Computational methods for tracking inertial particles in discrete incompressible flows.

Benjamin K Tapley\textsuperscript{a,*}, Helge I Andersson\textsuperscript{b}, Elena Celledoni\textsuperscript{a}, Brynjulf Owren\textsuperscript{a}

\textsuperscript{a}Department of Energy and Process Engineering, The Norwegian University of Science and Technology, 7491 Trondheim, Norway
\textsuperscript{b}Department of Mathematical Sciences, The Norwegian University of Science and Technology, 7491 Trondheim, Norway

Abstract

Calculating trajectories of small particles in numerical simulations of incompressible fluids is of great importance for natural and industrial applications, yet it is a difficult and computationally expensive challenge. The problem involves interpolating the fluid velocity field and its derivatives onto the location of the particle, calculating forces and torques, then integrating a set of rigid-body equations, hence this amounts to an interesting challenge from a numerical point of view. In this paper we investigate some computational methods for addressing this problem, including using regularised Stokeslet solutions to the steady Stokes equations to approximate the local fluid field around the particle. We show a simple equivalence between regularised Stokeslets and matrix-valued radial basis function (MRBF) interpolation, which is a well posed interpolation method. The resulting rigid body ODEs can be integrated using a splitting method that utilises the exact rigid body motion of the particle. We show numerically, for a variety of Stokes regimes, that the proposed interpolation and integration algorithm reduces error for trajectories of small inertial ellipsoidal particles compared to conventional methods in discrete Taylor-Green vortices, as an example. We also conduct experiments with 10,000 particles and measure statistical quantities of the particle system as a discrete probability distribution. We show that when compared to a polynomial interpolation scheme, a cheaper MRBF scheme can lead to more accurate distributions, despite having more error, when measured in a more traditional sense. This is numerical evidence to support the claim that interpolation errors are “averaged out” in simulations of many anisotropic particles when MRBF interpolation methods are used.

\textsuperscript{*}Corresponding author

\textit{Email addresses:} Benjamin.Tapley@ntnu.no (Benjamin K Tapley), Helge.I.Andersson@ntnu.no (Helge I Andersson), Elena.Celledoni@ntnu.no (Elena Celledoni), Brynjulf.Owren@ntnu.no (Brynjulf Owren)
1. Introduction

In this paper, we address methods for tracking pathlines of inertial, anisotropic particles in discrete incompressible flow fields. This is a challenging issue since the particles do not follow the pathlines of the fluid elements. Often, the trajectories of millions of particles are needed to gain statistically reliable results hence the calculation of these trajectories is a computational bottle-neck of these simulations and there is a demand for cheaper methods. In particular, this problem involves interpolating the fluid velocity and its derivatives at the location of the particle, calculating the forces and torques on the particle and then integrating a resulting set of 13 coupled ordinary differential equations (ODEs) in time. Being able to conduct low-cost, large-scale studies has significant impact in the simulation of natural and environmental processes, for example, paper making, the spread of pollutants in the atmosphere, pharmaceutical processes, ice cloud dynamics, pollen dispersion, migration of plankton, spread of micro-plastics in the ocean, the movement of nano particles in the blood stream etc. (see [1] and references therein).

In application, pathlines of particles in complex flows are required that usually involve a direct numerical simulation of the Navier-Stokes equations

\[
\rho \left( \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) - \mu \nabla^2 \mathbf{u} = -\nabla p + \mathbf{F} \quad \text{and} \quad \nabla \cdot \mathbf{u} = 0, \tag{1}
\]

where \( \mathbf{u} \) is the fluid velocity, \( p \) the pressure, \( \mathbf{F} \) the body force and \( \mu \) is the fluid viscosity. This means that the background fluid field is available only at discrete points in space and time and an interpolation scheme needs to be integrated into the algorithm. A number of authors whom have carried out simulations of spherical particles in numerical turbulence have claimed that the accuracy of the interpolation method is unimportant in deriving convergent Lagrangian statistics [2,3,4,5,6,7]. In particular, it is claimed that the individual errors of particles are “averaged out” when considering statistical quantities of many particles and therefore a linear interpolation scheme is said to be sufficient. While these claims may hold merit when considering pathlines of spherical particles in weakly fluctuating flows, they are not so strongly supported when treating non-spherical particles, nor in flows that vary significantly across the grid cells. The former is due to the fact that the rotational dynamics of non-spherical particles are strongly coupled with their translational dynamics, which can have a large effect on the final position of the particle due to the rich interplay between the rotational and translational terms of the rigid-body equations. The latter is valid when simulating high Reynolds number flows, where the length scales that need to be resolved become increasingly small to resolve small fluctuations in the flow. In addition, the fluid field possesses the fundamental quality of being divergence-free, a feature that is destroyed when conventional interpolation techniques are implemented.

To fully resolve all the fluid length scales in direct numerical simulations, the grid spacing \( \Delta x \) is set to be smaller than the Kolmogorov length, which is the scale...
where viscous effects dissipate vortices and hence can be considered the smallest length scale of the flow. At this scale the Reynolds number is low and the flow can locally be considered Stokesian. An idea that we explore in this paper is to use analytic solutions to the steady Stokes equations to do interpolation on discrete solutions to equation (1). In particular, we consider an interpolation scheme where we use scaled and shifted regularised Stokeslet solutions to the Stokes equation, which were derived by Cortez [8]. A question that arises when considering a new interpolation scheme, is whether or not the scheme is well-posed. That is, whether or not an interpolating surface exists and if it is unique. This is equivalent to asking if there exists an invertible interpolation matrix. We answer this question using theory developed in the context of matrix-valued radial basis functions, which is a well-studied interpolation scheme that is able to create divergence-free vector-valued approximations to vector-valued data for almost no extra cost to a polynomial scheme. The use of matrix-valued radial basis functions have a wide and far reaching application in fluid dynamics [9, 10], deep learning [11], magnetohydrodynamics [12], engineering [13] and other areas of computational sciences, (we refer to books by Buhmann [14] and Wendland [15]) but there are less studies on their use in particle dynamics and local interpolation on fluid fields. Matrix valued radial basis functions have, however, been used to calculate pathlines of massless, spherical tracer particles and were shown to supercede polynomial interpolation in terms of cost and accuracy [16].

In addition to interpolation, we also focus on a numerical integration scheme. In recent years, many numerical analysts have turned their attention towards geometric numerical techniques for solving ODEs, a field called geometric integration [17]. While conventional numerical schemes are largely focused purely on minimising numerical error for general ODEs, geometric integration techniques exploit particular geometric features and structures of the specific ODE under study to produce solutions that are qualitatively more accurate than conventional schemes. These geometric methods have seen great success over all-purpose multi-stage and multi-step methods, due to their ability to conserve and exploit particular features of the problem. The ODEs governing the dynamics of inertial particles in viscous flows carry features that make them suitable to geometric treatment, such as simple forcing terms and rigid-body motion. When considered separately, these two terms can be integrated exactly, which makes the resulting ODE a perfect arena for a splitting method that creates a numerical flow by composition of exact, sub-flows of the free rigid-body motion and fluid forces.

Provided a suitable model for calculating the forces and torques on the particle exist, the above methods are adaptable for arbitrarily shaped particles in a wide range of flows. In this paper, we will conduct experiments on the particles that can be modelled as small point-ellipsoids where the forces are due to Brenner [18] and the torques are due to Jeffery [19], however we note that the method’s applicability extends to other particle models, such as slender body models [20] etc.
In the following section, we will review interpolation methods using polynomials, scalar and matrix radial basis functions, and regularised Stokeslets. The section concludes by showing a simple equivalence between the latter two. The third section reviews the dynamics of inertial anisotropic particles immersed in viscous fluids, which leads into the formulation of the splitting method for integrating the resulting ODEs. We then focus our attention to a specific particle model, namely a well-studied small inertial rigid ellipsoid model for particles in creeping Stokes flow. The penultimate section is dedicated to numerical simulations in a discrete fluid velocity field generated by a stationary 3D Taylor-Green vortex. The errors of the methods are measured for a wide range of Stokes numbers and simulations with many particles are carried out. The final section is dedicated to conclusions and future work.

