Journal of Computational Physics 471 (2022) 111622

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Fast asymptotic-numerical method for coarse mesh particle simulation in channels of arbitrary cross section

Samuel Christensen [a,*], Raymond Chu [b], Christopher Anderson [b], Marcus Roper [a,b]

[a] *Dept. of Computational Medicine, Box 951766, Life Sciences Bldg., Los Angeles, CA 90095-1766, United States of America*
[b] *Dept. of Mathematics, Box 951555, Mathematical Sciences Bldg., Los Angeles, CA 90095-1555, United States of America*

ARTICLE INFO

Article history:
Received 26 October 2021
Received in revised form 10 July 2022
Accepted 12 September 2022
Available online 19 September 2022

Keywords:
Fluid dynamics
Inertial microfluidics
Inertial migration
Particles
Asymptotic analysis

ABSTRACT

Particles traveling through inertial microfluidic devices migrate to focusing streamlines. We present a numerical method that calculates migration velocities of particles in inertial microfluidic channels of arbitrary cross section by representing particles by singularities. Refinements to asymptotic analysis are given that improve the regularity of the singularity approximation, making finite element approximations of flow and pressure fields more effective. Sample results demonstrate that the method is computationally efficient and able to capture bifurcations in particle focusing positions due to changes in channel shape and Reynolds number.

1. Introduction

The micron-scale channels of microfluidic devices allow for manipulation of nano-liter volumes of fluid and small particles, including cells. At high enough fluid velocities, corresponding to Reynolds numbers between 1 and 100, interactions between particles and flow cause particles to move across streamlines, a phenomenon known as *inertial focusing* [1]. Inertial focusing is finding increasingly wide use in biotechnology: differential focusing of differently-sized particles can be used to separate (large) cancer cells from (small) blood cells [2–4]. Meanwhile, additional hydrodynamic interactions between inertially-focused particles cause them to organize into regularly spaced trains, aiding with their encapsulation or analysis [5,6]. Numerical simulation of inertial microfluidic devices requires solving for the 3D motion of suspended in a fluid flow occurring in a 3D geometry that is changing due to the movement of the particles. Moreover, the phenomenon of interest, particle migration, and in particular particle focusing occur at non-negligible Reynolds number, so cannot be modeled by numerical methods that are designed for small Reynolds number particle-flows such as Stokesian dynamics [7] or boundary integral methods [8].

The time evolving geometry of migrating particles, and need to include nonlinear terms, favors numerical methods such as immersed boundary [9,10] or immersed interface [11] that embed moving boundaries within a fixed computational grid on which the Navier-Stokes equations are solved. Relative easy parallelization has boosted the popularity of Lattice-Boltzmann methods (LBM) for approximating the Navier-Stokes equations [12]. Moving boundary LBM-based simulations have been used to study the dependences of focusing upon channel aspect ratio [13], particle size and Reynolds number [14],

[*] Corresponding author.
E-mail address: schristen@ucla.edu (S. Christensen).

https://doi.org/10.1016/j.jcp.2022.111622
0021-9991/© 2022 Published by Elsevier Inc.

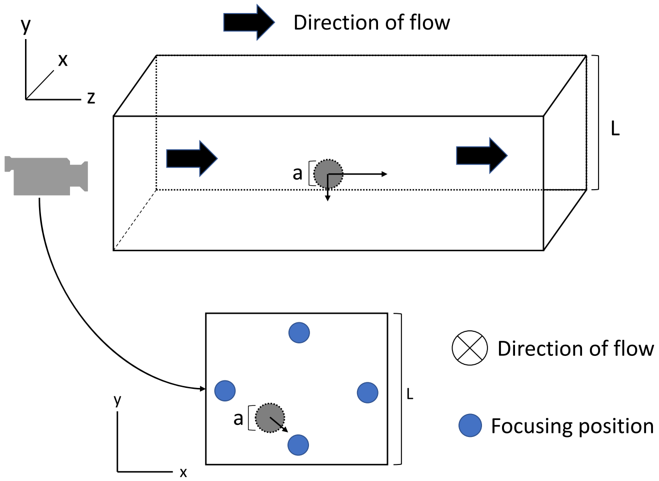

Fig. 1. Schematic showing sphere of radius a, freely flowing in a straight channel of uniform cross-section shape. Propelled by inertial migration forces, the particle converges to one of several focusing streamlines, shown in the cross-section view.

as well as to incorporate additional physics due to deformable particles [15], or interactions between multiple particles [16, 17]. Yet, although existing numerical simulations have illuminated the physics of inertial focusing, the high computational cost of nonlinear 3D simulations has meant that predictive simulations are not currently used to design or optimize inertial microfluidic devices.

Here we present a numerical method based on the singular asymptotic expansions associated with flows where the ratio of particle size to domain size is small [18]. When a suspended particle is small compared to the size of the domain, the fluid-filled domain may be divided into near- and far-field regions. Flow fields can be computed analytically within the near-field region, while in the far-field region the Navier-Stokes Equations are linearized and then solved numerically. By extending the asymptotic expansion of the near field solution to include higher order terms, we increase the regularity of the numerical far-field solution from discontinuous to continuous, which allows for second order convergence of a polynomial based finite element solution of the far field equations. Thus, the far-field solution is solved rapidly and accurately using polynomial finite elements. Although the method is generalizable to multiple particles and to fully three dimensional domains, we limit our attention here to predicting all particle focusing positions and focusing trajectories within channels of arbitrary cross-sections, a problem for which existing experimental and simulation data allows for testing of predictions. An implementation of the numerical method in Matlab may be downloaded from GitHub, and can be run on computers with as little as 8 GB of memory.

An existing method that also combines asymptotic and numerical techniques is the PHYSALIS resolved-particle simulation code [19]. The PHYSALIS simulation is based on a model that assumes in the rest frame of a rigid particle the velocity of the fluid close to the particle surface is close to zero. Leading order fluid acceleration terms can be absorbed into pressure, so in the near particle region, the fluid flow equations obey approximately the Stokes equations. The PHYSALIS code uses exact Stokes solutions to model fluid flow near particles and numerically couples the near field solutions to fully nonlinear fluid flow away from the particles. It can solve, in principle, for any particle shape where exact Stokes solutions are available, including for multiple spherical particles or 2D elliptical particles [19,20]. However the code still requires a fully resolved 3D Navier-Stokes simulation to solve for the flow away from particle neighborhoods. Rigorous analysis of the effects of where and how matching between Stokes and Navier-Stokes solutions remains an ongoing challenge. It is highly likely that the existing PHYSALIS code base could be used to model particle focusing in rectangular channels. The principal gains from the method described in this paper are that linear approximation in the outer region allows the outer field to be calculated using a single fast linear solve, and our use of finite element methods allows a diverse range of channel shapes to be modeled.

2. Derivation of approximate equations of motion

The problem considered in this paper is the prediction of particle trajectories in a straight microfluidic channel, with arbitrary cross-section. Due to inertial focusing forces, spherical particles introduced within the channel migrate across streamlines, arriving ultimately at one of a small number of inertial focusing locations (Fig. 1). We present a numerical method for predicting the focusing trajectory of particles, as well as the number of possible final focusing position and their basins of attraction.

Two Reynolds numbers can be defined for the flow: the channel Reynolds number $Re_c = \frac{\rho U_{\max} L}{\mu}$, constructed from the fluid viscosity, μ, density ρ, maximum velocity $U_{\max}$ and a channel length-scale (e.g. diameter), L. In typical microfluidic devices Re_c values are in the tens or hundreds. The particle Reynolds number $Re_p = \alpha^2 Re_c$, describes the velocity disturbances created by the particle, and involves an additional factor $\alpha \equiv \frac{a}{L}$ which is often much less than 1, so that Re_p is $O(1)$ or smaller.

We decompose fluid velocities within the microfluidic channel into two components: the flow without the particle (also called the background flow), $\bar{\mathbf{u}}$, and the disturbance velocity created by the particle: $\mathbf{u}'$. The total fluid velocity is $\mathbf{u} = \bar{\mathbf{u}} + \mathbf{u}'$. Furthermore, we introduce dimensionless variables $\mathbf{u}^* = \frac{\mathbf{u}' L}{a U_{max}}$, $\mathbf{x}^* = \frac{\mathbf{x}}{a}$, $t^* = \frac{t U_{max}}{L}$, $p^* = \frac{Lp}{\mu U_{max}}$, and we change the frame of reference so that $\mathbf{x}_p$, the center of our moving particle is now the origin. The equations for the dimensionless disturbance velocity are:

$$\begin{aligned} \Delta \mathbf{u}^* - \nabla p^* &= Re_p \left(\frac{\partial \mathbf{u}^*}{\partial t^*} + \left(\mathbf{U}^*_{\mathbf{p}}(t^*) + \bar{\mathbf{u}}(\mathbf{x}_p) + \mathbf{u}^* \right) \cdot \nabla \mathbf{u}^* + \mathbf{u}^* \cdot \nabla \bar{\mathbf{u}}' + \bar{\mathbf{u}}' \cdot \nabla \mathbf{u}^* \right) \\ \nabla \cdot \mathbf{u}^* &= 0 \\ \mathbf{u}^* &= \mathbf{0} \qquad \text{on the channel walls} \\ \mathbf{u}^* &= \mathbf{U}^*_p(t^*) + \mathbf{\Omega}_p(t^*) \times \mathbf{x} \qquad \text{on } |\mathbf{x}| = 1 \end{aligned} \tag{1}$$

where $\bar{\mathbf{u}}'(\mathbf{x}) = \bar{\mathbf{u}}(\mathbf{x}) - \bar{\mathbf{u}}(\mathbf{x}_p)$ is the background flow in frame of reference with the particle. The particle velocity and rotation, $\mathbf{U}^*_p$ and $\mathbf{\Omega}^*_p$ evolve over time to satisfy the force and torque free condition. The x and y components of $\mathbf{U}^*_p$ constitute the particle's migration velocity. Henceforth we drop the asterisks distinguishing dimensionless from dimensional quantities.

The asymptotic approximations incorporated into our numerical method are derived from the system of equations (1) in the limit $\alpha \to 0$ and $Re_p \to 0$. In these limits, flow field separates into an **inner region**, where the effects of the particle dominate but where inertial terms can be neglected at leading order, and we can expand the velocity field: $\mathbf{u} \sim \alpha Re_p^0 \mathbf{u}_1^{(0)} + \alpha Re_p \mathbf{u}_1^{(1)} + \ldots$ and an **outer region** where the inertia of the outer region interacts with the disturbance coming from the inner region to create a disturbance velocity: $\mathbf{u} \sim \alpha Re_p \mathbf{U}_1^{(1)} + \ldots$ [21–23,18]. We similarly expand the particle's translational velocity as $\mathbf{U}_p = \alpha Re_p^0 \mathbf{U}_{p,0}^{(0)} + \alpha Re_p^1 \mathbf{U}_{p,1}^{(1)} \ldots$ and the rotation speed as $\mathbf{\Omega}_p = \alpha Re_p^0 \mathbf{\Omega}_{p,1}^{(0)} + \alpha Re_p^1 \mathbf{\Omega}_{p,1}^{(1)} + \ldots$.

