



A Finite Element/Neural Network Framework for Modeling Suspensions of Non-spherical Particles

Concepts and Medical Applications

Martyna Minakowska¹ · Thomas Richter¹ · Sebastian Sager²

Received: 17 September 2020 / Accepted: 5 January 2021 / Published online: 19 March 2021
© The Author(s) 2021

Abstract

An accurate prediction of the translational and rotational motion of particles suspended in a fluid is only possible if a complete set of correlations for the force coefficients of fluid-particle interaction is known. The present study is thus devoted to the derivation and validation of a new framework to determine the drag, lift, rotational and pitching torque coefficients for different non-spherical particles in a fluid flow. The motivation for the study arises from medical applications, where particles may have an arbitrary and complex shape. Here, it is usually not possible to derive accurate analytical models for predicting the different hydrodynamic forces. The presented model is designed to be applicable to a broad range of shapes. Another important feature of the suspensions occurring in medical and biological applications is the high number of particles. The modelling approach we propose can be efficiently used for simulations of solid-liquid suspensions with numerous particles. Based on resolved numerical simulations of prototypical particles we generate data to train a neural network which allows us to quickly estimate the hydrodynamic forces experienced by a specific particle immersed in a fluid.

Keywords Non-spherical particles · Finite element method · Neural network

Mathematics Subject Classification (2010) 74F10 · 65M60 · 76D05 · 76M10

✉ Martyna Minakowska
martyna.minakowska@ovgu.de

Thomas Richter
thomas.richter@ovgu.de

Sebastian Sager
sebastian.sager@ovgu.de

¹ Institute of Analysis and Numerics, Otto-von-Guericke University Magdeburg, Universitätsplatz 2, 39106, Magdeburg, Germany

² Institute of Mathematical Optimization, Otto-von-Guericke University Magdeburg, Universitätsplatz 2, 39106, Magdeburg, Germany

1 Introduction

The prediction of the motion of non-spherical particles suspended in a fluid is crucial for the understanding of natural processes and industrial applications. In such processes, particles can have different shapes and sizes, may be deformed and can interact with each other. So far, in the majority of scientific studies, particulate flow modelling is investigated with the hypothesis of perfectly spherical particles, thereby eliminating orientation and shape effects. This assumption is very convenient due to its simplicity, the fact that the behaviour of spheres is well known and the availability of a number of models to describe the interaction with fluid flow. The study of suspensions of multiple, irregular-shaped, interacting and deformable particles has received less attention and still presents a challenge.

Particles come in all sort of shapes and sizes, in fact, due to the arbitrary nature of naturally occurring particles there are an indefinite number of possible shapes. On the other hand, there is a common understanding that particle shape has a strong influence on the dynamics of NSPS (non-spherical particulate systems). These two factors combined makes modelling of NSPS in general way impossible, since for describing the motion of non-spherical particles, detailed information on the fluid dynamic forces acting on such particles are necessary, but generally not available. Therefore, particular models place emphasis on different shapes and types of flow. In our work we focus on medical applications, namely on modelling of platelets dynamics under blood flow. Platelets play a main role in the process of blood coagulation and therefore are of great interest in the modelling of blood flow. The majority of models characterize platelet motion quantitatively and use approaches such as immersed boundary method [13], cellular Potts model [40–42] and dissipative particle dynamics [12, 36], treating platelets as points and thus neglecting entirely their shape. Some effort has been done to model platelets as rigid two- or three-dimensional spheres or spheroids [24, 35, 46] but this simplification of shape has been shown to affect processes in which platelets are involved (e.g. spherical platelets marginate faster than ellipsoidal and disc forms) [37].

On the other hand, several studies have been performed to model particles of irregular shapes, but motivation behind them usually arises from engineering applications (dispersion of pollutant, pulverized coal combustion, pneumatic transport) [48], where particles are much bigger and usually constitute a significant part of the volume of the suspension [34]. But even under given very specific circumstances there is no set of correlations of the forces acting on irregular-shaped particles suspended in a fluid (forces arising from fluid-particle interaction). Furthermore, in these kinds of models interactions between particles become dominant in terms of determining particle dynamics, whereas platelets are very dilute in blood and their contact is rather rare.

The motivation behind this work is fourfold and arises from the specificity of the modelled phenomenon. Firstly, platelets, because of their small size (compared to blood vessel), are often modelled quantitatively, ignoring the importance of shape effects. Secondly, platelets constitute only a very small volume of blood what makes most of the engineering applications-driven models, where particles are very dense and often interact with each other, inappropriate. Thirdly, highly irregular shape of platelets requires new methods for estimating force coefficients in fluid-particle interaction. Fourthly, platelets are numerous and the evolution of their movement needs to be evaluated quickly and efficiently.

A common approach to the problem of the modelling of NSPS consists in developing analytical models for fluid dynamic forces acting on particles, cf. [17, 21, 23, 45, 48]. In this contribution we go a different way and refrain from giving analytical expressions. Instead, by prototypical simulations we train a neural network model that takes several

parameters describing the particle shape, size and the flow configuration as input and which gives hydrodynamical coefficients like drag, lift and torque as output. We demonstrate that such a model can be efficiently trained in an offline phase for a range of particles. Later on, the coupling of the flow model with the particle system only requires the evaluation of the network for getting updates on these coefficients. This two-step approach with an offline phase for training a network based on the particle classes under consideration allows for a direct extension to further applications.

A similar approach is considered in [43]. Here, the authors design and train a radial basis function network to predict the drag coefficient of non-spherical particles in fluidized beds. Training is based on experimental data and the network input is the particle's sphericity and the Reynolds number, covering the Stokes and the intermediate regime. Our approach, based on training data generated by detailed simulations of prototypical particles for predicting drag, lift and torque coefficients could be augmented by including experimental data.

In the following section we describe prototypical medical applications where such a heterogeneous modeling approach can be applied. Then, in Section 3 we detail the general framework for coupling the Navier–Stokes equation with a discrete particle model. Section 4 introduces the neural network approach for estimating hydrodynamical coefficients and we describe the procedure for offline training of the network. Then, different numerical test cases are described in Section 5.1 that show the potential of such a heterogeneous modeling approach. Section 5.2 is devoted to a numerical study comprising many particles and shows the efficiency of the presented approach. We summarize with a short conclusion in Section 6.

2 Modeling of Suspensions with Non-spherical Particles and Medical Applications

Platelets are a vital component of the blood clotting mechanism. They are small non-nucleated cell fragments with the diameter of approximately $2 - 4 \mu\text{m}$, the thickness of $0.5 \mu\text{m}$, volume of about $7 \mu\text{m}^3$ and the number density of $1.5 - 4 \cdot 10^5 \mu\text{l}^{-1}$ [11] which leads to a volume fraction of only about $10^{-3} : 1$. In the rest state platelets shape is discoid, but they have the ability of deforming as a response to various stimuli (chemical and mechanical). They may become star shaped (rolling over blood vessels wall to inspect its integrity). During the clotting process they undergo deep morphological changes, from becoming spherical and emitting protuberances (filopodia or pseudopods) which favour mutual aggregation, as well as adhesion to other elements constituting the clot to fully flat, spread stage, to enable wound closure. Thrombocytes constitute approximately less than 1% of the blood volume, therefore individual platelets have a negligible effect on blood rheology [20].

Due to a significant effect of the particles shape on their motion and their practical importance in industrial applications, the non-sphericity started attracting attention in the modelling and simulations of particles transport in fluid flows [18, 23, 45]. Unfortunately, it is not possible to consider each shape in the implementation of numerical methods because of non-existence of a single approach describing accurately the sizes and shapes of non-spherical particles. Spheres can be described by a single characteristic value, i.e. the diameter, whereas non-spherical particles require more parameters. Even very regular shapes need at least two parameters. Moreover, the particles may have varying orientation with respect to the flow, what makes the description of their behaviour even more difficult. Even though several methods for shape parametrization and measurement have been

suggested, none has won greater acceptance. One of the most commonly used shape factor is a sphericity which was firstly introduced in [38] and defined as the ratio between the surface area of a sphere with equal volume as the particle and the surface area of the considered particle. Then the drag coefficient for non-spherical particle is estimated by using correlations for spherical particles and modified to take into account the sphericity factor [44]. Models using sphericity as a shape factor give promising results when restricted to non-spherical particles with aspect ratios β less than 1.7 [7], where $\beta = L/D$ with L and D being length and diameter of the considered particle. For particles having extreme shapes and those having little resemblance to a sphere, the sphericity concept fails to produce satisfying quantitative results [6]. In general, the lower the sphericity, the poorer is the prediction. Also, the same value of the sphericity may be obtained for two very varying shapes whose behaviour in the flow is different. Moreover, the sphericity does not take the orientation into account. In order to introduce orientation dependency in drag correlations, some researchers use two additional factors: the crosswise sphericity and lengthwise sphericity [17]. Most of these correlations employ also dependency on particle Reynolds number defined as

$$Re_p = \rho \bar{u} d_{eq} / \mu,$$

where ρ and μ are the fluid density and the viscosity, $\bar{u} = u_f - u_p$ is the velocity of the particle relative to the fluid velocity and d_{eq} is the equivalent particle diameter, i.e., the diameter of a sphere with the same volume as the considered particle in order to include the importance of fluid properties.

The shape factor concept may be described as an attempt to define a single correlation for drag for all shapes and orientations. Another approach appeared as an alternative consisting in obtaining drag coefficient expressions for a fixed shape and any orientations: the drag coefficient is determined at two extreme angles of incidence (0° and 90°) from existing correlations which are then linked by some functions resulting in the whole range of angles of incidence for non-spherical particles [31]. However, besides drag force, non-spherical particles are associated with orientation and shape induced lift along with pitching and rotational torques. Hölzer and Sommerfeld [18] investigated a few different shapes of non-spherical particles at different flow incident angles using the Lattice Boltzmann method to simulate the flow around the particle. Wachem et al. [45] proposed a new force correlations (for drag, lift, pitching and rotational torque) for particular shapes of non-spherical particles (two ellipsoids with different aspect ratio, disc and fibre) from data given by a direct numerical simulation (DNS) carried out with an immersed boundary method. Those correlations employ particle Reynolds number, angle of incidence and some shape-related coefficients. Ouchene et al. [26] determined force coefficients depending on particle Reynolds number, aspect ratio and angle of incidence by fitting the results extracted from DNS computations of the flow around prolate ellipsoidal particles. Discrete element methods (DEM) coupled with computational fluid dynamics (CFD) has been recognized as a promising method to meet the challenges of modelling of NSPS [49, 50]. DEM is a numerical approach for modelling a large number of particles interacting with each other. The simplest computational sequence for the DEM typically proceeds by solving the equations of motion, while updating contact force histories as a consequence of contacts between different discrete elements and/or resulting from contacts with model boundaries. It is designed to deal with very dense suspensions, where contacts between particles are very common and play a key role in determining the motion of particles, see [48] for an extensive overview of DEM.

Obviously, for a particle with a specific shape, the general expressions derived from the first factor shape approach tend to be less accurate than the specialized one for that shape. However, the efficiency of interpolations/extrapolations to the various shapes to provide

the general expression is an attractive perspective on engineering applications. On the other hand, particles occurring in biological processes are usually very numerous. Therefore, an effective method not only has to be accurate but also efficient in terms of computational time.

To overcome the aforementioned limitations in modelling of NSPS we employ the recently trending approach and use machine learning to design a method that enables us to model the behaviour of suspensions of particles of an arbitrary shape while maintaining the accuracy of shape-specific models. We also place an emphasis on the computational efficiency as usually there are plenty of particles involved in medical processes and engineering problems.

3 Model Description

This section describes a general numerical framework for suspensions of particles in a Navier–Stokes fluid. The discretization of the Navier–Stokes equations is realized in the finite element toolbox *Gascoigne 3D* and outlined in Section 3.1. Then, in Section 3.2 we describe a model for the motion of the particles.

