

Spectrally-Accurate Close Evaluation Schemes for Stokes Boundary Integral Operators

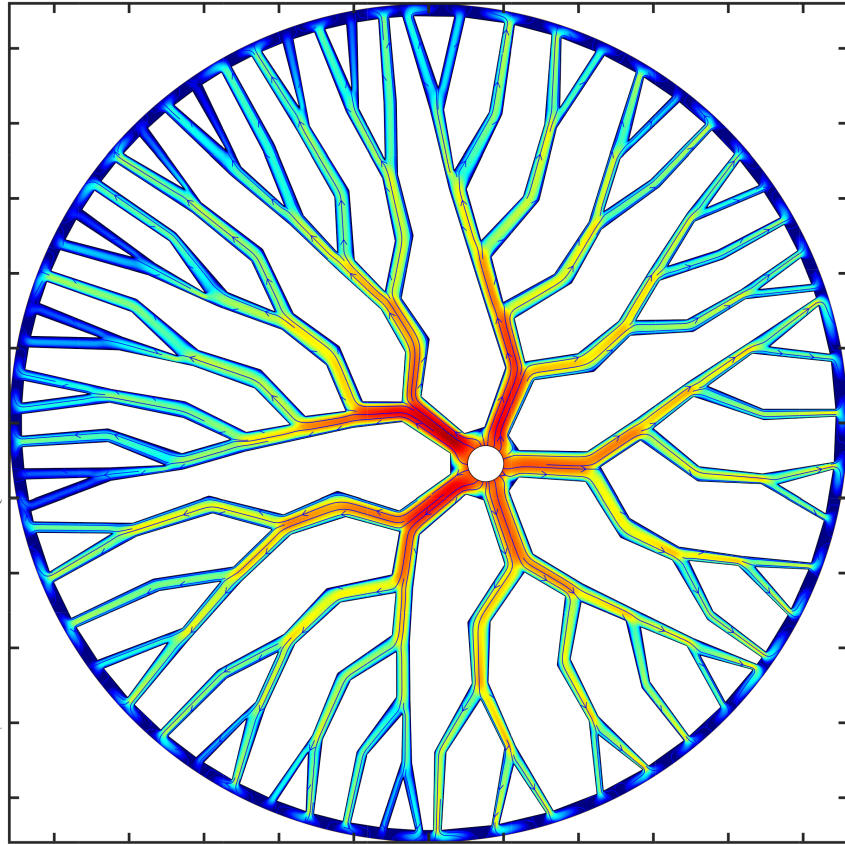
by

Bowei Wu

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Applied and Interdisciplinary Mathematics and Scientific Computing)
in The University of Michigan
2019

Doctoral Committee:

Associate Professor Shravan Veerapaneni, Chair
Associate Professor Silas Alben
Professor Robert Krasny
Associate Professor Allen Liu



Bowei Wu

boweiwu@umich.edu

ORCID iD: 0000-0003-2726-2812

© Bowei Wu 2019

All Rights Reserved

ACKNOWLEDGEMENTS

The works presented in this dissertation are collaborative research with my advisor Shravan Veerapaneni, Professor Alex Barnett, and my colleague Hai Zhu. I am grateful for their help on the course of making these exciting discoveries and developing the beautiful results.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	ii
LIST OF FIGURES	v
LIST OF TABLES	xi
ABSTRACT	xii
CHAPTER	
I. Introduction	1
II. Spectrally-Accurate Quadratures for Evaluation of Layer Potentials Close to the Boundary for the 2D Stokes and Laplace Equations	4
2.1 Introduction	4
2.2 Laplace and Stokes layer potentials	8
2.2.1 Stokes potentials expressed via Laplace potentials	9
2.3 Barycentric approximation of the interior and exterior Cauchy integral formulae	11
2.3.1 Interior case	13
2.3.2 Exterior case	14
2.3.3 First derivative and its barycentric form, interior case	15
2.3.4 First derivative with a barycentric form, exterior case	17
2.3.5 Numerical tests of values and derivatives close to the curve	18
2.4 Evaluation of Laplace layer potentials with global quadrature	20
2.4.1 Laplace double-layer potential	20
2.4.2 Laplace single-layer potential	22
2.4.3 Numerical tests of Laplace evaluation quadratures	28
2.5 Stokes close evaluation scheme and numerical results	30
2.6 Conclusions and discussion	38

III. Solution of Stokes Flow in Complex Nonsmooth 2D Geometries via a Linear-Scaling High-Order Adaptive Integral Equation Scheme	41
3.1 Introduction	41
3.2 Mathematical preliminaries	46
3.2.1 Boundary value problem and integral equation formulation	47
3.2.2 Fundamental contour integrals	49
3.2.3 Laplace layer potentials	50
3.2.4 Stokes velocity layer potentials	51
3.3 Nyström discretization and evaluation of layer potentials	53
3.3.1 Overview: discretization and the plain Nyström formula	53
3.3.2 Close-evaluation and self-evaluation corrections	56
3.3.3 Close-evaluation of potentials	58
3.3.4 Computation of close-evaluation matrix blocks	64
3.4 Adaptive panel refinement	65
3.5 Numerical results and discussion	71
3.6 Conclusions	78
IV. Electrohydrodynamics of Deflated Vesicles: Budding, Rheology and Pairwise Interactions	80
4.1 Introduction	80
4.2 Problem formulation	82
4.2.1 Governing equations	82
4.2.2 Numerical Method	85
4.3 Results and discussions	87
4.3.1 Isolated vesicle EHD: transition from squaring to budding in POP	88
4.3.2 Electro-rheology in the dilute limit	90
4.3.3 Two-body EHD interactions	92
4.4 Conclusions	97
V. Summary	99
APPENDIX	101
BIBLIOGRAPHY	106

LIST OF FIGURES

Figure

2.1	Motivational example: snapshots from a Stokes simulation of nine vesicles squeezing through a fixed-wall microfluidic device. Using the close-evaluation scheme presented here, accurate simulations can be run with only 32 points per vesicle. (Without the scheme, instabilities creep in and the simulation breaks down after a few time-steps.) Details of this application and its accuracy tests will be presented in [50].	6
2.2	(a) Curve (with $N = 200$ nodes shown) for the test of the interior Cauchy integral evaluation; points x (shown by + symbols along the straight line) lie at approximate distances from a node $0, 10^{-16}, 10^{-14}, \dots, 10^{-2}, 1$. The test function is $v(x) = 1/(x - b)$ where the pole $b = 1.1 + 1i$ (shown by *) is a distance of 0.5 from Γ . (b) Convergence of \log_{10} of the absolute error (see colorbar at right) for $v(x)$, using formula (2.16), plotted vs distance of x from a node (horizontal axis) and number of nodes N (vertical axis). (c) Convergence of $v'(x)$, using the formula (2.21). (d) Convergence of $v'(x)$, inserting (2.22) into (2.21) for distances below $\delta = 10^{-2}$. Exterior results are very similar, so we do not show them.	19
2.3	Evaluation error for the Laplace single-layer potential on a grid exterior to the closed curve Γ defined by the radial function $r(\theta) = 1 + 0.3 \cos 5\theta$. The color shows \log_{10} of the errors relative to a known solution to the exterior Neumann BVP. (a) uses the native periodic trapezoid rule (2.13), whereas (b) and (c) use the scheme of Section 2.4.2.2 (note the change in color scale). In all cases $N = 240$ nodes are used for solution and evaluation.	29

2.4	Performance of proposed globally compensated Stokes SLP quadrature in evaluating the hydrodynamic interaction force (2.43) between two elliptical bubbles parametrized as translations of $(\cos \theta, 2 \sin \theta)$. δ is the minimum distance between the two interfaces; we plot the l^∞ -norm error against the number of nodes N for $\delta = 0.1, 0.01$ and 0.001 . (b) Convergence of new globally compensated scheme. (c) Native quadrature (trapezoidal rule), showing very poor convergence for small δ values.	32
2.5	Absolute errors in computing the Stokes SLP at a fixed distance $\delta = 0.001$ away from the boundary of three different geometries with varying complexity: ellipses with aspect ratios 2, 4, and 8. (a) Shows ellipses and target point x (case $\delta = 0.1$ is shown to make the separation visible). (b) Results for our proposed scheme. In all cases, the errors decay exponentially with discretization size although, as expected, the absolute errors are higher for higher aspect ratio geometries. Therefore, in the case of our scheme, the given data (boundary, density) dictates the overall accuracy. (c) Results for native quadrature. In this case, convergence rate decreases but more striking is the fact the errors are almost the same for a particular number of points. This clearly implies that the near singular integral evaluation dominates the overall error.	33
2.6	(a) Streamlines of the velocity field generated by Stokeslets shown in the interior, for testing the exterior BVPs in Example 3 of Sec. 2.5. The boundary Γ is given by the polar function $f(\theta) = 1 + 0.3 \cos 5\theta$. (b) \log_{10} of absolute error in computing the velocity (on a grid of spacing 0.02) for the exterior Neumann Stokes BVP, using $N = 250$ discretization points on the boundary for both the Nyström solve and the evaluation as in Example 3 of Sec. 2.5.	34
2.7	Test of exterior Dirichlet BVP for the 20 elliptical vesicles of Example 4, using the scheme of Sec. 2.5. (a) Streamlines of the velocity field in the exterior of vesicles. (b) \log_{10} of error in velocity field using the Nyström method (with GMRES) to solve the density, using the close evaluation scheme for vesicle-vesicle interactions and for final flow evaluation, with $N = 150$ nodes per vesicle. (c) convergence of the sup norm of error in the velocity field over the grid of values plotted in (b).	36

2.8	Example 5: Exterior Stokes Dirichlet BVP for 20 close rigid ellipses with no-slip boundary condition, driven by a shear flow u_∞ , using the scheme of Sec. 2.5. (a) Streamlines of the velocity field (blue lines with arrows), with magnitude of density σ indicated on the boundaries (color scale on right). (b) \log_{10} of error in velocity field using the Nyström method (with GMRES) to solve the density, using the close evaluation scheme for vesicle-vesicle interactions and for final flow evaluation, with $N = 300$ nodes per vesicle. (c) Convergence of the sup norm of error in the velocity field over the grid of values plotted in (b).	38
3.1	Snapshot from a simulation of bacterial suspension flow in a microfluidic chip geometry, which is inspired from the design proposed in [36]. A squirmer model [38] is used for modeling the bacteria, which treats them as rigid bodies with a prescribed slip at the fluid-structure interface. Thereby, we solve the Stokes equations with a no-slip boundary condition on the microfluidic chip geometry, a prescribed tangential velocity on the squirmer boundaries and an imposed parabolic flow profile at the inlet and outlet. We used 730,080 discretization points for the chip boundary, resulting in 1,460,160 degrees of freedom, and 128 discretization points at each of the 120 squirmers. GMRES took about 10 hours to reach a relative residual of 5.6×10^{-8} , using an 8-core 3.6 GHz Intel Core i7 processor with 128 GB of RAM. Color indicates the magnitude of fluid velocity. The estimated PDE relative L_2 -norm error is 2×10^{-5}	43
3.2	Solution of the Stokes equation in a nonsmooth circular vascular network with Dirichlet boundary condition. We apply no-slip boundary condition at all branch walls, and it is driven by a uniform flow from inner to outer circle. Color here indicates log of the magnitude of fluid velocity. We used automatically generated panels for both smooth boundaries and 378 corners, resulting in 356,580 degrees of freedom. GMRES took about 1 hour to reach a relative residual of 7.61×10^{-11} on an 8-core 4.0 GHz Intel Core i7 desktop. The PDE solution has a relative L_2 -norm error of 1×10^{-9} . Three high-resolution \log_{10} error plots that correspond to different user-requested tolerance ϵ near the same reentrant corner are shown on the left; here the short normal lines show panel endpoints, and the black dots quadrature nodes. . .	46
3.3	Special handling of close evaluation branch cut when the panel is touching a reentrant corner. (a) The target panel Λ' is crossing the branch cut of the source panel Λ defined by (3.33), resulting in wrong close evaluation values. (b) Changing the sign of ϕ in (3.33) flips the branch cut to the other side; close evaluation at the targets are now correct.	62

3.4	(a) Linear shear flow past a starfish-shaped island. Streamlines of the flow and panel endpoints (small segments) are shown. Color represents the \log_{10} error of the velocity computed under tolerance $\epsilon = 10^{-12}$ (resulted in $2N = 2184$ degrees of freedom). (b) Convergence of the maximum relative error ϵ_{\max} versus requested tolerance ϵ , where the traction field is computed along the (1,2)-direction. (c) Convergence of the maximum error ϵ_{\max} versus the square root of the number of nodes.	72
3.5	(a) Linear shear flow around a shuriken-shaped island with 8 corners. Streamlines of the flow and panel endpoints (small segments) are shown. Color represents the \log_{10} error of the velocity computed under tolerance $\epsilon = 10^{-10}$ (resulted in $2N = 6640$ degrees of freedom). (b) Convergence of the maximum error ϵ_{\max} versus requested tolerance ϵ . (c) Convergence of the maximum error ϵ_{\max} versus the square root of the number of nodes; root-exponential convergence would result in a straight line.	73
3.6	(a) Streamlines of a shear flow past 50 randomly generated polygonal islands with a total number of 253 corners. Color on the polygon boundaries indicate the magnitude of density σ . (b) \log_{10} of absolute error of the velocity, computed using $2N = 222140$ degrees of freedom. Error is measured on a 1000×1000 grid (spacing $\Delta x \approx 8.3 \times 10^{-3}$) by comparing to the solution obtained with $\epsilon = 10^{-10}$. (c) Convergence of the maximum error ϵ_{\max} versus the square root of the number of nodes.	74
3.7	Error and timing of solving the vascular network BVP. (a) Convergence of errors in log-linear scale. (b) Log-log scale plot of the total CPU time per GMRES iteration, which consists of the FMM time, shown in (c), and the close correction time, shown in (d).	75
3.8	Convergence of a uniform flow past two touching disks that are $d = 10^{-6}$ apart, and whose radii are 1 and 0.1. The required number of unknowns in the adaptive scheme is much less than the global scheme with uniform resolution; see Example 5.	77
4.1	Snapshots from two different simulations of a single vesicle subjected to an external electric field, with $\Lambda = 0.1$, $G = 0$, $Ca = 0$ and (a) $\Delta = 0.9$, $\beta = 3.2$ and (b) $\Delta = 0.5$, $\beta = 12.8$. While the vesicle undergoes a prolate-oblate-prolate transition, the transient “square-like” shapes observed here (in (a)) and in prior numerical studies cannot be observed when the reduced volume is lowered. Instead, to sustain the electric compression forces, the vesicle forms buds as it undergoes the POP transition (more details on this phase are shown in Figure 4.2).	88

4.2	Streamlines (left) and electric field lines (middle) plotted at the moment when the vesicle with $\Delta = 0.5$ shown in Figure 4.1(b) forms buds while undergoing POP transition. In the left figure, the membrane color indicates the magnitude of tension while on the middle figure, it indicates the magnitude of the transmembrane potential. The right figure gives a closer look at the <i>narrowest</i> buds formed under different β 's, where the times correspond to this state for $\beta = 9.6, 12.8$ and 16 are $t = 0.253, 0.216$ and 0.184 , respectively. The neck of the buds becomes narrower as β increases.	90
4.3	Phase diagrams of vesicle dynamics for different reduced volumes as a function of the membrane conductivity G and electric field strength β . Here, the different phases of the dynamics are indicated by O when the vesicle remains oblate for all times or P when it remains prolate or POP when it transitions from prolate to oblate to prolate shapes. For all the cases, the conductivity ratio Λ is set to 0.1 , $Ca = 0$. . .	91
4.4	Single vesicle rheology when $G = 4$, $\beta = 6.4$ and $Ca = 10$. Plots of the effective viscosity (left), angle of inclination (middle) and the tangential velocity (right) when a vesicle is suspended in a linear shear flow as a function of the conductivity ratio. We can observe that the inclination angle increases as Λ is increased i.e., the vesicle tries to align with the electric field direction and away from the direction of shear. Thereby, it presents more resistance to imposed flow, leading to higher effective viscosity. One remarkable effect of low reduced volume, as is evident from the right panel, is that the vesicle tank-treads in the opposite direction compared to high reduced volume vesicles when Λ is small.	92
4.5	Dependence of effective viscosity $[\mu]$ on β and Δ . Conductivity $G = 4$ and $Ca = 10$ (top row) or $Ca = 50$ (bottom row). We note that (i) $[\mu]$ is higher whenever the equilibrium angle at which the vesicle tank-treads is away from the direction of shear and (ii) when Δ is close to 1 (vesicle closer to a circle), $[\mu]$ is nearly β -independent and shear-independent (as can be expected).	93
4.6	A summary of pairwise vesicle EHD interactions ($\Delta = 0.9$, $\beta = 3.2$, $Ca = 0$)	94
4.7	Snapshots from a simulation of two vesicles undergoing circulatory motion described in Figure 4.6 with $G = 0$ and $\Lambda = 0.5$. Here, one of the vesicles is colored by the magnitude of V_m (yellow indicates positive and blue indicates negative values respectively). We can observe that each vesicle undergoes tank-treading motion on its own (as indicated by the streamlines), they rotate about each other and the vesicles viewed as dipoles are always tilted with respect to the applied field direction.	95

4.8	Insensitivity of the EHD pairwise interactions to the initial offset from the aligned position. θ measures the angular offset of the two vesicles relative to the horizontally aligned position. (a) Chain formation. (b) Oscillatory motion. (c) Circulatory motion. In each case, the same pattern is observed regardless of the initial $\theta > 0$	96
4.9	Pairwise interactions for $G = 0$ vesicles of reduced volumes $\Delta = 0.7$ (with $\beta = 4.8$) and $\Delta = 0.99$ (with $\beta = 2.4$). $Ca = 0$. The behaviors (e.g. chain formation, oscillatory motion, circulatory motion) are the same as in the $\Delta = 0.9$ case (Fig. 4.6), showing that the mechanism of pairwise interactions is insensitive to the reduced volume. Note that the bud formation also happens with the case of $\Delta = 0.7, \Lambda = 0.1$.	97
4.10	Top figure: final configurations of eight separate simulations, each corresponding to a different membrane conductivity G . There is a continuous transition from a chain of prolates ($G \approx 0$) to a chain of oblates ($G \gg 0$). For certain intermediate value of G (e.g. $G = 0.096, 0.144, 0.192$) the chain formation process is accompanied with decaying oscillatory motions (lower left figure), while for more extreme values of G the vesicles directly form a chain without oscillations (lower right figure). For all simulations $\beta = 3.2, \Lambda = 0.9$, and $Ca = 0$	98

LIST OF TABLES

Table

2.1	Convergence of error for the Laplace layer potential evaluation scheme of Section 2.4 for BVPs solved on the curve shown in Figure 2.3. The maximum error in u or ∇u is taken over a square grid of spacing 0.01.	30
2.2	Convergence of velocity error (max over the evaluation grid) for the solutions of the four types of Stokes BVPs, on the curve shown in Fig. 2.6. The Nyström method is used to find the density, then the new close evaluation scheme is used, as described in Sec. 2.5.	33
3.1	<i>Results and statistics of solving the BVP in the vascular network in Figure 3.2 for various tolerance ϵ. Errors ϵ_{\max} and ϵ_{L_2} are measured on a 2160×2160 grid (spacing $\approx 2.5 \times 10^{-3}$) by comparing to the solution obtained from $\epsilon = 10^{-10}$. CPU time and RAM used are measured using [7].</i>	76

ABSTRACT

Dense particulate flow simulations using integral equation methods demand accurate evaluation of Stokes layer potentials on arbitrarily close interfaces. In this thesis, two spectrally-accurate integration schemes for close evaluation of 2D Stokes layer potentials are developed – a global quadrature for the moving particles (e.g., blood cells, vesicles) represented as smooth closed curves, and an adaptive panel quadrature for the stationary boundaries (e.g., vessel walls, microfluidic channels) which are more complex curves that can be non-smooth. Both schemes rely on expressing Stokes layer potentials in terms of Laplace potentials and related complex contour integrals, which are then evaluated accurately either through a singularity cancellation technique or using analytic expressions. Numerical examples are presented to demonstrate the robustness and super-algebraic convergence of both schemes. Finally, as an application of the integration schemes, we investigate the electrohydrodynamic interactions between (possibly deflated) vesicles, where interesting behaviors unique to vesicles, such as circulatory and oscillatory motions, are observed and analyzed.

CHAPTER I

Introduction

Complex fluids are fluids that exhibit unusual macroscopic behavior, even if the microscopic governing laws are simple and linear. One major category of complex fluids is the particulate flows – flows that contain rigid and/or deformable particles. These systems are ubiquitously seen in industrial and biomedical applications, examples include blood stream, vesicle suspensions, bacterial flow, and a variety of microfluidic systems. Besides experiments, direct numerical simulations are often the only means for gaining insights into their non-equilibrium behavior.

One of the main challenges for existing numerical methods to simulate particulate flows is to accurately resolve the particle-particle or the particle-wall interactions. In these complex fluid flows, more often than not, particles approach very close to each other or to the wall when subjected to flow and other effects (such as electric stress). Numerical instabilities arise when the near interactions are not computed accurately, jeopardizing the entire simulation.

The boundary integral equation (BIE) method is particularly suitable for particulate flow simulations because, under the BIE formulation, all the unknowns reside on the surfaces of the particles and walls, leading to a reduction of dimensionality compared to methods that discretize the equations directly in the bulk fluid. In addition, with an appropriate choice of integral operators (which are called *layer potentials* in

the language of potential theory), many equations that govern complex fluids can be reformulated as second kind integral equations. These equations are well-conditioned and lead to fast convergence when solved with iterative linear algebra solvers. Moreover, state-of-the-art fast algorithms such as the Fast Multipole Methods (FMM) are widely used to build powerful fast solvers for these linear systems.

Under the BIE framework, the particle-particle or the particle-wall close interactions demand accurate evaluation of nearly singular integrals associated to the Stokes layer potentials. This is called the *close evaluation* problem, which will be the main focus of this thesis.

We will extend Helsing-Ojala’s close evaluation technique for the Laplace layer potentials [31] to develop two close evaluation schemes for the Stokes layer potentials and their associated pressure and traction formulae. These potentials all take the form of

$$\int_{\Gamma} K(x, y)\tau(y) ds_y, \quad x \in \mathbb{R}^2 \tag{1.1}$$

which is defined on a planar curve Γ for some density function τ . When the target point x is close to a source point $y \in \Gamma$, the kernel $K(x, y)$ becomes nearly singular. The two close evaluation schemes we will develop in this thesis are

- a global scheme for when Γ is a smooth closed curve, and
- a panel scheme for when Γ is an open curve, which is suitable for more complex geometries that are represented as a collection of disjoint panels.

Both schemes rely on expressing Stokes layer potentials in terms of Laplace potentials and the related complex contour integrals, which are then accurately evaluated with the help of complex analysis and techniques such as singularity cancellation. In addition, for the panel quadrature scheme we also formulate a set of rules for adaptively refining the panels used to represent the boundaries, so that a user-specified error tolerance can be achieved automatically.

Finally, as an application of the global scheme, we will investigate the *electrohydrodynamics (EHD)* of vesicles. Vesicles are deformable particles sharing the same structural component as many biological cells, hence their EHD has been a paradigm for understanding how general biological cells behave when subjected to electric signals. We develop a new BIE formulation and numerical method for the vesicle EHD, which is able to handle multiple vesicles that can be deflated. Our numerical experiments investigate new phenomena such as “budding” and pairwise EHD interactions. Close evaluation is important in this study since vesicles are driven very close to each other by the electric stress.

This thesis is organized as follows. The next two chapters (Chapter II and III) introduce the close evaluation schemes, where the global quadrature is introduced in Chapter II and the adaptive panel quadrature in Chapter III. Chapter IV investigates the vesicle EHD. Then a summary is followed in Chapter V.

CHAPTER II

Spectrally-Accurate Quadratures for Evaluation of Layer Potentials Close to the Boundary for the 2D Stokes and Laplace Equations

Preamble. This and the next chapters focus on the accurate quadratures for 2D Stokes layer potentials. This chapter introduces a global quadrature scheme for smooth closed curves. In this scheme, the Stokes potentials are first split into Laplace potentials, which in turn are rewritten as complex contour integrals. These contour integrals are evaluated accurately using the periodic trapezoid rule (PTR) with the help of a barycentric-type formula for the Cauchy integral [37]. Our scheme is spectrally accurate for target points arbitrarily close to the source curve. This is a joint work with Alex Barnett and Shravan Veerapaneni that is published in [8].

2.1 Introduction

Dense suspensions of deformable particles in viscous fluids are ubiquitous in natural and engineering systems. Examples include drop, bubble, vesicle, swimmer and blood cell suspensions. Unlike simple Newtonian fluids, the laws describing their flow behavior are not well established, owing to the complex interplay between the deformable microstructure and the macroscale flow. Besides experiments, direct numer-

ical simulations are often the only means for gaining insights into the non-equilibrium behavior of such complex fluids. One of the main challenges for existing numerical methods to simulate *dense* or *concentrated* suspensions is to accurately resolve the particle-particle or the particle-wall interactions. In these complex fluid flows, more often than not, particles approach very close to each other when subjected to flow (e.g., see Figure 2.1). Numerical instabilities arise when the near interactions are not computed accurately, jeopardizing the entire simulation.

Boundary integral methods are particularly well-suited for vanishing Reynolds number problems where the Stokes equations govern the ambient fluid flow [60]. The advantages over grid- or mesh-based discretizations include: a much smaller number of unknowns (exploiting the reduced dimensionality), no need for smearing of interface forces onto a grid, the availability of very high-order discretizations, and of accelerated solvers such as the fast multipole method (FMM) [25] for handling the dense matrices in linear time.

The kernels of the integral operators such as the single-layer potential,

$$(\mathcal{S}\tau)(x) = \int_{\Gamma} G(x, y)\tau(y) ds_y, \quad x \in \mathbb{R}^2, \quad (2.1)$$

defined on a smooth closed planar curve Γ for some smooth density function τ , become *nearly singular* when the target point x is close to Γ . Neither smooth quadrature rules (such as the trapezoidal rule) nor singular quadratures are effective (uniformly convergent) for nearly singular integrals; for example, the error in a fixed smooth quadrature rule grows exponentially to $\mathcal{O}(1)$ as x approaches Γ [9, Thm. 3]. The objective of this paper is the design of numerical integration schemes for (2.1) that, given the smooth density τ sampled at the N nodes of a periodic trapezoid rule on Γ , exhibit *superalgebraic convergence* in N , with rate *independent* of the distance of x from Γ .

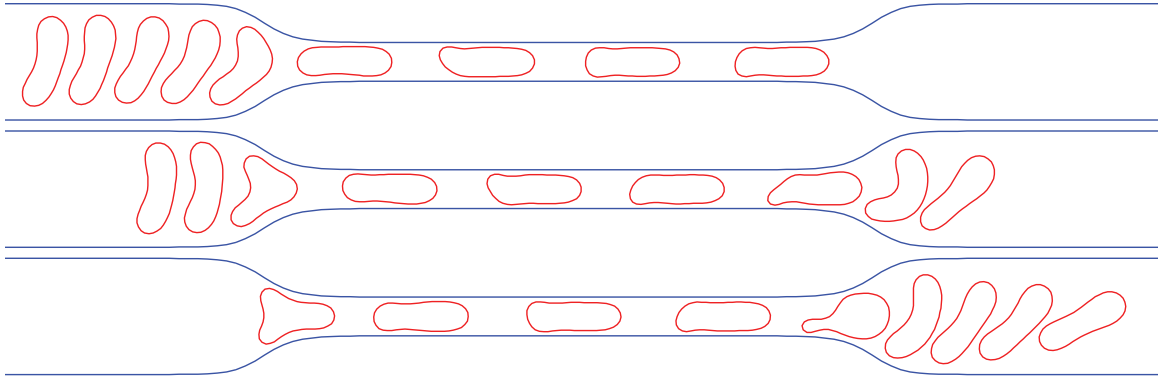


Figure 2.1: Motivational example: snapshots from a Stokes simulation of nine vesicles squeezing through a fixed-wall microfluidic device. Using the close-evaluation scheme presented here, accurate simulations can be run with only 32 points per vesicle. (Without the scheme, instabilities creep in and the simulation breaks down after a few time-steps.) Details of this application and its accuracy tests will be presented in [50].

Remark 1. For smooth geometries and data many simple boundary integral solution methods, such as Nyström’s method [44], exhibit superalgebraic convergence in the density. Our goal is thus to provide layer potential evaluations that are as accurate as the N -node spectral interpolant to the density itself, i.e. *limited only by the data samples available*. This means that in simulations, even those with close-to-touching geometry, only the smallest number of unknowns N required to capture the density is needed, and optimal efficiency results.

Specifically, we develop a new suite of tools for evaluating layer potentials such as (2.1) (and their derivatives) on smooth closed curves for Laplace’s equation in two dimensions (2D), and from this build evaluators for Stokes potentials that can handle close-to-touching geometries and flow field evaluations arbitrarily close to curves, with accuracies approaching machine precision.

Despite its importance in practical applications, very few studies have addressed the accuracy issue with nearly singular integrals. Adaptive quadrature on a target-by-target basis is impractically slow. Beale et al. [12, 80] proposed a regularized kernel approach that attains third-order accuracy by adding analytically determined

corrections. Ying et al. [97] developed a method that interpolates the potentials along extended surface normals. Quaife–Biros [61] applied this in the context of 2D vesicle flows, attaining 5th-order accuracy. Despite its high-order accuracy, the computational cost scales as a suboptimal $\mathcal{O}(N^{3/2})$, due to the need to compute values near the boundary using an upsampled trapezoid rule. Helsing–Ojala [31], exploiting Cauchy’s theorem and recurrence relations, developed 16th-order panel-based close evaluation schemes in 2D; these have recently been adapted to the complex biharmonic formulation of Stokes potentials by Ojala–Tornberg [56]. The recent QBX scheme [9, 41] can achieve arbitrarily high order for Laplace and Helmholtz potentials in 2D and 3D, but this requires upsampling the density by a factor of 4–6.

The pioneering work of Helsing–Ojala includes a “globally compensated” scheme for the 2D double-layer potential (DLP) [31, Sec. 3] which builds upon a second-kind barycentric-type formula for quadrature of Cauchy’s theorem due to Ioakimidis et al. [37]. The scheme we present extends this to the interior and exterior 2D single-layer potential (SLP); since the complex logarithmic kernel is not single-valued, this requires careful application of a spectrally-accurate product quadrature for the sawtooth function, in the style of Kress [43]. We also supply a true barycentric evaluation for first derivatives of layer potentials that is stable for target points arbitrarily close to nodes. Unlike [31], we prefer to use an underlying global quadrature (the periodic trapezoid rule) on Γ , since it is most commonly used for vesicle simulations such as Fig. 2.1, and (as our results show) is somewhat more efficient in terms of N than panel-based quadratures in this setting.¹

One advantage of our approach is that no auxiliary nodes or upsampling is needed; another is that the resulting discrete Cauchy sums are amenable for fast summation via the FMM. While the new tools for the planar Laplace equation are of interest in their own right, our main motivation and interest is to enable accurate close eval-

¹It is worth noting that even some panel-based schemes exploit global schemes for adaptivity [56].

uation of Stokes potentials to target applications in interfacial fluid mechanics. We accomplish this using the well-known fact that Stokes potentials can be written in terms of Laplace potentials and their derivatives [19, 81, 88]. Specifically in 2D, the Stokes SLP requires three Laplace SLP evaluations and the DLP requires five Laplace DLP evaluations (two of which are Cauchy-type). We demonstrate in several numerical experiments that our Stokes evaluations are very nearly as accurate as for Laplace.

The paper is organized as follows. We define Laplace and Stokes integral representations and set up notation in Section 2.2. Barycentric-type formulae for Cauchy integrals and their derivatives in the interior and exterior of Γ are discussed in Section 2.3. Based on these formulae, we derive spectrally accurate global quadratures for close evaluation of Laplace potentials in Section 2.4, and demonstrate their performance in evaluating all four types of boundary value problem (BVP) solutions. We test performance of the quadratures applied to Stokes potentials, and for the four Stokes BVPs, including one with close-to-touching boundaries, in Section 2.5. Finally, we summarize and discuss future work in Section 2.6.

