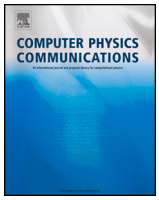
Computer Physics Communications 310 (2025) 109502

Contents lists available at ScienceDirect

Computer Physics Communications

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



Computational Physics

Efficient numerical methods for the Maxey-Riley-Gatignol equations with Basset history term ☆

Julio Urizarna-Carasa, Leon Schlegel, Daniel Ruprecht *

Lehrstuhl Computational Mathematics, Institut für Mathematik, Technische Universität Hamburg, Germany

ARTICLE INFO

The review of this paper was arranged by Prof. Andrew Hazel

Keywords:
Inertial particles
Maxey-Riley-Gatignol equations
Finite differences for unbounded domains
Bickley jet
Faraday flow

ABSTRACT

The Maxey-Riley-Gatignol equations (MRGE) describe the motion of a finite-sized, spherical particle in a fluid. Because of wake effects, the force acting on a particle depends on its past trajectory. This is modeled by an integral term in the MRGE, also called Basset force, that makes its numerical solution challenging and memory intensive. A recent approach proposed by Prasath et al. (2019) [9] exploits connections between the integral term and fractional derivatives to reformulate the MRGE as a time-dependent partial differential equation on a semi-infinite pseudo-space. They also propose a numerical algorithm based on polynomial expansions. This paper develops a numerical approach based on finite difference instead, by adopting techniques by Koleva (2005) [35] and Fazio and Jannelli (2014) [37] to cope with the issues of having an unbounded spatial domain. We compare convergence order and computational efficiency for particles of varying size and density of the polynomial expansion by Prasath et al., our finite difference schemes and a direct integrator for the MRGE based on multi-step methods proposed by Daitche (2013) [29]. While all methods achieve their theoretical convergence order for neutrally buoyant particles with zero initial relative velocity, they suffer from various degrees of order reduction if the initial relative velocity is non-zero or the particle has a different density than the fluid.

1. Introduction

The Maxey-Riley-Gatignol equations (MRGE) are a second-order system of integro-differential equations with a singular kernel that model the motion of finite-sized (inertial), spherical particles in a fluid [1]. They are used for the study of a wide range of phenomena, for example the formation of clouds in the atmosphere [2,3], the settling of plankton in the ocean ("marine snow") [4,5], the transmission of Covid-19 virus through sprays [6] and more.

While the motion of inertial particles can be described by other means, for example the Navier-Stokes equations with specific boundary conditions [7,8], the MRGE provide a simpler and computationally much more efficient model which is valid for spheric particles under the assumption that while the flow influences the particle, the particles do not alter the fluid. This is an appropriate assumption when "particles are far smaller than all relevant flow length scales, and when the suspension of particles is dilute" [9]. The MRGE are useful for particles that are too large to be treated as passive tracers but not yet big enough to substantially disturb the fluid. Candelier et al. [10, Figure 3] show a good match between theoretical and experimental trajectories for particles in a rotating cylinder.

According to Langlois et al. [11], the search for a suitable model that ultimately resulted in the MRGE dates back to 1851, when Stokes attempted for the first time to obtain a model for the motion of a pendulum in a fluid [12]. Later, the combined works of Basset [13], Boussinesq [14] and Oseen [15] let to a model called the Basset-Boussinesq-Oseen (BBO) equation for the settling of a sphere under gravity in a quiescent fluid. In 1947, Tchen [16] modified the BBO into a model for the motion of a rigid sphere in a nonuniform unsteady flow. This model underwent several amendments until it attained the form that is now known as the Maxey-Riley-Gatignol equations [1,17]. Throughout this paper, we consider the MRGE in the form

☆ This project is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – SFB 1615 – 503850735.
* Corresponding author.
E-mail addresses: julio.urizarna@tuhh.de (J. Urizarna-Carasa), leon.schlegel@tuhh.de (L. Schlegel), ruprecht@tuhh.de (D. Ruprecht).

https://doi.org/10.1016/j.cpc.2025.109502
Received 20 March 2024; Received in revised form 6 January 2025; Accepted 7 January 2025

Available online 17 January 2025

$$\dot{\boldsymbol{y}}(t) = \boldsymbol{v}(t), \tag{1a}$$

$$R\dot{\boldsymbol{v}}(t) = \frac{D\boldsymbol{u}}{Dt} - \frac{1}{S}(\boldsymbol{v} - \boldsymbol{u}) - \sqrt{\frac{3}{\pi S}}\left\{ \frac{1}{\sqrt{t}}(\boldsymbol{v}(0) - \boldsymbol{u}(0)) + \int_0^t \frac{(\dot{\boldsymbol{v}}(s) - \dot{\boldsymbol{u}}(s))}{\sqrt{t-s}} ds \right\}, \tag{1b}$$

given by Prasath et al. [9] which neglects the Faxen corrections by assuming that the particle radius is much smaller than the typical length scale of the flow [18]. Here, $\boldsymbol{u}(\boldsymbol{y}(t), t)$ is the fluid velocity at the particle position $\boldsymbol{y}(t)$ and $\boldsymbol{v}(t)$ is the particle velocity. The parameters are

$$\beta = \frac{\rho_p}{\rho_f}, \quad R = \frac{1+2\beta}{3}, \quad S = \frac{a^2}{3T\nu}, \tag{2}$$

where ρ_p is the density of the particle, ρ_f the density of the fluid, a the particle's diameter, ν the fluid's kinematic viscosity and T the time scale associated with the flow.

Due to the integral term in (1b), also known as Basset history or memory term, the force acting on a particle depends on its past trajectory and the MRGE are not a dynamical system. Since this term causes numerical difficulties and, in straightforward numerical integration, requires storage of all previously computed steps, it is often neglected in applications [6], modified [19–22] or approximated [23–26]. However, both theoretical [9] and empirical [10,27,28] studies have shown that the Basset history term is relevant even for small Stokes numbers and that neglecting it can lead to noticeable inaccuracies in the modeled trajectories.

Therefore, a handful of numerical approaches were developed to solve the full MRGE efficiently. Based on linear multistep methods, Daitche proposed the first direct integration methods that do not rely on approximations of the kernels [29]. While computationally fast, the approach still requires storage of all time steps which can lead to problems with memory in long simulations. Michaelides [30] propose a transformed version of the MRGE based on the Laplace transform that removes the implicit dependency on the dependent variable in the history integral. While this avoids the need to use a nonlinear iterative solver and thus reduces computing time, it does not remove the history integral.

Tatom in 1988 realized that the history term is equivalent to a half derivative of Riemann-Liouville type [31]. Using this, Prasath et al. [9] show that solving the MRGE is equivalent to solving a diffusive heat equation posed on a semi-infinite pseudo-space with a nonlinear boundary condition. Their reformulation removes the integral term and memory effect and allows to use standard numerical techniques for partial differential equations to solve the MRGE. However, it requires dealing with an unbounded spatial computational domain. Prasath et al. [9] also propose a numerical approach based on the integral form of the solution obtained by using Fokas's method [32]. An open-source reimplementation in Python of their numerical approach based on polynomial expansions was recently published [33]. However, their reformulation also opens up possibilities to apply other numerical techniques for partial differential equations to the MRGE. Very recently, Jaganathan et al. [34] proposed another transformation of the Maxey-Riley-Gatignol equation by embedding it into an extended state-space. This removes the non-locality in time and results in a dynamical system that can be solved with standard explicit numerical integrators. While we did not reimplement their Fortran code in Python, a preliminary comparison of performance based on published results can be found in §3.3. Comparing their method against Prasath et al.'s [9] and our approach would be an interesting next step.

To illustrate the opportunities offered by Prasath et al.'s [9] reformulation, we propose a finite difference method as solver for the MRGE. By modifying techniques developed by Koleva [35], Alshina et al. [36] and Fazio and Janelli [37] for solving initial-boundary value problems on infinite domains, we introduce a second and fourth order finite difference discretization. To efficiently deal with the nonlinearity at the boundary, we use implicit-explicit Runge-Kutta methods of order two and four, which avoid the need for an iterative nonlinear solver in every stage. We also provide an open-source Python implementation [38] of our finite difference approach, Prasath et al.'s algorithm [9] and Daitche's method [29]. The code can be used to reproduce our results or to explore regimes and parameter configurations beyond what is reported in the paper. To our knowledge, these are the only available algorithms that solve the MRGE without approximations to the kernel. We perform a comprehensive comparison of all three methods with respect to accuracy and computational efficiency for five different flow fields and particles of different size and density.

We demonstrate that all three methods reach their full theoretical order of convergence only if the particle is neutrally buoyant and initially has zero relative velocity. For particles that are not neutrally buoyant, we observe order reduction for all methods. Even stronger order reduction is observed if the particle has non-zero initial relative velocity. It is noteworthy that all methods experience this, albeit to varying degrees, even though they rely on different discretizations of different formulations of the MRGE.

Contributions. The main novel contributions of this paper are (i) the application of techniques for finite differences on semi-infinite domains to the reformulated MRGE, (ii) the idea to use an implicit-explicit Runge-Kutta method to avoid having to use a nonlinear solver, (iii) a comparison of different numerical methods for the full MRGE with respect to computational efficiency and (iv) a demonstration that the numerical solution of the full MRGE including history term in real-time is possible.

Structure of the paper. Section 2 summarizes the transformation of the MRGE into a diffusion-type PDE on a semi-infinite domain and introduces the finite differences used to solve the resulting system. Section 3 first introduces the five used benchmark problems. It then studies convergence rates of the investigated methods for particles of different densities. Finally the section compares different methods with respect to computational efficiency by investigating the wallclock times required to reach a certain error. Section 4 summarizes the paper and draws conclusions.

2. Numerical solution of the reformulated Maxey-Riley equations

Subsection 2.1 briefly summarizes the reformulation of the MRGE with history term as a diffusive PDE on a semi-infinite domain [9]. Subsection 2.2 introduces a second-order and fourth-order finite difference scheme to discretize this PDE in a method-of-lines approach. Subsection 2.3 describes two types of Runge-Kutta time stepping methods, diagonally implicit (DIRK) and implicit-explicit (IMEX), to efficiently integrate the resulting semi-discrete problem in time.

2.1. Transforming the MRGE with Basset history

We simply state the resulting transformed system here. For a detailed explanation of the reformulation, we refer to the original paper by Prasath et al. [9] and our description of a reimplementation of their approach in an open-source Python software [33]. The reformulated problem reads

$$\boldsymbol{q}_t(x,t) = \boldsymbol{q}_{xx}(x,t), \quad x > 0,\ t \in (0,T], \tag{3a}$$

$$\boldsymbol{q}(x,0) = \boldsymbol{0}, \quad x > 0, \tag{3b}$$

$$\boldsymbol{q}_t(0,t) + \alpha\boldsymbol{q}(0,t) - \gamma\boldsymbol{q}_x(0,t) = \boldsymbol{f}(\boldsymbol{q}(0,t), \boldsymbol{y}(t), t), \quad t \in [0,T], \tag{3c}$$

$$\dot{\boldsymbol{y}}(t) = \boldsymbol{q}(0,t) + \boldsymbol{u}(\boldsymbol{y}(t), t), \quad t \in [0,T], \tag{3d}$$

$$\lim_{t\to 0} \boldsymbol{q}(0,t) = \boldsymbol{v}_0 - \boldsymbol{u}_0, \tag{3e}$$

$$\boldsymbol{y}(0) = \boldsymbol{y}_0. \tag{3f}$$

where x is a pseudo-space without physical meaning, t is time, $\boldsymbol{y}(t)$ is, as in (1), the particle's position at time t, and $\boldsymbol{q}(x,t)$ is a function with identical dimensionality as $\boldsymbol{y}(t)$, that is two for a two-dimensional flow field and three for a 3D field. The boundary value $\boldsymbol{q}(0,t)$ is equal to the relative velocity of the particle at time t and $\dot{\boldsymbol{y}}(t)$ is its absolute velocity. Furthermore, we have the right hand side function

$$\boldsymbol{f}\left(\boldsymbol{q}(0,t),\boldsymbol{y}(t),t\right) := \left(\frac{1}{R}-1\right)\frac{D\boldsymbol{u}}{Dt} - \boldsymbol{q}(0,t)\cdot\nabla_y \boldsymbol{u}(\boldsymbol{y}(t),t) \tag{4}$$

for the boundary condition and physical parameters

$$\alpha := \frac{1}{RS}, \quad \gamma := \frac{1}{R}\sqrt{\frac{3}{S}} \tag{5}$$

with R and S defined in (2). Solving (3) produces the solution $\boldsymbol{y}(t)$ of (1) with the advantage that the history term is not present.

2.2. Finite differences for the transformed MRGE

The transformed problem (3) is defined on a semi-infinite computational domain, which requires some care when dealing with the boundary condition at infinity. To discretize equation (3) in space, we propose two finite difference schemes: the second order scheme by Koleva [35] based on the work by Alshina et al. [36], and a novel fourth order approximation obtained by using compact finite differences for uniform grids [39] plus a new technique based on the core idea in Fazio and Jannelli [37] to map equidistant nodes to the semi-infinite domain.