3.1 Fluid Dynamics

Consider a finite time interval $I = [0, T]$ and a bounded domain $\Omega \in \mathbb{R}^d$ for $d \in \{2, 3\}$. We assume incompressibility of fluid, which is modelled by the Navier–Stokes equations that take the form

$$\begin{aligned} \rho(\partial_t \mathbf{v} + (\mathbf{v} \cdot \nabla) \mathbf{v}) - \operatorname{div} \boldsymbol{\sigma}(\mathbf{v}, p) &= 0, \\ \operatorname{div} \mathbf{v} &= 0, \end{aligned}$$

where \mathbf{v} denotes the fluid velocity, $\boldsymbol{\sigma}$ is the Cauchy stress tensor

$$\boldsymbol{\sigma}(\mathbf{v}, p) = \rho\nu(\nabla \mathbf{v} + \nabla \mathbf{v}^T) - p\mathbf{I},$$

p the pressure, ρ the fluid mass density and ν the kinematic viscosity. Fluid density and viscosity are assumed to be nonnegative and constant.

The fluid boundary is split into an inflow boundary Γ_{in} , an outflow boundary Γ_{out} and rigid no-slip wall boundaries Γ_{wall} . On the inflow and walls we impose Dirichlet boundary conditions while on the outflow we apply the do-nothing condition (see e.g. [15])

$$\begin{aligned} \mathbf{v} &= \mathbf{v}_{in} && \text{on } \Gamma_{in} \times I, \\ \mathbf{v} &= 0 && \text{on } \Gamma_{wall} \times I, \\ (\rho\nu \nabla \mathbf{v} - p\mathbf{I})\mathbf{n} &= 0 && \text{on } \Gamma_{out} \times I, \end{aligned}$$

where \mathbf{v}_{in} is prescribed inflow-profile and \mathbf{n} is the outward unit normal vector.

Discretization For temporal discretization of the Navier–Stokes equations we introduce a uniform partitioning of the interval $I = [0, T]$ into discrete steps

$$0 = t_0 < t_1 < \dots < t_N = T, \quad k := t_n - t_{n-1}.$$

By $\mathbf{v}_n := \mathbf{v}(t_n)$ and $p_n := p(t_n)$ we denote the approximations at time t_n . We use a shifted version of the Crank–Nicolson time discretization scheme which is second order accurate

and which has preferable smoothing properties as compared to the standard version, see [14, Remark 1], i.e.

$$\rho \left(\frac{\mathbf{v}_n - \mathbf{v}_{n-1}}{k} + \theta(\mathbf{v}^n \cdot \nabla)\mathbf{v}^n + (1 - \theta)(\mathbf{v}^{n-1} \cdot \nabla)\mathbf{v}^{n-1} \right) - \theta \operatorname{div} \boldsymbol{\sigma}(\mathbf{v}^n, p^n) - (1 - \theta) \operatorname{div} \boldsymbol{\sigma}(\mathbf{v}^n, p^n) = 0,$$

where, typically, $\theta = \frac{1+k}{2}$. For spatial discretization we denote by Ω_h a quadrilateral (or hexahedral) finite element mesh of the domain Ω that satisfies the usual regularity assumptions required for robust interpolation estimates, see [30, Section 4.2]. Adaptive meshes are realized by introducing at most one hanging node per edge. Discretization is based on second order finite elements for pressure and velocity. To cope with the lacking inf-sup stability of this equal order finite element pair we stabilize with the local projection method [2]. Local projection terms are also added to stabilize dominating transport [3]. Finally, velocity $\mathbf{v}_n \in [V_h]^2$ and pressure $p_n \in V_h$ (where we denote by V_h the space of bi-quadratic finite elements on the quadrilateral mesh) are given as solution to

$$\begin{aligned} & (\rho \mathbf{v}_n, \phi_h)_\Omega + k\theta(\rho(\mathbf{v}_n \cdot \nabla)\mathbf{v}_n, \phi_h)_\Omega + k\theta(\rho v(\nabla \mathbf{v}_n + \nabla \mathbf{v}_n^T), \nabla \phi_h)_\Omega \\ & - k(p_n, \operatorname{div} \phi_h)_\Omega + k(\rho \operatorname{div} \mathbf{v}_n, \xi_h)_\Omega \\ & + k \sum_{K \in \Omega_h} \alpha_K (\nabla(p_n - \pi_h p_n), \nabla(\xi_h - \pi_h \xi_h))_K \\ & + k \sum_{K \in \Omega_h} \alpha_K \left((\mathbf{v}_n \cdot \nabla)(\mathbf{v}_n - \pi_h \mathbf{v}_n), (\mathbf{v}_n \cdot \nabla)(\phi_h - \phi_h^\xi) \right)_K \\ = & (\rho \mathbf{v}_{n-1}, \phi_h)_\Omega - k(1 - \theta) (\rho(\mathbf{v}_{n-1} \cdot \nabla)\mathbf{v}_{n-1}, \phi_h)_\Omega \\ & - k(1 - \theta) \left(\rho v(\nabla \mathbf{v}_{n-1} + \nabla \mathbf{v}_{n-1}^T), \nabla \phi_h \right)_\Omega \quad \forall (\phi_h, \xi_h) \in [V_h]^2 \times V_h, \end{aligned} \tag{1}$$

where the stabilization parameters are element-wise chosen as [5]

$$\alpha_K = \alpha_0 \left(\frac{\nu}{h_K^2} + \frac{\|\mathbf{v}\|_{L^\infty(K)}}{h_K} + \frac{1}{k} \right)^{-1},$$

where $h_K = \operatorname{diam}(K)$ is the diameter of the element K . Usually we choose $\alpha_0 = 0.1$. By $\pi_h : V_h \rightarrow V_h^{(1)}$ we denote the interpolation into the space of bi-linear elements on the same mesh Ω_h .

Solution of the discretized problem Discretization by means of (1) gives rise to a large system of nonlinear algebraic equation which we approximate by a Newton scheme based on the analytical Jacobian of (1). The resulting linear systems are solved by a GMRES iteration (Generalized minimal residual method [32]), preconditioned by a geometric multi-grid solver [1]. As smoother we employ a Vanka-type iteration based on the inversion of the submatrices belonging to each finite element cell. These local 27×27 (108×108 in 3d) matrices are inverted exactly. Essential parts of the complete solution framework are parallelized using OpenMP, see [10].

3.2 Particle Dynamics

The particles suspended in the fluid are described as rigid bodies and their dynamics is driven by the hydrodynamical forces of the flow. Each particle P with the center of

mass \mathbf{x}_P , the velocity \mathbf{V}_P and the angular velocity $\boldsymbol{\Omega}_P$ is governed by Newton’s law of motion

$$m_P \mathbf{X}_P''(t) = \mathbf{F}(\mathbf{v}, p; P) := - \int_{\partial P} \boldsymbol{\sigma}(\mathbf{v}, p) \mathbf{n}_P \, ds,$$

$$\mathbf{J}_P \boldsymbol{\Omega}_P'(t) = \mathbf{T}(\mathbf{v}, p; P) := \int_{\partial P} (\mathbf{x} - \mathbf{x}_P) \times (\boldsymbol{\sigma}(\mathbf{v}, p) \mathbf{n}_P) \, ds,$$

where m_P is the particle’s mass, and \mathbf{J}_P its moment of inertia given by

$$\mathbf{J}_P = \text{diag} \left\{ \rho_P \int_P (|\mathbf{x} - \mathbf{x}_P|^2 - (\mathbf{x}_i - [\mathbf{x}_P]_i)^2) \, d\mathbf{x}; i = 1, 2, 3 \right\},$$

with the (uniform) particle density ρ_P and \mathbf{n}_P being the unit normal vector on the particle boundary facing into the fluid.

A resolved simulation is out of bounds due to the large number of platelets and in particular due to the discrepancy in particle diameter (about 10^{-6} m) versus vessel diameter (about 10^{-3} m). Instead, we consider all platelets to be point-shaped and determine traction forces $\mathbf{F}(\mathbf{v}, p; P)$ and torque $\mathbf{T}(\mathbf{v}, p; P)$ based on previously trained neural networks. These coefficients will depend on the shape and the size of the particles but also on their relative orientation and motion in the velocity field of the fluid. Since the relative velocities (blood vs. particles) are very small the interaction lies within the Stokes regime with a linear scaling in terms of the velocity. The deep neural network will predict coefficients for drag C_d , lift C_l , pitching torque C_p and rotational torque C_r . The resulting forces exerted on each particle P are given by

$$\mathbf{F}_P = C_d(P, \psi_P)(\mathbf{v} - \mathbf{V}_P) + C_l(P, \psi_P)(\mathbf{v} - \mathbf{V}_P)^\perp,$$

$$\mathbf{T}_P = C_p(P, \psi_P)|\mathbf{v} - \mathbf{V}_P| + C_r(P, \psi_P)(\boldsymbol{\omega} - \boldsymbol{\Omega}_P),$$

where $P = (L_x, L_y, L_z, \alpha_{top}, \alpha_{bot})$ describes the particle shape and where ψ_P is the relative angle of attack which depends on the particle orientation but also on the relative velocity vector between blood velocity and particle trajectory, see Fig. 2. The coefficient functions C_d , C_l , C_p and C_r will be trained based on detailed numerical simulations using random particles in random configurations. By $(\mathbf{v} - \mathbf{V}_P)^\perp$ we denote the flow vector in lift-direction, orthogonal to the main flow direction. In 3d configurations, two such lift coefficients must be trained. Here we will however only consider 2d simplifications with one drag and one lift direction. Since we are in the Stokes regime, the representation of the fluid mechanical components is simplified. The neural network approach described in the next section will require very little training data, since the Reynolds number does not have to be considered as a free parameter in the fluid-particle interaction.

4 An Artificial Neural Network Model for Predicting Hydrodynamical Parameters

In this section we describe the neural network model used for coupling the Navier–Stokes equations with a suspension of non-spherical particles. The different hydrodynamical coefficients will be taken from a neural network, which is trained in an offline phase. Training data is achieved by resolved Navier–Stokes simulations using prototypical particles with random parameters.

The setting investigated in this work carries several special characteristics that differ from industrial applications.

- The particle density is very small—about $1.04\text{--}1.08 \cdot 10^3 \text{ g l}^{-1}$ and the particle and the fluid densities are similar (the average density of whole blood for a human is about $1.06 \cdot 10^3 \text{ g l}^{-1}$). Blood contains about 200 000–400 000 platelets per mm^3 summing up to less than 1% of the overall blood volume [39]. Hence we neglect all effects of the particles onto the fluid. This simplification is possible since we only model the platelets as rigid particles. Effects of the red blood cells, much larger and appearing in greater quantity, can be integrated by means of a non-Newtonian rheology.
- The particle Reynolds numbers are very small (with order of magnitude about 10^{-4} or less) such that we are locally in the Stokes regime. This is mainly due to the smallness of the platelets (diameter approximately $3 \mu\text{m}$) and the small flow velocities at (bulk) Reynolds numbers ranging from 50 to 1 000 depending on the specific vessel under investigation. We focus on coronary vessels with a diameter around 2 mm and with Reynolds number about 200.
- The platelets have a strongly non-spherical, disc-like shape. Their shape and size underlie a natural variation. Furthermore, under activation, the particles will take a spherical shape.

Instead of deriving analytical models for the transmission of forces from the fluid to the particles, we develop a neural network for the identification of drag, lift and torque coefficients based on several parameters describing the shape and the size of the platelets and the individual flow configuration.

