A fast solver for the Stokes PDE

Tejaswin Parthasarathy

December 22, 2017

1 Proposal

1.1 Deliverables

- Understanding of Integral Equation & PDE theory as applied to the Stokes PDE
- Code in python for solving the Stokes equation, building on the existing infrastructure of fast algorithms.
- The above code concentrates exclusively on the more relevant problem of exterior Dirichlet problems in two and three dimensions, prescribed with a velocity on a solid object surface
- Example test cases to setup/analyse flows generated by different geometries
- Documentation for the above code (this report)

1.2 Grading

- My understanding of how integral operator theory works, especially wrt the Stokes PDE
- The 'success' of the solver, tested against some standard validation cases for accuracy

1.3 Validation

- Simple cases governed by the Stokes PDE have an analytical solution which we can compare against (specified later)

This document is meant to serve as a reference (for me in the future) and hence goes into some detail.

2 Introduction

2.1 The Stokes PDE

In the limit of high viscosity, or when the governing physical scales are diminutive enough, the partial differential equations (PDE) governing incompressible Newtonian fluid flow (the celebrated Navier Stokes equations) reduce to the Stokes PDE.

$$\begin{aligned} \mu \Delta \boldsymbol{u} - \nabla p + \boldsymbol{f} &= 0 \\ \nabla \cdot \boldsymbol{u} &= 0 \end{aligned} \tag{1}$$

where $\boldsymbol{u}$ is the velocity of the fluid, p is the associated pressure, $\boldsymbol{f}$ is some representative force and μ is the (constant) dynamic viscosity in some domain D bounded by ∂D. The above formulation represents the steady formulation (time independent) of the generalized Stokes equation and hence is cast as a boundary value problem for three variables (u_i, $i = 1,2,3$ and p), with the associated boundary conditions.

2.2 PDE solution : Existence and Uniqueness

Given the above BVP, a natural question is related to the existence and uniqueness of the solution, especially applied to Dirichlet problems (interior and exterior) in u_i, $i = 1,2,3$. In short, the existence is non trivial and cannot be generalized (while uniqueness can, for Dirichlet problems [1, 2]). A notable case where existence is not guaranteed (and rightly so) is the case of two dimensional stokesian flow in an *infinite* domain past a stationary cylinder. However, the variational (weak) problem of Stokes flow (as used in the finite element formalism) is well posed (existence and uniqueness), in specific space pairs for $\boldsymbol{u}$ and p, for connected domains with a Lipschitz boundary [3, 4].

2.3 Numerical solution of the Stokes PDE

Given the well posedness of the weak form of the linear PDE, it seems surprising that accurate numerical simulation of Stokes flow is still an active area of research [5, 6, 7, 8, 9]. However, the Stokes PDE is notoriously difficult to solve. Techniques used in high Re Computational Fluid Dynamics (CFD) (both Finite Differencing (FDM) and Finite Volume (FVM) techniques) to solve the more general Navier Stokes equations fail because of stability restrictions in evolving the solution, and adapting them for Stokes flow simulations is not common. Finite Element Methods (FEM) are widely used, but they suffer from issues in selecting solution spaces, conditioning, solution order and time to solution. Integral Equation Methods seek to overcome the problems associated with solving the Stokes PDE with FDM, FVM and FEM. More specifically, if we have an integral equation formulation we can construct a matrix $\mathbf{A}$ and simply solve the linear system $\mathbf{A}x = b$ (where b represents the boundary conditions) to get the desired solution.

- The dimensions of the constant coefficient problem eq. (1) is reduced from d to $d-1$, using boundary integrals to represent the solution

- We forgo the requirement of remeshing to obtain a grid for representing moving particles in every time step to solve a quasi-time dependent BVP. Even frameworks like ALE (Arbitrary Lagrangian-Eulerian), IBM (Immersed Boundary Methods), IIM (Immersed Interface Methods) are rendered ineffective because of either (a) inefficiency in maintaining a separate grid for the particles or (b) inability to enforce boundary conditions rigorously
- $\boldsymbol{u}$ is guaranteed to be divergence free by the solution representation (thus removing the need to project to a divergence free subspace)
- The solutions can be meaningfully represented in simple polynomial spaces
- The condition number of the matrix $\mathbf{A}$ (the finite representation of the Stokes operator) can be restricted to a small constant number by careful construction of the matrix. Thus using iterative techniques like GMRES we can solve the linear system through a small number of mat-vecs
- The problem is conducive for acceleration through methods like FMM (an $\mathcal{O}(n)$ method), thus improving the time to solution (in addition to the dimension reduction discussed above)
- Obtaining higher order solution with an integral equation formulation is relatively straightforward

Thus the advantage of such methods. We go on to construct some integral equation representations of the solution.

