

Accepted Manuscript

A fast lattice Green's function method for solving viscous incompressible flows on unbounded domains

Sebastian Liska, Tim Colonius

PII: S0021-9991(16)30064-X
DOI: <http://dx.doi.org/10.1016/j.jcp.2016.04.023>
Reference: YJCPH 6551

To appear in: *Journal of Computational Physics*

Received date: 26 August 2015
Revised date: 7 April 2016
Accepted date: 10 April 2016

Please cite this article in press as: S. Liska, T. Colonius, A fast lattice Green's function method for solving viscous incompressible flows on unbounded domains, *J. Comput. Phys.* (2016), <http://dx.doi.org/10.1016/j.jcp.2016.04.023>

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.



A fast lattice Green's function method for solving viscous incompressible flows on unbounded domains

Sebastian Liska*, Tim Colonius*

Division of Engineering and Applied Science, California Institute of Technology, Pasadena, CA 91125, USA

Abstract

A computationally efficient method for solving three-dimensional, viscous, incompressible flows on unbounded domains is presented. The method formally discretizes the incompressible Navier-Stokes equations on an unbounded staggered Cartesian grid. Operations are limited to a finite computational domain through a lattice Green's function technique. This technique obtains solutions to inhomogeneous *difference* equations through the discrete convolution of source terms with the fundamental solutions of the *discrete* operators. The differential algebraic equations describing the temporal evolution of the discrete momentum equation and incompressibility constraint are numerically solved by combining an integrating factor technique for the viscous term and a half-explicit Runge-Kutta scheme for the convective term. A projection method that exploits the mimetic and commutativity properties of the discrete operators is used to efficiently solve the system of equations that arises in each stage of the time integration scheme. Linear complexity, fast computation rates, and parallel scalability are achieved using recently developed fast multipole methods for difference equations. The accuracy and physical fidelity of solutions is verified through numerical simulations of vortex rings.

Keywords: Incompressible viscous flow, Unbounded domain, Lattice Green's function, Projection method, Integrating factor, Half-explicit Runge-Kutta, Elliptic solver

1. Introduction

Numerical simulations of viscous, incompressible flows on unbounded fluid domains require numerical techniques that can accurately approximate unbounded computational domains using only a finite number of operations. Spatial truncation and artificial boundary conditions have been developed for this purpose but they can adversely affect the accuracy of the solution and even change the dynamics of the flow [1–4]. Furthermore, minimizing the error due to artificial boundaries by employing large computational domains increases the number of computational elements and often requires the use of solvers that are less efficient than those used on regular grids (e.g. FFT techniques, multigrid, etc.).

Recently, fast multipole methods (FMMs) for solving constant coefficient elliptic *difference* equations on unbounded regular grids have been developed for 2D [5, 6] and 3D [7] problems.

*Corresponding author

Email addresses: sliska@caltech.edu (Sebastian Liska), colonius@caltech.edu (Tim Colonius)

These methods obtain solutions to inhomogeneous *difference* equations by using fast summation techniques to evaluate the discrete convolution of source terms with the fundamental solutions of the *discrete* operators. The fundamental solutions of discrete operators on unbounded regular grids, or lattices, are also referred to as lattice Green's functions (LGFs).

Similar to particle and vortex methods, e.g. [8–20] and references therein, the LGF techniques discussed in [5–7] have efficient nodal distributions and automatically enforce free-space boundary conditions. As a result, needlessly large computational domains and artificial boundary conditions can be avoided when solving flows on unbounded regular grids by using LGF techniques to compute the action of solution operators. A significant advantage of recently developed particle and vortex methods is their ability to efficiently solve large scale problems relevant to 3D incompressible flows using fast, parallel methods based on techniques such as tree-codes, FMMs, dynamic error estimators, hybrid Eulerian-Lagrangian formulations, hierarchical grids, FFT methods, and domain decomposition techniques [11–13, 15, 17–20]. It is demonstrated in [7] that LGF FMMs can achieve computational rates and parallel scaling for 3D discrete (7-pt Laplacian) Poisson problems comparable to existing fast 3D Poisson solvers.

The present formulation numerically solves the incompressible Navier-Stokes equations expressed in the non-dimensional form given by

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \frac{1}{\text{Re}} \nabla^2 \mathbf{u}, \quad (1a)$$

$$\nabla \cdot \mathbf{u} = 0, \quad (1b)$$

where \mathbf{u} , p , and Re correspond to the velocity, the pressure, and the Reynolds number, respectively. The equations are defined on an unbounded domain in all directions, and are subject to the boundary conditions

$$\mathbf{u}(\mathbf{x}, t) \rightarrow \mathbf{u}_\infty(t) \text{ as } |\mathbf{x}| \rightarrow \infty, \quad (2)$$

where \mathbf{u}_∞ is a known time-dependent function. We limit our attention to flows in which the vorticity, $\boldsymbol{\omega} = \nabla \times \mathbf{u}$, decay exponentially fast as $|\mathbf{x}| \rightarrow \infty$.

The present formulation is simplified by considering the evolution of the velocity perturbation, $\mathbf{u}'(\mathbf{x}, t) = \mathbf{u}(\mathbf{x}, t) - \mathbf{u}_\infty(t)$, and pressure perturbation, $p'(\mathbf{x}, t) = p(\mathbf{x}, t) - p_\infty(\mathbf{x}, t)$. The freestream pressure, p_∞ , is given by

$$p_\infty(\mathbf{x}, t) = \frac{d\mathbf{u}_\infty}{dt} \cdot \mathbf{x}, \quad (3)$$

where we have taken the arbitrary time-dependent constant to be zero. Subtracting the uniform freestream equations from Eq. (1) yields

$$\frac{\partial \mathbf{u}'}{\partial t} + (\mathbf{u}' + \mathbf{u}_\infty) \cdot \nabla \mathbf{u}' = -\nabla p' + \frac{1}{\text{Re}} \nabla^2 \mathbf{u}', \quad \nabla \cdot \mathbf{u}' = 0, \quad (4)$$

subject to the boundary conditions $\mathbf{u}'(\mathbf{x}, t) \rightarrow 0$ as $|\mathbf{x}| \rightarrow \infty$. The boundary conditions on \mathbf{u}' and the irrotational nature of the flow at large distances imply that p' is subject to the compatibility condition¹

$$p'(\mathbf{x}, t) \rightarrow 0 \text{ as } |\mathbf{x}| \rightarrow \infty. \quad (5)$$

¹In the absence of sources and sinks, the velocity of an irrotational flow subject to zero boundary conditions at infinity is given by $\mathbf{v} = \nabla \phi$, where the leading order term of ϕ is $-\mathbf{M} \cdot \mathbf{x}/r^3$ [21]. Consequently, $p = -\left(\frac{\partial \phi}{\partial t} + \frac{1}{2}|\nabla \phi|^2\right) \rightarrow 0$ as $r \rightarrow \infty$, where we have taken the arbitrary time-dependent constant to be zero.

The remainder of the paper is organized as follows. In Section 2, we describe the spatial discretization of the governing equations on formally unbounded staggered Cartesian grids and discuss LGF techniques that can be used to obtain fast solutions to the associated discrete elliptic problems. Additionally, we present an integrating factor technique that facilitates the implementation of efficient, robust time integration schemes. In Section 3, the system of differential algebraic equations (DAEs) resulting from the spatial discretization and integrating factor techniques is numerically solved using a half-explicit Runge-Kutta method. We show that the linear systems of equations that arise at each stage of the time integration scheme can be efficiently solved, without splitting errors or additional stability constraints, by a fast projection method based on LGF techniques and the properties of the discrete operators. In Section 4, we demonstrate that an adaptive block-structured grid padded with appropriately sized buffer regions can be used to efficiently compute numerical solutions to a prescribed tolerance. In Section 5, we summarize the algorithm and discuss a few practical considerations including computational costs and performance optimization. Finally, in Section 6, we perform numerical experiments on vortex rings to verify the present formulation.

2. Spatial discretization

2.1. Unbounded staggered Cartesian grids

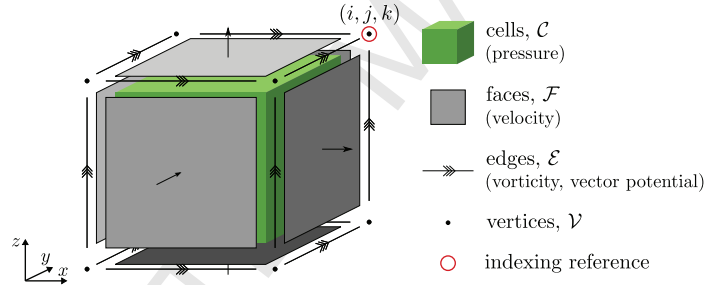


Figure 1: Unit cell of the staggered Cartesian grid. The vertex enclosed by the circle corresponds to the (i, j, k) vertex. The (i, j, k) cell, faces, and edges correspond to the depicted elements intersecting the (i, j, k) vertex. There are three faces and edges per vertex. The superscript “ (q) ” is used to denote faces (edges) normal (parallel) to x_q axis.

In this section we describe the discretization of Eq. (4) on a formally unbounded staggered Cartesian grid. Figure 1 depicts our staggered grid, which consists of cells (\mathcal{C}) and vertices (\mathcal{V}) that house scalar quantities, and faces (\mathcal{F}) and edges (\mathcal{E}) that house vector quantities. The notation \mathbb{R}^Q denotes the set of real-valued grid functions with values defined on $Q \in \{\mathcal{C}, \mathcal{F}, \mathcal{E}, \mathcal{V}\}$. The value of a grid function q evaluated at $\mathbf{n} = (i, j, k) \in \mathbb{Z}^3$ is given by $q(\mathbf{n})$ and $q_{i,j,k}$. For the case of a vector-valued grid function q , i.e. $q \in \mathbb{R}^{\mathcal{F}}$ or $q \in \mathbb{R}^{\mathcal{E}}$, $q^{(k)}(\mathbf{n})$ denotes the component of $q(\mathbf{n})$ in the k -th direction.

The spatial discretization of Eq. (4) is performed using the techniques of Nicolaides and Wu [22], and Zhang et al. [23]. The resulting discrete operators are similar or equivalent to those obtained from standard second-order finite-volume or finite-difference schemes, e.g. [24].

Yet we refer to the more general techniques of [22] and [23] since their discussions emphasize many of the algebraic properties of the discrete operators used by the present formulation. For convenience, point-operator representations of the discrete operators are included in [Appendix A](#).

The semi-discrete system of equations obtained from the spatial discretization of Eq. (4) is

$$\frac{d\mathbf{u}}{dt} + \mathbf{N}(\mathbf{u} + \mathbf{u}_\infty) = -\mathbf{G}\mathbf{p} + \frac{1}{\text{Re}}\mathbf{L}_\mathcal{F}\mathbf{u}, \quad \bar{\mathbf{D}}\mathbf{u} = 0, \quad (6)$$

where $\mathbf{u} \in \mathbb{R}^\mathcal{F} \times \mathbb{R}$ and $\mathbf{p} \in \mathbb{R}^C \times \mathbb{R}$ denote the time-dependent grid functions associated with the discrete velocity and pressure perturbation fields, respectively.² The time-dependent grid function $\mathbf{u}_\infty \in \mathbb{R}^\mathcal{F} \times \mathbb{R}$ is constant in space with values given by $u_\infty(\mathbf{n}, t) = \mathbf{u}_\infty(t)$. Discrete operators $\mathbf{G} : \mathbb{R}^C \mapsto \mathbb{R}^\mathcal{F}$, $\bar{\mathbf{D}} : \mathbb{R}^\mathcal{F} \mapsto \mathbb{R}^C$, and $\mathbf{L}_\mathcal{F} : \mathbb{R}^\mathcal{F} \mapsto \mathbb{R}^\mathcal{F}$ correspond to the discretizations of the gradient, divergence, and vector Laplacian operators, respectively. Finally, $\mathbf{N} : \mathbb{R}^\mathcal{F} \mapsto \mathbb{R}^\mathcal{F}$ denotes the discrete nonlinear operator approximating the convective term, i.e. $\mathbf{N}(\mathbf{u} + \mathbf{u}_\infty) \approx (\mathbf{u}' + \mathbf{u}_\infty) \cdot \nabla (\mathbf{u}' + \mathbf{u}_\infty) = (\mathbf{u}' + \mathbf{u}_\infty) \cdot \nabla \mathbf{u}'$.³

In addition to the aforementioned discrete operators, the subsequent discussion makes use of the discrete gradient operator $\bar{\mathbf{G}} : \mathbb{R}^\mathcal{V} \mapsto \mathbb{R}^\mathcal{E}$, the discrete curl operators $\mathbf{C} : \mathbb{R}^\mathcal{F} \mapsto \mathbb{R}^\mathcal{E}$ and $\bar{\mathbf{C}} : \mathbb{R}^\mathcal{E} \mapsto \mathbb{R}^\mathcal{F}$, and the discrete Laplacian operators $\mathbf{L}_Q : \mathbb{R}^Q \mapsto \mathbb{R}^Q$, where $Q \in \{C, \mathcal{E}, \mathcal{V}\}$. A summary of all the discrete vector operators and their definitions is also provided in [Appendix A](#).

The choice of discretization technique yields a numerical scheme with the following properties:

- *Second-order accuracy*: all discrete operators are second-order accurate in space.
- *Conservation properties*: using appropriate discretizations of the nonlinear convective term leads to a scheme that conserves momentum, kinetic energy, and circulation in the absence of time-differencing errors and viscosity [23, 25, 26]. The benefits of discrete conservation properties related to numerical stability and physical fidelity are discussed in the review by Perot [27] and references therein.
- *Mimetic properties*: discrete operators and their corresponding vector calculus operators satisfy similar symmetry and orthogonality properties in addition to similar integration by parts formulas [22, 23, 28, 29]. Specific properties pertinent to the discussion of the present method are:

$$\bar{\mathbf{D}} = -\mathbf{G}^\dagger, \quad \bar{\mathbf{C}} = \mathbf{C}^\dagger, \quad \bar{\mathbf{G}} = -\mathbf{D}^\dagger, \quad (7a)$$

$$\text{Im}(\mathbf{G}) = \text{Null}(\mathbf{C}), \quad \text{Im}(\mathbf{C}) = \text{Null}(\mathbf{D}), \quad (7b)$$

$$\mathbf{L}_C = -\mathbf{G}^\dagger \mathbf{G}, \quad \mathbf{L}_\mathcal{F} = -\mathbf{G}\mathbf{G}^\dagger - \mathbf{C}^\dagger \mathbf{C}, \quad \mathbf{L}_\mathcal{E} = -\mathbf{D}^\dagger \mathbf{D} - \mathbf{C}\mathbf{C}^\dagger, \quad \mathbf{L}_\mathcal{V} = -\mathbf{D}\mathbf{D}^\dagger. \quad (7c)$$

Many of the mimetic properties of discrete operators are closely related to the conservation properties [22, 23].

²In order to avoid a cumbersome notation, the prime symbols, $'$, are omitted from variables denoting grid functions associated with the perturbations of the discrete velocity and pressure fields.

³No particular form (e.g. convection, rotational, divergence, skew-symmetric) or discretization scheme for the convection term is assumed by Eq. (6).

- *Commutativity properties*: on unbounded staggered grids, discrete Laplacians and integrating factors (to be introduced in Section 2.3) are able to commute with other operators in the sense $AT_X = T_Y A$, where $A : \mathbb{R}^X \mapsto \mathbb{R}^Y$ is any of the previously mentioned linear operators, and T_X (T_Y) is either the discrete Laplacian or integrating factor mapping \mathbb{R}^X to \mathbb{R}^X (\mathbb{R}^Y to \mathbb{R}^Y). Similar commutativity properties result in discretizations of periodic domains using uniform staggered grids.

In subsequent sections we discuss how the mimetic and commutativity properties facilitate the construction of fast, stable methods for numerically solving Eq. (6).

It is convenient to define

$$\mathbf{d} = \mathbf{p} + \frac{1}{2} \mathbf{P}(\mathbf{u} + \mathbf{u}_\infty, \mathbf{u} + \mathbf{u}_\infty), \quad (8)$$

where $\mathbf{P} : \mathbb{R}^{\mathcal{F}} \times \mathbb{R}^{\mathcal{F}} \mapsto \mathbb{R}^C$ is an arbitrary discrete approximation of the vector dot-product, i.e. $\mathbf{P}(\mathbf{u}, \mathbf{v}) \approx \mathbf{u} \cdot \mathbf{v}$. The time-dependent grid function $\mathbf{d} \in \mathbb{R}^C \times \mathbb{R}$ can be regarded as a discrete approximation of the total pressure perturbation, i.e. $\mathbf{d} \approx p' + \frac{1}{2} |\mathbf{u}' + \mathbf{u}_\infty|^2$. Using Eq. (8), we express Eq. (6) as

$$\frac{d\mathbf{u}}{dt} + \tilde{\mathbf{N}}(\mathbf{u} + \mathbf{u}_\infty) = -\mathbf{G}\mathbf{d} + \frac{1}{\text{Re}} \mathbf{L}_{\mathcal{F}} \mathbf{u}, \quad \mathbf{G}^\dagger \mathbf{u} = 0, \quad (9)$$

where $\tilde{\mathbf{N}}(\mathbf{v}) = \mathbf{N}(\mathbf{v}) - \frac{1}{2} \mathbf{G}\mathbf{P}(\mathbf{v}, \mathbf{v})$. Consequently, $\tilde{\mathbf{N}}(\mathbf{u} + \mathbf{u}_\infty)$ is a discrete approximation of $\boldsymbol{\omega} \times (\mathbf{u} + \mathbf{u}_\infty)$.⁴ As will be demonstrated in Section 4, an advantage of using $\tilde{\mathbf{N}}(\mathbf{u} + \mathbf{u}_\infty)$ instead of $\mathbf{N}(\mathbf{u} + \mathbf{u}_\infty)$ is that the former typically has a smaller support than that of the latter, which in turn reduces the number of operations and storage required to numerically solve the flow. We emphasize that Eq. (9) is equivalent to Eq. (6), and no additional discretization errors have been introduced.

