A FFT-based convolution method for mix-periodic Boundary Condition

Zhe Chen*[1]

[1]Courant Institute of Mathematical Sciences, New York University

Oct. 2019

1 FFT-based convolution

Convolution is quite common in computational Mathematics, cheap and accurate numerical method of convolution is thus requisite. Take 2D convolution for example, direct discrete convolution would be $O(N^4)$ for $N \times N$ mesh grid, which is unaffordable. FFT method is usually the way to cut down complexity from $O(N^4)$ to $O(N^2 log(N))$. A basic theorem is circulant convolution theorem, which states that discrete fourier transform of circulant convolution is dot product of their accordingly discrete fourier transform. For two equal-length vector $x = (x_n),\ y = (y_n)$,

$$\mathcal{F}(x * y) = \mathcal{F}(x)\ \cdot\ \mathcal{F}(y),$$

where $\cdot$ denotes inner product, $\mathcal{F}(\cdot)$ demotes discrete fourier transform and $*$ denotes circulant convolution.

However, in practical situation, it is not always originally standard circulant convolution method. Periodicity of the two function would vary. If the function is in more than one dimension, periodicity would be more complicated. For example, consider convolving two function, ρ and K, in $\mathbb{R}$. There would be situations like following,

1. ρ is periodic but K is a-periodic,
2. both ρ and K is a-periodic, ρ is finite supported.

1.1 Case 1.

Assume ρ is periodic in $[0, L]$ and K is aperiodic in $\mathbb{R}$, $\rho_n = \rho(L/N \cdot n)$ $K_n = K(L/N \cdot n)$. By periodicity of ρ, convolution of these two function would be written as form of periodic summation,

$$v_n = (\rho * K)_n = \sum_{l=-\infty}^{+\infty} \rho_l K_{n-l} = \sum_{l=0}^{N-1} \rho_l K_{n-l}^N, \tag{1}$$

where $K_n^N = \sum_{p=-\infty}^{\infty} K(n - pN)$ is periodic summation.

1.1.1 Case 1.(a)

Thus, circulant discrete convolution of K^N and ρ is equivalent of convolution of K and ρ in $\mathbb{R}$. Except for computing FFT, extra computational effort is taken into periodic summation. For K that is finite supported, only several images are needed to compute periodic summation, even no image boxes needed if K is only supported in $[0, L]$. If K decays fast enough, we could also treat it as finite supported since periodic summation converges quickly.

*zc1291@nyu.edu

1.1.2 Case 1.(b)

However, for K that decays slowly, this method would not be good since periodic summation converges slowly. If we luckily know analytic Fourier transform of K, however, we could evaluates the continue FT at DFT modes $\frac{2\pi}{L}[-N/2, \ldots, -1, 0, 1, \ldots, N/2-1]$. Disadvantage is that we need enough modes to resolve this Kernel in fourier space if this Kernel has high frequency component.

1.2 Case 2.

For aperiodic ρ and K, circulant discrete convolution does not seemly work. However, there's a zero padding trick to make it work. Consider ρ is finite supported in $[0, L]$ and K is not finite supported. If we zero-padding N-length $[\rho_0, ..., \rho_{N-1}]$ to be $(2N-1)$-length vector $\tilde{\rho} = [\rho_0, ..., \rho_{N-1}, 0, ..., 0]$, let $\tilde{K} = [K_0, ..., K_{N-1}, K_{-(N-1)}, ..., K_{-1}]$. Then we do circulant discrete convolution $\tilde{\rho} * \tilde{K}$ and take the first N component. That's exactly v. By this zeros-padding trick, convolution of two aperiodic function, one of which is finite supported, could be solved by circulant discrete convolution and then solved quickly by FFT. Actually, in practical situation, I double zero-padding it to $2N$ length rather than $2N-1$ to keep length of vector to be power of 2, which benefits FFT algorithm. Then $\tilde{K} = [K_0, ..., K_{N-1}, K_{-N}, K_{-(N-1)}, ..., K_{-1}]$ this time.