2.2 Laplace and Stokes layer potentials

Let Γ be a smooth closed Jordan curve in \mathbb{R}^2 , with outwards-directed unit normal n_y at the point $y \in \Gamma$. Let Ω be the interior domain of Γ , and $\Omega^c := \mathbb{R}^2 \setminus \bar{\Omega}$ be the exterior domain. Let $\tau \in C(\Gamma)$ be a density function. We review some standard definitions [44, Ch. 6]. The Laplace SLP is defined by

$$(\mathcal{S}\tau)(x) := \frac{1}{2\pi} \int_{\Gamma} \left(\log \frac{1}{\rho} \right) \tau(y) ds_y \quad x \in \mathbb{R}^2. \quad (2.2)$$

where the distance is $\rho := |r|$, the displacement $r := x - y$, and $|x| := \sqrt{x_1^2 + x_2^2}$ is the Euclidean length of $x \in \mathbb{R}^2$. Finally, ds_y is the arc length element on Γ . The Laplace

DLP is defined by

$$(\mathcal{D}\tau)(x) := \frac{1}{2\pi} \int_{\Gamma} \left(\frac{\partial}{\partial n_y} \log \frac{1}{\rho} \right) \tau(y) ds_y = \frac{1}{2\pi} \int_{\Gamma} \left(\frac{r \cdot n_y}{\rho^2} \right) \tau(y) ds_y \quad x \in \mathbb{R}^2 \setminus \Gamma . \quad (2.3)$$

Associating \mathbb{C} with \mathbb{R}^2 , and noticing that the complex line element is $dy = in_y ds_y$, for purely real τ the DLP may also be written as the real part of a Cauchy integral, as follows,

$$(\mathcal{D}\tau)(x) = \operatorname{Re} v(x) , \quad \text{where} \quad v(x) := \frac{1}{2\pi i} \int_{\Gamma} \frac{\tau(y)}{x-y} dy \quad x \in \mathbb{C} \setminus \Gamma . \quad (2.4)$$

Let S be the restriction of \mathcal{S} to evaluation on Γ , in other words S is the boundary integral operator with kernel $k(x, y) = (1/2\pi) \log 1/\rho$. $(S\tau)(x)$ exists as an improper integral. Let D be the restriction of \mathcal{D} to evaluation on Γ , in other words D is the boundary integral operator with kernel $k(x, y) = (1/2\pi)(r \cdot n_y/\rho^2)$, taken in the principal value sense [44, Sec. 6.3]. D has a smooth kernel when Γ is smooth.

We define the interior and exterior boundary limits of a function u defined in $\mathbb{R}^2 \setminus \Gamma$, at the point $x \in \Gamma$, by $u^{\pm}(x) := \lim_{h \rightarrow 0^+} u(x \pm hn_x)$. Likewise, $u_n^{\pm}(x) := \lim_{h \rightarrow 0^+} n_x \cdot \nabla u(x \pm hn_x)$. We will need the following standard jump relations [44, Sec. 6.3]. For any C^2 -smooth curve Γ , and density function $\tau \in C(\Gamma)$,

$$(\mathcal{S}\tau)_n^{\pm} = (D^T \mp \frac{1}{2})\tau \quad (2.5)$$

$$(\mathcal{D}\tau)^{\pm} = (D \pm \frac{1}{2})\tau . \quad (2.6)$$

2.2.1 Stokes potentials expressed via Laplace potentials

Let $\sigma(y) = (\sigma_1(y), \sigma_2(y))$, for $y \in \Gamma$, be a smooth real-valued vector density function. The Stokes single and double layer potentials, denoted by \mathbf{S} and \mathbf{D} , are

defined by

$$(\mathbf{S}\sigma)(x) := \frac{1}{4\pi} \int_{\Gamma} \left(\log \frac{1}{\rho} I + \frac{r \otimes r}{\rho^2} \right) \sigma(y) ds_y, \quad (2.7)$$

$$(\mathbf{D}\sigma)(x) := \frac{1}{\pi} \int_{\Gamma} \left(\frac{r \cdot n_y}{\rho^2} \frac{r \otimes r}{\rho^2} \right) \sigma(y) ds_y, \quad (2.8)$$

where $r := x - y$ and $\rho := |r|$. In [19, 81, 88], fast algorithms to compute Stokes potentials were developed by expressing them in terms of Laplace potentials for which standard FMMs are applicable. We use the same strategy in this paper for close evaluation of Stokes potentials. Using the identity

$$\frac{r \otimes r}{\rho^2} \sigma = \frac{r}{\rho^2} (r \cdot \sigma) = (r \cdot \sigma) \nabla_x \log \rho,$$

we can rewrite the Stokes SLP in terms of the Laplace SLP (2.2) as

$$\begin{aligned} (\mathbf{S}\sigma)(x) &= \frac{1}{4\pi} \int_{\Gamma} \left(\log \frac{1}{\rho} \right) \sigma ds_y + \frac{1}{4\pi} \nabla \int_{\Gamma} \left(\log \frac{1}{\rho} \right) (y \cdot \sigma) ds_y \\ &\quad - \frac{1}{4\pi} x_1 \nabla \int_{\Gamma} \left(\log \frac{1}{\rho} \right) \sigma_1 ds_y - \frac{1}{4\pi} x_2 \nabla \int_{\Gamma} \left(\log \frac{1}{\rho} \right) \sigma_2 ds_y, \end{aligned} \quad (2.9)$$

where $\nabla = \nabla_x$ is assumed from now on. Therefore, three Laplace potentials (and their first derivatives), with density functions $y \cdot \sigma$, σ_1 , and σ_2 , need to be computed to evaluate the Stokes SLP. Similarly, using the identity

$$\nabla \left(\frac{r \cdot n_y}{\rho^2} \right) = \frac{n_y}{\rho^2} - (r \cdot n_y) \frac{2r}{\rho^4},$$

the Stokes DLP (2.8) can be written as

$$\begin{aligned}
(\mathbf{D}\sigma)(x) &= \frac{1}{2\pi} \int_{\Gamma} \frac{n_y}{\rho^2} (r \cdot \sigma) ds_y + \frac{1}{2\pi} \nabla \int_{\Gamma} \frac{r \cdot n_y}{\rho^2} (y \cdot \sigma) ds_y \\
&\quad - \frac{1}{2\pi} x_1 \nabla \int_{\Gamma} \frac{r \cdot n_y}{\rho^2} \sigma_1 ds_y - \frac{1}{2\pi} x_2 \nabla \int_{\Gamma} \frac{r \cdot n_y}{\rho^2} \sigma_2 ds_y .
\end{aligned} \tag{2.10}$$

The last three terms require Laplace DLP (2.3) potentials (and first derivatives) for the same three densities $y \cdot \sigma$, σ_1 , and σ_2 . However, the first term is not a strict Laplace DLP of the form (2.3): the derivative is taken in the σ rather than normal n_y direction. Yet, it can fit into our framework via two DLPs if we generalize slightly the Cauchy expression for the DLP (2.4) to allow complex densities τ , thus, using vector notation for the two components,

$$\frac{1}{2\pi} \int_{\Gamma} \frac{n_y}{\rho^2} (r \cdot \sigma) ds_y = \operatorname{Re} \frac{1}{2\pi i} \int_{\Gamma} \frac{(\tau_1, \tau_2)}{x - y} dy , \tag{2.11}$$

where

$$\tau_1 = (\sigma_1 + i\sigma_2) \frac{\operatorname{Re} n_y}{n_y} , \quad \tau_2 = (\sigma_1 + i\sigma_2) \frac{\operatorname{Im} n_y}{n_y} .$$

So, in total five Laplace DLPs are needed. Equations (3.16)–(2.11) allow us to compute Stokes potentials by simply applying accurate (Cauchy-form) Laplace close evaluation schemes, which are the focus of the next two sections.

2.3 Barycentric approximation of the interior and exterior Cauchy integral formulae

In this section we describe an efficient and accurate method to approximate a holomorphic function v from its boundary data sampled on a set of quadrature nodes on the closed curve Γ . The interior case is review of Ioakimidis et al. [37], but we extend the method to the exterior case in a different manner from Helsing–Ojala

[31, Sec. 3], and correct an accuracy problem in the standard formula for the first derivative. In Sec. 2.3.5 we present results showing uniform accuracy close to machine precision. We associate \mathbb{R}^2 with \mathbb{C} .

We fix a quadrature scheme on Γ , namely a set of nodes $y_j \in \Gamma$, $j = 1, \dots, N$, and corresponding weights w_j , $j = 1, \dots, N$, such that

$$\int_{\Gamma} f(y) ds_y \approx \sum_{j=1}^N w_j f(y_j)$$

holds to high accuracy for all smooth enough functions f . Let Γ be parametrized by the 2π -periodic map $Z : [0, 2\pi) \rightarrow \mathbb{R}^2$, with $Z(t) = Z_1(t) + iZ_2(t)$, such that $\Gamma = Z([0, 2\pi))$, and with “speed” $|Z'(t)| > 0$ for all $0 \leq t < 2\pi$. Then probably the simplest global quadrature arises from the N -point periodic trapezoid rule [18] with equal weights $2\pi/N$ and nodes

$$s_j := \frac{2\pi j}{N}, \quad j = 1, \dots, N. \quad (2.12)$$

By changing variable to arc-length on Γ , we get a boundary quadrature

$$y_j = Z(s_j), \quad w_j = \frac{2\pi}{N} |Z'(s_j)|, \quad j = 1, \dots, N. \quad (2.13)$$

It is well known that this rule can be exceptionally accurate: since the periodic trapezoid rule is exponentially convergent in N for analytic 2π -periodic integrands [17] [44, Thm. 12.6] [85], the rule (2.13) is exponentially convergent when Z_1 and Z_2 are analytic (hence Γ is an analytic curve), and the integrand f is analytic. The exponential rate is controlled by the size of the region of analyticity. In the merely smooth case we have superalgebraic convergence.

2.3.1 Interior case

Cauchy's formula states that any function v holomorphic in Ω whose limit approaching Γ from the inside is $v^- \in C(\Gamma)$ may be reconstructed from its boundary data alone:

$$\frac{1}{2\pi i} \int_{\Gamma} \frac{v^-(y)}{y-x} dy = \begin{cases} v(x), & x \in \Omega \\ 0, & x \in \Omega^c \end{cases} \quad (2.14)$$

Note that we have taken care to specify the data as the inside limit v^- ; this matters later when v will be *itself* generated by a Cauchy integral.

By combining (2.14) with the special case $(1/2\pi i) \int_{\Gamma} 1/(y-x) dy = 1$ for $x \in \Omega$, we have

$$\int_{\Gamma} \frac{v^-(y) - v(x)}{y-x} dy = 0, \quad x \in \Omega. \quad (2.15)$$

Even as the target point x approaches Γ , the integrand remains smooth (e.g. the neighborhood in which it is analytic remains large) because of the cancellation of the pole, and hence the quadrature rule (2.13) is accurate. Thus

$$\sum_{j=1}^N \frac{v_j^- - v(x)}{y_j - x} w_j \approx 0.$$

Rearranging derives a way to approximate $v(x)$, given the vector of values $v_j^- := v^-(y_j)$, namely

$$v(x) \approx \begin{cases} \frac{\sum_{j=1}^N \frac{v_j^-}{y_j - x} w_j}{\sum_{j=1}^N \frac{1}{y_j - x} w_j}, & x \in \bar{\Omega}, x \neq y_i, i = 1, \dots, N \\ v_i^-, & x = y_i \end{cases} \quad (2.16)$$

We have included x in the closure of Ω because in practical settings with roundoff error, targets may fall on Γ . The second formula is needed when x hits a node. We believe (2.16) is due to Ioakimidis et al. [37, (2.8)]; Helsing–Ojala call it “globally

compensated” quadrature [31]. It is in fact a *barycentric* Lagrange polynomial interpolation formula of the second form, with two crucial differences from the usual setting [13]:

- the nodes are no longer on the real axis, and
- the weights w_j come simply from quadrature weights on the curve rather than from the usual formula related to Lagrange polynomials.

In the case of Ω the unit disc with equispaced nodes, the equivalence of (2.16) to barycentric interpolation was recently explained by Austin–Kravanja–Trefethen [5, Sec. 2.6].

A celebrated key feature of barycentric formulae is numerical stability even as the evaluation point x approaches arbitrarily close to a node y_i . Although relative error grows without limit in both numerator and denominator of (2.16), due to roundoff error in the dominant terms $1/(y_i - x)$, these errors cancel (see [13, Sec. 7], [33] and references within). Thus close to full machine precision is attainable even in this limit.

2.3.2 Exterior case

We turn to the exterior Cauchy formula that states for v holomorphic in Ω^c ,

$$\frac{1}{2\pi i} \int_{\Gamma} \frac{v^+(y)}{y-x} dy = \begin{cases} v_{\infty}, & x \in \Omega \\ v_{\infty} - v(x), & x \in \Omega^c \end{cases} \quad (2.17)$$

where $v_{\infty} := \lim_{x \rightarrow \infty} v(x)$. We are interested only in v that can be generated by the exterior Cauchy integral, i.e. the case $v_{\infty} = 0$. We pick a simple non-vanishing function p with $p_{\infty} = 0$ that will play the role that the constant function played in the interior case. We choose $p(x) = 1/(x - a)$ where $a \in \Omega$ is a fixed arbitrary point

chosen not near Γ . Applying (2.17) this is generated by

$$\frac{1}{x-a} = \frac{-1}{2\pi i} \int_{\Gamma} \frac{(y-a)^{-1}}{y-x} dy$$

Multiplying both sides by $(x-a)$ we get a way to represent the constant function 1 via an exterior representation. Using this we create an exterior equivalent of (2.15),

$$\int_{\Gamma} \frac{v^+(y) - (y-a)^{-1}(x-a)v(x)}{y-x} dy = 0 \quad \text{for } x \in \Omega^c. \quad (2.18)$$

The integrand remains smooth and analytic even as the target x approaches Γ . Substituting the periodic trapezoid rule in (2.18) and rearranging as in the interior case gives

$$v(x) \approx \begin{cases} \frac{1}{x-a} \cdot \frac{\sum_{j=1}^N \frac{v_j^+}{y_j-x} w_j}{\sum_{j=1}^N \frac{(y_j-a)^{-1}}{y_j-x} w_j}, & x \in \Omega^c, x \neq y_i, i = 1, \dots, N \\ v_i^+, & x = y_i \end{cases} \quad (2.19)$$

which is our formula for accurate evaluation of the exterior Cauchy integral. It also has barycentric stability near nodes.

Remark 2. Helsing–Ojala [31, Eq. (27)] mention a different formula for the exterior case,

$$v(x) \approx \frac{\sum_{j=1}^N \frac{v_j^+}{y_j-x} w_j}{-2\pi i + \sum_{j=1}^N \frac{1}{y_j-x} w_j}, \quad x \in \Omega^c, x \neq y_i, i = 1, \dots, N.$$

This is marginally simpler than our (2.19) since the interior point a is not needed; we have not compared the two methods numerically, since our scheme performs so well.

2.3.3 First derivative and its barycentric form, interior case

Accurate Stokes evaluation demands accurate first derivatives of Laplace potentials and hence of the Cauchy representation. The interior Cauchy formula for the

first derivative is,

$$v'(x) = \frac{1}{2\pi i} \int_{\Gamma} \frac{v^-(y)}{(y-x)^2} dy, \quad x \in \Omega .$$

We can combine this with the Cauchy formula as in (2.15) to get,

$$\int_{\Gamma} \frac{v^-(y) - v(x) - (y-x)v'(x)}{(y-x)^2} dy = 0 \quad x \in \Omega , \quad (2.20)$$

which holds because the middle term vanishes (the contour integral of $1/(x-y)^n$ is zero for integer $n \neq 1$). The integrand is analytic and smooth even as x approaches Γ because the numerator kills Taylor terms zero and one in the expansion of v about x , so the trapezoid rule (2.13), as before, is accurate. Applying the quadrature (making sure to keep the middle term, which is mathematically zero but numerically necessary to compensate the quadrature) gives

$$v'(x) \approx \begin{cases} \frac{\sum_{j=1}^N \frac{v_j^- - v(x)}{(y_j - x)^2} w_j}{\sum_{j=1}^N \frac{1}{y_j - x} w_j}, & x \in \bar{\Omega}, x \neq y_i, i = 1, \dots, N \\ -\frac{1}{w_i} \sum_{j \neq i} \frac{v_j^- - v_i^-}{y_j - y_i} w_j, & x = y_i \end{cases} \quad (2.21)$$

Here the case where x coincides with a node is derived by taking the limit as x approaches a node. (2.21) is analogous to the derivative of the barycentric interpolant derived by Schneider–Werner [75, Prop. 11] (who also generalized to higher derivatives). The case for x not a node is equivalent to the derivative formula of Ioakimidis et al. [37, (2.9)], and that of [5, Algorithm P’].

However, (2.21) (and its equivalent forms cited above) suffers from catastrophic cancellation as x approaches a node, a point that we have not seen discussed in the literature. Even if $v(x)$ is computed to high accuracy (say, using (2.16)), the *relative* accuracy of the term $(v_j^- - v(x))$ in (2.21) deteriorates, in a way that involves no explicit cancellation of poles as in the second barycentric form. A solution which

regains true barycentric stability is to evaluate this term via

$$v_j^- - v(x) \approx \frac{\sum_{k \neq j} \frac{v_j^- - v_k^-}{y_k - x} w_k}{\sum_{k=1}^N \frac{1}{y_k - x} w_k} . \quad (2.22)$$

Because the numerator term $k = j$ is absent, then as x tends to node y_j an overall factor of $(y_j - x)$ dominates in a way that cancels (to high accuracy) one power in the term $(y_j - x)^2$ in (2.21). Thus using (2.22) for each term $v_j^- - v(x)$ is a true second barycentric form, which we believe is new. The problem is that it increases the effort from $\mathcal{O}(N)$ to $\mathcal{O}(N^2)$ per target point x . Our remedy is to realize that (2.22) is only helpful for small distances $|y_j - x|$: thus we use the value $v(x)$ from (2.16) unless $|y_j - x| < \delta$, in which case (2.22) is used. We choose $\delta = 10^{-2}$, since in most settings with Ω of size $\mathcal{O}(1)$ only a small fraction of targets lie closer than this to a node, and at most around 2 digits are lost due to the loss of barycentric stability for larger target-node distances. Unless many nodes are spaced much closer than δ , the method remains $\mathcal{O}(N)$ per target point.

Remark 3. It has been pointed out² to us that another way to alleviate the issue of the first formula in (2.21) not itself being barycentric is to instead compute $v'(y_i)$ at each of the nodes using the second formula in (2.21), then to use barycentric interpolation (2.16) from these values. I.e. one *interpolates the derivative instead of differentiating the interpolant*. We postpone comparing these two methods to future work, since we note that we already achieve close to machine precision errors, uniformly.

2.3.4 First derivative with a barycentric form, exterior case

Combining the ideas of (2.18) and (2.20) we have the identity

$$\int_{\Gamma} \frac{v^+(y) - v(x) - (x - a)(y - a)^{-1}(y - x)v'(x)}{(y - x)^2} dy = 0 , \quad \text{for } x \in \Omega^c ,$$

²L. N. Trefethen, personal communication.

which, as with (2.20), has smooth analytic integrand even as x approaches Γ because the first two Taylor terms are cancelled. Inserting the quadrature rule gives the approximation

$$v'(x) \approx \frac{1}{x-a} \cdot \frac{\sum_{j=1}^N \frac{v_j^+ - v(x)}{(y_j - x)^2} y_j' w_j}{\sum_{j=1}^N \frac{(y_j - a)^{-1}}{y_j - x} y_j' w_j}, \quad x \in \mathbb{R}^2 \setminus \Omega, \quad x \neq y_i, \quad i = 1, \dots, N \quad (2.23)$$

which, as in the interior case, does not have barycentric stability as x approaches a node. Unfortunately the formula analogous to (2.22) which uses (2.19) to write $v(x)$ also fails to give stability, because the $k = j$ term no longer vanishes and roundoff in this term dominates. However, it is easy to check that the following form is mathematically equivalent,

$$v_j^+ - v(x) \approx \frac{1}{x-a} \left[\frac{\sum_{k \neq j} \frac{v_j^+ (y_j - a)(y_k - a)^{-1} - v_k^+}{y_k - x} w_k}{\sum_{k=1}^N \frac{(y_k - a)^{-1}}{y_k - x} w_k} - (y_j - x) v_j^+ \right], \quad (2.24)$$

and does give barycentric stability when inserted into (2.23), because, as x approaches y_j , the factor $y_j - x$ in both terms is explicit. As with the interior case, we only use this when $|y_j - x| < \delta$. This completes our recipes for interior/exterior Cauchy values and first derivatives.

2.3.5 Numerical tests of values and derivatives close to the curve

We test the Cauchy integral evaluation formulae presented in this section on the smooth star-shaped domain of Fig. 2.2(a) given by the radial function $f(\theta) = 1 + 0.3 \cos(5\theta)$. (This shape is also used in [31].) For the interior case, the test points x lie along the line shown in Fig. 2.2(a), at a set of distances from a node y_j logarithmically spanning the range from machine precision to 1. A test point $x = y_j$ is also included (data for this test point appears in the left-most column of each plot (b)–(d)). The function v is a pole located outside of Γ and is therefore holomorphic in Ω ; its maximum

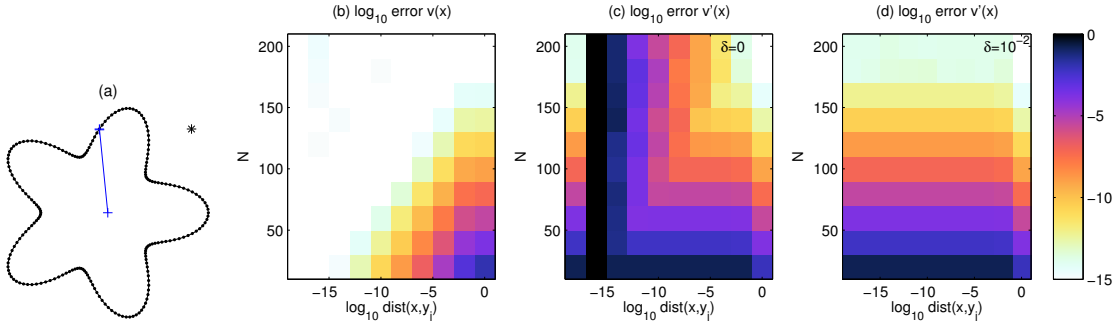


Figure 2.2: (a) Curve (with $N = 200$ nodes shown) for the test of the interior Cauchy integral evaluation; points x (shown by $+$ symbols along the straight line) lie at approximate distances from a node $0, 10^{-16}, 10^{-14}, \dots, 10^{-2}, 1$. The test function is $v(x) = 1/(x - b)$ where the pole $b = 1.1 + 1i$ (shown by $*$) is a distance of 0.5 from Γ . (b) Convergence of \log_{10} of the absolute error (see colorbar at right) for $v(x)$, using formula (2.16), plotted vs distance of x from a node (horizontal axis) and number of nodes N (vertical axis). (c) Convergence of $v'(x)$, using the formula (2.21). (d) Convergence of $v'(x)$, inserting (2.22) into (2.21) for distances below $\delta = 10^{-2}$. Exterior results are very similar, so we do not show them.

magnitude on Γ is of order 1.

Fig. 2.2(b) shows exponential convergence with 15 digit accuracy in value reached by $N = 180$ at all distances, and that small distances converge at the same rate but with a smaller prefactor. For comparison, applying the quadrature scheme (2.13) directly to the Cauchy integral, as is common practice, with the same $N = 180$, gives 15 digits of accuracy at the most distant point, but zero digits of accuracy at all other points.

The derivative formula (2.21) of Ioakimidis et al. is tested in plot (c): there is a clear loss of accuracy in inverse proportion to the distance from a node, regardless of N , simply due to non-barycentric loss of relative error in the term $v_j^- - v(x)$. (However, when x coincides with a node, convergence is again exponential). Finally, evaluating $v_j^- - v(x)$ by (2.22) for distances less than $\delta = 10^{-2}$ gives plot (d), which achieves 14 digit accuracy at all distances.

We also tested the exterior methods, choosing a generic interior point $a = -0.1$ in the method, and using v given by a pole at $b = 0.1 + 0.5i$ a distance 0.33 from Γ .

This v is holomorphic in Ω^c and has $v_\infty = 0$. Results for the value formula (2.19), and derivative formula (2.23) with $v_j^+ - v(x)$ evaluated via (2.24), are essentially identical to the interior case, with equally good achievable accuracies, so we do not show them.

Note that the *rate* of exponential convergence is clearly affected by the choice of holomorphic test function v : moving the pole of v closer to Γ worsens the rate since the data becomes less smooth. We chose poles not too far from Γ . For comparison, for an entire function, such as $v(x) = e^{2x}$, full convergence in the interior is achieved at only $N = 80$.

2.4 Evaluation of Laplace layer potentials with global quadrature

The goal of this section is to describe accurate methods to evaluate the single- and double-layer potential in the interior and exterior of a closed curve Γ , given only the density values $\tau_j := \tau(y_j)$ at the nodes y_j belonging to a global quadrature (2.13) on the curve. We remind the reader that direct application of the rule (2.13) to the layer potentials (2.2) and (2.3) is highly inaccurate near Γ . In contrast, our methods retain accuracy and efficiency for targets x arbitrarily close to Γ . The effort will be $\mathcal{O}(N(N + M))$ for N nodes and M targets (although see discussion in Sec. 2.6).

2.4.1 Laplace double-layer potential

Recall that the double-layer potential (2.3) can be written as the real part of the function v given by the Cauchy integral (2.4). v is holomorphic in Ω and in Ω^c . For interior evaluation, Helsing–Ojala [31] proposed a two-stage scheme, which for convenience we review in our setting of the global periodic trapezoid quadrature:

Step 1. Approximate the boundary data $v_j^- := v^-(y_j)$ which is the interior limit of the function (2.4) at each of the nodes.

Step 2. Use this data to numerically approximate the Cauchy integral (2.14) to generate $v(x)$ at any $x \in \Omega$, using the method of Section 2.3.1.

Finally, taking $u = \operatorname{Re} v$ extracts the desired potential. We extend this to first derivatives by including in Step 2 the method of Section 2.3.3 to evaluate $v'(x)$, then extracting the gradient as $\nabla u = (\operatorname{Re} v', -\operatorname{Im} v')$.

It only remains to present Step 1 (following [31, Sec. 3]). For $x \in \Gamma$, let $v^-(x) := \lim_{y \rightarrow x, y \in \Omega} v(y)$ be the interior limit of (2.4). It follows from the Sokhotski–Plemelj jump relation [44, Thm. 7.6] that

$$v^-(x) = -\frac{1}{2}\tau(x) - \frac{1}{2\pi i} \operatorname{PV} \int_{\Gamma} \frac{\tau(y)}{y-x} dy, \quad x \in \Gamma,$$

where PV indicates the principal value integral. We split the PV integral into an analytic and Cauchy part:

$$\operatorname{PV} \int_{\Gamma} \frac{\tau(y)}{y-x} dy = \int_{\Gamma} \frac{\tau(y) - \tau(x)}{y-x} dy + \tau(x) \operatorname{PV} \int_{\Gamma} \frac{1}{y-x} dy.$$

The latter integral is $-\frac{1}{2}$ for $x \in \Gamma$ for any closed curve Γ . Thus

$$v^-(x) = -\tau(x) - \frac{1}{2\pi i} \int_{\Gamma} \frac{\tau(y) - \tau(x)}{y-x} dy, \quad x \in \Gamma \quad (2.25)$$

The integrand is analytic (compare (2.15)), so the periodic trapezoid rule is again excellent. We need the boundary point $x = y_k$, i.e. the k th node, so must use the correct limit of the integrand at the diagonal $y = x$. If we define $\tilde{\tau}(t) := \tau(Z(t))$ as the density in the parameter variable, and let $\tau'_k := \tilde{\tau}'(s_k)$, then the integrand has diagonal limit $\tilde{\tau}'(s_k)/|Z'(s_k)|$, and applying quadrature (2.13) to (2.25) gives

$$v_k^- = -\tau_k - \frac{1}{2\pi i} \sum_{j \neq k} \frac{\tau_j - \tau_k}{y_j - y_k} w_j - \frac{\tau'_k}{iN}, \quad k = 1, \dots, N. \quad (2.26)$$

We compute the vector $\{\tau'_j\}_{j=1}^N$ by spectral differentiation [83] of the vector $\{\tau_j\}_{j=1}^N$ via the N -point fast Fourier transform (FFT). This completes Step 1 for the interior case.

The case of x exterior to Γ is almost identical, except that by the Sokhotski–Plemelj jump relation we change Step 1 to

$$v_k^+ = v_k^- + \tau_k = -\frac{1}{2\pi i} \sum_{j \neq k} \frac{\tau_j - \tau_k}{y_j - y_k} w_j - \frac{\tau'_k}{iN}, \quad k = 1, \dots, N, \quad (2.27)$$

and in Step 2 we now use the exterior methods of Sections 2.3.2 and 2.3.4 for $v(x)$ and $v'(x)$.

2.4.2 Laplace single-layer potential

This section is the heart of the contribution of this paper. Although we have not seen this in the literature (other than [9]), it is also possible to write the single-layer potential (2.2) as the real part of a holomorphic function,

$$u(x) = (\mathcal{S}\tau)(x) = \operatorname{Re} v(x) \quad (2.28)$$

where the function v is defined by

$$v(x) := \frac{1}{2\pi} \int_{\Gamma} \left(\log \frac{1}{y-x} \right) \tau(y) |dy|, \quad x \in \mathbb{R}^2 \setminus \Gamma. \quad (2.29)$$

Note that the integration element $|dy| = dy/in_y$ is real. We present the interior and exterior cases in turn.

2.4.2.1 Interior case

We first explain the (simpler) interior case $x \in \Omega$. In order that v be holomorphic in Ω , care needs to be taken with the branch cuts of the logarithm in (2.29), and

it must be considered as a function of two variables, $L(y, x) := \log 1/(y - x)$, which agrees with the standard logarithm up to the choice of Riemann sheet. Fixing a boundary point $y_0 \in \Gamma$, the following branch choices for L are sufficient for v to be holomorphic [9, Remark 7]: i) for each fixed $x \in \Omega$, as y loops around Γ a jump of $2\pi i$ occurs in L only at y_0 , and ii) for each fixed $y \in \Gamma$, $L(y, \cdot)$ is continuous in Ω .

As before we have two steps, only the first of which differs from the double-layer case: Step 1 finds the interior boundary data v_j^- for the function (2.29), then Step 2 evaluates v at arbitrary interior target points using (2.14) and the Cauchy integral method of Section 2.3.1. Finally the real part is taken. Accurate first partials are also found in the same way as the double-layer case.

All that remains is to explain Step 1. Inserting the parametrization of Γ from Section 2.3 and recalling $\tilde{\tau}(s) := \tau(Z(s))$, the interior boundary data of v at $x = Z(t) \in \Gamma$ is

$$v^-(x) := \lim_{\Omega \ni z \rightarrow x} \frac{1}{2\pi} \int_0^{2\pi} \left(\log \frac{1}{Z(s) - z} \right) \tilde{\tau}(s) |Z'(s)| ds . \quad (2.30)$$

To handle the logarithmically singular kernel which results in the limit, we exploit the identity

$$\log \frac{1}{Z(s) - Z(t)} = \log \frac{e^{is} - e^{it}}{Z(s) - Z(t)} - \log(e^{is} - e^{it}) . \quad (2.31)$$

For Γ smooth, the first term on the right-hand side is smooth as a function of s when the correct branch cuts of \log are taken, while the second term (relating to the single-layer kernel on the unit disc) can be handled analytically. Similar ideas are used in analytic PDE theory [82, Sec. 8]. Inserting this into (2.30), realizing that the sense

of the limit (interior) only affects the second term, gives

$$\begin{aligned}
v^-(Z(t)) &:= \frac{1}{2\pi} \int_0^{2\pi} \left(\log \frac{e^{is} - e^{it}}{Z(s) - Z(t)} \right) \tilde{\tau}(s) |Z'(s)| ds \\
&\quad - \frac{1}{2\pi} \int_0^{2\pi} \log(e^{is} - e^{i(t+i0)}) \tilde{\tau}(s) |Z'(s)| ds
\end{aligned} \tag{2.32}$$

where $t + i0$ indicates the limit where the imaginary part of t approaches zero from above, i.e. e^{it} approaches the unit circle from inside. We write $\log(e^{is} - e^{i(t+i0)}) = is + g(s - t)$ where g is a convolution kernel. We simply drop the non-convolutional term is because it contributes a purely imaginary constant to v that will have no effect on $\text{Re } v$. The sense of the limit allows a Taylor expansion of the logarithm in the kernel,

$$g(s) = \log(1 - e^{-i(s-i0)}) = \sum_{n>0} \frac{e^{-ins}}{n} =: \sum_{n \in \mathbb{Z}} g_n e^{ins} , \tag{2.33}$$

where the last term defines a Fourier series whose coefficients are read off as

$$g_n = \begin{cases} -\frac{1}{n}, & n < 0 \\ 0, & n \geq 0 \end{cases} \tag{2.34}$$

These coefficients can be used to construct weights that approximate the product quadrature with g , via trigonometric polynomials. Let any function f be smooth and 2π -periodic, then

$$\int_0^{2\pi} f(s) g(s) ds \approx \sum_{j=1}^N R_j f(s_j)$$

holds to spectral accuracy, where the general formula³ for the weights (as derived in

³We drop the factor of $\frac{1}{2}$ from the last term present in [43] and [27, Sec. 6], it being exponentially small.

the review [27, Sec. 6]), followed by the weights for our particular case (2.33) of g , is

$$R_j = \frac{2\pi}{N} \sum_{|n| < N/2} \bar{g}_n e^{-ins_j} = -\frac{2\pi}{N} \sum_{n=-N}^{-1} \frac{e^{-ins_j}}{n}, \quad j = 1, \dots, N. \quad (2.35)$$

The action of g as a convolution kernel on any f is thus approximated by a circulant matrix:

$$\int_0^{2\pi} f(s)g(s - s_k)ds \approx \sum_{j=1}^N R_{j-k}f(s_j), \quad k = 1, \dots, N.$$

We are now ready to apply quadrature rules to (2.32). For each boundary target $x = y_k = Z(s_k)$, the first (smooth) term is well approximated using the periodic trapezoid rule with the correct diagonal limit, while the second term uses the above product quadrature, giving

$$\begin{aligned} v_k^- \approx & \frac{1}{2\pi} \sum_{j \neq k} \left(\log \frac{e^{i(s_k - s_j)}}{Z(s_k) - Z(s_j)} \right) w_j \tau_j + \frac{1}{2\pi} \log \frac{ie^{is_k}}{Z'(s_k)} w_k \tau_k \\ & - \frac{1}{2\pi} \sum_{j=1}^N R_{j-k} w_j \tau_j, \quad k = 1, \dots, N. \end{aligned} \quad (2.36)$$

With (2.36) and (2.35) defined, Step 1 is complete.

