Received: 8 October 2022 | Accepted: 24 November 2022

DOI: 10.1002/pamm.202200242

A Python toolbox for the numerical solution of the Maxey-Riley equation

Julio Urizarna-Carasa[1,*], **Daniel Ruprecht**[1,**], **Alexandra von Kameke**[2,***], and **Kathrin Padberg-Gehle**[3,†]

[1] Hamburg University of Technology, Institute of Mathematics, Chair Computational Mathematics.
[2] Hamburg University of Applied Sciences, Department Maschinenbau und Produktion.
[3] Leuphana Universität Lüneburg, Institute of Mathematics and its Didactics, Applied Mathematics Group.

The Maxey-Riley equation (MRE) models the motion of a finite-sized, spherical particle in a fluid. It is a second-order integro-differential equation with a kernel with a singularity at initial time. Because solving the integral term is numerically challenging, it is often neglected despite its often non-negligible impact. Recently, Prasath et al. showed that the MRE can be rewritten as a time-dependent heat equation on a semi-infinite domain with a nonlinear, Robin-type boundary condition. This approach avoids the need to deal with the integral term. They also describe a numerical approach for solving the transformed MRE based on Fokas method. We provide a Python toolbox implementing their approach, verify it against some of their numerical examples and demonstrate its flexibility by computing the trajectory of a particle in a velocity field given by experimental data.

1 Introduction

The motion of inertial particles in a fluid is a fundamental problem of fluid dynamics that appears in many areas, for example spread of COVID-19 virus in the air [1], rain drop formation in clouds [2], settlement of plankton in the ocean [3] and more [4]. The Maxey-Riley equation (MRE) [5] has become an accepted model, at least for particles up to a certain size [6]. In the form used by Prasath et al. [7], the MRE reads

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

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

where $\boldsymbol{u}(\boldsymbol{y}(t),t)$ is the fluid's velocity at the particle's position $\boldsymbol{y}(t)$ and $\boldsymbol{v}(t)$ is the particle's velocity. The parameters are

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

where ρ_p, ρ_f are the particle and fluid density, respectively; a is the particle radius, T is the flow time scale, ν is the kinematic viscosity, R is the effective density ratio and S is the Stokes number. The three terms on the right hand side of (1b) correspond to the acceleration of the particle by the fluid, the Stokes drag and the influence of the lagging boundary layer forming behind the particle. This last term, involving the integral from 0 to t, is called the Basset history term. Since the history term complicates the direct numerical solution of the MRE [8], it is often neglected. However, there is both theoretical [7] as well as experimental [9] evidence that in many cases the impact of the term is not negligible.

In a recent paper, Prasath et al. [7] demonstrate that the MRE can be reformulated as a one-dimensional diffusive initial-boundary-value problem on a semi-infinite domain. This paves the way for the development of significantly more efficient numerical approaches as it sidesteps the complications arising from the history term. Prasath et al. suggest a numerical approach based on Fokas method for solving the transformed MRE and show numerical examples but do not provide software with their paper. They also only consider scenarios where the velocity field is given by an analytical expression which will often not be the case in realistic scenarios.

Contributions. This paper provides a summary of the numerical approach used by Prasath et al. for solving the transformed MRE together with an openly available Python implementation of their method[1] [10]. For verification, we reproduce three of their numerical examples. To illustrate the usefulness of our software, we also show a new example where we compute particle trajectories in a velocity field interpolated from experimental data.

