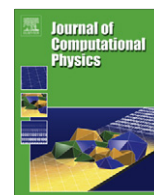




Contents lists available at ScienceDirect

Journal of Computational Physics

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

An efficient, second order method for the approximation of the Basset history force

M.A.T. van Hinsberg^{a,b}, J.H.M. ten Thijs Boonkkamp^b, H.J.H. Clercx^{a,c,*}^a Department of Physics, Eindhoven University of Technology, P.O. Box 513, 5600MB Eindhoven, The Netherlands^b Department of Mathematics and Computer Science, Eindhoven University of Technology, P.O. Box 513, 5600MB Eindhoven, The Netherlands^c Department of Applied Mathematics, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands

ARTICLE INFO

Article history:

Received 22 July 2010

Received in revised form 9 November 2010

Accepted 9 November 2010

Available online 4 December 2010

Keywords:

Basset history force

Numerical approximation

Particle laden flow

Maxey–Riley equation

Isotropic turbulence

ABSTRACT

The hydrodynamic force exerted by a fluid on small isolated rigid spherical particles are usually well described by the Maxey–Riley (MR) equation. The most time-consuming contribution in the MR equation is the Basset history force which is a well-known problem for many-particle simulations in turbulence. In this paper a novel numerical approach is proposed for the computation of the Basset history force based on the use of exponential functions to approximate the tail of the Basset force kernel. Typically, this approach not only decreases the cpu time and memory requirements for the Basset force computation by more than an order of magnitude, but also increases the accuracy by an order of magnitude. The method has a temporal accuracy of $\mathcal{O}(\Delta t^2)$ which is a substantial improvement compared to methods available in the literature. Furthermore, the method is partially implicit in order to increase stability of the computation. Traditional methods for the calculation of the Basset history force can influence statistical properties of the particles in isotropic turbulence, which is due to the error made by approximating the Basset force and the limited number of particles that can be tracked with classical methods. The new method turns out to provide more reliable statistical data.

© 2010 Elsevier Inc. Open access under the [Elsevier OA license](http://creativecommons.org/licenses/by/3.0/).

1. Introduction

The turbulent dispersion of small inertial particles plays an important role in environmental flows, and in this work we focus on small particles with densities of the same order as that of the surrounding fluid. Examples of such particles that may be present in well-mixed or in density stratified estuaries are plankton, algae, aggregates (all with densities similar to the fluid density) or resuspended sand from the sea bottom (particle densities in this case several times that of the fluid). Particle collisions and the formation of aggregates of marine particles or sediment depend on the details of the small-scale trajectories of the particles in locally homogeneous and isotropic turbulence. At these scales the details of the hydrodynamic force acting on (light) inertial particles are relevant.

Maxey and Riley [1] introduced the equation of motion for small ($d_p \ll \eta$, with d_p the particle diameter and η the Kolmogorov length scale) isolated rigid spherical particles in a nonuniform velocity field $\mathbf{u}(\mathbf{x}, t)$. An important assumption is that the particle Reynolds number $Re_p = d_p |\mathbf{u} - \mathbf{u}_p|/\nu \ll 1$, with \mathbf{u}_p the velocity of the particle and ν the kinematic viscosity of the fluid. As we consider small particle diameter and small volume fraction of particles we ignore the effect of two-way and four-way coupling. The relative importance of the terms in the hydrodynamic force depends on the ratio of particle-to-fluid

* Corresponding author at: Department of Physics, Eindhoven University of Technology, P.O. Box 513, 5600MB Eindhoven, The Netherlands. Tel.: +31 40 247 2680; fax: +31 40 246 4151.

E-mail address: h.j.h.clercx@tue.nl (H.J.H. Clercx).

density and the particle diameter. The computation of all the different forces in the Maxey–Riley equation is an expensive time- and memory consuming job. Therefore, assumptions are often made regarding the forces that can be neglected in the study of particle dispersion. The number of studies underpinning these assumptions, however, is rather limited due, for example, to the lack of efficient algorithms to take into account the effects of the Basset history force with sufficient numerical accuracy. This term was first discovered by Boussinesq in 1885. An elaborate overview of the work on the different terms in the Maxey–Riley equation and their numerical implementation can be found in the paper by Loth [2] and a historical account of the equation of motion was given in a review article by Michaelides [3].

The term most often neglected is the Basset history force because of its numerical complexity. Many recent studies underline the importance of the Basset force compared to the other forces contributions in the Maxey–Riley equation for particle transport in turbulent flows, see Refs. [4–7]. Moreover, it can affect the motion of a sedimenting particle [8] or bed-load sediment transport in open channels, where the Basset force becomes extremely important for sand particles [9,10]. It also might alter the particle velocity in an oscillating flow field [11] or modify the trapping of particles in vortices [12].

Fast and accurate computation of the Basset force is far from trivial. Although several attempts have been made [13–15], the computation of the Basset force is still far more time consuming and less accurate than the computation of the other forces in the MR equation. Therefore we present a new method that saves time, memory costs and is more accurate.

The MR equation and the subtleties with regard to the computation of the Basset history force are introduced in Section 2. Next, in Sections 3 and 4, the new method is introduced, where Section 3 focuses on the approximation of the tail of the Basset history force and Section 4 on the numerical integration of the Basset history force. Thereafter, validation of the method using analytical solutions is discussed in Section 5. A simulation of isotropic turbulence, with light inertial particles embedded in the flow, has been performed. In Section 6 we compare the results from this simulation with the new implementation of the full MR equation with the old version used by van Aartrijk and Clercx [6]. Finally, concluding remarks are given in Section 7.

2. Particle tracking

Particle trajectories in a Lagrangian frame of reference satisfy

$$\frac{d\mathbf{x}_p}{dt} = \mathbf{u}_p, \quad (1)$$

with \mathbf{x}_p the particle position and \mathbf{u}_p its velocity. According to Maxey and Riley [1] the equation of motion for an isolated rigid spherical particle in a nonuniform velocity field \mathbf{u} is given by

$$m_p \frac{d\mathbf{u}_p}{dt} = 6\pi a \mu \left(\mathbf{u} - \mathbf{u}_p + \frac{1}{6} a^2 \nabla^2 \mathbf{u} \right) + m_f \frac{D\mathbf{u}}{Dt} - (m_p - m_f) g \mathbf{e}_z + \frac{1}{2} m_f \left(\frac{D\mathbf{u}}{Dt} - \frac{d\mathbf{u}_p}{dt} + \frac{1}{10} a^2 \frac{d}{dt} (\nabla^2 \mathbf{u}) \right) + 6a^2 \rho \sqrt{\pi v} \int_{-\infty}^t K_B(t - \tau) \mathbf{g}(\tau) d\tau = \mathbf{F}_{St} + \mathbf{F}_P + \mathbf{F}_G + \mathbf{F}_{AM} + \mathbf{F}_B. \quad (2)$$

The equation of motion includes time derivatives of the form d/dt taken along the particle path and derivatives of the form D/Dt taken along the path of a fluid element. The particle mass is given by m_p , a is the radius of the particle, $\mu = \rho v$ is the dynamic viscosity, ρ and v are the density of the fluid and its kinematic viscosity, m_f is the mass of the fluid element with a volume equal to that of the particle and \mathbf{e}_z is the unit vector in the opposite direction of the gravitational force. The forces in the right-hand side of this equation denote the Stokes drag, local pressure gradient in the undisturbed fluid, gravitational force, added mass force and the Basset history force, respectively. The Faxén correction proportional to $\nabla^2 \mathbf{u}$ has been included in the Stokes drag, added mass and Basset force [16]. According to Homann et al. [17] these corrections reproduce dominant finite-size effects on velocity and acceleration fluctuations for neutrally buoyant particles with diameter up to four times the Kolmogorov scale η . For the added mass term the form described by Auton et al. [18] is used. Moreover, the history force convolution function $\mathbf{g}(t)$ and its kernel are

$$\mathbf{g}(t) = \frac{d\mathbf{f}(t)}{dt}, \quad \mathbf{f}(t) = \mathbf{u} - \mathbf{u}_p + \frac{1}{6} a^2 \nabla^2 \mathbf{u}, \quad K_B(t) = \frac{1}{\sqrt{t}}. \quad (3)$$

Eq. (2) is valid when $a \ll \eta$, but, as mentioned above, the Faxén correction can weaken this condition. Furthermore, the particle Reynolds number must be small ($Re_p \ll 1$), as are the velocity gradients around the particle. Finally, the initial velocity of the particle and fluid must be equal, if this is not the case a second term appears in the Basset history force [15]. The coupled system (1) and (2) is in principle suitable for integration by any standard method, e.g. the fourth order Runge–Kutta method.

