Matrix probing, skeleton decompositions, and sparse Fourier transform

by

Jiawei Chiu

Submitted to the Department of Mathematics
in partial fulfillment of the requirements for the degree of

Doctorate of Philosophy in Mathematics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2013

Author .....
Department of Mathematics
May 15, 2013

Certified by .....
Laurent Demanet
Assistant Professor
Thesis Supervisor

Accepted by .....
Michel Goemans
Chairman, Department Committee on Graduate Theses

Matrix probing, skeleton decompositions, and sparse Fourier transform

by

Jiawei Chiu

Submitted to the Department of Mathematics
on May 15, 2013, in partial fulfillment of the
requirements for the degree of
Doctorate of Philosophy in Mathematics

Abstract

In this thesis, we present three different randomized algorithms that help to solve matrices, compute low rank approximations and perform the Fast Fourier Transform.

Matrix probing and its conditioning

When a matrix A with n columns is known to be well approximated by a linear combination of basis matrices $B_1, \ldots, B_p$, we can apply A to a random vector and solve a linear system to recover this linear combination. The same technique can be used to obtain an approximation to A^{-1}. A basic question is whether this linear system is well-conditioned. This is important for two reasons: a well-conditioned system means (1) we can invert it and (2) the error in the reconstruction can be controlled. In this paper, we show that if the Gram matrix of the B_j's is sufficiently well-conditioned and each B_j has a high numerical rank, then $n \propto p \log^2 n$ will ensure that the linear system is well-conditioned with high probability. Our main application is probing linear operators with smooth pseudodifferential symbols such as the wave equation Hessian in seismic imaging. We also demonstrate numerically that matrix probing can produce good preconditioners for inverting elliptic operators in variable media.

Skeleton decompositions in sublinear time

A skeleton decomposition of a matrix A is any factorization of the form $A_{:C} Z A_{R:}$ where $A_{:C}$ comprises columns of A, and $A_{R:}$ comprises rows of A. In this paper, we investigate the conditions under which random sampling of C and R results in accurate skeleton decompositions. When the singular vectors (or more generally the generating vectors) are *incoherent*, we show that a simple algorithm returns an accurate skeleton in sublinear $O(\ell^3)$ time from $\ell \simeq k \log n$ rows and columns drawn uniformly at random, with an approximation error of the form $O(\frac{n}{\ell}\sigma_k)$ where σ_k is the k-th singular value of A. We discuss the crucial role that regularization plays in

forming the middle matrix U as a pseudo-inverse of the restriction A_{RC} of A to rows in R and columns in C. The proof methods enable the analysis of two alternative sublinear-time algorithms, based on the rank-revealing QR decomposition, which allow us to tighten the number of rows and/or columns sampled to k with an error bound proportional to σ_k.

Sparse Fourier transform using the matrix pencil method

One of the major applications of the FFT is to compress frequency-sparse signals. Yet, FFT algorithms do not leverage on this sparsity. Say we want to perform the Fourier transform on $x \in \mathbb{C}^N$ to obtain some $\hat{x}$ which is known to be S-sparse with some additive noise. Even when S is small, FFT still takes $O(N \log N)$ time. In contrast, SFT (sparse Fourier transform) algorithms aim to run in $\widetilde{O}(S)$ time ignoring log factors. Unfortunately, SFT algorithms are not widely used because they are faster than the FFT only when $S \ll N$. We hope to address this deficiency. In this work, we present the fastest known robust $\widetilde{O}(S)$-time algorithm which can run up to 20 times faster than the current state-of-the-art algorithm AAFFT. The major new ingredient is a mode collision detector using the matrix pencil method. This enables us to do away with a time-consuming coefficient estimation loop, use a cheaper filter and take fewer samples of x. We also speed up a crucial basic operation of many SFT algorithms by halving the number of trigonometric computations. Our theory is however not complete. First, we prove that the collision detector works for a few classes of random signals. Second, we idealize the behavior of the collision detector and show that with good probability, our algorithm runs in $O(\frac{S}{\varepsilon} \log^2 \frac{N}{S} \log N)$ time and outputs a $O(S)$-sparse $\hat{x}'$ such that $\|\hat{x}' - \hat{x}\|^2 \leq (1+\varepsilon) \|\hat{x}_* - \hat{x}\|^2$ where $\hat{x}_*$ is the best exact S-sparse approximation of $\hat{x}$.

Thesis Supervisor: Laurent Demanet
Title: Assistant Professor

Acknowledgments

I am deeply indebted to my advisor, Prof. Laurent Demanet. He has provided me invaluable advice, academic or non-academic, and has given me more support than I can ever ask for. I am also grateful to Prof. Alan Edelman, Prof. Piotr Indyk and Dr. Jeremy Orloff for their trust in me, and to Prof. Scott Sheffield for agreeing to be on my thesis committee. I would like to thank everyone in the Imaging and Computing Group, particularly Russell, Rosalie, Leo, for attending the group meeting even when I am presenting. I would also like to thank Jeffrey Pang, Yifeng Wei, Liang Jie Wong for their wonderful company during my stay at MIT. I am also thankful to A*STAR for their financial support over the years. Finally, I would like to dedicate this piece of work to the most important people of my life: my parents, my sister and my wife.

Contents

1 Overview **11**

1.1 Matrix probing and its conditioning 11

1.2 Sublinear randomized algorithms for skeleton decompositions 12

1.3 Sparse Fourier transform using the matrix pencil method 14

2 Matrix probing and its conditioning **19**

2.1 Introduction 19

2.1.1 Forward matrix probing 20

2.1.2 Conditioning of L 21

2.1.3 Multiple probes 24

2.1.4 When to probe 25

2.1.5 Backward matrix probing 26

2.2 Proofs 27

2.2.1 Proof of Theorem 2.1.3 27

2.2.2 Sketch of the proof for Theorem 2.1.4 30

2.2.3 Proof of Proposition 2.1.5 30

2.2.4 Proof of Corollary 2.1.6 31

2.3 Probing operators with smooth symbols 32

2.3.1 Basics and assumptions 32

2.3.2 User friendly representations of symbols 33

2.3.3 Symbol expansions 34

2.3.4 Chebyshev expansion of symbols 36

2.3.5 Order of an operator 39

2.4 Numerical examples 40
2.4.1 1D statistical study 40
2.4.2 Elliptic equation in 1D 43
2.4.3 Elliptic in 2D 45
2.4.4 Foveation 48
2.4.5 Inverting the wave equation Hessian 48
2.5 Conclusion and future work 50