2.1. Inner region expansion of the equations of motion

In the inner region we scale lengths by a, the particle radius, and extract the order $O(\alpha Re_p^0)$ terms in Eqn. (1) to describe the disturbance velocity near the particle [18]:

$$\begin{aligned} \Delta \mathbf{u}_1^{(0)} - \nabla p_1^{(0)} &= \mathbf{0} \\ \mathbf{u} &= -\boldsymbol{\gamma} \cdot \mathbf{x}\mathbf{e}_3 + \mathbf{\Omega}_{p,1}^{(0)} \times \mathbf{x} \quad \text{on } |\mathbf{x}| = 1 \\ \mathbf{u} &\to \mathbf{0} \;\; \text{as } |\mathbf{x}| \to \infty. \end{aligned} \tag{2}$$

Where $\boldsymbol{\gamma}$ describes the local velocity gradient, also known as the shear rate, of the background flow at the location of the particle. The rotation component is there to satisfy the torque free condition. This equation can be solved analytically to obtain $\mathbf{u}_1^{(0)} \sim \mathbf{u}_{\text{str}} + O\left(1/r^4\right)$ as $r \equiv |\mathbf{x}| \to \infty$ [24]; the *stresslet* flow field, defined by

$$\mathbf{u}_{\text{str}} = -\frac{1}{r^2}\left(\gamma_x \frac{5xz\mathbf{x}}{2r^3} + \gamma_y \frac{5yz\mathbf{x}}{2r^3} \right), \tag{3}$$

represents the slowest decaying component of the disturbance velocity far away from the particle. Importantly, $\mathbf{u}_1^{(0)}$ depends only on the shear rate $\boldsymbol{\gamma}$ and there is no migration velocity generated at this order [25].

2.2. Outer region expansion of the equations of motion

The presence of the particle disturbs the background flow locally and an investigation of how this disturbance interacts with the background flow throughout the channel leads to the construction of equations for fluid motion in the outer region. Far from the particle, the rapidly decaying viscous stress term ($\Delta \mathbf{u}_{\text{str}} = O\left(\frac{1}{r^4}\right)$) will decrease until it reaches equal magnitude with the more slowly decaying inertial term ($Re_p \left(\bar{\mathbf{u}}' \cdot \nabla \mathbf{u}_{\text{str}}\right) = O\left(\frac{Re_p}{r^2}\right)$), at a distance $r = O(Re_p^{-1/2})$. Accordingly we define coordinates $\mathbf{X} = Re_p^{1/2}\mathbf{x}$. Generalizing the result of Schönberg and Hinch [18] for the $O(\alpha Re_p)$ leading order disturbance velocity to 2D channel shapes, we obtain:

$$\Delta \mathbf{U}_1^{(1)} - \nabla P_1^{(1)} - \mathbf{U}_1^{(1)} \cdot \nabla \bar{\mathbf{U}}' - \bar{\mathbf{U}}' \cdot \nabla \mathbf{U}_1^{(1)} = -\begin{bmatrix} \gamma_x \frac{\partial}{\partial Z} \\ \gamma_y \frac{\partial}{\partial Z} \\ \gamma_x \frac{\partial}{\partial X} + \gamma_y \frac{\partial}{\partial Y} \end{bmatrix} \delta(\mathbf{X}) \tag{4}$$

$$\nabla \cdot \mathbf{U}_1^{(1)} = \mathbf{0}$$

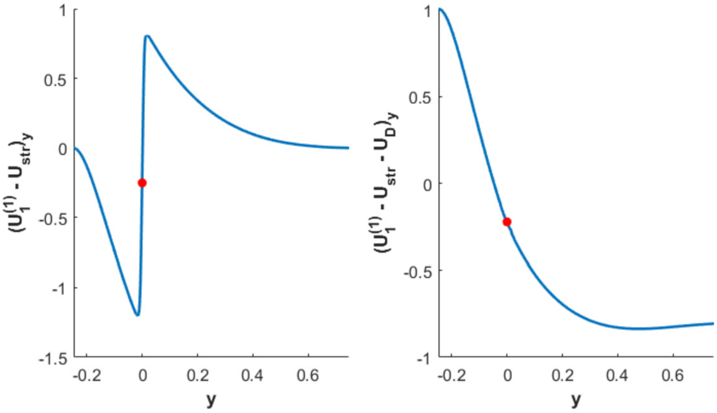

Fig. 2. The velocity $\mathbf{U}_1^{(1)} - \mathbf{U}_{\text{str}}$ is discontinuous, even once the stresslet contribution is subtracted. Left panel: the Y-component of the partially regularized velocity, $(\mathbf{U}_1^{(1)} - \mathbf{U}_{\text{str}})_Y$ along the line segment $X = Z = 0$ is discontinuous at $Y = 0$. Right panel: We accelerate particle focusing simulations by solving for the continuous velocity field $\mathbf{V} = \mathbf{U}_1^{(1)} - \mathbf{U}_{\text{str}} - \mathbf{U}_D$ using Eqn. (7). Velocity fields agree at $Y = 0$ (red dot), to give the particle migration velocity. Channel geometry is 4×1×10 rectangular channel with $Re_c = 1$. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

$$\mathbf{U}_1^{(1)} = \mathbf{0} \quad \text{on channel walls}$$

Matching of the outer region flows to the slowest decaying component of the inner region solution requires that $\mathbf{U}_1^{(1)} \sim \mathbf{u}_{\text{str}}$ as $\mathbf{X} \to \mathbf{0}$. In Eqn. (4), this matching is accomplished by adding derivatives of the Dirac delta function, $\delta(\mathbf{X})$, as forcing terms [18]. Once the stresslet has been subtracted, the limiting value of $\mathbf{U}_1^{(1)}$ as $\mathbf{X} \to 0$ can be used to compute the particle migration velocity at $O(Re_p)$. This PDE is much easier to solve numerically than the original system (Eqn. (1)) because it is linear and because there is no need to explicitly representing the moving particle boundary.

2.3. *Reformulating the problem for a continuous solution*

The solution of Eqn. (4) is singular at $\mathbf{X} = \mathbf{0}$. The singularity can be removed by subtracting off the stresslet velocity and pressure fields (Eqn. (3)) from $\mathbf{U}_1^{(1)}$ (see Fig. 2). However, the resulting non-singular solution remains discontinuous, and the polynomial finite element basis is a computationally inefficient representation for the solution's remaining three dimensional discontinuity, limiting the method to first order convergence. By removing the discontinuity we can construct a finite element solver that achieves second order convergence. We find the discontinuity analytically by examining $\mathbf{u}_1^{(1)}$, the inner solution at $O(\alpha Re_p)$, which satisfies:

$$\Delta \mathbf{u}_1^{(1)} - \nabla p_1^{(1)} = (\boldsymbol{\gamma} \cdot \mathbf{x}\mathbf{e_3}) \cdot \nabla \mathbf{u}_1^{(0)} + \mathbf{u}_1^{(0)} \cdot \nabla(\boldsymbol{\gamma} \cdot \mathbf{x}\mathbf{e_3}) \tag{5}$$

$$\nabla \cdot \mathbf{u}_1^{(1)} = 0\,, \tag{6}$$

with appropriate velocity boundary conditions to maintain a force and torque free particle. Howecwe, these boundary conditions do not affect the non-decaying component of the velocity solution. We are only interested in the non-decaying component of this solution which is not affected by the boundary condition. There is no migration or rotation generated from the inner region due to this solution [26,27], but components of $\mathbf{u}_1^{(1)}$ do not decay as $r \to \infty$ and therefore affect the leading order solution in the outer region. We solved for the nondecaying components of Eqn. (5) analytically using spherical harmonics and the projection method (see Appendix A).

In order to take advantage of *a priori* knowledge of the discontinuity, we decompose $\mathbf{U}_1^{(1)}$ and $P_1^{(1)}$ into regular and irregular (singular + discontinuous) components: $\mathbf{U}_1^{(1)} = \mathbf{V} + \mathbf{U}_{\text{str}} + \mathbf{U}_D$ and $P_1^{(1)} = Q + P_{\text{str}} + P_D$, where $(\mathbf{U}_{\text{str}}, P_{rmstr})$ is the stresslet solution from Eqn (3) and $(\mathbf{U}_D, P_D)$ are the discontinuous velocity fields given in Eqn. (A.9), rewritten in terms of the outer variable $\boldsymbol{X}$. Rewritten in terms of $(\mathbf{V}, Q)$, Eqn. (4) becomes:

$$\begin{aligned}
\Delta \mathbf{V} - \nabla Q - \mathbf{V} \cdot \nabla \bar{\mathbf{U}} + \bar{\mathbf{U}} \cdot \nabla \mathbf{V} = \mathbf{U}_{\text{str}} \cdot \nabla(\bar{\mathbf{U}} - \boldsymbol{\gamma} \cdot \mathbf{x}\mathbf{e}_3) + \mathbf{U}_{\text{str}} \cdot \nabla(\bar{\mathbf{U}} - \boldsymbol{\gamma} \cdot \mathbf{x}\mathbf{e}_3)& \\
+ \mathbf{U}_D \cdot \nabla \bar{\mathbf{U}} + \bar{\mathbf{U}} \cdot \nabla \mathbf{U}_D& \\
\nabla \cdot \mathbf{V} = \mathbf{0}& \\
\mathbf{V} = -\mathbf{U}_{\text{str}} - \mathbf{U}_D \quad \text{on walls of the channel}&
\end{aligned} \tag{7}$$

This reformulation increases numerical accuracy not just because the unknown velocity and pressure fields are made continuous at $\mathbf{X} = \mathbf{0}$, but because the order of the singularity in the right hand side of the first subequation in Eqn. (7) is reduced from $O(R^{-2})$ to $O(R^{-1})$.

2.4. Alternative regularization methods

Singular forcing terms are commonly encountered in computational models of fluid-body interactions, and there are existing methods that can be used to regularize a system of equations such as Eqn. (4). We compare the regularization from Section 2.3, with 1. a direct solve of Eqn. (4) with a regularized Dirac delta function and 2. velocity and pressure decomposition, in which a regularized form of the stresslet is subtracted from $\mathbf{U}_1^{(1)}$.