2.2. Lattice Green's function techniques

The procedure for solving difference equations on unbounded regular grids using LGFs is analogous to the procedure for solving inhomogeneous PDEs on unbounded domains using the fundamental solution of continuum operators. As a representative example, we consider the (continuum) scalar Poisson equation

$$[\Delta u](\mathbf{x}) = f(\mathbf{x}), \quad \text{supp}(f) \subseteq \Omega, \quad (10)$$

where $\mathbf{x} \in \mathbb{R}$ and Ω is a bounded domain in \mathbb{R}^3 . The solution to Eq. (10) is given by

$$u(\mathbf{x}) = [G * f](\mathbf{x}) = \int_{\Omega} G(\mathbf{x} - \mathbf{y}) f(\mathbf{y}) d\mathbf{y}, \quad (11)$$

where $G(\mathbf{x}) = -1/(4\pi|\mathbf{x}|)$ is the fundamental solution of the Laplace operator. Similarly, we consider the discrete scalar Poisson equation

$$[\mathbf{L}_Q \mathbf{u}](\mathbf{n}) = \mathbf{f}(\mathbf{n}), \quad \text{supp}(\mathbf{f}) \subseteq D, \quad (12)$$

⁴The discretization of Eq. (4) naturally assumes the form given by Eq. (9) if the convection term is discretized in its rotational form, $(\nabla \times \mathbf{v}) \times \mathbf{v} + \frac{1}{2} \nabla v^2$, with the gradient term approximated by $\frac{1}{2} \mathbf{G}\mathbf{P}(\mathbf{v}, \mathbf{v})$.

where $u, f \in \mathbb{R}^Q$, D is a bounded region in \mathbb{Z}^3 , and $Q \in \{C, V\}$. The solution to Eq. (12) is given by

$$u(\mathbf{n}) = [G_L * f](\mathbf{n}) = \sum_{\mathbf{m} \in D} G_L(\mathbf{n} - \mathbf{m})f(\mathbf{m}) \quad (13)$$

where $G_L : \mathbb{Z}^3 \mapsto \mathbb{R}$ is the fundamental solution, or LGF, of the discrete scalar Laplacian [6, 7]. Subsequently, we refer to the grid functions f and u as the source field and the induced field, respectively.

It is evident from the definitions of $L_{\mathcal{F}}$ and $L_{\mathcal{E}}$ that each component of a discrete vector Poisson problem corresponds to a discrete scalar Poisson problem. As a result, the q -th component of solutions to Eq. (12) for $Q \in \{\mathcal{F}, \mathcal{E}\}$ are given by Eq. (13) with $u \rightarrow u^{(q)}$ and $f \rightarrow f^{(q)}$. Procedures for obtaining expressions for $G_L(\mathbf{n})$ are discussed in [30–33]. For convenience, expressions for $G_L(\mathbf{n})$ are provided in Appendix B.

Fast numerical methods for evaluating discrete convolutions involving LGFs have recently been proposed in 2D by Gillman and Martinsson [6] and in 3D by Liska and Colonius [7]. Here, the 3D lattice Green's function fast multipole method (LGF-FMM) of [7] is used to evaluate discrete convolutions involving G_L . The LGF-FMM is a kernel-independent interpolation-based FMM specifically designed for solving difference equations on unbounded Cartesian grids. In addition to its asymptotic linear algorithmic complexity, it has been shown that the LGF-FMM achieves high computation rates and good parallel scaling for the case of G_L [7].

As final remark, the LGF-FMM is a direct solver that computes solutions to a prescribed tolerance ϵ , $\|y_{\text{true}} - y\|_{\infty} / \|y_{\text{true}}\|_{\infty} \leq \epsilon$, where y is the numerical solution and y_{true} is the exact solution to the system of *difference* equations. In order to obtain accurate error bounds for the LGF-FMM it is necessary to profile the method once for each kernel and scheme used. Error estimates for the discrete 7-pt Laplace kernel and different schemes are provided in [7]. In the present formulation, all instances of E_Q and L_Q^{-1} are computed using values of ϵ that are less than or equal to prescribed value of ϵ_{FMM} .

2.3. Integrating factor techniques

In this section we describe an integrating factor technique for integrating the stiff viscous term of Eq. (9) analytically. Analytical integration has the advantage of neither introducing discretization errors nor imposing stability constraints on the time marching scheme. Integrating factor techniques for the viscous term are widely used in Fourier pseudo-spectral methods. These methods typically compute the action of the integrating factor in Fourier-space. In contrast, the present method computes the action of the integrating factor in real-space, since the Fourier series of an arbitrary grid function on an unbounded domain is not computationally practical.

We consider integrating factors defined as the solution operators of the discrete diffusion equation of the form

$$\frac{dh}{dt} = \kappa L_Q h, \quad h(\mathbf{n}, t) \rightarrow h_{\infty}(t) \text{ as } |\mathbf{n}| \rightarrow \infty, \quad (14)$$

where $\kappa \in \mathbb{R}_{\geq 0}$ and $h \in \mathbb{R}^Q$. As discussed in Appendix A, the discrete Laplace operator L_Q is diagonalized by the Fourier series operator \mathcal{F}_Q ,

$$(\Delta x)^2 L_Q = \mathcal{F}_Q^{-1} \sigma_Q^{\perp} \mathcal{F}_Q, \quad (15)$$

where $\sigma_Q^{\perp}(\xi)$ for $\xi \in (\pi, \pi)^3$ is the spectrum of $(\Delta x)^2 L_Q$. Next, we define the exponential of the L_Q as

$$E_Q(\alpha) = \mathcal{F}_Q^{-1} \exp(\alpha \sigma_Q^{\perp}) \mathcal{F}_Q, \quad (16)$$

where $\alpha = \kappa(t - \tau)/(\Delta x)^2$. An immediate consequence of Eq. (16) is that

$$\frac{d}{d\alpha} E_Q(\alpha) = \mathcal{F}_Q^{-1} \sigma_Q^L \exp(\alpha \sigma_Q^L) \mathcal{F}_Q^{-1} = L_Q E_Q(\alpha) = E_Q(\alpha) L_Q, \quad (17)$$

which implies that the solution to Eq. (14) is given by

$$h(\mathbf{n}, t) = \left[E_Q \left(\frac{\kappa(t - \tau)}{(\Delta x)^2} \right) h_\tau \right](\mathbf{n}, t), \quad t \geq \tau, \quad \forall \mathbf{n} \in \mathbb{Z}^3, \quad (18)$$

where $h(\mathbf{n}, \tau) = h_\tau(\mathbf{n})$.

We now consider using $E_Q(\alpha)$ as an integrating factor for Eq. (9). Operating from the left on the semi-discrete momentum equation of Eq. (9) with $E_{\mathcal{F}} \left(\frac{t-\tau}{(\Delta x)^2 \text{Re}} \right)$ and introducing the transformed variable $\mathbf{v} = E_{\mathcal{F}} \left(\frac{t-\tau}{(\Delta x)^2 \text{Re}} \right) \mathbf{u}$ yields the transformed system of semi-discrete equations

$$\frac{d\mathbf{v}}{dt} = -H_{\mathcal{F}} \tilde{\mathbf{N}} (H_{\mathcal{F}}^{-1} \mathbf{v} + \mathbf{u}_\infty) - H_{\mathcal{F}} \mathbf{G} \mathbf{d}, \quad \mathbf{G}^\dagger H_{\mathcal{C}}^{-1} \mathbf{v} = 0, \quad (19)$$

where $H_Q = E_Q \left(\frac{t-\tau}{(\Delta x)^2 \text{Re}} \right)$. Using the commutativity properties of integrating factors, Eq. (19) simplifies to

$$\frac{d\mathbf{v}}{dt} = -H_{\mathcal{F}} \tilde{\mathbf{N}} (H_{\mathcal{F}}^{-1} \mathbf{v} + \mathbf{u}_\infty) - \mathbf{G} \mathbf{b}, \quad \mathbf{G}^\dagger \mathbf{v} = 0, \quad (20)$$

where $\mathbf{b} = H_{\mathcal{F}} \mathbf{d}$. We emphasize that the transformed system of equations Eq. (20) is equivalent to the original system of equation Eq. (9). Furthermore, as is the case for Eq. (9), Eq. (20) represents a system of DAEs of index 2.

The procedures for obtaining expressions $\mathbf{G}_L(\mathbf{n})$ can be readily extended to the case of $[\mathbf{G}_E(\alpha)](\mathbf{n})$, where $\mathbf{G}_E(\alpha)$ is the LGF of the integrating factor $E_Q(-\alpha)$. Expressions for $\mathbf{G}_E(\mathbf{n})$ are also provided in Appendix B. As for the case of L_Q^{-1} , fast solutions to expressions involving $\mathbf{G}_E(\alpha)$ are computed using the LGF-FMM.

An important distinction between $\mathbf{G}_L(\mathbf{n})$ and $[\mathbf{G}_E(\alpha)](\mathbf{n})$ is found in their asymptotic behavior. Whereas $|\mathbf{G}_L(\mathbf{n})|$ decays as $1/|\mathbf{n}|$ as $|\mathbf{n}| \rightarrow \infty$, $|\mathbf{G}_E(\alpha)](\mathbf{n})|$ decays faster than any exponential as $|\mathbf{n}| \rightarrow \infty$ for a fixed α .⁵ The fast decay of \mathbf{G}_E implies that, for typical computations, the application of E_Q can be consider a local operation, i.e. values computed at a particular grid location only depend on the values of a few neighboring grid cells. Consequently, the LGF-FMM requires significantly fewer operations to evaluate the action of E_Q compared to the action of L_Q^{-1} .⁶

3. Time integration

3.1. Half-explicit Runge-Kutta methods

Failing to properly identify the semi-discrete form of the governing equations, i.e. Eq. (9), as a system of differential algebraic equations (DAEs) of index 2 prior to choosing a time integration

⁵Consider $[\mathbf{G}_E(\alpha)](\mathbf{n})$ for the case $\mathbf{n} = (n, 0, 0)$. As $n \rightarrow \infty$, $[\mathbf{G}_E(\alpha)](\mathbf{n}) \sim \alpha^n/n!$. For $\alpha = 0.1$ and $\alpha = 1.0$, the value of $[\mathbf{G}_E(\alpha)](\mathbf{n})/[\mathbf{G}_E(\alpha)](\mathbf{0})$ is less than 10^{-10} at $n = 7$ and $n = 13$, respectively. The numerical simulations of Section 6 make use of integrating factors with $\alpha < 1$, but larger values of α are allowed.

⁶For the run parameters of the numerical experiments of Section 6, the action of E_Q only requires approximately 10% of the total number of operations required to compute L_Q^{-1} .

scheme can have undesirable consequences on the quality of the numerical solution [34, 35]. Half-explicit Runge-Kutta (HERK) methods are a type of one-step time integration schemes developed for DAEs of index 2 [34, 36, 37]. Although there are multiple HERK methods [34], we limit our attention to the original HERK method proposed by Hairer et al. [36].

Consider DAE systems of index 2 of the form

$$\frac{dy}{dt} = f(y, z), \quad g(y) = 0, \quad (21)$$

where f and g are sufficiently differentiable, and z is an unknown that must be computed so as to have y satisfy $g(y) = 0$. Problems of this form are of index 2 if the product of partial derivatives $g_y(y)f_z(y, z)$ is non-singular in a neighborhood of the solution. The HERK method applied to Eq. (21) is given by an algorithm similar to that of explicit Runge-Kutta (ERK) methods except that the implicit constraint equation $g(y) = 0$ is solved at each stage of the ERK scheme.

Similarly to standard RK methods, HERK methods can be described by their Butcher tableau:

$$\begin{array}{c|c} \mathbf{c} & \mathbf{A} \\ \hline & \mathbf{b}^\top \end{array}, \quad (22)$$

where $\mathbf{A} = [a_{i,j}]$ is the Runge-Kutta matrix, $\mathbf{b} = [b_i]$ is the weight vector, and $\mathbf{c} = [c_i]$ is the node vector. In subsequent sections, it is often convenient to use the *shifted* tableau notation:

$$\tilde{a}_{i,j} = \begin{cases} a_{i+1,j} & \text{for } i = 1, 2, \dots, s-1 \\ b_j & \text{for } i = s \end{cases}, \quad \tilde{c}_i = \begin{cases} c_{i+1} & \text{for } i = 1, 2, \dots, s-1 \\ 1 & \text{for } i = s \end{cases}. \quad (23)$$

We refer the reader to the discussions of [36, 37] for a detailed algorithm and a list of order-conditions for the general case of Eq. (21).

We now turn our attention to the special case of the transformed semi-discrete governing equations given by Eq. (20). It is convenient to express the non-autonomous system of Eq. (20) in terms of the autonomous system of Eq. (21). This is achieved by letting $y = [v, t]$ and $z = \mathbf{b}$, and by adding $t' = 1$ to Eq. (20). For this case, $g_y = [\mathbf{G}^\top, 0]$ and $f_z = g_y^\top$, where $g_y = [g_u, g_t]$ and $f_z = f_b$. By construction, the operator \mathbf{G} is a constant, which implies that f_z and g_y are also constants. As a result, order-conditions for the general system of Eq. (21) involving high-order derivatives of f_z and g_y are trivially satisfied for the case of Eq. (20). Fewer order-conditions permit a wider range of RK tableaux to be used for a given order of accuracy. This is particularly relevant for high-order HERK schemes, since the number of order-conditions is significantly larger than that of standard RK schemes [37].

The simplifications in the order-conditions obtained for the special case of constant f_z and g_y are well-described in the literature of HERK methods [34, 36–38]. Order-conditions up to order 4 for the y -component reduce to those of standard RK methods [38]. Similarly, order-conditions of order $r \leq 3$ for the z -component (up to fourth-order accurate z -component) reduce to having the shifted sub-tableau $[\tilde{a}_{i,j}]$ for $i, j = 1, 2, \dots, s-1$ satisfy the y -component order-conditions up to order r [37, 38]. It is beyond the scope of the present work to provide an extended discussion on the properties and implementation details of the HERK method for particular RK tableaux. Instead, the order of accuracy and linear stability of a few selected schemes used to perform the numerical experiments of Section 6 is discussed in Section 3.2 and Appendix C, respectively.

3.2. Combined integrating factor and half-explicit Runge-Kutta method

In this section we present a method for obtaining numerical solutions for the (untransformed) discrete velocity and total pressure perturbation by combining the integrating factor technique of

Section 2.3 with the HERK method of Section 3.1. The combined method, referred to as the IF-HERK method, integrates Eq. (6) over $t \in [0, T]$ subject to the initial condition $\mathbf{u}(\mathbf{n}, 0) = \mathbf{u}_0(\mathbf{n})$.

Formally, the IF-HERK method partitions the original problem into a sequences of n sub-problems, where the k -th sub-problem corresponds to numerical integration of Eq. (6) from t_k to t_{k+1} subject to the initial condition $\mathbf{u}(\mathbf{n}, t_k) = \mathbf{u}_k(\mathbf{n})$. We restrict our discussion to the case of equispaced time-steps, i.e. $t_k = t_{k-1} + \Delta t$, since the more general case of variable time-step size is readily deduced.

The k -th sub-problem is solved by first introducing the transformed variables

$$\mathbf{v}(\mathbf{n}, t) = \left[E_{\mathcal{F}} \left(\frac{\Delta t}{(\Delta x)^2 \text{Re}} \right) \right] \mathbf{u}(\mathbf{n}, t), \quad \mathbf{b}(\mathbf{n}, t) = \left[E_{\mathcal{F}} \left(\frac{\Delta t}{(\Delta x)^2 \text{Re}} \right) \right] \mathbf{q}(\mathbf{n}, t), \quad t \in [t_k, t_{k+1}], \quad (24)$$

and using $E_{\mathcal{F}} \left(\frac{\Delta t}{(\Delta x)^2 \text{Re}} \right)$ as an integrating factor for Eq. (9). Next, the HERK method is used to integrate the transformed nonlinear equations from t_k to t_{k+1} in order to obtain $\mathbf{v}_{k+1}(\mathbf{n}) \approx \mathbf{v}(\mathbf{n}, t_{k+1})$ and $\mathbf{b}_{k+1} \approx \mathbf{b}(\mathbf{n}, t_{k+1})$. Finally, values for the discrete velocity and total pressure perturbation at t_{k+1} , i.e. $\mathbf{u}_{k+1}(\mathbf{n}) \approx \mathbf{u}(\mathbf{n}, t_{k+1})$ and $\mathbf{d}_{k+1}(\mathbf{n}) \approx \mathbf{d}(\mathbf{n}, t_{k+1})$, are obtained from \mathbf{v}_{k+1} and \mathbf{b}_{k+1} by using the integrating factor $E_{\mathcal{F}} \left(\frac{-\Delta t}{(\Delta x)^2 \text{Re}} \right)$.

A computationally convenient algorithm for the k -th time-step of the IF-HERK method, subsequently denoted by $(\mathbf{u}_{k+1}, t_{k+1}, \mathbf{p}_{k+1}) \leftarrow \text{IF-HERK}(\mathbf{u}_k, t_k)$, is given by:

1. *initialize*: copy solution values from the k -th time-step,

$$\mathbf{u}_k^0 = \mathbf{u}_k, \quad t_k^0 = t_k. \quad (25)$$

2. *multi-stage*: for $i = 1, 2, \dots, s$, solve the linear system

$$\begin{bmatrix} (\mathbf{H}_{\mathcal{F}}^i)^{-1} & \mathbf{G} \\ \mathbf{G}^\dagger & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_k^i \\ \hat{\mathbf{d}}_k^i \end{bmatrix} = \begin{bmatrix} \mathbf{r}_k^i \\ 0 \end{bmatrix}, \quad (26)$$

where

$$\mathbf{H}_{\mathcal{F}}^i = E_{\mathcal{F}} \left(\frac{(\tilde{c}_i - \tilde{c}_{i-1}) \Delta t}{(\Delta x)^2 \text{Re}} \right), \quad \mathbf{r}_k^i = \mathbf{q}_k^i + \Delta t \sum_{j=1}^{i-1} \tilde{a}_{i,j} \mathbf{w}_k^{i,j} + \mathbf{g}_k^i, \quad (27)$$

$$\mathbf{g}_k^i = -\tilde{a}_{i,i} \Delta t \tilde{\mathbf{N}}(\mathbf{u}_k^{i-1} + \mathbf{u}_{\infty}(t_k^{i-1})), \quad t_k^i = t_k + \tilde{c}_i \Delta t. \quad (28)$$

For $i > 1$ and $j > i$, \mathbf{q}_k^i and $\mathbf{w}_k^{i,j}$ are recursively computed using⁷

$$\mathbf{q}_k^i = \mathbf{H}_{\mathcal{F}}^{i-1} \mathbf{q}_k^{i-1}, \quad \mathbf{q}_k^1 = \mathbf{u}_k^0 \quad (29)$$

$$\mathbf{w}_k^{i,j} = \mathbf{H}_{\mathcal{F}}^{i-1} \mathbf{w}_k^{i-1,j}, \quad \mathbf{w}_k^{i,i} = (\tilde{a}_{i,i} \Delta t)^{-1} (\mathbf{g}_k^i - \mathbf{G} \hat{\mathbf{d}}_k^i). \quad (30)$$

3. *finalize*: define the solution and constraint values of the $(k+1)$ -th time-step,

$$\mathbf{u}_{k+1} = \mathbf{u}_k^s, \quad \mathbf{d}_{k+1} = (\tilde{a}_{s,s} \Delta t)^{-1} \hat{\mathbf{d}}_k^s, \quad t_{k+1} = t_k^s. \quad (31)$$

⁷An efficient implementation of the IF-HERK algorithm recognizes that the application of $s-1$ integrating factors can be avoided during final, $i = s$, stage by computing $\mathbf{r}_k^s = \mathbf{H}_{\mathcal{F}}^{s-1} (\mathbf{q}_k^{s-1} + \Delta t \sum_{j=1}^{s-1} \tilde{a}_{s,j} \mathbf{w}_k^{s-1,j}) + \mathbf{g}_k^s$, as opposed to Eq. (28). This modification avoids having to explicitly compute \mathbf{q}_k^s and $\mathbf{w}_k^{s,j}$ for $j = 1, 2, \dots, s-1$.