2. Interpolation methods

To accurately specify the force and torque terms on the particle one needs to accurately represent the fluid velocity and often its derivatives at the location of the particle. The fluid field is available only at discrete points in space (usually on a regularly spaced grid) and therefore an interpolation scheme must be employed. Many authors [2, 3, 4, 5, 6, 7, 21, 22, 23, 24, 25, 26] use some form of polynomial interpolation due to its simplicity and accuracy. Typically, one uses a cube consisting of the $n^3$ nearest data points $\{u_i, x_i\}_{i=1}^{n^3}$ around the particle, where $u_i = u(x_i)$ is the fluid vector at the grid node located at $x_i = (x_i, y_i, z_i)^T$. As some models require the local fluid derivatives, for example the torque on an ellipsoid due to Jeffery [19], one is faced with the problem of also accurately representing the local derivatives of the velocity field at the location of the particle. One could create approximations to the fluid velocity Jacobian $\nabla u$ at the grid nodes using, for example, a finite-difference approximation, then perform nine separate interpolations on each component of $\nabla u$. Another option is to create an interpolating surface for the velocity field and then take exact derivatives. The latter option is about nine times faster, but less accurate, especially when the interpolation scheme is based on polynomials, where taking derivatives reduces the polynomial order by one, resulting in slower convergence of the derivatives [27]. On the other hand, radial basis function interpolation is based on creating an interpolating surface that is a linear combination of positive definite functions that can have infinitely many non-vanishing derivatives and have been shown to provide more accurate approximations to derivatives of data compared to conventional polynomial interpolation schemes [14, 15]. In addition, by using matrix-valued radial basis functions it is possible to reconstruct divergence-free surfaces from divergence-free data. This provides an advantage over polynomial interpolation where it is not a straightforward task to construct efficient divergence-free interpolating surfaces. Another advantage that is immediately clear is that radial basis functions are able to do interpolation on scattered data, which makes for a seamless transition into particle-laden simulations involving non-regular computational grids. Examples are simulations of flows such as turbulent boundary
layers, wall bounded turbulent flows or flows around complex solid objects where variable grid spacing and shapes are needed to resolve flow around such geometries. In these cases, polynomial interpolation becomes more cumbersome, whereas radial basis functions can be used in exactly the same way as with a regularly spaced grid.

We will now review conventional tri-polynomial interpolation, which will be used as a benchmark method in the following numerical experiments, then both scalar and matrix-valued radial basis functions are presented. We will then further motivate the use of matrix-valued radial basis functions by explaining their connection with solutions to the Stokes equations and the method of regularised Stokeslets. We refer to references such as [28, 29, 30, 31] for mathematical details in radial basis functions, including proofs of well-posedness, convergence, stability and error estimates and [12, 14] for details on the implementation of matrix-valued radial basis functions.

2.1. Tripolynomial interpolation

Tripolynomial interpolation of order $s$ involves finding the coefficients of the tri-variate polynomial

$$p(x) = \sum_{i,j,k=0}^{s} c_{ijk}x^iy^jz^k.$$ 

We will consider tri-linear, -quadratic and -cubic interpolation corresponding to $s = 1, 2$ and 3 respectively. The case $s = 3$ is analysed by Lekien and Marsden [32], which we refer to for more details. The $(s + 1)^3$ coefficients are chosen such that $p(x_i) = d_i$ found by solving a linear system

$$Ac = d$$

where $c \in \mathbb{R}^{(s+1)^3}$ is a vector containing the coefficients $c_{ijk}$, $d \in \mathbb{R}^{(s+1)^3}$ is a vector containing the (scalar) interpolating data and $A_{nm}$ element of the matrix $A$ is the monomial associated with the $n$th element of $c$ and evaluated at the location of the data point of the $n$th element in $d$. This is done for each component of the fluid vector field and therefore requires the solution of three linear systems for vector-valued data.

2.2. Scalar radial basis functions

Here we will give a brief introduction to radial basis functions (RBFs). RBF interpolation differs from classical polynomial interpolation in that the interpolating surface is a linear combination of positive definite radial functions $\psi(r_i)$ centred at the grid node $x_i$, and depend only on the distance $r_i = ||x - x_i||_2$ from that node. Such a surface is represented by

$$s(x) = \sum_{i=1}^{n} \psi(r_i)c_i.$$
where the constants $c_i$ are chosen such that the surface is consistent with the data points $s(x_i) = d_i$. This is done by solving a linear system with a $n^3 \times n^3$ positive definite coefficient matrix with components $A_{ij} = \psi(||x_i - x_j||_2)$. In this way we can construct a vector-valued interpolating surface by interpolating each component of the fluid field independently.

In addition to a more accurate interpolating surface, RBF interpolation has the advantage of approximating data, with a $C^\infty$ surface that has infinitely many non-vanishing derivatives. This means that we can find good approximations of derivatives by simply taking the exact derivative of the interpolating surface. Moreover, we are not restricted to using a particular number of interpolation points. In this way we can match the accuracy of the interpolation step to the accuracy requirements of the ODE solver and hence, is more accommodating when optimising the choice of $h$ and $\Delta x$. We are not afforded this freedom with a typical polynomial method, which often requires solving a linear system of fixed size to ensure the existence of a unique interpolating polynomial.

There are a number of known functions that satisfy the requirements of a radial basis function and the most commonly used ones are listed below

| Function Type          | Function Formulation                  |
|-----------------------|---------------------------------------|
| Gaussians             | $\exp(-\epsilon r^2)$                |
| Multiquadrics         | $\sqrt{\epsilon^2 + r^2}$            |
| Inverse multiquadrics | $\frac{1}{\sqrt{\epsilon^2 + r^2}}$ |
| Inverse quadrics      | $\left(\frac{\epsilon}{\epsilon^2 + r^2}\right)^k$ for $k = 1, 3, ...$ |
| Polyharmonic splines  | $\left(\frac{\epsilon}{\epsilon^2 + r^2}\right)^k \ln(\epsilon)$ for $k = 2, 4, ...$ |

Here, $\epsilon$ is called the shape parameter and determines the “flatness” of $\psi(r)$. In general, one should choose $\epsilon$ as low as possible, which results in more accurate representations of the data, although worse conditioned linear systems.

### 2.3. Matrix valued radial basis functions

As the underlying data is divergence-free, it is desirable that our interpolating surface also satisfies this quality, however the surface constructed from the above scalar RBF formalism is not guaranteed to reproduce a divergence-free field. This is easily remedied through the use of matrix-valued RBFs (MRBFs). In a similar fashion to the scalar RBF case, constructing a MRBF interpolant involves solving a linear system to find a set of, now vector-valued, coefficients. In this way, we can calculate the three components of the interpolating surface simultaneously and have that they produce a divergence-free field. A MRBF is given by

$$\Phi(r) = (\nabla \nabla^T - \Delta 1) \psi(r)$$
for some scalar radial basis function $\psi(r)$. Then the vector-valued interpolating surface $s$ is constructed by

$$s = \sum_{i=1}^{n} \Phi(||r_i||)c_i.$$  \hspace{1cm} (2)

Taking the divergence of $\Phi(r_i)c_i$ and with the aid of the double curl vector identity in $\mathbb{R}^3$, we arrive at

$$\nabla \cdot \Phi(r_i)c_i = \nabla \cdot (\nabla^T \nabla - \Delta I)(\psi(r_i)c_i)$$
$$= \nabla \cdot (\nabla \times (\nabla \times (\psi(r_i)c_i)))$$
$$= 0,$$

as the divergence of curl is zero. The vector coefficients $c_i$ are chosen such that $s(x_i) = u(x_i)$, which amounts to solving a single $3n^3 \times 3n^3$ linear system for the $n^3$ vector-valued coefficients $c_i$ (as opposed to the scalar case where we solve three $n^3 \times n^3$ linear systems). A theorem that is proved by Lowitzsch in [30] states that if $\psi(r_i)$ is positive definite, then the distance matrix created by the elements of $\Phi(r_i)$ is also positive definite. This means that the interpolation scheme is well-posed and there exists a unique surface $s$ that interpolates the data points.