To regularize the Dirac delta forcing term in Eqn. (4) directly, we replace $\delta(\mathbf{X})$ by normalized Gaussian of ϵ-variance, $\delta^{\epsilon}(\mathbf{x}) = \frac{1}{(2\pi\epsilon)^{\frac{3}{2}}} \exp\left(-\frac{||\mathbf{x}||^2}{2\epsilon^2}\right)$:

$$\Delta \mathbf{U}_{\mathbf{1}}^{(\mathbf{1})} - \nabla P_1^{(1)} - \mathbf{U}_1^{(1)} \cdot \nabla \bar{\mathbf{U}}' - \bar{\mathbf{U}}' \cdot \nabla \mathbf{U}_1^{(1)} = -\frac{10\pi}{3} \begin{bmatrix} \gamma_x \frac{\partial}{\partial z} \\ \gamma_y \frac{\partial}{\partial z} \\ \gamma_x \frac{\partial}{\partial x} + \gamma_y \frac{\partial}{\partial y} \end{bmatrix} \delta^{\epsilon}(\mathbf{x}) \tag{8}$$

$$\nabla \cdot \mathbf{U}_1^{(1)} = \mathbf{0}$$

A second regularization method is modeled on the regularized Stokeslet method for blunting point force singularities in zero-Re flows [28]. Specifically, we isolate the singular part of the velocity field; $\mathbf{U}_0^{(1)} = \mathbf{U}_{\text{str}} + \mathbf{V}_{ns}$ where $\mathbf{U}_{\text{str}}$ is the stresslet written in terms of the outer coordinates. We present a regularized version of $\mathbf{U}_{\text{str}}$ that is spread over nearby grid points, $\mathbf{U}_{\text{str}}^{\epsilon}$, by solving Stokes's equations with delta function forcing by a blob with length scale ϵ [28]:

$$\Delta \mathbf{U}_{\text{str}}^{\epsilon} - \nabla P_{\text{str}}^{\epsilon} = -\begin{bmatrix} \gamma_x \frac{\partial}{\partial z} \\ \gamma_y \frac{\partial}{\partial z} \\ \gamma_x \frac{\partial}{\partial x} + \gamma_y \frac{\partial}{\partial y} \end{bmatrix} \frac{15\epsilon^4}{8\pi\left(r^2+\epsilon^2\right)^{\frac{7}{2}}}$$

$$\nabla \cdot \mathbf{U}_{\text{str}}^{\epsilon} = 0$$

from which we derive (see Appendix B):

$$\mathbf{U}_{\text{str}}^{\epsilon} = -\frac{5}{4(\epsilon^2+r^2)^{5/2}} \begin{bmatrix} 2xz(\gamma_x x + \gamma_y y) + \epsilon^2 \gamma_x z \\ 2yz(\gamma_x x + \gamma_y y) + \epsilon^2 \gamma_y z \\ 2z^2(\gamma_x x + \gamma_y y) + \epsilon^2(\gamma_x x + \gamma_y y) \end{bmatrix} \tag{9}$$

After regularizing the Dirac delta function, Eqn. (4) is recast in terms of $\mathbf{V}_{\text{ns}} = \mathbf{U}_0^{(1)} - \mathbf{U}_{\text{str}}^{\epsilon}$ and $P_{\text{ns}} = P_0^{(1)} - P_{\text{str}}^{\epsilon}$:

$$\Delta \mathbf{V}_{\text{ns}} - \nabla P_{\text{ns}} - \mathbf{V}_{\mathbf{ns}} \cdot \nabla \bar{\mathbf{U}}' - \bar{\mathbf{U}}' \cdot \nabla \mathbf{V}_{\mathbf{ns}} = \bar{\mathbf{U}}' \cdot \nabla \mathbf{U}_{\text{str}}^{\epsilon} + \mathbf{U}_{\text{str}}^{\epsilon} \cdot \nabla \bar{\mathbf{U}}' \tag{10}$$

$$\nabla \cdot \mathbf{V}_{\mathbf{ns}} = \mathbf{0}$$

$$\mathbf{V}_{\mathbf{ns}} = -\mathbf{U}_{\text{str}}^{\epsilon} \text{ on channel walls .}$$

3. Numerical method

Equations (7), (8), (10) are derived from different regularizations of Eqn. (4). To prevent the scaling of our computational domain from depending on particle Reynolds numbers, we introduce new dimensionless variables with lengths non-dimensionalized by the channel length scale: $\mathbf{x} = \mathbf{x}^{**} L$, pressures as: $p = \frac{\mu U_{\max}}{L} p^{**}$, and velocities as $\mathbf{V} = U_{\max} \mathbf{V}^{**}$. All three equations then take a common form:

$$Re_c\left(\bar{\mathbf{U}}'^{**} \cdot \nabla \mathbf{V}^{**} + \mathbf{V}^{**} \cdot \nabla \bar{\mathbf{U}}'^{**}\right) - \Delta \mathbf{V}^{**} + \nabla Q^{**} = \mathbf{G} \tag{11}$$

$$\nabla \cdot \mathbf{V}^{**} = 0$$

in which the body force, $\mathbf{G}$, depends on the type of regularization chosen and the field Q^{**} stands in for $-P_D$, $-P_0^{(1)}$, or $-P_{\text{ns}}$ respectively. In all three systems, $\mathbf{V}$ satisfies Dirichlet boundary conditions on the channel walls. Scaling lengths by L means that channel geometries no longer depend on particle size and speed, at the cost of reintroducing Re_c as a parameter in the PDE. In all formulations we must first solve for the background flow: $\bar{\mathbf{U}}^{**} \equiv (0, 0, \bar{U}^{**}(x, y))$. Because the flow is rectilinear, $\bar{U}^{**}$ satisfies a 2D Poisson equation, which can be solved one time, for all particle locations and values of Re_c.

Returning to Eqn. (11), we simplify our solution geometry by taking a Fourier transform $z \to k$, at the same time omitting the double asterisks that denote dimensionless variables:

$$\begin{aligned} Re_c(-ik\bar{U}'\hat{V}_x) - \frac{\partial^2\hat{V}_x}{\partial x^2} - \frac{\partial^2\hat{V}_x}{\partial y^2} + k^2\hat{V}_x + \frac{\partial\hat{Q}}{\partial x} &= \hat{G}_x \\ Re_c(-ik\bar{U}'\hat{V}_y) - \frac{\partial^2\hat{V}_y}{\partial x^2} - \frac{\partial^2\hat{V}_y}{\partial y^2} + k^2\hat{V}_y + \frac{\partial\hat{Q}}{\partial y} &= \hat{G}_y \\ Re_c(\frac{\partial\bar{U}'}{\partial x}\hat{V}_x + \frac{\partial\bar{U}'}{\partial y}\hat{V}_y) - \frac{\partial^2\hat{V}_z}{\partial x^2} - \frac{\partial^2\hat{V}_z}{\partial y^2} + k^2\hat{V}_z - ik\hat{Q} &= \hat{G}_z \\ \frac{\partial\hat{V}_x}{\partial x} + \frac{\partial\hat{V}_y}{\partial y} - ik\hat{V}_z &= 0 \end{aligned} \tag{12}$$

thereby reducing our equations to set of uncoupled 2D-PDEs for each of the Fourier modes. The 2D-PDEs for different wavenumbers can be solved in parallel.

We discretize our equations in x and y using the Taylor-Hood P2+P1 finite element method, in which the velocity is approximated using piecewise quadratic functions, ϕ_j, and the pressure is approximated using piecewise linear functions, ψ_j [29]:

$$\hat{V}_i = \sum_j U_{ij}\phi_j \qquad \hat{Q} = \sum_j P_j\psi_j \tag{13}$$

and then we integrate both sides of Eqn. (12) against test functions ϕ for the momentum equations and ψ for the divergence-free condition to obtain the discrete approximation of Eqn. (12):

$$\begin{bmatrix} \mathbf{F} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \mathbf{P} \end{bmatrix} = \begin{bmatrix} \mathbf{G} \\ \mathbf{0} \end{bmatrix} \tag{14}$$

where $\mathbf{F}$ is our discrete integral of the momentum equations and $\mathbf{B}$ is the discrete integral approximation of -1 times the divergence operator. Strictly, $\mathbf{C} = \mathbf{0}$. However, the resulting system has a non-trivial kernel containing all solutions with $\mathbf{V} = \mathbf{0}$ everywhere, and Q = constant. To remove this degeneracy, we define $\mathbf{C} = \mathbf{0}$ for $k \neq 0$ and for $k = 0$ define $\mathbf{C} = c\mathbf{1}$, with c chosen so as not to change the spectra of the matrix: $\lambda_{min}(\mathbf{F}) < c < \lambda_{max}(\mathbf{F})$. Each wavenumber, k, produces a different matrix on the left-hand side of Eqn. (14); however the matrix equations are decoupled and can be solved in parallel.

Once all coefficients $\mathbf{U}$ and $\mathbf{P}$ are solved for, we reconstitute $\mathbf{V}$ and evaluate it at the particle location to obtain the inertial focusing velocity. To identify focusing positions we need to find inertial focusing velocities for many candidate particle positions, which is equivalent to solving Eqn. (14) with different forcing vectors $\mathbf{G}$ on the right-hand side but the matrices on the left-hand side that differ only by a Galilean transformation based on the particle's velocity. This transformation is small enough that the same preconditioner is effective regardless of particle velocity.

3.1. Preconditioning

Eqn. (14) is a large and sparse system of equations which, with an appropriately chosen preconditioner, can be solved using an iterative preconditioned Krylov subspace method (GMRES) that is more efficient and has less of a memory footprint than a direct method of solution based upon matrix factorization techniques, e.g. LU or QR factorization. An ideal right preconditioner for our matrix is

$$\begin{bmatrix} \mathbf{F} & \mathbf{B}^T \\ \mathbf{0} & -\mathbf{S} \end{bmatrix}^{-1} \tag{15}$$

where $\mathbf{S} = \mathbf{C} - \mathbf{B}\mathbf{F}^{-1}\mathbf{B}^T$ is the Schur complement. We call this preconditioner "ideal" because it will converge in just two iterations for all values of $\mathbf{G}$. To reduce the cost of applying this preconditioner, we can factor it as:

$$\begin{bmatrix} \mathbf{F} & \mathbf{B}^T \\ \mathbf{0} & -\mathbf{S} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{F}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\mathbf{B}^T \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & -\mathbf{S}^{-1} \end{bmatrix}. \tag{16}$$

However the cost of applying this preconditioner is more expensive than solving Eqn. (14) by a direct method, since it entails finding $\mathbf{F}^{-1}$ and $\mathbf{S}^{-1}$. We therefore construct an approximation of Eqn. (15) from good approximations for $\mathbf{S}$ and $\mathbf{F}$ that are easy to invert. An approximation of $\mathbf{S}$ is based on the idea that if we replace our discrete operators, which do not commute, with continuous ones, which do, we can perform algebraic manipulations that will simplify the construction of the preconditioner [30]. We approximate $\mathbf{F} \approx \tilde{\mathbf{F}} = -\Delta$ and:

$$\begin{aligned} \mathbf{S} &= \mathbf{C} - \mathbf{B}\mathbf{F}^{-1}\mathbf{B}^T \\ &\approx \mathbf{C} - \nabla \cdot (\tilde{\mathbf{F}})^{-1}\nabla \\ &= \mathbf{C} - (-\Delta)^{-1}\Delta \qquad (17) \\ &= \mathbf{C} + \mathbf{I}. \qquad (18) \end{aligned}$$

