Chemical Engineering Science 311 (2025) 121624

Contents lists available at ScienceDirect

Chemical Engineering Science

journal homepage: www.elsevier.com/locate/ces

Propagator-moments approximation for recurrence CFD: Application to species transport in turbulent flows

H. Lumetzberger [a,b], S. Pirker [a], T. Lichtenegger [a,*]

[a] Department of Particulate Flow Modelling, Johannes Kepler University, Linz, Austria
[b] K1-MET Metallurgical Competence Center, Stahlstraße 14, 4020, Linz, Austria

ARTICLE INFO

Keywords:
Time extrapolation
Recurrence CFD
Propagator formulation
Turbulent species transport

ABSTRACT

Accurate, long-term simulations of turbulent flows pose a large challenge for standard CFD algorithms. Recurrence CFD (rCFD) provides a simple, data-assisted approximation for recurrent dynamics. We present the theoretical foundations of this approach in terms of propagator theory for passive transport processes and derive expressions for convective and diffusive contributions to large-step species transfer. We tested our new implementation on a double-sided, cuboid lid-driven cavity and compared various treatments of diffusion against detailed CFD calculations. Based on these insights, we applied the methodology to a down-scaled, industrial case of impurity transport in a steelmaking tundish at Reynolds number $Re = 220,000$. We obtained results in mostly very good agreement with detached-eddy simulations at 1/2700 of their runtime, which made rCFD faster than real time.

1. Introduction

When applying CFD simulation methods to industrial scales, the calculations often become numerically expensive and ultimately infeasible. This is particularly true when the presence of small-scale phenomena requires both high spatial and temporal accuracy, while at the same time, the overall process demands long observation times.

To address this problem, we need to integrate the equations of motion (EOMs) applying at the microscale (defined as the required spatial and temporal resolution within the context of the chosen modeling approach) with macroscopic behavior (i.e., that observed over characteristic distances and process-relevant durations of a problem), using data that reflect the flow behavior at these scales. Over the last few years, machine learning (ML) techniques, especially deep neural networks (NNs), have received increasing attention in the application to fluid mechanics. Review articles by Brunton et al. (2019) and Karniadakis et al. (2021) provide an overview of recent developments in this regard. A particularly promising approach are deep operator networks of various architectures (Alkin et al., 2024; Oommen et al., 2022; Wang et al., 2021), which can learn the solution of a partial differential equation given a sufficient amount of training data.

A dedicated process-engineering perspective is given by Zhu et al. (2022) who discuss ML for multiphase reactors. A variety of publications on this topic such as Refs. Lu et al. (2022), Ouyang et al. (2022), Shirzadi et al. (2023) and Guo et al. (2024) illustrates the potential of NNs for engineering-relevant flows. However, because generating high-fidelity data for complex flows is very time-consuming, less data-hungry approaches would be highly desirable. In their seminal paper, Raissi and Karniadakis (2018) demonstrated how the underlying EOMs of a problem can be incorporated into the loss function of certain NNs. This reduces the required amount of training data significantly but increases the complexity of the training procedure (Wang et al., 2022). Initial efforts to include the governing equations as side constraints in operator networks have already been undertaken (Wang et al., 2021).

Besides these powerful but relatively complex and mostly data-heavy methods, there are simpler, though less flexible, data-assisted approaches. Recurrence CFD (rCFD) adopts a strategy to address the temporal scaling problem for highly dynamic, recurrent systems that requires much less data than what is usually needed for ML methods to reproduce the relevant features of a specific flow (Lichtenegger et al., 2017). It utilizes one or a few short time series from detailed CFD simulations and extrapolates them to longer observation times, where the length of these series should be at least one oscillation period of the lowest relevant frequency in the system. Alternatively, a more stringent criterion can be defined via the convergence of the leading statistical moments (Lichtenegger et al., 2022). If the recurrence requirement is met, rCFD has been shown to produce very good results at low runtimes, as demonstrated e.g. for fluidized beds (Lichtenegger et al., 2017) and turbulent jets (Abbasi et al., 2020).

* Corresponding author.
E-mail addresses: hannes.lumetzberger@k1-met.com (H. Lumetzberger), stefan.pirker@jku.at (S. Pirker), thomas.lichtenegger@jku.at (T. Lichtenegger).

https://doi.org/10.1016/j.ces.2025.121624
Received 9 January 2025; Received in revised form 13 March 2025; Accepted 3 April 2025
Available online 4 April 2025

Nomenclature

Greek letters

α	volume fraction

Latin of symbols

c	concentration
$\boldsymbol{u}$	velocity
D	diffusivity
$\boldsymbol{D}$	dispersion tensor
K	propagator
Re	Reynolds number
Sc	Schmidt number

Subscripts/superscripts

(dest)	destination
(or)	origin
(ic)	the ith centered moment
b	boundary
eff	effective
disp	dispersive
phys	physical
turb	turbulence

List of abbreviations

C2Cs	cell-to-cell shifts
CFD	computational fluid dynamics
DEM	discrete element method
rCFD	recurrence CFD

By iteratively applying the method of analogs, which assumes that similar initial evolve into similar final states (Cecconi et al., 2012), to short time series, rCFD predicts the long-term evolution of recurrent systems. Since there is no need to solve the EOMs anymore, the simulation becomes computationally inexpensive and consequently several orders of magnitude faster (Pirker and Lichtenegger, 2018) compared to the corresponding CFD simulation.

There are two recurrence-based approaches for database generation and utilization, differing in the type of information that is employed and in the potential speed up. Flow-based rCFD (Lichtenegger et al., 2017) involves storing and later reusing the velocity field according to the method of analogs. This allows for the calculation of transport of other quantities such as energy or chemical species with a passive transport equation. If the velocity field is not required explicitly e.g. for the calculation of forces on a secondary phase (Hansson et al., 2024), transport-based rCFD (Pirker and Lichtenegger, 2018) can be employed, where instead the information transfer pattern between mesh cells is stored, time-extrapolated and reused for transport calculations with significantly larger time-steps than admissible in flow-based simulations.

This paper presents propagator theory (Lichtenegger, 2024) as a stringent mathematical framework for transport-based rCFD that overcomes certain limitations of the intuitive formulation in terms of discrete cell-to-cell maps employed in previous studies. This enabled significant advancements in picturing convection and diffusion for cases with large velocity ranges and cells of vastly different size as found in turbulent flows. We demonstrate these improvements for the simple case of a two-sided lid-driven cavity, but also highlight the applicability to the more demanding scenario of species transport in liquid steel inside a tundish.

Our paper is organized as follows. Section 2 introduces the propagator formulation and a simple approximation that allows to obtain information about displacement and diffusion of cell content. In Section 3, we explain how such approximated propagators can be discretized on meshes of realistic problem settings. Section 4 describes the above mentioned simulation cases to which we applied the rCFD algorithm, and Section 5 discusses the corresponding results. Finally, we draw conclusions and give an outlook on future developments of rCFD in Section 6.

2. Theoretical background

2.1. Propagator formulation

A passive species concentration $c(\boldsymbol{r},t)$ is governed by the transport equation

$$\frac{\partial c}{\partial t} + \nabla \cdot c\boldsymbol{u} = \nabla \cdot D\nabla c, \tag{1}$$

where $\boldsymbol{u}(\boldsymbol{r},t)$ is the velocity and $D(\boldsymbol{r},t)$ the diffusivity field. Due to its linear nature, the concentration at an arbitrarily long time after t_0 must depend linearly on its values at t_0, i.e.,

$$c(\boldsymbol{r},t) = \int d^3r'\, K_{cc}(\boldsymbol{r},\boldsymbol{r}';t,t_0)\, c(\boldsymbol{r}',t_0), \tag{2}$$

where integration extends over the full domain including the boundaries where the propagator contains Dirac delta functions and gradients thereof. Therefore, the propagator $K_{cc}(\boldsymbol{r},\boldsymbol{r}';t,t_0)$ satisfies

$$\frac{\partial}{\partial t}K_{cc} + \nabla \cdot K_{cc}\boldsymbol{u} = \nabla \cdot D\nabla K_{cc} \tag{3}$$

for each $\boldsymbol{r}'$, where derivatives are taken with respect to the first argument $\boldsymbol{r}$. Thus, it depends on the temporal evolution of the velocity and diffusivity fields $\boldsymbol{u}(\boldsymbol{r},t)$ and $D(\boldsymbol{r},t)$ from t_0 to t.

We point out the close relationship of the propagator and the Green's function

$$G_{cc}(\boldsymbol{r},\boldsymbol{r}';t,t_0) = \Theta(t-t_0)K_{cc}(\boldsymbol{r},\boldsymbol{r}';t,t_0) \tag{4}$$

where $\Theta(t)$ is the Heaviside step function. It can easily be confirmed that G_{cc} satisfies

$$\frac{\partial}{\partial t}G_{cc} + \nabla \cdot G_{cc}\boldsymbol{u} = \nabla \cdot D\nabla G_{cc} + \delta(t-t_0)\delta(\boldsymbol{r}-\boldsymbol{r}'). \tag{5}$$