2 Numerical results

For a colloidal system near a wall, many interesting collective phenomenon would happen as a result of long-range and linearity of Stokes-flow. Here's a video from physical experiments. Those very small colloidal particles have equal radius a and almost at the same height h_g, which is only several a, as a result of balance between Brownian motion and gravity. At $t = 0$, they are uniformly distributed in a rectangle strip and given a uniform force in x-direction. Then they moves in this viscous fluid with low Reynold number. Although it's uniform in y at the beginning, it gradually develop some structure in y, which interests us to simulate it numerically.

Thus, in numerical model, we consider boundary condition to be periodic in y but a-periodic in x. In the domain $Lx \times Ly$, it's equal-space meshed. Density ρ is given on center point $x_m = (m - 1/2)h,\ y_n = (n - 1/2)h,\ m, n = 1, ..., N$. Initial condition is given that $\rho = \rho_0$ in a strip $Lx_{init} \times Ly$ at the left of the domain. Apply uniform force f in +x direction. Velocity at left and bottom edge of each mesh grid need to be computed at each time step. Thus, we should get v_x and v_y at following points.

$$\begin{aligned} (v_x)_{m,n} &= v_x(x_m - h/2, y_n) \\ (v_y)_{m,n} &= v_y(x_m, y_n - h/2) \end{aligned} \tag{2}$$

In this many-body hydrodynamics system, formula of velocity for given ρ_t could be built from Stokeslet, which is Green function of Stokes Equation for two point in infinity open space. We first give interaction of two particles through RPY kernel, which is actually integral of Stokeslet through two balls' surface. Then, a corrector to RPY kernel for half-space BC is added. Here we use Swan-Brady's construction, which actually uses reflection to satisfy BC at the wall. Thus, velocity is given by convolution of density ρ and Green function, i.e. Swan-Brady kernel, K, whose deriving process is shown in this Mathematica worksheet. However, this so-called RPY-Swan kernel is very complicated and long-ranged. And periodicity of this problem is mixed. It needs be to carefully treated to use FFT-based convolution.

We used to use direct convolution method, paralleled by GPU, to compute velocity. The periodic BC in y is mimicked by adding finite many periodic boxes on both side in y of original domain. However, this method is expensive, $O((MN^2)^2)$ for M image boxes, and converges slowly because of slow decay of RPY-Swan kernel.

For discrete convolution method itself, it has spectral accuracy in spatial step h for two analytic function and 1st-order convergence for function that has jump discontinuity, which is a trivial conclusion and we've verified before. The main problem should be the slow decay of K. In this report, I'm trying to take advantage of FFT to build a fast and cheap method to compute this convolution that has long-range kernel and is mix-periodic.

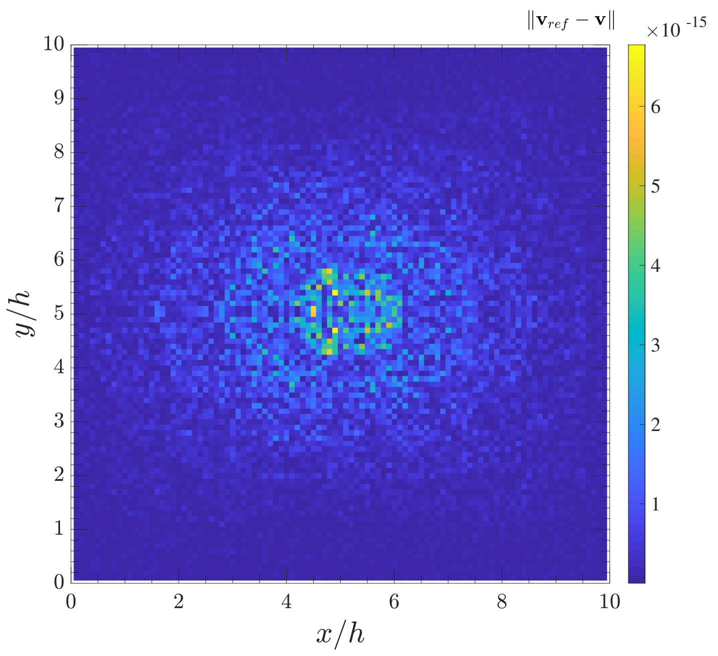


Figure 1: Error of velocity between FFT method and GPU-paralleled direct convolution method.