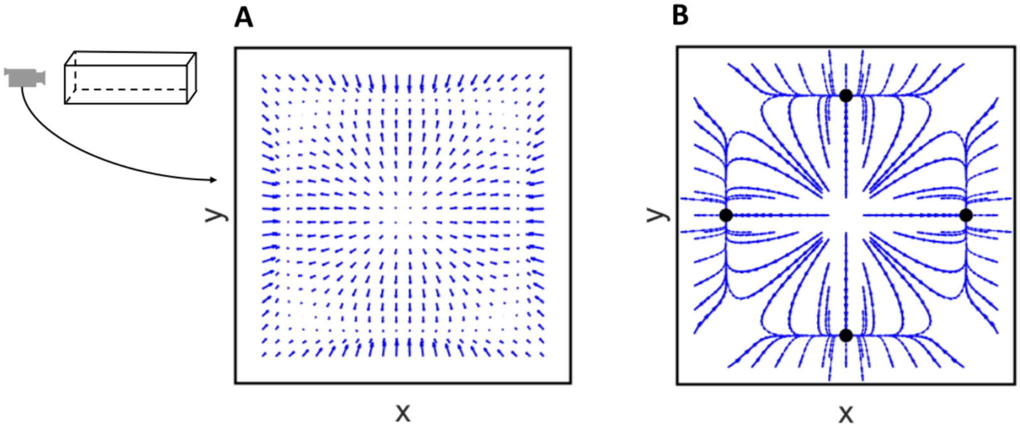

Fig. 3. Migration velocities are sampled at a grid of particle locations, and natural neighbor interpolation is used to evaluate velocities across the entire channel cross-section. All views show only the $x-$ and $y-$ components of the migration velocity. A: Migration velocities are sampled on a grid spaced by 0.03 in x and y. B: The sampled migration velocities are then interpolated to reconstruct particle trajectories by solving Eqn. (22) using ode45, and stable focusing positions are found as the limit points of trajectories (black dots).

Represented in the FE basis this operator becomes:

$$S_{ij} \approx \tilde{S}_{ij} = \mathbf{C} + \int_{\Omega} \psi_i \psi_j \, d\mathbf{x}. \tag{19}$$

Hence our approximate preconditioner is

$$\begin{bmatrix} \mathbf{F} & \mathbf{B}^T \\ \mathbf{0} & -\mathbf{S} \end{bmatrix} \approx \begin{bmatrix} \tilde{\mathbf{F}} & \mathbf{B}^T \\ \mathbf{0} & -\tilde{\mathbf{S}} \end{bmatrix} \tag{20}$$

where $\tilde{\mathbf{F}} = -\Delta$ and $\tilde{\mathbf{S}}$ is as defined in Eqn. (19). We factor the application of this preconditioner as

$$\begin{bmatrix} \tilde{\mathbf{F}} & \mathbf{B}^T \\ \mathbf{0} & -\tilde{\mathbf{S}} \end{bmatrix}^{-1} = \begin{bmatrix} \tilde{\mathbf{F}}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\mathbf{B}^T \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & -\tilde{\mathbf{S}}^{-1} \end{bmatrix}. \tag{21}$$

This makes the cost of preconditioning one application of $\tilde{\mathbf{F}}^{-1}$, one application of $\tilde{\mathbf{S}}^{-1}$, and one application of $\mathbf{B}^T$.

Calculating particle focusing positions for one channel involves tens of thousands of solves of Eqn. (14) for different wavenumbers and particle locations (Fig. 3). As each solve is independent, we parallelize over wavenumbers and assign each worker solving Eqn. (14) a single wavenumber but for all sampled particle locations. The worker constructs the preconditioner for this wavenumber along with its column- and row- pivoted LU factorization, and uses these to solve the matrix equation for every particle location. Column- and row- pivoting accelerates the LU factorization, and reduces the number of nonzero elements that need to be stored, thereby speeding up preconditioning. In our tests, GMRES converged with a relative residual of 10^{-4} in between 4 and 15 iterations for all values of k for $Re_c = 1$ and between 20 and 100 iterations for $Re_c = 100$.

3.2. Integration of the right hand side

$\mathbf{V}$ can be accurately approximated on a coarse finite element mesh because it is continuous, however the right hand side of Eqn. (14) diverges like $O(r^{-1})$ at the location of the particle, meaning that integrals of the right hand side are seldom accurate when approximated on the same coarse grid. To improve the accuracy, the integrals over the element containing the singularity and its D nearest neighboring elements were obtained by combining the integrals over a 4^n congruent sub-triangulation of the elements. The integrals over all triangles and sub-triangles were obtained using second order Gauss quadrature. Our experiments showed $D = 1$ and $n = 2$ were sufficient to accurately integrate the right-hand side and were used for all further calculations.

3.3. Calculating a particle trajectory

To calculate the trajectory of a particle across the channel cross-section, we must solve the ordinary differential equation:

$$\frac{d}{dt}(\mathbf{x}_p) = \mathbf{V}(\mathbf{x}_p; \mathbf{x}_p) \tag{22}$$

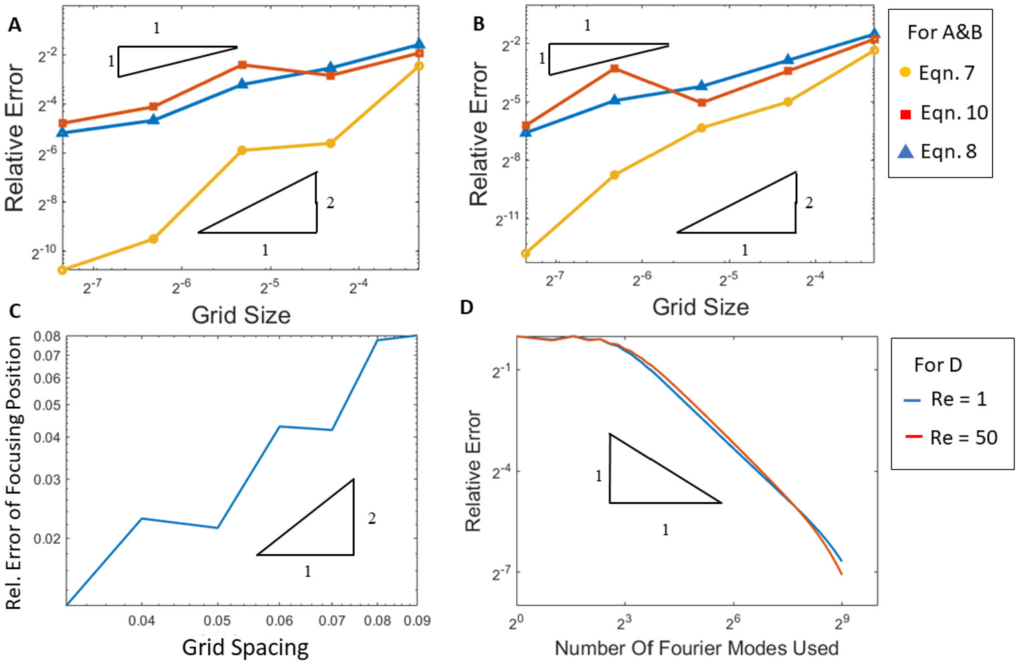

Fig. 4. Testing convergence of the inertial migration velocity with respect to mesh density at $R_c = 1,\ 50$ (A&B), sampling point density (C), and Fourier modes (D). A&B: Relative error of migration velocity at $R_c = 1$(A) and $R_c = 50$(B) of the 3 different methods of solution Eqn. (7) in yellow, Eqn. (8) in blue, and Eqn. (10) in red. C: Relative error of migration velocity versus number of Fourier modes included for $R_c = 1$ (blue) and $R_c = 50$ (red). D: Convergence of focusing positions (sec. 4.2) as we increase sampling point density with $R_c = 50$ and a channel cross section of scalene triangle with side lengths 1, 1.28, 1.02. A scalene triangle was chosen so that symmetry does affect convergence.

where $\mathbf{V}(\mathbf{x}_p; \mathbf{x}_p)$ denotes evaluating the solution of Eqn. (4) at $\mathbf{x}_p$ with the singularity located at position $\mathbf{x}_p$. Because the singularity must be at $\mathbf{x}_p$, the RHS of Eqn. (4) is different for every particle location, meaning one PDE solve is required for each time step.

Our goal is to calculate trajectories with initial conditions covering the channel, therefore it is more efficient to pre-calculate the migration velocity on a grid of sample positions (Fig. 3A) and interpolate between them to calculate $\mathbf{V}(\mathbf{x}_p; \mathbf{x}_p)$ in order to advance the particle trajectory. Once velocities are sampled across the entire channel cross-section, we advect the particles by solving Eqn (22) using the fourth order Runge-Kutta solver, ode45 in MATLAB (Mathworks, Natick, MA). We calculate the trajectories of many particles, identify stable focusing positions by considering the limiting behaviors of the particles (Fig. 3B).

In order to handle arbitrary channel shapes, we use natural neighbor interpolation [31] to interpolate velocities between arbitrarily spaced sampling points. This interpolation is once differentiable everywhere except the sampling points and yields quadratic convergence upon refinement.

4. Results

4.1. Numerical convergence

In this section we demonstrate the accuracy and efficiency of the migration velocity calculation and validate its second order convergence. To test our regularization method from Eqn. (7), we perform a self-convergence test on the calculated migration velocity for a particle in a $1 \times 1 \times 10$ square channel, with one particle located at $\mathbf{x}_p = [-0.1, 0.2]$ and $Re_c = 1,\ 50$. We also compare results and convergence rates the regularization method of Eqn. (7) with the existing regularization methods given by Eqns. (8) and (10) (Fig. 4A&B).

Mesh convergence for different regularization methods. We measure the rate of convergence for our migration velocity calculation as we decrease the 2D mesh size in a 0.1x0.1 box around the particle. Convergence is measured by comparing to computations done with half the grid size of the smallest grid size plotted. Relative error decreases proportional to $O(h^2)$ as the local grid-size around the discontinuity/singularity is reduced (Fig. 4A&B), which is the theoretical optimal rate of convergence for the $P2 + P1$ scheme.