The Basset history force \mathbf{F}_B presents additional challenges. First, the evaluation of the Basset force can become extremely time consuming and memory demanding. This is due to the fact that every time step an integral must be evaluated over the complete history of the particle. Several attempts have been made to solve this problem. Michaelides [15] uses a Laplace transform to find a novel way for computing the Basset force. This procedure can be used for linear problems, but is not suitable for space dependent velocity fields for which the coupled system (1) and (2) is nonlinear. Another solution is provided by Dorgan and Loth [14] and Bombardelli et al. [13]. In these papers the integral is evaluated over a finite window from $t - t_{win}$ until t . This can be represented by a change in the kernel of the Basset force. The window kernel is thus defined as

$$K_{\text{win}}(t) = \begin{cases} K_B(t) & \text{for } t \leq t_{\text{win}}, \\ 0 & \text{for } t > t_{\text{win}}. \end{cases} \quad (4)$$

The kernel of the Basset force is decreasing very slowly for $t \rightarrow \infty$, thus t_{win} must be chosen rather large. For Bombardelli et al. [13] this problem turned out to be less important because they used a different kernel, which decreases faster for $t \rightarrow \infty$. Although the application of the window kernel saves CPU time, the computation of the Basset force is still far more expensive than the evaluation of the other forces in the MR equation. It turns out to be approximately 100–1000 times more time consuming depending on the application.

A second issue concerns the kernel of the Basset force, which is singular for $t \rightarrow 0$. A standard approach to deal with the singularity of the Basset kernel is to employ specific quadrature rules such as the second order Euler–Maclaurin formula [19]. Another approach is presented by Tatom [20] who uses a fractional derivative method. This approach was tested by Bombardelli et al. [13]. From their results it can be easily shown that the integration method with specific quadrature rules has only temporal accuracy $\mathcal{O}(\sqrt{\Delta t})$ and that the fractional derivative approach has a temporal accuracy $\mathcal{O}(\Delta t)$. In computations of turbulent flows with particles, other discretization methods involved are at least second order. Therefore, it is not sufficient to have a first order integration method for the Basset force.

Our goal is to derive a robust and efficient method for the computation of the Basset force that overcomes all the problems mentioned above and to find an approach that is suitable for different forms of the kernel. Furthermore, our method must be stable and at least second order accurate in time. A third requirement is that it should be less time consuming and memory demanding than previous methods.

3. Approximation of the tail of the basset force

To get a better understanding of the Basset force we will first show that the contribution of this force is finite at any given time. To do this, some restrictions on $\mathbf{f}(t)$ and $\mathbf{g}(t) = \frac{d}{dt}\mathbf{f}(t)$ should be made. First, $\mathbf{f}(t)$ must be a continuous function and its derivative must exist almost everywhere. Further, $\mathbf{f}(t)$ and $\mathbf{g}(t)$ must be in the L^∞ space with norm B_1 and B_2 , respectively. The restrictions on $\mathbf{f}(t)$ and $\mathbf{g}(t)$ are thus:

$$\mathbf{f} \in C^0, \quad \|\mathbf{f}\|_\infty = B_1, \quad \|\mathbf{g}\|_\infty = B_2, \quad (5)$$

where $\|\cdot\|_\infty$ is defined as:

$$\|\mathbf{f}\|_\infty = \inf \{C \geq 0 : |\mathbf{f}(t)| \leq C \text{ almost everywhere}\}, \quad (6)$$

and $|\cdot|$ is the usual length of the vector. We assume that for particles in (turbulent) flows with $\mathbf{f}(t) = \mathbf{u} - \mathbf{u}_p + \frac{1}{6}a^2\nabla^2\mathbf{u}$ these conditions are satisfied as both the flow field and its Laplacian satisfy these conditions. With the conditions in (5) it is possible to find an upper bound for \mathbf{F}_B . The integral is split into two parts, in order to control both the singularity in the Basset kernel and the tail of the integral. This yields

$$\begin{aligned} \left| \frac{\mathbf{F}_B}{c_B} \right| &= \left| \int_{-\infty}^t K_B(t-\tau)\mathbf{g}(\tau)d\tau \right| \\ &= \left| \int_{-\infty}^{t-\frac{B_1}{B_2}} \frac{\mathbf{g}(\tau)}{\sqrt{t-\tau}}d\tau + \int_{t-\frac{B_1}{B_2}}^t \frac{\mathbf{g}(\tau)}{\sqrt{t-\tau}}d\tau \right| \\ &\leq \left| \left[\frac{\mathbf{f}(\tau)}{\sqrt{t-\tau}} \right]_{-\infty}^{t-\frac{B_1}{B_2}} - \int_{-\infty}^{t-\frac{B_1}{B_2}} \frac{\mathbf{f}(\tau)}{2(t-\tau)^{3/2}}d\tau \right| + \int_{t-\frac{B_1}{B_2}}^t \frac{|\mathbf{g}(\tau)|}{\sqrt{t-\tau}}d\tau \\ &\leq \sqrt{B_1B_2} + \frac{B_1}{2} \int_{-\infty}^{t-\frac{B_1}{B_2}} \frac{1}{(t-\tau)^{3/2}}d\tau + B_2 \int_{t-\frac{B_1}{B_2}}^t \frac{1}{\sqrt{t-\tau}}d\tau = 4\sqrt{B_1B_2}. \end{aligned} \quad (7)$$

Here $c_B = 6a^2\rho\sqrt{\pi\nu}$ is introduced for convenience. We now consider the window kernel for calculation of the Basset force $\mathbf{F}_{B-\text{win}}$. In the limit of $t_{\text{win}} \rightarrow \infty$ the difference between \mathbf{F}_B and $\mathbf{F}_{B-\text{win}}$ must vanish. Using integration by parts, one can derive

$$\left| \frac{\mathbf{F}_B - \mathbf{F}_{B-\text{win}}}{c_B} \right| = \left| \int_{-\infty}^t K_B(t-\tau)\mathbf{g}(\tau)d\tau - \int_{-\infty}^t K_{\text{win}}(t-\tau)\mathbf{g}(\tau)d\tau \right| = \left| \int_{-\infty}^{t-t_{\text{win}}} \frac{\mathbf{g}(\tau)}{\sqrt{t-\tau}}d\tau \right| \leq \frac{2B_1}{\sqrt{t_{\text{win}}}}. \quad (8)$$

The error made by using the window kernel instead of the Basset kernel is indeed becoming negligibly small for $t_{\text{win}} \rightarrow \infty$. Unfortunately, this convergence is very slow, implying that t_{win} must be very large, and a better approach for the computation of the Basset force must be found. This is done by introducing a new kernel with a modified tail, in short the modified Basset kernel $K_{\text{mod}}(t)$, as follows

$$K_{\text{mod}}(t) = \begin{cases} K_B(t) & \text{for } t \leq t_{\text{win}}, \\ K_{\text{tail}}(t) & \text{for } t > t_{\text{win}}, \end{cases}$$

$$\lim_{t \rightarrow \infty} K_{\text{tail}}(t) = 0. \quad (9)$$

This new kernel also implies a modified history force denoted by $\mathbf{F}_{B\text{-mod}}$. For now $K_{\text{tail}}(t)$ is not yet defined but must be chosen such as to approximate the Basset kernel as close as possible. Using integration by parts in the last step, the upper bound for the error induced by the modified Basset force $\mathbf{F}_{B\text{-mod}}$ becomes:

$$\begin{aligned} \left| \frac{\mathbf{F}_B - \mathbf{F}_{B\text{-mod}}}{C_B} \right| &= \left| \int_{-\infty}^t K_B(t - \tau) \mathbf{g}(\tau) d\tau - \int_{-\infty}^t K_{\text{mod}}(t - \tau) \mathbf{g}(\tau) d\tau \right| = \left| \int_{-\infty}^{t-t_{\text{win}}} (K_B - K_{\text{tail}})(t - \tau) \mathbf{g}(\tau) d\tau \right| \\ &\leq B_1 \left\{ |K_B(t_{\text{win}}) - K_{\text{tail}}(t_{\text{win}})| + \int_{t_{\text{win}}}^{\infty} \left| \frac{d(K_B - K_{\text{tail}})(t)}{dt} \right| dt \right\}. \end{aligned} \tag{10}$$

As the upper bound in relation (10) depends on t_{win} , it turns out to be beneficial to rescale the time and kernel as follows:

$$\tilde{K}_{\text{tail}}(\tilde{t}) = \frac{K_{\text{tail}}(t)}{K_B(t_{\text{win}})}, \quad \tilde{t} = \frac{t}{t_{\text{win}}}. \tag{11}$$

Applying the same scaling to $K_B(t) = 1/\sqrt{t}$ we find

$$\tilde{K}_B(\tilde{t}) = \frac{K_B(t)}{K_B(t_{\text{win}})} = K_B(\tilde{t}). \tag{12}$$

Note that this cannot be done for a general kernel. Eq. (10) can now be reformulated as

$$\left| \frac{\mathbf{F}_B - \mathbf{F}_{B\text{-mod}}}{C_B} \right| \leq \frac{B_1}{\sqrt{t_{\text{win}}}} \left\{ |1 - \tilde{K}_{\text{tail}}(1)| + \int_1^{\infty} \left| \frac{d(K_B - \tilde{K}_{\text{tail}})(\tilde{t})}{d\tilde{t}} \right| d\tilde{t} \right\}. \tag{13}$$

When comparing (8) and (13) one can see that a good approximation $\tilde{K}_{\text{tail}}(\tilde{t})$ of the tail reduces the error in (13) significantly in comparison with (8).