The above algorithm is obtained by applying the HERK method to either Eq. (20) or, equivalently, Eq. (19) for the k -th sub-problem, and introducing the auxiliary variables

$$\mathbf{u}_k^i(\mathbf{n}) = \left[\mathbf{E}_{\mathcal{F}} \left(\frac{-\tilde{c}_i \Delta t}{(\Delta x)^2 \text{Re}} \right) \right] \mathbf{v}_k^i(\mathbf{n}), \quad \mathbf{d}_k^i(\mathbf{n}) = \left[\mathbf{E}_{\mathcal{F}} \left(\frac{-\tilde{c}_i \Delta t}{(\Delta x)^2 \text{Re}} \right) \right] \mathbf{b}_k^i(\mathbf{n}), \quad (32)$$

for $i = 1, 2, \dots, s$. We clarify that the intermediate steps used to obtain the final form IF-HERK algorithm make use of the commutativity properties of \mathbf{E}_Q and the identity $\mathbf{E}_Q(\alpha_1)\mathbf{E}_Q(\alpha_2) = \mathbf{E}_Q(\alpha_1 + \alpha_2)$.

The linear operator on the left-hand-side (LHS) of Eq. (26) is symmetric positive semi-definite and its null-space is spanned by the set of $[0, \mathbf{a}]^\top$, where $\mathbf{a} \in \mathbb{R}^C \times \mathbb{R}$ is any discrete linear polynomial. Consequently, the compatibility condition on the pressure field given by Eq. (5) guarantees Eq. (26) has a unique solution. As presented, the IF-HERK algorithm is compatible with any HERK scheme since no assumptions have been made on the RK coefficients. Of course, more efficient versions of this algorithm can potentially be obtained for specific families of RK coefficients, but such details are beyond the scope of the present work.

The IF-HERK schemes used to perform the numerical experiments of Section 6 are given by the following tableaux:

$$\begin{array}{c|ccc} & \text{SCHEME A} & & \\ 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 1 & \frac{\sqrt{3}}{3} & \frac{3-\sqrt{3}}{3} & 0 \\ \hline & \frac{3+\sqrt{3}}{6} & -\frac{\sqrt{3}}{3} & \frac{3+\sqrt{3}}{6} \end{array}, \quad \begin{array}{c|ccc} & \text{SCHEME B} & & \\ 0 & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ 1 & -1 & 2 & 0 \\ \hline & 0 & \frac{3}{4} & \frac{1}{4} \end{array}, \quad \begin{array}{c|ccc} & \text{SCHEME C} & & \\ 0 & 0 & 0 & 0 \\ \frac{8}{15} & \frac{8}{15} & 0 & 0 \\ \frac{2}{3} & \frac{1}{4} & \frac{5}{12} & 0 \\ \hline & \frac{1}{4} & 0 & \frac{3}{4} \end{array}. \quad (33)$$

The order of accuracy, based on the simplified order-conditions discussed in Section 3.1, for each scheme is provided in Table 1. As a point of comparison, Table 1 also provides the expected order of accuracy for general semi-explicit DAEs of index 2, i.e. Eq. (21).

Table 1: Order of accuracy of the solution y variable (velocity perturbation) and constraint z variable (pressure perturbation) based on specialized HERK order conditions. The superscript * denotes values for general semi-explicit DAEs of index 2.

	y -Order	z -Order	y -Order*	z -Order*
Scheme A	2	2	2	2
Scheme B	3	2	3	2
Scheme C	3	1	2	1

The tableaux for Schemes B and C were obtained from [37] and [38]. As discussed in [38], the tableau for Scheme C corresponds to the RK coefficients of the popular three-stage fractional step method of [39]. Unlike Schemes B and C, the tableau for Scheme A was specifically defined for the IF-HERK method. An advantage of Scheme A over Schemes B and C is that the RK nodes, c_i 's, are equally spaced. As a result, the IF-HERK method only requires a single non-trivial integrating factor.⁸ This reduction in the number of distinct LGFs reduces the number of pre-processing operations and lowers the storage requirements of the LGF-FMM. Additionally, extensions of the present method including immersed surfaces, e.g. via the treatment of

⁸One additional integrating factor is required during the last stage of the IF-HERK algorithm, but for the case of $c_s = 1$ this additional integrating factor reduces to the identity operator.

immersed boundaries of [40], can potentially enjoy similar reductions in the computational costs of pre-processing operations by only having to consider a single non-trivial integrating factor. We will report on immersed boundary methods based on the present flow solver in subsequent publications. The linear stability analysis of the IF-HERK method is provided in [Appendix C](#).

3.3. Projection method

It is readily verified that the most computationally expensive operation performed by the IF-HERK method corresponds to solving Eq. (26) for each stage. Systems of continuum or discrete equations similar to Eq. (26) often arise in the literature of numerical methods for simulating incompressible flows. Solutions to these system are frequently obtained through classical projection, fractional-step, or pressure Schur complement methods [41, 42]. These methods can be regarded as approximate block-wise LU decompositions of the original system [41, 42]. More recently, *exact* projection techniques that are free of any matrix/operator approximations have been proposed, e.g. [40, 43]. These techniques have the advantage of not introducing any “splitting errors” and do not require artificial pressure boundary conditions. The present formulation uses an exact projection method to solve Eq. (26), but differs from the methods of [40, 43] in that it does not use the null-space of the discrete operators to obtain solutions to the linear system.

The block-wise LU decomposition of the operator in Eq. (26) suggests a solution procedure, expressed in the standard correction form, given by:

$$\mathbf{u}^* = \mathbf{H}_{\mathcal{F}}^i \mathbf{r}_k^i \quad (\text{compute intermediate velocity}) \quad (34a)$$

$$\mathbf{S} \hat{\mathbf{d}}_k^i = \mathbf{G}^\dagger \mathbf{u}^* \quad (\text{solve for total pressure}) \quad (34b)$$

$$\mathbf{u}_k^i = \mathbf{u}^* - \mathbf{H}_{\mathcal{F}}^i \mathbf{G} \hat{\mathbf{d}}_k^i \quad (\text{projection step}), \quad (34c)$$

where $\mathbf{S} = \mathbf{G}^\dagger \mathbf{H}_{\mathcal{F}}^i \mathbf{G}$ is the Schur complement of the system.⁹ By taking into account the commutativity and mimetic properties of the spatial discretization scheme the procedure given by Eq. (34) simplifies to:

$$\hat{\mathbf{d}}_k^i = -\mathbf{L}_C^{-1} \mathbf{G}^\dagger \mathbf{r}_k^i, \quad \mathbf{u}_k^i = \mathbf{H}_{\mathcal{F}}^i (\mathbf{r}_k^i - \mathbf{G} \hat{\mathbf{d}}_k^i), \quad (35)$$

where $\mathbf{x} = \mathbf{L}_C^{-1} \mathbf{y}$ is equivalent to solving $\mathbf{L}_C \mathbf{x} = \mathbf{y}$ subject to uniform boundary conditions at infinity. In this form, one of the two integrating factors has been eliminated and the original elliptic problem $\mathbf{G}^\dagger \mathbf{H}_{\mathcal{F}}^i \mathbf{G} \mathbf{x} = \mathbf{y}$ has been replaced by the Poisson problem $\mathbf{L} \mathbf{x} = \mathbf{y}$. Reducing the original discrete elliptic problem to a discrete Poisson problem is of significant practical importance since it permits the use of the LGF-FMM with known LGF expressions [7]. As will be discussed in Section 4, the operation count of our overall algorithm is dominated by the cost of solving for the discrete pressure perturbation; therefore, a projection method that is compatible with fast, robust discrete elliptic solvers greatly facilitates obtaining fast flow solutions.

4. Adaptive computational grid

4.1. Restricting operations to a finite computational grid

Thus far we have described algorithms for discretizing and computing the incompressible Navier-Stokes equations on unbounded grids. In this section, we present a method for computing

⁹Without additional information the (scaled) total pressure perturbation, $\hat{\mathbf{d}}_k^i$, obtained from Eq. (34b) is unique up to a discrete linear polynomial. Yet, a unique $\hat{\mathbf{d}}_k^i$ is obtained by taking into account the compatibility condition $\mathbf{p}(\mathbf{n}, t) \rightarrow 0$ as $|\mathbf{n}| \rightarrow \infty$, i.e. $\hat{\mathbf{d}}_k^i(\mathbf{n}) \rightarrow \mathbf{c}_k^i$ as $|\mathbf{n}| \rightarrow \infty$ where $\mathbf{c}_k^i = \frac{1}{2} |\mathbf{u}_\infty(t_k^i)|^2$, discussed in Section 2.1.

solutions, to a prescribed tolerance, using only a finite number of operations. This approximation is accomplished by limiting all operations to a finite computational grid obtained by removing grid cells of the original unbounded grid containing field values that are sufficiently small so as not to significantly affect the evolution of the flow field. As will be demonstrated in the following discussion, the ability of the present method to only track a finite region of the unbounded domain is a consequence of the exponential decay of the vorticity at large distances, which is assumed for all flows under consideration.

We first consider the error resulting from neglecting field values outside a finite region when solving the elliptic problems of the IF-HERK method.¹⁰ Using the notation of Section 2.2, the solution to the discrete Poisson problem of Eq. (35) is given by

$$\hat{\mathbf{d}}(\mathbf{n}) = [\mathbf{G}_L * \mathbf{f}](\mathbf{n}), \quad \mathbf{f}(\mathbf{n}) = [-\mathbf{G}^\dagger \mathbf{r}_k^i](\mathbf{n}). \quad (36)$$

The source field $\mathbf{G}^\dagger \mathbf{r}_k^i$ is a discrete approximation of $\nabla \cdot \boldsymbol{\ell}$ at $t \approx k\Delta t$, where $\boldsymbol{\ell} = \boldsymbol{\omega} \times \mathbf{u}$ is the Lamb vector. It follows from the assumption that $\boldsymbol{\omega}$ is exponentially small at large distances that $\nabla \cdot \boldsymbol{\ell}$ and $\mathbf{G}^\dagger \mathbf{r}_k^i$ must also be exponentially small at large distances. As a result, the induced field of Eq. (36) is computed to a prescribed tolerance by defining the finite computational domain such that it includes the region where the magnitude of $\mathbf{G}^\dagger \mathbf{r}_k^i$ is greater than some positive value.

The action of all operators present in the IF-HERK and projection algorithms, with the exception of \mathbf{L}_C^{-1} , are evaluated using only a few local operations. Many of these local operators act on fields that typically decay algebraically, e.g. \mathbf{u} and \mathbf{d} . As a result, the technique of only tracking regions with non-negligible source terms used for Eq. (35) is impractical for most other operations required by the IF-HERK method. Unlike the action of \mathbf{L}_C^{-1} , the action of local operators only incurs an error limited to a few cells near the boundary of a finite region if field values of outside the region are ignored, i.e. taken to be zero. Furthermore, repeated applications of local operators only propagate the error into the interior of the region by a few grid cells per application. This type of error is prevented from significantly affecting the solution in the interior by padding the interior with buffer grid cells and by periodically computing (“refreshing”) \mathbf{u} from the discrete vorticity, $\mathbf{w} = \mathbf{C}\mathbf{u}$, which, like $\mathbf{G}^\dagger \mathbf{r}_k^i$, has bounded approximate support. As a result, the approximate support of both $\mathbf{G}^\dagger \mathbf{r}_k^i$ and \mathbf{w} must be contained in the finite computational domain. Bounds for the error resulting from approximating the support of these fields and estimates for the number of time steps that can elapse before the velocity needs to be refreshed will be discussed in Sections 4.3 and 4.4, respectively.

We recall that the discrete velocity perturbation \mathbf{u} is subject to the constraint $\mathbf{G}^\dagger \mathbf{u} = 0$ and that the null-space of \mathbf{G}^\dagger is spanned by the image of \mathbf{C}^\dagger . As a result, it is possible to express \mathbf{u} as

$$\mathbf{u} = \mathbf{C}^\dagger \mathbf{a}, \quad (37)$$

where $\mathbf{a} \in \mathbb{R}^E$ can be regarded as the discrete vector potential or streamfunction. Additionally, we require $\mathbf{D}\mathbf{a} = 0$. The discrete vorticity, \mathbf{w} , can now be expressed in terms of \mathbf{a} as

$$\mathbf{w} = \mathbf{C}\mathbf{C}^\dagger \mathbf{a} = (\mathbf{C}\mathbf{C}^\dagger + \mathbf{D}^\dagger \mathbf{D}) \mathbf{a} = -\mathbf{L}_E \mathbf{a}. \quad \mathbf{D}\mathbf{w} = 0 \quad (38)$$

Finally, Eq. (37) and (38) provide an expression for \mathbf{u} in terms of \mathbf{w} ,

$$\mathbf{u} = -\mathbf{C}^\dagger \mathbf{L}_E^{-1} \mathbf{w}, \quad (39)$$

¹⁰Field values outside the finite region being tracked are treated as zero.

where $L_{\mathcal{E}}^{-1}$ imposes zero boundary conditions at infinity.¹¹ As expected, the expressions relating \mathbf{u} , \mathbf{w} , and \mathbf{a} are analogous to the continuum expressions relating the velocity, vorticity, and streamfunction fields. We emphasize that Eq. (37), (38), and (39) were obtained through the algebraic properties of the discrete operators, as opposed to the discretization of continuum equations.

The present formulation can be cast into an equivalent vorticity formulation simply by taking the discrete curl of Eq. (9) and computing \mathbf{u} , which is required to evaluate the non-linear term, using Eq. (39). This formulation is not pursued since each stage of the IF-HERK would require solving a discrete *vector* Poisson problem, as opposed to a discrete *scalar* Poisson problem, which would in turn roughly triple the cost of each stage.¹² The vorticity formulation has the advantage of not having to periodically evaluate Eq. (39) to refresh \mathbf{u} , but, as will be discussed in Sections 4.4, this operation occurs, at most, once per time step. Based on the stability analysis of Appendix C, RK schemes with a minimum of three stages are required to ensure stable solutions. As a result, the primitive variable formulation is approximately 1.5 to 3 times faster than the vorticity formulation. Differences in the errors between the two algebraically-equivalent formulations associated with the finite tolerances used to compute the LGF-FMM and the adaptive grid algorithms can be used to further distinguish each formulation. However, such differences in errors are not considered here since they are expected to be on the order of the prescribed tolerances, which, as will be discussed in Section 5, are specified to be much smaller than the discretization errors of practical flows.

4.2. Block-structured active computational grid

We now turn our attention to the formal definition of the finite region of the unbounded computational domain tracked by our formulation, which we refer to as the *active* computational domain. Consider partitioning the unbounded staggered Cartesian grid described in Section 2 into an infinite set of equally sized blocks arranged on a logically Cartesian grid. The block corresponding to the $\mathbf{n} = (i, j, k)$ location is denoted by $B(\mathbf{n})$ or, equivalently, $B_{i,j,k}$, and the union of all blocks is denoted by D_{∞} . Each block is defined as a finite staggered Cartesian grid of $n_1^b \times n_2^b \times n_3^b$ cells. We limit our attention to the case in which each block contains the same number of cells in each direction, i.e. $n_i^b = n^b$, but note that the subsequent discussion readily extends to the general case. As a practical consideration, a layer of buffer or ghost grid cells surrounding each block is introduced to facilitate the implementation of the present algorithm.

Figure 2 depicts the three nested sub-domains $D_{\text{supp}} \subseteq D_{\text{soln}} \subset D_{\text{xsoln}} \subset D_{\infty}$ that constitute the active computational domain. These sub-domains are defined as:

- *Support blocks* (D_{supp}): union of blocks that defines the support of the source field of the discrete Poisson problems of Eq. (35) and (38).
- *Solution blocks* (D_{soln}): union of blocks that tracks the solution fields \mathbf{u} and \mathbf{d} . All field values defined in the blocks belonging to D_{soln} are regarded as accurate approximations of the field values computed using an unbounded domain.

¹¹Without further considerations Eq. (38) implies that \mathbf{a} is unique up to a discrete linear polynomial. Given that \mathbf{w} is exponentially small at large distances and that \mathbf{u} tends to zero at infinity, it follows that \mathbf{a} is unique up to an arbitrary constant taken to be zero.

¹²For the test case of the extremely thin $\delta/R = 0.0125$ vortex ring discussed in Section 6.3, the wall-time ratio of a vector to a scalar discrete Poisson solve is approximately 2.8, which is slightly less than the expected ratio of 3 based on operation count estimates of the LGF-FMM due to the larger parallel communication costs per problem unknown for the scalar case.

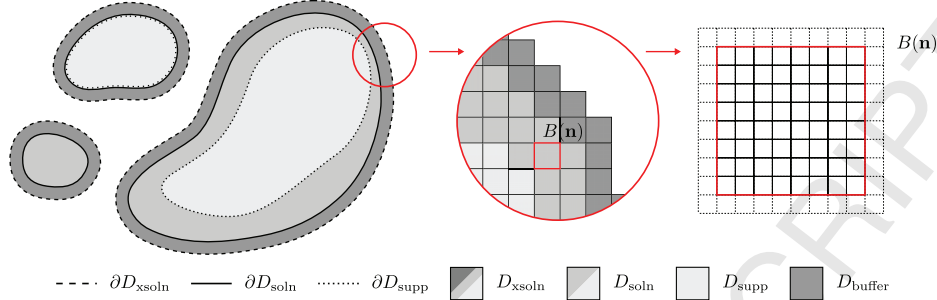


Figure 2: Depiction of the finite computational domain in two-dimensions. Distant view of the three nested sub-domains $D_{\text{supp}} \subseteq D_{\text{soln}} \subset D_{\text{xsoln}}$ defined in the main text (left). Zoomed-in view illustrating the union of blocks used to define the domain (middle). Magnified view of an individual block (right). Each block is defined as a finite staggered Cartesian grid; dashed cells surrounding the interior grid correspond to buffer or ghost grid cells.