2.1 Method 1.

First, we could easily develop a method that is equivalent with direct convolution that uses extra periodic boxes, but uses FFT instead.

In y-direction, it's periodic for ρ but aperiodic for K. Hence Case 1 is suitable. If we truncate periodic summation in eq.(1) at M on both sides, $K_n^N = \sum_{p=-M}^{M} K(n-pN)$, then this circulant discrete convolution in eq.(1) is equivalent with direct convolution that has M image boxes on both sides.

In x direction, it's case 2, so the algorithm is like following,

1. Double zero-padding in x direction as what case 2 does.
2. Do periodic summation eq.(1) in x direction for K up to M images on both sides. $K_n^N = \sum_{p=-M}^{M} K(n-pN)$. One should notice that Kernel is evaluated at different place for v_x and v_y as stated in eq.(2).
3. Use 2D FFT to compute this 2D circulant discrete convolution.
4. Take the first $Nx - by - Ny$ component as result.

To verify equivalence of these two method, I did a numerical test. Set $a = 0.01,\ h_g = 1,\ Lx = Ly = 10$, and uses one image box on both sides, i.e. $M = 1$. Density function ρ is set to be gaussian $\rho(x,y) = \exp\left(-3((x-Lx/2)^2 + (y-Ly/2)^2)Lx\right)$. It turns out that difference of results of these two methods is machine error (fig.1). However, computing complexity has a huge difference. FFT method decrease it from $O((MN^2)^2)$ to $O(N^2(log(N)+M))$ and they are equivalent. Thus, FFT method could at-least replaces direct convolution method.

Another convergence test we should do is about number of periodic boxes M. This would help us understand how slow decaying of Kernel would have a effect on accuracy of this method. Set $Nx = Ny = 128$ and keep other parameters the same as before and increase periodic boxes number. The emphirical error converges in 4th order as fig.2 shows. As we know that convergence in spatial step h is spectral, slow decay of kernel K is the main problem to compute velocity accurately. In the following, we'll develop a method that is far more accurate and converges fast but also has only FFT complexity.

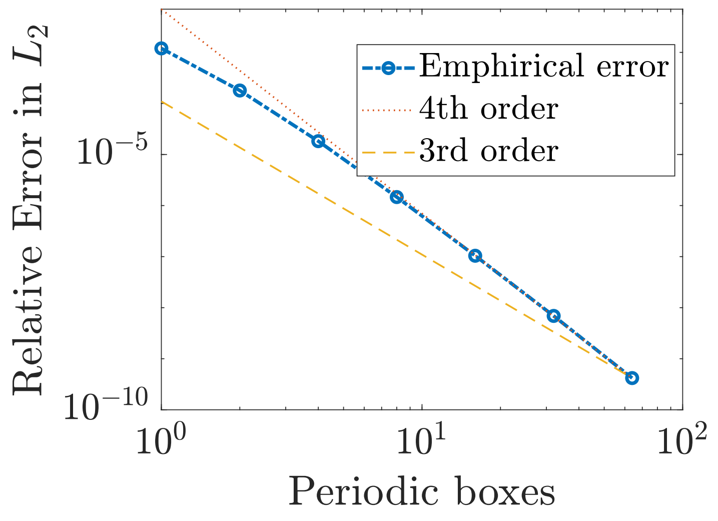


Figure 2: Emphirical error of velocity in L_2. Convergence of periodic boxes number M is 4th order

2.2 Method 2.

This section is to think about how to deal with the slow-decayed kernel. As case 1.(b) stated, if analytic fourier transform of the Kernel is known, it could be used into FFT to compute convolution of periodic ρ with long-ranged K. However, RPY-Swan kernel is too complicated to get analytic fourier transform, which is disappointing. Fortunately, for Rotne-Blake kernel, which is actually limit of RPY-Swan as $a \to 0^+$, has analytic fourier transform in y both for K_x and K_y. They are asymptotically the same as $(x, y) \to \infty$, which is a natural result from that it makes no difference at long distance for finte radius a or $a \to 0^+$. The way we develop its FT in y direction in Maple is shown in this Maple sheet. Hence, we can split the Kernel into long-ranged part Rotne-Blake kernel and short-ranged kernel as following,

$$K_{RPY} = (K_{RPY} - K_{RB}) + K_{RB}$$

Now, it's feasible to use Method 1 to deal with short-ranged part $K_{RPY} - K_{RB}$ numerically because it decays super fast. For long-ranged part, we should use the idea in case 1.(b). in y direction. In x direction, it's the same as Method 1. A zeros-padding is needed in x direction. The algorithm is like following,

