Computer Physics Communications 325 (2026) 110169

Contents lists available at ScienceDirect

Computer Physics Communications

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

Computational Physics

Quantum-inspired tensor-network fractional-step method for incompressible flow in curvilinear coordinates

Nis-Luca van Hülst[a,*], Pia Siegl[a,b], Paul Over[c], Sergio Bengoechea[c], Tomohiro Hashizume[a,d], Mario Guillaume Cecile[a], Thomas Rung[c], Dieter Jaksch[a,d,e]

[a] Institute for Quantum Physics, University of Hamburg, Luruper Chaussee 149, D-22761 Hamburg, Germany
[b] Institute of Software Methods for Product Virtualization, German Aerospace Center (DLR), Nöthnitzer Straße 46b, 01187 Dresden, Germany
[c] Institute for Fluid Dynamics and Ship Theory, Hamburg University of Technology, Hamburg D-21073, Germany
[d] The Hamburg Centre for Ultrafast Imaging, Luruper Chaussee 149, D-22761 Hamburg, Germany
[e] Clarendon Laboratory, University of Oxford, Parks Road, Oxford OX1 3PU, United Kingdom

ARTICLE INFO

Prof. Andrew Hazel

Keywords:
Quantum computational fluid dynamics
Quantics tensor trains
Quantum-inspired flow solver
Curvilinear coordinates
Reduced order modeling
Navier-Stokes flow

ABSTRACT

We introduce an algorithmic framework based on tensor networks for computing fluid flows around immersed objects in curvilinear coordinates. We show that the tensor network simulations can be carried out solely using highly compressed tensor representations of the flow fields and the differential operators and discuss the numerical implementation of the tensor operations required for computing fluid flows in detail. The applicability of our method is demonstrated by applying it to the paradigm example of steady and transient flows around stationary and rotating cylinders. We find excellent quantitative agreement in comparison to finite difference simulations for Strouhal numbers, forces and velocity fields. The properties of our approach are discussed in terms of reduced order models. We estimate the memory saving and potential runtime advantages in comparison to standard finite difference simulations. We find accurate results with errors of less than 0.3% for flow-field compressions by a factor of up to 20 and differential operators compressed by factors of up to 1000 compared to sparse matrix representations. We provide strong numerical evidence that the runtime scaling advantages of the tensor network approach with system size will provide substantial resource savings when simulating larger systems. Finally, we note that, like other tensor network-based fluid flow simulations, our algorithmic framework is directly portable to a quantum computer leading to further scaling advantages.

1. Introduction

Engineering computational fluid dynamics (CFD) is subjected to an ever-increasing demand for higher spatial and temporal resolution. An improved dynamic resolution is crucial to understand complex multiphysics and multiscale dynamics whose analysis has so far been impossible due to a lack of computing power. The most prominent example probably relates to the simulation of turbulent flows with the full resolution of all spatiotemporal scales in a direct numerical simulation (DNS). This requirement can pose enormous computational challenges because the range of relevant scales increases nonlinearly with the Reynolds number, which renders DNS infeasible for engineering CFD [1–3]. Classical remedies include turbulence closure models [4,5], which trade generality for reduced resolution requirements, and reduced-order models (ROMs) [6,7], which mitigate the curse of dimensionality [8] but often sacrifice long-term predictive reliability.

Quantum computers (QCs) offer the potential to exponentially reduce memory requirements for storing discretized flow fields [9,10], though the linear nature of quantum mechanics poses challenges for nonlinear problems, compounded by noise and decoherence on current hardware [11,12]. Various strategies have been proposed, including Carleman linearization [13,14], quantum linear solvers [15,16], quantum lattice Boltzmann methods [17,18], Schrödinger-equation mappings [19,20], and hybrid quantum-classical variational approaches [21–24].

While these quantum algorithms hold long-term promise, current hardware limitations motivate the development of quantum-inspired methods, which reformulate classical algorithms using tensor networks [25] as a programming paradigm [26,27] and can already be implemented and benchmarked on classical hardware. Fully scale-resolving quantum-inspired tensor network simulations are computationally attractive only if the entanglement – i.e., the internal

---

[*] Corresponding author.
E-mail address: nis-luca.van.huelst@uni-hamburg.de (N.-L. van Hülst).

https://doi.org/10.1016/j.cpc.2026.110169
Received 16 July 2025; Received in revised form 12 April 2026; Accepted 19 April 2026

correlations between different parts of the system – is limited, effectively turning them into ROMs [10,28]. As conventional intrusive ROMs [29], these methods allow complete CFD simulation within a compressed subspace, akin to studies in quantum many-body systems [30].

Quantics tensor trains [31], also known as a special type of matrix product states (MPS) in quantum physics, and hereafter referred to simply as tensor trains (TTs), have emerged as strong candidates for quantum-inspired tensor network CFD methods. Building on the pioneering work of Dolgov et al. [32], which utilized the TT format to solve semi-discrete parabolic problems, recent studies have successfully extended quantum-inspired ROMs to a broad variety of increasingly advanced CFD problems [32–43]. They mainly utilize conventional finite difference (FD) frameworks and address various challenges faced by conventional CFD, such as incorporating higher-order spatial and temporal approximation schemes. Through this approach, significant speedups have been reported: a 12-fold reduction in computational time for simulating a 2D decaying-turbulence flow on graphic-cards [35], one order of magnitude runtime improvement [36] compared to classical FD for the paradigmatic 2D lid-driven cavity problem and Gourianov et al. [44] reported a reduction of memory and computational cost up to six and three orders of magnitude, respectively, for the $(5+1)$D joint probability density-function simulation of a chemically reactive turbulent flow.

Applications of TT methods to complex geometries are still in their infancy and simulations around non-rectangular bodies represent a challenge. Nowadays, mesh-based classical CFD methods predominantly refer to body-fitted, unstructured polyhedral meshes [45,46]. Other approaches employ the immersed boundary method (IBM) to capture complex geometries usually embedded in structured Cartesian grids [47–50]. In addition, particularly in the area of complex multi-body flow simulations, there are approaches based on overlapping domain-specific grids, which are structured and body-fitted. A key challenge of overset grids is certainly the organization of the dynamic connectivity between grids and the related conservative inter-grid interpolation [51,52]. To date, only two studies have investigated TT-based CFD methods for complex domains. The first one was published by Peddinti et al. [38]. It employs a direct-forcing IBM approach on a Cartesian mesh to simulate flow around circular and square cylinders, as well as a NACA0042 airfoil. However, the added geometry-representing penalty terms inherent to this method generally introduce extra algorithmic complexity and computational overhead. The second one, reported by Kornev et al. [53], utilizes block-structured Cartesian meshes to simulate flow in a T-shaped channel, yet can not resolve non-rectangular objects.

The present paper reports the implementation of a TT-based CFD procedure for body-fitted structured-grid discretizations using curvilinear coordinates. The transformation of Cartesian differential operators into curvilinear operators within the TT format will be outlined in detail. Analogous to classical approaches, the procedure is based on a fractional-step algorithm [54–56]. This enables an accurate and efficient simulation of incompressible flows around submerged objects, extending the applicability of TT methods beyond Cartesian grids. An investigation of overset grids is, however, beyond the scope of this article but it paves the way for future TT applications on these type of grids.

The remainder of the paper is organized as follows. Section 2 introduces the TT format and its algebraic operations. Section 3 presents the physical model, curvilinear coordinate transformation, and time integration scheme. Sections 4.1-4.2 apply the framework to laminar flows around non-rotating and rotating circular cylinders $(20 \leq Re \leq 200)$.

We observe excellent prediction and rapid convergence of the TT algorithm toward the reference values characterizing the system dynamics, such as mean lift coefficients and Strouhal numbers. Remarkably, for the steady (non-rotating) cylinder case the proposed TT solver achieves a relative error below 0.3% in the velocity magnitude utilizing only 5.8% degrees of freedom of classical simulations, i.e., a 20-fold compression. The flow dynamics in the transient case are also accurately predicted with an absolute error below 1×10^{-3} in the Strouhal number at compression ratios as low as 18%. Section 4.3 discusses the computational performance of the TT solver on the transient flow case. We demonstrate that the computational effort required to refine the discretization is significantly reduced when using the TT method compared to classical FD simulations, particularly in cases where the bond dimension does not grow substantially with the system size. Furthermore, we show that the required curvilinear differential operators admit a compression of up to 1000-fold in the TT format, efficiently encoding the orthogonal cylindrical grid. The last section, Section 5, is devoted to conclusions and outlook, where we discuss possible implications for future developments of tensor-network-based fluid dynamics. Additional technical details, including the curvilinear discretization, TT decomposition procedures, grid specifications, and solver verification, are provided in the Appendices.

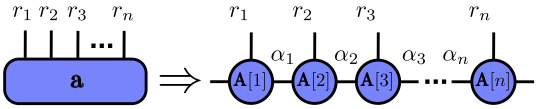

Fig. 1. Example for the decomposition of an arbitrary vector $\boldsymbol{a}$, reshaped as 2^n-component tensor $\mathbf{a}_{r_1,r_2,\dots,r_n}$, into the TT format by performing a series of SVDs, where $\mathbf{A}[k]$ are order-3 tensors with mode indices $r_k \in \{0,1\}$, and bond indices α_k. The corresponding mathematical interpretation is given in Eq. (1).

The publication employs lower-case Latin letters to describe field properties and uppercase Latin letters indicate reference quantities used to non-dimensionalize the field quantities. Dimensional field quantities are denoted by a tilde. In a symbolic representation, the number of underlines indicates the tensorial order, e.g., $\underline{\tilde{x}}$ for the position vector and $\underline{\nabla}$ for the non-dimensional spatial derivative vector. Spatial vectors and tensors are usually defined with reference to Cartesian coordinates. Furthermore, the notation for discretized quantities follows

$$\begin{aligned} \text{Vectors (order-1 tensor): } & \boldsymbol{v}, \\ \text{Matrices (order-2 tensor): } & \boldsymbol{A}, \\ \text{Order-3 or higher tensors: } & \mathbf{v},\ \mathbf{A}, \\ \text{Tensor Trains: } & v,\ \mathcal{A}\,. \end{aligned}$$

2. Tensor trains: Principles and fundamental operations

This section presents a reduced order modeling framework based on TTs in their binary (dyadic) form – usually called quantics tensor trains or understood as a special case of matrix product states – which we adopt for its compatibility with quantum hardware. Throughout the remainder of the paper, the term TT will always refer to this binary specialization.

We define the notation and encoding of state vectors and operators into the TT format, followed by the required algebraic operations (Section 2.1) and their complexity (Section 2.2). Throughout this section, we practice the Einstein summation convention, where repeated indices imply summation. Index fusion/extension assumes column ordering throughout, consistent with our JULIA-based implementation [57]; details are given in Appendix B.

For an introduction to the fundamentals of TTs, a 2^n-component vector is reshaped into an order-n tensor $\mathbf{a}_{r_1,r_2,\dots,r_n}$ with binary $r_k \in \{0,1\}$ for $k = 1,\dots,n$ and subsequently decomposed into a TT representation of n order-3 tensors $\mathbf{A}[k]$ (TT-cores) [53,58–60], as it is schematically depicted in Fig. 1. For the representation of tensors in this work, we follow the (graphical) *Penrose* notation [61], where a vector is an object with one leg (cf. Fig. 1), a line sticking out of the blue box/circle, a matrix has two legs, and an order-n tensor shows n legs. The number of outgoing legs thereby determines the order of the tensor. A link between any two tensors, indicated with a lowercase Greek letter, is referred to as a bond (or artificial) index implying summation. The outer indices instead are called physical (or real) indices with dimension 2 and indicated by a lowercase Latin letter.

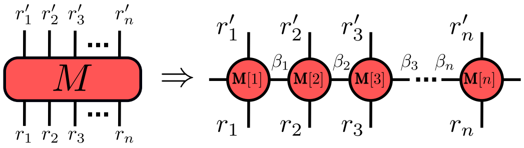

Fig. 2. Graphical depiction of the TT representation $\mathcal{M}$ of a matrix $\boldsymbol{M}$. The red circles with outgoing legs denote TT-cores, where $(r'_1, \ldots, r'_n)$ represents the rows of $\boldsymbol{M}$ and $(r_1, \ldots, r_n)$ enumerates its columns. The corresponding mathematical description is given in Eq. (2). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The decomposition in Fig. 1 is performed by successive singular value decompositions (SVDs), though the TT-cross algorithm [62,63] or analytical constructions for polynomial and trigonometric functions [64] may also be used (see Appendix D for an example). As the decomposition is not unique, we maintain a right-canonical form throughout [25,65].

The dimension of the bond indices α_k connecting individual order-3 tensors can be set by either neglecting singular values below a certain threshold or by manually selecting the first χ diagonal entries in the singular value matrix [25]. In this regard, the complete TT-vector reads as

$$\begin{aligned} a &:= \mathbf{A}[1]^{r_1}_{\alpha_0,\alpha_1} \mathbf{A}[2]^{r_2}_{\alpha_1,\alpha_2} \cdots \mathbf{A}[n]^{r_n}_{\alpha_{n-1},\alpha_n} \\ &\simeq \mathbf{a}_{r_1,r_2,\ldots,r_n} , \end{aligned} \tag{1}$$

where the dimensions of bond indices α_k follow $\dim(\alpha_k) = \min(2^k, 2^{n-k}, \chi)$ for $k = 0, \ldots, n$. Accordingly, χ is always referred to as the maximal value in the set $\{\dim(\alpha_0), \dim(\alpha_1), \ldots, \dim(\alpha_n)\}$, and controls the expressiveness of the representation, possibly affecting its accuracy. In the case where the bond dimension is not under truncation, the TT representation $a = \mathbf{a}_{r_1,r_2,\ldots,r_n}$ is exact.

Similarly, a matrix $\boldsymbol{M}$ can be represented as a TT-matrix $\mathcal{M}$ (TT-operator) by local order-4 tensors $\mathbf{M}[k]$ with incoming (not primed) and outgoing (primed) modes of the same mode size (square matrices), 2, such that the contraction of incoming mode indices with the mode indices of a TT-vector naturally gives a new TT-vector. The TT-matrix is illustrated in Fig. 2.

The matrix $\boldsymbol{M}$ can be encoded through the successive application of SVDs, defining the maximal bond dimension $\chi(\mathcal{M})$ as the largest value in the set $\{\dim(\beta_0), \dim(\beta_1), \ldots, \dim(\beta_n)\}$. The TT cross algorithm [62] may also be employed to further reduce computational effort when obtaining the TT-matrix. However, in most cases, numerical decompositions are not required, as the TTs in these instances can be derived analytically [66].

Each Latin index again holds a mode size of 2. The resulting TT-matrix [53], consisting of order-4 tensors, is given by

$$\begin{aligned} \mathcal{M} &:= \mathbf{M}[1]^{r'_1,r_1}_{\beta_0,\beta_1} \mathbf{M}[2]^{r'_2,r_2}_{\beta_1,\beta_2} \ldots \mathbf{M}[n]^{r'_n,r_n}_{\beta_{n-1},\beta_n} \\ &\simeq \mathbf{M}^{r'_1,r'_2,\ldots,r'_n}_{r_1,r_2,\ldots,r_n} . \end{aligned} \tag{2}$$

Again, without truncation, the above representation is exact.

2.1. Algebraic operations

The common algebraic operations, such as addition ($a + b$), matrix-vector product ($\mathcal{M} a$), and elementwise multiplication ($a \odot b$), are introduced in Sections 2.1.1–2.1.3, and the variational solution of linear systems in Section 2.1.4. Since these operations generally increase the bond dimension, we apply TT-rounding [25,58] after each step, truncating to a fixed bond dimension χ and additionally filters out singular values below a threshold of 10^{-14}.

2.1.1. Vector-vector addition

Given two TT-vectors a and b, with cores $\mathbf{A}[k]^{r_k}$ and $\mathbf{B}[k]^{r_k}$ and bond dimension $\chi(a)$ and $\chi(b)$, a summation of a and b results in the new TT-vector c. For the summation, each core $C[k]$ of c is built as

$$\mathbf{C}[k]^{r_k} = \begin{bmatrix} \mathbf{A}[k]^{r_k} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}[k]^{r_k} \end{bmatrix} = \mathbf{A}[k]^{r_k} \oplus \mathbf{B}[k]^{r_k} , \tag{3}$$

where the direct sum $\oplus$ is the shorthand notation for this block encoding. For $k = 1$ ($k = n$), the TT-cores of a and b are concatenated in a row (column) vector, respectively. The resulting TT-vector c will have a bond dimension $\chi(c) = \chi(a) + \chi(b)$, which may be truncated.