### 2.4. Interpolation with regularised Stokeslets

We have an interpolation scheme that is able to preserve the divergence-free condition of an arbitrary vector field via MRBFs, however, we are given extra knowledge of the underlying vector field that can be exploited; namely that the data is a numerical solution to the incompressible Navier-Stokes equations (equations (1)). We are only interpolating in space and hence approximating steady-state solutions to equations (1) and as the grid-spacing $\Delta x$ is comparable to the smallest length scales of the flow (i.e., the Kolmogorov scale), the Reynolds number is small and the nonlinear terms of equations (1) can be ignored. Under the above assumptions, a good approximation for the local flow in a grid-cell can be given by the steady Stokes equations, which with body force $F$, reads

$$\mu \nabla^2 u - \nabla p = -F \quad \text{and} \quad \nabla \cdot u = 0.$$ \hspace{1cm} (3)

The question we address now is: can we use analytic solutions to the Stokes equations to do interpolation on discrete solutions generated by Navier-Stokes equations? The answer can be made clear by first reviewing one solution method to equation (3), namely the method of regularised Stokeslets. Consider a point force at the origin $F = \delta(x)f_0$, where $\delta(x)$ is the delta function. Solving equation (3) by Green’s function gives rise to the Stokeslet solution, which reads

$$u = \frac{1}{8\pi\mu} \left( \frac{||x||^2 I + xx^T}{||x||^3} \right) f_0 \quad \text{and} \quad p = \frac{1}{4\pi} \frac{f_0 \cdot x}{||x||^3}.$$ \hspace{1cm} (4)
One idea is to treat Stokeslet solutions like MRBFs and compute an interpolating vector field from shifted and scaled Stokeslets, however this solution has discontinuous derivatives at the origin and therefore is less amenable to interpolation techniques unless you choose to center the Stokeslets outside the interpolation domain (which is possible, just a bit more cumbersome). One can avoid such singular solutions by regularising the Stokeslet.

We will now follow the theory presented by Cortez in [8] to develop what is called the regularized Stokeslet solution to the Stokes equations. Instead of a point body force, we now consider an approximate problem where the forces are spread out over a small ball centred around the origin. The force is maximum in the center and quickly decays to zero at the surface of the ball. This is done by replacing the body force term by \( \mathbf{F} = \phi_\epsilon(x) f_0 \), where \( \phi_\epsilon(x) \) is a radially symmetric smooth “blob” function that decays to zero at infinity and satisfies

\[
\int \phi_\epsilon(x) \, dx = 1 \quad \text{and} \quad \lim_{\epsilon \to 0} (\phi_\epsilon(x)) = \delta(x).
\]

Now define the functions \( G_\epsilon \) and \( B_\epsilon \) as the solutions to

\[
\Delta G_\epsilon = \phi_\epsilon(x) \quad \text{and} \quad \Delta B_\epsilon = G_\epsilon(x),
\]

which is a smooth approximations to Green’s function and an approximation to the solution to the biharmonic equation \( \Delta^2 B = \delta(x) \), respectively. Taking the divergence of the Stokes equation gives the following expression for the pressure term

\[
\Delta p = \nabla \cdot \mathbf{F},
\]

which has the particular solution

\[
p = f_0 \cdot \nabla G_\epsilon.
\]

This can be put back into the equation for \( \mathbf{u} \) to yield

\[
\mu \Delta \mathbf{u} = (f_0 \cdot \nabla) \nabla G_\epsilon - f_0 \phi_\epsilon,
\]

which has the particular solution

\[
\mu \mathbf{u}(x) = (f_0 \cdot \nabla) \nabla B_\epsilon(x) - f_0 G_\epsilon(x).
\]  

(5)

This is known as the regularized Stokeslet velocity due to Cortez. If there are multiple forces of magnitude \( f_\ell \) centered at the points \( x_\ell \) then the pressure and velocity are given by scaled and shifted Stokeslet velocities and pressures, which is possible due to the linearity of the Stokes equations [3].
2.5. Matrix radial basis functions as the solution to a local Stokes problem

Now consider the interpolation problem where we construct an interpolating vector-valued surface using scaled and shifted Stokeslets from equation (5). To determine if the interpolation problem is well-posed we make use of the following identity

\[(f_0 \cdot \nabla) \nabla B_\epsilon = \nabla \nabla^T (B_\epsilon f_0),\]  

which is true for any function \(B_\epsilon(x)\) and constant vector \(f_0\). This can be shown easily by writing the left hand side in index notation, where the \(i\)th component can be expressed as 
\[
\sum_{j=1}^{3} (f_j \partial_j) \partial_i B_\epsilon = \sum_{j=1}^{3} (\partial_i \partial_j) B_\epsilon f_j,
\]
which is exactly the right hand side of the identity (6). After inserting \(G_\epsilon = \nabla^2 B_\epsilon\) and making use of the above identity, we can re-write equation (5) as

\[u = (\nabla \nabla^T - I \Delta) B_\epsilon f_0\]

which is analogous to a MRBF element if we can identify \(B_\epsilon(x)\) with a positive-definite RBF \(\psi(||x||)\) and the force vectors are identified with the interpolation coefficient vectors \(f_0 = c\) from equation (2). This means that every RBF corresponds to a regularised Stokeslet solution, with force \(f_i = c_i\) if the function \(\nabla^4 \psi(r) = \phi_\epsilon(r)\) satisfies the definition of a blob function. Although the converse is not necessarily true: doing interpolation from shifted and scaled versions of the regularised Stokeslets does not necessarily correspond to a MRBF scheme, as we are not guaranteed that the problem corresponds to a RBF that is positive definite, and hence are not guaranteed that the interpolation matrix is invertible. We observe a few things here. Doing MRBF interpolation with the RBF \(\psi = r\), is equivalent to doing interpolation with scaled and shifted Stokeslets from equation (4). However, as we previously mentioned, this method leads to a singular interpolation matrix as \(r\) has discontinuous derivatives at \(r = 0\). This can be regularised by using \(\psi = \sqrt{\epsilon^2 + r^2}\), which is a Hardy multi quadric and leads to a regularised Stokeslet solution presented by Cortez in [8]. In the following experiments, we will use the \(C^\infty\) Gaussian RBF \(\psi(r) = e^{\epsilon^2 r^2}\) for the MRBF interpolation scheme. This corresponds to Stokeslet solutions with the blob function

\[
\nabla^4 \psi(r) = \phi_\epsilon = 16 \left( \epsilon^4 r^4 + 5 \epsilon^2 r^2 + \frac{15}{4} \right) e^{\epsilon^2 r^2}
\]

and the Stokeslet velocity

\[
u = (\nabla \nabla^T - I \Delta) e^{-\epsilon^2 r^2} f_0
= 4 \epsilon^2 \left( \left( 1 - \epsilon^2 r^2 \right) I + \epsilon^2 xx^T \right) e^{-\epsilon^2 r^2} f_0,
\]

which is a solution of equation (5) with \(F = \phi_\epsilon f_0\) and pressure term

\[p = 4 \epsilon^4 \left( 5 + 2 \epsilon^2 r^2 \right) e^{-\epsilon^2 r^2} (f_0 \cdot x).
\]
3. Dynamics