2.4 Integral Equation Formulation

Given the linear stokes PDE in D bounded by ∂D eq. (1), we establish an integral equation formalism using fundamental solutions. Using our knowledge of unbounded Green's function for the Laplace and Biharmonic operators in three dimensions, and after algebraic manipulation we arrive at the fundamental solutions for the Stokes PDE in eq. (2) and eq. (3). Here $\hat{\boldsymbol{x}}$ indicates $\boldsymbol{x}-\boldsymbol{y}$ with $\boldsymbol{x}$ being the target location and $\boldsymbol{y}$ being the source location. For convenience we represent $\|\hat{\boldsymbol{x}}\|$ as r.

$$\begin{aligned} G_{ij}(\boldsymbol{x},\boldsymbol{y}) &= -\frac{1}{8\pi\mu}\left(\frac{\delta_{ij}}{r}+\frac{\hat{x}_i\hat{x}_j}{r^3}\right) \\ \phi_i(\boldsymbol{x},\boldsymbol{y}) &= -\frac{1}{4\pi}\frac{\hat{x}_i}{r^3} \end{aligned} \tag{2}$$

$$\begin{aligned} T_{ijk}(\boldsymbol{x},\boldsymbol{y}) &= \frac{3}{4\pi}\frac{\hat{x}_i\hat{x}_j\hat{x}_k}{r^5} \\ \Pi_{ij}(\boldsymbol{x},\boldsymbol{y}) &= \frac{\mu}{2\pi}\left(-\frac{\delta_{ij}}{r^3}+3\frac{\hat{x}_i\hat{x}_j}{r^5}\right) \end{aligned} \tag{3}$$

The above kernels are for the three dimensional Stokes equations. The two dimensional variants are equivalently:

$$\begin{aligned} G_{ij}(\boldsymbol{x},\boldsymbol{y}) &= -\frac{1}{4\pi\mu}\left(-\delta_{ij}\log r+\frac{\hat{x}_i\hat{x}_j}{r^2}\right) \\ \phi_i(\boldsymbol{x},\boldsymbol{y}) &= -\frac{1}{2\pi}\frac{\hat{x}_i}{r^2} \end{aligned} \tag{4}$$

$$
\begin{aligned}
T_{ijk}(\boldsymbol{x},\boldsymbol{y}) &= \frac{1}{\pi}\frac{\hat{x}_i\hat{x}_j\hat{x}_k}{r^4} \\
\Pi_{ij}(\boldsymbol{x},\boldsymbol{y}) &= \frac{\mu}{\pi}\left(-\frac{\delta_{ij}}{r^2}+2\frac{\hat{x}_i\hat{x}_j}{r^4}\right)
\end{aligned}
\tag{5}
$$

The kernel G_{ij} are the 'Stokeslets' that map causal point 'charges' at $\boldsymbol{y}$ (representing 'forces' physically) to the fluid velocity at $\boldsymbol{x}$. The equivalent pressure kernel for the Stokeslets are the ϕ_i kernels mapping point 'charges' at $\boldsymbol{y}$ to the fluid pressure at $\boldsymbol{y}$. The kernel T_{ijk} are the 'Stresslets' that map causal point 'charges' at $\boldsymbol{y}$ to the fluid stress $\boldsymbol{\sigma}(\boldsymbol{x})$ at $\boldsymbol{x}$. (The fluid stress is defined as $\sigma_{ij} = -P\delta_{ij}+\mu\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)$). We will also see that the stresslets map causal point 'dipoles' (force dipoles physically) with some strength and direction at $\boldsymbol{y}$, to fluid velocity at $\boldsymbol{x}$. In this context Π_{ij} represents, akin to ϕ_i, an equivalent pressure kernel mapping the dipoles at $\boldsymbol{y}$ to the pressure at $\boldsymbol{x}$. It can be seen that we have effectively decoupled the velocities and pressure.

Using these kernels, we can now construct an integral equation representation, mapping source charges and dipoles to the velocity and pressure, representing the single (Stokeslet kernel) and double (Stresslet kernel) layer potentials. Using the reciprocal identity (Pozrikidis Thm (1.4.3), (2.3.11)) (quite akin to Green's second identity) we can represent the velocity-pressure fields at any target location as

$$
\begin{aligned}
u_i(\boldsymbol{x}) &= \int_{\partial D}\left(G_{ij}(\boldsymbol{x},\boldsymbol{y})q_j(\boldsymbol{y}) + T_{ijk}(\boldsymbol{x},\boldsymbol{y})\gamma_j(\boldsymbol{y})n_k(\boldsymbol{y})\right)d(\boldsymbol{y}) \\
p(\boldsymbol{x}) &= \int_{\partial D}\left(\phi_j(\boldsymbol{x},\boldsymbol{y})q_j(\boldsymbol{y}) + \Pi_{jk}(\boldsymbol{x},\boldsymbol{y})\gamma_j(\boldsymbol{y})n_k(\boldsymbol{y})\right)d(\boldsymbol{y})
\end{aligned}
\tag{6}
$$