* Corresponding author e-mail: julio.urizarna@tuhh.de, phone: +49 40 42878 3079
** E-mail: ruprecht@tuhh.de, phone: +49 40 42878 3279
*** E-mail: Alexandra.vonKameke@haw-hamburg.de, phone: +49 40 42875 8624
† E-mail: kathrin.padberg-gehle@leuphana.de, phone: +49 4131 677 1310

[1] https://dx.doi.org/10.5281/zenodo.7304316

2 Transformation by Prasath et al.

Prasath et al. [7] show that the MRE can be reformulated as the boundary value of the solution of the heat equation on a semi-infinite domain with a nonlinear boundary condition. They use that the history term is equivalent to the half-derivative of the relative velocity, a fact already shown by Tatom [11], which is equivalent to a Dirichlet-to-Neumann map. The Dirichlet-to-Neumann operator maps a given Neumann boundary condition for the heat equation to a Dirichlet boundary condition such that the solution is the same for both. As explained for example by Fokas [12], the initial-boundary value problems

Dirichlet boundary condition		Neumann boundary condition	
$\boldsymbol{q}_t(x,t) = \boldsymbol{q}_{xx}(x,t),$	(2a)	$\boldsymbol{q}_t(x,t) = \boldsymbol{q}_{xx}(x,t),$	(3a)
$\boldsymbol{q}(x,t_0) = \boldsymbol{0},$	(2b)	$\boldsymbol{q}(x,t_0) = \boldsymbol{0},$	(3b)
$\boldsymbol{q}(0,t) = \boldsymbol{h}(t).$	(2c)	$\boldsymbol{q}_x(0,t) = \boldsymbol{g}(t)$	(3c)

will have the same solution if

$$\boldsymbol{q}_x(0,t) = \boldsymbol{g}(t) = \left(\frac{\partial}{\partial t}\right)^{1/2} \boldsymbol{h}(t) = \left(\frac{\partial}{\partial t}\right)^{1/2} \boldsymbol{q}(0,t) = \frac{\boldsymbol{q}(t_0)}{\sqrt{\pi(t-t_0)}} + \int_{t_0}^{t} \frac{\dot{\boldsymbol{q}}(s)}{\sqrt{\pi(t-t_0)}} ds. \tag{4}$$

Note how the last term in (4) has the same form as the Basset history term in (1). Therefore, if we solve

$$\begin{aligned}
\boldsymbol{q}_t(x,t) &= \boldsymbol{q}_{xx}(x,t), & x>0, t\in(0,T], \quad (5a)\\
\boldsymbol{q}(x,0) &= \boldsymbol{0}, & x>0, \quad (5b)\\
\boldsymbol{q}_t(0,t) + \alpha\boldsymbol{q}(0,t) - \gamma\boldsymbol{q}_x(0,t) &= \boldsymbol{f}(\boldsymbol{q}(0,t),\boldsymbol{y}(t),t), & t\in[0,T], \quad (5c)\\
\dot{\boldsymbol{y}}(t) &= \boldsymbol{q}(0,t) + \boldsymbol{u}(\boldsymbol{y}(t)), & t\in[0,T], \quad (5d)\\
\lim_{t\to 0} \boldsymbol{q}(0,t) &= \boldsymbol{v}_0, \quad (5e)\\
\boldsymbol{y}(0) &= \boldsymbol{y}_0. \quad (5f)
\end{aligned}$$

with

$$\alpha = \frac{1}{RS}, \quad \gamma = \frac{1}{R}\sqrt{\frac{3}{S}}, \quad \boldsymbol{f}\left(\boldsymbol{q}(0,t),\boldsymbol{y}(t),t\right) = \left(\frac{1}{R}-1\right)\frac{D\boldsymbol{u}}{Dt} - \boldsymbol{q}(0,t)\cdot\nabla_y\boldsymbol{u}(\boldsymbol{y}(t),t),$$

then $\boldsymbol{q}(0,t)+\boldsymbol{u}(\boldsymbol{y}(t),(t)) =: \boldsymbol{v}(t)$ is a solution of (1). Since the absolute velocity $\boldsymbol{v}(t)$ has two or three components depending on whether the particle moves in a plane or in 3D, $\boldsymbol{q}$ is vector-valued and (5) consists of two or three independent diffusion problems, one for every component of the relative velocity.