We discretize the spatial domain with a quasi-uniform grid by defining a set of N uniform grid points $\xi_n := \frac{n}{N}$, $n \in \{0,1,\dots,N-1\}$ in the interval $[0,1)$. These are then mapped to $[0,\infty)$ via the logarithmic mapping

$$x_n := x(\xi_n) = -c\ln(1-\xi_n), \tag{6}$$

where c is a parameter that controls the distribution of nodes such that approximately half of the grid points are placed within the interval $[0,c]$. In contrast to the algebraic rule also proposed by Koleva [35] and Fazio [37], the logarithmic rule produces a higher density of nodes around $x_0 = 0$. Since our aim is to approximate the boundary value $\boldsymbol{q}(0,t)$, this is where we require the highest accuracy. We have, however, not performed a detailed comparison of the two mappings. Furthermore, while numerical experiments not documented here suggest that the value of c can have a substantial impact on the overall error, we could not establish a robust heuristic how to choose it. Therefore, we set $c = 20$ in all numerical experiments, which provided satisfactory results, and abstained from tuning. Significant further gains in accuracy might be possible using a more informed and may be even adaptive choice where c changes over time but studying this is left for future work.

Discretizing the spatial derivatives in (3a)-(3c) together with (3d) results in the semidisrete system

$$\underbrace{\begin{bmatrix}\dot{\boldsymbol{q}}(t)\\ \dot{\boldsymbol{y}}(t)\end{bmatrix}}_{=:\dot{\boldsymbol{\eta}}(t)} = \underbrace{\left[\begin{array}{ccccc|cc} & & A_s & & & 0 & 0\\ & & & & & \vdots & \vdots\\ & & & & & 0 & 0\\ \hline 1 & 0 & 0 & \dots & 0 & 0 & 0\\ 0 & 1 & 0 & \dots & 0 & 0 & 0\end{array}\right]}_{=:A} \underbrace{\begin{bmatrix}\boldsymbol{q}(t)\\ \boldsymbol{y}(t)\end{bmatrix}}_{=:\boldsymbol{\eta}(t)} + \underbrace{\begin{bmatrix}\boldsymbol{v}(\boldsymbol{q}_0(t),\boldsymbol{y}(t),t)\\ \boldsymbol{u}(\boldsymbol{y}(t),t)\end{bmatrix}}_{=:\boldsymbol{\omega}(\boldsymbol{q}_0(t),\boldsymbol{y}(t),t)} \tag{7}$$

where the specific forms of $A_s \in \mathbb{R}^{N\times N}$ and $\boldsymbol{v}(\boldsymbol{q}_0(t),\boldsymbol{y}(t),t) \in \mathbb{R}^N$ depend on whether the second or fourth order discretization described below is used. Furthermore,

$$\boldsymbol{q}(t) := \begin{bmatrix} q_0^{(1)}(t) & q_0^{(2)}(t) & q_1^{(1)}(t) & q_1^{(2)}(t) & \dots & q_{N-2}^{(1)}(t) & q_{N-2}^{(2)}(t)\end{bmatrix}^T, \tag{8}$$

is a vector with both the horizontal and vertical components of the relative velocity at each node and

$$\boldsymbol{q}_0(t) := \begin{bmatrix} q_0^{(1)}(t) & q_0^{(2)}(t)\end{bmatrix}^T. \tag{9}$$

Note that in a three-dimensional flow field we would have components $q_0^{(3)}(t)$, $q_1^{(3)}(t)$, etc. However, since we provide only numerical examples for 2D flow fields, we just show the two component case. Below, we describe a second-order and fourth-order accurate approach to construct A_s.

2.2.1. Second order spatial discretization

Koleva [35] provides the following derivative approximations for problems on one-dimensional semi-infinite spatial domains

$$\left.\frac{\partial q(\xi(x),t)}{\partial x}\right|_{x_{n+1/2}} \approx \frac{q_{n+1}(t)-q_n(t)}{2\zeta_n}, \tag{10}$$

$$\left.\frac{\partial^2 q(\xi(x),t)}{\partial x^2}\right|_{x_n} \approx \frac{1}{\psi_n}\left[\left.\frac{\partial q(\xi(x),t)}{\partial x}\right|_{x_{n+1/2}} - \left.\frac{\partial q(\xi(x),t)}{\partial x}\right|_{n-1/2}\right] \tag{11}$$

where q corresponds to one of the components of the vector $\boldsymbol{q}$ used in (3) and

$$\psi_n := x_{n+1/2} - x_{n-1/2}, \quad \zeta_n := x_{n+3/4} - x_{n+1/4}$$

for $n = 1,2,\dots,N-1$ are the mesh spacings in the pseudo-space coordinate. At the left boundary we use

$$\left.\frac{\partial q(\xi(x),t)}{\partial x}\right|_{x_0} \approx \frac{1}{2}\left[\left.\frac{\partial q(\xi(x),t)}{\partial x}\right|_{x_{1/2}} + \left.\frac{\partial q(\xi(x),t)}{\partial x}\right|_{x_{-1/2}}\right]. \tag{12}$$

The boundary condition (3c) is coupled to the PDE (3a) by using the ghost grid point at $-1/2$ in equations (12) and (11) for $n = 0$. With these approximations, the entries $(a_{i,j})_{1\le i\le 2N,\, 1\le j\le 2N}$ of A_s are

$$a_{11} = a_{22} = -\frac{\gamma + 2\alpha\zeta_0}{\zeta_0(2+\gamma\psi_0)}, \qquad a_{12} = a_{24} = \frac{\gamma}{\zeta_0(2+\gamma\psi_0)} \tag{13}$$

for the boundary values and

$$a_{i,i-2} = a_{i+1,i-1} = \frac{1}{2\psi_{\frac{i-1}{2}}\zeta_{\frac{i-3}{2}}}, \tag{14}$$

$$a_{i,i} = a_{i+1,i+1} = -\frac{1}{\psi_{\frac{i-1}{2}}}\left(\frac{\zeta_{\frac{i-3}{2}} + \zeta_{\frac{i-1}{2}}}{2\zeta_{\frac{i-3}{2}}\zeta_{\frac{i-1}{2}}}\right), \tag{15}$$

$$a_{i,i+2} = a_{i+1,i+3} = \frac{1}{2\psi_{\frac{i-1}{2}}\zeta_{\frac{i-1}{2}}}, \tag{16}$$

for $i = 2m+1$ with $m \in \{1,2,\dots,N-2\}$. All other matrix entries are zero so that A_s is a sparse pentadiagonal matrix with zero first upper and lower diagonals. The vector $\boldsymbol{v}(\boldsymbol{q}_0(t),\boldsymbol{y}(t),t)$, which contains the boundary condition function $\boldsymbol{f}(\boldsymbol{q}_0(t),\boldsymbol{y}(t),t)$, is given by

$$\boldsymbol{v}(\boldsymbol{q}_0(t),\boldsymbol{y}(t),t) := \begin{bmatrix} \frac{2}{2+\gamma\psi_0}\boldsymbol{f}^{(1)}(\boldsymbol{q}_0(t),\boldsymbol{y}(t),t)\\ \frac{2}{2+\gamma\psi_0}\boldsymbol{f}^{(2)}(\boldsymbol{q}_0(t),\boldsymbol{y}(t),t)\\ 0\\ \vdots\\ 0\end{bmatrix}. \tag{17}$$

2.2.2. Fourth order spatial discretization

As in Fazio's second order approximation [37], we start by using the chain rule for the derivative of $q(\xi(x),t)$ with respect to x at a node with index n to get

$$\left.\frac{\partial q(\xi(x),t)}{\partial x}\right|_{x_n} = \left.\frac{\partial q(\xi(x),t)}{\partial \xi}\right|_{x_n} \cdot \left.\frac{d\xi(x)}{dx}\right|_{x_n}. \tag{18}$$

Fazio approximates both derivatives on the right hand side of (18) by centered differences [37]. In contrast, we observe that $\left({}^{d\xi(x)}/_{dx}\right)_{x_n}$ can be obtained exactly from (6) which leads to simpler expressions when using higher order derivative approximations with wider stencils.

Since the logarithmic map (6) between grids is a bijection, we can obtain its inverse mapping

$$x(\xi) = -c\ln(1-\xi) \iff \xi(x) = 1 - e^{-x/c} \tag{19}$$

and calculate its derivative at integer points

$$\left.\frac{\partial \xi(x)}{\partial x}\right|_{x_n} = \frac{1}{c}e^{-x_n/c} = \frac{1}{c}\left(1-\xi_n\right) = \frac{1}{c}\left(1-\frac{n}{N}\right). \tag{20}$$

By plugging this into (18) we obtain

$$\left.\frac{\partial q(\xi(x),t)}{\partial x}\right|_{x_n} = \left.\frac{\partial q(\xi(x),t)}{\partial \xi}\right|_{x_n} \cdot \frac{1}{c}\left(1-\frac{n}{N}\right), \tag{21}$$

connecting derivatives on the infinite grid with derivatives on the equidistant, finite reference grid. Equation (21) allows to define finite difference approximations for the first order derivative in a bounded uniform mesh $(\xi_n)_{n=0,\dots,N-1}$ and then transform them to the mesh in the semi-infinite domain $(x_n)_{n=0,\dots,N-1}$.

To approximate the derivative $\left.\partial q(\xi(x),t)/\partial \xi\right|_{x_n}$ of q on the equidistant grid at the interior points, we use the fourth order compact finite difference by Lele [39, Eq. (2.1.6)] with $\alpha = \frac{1}{4}$, resulting in

$$\begin{aligned}&\left.\frac{\partial q(\xi(x),t)}{\partial \xi}\right|_{x_{n-1}} + \left.\frac{4\partial q(\xi(x),t)}{\partial \xi}\right|_{x_n} + \left.\frac{\partial q(\xi(x),t)}{\partial \xi}\right|_{x_{n+1}} \\ &\quad\approx \frac{3}{d}\left(q_{n+1}(t) - q_{n-1}(t)\right)\end{aligned} \tag{22}$$

with $d := \xi_{n+1} - \xi_n = \frac{1}{N}$ for the interior points with indices $n \in \{1,2,\dots,N-1\}$. Note that non-compact higher order finite differences would require additional ghost points at the left boundary which could be difficult to realize for the nonlinear boundary condition in (3).

We use (21) to replace the derivatives in uniform space by derivatives in pseudo-space so that (22) becomes

$$\begin{aligned}&\left(\frac{Nc}{N-n+1}\right)\left.\frac{\partial q(\xi(x),t)}{\partial x}\right|_{n-1} + \left(\frac{4Nc}{N-n}\right)\left.\frac{\partial q(\xi(x),t)}{\partial x}\right|_{n} + \\ &\qquad + \left(\frac{Nc}{N-n-1}\right)\left.\frac{\partial q(\xi(x),t)}{\partial x}\right|_{n+1} \\ &\qquad \approx \frac{3}{d}\left(q_{n+1}(t) - q_{n-1}(t)\right)\end{aligned} \tag{23}$$

Like in Section 2.2.1 we obtain the second derivative by applying the first order derivative twice. As a consequence, two different approximations for the point at the left boundary are required: one including the boundary condition obtained by the same procedure as described by Malele et al. [40] and another without the influence of the boundary condition obtained by using the fourth order upwinding scheme given by Lele [39].

The coefficients of the upwinding approximation at the boundary that includes the left boundary condition are calculated in Appendix A and result in the scheme

$$\begin{aligned}&c\left.\frac{\partial q(\xi(x),t)}{\partial x}\right|_{x_0} + \frac{Nc}{N-1}\left.\frac{\partial q(\xi(x),t)}{\partial x}\right|_{x_1} \\ &\quad\approx \left(\frac{12\alpha d c - 17\gamma}{18\gamma d}\right)q_0(t) + \frac{1}{2d}q_1(t) + \frac{1}{2d}q_2(t) - \frac{1}{18d}q_3(t) + \\ &\quad - \frac{2c}{3\gamma}\left(f(\boldsymbol{q}_0(t),\boldsymbol{y}(t),t) - \left.\frac{\partial q(\xi(x),t)}{\partial t}\right|_{x_0}\right).\end{aligned} \tag{24}$$

On the other hand, since the second upwinding scheme at $n=0$ does not include the boundary condition, the coefficients of the scheme on the uniform grid are taken directly from [39, Equation (4.1.4)], so that

$$\begin{aligned}&\left.\frac{\partial q(\xi(x),t)}{\partial \xi}\right|_{x_0} + 3\left.\frac{\partial q(\xi(x),t)}{\partial \xi}\right|_{x_1} \\ &\quad\approx \frac{1}{d}\left(-\frac{17}{6}q_0(t) + \frac{3}{2}q_1(t) + \frac{3}{2}q_2(t) - \frac{1}{6}q_3(t)\right).\end{aligned} \tag{25}$$

We again replace the derivatives in uniform space by derivatives in pseudo-space

$$\begin{aligned}&c\left.\frac{\partial q(\xi(x),t)}{\partial x}\right|_{x_0} + \left(\frac{3Nc}{N-1}\right)\left.\frac{\partial q(\xi(x),t)}{\partial x}\right|_{x_1} \\ &\quad\approx \frac{1}{d}\left(-\frac{17}{6}q_0(t) + \frac{3}{2}q_1(t) + \frac{3}{2}q_2(t) - \frac{1}{6}q_3(t)\right).\end{aligned} \tag{26}$$