A couple of practical words are needed. The vector $\{R_j\}_{j=1}^N$ is simply filled by taking the FFT of (2.34). Handling the branch cuts of the first term in (2.36) so that it corresponds to a smooth kernel may seem daunting. In fact this is done easily by filling the N -by- N matrix \mathbf{S} corresponding to the first two terms in (2.36), using the machine's standard branch cut for \log , then applying to it the following simple MATLAB code,

```
for i=1:numel(S)-1
    p = imag(S(i+1)-S(i));
    S(i+1) = S(i+1) - 2i*pi*round(p/(2*pi));
end
```

This loops through all matrix elements and adjusts them by an integer multiple of $2\pi i$ whenever they jump by more than π in imaginary part from a neighboring element. Once N is sufficiently large to resolve the kernel, such large jumps cannot occur unless a branch is being crossed.

2.4.2.2 Exterior case

Here we describe the differences from the interior case. In Step 1, the crucial sign change causes the convolutional part of the kernel to reverse direction as follows,

$$\log(e^{is} - e^{i(t-i0)}) = i\pi + \log(e^{it} - e^{i(s+i0)}) = i(\pi + t) + g(t - s) ,$$

where g is as in (2.33). Thus the exterior version of (2.32) is

$$\begin{aligned} v^+(Z(t)) := & \frac{1}{2\pi} \int_0^{2\pi} \left(\log \frac{e^{is} - e^{it}}{Z(s) - Z(t)} \right) \tilde{\tau}(s) |Z'(s)| ds + \frac{T}{2\pi i} (\pi + t) \\ & - \frac{1}{2\pi} \int_0^{2\pi} g(t - s) \tilde{\tau}(s) |Z'(s)| ds \end{aligned} \quad (2.37)$$

where the *total charge* of the single-layer density τ is

$$T := \int_{\Gamma} \tau(y) |dy| = \int_0^{2\pi} \tilde{\tau}(s) |Z'(s)| ds . \quad (2.38)$$

The first (smooth) term in (2.37) is identical to that in (2.32). The π in the middle term can be dropped since it has no effect on $\text{Re } v$. In the last term, the convolution kernel is $g(-s)$ instead of $g(s)$; we can achieve this by replacing R_j with R_{-j} . Thus

the exterior version of (2.36) is

$$v_k^+ \approx \frac{1}{2\pi} \sum_{j \neq k} \left(\log \frac{e^{i(s_k - s_j)}}{Z(s_k) - Z(s_j)} \right) w_j \tau_j + \frac{1}{2\pi} \log \frac{ie^{is_k}}{Z'(s_k)} w_k \tau_k + \frac{T}{2\pi i} t_k - \frac{1}{2\pi} \sum_{j=1}^N R_{k-j} w_j \tau_j. \quad (2.39)$$

This completes Step 1 for the Laplace single-layer exterior case.

Remark 4. Kress [43] gives formulae (due to Martensen–Kussmaul) for splitting a periodic kernel with a logarithmic singularity into a smooth part and the product of a smooth part and the convolution kernel $\log(4 \sin^2 \frac{s}{2})$. The formula $g(s) = \log(1 - e^{-i(s-i0)}) = \frac{1}{2} \log(4 \sin^2 \frac{s}{2}) + i \frac{s-\pi}{2}$, for $0 \leq s < 2\pi$, shows that (2.31) is the analogous Kress-type split for the *complex* logarithmic kernel case. The imaginary part of g is a “sawtooth wave” (periodized linear function) whose sign depends on from which side the limit is taken.

If the total charge $T = 0$ then v given by (2.29) is single-valued in Ω^c and has $v_\infty = 0$, so Step 2 may proceed just as for the double-layer case, and we are done. However, if $T \neq 0$, then there must be a branch cut in v connecting Γ to ∞ (along which the imaginary part must jump by T), and v grows logarithmically at ∞ . In this latter case we must subtract off the total charge as follows. Let $a \in \Omega$ be chosen not close to Γ , then define

$$w(x) := v(x) - \frac{T}{2\pi} \log \frac{1}{a-x}, \quad (2.40)$$

where v is as in (2.29). Then w , being holomorphic in Ω^c (the branch cut of \log in (2.40) can be chosen to cancel that of v), and having $w_\infty = 0$, is appropriate for representation by Step 2, using its boundary data $w_k^+ = v_k^+ - (T/2\pi) \log 1/(a - y_k)$, for $k = 1, \dots, N$. Finally, after Step 2 produces w and w' , the missing monopole must be added back in to get v via (2.40), and $v'(x) = w'(x) + (T/2\pi)/(a - x)$. Then as before, $u = \operatorname{Re} v$ and $\nabla u = (\operatorname{Re} v', -\operatorname{Im} v')$.

Remark 5. The reader may wonder how spectral accuracy is to be achieved in Step 2 using the periodic trapezoid rule when $T \neq 0$, given that then the boundary data (2.39) has a *discontinuity* due to the imaginary sawtooth (third term). In fact there is also a discontinuity of equal size and opposite sign, in w , introduced by the branch cut of the log in (2.40). Moreover, these discontinuities may occur at different boundary locations; but, since the number of nodes lying between them is fixed, the error introduced is a purely imaginary constant and has no effect on u . The overall scheme for evaluating u and ∇u , as for the interior case, is spectrally accurate.

2.4.3 Numerical tests of Laplace evaluation quadratures

In order to test the new global quadrature evaluation schemes at a variety of distances from the boundary, we set up simple BVPs on a smooth closed curve with boundary data corresponding to known Laplace solutions. We then solve each using an integral equation formulation and record the error between the numerical layer potential evaluation and the known solution. Introducing and solving the BVP is necessary since on a general curve there are very few known density functions which generate known analytic potentials (specifically, $\tau \equiv 1$ generating $u \equiv -1$ via the DLP is the only example known to the authors).

The Laplace BVPs we use for tests are the standard four possibilities of interior/exterior, Dirichlet/Neumann problems [44, Sec. 6.2]. The integral equation representations are those of [44, Sec. 6.4] with the exception of the exterior Dirichlet, which we do not modify since we choose the solution to vanish at ∞ . We use the representation $u = \mathcal{D}\tau$ for the Dirichlet BVP, and $u = \mathcal{S}\tau$ for the Neumann BVP, with boundary data f . Using the operator D and the jump relations (2.5)–(2.6), the

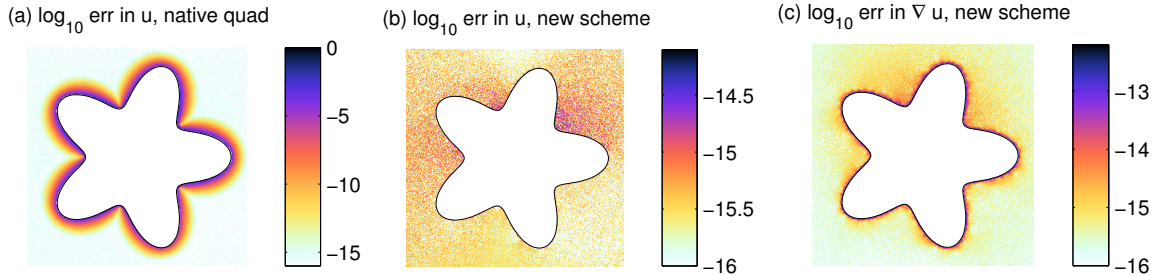


Figure 2.3: Evaluation error for the Laplace single-layer potential on a grid exterior to the closed curve Γ defined by the radial function $r(\theta) = 1 + 0.3 \cos 5\theta$. The color shows \log_{10} of the errors relative to a known solution to the exterior Neumann BVP. (a) uses the native periodic trapezoid rule (2.13), whereas (b) and (c) use the scheme of Section 2.4.2.2 (note the change in color scale). In all cases $N = 240$ nodes are used for solution and evaluation.

integral equations are thus,

$$\text{Dirichlet BVP : } (D \pm \frac{1}{2})\tau = f , \quad (+ \text{ exterior case, } - \text{ interior case}) \quad (2.41)$$

$$\text{Neumann BVP : } (D^T \mp \frac{1}{2})\tau = f , \quad (- \text{ exterior case, } + \text{ interior case}) \quad (2.42)$$

Since the kernel of D is smooth, we fill the system matrix via the Nyström method [44, Sec. 12.2] using the diagonal values $\lim_{s \rightarrow t} D(s, t) = -\kappa(t)/4\pi$ where $\kappa(t)$ is the curvature of Γ at $Z(t)$. The interior Neumann solution is only defined up to a constant, hence we fix this constant by defining the error at the origin to be zero. For a solution in Ω we use $u(x) = \text{Re } e^{i(1+x)}$, $x \in \mathbb{C}$, and in Ω^c we use $u(x) = \text{Re } 1/(x - 0.1 - 0.3i)$, which has $u_\infty = 0$. For the exterior Dirichlet and interior Neumann cases the operator, and hence Nyström matrix, has nullity 1; however, this does not pose a problem when a backward-stable dense linear solver is used (we use the backslash command in MATLAB).

Figure 2.3 shows results for the exterior single-layer potential on the curve used in Sec. 2.3.5, for the converged value of $N = 240$. As described in the introduction, we see that with the native quadrature scheme errors grow exponentially up to $\mathcal{O}(1)$ near Γ . In contrast, the new scheme of this section achieves 14 digits in value, and 12

N	DLP int		DLP ext		SLP int		SLP ext	
	u	∇u	u	∇u	u	∇u	u	∇u
100	2.9e-07	9.6e-06	8e-05	2.6e-03	7e-09	2.7e-07	1e-06	3.9e-05
150	7.8e-11	3.8e-09	6.7e-10	6.8e-08	1.4e-12	8.7e-11	7.9e-10	7.5e-08
200	2.1e-14	2e-12	2.6e-13	3.4e-11	9.8e-15	7e-13	2.7e-13	3.6e-11
250	2e-14	1.7e-12	4.7e-14	4.6e-12	5.9e-14	4.5e-12	4.9e-15	6.3e-13

Table 2.1: Convergence of error for the Laplace layer potential evaluation scheme of Section 2.4 for BVPs solved on the curve shown in Figure 2.3. The maximum error in u or ∇u is taken over a square grid of spacing 0.01.

digits in first derivative, uniformly for points arbitrarily close to Γ . The results for the other three BVPs are very similar. For the convergence in all four cases we refer to Table 2.1 which shows the worst-case error over target points lying on a square grid covering $[-1.5, 1.5]^2$ with grid spacing 0.01 (some of these target points lie exactly on nodes; the closest other ones are distance 3×10^{-4} from a node). High-order convergence is apparent from this table, with the interior schemes converging slightly faster in N than the exterior ones.

Remark 6. Our test curve is chosen to be the same as the interior Dirichlet BVP test in [31, Fig. 2–4], enabling a comparison of our scheme against their panel-based version. When we use data from their solution, $u(x) = \text{Re } 1/(z - 1.5 - 1.5i) + 1/(z + 0.25 - 1.5i) + 1/(z + .5 + 1.5i)$, we achieve uniform 14-digit accuracy by $N = 320$; from [31, Fig. 4] the panel-based version requires $N = 480$. So the periodic trapezoid rule is a factor 1.5 times more efficient in terms of unknowns, which is close to the expected factor $\pi/2$ [26].

We use MATLAB (R2012a) for this implementation (and others in this paper), and achieve around 10^7 source-target pairs per second on a 2.6 GHz i7 laptop.

2.5 Stokes close evaluation scheme and numerical results

We are at last in a position to describe how we evaluate Stokes potentials, given samples $\{\sigma_j\}_{j=1}^N$ of the vector density σ at N trapezoid rule nodes $\{y_j\}_{j=1}^N$ defining a

closed curve Γ . Throughout we generate all other geometric data (normals, curvature, etc) at the nodes using spectral differentiation via the N -point FFT.

For the SLP we use (3.16), where for each of the three Laplace SLPs we use the samples of the density ($y \cdot \sigma$, σ_1 or σ_2) at the same N nodes, fed into the new Laplace evaluation scheme of Sec. 2.4.2 (which itself relies on Sec. 2.3 for Step 2). For the DLP we similarly use (3.18), with the last three terms using samples at the given N nodes fed into the (Ioakimidis/Helsing) scheme of Sec. 2.4.1. However, for the first term in (3.18), we need the complex-valued densities τ_1 and τ_2 in (2.11); notice that the scheme of Sec. 2.4.1 works perfectly well when fed a complex τ , producing (2.4) rather than (2.3). We have found that this first term converges slower than the others,⁴ so to preserve overall accuracy we use FFT interpolation to upsample σ by a factor $\beta > 1$ when computing τ_1 and τ_2 . Thus this first term (and its two Laplace DLP evaluations) are done with densities sampled on βN nodes. We have found that $\beta = 2.2$ is sufficient to recover similar accuracy to the other terms.⁵ This increases the effective cost of the Stokes DLP from 5 to about 7 Laplace DLPs.

We now test the performance of our Stokes close evaluation scheme in settings relevant for vesicle simulations. We consider four examples: the first two study the effect of proximity of the target and complexity of the geometry on the error. In the last two, similar to the Laplace case, we set up BVPs with boundary data corresponding to known analytic solutions and compare against numerical solutions obtained through integral equation solves. The fourth example is special in that we also use the evaluation scheme to *apply* the Nyström matrix in the solve, given a boundary with close-to-touching components.

Example 1. Consider two identical bubbles separated by a small distance δ as shown in Fig. 2.4(a). Assuming unit surface tension, the interfacial force f on

⁴We suspect this is because the two appearances of the normal function in (2.11) cause a more oscillatory integrand.

⁵We note that panel-based schemes also need upsampling from 16 to 32 nodes per panel for full accuracy [31].

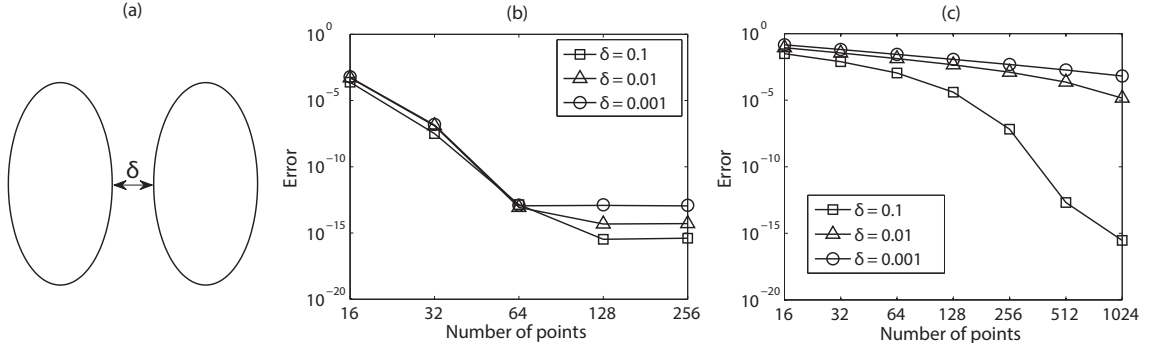


Figure 2.4: Performance of proposed globally compensated Stokes SLP quadrature in evaluating the hydrodynamic interaction force (2.43) between two elliptical bubbles parametrized as translations of $(\cos \theta, 2 \sin \theta)$. δ is the minimum distance between the two interfaces; we plot the l^∞ -norm error against the number of nodes N for $\delta = 0.1$, 0.01 and 0.001. (b) Convergence of new globally compensated scheme. (c) Native quadrature (trapezoidal rule), showing very poor convergence for small δ values.

each bubble is given by $f(y) = \kappa(y)n_y$ where κ is the curvature and n_y is the unit outward normal at a point y on the interface Γ . The hydrodynamic interaction force F experienced at target x on one bubble due to the presence of the other is simply the Stokes SLP with interfacial force as the density, that is,

$$F(x) = (\mathbf{S}f)(x) = \frac{1}{4\pi} \int_{\Gamma} \left(\log \frac{1}{\rho} I + \frac{r \otimes r}{\rho^2} \right) \kappa n_y ds_y . \quad (2.43)$$

If δ is large, the integrand in (2.43) is smooth and the native trapezoidal rule will yield superalgebraic convergence. When δ is small, this convergence rate is proportional to δ , due to the nearly singular integral; hence the N required scales like $1/\delta$, and is unacceptably large as shown in Fig. 2.4(c). Yet our new scheme achieves 13 digits with only $N = 64$ nodes, as shown in Fig. 2.4(b).

Example 2. We consider elliptical bubbles of aspect ratios 2, 4, and 8, and evaluate $(\mathbf{S}f)(x)$ for a target point x a distance $\delta = 10^{-3}$ from the highest-curvature point of the ellipse. This can be interpreted as the disturbance velocity at x induced by the nearby bubble. Fig. 2.5(b) shows the rapid exponential convergence of our SLP scheme. For aspect ratio 2, by $N = 128$ it has converged to 13 digits, and for

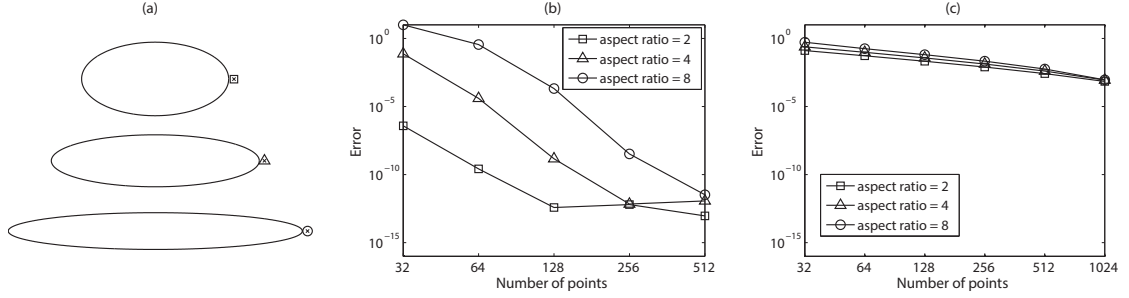


Figure 2.5: Absolute errors in computing the Stokes SLP at a fixed distance $\delta = 0.001$ away from the boundary of three different geometries with varying complexity: ellipses with aspect ratios 2, 4, and 8. (a) Shows ellipses and target point x (case $\delta = 0.1$ is shown to make the separation visible). (b) Results for our proposed scheme. In all cases, the errors decay exponentially with discretization size although, as expected, the absolute errors are higher for higher aspect ratio geometries. Therefore, in the case of our scheme, the given data (boundary, density) dictates the overall accuracy. (c) Results for native quadrature. In this case, convergence rate decreases but more striking is the fact the errors are almost the same for a particular number of points. This clearly implies that the near singular integral evaluation dominates the overall error.

higher aspect ratios the N required grows in proportion to the aspect ratio. This is as expected, since the density data (κn_y) changes more rapidly at the extreme point of the ellipse as its aspect ratio grows, thus requires a larger N to accurately interpolate. Thus the errors are limited by resolving the data, not by the close-evaluation scheme. In contrast, errors using the native scheme are unacceptable, and dominated by the small δ .

N	ext Dirichlet	int Dirichlet	ext Neumann	int Neumann
100	2.3e-05	3.6e-05	2.7e-04	1.8e-04
150	2.3e-07	9.9e-08	6.1e-08	1.4e-08
200	5.0e-09	1.9e-09	3.7e-12	4.0e-11
250	2.8e-10	1.4e-11	3.6e-13	1.2e-12
300	2.3e-11	8.3e-14	6.7e-13	2.0e-13
350	2.0e-12	4.9e-14	4.3e-13	1.1e-13

Table 2.2: Convergence of velocity error (max over the evaluation grid) for the solutions of the four types of Stokes BVPs, on the curve shown in Fig. 2.6. The Nyström method is used to find the density, then the new close evaluation scheme is used, as described in Sec. 2.5.

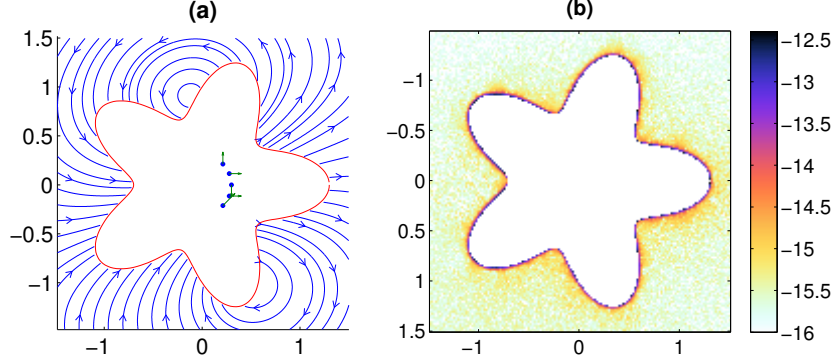


Figure 2.6: (a) Streamlines of the velocity field generated by Stokeslets shown in the interior, for testing the exterior BVPs in Example 3 of Sec. 2.5. The boundary Γ is given by the polar function $f(\theta) = 1 + 0.3 \cos 5\theta$. (b) \log_{10} of absolute error in computing the velocity (on a grid of spacing 0.02) for the exterior Neumann Stokes BVP, using $N = 250$ discretization points on the boundary for both the Nyström solve and the evaluation as in Example 3 of Sec. 2.5.

Example 3. We solve all four types (interior/exterior, Dirichlet/Neumann) of Stokes BVP on the star-shaped geometry introduced in Section 2.3.5 using a standard Nyström method, and test the velocity field evaluation error with the new scheme. Denoting the pressure by p and the boundary vector data by g , the (unit viscosity) Stokes BVPs are [34, Sec. 2.3.2]

$$-\Delta u + \nabla p = 0 \quad \text{in } \Omega^c \text{ (exterior) ,} \quad \text{or in } \Omega \text{ (interior) ,} \quad (2.44)$$

$$\nabla \cdot u = 0 \quad \text{in } \Omega^c \text{ (exterior) ,} \quad \text{or in } \Omega \text{ (interior) ,} \quad (2.45)$$

$$u = g \text{ on } \Gamma \quad \text{(Dirichlet),} \quad \text{or } T(u, p) = g \text{ on } \Gamma \text{ (Neumann),} \quad (2.46)$$

$$u(x) = \Sigma \log \rho + \mathcal{O}(1) \quad \text{as } \rho := |x| \rightarrow \infty \quad \text{(exterior only) ,} \quad (2.47)$$

where the traction at $y \in \Gamma$ is defined by $T(u, p) := -pn_y + (\nabla u + (\nabla u)^T) \otimes n_y$, and Σ is some constant vector. For the Dirichlet case g is the boundary velocity, for the Neumann case the boundary traction.

We use standard representations as follows. Let \mathbb{S} and \mathbb{D} denote the boundary integral operators with the Stokes SLP and DLP kernels, with \mathbb{D} taken in the principal

value sense (these are analogous to S and D in the Laplace case). Then for the Neumann BVPs we use $u = \mathbf{S}\sigma$, giving the integral equation $(\mathbb{D}^T \mp \frac{1}{2})\sigma = g$ (– exterior case, + interior case), just as in (2.42). For the interior Dirichlet BVP we use $u = \mathbf{D}\sigma$, giving $(\mathbb{D} - \frac{1}{2})\sigma = g$. For the exterior Dirichlet, since we wish to solve data g for which $\Sigma \neq \mathbf{0}$, we use the combined representation $u = (\mathbf{D} + \mathbf{S})\sigma$ [60, page 128], giving $(\mathbb{D} + \mathbb{S} + \frac{1}{2})\sigma = g$. This also eliminates the 1-dimensional nullspace in this case.⁶ All four integral equations are consistent, but the interior Dirichlet and exterior Neumann have 1-dimensional nullspaces which only affect the pressure solution p , and the interior Neumann integral equation has a 3-dimensional nullspace corresponding to rigid motions in the plane [34, Sec. 2.3.2]. In the last case we match these three components to those of the reference solution before computing the errors.

We use the boundary Γ shown in Fig. 2.6 for all four BVP tests. A simple exterior reference solution (u^*, p^*) is constructed via Stokeslets placed at random locations in the interior. At any exterior point x , the velocity and pressure are then given by

$$u^*(x) = \frac{1}{4\pi} \sum_{k=1}^q \left(\log \frac{1}{|x - y_k|} \right) f_k + \frac{(x - y_k) \cdot f_k}{|x - y_k|^2} (x - y_k) \quad (2.48)$$

$$p^*(x) = \frac{1}{2\pi} \sum_{k=1}^q \frac{(x - y_k) \cdot f_k}{|x - y_k|^2} \quad (2.49)$$

The number of Stokeslets q is set to 5 and their chosen locations $\{y_k\}$ and strengths $\{f_k\}$ are displayed in Fig. 2.6(a). For the interior cases the Stokeslets are moved to the exterior of Γ at a radius of 2. Then either the Dirichlet data $g(x) = u^*(x), x \in \Gamma$ or the Neumann data $g(x) = T(u^*, p^*)(x), x \in \Gamma$ is used. The sup norm of this data is always in the range 0.1 to 1.

The numerical velocity solution $u(x)$ is obtained by solving the integral equation for σ using the standard Nyström method, then using the new Stokes close evaluation

⁶Traditionally, a combination of Stokeslets and rotlets are used to eliminate this nullspace in the context of Stokes BVPs. A more general procedure for handling nullspaces is recently given in [77].

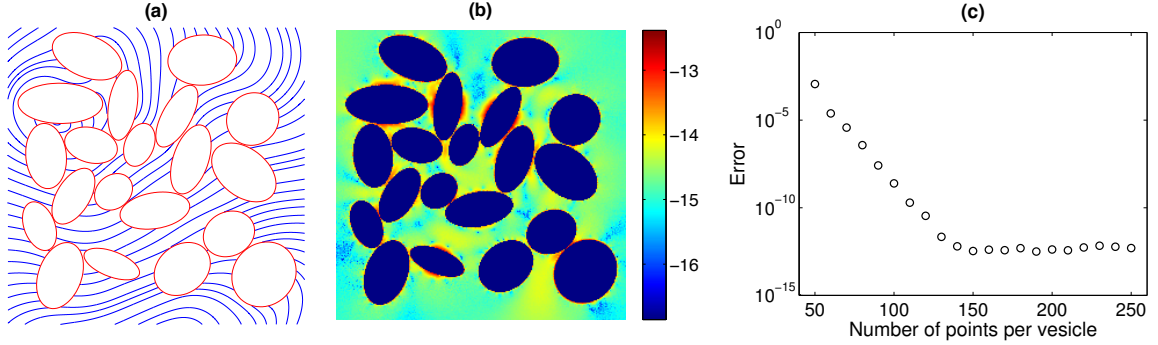


Figure 2.7: Test of exterior Dirichlet BVP for the 20 elliptical vesicles of Example 4, using the scheme of Sec. 2.5. (a) Streamlines of the velocity field in the exterior of vesicles. (b) \log_{10} of error in velocity field using the Nyström method (with GMRES) to solve the density, using the close evaluation scheme for vesicle-vesicle interactions and for final flow evaluation, with $N = 150$ nodes per vesicle. (c) convergence of the sup norm of error in the velocity field over the grid of values plotted in (b).

schemes for the SLP and/or DLP to evaluate the representation for u . We note that for the exterior Dirichlet case, the logarithmically-singular operator \mathbb{S} must be discretized; for this we use 16th-order Alpert corrections [3] as explained in [27, Sec. 4]. For the exterior Neumann case, the resulting error is plotted on a grid of spacing 0.02 in Fig. 2.6(b). The convergence of worst-case errors on this grid for all four BVPs is given in Table 2.2. It is superalgebraic, and clear that 12-digit accuracy results for N in the range 250 to 300, apart from the exterior Dirichlet case when 350 is needed.⁷ Away from a thin boundary layer the errors improve by at least 2 digits. Note that it is not very meaningful to do a direct comparison against the Laplace case since the smoothness of the data (i.e. distance away of the sources y_k) has a large effect on the convergence rate.

Example 4. We now focus on the exterior Dirichlet Stokes BVP, using the same combined representation $u = (\mathbf{D} + \mathbf{S})\sigma$ as before, with a more complicated boundary Γ comprising 20 elliptical vesicles that come very close to each other, shown in Fig. 2.7(a). Their minimum separation is 2×10^{-4} . This geometry is taken from [98,

⁷We verified that it was the Alpert correction for \mathbb{S} that caused this slightly slower convergence rate.

Table 4]. The number of quadrature nodes for each vesicle is the same, N . To resolve the Nyström matrix for the solution of σ one would need very high N when using the standard Nyström quadrature formulae, due to interacting vesicles. Therefore in this example we use an iterative solver (GMRES), using the new global close evaluation scheme for the SLP and DLP to apply all vesicle-vesicle interaction blocks of the system matrix. (The self-interactions of each vesicle are done by the standard Nyström formulae, with 8th-order Alpert corrections.) The number of GMRES iterations needed was high (around 600); however, our point is merely to demonstrate that our close evaluation scheme can be used to solve a close-to-touching geometry with a small N per vesicle.

The reference solution is the flow field due to a single Stokeslet of random strength at the center of each ellipse. We believe that this test case is crudely representative of the types of vesicle flows occurring in applications, with non-singular velocities where vesicles approach each other (this contrasts the case of approaching rigid bodies). The solution error for u is evaluated on a grid of spacing 0.016 for $N = 150$ in Fig. 2.7(b), and the convergence tested with respect to N in Fig. 2.7(c). Convergence is again superalgebraic, reaching 13 digits by $N = 150$. By comparison, the recent tests of [98] achieve only around 3 digits at this N .

Example 5. In our final example, we solve a more difficult Stokes Dirichlet problem with the same geometry as in Example 4. Consider 20 fixed elliptical rigid particles submerged in a linear shear flow with no-slip boundary conditions:

$$u_\infty(x) := \lim_{|x| \rightarrow \infty} u(x) = (x_2, 0), \quad u(x) = 0 \text{ for } x \in \Gamma. \quad (2.50)$$

Here we actually solve for the difference flow $u - u_\infty$ which obeys the exterior Dirichlet version of (2.44)–(2.47) with boundary data $g = -u_\infty|_\Gamma$. For this we use the same boundary integral representation as in Example 4. The exact solution is not

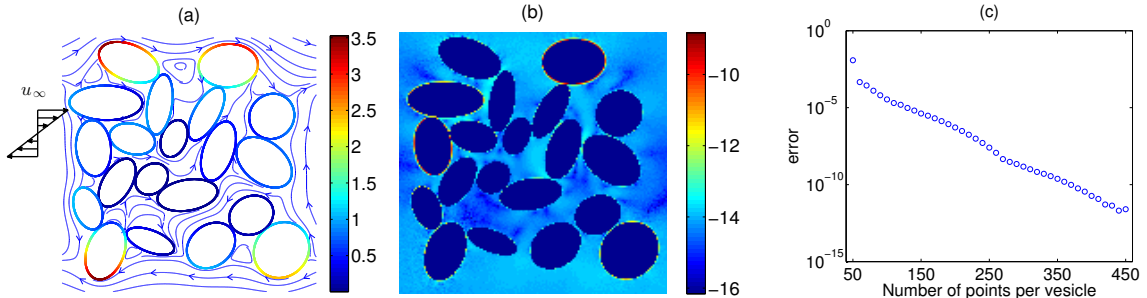


Figure 2.8: Example 5: Exterior Stokes Dirichlet BVP for 20 close rigid ellipses with no-slip boundary condition, driven by a shear flow u_∞ , using the scheme of Sec. 2.5. (a) Streamlines of the velocity field (blue lines with arrows), with magnitude of density σ indicated on the boundaries (color scale on right). (b) \log_{10} of error in velocity field using the Nyström method (with GMRES) to solve the density, using the close evaluation scheme for vesicle-vesicle interactions and for final flow evaluation, with $N = 300$ nodes per vesicle. (c) Convergence of the sup norm of error in the velocity field over the grid of values plotted in (b).

analytically known. In Fig. 2.8 we show the resulting numerical flow u and compare it against a reference solution obtained with $N = 500$ points per particle. Panel (b) shows the u error relative to this reference at $N = 300$; we achieve 9 digits uniformly, and 13 digits away from a very narrow boundary layer. Panel (c) demonstrates exponential convergence, as in Example 4, but with a slower rate. 12 digits of uniform accuracy are reached at $N = 430$. We believe that this slower convergence is due to the stronger singularities that may occur with close-to-touching rigid bodies discussed in the introduction, and we expect that in deformable particle flow applications the performance will be closer to that in Example 4.