In principle, Eq. (3) could be solved with the initial condition

$$K_{cc}(\boldsymbol{r},\boldsymbol{r}';t=t_0) = \delta(\boldsymbol{r}-\boldsymbol{r}') \tag{6}$$

and boundary conditions derived from those of the species field $c(\boldsymbol{r},t)$ (cf. Section 2.2). However, computing the full propagator which depends on six spatial coordinates is very expensive and requires substantial memory. Therefore, cheaper approximations are of interest. Without any diffusion, the propagator would simply be a delta function $\delta\left(\boldsymbol{r}' - \boldsymbol{r}^{(\mathrm{or})}(\boldsymbol{r})\right)$ that transports the concentration from a point of origin $\boldsymbol{r}^{(\mathrm{or})}(\boldsymbol{r})$ to $\boldsymbol{r}$ over an arbitrarily large time-step. However, diffusion broadens the delta function into a smeared-out peak around $\boldsymbol{r}^{(\mathrm{or})}(\boldsymbol{r})$. To proceed, one can either make further assumptions about the shape of the propagator or perform a low-order series expansion. In this work, we follow the latter approach and expand the convolution integral in Eq. (2) around the origin of influence, resulting in

$$\begin{aligned} c(\boldsymbol{r},t) = \int d^3r'\, K_{cc}(\boldsymbol{r},\boldsymbol{r}';t,t_0) \Big(& c(\boldsymbol{r}^{(\mathrm{or})},t_0) \\ & + (\boldsymbol{r}'-\boldsymbol{r}^{(\mathrm{or})})\cdot\nabla c(\boldsymbol{r}^{(\mathrm{or})},t_0) \\ & + \frac{1}{2}(\boldsymbol{r}'-\boldsymbol{r}^{(\mathrm{or})})(\boldsymbol{r}'-\boldsymbol{r}^{(\mathrm{or})}) : \nabla\nabla c(\boldsymbol{r}^{(\mathrm{or})},t_0) + \dots \Big) \\ = k^{(0\mathrm{c})}(\boldsymbol{r},t,t_0)\, c(\boldsymbol{r}^{(\mathrm{or})},t_0) & + \boldsymbol{k}^{(1\mathrm{c})}(\boldsymbol{r},t,t_0)\cdot\nabla c(\boldsymbol{r}^{(\mathrm{or})},t_0) \\ & + \frac{1}{2}\boldsymbol{k}^{(2\mathrm{c})}(\boldsymbol{r},t,t_0) : \nabla\nabla c(\boldsymbol{r}^{(\mathrm{or})},t_0) + \dots, \end{aligned} \tag{7}$$

where we have introduced the leading, centered moments

$$k^{(0\mathrm{c})}(\boldsymbol{r},t,t_0) \equiv \int d^3r'\, K_{cc}(\boldsymbol{r},\boldsymbol{r}';t,t_0), \tag{8}$$

$$\boldsymbol{k}^{(1\mathrm{c})}(\boldsymbol{r},t,t_0) \equiv \int d^3r'\, K_{cc}(\boldsymbol{r},\boldsymbol{r}';t,t_0)\,(\boldsymbol{r}'-\boldsymbol{r}^{(\mathrm{or})}), \tag{9}$$

$$\boldsymbol{k}^{(2\mathrm{c})}(\boldsymbol{r},t,t_0) \equiv \int d^3r'\, K_{cc}(\boldsymbol{r},\boldsymbol{r}';t,t_0)\,(\boldsymbol{r}'-\boldsymbol{r}^{(\mathrm{or})})(\boldsymbol{r}'-\boldsymbol{r}^{(\mathrm{or})}). \tag{10}$$

They are related to the plain, uncentered moments

$$k^{(0)}(\boldsymbol{r},t,t_0) \equiv \int d^3r'\, K_{cc}(\boldsymbol{r},\boldsymbol{r}';t,t_0), \tag{11}$$

$$\boldsymbol{k}^{(1)}(\boldsymbol{r},t,t_0) \equiv \int d^3r'\, K_{cc}(\boldsymbol{r},\boldsymbol{r}';t,t_0)\,\boldsymbol{r}', \tag{12}$$

$$\boldsymbol{k}^{(2)}(\boldsymbol{r},t,t_0) \equiv \int d^3r'\, K_{cc}(\boldsymbol{r},\boldsymbol{r}';t,t_0)\,\boldsymbol{r}'\boldsymbol{r}' \tag{13}$$

in a straightforward manner of

$$k^{(0c)}(\boldsymbol{r},t,t_0) = k^{(0)}(\boldsymbol{r},t,t_0), \tag{14}$$

$$\boldsymbol{k}^{(1c)}(\boldsymbol{r},t,t_0) = \boldsymbol{k}^{(1)}(\boldsymbol{r},t,t_0) - k^{(0)}(\boldsymbol{r},t,t_0)\boldsymbol{r}^{(or)}(\boldsymbol{r},t), \tag{15}$$

$$\begin{aligned}\boldsymbol{k}^{(2c)}(\boldsymbol{r},t,t_0) = {}& \boldsymbol{k}^{(2)}(\boldsymbol{r},t,t_0) - \boldsymbol{k}^{(1)}(\boldsymbol{r},t,t_0)\boldsymbol{r}^{(or)}(\boldsymbol{r},t) - \boldsymbol{r}^{(or)}(\boldsymbol{r},t)\boldsymbol{k}^{(1)}(\boldsymbol{r},t,t_0) \\ & + k^{(0)}(\boldsymbol{r},t,t_0)\boldsymbol{r}^{(or)}(\boldsymbol{r},t)\boldsymbol{r}^{(or)}(\boldsymbol{r},t).\end{aligned} \tag{16}$$

The uncentered moments can be computed directly from integrals of Eq. (3) without knowledge of the full propagator, i.e., by solving equations of the type

$$\frac{\partial}{\partial t}\boldsymbol{k}^{(m)} + \nabla\cdot\boldsymbol{u}\boldsymbol{k}^{(m)} = \nabla\cdot D\nabla\boldsymbol{k}^{(m)}, \tag{17}$$

for $m \leq 2$ with appropriate initial conditions

$$k^{(0)}(\boldsymbol{r},t=t_0) = 1, \tag{18}$$

$$\boldsymbol{k}^{(1)}(\boldsymbol{r},t=t_0) = \boldsymbol{r}, \tag{19}$$

$$\boldsymbol{k}^{(2)}(\boldsymbol{r},t=t_0) = \boldsymbol{r}\boldsymbol{r}. \tag{20}$$

Eq. (7) simplifies particularly if one defines $\boldsymbol{r}^{(or)}$ such that the first-order, centered moment $\boldsymbol{k}^{(1c)}$ vanishes. Employing $k^{(0)} = 1$ (which holds for or incompressible single-phase flows because it solves Eq. (17) under the initial condition Eq. (18)), this translates to

$$\boldsymbol{r}^{(or)} = \boldsymbol{k}^{(1)}. \tag{21}$$

If one further and neglects terms beyond the second order, Eq. (7) reduces to

$$c(\boldsymbol{r},t) \approx c(\boldsymbol{r}^{(or)},t_0) + \frac{1}{2}\boldsymbol{k}^{(2c)}(\boldsymbol{r},t) : \nabla\nabla c(\boldsymbol{r}^{(or)},t_0), \tag{22}$$

where we have used $k^{(0c)} = 1$. Higher-order terms can be assumed to play a negligible role if the concentration field varies over distances similar to or larger than those of the propagator. However, small-scale structures far below the width of the propagator get lost. Since the width grows with the prediction step size, the latter determines down to which spatial resolution large-step species transport according to Eq. (22) remains accurate.

Because changes in the total amount of species are solely due to influx and outflux, the above expression can be cast into the conservative form

$$c(\boldsymbol{r},t) = c(\boldsymbol{r}^{(or)},t_0) + \nabla\cdot(t-t_0)\boldsymbol{D}_{\text{eff}}(\boldsymbol{r},t,t_0)\cdot\nabla c(\boldsymbol{r}^{(or)},t_0), \tag{23}$$

with

$$\boldsymbol{D}_{\text{eff}}(\boldsymbol{r},t,t_0) \equiv \frac{1}{2(t-t_0)}\boldsymbol{k}^{(2c)}(\boldsymbol{r},t). \tag{24}$$

Notably, the effective dispersion tensor $\boldsymbol{D}_{\text{eff}}(\boldsymbol{r},t,t_0)$ is not a function of the diffusivity $D(\boldsymbol{r},t)$ alone because it includes contributions from gradients of the velocity field. However, for uniform velocity and diffusivity, it can be shown that $\boldsymbol{k}^{(2c)}(\boldsymbol{r},t,t_0) = 2(t-t_0)D\boldsymbol{1}$, so that the effective dispersion tensor reduces to $\boldsymbol{D}_{\text{eff}} = D\boldsymbol{1}$.

In the general case of non-uniform velocity fields, $\boldsymbol{D}_{\text{eff}}$ is anisotropic. In the present work, we restrict ourselves to an isotropic approximation obtained from the trace of the full, anisotropic dispersion tensor (cf. Section 5.1.1).

2.2. Boundary conditions for a propagator

Boundary conditions for species propagators and consequently for their moments can be obtained from the conditions of the concentration field. In the following, we restrict ourselves to the important cases of Dirichlet and Neumann conditions, but state that more general cases can be handled in a similar fashion.

At Dirichlet boundaries, e.g. an inlet with a prescribed concentration value, one has

$$c(\boldsymbol{r}_\text{b},t) = c(\boldsymbol{r}_\text{b},t_0), \tag{25}$$

if variations between t_0 and t are ignored. According to Eq. (2), this translates into

$$\int d^3r'\, K_{cc}(\boldsymbol{r}_\text{b},\boldsymbol{r}';t,t_0)\, c(\boldsymbol{r}',t_0) = \int d^3r'\, \delta(\boldsymbol{r}_\text{b}-\boldsymbol{r}')\, c(\boldsymbol{r}',t_0). \tag{26}$$

Because the propagator K_{cc} must satisfy Eq. (26) for any concentration field $c(\boldsymbol{r}',t_0)$, it has to do so both for the integral and the differential versions of the equation. This yields

$$K_{cc}(\boldsymbol{r}_\text{b},\boldsymbol{r}';t,t_0) = \delta(\boldsymbol{r}_\text{b}-\boldsymbol{r}'). \tag{27}$$

Application to Eqs. (11)–(13) gives

$$k^{(0)}(\boldsymbol{r}_\text{b},t) = \int d^3r'\, \delta(\boldsymbol{r}_\text{b}-\boldsymbol{r}') = 1 \tag{28}$$

$$\boldsymbol{k}^{(1)}(\boldsymbol{r}_\text{b},t) = \int d^3r'\, \delta(\boldsymbol{r}_\text{b}-\boldsymbol{r}')\boldsymbol{r}' = \boldsymbol{r}_\text{b} \tag{29}$$

$$\boldsymbol{k}^{(2)}(\boldsymbol{r}_\text{b},t) = \int d^3r'\, \delta(\boldsymbol{r}_\text{b}-\boldsymbol{r}')\boldsymbol{r}'\boldsymbol{r}' = \boldsymbol{r}_\text{b}\boldsymbol{r}_\text{b} \tag{30}$$

for the zeroth, first and second moment at Dirichlet boundary points $\boldsymbol{r}_\text{b}$.