- *Expanded solution blocks* (D_{xsoln}): union of blocks given by a non-trivial neighborhood of D_{soln} . We limit our attention to neighborhoods defined by the union of blocks that are at most N_b blocks away from any block contained in D_{soln} ,

$$D_{\text{xsoln}} = \{B(\mathbf{m}) : |\mathbf{n} - \mathbf{m}| \leq N_b, B(\mathbf{n}) \in D_{\text{soln}}, \mathbf{m}, \mathbf{n} \in \mathbb{Z}^3\}. \quad (40)$$

- *Buffer blocks* (D_{buffer}): union of blocks belonging to D_{xsoln} , but not belonging to D_{soln} , i.e. $D_{\text{buffer}} = D_{\text{xsoln}} \setminus D_{\text{soln}}$. (The domain D_{buffer} is not one of the three primary sub-domains, but it is introduced to facilitate the subsequent discussion.)

The criteria for selecting which blocks belong to D_{supp} and D_{soln} are discussed in Section 4.3, and the techniques for selecting values of N_n discussed in Section 4.4.

We now introduce the “mask operator” $M_Q^\gamma : \mathbb{R}^Q \mapsto \mathbb{R}^Q$ associated with the grid space Q and the domain γ , which is defined by

$$[M_Q^\gamma \mathbf{q}](\mathbf{n}) = \begin{cases} \mathbf{q}(\mathbf{n}) & \text{if } \mathbf{n} \in \text{ind}[B] \text{ and } B \in D_\gamma \\ 0 & \text{otherwise} \end{cases}, \quad (41)$$

where $\mathbf{q} \in \mathbb{R}^Q$, and $\text{ind}[B]$ denotes the set of all indices of the unbounded staggered grid associated with block B . Mask operators are subsequently used to formally define operations performed on finite domains. For example, the operation \mathbf{Gd} perform over D_{xsoln} is defined as $M_{\mathcal{F}}^{\text{xsoln}} \mathbf{G} M_C^{\text{xsoln}} \mathbf{d}$. For this particular operation, the values of $M_C^{\text{xsoln}} \mathbf{G} M_C^{\text{xsoln}} \mathbf{d}$ and \mathbf{Gd} are equivalent for grid cells in D_{xsoln} , except for a single layer of grid cells on the boundary of D_{xsoln} . Computationally efficient implementations of $M_Q^\gamma \mathbf{A} M_Q^\gamma$ recognize that all non-trivial numerical operations are limited to grid cells contained in either D_γ and $D_{\gamma'}$.

4.3. Adaptivity

In this section we discuss the criteria used to select the blocks belonging to D_{supp} and D_{soln} . It follows from subsequent discussions that the field values on $D_{\text{soln}} \setminus D_{\text{supp}}$ can be computed as a post-processing step from the field values on D_{supp} ; therefore, only the criteria used to define

the D_{supp} affects the accuracy of the computed flow field. We allow for $D_{\text{soln}} \neq D_{\text{supp}}$ in order to emphasize that the present algorithm is able to track values of u and d over arbitrary regions of interest.

Consider a function W that maps an unbounded grid of blocks, i.e. D_∞ , to an unbounded grid of positive real scalars. We define the support and solution regions as

$$D_{\text{supp}} = \{B(\mathbf{n}) : [W_{\text{supp}}(D_\infty)](\mathbf{n}) > \epsilon_{\text{supp}}, \mathbf{n} \in \mathbb{Z}^3\}, \quad (42a)$$

$$D_{\text{soln}} = \{B(\mathbf{n}) : [W_{\text{soln}}(D_\infty)](\mathbf{n}) > \epsilon_{\text{soln}}, \mathbf{n} \in \mathbb{Z}^3\}, \quad (42b)$$

respectively. The functions W_{supp} and W_{soln} , and the scalars ϵ_{supp} and ϵ_{soln} are referred to as weight functions and threshold levels, respectively.

Although the weight function W_{supp} can be defined to reflect any block selection criteria, we limit our attention to cases for which $[W_{\text{supp}}(D_\infty)](\mathbf{n})$ reflects the magnitude of the fields Cu and $G^\dagger \tilde{N}(u + u_\infty)$ over the block $B(\mathbf{n})$. This choice of W_{supp} facilitates establishing relationships between the threshold level ϵ_{supp} and the error incurred by neglecting source terms values outside D_{supp} when solving the discrete Poisson problems of Eq.(35) and (38). As a representative example, we consider the weight function W_{supp} given by

$$[W_{\text{supp}}(D_\infty)](\mathbf{n}) = \max(\mu(\mathbf{n})/\mu_{\text{global}}, \nu(\mathbf{n})/\nu_{\text{global}}), \quad (43a)$$

$$\mu(\mathbf{n}) = \max_{\mathbf{m} \in \text{ind}[B(\mathbf{n})]} |[Cu](\mathbf{n})|, \quad \mu_{\text{global}} = \max_{\mathbf{n} \in \mathbb{Z}^3} \mu(\mathbf{n}), \quad (43b)$$

$$\nu(\mathbf{n}) = \max_{\mathbf{m} \in \text{ind}[B(\mathbf{n})]} |[G^\dagger \tilde{N}(u + u_\infty)](\mathbf{n})|, \quad \nu_{\text{global}} = \max_{\mathbf{n} \in \mathbb{Z}^3} \nu(\mathbf{n}). \quad (43c)$$

In the absence of any error associated with computing the action of L_Q^{-1} , this expression for W_{supp} results in an upper bound of ϵ_{supp} for the point-wise normalized residual of the active domain approximations of Eq. (35) and (38).¹³ For these cases, the point-wise normalized residual is defined as $\|r\|_\infty / \|x\|_\infty$, where

$$r = x - M_{\text{supp}}^{\text{supp}} L_Q y, \quad y = M_Q^{\text{soln}} L_Q^{-1} M_Q^{\text{supp}} x, \quad (44)$$

and x is the source field of the corresponding discrete Poisson problem.

In general, as the solution changes over time the domain D_{supp} , as defined by Eq. (42) and Eq. (42a), will also change. Significant amounts of non-negligible source terms are prevented from being advected or diffused outside D_{supp} by recomputing and, if necessary, reinitializing the active domain at the beginning of a time-step. This operation is performed by first computing $w \leftarrow Cu$ and $q \leftarrow -G^\dagger \tilde{N}(u + u_\infty)$ on D_{xsoln} . Next, values of w and q of grid cells belonging to block in D_{buffer} that have been significantly contaminated by finite boundary errors are zeroed. Finally, $[W_{\text{supp}}(D_\infty)](\mathbf{n})$ and $[W_{\text{soln}}(D_\infty)](\mathbf{n})$ are computed using Eq. (42a) for all $\mathbf{n} \in \mathbb{Z}^3$ such that $B(\mathbf{n}) \in D_{\text{xsoln}}$ and are set to zero otherwise.

If either of the newly computed D_{supp} or D_{soln} differ from their respective previous values, then it is necessary to reinitialize the active grid and compute the discrete velocity perturbation,

¹³Formally, ϵ_{supp} is only an approximate upper bound for the active domain case of Eq. (35) since the source field for this problem is not exactly equal to $-G^\dagger \tilde{N}(u + u_\infty)$. Yet, for the present error estimates, numerical experiments of representative flows indicate that $-G^\dagger \tilde{N}(u + u_\infty)$ at $t = t_k$ is a good approximation to $G^\dagger r_k^i$ of each stage of the k -th time-step.

u , over the new D_{xsoln} . By construction, all non-negligible values of the discrete vorticity, w , are contained in D_{supp} ; therefore, u over D_{xsoln} can be computed as

$$a \leftarrow -M_{\mathcal{E}}^{\text{xsoln}} L_Q^{-1} M_{\mathcal{E}}^{\text{supp}} w, \quad u \leftarrow M_{\mathcal{E}}^{\text{xsoln}} C^\dagger M_{\mathcal{E}}^{\text{xsoln}} a. \quad (45)$$

Subsequently, we denote the procedure implied by Eq. (45) as $u \leftarrow \text{Vor2Vel}(w)$.

We emphasize that the present algorithm is also compatible with other choices of weight functions. Using weight functions that are well-suited for capturing the relevant flow physics of a particular application can potentially reduce the size of the active domain and the number of operations required to accurately simulate the flow. For example, if we are primarily interested in capturing the local physics of a flow over a particular region centered at \mathbf{x}_0 , then a weight function $|\mathbf{n} - \mathbf{x}_0|^{-\alpha} [W(D_\infty)](\mathbf{n})$ with $\alpha > 0$ and W given by Eq. (42a) might be an appropriate choice. Unless otherwise stated, subsequent discussions assume that W_{supp} is defined by Eq. (43a).

4.4. Velocity refresh

In this section we present a set of techniques for limiting the error introduced from truncating non-compact fields that decay algebraically, e.g. u and d , when computing the action of local operators. We limit the present discussion to issues that arise from evaluating expressions involving $E_Q^L(\alpha)$ on the finite active domain since this operator has the largest stencil of all local operators involved in the IF-HERK and projection methods.

We recall that the action of $E_Q^L(\alpha)$ on $q \in \mathbb{R}^Q$ is computed as $[G_E(\alpha) * q](\mathbf{n})$. Formally, $G_E(\alpha)$ has an infinite support, but, as discussed in Section 2.2, $[G_E(\alpha)](\mathbf{n})$ decays rapidly as $|\mathbf{n}| \rightarrow \infty$; therefore, it is possible to approximate $G_E(\alpha)$ to prescribed tolerance using a finite support. Consequently, for a given α , there exists some $n_E \in \mathbb{Z}$ such that the field induced from an arbitrary source field can be computed at a distance $n_E \Delta x$ from $\partial D_{\text{xsoln}}$ to a prescribed accuracy ϵ_E . By choosing the parameter N_b , used to define D_{xsoln} in Eq. (40), to be equal or greater than $\lceil n_E / n^b \rceil$ it is possible to evaluate the action of $E_Q^L(\alpha)$ on D_{soln} to an accuracy ϵ_E . As a result, the flow inside D_{soln} remains an accurate approximation of the flow that would have been obtained using the entire unbounded grid.

As the solution is evolved using the IF-HERK method, the operator $E_Q^L(\alpha)$ is repeatedly applied to various grid functions, causing the error associated with truncated non-compact source fields to progressively propagate into the interior of D_{xsoln} . The action of $\prod_{i=1}^n M_Q^{\text{xsoln}} E_Q^L(\alpha_i) M_Q^{\text{xsoln}}$ is well-approximated by $M_Q^{\text{xsoln}} E_Q^L(\beta) M_Q^{\text{xsoln}}$, where $\beta = \sum_{i=1}^n \alpha_i$. Given that the physical values of the nonlinear terms in the IF-HERK algorithm are approximately zero on D_{buffer} , the minimum buffer region required to integrate u over q time-steps is determined by the support of $G_E(q\beta)$, where $\beta = \sum_{i=1}^s \frac{\Delta \tilde{c}_i \Delta t}{(\Delta x)^2 \text{Re}} = \frac{\Delta t}{(\Delta x)^2 \text{Re}}$. A procedure for obtaining estimates for n_E from q and β is provided in Appendix B. This procedure is extended to obtain an upper bound, q_{max} , on the number of time-steps, q , before the error at prescribed distance $n_E \Delta x$ away from $\partial D_{\text{xsoln}}$ exceeds a prescribed value of ϵ_E . At its minimum, the depth of the buffer region is $n^b N_b \Delta x$; therefore, the present method takes n_E to be equal to $n^b N_b$.

Provided $q_{\text{max}} \geq 1$, the solution is integrated over multiple time-steps before the error from truncating non-compact source field starts to significantly affect the accuracy of the solution on D_{soln} .¹⁴ In order to maintain the prescribed accuracy, after q_{max} time-steps the discrete velocity perturbation on D_{xsoln} is recomputed or *refreshed* from the discrete vorticity on D_{supp} using the Vor2Vel procedure.

¹⁴Combinations of n^b , N_b , and β resulting in $q_{\text{max}} = 0$ are not allow. For a given β , the value of $q_{\text{max}} = 0$ can always be increased by using larger values of n^b or N_b .

5. Algorithm summary

The present method for solving the incompressible Navier-Stokes on formally unbounded Cartesian grids using a finite number of operations and storage, referred to as the NS-LGF method, is summarized in this section. Implementation details are omitted since they are beyond the scope of the present work. Instead, we refer the reader to the parallel implementation of the LGF-FMM [7], which can be readily extended to accommodate the additional operations required by the NS-LGF method.

An outline of the steps performed by the NS-LGF algorithm at k -th time-step is as follows:

1. *Preliminary*: compute the discrete vorticity, \mathbf{w}_k , and divergence of the Lamb vector, \mathbf{q}_k .

$$\mathbf{w}_k \leftarrow \mathbf{M}_{\mathcal{E}}^{\text{xsoln}} \mathbf{C} \mathbf{M}_{\mathcal{F}}^{\text{xsoln}} \mathbf{u}_k, \quad (46a)$$

$$\mathbf{q}_k \leftarrow -\mathbf{M}_{\mathcal{C}}^{\text{xsoln}} \mathbf{G}^\dagger \mathbf{M}_{\mathcal{F}}^{\text{xsoln}} \tilde{\mathbf{N}}(\mathbf{M}_{\mathcal{F}}^{\text{xsoln}}(\mathbf{u}_k + \mathbf{u}_{\infty}(t_k))). \quad (46b)$$

2. *Grid update*: update the computational grid based on prescribed criteria.

- (a) *Query*: use weight functions W_{supp} and W_{soln} , threshold values ϵ_{supp} and ϵ_{soln} , and fields \mathbf{w}_k and \mathbf{q}_k to determine whether D_{supp} or D_{soln} need to be updated.
- (b) *Update*: (if necessary) update D_{supp} , D_{soln} , and D_{xsoln} by adding or removing blocks. Copy the values of the discrete vorticity from the old to the new computational grid for $\forall B \in D_{\text{supp}}^{\text{new}} \cap D_{\text{supp}}^{\text{old}}$, where $D_{\text{supp}}^{\text{new}}$ and $D_{\text{supp}}^{\text{old}}$ denote D_{supp} before and after the update, respectively.

3. *Velocity refresh*: compute the discrete velocity perturbation, \mathbf{u}_k , from the discrete vorticity, \mathbf{w}_k .

- (a) *Query*: this operation is required if either the grid has been updated or if the number of time-steps since the last refresh is equal or greater than q_{max} .
- (b) *Refresh*: (if necessary) compute \mathbf{u}_k using:

$$\mathbf{u}_k \leftarrow \text{Vor2Vel}(\mathbf{w}_k), \quad (47)$$

where the Vor2Vel procedure given by Eq. (45).

4. *Time integration*: compute \mathbf{u}_{k+1} , t_{k+1} , and \mathbf{p}_{k+1} using:

$$(\mathbf{u}_{k+1}, t_{k+1}, \mathbf{p}_{k+1}) \leftarrow \text{xIF-HERK}(\mathbf{u}_k, t_k), \quad (48)$$

where the xIF-HERK algorithm is the finite computational grid version of the IF-HERK algorithm.

The xIF-HERK algorithm is identical to the IF-HERK algorithm, except for the presence of mask operators which are used to confine all operations to the finite active domain. With the exception of a few special cases, the xIF-HERK algorithm is obtained by operating from the left all operators and grid functions present in the IF-HERK algorithm by the appropriate $\mathbf{M}_Q^{\text{xsoln}}$, e.g. $\mathbf{A} \rightarrow \mathbf{M}_Q^{\text{xsoln}} \mathbf{A}$ and $\mathbf{y} \rightarrow \mathbf{M}_Q^{\text{xsoln}} \mathbf{y}$. The exceptions to this rule correspond to the expressions for \mathbf{g}_k^i and $\hat{\mathbf{d}}_k^i$, which are given by

$$\mathbf{g}_k^i = \tilde{a}_{i,i} \Delta t \mathbf{M}_{\mathcal{F}}^{\text{xsoln}} \tilde{\mathbf{N}}(\mathbf{M}_{\mathcal{F}}^{\text{xsoln}}(\mathbf{u}_k^{i-1} + \mathbf{u}_{\infty}(t_k^{i-1}))), \quad (49a)$$

$$\hat{\mathbf{d}}_k^i = -\mathbf{M}_{\mathcal{C}}^{\text{xsoln}} \mathbf{L}_{\mathcal{C}}^{-1} \mathbf{M}_{\mathcal{C}}^{\text{supp}} \mathbf{G}^\dagger \mathbf{M}_{\mathcal{F}}^{\text{xsoln}} \mathbf{r}_k^i. \quad (49b)$$

Both Eq. (49a) and (49b) reflect the fact that, by construction, the non-negligible physical values of \mathbf{w}_k and \mathbf{q}_k are contained in W_{supp} .

The operation count for the k -th time-step of the NS-LGF method, denoted by N_k^{NS} , is dominated by the number of operations required to evaluate the actions of L_Q^{-1} and E_Q^L . As a result, an estimate for N_k^{NS} is given by:

$$N_k^{\text{NS}} \approx sN_k^L + 3C(s)N_k^E + \lceil 3N_k^L \rceil_k, \quad (50)$$

where s is the number of stages of the HERK scheme. N_k^L and N_k^E denote the number of operations required to compute the action of $M_Q^{\text{xsoln}}L_Q^{-1}M_Q^{\text{supp}}$ and $M_Q^{\text{xsoln}}L_Q^{-1}M_Q^{\text{xsoln}}$, respectively, using the LGF-FMM for scalar grid spaces.¹⁵ Detailed estimates for the values of N_k^L and N_k^E can be obtained from the discussion of the LGF-FMM [7], but we note here that both N_k^L and N_k^E scale as $O(N)$ for sufficiently large values of N , where N is the total number of grid cells of the active domain. The notation $\lceil \cdot \rceil_k$ is used to clarify that cost associated with velocity update, i.e. $3N_k^L$, should only be included if a velocity update is performed. Lastly, $C(s)$ specifies the number of integrating factors required by an s -stage IF-HERK scheme. In general, $C(s)$ is equal to $C_0(s)$, where

$$C_0(s) = s + \left\lceil \frac{(s-1)s}{2} \right\rceil. \quad (51)$$

For special case of second-order IF-HERK schemes, $C(s)$ reduces to $C_0(s) - 1$.¹⁶

For convenience, a summary of the parameters used in our treatment of the active computational domain is provided by Table 2. Of the parameters listed in Table 2, only ϵ_{FMM} , ϵ_E , and ϵ_{supp}

Table 2: Parameters used in the treatment of the finite computational domain.

Symbol	Description	Section
N_b	Width of W_{buffer} (no. blocks)	4.1
n^b	Block size (no. cells)	4.1
ϵ_{FMM}	LGF-FMM tolerance	2.2
ϵ_{supp}	Support region threshold	4.3
ϵ_E	Buffer region tolerance	4.4

affect the accuracy of the numerical simulation. The *solution error* of the NS-LGF method, i.e. the error associated with approximately solving the fully discretized unbounded grid equations, is approximately bounded above by the sum of these three parameters.