2.6 Conclusions and discussion

We have presented a simple new scheme for evaluating the classical Stokes layer potentials on smooth closed curves with rapid spectral convergence in the number of quadrature nodes N , independent of the distance from the curve. We expect this to find applications in Stokes flow simulations with large numbers of close-to-touching vesicles. This builds upon Laplace layer potentials—including a new scheme

for single-layer potentials of independent interest—which in turn rely on barycentric-type quadratures for Cauchy’s formula. The underlying global periodic trapezoid rule, being simpler and potentially more efficient than panel-based schemes (see Remark 6), is the most common in vesicle applications. Thus our work complements recent panel-based Stokes quadratures [56].

We performed systematic tests, solving all eight boundary value problems (Laplace/Stokes, Dirichlet/Neumann, interior/exterior) via integral equations, and reach close to machine error in the solution with a small N that is essentially the same as that needed for the Nyström method itself. We also showed that our evaluator can be used to apply the operator in an iterative solution with close-to-touching boundaries (Example 4).

To evaluate at M targets the effort is $\mathcal{O}(N(N + M))$, although this would be easy to improve to $\mathcal{O}(N + M)$, by using the Cauchy FMM in the barycentric evaluations (2.16) and (2.19) and the Laplace Step 1 (splitting the sums in (2.26)–(2.27)), and a complex logarithmic FMM in (2.36) and (2.39). In vesicle applications N is small, so we leave this for future work.

In terms of future work, there are several variants that could be benchmarked, such as whether derivatives are best moved from Step 2 to Step 1, and alternatives in Remarks 2 and 3. However, given the results we presented, there is not much room for improvement in accuracy. For optimal speed, a Fortran/C/OpenMP library is certainly needed. Since the Cauchy formula quadratures of Sec. 2.3 (due to Ioakimidis) generalize barycentric interpolation to the complex plane, this connection is worth analyzing further (along these lines see [5]). Finally, we do not know of any way to extend the complex-analytic methods presented to 3D. For surfaces in 3D one cannot yet say which of the many existing schemes is to be preferred.

We provide documented MATLAB codes for the close evaluation of 2D Laplace and Stokes layer potentials, and driver codes to generate the tables and several figures

from this paper, at the following URL:

<https://users.flatironinstitute.org/~ahb/software/lsc2d.tgz>

CHAPTER III

Solution of Stokes Flow in Complex Nonsmooth 2D Geometries via a Linear-Scaling High-Order Adaptive Integral Equation Scheme

Preamble. We continue to present accurate close evaluation schemes for 2D Stokes layer potentials. The global scheme in the previous chapter has uniform resolution on a smooth closed curve, so it is most suitable for problems like vesicles and liquid drops. However, for problems with non-smooth geometries, such as fluid channels and microfluidic chips, a panel-based and adaptive quadrature is much more efficient and often necessary – this will be the focus of the current chapter. The panel scheme of this chapter first rewrites the Stokes layer potentials as a sum of contour integrals following the exact same procedure as in the global scheme, then contour integrals are accurately evaluated using polynomial approximations and recursive integration formulae. This is a joint work with Hai Zhu, Alex Barnett and Shravan Veerapaneni, which has been recently submitted.

3.1 Introduction

Since the pioneering work of Youngren and Acrivos [100, 99], boundary integral equation (BIE) methods have been widely used for studying various particulate Stokes

flow systems including drops, bubbles, capsules, vesicles, red blood cells and swimmers (e.g., see recent works [78, 66, 15] and references therein). BIE methods exploit the linearity of Stokes equations and offer several advantages such as reduction in dimensionality, exact satisfaction of far-field boundary conditions, and ease of handling moving geometries. Development of fast algorithms such as the fast multipole methods (FMMs) [24, 96, 91, 81, 21, 49], Ewald summation methods [71, 47, 46, 1] and others tailored to Stokes equations [74, 104, 93, 92], further extended their scope for solving problems in physically-realistic parameter regimes. Consequently, they are being used to investigate problems in a wide range of physical scales: from microhydrodynamics of isolated particles to large-scale flows generated by suspensions of particles (e.g., see [65, 54, 95]).

Despite their success, many numerical challenges still remain open for BIEs as applied to particulate Stokes flow problems. Prominent among them is the accurate handling of hydrodynamic interactions between surfaces that are almost in contact. For example, in dense suspension flows, the constituent particles often approach very close to each other or to the walls of the enclosing geometries (e.g., see Figure 3.1). To prevent artificial instabilities in this setting, numerical methods often require adaptive spatial discretizations, accurate nearly-singular integral evaluation schemes (note that while this issue is specific to BIEs, it manifests itself in other forms for other numerical methods) and accurate time-stepping schemes. The primary focus of this paper is addressing the first two issues for two-dimensional problems.

In this work, we introduce specialized panel quadrature schemes that can accurately evaluate layer potentials defined on a smooth open curve and for target points arbitrarily close to or on the curve. This helps the efficient tackling of dense particulate flows constrained in large multiply-connected domains such as in Figure 3.1. In addition, we formulate a set of rules for refining (or coarsening) the panels used to represent the boundaries, so that a user-specified error tolerance can be achieved

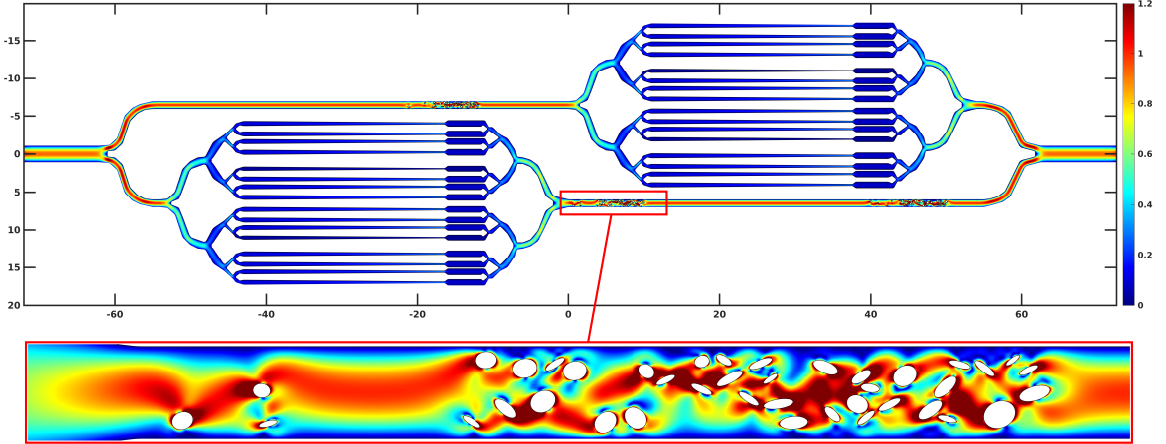


Figure 3.1: Snapshot from a simulation of bacterial suspension flow in a microfluidic chip geometry, which is inspired from the design proposed in [36]. A squirmer model [38] is used for modeling the bacteria, which treats them as rigid bodies with a prescribed slip at the fluid-structure interface. Thereby, we solve the Stokes equations with a no-slip boundary condition on the microfluidic chip geometry, a prescribed tangential velocity on the squirmer boundaries and an imposed parabolic flow profile at the inlet and outlet. We used 730,080 discretization points for the chip boundary, resulting in 1,460,160 degrees of freedom, and 128 discretization points at each of the 120 squirmers. GMRES took about 10 hours to reach a relative residual of 5.6×10^{-8} , using an 8-core 3.6 GHz Intel Core i7 processor with 128 GB of RAM. Color indicates the magnitude of fluid velocity. The estimated PDE relative L_2 -norm error is 2×10^{-5} .

automatically. One of the advantages of this adaptive panel refinement procedure is that it can handle geometries with corners (as in Figure 3.2) or nearly self-touching geometries using the same quadrature schemes for weakly- and nearly-singular integrals.

Our work is closely related to two recent efforts, that of Barnett et al. [8] and Ojala-Törnberg [56], both of which, in turn, are extensions to Stokes potentials of the Laplace work of Helsing-Ojala [31]. In [8], three of the co-authors developed high-order global quadrature schemes based on the periodic trapezoidal rule (PTR) for the evaluation of Laplace and Stokes layer potentials defined on a smooth closed planar curve, which achieved spectral accuracy for particles arbitrarily close to each other. Nevertheless, a key limitation of these close evaluation schemes is that they only work for closed curves and have uniform resolution on any part of a curve. Thereby, while

they are well-suited for moving rigid and deformable particles immersed in Stokes flow, new machinery is needed for the constraining complex geometries such as those in Figures 3.1 and 3.2.

On the other hand, Ojala–Törnberg [56] developed a panel quadrature scheme as done in this work. The main distinction is that [56] uses the BIE framework of Kropinski [45], converting Stokes equations into a biharmonic equation, then tailors it to a specific application (droplet hydrodynamics), whereas our scheme directly tackles all the Stokes layer potentials using physical variables (single-layer, double-layer and their associated pressure and traction kernels). Thereby, it can be integrated with existing BIE methods developed for various physical problems (colloids, drops, vesicles, squirmers and other suspensions) and flow conditions (imposed flow, pressure-, electrically- or magnetically-driven flows, etc) more naturally. We note that resolving nearly-singular integrals is an active area of research owing to its importance in solving several other linear elliptic partial differential equations (PDEs), such as Laplace, Helmholtz and biharmonic equations, via BIEs; other recent works that considered two-dimensional problems include [41, 9, 16, 64, 2, 58].

The predominant class of algorithms for adaptive meshing found in the Stokes BIE literature are the *reparameterization* schemes (also called the *resampling* techniques) that are dedicated to resolving boundary mesh distortions arising in deformable particle flow simulations (see [86] for a review on this topic). The primary focus of this work, on the other hand, is to determine an optimal distribution of boundary panels on a given domain so that a user-prescribed tolerance is met when computing the solution. Prior work in this area has mostly been restricted to low-order boundary element methods (BEMs); see [40] for a review. In the last 20 years, high-order *hp*-variants of BEM have been tested for Laplace and Helmholtz problems on polygons (for example [32, 6]). However, we are not aware of any Nyström *hp*-BIEs for the Stokes equations capable of handling arbitrary complex geometries. The recent

research of Rachh–Serkh [63] exploits a power-law basis resulting from analysis of the wedge problem to solve Stokes corner problems via a BIE. Here, we take a more pedestrian approach to handling corner singularities via the use of graded meshes. A prototypical example is shown in Figure 3.2.

The paper is organized as follows. In Section 3.2, we define the boundary integral operators and state the integral equation formulations for the viscous flow problems considered here. In Section 3.3, we present the panel quadrature rules for evaluating the Stokes layer potentials. We summarize our algorithm for adaptively discretizing a given geometry in Section 3.4. We consider several test cases and present results on the performance of our algorithms in Section 4.3, followed by conclusions in Section 3.6.

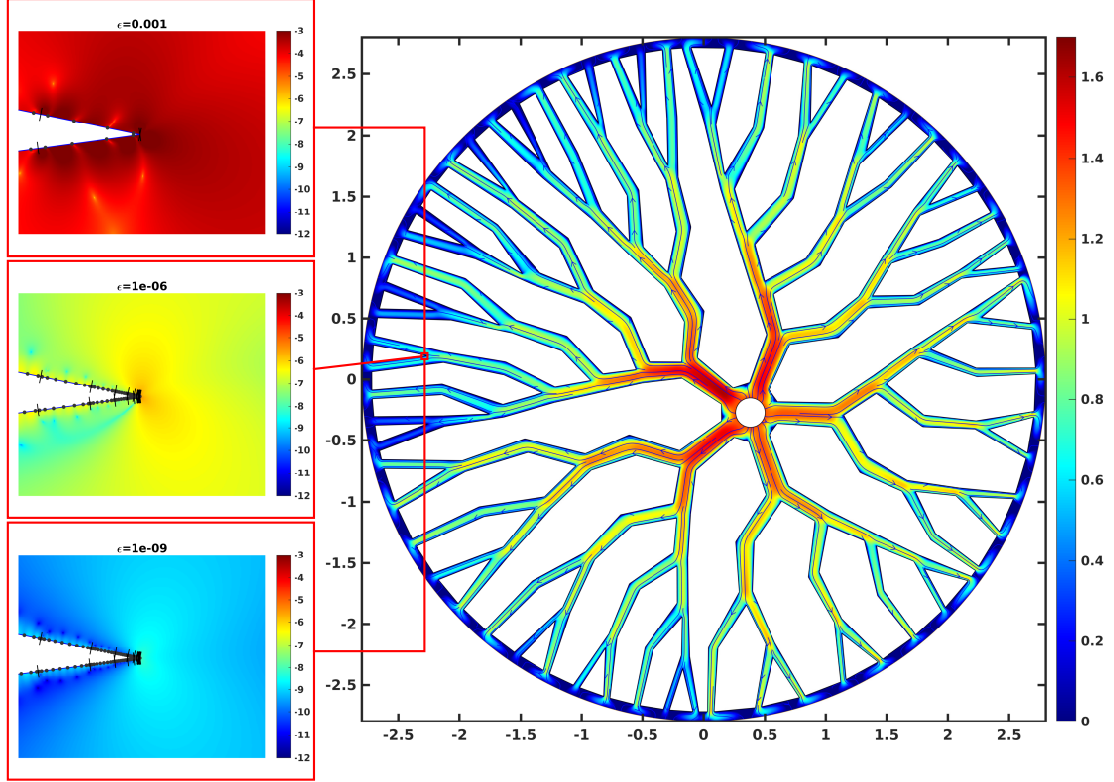


Figure 3.2: Solution of the Stokes equation in a nonsmooth circular vascular network with Dirichlet boundary condition. We apply no-slip boundary condition at all branch walls, and it is driven by a uniform flow from inner to outer circle. Color here indicates log of the magnitude of fluid velocity. We used automatically generated panels for both smooth boundaries and 378 corners, resulting in 356,580 degrees of freedom. GMRES took about 1 hour to reach a relative residual of 7.61×10^{-11} on an 8-core 4.0 GHz Intel Core i7 desktop. The PDE solution has a relative L_2 -norm error of 1×10^{-9} . Three high-resolution \log_{10} error plots that correspond to different user-requested tolerance ϵ near the same reentrant corner are shown on the left; here the short normal lines show panel endpoints, and the black dots quadrature nodes.

3.2 Mathematical preliminaries

In this section, we first state the PDE formulation for the example shown in Figure 3.2, reformulate it as a BIE and then discuss the evaluation of the resulting boundary integral operators.

3.2.1 Boundary value problem and integral equation formulation

The fluid domain Ω in Figure 3.2 is a multiply-connected region bounded by N_Γ closed curves, $\{\Gamma_i, i = 1, \dots, N_\Gamma\}$. Without loss of generality, let Γ_1 be the all-enclosing boundary—i.e., the outer circle in Figure 3.2—and let $\Gamma = \cup_i \Gamma_i$. Denoting the fluid viscosity by μ , the velocity by u and the pressure by p , the governing boundary-value problem, in the vanishing Reynolds number limit, is

$$-\mu\Delta u + \nabla p = 0 \quad \text{and} \quad \nabla \cdot u = 0 \quad \text{in} \quad \Omega, \quad (3.1a)$$

$$u = g \quad \text{on} \quad \Gamma. \quad (3.1b)$$

The Dirichlet data g must satisfy the consistency condition $\int_\Gamma g \cdot n ds = 0$, where n is the normal to Γ . For example, in Figure 3.2, the flow is driven by an outward flow condition at the inner circle and an inward flow condition at the outer circle. Their magnitude is chosen such that the consistency condition is respected. On the rest of the curves, a no-slip ($g = \mathbf{0}$) boundary condition is enforced.

While there are many approaches for reformulating the Dirichlet problem (3.1) as a BIE [60, 34], for simplicity, we use an indirect, combined-field BIE formulation that leads to a well-conditioned and non-rank-deficient linear system. We make the following *ansatz*:

$$u(x) = \sum_{i=1}^{N_\Gamma} (\mathbf{S}_i \sigma)(x) + (\mathbf{D}_i \sigma)(x) := (\mathbf{S}\sigma)(x) + (\mathbf{D}\sigma)(x), \quad x \in \Omega, \quad (3.2)$$

where σ is an unknown vector “density” function to be determined, \mathbf{S}_i is the velocity Stokes single layer potential (SLP) and \mathbf{D}_i is the double layer potential (DLP), defined

by

$$(\mathbf{S}_i\sigma)(x) = \frac{1}{4\pi\mu} \int_{\Gamma_i} \left(I \log \frac{1}{\rho} + \frac{r \otimes r}{\rho^2} \right) \sigma(y) ds_y, \quad x \in \Omega, \quad (3.3)$$

$$(\mathbf{D}_i\sigma)(x) = \frac{1}{\pi} \int_{\Gamma_i} \left(\frac{r \cdot n_y}{\rho^2} \frac{r \otimes r}{\rho^2} \right) \sigma(y) ds_y, \quad x \in \Omega. \quad (3.4)$$

Here, I is the 2-by-2 identity tensor, $r = x - y$, $\rho = \|r\|_2$, and ds_y is the arc length element on Γ . As (3.2) indicates, we use \mathbf{S} or \mathbf{D} to denote the sum of layer potentials due to all source curves. When combined with their associated pressure kernels (given in the Appendix), the convolution kernels above are fundamental solutions to the Stokes equations (3.1a); therefore, all that is left to ensure that the *ansatz* (3.2) solves the boundary-value problem is to enforce the velocity boundary condition (3.1b).

We introduce the operator block notation $\mathbb{S}_{ij}\sigma = (\mathbf{S}_j\sigma)(\Gamma_i)$, and similarly \mathbb{D}_{ij} for the double layer potential, i.e., the layer velocity potential restricted to the target curve Γ_i , using the source curve Γ_j . By taking the limit as x approaches Γ (from the interior for Γ_1 , and exterior for the rest of the curves), the standard jump relations for the single and double layer potentials [60, 34] give the interior case of the following $N_\Gamma \times N_\Gamma$ BIE system:

$$\left(\begin{array}{c} \left[\begin{array}{ccc} I/2 & 0 & \dots \\ 0 & I/2 & \dots \\ \vdots & \vdots & \ddots \end{array} \right] + \left[\begin{array}{ccc} \eta(\mathbb{S}_{11} + \mathbb{D}_{11}) & \mathbb{S}_{12} + \mathbb{D}_{12} & \dots \\ \eta(\mathbb{S}_{21} + \mathbb{D}_{21}) & \mathbb{S}_{22} + \mathbb{D}_{22} & \dots \\ \vdots & \vdots & \ddots \end{array} \right] \end{array} \right) \begin{array}{c} \left[\begin{array}{c} \eta\sigma(\Gamma_1) \\ \sigma(\Gamma_2) \\ \vdots \end{array} \right] \end{array} = \begin{array}{c} \left[\begin{array}{c} f(\Gamma_1) \\ f(\Gamma_2) \\ \vdots \end{array} \right], \end{array}$$

where $\eta = \begin{cases} -1, & \text{interior case,} \\ +1, & \text{exterior case.} \end{cases}$

(3.5)

The admixture of SLP and DLP in (3.2) insures that (3.5) is of Fredholm second kind, and that interior curves Γ_i , $i = 2, 3, \dots$ introduce no null-spaces [28, Thm. 2.1] [60, p.128]. Note that the negation of the density for the outer curve (first block column)

results in all positive identity blocks. Once we obtain the unknown density function σ on Γ , by solving (3.5), we can evaluate the velocity at any point in the fluid domain by using 3.2.

We also consider the *exterior* boundary-value problem, which can be thought of taking the above outer curve Γ_1 to infinity, removing it from the problem. The fluid domain Ω becomes the entire plane minus the closed interiors of the other curves; these curves we may relabel as $\Gamma = \cup_{i=1}^{N_\Gamma} \Gamma_i$. There is also now a given imposed background flow $u_\infty(x)$, which in applications is commonly a uniform, shear, or extensional flow. The decay condition $u(x) - u_\infty(x) = \Sigma \log \|x\| + O(1)$, for some $\Sigma \in \mathbb{R}^2$, must be appended to (3.5) to give a well-posed BVP [34]. The representation (3.2) of the physical velocity is now

$$u(x) = u_\infty(x) + (\mathbf{D}\sigma + \mathbf{S}\sigma)(x) . \quad (3.6)$$

The resulting BIE is given as the second case of (3.5). In the simplest case of physical no-slip boundary conditions, the data in (3.5) is now $f = -u_\infty|_\Gamma$, which one may check cancels the velocity on all boundaries.

The close-evaluation schemes of Helsing–Ojala [31] enable efficient and accurate evaluation of certain complex contour integrals arbitrarily close to the boundary, in the case where high order panel-based discretization is used. Thus, a route to evaluate the needed Stokes layer potentials close to the boundary, as in (3.5) and (3.2), is to express them in terms of Laplace potentials, which are in turn expressed via contour integrals. The rest of this section presents these formulae.

3.2.2 Fundamental contour integrals

Let τ be a given, possibly complex, scalar density function on Γ . We associate \mathbb{R}^2 with the complex plane \mathbb{C} . Let n_y be the outward-pointing unit vector at $y \in \Gamma$,

expressed as a complex number. Given a target point $x \in \Omega$, we define the potentials

$$I_L = (I_L\tau)(x) := \int_{\Gamma} \log|x-y|\tau(y)|dy| = \int_{\Gamma} \log|x-y|\frac{\tau(y)}{in_y}dy, \quad (\text{real logarithmic}) \quad (3.7)$$

$$I_C = (I_C\tau)(x) := \int_{\Gamma} \frac{\tau(y)}{y-x}dy, \quad (\text{Cauchy}) \quad (3.8)$$

$$I_H = (I_H\tau)(x) := \int_{\Gamma} \frac{\tau(y)}{(y-x)^2}dy, \quad (\text{Hadamard}) \quad (3.9)$$

$$I_S = (I_S\tau)(x) := \int_{\Gamma} \frac{\tau(y)}{(y-x)^3}dy. \quad (\text{supersingular}) \quad (3.10)$$

Note that, as functions of target point x , I_C , I_H and I_S are holomorphic functions in Ω , that $(d/dx)(I_C\tau)(x) = (I_H\tau)(x)$, and $(d/dx)(I_H\tau)(x) = 2(I_S\tau)(x)$. In contrast, I_L is not generally holomorphic; yet, for τ real, I_L is the real part of a holomorphic function.

3.2.3 Laplace layer potentials

Now, let τ be a real, scalar, density function on Γ . The Laplace single- and double-layer potentials are defined, respectively, by

$$(\mathcal{S}\tau)(x) := \frac{1}{2\pi} \int_{\Gamma} \left(\log \frac{1}{\rho} \right) \tau(y) ds_y, \quad x \in \Omega, \quad (3.11)$$

$$(\mathcal{D}\tau)(x) := \frac{1}{2\pi} \int_{\Gamma} \left(\frac{\partial}{\partial n_y} \log \frac{1}{\rho} \right) \tau(y) ds_y = \frac{1}{2\pi} \int_{\Gamma} \left(\frac{r \cdot n_y}{\rho^2} \right) \tau(y) ds_y, \quad x \in \Omega. \quad (3.12)$$

In terms of the contour integrals (3.7) and (3.8), using $ds_y = |dy| = dy/in_y$, and the restriction to τ real, we can rewrite the SLP and DLP as

$$(\mathcal{S}\tau)(x) = \frac{-1}{2\pi}(I_L\tau)(x) \quad \text{and} \quad (\mathcal{D}\tau)(x) = \text{Re} \frac{i}{2\pi}(I_C\tau)(x), \quad x \in \Omega. \quad (3.13)$$

We will also need the gradients and Hessians of Laplace layer potentials. The gradient of the SLP has components

$$\frac{\partial}{\partial x_1}(\mathcal{S}\tau)(x) = \frac{1}{2\pi} \text{Re} (I_C[\tau/in_y])(x), \quad \frac{\partial}{\partial x_2}(\mathcal{S}\tau)(x) = \frac{-1}{2\pi} \text{Im} (I_C[\tau/in_y])(x). \quad (3.14)$$

The gradient of the DLP requires the Hadamard kernel, and has components

$$\frac{\partial}{\partial x_1}(\mathcal{D}\tau)(x) = \frac{-1}{2\pi} \text{Im} (I_H\tau)(x), \quad \frac{\partial}{\partial x_2}(\mathcal{D}\tau)(x) = \frac{-1}{2\pi} \text{Re} (I_H\tau)(x). \quad (3.15)$$

The Hessians, which also require I_S , are discussed in Appendix A.

3.2.4 Stokes velocity layer potentials

As in [8], we rewrite the Stokes SLP in terms of Laplace potentials, and the DLP in terms of Laplace and Cauchy potentials. Using the identity

$$\frac{r \otimes r}{\rho^2} \sigma = \frac{r}{\rho^2} (r \cdot \sigma) = (r \cdot \sigma) \nabla_x \log \rho,$$

we can rewrite the Stokes SLP in terms of the Laplace SLP (3.11) as

$$\begin{aligned} (\mathbf{S}\sigma)(x) = \frac{1}{4\pi\mu} \left\{ \int_{\Gamma} \left(\log \frac{1}{\rho} \right) \sigma ds_y + \nabla \int_{\Gamma} \left(\log \frac{1}{\rho} \right) (y \cdot \sigma) ds_y \right. \\ \left. - x_1 \nabla \int_{\Gamma} \left(\log \frac{1}{\rho} \right) \sigma_1 ds_y - x_2 \nabla \int_{\Gamma} \left(\log \frac{1}{\rho} \right) \sigma_2 ds_y \right\}, \end{aligned} \quad (3.16)$$

where $\nabla = \nabla_x$ is assumed from now on. Therefore, three Laplace potentials (and their first derivatives) with real scalar density functions $y \cdot \sigma$, σ_1 , and σ_2 need to be computed to evaluate the Stokes SLP. Similarly, using the identity

$$\nabla \left(\frac{r \cdot n_y}{\rho^2} \right) = \frac{n_y}{\rho^2} - (r \cdot n_y) \frac{2r}{\rho^4}, \quad (3.17)$$

the Stokes DLP can be written as

$$\begin{aligned} (\mathbf{D}\sigma)(x) &= \frac{1}{2\pi} \int_{\Gamma} \frac{n_y}{\rho^2} (r \cdot \sigma) ds_y + \frac{1}{2\pi} \nabla \int_{\Gamma} \frac{r \cdot n_y}{\rho^2} (y \cdot \sigma) ds_y \\ &\quad - \frac{1}{2\pi} x_1 \nabla \int_{\Gamma} \frac{r \cdot n_y}{\rho^2} \sigma_1 ds_y - \frac{1}{2\pi} x_2 \nabla \int_{\Gamma} \frac{r \cdot n_y}{\rho^2} \sigma_2 ds_y. \end{aligned} \quad (3.18)$$

While the last three terms are gradients of Laplace DLPs, the first term requires a Cauchy integral. More concisely, we can write (3.16) and (3.18) as

$$(\mathbf{S}\sigma)(x) = \frac{1}{2\mu} \left((\mathcal{S}\sigma_1, \mathcal{S}\sigma_2) + \nabla \mathcal{S}[y \cdot \sigma] - x_1 \nabla \mathcal{S}\sigma_1 - x_2 \nabla \mathcal{S}\sigma_2 \right)(x), \quad x \in \Omega, \quad (3.19)$$

$$(\mathbf{D}\sigma)(x) = \left(\frac{1}{2\pi} \text{Re} (I_C(\tau_1), I_C(\tau_2)) + \nabla \mathcal{D}[y \cdot \sigma] - x_1 \nabla \mathcal{D}\sigma_1 - x_2 \nabla \mathcal{D}\sigma_2 \right)(x), \quad x \in \Omega, \quad (3.20)$$

where, as above, a pair (\cdot, \cdot) indicates two vector components, and a short calculation verifies that the complex scalar density functions τ_1 and τ_2

$$\tau_1 = (\sigma_1 + i\sigma_2) \frac{\text{Re } n_y}{n_y}, \quad \tau_2 = (\sigma_1 + i\sigma_2) \frac{\text{Im } n_y}{n_y}, \quad (3.21)$$

where n_y is interpreted as a complex number, makes the identity hold (i.e. the first term in (3.20) equals the first term in (3.18)).

In summary, the procedure discussed in this section enables us to express the velocity field anywhere in the fluid domain, represented by (3.2), in terms of funda-

mental contour integrals. Similar formulae exist for the fluid pressure and hydrodynamic stresses, which are commonly needed in several applications; we present these in Appendix A.

3.3 Nyström discretization and evaluation of layer potentials

3.3.1 Overview: discretization and the plain Nyström formula

Firstly, we need to specify a numerical approximation of the density function σ . For simplicity, consider the case of an exterior BVP on a single closed curve Γ parameterized by $Z : [0, 2\pi) \rightarrow \mathbb{R}^2$, such that $\Gamma = Z([0, 2\pi))$. The Stokes BIE (3.5) is then

$$(I/2 + \mathbb{K})\sigma = f, \quad \mathbb{K} := \mathbb{D} + \mathbb{S}, \quad (3.22)$$

where σ and f are 2-component vector functions on Γ .

Given the user requested tolerance ϵ , the boundary is split into n_Λ disjoint panels Λ_i , $i = 1, \dots, n_\Lambda$ using the adaptive algorithm to be described in Section 3.4. Each panel will have p nodes, giving $N = pn_\Lambda$ nodes in total, hence $2N$ density unknowns. The i th panel is $\Lambda_i = Z([a_{i-1}, a_i])$, where a_i , $i = 0, \dots, n_\Lambda$ are the parameter break-points of all panels (where $a_{n_\Lambda} \equiv a_0 \pmod{2\pi}$). Let the p -point Gauss–Legendre nodes and weights on the parameter interval $[a_{i-1}, a_i]$ be $t_j^{(i)}$ and $w_j^{(i)}$ respectively, for $j = 1, \dots, p$. Then, for smooth functions f on Γ , the quadrature rule

$$\begin{aligned} \int_{\Gamma} f(y) ds_y &= \int_0^{2\pi} f(Z(t)) |Z'(t)| dt \approx \sum_{i=1}^{n_\Lambda} \sum_{j=1}^p f(Z(t_j^{(i)})) |Z'(t_j^{(i)})| w_j^{(i)} \\ &=: \sum_{\ell=1}^N f(y_\ell) |Z'(t_\ell)| w_\ell \end{aligned} \quad (3.23)$$

holds to high-order accuracy. In the last formula above $\{t_\ell\}_{\ell=1}^N$ denotes the entire set of parameter nodes, $y_\ell = Z(t_\ell)$ their images on Γ , and w_ℓ their corresponding weights.

The Nyström method [44, Sec. 12.2] is then used to approximately solve the BIE (3.22). Broadly speaking, this involves substituting the quadrature rule (3.23) for the integral implicit in the BIE, then enforcing the equation at the quadrature nodes themselves. The result is the $2N$ -by- $2N$ linear system,

$$\mathbf{A}\boldsymbol{\sigma} = \mathbf{f} \tag{3.24}$$

where $\mathbf{f} := \{f(t_\ell)\}_{\ell=1}^N$ is the vector of 2-component values of the right-hand side f at the N nodes, and $\boldsymbol{\sigma} := \{\sigma_\ell\}_{\ell=1}^N$ is the vector of 2-component densities at the N nodes. $\mathbf{A} = \{A_{ij}\}_{i,j=1,\dots,n_\Lambda}$ is a $n_\Lambda \times n_\Lambda$ block matrix, where each block A_{ij} is a $2p \times 2p$ submatrix that represents the interaction from source panel Λ_j to target panel Λ_i . For targets that are “far” (in a sense discussed below) from a given source panel, the formula for filling corresponding elements of \mathbf{A} is simple. Letting the index be ℓ for such a target node, and ℓ' for a source lying in such a panel, using the smooth rule (3.23) gives the matrix element

$$\mathbf{A}_{\ell,\ell'} = K(y_\ell, y_{\ell'}) |Z'(t_{\ell'})| w_{\ell'} , \quad (\text{Nyström rule for } y_\ell \text{ “far” from panel containing } y_{\ell'}) \tag{3.25}$$

where $K(x, y) = S(x, y) + D(x, y)$, the latter being the kernels appearing in (3.3)–(3.4). Recall that each element in (3.25) is a 2×2 tensor, since K is. This defines the plain (smooth) rule for matrix elements (note that we need not include the diagonal $I/2$ from (3.22) since $\ell = \ell'$ is never a “far” interaction).