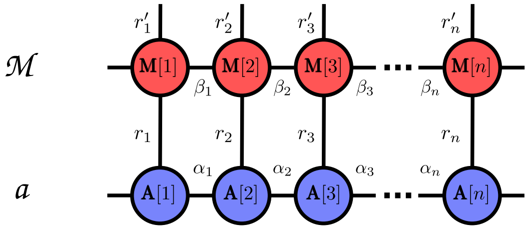

Fig. 3. Graphical representation of TT-matrix acting on a TT-vector, which is achieved by contraction over common physical indices.

2.1.2. Matrix-vector product

The matrix-vector product is realized as a combination of TTs [66]. For the application of a TT-matrix $\mathcal{M}$ to a TT-vector a, multiple contractions along the physical indices are required, as it is depicted in Fig. 3.

In mathematical terms, the TT-vector $b = \mathcal{M} a$ has the cores

$$\mathbf{B}[k]^{r'_k}_{1,(\alpha_k\beta_k)} = \sum_{r_k} \mathbf{M}[k]^{r'_k,r_k}_{1,\beta_k} \mathbf{A}[k]^{r_k}_{1,\alpha_k} \tag{4}$$

such that the resulting tensor train is

$$b = \mathbf{B}[1]^{r'_1}_{1,(\alpha_1\beta_1)} \mathbf{B}[2]^{r'_2}_{(\alpha_1\beta_1),(\alpha_2\beta_2)} \ldots \mathbf{B}[n]^{r'_n}_{(\alpha_n\beta_n),1} . \tag{5}$$

Here, $(\alpha_k\, \beta_k)$ denotes the fused bond index formed by combining α_k and β_k, with total dimension $\dim(\alpha_k) \times \dim(\beta_k)$ (cf.Appendix B). As a result, this increases the bond dimension of the resulting TT-vector, which requires regular truncation (e.g., using SVD) for operations to remain efficient.

2.1.3. Vector-vector elementwise multiplication

As the last fundamental operation, the vector-vector elementwise multiplication, indicated by $\odot$, is introduced. For three TT-vectors a, b, and c, the elementwise product $c = a \odot b$ is computed in two steps. The first step involves contracting the copy tensor $\boldsymbol{\delta}^{r'_k}_{r_k,r''_k}$ tensor, defined as [67]

$$\boldsymbol{\delta}^{r'_k}_{r_k,r''_k} = \begin{cases} 1 & \text{if } r_k = r'_k = r''_k \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

into each TT-core of a, which effectively doubles its physical dimension (from 2 to 4). This turns the TT-vector a into a diagonal TT-matrix $\mathcal{A}$ with the vector entries along its diagonal (cf.Appendix B). Secondly, the resulting TT $c = \mathcal{A} b$ is computed by the matrix-vector product, as it is described in Section 2.1.2. Both steps of the vector-vector elementwise operation are illustrated in Fig. 5. This increases the bond dimension of the resulting TT-vector according to the matrix-vector product detailed in the previous subsection.

2.1.4. Solving linear systems of equations

The discretization of partial differential equations typically leads to linear systems of equations (LSEs). In this section, we introduce a TT-based approach for solving such systems. In classical computing, the unknowns of an LSE can be determined in many ways (e.g., Gauss-elimination, Cholesky decomposition, …). Within the TT format, however, one typically employs dedicated solvers; examples are described in Refs. [68–71]. In the present work we adopt a single-site version

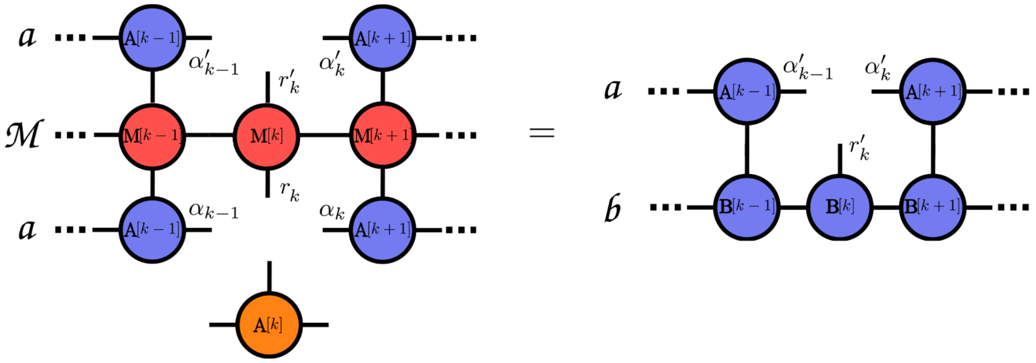

Fig. 4. Graphical representation of the local linear system for the k-th TT-core $\mathbf{A}[k]$, obtained by enforcing the stationarity condition $\partial C/\partial\mathbf{A}[k] = 0$. Solving this system yields the updated core $\mathbf{A}[k]$ (orange node).

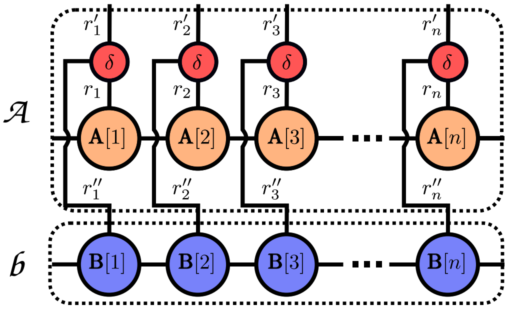

Fig. 5. Graphical representation of vector-vector elementwise multiplication between two TT-vectors a and b.

Table 1
Computational complexity of the TT operations introduced in Section 2, including subsequent truncation to a target bond dimension (TT-rounding), and variational complexities as discussed in Ref. [73].

Operation	Direct approach	Variational
$c = a + b$	$\mathcal{O}\big((\chi(a) + \chi(b))^3\big)$	$\mathcal{O}\big(\chi(c)(\chi(a) + \chi(b))^2\big)$
$c = a \odot b$	$\mathcal{O}\big(\chi(a)^3\chi(b)^3\big)$	$\mathcal{O}\big(\chi(a)\chi(b)^2\chi(c)$ $+ \chi(a)^2\chi(b)\chi(c)\big)$
mat-vec $b = \mathcal{M}a$	$\mathcal{O}\big(\chi(\mathcal{M})^3\chi(a)^3\big)$	$\mathcal{O}\big(\chi(\mathcal{M})\chi(a)^2\chi(b)\big)$
LSE $\mathcal{M}a = b$	N/A	$\mathcal{O}\big(\chi(\mathcal{M})\chi(a)^3$ $+ \chi(\mathcal{M})^2\chi(a)^2\big)$

of the variational density matrix renormalization group (DMRG) algorithm, also known as the alternating least-squares method, as detailed, for example, in Refs. [69,70].

Generally, given TT representations of a matrix $\mathcal{M}$, a vector a and a right-hand side (RHS) b the linear problem reads as

$$\mathcal{M}a = b\,. \tag{7}$$

For application of the DMRG, we define a cost function $C(a)$, whose global minimum corresponds to the exact solution of Eq. (7) if the matrix $\mathcal{M}$ is symmetric and positive-semidefinite [32,72]. If this condition is not satisfied, this cost function has still been shown to perform well [32]; alternatively, a least-squares formulation may be employed [72]. To approximate the solution of the given problem, we solve the optimization problem

$$\min_{a} C(a) := \frac{1}{2}\big(a\mathcal{M}a\big) - ab, \tag{8}$$

where the unknown TT-vector a is fully parameterized. Taking the derivative of the cost functional w.r.t. the individual TT-cores of a reads

$$\begin{aligned}\frac{\partial C}{\partial\mathbf{A}[k]} &= \frac{1}{2}\frac{\partial\big(a\mathcal{M}a\big)}{\partial\mathbf{A}[k]} - \frac{\partial(ab)}{\partial\mathbf{A}[k]}\\ &= \boldsymbol{M}^{\mathrm{eff}}_{(\alpha'_{k-1}r'_k\alpha'_k),(\alpha_{k-1}r_k\alpha_k)}\mathbf{A}[k]_{(\alpha_{k-1}r_k\alpha_k)}\\ &\quad - \boldsymbol{b}^{\mathrm{eff}}_{(\alpha'_{k-1}r'_k\alpha'_k)}\,,\end{aligned} \tag{9}$$

where $(\alpha_{k-1}r_k\alpha_k)$ denotes the fused bond index formed by combining indices α_{k-1}, r_k, and α_k (cf.Appendix B).

The algorithm is then realized via successive DMRG-style sweeps over all TT-cores [25,69]. During the update of core $\mathbf{A}[k]$, all other cores are held fixed and a small linear system for the optimal $\mathbf{A}[k]$ is obtained by enforcing the stationarity condition $\partial C/\partial\mathbf{A}[k] = 0$. This step is graphically depicted in Fig. 4. The local effective matrix $\boldsymbol{M}^{\mathrm{eff}}$ is of size $2\chi^2 \times 2\chi^2$, so that the matrix-vector product $\boldsymbol{M}^{\mathrm{eff}}\mathbf{A}[k]$ scales as $\mathcal{O}(\chi^4)$.

An alternative approach is to emulate the local matrix-vector product by keeping the constituent tensors of $\boldsymbol{M}^{\mathrm{eff}}$ and applying successive tensor contractions [32]. This is reported to scale as $\mathcal{O}\big(\chi(\mathcal{M})\chi^3 + \chi(\mathcal{M})^2\chi^2\big)$. However, numerical studies have shown poor convergence in the latter scheme in the case of ill-conditioned systems, rendering preconditioners mandatory. A strategy to find preconditioners directly in the TT format is outlined in [32].

2.2. *TT complexity*

Table 1 compares the computational complexity of direct TT operations (including TT-rounding) with variational alternatives [36,53,73]. Variational methods scale more favorably with χ and are employed here for the most expensive operations (elementwise multiplications and LSEs). Direct methods are used for cheaper operations to avoid the overhead and potential convergence issues of the variational approach. Further improvements are possible through the TT multiplication algorithm [39] or the zip-up method [74].

3. Computational model

This section briefly outlines the governing equations (Section 3.1) and the classical FD-based discretization strategy based on curvilinear coordinates (Section 3.2). In the latter, we describe how this discretization is implemented within the TT format, i.e., encoding two-dimensional scalar fields (3.2.1), generating a body-fitted grid (3.2.2), and transforming the corresponding differential operators (3.2.3). The section concludes with detailing the fractional step method employed to solve the incompressible Navier-Stokes equations (INSE) in Section 3.3.

3.1. Governing equations

The two-dimensional, laminar fluid-dynamics problem is governed by the INSE for a Newtonian fluid with constant density $\tilde{\rho}$. Using non-dimensional spatial coordinates $\underline{x} = \underline{\tilde{x}}/L$, velocities $\underline{v} = \underline{\tilde{v}}/U_\infty$ in addition to a non-dimensional time $t = \tilde{t}/T_{\text{ref}}$ and pressure $p = \tilde{p}/P_{\text{ref}}$, the system of momentum (10) and continuity (11) equations inside a physical domain Ω read

$$\frac{\partial \underline{v}}{\partial t} + \left(\underline{v} \cdot \underline{\nabla}\right)\underline{v} = -\underline{\nabla}\, p + \frac{1}{Re}\Delta\, \underline{v} + \underline{f} \quad \text{in } \Omega\,, \tag{10}$$

$$\underline{\nabla} \cdot \underline{v} = 0 \quad \text{in } \Omega\,, \tag{11}$$

where $\underline{\nabla}$ denotes the vector of the non-dimensional spatial derivatives, $Re = U_\infty L/\tilde{\nu}$ is the Reynolds number, $T_{\text{ref}} = L/U_\infty$ is a convective reference time, U_∞ is the reference (approach flow) velocity, $\tilde{\nu}$ refers to the (constant) kinematic viscosity of the fluid and L denotes a reference length, typically the diameter D of the investigated cylinder studied in this paper, cf. Fig. 7. The employed reference pressure refers to $P_{\text{ref}} = \tilde{\rho} U_\infty^2$ and $\underline{f} = \underline{\tilde{f}} L/U_\infty^2$ denotes a non-dimensional volume-specific body force.

The boundary of the object, denoted Γ, is subject to Dirichlet (no-slip) conditions $\underline{v}_\Gamma$ for the velocity and a natural (Neumann) condition for the pressure. The exterior boundaries of the domain $\partial\Omega$ are assigned to Dirichlet conditions for both velocity and pressure, viz.

$$\begin{aligned} \underline{v} - \underline{v}_\Gamma &= \underline{0}\,, \qquad & \underline{\nabla} p \cdot \underline{n} &= 0\,, \qquad & &\text{on } \Gamma\,, \\ \underline{v} - \underline{v}_\infty &= \underline{0}\,, \qquad & p &= 0\,, \qquad & &\text{on } \partial\Omega\,. \end{aligned} \tag{12}$$

Initial conditions are prescribed by a spatially dependent $\underline{v} = \underline{v}_{\text{init.}}$ for $t = 0$. Boundary conditions are incorporated using a ghost-point approach. These conditions enter the discrete differential operators as additive correction terms that augment the interior FD stencil. Thereby, each operator application can be expressed as a matrix-vector product plus an explicit boundary-correction vector, while boundary points themselves are excluded from the solution vector [75]. Contributions to the source term $\underline{f}$ as volume-specific body forces by the physical problem are neglected in the course of this work.

3.2. Discretization and representation

The spatial discretization of the two-dimensional physical domain Ω is based on a structured O-type grid $\mathcal{G}$ [76,77]. As outlined in Appendix A, the physical coordinates (x_{ij}, z_{ij}) are defined by a mapping to a unit-square computational domain Ω_0 using a simple transfinite interpolation approach [78]. The interior supporting points of the computational domain are located at

$$\left.\begin{aligned} \eta_{ij} &= (i+1)\delta\eta \\ \xi_{ij} &= j\delta\xi \end{aligned}\right\} \text{with} \quad \begin{aligned} i &= 0, \ldots, 2^{n_\eta} - 1\,, \\ j &= 0, \ldots, 2^{n_\xi} - 1\,. \end{aligned} \tag{13}$$

Hence, employing n_η and n_ξ TT-cores for the η- and ξ-directions, respectively. We set $N_\eta = 2^{n_\eta}$ and $N_\xi = 2^{n_\xi}$, thereby obtaining an $N_\eta \times N_\xi$ structured grid with uniform spacings $\delta\eta = 1/(N_\eta + 1)$ and $\delta\xi = 1/N_\xi$ (cf. A). Note that the auxiliary indices $i = -1$ and $i = N_\eta$ coincide with the inner (top) and exterior (bottom) boundaries of Ω (Ω_0) domain, while the ξ-direction is treated periodically.

Next, we define a coordinate transformation $\Phi : \Omega_0 \mapsto \Omega$ that maps the computational coordinates (ξ_{ij}, η_{ij}) to the physical coordinates (x_{ij}, z_{ij}) with $x_{ij} = x(\xi_{ij}, \eta_{ij})$ and $z_{ij} = z(\xi_{ij}, \eta_{ij})$ [77]. Using this transformation, we approximate the spatial derivatives in the momentum Eq. (10) and continuity Eq. (11) by their analogues in the computational domain, thereby obtaining curvilinear forms of the differential operators.

3.2.1. 2D field encoding

Before proceeding with the discretization in the TT format, we first introduce the encoding of two-dimensional scalar fields, here defined over the computational domain Ω_0, as TT-vectors. Consider, for example, a scalar field $a(\xi_{ij}, \eta_{ij})$, such as the x-coordinate $x(\xi_{ij}, \eta_{ij})$. To represent such a field as a TT-vector, a consistent enumeration of the grid points (ξ_{ij}, η_{ij}) must first be established. To each point (ξ_{ij}, η_{ij}), we assign a bit string $r = i_b \cup j_b$ of length $n = n_\eta + n_\xi$, formed by concatenating i_b, the length n_η binary representation of row index $i \in \{0, 1, \ldots, N_\eta - 1\}$, with j_b, the length n_ξ binary representation of column index $j \in \{0, 1, \ldots, N_\xi - 1\}$. Note that i and j are the indices defined in Eq. (13). Each bit in r is then interpreted as the two modes $\{0, 1\}$ of the indices of an order-n tensor $\mathbf{a}_{r_1, r_2, \ldots, r_n}$ that encodes $N = 2^n$ discretized field values as presented in Section 2, where r_k is the k^{th} bit of r. After decomposing $\mathbf{a}_{r_1, r_2, \ldots, r_n}$ for $n - 1$ times, as illustrated in Fig. 6, one obtains the TT-vector representation of the field

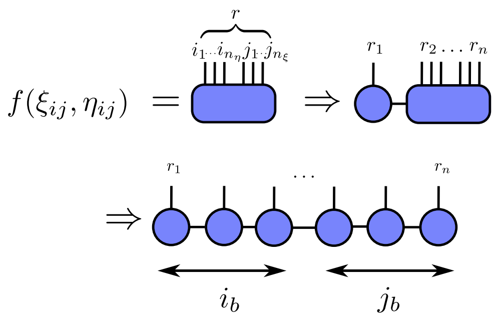

Fig. 6. Graphical representation of a TT decomposition of a 2D field. Here, i_k (j_k) represents the value of the k^{th} bit of bit string i_b (j_b).

$$a = \mathbf{A}[1]^{r_1} \mathbf{A}[2]^{r_2} \ldots \mathbf{A}[n]^{r_n} \simeq \mathbf{a}_{r_1, r_2, \ldots, r_n} = f(\xi_{ij}, \eta_{ij})\,. \tag{14}$$

Vector fields, such as the velocity $\underline{v}$, are treated by encoding x- and z-components of a field separately into two different TT-vectors.