4.1 Parametrization of the Platelets

We model the platelets as variations of an ellipsoid with major axes $L_x \times L_y \times L_z$ with $L_x \approx L_z \approx 3 \mu\text{m}$ and $L_y \approx 0.5 \mu\text{m}$. In y -direction upper α_{top} and lower α_{bot} semi-ellipsoids are modified to give them a more or less concave or convex shape. Altogether, each particle is described by a set of 5 parameters $P = (L_x, L_y, L_z, \alpha_{top}, \alpha_{bot})$. The surface of the platelets is given as zero contour of the levelset function

$$\Phi(P; x, y, z) = 1 - R(P)^2 - \left(\alpha + (1 - \alpha)R(P)^2 \right)^{-2} \left(\frac{2y}{L_y} \right)^2,$$

where we define

$$R(P)^2 := \left(\frac{2x}{L_x} \right)^2 + \left(\frac{2z}{L_z} \right)^2$$

and

$$\alpha := \begin{cases} \alpha_{top}, & y \geq 0, \\ \alpha_{bot}, & y < 0. \end{cases}$$

We assume that all parameters $L_x, L_y, L_z, \alpha_{top}, \alpha_{bot}$ are normally distributed with means indicated above and with standard deviation 0.3 for the lengths L_x, L_y, L_z and 0.4 for the shape parameters $\alpha_{top}, \alpha_{bot}$. We drop particles that exceed the bounds

$$L_x, L_z \in [2.5 \mu\text{m}, 3.5 \mu\text{m}], \quad L_y \in [0.15 \mu\text{m}, 1 \mu\text{m}], \quad \alpha_{top}, \alpha_{bot} \in [0.2, 2]. \quad (2)$$

In Fig. 1 we show some typical shapes of the platelets.

Next, we indicate mass, center of mass and moment of inertia for a parametrized particle $P = (L_x, L_y, L_z, \alpha_{top}, \alpha_{bot})$. Unless otherwise specified, all quantities are given in μm and g . The mass of a particle P is approximated by

$$m(P) := \rho_P L_x L_y L_z 0.2116 \left(\sqrt{\alpha_{bot} + 0.5313} + \sqrt{\alpha_{top} + 0.5313} \right).$$

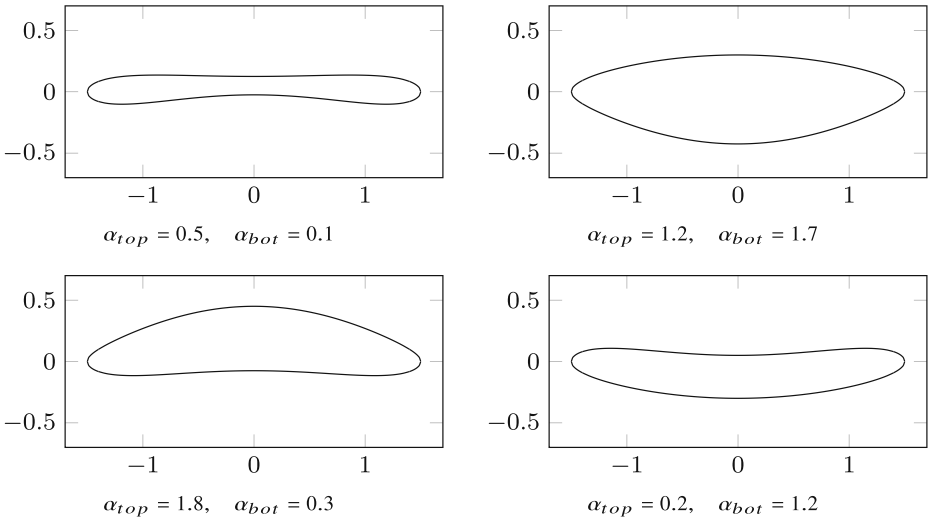


Fig. 1 Prototypical templates for variations of the parameters L_x, L_y, L_z (length) and $\alpha_{top}, \alpha_{bot}$ (shape). Within the blood flow, particles can appear at all angles of attack ϕ

This approximation is based on a weighted one-point Gaussian quadrature rule. It is accurate with an error of at most 2% for all $\alpha \in [0.2, 2]$.

The center of mass for a particle P is given by

$$m_x(P) = m_z(P) = 0, \quad m_y(P) = \frac{L_x L_y^2 L_z \pi}{96m(P)} (\alpha_{top} - \alpha_{bot}).$$

The moment of inertia in the x/y plane (the only axis of rotation that we will consider in the 2d simplification within this work) is given by

$$I_z(P) = \rho_P \int_P (x^2 + y^2) d(xyz) \approx \frac{L_x L_y L_z \rho_P \pi}{8} \left(0.24 + 0.12 (\alpha_{top} + \alpha_{bot}) + 0.0236 (\alpha_{top}^2 + \alpha_{bot}^2) \right),$$

which is accurate up to an error of at most 1% for all $\alpha \in [0.2, 2]$. These coefficients are computed once for each particle and stored as additional parameters.

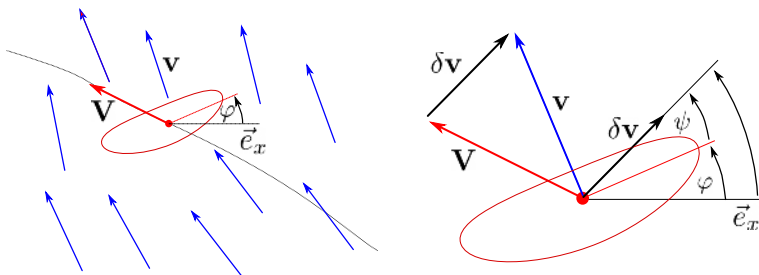
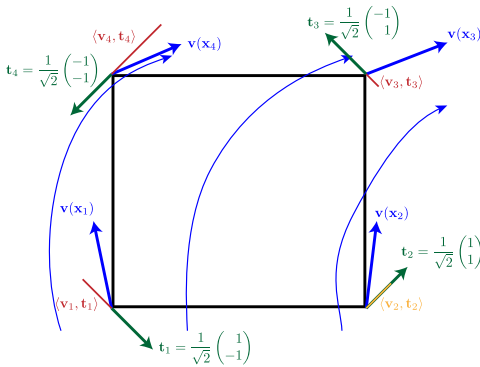


Fig. 2 Left: typical configuration of one platelet (in red) within velocity field of the blood (blue arrows). The angle ϕ is the orientation of the platelet relative to its standard orientation. Right: the black arrow $\delta \mathbf{v} = \mathbf{v} - \mathbf{V}$ is the velocity acting on the platelet. By $\psi := \angle(\delta \mathbf{v}, \mathbf{e}_x) - \phi$ we denote the effective angle of attack



We denote by $\mathbf{v}(\mathbf{x}_i)$ the Navier–Stokes velocity at node \mathbf{x}_i , by \mathbf{t}_i the unit-tangential vector at this node in counter-clockwise orientation. With $\langle \mathbf{v}_i, \mathbf{t}_i \rangle$ we denote the velocity contribution in tangential direction. Positive values are indicated in orange (at node \mathbf{x}_2), negative contributions are in red.

Fig. 3 Approximation of the rotational velocity on an element T

2d Simplification To start with, we apply a two dimensional simplification of the problem by assuming that the blood vessel is a layer of infinite depth (in z -direction) and that it holds $\mathbf{v}_3 = 0$ for the blood velocity and $\mathbf{V}_3 = 0$ for all particles. Further, given the symmetry of the particles w.r.t. rotation in the x/y -plane, no traction forces in z -direction will appear. Besides that, rotation is restricted to rotation around the x/y -plane. Hence, $\boldsymbol{\Omega} = (0, 0, \omega)$ is described by a scalar component. A complete particle is then described by

$$P = (L_x, L_y, L_z, \alpha_{top}, \alpha_{bot}; \mathbf{X}, \varphi, \mathbf{V}, \boldsymbol{\Omega}),$$

where $L_x, L_y, L_z, \alpha_{top}, \alpha_{bot}$ are the shape parameters and, \mathbf{X} is the (2d) position, φ the orientation w.r.t. the z -axis, \mathbf{V} the (2d) velocity and $\boldsymbol{\Omega}$ the angular velocity w.r.t. the z -axis rotation.

The two dimensional simplification is used in Section 5 to validate the effects of non-sphericity in a coupled Navier–Stokes particle simulation based on the neural network output. The training and testing of the neural network is however completely based on the full three dimensional Navier–Stokes problem.

To describe the forces acting on the particle suspended in the Navier–Stokes fluid we denote by $\delta \mathbf{v} := \mathbf{v} - \mathbf{V}$ the effective velocity vector, i.e., the relative velocity that is acting on the platelets. By $\psi := \angle(\mathbf{e}_x, \mathbf{v} - \mathbf{V}) - \varphi$ we denote the effective angle of attack which is the angle between relative velocity $\delta \mathbf{v}$ and the current orientation of the platelet, see Fig. 2 (right) and (2). It is computed as

$$\psi := \angle(\mathbf{e}_x, \mathbf{v} - \mathbf{V}) - \varphi. \tag{3}$$

Furthermore, denote by $\delta \omega := \omega(\mathbf{v}) - \omega$ the relative angular velocity. The angular part of the Navier–Stokes velocity is locally reconstructed from the velocity field in every lattice, i.e., in every finite element cell T , by means of

$$\omega_T = \frac{1}{2d_T} \sum_{i=1}^4 \langle \mathbf{v}(\mathbf{x}_i), \mathbf{t}_i \rangle, \tag{4}$$

where \mathbf{x}_i and \mathbf{t}_i , for $i = 1, 2, 3, 4$ are the four nodes and the tangential vectors on the circle passing through all four lattice nodes. By $d_T = \sqrt{2}h_T$ we denote the diameter of the lattice, see also Fig. 3.

4.2 Design of the Artificial Neural Network

We will train a deep neural networks for determining the coefficients C_d , C_l , C_p , and C_r . We use two separate neural networks since the input data for C_d , C_l and C_p depends on the angle of attack ψ , while that of C_r is invariant to the orientation of the particle. We call these artificial neural networks \mathcal{N} and \mathcal{N}_r . Both take the platelet parameters $(L_x, L_y, L_z, \alpha_{top}, \alpha_{bot})$ as input. \mathcal{N} further depends on the effective angle of attack ψ_p . Altogether

$$\mathcal{N} : \mathbb{R}^6 \rightarrow \mathbb{R}^3, \quad \mathcal{N}_r : \mathbb{R}^5 \rightarrow \mathbb{R}.$$

Both neural networks are fully connected feedforward networks with three hidden layers consisting of 50, 20 and 20 neurons in the case of the drag/lift network and 20 neurons each in the case of the rotational torque network. All neurons apart from the output layer are of ReLU type, i.e. using the activation function $f(x) = \max\{x, 0\}$. Figure 4 shows the general configuration. The network was designed by simple parameter studies. Different depths and layer widths have been tested. Further, we investigated whether a combination of coefficients or separate networks for drag, lift and torque are more advantageous. The architectures presented here have proven to be a good compromise between network size and accuracy. The two networks for drag, lift, (pitching) torque and rotational torque are based on different types of input data: while the first larger network takes the angle as configuration-dependent parameter, the second smaller network is invariant to the angle. A combination of both networks could not give satisfactory accuracies.

4.3 Generation of the Training Data

Training and test data is obtained by resolved simulations with random sampling of prototypical platelet shapes. Let $\Omega = \{\mathbf{x} \in \mathbb{R} : |\mathbf{x}|_2 < 50 \mu\text{m}\} \setminus P$ be the open ball with radius $R = 50 \mu\text{m}$ around a platelet P . Each platelet P is constructed by taking normally distributed values for $(L_x, L_y, L_z, \alpha_{top}, \alpha_{bot})$ as indicated in Section 4.1. Since only relative velocity and relative orientation matter, the platelets are fixed in the origin with $\mathbf{V} = 0$ and $\Omega = 0$ at angle $\varphi = 0$.