In order to find a good approximation $\tilde{K}_{\text{tail}}(\tilde{t})$ we start with (10). The right hand side of (10) can be minimized and thereby minimizing the error in $\mathbf{F}_{B\text{-mod}}$. When determining $K_{\text{tail}}(t)$ it is important that computation time is kept low. In order to achieve this, exponential functions are used because they can be implemented in a recursive way as explained later on. At first we start with one exponential function as follows,

$$K_{\text{tail}}(t) = a \exp(-bt). \tag{14}$$

Here a and b are two positive constants. As a first guess we require that $K_{\text{tail}}(t_{\text{win}}) = K_B(t_{\text{win}})$ and $\frac{d}{dt} K_{\text{tail}}(t_{\text{win}}) = \frac{d}{dt} K_B(t_{\text{win}})$ in order to determine a and b . In this way $K_{\text{mod}}(t)$, defined in (9), is continuously differentiable. Doing this results in

$$K_{\text{tail}}(t) = \sqrt{\frac{e}{t_{\text{win}}}} \exp\left(-\frac{t}{2t_{\text{win}}}\right). \tag{15}$$

Fig. 1 shows several kernels, where the modified Basset kernel is given by (15). The error by applying the modified Basset kernel is obviously smaller compared to the error for the window method. In order to minimize the error even more, multiple exponential functions can be used. Relation (15) provides an ansatz for the choice of a and b . Thus we write $K_{\text{tail}}(t)$ as

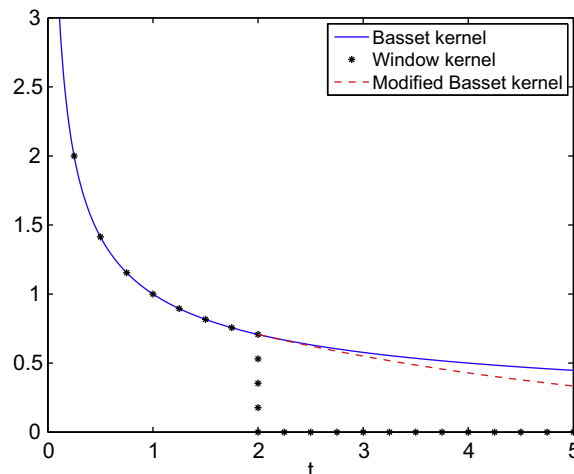


Fig. 1. Basset kernel (solid line), window kernel (dots) and the modified Basset kernel (dashed line) for $t_{\text{win}} = 2$.

$$K_{\text{tail}}(t) = \sum_{i=1}^m a_i K_i(t), \quad K_i(t) = \sqrt{\frac{e}{\tilde{t}_i}} \exp\left(-\frac{t}{2\tilde{t}_i}\right), \tag{16}$$

with a_i and \tilde{t}_i positive constants. The functions $K_i(t)$ satisfy the following properties: $K_i(\tilde{t}_i) = K_B(\tilde{t}_i)$ and $\frac{d}{dt}K_i(\tilde{t}_i) = \frac{d}{dt}K_B(\tilde{t}_i)$. Combining (11) and (16), we obtain the following dimensionless representation for the tail:

$$\tilde{K}_{\text{tail}}(\tilde{t}) = \sum_{i=1}^m a_i \tilde{K}_i(\tilde{t}), \quad \tilde{K}_i(\tilde{t}) = \sqrt{\frac{e}{\tilde{t}_i}} \exp\left(-\frac{\tilde{t}}{2\tilde{t}_i}\right), \quad \tilde{t}_i = \frac{t_i}{t_{\text{win}}}. \tag{17}$$

The coefficients a_i and \tilde{t}_i should be chosen in such a way that the upper bound in (13) is minimized. However, Newton iteration will not work for this problem, and instead we consider the expression

$$\left(1 - \tilde{K}_{\text{tail}}(1)\right)^2 + \int_1^\infty \tilde{t} \left(\frac{d(K_B - \tilde{K}_{\text{tail}})}{d\tilde{t}}\right)^2 d\tilde{t}, \tag{18}$$

which provides a good indication for the optimal values of a_i and \tilde{t}_i . In (18) an extra multiplication with \tilde{t} is introduced to correct for the change in norm. After minimizing the expression in (18), we can verify whether the error in (13) is of the same order. Since $\tilde{K}_i(\tilde{t}_i) = K_B(\tilde{t}_i)$ and $\frac{d}{d\tilde{t}}\tilde{K}_i(\tilde{t}_i) = \frac{d}{d\tilde{t}}K_B(\tilde{t}_i)$, the function $\tilde{K}_i(\tilde{t})$ approximate $K_B(\tilde{t})$ very well around \tilde{t}_i . The kernel K_B must be approximated over a large range of \tilde{t} -values and as a consequence \tilde{t}_i must also have a large range. Furthermore, K_B is changing slowly for large \tilde{t} so the small \tilde{t}_i must be close to each other whereas the large \tilde{t}_i can be far apart. The approach for finding a_i and \tilde{t}_i is thus the following. First, make a reasonable choice for \tilde{t}_i , and second, calculate a_i by minimizing (18). Finally, determine the term between brackets from (13). Another slightly different set of \tilde{t}_i -values can be chosen to see if a better approximation can be made. In Table 1 the result is shown for $m = 10$. Here one can see that some values of \tilde{t}_i are smaller than 1. This is surprising because the kernel K_B is not being approximated below $\tilde{t} = 1$. When tuning the \tilde{t}_i -values we found, however, that this improves the approximation.

From Fig. 2 it can be seen that \tilde{K}_{tail} approximates K_B relatively well over a wide range of \tilde{t} . From Fig. 3 one can see that the error decays for large \tilde{t} (note the huge range of \tilde{t} in both figures).

Table 1
Coefficients a_i and \tilde{t}_i in $\tilde{K}_{\text{tail}}(\tilde{t})$ with $m = 10$

\tilde{t}_i	a_i
0.1	0.23477481312586
0.3	0.28549576238194
1	0.28479416718255
3	0.26149775537574
10	0.32056200511938
40	0.35354490689146
190	0.39635904496921
1000	0.42253908596514
6500	0.48317384225265
50000	0.63661146557001

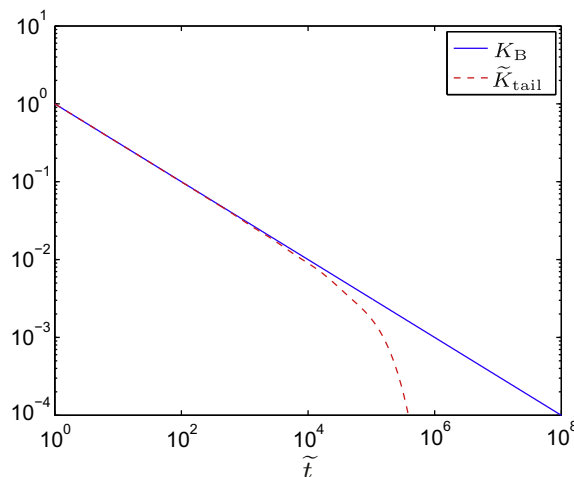


Fig. 2. The kernels $K_B(\tilde{t})$ and $\tilde{K}_{\text{tail}}(\tilde{t})$.

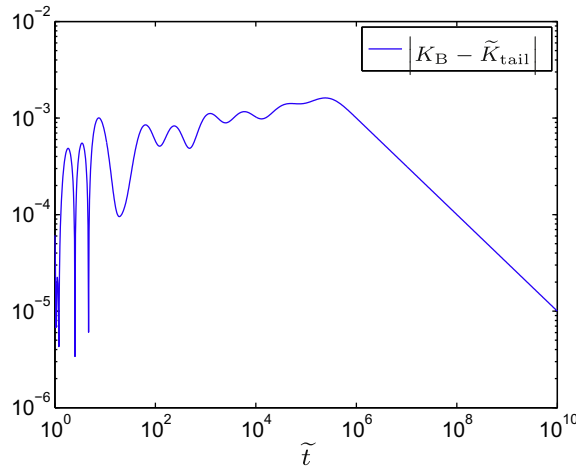


Fig. 3. The error $|K_B - \tilde{K}_{tail}|$.

Using (17) in combination with Table 1 for $\tilde{K}_{tail}(\tilde{t})$ the part between brackets in (13) can be calculated

$$|1 - \tilde{K}_{tail}(1)| + \int_1^\infty \left| \frac{d(K_B - \tilde{K}_{tail})(\tilde{t})}{d\tilde{t}} \right| d\tilde{t} \approx 9.5 \cdot 10^{-3}. \tag{19}$$

Comparing this result with the window method (8) a factor of more than 200 is gained in accuracy. When keeping the same accuracy but changing the window, t_{win} can be decreased by a factor of $200^2 = 40000$.