While the aforementioned method yields a flexible encoding of various fields, the computational cost is proportional to the number of grid points $N = N_\eta N_\xi$. Practically, a field with recognized features (e.g., uniform, harmonic, exponential, ...) has a low-rank representation that allows a direct construction of its TT-cores that only costs $\mathcal{O}\big(\log(N)\chi^2\big)$ [64], or it can be found through tensor cross interpolation [63]. For the simulations conducted in this paper, the TT-vectors of the initial flow fields are directly constructed within the TT format, thus avoiding the decomposition of order-n tensors. By keeping the bond dimension fixed to a constant value throughout each step of the algorithm described in Section 3.3, the time evolution and subsequent analysis are performed without requiring the full numerical representation of the fields, thereby preserving the overall logarithmic scaling in N.

3.2.2. Mesh generation in TT format

The discretization (x_{ij}, z_{ij}) is classically defined by a grid $\mathcal{G}$, which can either be read in or generated directly in the TT format. In either case, the number of interior grid points must satisfy the constraint $2^{n_\eta} \times 2^{n_\xi}$ to be compatible with the TT format. To preserve the logarithmic scaling in the number of points N, the latter approach is preferable, as it avoids the explicit construction and decomposition of full N-component tensors, and is hence discussed in the following.

For a convex shape Γ with supporting points on the surface given by $(x_{\Gamma j}, z_{\Gamma j})$, the interior grid points can be constructed either uniformly along the surface normal direction, or using a non-uniform distribution with step sizes that gradually increase with the distance from the surface Γ, e.g.,

$$s_i^{(\text{lin})} = Q(i+1)\delta\eta \qquad \text{(linear)} \quad \text{or} \tag{15}$$

$$s_i^{(\text{exp})} = Q\,\frac{e^{\kappa(i+1)\delta\eta} - 1}{e^{\kappa} - 1} \qquad \text{(exponential)}\,, \tag{16}$$

where Q denotes the maximal normal distance to the surface Γ and κ is a stretching parameter. For $\kappa \to 0$, the exponential $s_i^{(\text{exp})}$ spacing reduces to linear $s_i^{(\mathrm{lin})}$. The grid coordinates are constructed as a tensor product of 1D functions defined along the η- and ξ-directions, respectively:

$$x_{ij} = 1_i x_{\Gamma\, j} + s_i\, \underline{n}_j^x \quad \text{and} \tag{17}$$

$$z_{ij} = 1_i z_{\Gamma\, j} + s_i\, \underline{n}_j^z\,, \tag{18}$$

where s_i is either $s_i^{(\mathrm{lin})}$ or $s_i^{(\mathrm{exp})}$, 1_i denotes a vector of ones, and $\underline{n}_j^x$, $\underline{n}_j^z$ are the x- and z-components of the unit normal vector at surface point j. For the orthogonal cylinder mesh illustrated in Fig. 7, Eqs. (17), (18) agree with the results of a slightly more involved classical transfinite interpolation method [79].

To translate Eqs. (17) and (18) into TTs, the N_ξ-component vectors containing to the x- and z-coordinates of the object's surface Γ, $x_j^{(\Gamma)}$ and $z_j^{(\Gamma)}$, as well as the components of the unit normal vector, $\underline{n}_j^x$ and $\underline{n}_j^z$, with $j = 0, \dots, N_\xi - 1$ must each be encoded as TT-vectors of length n_ξ. These are denoted as $\mathcal{x}^{(\Gamma)}$, $\mathcal{z}^{(\Gamma)}$, $\mathcal{n}^x$, and $\mathcal{n}^z$, respectively. Furthermore, the TT-vectors of length n_η corresponding to 1_i and s_i, with $i = 0, \dots, N_\eta - 1$, are analytically prescribed and low-rank. The TT-cores of 1_i and linear spacing $s_i^{(\mathrm{lin})}$ can be constructed following Ref. [64], while the strategy for the exponential spacing $s_i^{(\mathrm{exp})}$ is described in Ref. [27], and denoting them by $\mathcal{1}$ and $\mathcal{s}$. The final TT-vectors of the interior coordinates ($x_{ij} \to \mathcal{x}$ and $z_{ij} \to \mathcal{z}$) are then realized by concatenating the above TT-vectors as

$$\mathcal{x} = \mathcal{1} \cup \mathcal{x}^{(\Gamma)} + \mathcal{s} \cup \mathcal{n}^x \quad \text{and} \tag{19}$$

$$\mathcal{z} = \mathcal{1} \cup \mathcal{z}^{(\Gamma)} + \mathcal{s} \cup \mathcal{n}^z\,, \tag{20}$$

where $\cup$ here denotes $\mathcal{a} \cup \mathcal{b} = \mathbf{A}[1] \dots \mathbf{A}[n_\eta]\, \mathbf{B}[1] \dots \mathbf{B}[n_\xi]$. The resulting TT-vector is then of length $n_\eta + n_\xi$.

3.2.3. Constructing curvilinear operators in TT format

The construction of curvilinear operators as TT-matrices exemplarily presented for the physical-space derivative ∂_x follows (cf. Appendix A)

$$\partial_x = \frac{1}{J}\Big((\partial_\eta z)\partial_\xi - (\partial_\xi z)\partial_\eta \Big)\,, \tag{21}$$

where J is the Jacobian determinant $J = (\partial_\xi x)(\partial_\eta z) - (\partial_\eta x)(\partial_\xi z)$. The same construction strategy applies analogously to the physical-space derivative ∂_z as well as to the Laplacian operator Δ. We note that all computational-space derivatives are discretized using a second-order central FD scheme (cf. A).

In convection-dominated regimes, it can be beneficial to employ upwind-biased discretizations to enforce numerical stability. As demonstrated by Ye and Loureiro [80] and Danis et al. [40], such schemes can be realized in the TT format using Tensor Cross Interpolation (TCI) [63]. Alternatively, an iterative algorithm can be used [81]. The methods introduced therein can likewise be extended to curvilinear coordinates. In this work, however, we restrict ourselves to central differences, which provide sufficient stability and accuracy for the parameter ranges investigated.

Given the TT-vector representations of the internal grid points, $\mathcal{x}$ and $\mathcal{z}$, as detailed in Section 3.2.2, the derivative operators ∂_ξ and ∂_η can be directly constructed as low-rank TT-matrices [66], $\mathcal{D}_\xi$ and $\mathcal{D}_\eta$, respectively. Accordingly, the TT-vector $\mathcal{j}$ of the Jacobian determinant J is computed as

$$\mathcal{j} = (\mathcal{D}_\xi \mathcal{x})(\mathcal{D}_\eta \mathcal{z}) - (\mathcal{D}_\eta \mathcal{x})(\mathcal{D}_\xi \mathcal{z})\,. \tag{22}$$

Next, the computation of $\mathcal{c}_1 = (\mathcal{D}_\eta \mathcal{z}) \odot \mathcal{j}^{-1}$ and $\mathcal{c}_2 = (\mathcal{D}_\xi \mathcal{z}) \odot \mathcal{j}^{-1}$ is executed where the inverse $\mathcal{j}^{-1}$ is found as the solution of the LSE $\mathcal{J}\mathcal{a} = \mathcal{1}$, where $\mathcal{J}$ denotes the diagonal TT-matrix representation of $\mathcal{j}$ and $\mathcal{1}$ a TT-vector of ones. Thus $\mathcal{c}_1$ and $\mathcal{c}_2$ are found by a combination of known operations, cf. Section 2.1.3–2.1.4. As a final step, the prefactors $\mathcal{c}_1$, $\mathcal{c}_2$ represented as TT-vectors, need to be multiplied into the TT-matrices $\mathcal{D}_\xi$ and $\mathcal{D}_\eta$ to construct the partial derivative in x-direction in TT format, viz. $\mathcal{D}_x = \mathcal{c}_1 \mathcal{D}_\xi - \mathcal{c}_2 \mathcal{D}_\eta$. This operation is realized as

$$\mathcal{D}_x = \mathcal{C}_1 \mathcal{D}_\xi - \mathcal{C}_2 \mathcal{D}_\eta\,, \tag{23}$$

where $\mathcal{C}_1$ ($\mathcal{C}_2$) are diagonal TT-matrices that have $\mathcal{c}_1$ ($\mathcal{c}_2$) on its diagonal using the $\boldsymbol{\delta}^{r'_k}_{r_k, r''_k}$ tensor as introduced in Section 2.1.3. The matrix-matrix product $\mathcal{C}_1 \mathcal{D}_\xi$ is performed similarly to a TT-matrix with TT-vector contraction with changes w.r.t. the physical dimension, and, in the worst case, scaling as $\mathcal{O}\big(\chi(\mathcal{D}_\xi)^3 \chi(\mathcal{C}_1)^3\big)$). Note that the bond dimension $\chi(\mathcal{C}_1)$ generally depends on the grid coordinates, while $\chi(\mathcal{D}_\xi)$ is low-rank and independent of the grid [66]. Moreover, they are independent of the flow regime, as it requires the construction of the curvilinear operators only once for a given FD scheme and discretization. Lastly, in constructing the curvilinear operators as TT-operators, a truncation is performed after each TT operation by discarding singular values below 10^{-14} (cf. Section 2).

3.3. Fractional step method

A fractional step method [75] is used to advance the flow field described by the momentum Eq. (10) and the continuity Eq. (11) from $t^{(m)}$ to $t^{(m+1)} = t^{(m)} + \delta t$. In this approach, the pressure gradient-free momentum equations are used to determine an intermediate velocity $\underline{v}^*$ [82]

$$\begin{aligned} &\frac{\underline{v}^* - \underline{v}^{(m)}}{\delta t} + \big(\underline{v}^{(m)} \cdot \underline{\nabla}\big)\underline{v}^{(m)} = \frac{1}{Re}\Delta\, \underline{v}^* \\ \Leftrightarrow\ &\Big(1 - \frac{\delta t}{Re}\Delta\Big)\underline{v}^* = \underline{v}^{(m)} - \delta t\big(\underline{v}^{(m)} \cdot \underline{\nabla}\big)\underline{v}^{(m)}. \end{aligned} \tag{24}$$

The intermediate velocity $\underline{v}^*$ is subsequently corrected by solving a pressure Poisson equation [75], which is controlled by the divergence of the intermediate velocity field, i.e.,

$$\Delta\, p^{(m+1)} = \frac{1}{\delta t}\, \underline{\nabla} \cdot \underline{v}^*\,. \tag{25}$$

The gradient of the computed pressure $p^{(m+1)}$ is used to enforce the continuity constraint for the velocity at the next time step $t^{(m+1)}$ with

$$\underline{v}^{(m+1)} = \underline{v}^* - \delta t \underline{\nabla}\, p^{(m+1)}\,. \tag{26}$$

Instead of an adaptive, Courant-based time step, δt is kept constant to maintain the efficiency in the TT format. The resulting method is equivalent to the fractional step method originally proposed by Chorin [54]. The Poisson Eq. (25) is solved in TT format using the variational optimization outlined in Section 2.1.4. This procedure is particularly well suited to elliptic equations, and we include the Jacobian determinant J to account for the varying cell sizes in the curvilinear discretization [72]. Consequently, the modified optimization problem for the Poisson Eq. (25) reads

$$\min_{\mathcal{a}} C(\mathcal{a}) := \frac{1}{2}\Big(\mathcal{a}\Big(\mathcal{j}\mathcal{M}^{(\Delta)}\Big)\mathcal{a}\Big) - \mathcal{a}(\mathcal{j} \odot \mathcal{b})\,. \tag{27}$$

Lastly, we note that the introduced methodology generalizes naturally to simulating Navier-Stokes equation in 3D curvilinear coordinates. This is detailed in A. In this article, however, we restrict our numerical investigation to 2D configurations to provide a focused and succinct assessment of the method. Furthermore, to facilitate the direct comparison of the TT solver with its classical counterpart, the fractional step method described above is implemented twofold. The first implementation follows the TT methodology, while the second refers to a classical FD implementation, which is verified against literature and finite-volume results generated with OpenFOAM. The interested reader finds more information in H.

4. Applications

In this section, we present a detailed comparison between our TT solver and a reference classical FD solver for the paradigmatic 2D cylinder flow, considering both a non-rotating and a rotating cylinder. We start by introducing the specific problem setup of the cylinder and its discretization. Further, we introduce flow characteristics and diagnostic quantities used to evaluate solution accuracy and computational efficiency. Results of the first Section 4.1 are dedicated to the non-rotating

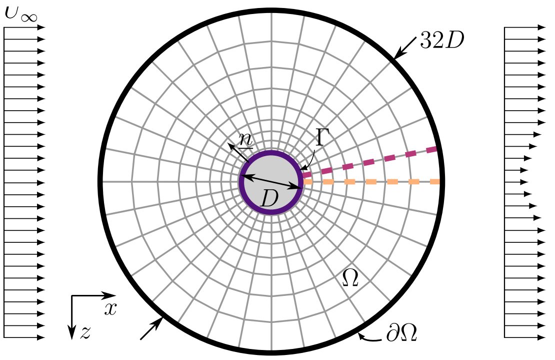

Fig. 7. Dimensionless illustration of the investigated, discretized 2D cylinder case with the grid lines indicated in light-gray and the exterior (interior) boundary of the physical domain Ω indicated in black (magenta) (cf. A).

cylinder in steady ($Re = 20$) and transient ($Re \geq 50$) regimes. The subsequent Section 4.2 discusses the transient flow around the rotating cylinder ($Re = 100$). Section 4.3 complements the numerical results with insights into computational performance. All runs were carried out on 8 cores of an AMD EPYC 9654 CPU and 32 GB of RAM, except stated otherwise.

Fig. 7 depicts the test case for a cylinder that features a unit diameter D boundary Γ which is centered in a circular domain Ω with the boundary ∂Ω placed with diameter $32D$.

The spatial discretization refers to a block-structured O-type grid, as shown by the light-gray grid lines in Fig. 7. Three successively refined grids, each equidistant in the circumferential direction, are considered:

$$\begin{aligned} \mathcal{G}_1 &: N = 256 \times 256, \; n_\eta = 8 \wedge n_\xi = 8 \,, \\ \mathcal{G}_2 &: N = 256 \times 512, \; n_\eta = 8 \wedge n_\xi = 9 \,, \\ \mathcal{G}_3 &: N = 512 \times 512, \; n_\eta = 9 \wedge n_\xi = 9 \,, \end{aligned} \tag{28}$$

with minimal grid spacings at the cylinder surface, each taken as the smaller of the normal spacing or the circumferential arc length, that are approximately $h_{\min} \approx 6.78 \times 10^{-3}$, $h_{\min} \approx 6.14 \times 10^{-3}$ and $h_{\min} \approx 3.39 \times 10^{-3}$, respectively. Here, $h_{\min} = \tilde{h}_{\min}/D$ denotes the smallest step size in the physical domain which usually occurs at the interior boundary in the radial direction. The grid points are specified in E. While the grids can be constructed directly within the TT format as outlined in Section 3.2.2, in this study we construct the grid points (x_{ij}, z_{ij}) and the associated metric terms in a classical way and subsequently encode them into TT-vectors. This ensures comparability between FD and TT-based solvers, which therefore operate with identical coordinate transformations. The computational overhead due to the transformation of the classical grid into the TT format is neglected here. Yet, its advantageous efficiency should be exploited for any other application.