Neumann boundary conditions arise e.g. at solid walls with a prescribed, fixed normal-gradient of $c(\boldsymbol{r},t)$, i.e.

$$\boldsymbol{n}\cdot\nabla c(\boldsymbol{r}_\text{b},t) = \boldsymbol{n}\cdot\nabla c(\boldsymbol{r}_\text{b},t_0) \tag{31}$$

for every wall point $\boldsymbol{r}_\text{b}$ with normal unit vector $\boldsymbol{n}(\boldsymbol{r}_\text{b})$. Arguing along the same lines as above, one finds

$$\begin{aligned}&\int d^3r'\, \boldsymbol{n}\cdot\nabla K_{cc}(\boldsymbol{r}_\text{b},\boldsymbol{r}';t,t_0)\, c(\boldsymbol{r}',t_0) \\ &= \int d^3r'\, \boldsymbol{n}\cdot\nabla\delta(\boldsymbol{r}_\text{b}-\boldsymbol{r}')\, c(\boldsymbol{r}',t_0).\end{aligned} \tag{32}$$

It follows that

$$\boldsymbol{n}\cdot\nabla K_{cc}(\boldsymbol{r}_\text{b},\boldsymbol{r}';t,t_0) = \boldsymbol{n}\cdot\nabla\delta(\boldsymbol{r}_\text{b}-\boldsymbol{r}') \tag{33}$$

and therefore

$$\boldsymbol{n}\cdot\nabla k^{(0)}(\boldsymbol{r}_\text{b},t) = \int d^3r'\, \boldsymbol{n}\cdot\nabla\delta(\boldsymbol{r}_\text{b}-\boldsymbol{r}') = 0, \tag{34}$$

$$\boldsymbol{n}\cdot\nabla\boldsymbol{k}^{(1)}(\boldsymbol{r}_\text{b},t) = \int d^3r'\, \boldsymbol{n}\cdot\nabla\delta(\boldsymbol{r}_\text{b}-\boldsymbol{r}')\boldsymbol{r}' = \boldsymbol{n}, \tag{35}$$

$$\boldsymbol{n}\cdot\nabla\boldsymbol{k}^{(2)}(\boldsymbol{r}_\text{b},t) = \int d^3r'\, \boldsymbol{n}\cdot\nabla\delta(\boldsymbol{r}_\text{b}-\boldsymbol{r}')\boldsymbol{r}'\boldsymbol{r}' = \boldsymbol{n}\boldsymbol{r}_\text{b} + \boldsymbol{r}_\text{b}\boldsymbol{n}, \tag{36}$$

for the zeroth, first and second moment at Neumann boundaries.

2.3. Conceptual difference to CFD algorithms

Standard CFD algorithms operate on sparse matrices, which keeps computations within each iteration manageable, but also implies small time-steps. In contrast, the propagator formulation Eq. (2) permits almost arbitrarily large steps but the number of non-zero entries in the matrices obtained upon discretization grows with the step size. To achieve real-time capable simulations, it takes approximations of the type Eq. (23) that simplify calculations to sparse matrices again.

While not directly related, our approach shares certain properties with the boundary element method (Brebbia and Partridge, 2012; Kirkup, 2019) in the sense that a linear partial differential equation (PDE) is transformed into integral form. However, the propagator collects information over the whole domain via Eq. (2), whereas the boundary element method employs a boundary integral form so that one deals with smaller but densely populated matrices instead of large, sparse ones. To this end, it requires knowledge of the PDEs Green's function.

In this regard, our approach seems more general because the propagator (its close relationship with the Green's function given by Eq. (4)) is computed from data of a CFD simulation and need not be known a priori.

3. Implementation

In this section, we explain aspects of the implementation of rCFD. We contrast two formulations how the information transfer pattern of a passive transport equation is obtained on a finite-volume mesh and address possible discretization errors.

3.1. The rCFD workflow

The overall workflow of rCFD is split into two parts. First, convection and diffusion data has to be extracted from an existing CFD simulation by adding either tracer particles (Section 3.2.1) or passive scalar fields corresponding to propagator moments (Section 3.2.2). An example of how this works can be found in Appendix A. After several CFD simulation time steps, the data are extracted and the tracer locations / fields are reset. The velocity data is immediately translated to cell-to-cell-shifts (C2Cs) (see Section 3.2), while the diffusivity is stored directly as a scalar field. This extraction process continues until a database with enough instances of different flow configurations is obtained to sample the underlying attractor sufficiently well.

Once this has been achieved, the rCFD simulation itself can be started by loading the database into RAM. In the spirit of a nearest-neighbor approach, the evolution of a flow field is approximated with that of the most similar state in the database. To transport species on top of this field, the corresponding C2Cs and diffusivity are employed. Upon iteration, long time series can be created.

The whole rCFD process has been implemented for Fluent as a set of User Defined Functions as well as Scheme files for workflow automation and is available in our public repository (rCF, n.d.).

3.2. From velocity fields to cell-to-cell shifts

For rCFD to work correctly, one has to translate a velocity field into large-step transport of information. In contrast to an instantaneous velocity field, such an information shift vector as illustrated in Fig. 1 can be calculated for an almost arbitrary time-step size because only the origin and destination points are important but not the path taken in between. This problem can be solved in a forward or a backward formulation. The forward method which was used in previous rCFD studies involves using massless tracer particles that are convected with the flow to determine the destination point $\boldsymbol{r}_i^{(\mathrm{dest})}$ (in general not a cell center) where information is transported to from the center of each cell i. The backward method employs the propagator moments formalism introduced in the preceding section. One solves passive scalar transport equations for each cell's coordinates to compute that location $\boldsymbol{r}_i^{(\mathrm{or})}$ (usually not the center of a cell) from where information for each cell i has originated.

3.2.1. Tracer particles

In previous rCFD investigations, massless tracer particles starting at each cell center were used to determine shift vectors from pathlines. However, Fig. 1a indicates how small changes in the initial position can result in significant variations in the final position, which necessitates a very accurate calculation of the trajectories.

As a major disadvantage, using one tracer per cell does not provide direct information on dispersion processes induced by diffusion and velocity gradients. Instead, one would have to track many tracers per cell each subject to additional stochastic forcing and calculate the center and variance of their locations. This can quickly cause significant computational costs and a cumbersome numerical setup.

3.2.2. Transport of propagator moments

To avoid these difficulties, shift vectors can also be computed based on the origin of influence $\boldsymbol{r}^{(\mathrm{or})}$ by transporting the cell position information according to a passive transport equation. As discussed in Section 2, this is equivalent to obtaining the first moment $\boldsymbol{k}^{(1)}(\boldsymbol{r},t)$ of the propagator from Eq. (15). After initialization with the corresponding cell position, i.e. $\boldsymbol{k}^{(1)}(\boldsymbol{r},0)=\boldsymbol{r}$, this vector field is propagated along with the classical CFD simulation. After a time Δt, each cell i contains information on where its current content has originated from, $\boldsymbol{r}_i^{(\mathrm{or})}$. The resulting displacement is shown in Fig. 1b as the dotted arrow. In contrast to tracer particles, which may start and end at arbitrary points in space, the shift vector calculated via the propagator always points to the center of its destination cell, with one displacement being calculated for each cell. Even for small velocities that the resulting displacement vector originates from and points to the same cell, a non-zero displacement length is obtained because the position information from neighboring cells partially flows into the current one.

Notably, this approach can easily provide information on dispersive processes via computation of the second moment $\boldsymbol{k}^{(2)}(\boldsymbol{r},t)$.

3.3. Spatial discretization of the propagator

A continuous description of the propagator allows for the exact calculation of movement for every point in space as a function of time.

However, when space is discretized into finite volumes, the shift of cell content over a given time-step Δt cannot be precisely mapped onto the underlying discretization. There are three primary reasons for this, which are also depicted in Fig. 2 from left to right:

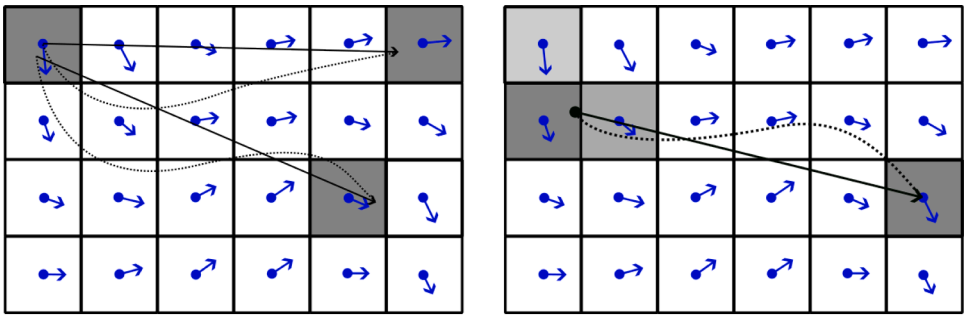

Fig. 1. Process of obtaining shift vectors. The displacements of cell content over multiple time-steps (dotted line) in a given velocity field (blue arrows) are replaced by shift vectors (solid line), which are then approximated by C2Cs (grey cells). The displacements are calculated using (a) tracers and (b) passive scalar fields. In (b), there are multiple C2Cs with different origin cells and the same destination cell whose weights are indicated by their shade of grey.

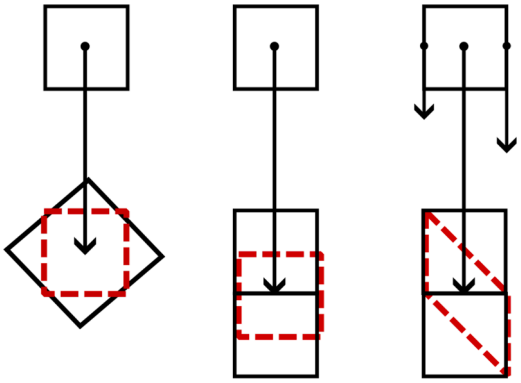

Fig. 2. Illustration depicting the errors introduced when using a displacement vector to describe convection.