3 Sublinear randomized algorithms for skeleton decompositions 53
3.1 Introduction 53
3.1.1 Skeleton decompositions 53
3.1.2 Overview 54
3.1.3 Related work 56
3.1.4 Notations 58
3.1.5 Main result 59
3.1.6 More on incoherence 61
3.2 Error estimates for $\widetilde{O}(k^3)$ algorithm 62
3.2.1 Notation 62
3.2.2 Two principles 63
3.2.3 Proof of Theorem 3.1.2 66
3.3 Alternative sublinear-time algorithms 72
3.3.1 Second algorithm 72
3.3.2 Third algorithm 75
3.3.3 Comparison of three algorithms 78
3.4 Examples 79
3.4.1 First toy example: convolution 79
3.4.2 Second toy example 80
3.4.3 Smooth kernel 82
3.4.4 Fourier integral operators 85

4 Sparse Fourier transform using the matrix pencil method **87**

4.1 Introduction 87

4.1.1 Review of sFFT3.0 89

4.1.2 Two limitations of sFFT3.0 94

4.1.3 MPFFT and main results 98

4.2 Matrix pencil method 102

4.2.1 Introduction 102

4.2.2 Identifying one mode and first order perturbations 105

4.2.3 Multiscale matrix pencil method 107

4.3 Collision detector 110

4.3.1 Total energy comparable to energy of dominant mode 113

4.3.2 Subdominant energy comparable to energy of second mode . . 116

4.3.3 Subdominant modes do not cancel one another 117

4.3.4 A few heavy modes with little noise 118

4.3.5 Two heavy modes 120

4.4 Binning 122

4.4.1 How binning works 122

4.4.2 Faster binning 127

4.4.3 Binning-related estimates 130

4.5 Analysis of MPFFT 133

4.5.1 Chance that a mode is identified and estimated well 133

4.5.2 Overall analysis of MPFFT 138

4.6 Implementation and numerical results 144

4.6.1 Numerical tests 148

4.6.2 Collision detection 153

A **157**

A.1 Khintchine inequalities 157

A.2 Other probabilistic inequalities 160

A.3 Linear algebra 160

Chapter 1

Overview

This dissertation consists of three main parts. In this chapter, we briefly describe what each of the three parts is about. Note that in each part, a different notation may be set up.

1.1 Matrix probing and its conditioning

Matrix probing is a simple idea that can be used for preconditioning and system identification. Let A be a large $n \times n$ matrix with not much information. Let p be a small positive integer. Suppose we know that $A \simeq \sum_{j=1}^{p} c_j B_j$ for some predefined B_j's. Let $c = (c_1, \ldots, c_p)^T$ be the vector of coefficients. The idea of matrix probing is to recover c by *applying A to random vectors.* Let u be a random Gaussian vector. Compute $v = Au$. Observe that

$$Lc \simeq v \text{ where } L = (B_1 u, \ldots, B_p u).$$

The linear system is then solved to find c. Our work addresses the following question: to recover c accurately, what assumptions do we need and how big does n have to be relative to p?

Let $\langle \cdot, \cdot \rangle$ be the Frobenius inner product. Our theory says that if each B_j is well-conditioned and act in a different way in the sense that $\langle B_i, B_j \rangle \simeq \delta_{ij}$, then

$n \gtrsim p$ ensures that with high probability, L is well-conditioned and we can recover c accurately and stably. For more details, see Theorem 2.1.3.

Matrix probing can be used to approximately *invert* a matrix. The idea is to apply A^{-1} to $v = Au$ where u is a random Gaussian vector. The steps are:

1) Generate one random Gaussian vector u. Compute $v = Au$.
2) Compute $B_1 v, \ldots, B_p v$ and form $L = (B_1 v, \ldots, B_p v)$.
3) Solve $Lc = u$ to estimate the coefficients c_i's.

In our applications, we considered structured matrices A for which a good choice is to take the B_j's to be elementary pseudodifferential symbols. More on pseudodifferential symbols can be found in Section 2.3. If matrix-vector multiplications take $O(n)$ time for our structured matrix A, then matrix probing takes only $O(np^2)$ time. This is comparable with multigrid methods which we find more restrictive. We find it intriguing that A^{-1} can be well-approximated merely by applying A to one random vector.

A major application is inverting the wave equation Hessian in seismic imaging [23]. Consider the least squares problem $\min_x \|b - Ax\|_2^2$ where b is data, A is the linearized forward operator and x is the model. A popular method to solve this is the Newton method. It converges in very few iterations, but requires us to compute $(A^*A)^{-1}$, the inverse of the Hessian. Our numerical experiments indicate that matrix probing can produce a high quality approximation of $(A^*A)^{-1}$.

1.2 Sublinear randomized algorithms for skeleton decompositions

We are interested in fast algorithms that produce low rank decompositions given partial information of a matrix. Unlike matrix completion [13], we do not perform any optimization. The algorithms we consider use only numerical linear algebra and run in sublinear time, i.e., $o(n^2)$ time if the matrix is of size n. Our work has many

applications. For example, it can be used to compute low rank approximations of Green's functions of many PDEs used in Boundary Integral Equation methods.

Let A be a $n \times n$ matrix. We seek to compute *matrix skeletons* [35] that approximate A. Matrix skeletons take the form of CUR where C is a column subset of A and R is a row subset of A. This representation is especially space-saving if we have a closed form or compressed representation of A. Suppose A is approximately rank k. Let $\ell \gtrsim k \log n$. We considered the following $O(\ell^3)$ time algorithm which returns a rank ℓ matrix skeleton. The pseudocode is on the left, while the Matlab code is on the right.

1) Uniformly sample ℓ rows of A to form R.	`r=randperm(n,ℓ); % R=A(r,:);`
2) Uniformly sample ℓ columns of A to form C.	`c=randperm(n,ℓ); % C=A(:,c);`
3) Let $Z \in \mathbb{C}^{\ell \times \ell}$ be the intersection of C, R.	`Z=A(r,c);`
4) Let Z' be Z with singular values less than δ removed. Let $U = Z'^{+}$, i.e., U is the *thresholded* pseudoinverse of Z.	`U=pinv(Z,delta);`
5) Return implicitly the matrix skeleton CUR.	(Return `r,c,U`)

The above algorithm works well in practice and is very simple compared to methods such as Adaptive Cross Approximation [5]. We like to understand when the above algorithm works well.