3 Numerical solution of the transformed Maxey-Riley equation

The reformulation of the MRE into an initial-boundary-value problem allows to use analytical methods for classical diffusion problems [4] instead of dealing with the memory term in direct integration of (1) as is done for example by Daitche [8]. Using Fokas' method [13], Prasath et al. obtain the solution

$$\boldsymbol{q}(0,t) = -\frac{2}{\pi}\int_0^\infty k e^{-k^2(t-t_0)}\, \Im\{\, \boldsymbol{\mathcal{G}}(k,t_0)\,\} dk + \frac{2}{\pi}\int_{t_0}^{t} \boldsymbol{f}(\boldsymbol{q}(0,s),\boldsymbol{x}(s),s)\; \mathcal{L}(t-s)\; ds \tag{6}$$

of (5) where

$$\Im\{\boldsymbol{\mathcal{G}}(k,t_0)\} = \Im\left\{\frac{\gamma\hat{\boldsymbol{q}}(-k,t_0) + \boldsymbol{q}(0,t_0)}{ik\gamma + (k^2-\alpha)}\right\}, \quad \mathcal{L}(m) = \int_0^\infty \frac{k^2\gamma e^{-k^2 m}}{k^2\gamma^2 + (k^2-\alpha)^2} dk, \tag{7}$$

and $\hat{\boldsymbol{q}}(k,t)$ is the truncated Fourier transform of $\boldsymbol{q}(x,t)$, i.e. $\hat{\boldsymbol{q}}(k,t) = \int_0^\infty e^{-ikx}\boldsymbol{q}(x,t)dx$. Solution (6) consists of two terms where the second depends implicitly on the solution $\boldsymbol{q}(0,t)$. We approximate $\boldsymbol{f}$ by a truncated series of Chebyshev polynomials

$$\boldsymbol{f}(\boldsymbol{q}(0,t),\boldsymbol{x}(t),t) \approx \sum_{n=0}^{N} \boldsymbol{c}_n T_n(t), \tag{8}$$

and obtain

$$\boldsymbol{q}(0,t) = \underbrace{-\frac{2}{\pi}\int_0^\infty k e^{-k^2(t-t_0)}\,\Im\{\,\boldsymbol{\mathcal{G}}(k,t_0)\,\}dk}_{=:\boldsymbol{\mathcal{H}}(t)} + \underbrace{\frac{2}{\pi}\sum_{n=0}^{N}\boldsymbol{c}_n\int_{t_0}^{t} T_n(s)\,\mathcal{L}(t-s)\,ds}_{=:\boldsymbol{\mathcal{F}}(t)}. \tag{9}$$

Inserting the quadrature nodes t_j into (9) results in the nonlinear root-finding problem

$$\boldsymbol{\mathcal{I}}(t_j) \equiv \boldsymbol{\mathcal{H}}(t_j) + \boldsymbol{\mathcal{F}}(t_j) - \boldsymbol{q}(0,t_j) = 0, \quad t_j = t_0 + \frac{\Delta t}{2}\left(1 - \cos\left(\frac{j\pi}{L}\right)\right) \tag{10}$$

for the coefficients $\boldsymbol{c}_n$ where $\Delta t = t_L - t_0$ is the length of each time step and $j = 1,\dots,L$ the index counting quadrature nodes. To solve (10), we use SciPy's Newton solver [14]. Since we do not provide a Jacobian, it reduces to a secant method. We use $\boldsymbol{\mathcal{H}}$ as initial guess, so that the iteration is initialized with $\tilde{\boldsymbol{q}}(0,t_j) = \boldsymbol{\mathcal{H}}(t_j)$. The term $\boldsymbol{\mathcal{G}}$ contains the information about the history of the particle until the initial time of the interval, i.e. $(0,t_0]$. Using the steps given in Appendix B in [7], its contribution to the following time step $(t_0, t_0+\Delta t]$ can be computed from

$$\boldsymbol{\mathcal{G}}(k,t_0+\Delta t) = e^{-k^2\Delta t}\boldsymbol{\mathcal{G}}(k,t_0) - e^{-k^2(t_0+\Delta t)}\int_{t_0}^{t_0+\Delta t} e^{k^2 s}\boldsymbol{q}(0,s)ds - e^{-k^2(t_0+\Delta t)}\frac{\tilde{\boldsymbol{f}}(k^2,t_0,t_0+\Delta t)}{\alpha - k^2 + ik\gamma}. \tag{11}$$