The dynamics of a rigid anisotropic particle immersed in a viscous fluid is described in this section. The variables $p$ and $m$ are the linear and angular momenta of the particle, the former defined in the inertial frame and latter being defined in the body frame of the particle (see figure 1). For a rigid particle with gravity $F_g$, hydrodynamic force $F_h$ and torque $N$, the momenta are determined by

$$\dot{p} = F_g + F_h,$$
$$\dot{m} = m \times (I^{-1}m) + N,$$

where $I$ is the diagonal moment of inertia tensor and the dot denotes $\frac{d}{dt}$. The particle orientation is specified using Euler parameters $q = (e_0, e_1, e_2, e_3)^T \in \mathbb{R}^4$, which satisfy the constraint $||q||_2 = 1$ and are determined by solving the ODE

$$\dot{q} = \frac{1}{2} q \cdot w,$$

where $w = (0, (I^{-1}m)^T)^T \in \mathbb{R}^4$ and $\cdot$ denotes the Hamilton product of two quaternions [33]. There is a 2-to-1 correspondence between Euler parameters and $3 \times 3$ rotation matrices given by the Euler-Rodriguez map $E : q \mapsto Q \in SO(3)$. Setting $e = (e_1, e_2, e_3)$, the rotation matrix $E(q) = Q$ is constructed via

$$Q = 1 + 2e_0 \dot{e} + 2\ddot{e},$$

Figure 1: A prolate spheroidal particle with coordinate lines of the inertial frame that is fixed in space (thick black arrows), translating frame whose origin is co-translating with the particle center of mass (thin black arrows) and the body frame that is co-translating and co-rotating with the particle (thin blue arrows).
where $\mathbb{1}$ is the $3 \times 3$ identity matrix and we have introduced the hat operator defined by $\hat{\cdot} : \mathbb{R}^3 \to \mathfrak{so}(3)$ and represents the skew-matrix form of a 3-vector, that is

$$
\begin{pmatrix}
\omega_1 \\
\omega_2 \\
\omega_3
\end{pmatrix} \mapsto \hat{\omega} =
\begin{pmatrix}
0 & -\omega_3 & \omega_2 \\
\omega_3 & 0 & -\omega_1 \\
-\omega_2 & \omega_1 & 0
\end{pmatrix},
$$

where $\mathfrak{so}(3)$ is the Lie algebra of $SO(3)$ containing $3 \times 3$ skew-symmetric matrices satisfying $\omega \times v = \hat{\omega}v$ for $\omega, v \in \mathbb{R}^3$. The Euler-Rodriguez map can be applied to equation (7) to yield the matrix ODE

$$
\dot{Q} = -\hat{\omega}Q.
$$

Finally, the particle position is found by solving

$$
\dot{x} = v.
$$

Letting $q_i$ to be the $i$th column of $Q^T$, we can write the above set of ODEs as

$$
\begin{align*}
\dot{p} &= F^T_h + F_g, \\
\dot{m} &= m \times (I^{-1}m) + N, \\
\dot{q}_i &= -\hat{\omega}q_i, \quad \text{for } i = 1, 2, 3 \\
\dot{x} &= v.
\end{align*}
$$

(8)

If we write the particle’s mechanical energy as

$$
H(y) = \frac{1}{2} p^T m^{-1} p + \frac{1}{2} m^T I^{-1} m + \frac{1}{2} m^T x^T g + \frac{1}{2} \sum_{i=1}^{3} q_i^T q_i + m x^T g,
$$

and set $y(t) = (p^T, m^T, q_1^T, q_2^T, q_3^T, x^T)^T \in \mathbb{R}^{18}$ then one can compactly write the system of differential equations (8) as

$$
\dot{y} = S \nabla H + F(y),
$$

(9)

where $S \in \mathbb{R}^{18 \times 18}$ is a skew-symmetric matrix given by

$$
S =
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & -\mathbb{1} \\
0 & \hat{m} & 0 & 0 & 0 & 0 \\
0 & 0 & -I^{-1} m & 0 & 0 & 0 \\
0 & 0 & 0 & -I^{-1} m & 0 & 0 \\
0 & 0 & 0 & 0 & -I^{-1} m & 0 \\
1 & 0 & 0 & 0 & 0 & -I^{-1} m
\end{pmatrix},
$$

and $F(y) = (F^T_h, N^T, 0, ..., 0)^T \in \mathbb{R}^{18}$ is a generalised force vector that can depend non-linearly on $y$, $u$ and its derivatives and depends on the particle model.
3.1. Splitting method

In the section 2 we addressed only spatial interpolation of the fluid velocity field and temporal interpolation is avoided to reduce cost. In practice, many authors use multi-step methods, for example Adams-Bashforth methods [4, 21, 22, 23], that require only the fluid velocity at integer multiples of the timestep $h$, therefore circumventing the costs associated with temporal interpolation of the fluid velocity field, which is needed when evaluating stage values between timesteps when using high-order Runge-Kutta methods, for example. Because of this, we would like to avoid such multi-stage methods, where multiple evaluations are needed at fractional time steps. A drawback, however, with multistep methods is that the fluid information and particle information is required from the previous time step. This multiplies the memory requirement, when compared to an explicit one stage method, like the forward Euler method, for example. Apart from multi-step methods, another promising candidate from the geometric integration community are splitting methods, which have the advantage of being explicit and have good stability properties when compared to other explicit methods. The method is constructed as follows. We split equation (9) into a rigid-body vector field and a vector field that depends on the fluid forces

\[ \dot{y} = S \nabla H := f_1(y), \]  
\[ \dot{y} = F(y) := f_2(y). \]  

The numerical flow is computed using the second-order Strang splitting method

\[ \Phi_h = \varphi_{h/2}^{[1]} \circ \varphi_{h}^{[2]} \circ \varphi_{h/2}^{[1]}, \]

where $\varphi_{h}^{[1]}$ and $\varphi_{h}^{[2]}$ represent the flow of equations (10) and (11) respectively. The flow operator $\varphi_{h}^{[1]}$ represents the exact solutions of the free rigid-body equations described in [34]. The flow operator $\varphi_{h}^{[2]}$ is found by integrating the generalised force vector $F(y)$. Whether or not an exact solution exists here depends on the the particle model, however the solution is made simpler under the fact that $q$, $x$ and $t$ are kept constant in this vector field, hence orientation, position and explicit time-dependencies of the particle forces are removed and the existence of an analytic solution only depends on how the force depends on the momenta $p$ and $m$. In the case of a rigid spheroid, which will be introduced in the following section, we have

\[ F(y) = Ay + b, \]  

where $A$ is a positive semi-definite matrix depending on the particle orientation and $b$ is a vector that depends linearly on the fluid velocity and the particle orientation, therefore both quantities are constant in the flow of $f_2(y)$. Equation (12) can therefore be integrated with the variation of constants formula. A splitting method for this specific force term was designed in [35, 36].
This splitting method is used for a number of reasons and we will now summarise the main results from [35]. When the fluid forces are small, i.e. $f_2(y) = \varepsilon \tilde{f}_2(y)$ for $\varepsilon << 1$, then the method has global order $O(\varepsilon h^2)$, a feature that conventional methods such as Adams-Bashforth are not afforded. In the presence of a stiff term, for example when the fluid term is large in magnitude $f_2(y) = \frac{1}{\varepsilon} \tilde{f}_2(y)$, then the splitting method suffers from order reduction in the stiff regime (i.e. when $h > \varepsilon$). This is not an unfavourable feature in contrast to explicit Adams-Bashforth methods which suffer from stability issues in this regime that render them unusable. Additionally, the splitting method does not require memory of the vector field from the previous step, which is an advantage when considering many particle systems. The method is of roughly the same cost as the two-step Adam-Bashforth method.

4. Particle model

For the following numerical experiments, we will conduct simulations on dilute solutions of point-like axisymmetric spheroids in a viscous fluid field. We ignore particle-particle collisions and assume the fluid field is not affected by the presence of the particles. In this section we will present a concise summary of this model and refer the reader to [22, 23, 21] for more technical details.

The surface of a spheroid is defined by the equation

$$\frac{x^2}{a^2} + \frac{y^2}{a^2} + \frac{z^2}{c^2} = 1,$$

where $a$ and $c$ are the distinct semi-axis lengths. The particle shape is characterised by the dimensionless aspect ratio $\lambda = c/a > 0$, which distinguishes between spherical ($\lambda = 1$), prolate ($\lambda > 1$) and oblate ($\lambda < 1$) particles (the latter two shapes are also called as rods and disks).