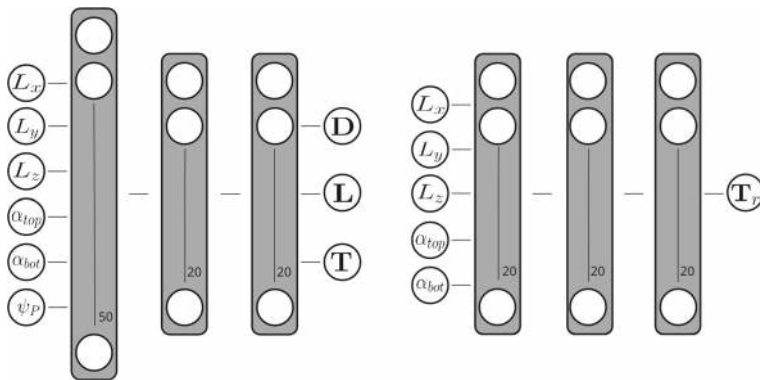


Fig. 4 Architecture for the neural networks \mathcal{N} (left) for predicting drag, lift and pitching torque and \mathcal{N}_r (right) for predicting the rotational torque. Both networks are fully connected feedforward networks with 3 hidden layers, having 50/20/20 and 20/20/20 neurons each

For training, the Stokes equations are formulated in the units μm for length, $\mu\text{m} \cdot \text{s}^{-1}$ for the velocity and μg for mass. With the blood viscosity $\mu = 3 \mu\text{g} \cdot \mu\text{m}^{-1} \cdot \text{s}^{-1}$ the Stokes equations

$$-\mu \Delta \mathbf{v} + \nabla p = 0, \quad \text{div } \mathbf{v} = 0,$$

are considered. We prescribe zero Dirichlet data on the platelet, $\mathbf{v} = 0$ on ∂P , and set a freestream velocity on the outer boundary $\partial \Omega \setminus \partial P$. This is either a uniform parallel flow field or a uniform rotational flow field that corresponds to the rotational velocity $\omega = 2\pi$, both given as Dirichlet data

$$\mathbf{v} = 0 \text{ on } \partial P, \quad \mathbf{v}_{\psi}^d := \begin{pmatrix} \cos(\psi) \\ \sin(\psi) \\ 0 \end{pmatrix} \quad \text{or} \quad \mathbf{v}_{\omega}^r := 2\pi \begin{pmatrix} -y \\ x \\ 0 \end{pmatrix} \quad \text{on } \partial \Omega \setminus \partial P,$$

where $\psi \in [0, 2\pi]$ is the relative angle of attack. For \mathbf{v}^d it holds $|\mathbf{v}_{\psi}^d| = 1 \mu\text{m} \cdot \text{s}^{-1}$ and in case of the rotational flow it holds $|\mathbf{v}_{\omega}^r| = 2\pi R \mu\text{m} \cdot \text{s}^{-1}$, such that it corresponds to a angular velocity of magnitude 2π in counter-clockwise direction around the z -axis.

The training data is generated as follows

Algorithm 1 (Generation of the training data).

Let $N \in \mathbb{N}$ be a prescribed number of experiments. For $n = 1, 2, \dots, N$

1. Generate a random particle P_n that satisfies the bounds (2).
2. For four random angles of attack $\psi_{n,i} \in [0, 2\pi]$, $i = 1, 2, 3, 4$, solve the Stokes equations for the directional Dirichlet data $\mathbf{v}_{\psi_{n,i}}^d$ and compute¹

$$\mathbf{D}_{n,i} := \int_{\partial P} (\mu \nabla \mathbf{v} - pI) \mathbf{n} \cdot \mathbf{v}_{\psi_{n,i}}^d \, ds,$$

$$\mathbf{L}_{n,i} := \int_{\partial P} (\mu \nabla \mathbf{v} - pI) \mathbf{n} \cdot \mathbf{v}_{\psi_{n,i}}^{d,\perp} \, ds,$$

$$\mathbf{T}_{n,i} := \int_{\partial P} (\mathbf{x} - \mathbf{x}_P)^\perp \cdot (\mu \nabla \mathbf{v} - pI) \mathbf{n} \, ds.$$

3. Solve the Stokes equations for the rotational Dirichlet data \mathbf{v}_{ω}^d and compute the rotational torque

$$\mathbf{T}_{n,r} := \int_{\partial P} (\mathbf{x} - \mathbf{x}_P)^\perp \cdot (\mu \nabla \mathbf{v} - pI) \mathbf{n} \, ds.$$

¹By $(\mathbf{x} - \mathbf{x}_P)^\perp$ and $\mathbf{v}_{\psi_{n,i}}^{d,\perp}$ we denote the counter-clockwise rotation of the corresponding vectors by 90°

Hereby, a set of $4N$ training data sets $(P_n, \psi_{n,i}; \mathbf{T}_{n,i})$ and N data sets for the rotational configuration $(P_n; \mathbf{T}_{n,r})$ are generated in an offline phase. Two different networks will be used for these two different settings.

The domain Ω is meshed with hexahedral elements and the finite element discretization is build on equal-order tri-quadratic finite elements for velocity and pressure. The curved boundaries (both the outer boundary and the platelet boundary) are approximated in an isoparametric setup to avoid dominating geometry errors see [30, Sec. 4.2.3]. A very coarse mesh with initially only 12 hexahedras is refined twice around the platelet boundary and

once globally. The resulting discretization has about 2 000 elements and 60 000 degrees of freedom. Details on the discretization are given in Section 3. The resulting (stationary) discrete finite element formulation is given by

$$\begin{aligned} \rho v(\nabla \mathbf{v}, \nabla \phi) - (p, \operatorname{div} \phi) &= 0 & \forall \phi \in [V_h]^2, \\ (\operatorname{div} \mathbf{v}, \xi) + \sum_{T \in \Omega_h} \frac{h_T^2}{\nu} (\nabla(p - \pi_h p), \nabla(\xi - \pi_h \xi))_T &= 0 & \forall \xi \in V_h. \end{aligned}$$

For the Stokes equation no transport stabilization must be added.

Given \mathbf{v} , p the resulting forces are computed by

$$\mathbf{F} = \int_{\partial P} (\mu \nabla \mathbf{v} - pI) \mathbf{n} \, d\sigma, \quad T = \int_{\partial P} (\mathbf{x} - \mathbf{x}_P)^\perp \cdot (\mu \nabla \mathbf{v} - pI) \mathbf{n} \, d\sigma.$$

The units of \mathbf{F} and T are

$$[\mathbf{F}] = \frac{\mu \text{m} \cdot \mu \text{g}}{\text{s}^2}, \quad [T] = \frac{\mu \text{m}^2 \cdot \mu \text{g}}{\text{s}^2}. \tag{5}$$

Training and test data is computed on an Intel Xeon E5-2640 CPU at 2.40 GHz using 20 parallel threads. A total of 58 500 data sets (46 600 for drag and lift, 11 900 for measuring the torque) have been generated. The overall computational time for all these 3d simulations was about 9 hours (less than one second for each simulation). All computations are done in Gascoigne 3D. In Fig. 5 we show snapshots of three such simulations.

4.3.1 Preparation and Normalization of Data/Training of the Neural Network

We produce a data set with N entries with random particles. We start by extracting drag, lift and torque according to Algorithm 1. To prepare the input data we encode as much model knowledge as possible. Assume that \mathbf{D} , \mathbf{L} , \mathbf{T} , \mathbf{T}_r are the vectors containing drag, lift and pitching torque and rotational torque. Then, we define the input data as (component-wise)

$$\mathbf{d} := \frac{\mathbf{D}}{70 - 10 \cos(2\varphi)}, \quad \mathbf{l} := \frac{\mathbf{L}}{10 \sin(2\varphi)}, \quad \mathbf{t}_p := \frac{\mathbf{T}}{2 \cos(\varphi)}, \quad \mathbf{t}_r := \frac{\mathbf{T}_r}{800}. \tag{6}$$

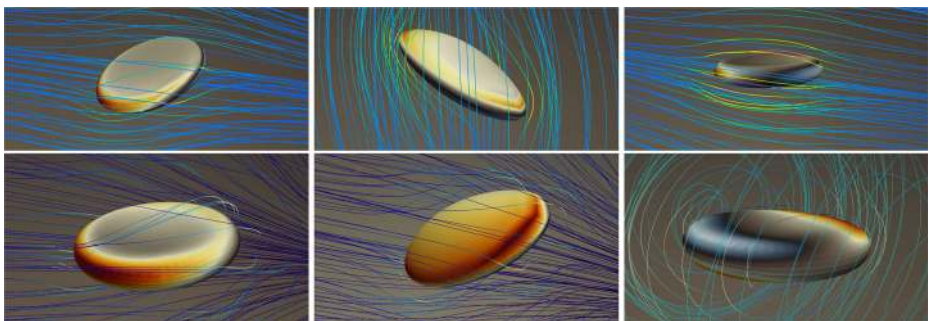


Fig. 5 Visualization of the resolved flow pattern around randomly created particles. The platelets vary in size ($L_x \times L_y \times L_z$) and in their convexity, further we vary the angle of attack. The upper row shows three simulations with random particle using different angles of attack. In the lower row, we consider the same particle for each of the three simulations. The two figures on the left correspond to the directional inflow at different angles of attack while the plot on the right corresponds to a simulation with the rotational Dirichlet pattern on the outer boundary of the domain

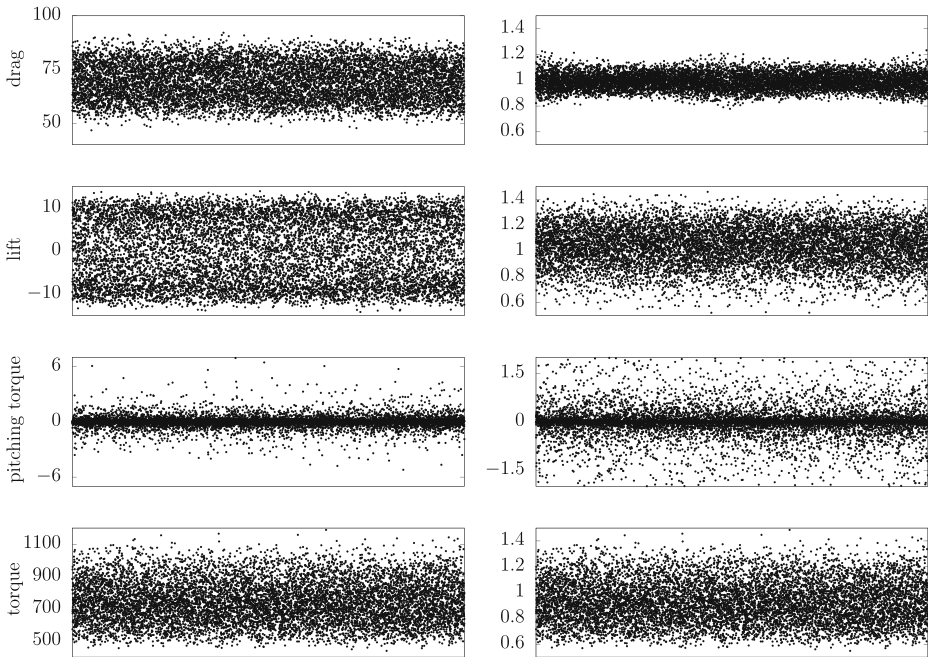


Fig. 6 Visualization of the training data. Left: raw values coming from 5000 experiments (plotted along the x -axis) with randomly generated platelets and random variation of the angle of attack. Right: scaled input data for the neural network according to (6). By pre-scaling the forces we can reduce the variation to about 20 – 40%. This remaining dependency of the quantities on the platelet- and flow-parameters will be learned in the artificial neural network

These simple relations have been found manually by analyzing the relation of the functional outputs on the different parameters. This scaling reduces the variation of the forces over all experiments to about 10 – 40% in the case of drag, lift and rotational force, see Fig. 6. A rescaling of the pitching torque (which has a rather low value) is more difficult since it depends on slight variations of the particle symmetry. The normalization of the data is required to equilibrate the inputs of the mean squares loss function and yields in better approximation results. Furthermore, the consideration of simple angle-dependent effects helped to speed up training process.

The neural network is implemented in *PyTorch* [28], using the *PyTorch C++ API* which has already been linked to our finite element framework *Gascoigne 3D* [4] in a neural network multigrid approach [16]. The randomly generated data sets originating from the detailed Navier–Stokes simulations are split into 80% serving as training data and the remaining 20% as test data. As loss function we consider square l_2 norm of the error. The training of the two very small networks is accomplished in a few minutes.