Using equations (23), (24) and (26) we obtain the system

$$M_1\, D_x\boldsymbol{q}(t) \approx B_1\boldsymbol{q}(t) + K_1(\boldsymbol{q}_0(t),\boldsymbol{y}(t),t) + V_1(\dot{\boldsymbol{q}}_0(t)), \tag{27a}$$

$$M_2\, D_x^2\boldsymbol{q}(t) \approx B_2\, D_x\boldsymbol{q}(t), \tag{27b}$$

where M_1, M_2, B_1, B_2, K_1 and V_1 are matrices and vectors defined in Appendix B and $D_x\boldsymbol{q}(t)$ is the spatial discretization of $\frac{\partial}{\partial x}\boldsymbol{q}(\xi(x),t)$. Multiplying (27a) by M_1^{-1} yields

$$D_x\boldsymbol{q}(t) \approx M_1^{-1}\, B_1\boldsymbol{q}(t) + M_1^{-1}\, K_1(\boldsymbol{q}_0(t),\boldsymbol{y}(t),t) + M_1^{-1}\, V_1(\dot{\boldsymbol{q}}_0(t)), \tag{28a}$$

$$M_2\, D_x^2\boldsymbol{q}(t) \approx B_2\, D_x\boldsymbol{q}(t). \tag{28b}$$

Substituting the right hand side of (28a) into (28b) yields

$$M_2 D_x^2\boldsymbol{q}(t) \approx B_2\left(M_1^{-1}\, B_1\boldsymbol{q}(t) + M_1^{-1}\, K_1(\boldsymbol{q}_0(t),\boldsymbol{y}(t),t) + M_1^{-1}\, V_1(\dot{\boldsymbol{q}}_0(t))\right). \tag{29}$$

After some algebra we obtain

$$\begin{aligned}&M_2\, D_x^2\boldsymbol{q}(t) - B_2M_1^{-1}V_1(\dot{\boldsymbol{q}}_0(t)) \\ &\quad\approx B_2M_1^{-1}B_1\boldsymbol{q}(t) + B_2M_1^{-1}\, K_1(\boldsymbol{q}_0(t),\boldsymbol{y}(t),t).\end{aligned}$$

We now use the semidiscretized version $\dot{\boldsymbol{q}}(t) = D_x^2\boldsymbol{q}(t)$ of the differential equation (3a) to transform the matrix $V_1(\dot{\boldsymbol{q}}_0(t))$ so that

$$V_1(\dot{\boldsymbol{q}}_0(t)) = \frac{2c}{3\gamma}\begin{bmatrix}\dot{q}_0^{(1)}(t)\\ \dot{q}_0^{(2)}(t)\\ 0\\ 0\\ \vdots\\ 0\end{bmatrix} = \underbrace{\frac{2c}{3\gamma}\begin{bmatrix}1&0&0&\dots\\ 0&1&0&\ddots\\ 0&0&0&\ddots\\ \vdots&\ddots&\ddots&\ddots\end{bmatrix}}_{=:\mathbb{P}}\dot{\boldsymbol{q}}(t) = \frac{2c}{3\gamma}\,\mathbb{P}\,\dot{\boldsymbol{q}}(t). \tag{30}$$

We then obtain the final approximation

$$\dot{\boldsymbol{q}}(t) = A_s\boldsymbol{q}(t) + \boldsymbol{\nu}(\boldsymbol{q}_0(t),\boldsymbol{y}(t),t) \tag{31}$$

to be used in (7) where

$$A_s := \Psi^{-1}B_2M_1^{-1}B_1, \tag{32}$$

$$\boldsymbol{\nu}(\boldsymbol{q}_0(t),\boldsymbol{y}(t),t) := \Psi^{-1}B_2M_1^{-1}K_1(\boldsymbol{q}_0(t),\boldsymbol{y}(t),t), \tag{33}$$

$$\Psi := M_2 - \frac{2c}{3\gamma}B_2M_1^{-1}\mathbb{P}. \tag{34}$$

2.3. Time stepping methods for the semi-discrete system

The semi-discrete system (7) can be solved numerically by using a time stepping method. In this section we briefly state the different time stepping schemes we investigate: a second order implicit method (trapezoidal rule), a fourth order Diagonally Implicit Runge-Kutta (DIRK) method and Implicit-Explicit (IMEX) Runge-Kutta methods of order two and four.

2.3.1. Trapezoidal rule

The fully implicit Trapezoidal Rule applied to (7) yields

$$\underbrace{\left(\mathbb{I} - \frac{\Delta t}{2}A\right)}_{=:M_{\text{left}}}\boldsymbol{\eta}^{k+1} = \left(\mathbb{I} + \frac{\Delta t}{2}A\right)\boldsymbol{\eta}^k + \frac{\Delta t}{2}\left(\boldsymbol{\omega}^k + \boldsymbol{\omega}^{k+1}\right), \tag{35}$$

where $\boldsymbol{\eta}^k := \boldsymbol{\eta}(t^k)$ and $\boldsymbol{\omega}^k := \boldsymbol{\omega}(\boldsymbol{q}_0(t^k),\boldsymbol{y}(t^k),t^k)$. Due to the $\boldsymbol{\omega}^{k+1}$ term on the right hand side this is an implicit system and requires a nonlinear solver. We use the Newton-Krylov method LGMRES with M_{left} as

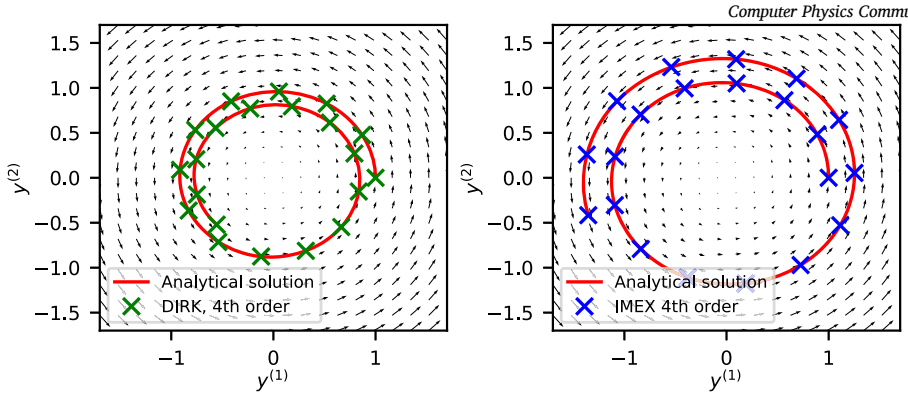


Fig. 1. Trajectories of two particles moving in the vortex flow field with initial position $\boldsymbol{y}(t_0) = (1, 0)^T$ and zero initial relative velocity with $S = 0.3$ and $R = 7/9$ (left) or $R = 4/3$ (right) for $t \in [0, 10]$.

preconditioner, implemented in the `newton_krylov` function of the `scipy.optimize` Python library [41–44]. We take the solution of the explicit system

$$\tilde{\boldsymbol{\eta}}^{k+1} = (\mathbb{I} + \Delta t\, A)\, \boldsymbol{\eta}^k + \Delta t\, \boldsymbol{\omega}^k \tag{36}$$

as starting value for the Newton method.

2.4. Diagonally Implicit Runge Kutta (DIRK) method

As implicit time integrator we use the fourth order DIRK method ESDIRK4(3)6L[2]SA [45]. Kennedy and Carpenter recommend this method as the "default method for solving stiff problems at moderate error tolerances". In ESDIRK methods, the first stage is explicit but all other stages are implicit and require a nonlinear solver. As for the trapezoidal rule, we use the Python function `newton_krylov`. The previous stage is used as starting value.

2.5. Implicit-Explicit Runge-Kutta (IMEX) methods

In the semi-discrete system (7), only the $\boldsymbol{\omega}$ term arising from the boundary condition is nonlinear. The A term is linear but stiff because of the discrete Laplacian. To avoid the overhead of having to use a fully nonlinear solver for every stage, we propose to use an implicit-explicit RKM instead. This allows to treat the boundary term explicitly, thus avoiding a nonlinear solver per stage, while treating the stiff discrete Laplacian implicitly, which avoids a very harsh time step restriction. We use the second-order IMEX midpoint rule by Ascher et al. [46] and the fourth order IMEX method combining the explicit ARK4(3)6L[2]SA–ERK and implicit ARK4(3)6L[2]SA–ESDIRK by Kennedy and Carpenter [45].

3. Numerical results

We compare accuracy and computational cost of (i) the finite difference (FD) schemes described in Section 2, (ii) the third-order direct numerical integrator proposed by Daitche [29] and (iii) our reimplementation of Prasath's polynomial expansion method [32] for the reformulated problem (3) by Prasath et al. [9]. Note that we rely on our own implementation of Prasath's methods since no publicly available code appears to exist [33]. Daitche [29] provided us with his code in private communication. We compared both our and his implementations, ensured that the coefficients computed in our version match those reported in the paper and finally reproduced figures 3 and 4 in his paper to confirm that both implementations deliver the same result. One difference, however, is that we only compute coefficients in double instead of quadruple precision, which in some cases leads to a reduction in convergence order for the 3rd order variant. This effect was already observed by Daitche [29] and Moreno-Casas and Bombardelli [22].

3.1. Benchmark problems

We consider three flow fields where analytical solutions to the MRGE are available: a steady vortex [10], a quiescent flow and an unsteady but spatially homogeneous oscillatory background [9]. Additionally, we consider two unsteady and inhomogeneous flow fields where no analytic solution is known, the Bickley jet proposed by Rypina et al. [47] with the parameters from Hadjighasem et al. [48] and an experimentally measured Faraday flow [49,50]. In the last two cases, we measure the error against a high resolution reference computed with Prasath et al.'s [9] algorithm. For each flow field, we investigate the accuracy of the methods when simulating trajectories of particles that are lighter ($R < 1$) or denser ($R > 1$) than the fluid or neutrally buoyant ($R = 1$) and cases with zero and non-zero initial relative velocity. Specifically, we use $R = 7/9$ where $\beta = 2/3$ and $R = 4/3$ where $\beta = 3/2$.

3.1.1. Quiescent, steady inhomogeneous and unsteady homogeneous flow field

Below, we describe in detail the three benchmark problems for which an analytical solution is available.

Vortex flow field. Fig. 1 shows the vortex flow field and two example trajectories for different values of R. The flow field

$$\boldsymbol{u} = \|\boldsymbol{y}(t)\|\, \omega\, \boldsymbol{e}_\theta = \omega \begin{bmatrix} -y^{(2)} \\ y^{(1)} \end{bmatrix}, \tag{37}$$

is a stationary vortex whose tangential velocity increases proportionally to the distance from the origin. We set the angular velocity of the vortex to $\omega = 1$. An analytical solution for this problem is given by Candelier et al. [10, Eq. (12)]. The markers in the figure indicate the numerical solution computed with a fully implicit fourth-order DIRK method (left) and a fourth-order implicit-explicit Runge-Kutta method (right). The analytical solutions are shown as red lines. For both the lighter and heavier particle, the solution agrees very well with the numerical approximation. Solutions computed with the other numerical methods are omitted to keep the plot readable but can be generated using the code provided with the paper. For the vortex flow field, an analytical solution is only available if the initial relative velocity of the particle is zero and thus we only consider this case.

Oscillatory flow field. Fig. 2 shows the analytical and numerical solutions for particles that are lighter (left) and denser (right) than the fluid in a spatially homogeneous oscillatory flow field

$$\boldsymbol{u}(t) = \begin{bmatrix} u^{(1)} \\ \sin(\lambda t) \end{bmatrix}, \tag{38}$$

where $u^{(1)} = 0.05$ is the horizontal component of the flow field and $\lambda = 6$ is the oscillation frequency. We show the numerical approximations provided by the trapezoidal rule (left) and the second-order IMEX method

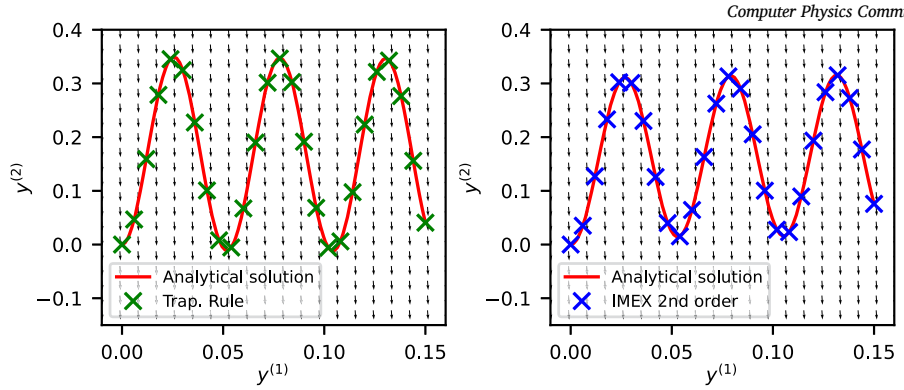


Fig. 2. Trajectories of two particles moving in the oscillatory flow field with initial position $\boldsymbol{y}(t_0) = (0,0)^T$ and zero initial relative velocity with $S = 0.3$ and $R = 7/9$ (left) or $R = 4/3$ (right) for $t \in [0,3]$. The arrows show the velocity field at $t = 3$. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

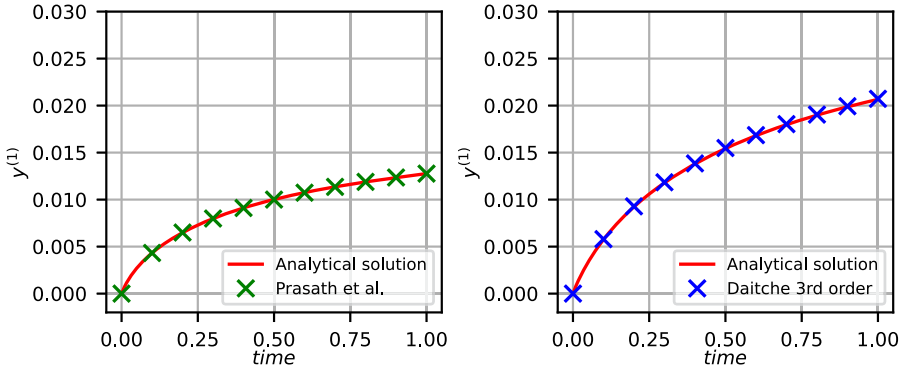


Fig. 3. Horizontal component of the position versus time for two particles decelerating in a quiescent flow with initial position $\boldsymbol{y}(t_0) = (0,0)^T$ and initial relative velocity $\boldsymbol{q}(0,t_0) = (0.1,0)^T$ with $S = 0.3$ and $R = 7/9$ (left) or $R = 4/3$ (right) for $t \in [0,1]$.

