University of Illinois, Urbana-Champaign

A Fast Stokes Solver

Isuru Fernando

December 21, 2017

1 Introduction to Stokes PDEs

In this report, we will look at a fast solver for Stokes Partial Differential Equations (PDEs) using integral equation methods. First let's look at what the Stokes PDEs are.

Stokes PDEs are a set of PDEs defined in vector form as

$$\mu\nabla^2 u - \nabla p + f = 0 \tag{1.1}$$
$$\nabla \cdot u = 0 \tag{1.2}$$

where u is the velocity vector, μ is the viscosity, p is the pressure and f is the force.

Here we are interested in the 2D and 3D problems. i.e. when u is a 2D or 3D vector. Writing these equations for 3D without the vector form is as follows.

$$\mu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right) - \frac{\partial p}{\partial x} + f_x = 0 \tag{1.3}$$
$$\mu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2}\right) - \frac{\partial p}{\partial y} + f_y = 0 \tag{1.4}$$
$$\mu\left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2}\right) - \frac{\partial p}{\partial z} + f_z = 0 \tag{1.5}$$
$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \tag{1.6}$$

1.1 Green's Function solution: The Stokeslet

A fundamental solution to the Stokes PDEs is the Stokeslet solution.

$$u(r) = F \cdot \mathbb{J}(r) \tag{1.7}$$

$$p(r) = \frac{F \cdot r}{4\pi |r|^3} \tag{1.8}$$

where

$$r = \begin{bmatrix} x & y & z \end{bmatrix} \tag{1.9}$$

$$\mathbb{J}(r) = \frac{1}{8\pi\mu}\left(\frac{\mathbb{I}}{|r|} + \frac{rr}{|r|^3}\right) \tag{1.10}$$

which can be written as a Matrix-Vector product

$$\begin{bmatrix} u \\ v \\ w \\ p \end{bmatrix} = \frac{1}{8\pi\mu|r|^3}\begin{bmatrix} |r|^2 + x^2 & xy & xz \\ xy & |r|^2 + y^2 & yz \\ xz & yz & |r|^2 + z^2 \\ 2\mu x & 2\mu y & 2\mu z \end{bmatrix}\begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} \tag{1.11}$$

2 Implementation

2.1 Current Implementation

Current implementation of the Stokeslet solutions in pytential uses 9 kernel functions. Since 3 out of the 12 kernels are repeated, there are only 9 independent kernels.

Drawbacks of this method are,

- Redundant computations
- Takes more storage than necessary
- Overhead of spinning up 9 kernel evaluations

2.2 New implementation

The new implementation computes all the kernels in one go, leading to the kernel functions and its derivatives being compiled in one function. This leads to compiler optimization called Common Subexpression Elimination (CSE).

2.2.1 Common Subexpression Elimination

Although the nine kernels are independent of each other, the derivatives of the kernels has common terms among them. Conside $\frac{\partial^3}{\partial y^3}\left(\frac{xy}{|r|^3}\right)$ and $\frac{\partial^3}{\partial y^3}\left(\frac{yz}{|r|^3}\right)$. If the two expressions were evaluated separately, it would take 36 arithmetic operations.

$$x_0 = y^2 \tag{2.1}$$

$$x_1 = x^2 + x_0 + z^2 \tag{2.2}$$

$$\frac{\partial^3}{\partial y^3}\left(\frac{xy}{|r|^3}\right) = \frac{3x}{x_1^{\frac{5}{2}}}\left(\frac{30x_0}{x_1} - 3 - \frac{35y^4}{x_1^2}\right) \tag{2.3}$$

$$x_2 = y^2 \tag{2.4}$$

$$x_3 = x^2 + x_2 + z^2 \tag{2.5}$$

$$\frac{\partial^3}{\partial y^3}\left(\frac{zy}{|r|^3}\right) = \frac{3z}{x_3^{\frac{5}{2}}}\left(\frac{30x_2}{x_3} - 3 - \frac{35y^4}{x_3^2}\right) \tag{2.6}$$