For constructing a matrix to solve the equation, we strive to attain a second kind Fredholm integral because of the implications of Riesz theory and Fredholm alternative for an operator of the form $\mathscr{I}+\mathscr{A}$ ($\mathscr{A}$ compact). This usually leads to a bounded condition number for the matrix $\mathbf{A}$ that does not grow with refinement. This warrants a pure double layer potential representation, which is also possible with the Stokes equations using only the Stresslet kernel. Thus we seek

$$
\begin{aligned}
u_i(\boldsymbol{x}) &= \int_{\partial D}\left(T_{ijk}(\boldsymbol{x},\boldsymbol{y})\gamma_j(\boldsymbol{y})n_k(\boldsymbol{y})\right)d(\boldsymbol{y}) \\
p(\boldsymbol{x}) &= \int_{\partial D}\left(\Pi_{jk}(\boldsymbol{x},\boldsymbol{y})\gamma_j(\boldsymbol{y})n_k(\boldsymbol{y})\right)d(\boldsymbol{y})
\end{aligned}
\tag{7}
$$

Close to the boundary then, we get the double layer potential in eq. (7) to be $\lim_{\epsilon\to 0}\mathscr{D}_i[\partial D,\boldsymbol{\gamma}](\boldsymbol{x}\pm\epsilon\boldsymbol{n}) = \pm\frac{\boldsymbol{\gamma}}{2}+\mathrm{PV}\,\mathscr{D}_i[\partial D,\boldsymbol{\gamma}]((\boldsymbol{x})$ where PV represents the principal value. This leads to a well conditioned system that can be solved with an iterative technique.

2.5 Integral equation solutions : Existence and Uniqueness

We need to prove that the chosen representations in eq. (6) and eq. (7) can faithfully represent the solution for different boundary conditions (we focus on Dirichlet and Neumann). Classical results [10] indicate that eq. (6) leads to existence and uniqueness of the solution for the integral equation representation. eq. (7) suffers from the same drawbacks as in the Laplace PDE. Because of the implications of the Fredholm Alternative, the exterior Dirichlet problem, which is usually well-posed, shares the null space with the interior Neumann operator. The conditions of uniqueness for interior Neumann

is no net force and torque (ie. $\int_{\partial D} q_i(\boldsymbol{y})d(\boldsymbol{y}) = 0$ and $\int_{\partial D} \epsilon_{ijk} y_j q_i(\boldsymbol{y})d(\boldsymbol{y}) = 0$), which carried over to exterior Dirichlet implies that eq. (7) is unable to represent distributions that correspond to non-zero forces and torques at the body (Pozrikidis Sec 4.3). More precisely, the cardinality of the null space of the the operator wrt exterior Dirichlet is 6, one each for every translational and rotational mode (each representing either a non-zero torque or force respectively).

Thus for complete representation of the flow, we need to deviate from the conventional $\frac{\mathscr{I}}{2} + \mathscr{D}$ representation and look for others (while still maintaining $\mathscr{I} + \mathscr{A}$, $\mathscr{A}$ compact). It is not surprising that many such representations exist. Let us then add a compact operator $\mathscr{V}_i[\partial D, \boldsymbol{\gamma}]$ to the first equation of eq. (7) to 'complete' the representation. $\mathscr{V}_i[\partial D, \boldsymbol{\gamma}]$ is called the completion flow and is usually a single layer operator (and hence continuous across the boundary) with some specified kernel and the **same** density as the double layer potential. With this solution representation ($u_i((\boldsymbol{x})) = \mathscr{D}_i[\partial D, \boldsymbol{\gamma}](\boldsymbol{x}) + \mathscr{V}_i[\partial D, \boldsymbol{\gamma}](\boldsymbol{x})$), we have two different sets of problems for the same boundary conditions:

2.5.1 The resistance problem

When the body motion $(\boldsymbol{U}, \boldsymbol{\Omega}, \boldsymbol{U}^{\text{def}})$ of the boundary is known, the resistance problem is to find the resulting force $\boldsymbol{f}$ and torque $\boldsymbol{t}$ exerted by the fluid on the boundary. These can be related to the double layer density $\boldsymbol{\gamma}$ through the definition in eq. (6), as

$$\begin{aligned} \boldsymbol{f}(\boldsymbol{\gamma}) &= \int_{\partial D} \boldsymbol{\gamma}(\boldsymbol{y}) d(\boldsymbol{y}) \\ \boldsymbol{t}(\boldsymbol{\gamma}) &= \int_{\partial D} \boldsymbol{\gamma}(\boldsymbol{y}) \times (\boldsymbol{y} - \boldsymbol{x_c}) d(\boldsymbol{y}) \end{aligned} \tag{8}$$

In other words given the LHS in the following equation, we need to solve for the vector density $\boldsymbol{\gamma}(\boldsymbol{y})$:

$$U_i + \boldsymbol{\Omega} + \epsilon_{ijk}\Omega_j(x - x^c)_k + U_i^{\text{def}} = \pm\frac{\gamma_i}{2} + \text{PV}\,\mathscr{D}_i[\partial D, \boldsymbol{\gamma}]((\boldsymbol{x}) + \mathscr{V}_i[\partial D, \boldsymbol{\gamma}](\boldsymbol{x}) \tag{9}$$

This is a straightforward $\mathbf{A}x = b$ problem. This is what we concentrate on in this study.