(right) as markers. Again, the solutions for the other numerical methods can be generated using the provided code. The analytical solution reads

$$y^{(1)}(t) = y_0^{(1)} + u^{(1)}t + 2\frac{v_0^{(1)}}{\pi}\int_0^\infty \frac{\gamma(1-e^{-k^2t})}{(\alpha-k^2)^2+(k\gamma)^2}dk, \tag{39a}$$

$$\begin{aligned} y^{(2)}(t) = y_0^{(2)} &+ \frac{1-\cos(\lambda t)}{\lambda} + \frac{2}{\pi}\int_0^\infty v_0^{(2)}\frac{\gamma(1-e^{-k^2t})}{k^2\gamma^2+(k^2-\alpha)^2}\,dk + \\ &+ \frac{2}{\pi}\frac{(1-R)\lambda}{R}\int_0^\infty \frac{k^2\gamma e^{-k^2t}}{(k^2\gamma^2+(k^2-\alpha)^2)(k^4+\lambda^2)}dk + \\ &+ \frac{2}{\pi}\frac{(1-R)\lambda}{R}\int_0^\infty \frac{k^4\gamma\sin(\lambda t)}{(k^2\gamma^2+(k^2-\alpha)^2)(k^4+\lambda^2)\lambda}dk + \\ &- \frac{2}{\pi}\frac{(1-R)\lambda}{R}\int_0^\infty \frac{k^2\gamma\cos(\lambda t)}{(k^2\gamma^2+(k^2-\alpha)^2)(k^4+\lambda^2)}dk. \end{aligned} \tag{39b}$$

It is shown as a solid red line in Fig. 2. The expressions for the horizontal (39a) and vertical (39b) components are obtained by using $f^{(1)} \equiv 0$ and $f^{(2)}(s) = \left(\frac{1}{R}-1\right)(\lambda\cos(\lambda s))$ in equation (3.16) in Prasath et al. [9]. As for the vortex, the numerical approximations shown as crosses match the analytical solutions.

Quiescent flow field. In this example, the flow field is zero everywhere and the particle, starting with a non-zero initial velocity, decelerates until it comes to a halt. The solution for a single component corresponds to equation (39a) with $u^{(1)} = 0$. Fig. 3 again shows that numerical and analytical solution agree. Since no motion is happening if the particle is initially at rest, we only consider the case of non-zero initial relative velocity.

3.1.2. Fully unsteady and inhomogeneous flow fields

Next we describe the two benchmark problems with flow fields that vary in both space and time where no analytical solution is available.

Bickley jet. The Bickley jet is a flow field with stream function

$$\Psi(x,y,t) = -U_0 L\tanh(y/L) + \sum_{i=1}^{3} A_i U_0 L\,\mathrm{sech}^2(y/L)\cos(k_i x - \sigma_i t) \tag{40}$$

and was proposed by Rypina et al. [47] as a model for atmospheric jets. We use the values for U_0, L, k_i and σ_i given by Hadjighasem et al. [48]. This flow field "serves as an idealized model of the stratospheric flow" [51]. Fig. 4 shows the reference solution computed with Prasath et al.'s algorithm [9] and solutions computed with the second- and fourth-order IMEX/finite difference approach.

Faraday flow. The second unsteady, inhomogeneous field is an experimentally measured Faraday flow, arising in a thin layer of water in a circular container attached to a shaker vibrating at a frequency of 50 Hz. Details of the experiment are given by Colombi et al. [49,50]. Measurements are taken at 115×86 points in space with 2 mm resolution with a snapshot in time taken every 40 ms and a total of 1056 snapshots. We use rectangular bivariate splines provided by `scipy` to interpolate between data points in space to obtain the velocity field at a given particle position and linear interpolation to interpolate between snapshots in time. Fig. 5 shows example trajectories of particles in the Faraday flow computed with two different algorithms for particles that are lighter (left) and heavier (right) than the fluid.

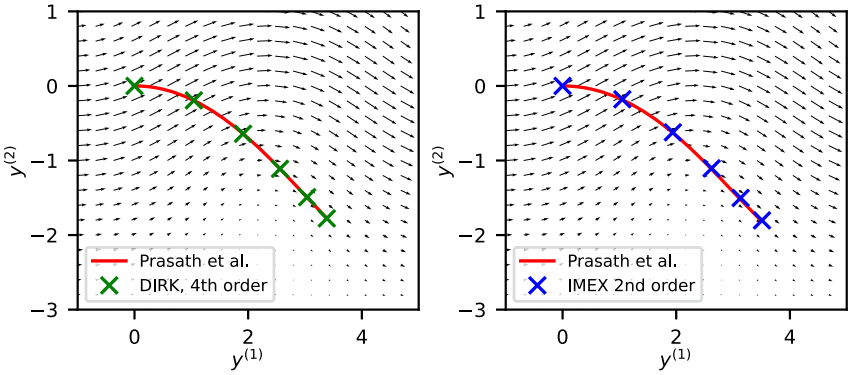


Fig. 4. Trajectory of two particles moving in the Bickley jet with initial position $\boldsymbol{y}(t_0) = (0,0)^T$ and zero initial relative velocity with $S = 0.3$ and $R = 7/9$ (left) and $R = 4/3$ (right) for $t \in [0,1]$. The arrows show the velocity field at $t = 1$.

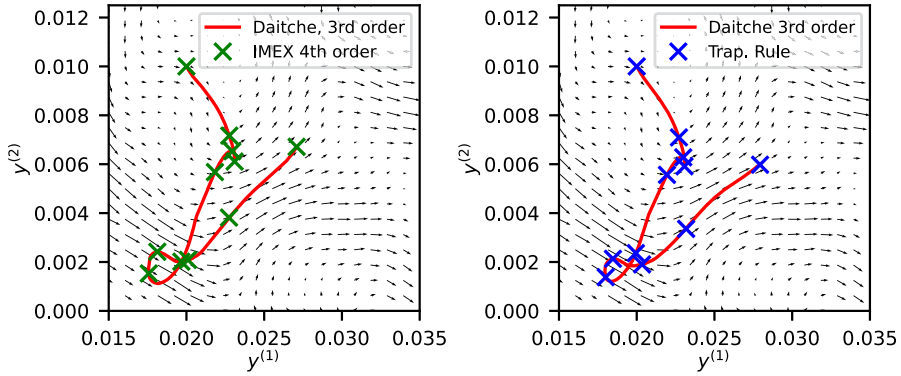


Fig. 5. Trajectory of two particles moving in the Faraday flow with initial position $\boldsymbol{y}(t_0) = (0.02, 0.01)^T$ and zero initial relative velocity, i.e. $\boldsymbol{q}(0,t_0) = (0,0)^T$, with $S = 0.3$ and either $R = 7/9$ (left) or $R = 4/3$ (right) for $t \in [0,5]$. The arrows show the velocity field at $t = 5$.

3.2. Convergence order and accuracy

This section analyzes the convergence orders of all three methods, Prasath al., Daitche and FD, for the five flow fields described above. We investigate the impact of changing the values of R and S and of whether the particle has zero or non-zero initial relative velocity. For the flow fields where an analytic solution is available we report the maximum relative l_2-error over all time steps. If no analytical solution is known and a numerically computed reference is used, we report the relative l_2-error at the final time step. We choose values of $R = 1/3$, corresponding to a particle of gas with density $\rho_p \to 0$ and $R = 7/3$, corresponding to the density of a particle of a heavy rock such as Basalt [52]. In addition, we simulate three intermediate cases of a light particle with $R = 7/9$, a neutrally buoyant particle with $R = 1$ and a heavy particle with $R = 4/3$. Values for the Stokes number are $S \in \{0.01, 0.1, 0.5, 1, 2, 4\}$.

3.2.1. Quiescent, steady inhomogeneous and unsteady homogeneous flow field

We first analyze convergence for the three flow fields for which we have analytical solutions. Throughout, N denotes the number of nodes in space and steps in time for the FD and Prasath et al.'s [9] polynomial expansion method, which we always keep equal. For Daitche's method [29], where no spatial discretization is needed, N simply indicates the number of time steps.

Vortex flow field. Fig. 6 shows convergence of all methods against the analytical solution for three different values of R. Lines with slopes two, three and four are shown as a guide to the eye. Second order FD converges with their theoretically expected rate in all three cases, both with a diagonally implicit or IMEX Runge-Kutta method. Fourth-order FD with DIRK or IMEX converges with orders between 3 and 4, slightly lower than what would be theoretically expected. Only for the neutrally buoyant particle with $R = 1$ do they achieve full fourth order. Daitche's [29] third order method achieves its expected convergence order only for the case of a neutrally buoyant particle ($R = 1$, middle) and converges with about order two for the lighter or denser particle. Most likely this is because coefficients were computed in double and not quadruple precision [22,29]. For very small values of S and R, Daitche's [29] method becomes unstable. Prasath's method converges with order between two and three for $R \neq 1$ and achieves what looks like spectral accuracy for $R = 1$, where it reaches round-off error with only very few nodes.

Oscillatory background flow field. Fig. 7 shows convergence of all methods for particles of different densities with zero relative velocity in the oscillatory flow field. Fig. 8 shows convergence for the same particles but with a small non-zero initial relative velocity. For zero initial relative velocity, results are very similar to those for the vortex. Second-order FD converges with its expected order with both implicit trapezoidal rule and IMEX2. Fourth-order FD reaches its full order only for $R = 1$ and converges with about order three otherwise. Daitche [29] achieves order three for $R = 1$ and an order of somewhat higher than two if $R \neq 1$. As before, Prasath converges with order between two and three unless the particle is neutrally buoyant, in which case it again achieves spectral convergence.

The picture changes dramatically if the particle starts with a non-zero initial relative velocity. Fig. 8 shows convergence for the same parameters as above except that now the velocity $\boldsymbol{v}(t_0)$ of the particle at the beginning is slightly different from the flow field at its starting point, leading to a small but non-zero initial relative velocity of $\boldsymbol{q}(0,t_0) = (0, 0.1)$. This leads to a discontinuity in the initial value for (3) since we have $q^{(2)}(0,t_0) > 0$ but $q^{(2)}(x,t_0) = 0$ for $x > 0$. For FDM, this leads to a massive deterioration in convergence order. Second-order FD

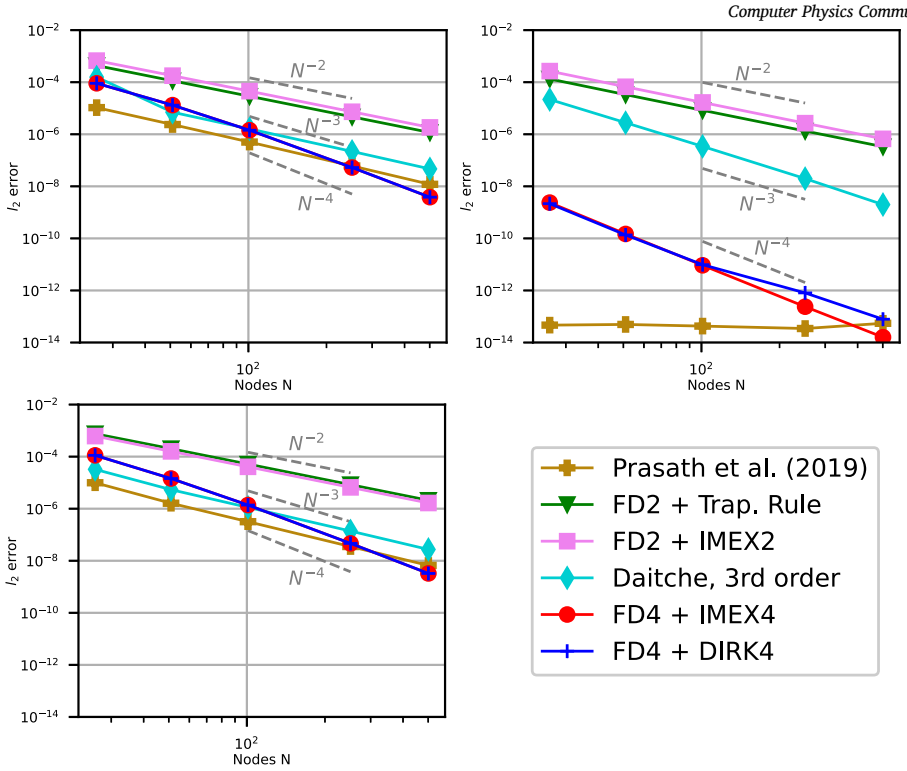


Fig. 6. Error versus N for a particle moving in the vortex with initial position $\boldsymbol{y}(t_0) = (1,0)^T$ and zero initial relative velocity with $S = 0.1$ and $R = 7/9$ (top left), $R = 1$ (top right) or $R = 4/3$ (bottom left) for $t \in [0,1]$.

