

We would like to thank the editor and reviewers for their time and valuable comments concerning our manuscript entitled “A Unified Framework for Oscillatory Integral Transforms: When to use NUFFT or Butterfly Factorization?”. We feel the revised manuscript are vastly improved based on the comments of the reviewers. To ease the reviewers, we mark the changes in the manuscript with blue color text. Below, we list our answer to each of the reviewers’ questions.

**Reviewer 1:** It is in keeping with the high standards of SIAM Journal on Scientific Computing and given the many applications of Fourier integral operators, I have no doubt it will be of interest to SISC’s readership. There are, however, a large number of minor typos, awkward phrase and a small number of places where I found the paper slightly unclear (see below). I recommend the paper be published after minor revisions to correct these issues.

1. The title should probably be changed to either “A unified framework for oscillatory integral transforms: When to use NUFFT or Butterfly Factorization?” or “A unified framework for the oscillatory integral transform: When to use NUFFT or Butterfly Factorization?”

**Response:** *Thank you for pointing out the typo. We have changed the title and added “s” to “transform”.*

2. Together Tables 1, 2 and 3 provide a good overview for the reader, but it took me a while to realize that the scenarios referred to in Table 2 are those listed in Table 3. I don’t believe Table 3 is referred to in the text until page 3 while Table 1 and 2 are referenced on page 2. Moreover, the caption for Table 2 does not reference Table 3. I would suggest mentioning Table 3 in the same paragraph in which you reference Tables 1 and 2 and adding a note to the caption of Table 2 to the effect that the scenarios it refers to are described in Table 3.

**Response:** *Thanks for pointing out the issue of ordering. I have mentioned that the scenarios in Table 2 are listed in Table 3 in the caption of Table 2. This makes it clear for reader to understand when they read Table 2. I have also referred Table 3 right after I introduce Table 1 and 2.*

3. 1st paragraph of the introduction, change “Oscillatory integral transform” to “Oscillatory integral transforms” or “The oscillatory integral transform”.

**Response:** *Thanks for this suggestion. Changed.*

4. First paragraph on page 2 “kernel function motivatives a series new ...” to “kernel function motivates a series of new ...”

**Response:** *Thanks. Changed.*

5. Page 6, second paragraph of Section 2.2 “coumsn” to “columns”.

**Response:** *Thanks. Corrected.*

6. I would suggest explicitly define the TV2 norm in Section 2.2 for the benefit of the reader.

**Response:** *We have defined TV1 and TV2 norms in the footnote on Page 7.*

7. Page 8, last paragraph “functin” to “function” and “disretization” to “discretization”.

**Response:** *Corrected.*

8. Page 17 first paragraph of Section 5 “per requested” to “per request”.

**Response:** *Corrected.*

**Reviewer 2:** There are a few interesting nuggets of ideas in the manuscript that I have not seen before: (a) The details of the approximate recovery of the phase function using  $TV^2$  in the form of a low rank factorization (see Algorithm 3) and (b) The details of the NUFFT and dimensional lifting in Section 3 when  $K = \alpha(x, \eta) \exp^{2\pi i \Phi(x, \eta)}$ . The main complexity gains over existing algorithms are mostly from replacing interpolative decompositions with randomized approximate SVDs when constructing low rank approximations. I also appreciate the manuscripts goal of trying to unify the application of the NUFFT and Butterfly factorization (see Figure 1) in the three situations outlined in Table 3. This is an important unification and, at the very least, has potential benefits in software systems such as the author’s ButterflyLab. I am supportive of the manuscripts goals.

However, my overall feeling after reading the manuscript several times is one of confusion. The manuscript is not carefully written and the precise motivation and details are too often missing. My confusion over the unification is summarized by the comment on p21: once the NUFFT is applicable, it is much more efficient than the BF and then Algorithm 6 where the user gets to select the rank parameter  $r$  to essentially decide if the NUFFT is determined to be applicable or not. The manuscripts answer to the proposed main question When to use NUFFT or Butterfly factorization? is lacking substance and precise details. In addition, the manuscript contains too many careless mistakes that are distracting.

The manuscript does contain several good ideas but the manuscript needs a lot more careful attention.

1. The set of points selected on p5 are very closely related to the so-called mock Chebyshev points on  $[1, N_A]$  (see [1]). In approximation theory, it is known that one must remove all but a set of  $O(\sqrt{N_A})$  points from an equispaced grid to have an associated Lebesgue constant for polynomial interpolation that does not grow exponentially (see Coppersmith and Rivlin inequality in [2]). Therefore, the construction of the low rank approximates from polynomial interpolation described on p5 is theoretically unstable if  $r \gg \sqrt{n}$  [2]. In the numerical examples,  $r$  is always less than  $\sqrt{n}$  but one will need to be careful to stay well within that regime.