2.5.2 The mobility problem

Consider the other case where the deformation and the forces of the body are known $(\boldsymbol{f}, \boldsymbol{t}, \boldsymbol{U}^{\text{def}})$. We now need to solve for the body velocity $(\boldsymbol{U}, \boldsymbol{\Omega})$. This is more involved and is usually a problem that can be cast as $\mathbf{A}x = \mathbf{B}x$ or $x = -\mathbf{A}x + b + c$ to be solved for x. Nevertheless, representations that can simplify this exist (Pozrikidis sec 4.9). This is not the focus of this work.

2.6 Exterior Dirichlet: Representation

As seen, we need to choose the completion flow representation, and the choice of $\mathscr{V}_i[\partial D, \boldsymbol{\gamma}]$ seems arbitrary. Nevertheless we have some standard representations that have guaranteed solution existence and uniqueness.

2.6.1 $d = 2$

For $d = 2$ we use the representation specified in [11]. It states that the density $\boldsymbol{\gamma}$ is given by solving the integral equation

$$\begin{aligned} -\frac{\boldsymbol{\gamma}}{2} - \mathscr{D}\boldsymbol{\gamma} + \eta'\mathscr{P}\boldsymbol{\gamma} &= \boldsymbol{f} \\ \mathscr{P}\boldsymbol{\gamma} &= \mathscr{S}\left(\boldsymbol{\gamma} - \frac{1}{l}\int_{\partial D}\boldsymbol{\gamma}(\boldsymbol{y})d(\boldsymbol{y})\right) - \frac{\alpha'}{2\pi}\int_{\partial D}\boldsymbol{\gamma}(\boldsymbol{y})d(\boldsymbol{y}) \end{aligned} \tag{10}$$

The operator $\mathscr{P}$ consists of a single layer of the density with the mean removed (l represnts the length of the contour ∂D), and another contribution from the mean component. The choices for η' and α' (the prime indicates a notation change from the original paper) are arbitrary, with restrictions for optimality. Using the criterion provided [11], we use $\eta' = 1$and $\alpha' = 1$. Using this representation, it is possible to represent the exterior solutions.

2.6.2 $d = 3$

For three dimensions, we choose the representation given by [12]. This involves using a combined double layer and a single layer potential operating on a linear function of the double layer density $\boldsymbol{\gamma}$. That is we get the solution by solving the integral equation

$$\frac{\boldsymbol{\gamma}}{2} + \mathscr{D}\boldsymbol{\gamma} + \mathscr{S}(\eta\boldsymbol{\gamma} + \beta) = \boldsymbol{f} \tag{11}$$

Once again η and β are arbitrary. WLOG, we choose $\eta = 1$ and $\beta = 0$ and represent the exterior solutions.

3 Numerical Implementation

With the representation for $d = 2,3$ we can now construct solutions for the exterior Dirichlet problems. The details of the implementation/numerics are discussed in this section. We use the `pytential`[13] library and its dependencies in the following section .

3.1 Surface discretization

For $d = 2$, we test the representation with straightforward curves, using `nelements` number of elements and nodes corresponding to `target_order` polynomial representation. For $d = 3$ we generate a geometry and mesh it external to the workflow, before importing it with the similar settings for a function discretization on the surface.

3.2 Integral Equation discretization

Using the Nyström discretization method [2], we obtain the coefficients for the (coarse) density $\boldsymbol{\gamma}$ at the nodes decribed in the generated mesh (with `nelements` $\cdot$ `target_order` points). As Nyström provides convergence and an upper bound on the error, we use this coarse density to go from given points to an upsampled number of points, described by `ovsmp_target_order`. The underlying architecture is structured so that it returns a discretization in which the quadrature of polynomials with order `ovsmp_target_order` can be done exactly (no quadrature error). With this particular discretization, we now construct the matrix $\mathbf{A}$ (the coefficients from the quadrature with unit density on the surface) by evaluating the (singular) integrals defined in eq. (6).

3.3 Evaluating Integrals

Of the numerous techniques for on-surface integral evaluation using quadrature for singular operators, we choose Quadrature by Expansion (QBX).[14] QBX exploits the smoothness of the integral off the surface (where both $\mathscr{S}$ and $\mathscr{D}$ return smooth functions) by using a local expansion constructed just off the surface to evaluate the on-surface integral. The number of terms in the local expansion used in QBX correspond to `qbx_order + 1` (with the same order of accuracy). In our case, the QBX implementation is accelerated using an adaptive FMM for deriving the local expansions, using `fmm_order` terms for the same. The implementation also ensures sufficient refinement to ensure that the error (additive error from truncation and quadrature) is bound. The actual quadrature scheme may vary, as spectral convergence is guaranteed even for trapezoidal schemes. In correspondence with the generated surface discretization, `pytential` uses a Gauss Legendre quadrature that can integrate polynomials of degree $2N+1$ exactly. We can now solve for the Integral Equation by enforcing the boundary condition on the discretized surface.

3.4 Solving for the density

We enforce the boundary condition point-wise and solve the linear system $\mathbf{A}\gamma = b$. Due to good conditioning properties (see section 2.4), we can solve this system through any iterative technique. If done

right, we end up solving a $3n \times 3n$ system for the vector density $\boldsymbol{\gamma}$ (and enforcing boundary conditions for $i = 1,2,3$). In `pytential`, a variant of the restarted GMRES is used, reducing the $\mathcal{O}\left(n^3\right)$ direct solve to $C\mathcal{O}\left(n^2\right)$, where C is small and is determined by $\kappa(\mathbf{A})$.