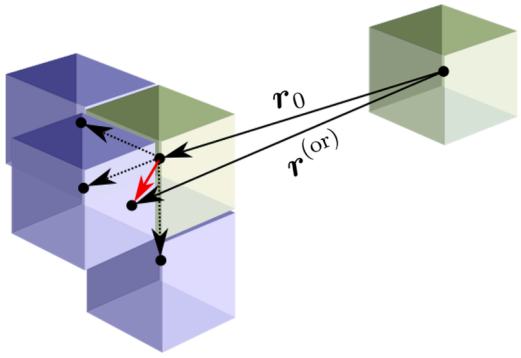

Fig. 3. Illustration of obtaining cell-to-cell shifts from a displacement vector. The dotted arrows form a base to describe the relative position (red) of $\boldsymbol{r}^{(\mathrm{or})}$ to the cell center.

1. For non-uniform meshes, errors arise for shifts between cells of different size or shape because of the inability to fit the content of the origin into the target cell. Hence, local conservation of mass cannot be guaranteed. However, a global mass balance can be introduced to ensure at least global conservation (see Section 3.3.1).
2. In general, the exact shift of each cell center for a given time-step Δt will not align with the center of another cell, necessitating the use of multiple cells for an accurate description via interpolation (see Section 3.3.2). While this is essential to picture transport correctly in the presence of both high- and low-velocity regions, it also introduces an error via additional, artificial diffusion.
3. Due to velocity gradients, the information contained within a cell volume becomes deformed over time. This deformation will eventually cause its content to disperse across multiple cells. The combined effect of diffusion and velocity gradients can be accounted for with an additional Laplacian term to the shift operation as specified by Eq. (23). The effective diffusion tensor $\boldsymbol{D}_{\mathrm{eff}}$ is calculated from the first and second propagator moments via Eq. (24).

3.3.1. Global species conservation

As mentioned above, cell-to-cell shifts do not automatically conserve species or any other transported quantity. Instead of local conservation, a global balance is introduced that controls the total amount inside a domain, which is known with high accuracy from the fluxes through the boundaries. Such balances can also be applied to sub-domains of sufficient size, which are then handled individually. In both cases the integral fluxes can be readily retrieved from the underlying CFD simulations.

If the total species amount in a sub-domain obtained through the cell-to-cell shifts differs from the target value calculated via the global balance, one has to insert or remove species. Since cell-to-cell shifts do not create or lose species in parts of the domain where no concentration gradients occur, it is reasonable to make adaptions proportional to such gradients. More specifically, we modify the diffusion operation which moves species between neighboring cells proportionally to the concentration gradient (Pirker and Lichtenegger, 2018). If there is too little species in a (sub-)domain, we add slightly more to a cell than we remove from its neighbor, and if there is too much, we remove somewhat more.

Global species conservation is enforced at the cost of an additional diffusion mechanism. Consequently, the applicability of rCFD is limited for diffusion-sensitive cases. However, one could enforce mass conservation already at the level of the rCFD database by appropriately scaling the weights connected to the shift operations (cf. Section 3.3.2). In this case, no on-the-fly corrections in terms of a global balance criterion would be needed because the cell-to-cell shifts (C2Cs) would not cause loss/creation of species anymore. Hence, improving the database quality by making it more mass conservative would decrease the impact of artificial diffusion. However, it remains to be seen in future work how such adaptions on the shift pattern can be constructed in a minimally invasive way.

3.3.2. Interpolation of cell-to-cell shifts

As discussed above, the shift vector obtained from tracer particles does not point from one cell center to another but rather to an arbitrary location in the domain. The first propagator moment points to a cell center but does not originate from one. An example for the latter case is shown in Fig. 3, where the origin of influence $\boldsymbol{r}^{(\mathrm{or})}$ of a cell i does not coincide with any cell center. However, the corresponding shift, i.e. the difference of the center of i and $\boldsymbol{r}^{(\mathrm{or})}$, can be represented as a superposition of center-to-center vectors. For the sake of simplicity, we choose a coordinate system located at the center of i. Let $\boldsymbol{r}_0$ be the center of that cell that contains $\boldsymbol{r}^{(\mathrm{or})}$ and $\boldsymbol{r}_{1,2,3}$ the centers of three neighbor cells such that

$$\boldsymbol{r}^{(\mathrm{or})} = \boldsymbol{r}_0 + \sum_{j\geq 1} w_j \left(\boldsymbol{r}_j - \boldsymbol{r}_0\right). \tag{37}$$

If the three neighbor cells are selected such that $\boldsymbol{r}^{(\mathrm{or})}$ lies within the tetrahedron formed by the four centers $\boldsymbol{r}_{0,1,2,3}$, the coefficients $w_0 \equiv 1 - w_1 - w_2 - w_3$, w_1, w_2 and w_3 each satisfy $0 \leq w_j \leq 1$. Hence, they can be used as weights for the contributions of their cells to the transported species concentration.

Calculating the standard deviation

$$\sigma = \sqrt{\frac{4}{3}\sum_{j\geq 0} w_i \left|\boldsymbol{r}_i - \boldsymbol{r}^{(\mathrm{or})}\right|^2}. \tag{38}$$

of this shift operation is straightforward because all the necessary shifts and weights are already defined and sum up to 1.0. It can be interpreted as an additional diffusivity

$$D_{\mathrm{C2C}} = \frac{\sigma^2}{2\Delta t_{\mathrm{rCFD}}}, \tag{39}$$

which emerges from the way the C2Cs operate, and can be explicitly accounted for (cf. Section 5.1.1). This correction mechanism could also be used in connection with coarser grids when fine-grained CFD data are mapped onto a coarse-grained representation to further speed up simulations.

3.3.3. Shifts originating from outside the domain

Due to numerical discretization errors, it is possible for the origin of a shift vector to lie outside the domain, particularly when boundary layers are not properly resolved. In such cases, the interpolation method described in Section 3.3.2 becomes inapplicable because no set of four cells can form a tetrahedron enclosing the origin point. Instead, we project it onto the boundary and interpolate between face centers.

4. Simulation setup

To demonstrate the improvements of the rCFD code with regard to a detailed representation of diffusion, we studied two simulation cases. The first case was a cuboid cavity with two moving walls, and the second a simplified tundish model similar to the setup described by Sheng (2020). For the data generating finite-volume CFD simulations, ANSYS Fluent 2022 R2 was used. Both test cases were regarded as incompressible and were treated in a transient fashion. For both geometries the meshes consisted of regular grids of 125,000 and 650,000 hexahedral cells, respectively.

4.1. Lid-driven cavity

As depicted in Fig. 4, we simulated the flow inside a cuboid cavity with an edge length of 1.0 m. The top and bottom walls moved with a velocity of $v_{\text{lid}} = 0.1\,\text{m}\,\text{s}^{-1}$ in opposing x-directions, and a Cartesian mesh with 125,000 cells was employed. The fluid inside the cavity was water with a density of $\rho = 1000\,\text{kg}\,\text{m}^{-3}$ and a viscosity of $\mu = 0.001\,\text{kg}\,\text{m}^{-1}\,\text{s}^{-1}$. These parameters corresponded to a Reynolds number $Re = 100,000$, which we accounted for with RANS simulations employing the SST-k-ω model (Menter, 1994) for turbulence modeling.

The fluid EOMs in pressure-velocity formulation were solved with the PISO algorithm (Ferziger and Perić, 2002). Pressure was computed on a staggered grid via Fluent's PRESTO! (Pressure Staggering Option) approach, which is similar to that described by Patankar (2018) but works for irregular grids. The spatial discretization was achieved via a least-squares method for the calculation of gradients (Mavriplis, 2003) and second-order upwind scheme for momentum, turbulent kinetic energy k and specific dissipation rate ω. The transient term was dealt with in implicit first-order approximation.

To provide a more comprehensive comparison between CFD and rCFD, two variations of passive scalar behavior were simulated: one with negligible diffusivity ($D \approx 0$) and the other with diffusivity set to $D = D_{\text{phys}} = \mu_{\text{eff}} Sc \approx \mu_{\text{eff}}$, where $\mu_{\text{eff}} = \mu_0 + \mu_{\text{turb}}$ is the effective viscosity consisting of a laminar and a turbulent contribution. For the sake of simplicity, the Schmidt number was set to $Sc = 1.0$. The fields for all cases were initialized with 0 for $x \leq 0$ and 1 for $x > 0$. This case setup with and without diffusion allowed for a separate analysis of both diffusion and convection effects.

4.2. Tundish

For the tundish simulation, the geometry from Sheng's study (Sheng, 2020) was utilized at a smaller scale of 1 : 2 with a mesh of 650,000 cells. The fluid in the tundish was molten steel with simulation parameters listed in Table 1. Given that the jet at the inlet was the primary source of turbulence, the Reynolds number was estimated at $Re \approx 220,000$. Similarly to the lid-driven cavity case, we chose a Schmidt number $Sc = 1.0$ which had also been used in other tundish simulations (Jha and Dash, 2004).

Table 1
Basic physical and simulation parameters of liquid steel in the tundish simulation case.

Parameter	Value	Units
density	7000	$\text{kg}\,\text{m}^{-3}$
viscosity	0.0019	$\text{kg}\,\text{m}^{-1}\,\text{s}^{-1}$
CFD time-step	0.01	s
rCFD time-step	0.05	s
inlet velocity	1.0	$\text{m}\,\text{s}^{-1}$
inlet diameter	0.06	m
outlet pressure	101,325	Pa

As a compromise between computational costs and accuracy of the rCFD database and validation data, we performed detached eddy simulations (DES) (Menter et al., 2003), an extension of LES where a RANS model (in this case, SST-k-ω) is utilized near boundaries. This approach avoids the complexity of addressing LES behavior with a high mesh resolution close to walls, allowing for a focus on the overall flow dynamics.

We used the SIMPLE algorithm (Ferziger and Perić, 2002) to deal with pressure-velocity coupling. Gradients were computed via a least-squares method (Mavriplis, 2003). Second-order upwind discretization was used for turbulent kinetic energy k and specific dissipation rate ω. A second-order method was used for the pressure. For the momentum bounded central differencing, a combination of standard central differencing and a first-order upwind scheme, was used. Temporal integration was carried out in second-order, implicit formulation.