4.3.2 Testing

To test the accuracy of the trained network we apply it to a set of testing data that was not used in the training of the network. In Fig. 7 we show for drag, lift and torque, 250 data points each that have been randomly taken from the test data set such that these data pairs have not been used in training. In the figure, we indicate the *exact values* as taken from

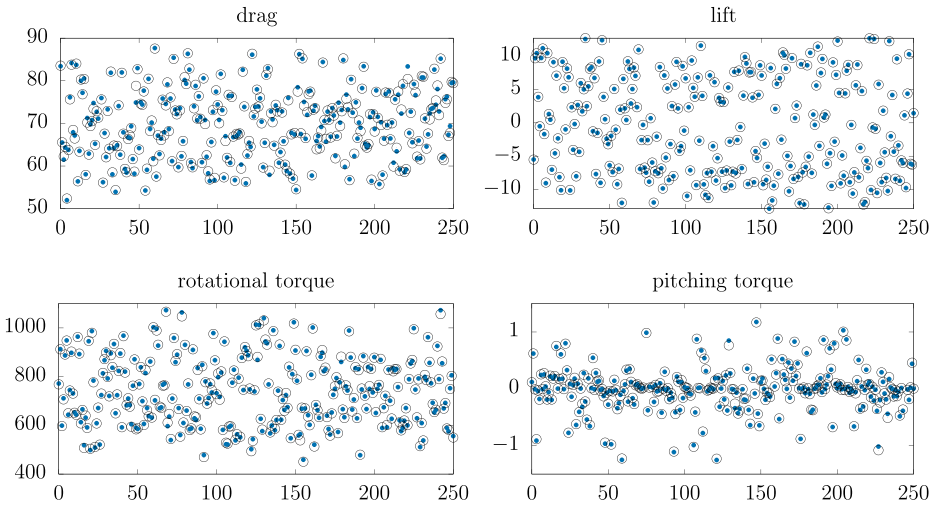


Fig. 7 Performance of the neural network in predicting drag, lift and torque coefficients for the flow around randomly created platelets. For 250 random particles each (all have not been used in training the network) we compare the prediction (blue bullets) with the coefficients obtained in a resolved finite element simulation. The coefficients are given in the units of the test reference system described in (5)

detailed finite element simulations that resolve the particle as large circles and the predicted DNN output as smaller bullets. We observe very good agreement in all three coefficients, best performance in the lift coefficient and highest deviation in the drag.

In Table 1 we indicate the mean (measured in the l_2 -norm) and the maximum error of the network applied to the training data and to the test data. Further, for getting an idea on the generalizability of the approach we also apply the network to additional testing data with random platelets, where at least one of the coefficients ($L_x, L_y, L_z, \alpha_{top}, \alpha_{not}$) does not satisfy the bounds specified in (2). We note that such particles are not appearing in the coupled Navier–Stokes particle simulation framework. The average errors appearing in all training and testing data is less than 1%. Maximum relative errors for few single particles reach values up to 4% in the case of the pitching torque, which is most sensitive with values close to zero. Even if we consider data points that are not within the bounds, as shown in the generalization test, average errors are still small, although substantial errors are found for

Table 1 Accuracy of the neural network model for predicting drag, lift and pitching and rotational torque in percent

	training data		test data		generalization	
	avg	max	avg	max	avg	max
Drag	0.40%	1.44%	0.40%	1.43%	0.87%	5.81%
Lift	0.34%	1.32%	0.34%	1.33%	0.65%	7.40%
Pitching torque	0.93%	3.75%	0.95%	3.87%	2.39%	12.76%
Rotational torque	0.45%	1.57%	0.45%	1.50%	1.03%	4.12%

We indicate the values for the training data, the test data and the hard test data that consists of data points outside the bounds (2)

single particles. Largest error are given for large values of the shape parameters α_{top} , α_{bot} , in particular for $\alpha_{top} > 2$ or $\alpha_{bot} > 2$. Less than 0.5% of the training data exceeds this threshold.

4.4 Application of the Neural Network

The neural network predicts the coefficients for drag \mathbf{d} , lift \mathbf{l} , pitching torque \mathbf{t}_p and rotational torque \mathbf{t}_r , which are scaled according to (6). While drag, lift and pitching torque depend on the effective angle of attack, the rotational torque is a fixed value that must be predicted only once for each particle. The former three values are recomputed whenever the configuration is changing, i.e., before every advection step.

Algorithm 2 (Neural network/particle/finite element coupling).

Let k be the macro time step used to predict the blood flow field, $k_p := k/M_p$ be the subcycling step for the particle dynamics and N_p be the number of particles. For $n = 1, 2, \dots$ iterate

1. Solve the Navier–Stokes equations $t_{n-1} \mapsto t_n$.
2. Transfer the local velocities from the finite element mesh to the particle lattice and locally compute the rotational velocity according to (4).
3. For $m = 1, \dots, M_p$ subcycle the particle dynamics with step size k_p

- (a) For each particle $\{P_1, \dots, P_{N_p}\}$, compute the effective angle of attack according to (3);
- (b) Evaluate the deep neural network for all particles

$$(L_x, L_y, L_z, \alpha_{top}, \alpha_{bot}, \psi)_i \mapsto (\mathbf{d}, \mathbf{l}, \mathbf{t}_p, \mathbf{t}_r)_i, \quad i = 1, \dots, N_p;$$

- (c) Rescale the coefficients according to (6) and to correct the reference units²

$$\begin{aligned} \mathbf{D} &= 10^{-6} \cdot (70 - 10 \cos(2\psi))\mathbf{t}, & \mathbf{L} &= 10^{-6} \cdot 10 \sin(2\psi)\mathbf{l}, \\ \mathbf{T}_p &= 10^{-9} \cdot 2 \cos(\psi)\mathbf{t}_p, & \mathbf{T}_r &= 10^{-9} \cdot 800\mathbf{t}_r; \end{aligned}$$

- (d) Advect all particles and perform collisions according to Remark 1.
-






² While the blood flow configuration is based in mm and g, the particle configuration is set up in μm and μg

Remark 1 (Collisions) In order to detect and perform collisions we treat particles as spheres with radius L_x and use model described in [22]. Since platelets constitutes only small part of the blood volume (less than 1%) collisions between them happen very rarely and this simplification does not affect validity of presented approach.

Usually we choose $M_p = 100$ subcycling iterations within each macro step. For further acceleration, steps 3(a)–3(c) of the inner loop can be skipped in most of these inner iterations and it will be sufficient to recalculate the coupling coefficients in approximately every tenth step.

Remark 2 (Parallelization) Steps 2, 3(a)–3(d) are parallelized using OpenMP. Particles are organized on a lattice mesh. The dimensions of the lattice are generated such that a small number of particles reside in each lattice. This gives a natural way for parallelization and

Table 2 Shape and parameters of 5 test particles. The spatial dimensions are given in μm

Particle	Shape	Lx	Ly	Lz	α_1	α_2
1		3.0	0.5	3.0	1.0	1.0
2		3.1	0.8	2.5	1.7	1.7
3		2.7	0.2	3.2	0.3	0.3
4		3.1	0.7	2.8	0.3	1.7
5		2.8	0.3	3.1	1.7	0.3

it also helps to keep the communication for performing particle-particle interactions local, compare [25] for details on this approach and for a review on further realization techniques.

Step 3(b) of the algorithm involves the evaluation of the deep neural network. Here, we integrate C++ bindings of the library PyTorch [28] into Gascoigne [4]. All particles are processed at once, such that the evaluation can be performed efficiently in the core of PyTorch. Considering larger networks or a larger number of particles, the use of a CUDA implementation is possible without further effort.

Finally, Step 1 of the algorithm requires to solve the Navier–Stokes equations in a finite element framework. The parallel framework that is used in Gascoigne 3D is described in [10, 19].

5 Numerical Examples

5.1 Evaluation of the Navier–Stokes/DNN Particle Coupling

We study how the different shapes of the particles affect their movement and whether the neural network model is able to give distinguished responses for different particle types, even if the variations of the considered particles are small. In order to do that we examine hydrodynamic forces acting on differently shaped particles. Simulations were performed for five particles (shown in Table 2) representing various shape features (symmetric, asymmetric, convex, convey and combinations).

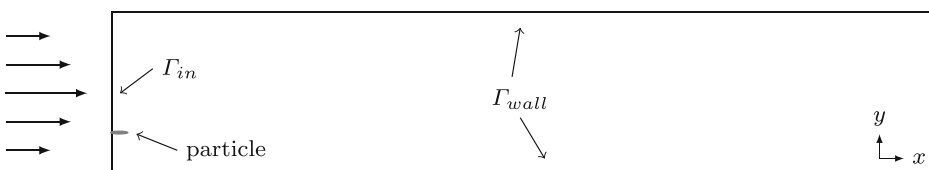


Fig. 8 Spatial configuration of the considered model

Our domain is a channel of diameter $L = 2$ mm and infinite length. The schematic geometry of the domain is described in Fig. 8. Platelets are variations of ellipsoids with major axes $L_x \times L_y \times L_z$ with $L_x \approx L_z \approx 3 \mu\text{m}$ and $L_y \approx 0.5 \mu\text{m}$, for more details see Section 4.1. The inflow data is defined by a time dependent parabolic inflow profile with inflow speed $v_{max} = 5$ mm/s, namely

$$\mathbf{v}_{in}(t) := y \frac{2-y}{2} t \cdot v_{max}.$$

All five particles are initially located at $y = 0.5176$, below the symmetry axis of the velocity profile such that a rotational velocity field attacks the particles. The fluid viscosity is set to $\mu = 3$ mg/mm · s, and particle and fluid density equal $\rho = \rho_p = 1.06$ mg/mm³. The parameters have been chosen so as to reflect a typical vessel, and realistic blood and platelet properties.

The simulations are carried out with the coupled interaction loop described in Algorithm 2. This means that after each Navier–Stokes step, the fluid velocity is transferred to the particle model and the coupling coefficients drag, lift and torques are updated based on the previously trained neural network. Detailed simulations around different particle shapes only enter the training phase by generating random data sets.

5.1.1 Drag

Drag is a force acting opposite to the relative motion of the particle moving with respect to a surrounding fluid. Shape-specific drag coefficients present in the literature are usually functions of the particle Reynolds number, angle of incidence and some shape parameters [17, 23, 45], while drag force itself usually depends on the properties of the fluid and on the size, shape, and speed of the particle.

In Fig. 9 the drag coefficient is plotted as a function of the angle of incidence (effective angle of attack) for five considered particles. The first main observation lies in the increase

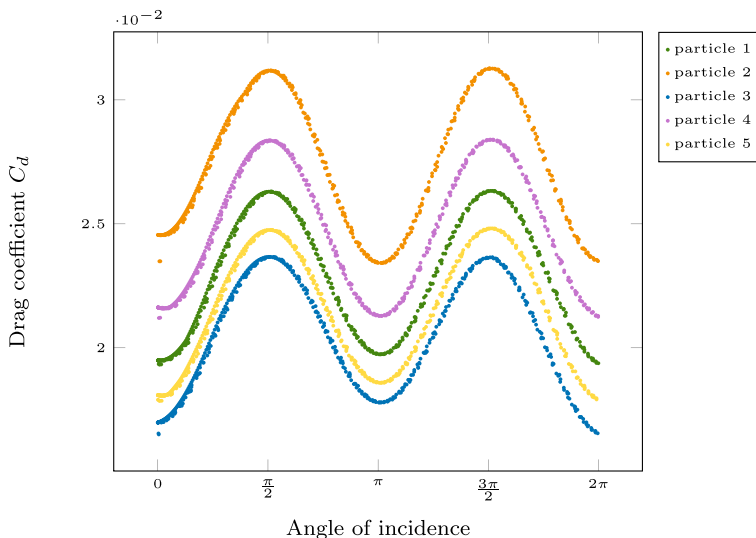


Fig. 9 Drag coefficient as a function of angle of incidence for the five particles defined in Table 2

of the drag values when the angle of incidence approaches $\psi = \frac{\pi}{2}$ and $\psi = \frac{3\pi}{2}$ so when particle is perpendicular to the flow, which means the biggest cross sectional area with respect to the flow. Correspondingly, the drag decreases when the angle of incidence reaches $\phi = 0$ or $\phi = 2\pi$ so when the cross sectional area gets smaller. Qualitatively, the present results are in good agreement with those reported in the literature since a similar trend is observed (see e.g. [33]).