1. Double zero-padding in x direction as what case 2 does.
2. Evaluate FT in y of K, $\mathcal{F}_y(K)$, at x-grid and $kk = \frac{2\pi}{Ly}[0, \ldots, Ny/2-1, -Ny/2, \ldots, -1]$
3. Do a drift $\mathcal{F}_y(K)_n = 1/dy \cdot \exp(i \cdot kk \cdot y_0) \cdot \mathcal{F}_y(K)_n$, where y_0 is where the first point in y-grid that velocity is evaluated, to match it to form of discrete fourier transform. For x velocity, $y_0 = 0$. For y velocity, $y_0 = -h/2$.
4. Do FFT of $\mathcal{F}_y(K)$ in x and do an inner-product with 2D FFT of ρ. Then do IFFT.
5. Take the first $Nx - by - Ny$ component as result.

Numerical verification

To verify consistence of this splitting method, I did the following tests.

Here's example of x direction component of RPY-Swan kernel. We set domain to be $[0, Lx] \times [-Ly/2, Ly/2]$. $h = 1,\ a = h/2,\ dx = h/10,\ Lx = 10h,\ Ly = 200h$. It's sampled at midpoint at each box. The reason why I choose Ly this big is to achieve high accuracy.

2.2.1 Integral along y

First we test its integral along y direction, which should be its $k_y = 0$ mode, i.e. $\widehat{K_{RPY}}(ky = 0)$. It could be computed by $\widehat{K_{RPY} - K_{RB}} + \widehat{K_{RB}}$. For the short range part, we use first entree of its FFT for zero mode.

For long range part, we have its analytic FT. However, since it has special Bessel function and could not be evaluated at $ky = 0$, we evaluate it at $k_y = \epsilon$ and let $\epsilon \to 0^+$ to get the zero mode. For comparison, we computed this integral $K_{RPYx}(x) = \int K_{RPY}(x, y)dy$ analytically, which could be found in this Mathematica sheet. Note that we should shift the grid points by $dx/2$ to avoid singularity. One should notice that Matlab's fft use special order of frequency such that we should take the first entry as $ky = 0$. Results are shown in fig. 3, with $dy = [h/32, h/64, h/128]$. As is shown in fig.3, error is mainly in $[0, 2a = 1]$ since RB kernel is singular at $x = 0$ and we are doing a numerical FT of $K_{RPY} - K_{RB}$.

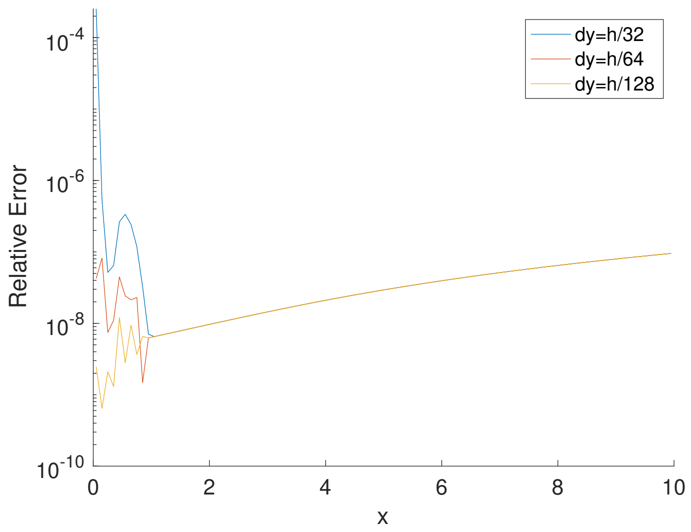


Figure 3: Difference of ky=0 component of Swan-Blake kernel's FT, compared with analytic results. Main error is in $[0, 2a = 1]$ since RB kernel is singular at $x = 0$ and we are doing a numerical FT of $K_{RPY} - K_{RB}$ although K_{RPY} is not singular at $x = 0$