We studied the transport behavior inside the tundish and evaluated its residence time and the spatial distribution of impurities such as non-metallic inclusions. To perform a residence time simulation, a different species of molten steel with concentration c was introduced via the inlet for a duration of 1 s and the simulation continued for a total time of 150 s. Regarding the impurity distribution, we restricted ourselves to small inclusion sizes where buoyancy-driven drift could be neglected. At the steel-air-interface, they were simply removed via the sink term

$$S(\boldsymbol{r}, t) = \begin{cases} -\frac{c}{\Delta t}, & \text{if } \boldsymbol{r} \in \text{slag-layer} \\ 0, & \text{otherwise} \end{cases} \tag{40}$$

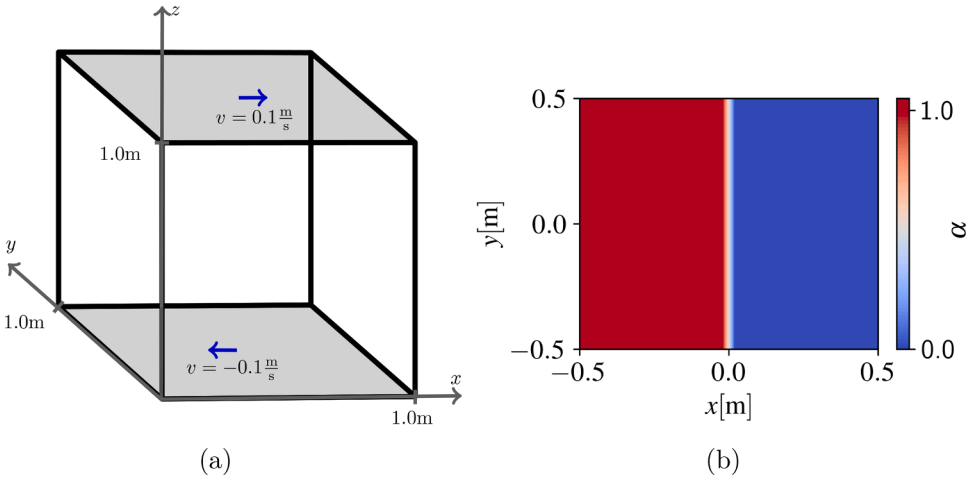

Fig. 4. Setup of the lid-driven cavity case. (a) Schematic of a cuboid, two-sided lid-driven cavity where the top and bottom walls move with a velocity of $\boldsymbol{v} = 0.1\,\text{m}\,\text{s}^{-1}$ in opposite x-directions. (b) Initial field of the passive scalar in the CFD simulation in the $y = 0$-plane.

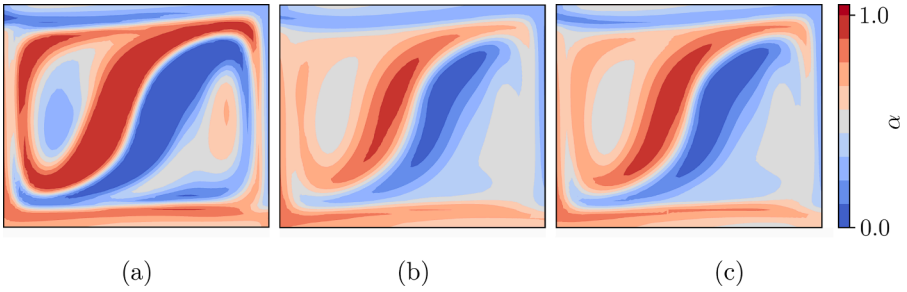

Fig. 5. Passive scalar mixing of the lid-driven cavity without diffusion. Results are shown after 200 s of simulation time with multiple configurations on the $y = 0$-plane with initial conditions as seen in Fig. 4b. The shown cases are (a) CFD with $D = 0$ and $\Delta t_{\text{CFD}} = 0.1\,\text{s}$, (b) rCFD with $\Delta t_{\text{rCFD}} = 1.0\,\text{s}$, and (c) rCFD with $\Delta t_{\text{rCFD}} = 2.0\,\text{s}$.

that mimicked the absorption into a slag layer. From the standard CFD perspective, the presence of this term generalized Eq. (1) to

$$\frac{\partial}{\partial t} c + \nabla \cdot c\boldsymbol{u} = \nabla \cdot D \nabla c + S, \tag{41}$$

while with the propagator approach, we applied it between steps.

5. Results

5.1. Lid-driven cavity

5.1.1. The impact of numerical diffusion

To assess the magnitude of the numerical diffusion that comes with the propagator formalism compared to standard CFD calculations, we first simulated the lid-driven cavity for 200 s without diffusion. Fig. 5 shows CFD results and those obtained with rCFD using two different time-steps $\Delta t_{\text{rCFD}} = 1.0\,\text{s}$ and $\Delta t_{\text{rCFD}} = 2.0\,\text{s}$. It can be immediately observed that the rCFD cases exhibited more diffusion which was caused by the interpolation of the displacement vector via C2Cs as explained in Section 3.3.2. Consequently, the results in Fig. 5c are more accurate than those in Fig. 5b because the number of interpolations and D_{C2C} was approximately halved. Of course, the relationship between D_{C2C} and $\Delta t_{\text{rCFD}}^{-1}$ is not exactly proportional, as D_{C2C} is not only dependent on the amount of interpolations but also on whether each interpolation distributes the shift weights among multiple cells equally or mainly on single cells.

To better illustrate this non-linearity, one can consider a regular, Cartesian grid with a distance d between cell centers. A range of possible values for D_{C2C} can be calculated via Eq. (39). D_{C2C} for a specific cell can range from 0, when data originates from only one origin cell, to $D_{\text{C2C}} = \frac{3d}{8\Delta t_{\text{rCFD}}}$ when it originates from four origin cells with equal weights $w_i = \frac{1}{4}$ as shown in Fig. 3. Very small changes in the displacement vector can lead to big changes for the weights w_i and subsequently D_{C2C}.

The impact of physical diffusion can be seen in Fig. 6. For the rCFD case shown in Fig. 6b, we used the same physical diffusion $D = D_{\text{phys}}$ as in the CFD calculation Fig. 6a. However, this led to insufficient mixing, and the result became even worse when D_{C2C} was subtracted as shown in Fig. 6c. What was missing in these rCFD simulations was the dispersion of the flow. Cell content should not only be displaced but also deformed due to velocity gradients. Using the propagator approach discussed in Section 2.1 gives the total, anisotropic diffusivity Eq. (24). For the sake of simplicity, we restricted ourselves to an isotropic approximation $\frac{1}{3}\text{Tr}\left[\boldsymbol{D}_{\text{eff}}\right]$. While calculations with the full, anisotropic dispersion tensor would not have been more expensive, it would take an anisotropic correction of numerical diffusion, i.e. a generalization of Eq. (39), to account for anisotropy effects consistently.

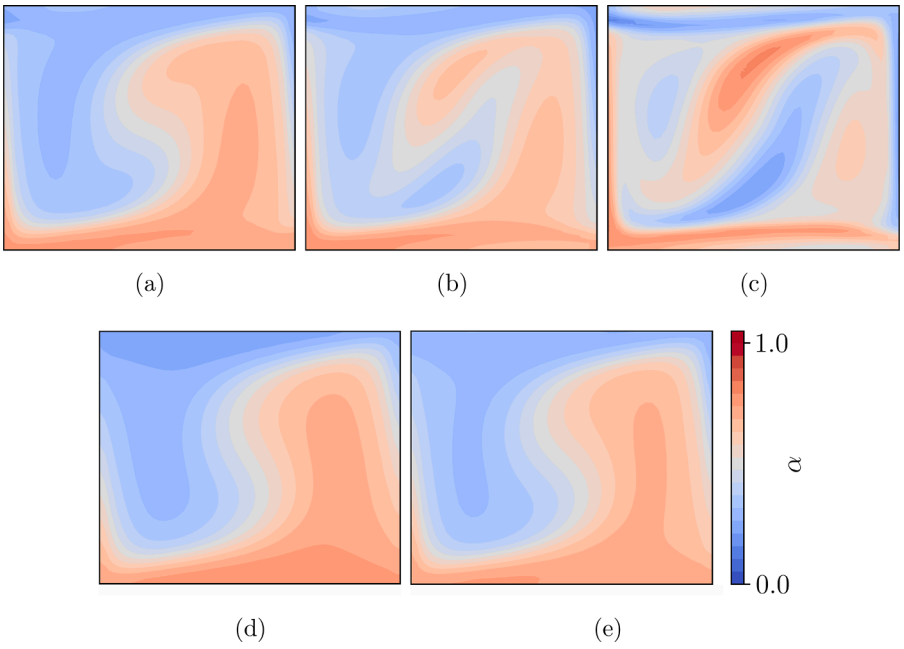

Fig. 6. Passive scalar mixing of the lid-driven cavity with diffusion. Results are shown after 200 s of simulation time with multiple configurations on the $y = 0$-plane with initial conditions as seen in Fig. 4b. (a) shows the CFD reference case with $D = D_{\text{phys}}$, (b) rCFD with $D = D_{\text{phys}}$, (c) rCFD with $D = D_{\text{phys}} - D_{\text{C2C}}$, (d) rCFD with $D = \text{Tr}\left[\boldsymbol{D}_{\text{eff}}\right]/3$, and (e) rCFD with $D = \text{Tr}\left[\boldsymbol{D}_{\text{eff}}\right]/3 - D_{\text{C2C}}$.