Furthermore, Fig. 9 shows various drag coefficient values for different particles. Particle 2 is characterized by the highest value of the drag. Reasons may be threefold: big size of the particle in comparison to others and hence bigger cross sectional area and higher particle Reynolds number. In contrast, particle 3 is characterized by the lowest value of the drag, which is a result of its small size in comparison to other particles. Particle 1 is an ellipse and serves as a reference. Its drag coefficient is in the middle which is in the line with intuition - particle 1 has intermediate values both in terms of size and convexity/concavity.

5.1.2 Lift

Lift force on a particle is a result of non-axisymmetric flow field. The pressure distribution on the surface of a particle inclined to the flow direction no longer follows the symmetry of that particle. This gives rise to a lift force due to the displacement of the center of pressure. Lift acts in the direction perpendicular to the fluid velocity and is present when the particles principle axis is inclined to the main flow direction. As in the case of drag, lift coefficient is usually a function of the particle Reynolds number, angle of incidence and some shape parameters [18, 26, 45], while lift force itself usually depends on the properties of the fluid and on the size, shape, and speed of the particle.

The lift coefficient behaviour at various angles of incidence for five studied particles is presented in Fig. 10. The figure shows that the lift coefficient reaches its maximum when the angle of incidence reaches $\psi = \frac{\pi}{4}$ or $\psi = \frac{5\pi}{4}$ and its minimum when $\psi = \frac{3\pi}{4}$ or $\psi = \frac{7\pi}{4}$

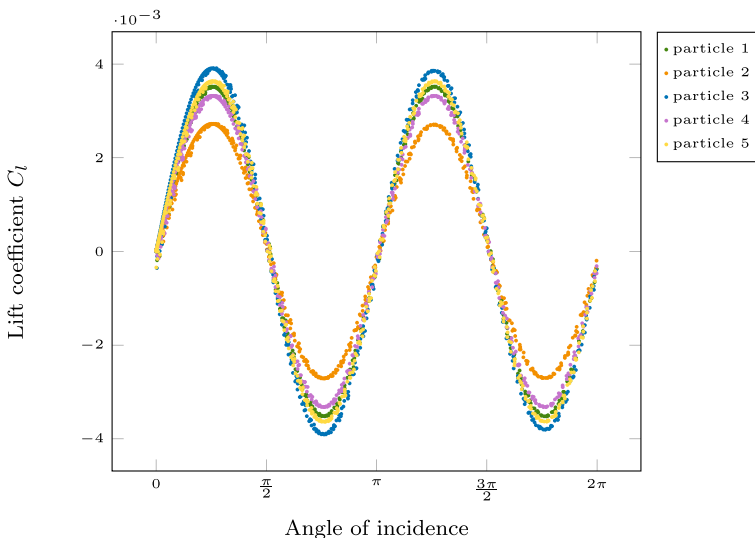


Fig. 10 Lift coefficient as a function of angle of incidence for the five particles defined in Table 2

and is equal to 0 for $\psi \in \{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$. These results are consistent with the definition of the lift force and are similar to other studies (see e.g. [27, 33]).

Moreover, one can notice that the lift coefficient takes the lowest value for particle 3 and the highest value for particle 2. It results from the difference in surface area, which is small for particle 3 and big for particle 2. Similarly to the drag, the lift of the reference particle 1 is in the middle which also corresponds to the intermediate value of its surface area.

5.1.3 Rotational Torque

There are two contributions to the rotational motion of the particle. The first is the inherent fluid vorticity, which acts on the particle as a torque due to the resistance on a rotating body.

Figure 11 illustrates the rotational torque coefficient plotted as a function of the angle of incidence for five examined particles. One can notice that the magnitude of the coefficients corresponds to the surface area of particle, with particle 2's rotational torque coefficient being the largest, while particle 3 experiencing the smallest rotational torque.

These results are consistent with the definition of the rotational torque and qualitatively are similar to those obtained in the literature (see e.g. [45]).

5.1.4 Pitching Torque

Since the center of pressure of the total aerodynamic force acting on each particle does not coincide with the particle's center of mass, a pitching torque is generated. This is the second factor that contributes to rotational motion. It accounts for the periodic rotation of the particle around an axis parallel to the flow direction.

In Fig. 12 the pitching torque coefficient is plotted as a function of the angle of incidence for all five considered particles. One can notice that the pitching torque coefficient is equal

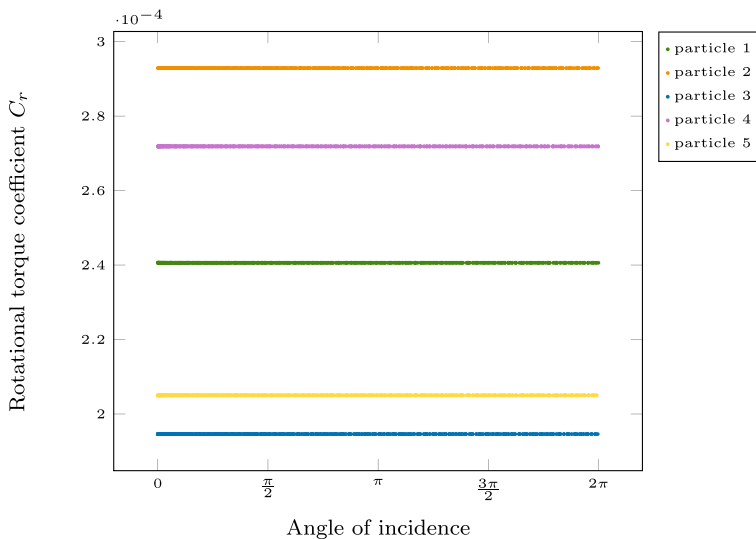


Fig. 11 Rotational torque coefficient as a function of angle of incidence for the five particles defined in Table 2. The rotational torque does not depend on the orientation of the particles since it is triggered by the symmetric rotational flow around the particle

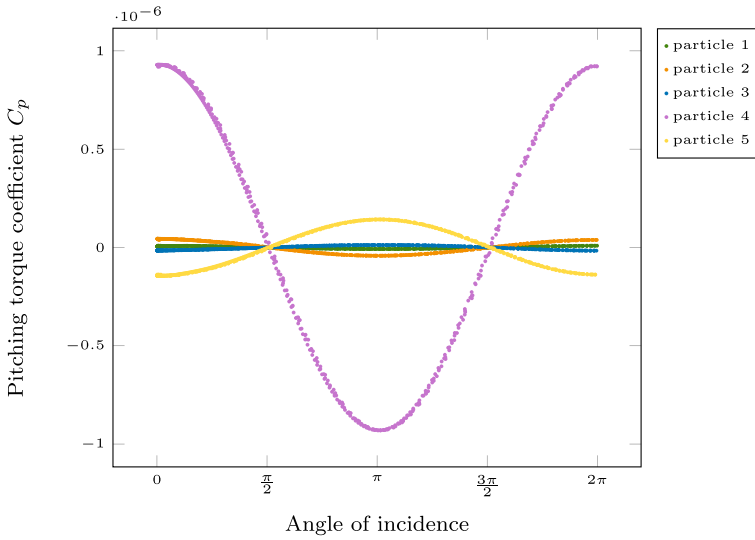


Fig. 12 Pitching torque coefficient as a function of angle of incidence for the five particles defined in Table 2

to 0 for $\psi = \frac{\pi}{2}$ and $\psi = \frac{3\pi}{2}$ for all five particles, so when particles are perpendicular to the flow. It means that for asymmetric particles, i.e., particles 4 and 5, the pitching torque is 0 when they are set symmetrically with respect to the flow. It may imply that this is their preferred orientation. In case of particle 4 the pitching torque coefficient reaches its minimum when the angle of incidence is $\psi = \pi$ and its maximum when $\psi = 0$ or $\psi = 2\pi$ are reached. It is caused by its asymmetric shape and setting with respect to the direction of the local fluid vorticity, namely particle 4 is convex “at the bottom” and concave “at the top” (for angle of incidence $\psi = 0$ or $\psi = 2\pi$), while the fluid around it is moving clockwise (see Fig. 13). For particle 5 the situation is analogous, however it is convex “at the bottom” and concave “at the top”. This is consistent with what happens for particle 4 and is reflected on the plot. For the remaining particles 1, 2, 3, the pitching torque is equal or close to 0, which results from their symmetry.

In the case of the pitching torque coefficient it is not straightforward to make a comparison between presented trends and those obtained in literature (e.g. [18, 26, 45]). Most simulations are performed for non-spherical but symmetric particles. Therefore, the discrepancy cannot be easily explained.

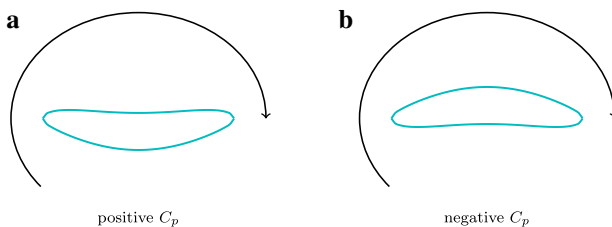


Fig. 13 The pitching torque coefficient C_p depends on particle settings

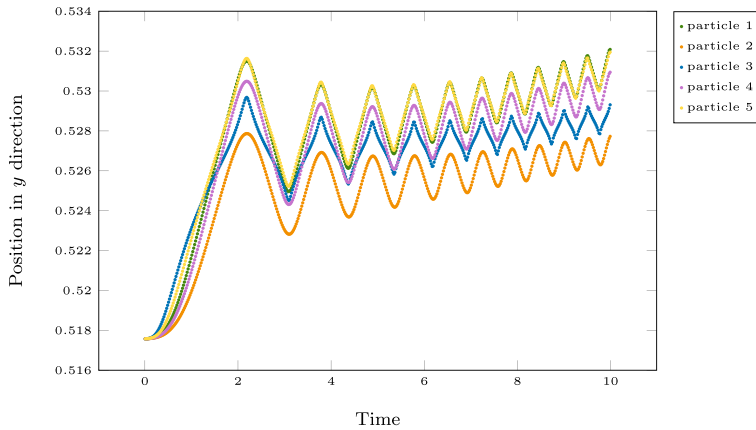


Fig. 14 Position in y direction for the five particles defined in Table 2

5.1.5 Oscillatory Translational Motion

The translational motion of non-spherical particles is characterized by an oscillatory motion. This is due to the fact that the pressure distribution causes the hydrodynamic forces to work at the center of pressure rather than at the center of mass. The non-coincidence of the center of pressure and center of mass causes the sustained oscillations (in y direction), see Fig. 14. The dominant factor of the movement in x direction is by the main flow profile. We did not observe oscillatory behaviour or variations of the horizontal velocity between the particles. Moreover, it is observed that every particle is also slowly moving up towards the horizontal axis of symmetry of the domain (all particles start below the axis, see Fig. 8). In Fig. 15 evolution in time of the aggregated lift of five studied particles is plotted together with the evolution in time of their lift coefficients. One can easily see that the oscillatory motion shown in Fig. 14 is a direct consequence of the lift force acting on the particles, while

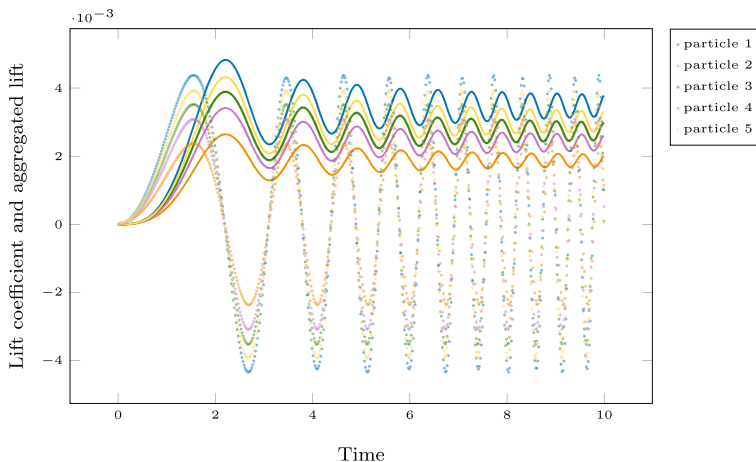


Fig. 15 Comparison of the lift coefficient for the different particles. In bold lines we show the aggregated lift over time

upward motion results from the aggregated lift being positive all the time. The behaviour of the y -velocity is also worth noting (see Fig. 16). One can notice that some particles (i.e., 1, 3, 5) decelerate when they are reaching local minimum or maximum. Those local maxima and minima appear for angle of incidence $\psi \in \{\frac{\pi}{4}, \frac{3\pi}{4}, \frac{5\pi}{4}, \frac{7\pi}{4}\}$, so when particles are inclined to the flow direction. Particle 3, the thinnest one, is subjected to the highest deceleration, whereas particles 2 and 4 move more smoothly.