Sections 3.3.2–3.3.4 will be devoted to defining “close” vs “far” and explaining how “close” matrix elements are filled. Assuming for now that this has been done, the result is a dense matrix A that is well-conditioned independent of N , because the underlying integral equation is of Fredholm second kind. Then an iterative solver for (3.24), such as GMRES, will converge rapidly. The result is the vector $\boldsymbol{\sigma}$ approximating the density at the nodes. Assuming that (3.24) has been solved exactly,

there is still a discretization error, whose convergence rate is known to inherit that of the quadrature scheme applied to the kernel [44, Sec. 12.2]. Given this, the density function may be evaluated at any point $y \in \Gamma$ using either the Nyström interpolant (which is global and hence inconvenient), or local p th-order Lagrange interpolation from just the points on the panel in which y lies. When needed, we will use the latter. The generalization of the above Nyström method to multiple closed curves is clear.

Remark III.1. When the problem size is large, the matrix-vector multiplication in (3.24), which requires $\mathcal{O}(N^2)$ time, can be rapidly computed using the fast multipole methods (FMM) in only $\mathcal{O}(N)$ time. This is because all but $\mathcal{O}(N)$ of the matrix elements involve the plain rule (3.25), for which applying the matrix is equivalent to evaluation of a potential with weighted source strengths. In our large examples (Figures 3.1 and 3.2), we use an OpenMP-parallelized Stokes FMM code due to Rachh, built upon the Goursat representation of the biharmonic kernel [62, 23].

Finally, once σ has been solved for, the evaluation of the velocity $u(x)$ at arbitrary targets $x \in \Omega$ is possible, by approximating the representation (3.2) or (3.6), as appropriate. By linearity, this breaks into a sum of contributions from each source panel on each curve, which may then be handled separately. Thus a given target $x \in \Omega$ may, again, fall “far” or “close” to a given source panel (denoted by $\Lambda = Z([a, b])$). If it is “far” (according to the same criterion as for matrix elements), then a simple plain rule is used. Letting u_Λ be the contribution to u from source panel Λ , this rule arises, as with (3.25), simply by substituting (3.23) into the representation, giving

$$\begin{aligned}
 u_\Lambda(x) &= \int_{\Lambda} K(x, y) \sigma(y) ds_y \\
 &\approx \sum_{j=1}^p K(x, y_j) |Z'(t_j)| w_j \sigma_j, \quad (\text{evaluation rule for } x \text{ “far” from } \Lambda)
 \end{aligned}
 \tag{3.26}$$

where $y_j := Z(t_j)$ are the nodes, and $\sigma_j := \sigma(y_j)$ the 2-component density values, belonging to Λ . For large N , the FMM is ideal for the task of evaluating u at many

targets, using this plain rule. An identical quadrature rule may be applied to the representations in the Appendix to evaluate pressure and traction at x .

3.3.2 Close-evaluation and self-evaluation corrections

If a target (on-surface node or off-surface evaluation point) is close enough to a source panel that (3.25) is inaccurate, a new *close-evaluation* formula is needed. A special case is when the target is a node from the source panel itself, which we call *self-evaluation*. The integrand is nearly singular for close-evaluation and singular for self-evaluation; in the latter case, the integral is understood as an improper integral (SLP) or a principal value integral (DLP). Conveniently, we will use the same rule for self-evaluation as both on-surface and off-surface close-evaluation (this contrasts much prior work, where the self-evaluation used another set of tailored high-order integration schemes; see, e.g., [27]).

We quantify “close” and “far” as follows. Given a panel $\Lambda_j = Z([a, b])$, a target point x is *close* to Λ_j if it lies inside the ellipse

$$|z - Z(a)| + |z - Z(b)| = C S, \quad (3.27)$$

otherwise x is *far* from Λ_j . A panel Λ_i is close to Λ_j if any point $z \in \Lambda_i$ lies inside the ellipse (3.27), otherwise Λ_i is far from Λ_j . (Note that Λ_j is close to itself.) In (3.27), S is the arc length of Λ_j and $C > 1$ is a constant. For the numerical examples in Section 4.3 we have picked $C = 2.5$, which is large enough to include all of both neighboring panels of Λ_j most of the time.

The rationale for the above heuristic is based upon the accuracy of the plain rule (3.26) (and its corresponding matrix element rule (3.25)). Examining (3.23), if the integrand $f(Z(t))|Z'(t)|$ is analytic with respect to t within a Bernstein ellipse (for the parameter domain $[a_{j-1}, a_j]$ for this panel) of size parameter $\varrho > 1$, then the error

convergence for the Gauss–Legendre rule for this panel is $\mathcal{O}(\varrho^{-2p})$, i.e. exponential in the panel degree p [84, Theorem 19.3]. Since in our case $f(Z(t)) = K(x, Z(t))\sigma(Z(t))$, and $K(x, y)$ is analytic for $x \neq y$, for such analyticity of the integrand, x must be outside the *image* of this Bernstein ellipse under Z . In the case where the panel is approximately flat, this image is approximated by the ellipse with foci $Z(a)$ and $Z(b)$, which gives the above geometric “far” criterion. The choice of C is empirically made to achieve an exponential error convergence rate in p no worse than that due to the next-neighboring panels discussed in Stage 1 of Section 3.4, in the case of panel shapes produced by the procedure in that section. Note that $\sigma(Z(t))$ must also be assumed to be analytic in the ellipse; we expect this to hold again because the panels will be sufficiently refined. For a more detailed error analysis of the plain panel rule using the Bernstein ellipse, see [2, Sect. 3.1].

So far we have presented (only in the “far” case) formulae for both filling Nyström matrix elements (3.25), and for evaluation of the resulting velocity potential (3.26). We now make the point that, in both the “far” and “close” cases, these are essentially the same task.

Remark III.2 (Matrix-filling is potential evaluation). The matrix element formula (3.25) is just a special case of the evaluation formula (3.26) for the on-surface target $x = y_\ell$, and with a Kronecker-delta density $\sigma_j = \delta_{j,\ell}$. I.e., one can compute (the “far” contributions to) $\mathbf{A}\boldsymbol{\sigma}$ from $\boldsymbol{\sigma}$, as needed for each iteration in the solution of (3.24), simply via the evaluation formula (3.26) with targets x as the set of nodes $\{y_\ell\}$. This will also apply to the special “close” formulae presented below. Henceforth we now present only formulae for evaluation; the corresponding matrix element formulae are easy to extract (see Section 3.3.4)

Note that the diagonal blocks A_{ii} when computed using the special close-evaluation quadratures will implicitly include the $I/2$ jump relation appearing in (3.22).

Finally, to accelerate the computation, the close- and self-evaluations can be pre-

computed as matrices (see Section 3.3.4) from which the (inaccurate) matrices involving the plain rule (3.25) are subtracted. The resulting “correction matrix” blocks are assembled and stored as a $2N$ -by- $2N$ sparse matrix. The entire application of \mathbf{A} to the density vector is then performed using the FMM with the plain rule (3.25), plus the action of this sparse matrix to replace the “close” interactions with their accurate values. This application is used to solve the whole linear system iteratively. We do this for our large-scale examples in Section 4.3.

3.3.3 Close-evaluation of potentials

Since we will perform all Nyström matrix filling using the same formulae as for close-evaluation of potentials (on- or off-surface), we now present formulae for the close-evaluation task. As before, we consider a single target point $x \in \Omega$ which is “close” to the single source panel $\Lambda = Z([a, b])$, on which a density σ_j is known at the nodes $j = 1, \dots, p$. Recall that the p -point Gauss–Legendre nodes and weights for the parameter interval $[a, b]$ are t_j and w_j .

We adapt special panel quadratures proposed for the Laplace case by Helsing and Ojala [31]. They use high-order polynomial interpolation in the complex plane to approximate the density function, then apply a recursion to exactly evaluate the near-singular integral for each monomial basis function. In Section 3.2 we showed that all the needed Stokes potentials may be written in terms of $I_L(x)$, $I_C(x)$, $I_H(x)$ and $I_S(x)$ from (3.7)–(3.10), involving scalar Cauchy densities τ derived from the given Stokes density σ . Thus in the following subsections we need only cover close-evaluation of each contour integral in turn. Although much of this is known [31], there are certain implementation details and choices that make a complete distillation valuable.

It turns out that the Helsing–Ojala polynomial approximation is most stable when assuming that $Z(a) = 1$ and $Z(b) = -1$, i.e. the panel endpoints are ± 1 in the

complex plane. Thus we start by making this assumption, then in Sec. 3.3.3.4 show how to correct the results for a panel with general endpoints.

Recall that σ , hence the derived scalar functions τ needed in the contour integrands, is available only at the p nodes of Λ . In order to improve the accuracy of the complex approximation for bent panels, firstly an upsampling is performed to $m > p$ “fine” nodes, using Lagrange interpolation in the parameter $t \in [a, b]$ from the p nodes to the m fine nodes. We find that $m = 2p$ is beneficial without incurring significant extra cost. Let $\tilde{\tau}_j$, $j = 1, \dots, m$ denote the fine density values, and $z_j = Z(\tilde{t}_j) \in \Lambda$ be the fine nodes, where \tilde{t}_j and \tilde{w}_j are the m -point Gauss–Legendre nodes and weights respectively for $[a, b]$. The following schemes now will use only the fine values and nodes.

3.3.3.1 Close evaluation of the Cauchy potential

We approximate τ on the panel Λ in the complex variable by the degree $m - 1$ polynomial

$$\tau(y) \approx \sum_{k=1}^m a_k y^{k-1}, \quad y \in \Lambda . \quad (3.28)$$

The vector of coefficients $\mathbf{a} := \{a_k\}_{k=1}^m$ is most conveniently found by using a dense direct solve of the square Vandermonde system

$$V \mathbf{a} = \tilde{\boldsymbol{\tau}} , \quad (3.29)$$

with elements $V_{jk} = z_j^{k-1}$, $j, k = 1, \dots, m$, and right hand side $\tilde{\boldsymbol{\tau}} := \{\tilde{\tau}_j\}_{j=1}^m$. It is known that, for any arrangement of nodes z_j other than those very close to the roots of unity, the condition number of V grows exponentially with m [57]. However, as discussed in [31, App. A], at least for $m < 50$, despite the extreme ill-conditioning, *backward stability* of the linear solver insures that the resulting polynomial matches the values at the nodes to close to machine precision. For this we use MATLAB

`mldivide` which employs standard partially-pivoted Gaussian elimination.

The remaining step is to compute the contour integral of each monomial,

$$p_k = p_k(x) := \int_{-1}^1 \frac{y^{k-1}}{y-x} dy, \quad k = 1, \dots, m, \quad (3.30)$$

which are, recalling (3.8), then combined using (3.28) to get

$$I_C \approx \sum_{k=1}^m a_k p_k. \quad (3.31)$$

By design, since the monomials are with respect to y in the complex plane (as opposed to, say, the parameter t), Cauchy's theorem implies that each p_k is independent of the curve Λ and depends only on the end-points. Specifically, for $k = 1$ we may integrate analytically by deforming Λ to the real interval $[-1, 1]$, so

$$p_1 := \int_{-1}^1 \frac{1}{y-x} dy = \log(1-x) - \log(-1-x) \pm 2\pi i \mathcal{N}_x \quad (3.32)$$

where $\mathcal{N}_x \in \mathbb{Z}$ is an integer winding number that depends on the choice of branch cut of the log function. For the standard cut on the negative real axis then $\mathcal{N}_x = 0$ when x is outside the domain enclosed by the oriented curve composed of Λ traversed forwards plus $[-1, 1]$ traversed backwards, $\mathcal{N}_x = +1$ (-1) when x is inside a region enclosed counterclockwise (clockwise) [31]. However, since it is inconvenient and error-prone to decide \mathcal{N}_x for points on or very close to Λ and $[-1, 1]$, we prefer to combine the two logs then effectively rotate its branch cut by a phase $\phi \in \mathbb{R}$, by using

$$p_1 = i\phi + \log \frac{1-x}{e^{i\phi}(-1-x)}, \quad (3.33)$$

where $\phi = -\pi/4$ when the upwards normal of the panel points into Ω (as for an interior curve), or $\phi = \pi/4$ otherwise. This has the effect of pushing the branch cut

“behind” the panel (away from Ω ; see Fig. 3.3), with the cut meeting ± 1 at an angle ϕ from the real axis. The potential is correct in the closure of Ω , including on the panel itself, without any topological tests needed (hence the unified handling of close and self evaluations in Sec. 3.3.2). This can fail if a panel is very curved (such a panel would be inaccurate anyway), or if a piece of Ω approaches close to the back side of the panel (which can be prevented by adaptive refinement as in Section 3.4). In practice we find that this is robust. However, we will mention one special situation where (3.33) could fail and therefore careful adjustment of the branch cut is critical; see Remark III.3 below.

The following 2-term recurrence is easy to check by adding and subtracting xy^{k-1} from the numerator of the formula (3.30) for p_{k+1} :

$$p_{k+1} = xp_k + (1 - (-1)^k)/k . \quad (3.34)$$

For $|x| < 1.1$ we find that the recurrence is sufficiently stable to use upwards from the value p_1 computed by (3.33), to get p_2, \dots, p_m . However, for targets outside this close disc, especially for larger m , the upwards direction is unstable. Thus, here instead we use numerical quadrature on (3.30) to get

$$p_m \approx \sum_{j=1}^m \frac{z_j^{m-1}}{z_j - x} Z'(\tilde{t}_j) \tilde{w}_j , \quad (3.35)$$

then apply (3.34) downwards to compute p_{m-1}, \dots, p_2 , and as before use p_1 from (3.33). Outside of the disc, no branch cut issues arise.

Remark III.3 (branch cuts at corners). When a panel Λ is directly touching a corner, directly applying (3.33) can fail no matter how much the panels are refined. In Figure 3.3a, the panel on the opposite side of the corner, Λ' , is always behind Λ and lying across the branch cut associated to Λ . Consequently, the close evaluation from Λ to Λ' results in completely wrong values, also leading to ill-conditioning of the whole

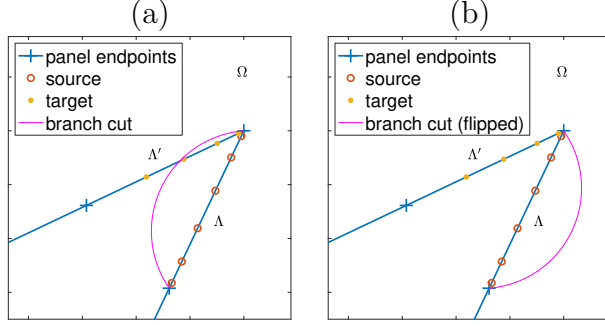


Figure 3.3: Special handling of close evaluation branch cut when the panel is touching a reentrant corner. (a) The target panel Λ' is crossing the branch cut of the source panel Λ defined by (3.33), resulting in wrong close evaluation values. (b) Changing the sign of ϕ in (3.33) flips the branch cut to the other side; close evaluation at the targets are now correct.

system (3.24). Instead, one can simply change the sign of ϕ in (3.33) to flip the branch cut to accommodate the targets on Λ' (Figure 3.3b). In practice, we find that this is robust for corners of arbitrary angles.

3.3.3.2 Close evaluation of the logarithmic potential

For τ real, we can write the final form in (3.7) as

$$I_L = \operatorname{Re} \int_{\Lambda} \log(y-x) \frac{\tau(y)}{in_y} dy, \quad (3.36)$$

which shows that the quantity to approximate on Λ as a complex polynomial is $\tau(y)/in_y$. Thus we find the coefficients in

$$\frac{\tau(y)}{in_y} \approx \sum_{k=1}^m b_k y^{k-1}, \quad y \in \Lambda, \quad (3.37)$$

by solving (3.29) but with modified right hand side $\{\tilde{\tau}_j/in_{z_j}\}_{j=1}^m$. Defining

$$q_k := \int_{-1}^1 y^{k-1} \log(y-x) dy, \quad k = 1, \dots, m \quad (3.38)$$

then (3.37) gives

$$\int_{-1}^1 \log(y-x) \frac{\tau(y)}{in_y} dy \approx \sum_{k=1}^m b_k q_k, \quad (3.39)$$

whose real part is I_L . Each q_k is computed from p_k of (3.30) as evaluated in Sec. 3.3.3.1, via a formula easily proven by integration by parts,

$$\begin{aligned} q_k &= \frac{-p_{k+1} + \log(1-x) - (-1)^k \log(1+x)}{k} \\ &= \begin{cases} (-p_{k+1} + i\phi + \log \frac{1-x}{e^{i\phi}(-1-x)})/k, & k \text{ even}, \\ (-p_{k+1} + \log[(1-x)(-1-x)]/k, & k \text{ odd}, \end{cases} \end{aligned} \quad (3.40)$$

where the latter form is that used in the code, needed to match the branch cut rotation used for p_k in (3.33). The final evaluation of I_L is then via

$$I_L \approx \operatorname{Re} \sum_{k=1}^m b_k q_k. \quad (3.41)$$

3.3.3.3 Close evaluation of the Hadamard and supersingular potentials

The double-layer Stokes velocity requires gradients of Laplace potentials (3.15), which require I_H . Also, the traction of the Stokes single-layer (A.5) involves I_H applied to $\tau(y)/in_y$, and the traction of the Stokes double-layer further involves I_S (Appendix A).

Using the complex monomial expansion (3.28), we have

$$I_H \approx \sum_{k=1}^m a_k r_k, \quad I_S \approx \sum_{k=1}^m a_k s_k, \quad (3.42)$$

where

$$r_k := \int_{-1}^1 \frac{y^{k-1}}{(y-x)^2} dy, \quad s_k := \int_{-1}^1 \frac{y^{k-1}}{(y-x)^3} dy, \quad k = 1, \dots, m. \quad (3.43)$$

The following formulae can be shown by integration by parts, and enable r_k and s_k to be found,

$$r_k = (k-1)p_{k-1} + \frac{(-1)^{k-1}}{-1-x} - \frac{1}{1-x}, \quad s_k = \frac{k-1}{2}r_{k-1} + \frac{(-1)^{k-1}}{2(-1-x)^2} - \frac{1}{2(1-x)^2}, \quad (3.44)$$

using p_k from (3.30) as computed in Sec. 3.3.3.1, and $p_0 = 0$.

3.3.3.4 Transforming for general panel endpoints

To apply close-evaluation methods in the above three sections to a general panel $\Lambda = Z([a, b])$, define the complex scale factor $s_0 := (Z(b) - Z(a))/2$ and origin $x_0 = (Z(b) + Z(a))/2$. Then the affine map

$$x = s(\tilde{x}) := \frac{\tilde{x} - x_0}{s_0}$$

takes any target \tilde{x} to its scaled version x . Likewise, the fine nodes are scaled by $z_j = s(Z(\tilde{t}_j))$, and the factor Z' in (3.35) is replaced by Z'/s_0 . Following Sec. 3.3.3.1 using these scaled target and fine nodes, no change in the result I_C is needed. However, the value of I_H computed in Sec. 3.3.3.3 must afterwards be divided by s_0 , and the value of I_S divided by s_0^2 . The value of I_L computed in Sec. 3.3.3.2 must be multiplied by $|s_0|$, and then have $|Z'(\tilde{t}_j)w_j/s_0| \log |s_0|$ subtracted.

3.3.4 Computation of close-evaluation matrix blocks

The above described how to evaluate $(I_L\tau)(x)$, $(I_C\tau)(x)$, $(I_H\tau)(x)$ and $(I_S\tau)(x)$ given known samples $\tilde{\tau}_j$ at a panel's fine nodes. In practice it is useful to instead precompute a matrix block A which takes any density values at a panel's original p nodes y_j to a set of n target values of the contour integral. Consider the case of the Cauchy kernel, and let A denote this n -by- p matrix. Let L be the m -by- p Lagrange interpolation matrix from the nodes t_j to fine nodes \tilde{t}_j , which need be filled once and

for all. Let P be the n -by- m matrix with entries $P_{ik} = p_k(x_i)$, given by (3.30), where $\{x_i\}_{i=1}^n$ is the set of desired targets. In exact arithmetic one has

$$A = PV^{-1}L .$$

However, since V is very ill-conditioned, filling V^{-1} and using it to multiply to the right is numerically unstable. Instead an *adjoint method* is used: one first solves the matrix equation $V^\top X = P^\top$, where \top indicates non-conjugate transpose, then forms the product

$$A = X^\top L .$$

The matrix solve is done in a backward stable fashion via MATLAB's `mldivide`. A further advantage of the adjoint approach is that if n is small, the solve is faster than computing V^{-1} .

The formulae for the logarithmic, Hadamard and supersingular kernels are analogous.

3.4 Adaptive panel refinement

In order to solve a BIE to high accuracy, it is necessary to set up panels such that the given complex geometry is correctly resolved. In this section, we describe a procedure that adaptively refine the panels based solely on the geometric properties. Specifically, our refinement algorithms take into account the accuracy of geometric representations (including arc length and curvature), the location of corners, and the distance between boundary components. It necessarily has some ad-hoc aspects, yet we find it quite robust in practice.

Suppose that for a user-prescribed tolerance ϵ , the goal is to find a partition $\Gamma = \bigcup_{i=1}^{n_\Lambda} \Lambda_i$ such that the error, ε , of evaluating boundary integral operators such as (3.22) satisfies $\varepsilon \lesssim \epsilon$. To this end, we describe our adaptive refinement scheme which

proceeds with three stages. In what follows, we again assume that the panel under consideration $\Lambda = Z([a, b])$ is rescaled such that its two endpoints are ± 1 .

Stage 1: Choice of overall p . Given tolerance ϵ , the goal is to determine a number of quadrature points, p , applied to all panels, such that the relative quadrature error on any panel is $\mathcal{O}(\epsilon)$. As mentioned above, the p -point Gauss–Legendre quadrature on $[-1, 1]$ has $\mathcal{O}(\varrho^{-2p})$ error if the integrand can be analytically extended to a Bernstein ellipse of parameter $\varrho > 1$, where the semi-major axis of this ellipse is $(\varrho + \varrho^{-1})/2$ [84, Thm. 19.3]. Therefore, making $\varrho^{-2p} \leq \epsilon$ we obtain the first term of the right-hand side in

$$p \geq \left\lceil \frac{\log_{10}(1/\epsilon)}{2 \log_{10} \varrho} \right\rceil + c, \quad (3.45)$$

where the second term accounts for unknown prefactors. Empirically we set $c = 1$.

To determine ϱ , we require that the Bernstein ϱ -ellipse of each panel encloses both its immediate neighboring panels. This insures that, when applying the smooth quadrature rule (3.23) between the nearest non-neighboring (“far”) panels that do not touch a corner, the integrand continues to a function analytic inside the ϱ -ellipse, so, by the above discussion, the relative error is no worse than ϵ . (This will not apply to panels touching a corner, but they are small enough to have negligible contributions.) Stages 2–3 below will place an upper bound of λ on the ratio of the lengths (with respect to parameter) of neighboring panels. Combining these two relations gives

$$\frac{\varrho + \varrho^{-1}}{2} = 1 + \frac{2}{\lambda}. \quad (3.46)$$

One then solves (3.46) for ϱ and substitutes it into (3.45) to obtain a lower bound for p . For example, $\lambda \leq 3$ holds in our examples, so $\varrho = 3$, and therefore we have as sufficient the simple rule $p = \lceil \log_{10}(1/\epsilon) + 1 \rceil$.

Stage 2: Local geometric refinement. In this stage, panels are split based on local geometric properties:

1. *Corner refinement.* Panels near a corner are refined geometrically so that each panel is a factor λ shorter in parameter than its neighbor (see lines 8–11 of Algorithm 1).

To each corner is associated a factor $\lambda \geq 2$. A rule of thumb is to use $\lambda = 2$ for sharper corners (e.g. whose angle θ is close to 0 or 2π) which are harder to resolve, and use $\lambda > 2$ for “flatter” corners (e.g. θ closer to π) to reduce the number of panels without affecting the overall achieved accuracy. In practice, we use $\lambda = 3$ for corners $\pi/2 \leq \theta \leq 3\pi/2$; for a problem with many flat corners, this can reduce the total number of unknowns by a factor of about $2/3$ (or about $\log_3 2$).

Near a corner, refinement stops when the panels touching the corner are shorter than ϵ^α , where α is an empirical power parameter chosen for each corner. Recent theoretical results for the plain double-layer formulation for the Stokes Dirichlet BVP state that the density is a constant plus a bounded singular function whose power exceeds $1/2$ for any corner angle in $(0, 2\pi)$ [63]; this is similar to the Laplace case [101]. For our $\mathbf{D} + \mathbf{S}$ formulation we observe a density behavior consistent with this. This might suggest choosing $\alpha = 1$ for any corner angle. In fact, for small (non-reentrant) angles we are able to reduce α somewhat without loss of accuracy, hence do so, to reduce the number of panels.

2. *Bent panel refinement.* Panels away from any corners are refined based on how well the smooth geometric properties are represented by the interpolants on their p Legendre nodes. We measure the accuracy of geometric representations by the interpolation errors of a set of test functions on a set of test points. First, we define the set of test functions $G = \{g_1, g_2, g_3, \dots\}$ to be approximated on

the panel Λ . The following list of functions are included in G whenever the necessary derivatives are available:

- $g_1(t) = Z(t)$, which resolves the geometry representation.
- $g_2(t) = |Z'(t)|$, which resolves arc length, recalling that arc length is

$$S = \int_{\Lambda} ds = \int_a^b |Z'(t)| dt$$

- $g_3(t) = \frac{|\text{Im}(Z''(t)/Z'(t))|^2}{|Z'(t)|}$, which resolves bending, since bending energy is

$$E = \int_{\Lambda} \kappa^2 ds = \int_a^b \left| \frac{Z'(t) \times Z''(t)}{|Z'(t)|^3} \right|^2 |Z'(t)| dt = \int_a^b \frac{\{\text{Im}(Z''(t)/Z'(t))\}^2}{|Z'(t)|} dt$$

Next, we define the test points to be the m equally spaced points on $[a, b]$, denoted $\tilde{t}_j^\Lambda, j = 1, \dots, m$, and let $t_j^\Lambda, j = 1, \dots, p$ be the Legendre nodes. Then for each $i = 1, 2, 3$, the relative error of approximating g_i is

$$\varepsilon_i = \frac{\|\tilde{\mathbf{g}}_i - \mathbf{M} \cdot \mathbf{g}_i\|}{\|\mathbf{g}_i\|},$$

where $\tilde{\mathbf{g}}_i := (g_i(\tilde{t}_1^\Lambda), \dots, g_i(\tilde{t}_m^\Lambda))$, $\mathbf{g}_i := (g_i(t_1^\Lambda), \dots, g_i(t_p^\Lambda))$, and \mathbf{M} is the $m \times p$ interpolation matrix from the Legendre nodes to the test points. The panels are refined until $\max_i \varepsilon_i < \epsilon^\beta$, where $\beta > 0$ is another power parameter, with default value $\beta = 1$.

The corner and bent panel refinement rules are applied to all panels recursively. The complete procedure is summarized in Algorithm 1.

Stage 3: Global closeness refinement. At this final stage, panels are further refined if they are (relatively) too close to any non-neighboring panels. Specifically,

Algorithm 1 Local geometric refinement

Require: The current panel $\Lambda = Z([a, b])$; tolerance ϵ ; corner information $C = \{t_j^c, \lambda_j, \alpha_j\}_{j=1}^k$; test function(s) $G = \{g_1, g_2, g_3, \dots\}$; β is the tolerance exponent for the test functions, default to be 1.

```
1: function REFINE( $Z([a, b]), \epsilon, C, G, \beta$ )
2:   Panel parametric length  $L = b - a$ 
3:   Panel arc length  $S = \int_a^b |Z'|$ 
4:   if geometry has corners then
5:     Let  $t_i^c$  be the corner closest to  $[a, b]$ 
6:     if  $L < \epsilon^{\alpha_i}$  or  $S < \epsilon^{\alpha_i}$  then
7:       return  $\{a, b\}$  ▷ panel length reached lower limit, do not split
8:     if  $t_i^c$  is close enough to the panel  $[a, b]$  then
9:       if  $a < t_i^c < b$  then  $s = t_i^c$  ▷ split right at the corner
10:      else if  $t_i^c < a$  then  $s = a + L/\lambda_i$  ▷ split towards the corner
11:      else if  $t_i^c > b$  then  $s = b - L/\lambda_i$  ▷ split towards the corner
12:    if split point  $s$  is not defined then
13:       $\mathbf{g}_i = g_i$  values at quadrature points
14:       $\tilde{\mathbf{g}}_i = g_i$  values at test points
15:       $\mathbf{M} =$  interpolation matrix from quadrature points to test points
16:       $\epsilon = \max_{g_i \in G} \|\tilde{\mathbf{g}}_i - \mathbf{M} \cdot \mathbf{g}_i\| / \|\mathbf{g}_i\|$  ▷ interpolation error of test function(s)
17:      if  $\epsilon > \epsilon^\beta$  then
18:         $s = (a + b)/2$  ▷ split in half
19:    if split point  $s$  is defined then
20:       $\mathbf{t}^1 = \text{REFINE}(Z([a, s]), \epsilon, C, G, \beta)$ 
21:       $\mathbf{t}^2 = \text{REFINE}(Z([s, b]), \epsilon, C, G, \beta)$ 
22:      return  $\mathbf{t}^1 \cup \mathbf{t}^2$  ▷ recursively refine panel
23:    return  $\{a, b\}$  ▷ do not split
```

let Λ^{left} and Λ^{right} be the two immediate neighboring panels of Λ , and define $\Gamma^{\text{far}} := \Gamma \setminus (\Lambda^{\text{left}} \cup \Lambda \cup \Lambda^{\text{right}})$ as all non-neighboring panels of Λ . Then the panel Λ is refined if $d(\Lambda, \Gamma^{\text{far}})$, its distance from Γ^{far} , is shorter than its arc length by a factor of 3 (see Line 4-10 of Algorithm 2); see Remark III.5 for an alternative, less restrictive, refinement criterion.

In practice, the distance $d(\Lambda, \Gamma^{\text{far}})$ can be approximated by $\min_{i,j} |y_i - y_j|$, where the minimum is searched among all pairs of nodes $y_i \in \Lambda$ and $y_j \in \Gamma^{\text{far}}$. A kd-tree algorithm [79] is used to accelerate this process for our large examples in Section 4.3, in which case the elliptical close neighborhood (3.27) is also replaced by $\bigcup_{i=1}^p B(Z_i, CS)$,

Algorithm 2 Global closeness refinement

Require: The refined panels $\Gamma = \bigcup_k \Lambda_k$ from Stage 2 (local geometric refinement).

```
1: function CLOSEREFINE( $\Gamma$ )
2:   Initialize the output set  $\mathbf{t} = \emptyset$ , which will contain the final panel endpoints
3:   Initialize the set of new endpoints  $\mathbf{t}^{\text{new}} = \{\text{endpoints of } \Gamma = \bigcup_k \Lambda_k\}$ 
4:   while  $\mathbf{t}^{\text{new}} \neq \emptyset$  do ▷ repeat until no further splitting
5:      $\mathbf{t} = \mathbf{t} \cup \mathbf{t}^{\text{new}}$ 
6:      $\mathbf{t}^{\text{new}} = \emptyset$ 
7:     Update panels  $\Gamma = \bigcup_k \Lambda_k$  based on  $\mathbf{t}$  ▷ ready for a new round of refinement
8:     for each panel  $Z([a, b]) \subset \Gamma$  do
9:       Locate  $\Lambda^{\text{left}}$  and  $\Lambda^{\text{right}}$ , the two immediate neighboring panels of  $Z([a, b])$ 
10:      Define  $\Gamma^{\text{far}} = \Gamma \setminus (\Lambda^{\text{left}} \cup Z([a, b]) \cup \Lambda^{\text{right}})$ 
11:      Compute the distance  $d = d(Z([a, b]), \Gamma^{\text{far}})$ 
12:      Calculate  $S = \text{arc length of } Z([a, b])$ 
13:      if  $\frac{1}{3}S > d$  then
14:         $\mathbf{t}^{\text{new}} = \mathbf{t}^{\text{new}} \cup \{\frac{a+b}{2}\}$  ▷ split in half
15:   return final panel endpoints  $\mathbf{t}$ 
```

the union of disks around each node on Λ , for convenience.