Suppose $A \simeq XBY^*$ where X, Y are $n \times k$ matrices with orthonormal columns and B is not necessarily diagonal. Like in matrix completion, assume that X, Y are *incoherent*. This means all the entries of X, Y are of magnitude $O(n^{-1/2})$. It is instructive to consider $Y = \binom{I_{k \times k}}{0}$, which is not incoherent. In this case, $C \simeq 0$ with high probability and the above algorithm fails. For uniform sampling to work, the incoherence assumption on X, Y seems to be necessary.

Here is what our main result or Theorem 3.1.2 says. Let $\varepsilon = \|A \simeq XBY^*\|$ where $\|\cdot\|$ is the operator norm. Suppose X, Y are incoherent, $\ell \gtrsim k \log n$ and the above

algorithm is run with $\delta \simeq \varepsilon$, then with high probability the algorithm will return a skeleton decomposition CUR satisfying

$$\|A - CUR\| \lesssim n\varepsilon/\ell.$$

Our numerical experiments in Section 3.4 show that the operator norm error can blow up as $\delta \to 0$, suggesting that the thresholding in Step 4 is indeed necessary. We also considered the following $\widetilde{O}(nk^2)$[1] time algorithm. Its main advantage is that only k columns are selected instead of $\ell \gtrsim k \log n$.

1) Uniformly sample p rows of A to form R.	`r=randperm(n,p); R=A(r,:);`
2) Run RRQR [36] on R to select k columns.	`c=rrqr(R,k); C=A(:,c);`
3) Let $Z \in \mathbb{C}^{p \times k}$ be the intersection of C, R.	`Z=A(r,c);`
4) Let $U = Z^+$. No thresholding is needed.	`U=pinv(Z);`
5) Return the matrix skeleton CUR.	

In Step 2, we use the Rank Reveal QR or Interpolative Decomposition [17, 36] to deterministically select k columns of R. Using the same proof framework, we show that under similar assumptions on A as before, i.e., $A \simeq XBY^*$, $\varepsilon = \|A - XBY^*\|$, $\ell \gtrsim k \log n$, X is incoherent, then the above algorithm will with high probability return a skeleton decomposition CUR satisfying $\|A - CUR\| \lesssim n\varepsilon$. For more details, see Theorem 3.3.2.

1.3 Sparse Fourier transform using the matrix pencil method

Frequency-sparse signals are abundant in our world. A natural question to ask is: if the signal is frequency-sparse, can we find its Fourier coefficients faster than the FFT? To be concrete, if the signal is of size N and has S large Fourier coefficients, can we find these coefficients in $\widetilde{O}(S)$ time instead of $O(N \log N)$ time?

[1] $\widetilde{O}(\cdot)$ is the $O(\cdot)$ notation with log factors dropped.

In 2002, Gilbert, Indyk et al. [33, 32] provided an algorithm which runs in $O(S\log^c N)$ time for some $c > 2$. Although its running time is near-optimal in theory, it is not very useful in practice. The reason is that we need S to be much smaller than N for it to be faster than FFT. Recently, Hassanieh, Indyk et al. [41] proposed a new algorithm sFFT1.0 which is significantly faster in practice. For $N = 2^{22}$, it is able to beat FFT for $S \lesssim 1000$[2]. The caveat is that its running time is $\widetilde{O}(\sqrt{NS})$ which does not scale well with N.

Our contribution is the design of the *fastest known robust* $\widetilde{O}(S)$-time sparse Fourier transform (SFT) algorithm. As of now, it is at least 5 times faster than AAFFT even when it is using conservative parameters that favor robustness to noise. For more, see Figure 4-1 and the numerical experiments in Section 4.6.1. Our main idea is to combine the matrix pencil method [45], a well-established spectral estimation tool, with sFFT3.0 [40], a fast but nonrobust SFT algorithm. By analyzing the spectral properties of matrices formed from translates of a signal, we are able to *detect "mode collision"* and speed up the estimation of coefficients. To understand how this works, we need a review of sFFT3.0. Here are its main steps.

1. Let $x \in \mathbb{C}^N$ be our signal and $\hat{x}$ be its Fourier transform. Bin the modes by convolving $\hat{x}$ with a smoothed boxcar filter of width $\sim \frac{1}{B}$ in the frequency space $[0, 1)$ where B is the number of bins. Obtain a $Y_0 \in \mathbb{C}^B$ such that

$$Y_0^b \simeq \sum_{k \text{ in bin } b} \hat{x}_k \text{ for any } b = 0, \ldots, B-1.$$

   By "k in bin b", we mean that $\lfloor kB/N \rceil = b$ or equivalently, $\frac{b}{B} \leq \frac{k}{N} < \frac{b+1}{B}$.

2. Let x^τ be x translated. Apply the previous step to x^τ and obtain $Y_\tau \in \mathbb{C}^B$. As $\hat{x}^\tau$ is $\hat{x}$ modulated,

$$Y_\tau^b \simeq \sum_{k \text{ in bin } b} \hat{x}_k e^{2\pi i k\tau/N} \text{ for any } b = 0, \ldots, B-1.$$

[2]We compile sFFT1.0 and sFFT2.0 using the same compiler flags as FFTW which include -O3 and `-mtune=native`. The latter turns on hardware optimization which is likely to be unfavorable to SFT algorithms. The hardware optimization however does not seem to affect the comparison with FFTW on the `FFTW_MEASURE` option.

3. Fix a bin b. Suppose there is only one mode k_0 in this bin. Then $Y_\tau^b \simeq \hat{x}_{k_0} e^{2\pi i k_0 \tau/N}$. We call k_0 an isolated mode. Estimate k_0 as $\frac{N}{2\pi} \arg \frac{Y_1^b}{Y_0^b}$ and $\hat{x}_{k_0}$ as Y_0^b.

4. By making $B \gtrsim S$, we expect a constant proportion of the modes to be isolated and found. Let x' be x with its modes randomly and implicitly shuffled. Repeat the above steps on x'.

The number of modes left to be found decay geometrically and only $O(\log S)$ repetitions are needed.