3.5 Post-processing

With the density $\boldsymbol{\gamma}$ solved for, the pressure p, the velocity $\boldsymbol{u}$ and the stress $\boldsymbol{\sigma}$ anywhere in the domain can be obtained as a post-processing step, using the representation eq. (10) or eq. (11) with suitably modified kernels. This computation is once again suitable for acceleration thorough an adaptive FMM. One important characteristic is the force and torque on the surface on which the boundary conditions are imposed. As we have seen in section 2.6, $\mathscr{D}_i$ is unable to represent these forces. With the fixed representation eq. (10) for $d = 2$ and eq. (11) for $d = 3$, the evaluation of forces and torques on this surface is reduced to the problem of evaluating the equivalent $\mathscr{P}\left(\boldsymbol{\gamma}\right)$ and $\mathscr{S}\left(\eta\boldsymbol{\gamma} + \beta\right)$. Thus we get complete information about the flow.

3.6 Time-stepping

With the evaluated forces and torques, we can identify the future positions of the surface through an explicit time-stepping (RK/AB) scheme. With the new positions we once again solve a quasi-time steady problem defined solely by the updated (translated, rotated and deformed) boundary elements and repeat the process.

Algorithm 1 Stokes solve through Integral Equation Methods

Require: $T > 0$, $D \in \mathbb{R}^d$, $\partial D \in \mathbb{R}^{d-1}$
Require: Function representation capability on ∂D through panels.$\mathscr{M}$
Ensure: $t = 0$
1: **while** $t <= T$ **do**
2: Evaluate kernels $\mathscr{D}$ and $\mathscr{P}$ (or) $\mathscr{S}$ on ∂D {Using accelerated QBX w/ Gauss quadrature and Nyström discretization}
3: Compute velocities on ∂D at time t with prescribed $(\boldsymbol{U}(t), \boldsymbol{\Omega}(t), \boldsymbol{U}^{\text{def}}(t))$ from body dynamics
4: Solve for the surface density $\boldsymbol{\gamma}$ eq. (10) or eq. (11) using operation of the form

$$\left(\frac{I}{2} + D + S\right) \cdot \boldsymbol{\gamma} = \boldsymbol{b}\left((\boldsymbol{U}(t), \boldsymbol{\Omega}(t), \boldsymbol{U}^{\text{def}}(t))\right)$$

5: Compute $p/\boldsymbol{u}/\boldsymbol{\sigma}$ in the desired regions as a post processing step using appropriate kernels with FMM
6: Compute $\boldsymbol{f}[\partial D]$ and $\boldsymbol{t}[\partial D]$ (efficiently as in [11, 12])
7: Use explicit time stepping to solve the following (3×1) (for $d = 2$) or (3×1) (for $d = 3$) second order ODE system, where $\boldsymbol{M}, \boldsymbol{I}$ are the mass and inertia matrices

$$\begin{bmatrix} \boldsymbol{M} & 0 \\ 0 & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{x}}(t) \\ \ddot{\boldsymbol{\theta}}(t) \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}(t) \\ \dot{\boldsymbol{t}}(t) \end{bmatrix}$$

8: Evolve the positions, mesh and geometry according to the solution of the previous step
9: **end while**

4 Some results

In this section, we show results from sample runs by solving the Stokes equations for two dimensions. The three dimensional simulations are skipped owing to their prohibitive costs. The simulation results are primitive. Code development was not the target - but rather the focus was a proof of concept and understanding the effect of simulation parameters on the solution and its error.

4.1 Validation case and convergence

We test the (artificial) case of a spinning circle in a viscous fluid, with the formulation given in eq. (10). The boundary condition imposed involves a combination of a point force (with direction vector $\hat{\boldsymbol{f}} = \hat{e}_x$, associated with a stokeslet) and a point couple (with direction $\hat{\boldsymbol{t}} = \hat{e}_z$, associated with a singularity called the rotlet) inside the solid domain (i.e. in $\mathbb{R}^2/D$). We know that the net field produced is a linear superposition of the field caused by the individual singularities. In two dimensions, the Stokeslet and Rotlet singularity kernels (without the scaling factors) are

$$\begin{aligned} G_{ij}(\boldsymbol{x},\boldsymbol{y}) &= \left(-\delta_{ij}\log r + \frac{\hat{x_i}\hat{x_j}}{r^2}\right) \\ R_{ij}(\boldsymbol{x},\boldsymbol{y}) &= \left(\epsilon_{ijk}\frac{\hat{x_k}}{r^2}\right) \end{aligned} \tag{12}$$

,to be multiplied respectively by their respective forces and couples. The said kernels (with the said forces/torques) are pictorially depicted in the figures given below.