The field values used to compute D_{supp} should represent field values that would be obtained using the unbounded grid in the absence of numerical errors associated with the evaluation of discrete operators. Spurious and unnecessary changes to the active domain are avoided by requiring

$$\max(\epsilon_{\text{FMM}}, \epsilon_E) < \alpha \epsilon_{\text{supp}}, \quad (52)$$

where $\alpha < 1$ is a safety parameter specifying the sensitivity of the adaptive scheme to the solution errors associated with ϵ_{FMM} and ϵ_E .¹⁷ Furthermore, using parameters that satisfy Eq. (52)

¹⁵The factor of 3 that appears in the second and third terms of Eq. (50) accounts for the additional operations required to solve vector Poisson problems and vector integrating factors.

¹⁶The expression $c_s = 1$ is one of the HERK order-conditions associated with second-order accurate constraints. For the case of $c_s = \bar{c}_{s-1} = 1$, the integrating factor $H_{\mathcal{F}}^1$, defined by Eq. (27), simplifies to the identity operator.

¹⁷Numerical experiments of representative flows have shown that $\alpha \approx 0.1$ is sufficiently small as to avoid most spurious and unnecessary changes to the computational grid.

eliminates the inclusion of blocks that only contain field values that are on the same order as the solution error.

The values for n^b and N_b can also significantly affect the number of numerical operations performed by the NS-LGF method. Smaller values of n^b typically result in smaller active domains, but require more frequent velocity updates and often require the use of LGF-FMM schemes with less than optimal computational rates. In practice, computationally efficient schemes are obtained by setting $N_b = 1$ and determining the lower bound for n^b , denoted by n_0^b , from the prescribed value of ϵ_E . Next, starting from n_0^b , progressively larger values of n^b are considered until an efficient LGF-FMM scheme that achieves the prescribed ϵ_{FMM} tolerance is obtained. The construction and computational performance of LGF-FMM schemes are discussed in [7].

6. Verification examples

The behavior of the NS-LGF method is verified through numerical simulations of thin vortex rings. We consider vortex rings of ring-radius R and core-radius δ , with circulation Γ and Reynolds number $\text{Re} = \frac{\Gamma}{\nu}$, where ν is the kinematic viscosity of the fluid. Unless otherwise stated, simulations are initiated with a vorticity distribution given by

$$\omega_\theta(r, z) = \frac{\Gamma}{\pi\delta^2} \exp\left(\frac{z^2 + (r - R)^2}{\delta^2}\right), \quad \omega_z(r, z) = 0, \quad (53)$$

where $r = x^2 + y^2$ and $\theta = \tan^{-1}(y/x)$. As a result, the vortex ring initially translates in the positive z -direction due to its self-induced velocity [21].

The numerical experiments discussed in this section are initialized by first specifying an initial discrete vorticity, w_0 , and then using Eq. (45) to obtain an initial discrete velocity perturbation, u_0 . This procedure naturally leads to a u_0 that is compatible with the IF-HERK method, i.e. $G^\dagger u_0 = 0$. The initial active domain is chosen such that the $|\omega| < 10^{-10}$ outside the D_{supp} . In order to avoid significant numerical artifacts due to the jump in the direction of the vorticity field at the ring origin, we limit our attention to vortex rings for which $|\omega_{\text{center}}| < 10^{-10} \max |\omega|$, where ω_{center} is the value of ω at the center of the ring. For the case of Eq. (53), this condition is satisfied for $\delta/R < 0.2$.

Provided a sufficiently large initial active domain, any sufficiently accurate process for computing w_0 from ω_0 can be used to initialize the numerical simulations. Yet it is convenient to use a process that naturally leads to a w_0 such that $Dw_0 \approx 0$. In the absence of any numerical errors, $\tilde{w}_0 = Cu_0$ is equal to w_0 if and only if $Dw_0 = 0$. For the case of $Dw_0 \neq 0$, the support of \tilde{w}_0 is typically larger than the support of w_0 , which in turn leads to larger active domains and complicates initial error estimates, i.e. $|w_0| < \epsilon$ in D_{supp} does not imply $|\tilde{w}_0| < \epsilon$ in D_{supp} . Provided $\nabla \cdot \omega = 0$, it is possible to construct w_0 such that the magnitude of Dw_0 is less than a prescribed tolerance by computing approximate values of the vorticity flux over the faces of the dual grid and applying the Divergence theorem to each dual cell.¹⁸ For all test cases, a high-order quadrature scheme is used to integrate the initial vorticity distribution over the faces of the dual grid such that the resulting w_0 satisfies $\|Dw_0\|_\infty \approx 10^{-10}$.

¹⁸The dual grid corresponds to a copy of the original staggered grid that has been shifted by half a grid cell in each direction. Cells, faces, edges, and vertices of the original grid can be regarded as vertices, edges, faces, and cells, respectively, of the dual grid.

Test cases are performed using $n^b = 16$ and $N_b = 1$. This choice of parameters leads to $\epsilon_{\text{FMM}} < 10^{-8}$ for all values of Δx , Δt , and Re considered. The values of ϵ_{supp} and ϵ_{E} are taken to be $\epsilon_{\text{supp}} = 0.1\epsilon^*$ and $\epsilon_{\text{E}} = \epsilon^*$. The value of ϵ^* is varied across different sets of simulations, but is always such that $10^{-8} \leq \epsilon^* \leq 10^{-2}$. The support domain D_{supp} is computed using Eq. (42a) and Eq. (43a), and the solution domain D_{soln} is set to be equal to D_{supp} . It follows from our choice of parameters that the overall solution error is always bounded above by ϵ^* .¹⁹

With the exception of a few test cases discussed in Section 6.1, all numerical experiments are performed using the IF-HERK scheme denoted as “Scheme A” in Section 3.2. The time-step size, Δt , is held fixed during each simulation and chosen such that the CFL, based on the maximum point-wise velocity magnitude, does not exceed 0.75. Unless otherwise stated, the freestream velocity, u_{∞} , is set to be zero.

6.1. Discretization error

The order of accuracy of the discretization techniques is verified using spatial and temporal refinement studies on the early evolution of a vortex ring at $\text{Re}_0 = 1,000$ with initial vorticity distributions given by

$$\omega_{\theta}(r, z) = \begin{cases} \alpha \frac{\Gamma}{R^2} \exp(-4s^2/(R^2 - s^2)) & \text{if } s \leq R \\ 0 & \text{otherwise} \end{cases}, \quad \omega_z(r, z) = 0, \quad (54)$$

where $s^2 = z^2 + (r - R)^2$ and α is chosen such that ω_{θ} integrates to Γ , i.e. $\alpha \approx 0.54857674$.²⁰ Test cases are performed using fixed grids that are sufficiently large such that at any time-step of the simulation the active domain corresponds to a value of ϵ^* less than 10^{-8} .

We use $\epsilon_{\mathbf{u}} = \|\mathbf{u} - T_{\mathcal{F}}\mathbf{u}^*\|_{\infty}/\|\mathbf{u}^*\|_{\infty}$ and $\epsilon_{\mathbf{p}} = \|\mathbf{p} - T_{\mathcal{C}}\mathbf{p}^*\|_{\infty}/\|\mathbf{p}^*\|_{\infty}$ to approximate the error at time T of the velocity field, \mathbf{u} , and the pressure field, \mathbf{p} , respectively. The superscript $*$ is used to denote grid functions obtained from the test case with the highest resolution, i.e. smallest Δx or Δt , included in the corresponding refinement study. Point-wise comparisons between grid functions at different refinement levels are made possible through the use of the coarsening operators $T_{\mathcal{F}}$ and $T_{\mathcal{C}}$. Finally, we define $\|\mathbf{x}\|_{\infty}$ as the maximum value of $|\mathbf{x}(\mathbf{n})|$ for all \mathbf{n} associated with grid locations in D_{soln} .

The spatial refinement study consists of seven test cases corresponding to $\Delta x/\Delta x_0 = 2^0, 2^{-1}, \dots, 2^{-6}$. Test cases are performed using the same Δt , and $\epsilon_{\mathbf{u}}$ and $\epsilon_{\mathbf{p}}$ are evaluated at $T = 10\Delta t$. The computational grids are constructed such that the location of vertices of coarser grids always coincide with the location of vertices of finer grids. This enables the coarsened solution fields $T_{\mathcal{C}}\mathbf{p}^*$ and $T_{\mathcal{F}}\mathbf{u}^*$ to be computed by recursively averaging the values of the 8 (4) fine grid cells (faces) occupying the same physical region as the corresponding coarse grid cell (face). The slope of the error curves depicted in the left plot of Figure 3 verifies that the solutions are second-order accurate in Δx .

Temporal refinement studies are performed using the three IF-HERK schemes, Scheme A–C, included in Section 3.2. For each scheme, a series of eight test cases is performed using $\Delta t/\Delta t_0 = 2^0, 2^{-1}, \dots, 2^{-7}$. All test cases employ the same computational grid, and $\epsilon_{\mathbf{u}}$ and $\epsilon_{\mathbf{p}}$ are evaluated at $T = 10\Delta t_0$. Consequently, $T_{\mathcal{F}}$ and $T_{\mathcal{C}}$ are taken to be identity operators. The

¹⁹The solution error, as defined in Section 5, should not be confused with the error of the solution.

²⁰The computational cost of the spatial convergence tests are reduced by using “fat” vortex rings such as those given by Eq. (54), which, unlike similar “fat” rings given by Eq. (53), are continuous and differentiable at the origin.

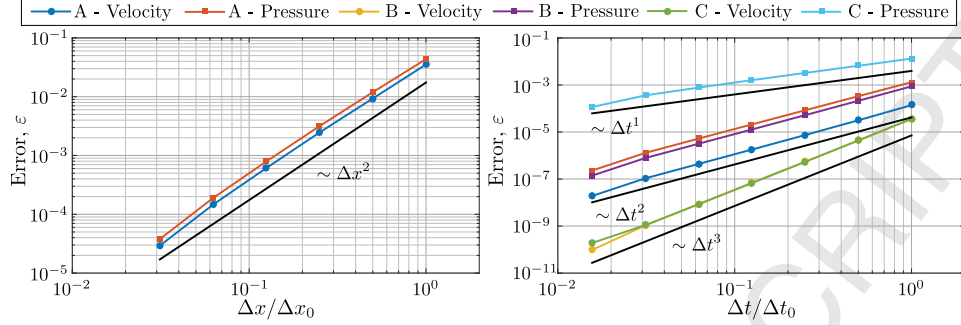


Figure 3: Velocity error, ε_u , and pressure error, ε_p , for test cases. Spatial refinement study verifies second-order accuracy of the spatial discretization technique (*left*). Temporal refinement studies verify the expected order of accuracy of the three time integration schemes defined in Section 3.2 (*right*).

slopes of the error curves depicted in the right plot of Figure 3 verify that the accuracy with respect to Δt of each scheme is the same as the order of accuracy expected from the IF-HERK order-conditions.²¹

6.2. Quality metrics for thin vortex rings

In this section we consider the laminar evolution of a thin vortex ring at $\text{Re}_0 = 7,500$ initiated with $\delta_0/R_0 = 0.2$. Six test cases for different values of Δx and Δt are performed. The ratio $\Delta t/\Delta x = 0.5734R_0/\Gamma_0$ is held constant across all test cases. Unlike the numerical experiments of Section 6.1, the grid is allowed to freely adapt as the solution evolves. For all test cases, ϵ^* is taken to be 10^{-6} , which is significantly smaller than the discretization error inferred from the discussion of Section 6.1.

The evolution of isolated vortex rings is often characterized by the time-history of a few fundamental volume integrals. Quantities considered in the following numerical experiments include the hydrodynamic impulse \mathcal{I} , the kinetic energy \mathcal{K} , enstrophy \mathcal{E} , the helicity \mathcal{J} , the Saffman-centroid \mathcal{X} , and the ring-velocity \mathcal{U} . Expressions for these quantities for unbounded fluid domains and exponentially decaying ω fields are given by [21]:

$$\begin{aligned} \mathcal{I}(t) &= \frac{1}{2} \int_{\mathbb{R}^3} \mathbf{x} \times \omega \, d\mathbf{x}, & \mathcal{J}(t) &= \int_{\mathbb{R}^3} \mathbf{u} \cdot \omega \, d\mathbf{x}, \\ \mathcal{K}(t) &= \int_{\mathbb{R}^3} \mathbf{u} \cdot (\mathbf{x} \times \omega) \, d\mathbf{x}, & \mathcal{X}(t) &= \frac{1}{2} \int_{\mathbb{R}^3} \frac{(\mathbf{x} \times \omega) \cdot \mathcal{I}}{|\mathcal{I}|^2} \mathbf{x} \, d\mathbf{x} - \int_0^t \mathbf{u}_\infty(t') \, dt' \\ \mathcal{E}(t) &= \frac{1}{2} \int_{\mathbb{R}^3} |\omega|^2 \, d\mathbf{x}, & \mathcal{U}(t) &= \frac{d\mathcal{X}}{dt}. \end{aligned} \quad (55)$$

The hydrodynamic impulse, \mathcal{I} , is a conserved quantity in the absence of non-conservative forces [21]. As a result, \mathcal{I} provides a useful metric for assessing the accuracy and physical fidelity of

²¹We note that the spatial discretization error associated with the computational grid is significantly larger than the temporal discretization error for some test cases. This does not affect the present refinement studies since the spatial discretization error is the same for all test cases and our error estimates are computed as the difference of two numerical solutions.

numerical solutions. The time rate of change of \mathcal{K} is related to \mathcal{E} by the relationship $\frac{d}{dt}\mathcal{K} = -2\nu\mathcal{E}$. Differences in the time history of $\frac{d}{dt}\mathcal{K}$ between different numerical simulations of the same flow are commonly used to characterize the accuracy of solutions of unsteady flows [44–46]. In the absence of viscosity, the helicity, \mathcal{J} , is an invariant of the flow and provides a measure for the degree of linkage of the vortex lines of the flow [47]. Although the present simulations consider viscous flows, differences in \mathcal{J} between test cases of the same flow are used as part of our quality metrics. Our definitions for the vortex ring centroid, \mathcal{X} , and propagation velocity, \mathcal{U} , are equivalent to those used by Saffman [21, 48]. Although all the integrals of Eq. (55) are formally over \mathbb{R}^3 , they can be accurately computed for solutions obtained by the NS-LGF method since the support of the integrands is approximately contained in D_{soln} .²²

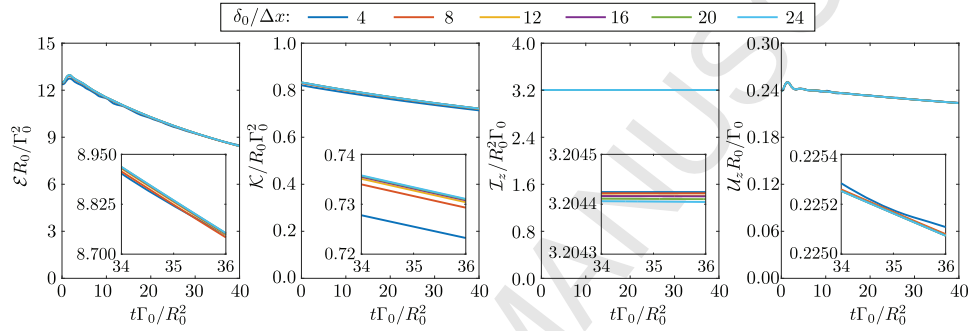


Figure 4: Time histories of \mathcal{E} , \mathcal{K} , \mathcal{I}_z , and \mathcal{U}_z (respectively, left to right) for a vortex ring at $\text{Re}_0 = 7,500$ initiated with $\delta_0/R_0 = 0.2$. Numerical experiments are performed using different values of $\delta_0/\Delta x$ while holding $\Delta t/\Delta x$ constant.

Table 3: Maximum difference in \mathcal{E} , \mathcal{K} , \mathcal{I}_z , and \mathcal{U}_z during $t\Gamma_0/R_0^2 \in [0, 40]$ between test cases with $\delta_0/\Delta x < 24$ and the test case with $\delta_0/\Delta x = 24$. Reported differences have been normalized by the maximum value of the respective quantity during $t\Gamma_0/R_0^2 \in [0, 40]$.

$\delta_0/\Delta x$	\mathcal{E}	\mathcal{K}	\mathcal{I}_z	\mathcal{U}_z
4	1.8×10^{-2}	1.5×10^{-2}	7.5×10^{-6}	4.9×10^{-3}
8	4.0×10^{-3}	3.5×10^{-3}	6.6×10^{-6}	4.8×10^{-4}
12	1.5×10^{-3}	1.3×10^{-3}	4.8×10^{-6}	1.7×10^{-4}
16	6.0×10^{-4}	5.3×10^{-4}	4.4×10^{-6}	7.3×10^{-5}
20	2.0×10^{-4}	2.2×10^{-4}	2.3×10^{-6}	2.7×10^{-5}

The time history for the values of \mathcal{E} , \mathcal{K} , \mathcal{I}_z , and \mathcal{U}_z , where subscripts “ q ” denotes the component of a vector quantity in q -th direction, are shown in Figure 4. The values for \mathcal{J} and the components of \mathcal{I} and \mathcal{U} in the x - and y -directions were also computed, but are not depicted since the magnitude of these values remained less than 10^{-8} , which is significantly smaller than ϵ^* , for all test cases. Visual inspection of the curves included in Figure 4 suggests good agreement between all tests cases. This is quantified by Table 3, which lists the maximum difference between test cases with $\delta_0/\Delta x < 24$ and the test case with $\delta_0/\Delta x = 24$.

²²Numerical solutions set the vorticity outside the computational to be zero. As a result, the only error involved in evaluating the integrals of Eq. (55) is the error resulting from their discretization.

Figure 4 demonstrates that \mathcal{E} , \mathcal{K} , and \mathcal{U}_z are most sensitive to changes in the resolution at early times, $t\Gamma_0/R_0^2 \in [0, 15]$. We attribute this to the rapid changes in the vorticity distribution observed shortly after the ring is initiated. For cases initiated with finite values of δ/R , it is well-known that flow undergoes an “equilibration” phase shortly after being initiated [44, 45, 49].²³ During this phase, vorticity starts to be shed into the wake and, over time, the core region of the ring assumes a more relaxed axisymmetric vorticity distribution in which ω_θ is no longer symmetric, but instead skewed so as to concentrate the vorticity away from the ring center. After the equilibration phase, i.e. approximately after $t\Gamma_0/R_0^2 > 15$ for test cases under consideration, the ring assumes a quasi-steady distribution that persists until the growth of linear instabilities causes the ring to transition into turbulence. This transition does not occur during the simulation time of the present study, but will be investigated in Section 6.4.

For each test case, the value of \mathcal{I} remained nearly constant throughout the simulation time, only exhibiting deviations on the same order as ϵ^* (taken to be 10^{-6} for all test cases). Interestingly, the value \mathcal{I} appears to be insensitive to changes in Δx , at least when maintaining $\Delta t/\Delta x$ constant, as demonstrated by Table 3. We refrain from speculating on whether the present method results in additional conservation properties beyond those mentioned in Section 2.1, since such investigations are beyond the scope of the present work. Instead, we simply note that \mathcal{I} appears to be conserved approximately up to the solution error, i.e. ϵ^* , which further verifies the physical fidelity of solutions obtained using the NS-LGF method.