We compare our new regularization method with two existing techniques, which were described in section 2.4. Both existing methods regularize the particle-singularity by blunting either the Dirac delta function (Eqn. (8)) or the stresslet velocity (Eqn. (10)), and introduce a blunting length-scale, ϵ. Our goal is to choose this length scale so as not to lose the physics induced by the singularity, while still representing it with our finite element basis. Empirically, we find minimum

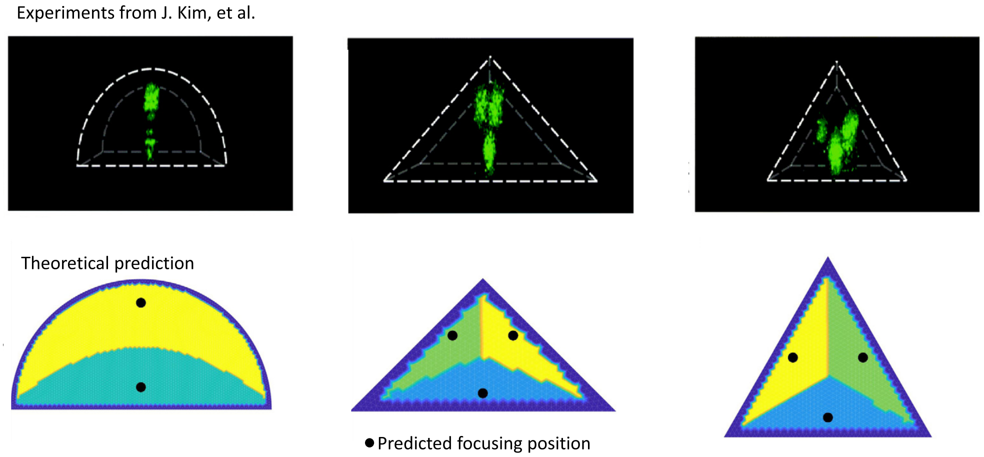

Fig. 5. Predicted focusing positions from our fast simulation method agree well with experimental data from [32]. From left to right: particle focusing positions in channels with semicircle, 2x1 isosceles triangle, and equilateral triangle cross section. Reynolds numbers were matched with those from experiment.

values for ϵ: $\epsilon = \text{gridsize}/2$ for Eqn. (8) and $\epsilon = \text{gridsize}/4$ for Eqn. (10). The existing methods had almost identical accuracy and $O(h)$ rates of convergence (Fig. 4A), and at the smallest mesh-sizes considered, produced errors that were around 16-fold larger than Eqn. (7). The slower convergence of the alternate methods is inevitable, since each makes an $O(\epsilon)$ approximation of the particle singularity. Indeed, even after the singularity is removed the velocity field remains discontinuous at the particle location. The remainder of the computational results presented utilize the method based upon Eqn. (7)

Convergence in number of velocity sampling points Rather than evaluating the migration velocity in Eqn. (22) directly, we pre-evaluate velocities on a grid of sampling points, and use interpolation to compute velocities at intermediate points. Our velocity interpolation method gives second order convergence as the spacing of sampling points is reduced [31] and the results from Fig. 4C shows this achieved for focusing position convergence. Fairly coarse sampling meshes can be used over much of the channel cross-section, however reconstructing the slow focusing manifold and the boundaries between the basins of attraction requires locally fine sampling meshes. Since the locations of these curves are not known *a priori*, we default to using a sampling mesh of 0.04.

Convergence in Fourier modes. The number of Fourier modes used to represent the solution is another parameter that influences the accuracy of the method. In Fig. 4D the convergence behavior is shown as an increasing number of Fourier modes are used to resolve the z-dependence of the particle disturbance flow. Errors in migration velocity decay like $O(N^{-1})$, where N is the number of Fourier modes. Convergence in Fourier modes is not spectral, likely because continuity of the velocity fields we are solving for fails at the level of the first derivative. Convergence is unaffected when changing R_c from 1 to 50.

4.2. Calculating focusing positions in channels of arbitrary cross section

Inertial migration causes particles to eventually collect at equilibrium streamlines, known as **focusing positions** (Fig. 3). We find stable focusing positions by integrating Eqn. (22) until particle velocities decrease below 10^{-5}. We identify all focusing positions by seeding the channel cross-section evenly with thousands of particles. We color code the seeding location according to which focusing position the particle converges to. This process partitions the cross section into basins of attraction (Fig. 5).

In a square channel, in common with previous studies [26], we find four symmetrically arranged focusing positions, one at each mid-face (Fig. 3B). The slow focusing manifold symmetrically links all four stable focusing positions. Although not shown explicitly, we infer that there must be four symmetrically arranged unstable focusing positions – that is streamlines that particles may remain on, but can not converge to – where the slow focusing manifold passes through the four diagonals of the square. Omitting these four unstable positions and the unstable center, the basins of attraction divide the channel into four triangular quadrants. We further validate predictions about focusing positions by comparing our computations with real experiments [32] using microfluidic channels with three different, non-rectangular cross-section shapes: two triangles of different base-height ratios and a semi-circle. We match Re_c to the parameters in [32]. Although no experimental data was presented on the trajectories of particles, we found close agreement between predicted and experimentally measured final focusing positions. In particular we see the same number of focusing positions (2 in a semicircular channel, and 3 in a triangular channel), as well as narrowing of the space between focusing positions in the larger base-height ratio triangle.

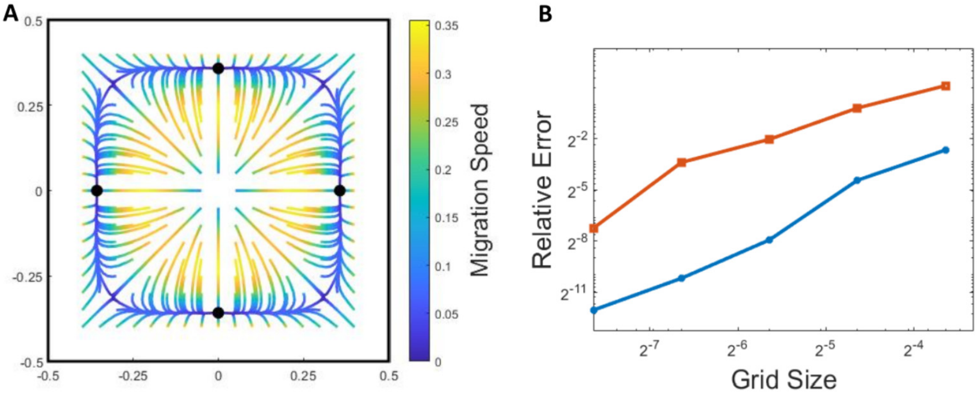

Fig. 6. Particles quickly migrate away from the center of the channel towards the slow focusing manifold, where they slowly migrate to the focusing positions. A: Particle trajectories are plotted with color denoting migration speed at a point. The black dots represent the focusing positions, the streamlines all particles will eventually gather upon. B: Computed migration velocities converge as $O(h^2)$ both on the slow manifold (red squares) and away from the slow manifold (blue circles). However, relative error is much larger on the slow manifold because velocities are ten-fold smaller. Tests were done in a square channel with $R_c = 1$ at $\mathbf{x}_p = (-0.2, -0.1)$ (far from the slow manifold) and $\mathbf{x}_p = (-0.1, -0.355)$ (close to the slow manifold).

4.3. Convergence along fast and slow manifolds

Inertial focusing occurs in two phases (Fig. 6) [33]: first particles move rapidly from their starting streamline toward a heteroclinic orbit within the channel cross-section (i.e. a manifold of streamlines when the stream-wise dimension is considered). Particles then move much more slowly along this manifold. Indeed, experimental evidence suggests that in real inertial microfluidic devices particles perform much of their fast focusing in the device inlet, leaving only the slow focusing parts of trajectories to be resolved within the channel [33]. Resolving the fast and slow dynamics is very important to understanding the speed at which particles migrate towards their focusing positions, as well as the size of the region of attraction.

Accurately calculating the migration velocity is more difficult along the slow manifold than elsewhere in the channel. This is because the magnitude of the error is similar but the migration velocity is ~ 10 fold smaller, thus the relative error is much higher (Fig. 6B). For example, we compare self convergence of the migration velocity in a square channel with $R_c = 1$, between a particle located at $(x_p, y_p) = (-0.2, -0.1)$, far from the slow focusing manifold, with a particle located at $(x_p, y_p) = (-0.1, -0.355)$, which lies on the slow focusing manifold. Although both velocities converge asymptotically as $O(h^2)$, relative errors remain 10-fold larger close to the slow focusing manifold.

As will be shown in Section 4.4, the location of the slow focusing manifold is sensitive to the physical parameters of the model; including channel shape and Reynolds number. The manifold shape also depends upon numerical parameters such as mesh size. Since whether a particle is placed on or off the slow manifold has a drastic effect upon the size of its inertial migration velocity, mesh size constraints are especially restrictive close to the slow manifold, since any perturbation of the manifold away from the particle will add a fast focusing velocity. For example, we compare self convergence of the migration velocity in a square channel with $R_c = 1$, between a particle located at $(x_p, y_p) = (-0.2, -0.1)$, far from the slow focusing manifold, with $(x_p, y_p) = (-0.1, -0.355)$, which lies on the slow focusing manifold. Although both velocities converge asymptotically as $O(h^2)$, relative errors remain 10-fold larger close to the slow focusing manifold. Although variation in relative errors across the cross-section means that we need to be cautious in evaluating the convergence of our algorithm, we found that a mesh-size of .02 could be used to reliably distinguish trajectories associated with different Reynolds numbers and channel shapes. However, resolving dynamics on the slow manifold, such as bifurcations affecting the number of stable focusing positions (Section 4.4) is particularly challenging numerically. Close to the critical shapes or Re_c-values that produce these bifurcations, focusing becomes extremely slow, potentially leading to the appearance of supernumerary stable focusing positions [14].

4.4. Varying channel shape and Reynolds number

Our algorithm is fast enough to test many channel geometries and particle Reynolds numbers, allowing for rapid exploration of the effect of channel shape and flow Reynolds number upon the number and location of focusing positions; an area in which there are still many unanswered questions.

As an example of the computational ability, we the effects of increasing Reynolds number in several geometries. Our calculations in square channels support previous studies that show that the slow focusing manifold and focusing positions move toward the channel walls as Re_c is increased (Fig. 7B). Di Carlo et al. [34] hypothesize that focusing positions arise from a balance of (shear) forces pushing the particle away from the channel center and (wall) forces that push particles away from channel walls, and that the shear forces increase more rapidly with Re_c. However, our modeling casts some

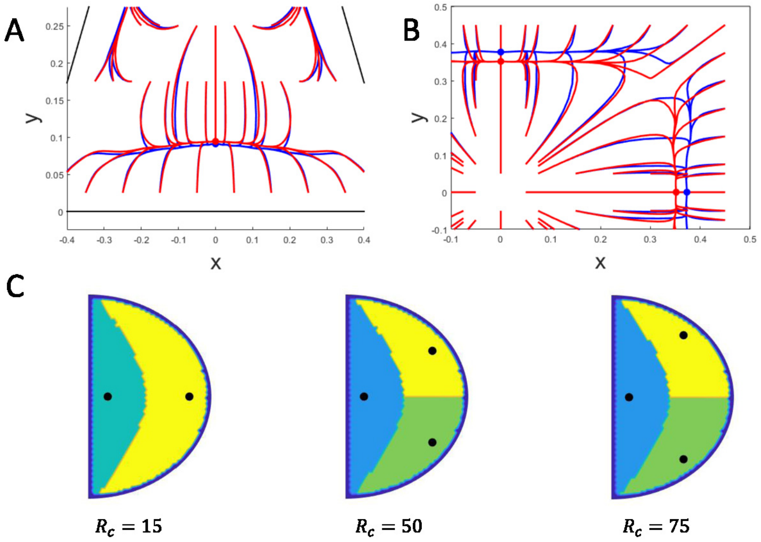

Fig. 7. Increasing channel Reynolds number changes the position and number of focusing streamlines. A-B, $Re_c = 1$ (red) and $Re_c = 100$ trajectories in an equilateral triangle and a square. In the square, increasing Re_c displaces focusing positions further from the center of the channel, but there is no effect in the triangle. C, Focusing positions bifurcate as Re_c is increased in a semi-elliptical channel with aspect ratio 1.2.

doubt on this mechanism, because we find no such shift in focusing positions as Re_c is increased for an equilateral triangle-shaped channel. In the equilateral triangle channel, focusing positions are symmetrically arranged near the middle of each channel wall, and maintain the same distance from the wall even as Re_c is increased 100-fold (Fig. 7A).