**Response:** *Thanks for the nice references. We have cited them and discussed them on Page 5. In all of our applications,  $r$  is an  $O(1)$  parameter independent of the problem size  $N$ . In the application of these Mock Chebyshev points, we have made sure that  $r$  is smaller than  $\sqrt{n}$  in our numerical implementation by controlling the trees of column and row indices in the butterfly factorization, where  $n$  is the smallest dimension of the low-rank matrix, to which we apply the polynomial interpolation. Furthermore, the Mock-Chebyshev points admit root-exponential convergence rate, and it is unknown whether they can achieve exponential convergence rate when we only care about the approximation error on the equispaced grid points in this paper.*

2. I am confused about the motivation of the so-called “middle indices”  $c_A$  and  $c_B$  in (3). I am assuming that the middle indices are something like  $\text{floor}(N_A/2)$  and  $\text{floor}(N_B/2)$ , though this needs to be clarified in the text. Since (3) is a low rank correction of  $\Phi$ , the rank of  $R(A, B)$  is essentially the same as  $\Phi(A, B)$  so I am confused why the low rank approximation of  $K(A, B)$  has been “reduced” to only  $e^{2\pi i R(A, B)}$ . Perhaps, the idea is to remove discontinuities? Either way, the motivation needs to be clarified.

**Response:** *Yes, they are similar to  $\text{floor}(N_A/2)$  and  $\text{floor}(N_B/2)$ . We have clarified them in the text. The low-rank approximation of  $e^{2\pi i R(A, B)}$  immediately gives the low-rank approximation of  $K(A, B)$  by (4). This is can be immediately seen if you visualize the equation in (4) as  $K = D_1 B D_2$ , where  $B$  represents  $e^{2\pi i R(A, B)}$ . If we have a low-rank approximation of  $B = LR$ , then  $(D_1 L)(R D_2)$  is the low-rank approximation of  $K$ .*

3. Phase retrieval is a well-established field of research, where TV-regularization is regularly employed. While I am not an expert in this area, I would appreciate if Section 2.2 connected better with this research topic.

**Response:** *You are right that the phase function recovery problem is similar to phase retrieval. However, these two problems have different setting and requirements. We would like to recover low-rank approximation of the smooth phase function in  $O(N)$  operations, which means that we can only touch  $O(N)$  entries of the phase function throughout the computation. In phase retrieval,  $O(N^2)$  observations of  $|e^{2\pi i\Phi}|$  are required and the state-of-the-art algorithm via PhaseLift with low-rank approximation requires  $O(N^2)$  operations per iterations to solve the phase retrieval problem. If we consider the number of iterations to guarantee the solution satisfies the constraints in Equation (12) within machine accuracy, existing phase retrieval algorithms have a very large prefactor. In phase retrieval, although the 2D target function has some smoothness in most area, it may still have some unknown and irregular discontinuity in some area, after discretization the resulting matrix might not be low-rank. What's more, many algorithms for phase retrieval use the hard constrain in Equation (12) as a soft constrain, which is not desired in our problem. We have added a short discussion in Section 2.2 to clarify the difference.*

4. Algorithm 4 is described without justification or motivation. I can see that the author is employing a second-order finite difference stencil to solve for entries of  $v$  (under constraint) as well as to detect discontinuities. However, step 16 seems to be trying to minimize an approximation to the third derivative of  $v$  (the difference of two 2nd order stencils), which I am not sure about. Also, as written the while loop never completes as the variable  $c$  never gets updated. This is a heuristic algorithm that needs motivation and discussion; particularly, exactly how is it approximately solving (12). To understand Algorithm 3, the reader needs to know Algorithm 4 so the author may wish to interchange the order of these two algorithms. Finally, the behavior of Algorithm 4 critically depends on  $\tau$ , the discontinuity parameter, and this is not discussed.