2.2.2 FT along y

The second way to verify our method is to compute FT numerically and do Inverse fourier transform, whose result should match original kernel very well. Algorithm is like following,

1. Do FFT of $K = K_{RPY} - K_{RB}$ along y numerically. It's frequency is ordered in $kk = \frac{2\pi}{L}[0, ..., n/2 - 1, -n/2, -n/2 + 1, ..., -1]$.

2. To match it with FT in continue form, we should do $\widehat{K} = dy \cdot \exp(-i \cdot kk(-Ly/2 + dy/2)) \cdot fft(K)$. Thus, we get $\widehat{K_{RPY}} = \widehat{K} + \widehat{K_{RB}(x, kk)}$.

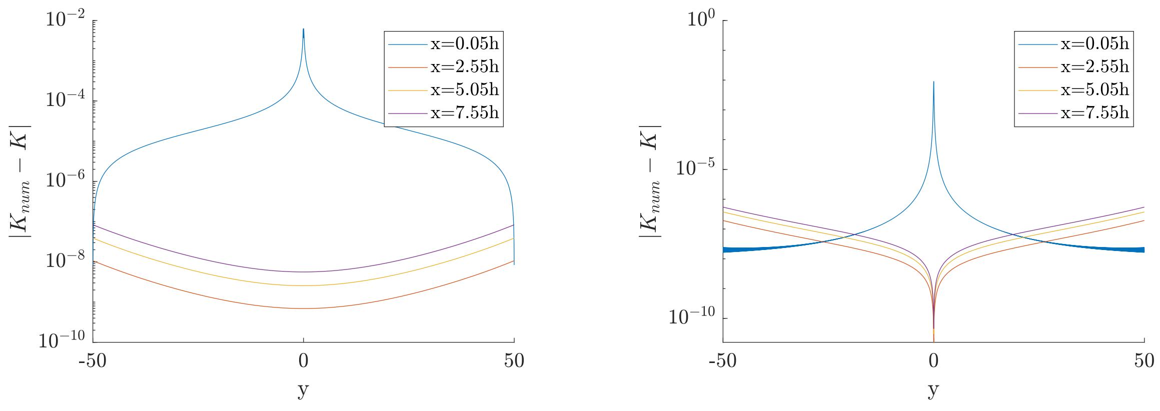
3. Use ifft to convert it into real space. We should do a drift first $1/dy \cdot \exp(i \cdot kk(-Ly/2 + dy/2)) \cdot K_{R\hat{P}Y}$. Then do ifft and compare with original K_{RPY}

Absolute error at different x grid point for both K_x and K_y are shown in fig.3, where we could see clearly that error is mainly near zero. This is due to the fact that K_{RB} is singular at zero. As x goes to 0, it becomes more and more singular. And DFT in y could not capture its high frequency mode since the biggest mode is $\frac{2\pi}{Ly} * N$.

2.2.3 Convergence

For method 2, it only requires FFT computation and evaluation of analytic FT, thus it's promising. Here we test its convergence in spatial step first.

Set $a = 0.01$, $h_g = 1$, $Lx = Ly = 10$. Density function ρ is set to be gaussian $\rho(x, y) = \exp(-3((x - Lx/2)^2 + (y - Ly/2)^2)Lx)$. We increase $Nx = Ny$ and compute the emphirical error in L_2. As shown in fig.5, the convergence is about 3rd order, but error is still kind of big for $Nx = 1024$.



(a) K_x (b) K_y

Figure 4: Absolute Error of Swan-Blake kernel's FT, compared with analytic results.