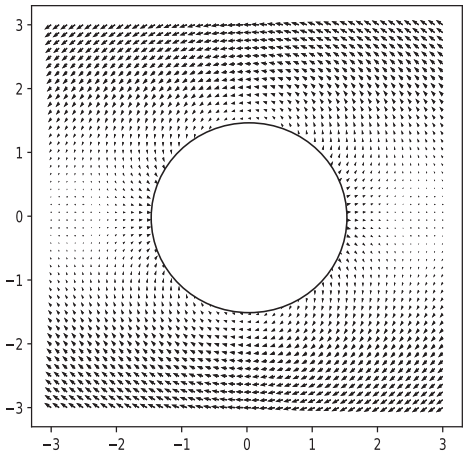

Figure 1: $\boldsymbol{u}$ field induced by the action of the Stokeslet kernel with a unit force $\boldsymbol{f} = \hat{e_x}$

We then impose force in the x direction with weighted strength (scaled with the kernel) `strength`, and a couple with `strength`/8, both at a location `fund_soln_loc`. The boundary condition on ∂D

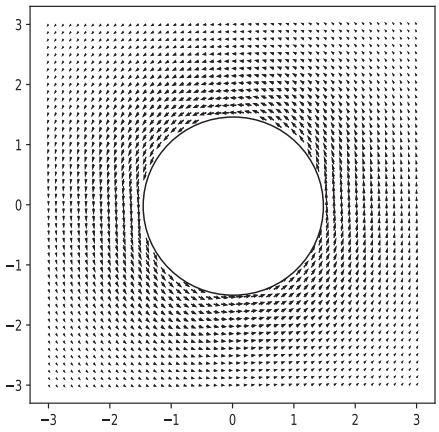

Figure 2: $\boldsymbol{u}$ field induced by the action of the Rotlet kernel with a unit couple $\boldsymbol{t} = \hat{e_z}$

is then imposed based on the stokeslet-rotlet solution. We use the solve-part of the procedure given in algorithm 1 to establish the fields. We then test the resulting velocity fields (based on our solution) with the analytical results from the Stokeslet-stresslet formula. The velocity field for a sample run is shown below:

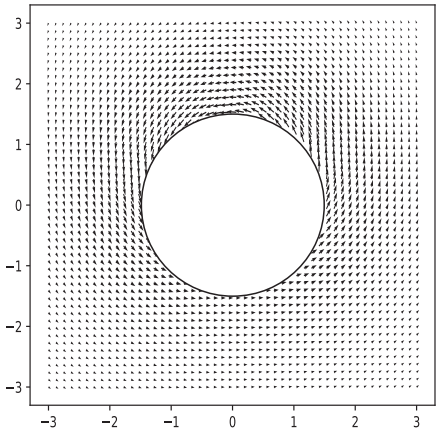

Figure 3: $\boldsymbol{u}$ field obtained from evaluating the solution outside ∂D

We compare the errors in the velocity for different element sizes (`n_elements`)and different qbx orders (`qbx_order`) in the table table 1 below, and the convergence plot fig. 4 following it.

qbx_order	h	$\|\text{error}\|_2$	$\|\text{error}\|_\infty$
	0.471238898	0.06875534615	0.02920006592
	0.235619449	0.01071585152	0.004890252719
2	0.1570796327	0.00343420811	0.001711021514
	0.1178097245	0.001508832439	0.0007496325353
	0.09424777961	0.0007917138205	0.0003926925152
order		2.77	2.66
	0.471238898	0.009608717878	0.004424500401
	0.235619449	0.0008170288352	0.0003987910161
3	0.1570796327	0.0001806385559	9.21E-05
	0.1178097245	6.31E-05	3.85E-05
	0.09424777961	2.73E-05	1.82E-05
order		3.64	3.42
	0.471238898	0.001415684831	0.0006817749502
	0.235619449	6.50E-05	3.26E-05
4	0.1570796327	1.63E-05	2.02E-05
	0.1178097245	1.81E-05	2.64E-05
	0.09424777961	1.01E-05	1.43E-05
order		3.03	3
	0.471238898	0.000213960101	0.0001137519624
	0.235619449	6.01E-06	4.18E-06
5	0.1570796327	1.30E-05	1.88E-05
	0.1178097245	1.79E-05	2.71E-05
	0.09424777961	1.03E-05	1.38E-05
order		1.54	0.84

Table 1: Convergence table for exterior flow around a cylinder for `strength = 100` and `fund_soln_loc = 0.0`. The numerical simulation parameters used are `mesh_order = 4`, `target_order = 4`, `fmm_order=10`, $\mu = 1$.

As can be seen, the error is seen to decrease on increasing the element size. The empirical order of convergence is also approximately `qbx_order + 1` as expected, except at higher orders. Increasing the mesh and target order had no impact - indicating that the information on the boundary is well resolved. Expected EOCs could be achieved by increasing the `FMM_order`, but with corresponding increase in the time to solution.

4.2 Effect of geometry on convergence

In microfluidic devices, where Stokes flow approximation is typically valid, the controllability of manufacturing methods is poor. This usually leads to "rough" geometries (which are analytically treated as smooth), leading to singularities. Thus, understanding the effect of shape induced singularities (sharp boundaries) on the solution is important.