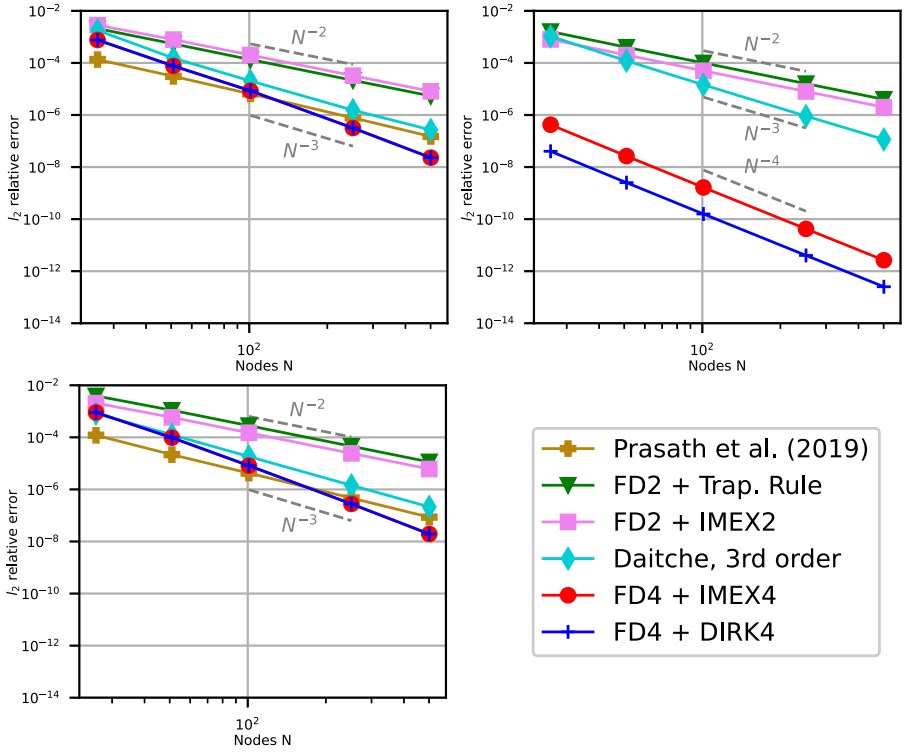


Fig. 7. Convergence for a particle moving in an oscillatory background starting from $\boldsymbol{y}(t_0) = (0,0)^T$ and zero initial relative velocity, $S = 0.1$ and either $R = 7/9$ (top left), $R = 1$ (top right) or $R = 4/3$ (bottom left) for $t \in [0,1]$.

methods are reduced to first order whereas the fourth-order FD variants converge even slower, with orders of about 0.7. Prasath's method looses the spectral accuracy we observed for zero initial relative velocity but still remains highly accurate and converges with approximately order two. While finite differences rely on Taylor series and thus are well-known to have big problems resolving discontinuities, Prasath's method uses an integral formulation which seems to be less affected. However, even Daitche's method [29], which does not rely on (3), is also reduced

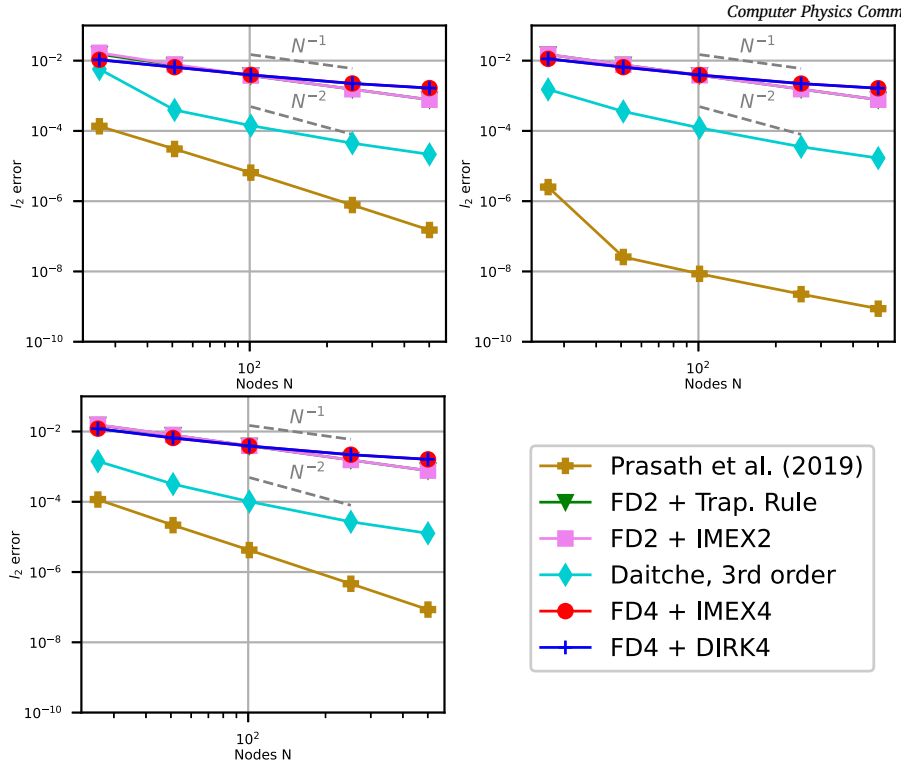


Fig. 8. Convergence for a particle moving in an oscillatory background starting from $\boldsymbol{y}(t_0) = (0,0)^T$ and initial relative velocity $\boldsymbol{q}(0,t_0) = (0,0.1)^T$, with $S = 0.1$ and $R = 7/9$ (top left), $R = 1$ (top right) or $R = 4/3$ (bottom left) for $t \in [0,1]$.

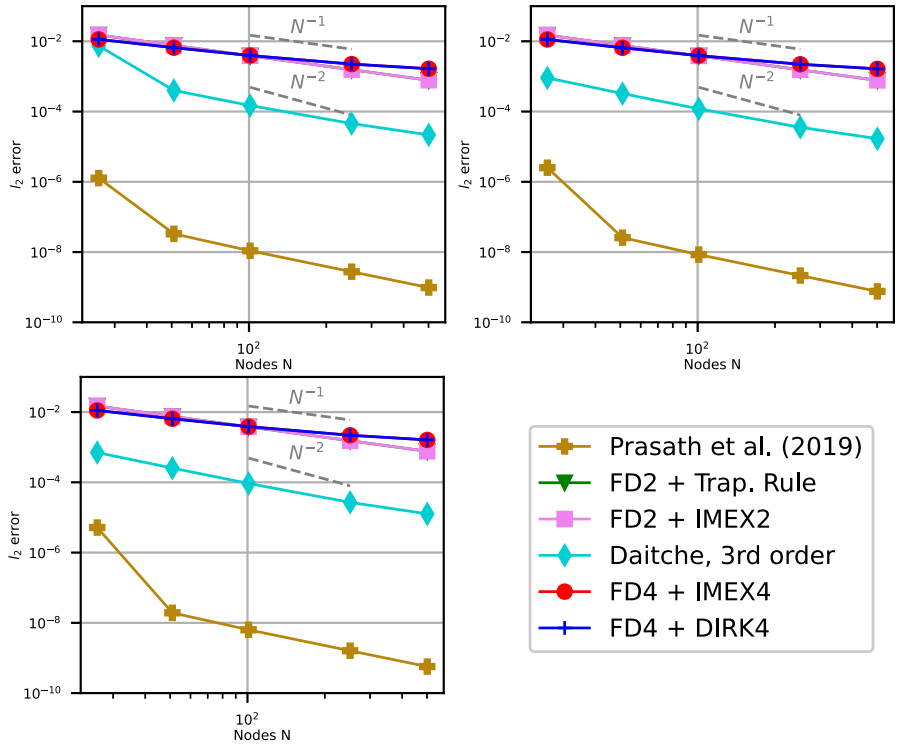


Fig. 9. Convergence plot for a particle moving in a quiescent flow, initialized at $\boldsymbol{y}(t_0) = (0,0)^T$ and initial relative velocity $\boldsymbol{q}(0,t_0) = (0.1,0)^T$, with $S = 0.1$ and $R = 7/9$ (top left), $R = 1$ (top right) or $R = 4/3$ (bottom left) for $t \in [0,1]$.

to slightly better than first order convergence. One reason is likely the very stiff initial relaxation of the particle velocity to the flow field. Order reduction for stiff ODEs is a well-documented phenomenon [53]. However, the choice of the parameter c, see §2.2, and other factors probably also play a role.

Relaxing particle. Fig. 9 shows the convergence plots for all methods against the analytical solution of a relaxing particle in a quiescent flow (39a) for three different values of R. Since the field is zero everywhere, the particle has to start with nonzero initial velocity to induce motion. The result is very similar to what we showed for the oscillatory

Table 1
Numerically computed convergence orders for a particle in the Bickley jet with for $t \in [0,1]$ and different values of R and S. The initial conditions are $\boldsymbol{y} = (0,0)^T$ and $\boldsymbol{q}(0,t_0) = (0,0)^T$.

Bickley jet with zero initial relative velocity and $S = 0.1$					
Method	$R = 1/3$	$R = 7/9$	$R = 1$	$R = 4/3$	$R = 7/3$
Prasath et al.	2.3	2.5	machine-ϵ	2.6	3.2
FD2 + Trap.	2.1	2.1	2.0	2.0	2.00
FD2 + IMEX2	2.1	2.1	2.0	2.0	2.0
Daitche3	unstable	2.9	3.0	2.8	2.7
FD4 + IMEX4	**2.8**	**3.2**	**4.1**	**3.2**	**3.5**
FD4 + DIRK4	**2.8**	**3.2**	3.9	**3.2**	**3.5**

Bickley jet with zero initial relative velocity and $R = 4/3$						
Method	$S = 0.01$	$S = 0.1$	$S = 0.5$	$S = 1$	$S = 2$	$S = 4$
Prasath et al.	2.4	2.6	2.7	3.4	3.0	**4.2**
FD2 + Trap.	2.0	2.0	2.0	2.0	2.0	2.0
FD2 + IMEX2	2.0	2.0	2.0	2.0	2.0	2.0
Daitche3	unstable	2.8	2.8	2.9	2.9	3.0
FD4 + IMEX4	**2.4**	**3.2**	**4.0**	**4.0**	**3.9**	3.9
FD4 + DIRK4	**2.4**	**3.2**	**4.0**	**4.0**	**3.9**	3.8

Table 2
Numerically computed convergence orders for a particle in the Bickley jet for $t \in [0,1]$ with different values for S and R. The initial conditions are $\boldsymbol{y} = (0,0)^T$ and $\boldsymbol{q}(0,t_0) = (0.5414,0)^T$.

Bickley jet with non-zero initial relative velocity and $S = 0.1$					
Method	$R = 1/3$	$R = 7/9$	$R = 1$	$R = 4/3$	$R = 7/3$
Prasath et al.	**2.3**	**2.5**	**2.6**	**2.7**	**3.3**
FD2 + Trap./IMEX2	0.9	0.9	1.0	1.0	1.0
Daitche3	unstable	1.9	1.4	1.3	1.3
FD4 + IMEX4 or DIRK4	0.5	0.5	0.5	0.5	0.6

Bickley jet with non-zero initial relative velocity and $R = 4/3$						
Method	$S = 0.01$	$S = 0.1$	$S = 0.5$	$S = 1$	$S = 2$	$S = 4$
Prasath et al.	**2.3**	**2.7**	**3.0**	**3.5**	**3.4**	**4.1**
FD2 + Trap./IMEX2	1.00	1.0	1.0	1.0	1.0	1.0
Daitche3	unstable	1.3	1.3	1.4	1.5	1.5
FD4 + IMEX4	0.5	0.5	0.6	0.6	0.6	0.7
FD4 + DIRK4	0.5	0.5	0.6	0.6	0.6	0.7

background flow field. FD2 converges with order one and FD4 converges even slower. Daitche [29] converges slightly faster than order one but is more accurate while Prasath is by far the most accurate and converges with order two or more.

3.2.2. Fully unsteady and inhomogeneous flow fields

Here we study convergence for the two unsteady and inhomogeneous fields where the error is measured against a numerically computed reference solution.

Bickley jet. Tables 1 and 2 show convergence rates of all methods for a particle in the Bickley jet with different values of R and S with zero and non-zero initial relative velocity. We observe a very similar pattern as for the three simpler test cases before: for zero initial relative velocity, FD2 and Daitche [29] largely achieve close to their theoretical order. For very small values of S and R, however, Daitche's method [29] eventually becomes unstable. The fourth order FD method suffers from order reduction for very light or very heavy particles, although the effect is more pronounced in the former case, as well as for very small Stokes numbers, but still provides the highest order in almost all cases. In these cases, relaxation of the particle velocity against the flow field is very rapid and the problem becomes very stiff. Prasath et al.'s [9] method converges with orders between two and four, depending on the values of R and S. A notable exception is zero initial relative velocity and $R = 1$ where it achieves machine precision.