5.2 Performance of the Coupled Model for Many Particles

In this section, we demonstrate the efficiency of the finite element/neural network approach and present numerical results with a multitude of particles. We test the computational effort for the particle model in comparison to the finite element Navier–Stokes discretization. Although some effort has been spent on the multicore implementation based on OpenMP, our implementation is by no means a high performance code. In particular, no use of GPU acceleration within the particle model is applied, neither in the coupling to the neural network model nor in the particle dynamics itself. Both is possible and in parts already standard in available software packages such like PyTorch C++ [28] or particle dynamics libraries such as LAMMPS [29].

All computations have been carried out on a two-socket system with Intel Xeon E5-2699A v4 processors running at 2.40 Ghz.

We will describe a prototypical blood-flow configuration and discuss the scaling of the implementation with respect to the number of cores. In particular we will investigate the relation between computational effort used in the particle model and in the Navier–Stokes solver. As in Section 5.1 all the parameters have been chosen so as to resemble vessel, blood and platelet properties.

5.2.1 Parallelization

The finite element model is implemented in Gascoigne 3D and outlined in Section 3.1, the discrete systems are approximated with a Newton–Krylow solver using geometric multi-grid preconditioning. Basic finite element routines and the linear algebra workflow is

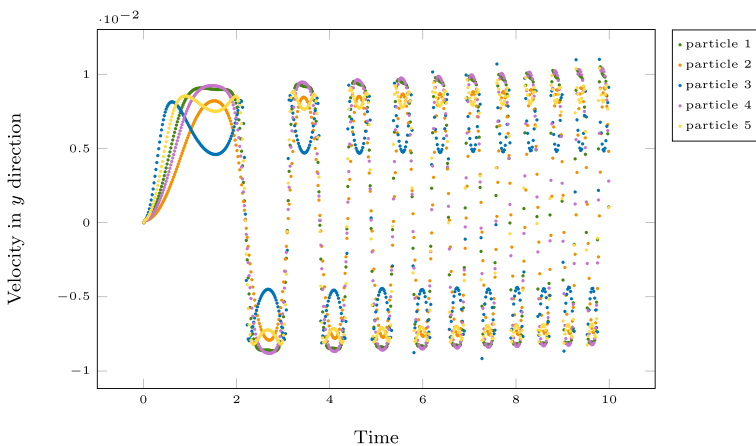


Fig. 16 Velocity in y direction for the five particles defined in Table 2

partially parallelized using OpenMP, see [10] for details. Since the mesh handling and i/o are not parallelized, substantial speedups are only reached for complex three dimensionals problems.

The particle model is based on a regular lattice mesh that covers the computational domain. The lattice elements of size $L_h \times L_h$ contain the individual particles. Detection of particle-particle collision is limited to those particles that reside in the same lattice element or that belong to directly adjacent elements. This substantially helps to reduce the computational effort which scales quadratically with the number of particles within each lattice element. Hence, we keep $L_h > 0$ small, such that an average of less than 100 particles resides in each element. On the other hand, L_h must be chosen large enough to avoid motion of particles across multiple elements in one time step, i.e. $k_{pd} V_p \leq L_h$, where $k_{pd} > 0$ is the time step size of the particle model and V_p the maximum velocity of the particles. Further, the lattice mesh is basis for parallelization since we can guarantee that no interaction between lattice elements which are separated by a complete layer can take place.

5.2.2 Configuration of the Test Case

We run simulations for the 2D flow in a channel with a local narrowing of 25% which should mimic a stenosed region of a blood channel. Figure 17 displays schematic geometry of the flow domain. The size of the domain, 2 mm \times 12 mm, is similar to the dimension of small arteries. The flow is driven by a Dirichlet profile on the inflow boundary Γ_{in} given by

$$v_{in}(x, y, t) = y \frac{2 - y}{2} \cdot v_{max}$$

with $v_{max} = 5$ mm/s. On the wall boundary Γ_{wall} we prescribe no-slip boundary conditions $\mathbf{v} = 0$ and on the outflow boundary Γ_{out} we use the do-nothing outflow condition $\mu \partial_n \mathbf{v} - p \mathbf{n} = 0$, see [14].

The fluid viscosity is set to $\mu = 3$ mg/mm \cdot s. Particle and fluid densities are equal $\rho = \rho_p = 1.06$ mg/mm³. Due to the fact that platelets constitute less than 1% of the blood volume [20] and the size of the domain we perform simulations with 165 000 particles.

In all numerical examples the temporal step size for solving Navier–Stokes equations is $k_{ns} = 0.005$ s, while time sub-step for particle advection is $k_{pd} = 0.00025$ s such that 20 subcycles are computed in each Navier–Stokes step. We update the force coefficients by evaluating the neural network every 10th step (i.e., twice in each Navier–Stokes step). The spatial finite element discretization is based on quadratic elements, with a total of 12819 degrees of freedom.

At the first time step we randomly seed about 165 000 particles distributed over the complete computational domain. Each particle is generated with random properties, i.e., specifying $P = (L_x, L_y, L_z, \alpha_{top}, \alpha_{bot})$ by means of the limits indicated in (2) such that the

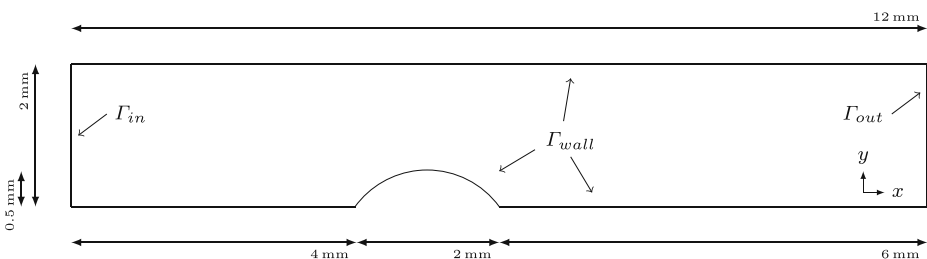


Fig. 17 Geometry of the numerical examples

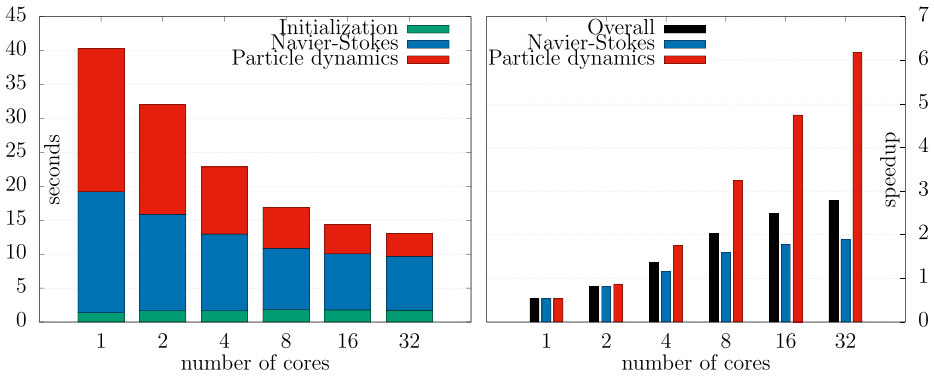


Fig. 18 Left: runtime (in seconds) for the coupled Navier–Stokes particle dynamics simulation for an increasing number of cores. Right: parallel speedup for the complete simulation and for the Navier–Stokes finite element simulation and the particle dynamics simulation separately

full variety of dimensions and shape is present. Details on the procedure for parametrization of the particles are given in Section 4.1. Then, 10 iterations of the interaction loop shown in Algorithm 2 are performed. Hence, 10 time steps of the Navier–Stokes problem and 200 particle dynamics substeps are performed. Figure 18 shows the runtime for all 200 iterations. Furthermore we indicate the parallel speedup. These results show that the allocation of computational time to the Navier–Stokes finite element solver and the particle dynamics system is rather balanced. While it is non-trivial to get a reasonable parallel speedup for highly efficient multigrid based finite element simulations (at least for simple 2d problems like this Navier–Stokes testcase), the scaling of the particle dynamics system is superior. These results demonstrate that the number of particles is not the limiting factor for such coupled simulations.

The key feature of our coupled Navier–Stokes particle dynamics scheme is the prediction of the hemodynamical coefficients by means of the previously trained neural network instead of using analytical models, which are not available, or running resolved simulations, which is not feasible for such a large number of particles. In Fig. 19 we give details on the

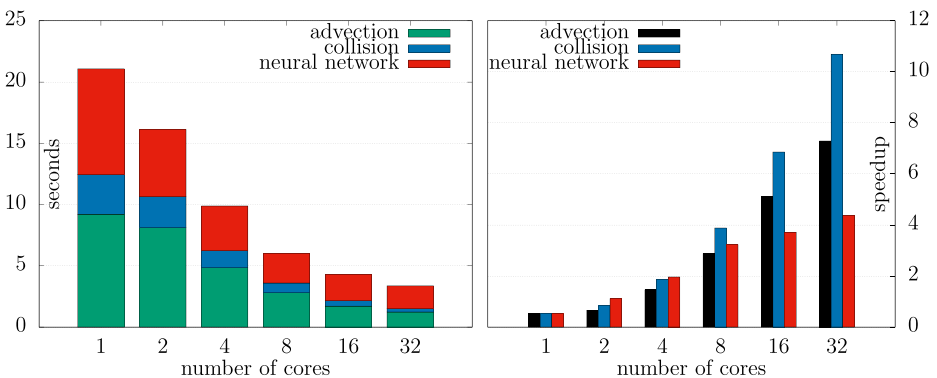


Fig. 19 Left: runtime (in seconds) for the particle dynamics simulation (all 200 substeps). Right: parallel speedup for the particle advection, handling of particle collisions and the neural network access for predicting the hemodynamical coefficients

computational time spend in the different parts of the particle dynamics system. Besides advection of the particles, the evaluation of the neural network is dominant, although we update the coefficients in every 10th step only. Here, a more systematic study of the impact of the update frequency should be performed. A further acceleration of the neural network evaluation is possible by using the GPU implementation of PyTorch.

The transition to more realistic three dimensional problems will substantially increase the effort in both parts, finite elements and particle dynamics. For the Navier–Stokes simulation it has been demonstrated that realistic 3d blood flow situations can be handled in reasonable time, see [8–10]. In case of substantially increased number of particles, the neural network coupling for estimating hemodynamic coefficients should be realized in a high performance package such as LAMMPS [29] that allows for an efficient GPU implementation.