4. Numerical approximation

In this section the numerical integration is discussed. First, the integration of the window and tail kernels are elaborated. Second, the overall numerical scheme for solving Eqs. (1) and (2) is explained.

The integration of the Basset force with the modified kernel (9) and (16) is split into two parts, the window kernel and the tail of the kernel as follows:

$$\begin{aligned} \mathbf{F}_{B-mod}(t) &= c_B \int_{-\infty}^t K_{mod}(t - \tau) \mathbf{g}(\tau) d\tau = c_B \int_{t-t_{win}}^t K_B(t - \tau) \mathbf{g}(\tau) d\tau + c_B \int_{-\infty}^{t-t_{win}} K_{tail}(t - \tau) \mathbf{g}(\tau) d\tau \\ &= \mathbf{F}_{B-win}(t) + \mathbf{F}_{B-tail}(t). \end{aligned} \tag{20}$$

In the following, methods are described for the calculation of \mathbf{F}_{B-win} and \mathbf{F}_{B-tail} .

First, we consider the Basset force due to the window kernel \mathbf{F}_{B-win} . The kernel of the Basset force is singular for $t \rightarrow 0$ which impedes use of the ordinary trapezoidal rule. In order to deal with the singularity we introduce an alternative, trapezoidal-based method, referred to as the TB-method. The idea is as follows. The trapezoidal rule is based on linear interpolation of the integrand on each subinterval. In our approach $\mathbf{g}(t)$ is approximated by its linear interpolant $\mathbf{P}_1(t)$, and subsequently the integration of $K_B(t - \tau) \mathbf{P}_1(\tau)$ is done exactly. For the numerical implementation we start with the discretization of the interval $[t - t_{win}, t]$, given by $\tau_n = t - n\Delta t$, $n = 0, 1, 2, \dots, N$ with $\Delta t = t_{win}/N$. Now the integral can be split as

$$\mathbf{F}_{B-win}(t) = c_B \sum_{n=1}^N \int_{\tau_n}^{\tau_{n-1}} \frac{\mathbf{g}(\tau)}{\sqrt{t - \tau}} d\tau. \tag{21}$$

The next step is to approximate $\mathbf{g}(\tau)$ by its linear interpolant on each subinterval, which yields

$$\mathbf{F}_{B-win}(t) \approx c_B \sum_{n=1}^N \int_{\tau_n}^{\tau_{n-1}} \frac{\mathbf{g}_n + (\mathbf{g}_{n-1} - \mathbf{g}_n)(\tau - \tau_n)/\Delta t}{\sqrt{t - \tau}} d\tau, \tag{22}$$

where $\mathbf{g}_n \equiv \mathbf{g}(\tau_n)$. After the change of variable $\tau' = t - \tau$ this integral can be evaluated and the following result¹ is obtained:

¹ This formulation is preferred to avoid loss of significant digits in the computation of \mathbf{F}_{B-win} .

$$\begin{aligned} \mathbf{F}_{B-\text{win}}(t) &\approx \frac{4}{3} c_B \mathbf{g}_0 \sqrt{\Delta t} + c_B \mathbf{g}_N \frac{\sqrt{\Delta t} (N - \frac{4}{3})}{(N-1)\sqrt{N-1} + (N - \frac{3}{2})\sqrt{N}} + c_B \sqrt{\Delta t} \\ &\times \sum_{n=1}^{N-1} \mathbf{g}_n \left(\frac{n + \frac{4}{3}}{(n+1)\sqrt{n+1} + (n + \frac{3}{2})\sqrt{n}} + \frac{n - \frac{4}{3}}{(n-1)\sqrt{n-1} + (n - \frac{3}{2})\sqrt{n}} \right). \end{aligned} \tag{23}$$

From the result above one can see that three inner products must be calculated each time step, one inner product for each spatial dimension. One vector contains all the values \mathbf{g}_n which must be shifted by one index each time step. The other vector containing the coefficients in (23) is calculated once at the start of the computation. In this way the computational time is kept minimal. The part with \mathbf{g}_0 will be treated in a different way as explained later on in order to improve stability.

Next, the numerical integration of the tail of the Basset force is discussed. The idea is to find a recursive formulation in order to minimize computation efforts. Using expression (16) for K_{tail} , $\mathbf{F}_{B-\text{tail}}$ becomes:

$$\mathbf{F}_{B-\text{tail}}(t) = \sum_{i=1}^m a_i c_B \int_{-\infty}^{t-t_{\text{win}}} K_i(t-\tau) \mathbf{g}(\tau) d\tau = \sum_{i=1}^m a_i \mathbf{F}_i(t), \tag{24}$$

Here, \mathbf{F}_i represents the contribution of the i -th exponential function. Now \mathbf{F}_i is split into two parts, as follows.

$$\mathbf{F}_i(t) = c_B \int_{t-t_{\text{win}}-\Delta t}^{t-t_{\text{win}}} K_i(t-\tau) \mathbf{g}(\tau) d\tau + c_B \int_{-\infty}^{t-t_{\text{win}}-\Delta t} K_i(t-\tau) \mathbf{g}(\tau) d\tau = \mathbf{F}_{i-\text{di}}(t) + \mathbf{F}_{i-\text{re}}(t), \tag{25}$$

where we have to compute $\mathbf{F}_{i-\text{di}}$ directly and where $\mathbf{F}_{i-\text{re}}$ can be computed recursively. For $\mathbf{F}_{i-\text{di}}$ the same procedure is followed as with the window kernel. Using this procedure the following result² can be obtained:

$$\begin{aligned} \mathbf{F}_{i-\text{di}}(t) &\approx c_B \sqrt{\frac{e}{t_i}} \int_{t_{\text{win}}}^{t_{\text{win}}+\Delta t} \exp\left(-\frac{\tau'}{2t_i}\right) \left(\mathbf{g}_N + \frac{t_{\text{win}} - \tau'}{\Delta t} (\mathbf{g}_N - \mathbf{g}_{N+1}) \right) d\tau' \\ &= 2c_B \sqrt{e t_i} \exp\left(-\frac{t_{\text{win}}}{2t_i}\right) \left\{ \mathbf{g}_N \left[1 - \varphi\left(-\frac{\Delta t}{2t_i}\right) \right] + \mathbf{g}_{N+1} \exp\left(-\frac{\Delta t}{2t_i}\right) \left[\varphi\left(\frac{\Delta t}{2t_i}\right) - 1 \right] \right\}, \end{aligned} \tag{26}$$

where $\varphi(z) = (e^z - 1)/z = 1 + \frac{1}{2}z + \frac{1}{6}z^2 + \mathcal{O}(z^3)$. Finally, $\mathbf{F}_{i-\text{re}}$ can be easily calculated using the value of \mathbf{F}_i at the previous time step:

$$\begin{aligned} \mathbf{F}_{i-\text{re}}(t) &= c_B \int_{-\infty}^{t-t_{\text{win}}-\Delta t} \sqrt{\frac{e}{t_i}} \exp\left(-\frac{t-\tau}{2t_i}\right) \mathbf{g}(\tau) d\tau = \exp\left(-\frac{\Delta t}{2t_i}\right) c_B \int_{-\infty}^{t-t_{\text{win}}-\Delta t} \sqrt{\frac{e}{t_i}} \exp\left(-\frac{t-\Delta t-\tau}{2t_i}\right) \mathbf{g}(\tau) d\tau \\ &= \exp\left(-\frac{\Delta t}{2t_i}\right) \mathbf{F}_i(t-\Delta t). \end{aligned} \tag{27}$$