If the two expressions were evaluated combined, common subexpressions of the two expression trees can be evaluated only once and then reused. It would take 19 arithmetic operations, which saves about 50% of operations. Following shows how the two derivatives are evaluated,

$$x_0 = y^2 \tag{2.7}$$

$$x_1 = x^2 + x_0 + z^2 \tag{2.8}$$

$$x_2 = \frac{3}{x_1^{\frac{5}{2}}}\left(\frac{30x_0}{x_1} - 3 - \frac{35y^4}{x_1^2}\right) \tag{2.9}$$

$$\frac{\partial^3}{\partial y^3}\left(\frac{xy}{|r|^3}\right) = xx_2 \tag{2.10}$$

$$\frac{\partial^3}{\partial y^3}\left(\frac{yz}{|r|^3}\right) = x_2 z \tag{2.11}$$

2.2.2 Reducing the number of kernels in local expansion

Let x be the targets, y be the sources, ϕ be the kernel and f be the charge/force

$$u(x,y) = f(y)\phi(x-y) \approx \sum_{|p|\le k} \frac{D_x^p f(y)\phi(x-y)|_{x=c}}{p!}(x-c)^p$$

Here $\frac{f(y)D_x^p\phi(x-y)|_{y=c}}{p!}$ depends on source and center, while $(x-c)^p$ depends on target and center

In the Stokes equations case u is a linear combination of 3 functions ϕ_1, ϕ_2, ϕ_3 and there are 3 charges f_1, f_2, f_3

$$u(x-y) = \sum_{i=1}^{3} f_i(y)\phi_i(x-y)$$
$$\approx \sum_{|p|\le k} \frac{D_x^p\left(\sum_{i=1}^3 f_i(y)\phi_i(x-y)\right)|_{x=c}}{p!}(x-c)^p$$

Therefore the output of the local expansion can be 4 polynomials instead of 12 polynomials. This leads to a significant reduction in computations in local-to-local expansions as we deal only with 4 polynomials.

2.2.3 Reducing the number of kernels in multipole expansion

Let x be the targets, y be the sources, ϕ be the kernel and f be the charge/force

$$f(y)\phi(x-y) \approx f(y)\sum_{|p|\le k} \frac{D_x^p\phi(x-y)|_{y=c}}{p!}(y-c)^p$$

or equivalently

$$f(y)\phi(x-y) \approx \sum_{|p|\le k} \frac{D_x^p\phi(x-y)|_{y=c}}{p!}f(y)(y-c)^p$$

Here $\frac{D_x^p\phi(x-y)|_{y=c}}{p!}$ depends on target and center, while $f(y)(y-c)^p$ depends on source and center

In the Stokes equation case consider the first velocity component u which is a linear combination of 3 functions ϕ_1, ϕ_2, ϕ_3 and there are 3 charges f_1, f_2, f_3

$$u(x-y) = \sum_{i=1}^{3} f_i(y)\phi_i(x-y) \approx \sum_{i=1}^{3}\left(f_i(y)\sum_{|p|\le k} \frac{D_x^p\phi_i(x-y)|_{y=c}}{p!}(y-c)^p\right)$$
$$u(x-y) \approx \sum_{|p|\le k}\left(\sum_{i=1}^{3} \frac{D_x^p\phi_i(x-y)|_{y=c}}{p!}f_i(y)(y-c)^p\right)$$

Note that this expression cannot be written as before to reduce the number of polynomials to 4 and therefore we need 12 polynomials and the calculations cannot be reduced further. (3 of which are reused)

2.2.4 Computing multipole-to-local translation for Stokes

As the new implementation uses 12 kernels for multipole expansion and 4 kernels for local expansion, we need to revise translation from multipole expansion to local expansion.
Let $\phi_{1,1}, \phi_{1,2}, \phi_{1,3}, \phi_{2,1}, \ldots, \phi_{3,3}$ be the 12 multipole kernels.

Then using the following equations we can translate from multipole to local.

$$u = \phi_{1,1} + \phi_{1,2} + \phi_{1,3}$$
$$v = \phi_{2,1} + \phi_{2,2} + \phi_{2,3}$$
$$w = \phi_{3,1} + \phi_{3,2} + \phi_{3,3}$$
$$p = \phi_{4,1} + \phi_{4,2} + \phi_{4,3}$$

2.2.5 Derivative reduction in local expansion

In local expansion, a CSE algorithm is used in eliminating common subexpressions, but the algorithm used is a heuristic and does not find the optimal reduction. Therefore this process can be further improved by using common subexpressions we know. For example, the PDE gives a relationship among the derivatives and this can be used for replacing one derivative using a combination of other derivatives.

Let's see how we can use derivative reduction for Stokes.