When there is noise, Step 3 does not work and we have to identify k_0 bit by bit. The idea of finding the index of an isolated mode in a multiscale fashion is not new. What is new here is a mode collision mechanism based on the matrix pencil method [45]. Fix a bin b. Define $X \in \mathbb{C}^N$ by $X_\tau = Y_\tau^b = \sum_{k \text{ in bin } b} \hat{x}_k e^{2\pi i k\tau/N}$. Apply the matrix pencil method to X to try to find the modes in bin b. This involves forming a $J \times J$ Toeplitz matrix (cf. (4.7))

$$A = \frac{1}{J} \begin{pmatrix} X_0 & X_{-1} & \dots & X_{-J+1} \\ X_1 & X_0 & \dots & X_{-J+2} \\ \vdots & \vdots & \vdots & \vdots \\ X_{J-1} & X_{J-2} & \dots & X_0 \end{pmatrix}.$$

Let $\mu^2 = \|A\|_F^2 - \|A\|^2 = \sum_{j=2}^J \sigma_j^2(A)$ where $\sigma_j(A)$ is the j-th largest singular value of A and $\|\cdot\|_F$ is the Frobenius norm. The idea is that if there are more than one mode in this bin, then it is likely that μ is much bigger than what we expect from noise. In other words, if μ is small, then we are confident that we have an isolated mode and will estimate the coefficient $\hat{x}_{k_0}$ using $(Y_j)_{|j| \le J-1}$. Existing iterative SFT algorithms have to run a separate loop to estimate these coefficients. This loop requires more bins than is optimal, a more expensive filter for binning, additional random shuffles, which amount to more signal samples and a slower algorithm.

In Section 4.3, we assume that the modes landing in a bin are fully randomly shuffled, and show that for some common cases, μ is unlikely to be small when there

is more than one heavy mode in the bin. One example is that the total energy of the heavy modes in the bin is sufficiently large relative to the heaviest mode in the bin. In Theorem 4.1.3, we idealize the effectiveness of the collision detector and show that with good probability, our algorithm will terminate in $O(\frac{S}{\varepsilon} \log N \log^2 \frac{N}{S})$ time with a $1+\varepsilon$ relative ℓ^2-error in the estimation of $\hat{x}_*$ where $\hat{x}_*$ is the best exact S-sparse approximation of $\hat{x}$. We argue informally at the beginning of Section 4.5 that our algorithm must run in $\Omega(\frac{S}{\varepsilon} \log N \log^2 \frac{N}{S})$ time.

Chapter 2

Matrix probing and its conditioning

2.1 Introduction

The earliest randomized algorithms include Monte Carlo integration and Monte Carlo Markov chains [2]. These are standard techniques in numerical computing with widespread applications from physics, econometrics to machine learning. However, they are often seen as the methods of last resort, because they are easy to implement but produce solutions of uncertain accuracy.

In the last few decades, a new breed of randomized algorithms has been developed by the computer science community. These algorithms remain easy to implement, and in addition, have failure probabilities that are provably negligible. In other words, we have rigorous theory to ensure that these algorithms perform consistently well. Moreover, their time complexity can be as good as the most sophisticated deterministic algorithms, e.g., Karp-Rabin's pattern matching algorithm [49] and Karger's min-cut algorithm [48].

In recent years, equally attractive randomized algorithms are being developed in the numerical community. For example, in compressed sensing [15], we can recover sparse vectors with random measurement matrices and ℓ^1 minimization. Another interesting example is that we can build a good low rank approximation of a matrix by *applying it to random vectors* [39].

Our work carries a similar flavor: often, the matrix A can be approximated as

a linear combination of a small number of matrices and the idea is to obtain these coefficients by applying A to a random vector or just a few of them. We call this "forward matrix probing." What is even more interesting is that we can also probe for A^{-1} by applying A to a random vector. We call this "backward matrix probing" for a reason that will be clear in Section 2.1.5.

Due to approximation errors, the output of "backward probing" denoted as C, is only an approximate inverse. Nevertheless, as we will see in Section 2.4, C serves very well as a preconditioner for inverting A, and we believe that its performance could match that of multigrid methods for elliptic operators in smooth media.

We like to add that the idea of "matrix probing" is not new. For example, Chan [19, 18] et. al. use the technique to approximate A with a sparse matrix. Another example is the work by Pfander et. al. [60] where the same idea is used in a way typical in compressed sensing. In the next section, we will see that their set-up is fundamentally different from ours.

2.1.1 Forward matrix probing

Let $\mathcal{B} = \{B_1, \ldots, B_p\}$ where each $B_j \in \mathbb{C}^{m\times n}$ is called a basis matrix. Note that $\mathcal{B}$ is specified in advance. Let u be a Gaussian or a Rademacher sequence, that is each component of u is independent and is either a standard normal variable or ± 1 with equal probability.

Define the matrix $L \in \mathbb{C}^{m\times p}$ such that its j-th column is $B_j u$. Let $A \in \mathbb{C}^{m\times n}$ be the matrix we want to probe and suppose A lies in the span of $\mathcal{B}$. Say

$$A = \sum_{i=1}^{p} c_i B_i \text{ for some } c_1, \ldots, c_p \in \mathbb{C}.$$

Observe that $Au = \sum_{i=1}^{p} c_i(B_i u) = Lc$. Given the vector Au, we can obtain the coefficient vector $c = (c_1, \ldots, c_p)^T$ by solving the linear system

$$Lc = Au. \tag{2.1}$$

In practice, A is not exactly in the span of a small $\mathcal{B}$ and (2.1) has to be solved in a least squares sense, that is $c = L^+(Au)$ where L^+ is the pseudoinverse of L.

We will assume that $p \leq n$. Otherwise there are more unknowns than equations and there is no unique solution if there is any. This differs from the set-up in [60] where $n \gg p$ but A is assumed to be a sparse linear combination of $B_1, \ldots, B_p$.

2.1.2 Conditioning of L

Whether (2.1) can be solved accurately depends on $\text{cond}(L)$, the condition number of L. This is the ratio between the largest and the smallest singular values of L and can be understood as how different L can stretch or shrink a vector.

Intuitively, whether $\text{cond}(L)$ is small depends on the following two properties of $\mathcal{B}$.

1. The B_i's "act differently" in the sense that $\langle B_j, B_k \rangle \simeq \delta_{jk}$ for any $1 \leq j, k \leq p$.[1]

2. Each B_i has a high rank so that $B_1 u, \ldots, B_p u \in \mathbb{C}^n$ exist in a high dimensional space.

When $\mathcal{B}$ possesses these two properties and p is sufficiently small compared to n, it makes sense that L's columns, $B_1 u, \ldots, B_p u$, are likely to be independent, thus guaranteeing that L is invertible, at least.