An inertial particle immersed in a fluid will experience forces on its surface that have magnitude governed by many parameters such as the particles density $\rho_p$, length $a$, fluid density $\rho_f$, kinematic viscosity $\nu$ and fluid relaxation time $\tau_f$. Hence, it is a logical step to non-dimensionalise our equations by introducing a dimensionless Stokes number. The particle Stokes number is formally defined as the ratio of the particle and fluid relaxation times $St = \tau_p/\tau_f$, where for a spherical particle the Stokes number is

$$St_0 = \frac{2Da^2}{9\nu\tau_f},$$

where $D = \rho_p/\rho_f$ is the particle-fluid density ratio. Note that this definition only depends on the particle size and inertia but not its shape. For spheroidal particles, the following shape dependent Stokes numbers are used, which are derived by Shapiro and Goldenberg [37] and Zhao, et al. [21]

$$St = \begin{cases} 
St_0 \lambda \log(\lambda + \sqrt{\lambda^2 - 1})/\sqrt{\lambda^2 - 1} & \text{for } \lambda > 1 \\
St_0 (\pi - k_0)/(2\sqrt{1 - \lambda^2}) & \text{for } \lambda < 1 
\end{cases}$$
where \( k_0 = \log((\lambda - \sqrt{\lambda^2 - 1})/(\lambda + \sqrt{\lambda^2 - 1})) \). Note that \( St \rightarrow St_0 \) as \( \lambda \rightarrow 1 \) from above or below. All equations are therefore implemented in their non-dimensional form and all parameters have dimension equal to 1.

The particle experiences a hydrodynamic drag force due to Brenner [18],

\[
F_h = QKQ^T(u - v),
\]

where \( u = u(x,t) \) is the fluid velocity evaluated at the location of the particle \( x \), \( v = p/m \) is the particle velocity. The resistance tensor \( K \) was calculated by Oberbeck [38] and is diagonal and positive definite. The gravity term is typically defined as \( F_g = (0, 0, 1 - 1/D)^T \) to account for the buoyancy force. The torque term \( N \) depends on the particle shape and the local fluid velocity derivatives, and is given in non-dimensional form by

\[
N_x = \frac{16\pi \lambda}{3(\beta_0 + \lambda^2 \gamma_0)} \left[ (1 - \lambda^2)S_{yz} + (1 + \lambda^2)(\Omega_x - \omega_y) \right],
\]

\[
N_y = \frac{16\pi \lambda}{3(\alpha_0 + \lambda^2 \gamma_0)} \left[ (\lambda^2 - 1)S_{zx} + (1 + \lambda^2)(\Omega_y - \omega_z) \right],
\]

\[
N_z = \frac{32\pi \lambda}{3(\alpha_0 + \beta_0)} (\Omega_z - \omega_z),
\]

where the dimensionless body frame strain rate tensor is \( S_{ij} = \frac{1}{2} (\partial_i u_j + \partial_j u_i) \) and the rotation rate vector is \( \Omega_i = \frac{1}{2} (\nabla \times u)_i \). The \( \lambda \) dependent parameters \( \alpha_0, \beta_0 \) and \( \gamma_0 \) are given by Siewert [39]. We define the positive definite matrices \( A_1 \) and \( A_2 \) as well as the vectors \( b_1 \) and \( b_2 \) that depend on the local fluid velocity and its derivatives such that

\[
F_h = -A_1 p + b_1,
\]

\[
N = -A_2 m + b_2.
\]

See [35] for details. The matrix \( A \) and vector \( b \) that appear in equation (12) are therefore given by \( A = \text{diag}(A_1, A_2, 0, ..., 0) \) and \( b = (b_1^T, b_2^T, 0, ..., 0)^T \). Note that the fluid terms \( F_h \) and \( N \) have magnitude that is inversely proportional to \( St \), that is \( |F_h| = \mathcal{O}(\frac{1}{St}) \) and \( |N| = \mathcal{O}(\frac{1}{St}) \).

5. Numerical experiments

In the following we will compare a number of solution methods that use either matrix radial basis functions from the \( n \times n \times n \) nearest data points (MRBF(\( n-1 \))) or order \( n-1 \) tripolynomials (TP(\( n-1 \))) for interpolation and either the splitting method (SP2) or the Adams-Bashforth two-step method (AB2) for integration. We will compare the methods against a reference solution that uses exact interpolation of the analytic fluid field and a classical fourth order Runge-Kutta method for time integration with a comparatively small timestep.
In this section we will use a discrete fluid field generated by a stationary Taylor-Green vortex solution \cite{40} to the Navier-Stokes equations (1), given by

\[ u = 2 \cos(2 \pi x) \sin(2 \pi y) \sin(2 \pi z), \]
\[ v = - \sin(2 \pi x) \cos(2 \pi y) \sin(2 \pi z), \]
\[ w = - \sin(2 \pi x) \sin(2 \pi y) \cos(2 \pi z), \]

with uniform sampling in each direction $\Delta x = \Delta y = \Delta z$. A representation of the three-dimensional fluid field is given in figure 2a.

For the rest of the paper, the RBF shape parameter is set to $\epsilon_1 = 0.31$, $\epsilon_2 = 0.23$ and $\epsilon_3 = 0.16$ corresponding to the MRBF1, MRBF2 and MRBF3 schemes and the discrete fluid field has a grid spacing of $\Delta x = 0.1$. These values are chosen empirically to coincide with the lowest error when compared to the exact Taylor-Green solution.

5.1. Computational cost

Here we will briefly summarise the costs associated with the above algorithms. The main cost associated with the TP interpolation method is solving a linear system of size $3n^3 \times 3n^3$. In \cite{32}, Lekien and Marsden give an exact expression for the inverse coefficient matrix for the tri-cubic interpolation scheme, which can be used to solve the linear system. The same can be done for the MRBF coefficient matrix on a regular grid, as this matrix has a block skew-symmetric structure, which can also be directly inverted exactly by using an algorithm by Akaike \cite{41}, hence both methods are afforded the same algorithmic speed-ups if one desires. However these techniques
are not implemented here and instead Gaussian elimination is used for simplicity and due to the fact that the systems are not so large (at most $192 \times 192$ for the MRBF3 and TP3 schemes). Other than this, MRBF interpolation involves evaluation of more complex basis functions (exponentials or fractional powers of polynomials), which is slightly more costly than evaluating polynomials for TP interpolation. This cost is small compared to that of solving the linear system, especially for the MRBF1 and MRBF2 schemes, which have similar cost to TP1 and TP2, respectively.

The Adams-Bashforth method requires memory of the particle dynamics and the fluid information at the previous timestep, which can lead to significant additional memory requirements when large numbers of particles are being used. Other than this the SP2 and AB2 methods are roughly equal in cost. Hence, the main cost of the algorithms are determined by the size of the interpolation stencil.

5.2. Interpolation errors

In this section we will directly compare the interpolation errors of the TP and MRBF schemes using quadratic polynomials ($n = 3$) as an example. The Taylor-Green fluid field from equation (13) is discretised with $\Delta x = 0.1$ and the fluid vectors are interpolated at $100^2$ points on the faces defined by $0 \leq x \leq 1$, $0 \leq y \leq 3/4$ and $0 \leq z \leq 1$ (same as the box in figure 2a) using the TP2 and MRBF2 methods. The error of the interpolated vector field $\mathbf{u}$ relative to the exact solution $\mathbf{u}_e$ is computed, using the 2-norm, by $||\mathbf{u} - \mathbf{u}_e||_2$. The fluid Jacobian is computed by taking the exact derivative of the interpolating function and the error is computed by $||\nabla \mathbf{u} - \nabla \mathbf{u}_e||$ where $|| \cdot ||$ here is the matrix 2-norm. These errors are plotted in figure 3. We see that the MRBF error is about an order lower than the TP2 solution for both fluid vector and Jacobian error. In addition, there are sharp jumps in the fluid velocity error across the grid cells. This is more closely examined in figure 3 which shows the value $||\mathbf{u} - \mathbf{u}_e||_2$ along the line $0 \leq x \leq 1$ at $y = \frac{5\Delta x}{2}$ and $z = \frac{3\Delta x}{2}$ (i.e. the black line shown in figure 2a). We observe discontinuities in error of both solutions, however the MRBF2 vector field has noticeably higher level of regularity compared to the TP2 field, which is expected to reduce numerical noise during simulations as particles traverse grid cells.