Therefore, the $\boldsymbol{\mathcal{G}}$ and thus $\boldsymbol{\mathcal{H}}$ term in (10) stay constant over the nonlinear iteration and only the $\boldsymbol{\mathcal{F}}$ term is updated. To manage the Chebyshev approximation (8) in the software, we rely on NumPy's Chebyshev library [15]. In particular, we use its `chebint` function to compute integrals. Only the integral in $\boldsymbol{\mathcal{F}}$ in (9) is computed with SciPy's `quad` function [16].

4 Verification

To verify that our implementation of Fokas method for the MRE is correct, we reproduce three of the numerical examples shown in Prasath et al. [7], a particle relaxing in a quiescient fluid, a particle in a Couette flow and finally a particle in a single point vortex.

4.1 Example 1: Relaxing particle

For a quiescient background flow we have $\boldsymbol{f} = 0$. In this case, the term $\boldsymbol{\mathcal{F}}$ disappears and thus

$$\boldsymbol{q}(0,t) = \boldsymbol{\mathcal{H}}(t) = -\frac{2}{\pi}\int_0^\infty k e^{-k^2(t-t_0)}\,\Im\left\{\frac{\gamma\hat{\boldsymbol{q}}(-k,t_0) + \boldsymbol{q}(0,t_0)}{ik\gamma + (k^2-\alpha)}\right\}dk. \tag{12}$$

We use the parameters from Example 1 in Prasath et al. The Stokes number is set to $S = 1$ and we consider three different particle densities: a light particle $\beta = 0.01$, a neutrally buoyant $\beta = 1$ and a heavy particle $\beta = 5$. All particles are initialised at position $(1,0)$ and have an initial velocity of $(1,0)$. We compute trajectories from $t = 0$ to $t = 15$ with a time step of $\Delta t = 0.1$ and a tolerance of 10^{-13} for the nonlinear solver. Figure 1 shows the relative velocity (left) and position (right) of the particles. The relative velocity agrees well with Figure 4 in Prasath et al. (the position is not shown there). Note that the dashed lines from their figure are missing in Figure 1 since we do not show solutions without history term.

4.2 Example 2: Couette flow

The Couette flow appears in a layer of fluid between two plates moving at different speeds. Its profile is given by $\boldsymbol{u}(\boldsymbol{x}(t)) = \left(\lambda y^{(1)}, 0\right)^T$. The integral form of the solution of (6) then becomes

$$\pi\boldsymbol{q}(0,t) = -\int_{-\infty}^{\infty} ike^{-k^2 t}\begin{bmatrix}(k^2-\alpha-ik\gamma)^{-1} & \lambda(k^2-\alpha-ik\gamma)^2\\ 0 & (k^2-\alpha-ik\gamma)^{-1}\end{bmatrix}\boldsymbol{q}(0,0)dk. \tag{13}$$

We set the same parameters as in Example 4 in Prasath et al. They do not state the used time step size but we use $\Delta t = 0.1$ for the case $S = 1$ and $\Delta t \approx 0.0043$ for the case $S = 0.01$. The density ratio is fixed to $\beta = 5$ in all cases. The chosen tolerance for the nonlinear solver is 10^{-13}. Figure 2 shows the trajectories (left) and relative velocities (right) for a particle with Stokes number $S = 0.01$ (upper) and $S = 1.0$ (lower). They correspond to the solid lines shown in the left and center plots in Figure 8 in Prasath et al. which they match well, except for our upper-left plot. However, a comparison against the analytic solution formula (13) confirms that Figure 2 agrees well with the solution for the stated parameters. Note that we do not show trajectories without the history term, which are shown as dashed lines by Prasath et al.