We thus repeat the experiment for various geometries (and unsurprisingly, we see that our solution

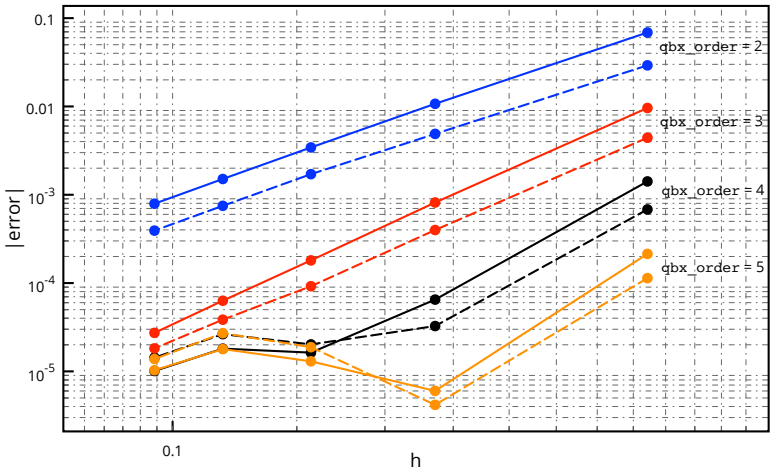

Figure 4: The trends of convergence for table 1 above. As expected, `qbx_order` corresponds well with the empirical order of solution (indicated as order), except at higher orders (where truncation errors are more relevant). The solid lines represent the error measured by L2 norm while the dashed lines represent the L-inf norm error

remains unchanged, as we impose the boundary condition based on a fundamental singularity). We run parametrized cases to understand the effect of geometry on the solution (and ultimate failure from QBX) - concentrating on the role of singular points in the geometry. For this study we choose a shape parametrization formulated for $0 \leq t \leq 2\pi$ below:

$$\begin{aligned} x(t) &= \cos(t) + a\sin(bt) \\ y(t) &= \sin(t) - a\cos(bt) \end{aligned} \tag{13}$$

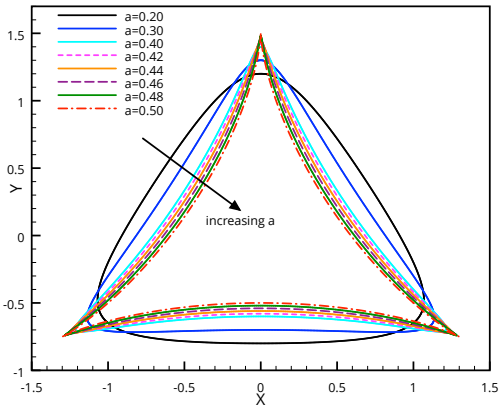

Figure 5: The shape parametrization for which the error order and convergence plots are discussed further on

In all our examples, we fix $b = -2.0$, but change a to introduce a shape singularity (sharpness) as a increases methodically from $a = 0.2$ to 0.5. The shape parametrization is shown in fig. 5.

It can be conclusively seen that increasing the parameter a gradually to 0.5 introduces a shape singularity and so the solution is expected to degrade. We run the simulations with the same boundary conditions for all the cases and observe the error behavior, order of convergence and the number of iterations taken to solve the system. For doing so we fix the other parameters to `strength` = 100, `fund_soln_loc = 0.0`, `mesh_order` = 4, `target_order` = 4, `qbx_order` = 3, `fmm_order=10`, $\mu = 1$. The convergence behavior is seen in fig. 6, where the empirical convergence is also listed besides the chart. It can be seen that the empirical order of convergence decreases steadily and then stagnates as the shape singularity increases.

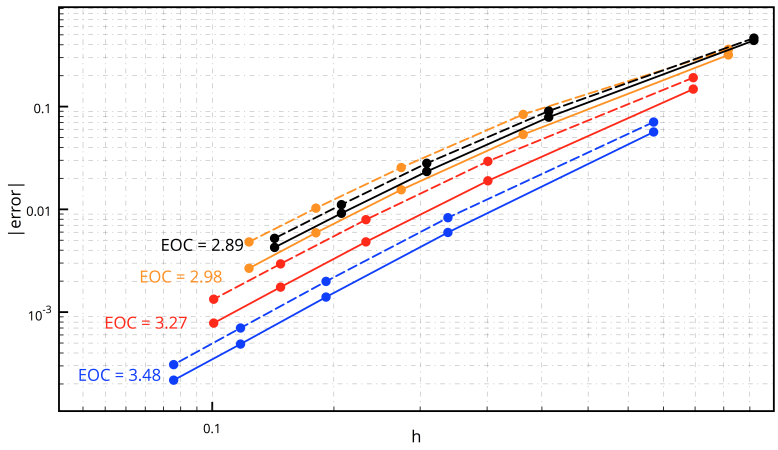

Figure 6: The trends of convergence for the different shapes listed above. Shown are the error curves corresponding to $a = 0.2$ (blue), $a = 0.3$ (red), $a = 0.4$ (orange) and $a = 0.48$ (black). The solid lines represent the error measured by L2 norm while the dashed lines represent the L-inf norm error

We now concentrate on the time-to-solution, represented by the time to solve the system using GMRES. The number of GMRES iterations for each value of a is shown in fig. 7 for a constant tolerance of $1e-9$. The variation for each parameter is depicted using error bars. It can be seen that for same residual order in GMRES, the amount of iterations needed grows exponentially! Our constructed operator then has more singular values that do not decay rapidly, indicating that a careful design of solution is needed in case we deal with sharp corners and domains. Thus with sharper domains, we tend to lose accuracy while increasing the time to solution.