5.3. Ring of particles

In this experiment, 1000 disk-like particles are placed in a ring of radius 0.05, centered about the point $\mathbf{x}_0 = \left[\frac{2}{3}, \frac{2}{3}, \frac{2}{3}\right]$. The Stokes number and aspect ratio are $St = 5$ and $\lambda = \frac{1}{2}$. Five methods are used to calculate the path of the 1000 particles: TP1+AB2, TP2+AB2, TP3+AB2, MRBF1+SP2 and MRBF2+SP2. The aim of this experiment is to first gain a qualitative understanding of the particle-fluid system under study and how the interpolation methods effect the resulting particle positions. Figure 4 shows snapshots in the interval $T = [0, 6]$ of the particles from the MRBF2+SP2 solution (red dots) and the reference solution (black line) at four moments in time. The ring of particles are distorted due to the fluid forces as well
Figure 3: The errors of the interpolated vector fields at $100^2$ points on the faces defined by $0 \leq x \leq 1$, $0 \leq y \leq 3/4$ and $0 \leq z \leq 1$. Note the difference in scales on the colour bars.
as due to numerical errors from the interpolation and integration. We assume that the reference solution has insignificant numerical error and that all distortions here are purely due to the fluid. Here, the reference solution and the MRBF2+SP2 are almost indistinguishable over the interval.

We now turn our attention to the figure 5 which shows the final $t = 6$ snapshot of the previous experiment using the other solution methods. There is a large mismatch between the position of the TP1+AB2 solution and the reference solution. In addition, the TP1+AB2 solution does not produce an evenly spaced distribution of particles and we instead see gaps (discontinuities) in the final position where particles in different grid cells erroneously travel at significantly different speeds. These discontinuities roughly correspond to the location of the cell faces and are also observed in the TP2+AB2 solution but to a lesser extent. We note that the TP2+AB2 method otherwise does a significantly better job at preserving the overall shape of the ring. At a visual level, the TP3+AB2 and MRBF1+SP2 solutions are closer to the reference solution and exhibit an even concentration of particles along the ring. This can be explained by the fact that TP3 interpolated surfaces are $C^1$ continuous between cell faces [32], as opposed to only $C^0$ for TP1 and TP2 interpolation. It isn’t clear how smooth MRBF surfaces are over cell faces but figure 2b shows that MRBF can be smoother than the TP interpolation of equal order which could explain why the MRBF1+SP2 solution does a much better job than the conventional method of equal cost, i.e., the TP1+AB2 solution.

The divergence of the interpolated surface is measured at the location of the particle and their values are plotted in figure 6. For the TP interpolated vector fields, the average divergence error is reduced by about an order of magnitude as the interpolating polynomial order is increased from linear to cubic, however the errors are still very significant compared to the MRBF interpolation surface, which has divergence at machine precision.

5.4. Direct measure of error in time

We now turn to some experiments to directly quantify the error of the algorithms. In the next simulation, $N_p = 100$ particles with random positions and orientations are placed in the unit cube $\Omega = [0,1]^3$ and their dynamics is calculated over an interval of time. The experiment is repeated for two choices of $\lambda$ and $St$ that correspond to a stiff and perturbed fluid-particle system. The components of the dynamics are strongly coupled, that is, the position of the particles are dependent on the forces, which is in turn affected by the orientation and torque. A good measure of the numerical algorithm’s accuracy can therefore be measured as the difference in position of the particle and a reference solution as errors in the orientation, angular and linear momenta will show up in the position. Taking the average over the $N_p$ particles yields the error expression

$$\text{error} = \frac{1}{N_p} \sum_{i=1}^{N_p} ||x_i - x_i^{\text{ref}}||_2.$$
Figure 4: The positions of a ring of 1000 particles in the interval at times $t = 2, 4, 5$ and 6. The red dots represent the positions of particles calculated by the MRBF2+SP2 method and the black line is the reference solution.
Figure 5: The position of 1000 particles calculated using different methods (coloured dots) and the reference solution (black line) at $t = 6$.

Figure 6: Average divergence of interpolated vector field averaged over the locations of the 1000 particles.
Figure 7: The error of the solution methods for two particle-fluid systems.

Figure 7 shows the error as a function of time for the three simulations. Note that when comparing algorithms in terms of computational cost, one should look at the order of interpolation, for example TP1 solutions are roughly equal cost as the MRBF1 solutions, which are plotted with the same marker face shape. There are many observations to be made here. First, the error of the stiff simulation is more dependent on the fluid behaviour and so the error fluctuates with the fluid. It can be seen, however that the MRBF solutions and the TP3 solution with SP2 integration are all very competitive, while the AB2 methods suffer from stability issues from the stiffness. Second, it is clear that, generally, the methods involving either MRBFs or SP2 integration achieve more accurate solutions. In both cases, the TP1+AB2 and TP2+AB2 solutions are amongst the worst performers. While the MRBF solutions outperform the TP solutions of equal cost.

5.5. Measuring the error as a function of the Stokes number

In the following, we aim to display how the methods perform for a variety of Stokes numbers. Here, 750 particles are given random initial positions and orientations in the domain $\Omega = [0, 1]^3$ and are allowed to evolve in the time interval $T = [0, 1]$. The rod-like particles have an aspect ratio of $\lambda = 3$ and a Stokes number taking one of 15 values in the range $10^{-3} \leq St \leq 10^4$ (i.e. 50 particles for each of the 15 values of $St$). Figure 8 presents the average error in terms of $St$ at the end of the time interval. The main observation here is that the SP2 algorithms (magenta and blue lines) become more accurate the larger $St$ becomes, while the AB2 methods (green and red lines) get to a certain point where the error is independent of $St$. Second, it is clear that, generally, the methods involving either MRBFs or SP2 integration achieve more accurate solutions. In both cases, the TP1+AB2 and TP2+AB2 solutions are amongst the worst performers. While the MRBF solutions outperform the TP solutions of equal cost.
is independent of $St$. Additionally, the AB2 methods see stability issues for low $St$, while the SP method retains stability for a larger range of $St$. These observations are in agreement with other observations for splitting methods applied to stiff and perturbed systems [42, 43]. In this time interval the MRBF methods give more accurate solutions than the TP solutions of equal cost.

5.6. Comparing particle distributions due to interpolation methods

Up until now we have mainly focused on the average error in the positions of individual particles. However, in practice, simulations involving many particles are carried out to derive statistical measures from distributions of particles and their individual error is less important. In particular, it is a more desirable quality of an algorithm to be able to reproduce accurate distributions of particles rather than minimising the absolute error of each individual particle. It is argued in many studies (e.g., [6, 7, 44]) that the error from polynomial interpolation is unimportant when considering statistics due to error being “averaged out” over many particles. It is difficult to know whether or not this argument holds in general and there have been few studies conducted to confirm the validity of this argument for non-spherical particles. In this section we will show that, cheap MRBF interpolation can lead to distributions that are closer to the reference distribution compared to expensive TP interpolation, despite having worse error per particle on average. To do so we will view systems of particles as discrete probability distributions and measure the “distance” between the distribution and a reference distribution. In our context, a distribution $P = \{(x_i, w_i)\}_{i=1}^n$ is a set of equally sized cells, where $x_i$ is the location of the cell center and $w_i$ is a weight that equals the number of particles in that cell.

A natural method of comparing the distance between two probability distributions of equal size is the first Wasserstein distance (also known as the Earth Mover
Distance in the computer science community. The first Wasserstein distance be-
tween two probability distributions is denoted by $W_1(P_1, P_2)$ and is a measure of 
the cost of transporting the distribution $P_1$ into $P_2$ in the cheapest way possible, 
hence involves solving an optimisation problem. The cost is measured as the distance
between cell centers, measured in the 2-norm and weighted by the number of 
particles being transported. For mathematical details about the first Wasserstein
distance, we refer the reader to [45] and the numerical calculations are computed 
using a publicly available MATLAB code [46].