The assessment of transient flow phenomena employs the Strouhal number

$$St = \tilde{\omega} D / 2\pi U_\infty \,, \tag{29}$$

where $\tilde{\omega}\,[\mathrm{rad/s}]$ is the circular frequency of the oscillating lift force $\tilde{F}_\mathrm{L}$. The lift force $\tilde{F}_\mathrm{L}$ is defined as the force per unit length acting perpendicular to the approach flow, whereas the drag force $\tilde{F}_\mathrm{D}$ acts parallel to it. The corresponding non-dimensional coefficients read

$$C_\mathrm{L} = 2\tilde{F}_\mathrm{L}/(\tilde{\rho} U_\infty^2\, D) \quad \text{and} \quad C_\mathrm{D} = 2\tilde{F}_\mathrm{D}/(\tilde{\rho} U_\infty^2\, D)\,. \tag{30}$$

The predictive accuracy is assessed by the L_2-norm of a scalar field $s(\xi, \eta)$, which accounts for the variable grid spacing across the domain, i.e.,

$$||s||_{\mathrm{L}_2} = \sqrt{\int_{\Omega_0} |s(\xi,\eta)|^2 J \, d\xi d\eta} \approx \sqrt{\delta\xi\delta\eta |s_{ij}|^2 J_{ij}}\,. \tag{31}$$

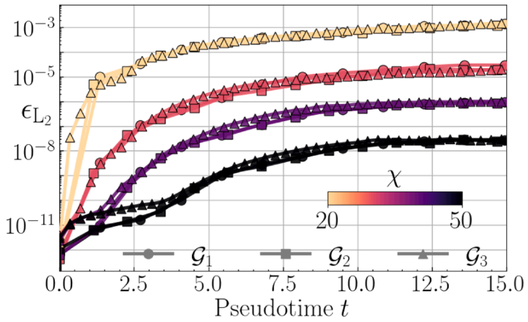

Fig. 8. Grid refinement study at $Re = 20$ under fixed bond dimension constraints. Evolution of the relative L_2 error, ϵ_{L_2}, between the FD and TT solutions is shown as a function of nondimensional pseudotime t for grids $\mathcal{G}_1$ (circles), $\mathcal{G}_2$ (squares), and $\mathcal{G}_3$ (triangles) evolved with the nondimensional pseudotime steps are $\delta t_1 = 4.5 \times 10^{-3}$, $\delta t_2 = 3.8 \times 10^{-3}$, and $\delta t_3 = 1.1 \times 10^{-3}$ respectively. Line colors correspond to the bond dimension χ as indicated in the colorbar. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Consequently, a relative difference between a TT solution s and a classical solution s reads

$$\epsilon_{\mathrm{L}_2} = \frac{||s - s||_{\mathrm{L}_2}}{||s||_{\mathrm{L}_2}}\,, \tag{32}$$

and is used to evaluate the error of the TT format. We also consider the maximal difference error

$$\epsilon_{\max} = \|s - s\|_\infty\,. \tag{33}$$

Moreover, time averages are denoted by an overline, e.g., the error $\overline{\epsilon_{\mathrm{L}_2}}$. To quantify the compression achievable through the TT representation, we introduce the number of variables parameterizing the solution (NVPS), which is the total number of values needed to be stored to represent a state [36]. The NVPS of a TT-vector depends on the dimensions of its indices and is computed by

$$\mathrm{NVPS}_\mathrm{TT} = 2\sum_{k=1}^{n} \alpha_{k-1}\alpha_k\,, \tag{34}$$

where α_k denotes again the bond index between the k-th and $(k+1)$-th TT-core, and $n = \log(N)$ the number of TT-cores. Therefore, the NVPS for a TT-vector grows only as $\mathcal{O}(n\chi^2)$ [36]. In contrast, in a classical FD description, the solution vector length grows exponentially as $\mathcal{O}(2^n)$.

4.1. Non-rotating cylinder

Within this section, the non-rotating cylinder is modeled by zero velocity conditions such that $\underline{v}_\Gamma = 0$ in Eq. (12). The initial Section 4.1.1 compares the INSE solutions (10–12) obtained from the classical FD method with TT-based solutions for the steady state flow at $Re = 20$, and scrutinizes the accuracy of the TT's steady-state solution for several bond dimensions. The subsequent Section 4.1.2 investigates the transient regime for increasing Reynolds numbers, where emphasis is given to the ability of TT to accurately predict vortex shedding and the related oscillating forces.

4.1.1. Steady flow (Re = 20)

To study the effects introduced by the compression inherent to the TT format, the differences between both strategies are analyzed for increasing bond dimensions χ. We note that the bond dimension is imposed uniformly on the TT-vectors of the two cartesian velocities u, v, and the pressure p, such that each field has the same value of χ. For the sake of completeness, the accuracy of the classical FD code is verified against a frequently used finite volume procedure in H. The simulations are started from a uniform initial condition with $\underline{v}_{\mathrm{init.}} = (U_\infty, 0)^\intercal$.

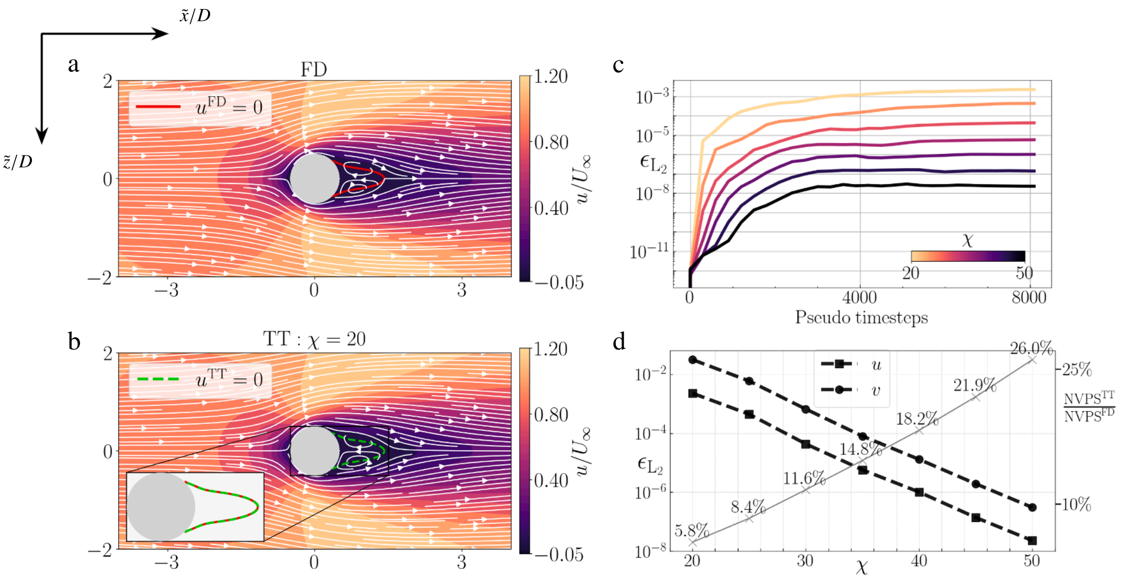

Fig. 9. Flow around the cylinder at $Re = 20$ using the medium-fine grid $\mathcal{G}_2$ and $\delta t_2 \approx 3.8 \times 10^{-3}$. Streamlines and horizontal velocity (u) contours for the steady-state flow, predicted by (a) the classical FD method and (b) the TT method with $\chi = 20$. The $u = 0$ isoline is highlighted in solid red for FD and dashed green for TT, with an inset in panel (b) showing a close-up comparison of both isolines. Predictive difference ϵ_{L_2} (32) between the velocity magnitude field (c) and the individual Cartesian velocity components u, v (d) obtained from the TT and the classical FD simulations. Panel (c) shows the temporal evolution for different bond dimensions χ. Panel (d) depicts the steady state results for u (circles) and v (squares), supplemented by the NVPS fraction for each χ (crosses) to indicate the corresponding compression rate. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

In order to obtain the steady-state solution, we solve the dimensionless Navier-Stokes equations by solving the pseudo-transient initial value problem. In this context, time t is interpreted as pseudotime that keeps track of the duration of the evolution from the initial state. We begin by examining the scaling behavior of the TT approach under grid refinement for the steady case at $Re = 20$. To this end, we simulate the flow on our three progressively finer grids, $\mathcal{G}_1$, $\mathcal{G}_2$, and $\mathcal{G}_3$, while keeping the bond dimension χ fixed at several levels. For each grid $\mathcal{G}_k$, we compute the relative error ϵ_{L_2} with respect to the corresponding FD reference solution obtained on the same grid. For the stability criterion of the convective term, a maximum velocity as $U_{\max} = \gamma U_\infty$ is estimated following the usual practice in the literature (cf. Ref. [75]); here we adopt $\gamma = 1.5$. Accordingly, for each grid $\mathcal{G}_k$ we choose a pseudotime step $\delta t_k = \tilde{\delta t}_k U_\infty / D$ such that the resulting convective Courant number remains below unity with specific values given by $\delta t_1 = 4.5 \times 10^{-3}$, $\delta t_2 = 3.8 \times 10^{-3}$, and $\delta t_3 = 1.1 \times 10^{-3}$ respectively.

Fig. 8 illustrates the evolution of the relative L_2 error toward the steady state, plotted against the pseudotime for a duration of 15 passage times (D/U_∞). For a fixed bond dimension χ, the error trajectories across the grids collapse onto nearly identical curves. This indicates that the approximation error is primarily governed by the bond dimension and remains effectively independent of the grid resolution N within the investigated range.

We now proceed with a detailed analysis of the flow fields and compression properties using the representative medium–fine grid $\mathcal{G}_2$. While the previous grid-refinement study included the most computationally demanding grid $\mathcal{G}_3$ and was thus limited to 15 passage times, the use of $\mathcal{G}_2$ allows us to examine a more extended evolution. To this end, we extract solutions after 8100 pseudotime steps. This corresponds to more than 30 passage times (D/U_∞).

Fig. 9(a) displays the steady-state streamlines and horizontal-velocity (u) contours at $Re = 20$ predicted by the classical FD method, while Fig. 9(b) shows the corresponding TT solution with $\chi = 20$. In both panels the $u = 0$ isoline is depicted, solid red for the FD and dashed green for the TT. Moreover, Fig. 9(b) includes an inset that magnifies these isolines for direct comparison. Qualitatively, the predicted fields show excellent agreement, and the classical Föppl vortex pair behind the cylinder is correctly captured.

Fig. 9 (c) depicts the evolution of the relative difference (error) ϵ_{L_2} between the velocity magnitude obtained from TT-based simulations and FD predictions, evaluated across a range of fixed bond dimensions χ. The error increases during the initial transient phase before stabilizing after around 25 passage times at distinct levels for each χ, remaining below 2×10^{-3}. The maximum difference in u is bounded by $\epsilon_{\max} \leq 9.9 \times 10^{-3}$ ($\chi = 20$), decreasing to 2.2×10^{-7} ($\chi = 50$). Similarly, the v-field satisfies $\epsilon_{\max} \leq 1.3 \times 10^{-2}$ ($\chi = 20$), dropping to $\epsilon_{\max} \leq 7.9 \times 10^{-8}$ ($\chi = 50$). The negligible lift coefficient of $|C_{\mathrm{L}}| = 1 \times 10^{-4}$ for the FD solver exhibits its largest deviation w.r.t. the TT solution at $\chi = 25$ with $|C_{\mathrm{L}}| = 3 \times 10^{-3}$. For all the bond dimension $\chi \geq 30$ the lift coefficient is upper bounded by $|C_{\mathrm{L}}| \leq 1.7 \times 10^{-4}$, closely resembling the FD solution. Moreover, in the energy norm commonly used to assess ROMs (cf. Ref. [83]), the steady-state relative error in total kinetic energy $E_{\mathrm{kin}} = 1/2 \, \|\underline{v}\|^2_{\mathrm{L}_2}$ compared to FD decays rapidly with bond dimension, from 7.65×10^{-5} at $\chi = 20$, to 7.16×10^{-8} at $\chi = 35$, and down to 3.62×10^{-11} at $\chi = 50$. Fig. 9 (d) shows the final differences in u and v after 8100 time steps.

In general, a remarkable agreement between the TT and FD solution is observed. It is evident that with approximately 5.8% of the NVPS, the TT-based approach achieves a relative error of 3×10^{-3} (or 0.3%) in the velocity magnitude, achieving an outstanding 20-fold compression. Increasing the NVPS fraction to approximately 15% further reduces the relative error by three magnitudes to 5.8×10^{-6}. In the most demanding case with $\chi = 50$, the method still requires only around 1/4 of the NVPS while achieving highly accurate solutions with relative errors on the order of 10^{-8}. An extrapolation of the errors suggests that the accuracy can in principle be reduced to machine precision by further increasing the bond dimension. More details will be found in the analysis of the time-dependent flow in the following section.

4.1.2. *Transient flow*

We now proceed to analyze unsteady flow regimes that emerge for $Re \geq 50$. Our goal is to demonstrate that TT can not only efficiently

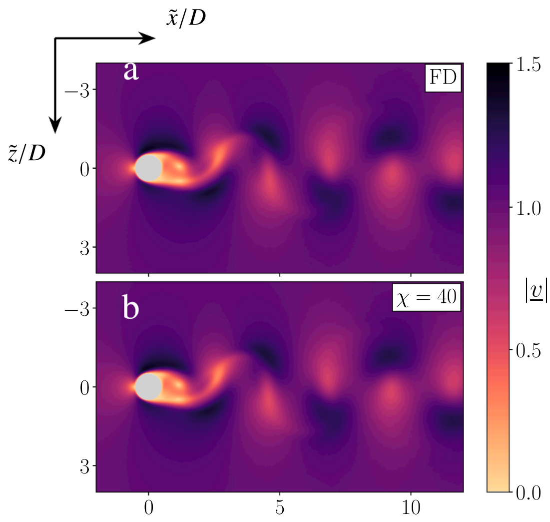

Fig. 10. Snapshots of velocity magnitude $|\underline{v}|$ extracted at $t = 84.85$ from TT in (a) ($\chi = 40$) and a classical FD (b) simulation at $Re = 150$ when using the coarse grid $\mathcal{G}_1$.

capture simple steady-state flows but also faithfully reproduce transient dynamics. In particular, we investigate the trade-off between parametrization (NVPS) complexity and pointwise solution errors, noting that exact local accuracy may not be required to capture the essential dynamics, aligning with the core idea of ROMs. The simulations apply the two coarser grids $\mathcal{G}_1$ and $\mathcal{G}_2$ with a time step of $\delta t_1 = \tilde{\delta t}_1\, U_\infty/D \approx 3.4 \times 10^{-3}$ and $\delta t_2 = \tilde{\delta t}_2\, U_\infty/D \approx 3.1 \times 10^{-3}$, respectively. Here, the maximum velocity is estimated with $\gamma = 2$, i.e., $U_{\text{max}} = \gamma U_\infty$, so that the resulting Courant number remains below unity for all Re considered. Expecting a Strouhal number of $St \approx 0.2$ corresponds to a temporal resolution of approximately 60 time steps per period. Two algorithmic modifications are employed to compute the transient flows: (1) we trigger transient effects by employing asymmetries within initial conditions. To this end, we add an initial perturbation $u_{\text{pert.}}$ to the initial velocity field, which consists of a 5% velocity modulation (as given in G) and (2) we apply a preconditioning of the Poisson equation within the TT format, allowing to efficiently simulate higher bond dimensions [32].

Fig. 10 displays snapshots of the TT for $\chi = 40$ and the FD-predicted velocity magnitudes $|\underline{v}|$ for $Re = 150$. Results refer to the coarse grid $\mathcal{G}_1$ at around 84.85 passage times, i.e., approximately 2.5×10^4 time steps when the flow has reached a periodic state.

Qualitatively, both velocity fields are almost indistinguishable from each other, demonstrating that the TT simulation at $\chi = 40$ is capable of capturing the Kármán vortex street similar to the classical approach. Fig. 11 illustrates the corresponding pointwise differences of the velocity magnitude $|\underline{v}|$, defined as $||\underline{v}^{\text{TT}}| - |\underline{v}^{\text{FD}}||$, in the region $\tilde{x}/D \in [-2, 12]$, $\tilde{z}/D \in [-4, 4]$. The bond dimension χ increases from top to bottom across the panels, with $\chi = 40$, 60, and 80. Results are again evaluated after around 2.5×10^4 time steps, when the flow reflects a periodic behavior in time.

Fig. 11 demonstrates that each increase of χ by 20 bond units reduces the local deviation by roughly two orders of magnitude.

For instance, at the location $(\tilde{x}/D, \tilde{z}/D) = (8, 2)$ (marked by '×' in Fig. 11), the deviation reads 5.2×10^{-3}, 6.3×10^{-5}, and 9.7×10^{-8} for increasing values of χ. This aligns with the convergence of TT toward FD in the limit of increasing χ. For the given system, the maximum possible bond dimension is $\chi = 256$. With a reduction to $\chi = 80$ and a corresponding 87% of NVPS compared to FD, a maximum error of 2.54×10^{-6} in the velocity magnitude $|\underline{v}|$ is recovered, cf. Fig. 11 (c). Nevertheless, lower χ values are sufficient to capture the essential physics for this Re, as it was concluded before. This is confirmed by the error patterns in the cylinder's wake. As expected, the error patterns are aligned to the wake and appear similar for different values of χ. This indicates a consistent resolution of the dynamics with only differences in the resolution accuracy, and underlines the suitability of the TT-based approach as a ROM.