In this last part the overall numerical scheme is discussed. To solve Eqs. (1) and (2) numerically the second-order Adams-Bashforth (AB2) method is implemented. For a differential equation $\frac{d\mathbf{y}}{dt} = \mathbf{h}(t, \mathbf{y})$ the scheme reads $\mathbf{y}_{n+1} = \mathbf{y}_n + \frac{\Delta t}{2} (3\mathbf{h}^n - \mathbf{h}^{n-1})$, where $\mathbf{h}^n = \mathbf{h}(t^n, \mathbf{y}^n)$. Eq. (1) can be directly integrated with this scheme but for Eq. (2) some modifications are needed. In order to have a stable scheme, the $\frac{d\mathbf{u}_p}{dt}$ term in the added mass force is treated in an implicit way instead of explicit. Moreover, it turned out that the AB2-method has poor stability properties for the calculation of the Basset force using the window method. Extremely small time steps must be taken in order to have a stable solution. An alternative method circumventing stability problems is to bring a part of the Basset force (the contribution $\frac{d\mathbf{u}_p}{dt}$ evaluated at t) to the left hand side. Eq. (2) is then reformulated as

$$\left(m_p + \frac{1}{2} m_f + \frac{4}{3} c_B \sqrt{\Delta t} \right) \frac{d\mathbf{u}_p}{dt} = \mathbf{F}_{\text{St}} + \mathbf{F}_P + \mathbf{F}_G + \mathbf{F}'_{\text{AM}} + \mathbf{F}'_B, \tag{28}$$

with $\mathbf{F}'_{\text{AM}} = \frac{1}{2} m_f \left(\frac{d\mathbf{u}}{dt} + \frac{1}{10} a^2 \frac{d}{dt} (\nabla^2 \mathbf{u}) \right)$ and $\mathbf{F}'_B = \mathbf{F}_B - \frac{4}{3} c_B \sqrt{\Delta t} \frac{d\mathbf{u}_p}{dt}$. In this way the Basset force becomes partially implicit instead of completely explicit. Finally, as only the time derivative along the particle path $\frac{d\mathbf{u}}{dt}$ is available, the time derivative along the path of a fluid element $\frac{d\mathbf{u}}{Dt}$ is computed according to

$$\frac{D\mathbf{u}}{Dt} = \frac{\partial \mathbf{u}}{\partial t} + u_j \frac{\partial \mathbf{u}}{\partial x_j} = \frac{\partial \mathbf{u}}{\partial t} + u_{p,j} \frac{\partial \mathbf{u}}{\partial x_j} + (u_j - u_{p,j}) \frac{\partial \mathbf{u}}{\partial x_j} = \frac{d\mathbf{u}}{dt} + (u_j - u_{p,j}) \frac{\partial \mathbf{u}}{\partial x_j}. \tag{29}$$

5. Validation of the Basset force integration

In this section four test cases are presented in order to validate the methods for the integration of the Basset force. The first example tests the trapezoidal-based (TB) method and compares the results with the semi-derivative (SD) approach by Bombardelli et al. [13]. Examples 2 and 3 test the the overall numerical scheme. Here both stability and convergence are

² Note that in Eq. (26) Taylor series must be used for $\varphi(-\frac{\Delta t}{2t_i})$ when $\Delta t \ll t_i$.

tested for the explicit and the partially implicit TB-method. Finally, **Example 4** shows the efficiency of the Basset force using the tail kernel.

Example 1. Basset integral for a given convolution function

In order to demonstrate the advantages of the TB-method, the convergence of this method is compared with the SD-approach of Bombardelli et al. [13]. To that end the arbitrary test function $g(\tau) = \cos \tau$ is used. The exact Basset integral is given by

$$F_B(t) = c_B \int_0^t \frac{\cos \tau}{\sqrt{t-\tau}} d\tau = 2c_B \int_0^{\sqrt{t}} \cos(t-\sigma^2) d\sigma = c_B \sqrt{2\pi} \left(C(\sqrt{2t/\pi}) \cos t + S(\sqrt{2t/\pi}) \sin t \right), \quad (30)$$

with $\sigma = \sqrt{t-\tau}$ and $C(t)$ and $S(t)$ the Fresnel cosine and sine functions [21], respectively.

The Basset integral F_B was evaluated at $t = 50\pi$ with different numbers of points N uniformly distributed in the interval $[0, 50\pi]$. The results for both the SD-approach and the TB-method are presented in **Table 2**. Here, it can be seen that the error of the TB-method is substantially smaller than that of the SD-approach. When increasing the number of points N it can be seen that the TB-method is second-order accurate in time (in agreement with analysis that can be done by using Taylor series), whereas the SD-approach is first-order accurate in time. More methods have been compared by Bombardelli et al. [13] but these methods have even lower order of convergence than the SD-approach.

Example 2. Space-dependent steady velocity field

In order to test the overall numerical scheme for the computation of particle trajectories we have implemented a particular space-dependent steady velocity field. The particle trajectory is a circle and given by $(x(t), y(t)) = (r \cos \omega t, -r \sin \omega t)$, where r and ω denote the radius and the angular velocity, respectively. The velocity field and its derivation is given in **Appendix A**. For the test case, exactly one revolution is simulated, from $t = 0$ until $t = 2\pi$. In order to test the stability of the overall scheme two different approaches have been tested. One with the completely explicit time integration procedure for the Basset force and the other with the partially implicit procedure, see Section 4. For both the implicit and explicit method the Basset force is computed with the TB-method and show second-order convergence in Δt . The relative error is computed with $\mathbf{x}_p(2\pi)$. The results are presented in **Table 3** and clearly indicate that the explicit scheme is very unstable when the number of time steps is smaller than 256. Even when taking 256 time steps the explicit procedure may be unstable and the data in **Table 3** are put in parenthesis to indicate this uncertainty. The partially implicit scheme remains stable even with the number of time steps as small as 16.

Table 2

Relative error and order of convergence for the Basset integral, for the SD-approach [13] and the TB-method.

Points N	Relative error SD	Order SD	Relative error TB	Order TB
81	$4.03 \cdot 10^{-1}$		$1.34 \cdot 10^{-1}$	
243	$1.37 \cdot 10^{-1}$	1.0	$2.54 \cdot 10^{-2}$	1.5
729	$4.66 \cdot 10^{-2}$	1.0	$3.29 \cdot 10^{-3}$	1.9
2,187	$1.56 \cdot 10^{-2}$	1.0	$3.93 \cdot 10^{-4}$	1.9
6561	$5.22 \cdot 10^{-3}$	1.0	$4.54 \cdot 10^{-5}$	2.0
19,683	$1.74 \cdot 10^{-3}$	1.0	$5.15 \cdot 10^{-6}$	2.0
59,049	$5.80 \cdot 10^{-4}$	1.0	$5.80 \cdot 10^{-7}$	2.0
177,147	$1.93 \cdot 10^{-4}$	1.0	$6.49 \cdot 10^{-8}$	2.0
531,441	$6.45 \cdot 10^{-5}$	1.0	$7.24 \cdot 10^{-9}$	2.0
1,594,323	$2.15 \cdot 10^{-5}$	1.0	$8.06 \cdot 10^{-10}$	2.0

Table 3

Relative error and order of convergence for the overall numerical scheme, tested for the trajectory of a small particle in a space dependent steady velocity field.

Number of time steps	Relative error explicit	Order explicit	Relative error implicit	Order implicit
16	Unstable		$3.63 \cdot 10^{-1}$	
32	Unstable		$8.32 \cdot 10^{-2}$	2.1
64	Unstable		$2.09 \cdot 10^{-2}$	2.0
128	Unstable		$5.28 \cdot 10^{-3}$	2.0
256	$(4.80 \cdot 10^{-2})$		$1.33 \cdot 10^{-3}$	2.0
512	$3.05 \cdot 10^{-4}$		$3.33 \cdot 10^{-4}$	2.0
1024	$7.68 \cdot 10^{-5}$	2.0	$8.34 \cdot 10^{-5}$	2.0
2048	$1.93 \cdot 10^{-5}$	2.0	$2.09 \cdot 10^{-5}$	2.0
4096	$4.84 \cdot 10^{-6}$	2.0	$5.21 \cdot 10^{-6}$	2.0

Example 3. Time-dependent velocity field

The trajectory of a spherical particle in an arbitrary time-dependent velocity field can rather straightforwardly be computed as long as the velocity field is smooth enough. The derivation of the particle trajectory uses Laplace transforms and the analytical procedure is given in [Appendix B](#). The overall numerical scheme is tested by computing the trajectory of a particle in the following one-dimensional, time-dependent velocity field

$$u(t) = \frac{(m_p - m_f)g}{6\pi a\mu} \cos 2t. \quad (31)$$

The total force on the particle is zero at $t = 0$, i.e., \mathbf{F}_{St} and \mathbf{F}_G are in balance. In order to compute the Basset force the implicit TB-method is used. The integration is carried out from $t = 0$ until $t = 2\pi$. The relative error is computed for $u_p(2\pi)$ and is presented in [Table 4](#), where once again second-order time accuracy is confirmed. From these test cases, using both a time-dependent and a space-dependent velocity field for the computation of particle trajectories, we can conclude that the (partially implicit) TB-method is stable and second-order accurate in time, and conjecture that this remains the case for particles in arbitrary time- and space-varying flow fields.

Example 4. Computational efficiency due to modified kernel integration

In this example the computational savings when using the modified tail kernel, given in [\(9\)](#) and [\(16\)](#), is investigated based on analysis of the number of flops per time step, per particle and per space dimension. For the window kernel this is $N + 1$ flops because only one vector dot product is calculated. For each exponential function three extra flops are needed. To see how efficient the tail kernel works the upper bound [\(13\)](#) for the error is plotted as a function of the computation time, [Fig. 4](#). Different numbers (indicated by m) of exponential functions are taken into account. The results are plotted in [Fig. 4](#). Here it can be seen that the best choice for m depends on the particular situation. Because the computation time is directly connected with N as mentioned above, N must be chosen optimally. From [Fig. 4](#) one can see that N must be chosen small, otherwise increasing m would be a better option. Because $t_{win} = N\Delta t$ this immediately gives an optimal value for t_{win} . Furthermore, the results show a significant saving in computation time. This can easily be a factor of 100 or more. When looking to the

Table 4
Relative error and order of convergence for the overall numerical scheme, for the velocity field [\(31\)](#).