The above refinement process is applied to each panel from the output of the previous stage and repeats until no further splitting. The algorithm for this stage is summarized in Algorithm 2. We note that since our algorithm is panel-based, it is agnostic of whether two touching panels belong to the same boundary component or not. Hence this algorithm handles two situations simultaneously: the case of close-touching between different boundary components, as well as the case of “self-touching” where a boundary component is almost touching itself.

Remark III.4 (Expected convergence rate with corners). The above three stages involve two quantities—the panel order p , and a resulting number of panels per corner—both of which grow linearly with $\log 1/\epsilon$. However, N is the product of these two quantities, thus, in the presence of corners, one expects asymptotically $N = \mathcal{O}(\log^2 1/\epsilon)$ as $\epsilon \rightarrow 0$. In other words, the error converges *root exponentially* in N , i.e. as $\mathcal{O}(e^{-c\sqrt{N}})$. This matches the theoretical convergence rate for *hp*-BEM on polygons by Heuer–Stephan [32]. This rate has also recently been observed and proven for a geometrically

graded “method of fundamental solutions” approach to polygons by Gopal–Trefethen [22].

3.5 Numerical results and discussion

In what follows, numerical examples will be presented to test the overall solution scheme presented so far. In each example, we solve a Dirichlet problem in the domain exterior to the given geometries. The BIE formulation of the problem is $(\frac{1}{2} + \mathbb{S} + \mathbb{D})\sigma = f$ as described in Section 3.2.1.

The solution procedure is to first adaptively refine the representation of the geometry using our refinement scheme (Section 3.4), then the BIE is discretized using the special quadrature (Section 3.3) and solved for the density σ , and finally the solution $u = u_\infty + (\mathbf{S} + \mathbf{D})\sigma$ is evaluated everywhere in the exterior domain on a grid of spacing $\Delta x = 0.02$.

We mention that our solution scheme has been tested on boundary value problems with inhomogeneous boundary data extracted from an analytically known smooth flow u , and, as expected, achieves superalgebraic convergence. However, in the presence of corners, such smooth test problems do *not* involve the corner singularities that generically arise in physical problems. For this reason, we only present results on physical flows such as imposed uniform or linear shear flows. In all the examples, the exact solution is not known analytically; therefore, we use the finest grid solution as the reference solution.

Example 1. Smooth domain. This example tests our scheme on a linear shear flow around a smooth starfish-shaped island defined by the polar function $r(\theta) = 1 + 0.3 \cos 5\theta$, with no-slip boundary condition $u|_\Gamma = 0$ and $u_\infty(x_1, x_2) = (-x_2, 0)$ as $|x| \rightarrow \infty$. We have used $\beta = 0.8$ (Line 17 of Algorithm 1) for this problem to reduce N . In addition to the velocity field, we have also investigated the convergence

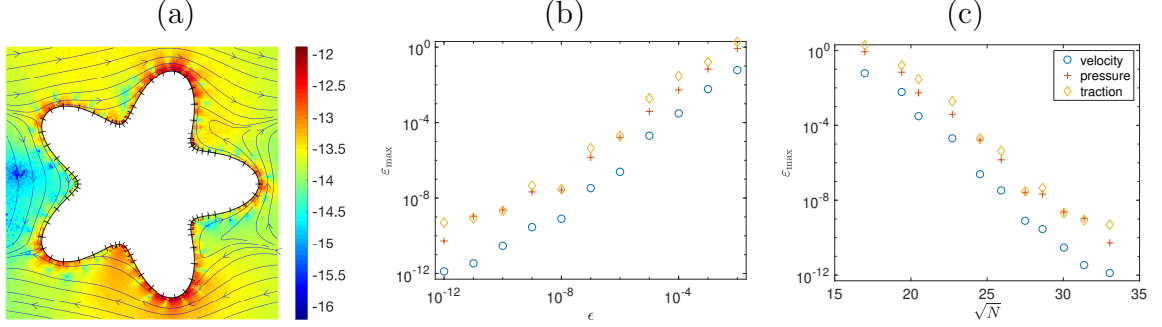


Figure 3.4: (a) Linear shear flow past a starfish-shaped island. Streamlines of the flow and panel endpoints (small segments) are shown. Color represents the \log_{10} error of the velocity computed under tolerance $\epsilon = 10^{-12}$ (resulted in $2N = 2184$ degrees of freedom). (b) Convergence of the maximum relative error ϵ_{\max} versus requested tolerance ϵ , where the traction field is computed along the $(1, 2)$ -direction. (c) Convergence of the maximum error ϵ_{\max} versus the square root of the number of nodes.

of the pressure field and the traction field in the $(1, 2)$ -direction, both of which are obtained using our close evaluation scheme (Section 3.3.3 and Appendix A). Our scheme achieved accuracies that are matching the requested tolerance (Figure 3.4b). All the solution fields converge super-algebraically with respect to the problem size (Figure 3.4c).

Example 2. Domain with corners. The smooth geometry in Example 1 is now replaced with a non-convex polygon. Figure 3.5 shows a linear shear flow around a “shuriken” domain with eight corners, the outer four of which are reentrant (with respect to Ω) corners of angle $\theta = 1.74\pi$. With $\alpha = 0.5$ for the flatter corners and $\alpha = 1.1$ for the sharper ones, our scheme achieved accuracies that are matching the requested tolerance (Figure 3.5b). Note that the convergence with respect to problem size is super-algebraic (Figure 3.5c), and consistent with root-exponential convergence, as expected for problems with corner singularities (see Remark III.4). We used the numerical solution obtained using $\epsilon = 10^{-10}$ as the reference solution.

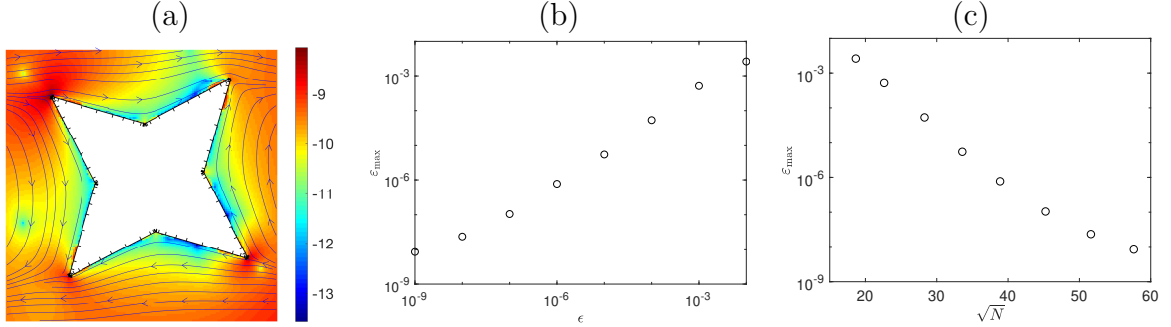


Figure 3.5: (a) Linear shear flow around a shuriken-shaped island with 8 corners. Streamlines of the flow and panel endpoints (small segments) are shown. Color represents the \log_{10} error of the velocity computed under tolerance $\epsilon = 10^{-10}$ (resulted in $2N = 6640$ degrees of freedom). (b) Convergence of the maximum error ϵ_{\max} versus requested tolerance ϵ . (c) Convergence of the maximum error ϵ_{\max} versus the square root of the number of nodes; root-exponential convergence would result in a straight line.

Example 3. Multiple polygonal islands. This example models a porous media flow through a collection of non-smooth, non-convex and closely packed boundaries: we set up 50 polygonal islands with a total number of 253 corners (Figure 3.6). The computational domain has width ≈ 8 and the closest distance between the polygons is about 10^{-2} . The background flow is the same as in the previous examples. With $\alpha = 0.75$ and $\lambda = 2$ for all corners, the convergence (Figure 3.6c) is similar to the single-polygon island example (Figure 3.5b–c), achieving more than 8 digits using approximately 800 degrees of freedom per corner. This demonstrates the robustness of our adaptive scheme, that is, the performance is as good for a more complex example as for a simple one.

Example 4. Artificial vascular network. We now turn to the example shown in Figure 3.2. We construct an artificial vascular network (with 378 corners) that mimics those observed in an eye of a zebra fish [4]. The flow in this network is driven by a uniform influx from the circular wall at the center and a uniform outflux at the outer circular wall, such that the overall volume is conserved; all other boundaries have a no-slip condition. We solve the BIE for this problem using GMRES with a block

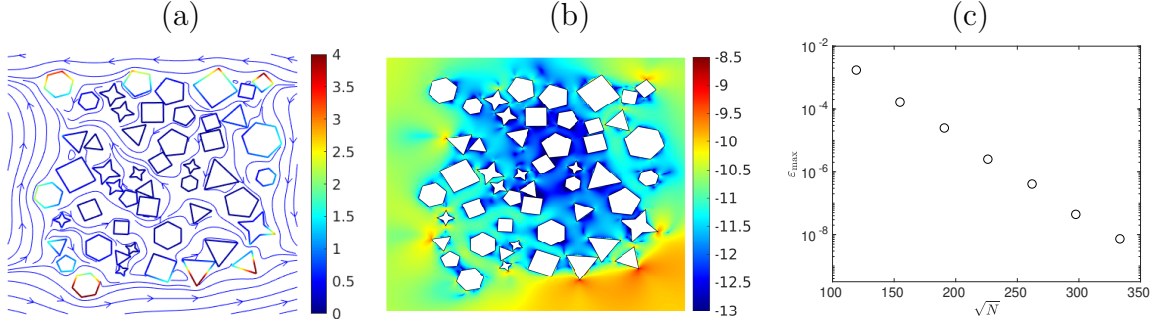


Figure 3.6: (a) Streamlines of a shear flow past 50 randomly generated polygonal islands with a total number of 253 corners. Color on the polygon boundaries indicate the magnitude of density σ . (b) \log_{10} of absolute error of the velocity, computed using $2N = 222140$ degrees of freedom. Error is measured on a 1000×1000 grid (spacing $\Delta x \approx 8.3 \times 10^{-3}$) by comparing to the solution obtained with $\epsilon = 10^{-10}$. (c) Convergence of the maximum error ϵ_{\max} versus the square root of the number of nodes.

diagonal preconditioner consisting of the diagonal panel-wise blocks of the BIE system itself (i.e., the self-evaluation blocks for each panel). The FMM is used for applying the matrix and for final flow evaluations; see Remark III.1. The sparse correction matrix (see Section 3.3.2) is applied via MATLAB’s single-threaded built-in matrix-vector multiplication; its rows have been precomputed as described in Section 3.3.4. All computations are done on an 8-core 4.0 GHz Intel Core i7 desktop.

Table 3.1 shows, for various tolerance ϵ , the relative L_2 -error ϵ_{L_2} , the relative maximum error in velocity ϵ_{\max} , the total number of panels used n_Λ , the number of degrees of freedom $2N$, memory (RAM) usage, the number of GMRES iterations and time used, the setup time for precomputing the close-correction matrices, and the percentage time for applying Stokes FMM during GMRES. Several observations are in order:

- 1) Both ϵ_{L_2} and ϵ_{\max} decay super-algebraically with the number of degrees of freedom; this data is plotted in Figure 3.7a. The closeness between ϵ_{L_2} and ϵ_{\max} shows that our scheme has achieved high accuracies near the sharper reentrant corners (hard) that are similar to those near the smooth edges (easy). This error analysis remains

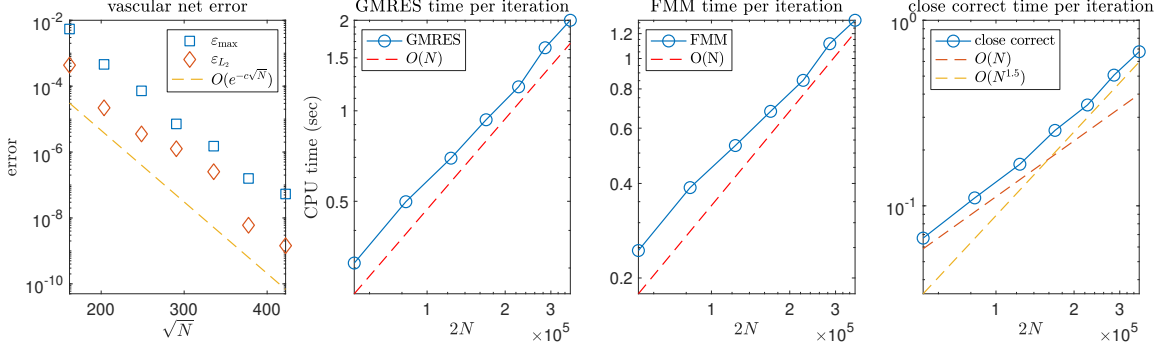


Figure 3.7: Error and timing of solving the vascular network BVP. (a) Convergence of errors in log-linear scale. (b) Log-log scale plot of the total CPU time per GMRES iteration, which consists of the FMM time, shown in (c), and the close correction time, shown in (d).

valid even in the zoomed-in high-resolution error plots in Figure 3.2. Furthermore, the convergence performance of this example is the same as the previous two examples—we achieved more than 8 digits, with a ratio $\frac{\text{degrees of freedom}}{\#\text{corners}} \approx 943$, which is similar to the ratios in Example 2 (830) and Example 3 (878). This once again demonstrates the robustness of our overall scheme to problem complexity.

- 2) The number of GMRES iterations increases only because we have requested smaller tolerance. Each additional digit needs about 100 more iterations. The GMRES convergence *rate* is stable, which demonstrates that our second kind BIE formulation is well-conditioned even in the presence of corners.
- 3) The fact that the Stokes FMM time is the main cost shows that our algorithm has achieved close to optimal efficiency. The slight decrease of the percentage FMM times at smaller ϵ is due to the fact that the FMM time grows only linearly with N , while the close evaluation matrix-vector multiplication time grows like $O(N^{3/2})$. The latter estimate is obtained as follows. The number of matrix-vector multiplications grows like $O(n_\Lambda) = O(\log 1/\epsilon) = O(p)$, where each matrix-vector product takes $O(p^2)$ time. Note that $N = n_\Lambda \times p = O(p^2)$, so the total close evaluation time grows as $O(p^3) = O(N^{3/2})$. (See Figure 3.7b–d.)

ϵ	ϵ_{L_2}	ϵ_{\max}	n_{pan}	$2N$	RAM used (gb)	GM- RES itera- tion	GM- RES time (s)	setup time (s)	% FMM time
1e-03	4.34e-04	5.43e-03	6549	52392	2.3	796	248	48	78.50
1e-04	2.20e-05	4.58e-04	8281	82810	2.9	919	458	63	77.86
1e-05	3.55e-06	7.21e-05	10301	123612	3.7	1091	759	84	75.92
1e-06	1.26e-06	7.15e-06	12061	168854	5.0	1282	1197	106	72.69
1e-07	2.53e-07	1.51e-06	14079	225264	6.9	1390	1670	135	70.91
1e-08	6.01e-09	1.58e-07	15839	285102	9.0	1501	2433	164	68.77
1e-09	1.44e-09	5.34e-08	17829	356580	12	1597	3195	204	66.16

Table 3.1: *Results and statistics of solving the BVP in the vascular network in Figure 3.2 for various tolerance ϵ . Errors ϵ_{\max} and ϵ_{L_2} are measured on a 2160×2160 grid (spacing $\approx 2.5 \times 10^{-3}$) by comparing to the solution obtained from $\epsilon = 10^{-10}$. CPU time and RAM used are measured using [7].*

Example 5. Uniform versus adaptive for close-to-touching curves. Finally, to demonstrate the advantage of using an adaptive scheme over a uniform discretization, let us consider a uniform flow past two close-to-touching disks (Figure 3.8). The background flow is a constant $u_\infty = (1, 0)$, the separation is $d = 10^{-6}$, and the radii 1 and 0.1. For the uniform-resolution scheme we use a global periodic trapezoid grid on each circle, where, in order to have similar node spacings, the larger circle has 9 times as many points as the smaller one. Here, global close-evaluation is done using the spectrally accurate quadrature from [8]. On the other hand, the adaptive quadrature uses a grid that is determined by our adaptive refinement scheme of Section 3.4, with one modification that improves the scaling in the number of refined panels (see Remark III.5). We observe that, for more than 4 accurate digits, the number of unknowns required by the adaptive scheme is much less than that of the uniform-resolution scheme (Figure 3.8). The smoothness of the density function discussed in the remark below suggests that, at fixed ϵ , the uniform scheme (and also the original refinement scheme) needs $N = O(1/\sqrt{d})$ unknowns, whereas the modified adaptive scheme needs only $N = O(\log(1/d))$. The latter is close to optimal, and is what we recommend for closely-interacting curves. (See also Examples 4 and 6 in [10] for a

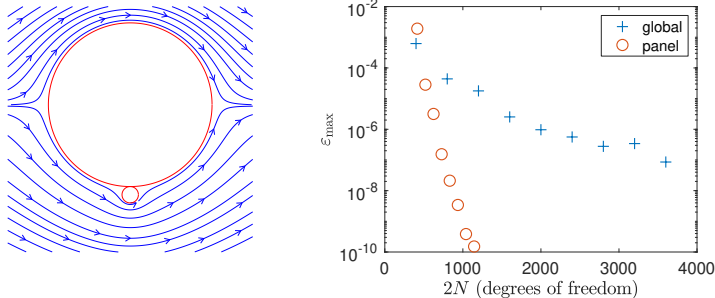


Figure 3.8: Convergence of a uniform flow past two touching disks that are $d = 10^{-6}$ apart, and whose radii are 1 and 0.1. The required number of unknowns in the adaptive scheme is much less than the global scheme with uniform resolution; see Example 5.

“globally adaptive” variant.)

Remark III.5 (Refinement at close-to-touching smooth surfaces). For viscous flow in the region between two smooth curves separated by a small distance d , asymptotic analysis gives that the width of the “bump” in fluid force scales as $O(\sqrt{d})$ [73]. By dimensional analysis, if the sum of the two curvatures of the surfaces near the contact point is κ , then the width in fact scales as $O(\sqrt{d/\kappa})$. Assuming that this also applies to the density σ , this suggests a looser criterion for refinement: panels should be refined only when they are longer than this width scale. This allows panels to come much closer than their length, without being refined. In the case of close smooth curves, the test in line 13 of Algorithm 2 can thus be modified to $(c'\sqrt{\kappa}S)^2 > d$. We find that the constant $c' = 0.7$ achieves the requested tolerance. The resulting N can be estimated as follows. Setting $\kappa = 1$ for simplicity, a generic local model of the separation is $h(t) \approx d + t^2$ as a function of parameter t , and $n_\Lambda = O(\int_{-1}^1 dt/S(h(t)))$ where $S(h)$ is the local panel size as a function of separation. The original refinement scheme, $S(h) = O(h)$, thus gives $n_\Lambda = O(d^{-1/2})$, whereas the modified $S(h) = O(\sqrt{h})$ gives $n_\Lambda = O(\log(1/d))$.

3.6 Conclusions

We have presented a set of panel quadrature rules for accurate evaluation of single- and double-layer Stokes potentials and their associated pressure and traction boundary integrals. They can be used for targets that are either on or off the boundary, and can be located arbitrarily close to it. In addition, we formulated an adaptive panel refinement procedure that sets the length of panels on the boundary (“ h -adaptivity”) and the overall degree of approximation p (“ p -adaptivity”) required to achieve a user-prescribed tolerance. We demonstrated via numerical experiments that our algorithm achieves super-algebraic convergence even for complex geometries with corners, and that the CPU time grows linearly with problem size, and is dominated by the cost of FMMs for large-scale problems. More sophisticated quadratures and techniques designed for corner singularities, such as the RCIP [29, 30] (or the work of [63]), are expected to further improve the performance of our BIE solver. It is also expected that adapting p on a per-panel basis (i.e. full hp -adaptivity) would reduce the total number of degrees of freedom needed, although only by a factor less than two. Applications of our work include providing design tools for rapid prototyping of microfluidic chips (for cell sorting, mixing or other manipulations e.g., [39]), shape optimization (e.g., [14]) and simulating cellular-level blood flow in microvasculature.

We envision building a fast 2D particulate flow software library by utilizing the algorithms developed in this work for the fixed complex geometries (such as microfluidic chips or vascular networks) and our global close evaluation schemes developed in [8] for the moving rigid or deformable particles (such as colloids, drops or vesicles). Another key ingredient would be a *fast direct solver* for solving the BIEs on the fixed geometries, similar to that developed in [50], wherein the boundary integral operators were compressed by exploiting their low-rank structures, inverted as a precomputation step, and applied at an optimal $\mathcal{O}(N)$ cost at every time-step of the particulate flow simulation as particles move through the fixed geometry. One open

research question in this context is: *Can we update the compressed representations as the boundary panels are refined (or coarsened) without rebuilding them?* A similar question was recently investigated in [103], where the authors report a $3\times$ speedup when locally perturbing the geometry. We plan to explore their approach and report its performance in the context of our adaptive panel refinement procedure.

CHAPTER IV

Electrohydrodynamics of Deflated Vesicles: Budding, Rheology and Pairwise Interactions

Preamble. In this chapter, we will present one specific scientific application – *vesicle electrohydrodynamics* (EHD), which studies how vesicles react and interact with each other under the influence of electric signals. Electric stresses often drive vesicles to come very close to each other or near the wall boundaries, therefore close evaluation schemes such as those in Chapter II and III are essential for any robust EHD simulations. This chapter focuses on the vesicle EHD in an open domain, for which the global scheme (Chapter II) is effective and sufficient. The adaptive scheme (Chapter III) will be used in the future for studies with more complex boundary conditions. This is a joint work with Shravan Veerapaneni that is published in [94].

4.1 Introduction

Understanding the electrohydrodynamics (EHD) of the so-called giant unilamellar vesicles (GUVs) has received much attention in the recent past [59, 89]. Vesicles share the same structural component of a biological cell, the bilipid membrane, and hence their EHD has been a paradigm for understanding how general biological cells behave under an electric field. The dynamics of this system is characterized by a competition

between viscous, elastic, and electric stresses on the individual membranes and the nonlocal hydrodynamic interactions. Studying the microstructural response of isolated vesicles and vesicle pairs subjected to electric fields can bring insights into the macroscopic properties of vesicle suspensions. Several recent theoretical and numerical works have focused on isolated, nearly spherical (or circular) vesicles; however, the dynamics of highly deformable deflated vesicles as well as the pair-wise dynamics of vesicle suspensions remains largely unexplored. The primary focus of this work is to develop a robust numerical scheme to enable study of these dynamics.

Theoretical investigation of vesicle EHD has been done via small deformation theory [90, 76] and semi-analytic studies using spheroidal models [102, 55]. Numerical solution of the coupled electric, elastic and hydrodynamic governing equations were computed using the boundary integral equation (BIE) methods [51, 72, 87] and immersed interface or immersed boundary methods [42, 35]. Advantages of BIE methods are well-known—exact satisfaction of far-field boundary conditions eliminating the need for artificial boundary conditions, reduction in dimensionality leading to reduced problem sizes, and well-conditioned linear systems through carefully chosen integral representations.

All of the aforementioned works, however, considered EHD of a single vesicle only. Vesicles are known to segregate when subjected to electric fields [69], thereby, pose significant challenges for direct numerical simulations. In the case of BIE methods, for instance, the integral representations of the hydrodynamic and electric interaction forces become nearly-singular, requiring specialized quadratures. Domain discretization methods, on the other hand, require finer meshes (locally, in the case of adaptive methods), worsening the conditioning issue of linear systems and increasing the overall computational expense.

Leveraging on our recently developed spectrally-accurate algorithm for evaluating nearly singular integrals [8] and the second-kind BIE formulation for three-

dimensional vesicle EHD [87], we develop a BIE method for simulating multiple vesicle EHD in this work. We apply it to analyze the pairwise interactions in a monodisperse suspension. We provide the integral equation formulation and the description of our numerical method in §4.2, followed by analysis and discussion of the results in §4.3.

4.2 Problem formulation

4.2.1 Governing equations

Let us first consider a single vesicle suspended in a two-dimensional unbounded viscous fluid domain, subjected to an imposed flow $\mathbf{v}_\infty(\mathbf{x})$, for any $\mathbf{x} \in \mathbb{R}^2$. The vesicle membrane is denoted by γ . Assume that the fluids interior and exterior to γ have the same viscosity μ and the same dielectric permittivity ϵ while their conductivities differ, given by σ_i and σ_e , respectively. In the vanishing Reynolds number limit, the governing equations for the ambient fluid can then be written as:

$$-\nabla p + \mu \Delta \mathbf{v} = 0 \quad \text{in } \mathbb{R}^2 \setminus \gamma, \quad (4.1a)$$

$$\nabla \cdot \mathbf{v} = 0 \quad \text{in } \mathbb{R}^2 \setminus \gamma, \quad (4.1b)$$

$$\mathbf{v}(\mathbf{x}) \rightarrow \mathbf{v}_\infty(\mathbf{x}) \quad \text{as } \|\mathbf{x}\| \rightarrow \infty. \quad (4.1c)$$

The fluid motion is coupled to the membrane motion via the kinematic boundary condition $\dot{\mathbf{x}} = \mathbf{v}$ on γ , where \mathbf{x} is a material point on the membrane. Using the boundary integral equation formulation, we can now write the membrane evolution equation by combining the kinematic condition with the governing equation (4.1) as [88],

$$\dot{\mathbf{x}} = \mathbf{v}_\infty(\mathbf{x}) + \int_{\gamma} G_s(\mathbf{x} - \mathbf{y}) \mathbf{f}_{hd}(\mathbf{y}) d\gamma(\mathbf{y}), \quad \nabla_\gamma \cdot \dot{\mathbf{x}} = 0, \quad (4.2)$$

where \mathbf{f}_{hd} is the hydrodynamic traction jump across the membrane and G_s is the free-space Green's function for the Stokes equations, given by

$$G_s(\mathbf{x} - \mathbf{y}) = \frac{1}{4\pi\mu} \left(-\log\|\mathbf{x} - \mathbf{y}\| \mathbf{I} + \frac{(\mathbf{x} - \mathbf{y}) \otimes (\mathbf{x} - \mathbf{y})}{\|\mathbf{x} - \mathbf{y}\|^2} \right). \quad (4.3)$$

The second equation in (4.2) expresses the local inextensibility constraint on the membrane.

For a given vesicle configuration, \mathbf{f}_{hd} can be evaluated by performing a force balance at the membrane. The elastic forces acting on the membrane are comprised of the bending and the tension forces, defined respectively as

$$\mathbf{f}_b = \kappa_B \left(\kappa_{ss} + \frac{\kappa^3}{2} \right) \mathbf{n}, \quad \mathbf{f}_\lambda = (\lambda \mathbf{x}_s)_s, \quad (4.4)$$

where κ_B is the bending modulus, κ is the curvature, s is the arclength parameter, \mathbf{n} is the outward normal to γ and the tension λ acts as a Lagrange multiplier to enforce the inextensibility constraint. A force balance at the membrane yields $\mathbf{f}_{hd} = \mathbf{f}_b + \mathbf{f}_\lambda - \mathbf{f}_{el}$, where \mathbf{f}_{el} is the electric force that is determined by solving for the electric potential.

In the *leaky-dielectric* model, the electric charges are assumed to be present only at the interface and not in the bulk. Let $\phi(\mathbf{x})$ be the electric potential at \mathbf{x} , so that $\mathbf{E} = -\nabla\phi$. Assuming that the vesicle membrane is charge-free and has a conductivity G_m , a capacitance C_m , the boundary value problem for the electric potential can be summarized as [76]:

$$-\Delta\phi = 0 \quad \text{in } \mathbb{R}^2 \setminus \gamma, \quad (4.5a)$$

$$-\nabla\phi(\mathbf{x}) \rightarrow \mathbf{E}_\infty(\mathbf{x}) \quad \text{as } \|\mathbf{x}\| \rightarrow \infty, \quad \llbracket \mathbf{n} \cdot (\sigma \nabla\phi) \rrbracket_\gamma = 0, \quad \llbracket \phi \rrbracket_\gamma = V_m, \quad (4.5b)$$

$$C_m \dot{V}_m + G_m V_m = -\mathbf{n} \cdot (\sigma_i \nabla\phi_i) \quad \text{on } \gamma. \quad (4.5c)$$

Here, \mathbf{E}_∞ is the imposed electric field, $\llbracket \cdot \rrbracket_\gamma$ denotes the jump across the interface

(e.g., $[[\sigma]]_\gamma = \sigma_i - \sigma_e$) and V_m is the transmembrane potential. The electric force on the membrane is then defined by $\mathbf{f}_{el} = [[\mathbf{n} \cdot \Sigma^{el}]]_\gamma$, where the Maxwell stress tensor, $\Sigma^{el} = \epsilon \mathbf{E} \otimes \mathbf{E} - \frac{1}{2} \epsilon \|\mathbf{E}\|^2 \mathbf{I}$. Therefore, we need to determine the electric field on both sides of the membrane by solving (4.5) to evaluate \mathbf{f}_{el} .

Since we are only interested in interfacial variables and (4.5) is a linear partial differential equation, we can recast it as a BIE with the unknowns residing only on the interface. We will employ an *indirect* integral equation formulation to solve for the electric potential ϕ . Assume that the electric potential in the domain interior and exterior of the membrane is given by [87],

$$\phi(\mathbf{x}) = \phi_\infty(\mathbf{x}) + \mathcal{S}[q](\mathbf{x}) - \mathcal{D}[V_m](\mathbf{x}) \quad (4.6)$$

where the membrane charge density, $q = [[\partial\phi/\partial\mathbf{n}]]_\gamma$ and the Laplace single and double layer integral operators are defined by

$$\mathcal{S}[q](\mathbf{x}) = \int_\gamma G(\mathbf{x} - \mathbf{y})q(\mathbf{y}) d\gamma(\mathbf{y}) \quad \text{and} \quad \mathcal{D}[V_m](\mathbf{x}) = \int_\gamma \frac{\partial G(\mathbf{x} - \mathbf{y})}{\partial \mathbf{n}(\mathbf{y})} V_m(\mathbf{y}) d\gamma(\mathbf{y}), \quad (4.7)$$

respectively. Here $G(\cdot)$ is the Laplace fundamental solution in the free space.

Note that, by construction, equation (4.6) implies $[[\phi]]_\gamma = V_m$ since the single layer potential is continuous across γ . Applying the current continuity condition and using the standard jump conditions for the Laplace layer potentials, we arrive at the *second-kind* integral equation for the unknown q :

$$\left(\frac{1}{2} + \eta \mathcal{S}'\right) q = \eta \mathbf{E}_\infty \cdot \mathbf{n} + \eta \mathcal{D}'[V_m], \quad (4.8)$$

where $\eta = (\sigma_i - \sigma_e)/(\sigma_i + \sigma_e)$, \mathcal{S}' and \mathcal{D}' denote the normal derivatives of the single and double layer potentials respectively. Furthermore, the interfacial conditions

$[[\partial\phi/\partial\mathbf{n}]_\gamma = q$ and $[[\sigma\partial\phi/\partial\mathbf{n}]_\gamma = 0$ imply that $-\mathbf{n} \cdot (\sigma_i \nabla \phi_i) = (\sigma_i \sigma_e / (\sigma_i - \sigma_e)) q$. Substituting this result in (4.5c) and using (4.8), we arrive at the following integro-differential equation for the evolution of V_m :

$$C_m \dot{V}_m + G_m V_m = \frac{\sigma_i \sigma_e}{\sigma_i + \sigma_e} \left(\frac{1}{2} + \eta \mathcal{S}' \right)^{-1} (\mathbf{E}_\infty \cdot \mathbf{n} + \mathcal{D}'[V_m]). \quad (4.9)$$

The steps involved within a time-stepping procedure for the electric problem for a given vesicle shape can now be summarized as follows: update V_m using (4.9), which also gives q since the right-hand side of (4.9) is just $(\sigma_i \sigma_e / (\sigma_i - \sigma_e)) q$, then evaluate the membrane electric force \mathbf{f}_{el} by computing \mathbf{E}_i and \mathbf{E}_e using (4.6).

Finally, the formulation generalizes to the two- (or multiple-) vesicle case in a trivial manner. Let γ now denote the union of the vesicle membranes i.e., $\gamma = \bigcup_{i=1}^2 \gamma_i$, where γ_i is the boundary of the i -th vesicle. Then, the definition of the boundary integral operators introduced earlier hold as is; for example,

$$\mathcal{S}[q](\mathbf{x}) = \int_{\gamma} G(\mathbf{x} - \mathbf{y}) q(\mathbf{y}) d\gamma(\mathbf{y}) := \sum_{j=1}^2 \int_{\gamma_j} G(\mathbf{x} - \mathbf{y}) q(\mathbf{y}) d\gamma_j(\mathbf{y}). \quad (4.10)$$

4.2.2 Numerical Method

We now describe a numerical scheme to solve the coupled integro-differential equations for the evolution of vesicle position (4.2) and its transmembrane potential (4.9). It directly follows from ideas introduced in [88], [8] and [87]. Each vesicle boundary is parametrized by a Lagrangian variable $\alpha \in [0, 2\pi]$ and a uniform discretization in α is employed. Derivatives of functions defined on the boundary are then computed using spectral differentiation in the Fourier domain, accelerated by the fast Fourier transform.