The difference between the LHS and RHS of $\frac{d}{dt}\mathcal{K} = -2\nu\mathcal{E}$ is often used as a metric for the spatial discretization error. The maximum value of $|\frac{d}{dt}\mathcal{K} - (-2\nu\mathcal{E})| / (2\nu\mathcal{E})$ for $t\Gamma_0/R_0^2 \in [0, 40]$ is 6.8×10^{-2} , 2.1×10^{-2} , 9.6×10^{-3} , 5.3×10^{-3} , 3.4×10^{-3} , and 2.3×10^{-3} for the tests cases considered, sorted in ascending order of $\delta_0/\Delta x$. Values for $\frac{d\mathcal{K}}{dt}$ and $2\nu\mathcal{E}$ were computed at each half-time step using standard second-order differencing and averaging, respectively.

6.3. Propagation speed of thin vortex rings

The results of this section verify that the solutions obtained using the NS-LGF method are indeed physical solutions to the incompressible Navier-Stokes equations. The translational speed of laminar vortex rings has been extensively studied through experimental, numerical, and theoretical investigations [21, 44, 50–52]. Saffman [48] showed that the propagation speed of a viscous vortex ring with a vorticity distributions given by Eq. (53), in the limit of $\delta/R \rightarrow 0$, is

$$U_{\text{Saffman}} = \frac{\Gamma_0}{4\pi R_0} \left[\log\left(\frac{8}{\varepsilon}\right) - \beta_0 + O(\varepsilon \log \varepsilon) \right], \quad (56)$$

where $\varepsilon = \delta/R$, $\beta_0 = \frac{1}{2}(1 - \gamma + \log 2) \simeq 0.557966$, and $\gamma \simeq 0.577216$ is Euler’s constant. Subsequent numerical [44] and theoretical [53] investigations have shown that the error term is actually smaller, and is given by $O(\varepsilon^2 \log \varepsilon)$.

The initial propagation speed of a vortex ring, taken to be \mathcal{U}_z as defined in Eq. (55), is computed for test cases at $\text{Re}_0 = 7,500$ that have been initiated with $\varepsilon = 0.2, 0.1, 0.05, 0.025$, and 0.0125 . For all test cases, $\delta_0/\Delta x = 20$, $\Delta t\Gamma_0/R_0^2 = 10^{-6}$ and $\epsilon^* = 10^{-6}$. Values of \mathcal{U}_z are computed via central differencing the values of \mathcal{X} between adjacent time-steps. The value \mathcal{U}_z at $t^* = \Delta t/2$ for each test case is shown in the left plot of Figure 5. Visual inspection indicates good

²³A vortex ring initiated with a vorticity distribution given by Eq. (53) is a solution to the Navier-Stokes equations only in the limit of $\delta/R \rightarrow 0$.

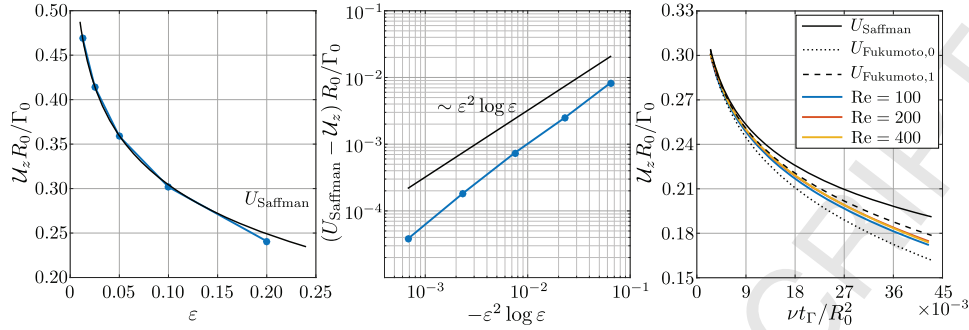


Figure 5: Propagation speed of a thin vortex ring at $Re_0 = 7,500$ for the different values of $\epsilon = \delta_0/R_0$ (left). Difference between the computed value, U_z , and the theoretical estimate, $U_{Saffman}$, for the propagation speed of a vortex ring at $Re_0 = 7,500$ (middle). Time history of the propagation speed of a vortex ring initiated with $\delta_0/R_0 = 0.1$ at different Re (right).

agreement between U_z and $U_{Saffman}$, which in turn verifies that numerical solutions obtained by the NS-LGF method approximate actual physical solutions.

We further verify the present formulation by confirming the form of the error term of $U_{Saffman}$, i.e. $O(\epsilon^2 \log \epsilon)$. Theoretical estimates for the effective ring and core radii for early times²⁴ indicate that, at time t^* , the ring and core size have not deviated enough from their initial values to significantly affect the value $U_{Saffman}$ as to hinder the present comparison. The middle plot of Figure 5 shows the difference in the ring propagation speed between the numerical experiments, U_z , and theoretical estimates, $U_{Saffman}$. For large values of ϵ , i.e. $\epsilon > 0.05$, the rate of change of $\Delta \tilde{U}_z = (U_{Saffman} - U_z) R_0 / \Gamma_0$ with respect to ϵ is consistent with the theoretical $O(\epsilon^2 \log \epsilon)$ error estimate. On the other hand, for $\epsilon < 0.05$ the rate of change of $\Delta \tilde{U}_z$ with respect to ϵ is slightly faster than $O(\epsilon^2 \log \epsilon)$. We refrain from attributing any physical meaning to the difference in the behavior of the error at smaller values of ϵ since we have not thoroughly determined the numerical error for such test cases.²⁵

We further verify the present implementation by comparing the time and Reynolds number dependence of U_z with previously reported theoretical [52] and numerical [44] results. To facilitate the comparisons, it is convenient to define

$$t_\Gamma = \frac{\delta_0^2}{4\nu} + t. \quad (57)$$

The discussion of [52] provides theoretical bounds on U_z based on the low and high Re limits of

²⁴The radius of the core and the vorticity centroid in the radial direction are approximately $2\sqrt{vt}$ and $R_0 + 3vt/R_0$ at $\sqrt{vt} \ll R_0$ [52].

²⁵Extrapolating from the results of Table 3 to the present tests cases, we estimate that the error of U_z to be between 10^{-5} and 10^{-4} . As a result, the assumption that U_z is more accurate than $U_{Saffman}$ might need to be revisited for test cases resulting in values of $\Delta \tilde{U}_z < 10^{-4}$.

a vortex ring initiated with $\delta/R \rightarrow 0$,

$$U_{\text{Fukumoto},0} = \frac{\Gamma_0}{4\pi R_0} \left[\log \left(\frac{4R_0}{\sqrt{\nu t_\Gamma}} \right) - \beta_0 - \frac{9}{5} \left(\log \left(\frac{4R_0}{\sqrt{\nu t_\Gamma}} \right) - \beta_1 \right) \frac{\nu t_\Gamma}{R_0^2} \right] \text{(low-Re),} \quad (58a)$$

$$U_{\text{Fukumoto},1} = \frac{\Gamma_0}{4\pi R_0} \left[\log \left(\frac{4R_0}{\sqrt{\nu t_\Gamma}} \right) - \beta_0 - \beta_2 \frac{\nu t_\Gamma}{R_0^2} \right] \text{(high-Re),} \quad (58b)$$

where β_0 is the same as in Eq. (56), $\beta_1 \approx 1.057967$, and $\beta_2 \approx 3.671591$. For all test cases, $\delta_0/\Delta x = 15$ and Δt is determined by requiring the initial CFL to be 0.5. Test cases correspond to a vortex ring at $\text{Re}_0 = 100, 200$, and 400 that are initiated with $\delta_0/R_0 = 0.1$. The right plot of Figure 5 demonstrates that, for all test cases, \mathcal{U}_z remains bounded between $U_{\text{Fukumoto},0}$ and $U_{\text{Fukumoto},1}$, except at early times for the case of $\text{Re}_0 = 400$ where the numerical \mathcal{U}_z slightly exceeds the $U_{\text{Fukumoto},1}$. This discrepancy is not surprising since the theory of Fukumoto [52] assumes that the vortex ring is initiated with $\delta/R \rightarrow 0$, and, as a result, does not properly account for the changes in the vorticity distribution that occur during the equilibration phase of a vortex ring initiated with a finite δ/R value. Although not shown in Figure 5, the time history of \mathcal{U}_z for all test cases has been compared to the numerical results of [44], and found to be in good agreement (overlying the curves of both investigations reveal nearly identical results).

6.4. Finite active computational domain error

In this section, we investigate the effect that our adaptive grid technique has on the numerical solutions by considering the evolution of a thin vortex ring computed using different values of ϵ^* . These test cases are used to verify that the solutions converge as ϵ^* tends to zero and to verify, via comparisons with numerical investigations of other authors, the physical fidelity of the solutions.

For all test cases, the vortex ring is initiated with $\delta_0/R_0 = 0.2$ and a constant uniform flow, $\mathbf{u}_\infty = [0, 0, u_\infty^{(z)}]$, is superimposed to partially oppose the translational motion of the vortex ring. The value of $u_\infty^{(z)} R_0/\Gamma_0$ is taken to be -0.18686 , which reduces the initial speed of the vortex ring by approximately 75%. Solutions are computed using $\delta_0/\Delta x = 10$ and $\Delta t \Gamma_0/R_0^2 \approx 0.01721$. The error estimates of Section 6.2 indicate that, for all test cases, the discretization error is on the order of 10^{-3} .

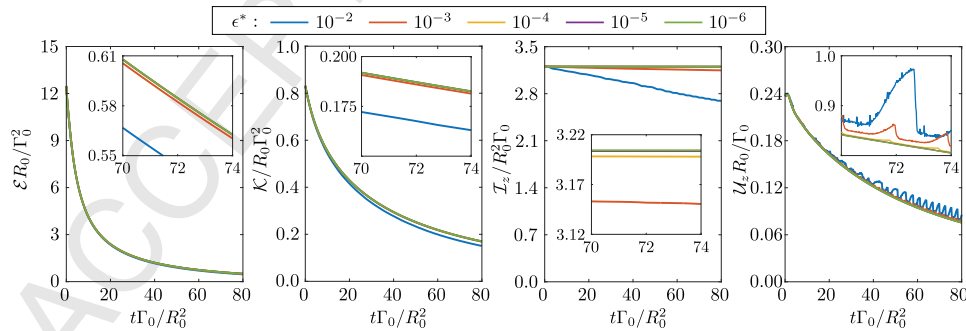


Figure 6: Time histories of \mathcal{E} , \mathcal{K} , \mathcal{I}_z , and \mathcal{U}_z (respectively, left to right) for a vortex ring at $\text{Re}_0 = 500$ initiated with $\delta_0/R_0 = 0.2$. All parameters, with the exception of ϵ^* , are held constant across all test cases.

Figure 6 depicts the time histories of \mathcal{E} , \mathcal{K} , \mathcal{I}_z , and \mathcal{U}_z for a vortex ring at $\text{Re} = 500$ computed using $\epsilon^* = 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}$, and 10^{-6} . The smooth decay of \mathcal{E} and \mathcal{K} indicates that the vortex ring remains laminar throughout the entire simulation time. This follows from the fact that a pronounced peak in \mathcal{E} is observed during the transition to the early stages of turbulence resulting from a significant increase in the stretching of vortex filaments [45]. Figure 6 verifies that, for laminar flows, numerical solutions converge as ϵ^* tends to zero. For all test cases with values of $\epsilon^* > 10^{-2}$, the error²⁶ in the computed values \mathcal{E} , \mathcal{K} and \mathcal{I}_z is inversely proportional to ϵ^* for $t\Gamma_0/R_0^2 \in [10, 80]$. The large oscillations in \mathcal{U}_z are due to shifts in \mathcal{X} resulting from the addition or removal of a single layer blocks in the z -direction. For times at which all test cases exhibit an approximate local minimum in \mathcal{U}_z , e.g. $t\Gamma_0/R_0^2 \approx 70.5$, the error in \mathcal{U}_z is also inversely proportional to ϵ^* .

Next, we consider the effect ϵ^* has on solutions of unsteady flows that are sensitive to small perturbations. The numerical investigations of [45, 54] on thin vortex rings with Gaussian vorticity distributions at $\text{Re}_0 = 7,500$ have shown that small sinusoidal perturbations to the vortex ring centerline result in the growth of azimuthal instabilities, which in turn facilitate the laminar to turbulent transition of the flow. Here, we consider the evolution of a vortex ring at $\text{Re}_0 = 7,500$ computed using values of $\epsilon^* = 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}$, and 10^{-6} . Unlike the numerical experiments of [45, 54], the vortex ring is initiated without imposing any perturbations beyond those implied by the numerical scheme.

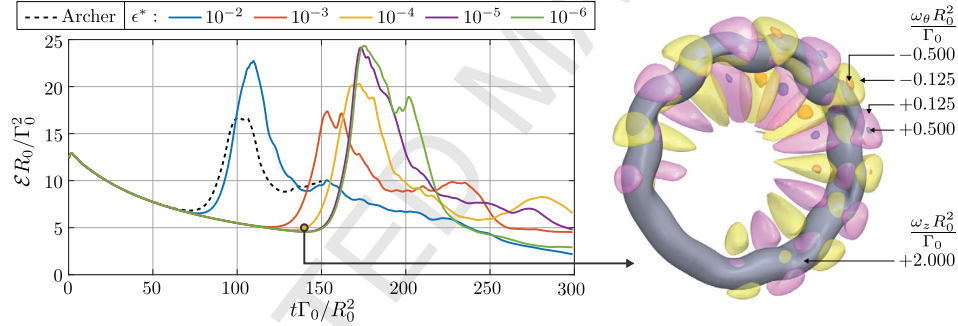


Figure 7: Time history of \mathcal{E} for a vortex ring at $\text{Re}_0 = 7,500$ initiated with $\delta_0/R_0 = 0.2$ (left). Data points labeled as “Archer” correspond values reported in Archer et al. [45]. All parameters, with the exception of ϵ^* , are held constant across all test cases. Vorticity iso-surfaces at $t\Gamma_0/R_0^2 = 137.6$ for test case $\epsilon^* = 10^{-4}$ (right).

The time history of \mathcal{E} for all test cases is shown in the left plot of Figure 7. The transition into the early stages of turbulence, characterized by a peak in \mathcal{E} resulting from an increase in the stretching of vortex filaments, is observed for all test cases. The growth of azimuthal instabilities and the development of secondary or “halo” vortices occurring at beginning of the transition phase [45, 54] are depicted in the right plot of Figure 7.

As expected from the previous test cases for $\text{Re}_0 = 500$, the values of \mathcal{E} during the laminar regime for all test cases converge as ϵ^* tends zero. Also included in Figure 7 are the values of \mathcal{E} reported in the numerical investigations of Archer et al. [45] for same vortex ring, which are

²⁶ The error is estimated by assuming that the test case corresponding to $\epsilon^* = 10^{-6}$ is the true solution.

nearly identical to values obtained from our test cases during the laminar regime.²⁷ Additionally, the vorticity iso-surfaces shown in right plot of Figure 7 are qualitatively similar to the vorticity iso-surfaces provided by Archer et al. [45] depicting the nonlinear growth of instabilities. In particular, the iso-surfaces of both investigations demonstrate the noticeable presence of the $n = 1$ azimuthal Fourier mode and the presence of halo vortices (iso-surfaces of ω_z in Figure 7) of similar magnitudes but alternating sign wedged between the approximately sinusoidally displaced inner-core (iso-surfaces of ω_θ in Figure 7).

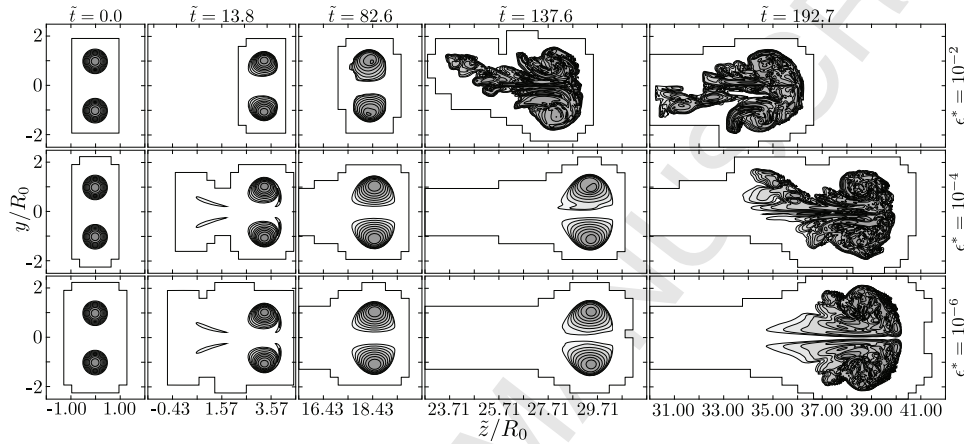


Figure 8: Vorticity magnitude on the y - z plane at $x = 0$ for test cases of $\epsilon^* = 10^{-2}$, 10^{-4} , and 10^{-6} at different times, $\tilde{t} = t\Gamma_0/R_0^2$. Contours correspond to values of $|\omega|R_0^2/\Gamma_0 = 4 \times \left(\frac{1}{2}\right)^i$ for $i = 8, 7, \dots, 0$. Contours have been shifted the z -direction to account for the constant freestream velocity, $\tilde{z} = z - u_\infty^{(z)}t$. Thick lines depict the boundary of D_{xsoln} .

The time histories of \mathcal{E} shown in Figure 7 indicate that the time at which \mathcal{E} starts to increase prior to reaching its peak value, i.e. the time at which the flow starts to transition, increases as ϵ^* decreases, but converges as ϵ^* tend to zero. This trend is an expected consequence of the present adaptive grid technique since the flow field is slightly perturbed each time a block is removed, i.e. vorticity is implicitly set to zero outside D_{supp} . The magnitude of these perturbations is correlated to the value of ϵ^* used to compute the numerical solution. Over time, the perturbations introduced by the adaptive grid lead to changes in the flow field that break the axial symmetry of the solution, which in turn promotes the growth of instabilities. Figure 8 provides vorticity contours at different times that depict the breakdown of axial symmetry and the subsequent laminar to turbulent transition for a few test cases.