For nonzero initial relative velocity, the finite difference method suffers from dramatic order reduction. FD2 at least remains first order accurate throughout but FD4 is reduced to about order 0.5, rendering it

Table 3
Numerically computed convergence orders for a particle in the Faraday flow field for $t \in [0,1]$ and different values of S and R. The initial conditions are $\boldsymbol{y} = (0.02, 0.01)^T$ and $\boldsymbol{q}(0,t_0) = (0,0)^T$.

Faraday flow with zero initial relative velocity and $S = 0.1$					
Method	$R = 1/3$	$R = 7/9$	$R = 1$	$R = 4/3$	$R = 7/3$
Prasath et al.	**2.2**	2.5	2.9	2.6	**2.8**
FD2 + Trap.	2.2	2.1	2.1	1.9	2.2
FD2 + IMEX2	2.1	2.1	2.1	2.3	2.1
Daitche3	unstable	**3.2**	2.7	2.6	2.5
FD4 + IMEX4	2.1	2.6	**4.0**	**2.7**	2.7
FD4 + DIRK4	2.1	2.0	2.1	2.6	2.6

Faraday flow with zero initial relative velocity and $R = 4/3$						
Method	$S = 0.01$	$S = 0.1$	$S = 0.5$	$S = 1$	$S = 2$	$S = 4$
Prasath et al.	2.3	2.6	2.4	**2.7**	2.6	**3.1**
FD2 + Trap.	2.1	1.9	1.9	2.1	2.1	2.2
FD2 + IMEX2	2.0	2.3	2.1	2.0	2.0	2.0
Daitche3	unstable	2.6	**2.5**	2.4	2.4	2.5
FD4 + IMEX4	2.3	**2.7**	2.5	2.5	2.2	2.7
FD4 + DIRK4	**2.3**	2.6	2.2	2.3	**2.7**	2.8

Table 4
Numerically computed convergence orders for a particle in the Faraday flow field with $S = 0.1$ and different values of R for $t \in [0,1]$. The initial conditions are $\boldsymbol{y} = (0.02, 0.01)^T$ and $\boldsymbol{q}(0,t_0) = (0.00052558, -0.00064947)^T$.

Faraday flow with non-zero initial relative velocity and $S = 0.1$					
Method	$R = 1/3$	$R = 7/9$	$R = 1$	$R = 4/3$	$R = 7/3$
Prasath et al.	**2.2**	2.5	**2.6**	**2.7**	**2.8**
FD2 + Trap.	0.9	1.0	1.0	1.0	1.0
FD2 + IMEX2	1.1	1.0	1.0	1.0	0.9
Daitche3	unstable	**2.7**	2.1	2.2	2.5
FD4 + IMEX4	0.6	0.6	0.6	0.6	0.6
FD4 + DIRK4	0.6	0.6	0.6	0.6	0.6

Faraday flow with non-zero initial relative velocity and $R = 4/3$						
Method	$S = 0.01$	$S = 0.1$	$S = 0.5$	$S = 1$	$S = 2$	$S = 4$
Prasath et al.	**2.3**	**2.7**	**2.6**	**3.1**	**2.6**	**3.4**
FD2 + Trap.	0.9	1.0	1.0	1.0	1.0	1.0
FD2 + IMEX2	1.0	1.0	0.9	0.9	0.8	0.7
Daitche3	unstable	2.2	2.4	2.4	2.4	2.6
FD4 + IMEX4	0.6	0.6	0.6	0.6	0.6	0.6
FD4 + DIRK4	0.6	0.6	0.6	0.6	0.6	0.6

largely useless in this case. Most likely this is because the initial value in (3) is not continuous for a non-zero initial relative velocity and finite difference are well known to cope badly with discontinuities. Daitche's method [29] also suffers from reduced order but not as dramatic as the FDM, achieving orders between one and two. By contrast, convergence of Prasath's method remains largely unaffected, probably because it relies on an integral formulation of the PDE which can better handle discontinuities. It delivers the highest order in all cases.

Faraday flow. Tables 3 and 4 show convergence rates of all methods for a particle in the Faraday flow field with different values of R and S and zero and non-zero initial relative velocity. The results are similar to those for the Bicklet jet, although Daitche [29] and FDM show somewhat lower convergence orders, even for zero initial relative velocity. Second-order FD achieves order two for zero and about first order for non-zero initial relative velocity, although FD-IMEX2 is somewhat below order one for large values of S. The instability of Daitche's method [29] for very small values of S and R as well as the dramatic order reduction for fourth-order FD for non-zero initial relative velocity remain: in the latter case, FD4 converges only with order slightly better than one-half. Prasath et al.'s [9] method provides the highest order in almost all cases with non-zero initial relative velocity. In contrast to the Bickley jet, there is some variation in which method provides the highest order for zero relative initial velocity. Depending on the specific values for S and R, it can be Prasath [9], Daitche [29] or FD4.

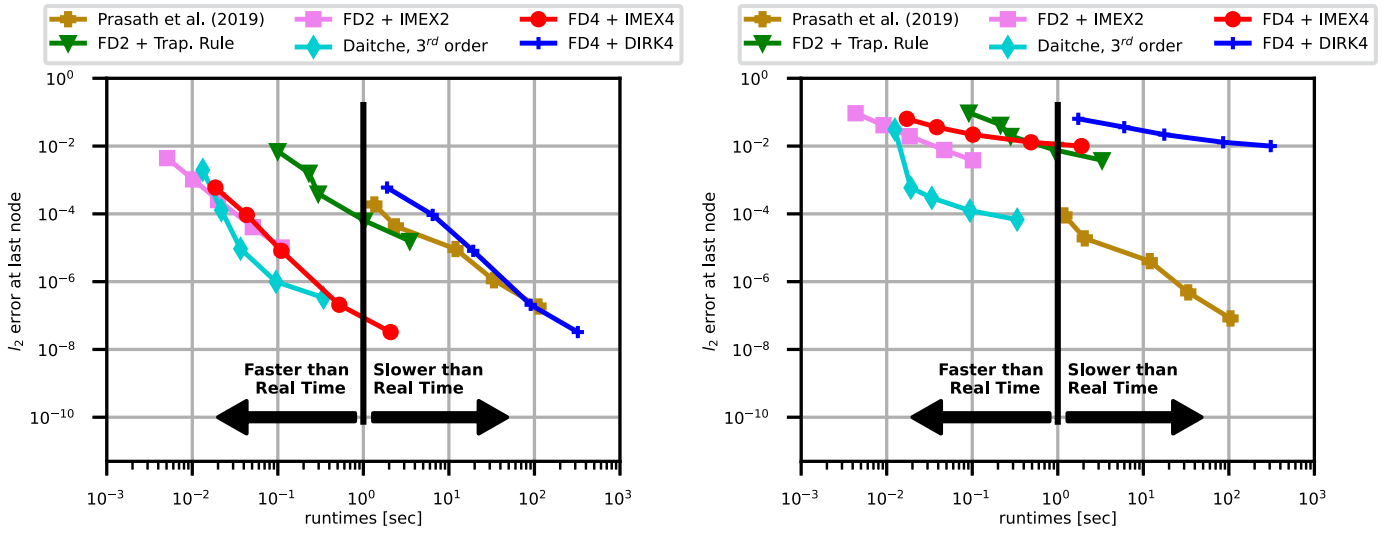


Fig. 10. Error against time-to-solution for a particle moving in the Bickley Jet with $S = 0.1$ and $R = 7/9$ for $t \in [0, 1]$. The initial conditions are $\boldsymbol{y}(t_0) = (0, 0)^T$ and $\boldsymbol{q}(0, t_0) = (0, 0)^T$ (left) and $\boldsymbol{q}(0, t_0) = (0.5414, 0)^T$ (right).

3.3. Computational efficiency

This last part analyzes computational efficiency of all methods by means of work-precision plots for the two unsteady, inhomogeneous flow fields. Work-precision plots show computational effort, here measured as runtime in seconds, on the x-axis against error on the y-axis. This allows to quickly see either which method is the fastest to produce a solution of a certain numerical accuracy or which method can produce the most accurate solution within a certain runtime. Runtime is measured with the `time` module in Python and includes the time required to compute all time steps of the trajectory but excludes setup times.

Note that the Fortran implementation of the method by Jaganathan et al. [34] for the oscillatory background flow field achieved l_2-errors of 10^{-4} in 10^{-3} seconds wallclock time for a 5 second trajectory. While we only show efficiency results for Bickley jet and Faraday flow here, results with the same code published elsewhere [54] show that our Python implementation of Daitche's method [29] requires around 10^{-2} seconds wallclock time to match this accuracy for a 1 second trajectory. Whether Jaganathan et al.'s [34] better performance is due to the numerical method or because compiled languages like Fortran are significantly faster than interpreted languages like Python is unknown. Refactoring our Python code into Fortran and providing a detailed comparison is beyond the scope of this paper but would be a very useful undertaking in the future.

Bickley jet. Fig. 10 shows error versus runtime for a particle in the Bickley jet with $S = 0.1$ and $R = 7/9$ and zero (left) and non-zero (right) initial relative velocity. The trajectory of the particle is simulated over 1 s, so a runtime of less than 1 s is considered faster than real-time even though effective use of the model for closed loop control will not be realistically possible throughout that whole range. For zero initial relative velocity, Daitche [29], FD2 + IMEX2 and FD4 + IMEX4 can all provide errors below 10^{-2} with runtimes between 10 ms to 100 ms. Prasath et al.'s [9] method is too computationally costly to be capable of producing real-time results. For non-zero initial relative velocity, the performance of the FD methods suffers significantly from the reduction to order one. While FD2 + IMEX2 can still produce errors of almost 10^{-2} with runtimes of around 100 ms, Daitche's method [29] is clearly superior in this case. Note that the provided code is written in Python and is not using multi-threading. An optimized implementation in a compiled language would be several times faster.

Finally, when comparing the fully implicit FD2 + Trapezoidal rule and FD4 + DIRK4 methods with their implicit-explicit counterparts FD2 + IMEX2 and FD4 + IMEX4, both show similar convergence behavior. However, the IMEX variants require noticeably less runtime, making them more efficient. This underscores the value of the IMEX approach to avoid a costly fully nonlinear solver.

Faraday flow. Fig. 11 shows error against runtime for a particle with $S = 0.1$ and $R = 7/9$ in the Faraday flow, again for trajectories computed over 1 s. All methods are more accurate than for the Bicklet jet, likely because the flow velocities in the Faraday flow are smaller. For zero initial relative velocity, FD2 + IMEX2, FD4 + IMEX4 and Daitche [29] all perform very similarly with a slight advantage for FD4 + IMEX4 for higher accuracies. For non-zero initial relative velocity, Daitche [29] again overtakes the FD methods with respect to speed of computation except for errors above 10^{-4}, where FD2 becomes more efficient. As for the Bickley jet, while Prasath et al.'s [9] method is highly accurate, it is too computationally costly to provide results in real-time. Again, comparing the second- and fourth-order fully implicit methods with their implicit-explicit counterparts illustrates that the latter provides substantial gains in efficiency.

4. Conclusions

The Maxey-Riley-Gatignol equations (MRGE) describe the movement of inertial particles in a fluid and are used in a wide range of applications. They contain an integral term, called Basset history term, that makes them difficult to solve numerically. Hence this term is often neglected, even though it has been shown in theory and experiments that this can lead to significant errors. While direct numerical solvers for the full MRGE exist [29], they need to store all computed time steps to evaluate the integral term which can lead to memory problems, in particular when trajectories of many particles are computed concurrently [55]. Approximations to the history term based on windowing are used but require careful calibration of parameters that do not necessarily generalize to very different flow fields [21]. A recent result by Prasath et al. [9] shows that solving the full MRGE is equivalent to solving a one-dimensional diffusive partial differential equation with nonlinear boundary condition on a semi-infinite domain. This allows to apply well established techniques for the numerical solution of partial differential equations to the MRGE and to avoid the memory demands of direct solvers.

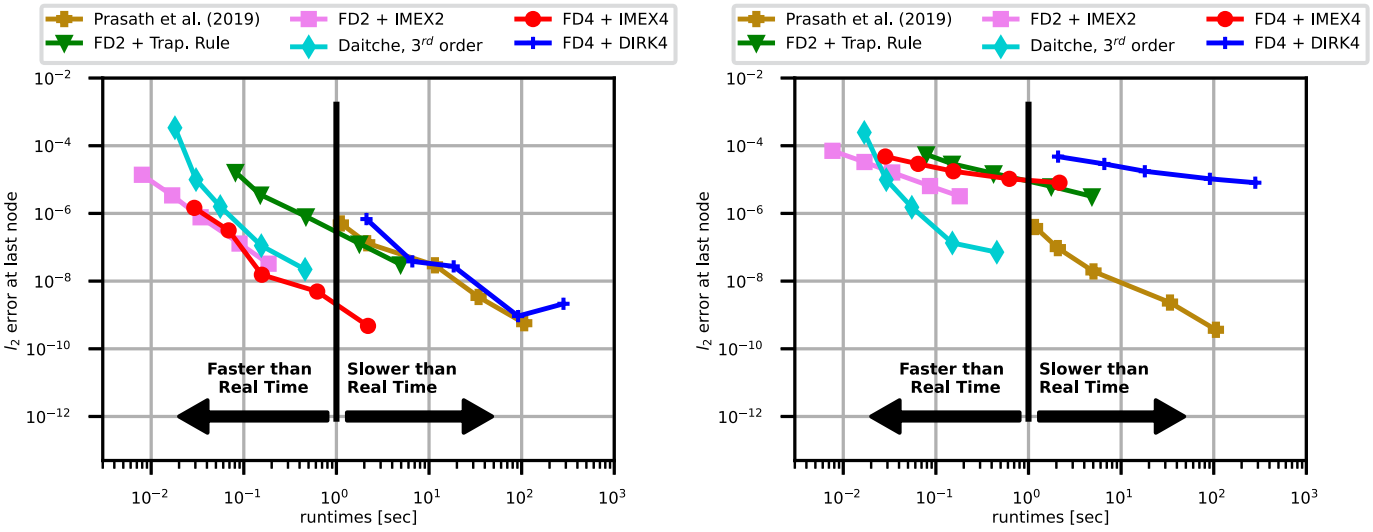


Fig. 11. Error against time-to-solution for a particle moving in the Faraday flow field with $S = 0.1$ and $R = 7/9$ for $t \in [0, 1]$. The initial position is $\boldsymbol{y}(t_0) = (0.02, 0.01)^T$ and the initial relative velocity is $\boldsymbol{q}(t_0) = (0, 0)^T$ (left) and $\boldsymbol{q}(0, t_0) = (0.00052558, -0.00064947)^T$ (right).