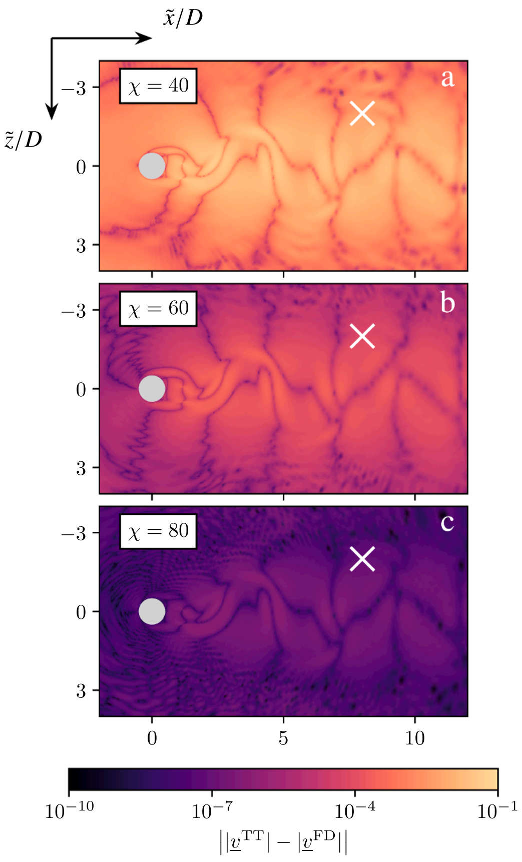

Fig. 11. Pointwise difference of the velocity magnitude $|\underline{v}|$ predicted by the TT and classical FD method for the flow around a non-rotating cylinder at $Re = 150$ on the coarse grid $\mathcal{G}_1$. White crosses indicate the probe location $(\tilde{x}/D, \tilde{z}/D) = (8, 2)$, for the comparison of velocity components.

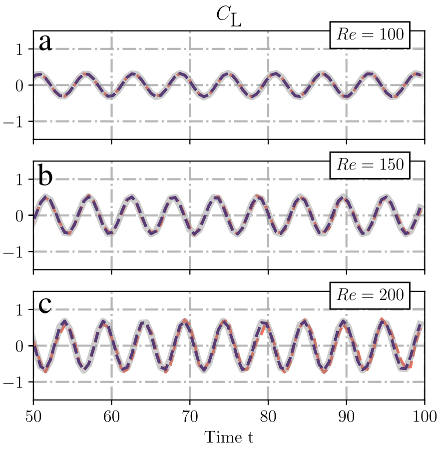

Fig. 12. Temporal evolution of the lift coefficient C_{L} for the flow around a non-rotating cylinder on the medium grid $\mathcal{G}_2$. Results predicted by the TT method are shown for bond dimensions $\chi = 40$ (orange dashed) and $\chi = 60$ (magenta dashed), with the FD reference solution (gray solid).

The capability of TTs to predict the relevant integral force quantities is assessed next. To this end, the Strouhal number is determined

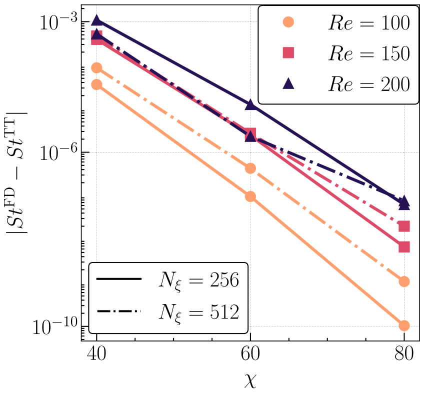

Fig. 13. Absolute difference of the Strouhal number St predicted by the TT and the classical FD method for the flow around a non-rotating cylinder on the two coarser grids $\mathcal{G}_1$ (straight) and $\mathcal{G}_2$ (dot-dashed).

from the oscillations of the lift force coefficient C_L (30). Accordingly, the computation of C_L can be performed entirely in the TT format, without requiring an expensive contraction to the full classical representation. Specifically, it involves extracting one-dimensional TT-vectors of η-constant lines adjacent to the surface Γ (cf. Ref. [36]), combining TT operations outlined in Section 2, and applying a summation (average-like) operation as reported in [38]. Furthermore, the mapping from the computational domain Ω_0 to the physical domain Ω must again be taken into account, as detailed in F.

Fig. 12 presents the temporal evolution of C_L over 50 passage times for $Re = 100, 150$, and 200. The figure compares the results obtained from the TT approach on the medium grid $\mathcal{G}_2$ for two different bond dimensions, i.e., $\chi = 40$ (orange) and $\chi = 60$ (magenta), and the FD method (gray). The TT results clearly capture distinct oscillations. However, the agreement between $\chi = 40$ and FD slightly deteriorates as Re increases. Notably, the phase seems to be shifting during the simulation of the higher Re number in combination with $\chi = 40$. This is attributed to an expected increase of the required bond dimension with Re at the same accuracy, as also seen in Ref. [36].

To study how the observed deterioration in Fig. 12 affects the accuracy of the predicted Strouhal numbers, the angular frequencies of the lift forces were evaluated for a range of bond dimensions for the two coarser grids $\mathcal{G}_1$ and $\mathcal{G}_2$, respectively. Results were obtained from successive (interpolated) zero crossings of C_L over 50 passage times ($t \in [50, 100]$); these crossings were then used to compute the averaged shedding period.

The difference between Strouhal numbers obtained from the TT simulations and the corresponding FD are presented in Fig. 13 for bond dimensions $\chi = 40$, 60 and $\chi = 80$. As a point of reference, we obtain $St \approx 0.1916$ for $Re = 200$ from the FD results using $\mathcal{G}_2$ which presents a 10^{-3} absolute error w.r.t. the value of $St \approx 0.1977$ reported in Ref. [75]. This allows us to conclude that the prediction by TT is indeed generally very accurate, as confirmed by Fig. 13. Results demonstrate that as χ increases, the difference in the predicted Strouhal number decreases monotonically, consistent with the observed decrease in pointwise field differences shown in Fig. 11. Moreover, we observe the expected trend where, for a fixed system size and bond dimension, the error in St increases with growing Re. Despite this, the TT-based model is particularly promising, as increases in the bond dimension χ can provide increasingly accurate Strouhal number estimates, which may be used in a Richardson extrapolation scheme to approximate the exact value.

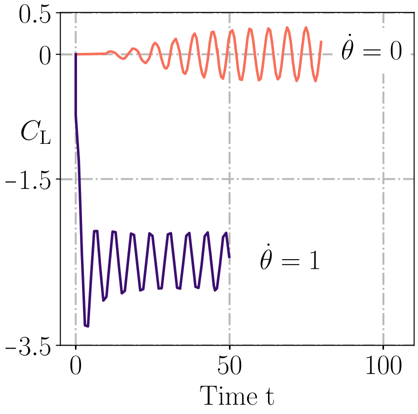

Fig. 14. Temporal evolution of the TT-predicted ($\chi = 40$) lift-force coefficient C_L for the flow around a rotating cylinder ($U_\Gamma = U_\infty \dot{\theta}$) at Reynolds $Re = 100$ on the coarse grid $\mathcal{G}_1$.

Table 2
Comparison of $\overline{C}_\mathrm{L}$ and lift St values for the flow around a rotating cylinder at $Re = 100$, $\dot{\theta} = 1$. Values reported in Ref. [85] (bold) are compared with values obtained using the TT method with different bond dimensions χ and the classical FD solver.

Discretization (interior)	χ	$\overline{C}_\mathrm{L}$		St	
		abs	%	abs	%
241 × 241	–	**2.4881**	–	**0.1655**	–
256 × 256	40	2.4919	0.15	0.1662	0.42
256 × 256	50	2.4908	0.11	0.1663	0.49
256 × 256	60	2.4908	0.11	0.1663	0.49
256 × 256	FD	2.4908	0.11	0.1663	0.49

4.2. Rotating cylinder

As a further demonstration of the framework's versatility, we examine the case of a cylinder rotating with the non-dimensional rotation rate $\dot{\theta} = D\,\dot{\tilde{\theta}}/(2\,U_\infty) = 1$, which gives rise to the widely studied phenomenon known as the Magnus effect. To this end, the boundary condition of the cylinder is changed to a no-slip condition with a non-vanishing prescribed tangential velocity

$$U_\Gamma = \dot{\theta} U_\infty \,. \tag{35}$$

The components of a normalized velocity $\underline{v}_\Gamma$ at the discrete surface points are given by

$$\begin{aligned} u_{\Gamma_j} &= \dot{\theta} \cdot \sin(2\pi j\, \delta\xi) \qquad \text{and} \\ v_{\Gamma_j} &= -\dot{\theta} \cdot \cos(2\pi j\, \delta\xi)\,, \end{aligned} \tag{36}$$

where $j = 0, \ldots, N_\xi - 1$. The initial conditions are set to a uniform flow field $\underline{v}_{\mathrm{init.}} = (U_\infty, 0)^\intercal$ with no added perturbation, as employed in Section 4.1.1, since the asymmetry of the boundary conditions is sufficient to trigger the vortex shedding at $\dot{\theta} = 1$ [84].

The application of Eq. (36) realizes a counter-clockwise rotation. As a consequence of the rotation, the cylinder is subject to time-averaged circulation, which mainly results in a shift of the oscillating lift force. Fig. 14 shows the temporal evolution of the lift-force coefficient C_L (30) for $\dot{\theta} = 1$ (magenta) in comparison to the non-rotating case ($\dot{\theta} = 0$, orange) predicted by the TT method for $Re = 100$ on the coarse grid $\mathcal{G}_1$.

The maximum velocity is estimated with $\gamma = 3$, i.e., $U_{\max} = \gamma U_\infty$, so that the resulting Courant number is below unity. Accordingly, the employed time step refers to $\delta t_1 = \delta\tilde{t}_1 U_\infty / D = 2 \times 10^{-3}$. The TT-predicted time-averaged lift force $\overline{C}_\mathrm{L}$ and the lift Strouhal number are outlined in Table 2 for three different bond dimensions χ = 40, 50, and 60.

These predictions compare favorably with our FD-predicted and literature-reported values extracted from [84,85] which were obtained for a similar resolution.

It is noteworthy, that the TT approximation used for the low relative error of 0.15% for $\overline{C}_\mathrm{L}$ and 0.42% for the Strouhal prediction just uses $\chi = 40 \sim 30\%$ of NVPS on the coarse grid $\mathcal{G}_1$. A convergence to-

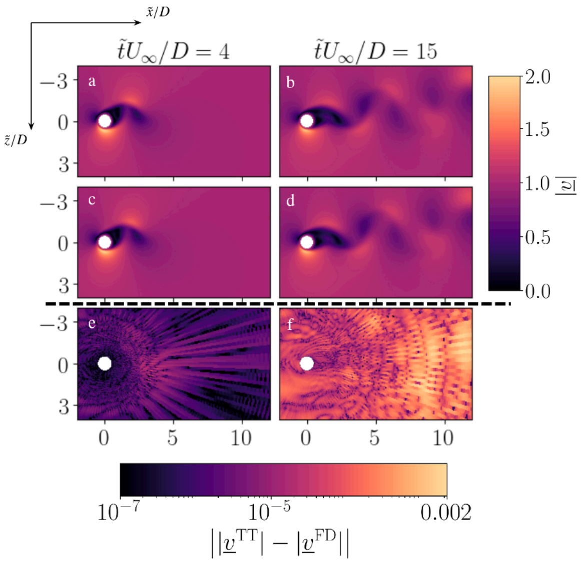

Fig. 15. Velocity magnitude snapshots predicted by the FD method (a,b) and the TT method ($\chi = 40$, c,d) on the coarse grid $\mathcal{G}_1$ for the flow around a rotating cylinder ($\dot{\theta} = 1$) at $Re = 100$. Panels (e) and (f) show the corresponding pointwise errors between both methods. All error values below 10^{-7} are rendered as black in the visualization.

wards the corresponding FD solver is observed as χ increases, whereas convergence toward literature-reported values is not observed, which might be attributed to differences in the simulation methods and grids. Nonetheless, we conclude that the TT-predicted values exhibit excellent agreement with the literature while reducing the computational effort at least by a third.

Fig. 15 shows the corresponding velocity magnitude fields predicted by the FD solver (a,b) and the TT solver with $\chi = 40$ (c,d), and the respective pointwise error distribution (e,f) analogously to Fig. 11. The depicted snapshots clearly describe the formation of the wake and include the initial transient state ($t \approx 4$) and the final periodic state ($t \approx 15$). The deflection of the flow is visible and differs from the structure in Fig. 10, which is due to the rotation of the cylinder. We observe well-defined vortices are convected downstream in the wake region. An increase in errors is detected in time, which we attribute to the accumulation of numerical noise and truncation errors in the TT format.

4.3. Computational characteristics & performance

In this section, we examine the computational effort of the algorithm. We confine the analysis to the assessment of the transient, non-rotating cylinder case at $Re = 150$ (cf. Section 4.1.2) and assume that changes of the physical model are not affecting the parameterization of the numerics.

Following the fractional step approach introduced in Section 3.3, the three most computationally expensive operations are identified. These are (i) computing the RHS of the rearranged momentum Eq. (24), involving elementwise multiplications of TT-vectors to evaluate the convection term, which scales as $\mathcal{O}(\chi^4)$, (ii) solving an LSE (cf. Eq. (24)) for both components of $\underline{v}^*$, scaling as $\mathcal{O}(\chi^3)$, and (iii) solving the Poisson Eq. (25) with $\mathcal{O}(\chi^3)$. The given scalings are taken from Table 1. The breakdown of the absolute computational effort required by these three most expensive operations is evaluated over 2.4×10^4 time steps and displayed in Fig. 16.

As evident in Fig. 16, each computational effort (i)-(iii) grows with rising χ while the ratios between the operations remain nearly constant over time. The results of Fig. 16 indicate that the wall-clock time per time step remains nearly constant throughout the simulation. We attribute this behavior to the Poisson preconditioner, which enables a rapid—if not fully optimal—pressure projection even when the Poisson system is ill-conditioned and the RHS is noisy. In contrast, an unpreconditioned DMRG solver performs less effectively with the truncation and rounding errors that accumulate over time, leading to an increase in the number of DMRG sweeps and, consequently, a higher cost per step.

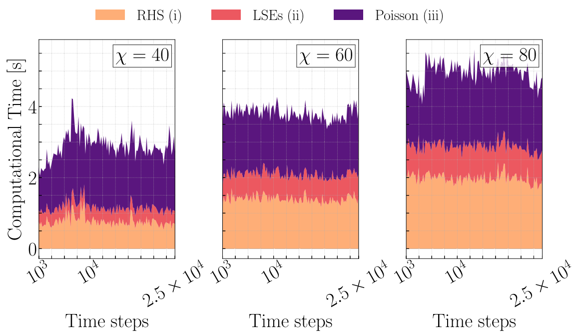

Fig. 16. Illustration of the computing time of the three most expensive algorithmic tasks per time step for the flow around a cylinder at $Re = 150$ on the coarse grid $\mathcal{G}_1$ for three different bond dimensions.

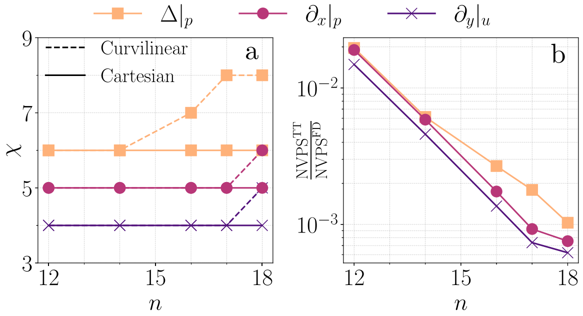

Fig. 17. Compressibility analysis of the generalized TT-operators. Panel (a) depicts the required bond dimension of the curvilinear and Cartesian operators. Panel (b) shows the ratio of NVPS needed to represent the curvilinear operators in TT and sparse format. Please note that the lines connecting the data points are added solely for better visualization of the emerging trends.

With a focus on the individual operations, solving the linear system (ii), for $\underline{v}^*$ requires the least computational effort, significantly less than solving the Poisson problem (iii), despite both being LSEs (cf. Fig. 16). This discrepancy can be attributed to the better conditioning of the left-hand side matrix $\left(1 + \frac{\delta t}{Re}\nabla^2\right)$ in the momentum update, resulting in faster convergence in the variational optimization. Moreover, we observe that computing the RHS (i) via elementwise multiplications of TT-vectors has roughly the same computational effort as solving the Poisson problem (iii).