Figure 8 also depicts the computational domains that result from using different values of ϵ^* . As expected, higher values of ϵ^* result in tighter domains, but lead to some significant changes in the flow that are potentially relevant to specific applications. For example, Figure 8 indicates that using a value ϵ^* of 10^{-2} is sufficient to accurately track the laminar evolution of the vortex

²⁷In the discussion of Archer et al. [45], the test case corresponding to a vortex ring at $\text{Re}_0 = 7,500$ initiated with $\delta_0/R_0 = 0.2$ is denoted as case “B3”. Unlike the present test cases, the initial vorticity distribution for case B3 of Archer et al. [45] was slightly perturbed to promote an early transition.

core, but does not adequately capture the large wake that develops behind the vortex ring.²⁸ We recall that the computational domain is determined by the particular choice of W_{supp} and ϵ_{supp} , both of which can be readily modified to accurately and efficiently capture the relevant physics of specific applications.

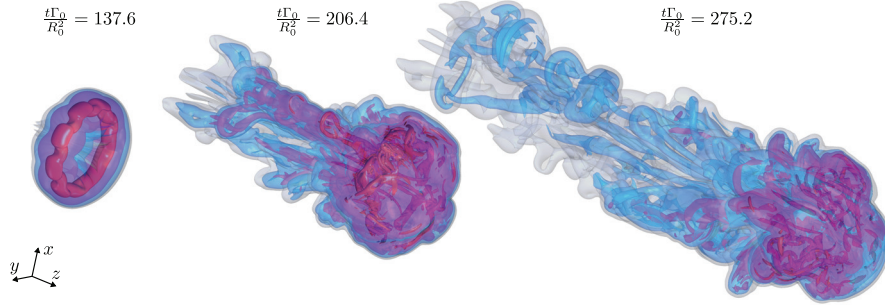


Figure 9: Translucent iso-surfaces of the vorticity magnitude for the test case of $\epsilon^* = 10^{-4}$ at different times. Iso-surfaces correspond to values of $|\omega|R_0^2/\Gamma_0 = 0.03125, 0.125, 0.5$, and 2 .

Figure 9 depicts vorticity iso-surfaces during the transition phase ($t\Gamma_0/R_0^2 = 137.6$) and early turbulent regime ($t\Gamma_0/R_0^2 = 206.4$ and 275.2) for the test case of $\epsilon^* = 10^{-4}$. At $t\Gamma_0/R_0^2 = 206.4$ and 275.2 , the presence of multiple thin vortex filaments and the absence of a coherent core indicate that the vortex ring is in its early turbulent regime [45, 54]. A comparison of the vorticity iso-surfaces at $t\Gamma_0/R_0^2 = 206.4$ and at $t\Gamma_0/R_0^2 = 275.2$ demonstrates that interwoven vorticity filaments near the core region are gradually pushed into the wake. As some of these structures are convected into the wake, they form hairpin vortices which persist for some time in the wake region. The periodic shedding of hairpin vortices into the wake is consistent with the numerical investigations of [45, 54], which in turn further verifies the physical fidelity of our solutions.

7. Conclusions

We have reported on a new fast, parallel solver for 3D, viscous, incompressible flows on unbounded domains based on LGFs. In this method, the incompressible Navier-Stokes equations are formally discretized on an unbounded staggered Cartesian grid using a second-order finite-volume scheme. This discretization technique has the advantage of enforcing discrete conservation laws and producing discrete operators with mimetic and commutativity properties that facilitate the implementation of fast, robust solvers. The system of DAEs resulting from the spatial-discretization of the momentum equation and the incompressibility constraint are integrated in time by using an integrating factor technique for the viscous terms and a HERK scheme for the convective term and the incompressibility constraint. Computationally efficient expressions for the integrating factors are obtained via Fourier analysis on unbounded Cartesian grids. A projection method that takes advantage of the mimetic and commutativity properties of the

²⁸The maximum length, in terms of R_0 , of the computational in the z -direction for is approximately 10, 26, 34, 46, 46 for test case with ϵ^* equal to 10^{-2} , 10^{-3} , 10^{-4} , 10^{-5} , and 10^{-6} , respectively.

discrete operators is used to efficiently solve the linear system of equations arising at each stage of the time integration scheme. This projection technique has the advantage of being equivalent to the LU decomposition of the system of equations, and, as a result, does not introduce any splitting-error and does not change the stability of the discretized equations.

In our formulation, solutions to the discrete Poisson problems and integration factor that are required to advance the flow are obtained through LGF techniques. These techniques express the solutions to inhomogeneous difference equations as the discrete convolution between source terms and the fundamental solutions of the discrete operators on unbounded regular grids. Fast, parallel solutions to the expressions resulting from the application of LGF techniques to discrete Poisson problems and integrating factors are obtained using the FMM for LGFs of [7].

As a result of our LGF formulation, the flow is solved using only information contained in the grid region where the vorticity and the divergence of the Lamb vector have non-negligible values. An adaptive block-structured grid and a velocity refresh technique are used to limit operations to a small finite computational domain. In order to efficiently compute solutions to a prescribed tolerance, weight functions and threshold values are used to determine the behavior of the adaptive grid.

The order of accuracy of the discretization and solution techniques is verified through refinement studies. The physical fidelity of the method is demonstrated in comparisons between computed and theoretical values for the propagation speed of a thin vortex ring. Additionally, results for the evolution of a thin vortex ring at $Re_0 = 7,500$ from the laminar to the early turbulent regime are shown to be in good agreement with investigations of other authors.

Acknowledgments

This work was partially supported by the United States Air Force Office of Scientific Research (FA950-09-1-0189) and the Caltech Field Laboratory for Optimized Wind Energy with Prof. John Dabiri as PI under the support of the Gordon and Betty Moore Foundation.

Appendix A. Discrete operators

In this appendix we provide point-operator and Fourier representations for the discrete operators of the present formulation. For operators that map onto $\mathbb{R}^{\mathcal{F}}$ or $\mathbb{R}^{\mathcal{E}}$, expressions for only one component of the resulting vector fields are provided since expressions for the other components are readily deduced. In the following discussion $\mathbf{c} \in \mathbb{R}^{\mathcal{C}}$, $\mathbf{f} \in \mathbb{R}^{\mathcal{F}}$, $\mathbf{e} \in \mathcal{E}$, and $\mathbf{v} \in \mathbb{R}^{\mathcal{V}}$ are arbitrary grid functions.

Point-operator representation based on the indexing convention depicted in Figure 1 are as follows:

- *Discrete gradient operators:* $\mathbf{G} : \mathbb{R}^{\mathcal{C}} \mapsto \mathbb{R}^{\mathcal{F}}$ and $\bar{\mathbf{G}} = -\mathbf{D}^\dagger : \mathbb{R}^{\mathcal{V}} \mapsto \mathbb{R}^{\mathcal{E}}$, where

$$\Delta x[\mathbf{G}\mathbf{c}]_{i,j,k}^{(1)} = \mathbf{c}_{i+1,j,k} - \mathbf{c}_{i,j,k}, \quad (\text{A.1a})$$

$$\Delta x[\bar{\mathbf{G}}\mathbf{v}]_{i,j,k}^{(1)} = \mathbf{v}_{i,j,k}^{(1)} - \mathbf{v}_{i-1,j,k}^{(1)}. \quad (\text{A.1b})$$

- *Discrete curl operators:* $\mathbf{C} : \mathbb{R}^{\mathcal{F}} \mapsto \mathbb{R}^{\mathcal{E}}$ and $\bar{\mathbf{C}} = \mathbf{C}^\dagger : \mathbb{R}^{\mathcal{E}} \mapsto \mathbb{R}^{\mathcal{F}}$, where

$$\Delta x[\mathbf{C}\mathbf{f}]_{i,j,k}^{(1)} = \mathbf{f}_{i,j,k}^{(2)} - \mathbf{f}_{i,j,k+1}^{(2)} + \mathbf{f}_{i,j+1,k}^{(3)} - \mathbf{f}_{i,j,k}^{(3)}, \quad (\text{A.2a})$$

$$\Delta x[\bar{\mathbf{C}}\mathbf{e}]_{i,j,k}^{(1)} = \mathbf{e}_{i,j,k-1}^{(2)} - \mathbf{e}_{i,j,k}^{(2)} + \mathbf{e}_{i,j,k}^{(3)} - \mathbf{e}_{i,j-1,k}^{(3)}. \quad (\text{A.2b})$$

- *Discrete divergence operators:* $\mathbf{D} : \mathbb{R}^{\mathcal{E}} \mapsto \mathbb{R}^{\mathcal{V}}$ and $\bar{\mathbf{D}} = -\mathbf{G}^\dagger : \mathbb{R}^{\mathcal{F}} \mapsto \mathbb{R}^{\mathcal{C}}$, where

$$\Delta x[\mathbf{De}]_{i,j,k} = \mathbf{e}_{i+1,j,k}^{(1)} + \mathbf{e}_{i,j+1,k}^{(2)} + \mathbf{e}_{i,j,k+1}^{(3)} - \sum_{q=1}^3 \mathbf{e}_{i,j,k}^{(q)}, \quad (\text{A.3a})$$

$$\Delta x[\bar{\mathbf{D}}\mathbf{f}]_{i,j,k} = -\mathbf{f}_{i-1,j,k}^{(1)} - \mathbf{f}_{i,j-1,k}^{(2)} - \mathbf{f}_{i,j,k-1}^{(3)} + \sum_{q=1}^3 \mathbf{f}_{i,j,k}^{(q)}. \quad (\text{A.3b})$$

- *Discrete Laplace operators:* $\mathbf{L}_Q : \mathbb{R}^Q \mapsto \mathbb{R}^Q$ for all Q in $\{\mathcal{C}, \mathcal{F}, \mathcal{E}, \mathcal{V}\}$, where

$$\mathbf{L}_\mathcal{C} = -\mathbf{G}^\dagger \mathbf{G}, \quad \mathbf{L}_\mathcal{V} = -\mathbf{D} \mathbf{D}^\dagger, \quad \mathbf{L}_\mathcal{F} = -\mathbf{G} \mathbf{G}^\dagger - \mathbf{C}^\dagger \mathbf{C}, \quad \mathbf{L}_\mathcal{E} = -\mathbf{D}^\dagger \mathbf{D} - \mathbf{C} \mathbf{C}^\dagger, \quad (\text{A.4})$$

and $[\mathbf{L}_\mathcal{C} \mathbf{c}]$, $[\mathbf{L}_\mathcal{V} \mathbf{v}]$, $[\mathbf{L}_\mathcal{F} \mathbf{f}]^{(\ell)}$, and $[\mathbf{L}_\mathcal{E} \mathbf{e}]^{(\ell)}$ can be computed as

$$(\Delta x)^2[\mathbf{La}]_{i,j,k} = -6\mathbf{a}_{i,j,k} + \sum_{q \in \{-1,1\}} (\mathbf{a}_{i+q,j,k} + \mathbf{a}_{i,j+q,k} + \mathbf{a}_{i,j,k+q}). \quad (\text{A.5})$$

- *Discrete nonlinear operator:* $\tilde{\mathbf{N}} : \mathbb{R}^{\mathcal{F}} \mapsto \mathbb{R}^{\mathcal{F}}$, where

$$[\tilde{\mathbf{N}}(\mathbf{f})]_{i,j,k}^{(1)} = \frac{1}{4} \sum_{q \in \{-1,0\}} [\mathbf{e}_{i,j,k+q}^{(2)} (\mathbf{f}_{i,j,k+q}^{(3)} + \mathbf{f}_{i+1,j,k+q}^{(3)}) - \mathbf{e}_{i,j+q,k}^{(3)} (\mathbf{f}_{i,j+q,k}^{(2)} + \mathbf{f}_{i+1,j+q,k}^{(2)})], \quad (\text{A.6})$$

and $\mathbf{e} = \mathbf{C}\mathbf{f}$.²⁹

- *Linearized discrete nonlinear operator:* the linearized form of $\tilde{\mathbf{N}}(\mathbf{f})$ about a constant uniform base flow, $\mathbf{f}_{\text{base}}(\mathbf{n}, t) = \mathbf{f}_{\text{base}}$, is given by $\mathbf{M}\mathbf{f}' = [\mathbf{K}(\mathbf{f}_{\text{base}})]\mathbf{C}\mathbf{f}'$, where $\mathbf{f}' = \mathbf{f} - \mathbf{f}_{\text{base}}$ and

$$\mathbf{K}(\mathbf{f}_{\text{base}}) : \mathbb{R}^{\mathcal{E}} \mapsto \mathbb{R}^{\mathcal{F}}, \quad [[\mathbf{K}(\mathbf{f}_{\text{base}})]\mathbf{e}']_{i,j,k}^{(1)} = \frac{1}{2} \sum_{q \in \{-1,0\}} (\mathbf{f}_{\text{base}}^{(3)} \mathbf{e}_{i,j,k+q}^{(2)} - \mathbf{f}_{\text{base}}^{(2)} \mathbf{e}_{i,j+q,k}^{(3)}). \quad (\text{A.7})$$

Discussions regarding the properties of discrete operators are often facilitated by using a block vector/matrix notation to describe the grid functions and linear operators. Consider the grid spaces \mathcal{X} and \mathcal{Y} corresponding to either \mathcal{F} or \mathcal{E} . Using block vector notation, a vector-valued grid function $\mathbf{x} \in \mathbb{R}^{\mathcal{X}}$ is expressed as

$$\mathbf{x} = \mathbb{S}_\mathcal{X}[\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, \bar{\mathbf{x}}_3]^\dagger, \quad (\text{A.8})$$

where the q -th block, $\bar{\mathbf{x}}_q$, corresponds to the values of the q -th component of \mathbf{x} . Each \mathbf{x}_q is a scalar real-valued grid function defined on an infinite Cartesian reference grid, which we denote by \mathbb{R}^Λ .³⁰ The shift operator $\mathbb{S}_\mathcal{X} : \mathbb{R}^\Lambda \mapsto \mathbb{R}^{\mathcal{X}}$ is used to transfer, or “shift”, the values of grid functions defined on \mathbb{R}^Λ to $\mathbb{R}^{\mathcal{X}}$ such that $[\mathbf{x}]^{(q)}(\mathbf{n}) = \bar{\mathbf{x}}_q(\mathbf{n})$. Similarly, the transpose of $\mathbb{S}_\mathcal{X}$, denoted by $\mathbb{S}_\mathcal{X}^\dagger$, transfers values of grid functions defined on $\mathbb{R}^{\mathcal{X}}$ to \mathbb{R}^Λ . The block vector notation and shift

²⁹The discrete nonlinear operator presented here is based on the discretization of the convective term in its rotational form, i.e. $\boldsymbol{\omega} \times \mathbf{u} - \frac{1}{2} \nabla(\mathbf{u} \cdot \mathbf{u})$, following the technique described in Zhang et al. [23]. As discussed in Section 2.1, $\tilde{\mathbf{N}}(\mathbf{f})$ is an approximation of $(\nabla \times \mathbf{f}) \times \mathbf{f}$.

³⁰Grid functions in \mathbb{R}^Λ can also be regarded as functions mapping \mathbb{Z}^3 to \mathbb{R} .

operators readily extend to the case of linear operators. Using block matrix notation, a discrete linear operator $\mathbf{T} : \mathbb{R}^X \mapsto \mathbb{R}^Y$ is expressed as

$$\mathbf{T} = \mathbb{S}_Y[\bar{\mathbf{T}}_{i,j}] \mathbb{S}_X^\dagger, \quad i, j = 1, 2, 3, \quad (\text{A.9})$$

where $\bar{\mathbf{T}}_{i,j} : \mathbb{R}^\Lambda \mapsto \mathbb{R}^\Lambda$.

We now turn our attention to the Fourier representations of grid functions and discrete linear operators. Consider the Fourier series, \mathfrak{F} , and the inverse Fourier transform, \mathfrak{F}^{-1} , given by:

$$[\mathfrak{F}\bar{\mathbf{u}}](\boldsymbol{\xi}) = \sum_{\mathbf{m} \in \mathbb{Z}^3} e^{i\mathbf{m} \cdot \boldsymbol{\xi}} \bar{\mathbf{u}}, \quad [\mathfrak{F}^{-1}\hat{\mathbf{u}}](\mathbf{m}) = \frac{1}{(2\pi)^3} \int_{\boldsymbol{\xi} \in \Pi} e^{-i\boldsymbol{\xi} \cdot \mathbf{m}} \hat{\mathbf{u}}(\boldsymbol{\xi}) d\boldsymbol{\xi}, \quad (\text{A.10})$$

respectively, where $\Pi = (-\pi, \pi)^3$, $\mathbf{u} : \mathbb{Z}^3 \mapsto \mathbb{R}$, and $\hat{\mathbf{u}} : \Pi \mapsto \mathbb{C}$. Using block matrix notation, we extend \mathfrak{F} and \mathfrak{F}^{-1} to the case of grid functions in \mathbb{R}^X by defining:

$$\mathfrak{F}_X = \text{diag}(\mathfrak{F}, \mathfrak{F}, \mathfrak{F}) \mathbb{S}_X, \quad \mathfrak{F}_X^{-1} = \mathbb{S}_X^\dagger \text{diag}(\mathfrak{F}, \mathfrak{F}, \mathfrak{F}). \quad (\text{A.11})$$

Next, let Ξ denote the set of all linear operators $\bar{\mathbf{Q}} : \mathbb{R}^\Lambda \mapsto \mathbb{R}^\Lambda$ such that the action of $\bar{\mathbf{Q}}$ on an arbitrary grid function $\bar{\mathbf{u}} \in \mathbb{R}^\Lambda$ is given by

$$[\bar{\mathbf{Q}}\bar{\mathbf{u}}](\mathbf{n}) = [\bar{\mathbf{K}}_Q * \bar{\mathbf{u}}](\mathbf{n}) = \sum_{\mathbf{m} \in \mathbb{Z}^3} \bar{\mathbf{K}}_Q(\mathbf{m} - \mathbf{n}) \bar{\mathbf{u}}(\mathbf{m}), \quad (\text{A.12})$$

where $\bar{\mathbf{K}}_Q : \mathbb{Z}^3 \mapsto \mathbb{R}$ is a well-behaved discrete kernel function. Any operator belonging to Ξ is diagonalized using \mathfrak{F} and \mathfrak{F}^{-1} ,

$$[\bar{\mathbf{Q}}\bar{\mathbf{u}}](\mathbf{n}) = [\bar{\mathbf{K}}_Q * \bar{\mathbf{u}}](\mathbf{n}) = [\mathfrak{F}^{-1}(\hat{\mathbf{K}}_Q \hat{\mathbf{u}})](\mathbf{n}), \quad (\text{A.13})$$

where $\hat{\mathbf{K}}_Q = \mathfrak{F}\bar{\mathbf{K}}_Q$ and $\hat{\mathbf{u}} = \mathfrak{F}\bar{\mathbf{u}}$. The block operators of all linear operators used in the present method belong to Ξ .