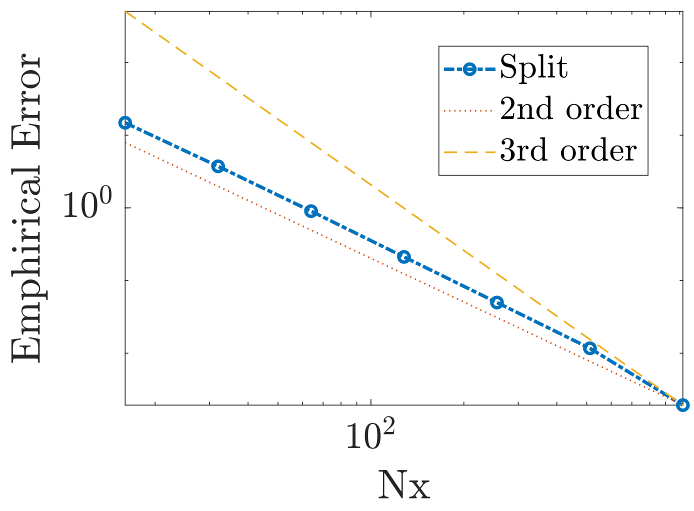


Figure 5: Emphirical convergence in spatial grid of Method 2.

We also expect it to be consistent with Method 1 in the limitation that periodic boxes number goes to infinity. Hence, we fix $Nx = Ny = 128$ and increase M and compute relative error in L_2 between Method 1 and Method 2. As fig.6 shows, it converges quickly in the solution that still has about 0.245 relative error with Method 2, which is not good. Even if I increase $Nx = Ny$ to the range like $Nx = Ny = 2048$, this error still exists and doesn't decrease much.

I think this could be explained by what we saw in section.2.2.2. The idea to use analytic FT to compute DFT would make error since this kernel goes to infinity at zeros. For small x, DFT modes $\frac{2\pi}{Ly}[-Ny/2, \dots, 0, \dots, Ny/2-1]$ could not interpolate high frequency of Fourier transform. If we increase $Nx = Ny$, although we have higher frequency modes, $K_{x=dx/2}(y)$ becomes more singular too, which makes it need even more higher frequency modes. Also, this singularity makes short-ranged part converges slowly too, which could also become source of error.

My major question here is about how to take advantage of this analytic fourier transform, which is wide-ranged in frequency domain, though we only have finite many modes for DFT. Aliasing of high frequency modes would become a problem.

What I'm thinking now is the following two solution.

1. For a given $Nx \times Ny$ mesh grid, we fix Nx so that $K_x(y)$ is fixed and refine y grid. For example, we

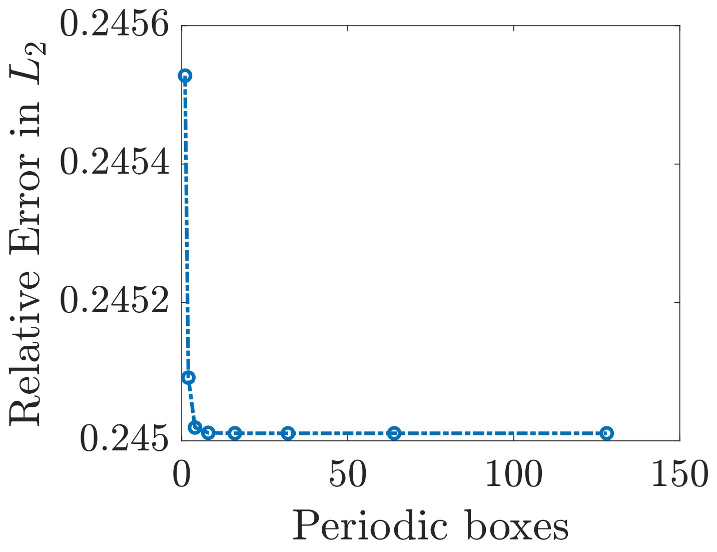


Figure 6: Error in L_2 between Method 1 and Method2.

refine y-grid to be $2^p Ny$ and then computes convolution of this new system by Method 2. Then coarse the grid to get velocity we need in $Nx \times Ny$ grid.

2. Try to find analytic Fourier Transform of an non-singular function that is asymptotic the same as RPY-Swan kernel. Maybe we could somehow regular Rotne-Blake kernel near zero.