We now make the above two properties more precise. Let

$$M = L^* L \in \mathbb{C}^{p \times p} \text{ and } N = \mathbb{E}\, M. \tag{2.2}$$

Clearly, $\text{cond}(M) = \text{cond}(L)^2$. If $\mathbb{E}\, M$ is ill-conditioned, there is little chance that M or L is well-conditioned. This can be related to Property 1 by observing that

$$N_{jk} = \mathbb{E}\, M_{jk} = \text{tr}(B_j{}^* B_k) = \langle B_j, B_k \rangle . \tag{2.3}$$

[1]Note that $\langle \cdot, \cdot \rangle$ is the Frobenius inner product and δ_{jk} is the Kronecker delta.

If $\langle B_j, B_k \rangle \simeq \delta_{jk}$, then the Gram matrix N is approximately the identity matrix which is well-conditioned. Hence, a more quantitative way of putting Property 1 is that we have control over $\kappa(B)$ defined as follows.

Definition 2.1.1. *Let $\mathcal{B} = \{B_1, \ldots, B_p\}$ be a set of matrices. Define its condition number $\kappa(\mathcal{B})$ as the condition number of the matrix $N \in \mathbb{C}^{p \times p}$ where $N_{jk} = \langle B_j, B_k \rangle$.*

On the other hand, Property 2 can be made precise by saying that we have control over $\lambda(\mathcal{B})$ as defined below.

Definition 2.1.2. *Let $A \in \mathbb{C}^{m \times n}$. Define its weak condition number*[2] *as*

$$\lambda(A) = \frac{\|A\| n^{1/2}}{\|A\|_F}.$$

Let $\mathcal{B}$ be a set of matrices. Define its (uniform) weak condition number as

$$\lambda(\mathcal{B}) = \max_{A \in \mathcal{B}} \lambda(A).$$

We justify the nomenclature as follows. Suppose $A \in \mathbb{C}^{n \times n}$ has condition number k, then $\|A\|_F^2 = \sum_{i=1}^n \sigma_i^2 \geq n\sigma_{\min}^2 \geq n \|A\|^2 / k^2$. Taking square root, we obtain $\lambda(A) \leq k$. In other words, any well-conditioned matrix is also weakly well-conditioned. And like the usual condition number, $\lambda(\mathcal{A}) \geq 1$ because we always have $\|A\|_F \leq n^{1/2} \|A\|$.

The numerical rank of a matrix A is $\|A\|_F^2 / \|A\|^2 = n\lambda(A)^{-2}$, thus having a small $\lambda(A)$ is the same as having a high numerical rank. We also want to caution the reader that $\lambda(\mathcal{B})$ is defined very differently from $\kappa(\mathcal{B})$ and is not a weaker version of $\kappa(\mathcal{B})$.

Using classical concentration inequalties, it was shown [23] that when $\lambda(\mathcal{B})$ and $\kappa(\mathcal{B})$ are fixed, $p = \tilde{O}(n^{1/2})$[3] will ensure that L is well-conditioned with high probability.

Here we establish a stronger result, namely that $p = \tilde{O}(n)$ suffices. The implication is that we can expect to recover $\tilde{O}(n)$ instead of $\tilde{O}(n^{1/2})$ coefficients. The exact

[2]Throughout the chapter, $\|\cdot\|$ and $\|\cdot\|_F$ denote the spectral and Frobenius norms respectively.

[3]Note that $\tilde{O}(n)$ denotes $O(n \log^c n)$ for some $c > 0$. In other words, ignore log factors.

statement is presented below.

Theorem 2.1.3 (Main result). *Let $C_1, C_2 > 0$ be numbers given by Remark A.1.3 in the Appendix A.1. Let $\mathcal{B} = \{B_1, \ldots, B_p\}$ where each $B_j \in \mathbb{C}^{m\times n}$. Define $L \in \mathbb{C}^{n\times p}$ such that its j-th column is $B_j u$ where u is either a Gaussian or Rademacher sequence. Let $M = L^*L$, $N = \mathbb{E}\, M$ $\kappa = \kappa(\mathcal{B})$ and $\lambda = \lambda(\mathcal{B})$. Suppose*

$$n \geq p\,(C\kappa\lambda \log n)^2 \text{ for some } C \geq 1.$$

Then

$$\mathbb{P}\left(\|M - N\| \geq \frac{t\,\|N\|}{\kappa}\right) \leq 2C_2 p n^{1-\alpha} \text{ where } \alpha = \frac{tC}{eC_1}.$$

The number C_1 is small. C_2 may be large but it poses no problem because $n^{-\alpha}$ decays very fast with larger n and C. With $t = 1/2$, we deduce that with high probability,

$$\mathrm{cond}(M) \leq 2\kappa + 1.$$

In general, we let $0 < t < 1$ and for the probability bound to be useful, we need $\alpha > 2$, which implies $C > 2eC_1 > 1$. Therefore the assumption that $C \geq 1$ in the theorem can be considered redundant.

We remark that Rauhut and Tropp have a new result (a Bernstein-like tail bound) that may be used to refine the theorem. This will be briefly discussed in Section 2.4.1 where we conduct a numerical experiment.

Note that when u is a Gaussian sequence, M resembles a Wishart matrix for which the distribution of the smallest eigenvalue is well-studied [27]. However, each row of L is not independent, so results from random matrix theory cannot be used in this way.

An intermediate result in the proof of Theorem 2.1.3 is the following. It conveys the essence of Theorem 2.1.3 and may be easier to remember.

Theorem 2.1.4. *Assume the same set-up as in Theorem 2.1.3. Suppose $n = \tilde{O}(p)$. Then*

$$\mathbb{E}\,\|M - N\| \leq C(\log n)\,\|N\|\,(p/n)^{1/2}\lambda \text{ for some } C > 0.$$

A numerical experiment in Section 2.4.1 suggests that the relationship between p and n is not tight in the log factor. Our experiment show that for $\mathbb{E}\,\|M - N\| / \|N\|$ to vanish as $p \to \infty$, n just needs to increase faster than $p\log(np)$, whereas Theorem 2.1.4 requires n to grow faster than $p\log^2 n$.

Next, we see that when L is well-conditioned, the error in the reconstruction is also small.

