HPC Final Project

May 16, 2019

1 Algorithm description

Periodic Stokes potentials Given the Stokes equation, the Green's function for the velocity field for the free-space problem is given by the Oseen-Burgers tensor:

$$\mathbf{S}(\mathbf{x}) = \frac{\mathbf{1}}{|\mathbf{x}|} + \frac{\mathbf{x}\mathbf{x}}{|\mathbf{x}|^3}. \tag{1.1}$$

Now we consider a system of N point sources at location $\mathbf{x}_n$ with strength $\mathbf{f}_n$, in the periodic setting, the velocity field is given as

$$\mathbf{u}(\mathbf{x}) = \sum_{n=1}^{N} \sum_{\mathbf{p}} \mathbf{S}(\mathbf{x} - \mathbf{x_n} + \mathbf{p})\mathbf{f}_n, \tag{1.2}$$

where $\mathbf{p}$ form the discrete set $\{[iL_x \; jL_y \; kL_z] : (i, j, k) \in \mathbb{Z}^3\}$ and L_x, L_y and L_z are the periodic lengths in the three directions.

Ewald summation for Stokes Applying the idea of Ewald decomposition, the Eq.(1.2) can be split as following:

$$\mathbf{u}(\mathbf{x}_m) = \sum_{n=1}^{N} \sum_{\mathbf{p}} \mathbf{A}(\xi, \mathbf{x_m} - \mathbf{x_n} + \mathbf{p})\mathbf{f}_n + \frac{1}{V} \sum_{\mathbf{k} \neq 0} \mathbf{B}(\xi, \mathbf{k}) e^{-k^2/4\xi^2} \sum_{n=1}^{N} \mathbf{f}_n e^{-i\mathbf{k}\cdot(\mathbf{x}_m - \mathbf{x}_n)} - \mathbf{u}_{\text{self}} \tag{1.3}$$

where $\mathbf{k} \in \{[2\pi k_i/L_i] : k_i \in \mathbb{Z}, i = 1, 2, 3\}, k = |\mathbf{k}|, V = L_x L_y L_z$ and ξ is a positive constant known as the Ewald parameter. From the Eq.(1.3) we can see that the velocity field has been split into three parts: one sum in real space ($\mathbf{u}^R$), one sum in frequency domain ($\mathbf{u}^F$) and a self-contribution $\mathbf{u}_{\text{self}}$.

From the formulation by Hasimoto, we have

$$\mathbf{A}(\xi, \mathbf{x}) = 2\left(\frac{\xi e^{-\xi^2 r^2}}{\sqrt{\pi} r^2} + \frac{\text{erfc}(\xi r)}{2r^3}\right)(r^2\mathbf{I} + \mathbf{x}\mathbf{x}) - \frac{4\xi}{\sqrt{\pi}} e^{-\xi^2 r^2}\mathbf{I} \tag{1.4}$$

with $r = |\mathbf{X}|$ and

$$B(\xi, \mathbf{k}) = 8\pi\left(1 + \frac{k^2}{4\xi^2}\right)\frac{1}{k^4}(k^2\mathbf{I} - \mathbf{k}\mathbf{k}) \tag{1.5}$$

and

$$\mathbf{u}_{\text{self}}(\mathbf{x}_m) = \frac{4\xi}{\sqrt{\pi}}\mathbf{f}_m \tag{1.6}$$

Nonuniform fast Fourier transform The nonuniform discrete Fourier transform (NuFFT) of type 1 and 2 is defined as:

$$F(\mathbf{k}) = \frac{1}{N}\sum_{n=1}^{N} f_j e^{-i\mathbf{k}\cdot\mathbf{x}_n} \tag{1.7}$$

and

$$f(x_n) = \sum_{\mathbf{k}} F(\mathbf{k}) e^{i\mathbf{k}\cdot\mathbf{x}_n} \tag{1.8}$$

They can be computed efficiently by fast Gaussian gridding and FFT, we will introduce it in the later part.

Fast summation method in frequency domain Now we focus on the frequency domain part of the summation based on NuFFT:

$$\mathbf{u}^F(\mathbf{x}_m) = \frac{1}{V}\sum_{\mathbf{k}\neq 0}\mathbf{B}(\xi, \mathbf{k}) e^{-k^2/4\xi^2}\sum_{n=1}^{N}\mathbf{f}_n e^{-i\mathbf{k}\cdot(\mathbf{x}_m - \mathbf{x}_n)} \tag{1.9}$$

The formula can be split into

$$\mathbf{u}^F(\mathbf{x}_m) = \frac{1}{V}\sum_{\mathbf{k}\neq 0}\mathbf{B}(\xi, \mathbf{k}) e^{-k^2/4\xi^2}\left(\sum_{n=1}^{N}\mathbf{f}_n e^{i\mathbf{k}\cdot\mathbf{x}_n}\right) e^{-i\mathbf{k}\cdot\mathbf{x}_m} \tag{1.10}$$

which is one NuFFT (type 1) combined with scaling and another NuFFT (type 2).

Reformulation by Gaussian Gridding & FFT The basic idea is convolution by the Gaussian... leave it for later.