All these operations scale with the bond dimension χ of the fluid field representations. However, it is important to emphasize that the overall complexity also depends on the bond dimensions of the TT-matrices representing the differential operators (cf. Table 1). Accordingly, Fig. 17 (a) illustrates the bond dimension of the curvilinear operators as TT (dashed line) w.r.t. the corresponding operators defined on a Cartesian mesh (solid line). As it is evident in this figure, the generalized operators require a comparable bond dimension, increasing only slightly for larger grid sizes. Furthermore, the number of NVPS of the TT ($\mathrm{NVPS}^{\mathrm{TT}}$) exhibits a remarkable reduction – up to three orders of magnitude – compared to the conventional sparse matrix representation (cf. Fig. 17 (b)). For this comparison, we consider a sparse matrix size of $\mathrm{NVPS}^{\mathrm{FD}} = 5N$ that represents the central point and its four neighboring points in the second-order central FD approximation of the Laplacian operator.

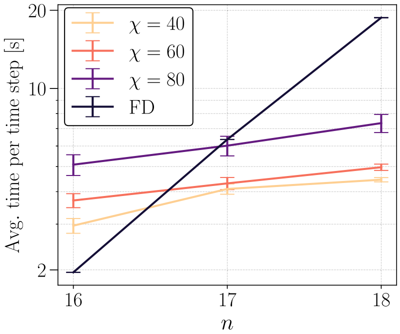

Fig. 18. Scaling of the average wall-clock time per iteration (sampled every 200th iteration) for the $Re = 150$ simulations discussed in Section 4.1.2.

Diagonal neighbors are not considered, due to the orthogonality of the grid.

The wall-clock time of our TT-based approach is compared to the classical FD in Fig. 18. As it is depicted in this figure, both approaches reach similar computational times for the given system sizes. The slopes of the TT simulations remain constant and the growth in bond dimension χ for increasing system size is only reflected in a y-axis offset. On the other hand, the computational complexity of the classic FD scales exponentially with the number of tensors (TT-cores) n. Despite the different LSE solving strategies applied to TT and FD, such as the non-variational approach or the absence of preconditioning in the FD, the extracted conclusions from Fig. 18 remain valid. An improvement in the FD computation times would shift the break-even point along the x-axis toward higher tensor counts. Given that industrial CFD cases routinely exceed 20 million NVPS ($\approx$ 24 tensors), we anticipate that tensor-network methods like the one proposed here could deliver substantial computational gains.

5. Conclusions & outlook

In this work, we demonstrated that the TT format can serve as an efficient ROM framework for simulating INSE in curvilinear coordinates. To the best of our knowledge, this study presents the first application of tensor network methods to the encoding and simulation of fluid fields on non-uniform, body-fitted meshes, extending their use beyond the uniform discretizations commonly considered in previous works [36,38,53]. This development marks a significant advancement toward the realization of an industrial-grade Quantum Computational Fluid Dynamics toolbox, as body-fitted meshes are integral to many established CFD programs [75,86,87].

The methodology presented here ranges from outlining the mesh generation process, formulated within the TT format, to describing the curvilinear transformation of differential operators using known TT operations, and includes the efficient readout of relevant quantities of interest, such as the lift coefficient and the Strouhal number, directly in the TT format [38]. Furthermore, the approach is reinforced through the ghost point strategy for boundary treatment that eliminates the need for penalty terms in the system of equations. This enables the versatile application of common engineering boundaries within the TT format. Similarly, the fractional-step method avoids determining additional degrees of freedom required to enforce the pressure-velocity coupling as realized in Refs. [35,38,88,89].

This allows the implementation of various flow scenarios around an immersed cylinder, ranging from Stokes flows to vortex streets with instabilities at large Reynolds numbers, and even the Magnus effect induced by cylinder rotation. In all cases, the TT-based solver demonstrates outstanding agreement with a significant reduction up to 20-fold of the NVPS required to represent fields and up to 1000-fold for the operators.

This work lays the foundation for future investigations into how obstacle geometry, mesh generation, grid properties (e.g., orthogonality), and parallelization [90] further increase the TT's operational efficiency. In particular, certain mesh types-such as elliptic grids-may naturally yield low-rank, geometry-aware operators, offering potential for shape optimization and other industrial applications. To address high-Reynolds number regimes, the second-order central discretization can be extended to upwind-biased [24,80] or WENO-type schemes [40] to ensure stability in convection-dominated regimes. The framework generalizes naturally to three-dimensional configurations, which are planned for future investigations (see Appendix A). A compelling next step is to investigate laminar and turbulent flows around immersed ellipsoids and NACA airfoils [91]. In such cases, the curvilinear approach is expected to be particularly advantageous compared to standard IBM.

CRediT authorship contribution statement

Nis-Luca van Hülst: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization; **Pia Siegl:** Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization; **Paul Over:** Writing – review & editing, Writing – original draft, Visualization, Methodology, Investigation, Formal analysis, Data curation, Conceptualization; **Sergio Bengoechea:** Writing – review & editing, Writing – original draft, Visualization, Methodology, Investigation, Formal analysis, Data curation, Conceptualization; **Tomohiro Hashizume:** Writing – review & editing, Writing – original draft, Visualization, Supervision, Software, Methodology, Investigation, Conceptualization; **Mario Guillaume Cecile:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Methodology, Investigation, Conceptualization; **Thomas Rung:** Writing – review & editing, Supervision, Resources, Project administration, Methodology, Funding acquisition, Conceptualization; **Dieter Jaksch:** Writing – review & editing, Supervision, Resources, Project administration, Methodology, Funding acquisition, Conceptualization.

Data availability

A software example for simulating the 2D heat equation on the employed cylindrical grid is available via Ref. [92]. The data that support the findings of this study are available upon reasonable request from the authors and at Ref. https://doi.org/10.25592/uhhfdm.17687.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This publication and the current work have received funding from the European Union's Horizon Europe research and innovation program (HORIZON-CL4-2021-DIGITAL-EMERGING-02-10) under grant agreement No. 101080085 QCFD. PS acknowledges financial support by the DLR-Quantum-Fellowship Program. DJ and TH are partly funded by the Cluster of Excellence 'Advanced Imaging of Matter' of the Deutsche Forschungsgemeinschaft (DFG) -EXC 2056- project ID390715994. DJ acknowledges the support by DFG project "Quantencomputing mit neutralen Atomen" (JA 1793/1-1, Japan-JST-DFG-ASPIRE 2024) and the Hamburg Quantum Computing Initiative (HQIC) project EFRE. The EFRE project is co-financed by ERDF of the European Union and by "Fonds of the Hamburg Ministry of Science, Research, Equalities and Districts (BWFGB)". The numerical data is acquired using the computational cluster Hummel-2 of the University of Hamburg, which is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - 498394658. This work was inspired in part by the engaging discussions and expert input from Airbus, BMW Group, and AWS during the *Quantum Mobility Quest* challenge. The authors thank Philipp Thoma for fruitful discussions.

Appendix A. Discretization and Coordinate Transformation

In Fig. A.19 the two domains of interest are illustrated, i.e., the computational (a) and the physical (b) domain. The former corresponds to a uniformly discretized unit square $\Omega_0 = [0,1] \times [0,1)$, while the latter is discretized using an O-type grid. The coloring highlights the corresponding boundaries of the two domains; for instance, the outer boundary $\partial\Omega$ of the physical domain (shown in black) maps to the top boundary of the unit square (also black).

The computational domain Ω_0 is discretized using $(2^{n_\eta}+2) \times 2^{n_\xi}$ grid points in the $\eta(i)$- and $\xi(j)$-directions, respectively, given by

$$\left.\begin{aligned}\eta_{ij} &= (i+1)\delta\eta\\ \xi_{ij} &= j\delta\xi\end{aligned}\right\}\text{with}\quad \begin{aligned} i &= -1,\dots,2^{n_\eta}\,,\\ j &= 0,\dots,2^{n_\xi}-1\,.\end{aligned} \tag{A.1}$$

Here, $i=-1$ and $i=2^{n_\eta}$ correspond to ghost points representing the top (black) and bottom (magenta) boundaries, respectively. These grid points are not part of the solution domain but are incorporated through the ghost-point approach. The associated grid spacings are $\delta\eta = 1/(2^{n_\eta}+1)$ and $\delta\xi = 1/2^{n_\xi}$. The difference in spacing reflects the boundary conditions: periodicity in the ξ-direction versus non-periodicity in η.

The Cartesian spatial derivatives required by the momentum (10) and continuity (11) equations (INSE) are expressed in the reference coordinate system (ξ,η) by applying the chain rule, e.g., $\partial_x = \partial_x\xi\,\partial_\xi + \partial_x\eta\,\partial_\eta$, and using a coordinate transformation, i.e., $\partial_x\xi = \partial_\eta z/J$, $\partial_x\eta = -\partial_\xi z/J$, $\partial_y\xi = -\partial_\eta x/J$ and $\partial_y\eta = \partial_\xi x/J$, viz.

$$(\partial_x,\partial_z) = \frac{1}{J}\begin{pmatrix}\partial_\eta z & -\partial_\xi z\\ -\partial_\eta x & \partial_\xi x\end{pmatrix}(\partial_\xi,\partial_\eta)^\intercal\,, \tag{A.2}$$

for first spatial derivatives and

$$\Delta = \frac{1}{J}(\partial_\xi,\partial_\eta)\frac{1}{J}\begin{pmatrix} g_{22} & -g_{12}\\ -g_{21} & g_{11}\end{pmatrix}(\partial_\xi,\partial_\eta)^\intercal\,, \tag{A.3}$$

for the Laplace operator. Here, J denotes the Jacobian determinant and g the metric tensor. They are given by the following expressions

$$J = (\partial_\xi x)(\partial_\eta z) - (\partial_\eta x)(\partial_\xi z) = \det\begin{pmatrix}\partial_\xi x & \partial_\eta x\\ \partial_\xi z & \partial_\eta z\end{pmatrix}\,, \tag{A.4}$$

$$g_{11} = (\partial_\xi x)^2 + (\partial_\xi z)^2\,, \tag{A.5}$$

$$g_{22} = (\partial_\eta x)^2 + (\partial_\eta z)^2\,, \tag{A.6}$$

$$g_{12} = (\partial_\xi x)(\partial_\eta x) + (\partial_\xi z)(\partial_\eta z)\,. \tag{A.7}$$

Writing out the Laplacian operator one obtains a sum of seven operator terms

$$\begin{aligned}\Delta = \frac{1}{J}\Bigg\{ &\partial_\xi\left(\frac{g_{22}}{J}\right)\partial_\xi + \frac{g_{22}}{J}\,\partial_{\xi\xi} - \partial_\xi\left(\frac{g_{12}}{J}\right)\partial_\eta - \partial_\eta\left(\frac{g_{12}}{J}\right)\partial_\xi\\ &-2\frac{g_{12}}{J}\,\partial_{\xi\eta} + \partial_\eta\left(\frac{g_{11}}{J}\right)\partial_\eta + \frac{g_{11}}{J}\,\partial_{\eta\eta}\Bigg\}\,.\end{aligned} \tag{A.8}$$

The derivatives w.r.t. ξ and η are approximated with second-order central difference methods. The first derivatives of a generic scalar field ϕ read

$$\partial_\xi\phi\Big|_{ij} = \frac{\phi_{i,j+1}-\phi_{i,j-1}}{2\delta\xi} + \mathcal{O}(\delta\xi^2), \tag{A.9}$$

$$\partial_\eta\phi\Big|_{ij} = \frac{\phi_{i+1,j}-\phi_{i-1,j}}{2\delta\eta} + \mathcal{O}(\delta\eta^2)\,. \tag{A.10}$$

Similarly, the corresponding second derivatives read

$$\partial_{\xi\xi}\phi\Big|_{ij} = \frac{\phi_{i,j+1}-2\phi_{i,j}+\phi_{i,j-1}}{\delta\xi^2} + \mathcal{O}(\delta\xi^2)\,, \tag{A.11}$$

$$\partial_{\eta\eta}\phi\Big|_{ij} = \frac{\phi_{i+1,j}-2\phi_{i,j}+\phi_{i-1,j}}{\delta\eta^2} + \mathcal{O}(\delta\eta^2)\,. \tag{A.12}$$

and finally mixed derivatives are approximated by

$$\begin{aligned}\partial_{\xi\eta}\phi\Big|_{i,j} = &\frac{\phi_{i+1,j+1}-\phi_{i+1,j-1}-\phi_{i-1,j+1}+\phi_{i-1,j-1}}{4\delta\xi\delta\eta}\\ &+\mathcal{O}(\delta\xi^2+\delta\eta^2)\,.\end{aligned} \tag{A.13}$$

Extension to three dimensions. The curvilinear TT framework generalizes naturally to three-dimensional configurations. From an algorithmic perspective, the extension is straightforward since higher-dimensional vector fields are represented by extending the tensor chain to a total length $n = n_x + n_y + n_z$. The fundamental TT algebra and solver routines remain conceptually identical to the 2D case. Mesh generation is treated as an independent preprocessing step and therefore constitutes an exchangeable module, allowing the use of standard 3D mesh generators. Given a body-fitted curvilinear grid, the required three-dimensional metric tensor g_{ij}, Jacobian J, and coordinate derivatives are computed and passed to the TT operator construction. While the transition to 3D increases the number of terms in the discretized equations, the operator assembly in TT form remains unchanged. Regarding computational efficiency, the effectiveness of TT compression depends more on inter-scale correlations than on spatial dimensionality. Recent studies on 3D fluid simulations suggest that comparatively low bond dimensions can capture key flow properties in three dimensions [41,88].

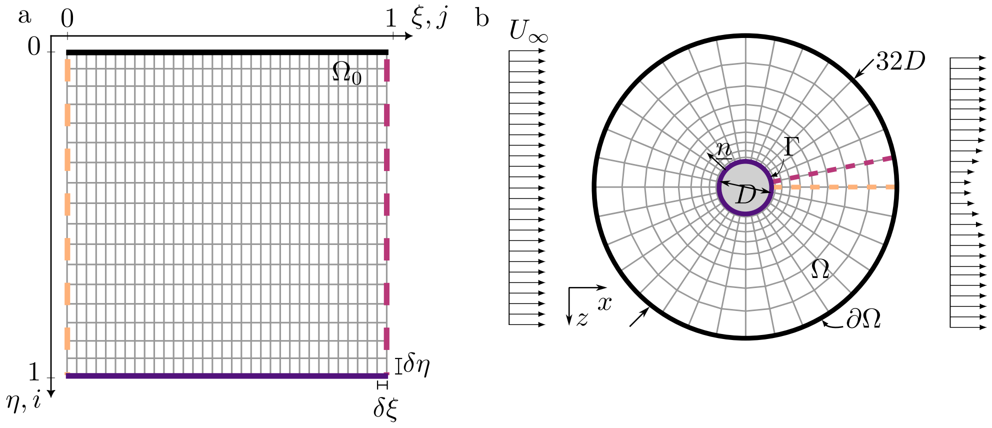

Fig. A.19. Computational domain Ω_0 with uniform discretization (left) and physical domain Ω around a cylinder discretized using an O-type grid (right). In both, light-gray points indicate grid nodes, and colored lines highlight corresponding boundaries.

Appendix B. Sorting Vector to Matrix and Reverse

The column-major ordered vectorization of matrix $A \in \mathbb{R}^{3\times3}$, denoted $\mathrm{vec}(A)$, is

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \tag{B.1}$$

$$\rightarrow \mathrm{vec}(A) = [a_{11}\ a_{21}\ a_{31}\ a_{12}\ a_{22}\ a_{32}\ a_{13}\ a_{23}\ a_{33}]^{\intercal}\ .$$

Appendix C. Read-out of Column or Row

Assuming the two-dimensional scalar field $f_{ij} := f(\xi_{ij}, \eta_{ij}) \in \mathbb{R}^{2^{n_\eta}\times 2^{n_\xi}}$ is represented as a TT, we write

$$f_{ij} \simeq \mathbf{A}[1]^{i_1}\mathbf{A}[2]^{i_2}\cdots\mathbf{A}[n_\eta]^{i_{n_\eta}}\mathbf{A}[n_\eta+1]^{j_1}\cdots\mathbf{A}[n]^{j_{n_\xi}}\ , \tag{C.1}$$

where total number of TT-cores is $n = n_\eta + n_\xi$. Each core $\mathbf{A}[\ell] \in \mathbb{R}^{\alpha_{\ell-1}\times 2\times\alpha_\ell}$ is equipped with a corresponding binary digit $i_k, j_k \in \{0,1\}$ of the row and column indices, respectively.