Time steps	Relative error	Order
16	$9.96 \cdot 10^{-2}$	
32	$2.38 \cdot 10^{-2}$	2.1
64	$5.57 \cdot 10^{-3}$	2.1
128	$1.31 \cdot 10^{-3}$	2.1
256	$3.13 \cdot 10^{-4}$	2.1
512	$7.56 \cdot 10^{-5}$	2.0
1024	$1.84 \cdot 10^{-5}$	2.0
2048	$4.53 \cdot 10^{-6}$	2.0
4096	$1.12 \cdot 10^{-6}$	2.0

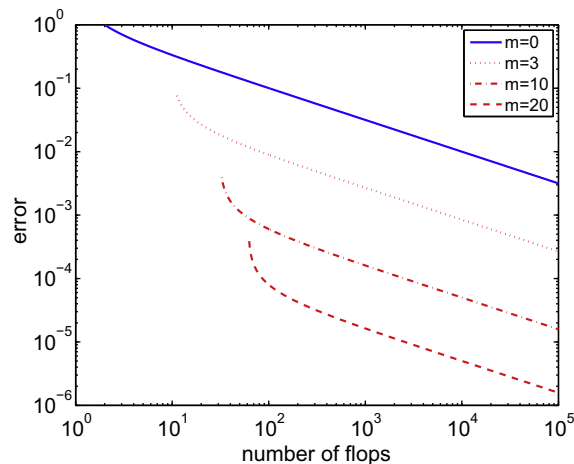


Fig. 4. Upper bound for the error in the approximation of the Basset force as a function of the number of flops for different number of exponential functions (indicated by m).

memory requirements the results are even better. For the window method as many memory locations as flops are needed whereas each exponential function only takes one memory location instead of 3 flops. So using the tail kernel not only saves time but also memory.

Typically, the use of the tail kernel reduces the computational costs of the Basset force by more than an order of magnitude, whereas the memory requirement is even reduced more. Furthermore, the error is reduced by more than an order of magnitude. The question remains, of course, whether the computational savings directly result in faster simulations. This depends on the remaining part of the simulation. Although the other force contributions in (28) can be calculated much faster than the Basset force this does not have to hold for the interpolation of the velocities in a turbulence simulation. The velocity of the flow field is only computed at the grid points and an interpolation must be carried out to compute the velocity at the particle position. This may be very time consuming and it can become the new bottleneck. The reduction of CPU-time might then not be as big as expected but it remains significant. Additionally, the decrease in memory requirement may become essential when increasing the number of particles in turbulence simulations.

6. Light particles in isotropic turbulence

In this section a brief statistical analysis of velocities of particles, released in an isotropic turbulent flow, is provided. The isotropic turbulence simulation is performed by means of direct numerical simulations. The numerical code consist of two parts. First the Navier–Stokes equations with the Boussinesq approximation are solved on a triple periodic domain using a pseudo-spectral code [22,23] (Eulerian approach). Second, the particle trajectories are obtained by the Lagrangian approach as explained in the previous sections. The simulation is performed on a 128^3 grid. The number of (light) particles is 20,000 and the particle-to-fluid density ratio $\rho_p/\rho_f = 4$ (thus all terms in the MR equation are relevant, see Refs. [6,7]). The integral-scale Reynolds number is $Re = UL/\nu = 1333$, with U the typical root-mean-square velocity and L the integral length scale. The Stokes number St is typically in the range $0.1 \leq St \leq 1.0$ [7] and particles are tracked for a period of approximately two eddy turnover times.

Two simulations have been carried out under exactly the same flow conditions and particle tracking is either based on the classical approach (window method) or on the novel integration method (exponential method) for the Basset kernel. In the first simulation only the window kernel (4) has been used, where the number of time steps in the window is $n = 500$. The other one uses the modified window kernel, given in (9) and (16). In this case only five time steps are taken into account in the window, so $n = 5$. For the tail of the Basset kernel the number of exponential functions $m = 10$.

In order to study a particle trajectory we start with considering the energy spectrum of the particle. To obtain the energy spectrum, we first need to calculate the autocorrelation $R(\tau)$ of the velocity, which is defined by

$$R(\tau) = \frac{\langle u_p(t)u_p(t + \tau) \rangle}{\langle u_p(t)^2 \rangle}. \quad (32)$$

Here, $\langle \cdot \rangle$ denotes the average in over the different particles. The particles are embedded in a homogeneous isotropic turbulent flow and no gravitation is applied. Therefore, we are allowed to average over the components of the velocity vector of all particles. No time averaging has been applied for the present velocity data as this run covers only one or two eddy turnover times. The results for the autocorrelation of the velocity are shown in Fig. 5 and we see that the results for both the window method and the exponential method are comparable. The energy spectrum obtained from the particle velocities can be calculated by taking the cosine transform of the autocorrelation function, and is shown in Fig. 6. Although the results are similar

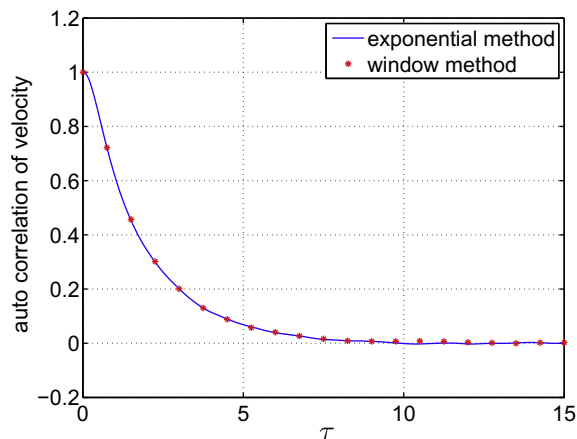


Fig. 5. Autocorrelation of the particle velocity u_p . The solid line represents the result from the exponential method and the dots those from the window method.

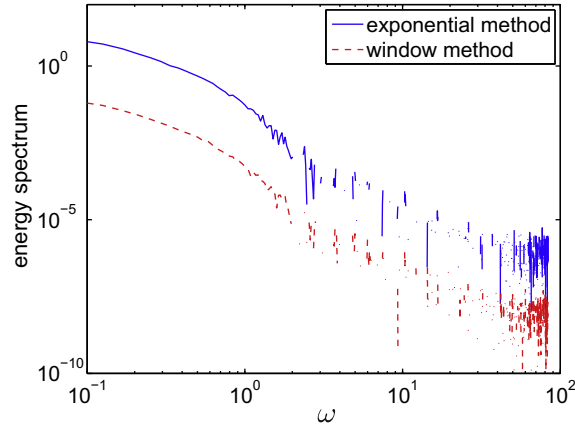


Fig. 6. Energy spectrum of the particle velocity. The graph of the window method (dashed line) is shifted downward with respect to the spectrum from the exponential method (solid line) by a factor of 100 for clarity.

we are interested in possible differences between the two spectra. If these differences have an overall trend this would mean that statistical properties can be influenced by the different methods of evaluating the Basset kernel. However, to observe any error in the evaluation of the Basset force kernel the differences should be larger than the statistical noise.

A starting point for an analytical evaluation of possible differences between the window method and the exponential method consists of the response of a single particle in a uniform oscillating flow field. We are therefore interested in the periodic solution u_p of a spherical particle responding to an oscillating velocity field $u = \cos \omega t$ (or $u = \mathcal{R}[\exp(i\omega t)]$, with i the imaginary unit and \mathcal{R} denoting the real part of this expression). The particle velocity can then be expressed as $u_p = \mathcal{R}[V \exp(i\omega t)]$ with V a complex amplitude, which is dependent on the method chosen to evaluate the Basset force kernel. For the window method and the exponential method we introduce V_{win} and V_{exp} , respectively. For V_{exp} the exact solution V_{ex} is used since the error of the exponential method is assumed to be negligibly small, see also Fig. 3. In general, $|V_{\text{win}}| \neq |V_{\text{ex}}|$ which means that some frequencies are suppressed with the window method while others may be amplified. This should become visible in the energy spectrum of particle velocities.