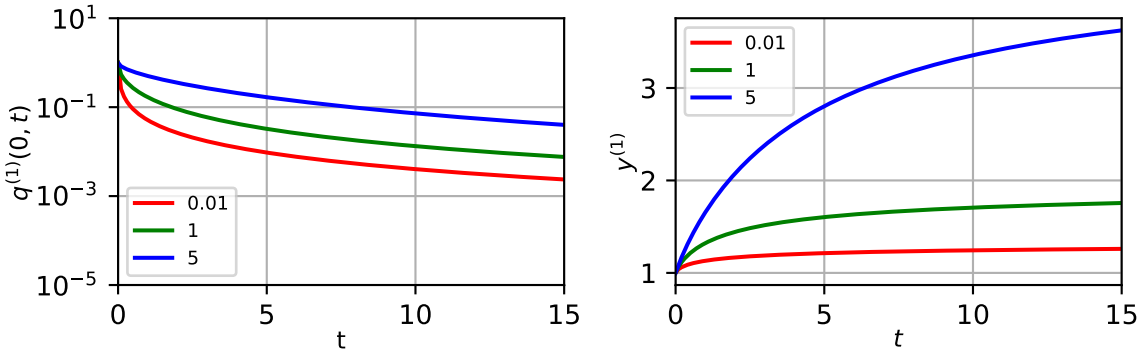

Fig. 1: Reproduction of Figure 4 by Prasath et al. [7]. Note that we do not show trajectories without the history term. Relative velocity (left) and position (right) of particles with density smaller than the fluid ($\beta = 0.01$), equal density ($\beta = 1.0$) and higher density than the fluid ($\beta = 5$) in a quiescient fluid .