Emerging methods for microfabrication now allow microfluidic channels to be built with ever more complex shapes, including channels with curved walls [32]. Our computations show that the number of focusing positions undergoes a finite-Re_c bifurcation in some curved channels. Inspired by [32], we simulated a semi-elliptical channel with major:minor axis aspect ratio 1.2. Migration velocity sampling points were taken on a square mesh with density 0.03. We found that for this channel there is an apparent supercritical pitchfork bifurcation at $Re_c \approx 40$ in which a single stable fixed point in the mid-point of the channel curved face splits into two stable fixed points, separated by an unstable fixed point (Fig. 7C). The emergence of the fixed point by a pitchfork bifurcation stands in contrast to square channels, where, at $Re_c \approx 250$, new focusing positions emerge near the channel diagonals via *saddle point* bifurcations [35]. Close to the bifurcation, migration velocities along the slow manifold become extremely slow ('critical slowing down' [36]), which increases the challenge of resolving the precise number of stable fixed points by studying the long time convergence of particles. Moreover, our model for inertial migration is only exact in the limit as $\alpha \to 0$ and $Re_p \to 0$; we hypothesize that perturbations associated with modeling finite-size particles could create the imperfect bifurcations [36] seen in [35].

We also explore how number and location of stable focusing positions varies over the space of channel shapes. Because of the limitations of photo-lithographic methods, inertial microfluidic devices typically have rectangular channels. Experimental data seems to suggest that in contrast to square channels which have at least four stable focusing positions, channels with sufficiently high aspect (width to height) ratios, may have only two stable focusing positions, centered at the two larger edges of the channel [1,37,5]. We simulated inertial focusing in 4 rectangular channels whose aspect ratios ranged from 1 to 4. Migration velocities were sampled on a rectangular mesh of points with spacing chosen to be a fraction 0.04 of the channel size in each dimension, e.g. for 4×1 rectangle the spacing was 0.16 in x and 0.04 in y.

We observe for low aspect ratio channels and consistent with experiments [1], that channels have four stable focusing positions, each centered on one of the channel's four sides. However, even for the most 'slot'-like channel geometries, our simulations showed particles continuing to converge to any of four stable focusing positions even in the high aspect ratio channels (Fig. 8, left panels), consistent with [33]. We studied the basins of attraction of the short-side focusing positions. We found that, rather than disappearing, the basins of attraction as the aspect ratio of the channel is increased, the basins of attraction for these short-side focusing positions approach a limiting shape as channel aspect ratio increases (the basins of attraction are shown for $Re_c = 1$ in Fig. 8, right panels). While an isthmus between the two major basins of attraction must exist, it quickly becomes too slender to resolve in our simulations. Since our method leverages a very different set of assumptions about channel Reynolds numbers to those of [26], our simulations again support the conclusion reached by [33], that the disappearance of short-side focusing positions in experiments is an artifact of the prefocusing that occurs in the inlets that feed particles into real inertial microfluidic channels.

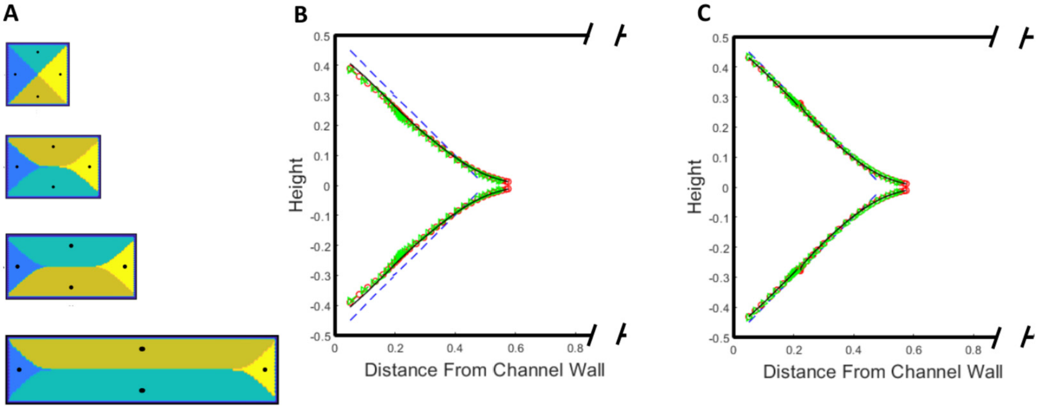

Fig. 8. A sweep over a range of parameters shows that short-side focusing positions persist even in high aspect ratio channels. The basins of attraction of these focusing positions approach a limiting shape as aspect ratio is increased. Left: Basins of attraction for channels of aspect ratio 1, 1.5, and 2. Right: Basins of attraction along the minor axis for rectangles of different aspect ratios. The blue dashed line represents AR=1, while the red circles, green triangles, and solid black line represent AR = 1.5, 2, and 4. For all simulations $Re_c = 1$. While the channels had different velocity sampling points (see main text), the basins of attraction created via interpolation were unaffected by sample spacing.

5. Conclusion

By representing particles submerged in fluids by singularities Schönberg and Hinch [18] were able to accurately calculate particle behavior without solving the nonlinear Navier-Stokes equations or explicitly meshing the 3D and time-varying fluid domain between particles and channel walls. We have taken the asymptotic approximations created in [18] and extended them to create a fast, efficient numerical method for calculating particle migration velocities in straight channels of uniform but arbitrary cross section. A proper accounting for singular and discontinuous terms that must be matched between inner and outer expansions allowed us to reduce the computation to one of finding only continuous components of the velocity field, improving the order of accuracy of the method and allowing for fast, stable calculations using a uniform mesh, including at the location of the particle itself. Numerical tests reveal that at moderate mesh sizes this method of regularizing the solution produces 10-fold higher accuracy than existing methods of blunting singularities, namely regularized stresslets and blob function approximations.

The method can be extended to particles of other shapes. Singularity modeling of the particle needs only the stresslet strength associated with the Stokes solution around the particle. This stresslet strength is already known for ellipsoids in shear flow [38] and for other particles can be found by solving for the motion of the particle in Stokes flow, for which there are many approximate or numerical methods [27,24]. However, a fully accelerated computation also requires that the discontinuous components of the velocity, $\mathbf{U}_D$, be found from the $O(\alpha R)$ inhomogeneous Stokes equation. For spherical particles we were able to isolate and find these terms analytically but for more complex particle geometries, this regularization could be given to a specialized solver designed to solve the inhomogeneous Stokes equations.

We note that subtracting the discontinuity leaves a velocity field that is continuous but that still has discontinuous derivatives. Higher order discontinuities can, in principle, be computed analytically by continuing the inner expansion to high order, that is, we can compute the discontinuities in the first derivative of $\mathbf{V}$ by continuing our expansion of the velocity field to $\mathbf{u}_2^{(1)}$, but numerical experiments in COMSOL Multiphysics (COMSOL Inc, Los Angeles, results not shown) showed that subtracting higher order discontinuities did not allow any reduction in the number of elements. Moreover, we found that the solver produces sufficiently accurate velocities at mesh sizes that are large enough to be applied uniformly across the entire of the channel cross-section, so discontinuity regularization alone is enough to achieve our goal of avoiding remeshing for different particle positions. However, the full potential of the Fourier basis used to represent the z-dependence of the solution remains unrealized.

Further expansions in terms of particle size and Reynolds number are possible, and would add either higher order forcing terms within our Oseen equation, or would require that we impose additional matching conditions to model the particle. Such an extension could be useful for predicting differential focusing of different particles within the same channel. Currently, although our simulations provide quick and accurate tools for predicting how particle focusing positions are affected by channel shape and channel Reynolds number, they can not reveal the effects of size upon focusing positions. In straight channels the effects of size are often mild [32], but particles of different sizes can follow very different focusing trajectories in non-straight channels, allowing particles of different sizes to be separated [39,40], and extending models to incorporate explicit dependence on particle size, offers a first step toward modeling the complex 3D geometries of a wide range of inertial microfluidic channels. Hood et al. [26] incorporated size-dependent focusing effects by calculating the next order term in α: $\mathbf{u}_2^{(0)}$. Although our model relies on different asymptotic limits, it could be similarly extended, though the calculation would require generating new solutions in both inner and outer regions, but still of the form Eqn (11), and thus we expect that the extension would not require any great increase of compute times.

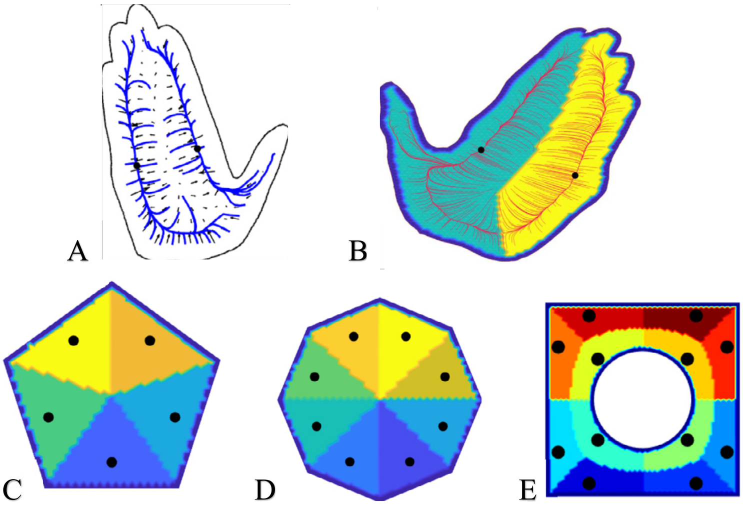

Fig. 9. Memory costs and time for individual velocity solves are quite small (∼200MB and ∼20 seconds on 1 core when averaged over simultaneous solves), allowing particle focusing velocities to be computed on a laptop computer. However, for complex channel geometries hundreds of particle positions are required for accurate interpolation of the migration velocity field (Fig. 3), and inference of focusing positions. Symmetry can be exploited to reduce the number of computations. Channel simulations in panels A-B took 200-250 particle locations and about 1 hour to be solved. However, simulations in panel D took only ∼15 minutes on a laptop computer when taking advantage of 2 reflection symmetries, so that only one quarter of migration velocities needed to be computed, while panel C took ∼30 minutes when taking advantage of 1 dimension of symmetry, so that only one half of migration velocities were computed.