Appendix B. Lattice Green's functions representations

The NS-LGF method uses the LGFs \mathbf{G}_L and $\mathbf{G}_E(\alpha)$ to computed the action of \mathbf{L}_Q^{-1} and $\mathbf{E}_Q(\alpha)$, respectively. Fourier and Bessel integrals for \mathbf{G}_L and \mathbf{G}_E are given by

$$(\Delta x)^2 \mathbf{G}_L(\mathbf{n}) = \frac{1}{8\pi^3} \int_{\Pi} \frac{\exp(-i\mathbf{n} \cdot \boldsymbol{\xi})}{\sigma(\boldsymbol{\xi})} d\boldsymbol{\xi} = - \int_0^\infty e^{-6t} I_{n_1}(2t) I_{n_2}(2t) I_{n_3}(2t) dt \quad (\text{B.1a})$$

$$[\mathbf{G}_E(\alpha)](\mathbf{n}) = \frac{1}{8\pi^3} \int_{\Pi} \exp(-i\mathbf{n} \cdot \boldsymbol{\xi} - \sigma(\boldsymbol{\xi})) d\boldsymbol{\xi} = e^{-6\alpha} I_{n_1}(2\alpha) I_{n_2}(2\alpha) I_{n_3}(2\alpha) \quad (\text{B.1b})$$

where $\sigma(\boldsymbol{\xi}) = 2 \cos(\xi_1) + 2 \cos(\xi_2) + 2 \cos(\xi_3) - 6$, $\Pi = (-\pi, \pi)^3$, and $I_n(z)$ is the modified Bessel function of the first kind of order n .

Insights into the approximate behavior of $\mathbf{G}_L(\mathbf{n})$ can be obtained by considering the case of $|\mathbf{n}| \rightarrow \infty$. Asymptotic expansions in terms of unique rational functions for $\mathbf{G}_L(\mathbf{n})$ are provided in [33]. For example,

$$(\Delta x)^2 \mathbf{G}_L(\mathbf{n}) = -\frac{1}{4\pi|\mathbf{n}|} - \frac{n_1^4 + n_2^4 + n_3^4 - 3n_1^2 n_2^2 - 3n_1^2 n_3^2 - 3n_2^2 n_3^2}{16\pi|\mathbf{n}|^7} + O(|\mathbf{n}|^{-5}), \quad (\text{B.2})$$

as $|\mathbf{n}| \rightarrow \infty$. As expected, the leading order term corresponds to the fundamental solution of the Laplace operator.

Numerical procedures for efficiently evaluating $G_L(\mathbf{n})$ are provided in [7]. Values for $[G_E(\alpha)](\mathbf{n})$ can be readily computed using its Bessel form given by Eq. (B.1b). Although computing values of $G_L(\mathbf{n})$ and $[G_E(\alpha)](\mathbf{n})$ can potentially require a non-trivial number of operations, the LGF-FMM, used to compute the action of L_Q^{-1} and $E_Q(\alpha)$, employs pre-processing techniques that limit the evaluation of point-wise values of LGFs to once per simulation.

Appendix C. Stability analysis

Consider the linearization of Eq. (20) with respect to \mathbf{v} about a uniform, constant base flow, $\mathbf{v}_{\text{base}}(\mathbf{n}, t) = \tilde{\mathbf{u}}$, for the case of $u_\infty = 0$,

$$\frac{d\mathbf{v}'}{dt} = [\mathbf{K}(\tilde{\mathbf{u}})]\mathbf{C}\mathbf{v}' + \mathbf{G}\mathbf{b}', \quad \mathbf{G}^\dagger \mathbf{v}' = 0, \quad (\text{C.1})$$

where $\mathbf{v} = \mathbf{v}_{\text{base}} + \mathbf{v}'$ and $\mathbf{K}(\tilde{\mathbf{u}})$ is defined by Eq. (A.7).³¹ The stability analysis of Eq. (C.1) is facilitated by using a null-space approach to transform the original DAE index 2 system to an equivalent ODE,

$$\frac{d\mathbf{q}}{dt} = \mathbf{C}^\dagger [\mathbf{K}(\mathbf{v}_{\text{base}})]\mathbf{q} \quad (\text{C.2})$$

where $\mathbf{q} = \mathbf{C}\mathbf{v}'$, $\mathbf{v}' = \mathbf{C}^\dagger \mathbf{s}$, and $L_E \mathbf{s} = \mathbf{q}$ with $\mathbf{s} \rightarrow 0$ as $|\mathbf{n}| \rightarrow 0$. The details regarding the feasibility and equivalence of this transformation will be discussed in Section 4.1. It is readily verified that the discrete equations corresponding to the HERK method for Eq. (C.1) and for Eq. (C.2) are also equivalent; hence, Eq. (C.1) and Eq. (C.2) have the same stability region.

The ODE given by Eq. (C.2) is diagonalized by the component-wise Fourier series \mathfrak{F}_E , defined by Eq. (A.11),

$$\frac{d\hat{q}_k}{dt} = \frac{|\tilde{\mathbf{u}}|\Delta t}{\Delta x} \sigma(\xi) \hat{q}_k \quad \forall i = 1, 2, 3, \quad (\text{C.3a})$$

$$\sigma(\xi) = -i \sum_{j=1}^3 \frac{\tilde{u}_j}{|\tilde{\mathbf{u}}|} \sin \xi_j, \quad (\text{C.3b})$$

where $\xi \in \Pi = (-\pi, \pi)$.³² It follows from Eq. (C.3b) that $\Re(\sigma(\xi)) = 0$ and $|\Im(\sigma(\xi))| \leq \sqrt{3}$ for all $\xi \in \Pi$. As a result, the linear stability Eq. (20) is determined by the stability of the scalar ODEs:

$$\frac{dy}{dt} = i\mu y \quad \forall \mu \in (-\gamma, \gamma), \quad \gamma = \sqrt{3} \frac{|\tilde{\mathbf{u}}|\Delta t}{\Delta x}. \quad (\text{C.4})$$

Consider integrating the ODE given by Eq. (C.4) using the HERK method. In the absence of algebraic constraints, an HERK scheme reduces to a standard ERK scheme with the same tableau. Consequently, the region of absolute stability for the ODE of Eq. (C.4) is given by

$$\Omega = \{\mu \in \mathbb{R} : |R(i\mu)| < 1\}, \quad R(z) = 1 + z\mathbf{b}^\dagger (\mathbf{I} - z\mathbf{A})^{-1} \mathbf{e}, \quad (\text{C.5})$$

³¹It is not necessary to linearize the integrating factors present in Eq. (20), since they can be commuted and made to cancel out after the linearization of $\tilde{\mathbf{N}}$.

³²In order to simplify the expression for $\sigma(\xi)$ to the form given by Eq. (C.3a) it is necessary to account for $D\mathbf{q} = 0$.

where \mathbf{b} and \mathbf{A} are defined by Eq. (22), and $\mathbf{e} = [1, 1, \dots, 1]$ [34]. Eq. (C.5) implies that the IF-HERK method is linearly stable if the following CFL condition is satisfied:

$$\text{CFL} = \frac{|\tilde{\mathbf{u}}|\Delta t}{\Delta x} < \text{CFL}_{\max}, \quad \text{CFL}_{\max} = \frac{\mu^*}{\gamma} \quad (\text{C.6})$$

where $\mu^* = \sup(\Omega)$ depends on the RK coefficients of the scheme. For all the IF-HERK schemes defined in Eq. (33), the value of CFL_{\max} is unity.

Appendix D. Error estimates for integrating factors operating on truncated source fields

In this appendix we provide estimates for the difference between $E_Q(\alpha)$ and $M_Q^\gamma E_Q(\alpha) M_Q^\gamma$ inside D_γ , which are pertinent to the discussion of Section 4.4. Consider the constant uniform scalar field $u \in \mathbb{R}^Q$ and the domain D_γ , where D_γ is infinite in the x - and y -directions and semi-infinite in the z -direction. For this simplified case, it is sufficient to consider the 1D problem of computing

$$y = [E'(\alpha) - M'E'(\alpha)M']u = [I - M'E'(\alpha)M']u, \quad (\text{D.1})$$

where I is the identity operator,

$$E'(\alpha)u = G'_E(\alpha) * u, \quad G'_E(n) = e^{-2\alpha} I_n(2\alpha), \quad (\text{D.2})$$

and

$$[M'u](k) = \begin{cases} u(k) & \text{if } k > 0 \\ 0 & \text{otherwise} \end{cases}. \quad (\text{D.3})$$

As a result, the magnitude of the normalized difference, d , at $k > 0$ is given by

$$d(k) = \frac{y(k)}{|u|} = \sum_{j=0}^{\infty} e^{-2\alpha} I_{k-j+1}(2\alpha), \quad (\text{D.4})$$

where $|u|$ is the magnitude of the uniform field u . Numerical approximations for $d(k)$ are obtained by truncating the infinite sum of Eq. (D.4) to a finite number of terms, N , such that $I_{k-N+1}(2\alpha)/I_{k+1}(2\alpha)$ is less than a prescribed value.³³

As discussed in Section 4.4, the current implementation of the NS-LGF method uses Eq. (D.4) to estimate the error associated with approximating $E_Q(\alpha)u$ by $M_Q^{\text{xsoln}} E_Q(\alpha) M_Q^{\text{xsoln}} u$, where u is the velocity perturbation field. For this case, $|u|$ in Eq. (D.4) is set to be the maximum value of any component of u in D_{soln} . Numerical experiments of flows similar to those considered in Section 6 demonstrate that this technique leads to fairly conservative error estimates; in all experiments the actual error was less than 10% of the estimated error. Tighter error bounds that account for the domain shape and the distribution of u can potentially be obtained, but are not explored in the present work.

³³For a fixed $z > 0$, $I_n(z)$ decreases as n increases. For a fixed $z > 0$, $I_n(z)$ decays faster than any exponential as $n \rightarrow \infty$.

References

- [1] S. V. Tsynkov, Numerical solution of problems on unbounded domains. A review, *Applied Numerical Mathematics* 27 (1998) 465–532.
- [2] T. Colonius, Modeling artificial boundary conditions for compressible flow, *Annual Review of Fluid Mechanics* 36 (2004) 315–345.
- [3] D. S. Pradeep, F. Hussain, Effects of boundary condition in numerical simulations of vortex dynamics, *Journal of Fluid Mechanics* 516 (2004) 115–124.
- [4] S. Dong, G. E. Karniadakis, C. Chrysosostomidis, A robust and accurate outflow boundary condition for incompressible flow simulations on severely-truncated unbounded domains, *Journal of Computational Physics* 261 (2014) 83–105.
- [5] A. Gillman, P. G. Martinsson, Fast and accurate numerical methods for solving elliptic difference equations defined on lattices, *Journal of Computational Physics* 229 (2010) 9026–9041.
- [6] A. Gillman, P. G. Martinsson, A fast solver for Poisson problems on infinite regular lattices, *Journal of Computational and Applied Mathematics* 258 (2014) 42–56.
- [7] S. Liska, T. Colonius, A parallel fast multipole method for elliptic difference equations, *Journal of Computational Physics* 278 (2014) 76–91.
- [8] A. Leonard, Vortex methods for flow simulation, *Journal of Computational Physics* 37 (1980) 289–335.
- [9] L. Greengard, V. Rokhlin, A fast algorithm for particle simulations, *Journal of Computational Physics* 73 (1987) 325–348.
- [10] G. S. Winckelmans, A. Leonard, Contributions to vortex particle methods for the computation of three-dimensional incompressible unsteady flows, *Journal of Computational Physics* 109 (1993) 247–273.
- [11] M. S. Warren, J. K. Salmon, A parallel hashed oct-tree n-body algorithm, in: *Proceedings of the 1993 ACM / IEEE conference on Supercomputing*, ACM, 1993, pp. 12–21.
- [12] H. Cheng, L. Greengard, V. Rokhlin, A fast adaptive multipole algorithm in three dimensions, *Journal of Computational Physics* 155 (1999) 468–498.
- [13] P. Ploumhans, G. S. Winckelmans, Vortex methods for high-resolution simulations of viscous flow past bluff bodies of general geometry, *Journal of Computational Physics* 165 (2000) 354–406.
- [14] G.-H. Cottet, P. D. Koumoutsakos, *Vortex methods: theory and practice*, Cambridge University Press, 2000.
- [15] L. Ying, G. Biros, D. Zorin, A kernel-independent adaptive fast multipole algorithm in two and three dimensions, *Journal of Computational Physics* 196 (2004) 591–626.
- [16] G. S. Winckelmans, *Vortex Methods*, John Wiley & Sons, Ltd, 2004.
- [17] R. Cogle, G. Winckelmans, G. Daeninck, Combining the vortex-in-cell and parallel fast multipole methods for efficient domain decomposition simulations, *Journal of Computational Physics* 227 (2008) 9091 – 9120.
- [18] P. Chatelain, P. Koumoutsakos, A Fourier-based elliptic solver for vortical flows with periodic and unbounded directions, *Journal of Computational Physics* 229 (2010) 2425–2431.
- [19] J. T. Rasmussen, G.-H. Cottet, J. H. Walther, A multiresolution remeshed vortex-in-cell algorithm using patches, *Journal of Computational Physics* 230 (2011) 6742–6755.
- [20] M. M. Hejlesen, J. T. Rasmussen, P. Chatelain, J. H. Walther, A high order solver for the unbounded Poisson equation, *Journal of Computational Physics* 252 (2013) 458–467.
- [21] P. G. Saffman, *Vortex dynamics*, Cambridge University Press, 1992.
- [22] R. A. Nicolaides, X. Wu, Covolume solutions of three-dimensional div-curl equations, *SIAM Journal on Numerical Analysis* 34 (1997) 2195–2203.
- [23] X. Zhang, D. Schmidt, B. Perot, Accuracy and conservation properties of a three-dimensional unstructured staggered mesh scheme for fluid dynamics, *Journal of Computational Physics* 175 (2002) 764–791.
- [24] F. H. Harlow, J. E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, *Physics of Fluids* 8 (1965) 2182–2189.
- [25] D. K. Lilly, On the computational stability of numerical solutions of time-dependent non-linear geophysical fluid dynamics problems, *Monthly Weather Review* 93 (1965).
- [26] Y. Morinishi, T. S. Lund, O. V. Vasilyev, P. Moin, Fully conservative higher order finite difference schemes for incompressible flow, *Journal of Computational Physics* 143 (1998) 90–124.
- [27] B. Perot, Discrete conservation properties of unstructured mesh schemes, *Annual Review of Fluid Mechanics* 43 (2011) 299–318.
- [28] R. A. Nicolaides, Direct discretization of planar div-curl problems, *SIAM Journal on Numerical Analysis* 29 (1992) 32–56.
- [29] B. Perot, Conservation properties of unstructured staggered mesh schemes, *Journal of Computational Physics* 159 (2000) 58–89.
- [30] W. H. McCrea, F. J. W. Whipple, Random paths in two and three dimensions, *Proceedings of the Royal Society of Edinburgh* 60 (1940) 281–298.

- [31] R. J. Duffin, E. P. Shelly, Difference equations of polyharmonic type, *Duke Mathematical Journal* 25 (1958) 209–238.
- [32] O. Buneman, Analytic inversion of the five-point Poisson operator, *Journal of Computational Physics* 8 (1971) 500–505.
- [33] P. G. Martinsson, G. J. Rodin, Asymptotic expansions of lattice Green’s functions, *Proceedings of the Royal Society of London Series A* 458 (2002) 2609–2622.
- [34] E. Hairer, G. Wanner, *Solving ordinary differential equations II: stiff and differential-algebraic problems*, Berlin Springer, 1996.
- [35] U. M. Ascher, L. R. Petzold, *Computer methods for ordinary differential equations and differential-algebraic equations*, volume 61, SIAM, 1998.
- [36] E. Hairer, C. Lubich, M. Roche, *The numerical solution of differential-algebraic systems by Runge-Kutta methods*, Lecture Notes in Mathematics, Springer-Verlag, 1989.
- [37] V. Brasey, E. Hairer, Half-explicit Runge-Kutta methods for differential-algebraic systems of index 2, *SIAM Journal on Numerical Analysis* 30 (1993) 538–552.
- [38] B. Sanderse, B. Koren, Accuracy analysis of explicit Runge-Kutta methods applied to the incompressible Navier-Stokes equations, *Journal of Computational Physics* 231 (2012) 3041–3063.
- [39] H. Le, P. Moin, An improvement of fractional step methods for the incompressible Navier-Stokes equations, *Journal of Computational Physics* 92 (1991) 369–379.
- [40] T. Colonius, K. Taira, A fast immersed boundary method using a nullspace approach and multi-domain far-field boundary conditions, *Computer Methods in Applied Mechanics and Engineering* 197 (2008) 2131–2146.
- [41] B. Perot, An analysis of the fractional step method, *Journal of Computational Physics* 108 (1993) 51–58.
- [42] S. Turek, *Efficient Solvers for Incompressible Flow Problems: An Algorithmic and Computational Approach*, volume 6, Springer Science & Business Media, 1999.
- [43] W. Chang, F. Giraldo, B. Perot, Analysis of an exact fractional step method, *Journal of Computational Physics* 180 (2002) 183–199.
- [44] S. K. Stanaway, B. J. Cantwell, P. R. Spalart, A numerical study of viscous vortex rings using a spectral method, *NASA Technical Memorandum* (1988) 101041.
- [45] P. J. Archer, T. G. Thomas, G. N. Coleman, Direct numerical simulation of vortex ring evolution from the laminar to the early turbulent regime, *Journal of Fluid Mechanics* 598 (2008) 201–226.
- [46] M. Cheng, J. Lou, T. T. Lim, Leapfrogging of multiple coaxial viscous vortex rings, *Physics of Fluids* 27 (2015) 031702.
- [47] H. K. Moffatt, A. Tsinober, Helicity in laminar and turbulent flow, *Annual Review of Fluid Mechanics* 24 (1992) 281–312.
- [48] P. G. Saffman, The velocity of viscous vortex rings, *Studies in Applied Mathematics* 49 (1970) 371–380.
- [49] K. Shariff, R. Verzicco, P. Orlandi, A numerical study of three-dimensional vortex ring instabilities: viscous corrections and early nonlinear stage, *Journal of Fluid Mechanics* 279 (1994) 351–375.
- [50] D. G. Akhmetov, *Vortex rings*, Springer Science & Business Media, 2009.
- [51] I. S. Sullivan, J. J. Niemela, R. E. Hershberger, D. Bolster, R. J. Donnelly, Dynamics of thin vortex rings, *Journal of Fluid Mechanics* 609 (2008) 319–347.
- [52] Y. Fukumoto, Global time evolution of viscous vortex rings, *Theoretical and Computational Fluid Dynamics* 24 (2010) 335–347.
- [53] Y. Fukumoto, H. K. Moffatt, Motion and expansion of a viscous vortex ring. Part 1. A higher-order asymptotic formula for the velocity, *Journal of Fluid Mechanics* 417 (2000) 1–45.
- [54] M. Bergdorf, P. Koumoutsakos, A. Leonard, Direct numerical simulations of vortex rings at $Re = 7500$, *Journal of Fluid Mechanics* 581 (2007) 495–505.