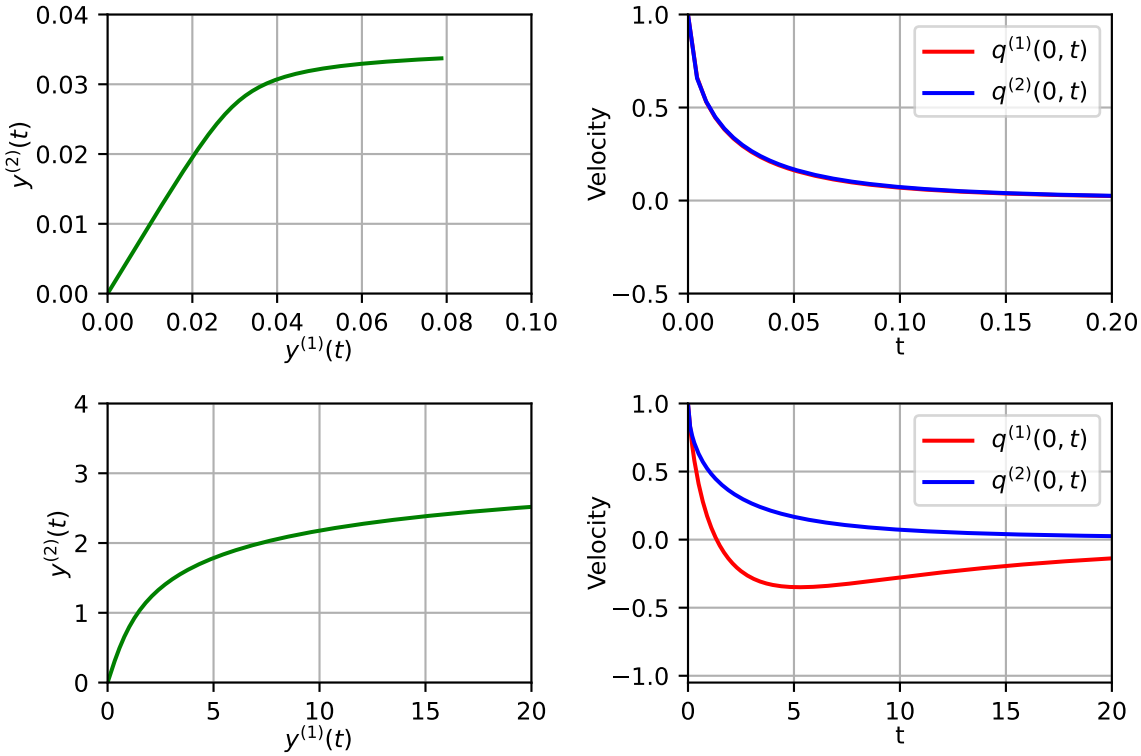

Fig. 2: Reproduction of Figure 8 in Prasath et al. [7]. Note that we do not show their sensitivity results and hence their dashed lines are missing in our plots. Particle trajectory (left) and relative velocity (right) in a Couette flow for Stokes numbers $S = 0.01$ (upper) and $S = 1.0$ (lower).

4.3 Example 3: Single point vortex

The velocity field for the single point vortex is given by

$$\boldsymbol{u} = \frac{\Gamma}{2\pi} \begin{bmatrix} -\frac{y^{(2)}}{|\boldsymbol{y}|^2} \\ \frac{y^{(1)}}{|\boldsymbol{y}|^2} \\ 0 \end{bmatrix},$$

where $\Gamma = 2\pi$ and $\boldsymbol{y} = (y^{(1)}, y^{(2)})$ is the particle's position. Since the initial position of particles is not stated by Prasath et al., we do not focus on the reproduction of the transient solutions shown in their Figure 11 but on matching the asymptotic scalings they found for the radial distance and velocity. In line with the results shown by Prasath et al., we compute the trajectories of five particles with $S = 0.01$ and $\beta = 10$, initial relative velocity $\boldsymbol{v}_0 = (0.05, 0.05)^T$ and distances from the

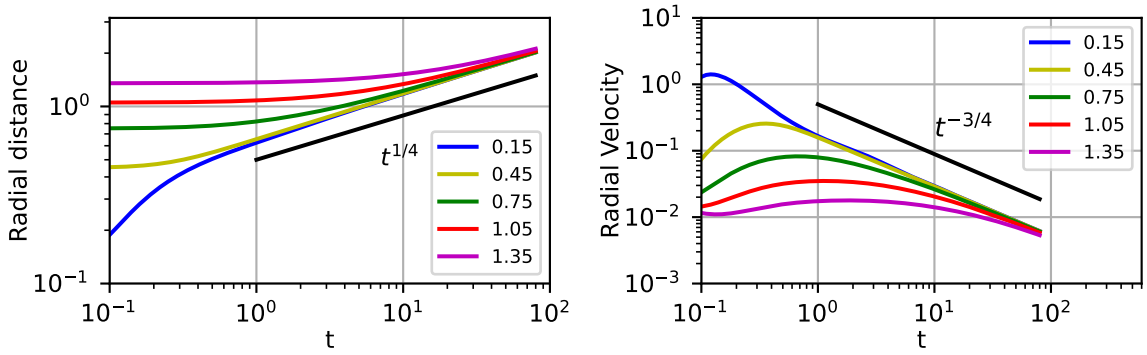

Fig. 3: Reproduction of Figure 11 in Prasath et al. [7]. Radial distance (left) and radial velocity (right) for particles starting at increasing distance from the vortex center. In agreement with their results, after an initial transient period, the radial distance increases like $t^{1/4}$ while the radial velocity decreases like $t^{-\frac{3}{4}}$.

vortex center of 0.15, 0.45, 0.75, 1.05 and 1.35. All particles lie on the horizontal axis, meaning that their initial positions are $(0, 0.15)$, $(0, 0.45)$, $(0, 0.75)$, $(0, 1.05)$ and $(0, 1.35)$. The chosen tolerance for the nonlinear solver is 10^{-7}. Figure 3 shows the radial distance and velocity for all five particles over time. In agreement with the results presented by Prasath et al., after a transient period lasting until around $t = 1$, the radial velocity decreases proportional to $t^{-\frac{3}{4}}$ while radial distance increases proportional to $t^{1/4}$.