We have implemented the algorithm described in this paper in Matlab (Mathworks, Natick, MA), and the resulting package (called *INFOCUS*) is freely downloadable from the first author's website. *INFOCUS* is intended to be usable by builders of inertial microfluidic devices, can be run on a laptop computer, and needs minimal programming experience. It is designed for building quick predictions of where and how fast particles will focus inside straight microfluidic channels with arbitrary cross-section shapes. Users need only need define channel shape by specifying vertices for polygonal-shaped channels or a level-set function for curved boundaries, and specify their channel Reynolds number based on the maximum flow velocity, or on the flow rate. *INFOCUS* can then calculate particle focusing positions, particle trajectories, and basins of attraction. The tool is capable of calculating the focusing positions of channels with simple geometries on a laptop computer with 8 GB of RAM in under an hour, while the most complex channel geometries that have assayed can be run on a desktop computer with 64 GB of RAM in 1-2 hours (Fig. 9). The implementation of our algorithm is able to handle arbitrary shapes and its speed makes parameter sweeps or shape optimization feasible.

The authors acknowledge financial support from the National Science Foundation through grants DMS-1351860 and DMS-2009317. We are grateful to Prof. Claudia Falcon, Izak Oltman, Siting Liu, Dominic Diaz, Alejandro Diaz, and Prof. Hangjie Ji, for scientific discussions.

CRediT authorship contribution statement

Samuel Christensen: Conceptualization, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Raymond Chu:** Conceptualization, Formal analysis, Methodology. **Christopher Anderson:** Supervision, Writing – review & editing. **Marcus Roper:** Conceptualization, Formal analysis, Investigation, Methodology, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Appendix A. Discontinuity solution

We wish to solve Eqn (5) analytically by transforming the problem into 3 separate inhomogeneous Laplace equations and solving those using Spherical Harmonics, denoted Y_i^j, which have the property that

$$\Delta\left(r^n Y_j^i(\theta,\phi)\right) = (n(n-1) - j(j+1))\, r^{n-2} Y_j^i(\theta,\phi) \tag{A.1}$$

making them pseudo-eigenvectors of the Laplacian [41]. All code is in Mathematica (Wolfram Research, Urbana Champaign, IL) and is based on code from [42]. We make the Helmholtz decomposition:

$$(\boldsymbol{\gamma}\cdot\mathbf{xe_3})\cdot\nabla\mathbf{u}_1^{(0)} + \mathbf{u}_1^{(0)}\cdot\nabla(\boldsymbol{\gamma}\cdot\mathbf{xe_3}) = \nabla\Phi + \nabla\times\mathbf{A} \tag{A.2}$$

Our goal is not to calculate $\mathbf{u}_1^{(1)}$ completely, but only to isolate the terms that are $O(r^0)$ as $r\to\infty$. Throughout our calculation, we will isolate only terms that contribute at this order.

We take the divergence of both sides of Eqn. (5), noting that since $\mathbf{u}_1^{(1)}$ is divergence-free:

$$\begin{aligned}\Delta p_1^{(1)} = \Delta\Phi &= \nabla\cdot\left((\boldsymbol{\gamma}\cdot\mathbf{xe_3})\cdot\nabla\mathbf{u}_1^{(0)} + \mathbf{u}_1^{(0)}\cdot\nabla(\boldsymbol{\gamma}\cdot\mathbf{xe_3})\right)\\ &= \nabla\cdot((\boldsymbol{\gamma}\cdot\mathbf{xe_3})\cdot\nabla\mathbf{u}_{\mathrm{str}} + \mathbf{u}_{\mathrm{str}}\cdot\nabla(\boldsymbol{\gamma}\cdot\mathbf{xe_3})) + o(r^{-3})\,.\end{aligned} \tag{A.3}$$

Using the fact that spherical harmonics make a perpendicular basis, we rewrite the first term of Eqn. (A.3) as a sum of spherical harmonics. $\sum_{j=0}^{\infty}\sum_{i=-j}^{j} a_{ij} r^{-3} Y_j^i$, where:

$$a_{ij} = \int_0^{\pi}\int_0^{2\pi} \nabla\cdot((\boldsymbol{\gamma}\cdot\mathbf{xe_3})\cdot\nabla\mathbf{u}_{\mathrm{str}} + \mathbf{u}_{\mathrm{str}}\cdot\nabla(\boldsymbol{\gamma}\cdot\mathbf{xe_3}))\, Y_j^i \sin\theta\, d\phi\, d\theta$$

We find a particular integral for this equation using the fact that spherical harmonics diagonalize the Laplace operator and lead to simple inversion

$$p_\Phi = \sum_{j=0}^{\infty}\sum_{i=-j}^{j} r^{-1} n_{ij}^{\Phi} Y_i^j \text{ where } n_{ij}^{\Phi} = \frac{a_{ij}}{2 - j(j+1)}\,. \tag{A.4}$$

Our first step toward calculating a leading order expression for $\mathbf{u}_1^{(1)}$ is to find a velocity field, $\mathbf{u}_\Phi$ associated with p_Φ, and satisfying the equation:

$$\Delta\mathbf{u}^{\Phi} = (\boldsymbol{\gamma}\cdot\mathbf{xe_3})\cdot\nabla\mathbf{u}_1^{(0)} + \mathbf{u}_1^{(0)}\cdot\nabla(\boldsymbol{\gamma}\cdot\mathbf{xe_3}) + \nabla p_\Phi \tag{A.5}$$

We decompose each Cartesian component of the right hand side of (A.5) into spherical harmonics: $\sum_{k=1}^{3}\sum_{j=1}^{\infty}\sum_{i=-j}^{j} b_{ijk} Y_j^i \mathbf{e}_k$ where:

$$b_{ijk} = \int_0^{\pi}\int_0^{2\pi}\left((\boldsymbol{\gamma}\cdot\mathbf{xe_3})\cdot\nabla\mathbf{u}_1^{(0)} + \mathbf{u}_1^{(0)}\cdot\nabla(\boldsymbol{\gamma}\cdot\mathbf{xe_3}) + \nabla p_\Phi\right)\cdot\mathbf{e}_k Y_j^i \sin\theta\, d\phi\, d\theta \tag{A.6}$$

We can then invert the Laplacian algebraically: $\mathbf{u}_\Phi = \sum_{k=1}^{3}\sum_{j=0}^{\infty}\sum_{i=-j}^{j}\alpha_{ijk}^{\Phi} Y_i^j \mathbf{e}_k$ where $\alpha_{ijk}^{\Phi} = -\frac{b_{ijk}}{j(j+1)}$.

$\mathbf{u}_\Phi$ is not the solution at O(1) in the inner region because it has non zero divergence and doesn't satisfy the boundary conditions, this makes sense as p_Φ is not the full pressure. To remedy this we simply add a potential to the pressure that would remove this divergence $p = p_\Phi + \nabla\cdot\mathbf{u}_\Phi$ and get a new RHS for a Laplace equation to represent with spherical harmonics

$$\begin{aligned}\Delta\mathbf{u}_1^{(1)} - \nabla(p_\Phi + \nabla\cdot\mathbf{u}_\phi) &= \left((\boldsymbol{\gamma}\cdot\mathbf{xe_3})\cdot\nabla\mathbf{u}_1^{(0)} + \mathbf{u}_1^{(0)}\cdot\nabla(\boldsymbol{\gamma}\cdot\mathbf{xe_3})\right)\\ \sum\sum\sum c_{ijk}\mathbf{e}_k &= \left((\boldsymbol{\gamma}\cdot\mathbf{xe_3})\cdot\nabla\mathbf{u}_1^{(0)} + \mathbf{u}_1^{(0)}\cdot\nabla(\boldsymbol{\gamma}\cdot\mathbf{xe_3}) + \nabla p_\Phi\right)\\ \text{where } c_{ijk} &= \int_0^{\pi}\int_0^{2\pi}\left((\boldsymbol{\gamma}\cdot\mathbf{xe_3})\cdot\nabla\mathbf{u}_1^{(0)} + \mathbf{u}_1^{(0)}\cdot\nabla(\boldsymbol{\gamma}\cdot\mathbf{xe_3}) + \nabla\left(p_\Phi + \nabla\cdot\mathbf{u}_\phi\right)\right)\cdot\mathbf{e}_k Y_i^j d\phi d\theta\end{aligned} \tag{A.7}$$

We can then perform one Laplace inverse for each of our 3 coordinate components:

$$\begin{aligned}\implies \left(\mathbf{u}_1^{(1)}\right)_k &= \sum\sum\alpha_{ijk} Y_i^j \mathbf{e}_k\\ \text{where } \alpha_{ijk} &= -\frac{c_{ijk}}{j(j+1)}\end{aligned} \tag{A.8}$$

Converting back to Cartesian coordinates we have our answer

$$\begin{aligned}
(\mathbf{u}_1^{(1)})_x &= \frac{-5\gamma_x^2 x\left(4x^4+x^2\left(7y^2+z^2\right)+3y^2\left(y^2+z^2\right)\right)-10\gamma_x\gamma_y y\left(3x^4+5x^2y^2+2y^2\left(y^2+z^2\right)\right)\dots}{72\left(x^2+y^2+z^2\right)^{5/2}}\\
&\quad\dots\frac{+5\gamma_y^2 x\left(x^4+x^2\left(y^2+z^2\right)+3y^2z^2\right)}{72\left(x^2+y^2+z^2\right)^{5/2}}\\
(\mathbf{u}_1^{(1)})_y &= \frac{5\gamma_x{}^2 y\left(x^2\left(y^2+3z^2\right)+y^2\left(y^2+z^2\right)\right)-10\gamma_x\gamma_y x\left(2x^4+x^2\left(5y^2+2z^2\right)+3y^4\right)\dots}{72\left(x^2+y^2+z^2\right)^{5/2}}\\
&\quad\dots\frac{-5\gamma_y{}^2 y\left(3x^4+z^2\left(3x^2+y^2\right)+7x^2y^2+4y^4\right)}{72\left(x^2+y^2+z^2\right)^{5/2}}\\
(\mathbf{u}_1^{(1)})_z &= \frac{5z\left(-6\gamma_x\gamma_y xy\left(x^2+y^2\right)+\gamma_x{}^2\left(5z^2\left(x^2+y^2\right)+3y^2\left(x^2+y^2\right)+2z^4\right)+\gamma_y{}^2\left(5z^2\left(x^2+y^2\right)+3x^2\left(x^2+y^2\right)+2z^4\right)\right)}{72\left(x^2+y^2+z^2\right)^{5/2}}
\end{aligned} \tag{A.9}$$