Proposition 2.1.5. *Assume the same set-up as in Theorem 2.1.3. Suppose* $A = \sum_{j=1}^{p} d_j B_j + E$ *where* $\|E\| \leq \varepsilon$ *and assume whp,*

$$\|M - N\| \leq \frac{t\,\|N\|}{\kappa} \text{ for some } 0 < t < 1.$$

Let $c = L^{+}Au$ *be the recovered coefficients. Then whp,*

$$\left\| A - \sum_{j=1}^{p} c_j B_j \right\| \leq O\left(\varepsilon\lambda \left(\frac{\kappa p}{1-t} \right)^{1/2} \right).$$

If $\varepsilon = o(p^{-1/2})$, then the proposition guarantees that the overall error goes to zero as $p \to \infty$. Of course, a larger n and more computational effort are required.

2.1.3 Multiple probes

Fix n and suppose $p > n$. L is not going to be well-conditioned or even invertible. One way around this is to probe A with multiple random vectors $u_1, \ldots, u_q \in \mathbb{C}^n$ at one go, that is to solve

$$L'c = A'u,$$

where the j-th column of L' and $A'u$ are respectively

$$\begin{pmatrix} B_j u_1 \\ \vdots \\ B_j u_q \end{pmatrix} \text{ and } \begin{pmatrix} Au_1 \\ \vdots \\ Au_q \end{pmatrix}.$$

For this to make sense, $A' = I_q \otimes A$ where I_q is the identity matrix of size q. Also

define $B'_j = I_q \otimes B_j$ and treat the above as probing A' assuming that it lies in the span of $\mathcal{B}' = \{B'_1, \ldots, B'_p\}$.

Regarding the conditioning of L', we can apply Theorem 2.1.3 to A' and $\mathcal{B}'$. It is an easy exercise (cf. Proposition A.3.1) to see that the condition numbers are unchanged, that is $\kappa(\mathcal{B}) = \kappa(\mathcal{B}')$ and $\lambda(\mathcal{B}) = \lambda(\mathcal{B}')$. Applying Theorem 2.1.3 to A' and $\mathcal{B}'$, we deduce that $\mathrm{cond}(L) \leq 2\kappa + 1$ with high probability provided that

$$nq \propto p(\kappa\lambda \log n)^2.$$

Remember that A has only mn degrees of freedom; while we can increase q as much as we like to improve the conditioning of L, the problem set-up does not allow $p > mn$ coefficients. In general, when A has rank $\tilde{n}$, its degrees of freedom is $\tilde{n}(m + n - \tilde{n})$ by considering its SVD.

2.1.4 When to probe

Matrix probing is an especially useful technique when the following holds.

1. We know that the probed matrix A can be approximated by a small number of basis matrices that are specified in advance. This holds for operators with smooth pseudodifferential symbols, which will be studied in Section 2.3.

2. Each matrix B_i can be applied to a vector in $\tilde{O}(\max(m, n))$ time using only $\tilde{O}(\max(m, n))$ memory.

The second condition confers two benefits. First, the coefficients c can be recovered fast, assuming that u and Au are already provided. This is because L can be computed in $\tilde{O}(\max(m, n)p)$ time and (2.1) can be solved in $O(mp^2 + p^3)$ time by QR factorization or other methods. In the case where increasing m, n does not require a bigger $\mathcal{B}$ to approximate A, p can be treated as a constant and the recovery of c takes only $\tilde{O}(\max(m, n))$ time.

Second, given the coefficient vector c, A can be applied to any vector v by summing over $B_i v$'s in $\tilde{O}(\max(m, n)p)$ time . This speeds up iterative methods such as GMRES

Require: $A^+ \simeq \sum_{i=1}^{p} c_i B_i$.
  procedure BACKWARDPROBING($A, B_1, \ldots, B_p$)
    Generate $u \sim N(0,1)^n$ iid.
    Compute $v = Au$.
    Filter away u's components in null(A). Call this $\tilde{u}$.
    Compute L by setting its j-column to $B_j v$.
    Solve for c the system $Lc = \tilde{u}$ in a least squares sense.
    return c
  end procedure

Figure 2-1: Backward matrix probing.

and Arnoldi.

2.1.5 Backward matrix probing

A compelling application of matrix probing is computing the pseudoinverse A^+ of a matrix $A \in \mathbb{C}^{m \times n}$ when A^+ is known to be well-approximated in the space of some $\mathcal{B} = \{B_1, \ldots, B_p\}$. This time, we probe A^+ by applying it to a random vector $v = Au$ where u is a Gaussian or Rademacher sequence that we generate.

Like in Section 2.1.1, define $L \in \mathbb{C}^{n \times p}$ such that its j-th column is $B_j v = B_j Au$. Suppose $A^+ = \sum_{i=1}^{p} c_i B_i$ for some $c_1, \ldots, c_p \in \mathbb{C}$. Then the coefficient vector c can be obtained by solving

$$Lc = A^+ v = A^+ Au. \tag{2.4}$$

The right hand side is u projected onto null$(A)^\perp$ where null(A) is the nullspace of A. When A is invertible, $A^+ Au$ is simply u. We call this "backward matrix probing" because the generated random vector u appears on the opposite side of the matrix being probed in (2.4). The equation suggests a framework for probing A^+ as shown in Figure 2-1.

In order to perform the filtering in Step 3 efficiently, prior knowledge of A may be needed. For example, if A is the Laplacian with periodic boundary conditions, its nullspace is the set of constant functions and Step 3 amounts to subtracting the mean from u. A more involved example can be found in [23]. In this work, we invert

the wave equation Hessian, and Step 3 entails building an illumination mask. Further comments on [23] are located in Section 2.4.5.

For the conditioning of L, we may apply Theorem 2.1.3 with $\mathcal{B}$ replaced with $\mathcal{B}_A := \{B_1A, \ldots, B_pA\}$ since the j-th column of L is now B_jAu. Of course, $\kappa(\mathcal{B}_A)$ and $\lambda(\mathcal{B}_A)$ can be very different from $\kappa(\mathcal{B})$ and $\lambda(\mathcal{B})$; in fact, $\kappa(\mathcal{B}_A)$ and $\lambda(\mathcal{B}_A)$ seem much harder to control because it depends on A. Fortunately, as we shall see in Section 2.3.5, knowing the "order" of A^+ as a pseudodifferential operator helps in keeping these condition numbers small.

When A has a high dimensional nullspace but has comparable nonzero singular values, $\lambda(\mathcal{B}_A)$ may be much larger than is necessary. By a change of basis, we can obtain the following tighter result.