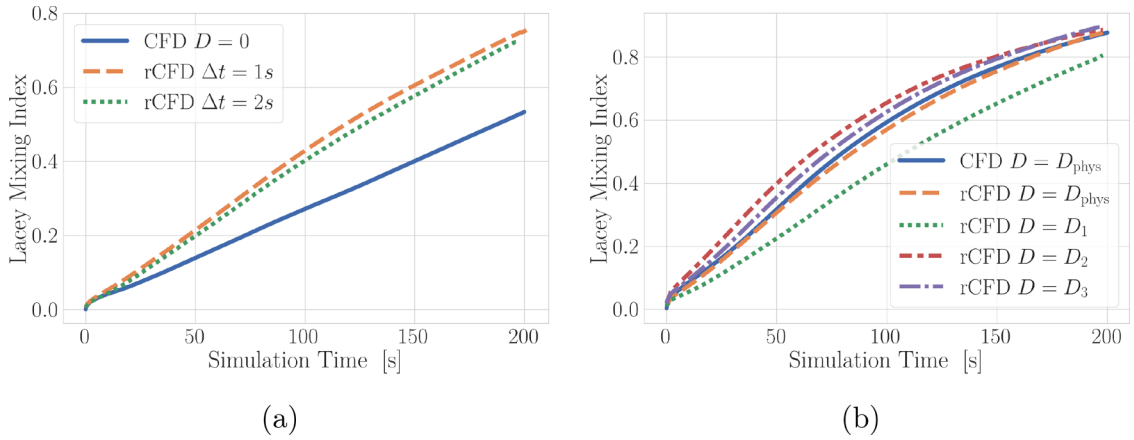

Fig. 7. Lacey mixing index for the lid-driven cavity. Results are displayed (a) without and (b) with diffusion. In (b) the abbreviations $D_1 = D_{\text{phys}} - D_{\text{C2C}}$, $D_2 = \text{Tr}[\boldsymbol{D}_{\text{eff}}]/3$ and $D_3 = \text{Tr}[\boldsymbol{D}_{\text{eff}}]/3 - D_{\text{C2C}}$ are used.

The result, shown in Fig. 6d, closely matched the CFD case from Fig. 6a. The agreement improved further when D_{C2C} was additionally subtracted as seen in Fig. 6e.

5.1.2. Mixing behavior

The diffusive behavior of CFD and rCFD can also be compared using the Lacey mixing index (Lacey, 1954). It quantifies the mixing state between two sets of particles or phases with values ranging from 0 (completely unmixed or initial state) to 1 (fully mixed or random state). Over time, the index value progressively increases towards the mixed state.

In Fig. 7a, the Lacey mixing index is shown for the three simulation cases of the lid-driven cavity without diffusion. As previously discussed, the rCFD cases demonstrated higher diffusivity than the CFD case, which reflected in their higher mixing index values. Furthermore, the mixing index for the rCFD case with $\Delta t = 2.0\,\text{s}$ was lower than that for the case with $\Delta t = 1.0\,\text{s}$, since D_{C2C} is proportional to Δt.

In Fig. 7b, the Lacey mixing index is displayed for the five simulation variants of the lid-driven cavity with diffusion. With the exception of the rCFD case where $D = D_{\text{phys}} - D_{\text{C2C}}$, the others appear to closely match the CFD case, with the best alignment seen for $D = D_{\text{phys}}$. However, this result can be misleading, as the Lacey mixing index represents an average mixing level over the entire volume. There were regions where the diffusivity was either too high or too low due to the exclusion of D_{C2C} or by using D_{phys} instead of $\text{Tr}[\boldsymbol{D}_{\text{eff}}]/3$, resulting in a misleadingly close match to the CFD simulation's average mixing behavior. Conversely, the cases where $D_2 = \text{Tr}[\boldsymbol{D}_{\text{eff}}]/3$ and $D_3 = \text{Tr}[\boldsymbol{D}_{\text{eff}}]/3 - D_{\text{C2C}}$ tended to slightly overestimate the mixing. This observation was likely due to cells where $\text{Tr}[\boldsymbol{D}_{\text{eff}}]/3 < D_{\text{C2C}}$, which made it impossible to fully compensate for D_{C2C} using the other diffusivity terms as the lowest possible diffusivity in a cell was $D = 0$. After 200 s, the Lacey mixing index for the CFD case amounted to 0.877, while our closest representation with $D = D_3$ was 0.894.

5.2. Tundish

5.2.1. Residence time

To demonstrate that rCFD can approximate CFD simulations in complex, industrially relevant geometries at much lower numerical costs, we carried out a residence time simulation for a tundish geometry downscaled from industrial dimensions. We injected a pulse with a duration of 1 s into the system and monitored the total amount of species over time. The rCFD database consisted of a 20 s CFD simulation, which was then extrapolated to longer durations of up to 200 s.

As shown in Fig. 8a and b, rCFD successfully captured the flow throughout the entire volume. It accurately reproduced the fast jet at the inlet, the complex motion through the weir and the dam, and the slow-moving flow between the dam and the outlet. Some regions showed different instantaneous concentration patterns between the rCFD and CFD simulations, particularly behind the dam. These discrepancies can be attributed to diverging trajectories between the rCFD database and long-term CFD simulations due to turbulence effects.

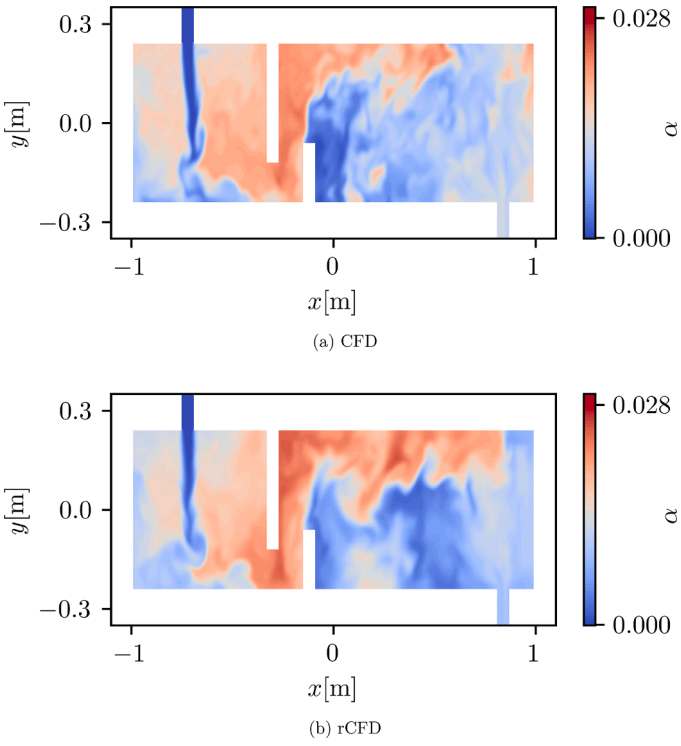

Fig. 8. Species concentrations in the tundish. (a) CFD and (b) rCFD simulation results are shown for the $y = 0$-plane at $t = 25$ s, i.e. 24 s after the inlet pulse had stopped.

However, the overall agreement was still satisfactory as evidenced by Fig. 9 which illustrates the time evolution of volume fractions in various regions. In the beginning the total amount of species decreased slightly faster in the CFD case than for rCFD. This behavior reflected in the outlet volume fraction shown in Fig. 9b, where the onset of non-zero volume fraction occurred a little bit earlier in CFD than in rCFD. This can be quantified using a minimal residence time, which we

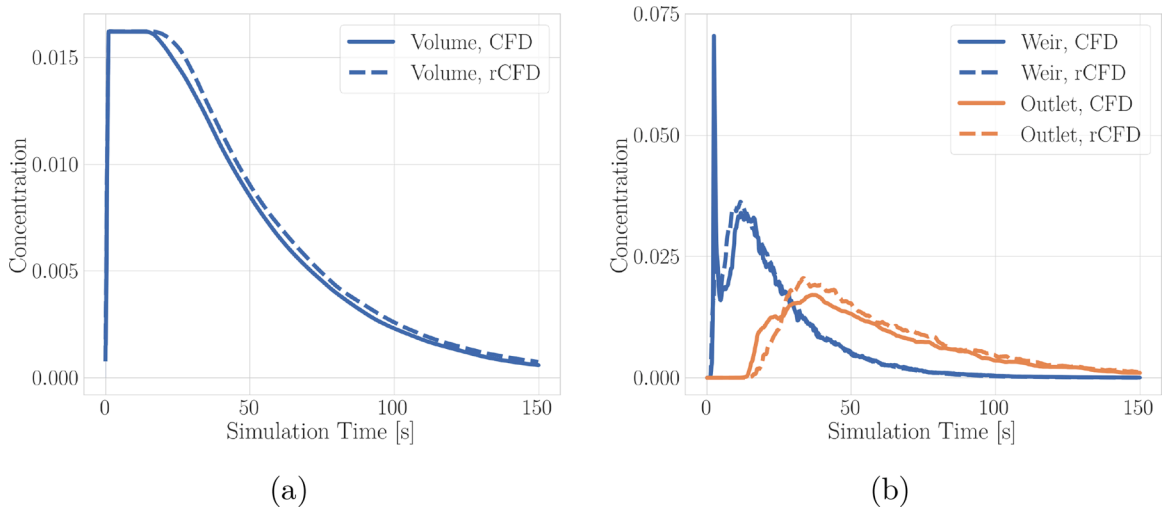

Fig. 9. Species evolution in the tundish. Concentrations after an initial pulse are shown (a) integrated over the whole volume and (b) beneath the weir and at the outlet.

defined as that duration until the concentration at the outlet reached 1 % of its maximum value. This happened after 13.6 s for CFD and after 15.1 s for rCFD. Overall, rCFD replicated the behavior of fluid flow through the entire tundish very well. We ascribe the deviations between the two simulations to steep concentration gradients in cells where D_{C2C} exceeded the total diffusivity $\mathrm{Tr}\left[\boldsymbol{D}_{\mathrm{eff}}\right]/3$ of the rCFD simulation, which remains a source of inaccuracy. Large and narrow peaks of volume fraction were automatically smeared out in the rCFD simulation. However, as steep gradients vanished, the average flow was replicated well.

5.2.2. Inclusion removal

The cleaning of steel via the separation of inclusion particles towards the top slag layer was simulated as an impurity concentration field with a sink in the uppermost cell layer. This field took 150 s to reach an equilibrium state in the CFD simulation which was continued for another 150 s. For rCFD the same process durations were used and the database was generated based on 20 s of the CFD simulation. In both cases, the simulation was initialized with steel with a maximum concentration of inclusions which were then cleaned at the slag layer until an equilibrium state was reached.

