME **393**, (2022).