In order to find V_{win} Eq. (2) should be solved for $u = \mathcal{R}[\exp(i\omega t)]$ and $u_p = \mathcal{R}[V \exp(i\omega t)]$, resulting in the following integro-differential equation:

$$i\omega m_p V_{\text{win}} = 6\pi\alpha\mu(1 - V_{\text{win}}) + \frac{i\omega}{2}m_f(3 - V_{\text{win}}) + i\omega c_B(1 - V_{\text{win}}) \int_{t-t_{\text{win}}}^t \frac{e^{-i\omega(t-\tau)}}{\sqrt{t-\tau}} d\tau. \quad (33)$$

Here, we used the fact that the velocity field is uniform, one dimensional and that no gravity is applied. Applying the change in variables $\sigma = \sqrt{(t-\tau)\omega}$, allows us to find an expression for V_{win} i.e.,

$$V_{\text{win}} = 1 + \frac{(m_f - m_p)i\omega}{6\pi\alpha\mu + (\frac{1}{2}m_f + m_p)i\omega + c_B\sqrt{2\omega\pi}Q(\sqrt{2t_{\text{win}}\omega\pi})}, \quad (34)$$

where $Q(t) = S(t) + iC(t)$, with $C(t)$ and $S(t)$ the Fresnel cosine and sine functions, respectively [21]. V_{ex} can now be found by taking $V_{\text{ex}} = \lim_{t_{\text{win}} \rightarrow \infty} V_{\text{win}}$ which results in

$$V_{\text{ex}} = 1 + \frac{(m_f - m_p)i\omega}{6\pi\alpha\mu + (\frac{1}{2}m_f + m_p)i\omega + c_B\sqrt{\frac{\omega\pi}{2}}(1+i)}. \quad (35)$$

Inspection of the energy spectrum displayed in Fig. 6 reveals that the noise becomes more important for increasing ω . The effects from the different methods for the computation of the Basset force kernel turned out to be most important for $\omega \geq 1$. Unfortunately, the noise in the energy spectrum is already larger than predicted for the differences between the window and exponential method. One way of decreasing the error would be averaging over time, but with the limited number of eddy turnover times in the present simulation this is not feasible. However, an alternative approach exists in comparing the autocorrelation of the particle acceleration.

The autocorrelation of the particle acceleration is plotted in Fig. 7. Here, the typical time scale is much shorter than that of the particle velocity, therefore it is possible to also average over time. The spectrum is calculated by taking the cosine transform of the autocorrelation acceleration function and is displayed in Fig. 8. Because the particle acceleration is used instead of the particle velocity, higher frequencies (shorter time scales) become more important. In this way deteriorating influence of the noise on the spectrum is shifted to higher frequencies. Nevertheless, the computed spectrum should still be affected by the method that is chosen for the evaluation of the Basset force kernel. In order to observe the differences the best approach is to plot the ratio of the two spectra as function of frequency. When no essential difference exists between the window and

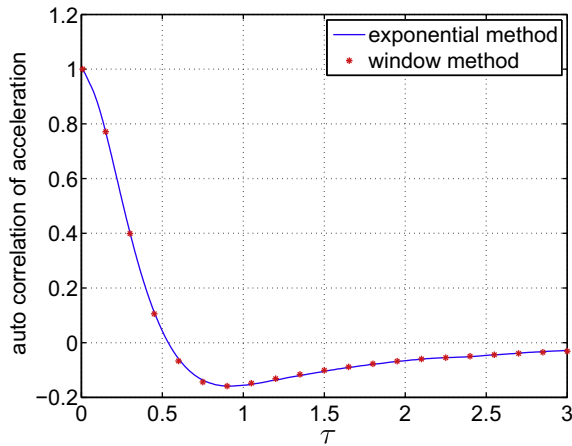


Fig. 7. Autocorrelation of the particle acceleration $a_p = du_p/dt$. The solid line represents the result from the exponential method and the dots those from the window method.

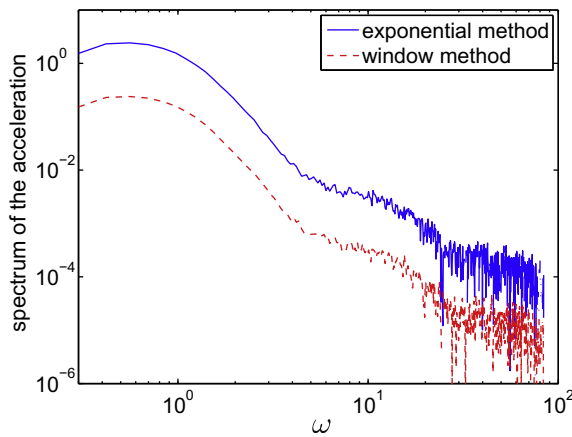


Fig. 8. Spectrum of the particle acceleration. The graph of the window method (dashed line) is shifted downward with respect to the spectrum from the exponential method (solid line) by a factor of 10 for clarity.

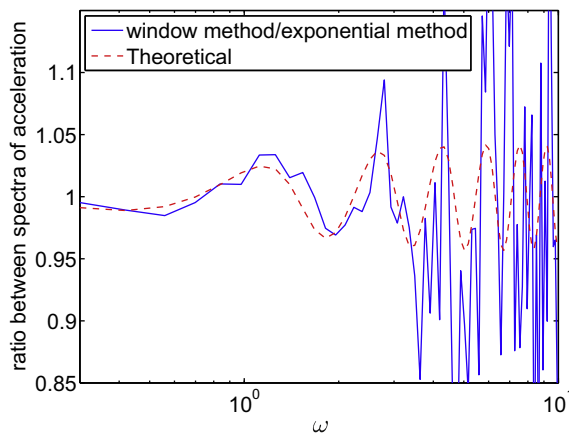


Fig. 9. The theoretical ratio $\left(\frac{V_{\text{window}}}{V_{\text{exponential}}}\right)^2$ (dashed line) compared with a similar ratio of the particle acceleration spectra (solid line).

exponential method their ratio would be equal to one with some noise added to it. However, the window method suppresses some frequency components while others are amplified, so the deviation from one is a measure for the error in the window

method. In Fig. 9 the ratio of both spectra is shown in combination with the theoretical ratio defined by $\left(\frac{|V_{\text{win}}|}{|V_{\text{ex}}|}\right)^2$. From Fig. 9 it can be seen that the theoretical ratio predicts the ratio obtained from the simulation, including the local maxima and minima quite well, provided the frequency is not too high. For higher frequencies the noise becomes larger but the theoretical and computational ratios still seem to have the same trend. The novel exponential method to evaluate the Basset force kernel might be considered as an excellent and efficient method for tracking of many particles in turbulent flows.

7. Conclusions

We have introduced a novel method for the evaluation of the Basset force kernel and analysed several aspects of its implementation. The tail of the Basset force kernel is approximated by exponential functions. The contribution of these exponential functions can be calculated in a recursive way which makes it very efficient. Typically the use of the tail kernel reduces the computational costs of the Basset force by more than an order of magnitude, whereas the memory requirement is reduced even more. Furthermore, the error in the tail of the Basset force is also reduced by more than an order of magnitude in comparison with the traditional window method.

A trapezoidal-based method is developed in order to deal with the singularity of the Basset force. This method has a temporal accuracy of $\mathcal{O}(\Delta t^2)$ where other methods only have a temporal accuracy of $\mathcal{O}(\Delta t)$ or lower. This method is made partially implicit in order to make it more stable.

The method has been implemented in a tracking algorithm for (light) inertial particles in turbulent flows. The isotropic turbulence simulation shows that the error made by the window method can influence statistics on the particle trajectories. This has been illustrated with the velocity and acceleration spectra. Therefore, the novel exponential method is preferred over the classical window method. Because the new implementation is much faster than the classical one, more particles can be taken into account in simulations, which opens possibilities for further research.

Acknowledgement

We thank Ben van den Broek for contributing to the derivation of the analytical solution presented in Appendix B.

Appendix A. Flow field for circular particle trajectories

In this appendix the space dependent velocity field is derived that allows a circular particle trajectory as solution of the MR equation. Suppose the particle trajectory and velocity is given by

$$\mathbf{x}_p = r\mathcal{R}[\mathbf{e}e^{-i\omega t}], \quad \mathbf{u}_p = -r\omega\mathcal{R}[\mathbf{i}ee^{-i\omega t}], \quad (\text{A.1})$$

with $\mathbf{e} = \mathbf{e}_x - i\mathbf{e}_y$. For the flow velocity field we are looking for solutions of (2) of the form

$$\mathbf{u} = -\mathcal{R}[s(x + iy)\mathbf{e}] + 2\alpha z\mathbf{e}_z, \quad (\text{A.2})$$

with $s = \alpha + i\beta$ a complex constant. For $\beta = 0$ the velocity field represents a sink flow $\mathbf{u} = (-\alpha x, -\alpha y, 2\alpha z)$ and for $\alpha = 0$ it represents solid body rotation: $\mathbf{u} = (\beta y, -\beta x, 0)$. A spherical particle released in the plane $z = 0$ will remain there due to the symmetry of the flow. Substituting (A.1) and (A.2) (assuming $z = 0$) in Eq. (2), and taking into account that no gravity is applied, the Faxén corrections are 0 for a linear velocity field and $\frac{D\mathbf{u}}{Dt} = \mathcal{R}[s^2(x + iy)\mathbf{e}]$, yields the following quadratic relation for s :

$$-\omega^2\left(m_p + \frac{1}{2}m_f\right) = 6\pi a\mu(-s + i\omega) + \frac{3}{2}m_f s^2 + c_B\sqrt{\frac{\omega\pi}{2}}(\omega + is)(1 + i). \quad (\text{A.3})$$

There are two solutions for s , but one of the solutions of s results in an unphysical particle trajectory and is therefore discarded.