6 Conclusions

Suspensions of arbitrarily-shaped particles in a fluid are of great importance both in engineering and medical applications. However, the interaction of the non-spherical particles with a fluid flow is a complex phenomenon, even for regularly-shaped particles in the simple fluid flows. The main difficulty lies in determining hydrodynamic forces acting on a particle due to their strong dependence on both particle shape and its orientation with respect to the fluid flow.

In this paper, a model is successfully derived to simulate the motion of non-spherical particles in a non-uniform flow field, including translation and rotation aspects. The model is designed to reflect platelets in a blood flow, both in terms of particle parameters and fluid configuration. The very good agreement of these results obtained by the coupled finite element/neural network/particle dynamics simulation with state of the art documentations in literature indicates an effectiveness of the presented approach and hence an encouraging potential toward medical applications. Furthermore, the big improvement over usual analytical interaction models is clearly seen as the neural network based model holds for a broad range of different shapes at any orientation. Moreover, using neural network to identify the transmission of forces from fluid to the particles provides a possibility to adopt the model to any desired shape of particle, making this method very promising.

We have further documented details on the scaling of the approach to many particles, which, in 2d blood flow simplifications, matches the typical particle density found for thrombocytes in blood flows. The computational effort is well balanced into the Navier–Stokes finite element part, the particle advection and the evaluation of the neural network.

The validation of the finite element/particle approach based on the neural network for identifying the coefficients by resolved simulations is still an open point. Due to the large range of spatial scales to be bridged (from the vessel to the particle scale) this will require the use of adaptive and dynamic meshes. Furthermore, since we consider non-spherical particles a Eulerian description based on, e.g., cut finite elements [47] will be required.

Acknowledgements The authors acknowledge the financial support by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - 314838170, GRK 2297 MathCoRe. TR acknowledges the support by the Federal Ministry of Education and Research of Germany, grant number 05M16NMA.

Funding Open Access funding enabled and organized by Projekt DEAL.

Declarations

Conflict of Interests The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Becker, R., Braack, M.: Multigrid techniques for finite elements on locally refined meshes. *Numer. Linear Algebra Appl.* **7**, 363–379 (2000)
2. Becker, R., Braack, M.: A finite element pressure gradient stabilization for the Stokes equations based on local projections. *Calcolo* **38**, 173–199 (2001)
3. Becker, R., Braack, M.: A two-level stabilization scheme for the navier-stokes equations. In: Feistauer, M., Dolejší, V., Knobloch, P., Najzar, K. (eds.) *Numerical Mathematics and Advanced Applications*, ENUMATH 2003, pp. 123–130. Springer, Berlin, Heidelberg (2004)
4. Becker, R., Braack, M., Meidner, D., Richter, T., Vexler, B.: The finite element toolkit Gascoigne. <https://www.gascoigne.de> (2005)
5. Braack, M., Burman, E., John, V., Lube, G.: Stabilized finite element methods for the generalized Oseen problem. *Comput. Methods Appl. Mech. Eng.* **196**, 853–866 (2007)
6. Chhabra, R.P., Agarwal, L., Sinha, N.K.: Drag on non-spherical particles: an evaluation of available methods. *Powder Technol.* **101**, 288–295 (1999)
7. Clift, R., Grace, J., Weber, M.: *Bubbles, Drops, and Particles*. Academic Press, New York (1978)
8. Failer, L., Minakowski, P., Richter, T.: On the impact of fluid structure interaction in blood flow simulations. *Vietnam J Math.* <https://doi.org/10.1007/s10013-020-00456-6> (2020)
9. Failer, L., Richter, T.: A Newton multigrid framework for optimal control of fluid–structure interactions. *Optim Eng.* <https://doi.org/10.1007/s11081-020-09498-8> (2020)
10. Failer, L., Richter, T.: A parallel Newton multigrid framework for monolithic fluid–structure interactions. *J. Sci. Comput.* **82**, 28 (2020)
11. Fasano, A., Sequeira, A.: Blood coagulation. In: *Hemomath*, pp. 79–158. Springer International Publishing, Cham (2017)
12. Filipovic, N., Kojic, M., Tsuda, A.: Modelling thrombosis using dissipative particle dynamics method. *Philos. Trans. R. Soc. Ser. A* **366**, 3265–3279 (2008)
13. Fogelson, A.L., Guy, R.D.: Immersed-boundary-type models of intravascular platelet aggregation. *Comput. Methods Appl. Mech. Eng.* **197**, 2087–2104 (2008)
14. Heywood, J., Rannacher, R.: Finite-element approximation of the nonstationary Navier–Stokes problem. Part IV: Error analysis for second-order time discretization. *SIAM J. Numer. Anal.* **27**, 353–384 (1990)
15. Heywood, J., Rannacher, R., Turek, S.: Artificial boundaries and flux and pressure conditions for the incompressible Navier–Stokes equations. *Int. J. Numer. Methods Fluids* **22**, 325–352 (1996)
16. Hartmann, D., Lessig, C., Margenberg, N., Richter, T.: A neural network multigrid solver for the Navier–Stokes equations. *arXiv:2008.11520* (2020)
17. Hölzer, A., Sommerfeld, M.: New simple correlation formula for the drag coefficient of non-spherical particles. *Powder Technol.* **184**, 361–365 (2008)
18. Hölzer, A., Sommerfeld, M.: Lattice Boltzmann simulations to determine drag, lift and torque acting on non-spherical particles. *Comput. Fluids* **38**, 572–589 (2009)

19. Kimmritz, M., Richter, T.: Parallel multigrid method for finite element simulations of complex flow problems on locally refined meshes. *Numer. Linear Algebra Appl.* **18**, 615–636 (2010)
20. Fogelson, A.L., Neeves, K.B.: Fluid mechanics of blood clot formation. *Ann. Rev. Fluid Mech.* **47**, 377–403 (2015)
21. Lu, G., Third, J., Müller, C.: Discrete element models for non-spherical particle systems: from theoretical developments to applications. *Chem. Eng. Sci.* **127**, 425–465 (2015)
22. Luding, S.: Collisions & contacts between two particles. In: Herrmann, H.J., Hovi, J.-P., Luding, S. (eds.) *Physics of Dry Granular Media*. NATO ASI Series, Vol. 350, pp. 285–304. Springer, Dordrecht (1998)
23. Mandø, M., Rosendahl, L.: On the motion of non-spherical particles at high Reynolds number. *Powder Technol.* **202**, 1–13 (2010)
24. Mody, N.A., King, M.R.: Three-dimensional simulations of a platelet-shaped spheroid near a wall in shear flow. *Phys. Fluids* **17**, 113302 (2005)
25. Muth, B., Müller, M.-K., Müller, M.K., Eberhard, P., Luding, S.: Collision detection and administration methods for many particles with different sizes. In: Cleary, P. (ed.) *Discrete Element Methods, DEM 07, 4th International Conference on Discrete Element Methods, DEM 4 - Brisbane, Australia*, pp. 1–18. Minerals Engineering Int. (2007)
26. Ouchene, R., Khalij, M., Arcen, B., Tanière, A.: A new set of correlations of drag, lift and torque coefficients for non-spherical particles and large Reynolds numbers. *Powder Technol.* **303**, 33–43 (2016)
27. Ouchene, R., Khalij, M., Tanière, A., Arcen, B.: Drag, lift and torque coefficients for ellipsoidal particles: From low to moderate particle Reynolds numbers. *Comput. Fluids* **113**, 53–64 (2015)
28. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché Buc, F., Fox, E., Garnett, R. (eds.) *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc (2019)
29. Plimpton, S.: Fast parallel algorithms for short-range molecular dynamics. *J. Comput. Phys.* **117**, 1–19 (1995)
30. Richter, T.: *Fluid-structure Interactions. Models, Analysis and Finite Elements*. Lecture Notes in Computational Science and Engineering, vol. 118. Springer, Cham, Berlin (2017)
31. Rosendahl, L.: Using a multi-parameter particle shape description to predict the motion of non-spherical particle shapes in swirling flow. *Appl. Math. Model.* **24**, 11–25 (2000)
32. Saad, Y.: *Iterative Methods for Sparse Linear Systems*. PWS Publishing Company (1996)
33. Sanjeevi, S., Kuipers, J., Padding, J.: Drag, lift and torque correlations for non-spherical particles from Stokes limit to high Reynolds numbers. *Int. J. Multiph. Flow* **106**, 325–337 (2018)
34. Shardt, O., Derksen, J.: Direct simulations of dense suspensions of non-spherical particles. *Int. J. Multiph. Flow* **47**, 25–36 (2012)
35. Sweet, C.R., Chatterjee, S., Xu, Z., Bisordi, K., Rosen, E.D., Alber, M.: Modelling platelet-blood flow interaction using the subcellular element Langevin method. *J. R. Soc. Interface* **8**, 1760–1771 (2011)
36. Tosenberger, A., Ataullakhanov, F., Bessonov, N., Pantelev, M., Tokarev, A., Volpert, V.: Modelling of thrombus growth in flow with a DPD-PDE method. *J. Theor. Bio.* **337**, 30–41 (2013)
37. Viallat, A., Abkarian, M.: *Dynamics of Blood Cell Suspensions in Microflows*. CRC Press, Boca Raton (2019)
38. Wadell, H.: The coefficient of resistance as a function of Reynolds number for solids of various shapes. *J. Franklin Inst.* **217**, 459–490 (1934)
39. Wiwanitkit, V.: Plateletcrit, mean platelet volume, platelet distribution width: Its expected values and correlation with parallel red blood cell parameters. *Clin. Appl. Thromb/Hemost.* **10**, 175–178 (2004)
40. Xu, Z., Chen, N., Kamocka, M., Rosen, E., Alber, M.: A multiscale model of thrombus development. *J. R. Soc. Interface* **5**, 705–722 (2008)
41. Xu, Z., Chen, N., Shadden, S.C., Marsden, J.E., Kamocka, M.M., Rosen, E.D., Alber, M.: Study of blood flow impact on growth of thrombi using a multiscale model. *Soft Matter* **5**, 769–779 (2009)
42. Xu, Z., Lioi, J., Mu, J., Kamocka, M.M., Liu, X., Chen, D.Z., Rosen, E.D., Alber, M.: A multiscale model of venous thrombus formation with surface-mediated control of blood coagulation cascade. *Biophys. J.* **98**, 1723–1732 (2010)
43. Yan, S., He, Y., Tang, T., Wang, T.: Drag coefficient prediction for non-spherical particles in dense gas–solid two-phase flow using artificial neural network. *Powder Technol.* **354**, 115–124 (2019)

44. Yow, H., Pitt, M., Salman, A.: Drag correlations for particles of regular shape. *Adv. Powder Technol.* **16**, 363–372 (2005)
45. Zastawny, M., Mallouppas, G., Zhao, F., van Wachem, B.: Derivation of drag and lift force and torque coefficients for non-spherical particles in flows. *Int. J. Multiph. Flow* **39**, 227–239 (2012)
46. Zhang, P., Gao, C., Zhang, N., Slepian, M., Deng, Y., Bluestein, D.: Multiscale particle-based modeling of flowing platelets in blood plasma using dissipative particle dynamics and coarse grained molecular dynamics. *Cel. Mol. Bioeng.* **7**, 552–574 (2014)
47. von Wahl, H., Richter, T., Frei, S., Hagemeyer, T.: Falling balls in a viscous fluid with contact: Comparing numerical simulations with experimental data. [arXiv:2011.08691](https://arxiv.org/abs/2011.08691) (2020)
48. Zhong, W., Yu, A., Liu, X., Tong, Z., Zhang, H.: DEM/CFD-DEM Modelling of non-spherical particulate systems: Theoretical developments and applications. *Powder Technol.* **302**, 108–152 (2016)
49. Zhu, H., Zhou, Z., Yang, R., Yu, A.: Discrete particle simulation of particulate systems: Theoretical developments. *Chem. Eng. Sci.* **62**, 3378–3396 (2007)
50. Zhu, H., Zhou, Z., Yang, R., Yu, A.: Discrete particle simulation of particulate systems: a review of major applications and findings. *Chem. Eng. Sci.* **63**, 5728–5770 (2008)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.