The time-averaged steel grade fields for the $y = 0$-plane are shown in Fig. 10, where averaging started after the flow had equilibrated. They were similar with only minor deviations. Above the outlet, the steel without inclusions had a slightly higher concentration in rCFD than in CFD. This can be attributed to the jumps between different frames in rCFD. In large-scale cases with a lot of turbulence, it is highly unlikely to find two very similar states within a short time series. Notably, rCFD can work even without pronounced recurrences because it still reproduces the correct average flow properties (Lichtenegger et al., 2022). However, abrupt variations due to jumps between two non-consecutive, not very similar frames can be viewed as additional turbulence causing more diffusion on a macroscopic scale because the velocity field suddenly changes. Such processes might have transported more inclusions towards the top layer where they were removed so that they did not leave the system through the outlet.

Fig. 11 illustrates the time evolution of volume fractions in various tundish regions. Starting from a state with maximum inclusion concentration, its value decreased because of slag separation towards the same equilibrium value in CFD and in rCFD. The equilibrated concentrations as well as their standard deviations were of a similar degree at the dam, the outlet as well as the whole volume and agreed fairly well between CFD and rCFD (cf. Table 2). The only difference was a faster equilibration of the CFD simulation after 94.5 s, while rCFD equilibrated after 120.0 s. A simulation was considered equilibrated when the concentration of inclusions in the whole volume became smaller than $c_{\mathrm{min,Vol}} + 0.02$ for the first time, where $c_{\mathrm{min,Vol}}$ is the smallest measured concentration in that simulation.

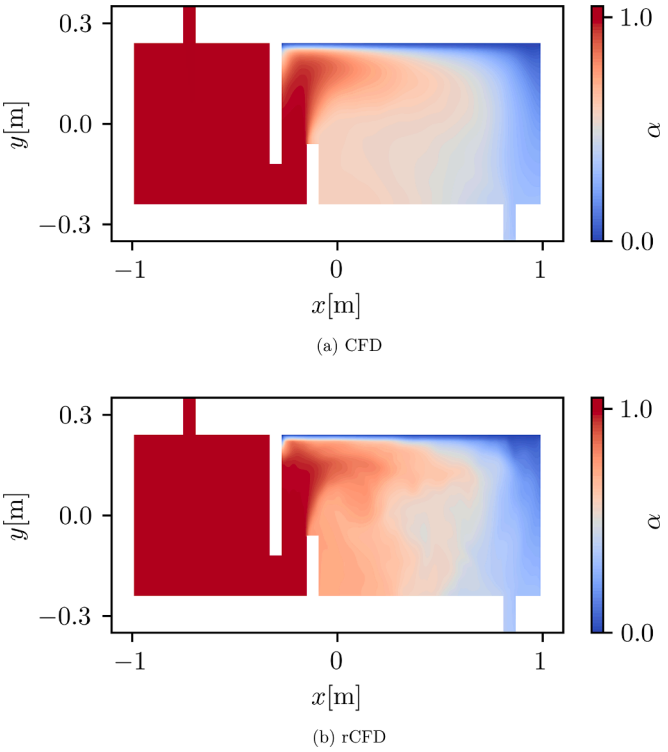

Fig. 10. Time-averaged steel grades for top-layer inclusion removal. Results from (a) CFD and (b) rCFD simulations are shown for the $y = 0$-plane.

Table 2
Concentrations of species and their standard deviations after equilibration in the grade change case. The CFD simulation was equilibrated after 94.5 s and the rCFD simulation after 120.0 s.

Parameter	CFD	rCFD
Volume	0.7541 ± 0.0022	0.7584 ± 0.0029
Dam	0.808 ± 0.011	0.817 ± 0.012
Outlet	0.485 ± 0.019	0.48 ± 0.016

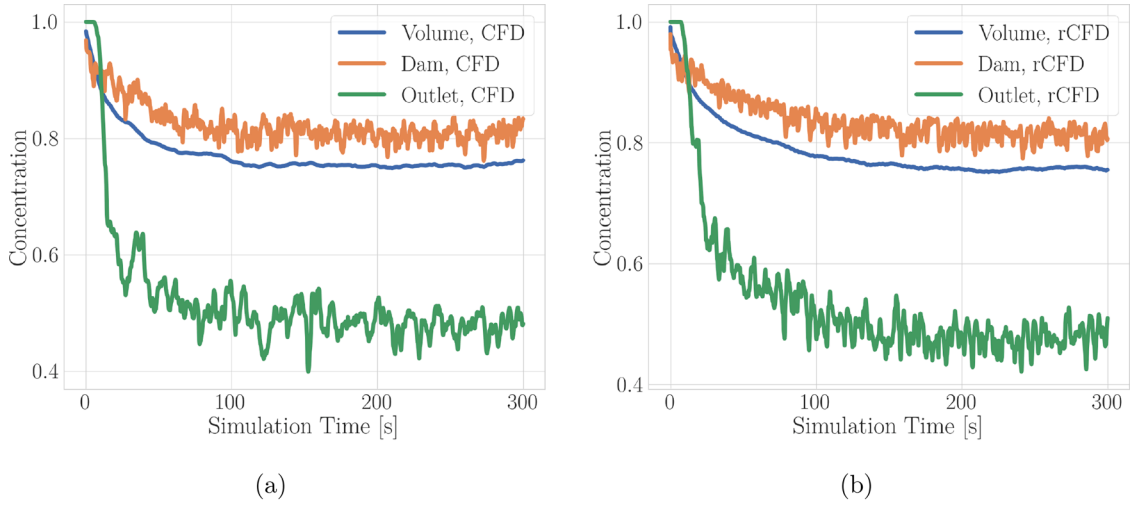

Fig. 11. Time evolution of species volume fraction during a grade change simulation. Both (a) CFD and (b) rCFD results are shown for the whole volume, above the dam and at the outlet.

5.2.3. Computational resources

The simulation time for 1.0 s of simulated process time amounted to approximately 2000 s for CFD and 0.75 s for rCFD using 8 processors 12th Gen Intel Core i9-12900K for both cases. Once the database had been generated, rCFD sped up the simulations by a factor of almost 2700. This came at the cost of increased RAM requirements of 19.8 GB instead of 5.9 GB.

6. Conclusion and outlook

In this work, we have derived the theoretical framework for large-step predictions of transport processes in recurrent flows. In addition to convection, this enabled a detailed description of the combined effect of physical diffusion and dispersion due to velocity gradients. Furthermore, it allowed us to quantify the numerical diffusion inherent to C2Cs and systematically account for it. The example of a cuboid cavity with two moving walls clearly demonstrated the superiority of this strategy compared to attempts with a less accurate handling of diffusion.

To highlight the significant benefits of our data-assisted simulations compared to standard CFD methods, we applied rCFD to a down-scaled tundish geometry from a realistic industrial process in steelmaking. Especially regarding global observables such as residence times, we found excellent agreement with detailed CFD data at about 1/2700 of their runtime. Notably, rCFD took 0.75 s per second of the real process making it faster than real time. Other properties like local probe values or the spatial distribution of species showed mostly very good accuracy, too. However, we noted certain deviations close to regions of strong gradients. There, numerical diffusion can get too strong for a simple correction as proposed in Section 5.1.1. Future work will explore how to improve predictions under such demanding conditions, in particular if the full, anisotropic diffusion tensor Eq. (24) offers a significant benefit over an isotropic approximation as used in the present work.

Furthermore, we will extend our industrially inspired test case of a small tundish towards more realistic settings in terms of size and physical modeling. Upscaling will make the data generation process more demanding and require a coarse-grained C2C representation. Regarding the second direction of improvement, we will address questions of heat transport and inclusion removal at bubble curtains. Both phenomena pose challenges for rCFD because thermal expansion affects the underlying C2C pattern, and mass transfer from the liquid to a bubble phase takes place over length scales much shorter than typical displacements. Overall, these steps will take us closer to real-time simulations of realistic, industrial processes.

Data availability

Data will be made available on request.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the authors used ChatGPT in order to correct spelling and grammatical errors. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Thomas Lichtenegger reports financial support was provided by K1-MET GmbH. Hannes Lumetzberger reports financial support was provided by K1-MET GmbH. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

H. Lumetzberger: Writing – original draft, Software, Methodology, Investigation; **S. Pirker:** Supervision, Software, Funding acquisition; **T. Lichtenegger:** Writing – original draft, Supervision, Methodology, Conceptualization.

Acknowledgment

This work was funded by K1-MET GmbH, Metallurgical Competence Center. The research program of the K1-MET competence center is supported by COMET (Competence Center for Excellent Technologies), the Austrian program for competence centers. COMET is funded by the Federal Ministry for Transport, Innovation and Technology, the Federal Ministry for Digital and Economic Affairs, and the Provinces of Upper Austria, Tyrol, and Styria.

HL acknowledges helpful discussions on the manuscript and the technical setup of the simulations with Tobias Kronlachner.

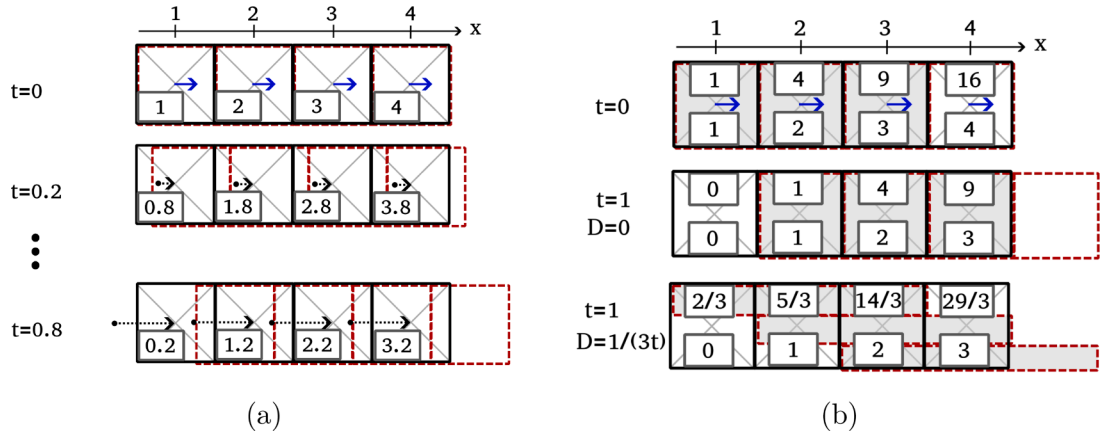

Fig. A.12. Propagator moments calculation. (a) The displacement vector (dotted arrows) and (b) the local diffusivity D_{eff} are obtained from passive scalar fields for a simple 1D example with constant horizontal velocity (blue arrows).