Appendix B. Time dependent velocity field

In this appendix the particle trajectory is derived given the uniform, time dependent velocity field (31). The particle is released with an initial velocity $\mathbf{u}_p(0)$. For a uniform velocity field Eq. (2) can be simplified to

$$-\left(m_p + \frac{1}{2}m_f\right)\frac{d\mathbf{w}}{dt} = 6\pi a\mu\mathbf{w} + (m_f - m_p)\frac{d\mathbf{u}}{dt} - (m_p - m_f)\mathbf{g}\mathbf{e}_z + c_B\int_0^t K_B(t - \tau)\frac{d\mathbf{w}(\tau)}{d\tau}d\tau, \quad (\text{B.1})$$

where $\mathbf{w} = \mathbf{u} - \mathbf{u}_p$. The velocity field \mathbf{u} will be expanded in a Fourier series, $\mathbf{u}(t) = \sum_{n=-\infty}^{\infty}\mathbf{u}_n e^{in\omega t}$. The Laplace transform of \mathbf{w} is given by $\mathbf{W}(s) = \int_0^{\infty} e^{-st}\mathbf{w}(t)dt$, and the Laplace transform of Eq. (B.1) reads

$$-\left(m_p + \frac{1}{2}m_f + c_B\sqrt{\frac{\pi}{s}}\right)(s\mathbf{W} - \mathbf{w}(0)) = 6\pi a\mu\mathbf{W} - \frac{(m_p - m_f)}{s}\mathbf{g}\mathbf{e}_z + (m_f - m_p)\sum_{n=-\infty}^{\infty}\mathbf{u}_n\frac{i n\omega}{s - i n\omega}. \quad (\text{B.2})$$

Using spitting in partial fractions this yields for $\mathbf{W}(s)$

$$\begin{aligned} \mathbf{W}(s) = & \frac{c}{\sqrt{s}} \left(\frac{c_+}{\sqrt{s} + c_-} - \frac{c_-}{\sqrt{s} + c_+} \right) \left(\mathbf{w}(0) - \frac{m_p - m_f}{6\pi a \mu} \mathbf{g} \mathbf{e}_z \right) + \frac{m_p - m_f}{6\pi a \mu} \frac{\mathbf{g}}{s} \mathbf{e}_z \\ & + \sum_{n=-\infty}^{\infty} \mathbf{c}_n \left\{ \frac{1}{(c_+ + \sqrt{in\omega})(c_- + \sqrt{in\omega})(s - in\omega)} + \frac{\sqrt{in\omega}(c_+ + c_-)}{(c_+^2 - in\omega)(c_-^2 - in\omega)\sqrt{s}(\sqrt{s} + \sqrt{in\omega})} \right. \\ & \left. + \frac{c}{\sqrt{s}} \left(\frac{c_+}{(c_+^2 - in\omega)(\sqrt{s} + c_+)} - \frac{c_-}{(c_-^2 - in\omega)(\sqrt{s} + c_-)} \right) \right\}, \end{aligned} \quad (\text{B.3})$$

with c_+ , c_- , c and \mathbf{c}_n constants given by

$$c_{\pm} = \frac{c_B \sqrt{\pi} \pm \sqrt{c_B^2 \pi - 12\pi a \mu (2m_p + m_f)}}{2m_p + m_f}, \quad (\text{B.4})$$

$$c = \frac{2m_p + m_f}{2\sqrt{c_B^2 \pi - 12\pi a \mu (2m_p + m_f)}}, \quad (\text{B.5})$$

$$\mathbf{c}_n = \frac{in\omega(m_p - m_f)\mathbf{u}_n}{m_p + \frac{1}{2}m_f}. \quad (\text{B.6})$$

Transformation back to physical space results in

$$\begin{aligned} \mathbf{w}(t) = & c \left[c_+ \psi(c_- \sqrt{t}) - c_- \psi(c_+ \sqrt{t}) \right] \left(\mathbf{w}(0) - \frac{m_p - m_f}{6\pi a \mu} \mathbf{g} \mathbf{e}_z \right) + \frac{m_p - m_f}{6\pi a \mu} \mathbf{g} \mathbf{e}_z \\ & + \sum_{n=-\infty}^{\infty} \mathbf{c}_n \left\{ \frac{1}{(c_+ + \sqrt{in\omega})(c_- + \sqrt{in\omega})} e^{in\omega t} + \frac{\sqrt{in\omega}(c_+ + c_-)}{(c_+^2 - in\omega)(c_-^2 - in\omega)} \psi(\sqrt{in\omega t}) \right. \\ & \left. + \frac{cc_+}{c_+^2 - in\omega} \psi(c_+ \sqrt{t}) - \frac{cc_-}{c_-^2 - in\omega} \psi(c_- \sqrt{t}) \right\}, \end{aligned} \quad (\text{B.7})$$

with $\psi(z) = \exp(z^2) \operatorname{erfc}(z)$.

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] E. Loth, Numerical approaches for motion of dispersed particles, droplets and bubbles, *Prog. Energy Combust.* 26 (2000) 161–223.
- [3] E.E. Michaelides, Hydrodynamic force and heat/mass transfer from particles, bubbles, and drops. The Freeman scholar lecture, *J. Fluids Eng.* 125 (2) (2003) 209–238.
- [4] R. Mei, R.J. Adrian, T.J. Hanratty, Particle dispersion in isotropic turbulence under Stokes drag and Basset force with gravitational settling, *J. Fluid Mech.* 225 (1991) 481–495.
- [5] V. Armenio, V. Fiorotto, The importance of the forces acting on particles in turbulent flows, *Phys. Fluids* 13 (2001) 2437–2440.
- [6] M. van Aartrijk, H.J.H. Clercx, Vertical inertial particle dispersion in stably stratified turbulence: the influence of the Basset force, *Phys. Fluids* 22 (2010) 013301-1–013301-9.
- [7] M. van Aartrijk, H.J.H. Clercx, The dynamics of small inertial particles in weakly stratified turbulence, *J. Hydro-Environ. Res.* 4 (2010) 103–114.
- [8] Y.D. Sobral, T.F. Oliveira, F.R. Cunha, On the unsteady forces during the motion of a sedimenting particle, *Powder Technol.* 178 (2007) 129–141.
- [9] I. Niño, M. García, Using Lagrangian particle saltation observations for bedload sediment transport modelling, *Hydrolog. Process* 12 (1998) 1197–1218.
- [10] N. Mordant, J.F. Pinton, Velocity measurement of a settling sphere, *Eur. Phys. J. B* 18 (2000) 343–352.
- [11] D.J. Vojir, E.E. Michaelides, Effect of the history term on the motion of rigid spheres in a viscous fluid, *Int. J. Multiphase Flow* 20 (1994) 547–556.
- [12] P. Tanga, A. Provenzale, Dynamics of advected tracers with varying buoyancy, *Physica D* 76 (1994) 202–215.
- [13] F.A. Bombardelli, A.E. González, Y.I. Niño, Computation of the particle basset force with a fractional-derivative approach, *J. Hydraul. Div.* 134 (2008) 1513–1520.
- [14] A.J. Dorgan, E. Loth, Efficient calculation of the history force at finite Reynolds numbers, *Int. J. Multiphase Flow* 33 (2007) 833–848.
- [15] E.E. Michaelides, A novel way of computing the Basset term in unsteady multiphase flow computations, *Phys. Fluids A* 4 (7) (1992) 1579–1582.
- [16] H. Faxén, Die Bewegung einer starren Kugel längs der Achse eines mit zäher Flüssigkeit gefüllten Rohres, *Ark. Mat. Astron. Fys.* 17 (1923) 1–28.
- [17] H. Homann, J. Bec, Finite-size effects in the dynamics of neutrally buoyant particles in turbulent flow, *J. Fluid Mech* 651 (2010) 81–91.
- [18] T.R. Auton, J.C.R. Hunt, M. Prud'homme, The force exerted on a body in inviscid unsteady non-uniform rotational flow, *J. Fluid Mech.* 197 (1988) 241–257.
- [19] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, *Numerical Recipes*, Cambridge University Press, New York, 1992.
- [20] F.B. Tatom, The Basset force as a semiderivative, *Appl. Sci. Res.* 45 (1988) 283–285.
- [21] A. Jeffrey, *Handbook of Mathematical Formulas and Integrals*, Elsevier, United Kingdom, 2004, pp. 245–248.
- [22] M. van Aartrijk, Dispersion of inertial particles in stratified turbulence. PhD thesis, Eindhoven University of Technology, 2008.
- [23] M. van Aartrijk, H.J.H. Clercx, K.B. Winters, Single-particle, particle-pair, and multiparticle dispersion of fluid particles in forced stably stratified turbulence, *Phys. Fluids* 20 (2008) 025104-1–025104-16.