Evaluating boundary integrals. We use the standard periodic trapezoidal rule for computing boundary integrals that are smooth (e.g., the double-layer potential

defined in (4.7)), which yields spectral accuracy. On the other hand, we discretize the weakly singular operators such as the single-layer potential defined in (4.7) using a spectrally-accurate Nyström method (with periodic Kress corrections for the log singularity, ([44], Sec. 12.3)). The same method is also applied for computing the Stokes single-layer potential (4.2).

The operator $\mathcal{D}'[\cdot]$ requires special attention as its kernel is *hyper-singular*. We employ the following standard transformation ([34]) to turn it into a weakly singular integral:

$$\begin{aligned} \mathcal{D}'[V_m](\mathbf{x}) &= \frac{\partial}{\partial \mathbf{n}(\mathbf{x})} \int_{\gamma} \frac{\partial G(\mathbf{x} - \mathbf{y})}{\partial \mathbf{n}(\mathbf{y})} V_m(\mathbf{y}) d\gamma \\ &= \frac{\partial}{\partial s(\mathbf{x})} \left(\int_{\gamma} G(\mathbf{x} - \mathbf{y}) \frac{\partial V_m(\mathbf{y})}{\partial s(\mathbf{y})} d\gamma \right) \quad \forall \mathbf{x} \in \gamma. \end{aligned} \quad (4.11)$$

The surface gradients, $\partial/\partial s(\mathbf{x})$ and $\partial/\partial s(\mathbf{y})$, are computed via spectral differentiation.

Lastly, when the vesicles are located arbitrarily close to each other, the boundary integrals evaluating the interaction forces becomes *nearly-singular*. For example, consider the integral,

$$\int_{\gamma_1} G(\mathbf{x} - \mathbf{y}) q(\mathbf{y}) d\gamma_1(\mathbf{y}), \quad \text{where } \mathbf{x} \in \gamma_2. \quad (4.12)$$

The periodic trapezoidal rule loses its uniform spectral convergence in evaluating this integral as \mathbf{x} approaches γ_1 ; moreover, the singular quadrature rule is also ineffective for this integral. These inaccuracies, in turn, may lead to numerical instabilities and breakdown of the simulation. To remedy this problem, we employ the recently developed close evaluation scheme of [8] whenever vesicles are located closer than a cutoff distance (which is heuristically chosen to be five times the minimum spacing between the nodes, the so-called “5*h*-rule”). This scheme achieves spectral accuracy in evaluating (4.11), regardless of the distance of \mathbf{x} from γ_1 . We use this scheme to

accurately evaluate the Stokes layer potential in (4.2) as well.

Time-stepping scheme. The numerical stiffness associated with the bending force on the vesicle membranes is overcome by using the semi-implicit scheme proposed in [88] to discretize (4.2) in time. Following [51] and [87], we treat the electric force on the membrane explicitly, thereby, decoupling the evolution equations (4.2) and (4.9). Then, we use a semi-implicit scheme to evolve the transmembrane potential independently, which we describe next.

Let Δt be the time-step size, $V_m^n(\mathbf{x})$ be the transmembrane potential at time $n\Delta t$ at a point \mathbf{x} on the membrane. Our semi-implicit time-stepping scheme for (4.9) is given by

$$C_m \frac{V_m^{n+1} - V_m^n}{\Delta t} + G_m V_m^{n+1} = \frac{\sigma_i \sigma_e}{\sigma_i + \sigma_e} \left(\frac{1}{2} + \eta \mathcal{S}' \right)^{-1} (\mathbf{E}_\infty \cdot \mathbf{n} + \mathcal{D}' V_m^{n+1}), \quad (4.13)$$

where the boundary integral operators are treated explicitly i.e., evaluated using the boundary position at $n\Delta t$. This linear system for the unknown V_m^{n+1} is solved using an iterative method (GMRES).

4.3 Results and discussions

We now turn to analyzing the simulation results obtained using the numerical method outlined above. We first compare our results on single vesicle EHD with those obtained in prior studies as well as present some new insights on dynamics and rheology of dilute suspensions, followed by analysis of pairwise dynamics. Let A and L denote the area and perimeter of the vesicle. Setting the characteristic length scale as $a = L/2\pi$, we characterize our results on the following six nondimensional parameters,

reduced volume: $\Delta = 4\pi A/L^2,$

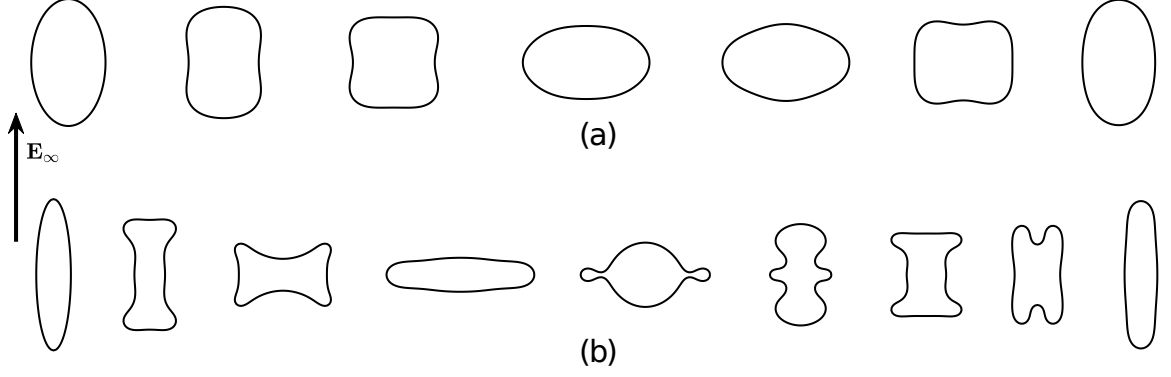


Figure 4.1: Snapshots from two different simulations of a single vesicle subjected to an external electric field, with $\Lambda = 0.1$, $G = 0$, $Ca = 0$ and (a) $\Delta = 0.9$, $\beta = 3.2$ and (b) $\Delta = 0.5$, $\beta = 12.8$. While the vesicle undergoes a prolate-oblate-prolate transition, the transient “square-like” shapes observed here (in (a)) and in prior numerical studies cannot be observed when the reduced volume is lowered. Instead, to sustain the electric compression forces, the vesicle forms buds as it undergoes the POP transition (more details on this phase are shown in Figure 4.2).

<i>conductivity ratio:</i>	$\Lambda = \sigma_i / \sigma_e,$
<i>membrane conductivity:</i>	$G = aG_m / \sigma_e,$
<i>electric field strength:</i>	$\beta = \epsilon \mathbf{E}_\infty ^2 a C_m / \mu \sigma_e,$
<i>capillary number:</i>	$Ca = \mu \dot{\gamma} a^3 / \kappa_B,$
<i>bending rigidity:</i>	$\chi = C_m \kappa_B / \sigma_e \mu a^2,$

where $\dot{\gamma}$ is the shear rate e.g., for imposed linear shear flow, we have $\mathbf{v}_\infty(\mathbf{x}) = (\dot{\gamma}x_2, 0)$. In all the simulations, the time is non-dimensionalized by the bending relaxation timescale $t_{\kappa_B} = \mu a^3 / \kappa_B$ and the bending rigidity, $\chi \approx 0.08$.

4.3.1 Isolated vesicle EHD: transition from squaring to budding in POP

When an arbitrarily shaped vesicle is subjected to uniform electric field, it is known to transform into either a prolate shape or an oblate shape at equilibrium [68, 70]. Since ours is a 2D construct, we refer to ellipses whose major axis aligns with the electric field direction as “prolates”; similarly, those whose minor axis aligns as

“oblates”. A classical observation in vesicle EHD studies is the *prolate-oblate-prolate* (POP) transition that arises in certain parameter regimes. Figure 4.1(a) illustrates the POP transition simulated using our numerical method.

Three conditions are generally required for a vesicle to undergo POP transition: 1) G is very small so that the vesicle membrane acts more like a capacitor than a conductor, 2) Λ is less than one and 3) β is strong enough. Since $\Lambda < 1$, charges accumulate faster on the membrane exterior initially, thereby, the vesicle appears to be negatively charged at the top and positively charged at the bottom, leading to a compressional force from the applied electric field and the vesicle transitions from a prolate to an oblate shape. At longer times, once the membrane, acting as a capacitor, is fully charged, the apparent charge becomes zero and the vesicle transforms back into a prolate shape, which minimizes the electrostatic energy [52].

A notable feature of the POP transition is the *squaring effect*—a transient shape of the vesicle with four smoothed corners (as can be observed in Figure 4.1(a))—which attracted attention of researchers due to its implications on electroporation. Since the reduced volume of a square is around 0.785, a question naturally arises: What transient shapes would a vesicle with much lower reduced volume assume? In Figure 4.1(b), we illustrate the POP transition of a vesicle with $\Delta = 0.5$. Since the fluid incompressibility acts to preserve its enclosed area, the vesicle forms small protrusions or “buds” to sustain the electrical compression forces. Figure 4.2 shows more details of this bud formation phase. The tension becomes negative, as expected, in the neck region of the buds. These intermediary shapes are reminiscent of those obtained by growing microtubules within the vesicles [20]; the notable feature here, however, is that only body forces are applied as opposed to local microtubule-membrane forces.

We further characterize the POP mechanism in Figure 4.3 for different reduced volumes. In all the cases, we observe that there exists some critical field strength β_0 for POP transition to happen (e.g., from the figure, for $G = 0$, $\beta_0 \approx \{1.9, 2.6, 5.1\}$)

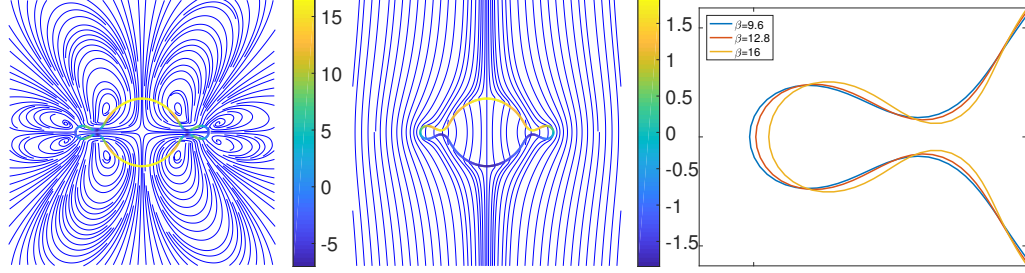


Figure 4.2: Streamlines (left) and electric field lines (middle) plotted at the moment when the vesicle with $\Delta = 0.5$ shown in Figure 4.1(b) forms buds while undergoing POP transition. In the left figure, the membrane color indicates the magnitude of tension while on the middle figure, it indicates the magnitude of the transmembrane potential. The right figure gives a closer look at the *narrowest* buds formed under different β 's, where the times correspond to this state for $\beta = 9.6, 12.8$ and 16 are $t = 0.253, 0.216$ and 0.184 , respectively. The neck of the buds becomes narrower as β increases.

corresponding to $\Delta = \{0.9, 0.8, 0.6\}$ respectively). On the other hand, when the field strength is weak, the vesicle remains a prolate and when the membrane conductivity is high, it transitions to an equilibrium oblate shape. These results are in qualitative agreement with [52], where similar phase diagrams were presented but only for higher reduced volume vesicles. Thus the phase diagrams in Figure 4.3 show that the POP mechanism works consistently for different Δ .

Finally, in the case when $\Lambda > 1$, the EHD forces act to extend the vesicle and it remains a prolate throughout the simulation.

4.3.2 Electro-rheology in the dilute limit

We next look at the combined effect of an imposed shear flow and a DC electric field on a single vesicle. In the presence of both fields, the dynamics is characterized by a competition between the electrical and hydrodynamical shear stresses and the migration of electric charges along the vesicle membrane.

Figure 4.4 shows the rheological properties of a vesicle subjected to an applied linear shear and an applied uniform electric field. In this case, where the membrane has

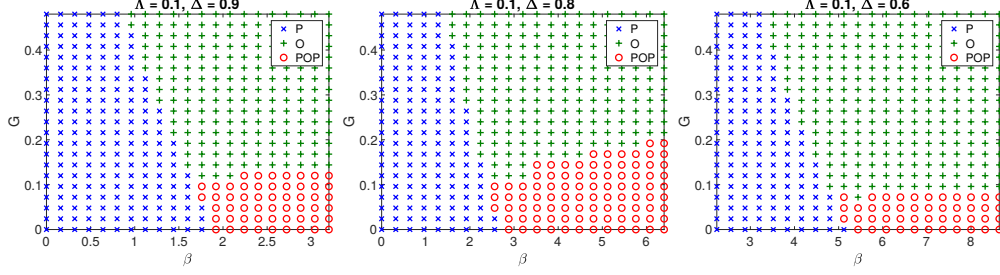


Figure 4.3: Phase diagrams of vesicle dynamics for different reduced volumes as a function of the membrane conductivity G and electric field strength β . Here, the different phases of the dynamics are indicated by O when the vesicle remains oblate for all times or P when it remains prolate or POP when it transitions from prolate to oblate to prolate shapes. For all the cases, the conductivity ratio Λ is set to 0.1, $Ca = 0$.

non-zero G , we observe that the vesicles with different reduced volumes all stabilize into a tank-treading motion and that the tank-treading speed and angle of inclination are affected nonlinearly by the conductivity ratio Λ . Note that as Λ is increased, the vesicle tries to align with the electric field direction and away from the direction of shear, presenting higher resistance to the imposed flow and hence leading to higher effective viscosity. Here, the effective viscosity $[\mu]$ is computed using the usual formula [67]:

$$[\mu] := \frac{1}{\dot{\gamma}\mu(T_e - T_i)} \int_{T_i}^{T_e} \langle \sigma_{12}^p \rangle dt, \quad \text{where} \quad \langle \sigma^p \rangle = \frac{1}{A} \int_{\gamma} (\mathbf{f}_b + \mathbf{f}_\lambda - \mathbf{f}_{el}) \otimes \mathbf{x} ds, \quad (4.14)$$

A is the area of the vesicle and σ^p represents the perturbation in the stress due to membrane forces. After the vesicle reaches a steady-state, the effective viscosity is measured over an arbitrary time interval $[T_i, T_e]$.

We further characterize the rheology in Figure 4.5 by plotting the effective viscosity as Δ is varied. Highly deflated vesicles prominently display shear-rate and β -dependent rheology since their shapes at equilibrium tank-treading dynamics are different, thereby, presenting varied resistance to applied shear.

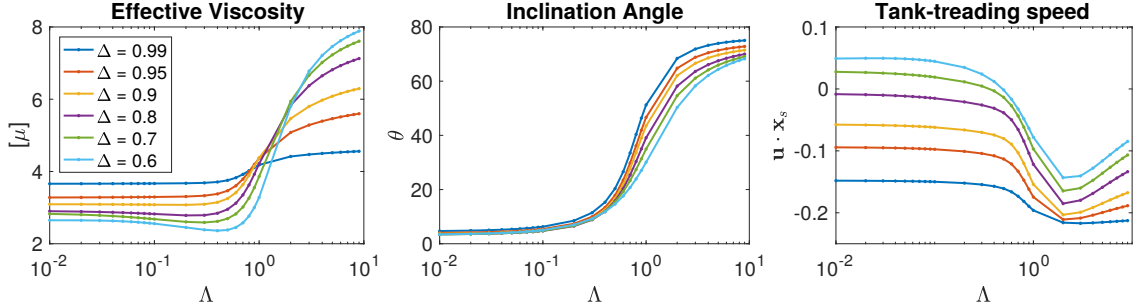


Figure 4.4: Single vesicle rheology when $G = 4$, $\beta = 6.4$ and $Ca = 10$. Plots of the effective viscosity (left), angle of inclination (middle) and the tangential velocity (right) when a vesicle is suspended in a linear shear flow as a function of the conductivity ratio. We can observe that the inclination angle increases as Λ is increased i.e., the vesicle tries to align with the electric field direction and away from the direction of shear. Thereby, it presents more resistance to imposed flow, leading to higher effective viscosity. One remarkable effect of low reduced volume, as is evident from the right panel, is that the vesicle tank-treads in the opposite direction compared to high reduced volume vesicles when Λ is small.

In the case when G is set to zero, the rheological behavior becomes much more complex, primarily because of the tendency of vesicles to undergo a POP transition while at the same time tank-tread due to the applied shear. For different values of Λ and Δ , we observed various behaviors such as tumbling, staggering (tank-treading with periodically varying inclination angles), “mirrored” tank-treading (tank-treading in the opposite direction and with inclination against the applied shear direction), and even chaotic staggering. A detailed analysis and characterization of these dynamics are currently being investigated and will be reported at a later date.

4.3.3 Two-body EHD interactions

Next we present results from simulation of two-body vesicle interactions in applied electric field and in the absence of imposed flow. As before, we assume that the viscosity and permittivity of the interior and exterior fluids are the same. We set the initial shape of both the vesicles to be identical and their initial location not

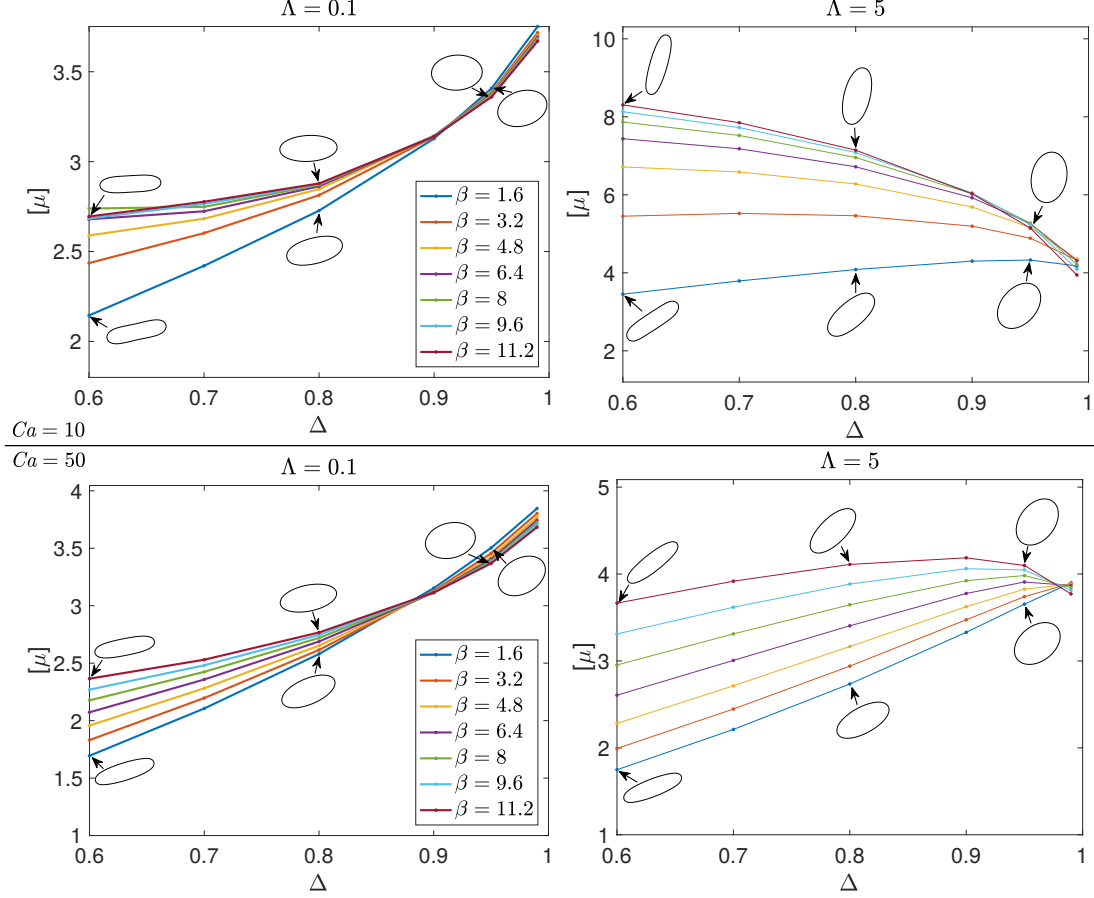


Figure 4.5: Dependence of effective viscosity $[\mu]$ on β and Δ . Conductivity $G = 4$ and $Ca = 10$ (top row) or $Ca = 50$ (bottom row). We note that (i) $[\mu]$ is higher whenever the equilibrium angle at which the vesicle tank-treads is away from the direction of shear and (ii) when Δ is close to 1 (vesicle closer to a circle), $[\mu]$ is nearly β -independent and shear-independent (as can be expected).

symmetric with respect to the electric field direction¹. We apply a DC electric field, pointing upwards, strong enough to cause the POP transition when $\Lambda = 0.1$ (i.e., $\beta > \beta_0$). Under these conditions, the different representative classes of dynamics observed are summarized in Figure 4.6.

The complex nature of these pairwise interactions can be understood from three predominant, competing mechanisms: (i) The electrically-driven vesicle alignment due to one vesicle appearing as a dipole (to leading order) in the far-field electrical

¹When they are aligned along \mathbf{E}_∞ , they simply attract each other (after transient shape changes) and when aligned in the perpendicular direction, they simply repel each other—both results are consequences of one vesicle appearing to the other as a dipole with same orientation.

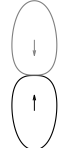

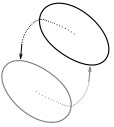
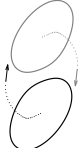
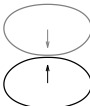
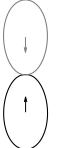
	$\Lambda = 0.1$	$\Lambda = 0.3$	$\Lambda = 0.5$	$\Lambda > 1$
$G = 0$	 chain formation (with POP)	 oscillatory motion	 circulatory motion (counterclockwise)	 circulatory motion (clockwise)
$G > 0$	 chain formation (oblates)		 chain formation (prolates)	

Figure 4.6: A summary of pairwise vesicle EHD interactions ($\Delta = 0.9$, $\beta = 3.2$, $Ca = 0$)

disturbance produced by the second vesicle. The two vesicles always tend to form a chain along the direction of dipole orientation; (ii) The EHD flow induced by the tangential electrical stresses at the fluid-vesicle interfaces, driving the vesicles to rotate about each other; (iii) The prolate-oblate deformation mentioned in Section 4.3.1, generating extensional flows around each vesicle.

First, let us consider the case of $G = 0$ i.e., the vesicle membranes are impermeable to charges. Three different types of dynamics can be observed from Figure 4.6. The first is *chain formation*, observed when Λ is small enough, wherein, pronounced deformation, due to mechanism (iii), induces flows that dominate the circulatory flow of mechanism (ii). Thereby, it completely halts the tank-treading motion. At the end of their POP cycle, both vesicles become almost vertically-aligned. Then, mechanism (i) slowly drives them to form a stable chain. From our numerical experiments, we noticed that the thin layer of fluid between the vesicles gets continuously drained albeit at a very slow pace (distance between them decays exponentially with time).

The second type is a *circulatory motion*, observed when Λ is large enough, wherein, mechanism (iii) becomes negligible. As the two vesicles move to form a chain, mechanism (ii) causes both of them to tank-tread. Consequently, the induced disturbance

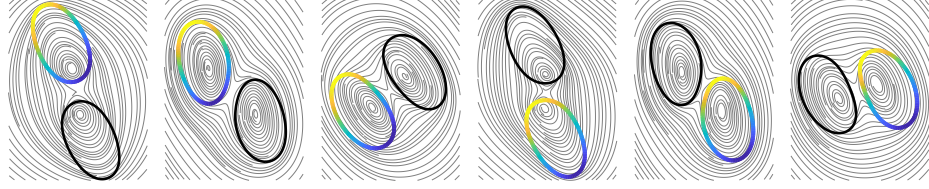


Figure 4.7: Snapshots from a simulation of two vesicles undergoing circulatory motion described in Figure 4.6 with $G = 0$ and $\Lambda = 0.5$. Here, one of the vesicles is colored by the magnitude of V_m (yellow indicates positive and blue indicates negative values respectively). We can observe that each vesicle undergoes tank-treading motion on its own (as indicated by the streamlines), they rotate about each other and the vesicles viewed as dipoles are always tilted with respect to the applied field direction.

flow on each vesicle becomes dominant and they start to rotate about each other. The tank-treading motion also causes the vesicles to appear as tilted dipoles, so they tend to form a tilted chain. The circulatory motion is periodically reinforced by the tilted-chain formation process. The direction of rotation depends on the net torque on each vesicle, which has opposite orientations for $\Lambda > 1$ and $\Lambda \leq 1$.

The last type is an *oscillatory motion*, where the two vesicles form an unstable chain and oscillate about each other. This is a transitional situation between the first two types, observed when Λ is between the values of those types. In this case, neither the circulatory flow of mechanism (ii) is strong enough to keep vesicles rotating about each other nor the deformational flow of mechanism (iii) is strong enough to completely halt the rotations. The two vesicles tend to form a chain that is periodically tilted one way or the other; each time the vesicles passing a tilted-chain position, tank-treading slows down and the dipole orientation oscillates back. Therefore, mechanisms (i) and (ii) collaborate to keep the vesicles oscillating near the vertical chain position.

On the other hand, the dynamics are much simpler when the membrane is permeable to charges i.e., $G \gg 0$. After a very short period of initial charging, the electric stresses become almost normal to the surface of each vesicle, so mechanism (ii) doesn't arise at all. By mechanism (iii) the vesicles eventually become oblate

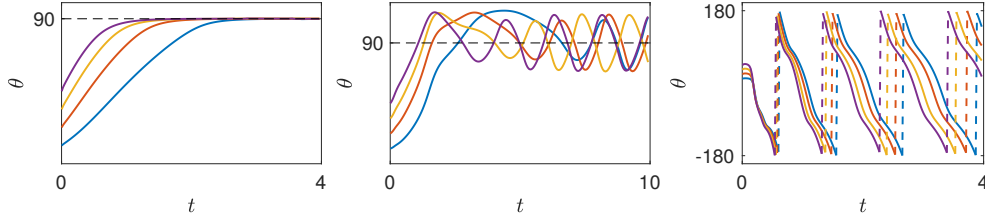


Figure 4.8: Insensitivity of the EHD pairwise interactions to the initial offset from the aligned position. θ measures the angular offset of the two vesicles relative to the horizontally aligned position. (a) Chain formation. (b) Oscillatory motion. (c) Circulatory motion. In each case, the same pattern is observed regardless of the initial $\theta > 0$.

when $\Lambda < 1$ (with strong enough β) and become prolate when $\Lambda > 1$, and mechanism (i) drives the vesicles to form a vertical chain.

Sensitivity to positions and shapes. Note that all of the aforementioned dynamics are insensitive to the initial offset or shapes of the vesicles. In Figure 4.8, we demonstrate that for different initial angular offsets from the aligned position, the vesicles undergo the same type of pairwise interaction that corresponds to the given Λ and G . Furthermore, Figure 4.9 shows that the similar kind of dynamics are observed for vesicles with different reduced volumes, therefore, the pairwise EHD interaction mechanisms appear to be consistent for highly-deflated or close-to-circular vesicles.

Continuous transition. Finally, we note that the dynamics transitioning from $G = 0$ to $G > 0$, as shown in Figure 4.6, are not abrupt. To illustrate this, we show in Figure 4.10 the pairwise dynamics of vesicles with $\Lambda = 0.1$, demonstrating a continuous transition from a chain of prolates ($G = 0$) to a chain of oblates ($G \gg 0$); for certain intermediate values of G , one can even observe interesting kidney-like shapes as well as decaying oscillations of the vesicles as they settle into their equilibrium shapes.

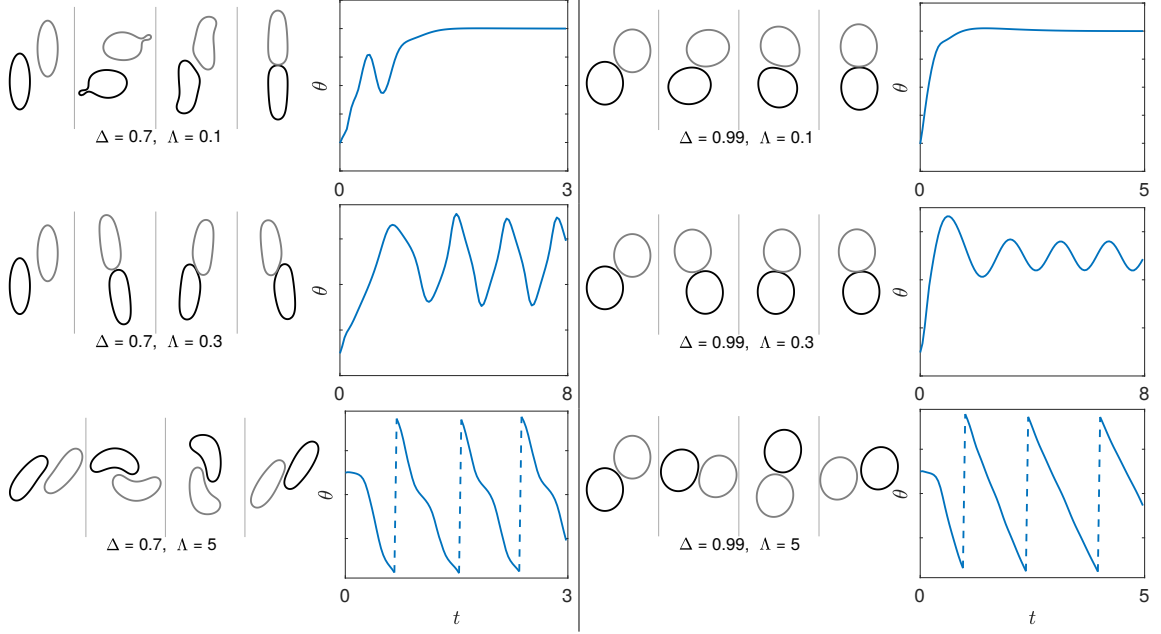


Figure 4.9: Pairwise interactions for $G = 0$ vesicles of reduced volumes $\Delta = 0.7$ (with $\beta = 4.8$) and $\Delta = 0.99$ (with $\beta = 2.4$). $Ca = 0$. The behaviors (e.g. chain formation, oscillatory motion, circulatory motion) are the same as in the $\Delta = 0.9$ case (Fig. 4.6), showing that the mechanism of pairwise interactions is insensitive to the reduced volume. Note that the bud formation also happens with the case of $\Delta = 0.7, \Lambda = 0.1$.