Appendix A. Example of propagator moments in one dimension

For a more practical understanding of the rCFD process, a simple 1D example as shown in Fig. A.12 can be used. The domain consists of a linear array of 4 cells with a constant velocity in the x-direction. For the sake of simplicity, we choose units such that the content in each cell is shifted exactly one cell to the right at $t = 1$. There are no boundaries like inlets or outlets, and it can be assumed that Fig. A.12 shows only a section of an infinitely long linear domain.

A.1. Displacement calculation

Fig. A.12a illustrates the displacement vector calculation via the first moment $\boldsymbol{k}^{(1)}(\boldsymbol{r}, t)$. At $t = 0$, the field, displayed at the bottom left of each cell, is initialized with the corresponding cell center position. As time progresses, the cell content shifts to the right, moving the values of the field as well. The values decrease over time as the content of each cell consists increasingly of its left neighbor.

The value of this field can be interpreted as the average origin of the current cell content or, in other words, as the shift vector. In contrast to shift vectors calculated from tracers, the origin, not the destination, changes over time.

A.2. Diffusion calculation

In Fig. A.12b, the cells are initialized with their center positions and an additional field for their squared positions equivalent to $\boldsymbol{k}^{(2)}(\boldsymbol{r}, t)$. Without diffusion, one finds the values shown in the second row of Fig. A.12b after $t = 1$. Using Eqs. (16), (21), and (24), the effective diffusion D_{eff} is obtained from $\boldsymbol{k}^{(1)}(\boldsymbol{r}, t)$ and $\boldsymbol{k}^{(2)}(\boldsymbol{r}, t)$ via

$$\boldsymbol{D}_{\text{eff}}(\boldsymbol{r}, t) \equiv \frac{1}{2}\left[\boldsymbol{k}^{(2)}(\boldsymbol{r}, t) - \boldsymbol{k}^{(1)}(\boldsymbol{r}, t)\boldsymbol{k}^{(1)}(\boldsymbol{r}, t)\right]. \tag{A.1}$$

As expected, the calculation yields $D_{\text{eff}}(x, 1) = 0$ for each cell in the figure.

In the third row, we retain diffusion, and the cell content is distributed equally among three neighboring cells. For $\boldsymbol{k}^{(1)}(\boldsymbol{r}, t)$, this has no effect in this 1D case, but for $\boldsymbol{k}^{(2)}(\boldsymbol{r}, t)$, the values increase. Finally, one gets

$$D_{\text{eff}}(x, 1) = \frac{1}{2}\left[\frac{2}{3}\right] = \frac{1}{2}\left[\frac{5}{3} - 1^2\right] = \frac{1}{2}\left[\frac{14}{3} - 2^2\right] = \frac{1}{2}\left[\frac{29}{3} - 3^2\right] = \frac{1}{3} \tag{A.2}$$

for all four cells.

References

Abbasi, S., Puttinger, S., Pirker, S., Lichtenegger, T., 2020. Recurrence analysis and time extrapolation of a confined turbulent jet using modal decomposition. Phys. Fluids 32 (7), 075115. https://doi.org/10.1063/5.0010315

Alkin, B., Fürst, A., Schmid, S., Gruber, L., Holzleitner, M., Brandstetter, J., 2024. Universal physics transformers. https://doi.org/10.48550/arXiv.2402.12365

Brebbia, C.A., Partridge, P.W., 2012. Boundary Elements in Fluid Dynamics. Springer, Netherlands.

Brunton, S., Noack, B., Koumoutsakos, P., 2019. Machine learning for fluid mechanics. Annu. Rev. Fluid Mech. 52. https://doi.org/10.1146/annurev-fluid-010719-060214

Cecconi, F., Cencini, M., Falcioni, M., Vulpiani, A., 2012. Predicting the future from the past: an old problem from a modern perspective. Am. J. Phys. 80 (11), 1001–1008. https://doi.org/10.1119/1.4746070

Ferziger, J.H., Perić, M., 2002. Computational Methods for Fluid Dynamics. Springer.

Guo, X., Hu, C., Dai, Y., Xu, H., Zeng, L., 2024. Learning dense gas-solids flows with physics-encoded neural network model. Chem. Eng. J. 485, 150072. https://doi.org/10.1016/j.cej.2024.150072

Hansson, J., Lichtenegger, T., Pirker, S., Sasic, S., Ström, H., 2024. Recurrence CFD for efficient predictions of long-term particle deposition on a cylinder. submitted to Comput. Fluids.

Jha, P., Dash, S., 2004. Employment of different turbulence models to the design of optimum steel flows in a tundish. Int. J. Numer. Methods Heat Fluid Flow 14, 953–979. https://doi.org/10.1108/09615530410544283

Karniadakis, G.E., Kevrekidis, I.G., Lu, L., Perdikaris, P., Wang, S., Yang, L., 2021. Physics-informed machine learning. Nat. Rev. Phys. 3 (6), 422–440. https://doi.org/10.1038/s42254-021-00314-5

Kirkup, S., 2019. The boundary element method in acoustics: a survey. Appl. Sci. 9 (8). https://doi.org/10.3390/app9081642

Lacey, P.M.C., 1954. Developments in the theory of particle mixing. J. Appl. Chem. 4 (5), 257–268. https://doi.org/10.1002/jctb.5010040504

Lichtenegger, T., 2024. Data-assisted, physics-informed propagators for recurrent flows. Phys. Rev. Fluids 9, 024401. https://doi.org/10.1103/PhysRevFluids.9.024401

Lichtenegger, T., Abbasi, S., Pirker, S., 2022. Transport in turbulent, recurrent flows: time-extrapolation and statistical symmetrization. Chem. Eng. Sci. 259, 117795. https://doi.org/10.1016/j.ces.2022.117795

Lichtenegger, T., Peters, E.A.J.F., Kuipers, J.A.M., Pirker, S., 2017. A recurrence CFD study of heat transfer in a fluidized bed. Chem. Eng. Sci. 172, 310–322. https://doi.org/10.1016/j.ces.2017.06.022

Lu, L., Brennan Pecha, M., Wiggins, G.M., Xu, Y., Gao, X., Hughes, B., Shahnam, M., Rogers, W.A., Carpenter, D., Parks, J.E., 2022. Multiscale CFD simulation of biomass fast pyrolysis with a machine learning derived intra-particle model and detailed pyrolysis kinetics. Chem. Eng. J. 431, 133853. https://doi.org/10.1016/j.cej.2021.133853

Mavriplis, D., 2003. Revisiting the least-squares procedure for gradient reconstruction on unstructured meshes. In: 16th AIAA Computational Fluid Dynamics Conference, p. 3986. https://doi.org/10.2514/6.2003-3986

Menter, F.R., 1994. Two-equation eddy-viscosity turbulence models for engineering applications. AIAA J. 32 (8), 1598–1605. https://doi.org/10.2514/3.12149

Menter, F.R., Kuntz, M., Langtry, R., et al., 2003. Ten years of industrial experience with the SST turbulence model. Turbul., Heat Mass Transf. 4 (1), 625–632.

Oommen, V., Shukla, K., Goswami, S., Dingreville, R., Karniadakis, G.E., 2022. Learning two-phase microstructure evolution using neural operators and autoencoder architectures. npj Comput. Mater. 8 (1), 190. https://doi.org/10.1038/s41524-022-00876-7

Ouyang, B., Zhu, L.-T., Luo, Z.-H., 2022. Machine learning for full spatiotemporal acceleration of gas-particle flow simulations. Powder Technol. 408, 117701. https://doi.org/10.1016/j.powtec.2022.117701

Patankar, S., 2018. Numerical Heat Transfer and Fluid Flow. CRC Press.

Pirker, S., Lichtenegger, T., 2018. Efficient time-extrapolation of single- and multiphase simulations by transport based recurrence CFD (rCFD). Chem. Eng. Sci. 188, 65–83. https://doi.org/10.1016/j.ces.2018.04.059

Raissi, M., Karniadakis, G.E., 2018. Hidden physics models: machine learning of nonlinear partial differential equations. J. Comput. Phys. 357, 125–141. https://doi.org/10.1016/j.jcp.2017.11.039

rCFD. https://github.com/ParticulateFlow/rCFDreloaded.

Sheng, D.-Y., 2020. Mathematical modelling of multiphase flow and inclusion behavior in a single-strand tundish. Metals 10 (9). https://doi.org/10.3390/met10091213

Shirzadi, M., Fukasawa, T., Fukui, K., Ishigami, T., 2023. Application of deep learning neural networks for the analysis of fluid-particle dynamics in fibrous filters. Chem. Eng. J. 455, 140775. https://doi.org/10.1016/j.cej.2022.140775

Wang, S., Wang, H., Perdikaris, P., 2021. Learning the solution operator of parametric partial differential equations with physics-informed DeepONets. Sci. Adv. 7 (40), eabi8605. https://doi.org/10.1126/sciadv.abi8605

Wang, S., Yu, X., Perdikaris, P., 2022. When and why PINNs fail to train: a neural tangent kernel perspective. J. Comput. Phys. 449, 110768. https://doi.org/10.1016/j.jcp.2021.110768

Zhu, L.-T., Chen, X.-Z., Ouyang, B., Yan, W.-C., Lei, H., Chen, Z., Luo, Z.-H., 2022. Review of machine learning for hydrodynamics, transport, and reactions in multiphase flows and reactors. Ind. Eng. Chem. Res. 61 (28), 9901–9949. https://doi.org/10.1021/acs.iecr.2c01036