We will also compare distributions by the relative entropy (also known as the 
Kullback-Leibler divergence) [47], which is a measure of how much information is 
lost from a reference distribution $P_2$ when an approximate distribution $P_1$ is used. 
The relative entropy is calculated by

$$E(P_1, P_2) = \sum_{x_i \in \Omega_P} P_1(x_i) \log \left( \frac{P_1(x_i)}{P_2(x_i)} \right),$$

where $P(x_i) = w_i$ is the number of particles in the cell at $x_i$ and $\Omega_P$ is the support
of the two distributions. If there is an empty cell in one distribution and not the other, say at $x = x_0$ we use $P(x_0) = \delta$ for the empty cell, where $\delta = 10^{-6} \ll 1$, to avoid singularities. This penalises the approximate solution for predicting a non-zero probability of having a particle in a cell that should have zero particles according to the reference distribution.

In this section we will use the methods TP1+SP2, TP2+SP2, TP3+SP2, MRBF1+SP2, 
MRBF1+SP2 and MRBF2+SP2 to compute the path of 10,000 rod-like particles
with $\lambda = 10$. The particles are given random positions and orientations in a cube of 
width 0.05 in the domain. The experiment is done twice, once with $St = 5$ on the interval
$T = [0, 6]$ and once with $St = 1/3$ on the interval $T = [0, 3]$.

Snapshots of the distributions at the end of the intervals are presented in figures 9 and 10 where the reference solution particle locations are represented by black dots and the numerical solution by coloured dots. In figure 9 we see that the MRBF and TP3 solutions are difficult to distinguish from the reference solution, whereas there are regions of erroneous clustering in the TP1 and TP2 solutions, as seen by the regions of black dots that are vacant of coloured dots. Similar observations are seen in figure 10.

To gain a more quantitative sense of the accuracy of the distributions, the average error, first Wasserstein distance and relative entropy are calculated at the end of the simulation for each experiment and the results are presented in tables 1 and 2. Note that the values in the tables are all relative to the TP3+SP2 solution to gain a relative sense of the methods performance, for example $W_{rel}(P) = W_1(P_{ref}, P)/W_1(P_{ref}, P_{TP3})$ and similarly for $E_{rel}(P)$ and the relative error.

From table 1 the main observations are that the MRBF1+SP2 and MRBF2+SP2 solutions, despite having more error than the TP3+SP2 solution, have a lower relative entropy. The same can be said about the relative first Wasserstein distance
for the TP3+SP2 solution and hence represents a distribution that is more similar to the reference distribution. The MRBF3+SP2 solution is about 0.31 times more accurate in all measures. Similar observations are made in table 2, although here, in addition, the MRBF1+SP2 solution outperforms the TP2 solutions in all measures.

The fact that the MRBF solutions have lower first Wasserstein distances than the TP solutions, despite larger errors, is supporting evidence towards the claim that numerical errors are averaged out when doing statistics on large numbers of particles. However, this evidence is stronger when the divergence-free interpolation is used.

| $P_n$     | $E_{rel}(P_n)$ | $W_{rel}(P_n)$ | rel. error |
|-----------|----------------|----------------|------------|
| MRBF1+SP2 | 0.9147         | 1.3288         | 1.1649     |
| MRBF2+SP2 | 0.3189         | 0.9035         | 1.2140     |
| MRBF3+SP2 | 0.3171         | 0.3133         | 0.3150     |
| TP1+SP2   | 23.7032        | 10.5297        | 11.0731    |
| TP2+SP2   | 32.0349        | 16.7479        | 31.2162    |
| TP3+SP2   | 1.0000         | 1.0000         | 1.0000     |

Table 1: Experiment 1 with $St = 5$. 

Figure 9: Snapshots at $t = 6$ from experiment 1. The different interpolation methods are given as coloured dots and the reference solution as black dots. The Stokes number is $St = 5$. 


Figure 10: Snapshots at $t = 3$ from experiment 2. The different interpolation methods are given as coloured dots and the reference solution as black dots. The Stokes number is $St = 1/3$.

| $P_n$  | $E_{rel}(P_n)$ | $W_{rel}(P_n)$ | rel. error |
|--------|----------------|----------------|-------------|
| MRBF1+SP2 | 1.1131 | 1.1352 | 3.3758 |
| MRBF2+SP2 | 0.9421 | 0.6520 | 1.1263 |
| MRBF3+SP2 | 0.6851 | 0.4666 | 0.5820 |
| TP1+SP2 | 12.3333 | 12.6987 | 12.5536 |
| TP2+SP2 | 1.1226 | 2.0907 | 4.3909 |
| TP3+SP2 | 1.0000 | 1.0000 | 1.0000 |

Table 2: Experiment 2 with $St = 1/3$.

6. Conclusion

In this paper we have proposed the use of matrix-valued radial basis function integration and splitting methods for solving for inertial spheroidal particles in incompressible discrete vector fields. The methods are compared against conventional methods involving Adams-Bashforth schemes and polynomial interpolation. The methods are tested in a range of Stokes numbers that encapsulate the dynamics of a wide range of particle applications and in most cases are shown to supersede the conventional methods in terms of complexity and accuracy. In particular, we show that the splitting method gains accuracy for perturbed systems (corresponding to large Stokes numbers) and retains stability for smaller Stokes numbers where the Adams-Bashforth method blow up. The use of matrix radial basis functions allow the construction of divergence-free vector field approximations to the local fluid field for almost no extra cost. We show, using the first Wasserstein distance, that us-
ing cheap divergence-free interpolation leads to more accurate particle distributions even though the average error per particle may be larger than high-order polynomial interpolants. This is evidence to support the claim that errors are “averaged out” when looking at statistical properties of many particles but is more so supported when divergence-free interpolation is used. This good collective particle behaviour can be somewhat argued from the fact that the particles see an interpolated vector field that is the exact solution to a Stokes problem, which we have shown arises from the method of regularised Stokeslets. On the other hand, we observe erroneous clustering and spreading of particles when tri-linear and tri-quadratic interpolation is used in our discrete Taylor-Green flow experiments. This can be circumvented by using the more expensive tri-cubic interpolation or a cheap matrix-valued radial basis function interpolation. We have no reason to believe that the methods proposed in this paper would not easily extend to numerical turbulence as obtained in a fully resolved direct numerical simulation.

7. Acknowledgements

This work has received funding from the European Unions Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement (no. 691070) as well as the SPIRIT project (no. 231632) under the Research Council of Norway FRIPRO funding scheme. The authors would like to thank the Isaac Newton Institute for Mathematical Sciences for support and hospitality during the programme Geometry, compatibility and structure preservation in computational differential equations when work on this paper was undertaken. This work was supported by: EPSRC grant number EP/R014604/1.

References

References

[1] G. A. Voth, A. Soldati, Anisotropic particles in turbulence, Annual Review of Fluid Mechanics 49 (2017) 249–276.

[2] J. W. Deardorff, R. L. Peskin, Lagrangian statistics from numerically integrated turbulent shear flow, The Physics of Fluids 13 (3) (1970) 584–595. arXiv:https://aip.scitation.org/doi/pdf/10.1063/1.1692964 doi:10.1063/1.1692964
URL https://aip.scitation.org/doi/abs/10.1063/1.1692964

[3] J. J. Riley, G. S. Patterson, Diffusion experiments with numerically integrated isotropic turbulence, The Physics of Fluids 17 (2) (1974) 292–297. arXiv:https://aip.scitation.org/doi/pdf/10.1063/1.1694714 doi:10.1063/1.1694714
URL https://aip.scitation.org/doi/abs/10.1063/1.1694714
[4] P. S. Bernard, M. F. Ashmawey, R. A. Handler, An analysis of particle trajectories in computer-simulated turbulent channel flow, Physics of Fluids A: Fluid Dynamics 1 (9) (1989) 1532–1540. arXiv:https://doi.org/10.1063/1.857330, doi:10.1063/1.857330.
URL https://doi.org/10.1063/1.857330.