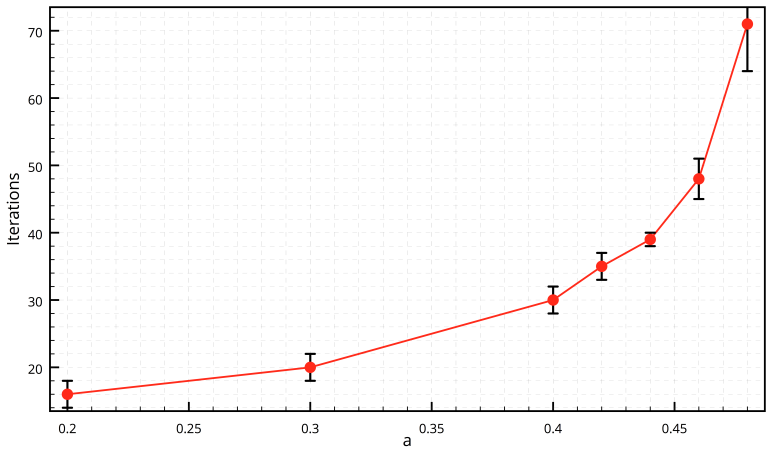

Figure 7: The number of iterations needed in GMRES to satisfy $\texttt{tol} = 1e-9$, for different values of a (more sharp as a increases)

5 Conclusion

In conclusion, I have surveyed and understood the application of integral equation theory to the Stokes PDE completely - covering solution representation, existence, uniqueness, altering exterior dirichlet null-space and application to an external dirichlet problem. Through [13], the stokes PDE could be solved in two dimensions for a toy problem that incorporates all the physics. The solution was validated and its error properties observed, on changing the simulation parameters. Due to the importance of shape singularities in such flows, I attempted to understand the degradation of solution properties with increasing (but controlled) sharpness of a representative shape. I wish to extend the solver capabilities in the future to realistically simulate Stokes flows.

References

[1] George Keith Batchelor. *An introduction to fluid dynamics*. Cambridge university press, 2000.

[2] Constantine Pozrikidis. *Boundary integral and singularity methods for linearized viscous flow*. Cambridge University Press, 1992.

[3] James H Bramble. "A proof of the inf–sup condition for the Stokes equations on Lipschitz domains". In: *Mathematical Models and Methods in Applied Sciences* 13.03 (2003), pp. 361–371.

[4] SA Nazarov and K Pileckas. "On the solvability of the Stokes and Navier-Stokes problems in the domains that are layer-like at infinity". In: *Journal of Mathematical Fluid Mechanics* 1.1 (1999), pp. 78–116.

[5] George Biros, Lexing Ying, and Denis Zorin. "A fast solver for the Stokes equations with distributed forces in complex geometries". In: *Journal of Computational Physics* 193.1 (2004), pp. 317–348.

[6] Dhairya Malhotra, Amir Gholami, and George Biros. "A volume integral equation stokes solver for problems with variable coefficients". In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE Press. 2014, pp. 92–102.

[7] Ludvig af Klinteberg and Anna-Karin Tornberg. "A fast integral equation method for solid particles in viscous flow using quadrature by expansion". In: *Journal of Computational Physics* 326 (2016), pp. 420–445.

[8] Eduardo Corona, Leslie Greengard, Manas Rachh, and Shravan Veerapaneni. "An integral equation formulation for rigid bodies in Stokes flow in three dimensions". In: *Journal of Computational Physics* 332 (2017), pp. 504–519.

[9] Young Ok Choi and Johannes Tausch. "The Galerkin boundary element method for transient Stokes flow". In: *Advances in Computational Mathematics* 43.3 (2017), pp. 473–493.

[10] Olga A Ladyzhenskaya. *The mathematical theory of viscous incompressible flow*. Vol. 12. 3. Gordon & Breach New York, 1969.

[11] George C Hsiao and Rainer Kress. "On an integral equation for the two-dimensional exterior Stokes problem". In: *Applied Numerical Mathematics* 1.1 (1985), pp. 77–93.

[12] Friecrich-Karl Hebeker. "Efficient boundary element methods for three-dimensional exterior viscous flows". In: *Numerical Methods for Partial Differential Equations* 2.4 (1986), pp. 273–297.

[13] Andreas Klöckner. *pytential*. `https://github.com/inducer/pytential`. 2017.

[14] Andreas Klöckner, Alexander Barnett, Leslie Greengard, and Michael O'Neil. "Quadrature by expansion: A new method for the evaluation of layer potentials". In: *Journal of Computational Physics* 252 (2013), pp. 332–349.