Corollary 2.1.6. *Let $C_1, C_2 > 0$ be numbers given by Remark A.1.3 in the Appendix A.1. Let $A \in \mathbb{C}^{m\times n}$, $\tilde{n} = \text{rank}(A)$ and $\mathcal{B}_A = \{B_1A, \ldots, B_pA\}$ where each $B_j \in \mathbb{C}^{n\times m}$. Define $L \in \mathbb{C}^{n\times p}$ such that its j-th column is B_jAu where $u \sim N(0,1)^n$ iid. Let $M = L^*L$, $N = \mathbb{E}\,M$, $\kappa = \kappa(\mathcal{B}_A)$ and $\lambda = (\tilde{n}/n)^{1/2}\lambda(\mathcal{B}_A)$. Suppose*

$$\tilde{n} \geq p\left(C\kappa\lambda \log \tilde{n}\right)^2 \text{ for some } C \geq 1.$$

Then

$$\mathbb{P}\left(\|M - N\| \geq \frac{t\,\|N\|}{\kappa}\right) \leq (2C_2p)\tilde{n}^{1-\alpha} \text{ where } \alpha = \frac{tC}{eC_1}.$$

Notice that $\tilde{n} = \text{rank}(A)$ has taken the role of n, and our λ is now $\max_{1\leq j\leq p} \frac{\tilde{n}^{1/2}\|B_jA\|}{\|B_jA\|_F}$, which ignores the $n - \tilde{n}$ zero singular values of each B_jA and can be much smaller than $\lambda(\mathcal{B}_A)$.

2.2 Proofs

2.2.1 Proof of Theorem 2.1.3

Our proof is decoupled into two components: one linear algebraic and one probabilistic. The plan is to collect all the results that are linear algebraic, deterministic in

nature, then appeal to a probabilistic result developed in the Appendix A.1.

To facilitate the exposition, we use a different notation for this section. We use lower case letters as *superscripts* that run from 1 to p and Greek symbols as subscripts that run from 1 to n or m. For example, the set of basis matrices is now $\mathcal{B} = \{B^1, \ldots, B^p\}$.

Our linear algebraic results concern the following variables.

1. Let $T^{jk} = B^{j*}B^k \in \mathbb{C}^{n\times n}$ and $T_{\xi\eta} \in \mathbb{C}^{p\times p}$ such that the (j,k)-th entry of $T_{\xi\eta}$ is the (ξ,η)-th entry of T^{jk}.

2. Let $Q = \sum_{1\le\xi,\eta\le n} T^*_{\xi\eta}T_{\xi\eta}$.

3. Let $S = \sum_{j=1}^p B^j B^{j*} \in \mathbb{C}^{m\times m}$.

4. Let F and G be block matrices $(T_{\xi\eta})_{1\le\xi,\eta\le n}$ and $(T^*_{\xi\eta})_{1\le\xi,\eta\le n}$ respectively.

The reason for introducing T is that M can be written as a quadratic form in $T_{\xi\eta}$ with input u:

$$M = \sum_{1\le\xi,\eta\le n} u_\xi u_\eta T_{\xi\eta}.$$

Since u_ξ has unit variance and zero mean, $N = \mathbb{E}\,M = \sum_{\xi=1}^n T_{\xi\xi}$.

Probabilistic inequalties applied to M will involve $T_{\xi\eta}$, which must be related to $\mathcal{B}$. The connection between these n by n matrices and p by p matrices lies in the identity

$$T^{jk}_{\xi\eta} = \sum_{\zeta=1}^m \overline{B^j_{\zeta\xi}} B^k_{\zeta\eta}. \tag{2.5}$$

The linear algebraic results are contained in the following propositions.

Proposition 2.2.1. *For any $1 \le \xi,\eta \le n$, $T_{\xi\eta} = T^*_{\eta\xi}$. Hence, $T_{\xi\xi}, N$ are all Hermitian. Moreover, they are positive semidefinite.*

Proof. Showing that $T_{\xi\eta} = T^*_{\eta\xi}$ is straightforward from (2.5). We now check that $T_{\xi\xi}$ is positive semidefinite. Let $v \in \mathbb{C}^p$. By (2.5), $v^*T_{\xi\xi}v = \sum_\zeta \sum_{jk} \overline{v^j} v^k \overline{B^j_{\zeta\xi}} B^k_{\zeta\xi} = \sum_\zeta \left|\sum_k v^k B^k_{\zeta\xi}\right|^2 \ge 0$. It follows that $N = \sum_\xi T_{\xi\xi}$ is also positive semidefinite. □

Proposition 2.2.2. *$Q^{jk} = \operatorname{tr}(B^{j*}SB^k)$ and $Q = \sum_{1\le\xi,\eta\le n} T_{\xi\eta}T^*_{\xi\eta}$.*

Proof. By (2.5), $Q^{jk} = \sum_l \langle T^{lj}, T^{lk}\rangle = \sum_l \operatorname{tr}(B^{j*}B^lB^{l*}B^k)$. The summation and trace commute to give us the first identity. Similarly, the (j,k)-th entry of $\sum_{\xi\eta} T_{\xi\eta}T^*_{\xi\eta}$ is $\sum_l \langle T^{kl}, T^{jl}\rangle = \sum_l \operatorname{tr}(B^{l*}B^kB^{j*}B^l)$. Cycle the terms in the trace to obtain Q^{jk}. □

Proposition 2.2.3. *Let $u \in \mathbb{C}^p$ be a unit vector. Define $U = \sum_{k=1}^p u^kB^k \in \mathbb{C}^{m\times n}$. Then $\|U\|_F^2 \le \|N\|$.*

Proof. $\|U\|_F^2 = \operatorname{tr}(U^*U) = \operatorname{tr}(\sum_{jk} \overline{u^j}u^kB^{j*}B^k)$. The sum and trace commute and due to (2.3), $\|U\|_F^2 = \sum_{jk} \overline{u^j}u^kN^{jk} \le \|N\|$. □

Proposition 2.2.4. $\|Q\| \le \|S\|\,\|N\|$.