5 Trajectories in an experimentally measured velocity field

In the verification examples above, the velocity field is given by an analytical expression. However, in most real-world cases, velocity fields will only be available from either measurements or as output from CFD models. To demonstrate that the toolbox can deal with such cases, we compute a trajectory in a thin layer of water in a circular container attached to a shaker. The shaking with a frequency of $50\,\mathrm{Hz}$ generates Faraday waves at the water surface and a turbulent space- and time-dependent 2D-velocity field. The velocity field was measured using particle image velocimetry (PIV). Details of the experimental setup are given by Colombi et al. [17, 18]. The generated data set consists of values for both velocity components, v_x and v_y, at 115×86 grid points over a roughly $7\,\mathrm{cm}$ by $5\,\mathrm{cm}$ area. A total of 1056 such snapshots are generated over a time period of $42.24\,\mathrm{s}$. SciPy's bivariate rectangular Spline interpolation is used to interpolate from data points to particle position [19].

Figure 4 shows the resulting trajectory in the x-y-plane (left) and relative velocity over time (right). The simulated particle, starting at $(x, y) = (0.02, 0.02)$ with initial velocity $\boldsymbol{v} = (0.0015, 0.0015)$, density ratio $\beta = 2$ and Stokes number $S = 0.1$, follows a loop shaped path. The trajectory is computed from $t = 0$ to $t = 5$ with a time step of $\Delta t = 0.01$ and a tolerance of 10^{-5}.

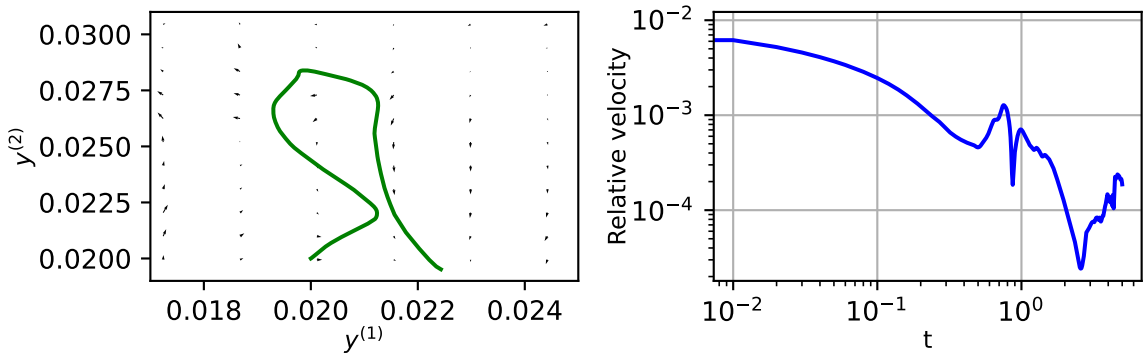

Fig. 4: Particle's trajectory in an experimentally measured velocity field in a very thin layer of fluid subjected to shaking. The shaking generates Faraday waves and a complex space- and time-dependent flow field.

Figure 5 shows two snapshots of the obtained velocity field over the whole measured area, one at initial time and another one at final time.