For NuFFT of type 1, introducing a free parameter η, we have that

$$\sum_{n=1}^{N} \mathbf{f}_n e^{i\mathbf{k}\cdot\mathbf{x}_m} = e^{\eta k^2/8\xi^2} \sum_{n=1}^{N} \mathbf{f}_n e^{-\eta k^2/8\xi^2} e^{-i(-\mathbf{k})\cdot\mathbf{x}_m} := e^{\eta k^2/8\xi^2} \hat{H}_{-\mathbf{k}} \tag{1.11}$$

while $\hat{H}_{\mathbf{k}}$ is the Fourier transform of

$$H(\mathbf{x}) = \left(\frac{2\xi^2}{\pi\eta}\right)^{3/2} \sum_{n=1}^{N} \mathbf{f}_n e^{-2\xi^2|\mathbf{x}-\mathbf{x}_n|_*^2/\eta} \tag{1.12}$$

where $|\cdot|_*$ denotes distance to closest periodic image. (*Remark from Guanchun: The distance is a little weird, I haven't checked if it's the Fourier transform or not. Is it common in periodic setting?*)

For the part of type 2 NuFFT, under the same parameter η, first we can simplify

$$\mathbf{u}^F(\mathbf{x}_m) = \frac{1}{V} \sum_{\mathbf{k}\neq 0} \mathbf{B}(\xi, \mathbf{k}) e^{-k^2/4\xi^2} e^{\eta k^2/8\xi^2} \hat{H}_{-\mathbf{k}} e^{-i\mathbf{k}\cdot\mathbf{x}_m} \tag{1.13}$$

$$= \frac{1}{V} \sum_{\mathbf{k}\neq 0} \hat{\hat{H}}_{\mathbf{k}} e^{-\eta k^2/8\xi^2} e^{i\mathbf{k}\cdot\mathbf{x}_m} \tag{1.14}$$

with

$$\hat{\hat{H}}_{\mathbf{k}} := \mathbf{B}(\xi, -\mathbf{k}) \hat{H}_{\mathbf{k}} e^{-(1-\eta)k^2/4\xi^2} \tag{1.15}$$

Then we denote $\tilde{H}(\mathbf{x}_i)$ as the inverse Fourier transform of $\hat{\hat{H}}_{\mathbf{k}}$.

According to the convolution argument, we have that

$$\mathbf{u}^F(\mathbf{x}_m) = \left(\frac{2\xi^2}{\pi\eta}\right)^{3/2} \int_{\Omega} \tilde{H}(\mathbf{x}) e^{-2\xi^2|\mathbf{x}-\mathbf{x}_m|_*^2/\eta} \mathrm{d}\mathbf{x} \tag{1.16}$$

$$\approx \frac{V}{N_{\text{grid}}} \left(\frac{2\xi^2}{\pi\eta}\right)^{3/2} \sum_{\mathbf{x}_{(i)} \text{ in equi-space grid}} \tilde{H}(\mathbf{x}_{(i)}) e^{-2\xi^2|\mathbf{x}_{(i)}-\mathbf{x}_m|_*^2/\eta} \tag{1.17}$$

The approximation integral is spectrally accurate since the integrand is periodic.

Now the only left problem is how to compute Eq.(1.12) and Eq.(1.17). The formula is known as **Gaussian gridding**.

Fast Gaussian Gridding The idea of fast Gaussian gridding is pre-compute and store the exponential and only do necessary multiplications.

Roughly speaking, for 1D situation, we have that ($x_{(i)} = ih$)

$$e^{-\alpha|x_{(i)}-x_n|^2} = e^{-\alpha|ih-x_n|^2} = e^{-\alpha(ih)^2}\left(e^{2\alpha h x_n}\right)^i e^{-\alpha x_n^2} \tag{1.18}$$

All $e^{-\alpha(ih)^2}, e^{2\alpha h x_n}, e^{-\alpha x_n^2}$ can be precomputed and then each time we only need to do some multiplications.

Remark from Guanchun: It's said to be 5 to 10 times faster in 2D than naive method and will perform better in higher dimension. However, I was wondering if we do the naive matrix multiplication on GPU, maybe it's still fast enough?

The Description of Algorithm The algorithm is conducted in the following way:

(1) Choose the free parameter ξ and η wisely and construct the uniform grid according to the problem.

(2) Evaluate $H(\mathbf{x})$ on the grid according to Eq.(1.12) with fast Gaussian gridding.

(3) Conduct FFT on $H(\mathbf{x})$ to get $\hat{H}_{\mathbf{k}}$.

(4) Apply the scaling according to Eq.(1.15) to get $\hat{\tilde{H}}_{\mathbf{k}}$.

(5) Conduct iFFT on $\hat{\tilde{H}}_{\mathbf{k}}$ to get $\tilde{H}(\mathbf{x})$.

(6) Evaluate the $\mathbf{u}^F(\mathbf{x}_m)$ according to Eq.(1.17) with fast Gaussian gridding.

(7) Evaluate the $\mathbf{u}^R(\mathbf{x}_m)$ in real space and get the final result.

Pseudo-code