To extract the I-th row, i.e., the slice f_{Ij}, we fix the first n_η physical indices using the binary representation $(I_1, \dots, I_{n_\eta}) \in \{0,1\}^{n_\eta}$ of I, resulting in the contracted form

$$\begin{aligned} f_{Ij} \simeq\ & \mathbf{A}[1]^{i_1=I_1}\mathbf{A}[2]^{i_2=I_2}\cdots\mathbf{A}[n_\eta]^{i_{n_\eta}=I_{n_\eta}} \\ & \mathbf{A}[n_\eta+1]^{j_1}\cdots\mathbf{A}[n]^{j_{n_\xi}}\ . \end{aligned} \tag{C.2}$$

This procedure effectively contracts the first n_η tensor cores to form a matrix $\boldsymbol{M}_I \in \mathbb{R}^{1\times\alpha_{n_\eta}}$, which is then left-multiplied into the remaining TT-cores

$$f_{Ij} \simeq \boldsymbol{M}_I \cdot \left(\mathbf{A}[n_\eta+1]^{j_1}\cdots\mathbf{A}[n]^{j_{n_\xi}}\right)\ . \tag{C.3}$$

This yields a TT representation of the row $f_{Ij} \in \mathbb{R}^{2^{n_\xi}}$. Similarly, to extract the J-th column, i.e., the slice f_{iJ}, we fix the last n_ξ physical indices using the binary representation $(J_1, \dots, J_{n_\xi}) \in \{0,1\}^{n_\xi}$ of J, resulting in the contracted form

$$\begin{aligned} f_{iJ} \simeq\ & \mathbf{A}[1]^{i_1}\mathbf{A}[2]^{i_2}\cdots\mathbf{A}[n_\eta]^{i_{n_\eta}} \\ & \mathbf{A}[n_\eta+1]^{j_1=J_1}\cdots\mathbf{A}[n]^{j_{n_\xi}=J_{n_\xi}}\ . \end{aligned} \tag{C.4}$$

This procedure effectively contracts the last n_ξ tensor cores to form a matrix $\boldsymbol{M}_J \in \mathbb{R}^{\alpha_{n_\eta}\times 1}$, which is then right-multiplied into the remaining TT-cores:

$$f_{iJ} \simeq \left(\mathbf{A}[1]^{i_1}\mathbf{A}[2]^{i_2}\cdots\mathbf{A}[n_\eta]^{i_{n_\eta}}\right)\cdot\boldsymbol{M}_J\ . \tag{C.5}$$

This yields a TT representation of the column $f_{iJ} \in \mathbb{R}^{2^{n_\eta}}$.

Appendix D. Vector to Tensor Train Decomposition

The decomposition of a vector into a TT is exemplified here analytically for a vector of size $2^3 = 8$ ($n = 3$). As a preliminary requirement, mention the rank core product, denoted by $\bowtie$, which acts as an array multiplication given vector entities as

$$\left[\begin{pmatrix} a \\ b \end{pmatrix}\begin{pmatrix} c \\ d \end{pmatrix}\right] \bowtie \begin{bmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \\ \begin{pmatrix} \gamma \\ \delta \end{pmatrix} \end{bmatrix} = \begin{pmatrix} a \\ b \end{pmatrix} \otimes \begin{pmatrix} \alpha \\ \beta \end{pmatrix} + \begin{pmatrix} c \\ d \end{pmatrix} \otimes \begin{pmatrix} \gamma \\ \delta \end{pmatrix} = \begin{pmatrix} a\alpha + c\gamma \\ a\beta + c\delta \\ b\alpha + d\gamma \\ b\beta + d\delta \end{pmatrix}, \tag{D.1}$$

$$\left[\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}\begin{pmatrix} e \\ f \\ g \\ h \end{pmatrix}\right] \bowtie \begin{bmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \\ \begin{pmatrix} \gamma \\ \delta \end{pmatrix} \end{bmatrix} = \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} \otimes \begin{pmatrix} \alpha \\ \beta \end{pmatrix} + \begin{pmatrix} e \\ f \\ g \\ h \end{pmatrix} \otimes \begin{pmatrix} \gamma \\ \delta \end{pmatrix} = \begin{pmatrix} a\alpha + e\gamma \\ a\beta + e\delta \\ b\alpha + f\gamma \\ b\beta + f\delta \\ c\alpha + g\gamma \\ c\beta + g\delta \\ d\alpha + h\gamma \\ d\beta + h\delta \end{pmatrix}. \tag{D.2}$$

If one now defines a vector $\underline{t} = (1,2,3,4,5,6,7,8)^{\intercal}$, its tensor decomposition is

$$\left[\begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \begin{pmatrix} 0 \\ 1 \end{pmatrix}\right] \bowtie \begin{bmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} & \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ \begin{pmatrix} 0 \\ 0 \end{pmatrix} & \begin{pmatrix} 2 \\ 2 \end{pmatrix} \end{bmatrix} \bowtie \begin{bmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} \\ \begin{pmatrix} 2 \\ 2 \end{pmatrix} \end{bmatrix} = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{pmatrix}. \tag{D.3}$$

Each of the square brackets represents one tensor core $\mathbf{A}[k]$, where $k \in \{0,1,2\}$. Furthermore, the bond dimension and the physical dimension are 2 in all cases.

Appendix E. Grid points in Application

The classically generated grid consists of $N = N_\eta \times N_\xi := 2^{n_\eta} \times 2^{n_\xi}$ interior support points, with indices $i = 0, \dots, N_\eta - 1$ and $j = 0, \dots, N_\xi - 1$, provided as a tensor (outer) product of 1D functions as

$$x_{ij} = \frac{D}{2}\left(\frac{D_{\partial\Omega}}{D}\right)^{\frac{i+1}{N_\eta+1}} \cos(2\pi j\delta\xi)\ , \tag{E.1}$$

$$z_{ij} = \frac{D}{2}\left(\frac{D_{\partial\Omega}}{D}\right)^{\frac{i+1}{N_\eta+1}} \sin(2\pi j\delta\xi)\ . \tag{E.2}$$

Appendix F. Force Computation

The Strouhal number can be computed using different methods, depending on the available data and flow characteristics. In this study, we determine the Strouhal number from the oscillations in the dimensionless lift force F_{L}, following the approach of [75]. The lift force corresponds to the z-projection of the total force as

$$\begin{aligned} F_{\mathrm{L}} &= \underline{e}_z^{\intercal}\int_\Gamma \underline{\underline{\tau}}(\underline{v}, p)\underline{n}\, ds \\ &\approx 2\pi\delta\xi\frac{1}{2}\sum_{j=0}^{N_\xi-1}(\tau_{21}, \tau_{22})\begin{pmatrix} \cos 2\pi j\delta\xi \\ \sin -2\pi j\delta\xi \end{pmatrix}\ . \end{aligned} \tag{F.1}$$

Here, the stress tensor components need to be determined on the cylinder surface via

$$\tau_{21} = \frac{2}{Re}\left(\frac{1}{2}\left(\frac{\partial v}{\partial x}\Big|_\Gamma + \frac{\partial u}{\partial z}\Big|_\Gamma\right)\right), \tag{F.2}$$

$$\tau_{22} = -\left(p\Big|_\Gamma + \frac{2}{3Re}(\nabla\underline{v})\Big|_\Gamma\right) + \frac{2}{Re}\frac{\partial v}{\partial z}\Big|_\Gamma\ , \tag{F.3}$$

where derivatives are computed in the computational domain Ω_0 and then transformed. A similar explicit formula can be derived for the imensionless drag force F_D.

Appendix G. Algorithm Modifications for Transient

First, we implement an initial modulation onto the velocity field to allow the vortex shedding to emerge. To do so, we employed the following modulation

$$u_{ij}(t=0) = A_0\frac{e^{(i\delta\eta)^6} - 1}{e - 1}\sin(2\pi j\delta\xi)\ , \tag{G.1}$$

with perturbation strength $A_0 = 0.05$. This modulation introduces an asymmetry w.r.t. the branch cut, while it vanishes radially towards the far-field boundary $\partial\Omega$. This was motivated to avoid strong perturbations on the comparably large cells in the far field, potentially making the system stiffer. Second, we employ a preconditioner for solving the Poisson equation within the TT approach to enable faster convergence in terms of the number of sweeps in the variational scheme [25], and to avoid dealing with highly ill-conditioned LSE in the local variational updates.

This allows us to make use of the *fast matvec* routine outlined in [32], i.e., emulating the local matrix-vector multiplication by successive tensor contractions scaling with $O(\chi^3)$ instead of assembling the local matrix and naively applying it to a vector with scaling of $O(\chi^4)$.

Appendix H. Verification of the Finite Difference Reference

To validate the feasibility of the presented results, the in-house FD code is benchmarked against reference values from OpenFOAM [93] and the literature [75,94,95] for the case of flow around a cylinder. For a quantitative comparison, the drag coefficient C_D as a function of the Reynolds number over the range $Re \in [1, 200]$ is given in Fig. H.20.

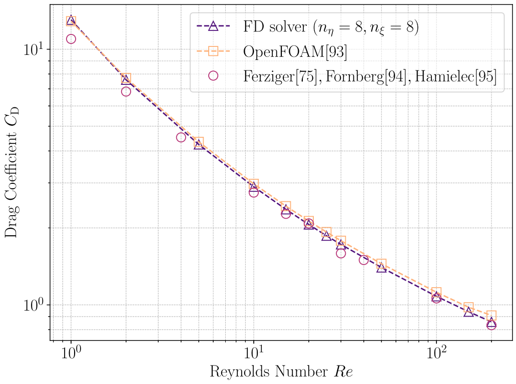

Fig. H.20. Drag coefficient as a function of the Reynolds number Re for the flow around a cylinder from Stokes to subcritical regimes. The benchmark study compares results from literature [75,94,95], OpenFOAM [93] and the developed FD solver.

The verification uses ○ marks for literature references, □ marks for the numerical results from OpenFOAM [93], and △ for results of the FD solver under validation. Overall, the results confirm good agreement, though minor discrepancies are apparent, especially at lower Re. These deviations can be attributed mainly to two factors: 1) discretization differences between the numerical models (finite approximation, grid resolution) and 2) differences in the computational setup (boundary layers, pressure correction approach, boundary conditions).

1) The application of different meshes between the models in Fig. H.20 is a source of discrepancy in the results. Furthermore, the references either approximate the vorticity-streamfunction formulation in [94,95], or apply finite volume discretizations [75,93], which inherently conserve fluxes, whereas the in-house FD solver uses central stencils introducing dispersion errors.

2) Another source of error lies in the computational model itself. As conventional CFD solvers can handle more complex flow phenomena, different types of boundaries and pressure correction approaches are implemented. In contrast, the developed FD code relies on Dirichlet-Neumann pressure boundaries to guarantee numerical stability. The differences in the boundary conditions are important as the wall treatment directly influences the computation of forces and reflects on the drag coefficient C_D. As a consequence, the corresponding literature symbols do not correctly align with the numerical results that include inertial contributions.

In conclusion, the FD solver delivers good accuracies and shows a constant offset to the given references, indicating a well-functioning of the model and motivating a reliable basis for benchmarking the TT solver under discussion.

References

[1] R.D. Moser, Chapter 1 - Numerical challenges in turbulence simulation, in: R.D. Moser (Ed.), Numerical Methods in Turbulence Simulation, Numerical Methods in Turbulence, Academic Press, 2023, pp. 1–43. https://doi.org/10.1016/B978-0-32-391144-3.00007-3

[2] J.P. Slotnick, A. Khodadoust, J. Alonso, D. Darmofal, W. Gropp, E. Lurie, D.J. Mavriplis, CFD vision 2030 study: a path to revolutionary computational aerosciences, Technical Report, NASA, 2014. https://ntrs.nasa.gov/citations/20140003093.

[3] A.S. Monin, A.M. Yaglom, J.L. Lumley, Statistical Fluid Mechanics: Mechanics of Turbulence, number Vol. 1 in Dover books on physics, Dover Publications, 2007.

[4] D. Wilcox, Turbulence Modeling for CFD (Third Edition), D C W Industries, 2006.

[5] S.B. Pope, Turbulent Flows, Cambridge University Press, 2000. https://doi.org/10.1017/CBO9780511840531

[6] K. Taira, S.L. Brunton, S.T.M. Dawson, C.W. Rowley, T. Colonius, B.J. McKeon, O.T. Schmidt, S. Gordeyev, V. Theofilis, L.S. Ukeiley, Modal analysis of fluid flows: an overview, AIAA J. 55 (12) (2017) 4013–4041. https://doi.org/10.2514/1.J056060

[7] S.L. Brunton, J.N. Kutz, Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control, Cambridge University Press, 2022. https://doi.org/10.1017/9781009089517

[8] R. Bellman, S. Dreyfus, Dynamic Programming, 33, Princeton University Press, 2010. http://www.jstor.org/stable/j.ctv1nxcw0f.

[9] P. Givi, A.J. Daley, D. Mavriplis, M. Malik, Quantum speedup for aeroscience and engineering, AIAA J. 58 (8) (2020) 3715–3727. https://doi.org/10.2514/1.J059183

[10] D. Jaksch, P. Givi, A.J. Daley, T. Rung, Variational quantum algorithms for computational fluid dynamics, AIAA J. 61 (5) (2023) 1885–1894. https://doi.org/10.2514/1.J062426

[11] S.S. Bharadwaj, K.R. Sreenivasan, Quantum computation of fluid dynamics, in: Indian Academy of Scieinces Conference Series, 3, 2020, pp. 77–96. https://doi.org/10.29195/iascs.03.01.0015

[12] C.A. Riofrío, J. Klepsch, J.R. Finžgar, F. Kiwit, L. Hölscher, M. Erdmann, L. Müller, C. Kumar, Y.A. Berrada, A. Luckow, Quantum Computing for Automotive Applications, 2024. 2409.14183

[13] C. Sanavio, R. Scatamacchia, C. de Falco, S. Succi, Three Carleman routes to the quantum simulation of classical fluids, Phys. Fluids 36 (5) (2024) 057143. https://doi.org/10.1063/5.0204955

[14] C. Sanavio, E. Mauri, S. Succi, Explicit quantum circuit for simulating the advection-diffusion-reaction dynamics, IEEE Trans. Quantum Eng. 6 (2025) 1–12. https://doi.org/10.1109/TQE.2025.3544839

[15] A.W. Harrow, A. Hassidim, S. Lloyd, Quantum algorithm for linear systems of equations, Phys. Rev. Lett. 103 (2009) 150502. https://doi.org/10.1103/PhysRevLett.103.150502

[16] A.M. Childs, R. Kothari, R.D. Somma, Quantum algorithm for systems of linear equations with exponentially improved dependence on precision, SIAM J. Comput. 46 (6) (2017) 1920–1950. https://doi.org/10.1137/16M1087072.