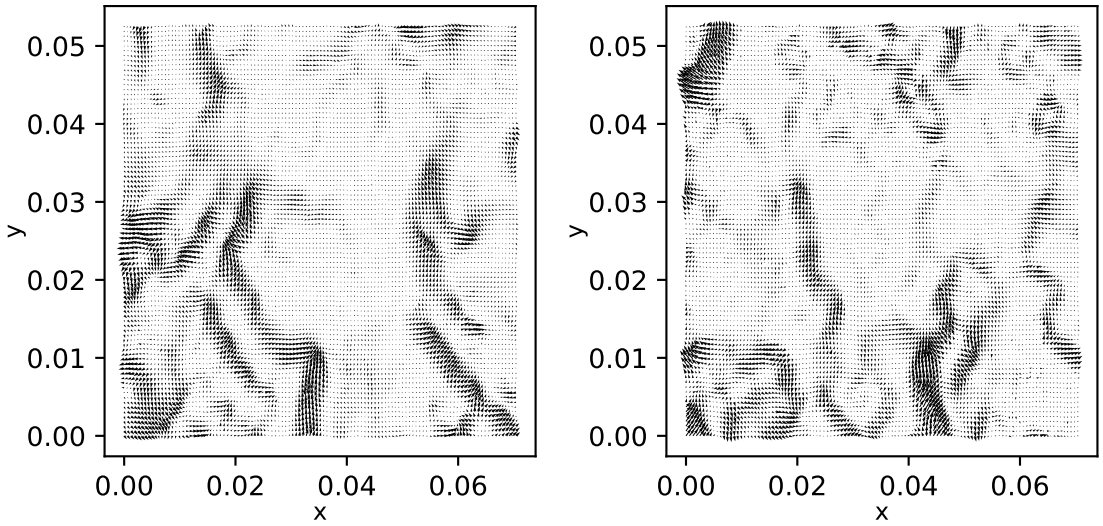

Fig. 5: Experimentally measured velocity field at initial time (left) and final time (right). The data was generated by Colombi et al. [17, 18].

6 Outlook

The reformulation of the Maxey Riley equation (MRE) as a one-dimensional diffusive partial differential equation proposed by Prasath et al. [7] provides a new approach for their numerical solution that sidesteps the difficulties arising from the Basset history term. A combination of new numerical methods and optimized implementation might allow for the solution of the Maxey-Riley equations in real-time, opening up possibilities to use them for example for closed-loop control or filtering in real-world applications. The Python software introduced in this paper provides a useful framework for such developments.

Acknowledgements Open access funding enabled and organized by Projekt DEAL.

References

[1] C. P. Cummins, O. J. Ajayi, F. V. Mehendale, R. Gabl, and I. M. Viola, Physics of Fluids **32**(8), 083302 (2020).
[2] M. Pinsky and A. Khain, Journal of Aerosol Science **28**(7), 1177–1214 (1997).
[3] M. N. Ardekani, G. Sardina, L. Brandt, L. Karp-Boss, R. N. Bearon, and E. A. Variano, Sedimentation of elongated non-motile prolate spheroids in homogenous isotropic turbulence, 2016.
[4] G. Haller, Journal of Fluid Mechanics **874**, 1–4 (2019).
[5] M. R. Maxey and J. J. Riley, The Physics of Fluids **26**(4), 883–889 (1983).
[6] F. Candelier, J. Angilella, and M. Souhar, Physics of Fluids **16**(5), 1765–1776 (2004).
[7] S. G. Prasath, V. Vasan, and R. Govindarajan, Journal of Fluid Mechanics **868**, 428–460 (2019).
[8] A. Daitche, Journal of Computational Physics **254**, 93–106 (2013).
[9] A. Daitche, Journal of Fluid Mechanics **782**, 567–593 (2015).
[10] JulioUri, Juliouri/maxey_riley: Submission for pamm paper, October 2022.
[11] F. B. Tatom, Applied Scientific Research **45**, 283 – 285 (1988).
[12] A. S. Fokas, A unified approach to boundary value problems (SIAM, 2008).
[13] A. S. Fokas, Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences **453**(1962), 1411–1443 (1997).
[14] scipy.optimize.newton, Accessed 8 October 2022.
[15] numpy.polynomial.chebyshev, Accessed 8 October 2022.
[16] scipy.integrate.quad, Accessed 8 October 2022.
[17] R. Colombi, N. Rohde, M. Schlüter, and A. von Kameke, Fluids **7**(5), 148 (2022).
[18] R. Colombi, M. Schlüter, and A. von Kameke, Experiments in Fluids **62**(1), 8 (2020).
[19] scipy.interpolate.rectbivariatespline, Accessed 8 October 2022.