**Response:** *Thanks for the comments. We have rewritten Section 2.2 according to your suggestions. 1) We have changed the order of Algorithm 3 and 4. A vector recovery problem is described to motivate Algorithm 4 (now it is Algorithm 3 in the revision). 2) You are right that it is more appropriate to us TV3 norm minimization to summarize Algorithm 3 and 4, because we try to minimize the difference of third order derivatives. 3) We have updated the variable “ $c$ ” in Line 20 of Algorithm 3. 4) It might not be important to understand how the proposed algorithms solves (12) since we don't require an exact solution. Any solution that are piecewise smooth such that the low-rank recovery algorithm works is good enough. 5)  $\tau$  is not an important parameter, since we don't require exact identification of discontinuous points. We have tested different  $\tau$  in a wide range, as long as  $\tau$  is not close to 0, which will result in too many fake discontinuous points that fail the scaling of the proposed algorithm, the matrix recovery algorithm will work. We have commented this in the end of Page 8.*

5. Algorithm 6 determines when a NUFFT is applicable or not. Is an oscillatory integral transform either NUFFT-friendly or IBF-MAT-friendly problem? I imagine that there is a \*huge\* middle ground here where neither a NUFFT nor an IBF-MAT is exploiting the right structure to be computationally efficient. Also, it seems that the user can select a huge range of rank parameters  $r$  to make Algorithm 6 decide that a NUFFT is applicable or not. I am therefore confused about what exactly has been unified in the manuscript.

**Response:** *Yes, all oscillatory integral transform with a kernel function  $\alpha(x, y)e^{2\pi i\Phi(x, y)}$  is*

either NUFFT-friendly or IBF-MAT-friendly, as long as  $\alpha$  and  $\Phi$  are piecewise smooth functions, which is the claim of our paper. Let us prove this point below assuming no discontinuous points and  $\alpha(x, y) = 1$ .

In a simple case when the kernel is from the discrete Fourier transform. For  $x$  and  $y = 0, \frac{1}{N}, \dots, \frac{N-1}{N}$ , Taylor expansion of  $Nx \cdot y$  at the position  $(x, y) = (c_A, c_B)$  results in the following factorization:

$$e^{2\pi i N x \cdot y} = e^{2\pi i N x \cdot c_B} e^{2\pi i N (x - c_A) \cdot (y - c_B)} e^{2\pi i N c_A \cdot y} e^{-2\pi i N c_A \cdot c_B}$$

Hence,

$$e^{2\pi i N (x - c_A) \cdot (y - c_B)}$$

is numerically low-rank for  $(x, y) \in A \times B$ , if

$$w_A * w_B \leq \frac{1}{N},$$

where  $A$  is a domain of  $x$  center at  $c_A$  with width  $w_A$ ,  $B$  is a domain of  $y$  center at  $c_B$  with width  $w_B$ . This shows that the DFT matrix is complementary low-rank. We emphasize here that a larger  $N$  results in smaller low-rank matrices.

In a general case, Taylor expansion of the kernel  $e^{2\pi i \Phi(x, y)}$  at the position  $(x, y) = (c_A, c_B)$  gives

$$e^{2\pi i \Phi(x, y)} \approx e^{2\pi i \alpha(x)} e^{2\pi i (x - c_A)(y - c_B) \partial_{xy} \Phi(c_A, c_B)} e^{2\pi i \beta(y)}.$$

In most high-frequency wave computation, we have a phase function satisfying  $\partial_x^n \partial_y^m \Phi(x, y) = O(k^{m+n})$ , where  $k$  is the frequency of the problem. Hence,

$$e^{2\pi i (x - c_A)(y - c_B) \partial_{xy} \Phi(c_A, c_B)}$$

is numerically low-rank for  $(x, y) \in A \times B$ , if

$$w_A * w_B \leq \frac{1}{k^2}.$$

This shows that the kernel matrix is complementary low-rank. We emphasize here that a larger  $k$  results in smaller low-rank matrices.

In sum, IBF can always give a nearly optimal implementation of the oscillatory integral transform. The derivation above is standard and well-known. Hence, we don't include it in the paper.

In special cases when the phase function has a very small rank then the NUFFT approach is obviously suitable for the oscillatory integral transform.

Hence, there might not be a regime of oscillatory integral transforms considered in this paper such that the NUFFT or IBF-MAT cannot achieve nearly optimal scaling. The third famous approach for oscillatory integral transform is the wide-band FMM, which might be more complicated than NUFFT and IBF and usually is kernel-dependent, i.e., once the kernel changes, we need to make a new code. The NUFFT and IBF approach in this paper doesn't rely on the explicit formulas of the amplitude and phase functions, instead they work on a matrix form. Hence, they are kernel independent.

This unified framework is to show that we can always use NUFFT and IBF to perform the oscillatory integral transform in ALL scenarios in Table 3. The unified framework is a survey

containing possibly most application scenarios and demand for oscillatory transform, and a discussion on the key factors for deciding which method to use. The NUFFT approach is a general method that includes many specialized method based on FFT. Hence, the proposed framework has included many existing algorithms for oscillatory integral transform. We have listed all factors that help to decide whether NUFFT or IBF is better.