[17] B. Ljubomir, Quantum algorithm for the Navier-Stokes equations by using the streamfunction-vorticity formulation and the lattice Boltzmann method, Int. J. Quantum Inf. 20 (02) (2022) 2150039. https://doi.org/10.1142/S0219749921500398

[18] W. Itani, K.R. Sreenivasan, S. Succi, Quantum algorithm for lattice Boltzmann (QALB) simulation of incompressible fluids with a nonlinear collision term, Phys. Fluids 36 (1) (2024) 017112. https://doi.org/10.1063/5.0176569

[19] Z. Meng, Y. Yang, Quantum spin representation for the Navier-Stokes equation, Phys. Rev. Res. 6 (2024) 043130. https://doi.org/10.1103/PhysRevResearch.6.043130

[20] P. Over, S. Bengoechea, P. Brearley, S. Laizet, T. Rung, Quantum algorithm for the advection-diffusion equation by direct block encoding of the time-marching operator, Phys. Rev. A 112 (2025) L010401. https://doi.org/10.1103/d8hb-fv93

[21] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P.J. Love, A. Aspuru-Guzik, J.L. O'Brien, A variational eigenvalue solver on a photonic quantum processor, Nat. Commun. 5 (1) (2014). https://doi.org/10.1038/ncomms5213

[22] M. Lubasch, J. Joo, P. Moinier, M. Kiffner, D. Jaksch, Variational quantum algorithms for nonlinear problems, Phys. Rev. A 101 (2020) 010301. https://doi.org/10.1103/PhysRevA.101.010301

[23] P. Over, S. Bengoechea, T. Rung, F. Clerici, L. Scandurra, E. de Villiers, D. Jaksch, Boundary treatment for variational quantum simulations of partial differential equations on quantum computers, Comput. Fluids 288 (2025) 106508. https://doi.org/10.1016/j.compfluid.2024.106508

[24] S. Bengoechea, P. Over, D. Jaksch, T. Rung, Toward variational quantum algorithms for generalized linear and nonlinear transport phenomena, AIAA J. 0 (0) (0) 1–20. https://doi.org/10.2514/1.J065582

[25] U. Schollwöck, The density-matrix renormalization group in the age of matrix product states, Ann. Phys. 326 (1) (2011) 96–192. https://doi.org/10.1016/j.aop.2010.09.012

[26] A. Termanova, A. Melnikov, E. Mamenchikov, N. Belokonev, S. Dolgov, A. Berezutskii, R. Ellerbrock, C. Mansell, M.R. Perelshtein, Tensor quantum programming, New J. Phys. 26 (12) (2024) 123019. https://doi.org/10.1088/1367-2630/ad985b

[27] P. Siegl, G.S. Reese, T. Hashizume, N.-L. van Hülst, D. Jaksch, Tensor-Programmable Quantum Circuits for Solving Differential Equations, 2025. 2502.04425https://doi.org/10.48550/arXiv.2502.04425

[28] M.D. García, A. Márquez Romero, Survey on computational applications of tensor-network simulations, IEEE Access 12 (2024) 193212–193228. https://doi.org/10.1109/ACCESS.2024.3519676

[29] G. Rozza, G. Stabile, F. Ballarin, Advanced Reduced Order Methods and Applications in Computational Fluid Dynamics, Society for Industrial and Applied Mathematics, Philadelphia, PA, Philadelphia, PA, 2022. https://doi.org/10.1137/1.9781611977257

[30] J. Eisert, M. Cramer, M.B. Plenio, Colloquium: area laws for the entanglement entropy, Rev. Mod. Phys. 82 (2010) 277–306. https://doi.org/10.1103/RevModPhys.82.277

[31] B.N. Khoromskij, O(dlogN)-quantics approximation of N-d tensors in high-dimensional numerical modeling, Constructive Approximation 34 (2) (2011) 257–280. https://doi.org/10.1007/s00365-011-9131-1

[32] S.V. Dolgov, B.N. Khoromskij, I.V. Oseledets, Fast solution of parabolic problems in the tensor train/quantized tensor train format with initial application to the fokker–planck equation, SIAM J. Sci. Comput. 34 (6) (2012) A3016–A3038. https://doi.org/10.1137/120864210

[33] L. Richter, L. Sallandt, N. Nüsken, Solving high-dimensional parabolic PDEs using the tensor train format, in: M. Meila, T. Zhang (Eds.), Proceedings of the 38th International Conference on Machine Learning, 139 of *Proceedings of Machine Learning Research*, PMLR, 2021, pp. 8998–9009. https://proceedings.mlr.press/v139/richter21a.html.

[34] M. Lubasch, P. Moinier, D. Jaksch, Multigrid renormalization, J. Comput. Phys. 372 (2018) 587–602. https://doi.org/10.1016/j.jcp.2018.06.065

[35] L. Hölscher, P. Rao, L. Müller, J. Klepsch, A. Luckow, T. Stollenwerk, F.K. Wilhelm, Quantum-inspired fluid simulation of two-dimensional turbulence with GPU acceleration, Phys. Rev. Res. 7 (2025) 013112. https://doi.org/10.1103/PhysRevResearch.7.013112

[36] M. Kiffner, D. Jaksch, Tensor network reduced order models for wall-bounded flows, Phys. Rev. Fluids 8 (12) (2023) 124101. https://doi.org/10.1103/physrevfluids.8.124101

[37] E. Ye, N.F. Loureiro, Quantized tensor networks for solving the Vlasov-Maxwell equations, J. Plasma Phys. 90 (3) (2024) 805900301. https://doi.org/10.1017/S0022377824000503

[38] R.D. Peddinti, S. Pisoni, A. Marini, P. Lott, H. Argentieri, E. Tiunov, L. Aolita, Quantum-inspired framework for computational fluid dynamics, Commun. Phys. 7 (1) (2024) 135. https://doi.org/10.1038/s42005-024-01623-8

[39] A.A. Michailidis, C. Fenton, M. Kiffner, Tensor Train Multiplication, 2024. https://arxiv.org/abs/2410.19747. 2410.19747

[40] M.E. Danis, D. Truong, I. Boureima, O. Korobkin, K.Ø. Rasmussen, B.S. Alexandrov, Tensor-train WENO scheme for compressible flows, J. Comput. Phys. 529 (2025) 113891. https://doi.org/10.1016/j.jcp.2025.113891

[41] S. Pisoni, R.D. Peddinti, E. Tiunov, S.E. Guzman, L. Aolita, Compression, simulation, and synthesis of turbulent flows with tensor trains, 2025. https://arxiv.org/abs/2506.05477. 2506.05477

[42] K. Lively, V. Pagni, G. Camacho, A Quantum-Inspired Algorithm for Wave Simulation Using Tensor Networks, 2025. https://arxiv.org/abs/2504.11181. 2504.11181

[43] L. Arenstein, M. Mikkelsen, M. Kastoryano, Fast and Flexible Quantum-Inspired Differential Equation Solvers with Data Integration, 2025. https://arxiv.org/abs/2505.17046. 2505.17046

[44] N. Gourianov, P. Givi, D. Jaksch, S.B. Pope, Tensor networks enable the calculation of turbulence probability distributions, Sci. Adv. 11 (5) (2025) eads5990. https://doi.org/10.1126/sciadv.ads5990

[45] I. Demirdžić, S. Muzaferija, Numerical method for coupled fluid flow, heat transfer and stress analysis using unstructured moving meshes with cells of arbitrary topology, Comput. Methods Appl. Mech. Eng. 125 (1) (1995) 235–255. https://doi.org/10.1016/0045-7825(95)00800-G

[46] H.G. Weller, G. Tabor, H. Jasak, C. Fureby, A tensorial approach to computational continuum mechanics using object-oriented techniques, Comput. Phys. 12 (6) (1998) 620–631. https://doi.org/10.1063/1.168744

[47] M.-C. Lai, C.S. Peskin, An immersed boundary method with formal second-Order accuracy and reduced numerical viscosity, J. Comput. Phys. 160 (2) (2000) 705–719. https://doi.org/10.1006/jcph.2000.6483

[48] C.S. Peskin, The immersed boundary method, Acta Numer. 11 (2002) 479-517. https://doi.org/10.1017/S0962492902000077

[49] K. Taira, T. Colonius, The immersed boundary method: a projection approach, J. Comput. Phys. 225 (2) (2007) 2118–2137. https://doi.org/10.1016/j.jcp.2007.03.005

[50] R. Verzicco, Immersed boundary methods: historical perspective and future outlook, Annu. Rev. Fluid Mech. 55 (2023) 129–155. https://doi.org/10.1146/annurev-fluid-120720-022129

[51] S. Völkner, J. Brunswig, T. Rung, Analysis of non-conservative interpolation techniques in overset grid finite-volume methods, Comput. Fluids 148 (2017) 39–55. https://doi.org/10.1016/j.compfluid.2017.02.010

[52] A. Sharma, S. Ananthan, J. Sitaraman, S. Thomas, M.A. Sprague, Overset meshes for incompressible flows: on preserving accuracy of underlying discretizations, J. Comput. Phys. 428 (2021) 109987. https://doi.org/10.1016/j.jcp.2020.109987

[53] E. Kornev, S. Dolgov, K. Pinto, M. Pflitsch, M. Perelshtein, A. Melnikov, Numerical solution of the incompressible Navier-Stokes equations for chemical mixers via quantum-inspired Tensor Train Finite Element Method, 2023. 2305.10784https://doi.org/10.48550/ARXIV.2305.10784

[54] A.J. Chorin, Numerical solution of the Navier-Stokes equations, Math. Comput. 22 (104) (1968) 745–762. http://www.jstor.org/stable/2004575.

[55] J. Kim, P. Moin, Application of a fractional-step method to incompressible Navier-Stokes equations, J. Comput. Phys. 59 (2) (1985) 308–323. https://doi.org/10.1016/0021-9991(85)90148-2

[56] S. Patankar, Numerical heat transfer and fluid flow, CRC press, 2018. https://doi.org/10.1201/9781482234213

[57] J. Bezanson, A. Edelman, S. Karpinski, V.B. Shah, Julia: a fresh approach to numerical computing, SIAM Rev. 59 (1) (2017) 65–98. https://doi.org/10.1137/141000671.

[58] I.V. Oseledets, Tensor-train decomposition, SIAM Journal on Scientific Computing 33 (5) (2011) 2295–2317. https://doi.org/10.1137/090752286

[59] I.V. Oseledets, E.E. Tyrtyshnikov, Breaking the curse of dimensionality, or how to use SVD in many dimensions, SIAM J. Sci. Comput. 31 (5) (2009) 3744–3759. https://doi.org/10.1137/090748330

[60] B.N. Khoromskij, Tensor Numerical Methods for High-dimensional PDEs: Basic Theory and Initial Applications, 2014. https://doi.org/10.48550/ARXIV.1408.4053

[61] R. Penrose, et al., Applications of negative dimensional tensors, Comb. Math. Appli. 1 (221–244) (1971) 3.

[62] I. Oseledets, E. Tyrtyshnikov, TT-cross approximation for multidimensional arrays, Linear Algebra Appl. 432 (1) (2010) 70–88. https://doi.org/10.1016/j.laa.2009.07.024

[63] M.K. Ritter, Y. Núñez Fernández, M. Wallerberger, J. von Delft, H. Shinaoka, X. Waintal, Quantics tensor cross interpolation for high-Resolution parsimonious representations of multivariate functions, Phys. Rev. Lett. 132 (2024) 056501. https://doi.org/10.1103/PhysRevLett.132.056501

[64] I.V. Oseledets, Constructive representation of functions in low-rank tensor formats, Constructive Approximation 37 (1) (2012) 1–18. https://doi.org/10.1007/s00365-012-9175-x

[65] R. Orus, A practical introduction to tensor networks: matrix product states and projected entangled pair states, Ann. Phys. 349 (2013). https://doi.org/10.1016/j.aop.2014.06.013

[66] V.A. Kazeev, B.N. Khoromskij, Low-rank explicit QTT representation of the laplace operator and its inverse, SIAM J. Matrix Anal. Appl. 33 (3) (2012) 742–758. https://doi.org/10.1137/100820479

[67] J.D. Biamonte, S.R. Clark, D. Jaksch, Categorical tensor network states, AIP Adv. 1 (4) (2011). https://doi.org/10.1063/1.3672009

[68] S.V. Dolgov, TT-GMRES: solution to a linear system in the structured tensor format, Russ. J. Numer. Anal. Math. Model. 28 (2) (2013) 149–172. https://doi.org/10.1515/rnam-2013-0009

[69] S. Holtz, T. Rohwedder, R. Schneider, The alternating linear scheme for tensor optimization in the tensor train format, SIAM J. Sci. Comput. 34 (2) (2012) A683–A713. https://doi.org/10.1137/100818893

[70] I.V. Oseledets, S.V. Dolgov, Solution of linear systems and matrix inversion in the TT-Format, SIAM J. Sci. Comput. 34 (5) (2012) A2718–A2739. https://doi.org/10.1137/110833142

[71] S.V. Dolgov, D.V. Savostyanov, Alternating minimal energy methods for linear systems in higher dimensions, SIAM J. Sci. Comput. 36 (5) (2014) A2248–A2271. https://doi.org/10.1137/140953289

[72] C. Grossmann, H.-G. Roos, M. Stynes, Numerical Treatment of Partial Differential Equations, Springer Berlin Heidelberg, 2007. https://doi.org/10.1007/978-3-540-71584-9

[73] S. Paeckel, T. Köhler, A. Swoboda, S.R. Manmana, U. Schollwöck, C. Hubig, Time-evolution methods for matrix-product states, Ann. Phys. 411 (2019) 167998. https://doi.org/10.1016/j.aop.2019.167998

[74] E.M. Stoudenmire, S.R. White, Minimally entangled typical thermal state algorithms, New J. Phys. 12 (5) (2010) 055026. https://doi.org/10.1088/1367-2630/12/5/055026

[75] J.H. Ferziger, M. Perić, R.L. Street, Computational Methods for Fluid Dynamics, Springer International Publishing, 2020. https://doi.org/10.1007/978-3-319-99693-6

[76] J.F. Thompson, F.C. Thames, C.W. Mastin, Automatic numerical generation of body-fitted curvilinear coordinate system for field containing any number of arbitrary two-dimensional bodies, J. Comput. Phys. 15 (3) (1974) 299–319. https://doi.org/10.1016/0021-9991(74)90114-4

[77] J.F. Thompson, General curvilinear coordinate systems, Appl. Math. Comput. 10–11 (1982) 1–30. https://doi.org/10.1016/0096-3003(82)90185-0

[78] W.J. Gordon, C.A. Hall, Construction of curvilinear co-ordinate systems and applications to mesh generation, Int. J. Numer. Methods Eng. 7 (4) (1973) 461–477. https://doi.org/10.1002/nme.1620070405

[79] W.J. Gordon, C.A. Hall, Transfinite element methods: blending-function interpolation over arbitrary curved element domains, Numer. Math. 21 (2) (1973) 109–129. https://doi.org/10.1007/BF01436298

[80] E. Ye, N.F.G. Loureiro, Quantum-inspired method for solving the Vlasov-Poisson equations, Phys. Rev. E 106 (3) (2022) 035208. https://doi.org/10.1103/physreve.106.035208

[81] X. Wang, T. Hashizume, P. Siegl, D. Jaksch, Iterative tensor network transforms for element-wise evaluation of elementary and filtering functions, 2026. Publication in progress.

[82] U.M. Ascher, S.J. Ruuth, R.J. Spiteri, Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations, Appl. Numer. Math. 25 (2) (1997) 151–167. Special Issue on Time Integration. https://doi.org/10.1016/S0168-9274(97)00056-1

[83] B. Koc, S. Rubino, T. Chacón Rebollo, T. Iliescu, Residual-based data-driven variational multiscale reduced order models for parameter-dependent problems, Comput. Appl. Math. 44 (2025). https://doi.org/10.1007/s40314-025-03273-0

[84] S. Mittal, B. Kumar, Flow past a rotating cylinder, J. Fluid Mech. 476 (2003) 303-334. https://doi.org/10.1017/S0022112002002938

[85] S. Kang, H. Choi, S. Lee, Laminar flow past a rotating circular cylinder, Phys. Fluids 11 (11) (1999) 3312-3321. https://doi.org/10.1063/1.870190

[86] M. Visbal, D. Rizzetta, Large-eddy simulation on curvilinear grids using compact differencing and filtering schemes, J. Fluids Eng. 124 (4) (2002) 836–847. https://doi.org/10.1115/1.1517564

[87] S. Deck, P.-E. Weiss, N. Renard, A rapid and low noise switch from RANS to WMLES on curvilinear grids with compressible flow solvers, J. Comput. Phys. 363 (2018) 231–255. https://doi.org/10.1016/j.jcp.2018.02.028

[88] N. Gourianov, M. Lubasch, S. Dolgov, Q.Y. van den Berg, H. Babaee, P. Givi, M. Kiffner, D. Jaksch, A quantum-inspired approach to exploit turbulence structures, Nat. Comput. Sci. 2 (1) (2022) 30–37. https://doi.org/10.1038/s43588-021-00181-1

[89] N. Gourianov, Exploiting the Structure of Turbulence with Tensor Networks, Phd thesis, University of Oxford, Oxford, UK, 2022. https://ora.ox.ac.uk/objects/uuid:9e3f4786-ad68-4913-9a0d-e9b1e108128f.

[90] P. Secular, N. Gourianov, M. Lubasch, S. Dolgov, S.R. Clark, D. Jaksch, Parallel time-dependent variational principle algorithm for matrix product states, Phys. Rev. B 101 (2020) 235123. https://doi.org/10.1103/PhysRevB.101.235123

[91] L. Prandtl, Applications of modern hydrodynamics to aeronautics, Technical Report, NASA, 1923. https://ntrs.nasa.gov/citations/19930091180.

[92] N.-L. van Hülst, P. Siegl, P. Over, S. Bengoechea, T. Hashizume, M.G. Cecile, T. Rung, D. Jaksch, A software example for “Quantum-Inspired Fractional Step Method in Curvilinear Coordinates”, 2025. https://doi.org/10.25592/uhhfdm.17679.

[93] E. Ltd, HELYX, 2025. https://engys.com/helyx/.

[94] B. Fornberg, Steady viscous flow past a circular cylinder up to reynolds number 600, J. Comput. Phys. 61 (2) (1985) 297–320. https://doi.org/10.1016/0021-9991(85)90089-0

[95] A.E. Hamielec, J.D. Raal, Numerical studies of viscous flow around circular cylinders, Phys. Fluids 12 (1) (1969) 11–17. https://doi.org/10.1063/1.1692253