However, solving the reformulated MRGE requires to deal with the semi-infinite computational domain. Prasath et al. [9] propose a numerical approach based on polynomial expansions. While accurate, this method is computationally expensive. Instead, we adopt techniques developed by Koleva [35] and Fazio and Jannelli [37] to develop finite difference discretizations of order two and four for the reformulated MRGE, resulting in a semi-discrete initial value problem. Applying an explicit method would be inefficient due to the stiffness of the discrete Laplacian, whilst a fully implicit method would require nonlinear Newton solves in each step because of the nonlinear boundary condition. Therefore, we propose to use an implicit-explicit RKM that treats the nonlinear boundary condition explicitly but the discrete Laplacian implicitly. This way, only linear solvers are required but the strict stability limit from the discrete Laplacian is avoided.

We test all methods, Prasath et al.'s polynomial expansion, Daitche's direct integrator and our FD2 and FD4 methods, for five different flow fields. For three of them, a quiescent flow, a steady, inhomogeneous vortex and a non-steady, homogeneous oscillating background, analytical solutions are available. For two unsteady, inhomogeneous fields, the Bickley jet and a Faraday flow interpolated from experimental data, we compute a reference solution with Prasath et al.'s polynomial expansion approach with very high resolution. Our numerical experiments suggest that:

1. Daitche's method is efficient for both zero and non-zero relative velocity and a good overall choice. However, it can become unstable for very small density ratios and Stokes numbers.
2. Prasath's polynomial expansion based method is very accurate for both zero and non-zero relative velocity but computationally expensive.
3. The finite difference method of order two is efficient for both zero and non-zero initial relative velocity but, particularly in the latter case, Daitche's method is significantly more efficient.
4. The finite difference method of order four is efficient for zero initial relative velocity and could outperform both Daitche and FD2 + IMEX2 for the Faraday flow.

Daitche and FD2 + IMEX2 could both simulate 1 s trajectories in significantly less than one second runtime, illustrating their potential to provide solutions to the MRGE in real-time. For the Faraday flow, for example, FD2 + IMEX2 could compute a 1 s-trajectory with a relative discretization error of 7×10^{-5} in 8 ms. An optimized implementation in a compiled language plus effective use of vectorization and multithreading would make simulations even faster. This could potentially allow to use the full MRGE as a model in either closed-loop control or for model-based filtering to enable real-time particle tracking, for example of Lagrangian sensors that are too large to be considered perfect tracers [56].

CRediT authorship contribution statement

Julio Urizarna-Carasa: Writing – original draft, Software, Methodology, Investigation. **Leon Schlegel:** Software, Methodology. **Daniel Ruprecht:** Writing – review & editing, Supervision, Project administration, Methodology, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Julio Urizarna-Carasa reports financial support was provided by German Research Foundation (503850735). If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We are grateful to Prof. Alexandra von Kameke for providing us with the data from the experimental measurements of the Faraday flow and to Prof. Kathrin Padberg-Gehle for providing the Bickley jet example.

Appendix A. Boundary conditions for the 4th order upwind scheme

Here we explain how to obtain a finite difference approximation of order four for the spatial derivative at the boundary $\boldsymbol{q}_x(0,t)$ in (3). The procedure is inspired by [40], where upwinding schemes for first and second derivatives for Robin boundary conditions are discussed. Their general form is

$$a_0(\partial_{\tilde{\xi}}q)_0 + a_1(\partial_{\tilde{\xi}}q)_1 = \hat{b}_0\left(\alpha q_0 - \gamma(\partial_x q)_0\right) + \frac{1}{d}\left\{b_0 q_0 + b_1 q_1 + b_2 q_2 + b_3 q_3\right\}, \tag{A.1}$$

with $a_0, a_1, b_0, \hat{b}_0, b_1, \hat{b}_1, b_2, b_3 \in \mathbb{R}$. We use the relation between mesh derivatives (21) to change the spatial derivative in the boundary condition from pseudo-space to the equidistant grid and obtain

$$a_0(\partial_{\tilde{\xi}}q)_0 + a_1(\partial_{\tilde{\xi}}q)_1 = \hat{b}_0\left(\alpha q_0 - \frac{\gamma}{c}(\partial_{\tilde{\xi}}q)_0\right) + \frac{1}{d}\left\{b_0 q_0 + b_1 q_1 + b_2 q_2 + b_3 q_3\right\}. \tag{A.2}$$

We then expand all the elements of equation (A.2) around ξ_0 and find

$$\begin{aligned}
(\partial_{\tilde{\xi}}q)_0 &: 0 && +(\partial_{\tilde{\xi}}q)_0, \\
(\partial_{\tilde{\xi}}q)_1 &: 0 && +(\partial_{\tilde{\xi}}q)_0 && +d\,(\partial_{\tilde{\xi}}^2 q)_0 && +\frac{d^2}{2}(\partial_{\tilde{\xi}}^3 q)_0 && +\frac{d^3}{6}(\partial_{\tilde{\xi}}^4 q)_0 && +\frac{d^4}{24}(\partial_{\tilde{\xi}}^5 q)_0, \\
q_0 &: q_0, \\
q_1 &: q_0 && +d(\partial_{\tilde{\xi}}q)_0 && +\frac{d^2}{2}(\partial_{\tilde{\xi}}^2 q)_0 && +\frac{d^3}{6}(\partial_{\tilde{\xi}}^3 q)_0 && +\frac{d^4}{24}(\partial_{\tilde{\xi}}^4 q)_0 && +\frac{d^5}{120}(\partial_{\tilde{\xi}}^5 q)_0, \\
q_2 &: q_0 && +2d(\partial_{\tilde{\xi}}q)_0 && +\frac{4d^2}{2}(\partial_{\tilde{\xi}}^2 q)_0 && +\frac{8d^3}{6}(\partial_{\tilde{\xi}}^3 q)_0 && +\frac{16d^4}{24}(\partial_{\tilde{\xi}}^4 q)_0 && +\frac{32d^5}{120}(\partial_{\tilde{\xi}}^5 q)_0, \\
q_3 &: q_0 && +3d(\partial_{\tilde{\xi}}q)_0 && +\frac{9d^2}{2}(\partial_{\tilde{\xi}}^2 q)_0 && +\frac{27d^3}{6}(\partial_{\tilde{\xi}}^3 q)_0 && +\frac{81d^4}{24}(\partial_{\tilde{\xi}}^4 q)_0 && +\frac{243d^5}{120}(\partial_{\tilde{\xi}}^5 q)_0.
\end{aligned}$$

By comparing coefficients we obtain the fourth order method

$$\begin{aligned}
q_0 &: && 0 && +0 && +\alpha\hat{b}_0 && +\frac{1}{d}b_0 && +\frac{1}{d}b_1 && +\frac{1}{d}b_2 && +\frac{1}{d}b_3 && = 0, \\
(\partial_{\tilde{\xi}}q)_0 &: && a_0 && +a_1 && +\frac{\gamma}{c}\hat{b}_0 && +0 && -b_1 && -2b_2 && -3b_3 && = 0, \\
(\partial_{\tilde{\xi}}^2 q)_0 &: && 0 && +da_1 && +0 && +0 && -\frac{d}{2}b_1 && -\frac{4d}{2}b_2 && -\frac{9d}{2}b_3 && = 0, \\
(\partial_{\tilde{\xi}}^3 q)_0 &: && 0 && +\frac{d^2}{2}a_1 && +0 && +0 && -\frac{d^2}{6}b_1 && -\frac{8d^2}{6}b_2 && -\frac{27d^2}{6}b_3 && = 0, \\
(\partial_{\tilde{\xi}}^4 q)_0 &: && 0 && +\frac{d^3}{6}a_1 && +0 && +0 && -\frac{d^3}{24}b_1 && -\frac{16d^3}{24}b_2 && -\frac{81d^3}{24}b_3 && = 0, \\
(\partial_{\tilde{\xi}}^5 q)_0 &: && 0 && +\frac{d^4}{24}a_1 && +0 && +0 && -\frac{d^4}{120}b_1 && -\frac{32d^4}{120}b_2 && -\frac{243d^4}{120}b_3. &&
\end{aligned}$$

We set $a_0 = a_1 = 1$. This results in the coefficients

$$\hat{b}_0 = -\frac{2c}{3\gamma} \quad b_0 = \frac{12\alpha dc - 17\gamma}{18\gamma} \quad b_1 = \frac{1}{2} \quad b_2 = \frac{1}{2} \quad b_3 = -\frac{1}{18},$$

and a leading error term of $\frac{d^4}{60}(\partial_{\tilde{\xi}}^5 q)_0$. In summary, the fourth order approximation at the boundary becomes

$$(\partial_{\tilde{\xi}}q)_0 + (\partial_{\tilde{\xi}}q)_1 = -\frac{2c}{3\gamma}\left(\alpha q_0 - \frac{\gamma}{c}(\partial_{\tilde{\xi}}q)_0\right) + \frac{1}{d}\left(\frac{12\alpha dc - 17\gamma}{18\gamma}\right)q_0 + \frac{1}{2d}q_1 + \frac{1}{2d}q_2 - \frac{1}{18d}q_3,$$

or written as a derivative with respect to x,

$$\begin{aligned}
c(\partial_x q)_0 + \frac{Nc}{N-1}(\partial_x q)_1 = &-\frac{2c}{3\gamma}\left(\alpha q_0 - \gamma(\partial_x q)_0\right) \\
&+ \frac{1}{d}\left(\frac{12\alpha dc - 17\gamma}{18\gamma}\right)q_0 + \frac{1}{2d}q_1 + \frac{1}{2d}q_2 - \frac{1}{18d}q_3.
\end{aligned}$$

Finally, we use (3c) to get

$$\begin{aligned}
c(\partial_x q)_0 + \frac{Nc}{N-1}(\partial_x q)_1 = &-\frac{2c}{3\gamma}\left(f(q_0, x_0, t) - (\partial_t q)_0\right) \\
&+ \frac{1}{d}\left(\frac{12\alpha dc - 17\gamma}{18\gamma}\right)q_0 + \frac{1}{2d}q_1 + \frac{1}{2d}q_2 - \frac{1}{18d}q_3.
\end{aligned}$$

Appendix B. Matrices for the fourth order finite difference method

$$M_1 := \begin{bmatrix} c & 0 & \frac{Nc}{N-1} & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & c & 0 & \frac{Nc}{N-1} & 0 & \ddots & \ddots & \ddots & 0 \\ c & 0 & \frac{4Nc}{N-1} & 0 & \frac{Nc}{N-2} & 0 & \ddots & \ddots & 0 \\ 0 & c & 0 & \frac{4Nc}{N-1} & 0 & \frac{Nc}{N-2} & 0 & \ddots & 0 \\ \vdots & 0 & \frac{Nc}{N-1} & 0 & \frac{4Nc}{N-2} & 0 & \frac{Nc}{N-3} & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \frac{Nc}{4} & 0 & \frac{4Nc}{3} & 0 & \frac{Nc}{2} & 0 \\ \vdots & \ddots & \ddots & \ddots & \frac{Nc}{4} & 0 & \frac{4Nc}{3} & 0 & \frac{Nc}{2} \\ 0 & \ddots & \ddots & \ddots & \ddots & \frac{Nc}{3} & 0 & \frac{4Nc}{2} & 0 \\ 0 & 0 & \dots & \dots & \dots & 0 & \frac{Nc}{3} & 0 & \frac{4Nc}{2} \end{bmatrix},$$

$$M_2 := \begin{bmatrix} c & 0 & 3\frac{Nc}{N-1} & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & c & 0 & 3\frac{Nc}{N-1} & 0 & \ddots & \ddots & \ddots & \ddots & 0 \\ c & 0 & \frac{4Nc}{N-1} & 0 & \frac{Nc}{N-2} & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & c & 0 & \frac{4Nc}{N-1} & 0 & \frac{Nc}{N-2} & 0 & \ddots & \ddots & 0 \\ 0 & 0 & \frac{Nc}{N-1} & 0 & \frac{4Nc}{N-2} & 0 & \frac{Nc}{N-3} & 0 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \frac{Nc}{4} & 0 & \frac{4Nc}{3} & 0 & \frac{Nc}{2} & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 & \frac{Nc}{4} & 0 & \frac{4Nc}{3} & 0 & \frac{Nc}{2} \\ 0 & \ddots & \ddots & \ddots & \ddots & 0 & \frac{Nc}{3} & 0 & \frac{4Nc}{2} & 0 \\ 0 & 0 & \dots & \dots & \dots & \dots & 0 & \frac{Nc}{3} & 0 & \frac{4Nc}{2} \end{bmatrix},$$