You are right that there is a rank parameter to decide whether NUFFT is applicable. But please note that Algorithm 6 is only to decide whether NUFFT is applicable or not; it is not for deciding whether NUFFT is better than BF. To answer whether NUFFT or BF should be applied, we use a numerical example of FIO in solving wave equations. As shown in Table 7, both the factorization and application time of BF are more expensive than NUFFT when we fix the approximation accuracy. Hence, we claim that once NUFFT is applicable, we should use NUFFT for oscillatory integral transform, no matter how many vectors in the matvec.

6. The proposed framework needs to be fully implemented into ButterflyLab before publication as there are still plenty of user-defined parameters ( $\tau$ ,  $r$ ,  $q$ , etc.) floating around, which probably makes the comparison as much about parameter-tuning as algorithms. There are also two implicit assumptions that the author is making that could be clarified: (i) Compute  $g = K*f$  where there is only one righthand sides of interest (so precomputation cannot be ignored) and (ii) The algorithms run in serial, as opposed to parallel.

**Response:** We will make the unified framework available in ButterflyLab online after the acceptance of the paper for the purpose of reproducible research. We are sorry that since it is in the reviewing process and it is not convenient for us to publish this code. We hope the reviewer could understand this.

Approximate fast algorithms naturally come with parameters that cannot be fixed and have to be decided by users depending on their demand, e.g., speed vs accuracy. We have shown that  $\tau$  is not an important parameter and our algorithm is not sensitive to it.  $r$  and  $q$  are standard parameters in randomized low-rank approximation. Hence, we believe that leaving  $r$  and  $q$  as user defined parameters is reasonable. In spite of the comparison, the proposed algorithms themselves are novel and they are the main focus of our paper.

The comparison is only for the purpose of showing a rough idea about how to choose these algorithms. We don't aim at a parameter-tuning paper and hence it might not be necessary to compare these two algorithms with too many different choices of parameters. For simplicity, we fix the accuracy for matvec and tune parameters to make sure that NUFFT and BF achieve the same accuracy. As shown in Table 7, both the factorization and application time of BF are more expensive than NUFFT when we fix the approximation accuracy. Hence, we claim that once NUFFT is applicable, we should use NUFFT for oscillatory integral transform, no matter how many vectors in the matvec.

We have emphasized that our discussion on how to choose these algorithms is in a serial computational environment on Page 2.

## 7. MINOR CLARIFICATIONS REQUIRED

- \* Is the  $O(r)$  in Alg 1 actually  $rq$ ?
- \* The  $U$  and  $V$  in Alg 2 are used for both the  $\Phi$  and  $\exp^{2\pi i\Phi}$ .
- \* What values are you thinking about for  $\tau$  in Alg 4?
- \* Step 5 of Alg 6 selects  $n$  so that one expects that it is a lower bound on the numerical rank of  $(U_2 V_2^*)$ . \*  $\exp^{2\pi i(U_1 V_1^* - P Q^*)}$ . Do you need to assume that  $n \leq r_\epsilon$ ?

**Response:** 1) No,  $O(r)$  is not exactly  $rq$  because in Line 6 in ALg 1 we mentioned that Line 4 and 5 may be repeated for a few times. For example, the union of  $S_{\text{row}}$  and  $\Pi_{\text{row}}$  may generate more than  $rq$  selected rows in the QR factorization.

2) You're right. Corrected.

3) Answered in the previous answer.

4) Yes, we expect that  $n \leq r_\epsilon$ . This is not an assumption, but a natural requirement due to the current development of NUFFT, since there is no NUFFT with higher than 3D available. Higher dimensional NUFFT could be an interesting future work. Hence,  $n \leq r$  and  $r \leq 3$ .  $r_\epsilon$  can be as large as  $O(100)$ . There might be no computational meaning for clarifying  $n \leq r_\epsilon$  since  $r$  and  $r_\epsilon$  are two independent parameters. If a very high dimensional NUFFT is available,  $n > r_\epsilon$  might not be a bad idea. Hence, we didn't mention the relation of  $n$  and  $r_\epsilon$ . We have commented the size of  $r$  right after Algorithm 6.

8. There are a huge number of typos in the manuscript. It makes the manuscript hard to read.

**Response:** Corrected

[1] J.P. Boyd, F. Xu, Divergence (Runge Phenomenon) for least-squares polynomial approximation on an equispaced grid and Mock Chebyshev subset interpolation. Appl. Math. Comput. 210 (2009) 158168.

[2] R. B. Platte, L. N. Trefethen, and A. B. J. Kuijlaars, Impossibility of fast stable approximation of analytic functions from equispaced samples, SIAM Review, 53.2 (2011), pp. 308-318.