$\frac{\partial(1.3)}{\partial x} + \frac{\partial(1.4)}{\partial y} + \frac{\partial(1.5)}{\partial z}$ gives us

$$\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} + \frac{\partial^2 p}{\partial z^2} = 0 \tag{2.12}$$

Pressure term satisfies the Laplace equation! For laplace equation, derivatives with z terms can be rewritten using derivatives with x, y terms and one or less z term. Figure 2.1 shows the coefficients as a tetrahedron and all the coefficients can be rewritten in terms of the coefficients represented in the base of the tetrahedron.

Using (2.12) we can obtain,

$$\frac{\partial^{a+b+2} p}{\partial x^{a+2} y^b} + \frac{\partial^{a+b+2} p}{\partial x^a y^{b+2}} + \frac{\partial^{a+b+2} p}{\partial x^a y^b z^2} = 0 \tag{2.13}$$

$$\frac{\partial^{a+b+2} p}{\partial x^a y^b z^2} = -\frac{\partial^{a+b+2} p}{\partial x^{a+2} y^b} - \frac{\partial^{a+b+2} p}{\partial x^a y^{b+2}} \tag{A}$$

Using (1.3) we can obtain,

$$\frac{\partial^{a+b+2} u}{\partial x^a y^b z^2} = \frac{1}{\mu}\left(\frac{\partial^{a+b+1} p}{\partial x^{a+1} y^b} - \frac{\partial^{a+b+2} u}{\partial x^{a+2} y^b} - \frac{\partial^{a+b+2} u}{\partial x^a y^{b+2}}\right) \tag{B}$$

Using (1.4) we can obtain,

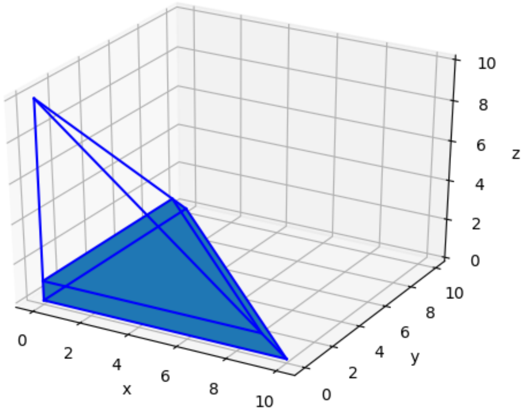

Figure 2.1: Representation of coefficients of the taylor series

$$\frac{\partial^{a+b+2} v}{\partial x^a y^b z^2} = \frac{1}{\mu}\left(\frac{\partial^{a+b+1} p}{\partial x^a y^{b+1}} - \frac{\partial^{a+b+2} v}{\partial x^{a+2} y^b} - \frac{\partial^{a+b+2} v}{\partial x^a y^{b+2}}\right) \tag{C}$$

Using (1.6) we can obtain,

$$\frac{\partial^{a+b+1} w}{\partial x^a y^b z} = -\frac{\partial^{a+b+1} u}{\partial x^{a+1} y^b} - \frac{\partial^{a+b+1} v}{\partial x^a y^{b+1}} \tag{D}$$

Equations $(A), (B), (C), (D)$ is used for reducing the z term of the derivatives

2.2.6 Derivative reduction in multipole expansion

In multipole expansion, the coefficients are the multi-index powers. Using the derivative recurrences, we can reduce the number of coefficients needed and therefore save some computations.

u, v, w, p in 1.11 satisfies the Stokes PDEs for any f_x, f_y, f_z.

Setting $f_x = 1$, $f_y = 0$ and $f_z = 0$ gives us

$$\begin{bmatrix} u_0 \\ v_0 \\ w_0 \\ p_0 \end{bmatrix} = \frac{1}{8\pi\mu|r|^3}\begin{bmatrix} |r|^2 + x^2 \\ xy \\ xz \\ 2\mu x \end{bmatrix} \tag{2.14}$$

u_0, v_0, w_0, p_0 satisfies the Stokes PDEs and therefore the relationships of the 4 kernels can be extended to the 12 kernels for use in multipole expansion.

3 Project Status & Future Work

Currently, the stokes solver is being integrated into Sumpy, a library that provides symbolic code generators for multipole and local expansions and translations. Following operations work in the Stokes PDE solver.

- Multipole expansion
- Local expansion
- Local-to-local translation
- Multipole-to-multipole translation
- Multipole-to-local translation

Future work includes

- StokesDerivativeWrangler to reduce derivatives
- Stresslet kernels