Appendix B. Regularizing the stresslet

Formally the stresslet solves a $Re = 0$ form of the Navier-Stokes equations:

$$\mu\nabla^2\mathbf{u}_{\text{str}} - \nabla p_{\text{str}} + \mathbf{f}_{\text{str}} = \mathbf{0}\ ,\ \ \nabla\cdot\mathbf{u}_{\text{str}} = 0\ , \tag{B.1}$$

for

$$\mathbf{f}_{\text{str}} = -\frac{10\pi\mu\gamma_x}{3}\left(\frac{\partial\delta(\mathbf{x})}{\partial z}, 0, \frac{\partial\delta(\mathbf{x})}{\partial x}\right)^T - \frac{10\pi\mu\gamma_y}{3}\left(0, \frac{\partial\delta(\mathbf{x})}{\partial z}, \frac{\partial\delta(\mathbf{x})}{\partial y}\right)^T\ , \tag{B.2}$$

where $\delta(\mathbf{x})$ is the Dirac delta function and (γ_x, γ_y) are the components of the shear vector. We follow [28] in deriving a non-singular approximation of $\mathbf{u}_{\text{str}}$ in which $\delta(\mathbf{x})$ is replaced by a function, ϕ_ϵ;

$$\phi_\epsilon(r) = \frac{15\epsilon^4}{8\pi(r^2+\epsilon^2)^{7/2}} \tag{B.3}$$

for which an exact solution of Eqn (B.1) is possible. For this choice of ϕ_ϵ, we may define auxiliary functions:

$$\begin{aligned}
\nabla^2 G &= \phi_\epsilon & \nabla^2 B &= G\\
H_1(r) &= r^{-1}B' - G & H_2(r) &= r^{-3}(rB'' - B')
\end{aligned} \tag{B.4}$$

where $(\cdot) = \frac{d(\cdot)}{dr}$, from which we can construct the components of the regularized stresslet:

$$u^\epsilon_{\text{str}\,i} = \frac{H_2'}{r}x_i x_j x_k Z_{jk} + \left(H_2 + \frac{H_1'}{r}\right)x_k Z_{ik}\ , \tag{B.5}$$

where

$$\mathbf{Z} = \frac{10a^3\pi}{3}\begin{pmatrix} 0 & 0 & \gamma_x \\ 0 & 0 & \gamma_y \\ \gamma_x & \gamma_y & 0 \end{pmatrix}. \tag{B.6}$$

We solve the system Eqn (B.4) using Maple (Maplesoft, Waterloo, Canada), obtaining:

$$\mathbf{u}^\epsilon_{\text{str}} = \frac{-5a^3}{4(\epsilon^2+r^2)^{5/2}}\begin{pmatrix} 2xz(\gamma_x x + \gamma_y y) + \epsilon^2\gamma_x z \\ 2yz(\gamma_x x + \gamma_y y) + \epsilon^2\gamma_y z \\ 2z^2(\gamma_x x + \gamma_y y) + \epsilon^2(\gamma_x x + \gamma_y y) \end{pmatrix}. \tag{B.7}$$

References

[1] D. Di Carlo, D. Irimia, R.G. Tompkins, M. Toner, Continuous inertial focusing, ordering, and separation of particles in microchannels, Proc. Natl. Acad. Sci. 104 (48) (2007) 18892–18897.

[2] J. Che, V. Yu, M. Dhar, C. Renier, M. Matsumoto, K. Heirich, E.B. Garon, J. Goldman, J. Rao, G.W. Sledge, et al., Classification of large circulating tumor cells isolated with ultra-high throughput microfluidic vortex technology, Oncotarget 7 (11) (2016) 12748.

[3] M. Dhar, J. Wong, J. Che, M. Matsumoto, T. Grogan, D. Elashoff, E.B. Garon, J.W. Goldman, E.S. Christen, D. Di Carlo, et al., Evaluation of pd-l1 expression on vortex-isolated circulating tumor cells in metastatic lung cancer, Sci. Rep. 8 (1) (2018) 1–10.

[4] J. Chen, J. Li, Y. Sun, Microfluidic approaches for cancer cell detection, characterization, and separation, Lab Chip 12 (10) (2012) 1753–1767.

[5] W. Lee, H. Amini, H.A. Stone, D. Di Carlo, Dynamic self-assembly and control of microfluidic particle crystals, Proc. Natl. Acad. Sci. 107 (52) (2010) 22413–22418.

[6] K. Hood, M. Roper, Pairwise interactions in inertially driven one-dimensional microfluidic crystals, Phys. Rev. Fluids 3 (9) (2018) 094201.
[7] A. Sierou, J.F. Brady, Accelerated stokesian dynamics simulations, J. Fluid Mech. 448 (2001) 115–146.
[8] N. Liron, E. Barta, Motion of a rigid particle in Stokes flow: a new second-kind boundary-integral equation formulation, J. Fluid Mech. 238 (1992) 579–598.
[9] H.T. Kazerooni, W. Fornari, J. Hussong, L. Brandt, Inertial migration in dilute and semidilute suspensions of rigid particles in laminar square duct flow, Phys. Rev. Fluids 2 (8) (2017) 084301.
[10] N. Nakagawa, T. Yabu, R. Otomo, A. Kase, M. Makino, T. Itano, M. Sugihara-Seki, Inertial migration of a spherical particle in laminar square channel flows from low to high Reynolds numbers, J. Fluid Mech. 779 (2015) 776.
[11] S. Xu, B. Gao, A. Lofquist, M. Fernando, M.-C. Hsu, H. Sundar, B. Ganapathysubramanian, An octree-based immersogeometric approach for modeling inertial migration of particles in channels, Comput. Fluids (2020) 104764.
[12] S.R. Bazaz, A. Mashhadian, A. Ehsani, S.C. Saha, T. Krüger, M.E. Warkiani, Computational inertial microfluidics: a review, Lab Chip 20 (6) (2020) 1023–1048.
[13] D.-K. Sun, Y. Wang, A.-P. Dong, B.-D. Sun, A three-dimensional quantitative study on the hydrodynamic focusing of particles with the immersed boundary–lattice Boltzmann method, Int. J. Heat Mass Transf. 94 (2016) 306–315.
[14] B. Chun, A. Ladd, Inertial migration of neutrally buoyant particles in a square duct: an investigation of multiple equilibrium positions, Phys. Fluids 18 (3) (2006) 031704.
[15] C. Schaaf, F. Rühle, H. Stark, A flowing pair of particles in inertial microfluidics, Soft Matter 15 (9) (2019) 1988–1998.
[16] K.J. Humphry, P.M. Kulkarni, D.A. Weitz, J.F. Morris, H.A. Stone, Axial and lateral particle ordering in finite Reynolds number channel flows, Phys. Fluids 22 (8) (2010) 081703.
[17] S. Kahkeshani, H. Haddadi, D. Di Carlo, Preferred interparticle spacings in trains of particles in inertial microchannel flows, J. Fluid Mech. 786 (2016).
[18] J.A. Schonberg, E. Hinch, Inertial migration of a sphere in Poiseuille flow, J. Fluid Mech. 203 (1989) 517–524.
[19] A. Prosperetti, H. Oguz, Physalis: a new o (n) method for the numerical simulation of disperse systems: potential flow of spheres, J. Comput. Phys. 167 (1) (2001) 196–216.
[20] A.J. Sierakowski, A. Prosperetti, Resolved-particle simulation by the physalis method: enhancements and new capabilities, J. Comput. Phys. 309 (2016) 164–184.
[21] M. Van Dyke, Perturbation Methods in Fluid Mechanics, annotated edition, STIA, vol. 75, 1975, p. 46926.
[22] I. Proudman, J. Pearson, Expansions at small Reynolds numbers for the flow past a sphere and a circular cylinder, J. Fluid Mech. 2 (3) (1957) 237–262.
[23] S. Kaplun, P. Lagerstrom, Asymptotic expansions of Navier-Stokes solutions for small Reynolds numbers, J. Math. Mech. (1957) 585–593.
[24] S. Kim, S.J. Karrila, Microhydrodynamics: Principles and Selected Applications, Courier Corporation, 2013.
[25] R. Cox, H. Brenner, The lateral migration of solid particles in Poiseuille flow—i theory, Chem. Eng. Sci. 23 (2) (1968) 147–173.
[26] K. Hood, S. Lee, M. Roper, Inertial migration of a rigid sphere in three-dimensional Poiseuille flow, J. Fluid Mech. 765 (2015) 452–479, https://doi.org/10.1017/jfm.2014.739.
[27] J. Happel, H. Brenner, Low Reynolds Number Hydrodynamics: with Special Applications to Particulate Media, vol. 1, Springer Science & Business Media, 2012.
[28] R. Cortez, The method of regularized stokeslets, SIAM J. Sci. Comput. 23 (4) (2001) 1204–1225.
[29] C. Taylor, P. Hood, A numerical solution of the Navier-Stokes equations using the finite element technique, Comput. Fluids 1 (1) (1973) 73–100.
[30] A. Wathen, D. Loghin, D. Kay, H. Elman, D. Silvester, A new preconditioner for the Oseen equations, in: Numerical Mathematics and Advanced Applications, Springer, 2003, pp. 979–988.
[31] R. Sibson, A brief description of natural neighbour interpolation, in: Interpreting Multivariate Data, 1981.
[32] J.-A. Kim, J. Lee, C. Wu, S. Nam, D. Di Carlo, W. Lee, Inertial focusing in non-rectangular cross-section microchannels and manipulation of accessible focusing positions, Lab Chip 16 (6) (2016) 992–1001.
[33] K. Hood, S. Kahkeshani, D. Di Carlo, M. Roper, Direct measurement of particle inertial migration in rectangular microchannels, Lab Chip 16 (15) (2016) 2840–2850.
[34] D. Di Carlo, J.F. Edd, K.J. Humphry, H.A. Stone, M. Toner, Particle segregation and dynamics in confined flows, Phys. Rev. Lett. 102 (9) (2009) 094503.
[35] H. Yamashita, T. Itano, M. Sugihara-Seki, Bifurcation phenomena on the inertial focusing of a neutrally buoyant spherical particle suspended in square duct flows, Phys. Rev. Fluids 4 (12) (2019) 124307.
[36] S.H. Strogatz, Nonlinear Dynamics and Chaos with Student Solutions Manual: With Applications to Physics, Biology, Chemistry, and Engineering, CRC Press, 2018.
[37] J.F. Edd, D. Di Carlo, K.J. Humphry, S. Köster, D. Irimia, D.A. Weitz, M. Toner, Controlled encapsulation of single-cells into monodisperse picolitre drops, Lab Chip 8 (8) (2008) 1262–1264.
[38] A.T. Chwang, T. Wu, Hydromechanics of low-Reynolds-number flow. Part 2. Singularity method for Stokes flows, J. Fluid Mech. 67 (4) (1975) 787–815.
[39] S.S. Kuntaegowdanahalli, A.A.S. Bhagat, G. Kumar, I. Papautsky, Inertial microfluidics for continuous particle separation in spiral microchannels, Lab Chip 9 (20) (2009) 2973–2980.
[40] A.J. Mach, J.H. Kim, A. Arshi, S.C. Hur, D. Di Carlo, Automated cellular sample preparation using a centrifuge-on-a-chip, Lab Chip 11 (17) (2011) 2827–2834.
[41] R. Courant, D. Hilbert, Methods of Mathematical Physics: Partial Differential Equations, John Wiley & Sons, 2008.
[42] K. Hood, lambs-soln-exterior, https://github.com/kthood/lambs-soln-exterior, 2015.