4.4 Conclusions

We presented a well-conditioned boundary integral equation formulation for solving the leaky-dielectric model describing the EHD of deformable vesicles. A collection of numerical advances—semi-implicit time-stepping, spectrally-accurate evaluation of weakly-singular, nearly-singular and hyper-singular integrals—enabled us to shed light onto the mechanics of highly deflated vesicles, study their rheology and pairwise dynamics in DC electric fields. We showed that much richer set of pairwise interactions can be observed when the membranes are impermeable to charges. This is somewhat unique to vesicle EHD compared to other systems such as drops [11], driven mainly by the capacitative nature of the membranes. However, we explored only a small fraction of the possible dynamics; relaxing our simplifying assumptions—varying the viscosity and permittivity contrasts, imposing an AC electric field, ac-

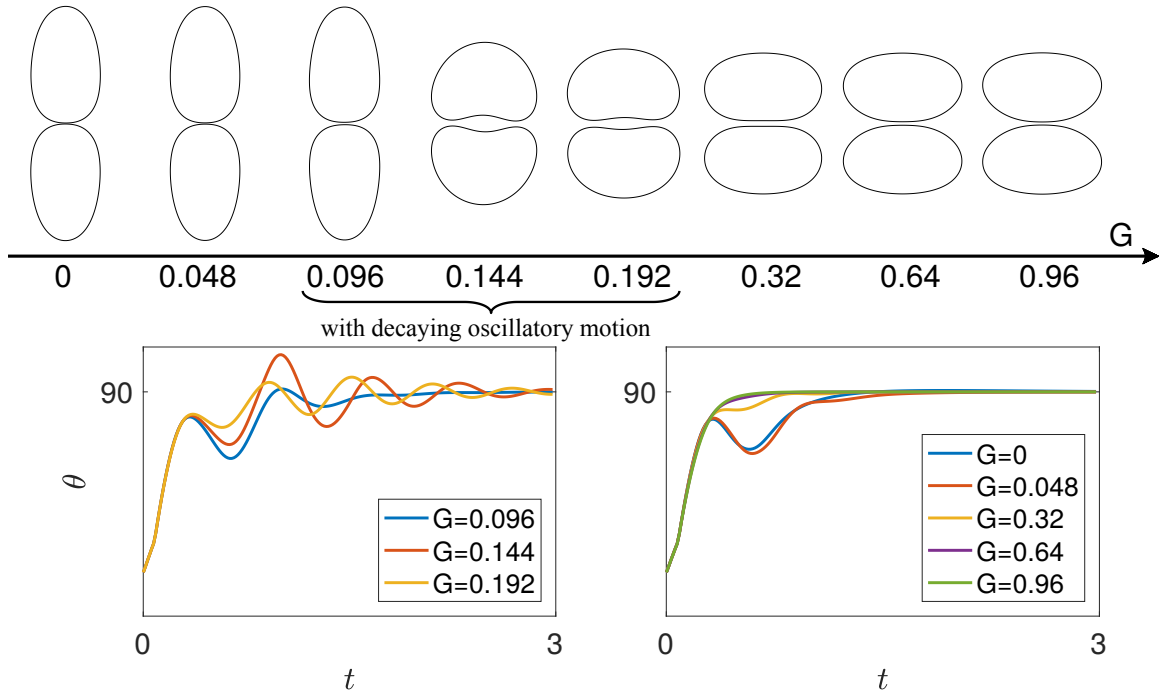


Figure 4.10: Top figure: final configurations of eight separate simulations, each corresponding to a different membrane conductivity G . There is a continuous transition from a chain of prolates ($G \approx 0$) to a chain of oblates ($G \gg 0$). For certain intermediate value of G (e.g. $G = 0.096, 0.144, 0.192$) the chain formation process is accompanied with decaying oscillatory motions (lower left figure), while for more extreme values of G the vesicles directly form a chain without oscillations (lower right figure). For all simulations $\beta = 3.2$, $\Lambda = 0.9$, and $Ca = 0$.

counting for charge convection along the membrane—is expected to enrich the space much further. We are currently exploring these as well as analyzing the collective dynamics of dense suspensions in periodic domains using the periodization techniques developed recently in [50] and [10]. Another important direction we are currently pursuing is to extend our numerical scheme to handle more general EHD models such as those discussed in the recent work of [53].

CHAPTER V

Summary

In this thesis, we first presented two spectrally-accurate quadrature schemes for the close evaluation of 2D Stokes layer potentials. The global quadrature scheme in Chapter II is effective for smooth closed curves and converge exponentially in the number of quadrature node for a target point arbitrarily close to the source curve. The panel quadrature scheme further extends the close evaluation capacity to arbitrary geometries that could be non-smooth, and can handle both targets that are on and off the boundary equally well, achieving super-algebraic convergence in the number of unknowns. Further more, an adaptive panel refinement procedure is proposed that can efficiently assign panels to achieve a user-requested tolerance for any complex geometry.

The global quadrature scheme is excellent for the particle-particle close interactions, hence is used in Chapter IV for the study of vesicle EHD. In this study, we have developed a new scheme that can handle the EHD interaction between vesicles as well as when the vesicles are deflated. We observed and analyzed new EHD phenomena including the “budding” phenomenon of a deflated vesicle and a rich set of pairwise interactions between vesicles.

There are many possible future directions.

In terms of numerical improvements, we expect to further improve the performance

of our panel quadrature scheme by incorporating more sophisticated quadratures designed for corner singularities, such as the RCIP [29] and the recent work of [63].

In terms of software developments, we envision building a fast 2D particulate flow software library that includes the following components. (1) Use the two close evaluation schemes developed in this thesis to efficiently handle the particle-particle and particle-wall interactions as well as the complex geometries. (2) Incorporate the periodization schemes in [50] and [10] to enable periodic particulate flow simulations. (3) Develop fast direct solvers similar to those in [50] to allow massive simulations.

In terms of applications, the vesicle EHD has much more rich dynamics to be explored. For example, one can further investigate the vesicle EHD by varying key parameters in addition to the fluid conductivity contrasts, imposing an AC electric field, or exploring the effect of charge convection along the membrane. The collective EHD dynamics of dense suspensions may also be analyzed in periodic domains using the techniques in [10].

Finally, it is unclear how the 2D schemes in this thesis can be generalized for problems in 3D since our 2D schemes required complex analysis, whose higher-dimensional analogs are not as well behaved. What would be the optimal close evaluation scheme for 3D surfaces is still an open question.

APPENDIX

APPENDIX A

Stokes pressure and traction in terms of contour integrals

Here, we give formulae for the traction vector \mathbf{T} induced at a target point with given surface normal, and the associated pressure field p , when the velocity field is represented by a Stokes single or double layer potential. The goal is to write the traction and pressure in terms of the four contour integrals of Sec. 3.2.2, to which close-evaluation methods of Sec. 3.3.3 may then be applied. This enables uniformly accurate force calculations on bodies, or solution of traction BVPs. We use the notation of Sec. 3.2: recall that $r = x - y$, n_x and n_y are the normal vectors at the target x and source y respectively, $\rho = |r|$, and I denotes the 2×2 identity operator.

We first consider the single layer potential (3.3). Its traction is

$$(\mathbf{T}^S \sigma)(x) = -\frac{1}{\pi} \int_{\Gamma} \frac{r \cdot n_x}{\rho^2} \frac{r \otimes r}{\rho^2} \sigma(y) ds_y, \quad (\text{A.1})$$

which turns out to be the negative of the Stokes DLP (3.4) with n_y replaced by n_x .

While (3.17) is no longer useful in this case, we can instead write the traction as

$$(\mathbf{T}^S \sigma)(x) = -\frac{1}{\pi} \int_{\Gamma} (r \cdot \sigma(y)) \frac{r}{\rho^4} (r \cdot n_x) ds_y \quad (\text{A.2})$$

and use the slightly different identity

$$\nabla_x \left(\frac{r \cdot \sigma}{\rho^2} \right) = \frac{\sigma}{\rho^2} - (r \cdot \sigma) \frac{2r}{\rho^4} \quad (\text{A.3})$$

to write the traction kernel as

$$\begin{aligned} (\mathbf{T}^S \sigma)(x) &= \frac{-1}{2\pi} \int_{\Gamma} \frac{r \cdot n_x}{\rho^2} \sigma ds_y + \frac{1}{2\pi} (x \cdot n_x) \nabla \int_{\Gamma} \frac{r \cdot \sigma}{\rho^2} ds_y \\ &\quad - \frac{1}{2\pi} n_{x,1} \nabla \int_{\Gamma} \frac{r \cdot \sigma}{\rho^2} y_1 ds_y - \frac{1}{2\pi} n_{x,2} \nabla \int_{\Gamma} \frac{r \cdot \sigma}{\rho^2} y_2 ds_y, \end{aligned} \quad (\text{A.4})$$

where $(n_{x,1}, n_{x,2}) =: n_x$ are the two components of n_x . As did in the case of velocity potentials, we can concisely write (A.4) as

$$\mathbf{T}^S \sigma = ((\mathcal{S}\sigma_1)_n + i(\mathcal{S}\sigma_2)_n) + \frac{1}{2\pi} \left(\text{Re}(x/n_x) \overline{I_H(\sigma/n_y)} - n_{x,1} \overline{I_H(\sigma y_1/n_y)} - n_{x,2} \overline{I_H(\sigma y_2/n_y)} \right), \quad (\text{A.5})$$

where all the \mathbb{R}^2 vectors are now understood as complex numbers in \mathbb{C} , the over line in $\overline{I_H(\cdot)}$ denotes the complex conjugate of $I_H(\cdot)$ and the dot product $x \cdot n_x = \text{Re}(x/n_x)$ is due to the fact that $1 = |n_x|^2 = n_x \overline{n_x}$.

The single layer pressure associated to (3.3) is

$$(P^S \sigma)(x) = \frac{1}{2\pi} \int_{\Gamma} \frac{r \cdot \sigma}{\rho^2} ds_y, \quad (\text{A.6})$$

which again in the complex plane can be written as

$$P^S \sigma = \text{Re} \frac{i}{2\pi} I_C(\sigma/n_y). \quad (\text{A.7})$$

We now turn to the Stokes double layer potential (3.4). The traction kernel and its associated pressure kernel are given by [48, (5.27)] [10, (3.37)],

$$\begin{aligned}
(\mathbf{T}^D\sigma)(x) &= \frac{\mu}{\pi} \int_{\Gamma} \left(-8 \frac{r \otimes r}{\rho^6} (r \cdot n_x)(r \cdot n_y) + \frac{r \otimes n_x}{\rho^4} (r \cdot n_y) + \frac{r \otimes r}{\rho^4} (n_x \cdot n_y) \right. \\
&\quad \left. + I \frac{1}{\rho^4} (r \cdot n_x)(r \cdot n_y) + \frac{n_y \otimes r}{\rho^4} (r \cdot n_x) + \frac{n_x \otimes n_y}{\rho^2} \right) \sigma(y) ds_y \\
(P^D\sigma)(x) &= \frac{\mu}{\pi} \int_{\Gamma} \left(-\frac{n_y \cdot \sigma(y)}{\rho^2} + 2 \frac{r \cdot \sigma(y)}{\rho^4} (r \cdot n_y) \right) ds_y.
\end{aligned} \tag{A.8}$$

The corresponding boundary integral operators $(\mathbf{T}^D\sigma)(x)$ and $(P^D\sigma)(x)$ are hyper-singular. We can easily derive the following equation expressing this operator in terms of the Laplace double layer potential:

$$\begin{aligned}
\frac{1}{\mu} \mathbf{T}^D\sigma &= -2(x \cdot n_x \nabla \nabla \mathcal{D}[\sigma] - n_{x,1} \nabla \nabla \mathcal{D}[y_1\sigma] - n_{x,2} \nabla \nabla [y_2\sigma]) \\
&\quad + 3I(n_x \cdot \nabla \mathcal{D})[\sigma] - (n_x \otimes \nabla \mathcal{D})[\sigma] - (\nabla \mathcal{D} \otimes n_x)[\sigma] \\
&\quad + \begin{bmatrix} 1 \\ -1 \end{bmatrix} (n_x \cdot \nabla \mathcal{D} + n_x \otimes \nabla \mathcal{D} - \nabla \mathcal{D} \otimes n_x) \begin{bmatrix} \bar{n}_y \\ n_y \end{bmatrix} \sigma, \tag{A.9} \\
P^D\sigma &= -2\mu \left(\frac{\partial}{\partial x_1} \mathcal{D}[\sigma_1] + \frac{\partial}{\partial x_2} \mathcal{D}[\sigma_2] \right) \\
&= \frac{\mu}{\pi} (\text{Im}(I_H(\sigma_1)) + \text{Re}(I_H(\sigma_2))).
\end{aligned}$$

Here $\frac{\bar{n}_y}{n_y} \sigma = [\frac{\bar{n}_y}{n_y} \sigma_1, \frac{\bar{n}_y}{n_y} \sigma_2]^T$, ∇ is short for ∇_x , and $\nabla \nabla$ is the Hessian tensor. The gradients of Laplace double-layer potentials needed above are expressed in terms of Hadamard integrals using (3.15). The Hessians are given in terms of supersingular

integrals as follows:

$$\nabla\nabla\mathcal{D}[\sigma](x) = \begin{bmatrix} \operatorname{Re}\frac{i}{\pi}(I_S(\sigma_1))(x) & -\operatorname{Im}\frac{i}{\pi}(I_S(\sigma_2))(x) \\ -\operatorname{Im}\frac{i}{\pi}(I_S(\sigma_1))(x) & -\operatorname{Re}\frac{i}{\pi}(I_S(\sigma_2))(x) \end{bmatrix}. \quad (\text{A.10})$$

The close evaluation formulae for these are in Sec. 3.3.3.3.

To validate the above formulae, we include in Fig. 3.4(b–c) the convergence of the maximum error in pressure and traction for the smooth domain of Example 1 from Sec. 4.3. The convergence rate is very similar to that of velocity albeit a loss of 1–2 digits, which is expected due to the extra derivatives.

BIBLIOGRAPHY

BIBLIOGRAPHY

- [1] Ludvig af Klinteberg, Davoud Saffar Shamshegar, and Anna-Karin Tornberg. Fast ewald summation for free-space Stokes potentials. *Research in the Mathematical Sciences*, 4(1):1, 2017.
- [2] Ludvig af Klinteberg and Anna-Karin Tornberg. Adaptive quadrature by expansion for layer potential evaluation in two dimensions. *SIAM Journal on Scientific Computing*, 40, 04 2017.
- [3] Bradley K Alpert. Hybrid Gauss-trapezoidal quadrature rules. *SIAM Journal on Scientific Computing*, 20(5):1551–1584, 1999.
- [4] Yolanda Alvarez, Maria L Cederlund, David C Cottell, Brent R Bill, Stephen C Ekker, Jesus Torres-Vazquez, Brant M Weinstein, David R Hyde, Thomas S Vihtelic, and Breandan N Kennedy. Genetic determinants of hyaloid and retinal vasculature in zebrafish. *BMC developmental biology*, 7(1):114, 2007.
- [5] Anthony P Austin, Peter Kravanja, and Lloyd N Trefethen. Numerical algorithms based on analytic function values at roots of unity. *SIAM Journal on Numerical Analysis*, 52(4):1795–1821, 2014.
- [6] Markus Bantle and Stefan Funken. Efficient and accurate implementation of *hp*-BEM for the Laplace operator in 2D. *Applied Numerical Mathematics*, 95:51 – 61, 2015. Fourth Chilean Workshop on Numerical Analysis of Partial Differential Equations (WONAPDE 2013).
- [7] Alex Barnett. `memorygraph`: a MATLAB/octave unix tool to record true memory and CPU usage vs time. <https://github.com/ahbarnett/memorygraph>, 2018.
- [8] Alex Barnett, Bowei Wu, and Shravan Veerapaneni. Spectrally accurate quadratures for evaluation of layer potentials close to the boundary for the 2D Stokes and Laplace equations. *SIAM Journal on Scientific Computing*, 37(4):B519–B542, 2015.
- [9] Alex H Barnett. Evaluation of layer potentials close to the boundary for Laplace and Helmholtz problems on analytic planar domains. *SIAM Journal on Scientific Computing*, 36(2):A427–A451, 2014.

- [10] Alex H Barnett, Gary R Marple, Shravan Veerapaneni, and Lin Zhao. A unified integral equation scheme for doubly periodic Laplace and Stokes boundary value problems in two dimensions. *Communications on Pure and Applied Mathematics*, 71(11):2334–2380, 2018.
- [11] James C Baygents, NJ Rivette, and Howard A Stone. Electrohydrodynamic deformation and interaction of drop pairs. *Journal of Fluid Mechanics*, 368:359–375, 1998.
- [12] J Thomas Beale and Ming-Chih Lai. A method for computing nearly singular integrals. *SIAM Journal on Numerical Analysis*, 38(6):1902–1925, 2001.
- [13] Jean-Paul Berrut and Lloyd N Trefethen. Barycentric lagrange interpolation. *SIAM review*, 46(3):501–517, 2004.
- [14] Marc Bonnet, Ruowen Liu, and Shravan Veerapaneni. Shape optimization of Stokesian peristaltic pumps using boundary integral methods. *arXiv preprint arXiv:1903.03634*, 2019.
- [15] Spencer H Bryngelson and Jonathan B Freund. Global stability of flowing red blood cell trains. *Physical Review Fluids*, 3(7):073101, 2018.
- [16] Camille Carvalho, Shilpa Khatri, and Arnold D Kim. Asymptotic analysis for close evaluation of layer potentials. *Journal of Computational Physics*, 355:327–341, 2018.
- [17] P. J. Davis. On the numerical integration of periodic analytic functions. In R. E. Langer, editor, *Proceedings of a Symposium on Numerical Approximations*. University of Wisconsin Press, 1959.
- [18] Philip J Davis and Philip Rabinowitz. *Methods of numerical integration*. Courier Corporation, 2007.
- [19] Yuhong Fu and Gregory J Rodin. Fast solution method for three-dimensional Stokesian many-particle problems. *Communications in Numerical Methods in Engineering*, 16(2):145–149, 2000.
- [20] Deborah Kuchnir Fygenson, John F Marko, and Albert Libchaber. Mechanics of microtubule-based membrane extension. *Physical review letters*, 79(22):4497, 1997.
- [21] Zydrunas Gimbutas and Leslie Greengard. Computational software: Simple FMM libraries for electrostatics, slow viscous flow, and frequency-domain wave propagation. *Communications in Computational Physics*, 18(2):516–528, 2015.
- [22] Abinand Gopal and Lloyd N. Trefethen. Solving Laplace problems with corner singularities via rational functions. *SIAM Journal on Numerical Analysis (to appear)*, 2019.

- [23] A Greenbaum, L Greengard, and A Mayo. On the numerical solution of the biharmonic equation in the plane. *Physica D*, 60(1–4):216–225, 1992.
- [24] Leslie Greengard, Mary Catherine Kropinski, and Anita Mayo. Integral equation methods for Stokes flow and isotropic elasticity in the plane. *Journal of Computational Physics*, 125(2):403–414, 1996.
- [25] Leslie Greengard and Vladimir Rokhlin. A fast algorithm for particle simulations. *Journal of computational physics*, 73(2):325–348, 1987.
- [26] Nicholas Hale and Lloyd N Trefethen. New quadrature formulas from conformal maps. *SIAM Journal on Numerical Analysis*, 46(2):930–948, 2008.
- [27] S. Hao, A. H. Barnett, P. G. Martinsson, and P. Young. High-order accurate Nyström discretization of integral equations with weakly singular kernels on smooth curves in the plane. *Adv. Comput. Math.*, 40(1):245–272, 2014.
- [28] F.-K. Hebeker. Efficient boundary element methods for three-dimensional exterior viscous flows. *Numer. Methods Partial Differential Equations*, 2:273–297, 1986.
- [29] Johan Helsing. Solving integral equations on piecewise smooth boundaries using the RCIP method: a tutorial. In *Abstract and Applied Analysis*, volume 2013. Hindawi, 2013.
- [30] Johan Helsing and Shidong Jiang. On integral equation methods for the first Dirichlet problem of the biharmonic and modified biharmonic equations in non-smooth domains. *SIAM Journal on Scientific Computing*, 40(4):A2609–A2630, 2018.
- [31] Johan Helsing and Rikard Ojala. On the evaluation of layer potentials close to their sources. *Journal of Computational Physics*, 227(5):2899–2921, 2008.
- [32] N Heuer and E P Stephan. The hp -version of the boundary element method on polygons. *J. Integral Equ. Appl.*, 8(2):173–212, 1996.
- [33] Nicholas J Higham. The numerical stability of barycentric Lagrange interpolation. *IMA Journal of Numerical Analysis*, 24(4):547–556, 2004.
- [34] G. C. Hsiao and W. L. Wendland. *Boundary integral equations*, volume 164 of *Applied Mathematical Sciences*. Springer, 2008.
- [35] Wei-Fan Hu, Ming-Chih Lai, Yunchang Seol, and Yuan-Nan Young. Vesicle electrohydrodynamic simulations by coupling immersed boundary and immersed interface method. *Journal of Computational Physics*, 317:66–81, 2016.
- [36] S Elizabeth Hulme, Sergey S Shevkoplyas, Javier Apfeld, Walter Fontana, and George M Whitesides. A microfabricated array of clamps for immobilizing and imaging *C. elegans*. *Lab on a Chip*, 7(11):1515–1523, 2007.

- [37] N. I. Ioakimidis, K. E. Papadakis, and E. A. Perdios. Numerical evaluation of analytic functions by Cauchy’s theorem. *BIT Numerical Mathematics*, 31:276–285, 1991.
- [38] Takuji Ishikawa and TJ Pedley. The rheology of a semi-dilute suspension of swimming model micro-organisms. *Journal of Fluid Mechanics*, 588:399–435, 2007.
- [39] Gökberk Kabacaoglu and George Biros. Optimal design of deterministic lateral displacement device for viscosity-contrast-based cell sorting. *Physical Review Fluids*, 3(12):124201, 2018.
- [40] Eisuke Kita and Norio Kamiya. Error estimation and adaptive mesh refinement in boundary element method, an overview. *Engineering Analysis with Boundary Elements*, 25(7):479–495, 2001.
- [41] Andreas Kloeckner, Alexander H. Barnett, Leslie Greengard, and Michael O’Neil. Quadrature by expansion: A new method for the evaluation of layer potentials. *Journal of Computational Physics*, 252:332–349, 2013.
- [42] Ebrahim M Kolahdouz and David Salac. Dynamics of three-dimensional vesicles in dc electric fields. *Physical Review E*, 92(1):012302, 2015.
- [43] Rainer Kress. Boundary integral equations in time-harmonic acoustic scattering. *Mathematical and Computer Modelling*, 15(3-5):229–243, 1991.
- [44] Rainer Kress. *Linear Integral Equations*, volume 82 of *Applied Mathematical Sciences*. Springer, 2nd edition, 1999.
- [45] M. C. A. Kropinski. An efficient numerical method for studying interfacial motion in two-dimensional creeping flows. *Journal of Computational Physics*, 171(2):479–508, 2001.
- [46] Amit Kumar and Michael D Graham. Accelerated boundary integral method for multiphase flow in non-periodic geometries. *Journal of Computational Physics*, 231(20):6682–6713, 2012.
- [47] Dag Lindbo and Anna-Karin Tornberg. Spectrally accurate fast summation for periodic Stokes potentials. *Journal of Computational Physics*, 229(23):8994–9010, 2010.
- [48] Y Liu. *Fast Multipole Boundary Element Method: Theory and Applications in Engineering*. Cambridge University Press, 2009.
- [49] Dhairya Malhotra and George Biros. PVFMM: A parallel kernel independent FMM for particle and volume potentials. *Communications in Computational Physics*, 18(3):808–830, 2015.

- [50] Gary R Marple, Alex Barnett, Adrianna Gillman, and Shravan Veerapaneni. A fast algorithm for simulating multiphase flows through periodic geometries of arbitrary shape. *SIAM Journal on Scientific Computing*, 38(5):B740–B772, 2016.
- [51] Lane C McConnell, Michael J Miksis, and Petia M Vlahovska. Vesicle electrohydrodynamics in dc electric fields. *IMA Journal of Applied Mathematics*, 78(4):797–817, 2013.
- [52] Lane C McConnell, Petia M Vlahovska, and Michael J Miksis. Vesicle dynamics in uniform electric fields: squaring and breathing. *Soft Matter*, 11(24):4840–4846, 2015.
- [53] Yoichiro Mori and Y-N Young. From electrodiffusion theory to the electrohydrodynamics of leaky dielectrics through the weak electrolyte limit. *Journal of Fluid Mechanics*, 855:67–130, 2018.
- [54] Ehssan Nazockdast, Abtin Rahimian, Denis Zorin, and Michael Shelley. A fast platform for simulating semi-flexible fiber suspensions applied to cell mechanics. *Journal of Computational Physics*, 329:173–209, 2017.
- [55] H Nganguia and Y-N Young. Equilibrium electrodeformation of a spheroidal vesicle in an ac electric field. *Physical Review E*, 88(5):052718, 2013.
- [56] Rikard Ojala and Anna-Karin Tornberg. An accurate integral equation method for simulating multi-phase Stokes flow. *Journal of Computational Physics*, 298:145–160, 2015.
- [57] V Y Pan. How bad are Vandermonde matrices? *SIAM J. Matrix Anal. Appl.*, 37(2):676–694, 2016.
- [58] Carlos Pérez-Arancibia, Luiz M Faria, and Catalin Turc. Harmonic density interpolation methods for high-order evaluation of Laplace layer potentials in 2D and 3D. *Journal of Computational Physics*, 376:411–434, 2019.
- [59] Dayinta L Perrier, Lea Rems, and Pouyan E Boukany. Lipid vesicles in pulsed electric fields: fundamental principles of the membrane response and its biomedical applications. *Advances in colloid and interface science*, 249:248–271, 2017.
- [60] Constantine Pozrikidis. *Boundary integral and singularity methods for linearized viscous flow*. Cambridge University Press, 1992.
- [61] Bryan Quaife and George Biros. High-volume fraction simulations of two-dimensional vesicle suspensions. *Journal of Computational Physics*, 274:245–267, 2014.
- [62] M Rachh. `bhfm2d`: parallel Fortran code for the biharmonic FMM in 2D, 2012.

- [63] Manas Rachh and Kirill Serkh. On the solution of Stokes equation on regions with corners. *arXiv preprint arXiv:1711.04072; submitted to Comm. Pure Appl. Math.*, 2017.
- [64] Abtin Rahimian, Alex Barnett, and Denis Zorin. Ubiquitous evaluation of layer potentials using quadrature by kernel-independent expansion. *BIT Numerical Mathematics*, 58(2):423–456, 2018.
- [65] Abtin Rahimian, Ilya Lashuk, Shravan Veerapaneni, Aparna Chandramowliswaran, Dhairya Malhotra, Logan Moon, Rahul Sampath, Aashay Shringarpure, Jeffrey Vetter, Richard Vuduc, et al. Petascale direct numerical simulation of blood flow on 200k cores and heterogeneous architectures. In *Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis, SC '10*, pages 1–11. IEEE Computer Society, 2010.
- [66] Abtin Rahimian, Shravan K Veerapaneni, Denis Zorin, and George Biros. Boundary integral method for the flow of vesicles with viscosity contrast in three dimensions. *Journal of Computational Physics*, 298:766–786, 2015.
- [67] Abtin Rahimian, Shravan Kumar Veerapaneni, and George Biros. Dynamic simulation of locally inextensible vesicles suspended in an arbitrary two-dimensional domain, a boundary integral method. *Journal of Computational Physics*, 229(18):6466–6484, 2010.
- [68] Karin A Riske and Rumiana Dimova. Electro-deformation and poration of giant vesicles viewed with high temporal resolution. *Biophysical journal*, 88(2):1143–1155, 2005.
- [69] William D Ristenpart, Olivier Vincent, Sigolene Lecuyer, and Howard A Stone. Dynamic angular segregation of vesicles in electrohydrodynamic flows. *Langmuir*, 26(12):9429–9436, 2010.
- [70] Mohamed M Sadik, Jianbo Li, Jerry W Shan, David I Shreiber, and Hao Lin. Vesicle deformation and poration under strong dc electric fields. *Physical Review E*, 83(6):066316, 2011.
- [71] David Saintillan, Eric Darve, and Eric SG Shaqfeh. A smooth particle-mesh Ewald algorithm for Stokes suspension simulations: The sedimentation of fibers. *Physics of Fluids*, 17(3):033301, 2005.
- [72] Paul F Salipante and Petia M Vlahovska. Vesicle deformation in dc electric pulses. *Soft matter*, 10(19):3386–3393, 2014.
- [73] Ashok S Sangani and Guobiao Mo. Inclusion of lubrication forces in dynamic simulations. *Physics of fluids*, 6(5):1653–1662, 1994.
- [74] Ashok S Sangani and Guobiao Mo. An $O(N)$ algorithm for Stokes and Laplace interactions of particles. *Physics of Fluids*, 8(8):1990–2010, 1996.

- [75] Claus Schneider and Wilhelm Werner. Some new aspects of rational interpolation. *Mathematics of Computation*, 47(175):285–299, 1986.
- [76] Jonathan T Schwalbe, Petia M Vlahovska, and Michael J Miksis. Vesicle electrohydrodynamics. *Physical Review E*, 83(4):046309, 2011.
- [77] Josef Sifuentes, Zydrunas Gimbutas, and Leslie Greengard. Randomized methods for rank-deficient linear systems. *Electronic Transactions on Numerical Analysis*, 44(Electronic Transactions on Numerical Analysis), 2015.
- [78] Chiara Sargentone and Anna-Karin Tornberg. A highly accurate boundary integral equation method for surfactant-laden drops in 3d. *Journal of Computational Physics*, 360:167–191, 2018.
- [79] Andrea Tagliasacchi. `kdtree`. <https://www.mathworks.com/matlabcentral/fileexchange/21512-ataiya-kdtree>, 2017.
- [80] Svetlana Tlupova and J Thomas Beale. Nearly singular integrals in 3D Stokes flow. *Communications in Computational Physics*, 14(5):1207–1227, 2013.
- [81] Anna-Karin Tornberg and Leslie Greengard. A fast multipole method for the three-dimensional Stokes equations. *Journal of Computational Physics*, 227(3):1613–1619, 2008.
- [82] John A. Toth and Steve Zelditch. Counting nodal lines which touch the boundary of an analytic domain. *J. Differential Geom.*, 81(3):649–686, 03 2009.
- [83] Lloyd N. Trefethen. *Spectral Methods in MatLab*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.
- [84] Lloyd N Trefethen. *Approximation theory and approximation practice*, volume 128. SIAM, 2013.
- [85] Lloyd N Trefethen and JAC Weideman. The exponentially convergent trapezoidal rule. *SIAM Review*, 56(3):385–458, 2014.
- [86] S. K. Veerapaneni, A. Rahimian, G. Biros, and D. Zorin. A fast algorithm for simulating vesicle flows in three dimensions. *Journal of Computational Physics*, 2011.
- [87] Shravan Veerapaneni. Integral equation methods for vesicle electrohydrodynamics in three dimensions. *Journal of Computational Physics*, 326:278–289, 2016.
- [88] Shravan K Veerapaneni, Denis Gueyffier, Denis Zorin, and George Biros. A boundary integral method for simulating the dynamics of inextensible vesicles suspended in a viscous fluid in 2D. *Journal of Computational Physics*, 228(7):2334–2353, 2009.

- [89] Petia M Vlahovska. Electrohydrodynamics of drops and vesicles. *Annual Review of Fluid Mechanics*, 51:305–330, 2019.
- [90] Petia M Vlahovska, Ruben Serral Gracia, Said Aranda-Espinoza, and Rumi-ana Dimova. Electrohydrodynamic model of vesicle deformation in alternating electric fields. *Biophysical journal*, 96(12):4789–4803, 2009.
- [91] Haitao Wang, Ting Lei, Jin Li, Jingfang Huang, and Zhenhan Yao. A parallel fast multipole accelerated integral equation scheme for 3D Stokes equations. *International journal for numerical methods in engineering*, 70(7):812–839, 2007.
- [92] Lei Wang, Svetlana Tlupova, and Robert Krasny. A treecode algorithm for 3D Stokeslets and stresslets. *arXiv preprint arXiv:1811.12498*, 2018.
- [93] Xin Wang, Joe Kanapka, Wenjing Ye, Narayan R Aluru, and Jacob White. Algorithms in FastStokes and its application to micromachined device simulation. *IEEE Transactions on computer-aided design of integrated circuits and systems*, 25(2):248–257, 2006.
- [94] B. Wu and S. Veerapaneni. Electrohydrodynamics of deflated vesicles: budding, rheology and pairwise interactions. *Journal of Fluid Mechanics*, 867:334–347, 2019.
- [95] Wen Yan, Eduardo Corona, Dhairya Malhotra, Shravan Veerapaneni, and Michael Shelley. A scalable computational platform for particulate Stokes suspensions. *Under Review*, 2019.
- [96] Lexing Ying, George Biros, and Denis Zorin. A kernel-independent adaptive fast multipole algorithm in two and three dimensions. *Journal of Computational Physics*, 196(2):591–626, 2004.
- [97] Lexing Ying, George Biros, and Denis Zorin. A high-order 3D boundary integral equation solver for elliptic PDEs in smooth domains. *Journal of Computational Physics*, 219(1):247–275, 2006.
- [98] Wenjun Ying and J Thomas Beale. A fast accurate boundary integral method for potentials on closely packed cells. *Communications in Computational Physics*, 14(4):1073–1093, 2013.
- [99] GK Youngren and A Acrivos. On the shape of a gas bubble in a viscous extensional flow. *Journal of fluid mechanics*, 76(3):433–442, 1976.
- [100] GK Youngren and Andrias Acrivos. Stokes flow past a particle of arbitrary shape: a numerical method of solution. *Journal of fluid Mechanics*, 69(2):377–403, 1975.
- [101] S S Zargaryan and V G Maz’ya. The asymptotic form of the solutions of the integral equations of potential theory in the neighbourhood of the corner points of a contour. *Prikl. Matem. Mekhan. U.S.S.R.*, 48(1):120–124, 1984.

- [102] Jia Zhang, Jeffrey D Zahn, Wenchang Tan, and Hao Lin. A transient solution for vesicle electrodeformation and relaxation. *Physics of Fluids*, 25(7):071903, 2013.
- [103] Yabin Zhang and Adrianna Gillman. A fast direct solver for boundary value problems on locally perturbed geometries. *Journal of Computational Physics*, 356:356–371, 2018.
- [104] Alexander Z Zinchenko and Robert H Davis. An efficient algorithm for hydrodynamical interaction of many deformable drops. *Journal of Computational Physics*, 157(2):539–587, 2000.