$$B_1 := \frac{1}{d}\begin{bmatrix} \frac{12\alpha dc-17\gamma}{18\gamma} & 0 & 1/2 & 0 & 1/2 & 0 & -1/18 & 0 & \dots & 0 \\ 0 & \frac{12\alpha dc-17\gamma}{18\gamma} & 0 & 1/2 & 0 & 1/2 & 0 & -1/18 & \ddots & 0 \\ -3 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & \dots & 0 \\ 0 & -3 & 0 & 0 & 0 & 3 & 0 & 0 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 & -3 & 0 & 0 & 0 \\ 0 & \dots & \dots & \dots & \dots & \dots & 0 & -3 & 0 & 0 \end{bmatrix},$$

$$B_2 := \frac{1}{d}\begin{bmatrix} -17/6 & 0 & 3/2 & 0 & 3/2 & 0 & -1/6 & 0 & \dots & 0 \\ 0 & -17/6 & 0 & 3/2 & 0 & 3/2 & 0 & -1/6 & \dots & 0 \\ -3 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & \dots & 0 \\ 0 & -3 & 0 & 0 & 0 & 3 & 0 & \ddots & \ddots & 0 \\ 0 & 0 & -3 & 0 & 0 & 0 & 3 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 & -3 & 0 & 0 & 0 \\ 0 & \dots & \dots & \dots & \dots & \dots & 0 & -3 & 0 & 0 \end{bmatrix},$$

$$K_1 := \begin{bmatrix} -\frac{2c}{3\gamma} f^{(1)}(\boldsymbol{q}_0(t), \boldsymbol{y}_0(t), t) \\ -\frac{2c}{3\gamma} f^{(2)}(\boldsymbol{q}_0(t), \boldsymbol{y}_0(t), t) \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \text{and} \quad V_1 := \frac{2c}{3\gamma}\begin{bmatrix} \left.\frac{\partial q^{(1)}(\xi(x),t)}{\partial t}\right|_0 \\ \left.\frac{\partial q^{(2)}(\xi(x),t)}{\partial t}\right|_0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Data availability

The used code is published on GitHub [38]. The experimental data for the Faraday flow can be made available upon request.

References

[1] M.R. Maxey, J.J. Riley, Equation of motion for a small rigid sphere in a nonuniform flow, Phys. Fluids 26 (1983) 883–889.

[2] R. Govindarajan, S. Ravichandran, Cloud microatlas, Resonance 22 (2017) 269–277.

[3] G. Falkovich, A. Fouxon, M. Stepanov, Acceleration of rain initiation by cloud turbulence, Nature 419 (2002) 151–154.

[4] M. Niazi Ardekani, G. Sardina, L. Brandt, L. Karp-Boss, R. Bearon, e. Variano, Sedimentation of elongated non-motile prolate spheroids in homogenous isotropic turbulence, arXiv:e-prints, arXiv–1611, 2016.

[5] K. Guseva, A. Daitche, U. Feudel, T. Tél, History effects in the sedimentation of light aerosols in turbulence: the case of marine snow, Phys. Rev. Fluids 1 (2016) 074203.

[6] C.P. Cummins, O.J. Ajayi, F.V. Mehendale, R. Gabl, I.M. Viola, The dispersion of spherical droplets in source–sink flows and their relevance to the COVID-19 pandemic, Phys. Fluids 32 (2020) 083302.

[7] G.P. Galdi, R. Rannacher, A.M. Robertson, S. Turek, Hemodynamical Flows, Oberwolfach Seminars, vol. 37, 2008.

[8] J.H. Cartwright, U. Feudel, G. Károlyi, A.d. Moura, O. Piro, T. Tél, Dynamics of finite-size particles in chaotic fluid flows, in: Nonlinear Dynamics and Chaos: Advances and Perspectives, Springer, 2010, pp. 51–87.

[9] S.G. Prasath, V. Vasan, R. Govindarajan, Accurate solution method for the Maxey–Riley equation, and the effects of Basset history, J. Fluid Mech. 868 (2019) 428–460.

[10] F. Candelier, J. Angilella, M. Souhar, On the effect of the Boussinesq–Basset force on the radial migration of a Stokes particle in a vortex, Phys. Fluids 16 (2004) 1765–1776.

[11] G.P. Langlois, M. Farazmand, G. Haller, Asymptotic dynamics of inertial particles with memory, J. Nonlinear Sci. 25 (2015) 1225–1255.

[12] G.G. Stokes, et al., On the effect of the internal friction of fluids on the motion of pendulums, 1851.

[13] A.B. Basset, A Treatise on Hydrodynamics: with Numerous Examples, vol. 2, Deighton, Bell and Company, 1888.
[14] J. Boussinesq, Sur la resistance qu'oppose un fluide indefini en repos, sans pesanteur, au mouvement varie d'une sphere solide qu'il mouille sur toute sa surface, quand les vitesses restent bien continues et assez faibles pour que leurs carres et produits soient negligiables, C. R. Acad. Sci. Paris 100 (1885) 935–937.
[15] C.W. Oseen, Neuere Methoden und Ergebnisse in der Hydrodynamik, Akademische Verlagsgesellschaft m. b H., Leipzig, 1927.
[16] C.-M. Tchen, Mean Value and Correlation Problems Connected with the Motion of Small Particles Suspended in a Turbulent Fluid, Springer, 2013.
[17] R. Gatignol, The Faxen formulae for a rigid particle in an unsteady non-uniform Stokes flow, J. Méc. Théor. Appl. 2 (1983) 143–160.
[18] F.J. Beron-Vera, M.J. Olascoaga, P. Miron, Building a Maxey–Riley framework for surface ocean inertial particle dynamics, Phys. Fluids 31 (2019) 096602.
[19] P.M. Lovalenti, J.F. Brady, The hydrodynamic force on a rigid particle undergoing arbitrary time-dependent motion at small Reynolds number, J. Fluid Mech. 256 (1993) 561–605.
[20] R. Mei, Flow due to an oscillating sphere and an expression for unsteady drag on the sphere at finite Reynolds number, J. Fluid Mech. 270 (1994) 133–174.
[21] A. Dorgan, E. Loth, Efficient calculation of the history force at finite Reynolds numbers, Int. J. Multiph. Flow 33 (2007) 833–848.
[22] P.A. Moreno-Casas, F.A. Bombardelli, Computation of the Basset force: recent advances and environmental flow applications, Environ. Fluid Mech. 16 (2016) 193–208.
[23] J. Klinkenberg, H. De Lange, L. Brandt, Linear stability of particle laden flows: the influence of added mass, fluid acceleration and Basset history force, Meccanica 49 (2014) 811–827.
[24] H.A. Elghannay, D.K. Tafti, Development and validation of a reduced order history force model, Int. J. Multiph. Flow 85 (2016) 284–297.
[25] M. Parmar, S. Annamalai, S. Balachandar, A. Prosperetti, Differential formulation of the viscous history force on a particle for efficient and accurate computation, J. Fluid Mech. 844 (2018) 970–993.
[26] M. Van Hinsberg, J. ten Thije Boonkkamp, H.J. Clercx, An efficient, second order method for the approximation of the Basset history force, J. Comput. Phys. 230 (2011) 1465–1478.
[27] Y. Niño, M. García, Using Lagrangian particle saltation observations for bedload sediment transport modelling, Hydrol. Process. 12 (1998) 1197–1218.
[28] N. Mordant, J.-F. Pinton, Velocity measurement of a settling sphere, Eur. Phys. J. B, Condens. Matter Complex Syst. 18 (2000) 343–352.
[29] A. Daitche, Advection of inertial particles in the presence of the history force: higher order numerical schemes, J. Comput. Phys. 254 (2013) 93–106.
[30] E.E. Michaelides, A novel way of computing the Basset term in unsteady multiphase flow computations, Phys. Fluids A, Fluid Dyn. 4 (1992) 1579–1582.
[31] F. Tatom, The Basset term as a semiderivative, Appl. Sci. Res. 45 (1988) 283–285.
[32] A.S. Fokas, A unified transform method for solving linear and certain nonlinear PDEs, Proc. R. Soc. Lond. Ser. A, Math. Phys. Eng. Sci. 453 (1997) 1411–1443.
[33] J. Urizarna-Carasa, D. Ruprecht, A. von Kameke, K. Padberg-Gehle, A Python toolbox for the numerical solution of the Maxey-Riley equation, PAMM 22 (2023) e202200242.
[34] D. Jaganathan, R. Govindarajan, V. Vasan, Explicit Runge-Kutta algorithm to solve non-local equations with memory effects: case of the Maxey-Riley-Gatignol equation, arXiv:2308.09714, 2023.
[35] M.N. Koleva, Numerical solution of the heat equation in unbounded domains using quasi-uniform grids, in: International Conference on Large-Scale Scientific Computing, Springer, 2005, pp. 509–517.
[36] E.A. Alshina, N.N. Kalitkin, S. Panchenko, Numerical solution of boundary value problems in unlimited area, Mat. Model. 14 (2002) 10–22.
[37] R. Fazio, A. Jannelli, Finite difference schemes on quasi-uniform grids for BVPs on infinite intervals, J. Comput. Appl. Math. 269 (2014) 14–23.
[38] J. Urizarna-Carasa, L. Schlegel, JulioUri/CFD_Numerics-for-the-Maxey-Riley- Equation: v1.1 - new submission, https://doi.org/10.5281/zenodo.14226715, 2024.
[39] S.K. Lele, Compact finite difference schemes with spectral-like resolution, J. Comput. Phys. 103 (1992) 16–42.
[40] J. Malele, P. Dlamini, S. Simelane, Highly accurate compact finite difference schemes for two-point boundary value problems with Robin boundary conditions, Symmetry 14 (2022) 1720.
[41] P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S.J. van der Walt, M. Brett, J. Wilson, K.J. Millman, N. Mayorov, A.R.J. Nelson, E. Jones, R. Kern, E. Larson, C.J. Carey, İ. Polat, Y. Feng, E.W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E.A. Quintero, C.R. Harris, A.M. Archibald, A.H. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0 contributors, SciPy 1.0: fundamental algorithms for scientific computing in python, Nat. Methods 17 (2020) 261–272.
[42] C.T. Kelley, Solving Nonlinear Equations with Newton's Method, SIAM, 2003.
[43] D.A. Knoll, D.E. Keyes, Jacobian-free Newton–Krylov methods: a survey of approaches and applications, J. Comput. Phys. 193 (2004) 357–397.
[44] A.H. Baker, E.R. Jessup, T. Manteuffel, A technique for accelerating the convergence of restarted GMRES, SIAM J. Matrix Anal. Appl. 26 (2005) 962–984.
[45] C.A. Kennedy, M.H. Carpenter, Diagonally implicit Runge-Kutta methods for ordinary differential equations. A review, Technical Report, 2016.
[46] U.M. Ascher, S.J. Ruuth, R.J. Spiteri, Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations, Appl. Numer. Math. 25 (1997) 151–167.
[47] I. Rypina, M.G. Brown, F.J. Beron-Vera, H. Koçak, M.J. Olascoaga, I. Udovydchenkov, On the Lagrangian dynamics of atmospheric zonal jets and the permeability of the stratospheric polar vortex, J. Atmos. Sci. 64 (2007) 3595–3610.
[48] A. Hadjighasem, D. Karrasch, H. Teramoto, G. Haller, Spectral-clustering approach to Lagrangian vortex detection, Phys. Rev. E 93 (2016) 063107.
[49] R. Colombi, N. Rohde, M. Schlüter, A. von Kameke, Coexistence of inverse and direct energy cascades in Faraday waves, Fluids 7 (2022) 148.
[50] R. Colombi, M. Schlüter, A.v. Kameke, Three dimensional flows beneath a thin layer of 2d turbulence induced by Faraday waves, Exp. Fluids 62 (2021) 1–13.
[51] K. Padberg-Gehle, C. Schneide, Network-based study of Lagrangian transport and mixing, Nonlinear Process. Geophys. 24 (2017) 661–671.
[52] P.V. Sharma, Environmental and Engineering Geophysics, Cambridge University Press, 1997.
[53] R. Frank, J. Schneid, C.W. Ueberhuber, Order results for implicit Runge–Kutta methods applied to stiff systems, SIAM J. Numer. Anal. 22 (1985) 515–534.
[54] J. Urizarna-Carasa, An efficient numerical implementation for the Maxey-Riley Equation, Ph.D. thesis, Hamburg University of Technology, 2024, in preparation.
[55] P. Kopper, A. Schwarz, S.M. Copplestone, P. Ortwein, S. Staudacher, A. Beck, A framework for high-fidelity particle tracking on massively parallel systems, Comput. Phys. Commun. 289 (2023) 108762.
[56] J. Bisgaard, M. Muldbak, S. Cornelissen, T. Tajsoleiman, J.K. Huusom, T. Rasmussen, K.V. Gernaey, Flow-following sensor devices: a tool for bridging data and model predictions in large-scale fermentations, Comput. Struct. Biotechnol. J. 18 (2020) 2908–2919.