[5] L. M. Portela, R. V. A. Oliemans, Eulerian-Lagrangian DNS/LES of particle-turbulence interactions in wall-bounded flows, International Journal for Numerical Methods in Fluids 43 (9) 1045–1065. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/fld.616, doi:10.1002/fld.616.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.616.

[6] B. A. van Haarlem, The dynamics of particles and droplets in atmospheric turbulence-a numerical study, Ph.D. thesis (2000).

[7] K. D. Squires, J. K. Eaton, Preferential concentration of particles by turbulence, Physics of Fluids A: Fluid Dynamics 3 (5) (1991) 1169–1178.

[8] R. Cortez, The method of regularized stokeslets, SIAM Journal on Scientific Computing 23 (4) (2001) 1204–1225.

[9] B. Šarler, A radial basis function collocation approach in computational fluid dynamics, Computer Modelling in Engineering & Sciences 7 (2005) 185–193.

[10] H. Wendland, Divergence-free kernel methods for approximating the stokes problem, SIAM Journal on Numerical Analysis 47 (4) (2009) 3158–3179.

[11] J. Park, I. W. Sandberg, Universal approximation using radial-basis-function networks, Neural computation 3 (2) (1991) 246–257.

[12] C. P. McNally, Divergence-free interpolation of vector fields from point values - exact $\nabla \cdot b = 0$ in numerical simulations, Monthly Notices of the Royal Astronomical Society: Letters 413 (1) (2011) L76–L80.

[13] G. S. Bhatia, G. Arora, Radial basis function methods for solving partial differential equations-a review, Indian Journal of Science and Technology 9 (45) (2016) 1–18.

[14] M. D. Buhmann, Radial basis functions: theory and implementations, Vol. 12, Cambridge university press, 2003.

[15] H. Wendland, Scattered data approximation, Vol. 17, Cambridge university press, 2004.

[16] B. K. Tapley, Computing cost-effective particle trajectories in numerically calculated incompressible fluids using geometric methods, arXiv preprint arXiv:1901.05236.
[17] E. Hairer, C. Lubich, G. Wanner, Geometric numerical integration: structure-preserving algorithms for ordinary differential equations, Vol. 31, Springer Science & Business Media, 2006.

[18] H. Brenner, The stokes resistance of an arbitrary particle, Chemical Engineering Science 18 (1) (1963) 1–25.

[19] G. B. Jeffery, The motion of ellipsoidal particles immersed in a viscous fluid, Proceedings of the Royal Society of London. Series A, Containing papers of a mathematical and physical character 102 (715) (1922) 161–179.

[20] L. Ohm, B. K. Tapley, H. I. Andersson, E. Celledoni, B. Owren, A slender body model for thin rigid fibers: validation and comparisons, arXiv preprint arXiv:1906.00253.

[21] L. Zhao, N. R. Challabotla, H. I. Andersson, E. A. Variano, Rotation of non-spherical particles in turbulent channel flow, Physical review letters 115 (24) (2015) 244501.

[22] N. R. Challabotla, L. Zhao, H. I. Andersson, Orientation and rotation of inertial disk particles in wall turbulence, Journal of Fluid Mechanics 766.

[23] P. Mortensen, H. Andersson, J. Gillissen, B. Boersma, Dynamics of prolate ellipsoidal particles in a turbulent channel flow, Physics of Fluids 20 (9) (2008) 093302.

[24] S. Balachandar, M. Maxey, Methods for evaluating fluid velocities in spectral simulations of turbulence. Journal of Computational Physics 83 (1) (1989) 96 – 125. doi:https://doi.org/10.1016/0021-9991(89)90224-6. URL http://www.sciencedirect.com/science/article/pii/0021999189902246.

[25] J. McLaughlin, Numerical computation of particles-turbulence interaction. International Journal of Multiphase Flow 20 (1994) 211 – 232. doi:https://doi.org/10.1016/0301-9322(94)90073-6. URL http://www.sciencedirect.com/science/article/pii/0301932294900736.

[26] P. K. Yeung, S. B. Pope, Lagrangian statistics from direct numerical simulations of isotropic turbulence, Journal of Fluid Mechanics 207 (1989) 531586. doi:10.1017/S0022112089002697.

[27] R. Carlson, C. Hall, Error bounds for bicubic spline interpolation, Journal of Approximation Theory 7 (1) (1973) 41–47.
[28] F. J. Narcowich, J. D. Ward, Generalized hermite interpolation via matrix-valued conditionally positive definite functions, Mathematics of Computation 63 (208) (1994) 661–687.

[29] S. Lowitzsch, Error estimates for matrix-valued radial basis function interpolation, Journal of Approximation Theory 137 (2) (2005) 238–249.

[30] S. Lowitzsch, Approximation and interpolation employing divergence-free radial basis functions with applications, Ph.D. thesis, Texas A&M University (2004).

[31] E. J. Fuselier Jr, Refined error estimates for matrix-valued radial basis functions, Ph.D. thesis, Texas A&M University (2007).

[32] F. Lekien, J. Marsden, Tricubic interpolation in three dimensions, International Journal for Numerical Methods in Engineering 63 (3) (2005) 455–471.

[33] H. Goldstein, C. Poole, J. Safko, Classical mechanics (2002).

[34] E. Celledoni, F. Fassò, N. Säfström, A. Zanna, The exact computation of the free rigid body motion and its use in splitting methods, SIAM Journal on Scientific Computing 30 (4) (2008) 2084–2112.

[35] B. Tapley, E. Celledoni, B. Owren, H. I. Andersson, A novel approach to rigid spheroid models in viscous flows using operator splitting methods, Numerical Algorithms (2019) 1–19.

[36] E. Celledoni, E. H. Høiseth, N. Ramzina, Passivity-preserving splitting methods for rigid body systems, Multibody System Dynamics 44 (3) (2018) 251–275.

[37] M. Shapiro, M. Goldenberg, Deposition of glass fiber particles from turbulent air flow in a pipe, Journal of aerosol science 24 (1) (1993) 65–87.

[38] A. Oberbeck, Uber stationare flussigkeitsbewegungen mit berucksichtigung der inner reibung, J. reine angew. Math. 81 (1876) 62–80.

[39] C. Siewert, R. Kunnen, M. Meinke, W. Schröder, Orientation statistics and settling velocity of ellipsoids in decaying turbulence, Atmospheric research 142 (2014) 45–56.

[40] G. I. Taylor, A. E. Green, Mechanism of the production of small eddies from large ones, Proceedings of the Royal Society of London. Series A-Mathematical and Physical Sciences 158 (895) (1937) 499–521.

[41] H. Akaike, Block toeplitz matrix inversion, SIAM Journal on Applied Mathematics 24 (2) (1973) 234–241.
[42] R. Kozlov, A. Kvaernø, B. Owren, The behaviour of the local error in splitting methods applied to stiff problems, J. Comput. Phys. 195 (2003) 576593. doi: 10.1016/j.jcp.2003.10.011

[43] R. I. McLachlan, Composition methods in the presence of small parameters, BIT Numerical Mathematics 35 (2) (1995) 258–268.

[44] W. Uijttewaal, R. Oliemans, Particle dispersion and deposition in direct numerical and large eddy simulations of vertical pipe flows, Physics of Fluids 8 (10) (1996) 2590–2604.

[45] Y. Rubner, C. Tomasi, L. J. Guibas, The earth mover’s distance as a metric for image retrieval, International Journal of Computer Vision 40 (2) (2000) 99–121.

[46] The Earth Mover’s Distance, https://se.mathworks.com/matlabcentral/fileexchange/22962-the-earth-mover-s-distance accessed: 01-02-2019.

[47] S. Kullback, R. A. Leibler, On information and sufficiency, The annals of mathematical statistics 22 (1) (1951) 79–86.