Proof. Q is Hermitian, so $\|Q\| = \max_u u^*Qu$ where $u \in \mathbb{C}^p$ has unit norm. Now let u be an arbitrary unit vector and define $U = \sum_{k=1}^p u^kB^k$. By Proposition 2.2.2, $u^*Qu = \sum_{jk} \overline{u^j}u^kQ^{jk} = \operatorname{tr}(\sum_{jk} \overline{u^j}u^kB^{j*}SB^k) = \operatorname{tr}(U^*SU)$. Since S is positive definite, it follows from "$\|AB\|_F \le \|A\|\,\|B\|_F$" that $u^*Qu = \left\|S^{1/2}U\right\|_F^2 \le \|S\|\,\|U\|_F^2$. By Proposition 2.2.3, $u^*Qu \le \|S\|\,\|N\|$. □

Proposition 2.2.5. *For any $1 \le j \le p$, $\|B^j\| \le \lambda n^{-1/2}\|N\|^{1/2}$. It follows that $\|Q\| = \left\|\sum_{\xi\eta} T_{\xi\eta}T^*_{\xi\eta}\right\| \le p\lambda^2\|N\|^2/n$.*

Proof. We begin by noting that $\|N\| \ge \max_j |N^{jj}| = \max_j \langle B^j, B^j\rangle = \max_j \|B^j\|_F^2$. From Definition 2.1.2, $\|B^j\| \le \lambda n^{-1/2}\|B^j\|_F \le \lambda n^{-1/2}\|N\|^{1/2}$ for any $1 \le j \le p$, which is our first inequality. It follows that $\|S\| \le \sum_{j=1}^p \|B^j\|^2 \le p\lambda^2\|N\|/n$. Apply Propositions 2.2.4 and 2.2.2 to obtain the second inequality. □

Proposition 2.2.6. *F, G are Hermitian, and $\max(\|F\|, \|G\|) \le \lambda^2\|N\|\,(p/n)$.*

Proof. That F, G are Hermitian follow from Proposition 2.2.1. Define $F' = (T^{jk})$ another block matrix. Since reindexing the rows and columns of F does not change its norm, $\|F\| = \|F'\|$. By Proposition 2.2.5,

$$\|F'\|^2 \le \sum_{j,k=1}^p \left\|T^{jk}\right\|^2 \le \sum_{j,k=1}^p \left\|B^j\right\|^2\left\|B^k\right\|^2 \le \lambda^4\|N\|^2(p/n)^2.$$

The same argument works for G. □

We now combine the above linear algebraic results with a probabilistic result in Appendix A.1. Prepare to apply Proposition A.1.7 with A_{ij} replaced with $T_{\xi\eta}$. Note that $R = \sum_{\xi\eta} T_{\xi\eta} T_{\xi\eta}^* = Q$ by Proposition 2.2.2. Bound σ using Propositions 2.2.5 and 2.2.6:

$$\begin{aligned}
\sigma &= C_1 \max(\|Q\|^{1/2}, \|R\|^{1/2}, \|F\|, \|G\|) \\
&\leq C_1 \|N\| \max((p/n)^{1/2}\lambda, (p/n)\lambda^2) \\
&\leq C_1 \|N\| (p/n)^{1/2}\lambda.
\end{aligned}$$

The last step goes through because our assumption on n guarantees that $(p/n)^{1/2}\lambda \leq 1$. Finally, apply Proposition A.1.7 with $t\|N\|/\kappa = e\sigma u$. The proof is complete.

2.2.2 Sketch of the proof for Theorem 2.1.4

Follow the proof of Proposition A.1.7. Letting $s = \log n$, we obtain

$$\begin{aligned}
\mathbb{E}\|M - N\| &\leq (\mathbb{E}\|M - N\|^s)^{1/s} \\
&\leq C_1(2C_2np)^{1/s} s \max(\|Q\|^{1/2}, \|R\|^{1/2}, \|F\|, \|G\|) \\
&\leq C(\log n)\|N\|(p/n)^{1/2}\lambda.
\end{aligned}$$

2.2.3 Proof of Proposition 2.1.5

Recall that A is approximately the linear combination $\sum_{j=1}^p d^j B^j$, while $\sum_{j=1}^p c^j B^j$ is the recovered linear combination. We shall first show that the recovered coefficients c is close to d:

$$\begin{aligned}
\|d - c\| &= \left\|L^+ Au - c\right\| \\
&= \left\|L^+(Lc + Eu) - c\right\| \\
&= \left\|L^+ Eu\right\| \\
&\leq \varepsilon\|u\| \left(\frac{\kappa}{(1-t)\|N\|}\right)^{1/2}.
\end{aligned}$$

Let v be a unit n-vector. Let L' be a $n \times p$ matrix such that its j-th column is $B^j v$. Now,

$$Av - \sum_{j=1}^{p} c^j B^j v = (L'd + Ev) - L'c = Ev + L'(d - c).$$

Combining the two equations, we have

$$\left\| A - \sum_{j=1}^{p} c^j B^j \right\| \leq \varepsilon + \varepsilon \left\| L' \right\| \left\| u \right\| \left(\frac{\kappa}{(1-t)\left\| N \right\|} \right)^{1/2}. \tag{2.6}$$

With overwhelming probability, $\|u\| = O(\sqrt{n})$. The only term left that needs to be bounded is $\|L'\|$. This turns out to be easy because $\|B^j\| \leq \lambda n^{-1/2} \|N\|^{1/2}$ by Proposition 2.2.5 and $\|L'\|^2 \leq \sum_{j=1}^{p} \|B^j v\|^2 \leq \lambda^2 \|N\| p/n$. Substitute this into (2.6) to finish the proof.

2.2.4 Proof of Corollary 2.1.6

Let $u \sim N(0,1)^n$ iid. Say A has a singular value decomposition $E\Lambda F^*$ where Λ is diagonal. Do a change of basis by letting $u' = F^* u \sim N(0,1)^n$ iid, $B'_j = F^* B_j E$ and $\mathcal{B}'_\Lambda = \{B'_1\Lambda, \ldots, B'_p\Lambda\}$. (2.1) is reduced to $L'c = \Lambda u'$ where the j-th column of L' is $B'_j \Lambda u'$.

Since Frobenius inner products, $\|\cdot\|$ and $\|\cdot\|_F$ are all preserved under unitary transformations, it is clear that $\kappa(\mathcal{B}'_\Lambda) = \kappa(\mathcal{B}_A)$ and $\lambda(\mathcal{B}'_\Lambda) = \lambda(\mathcal{B}_A)$. Essentially, for our purpose here, we may pretend that $A = \Lambda$.

Let $\tilde{n} = \operatorname{rank}(A)$. If A has a large nullspace, i.e., $\tilde{n} \ll \min(m,n)$, then $B'_j\Lambda$ has $n - \tilde{n}$ columns of zeros and many components of u' are never transmitted to the B'_j's anyway. We may therefore truncate the length of u' to $\tilde{n}$, let $\tilde{B}_j \in \mathbb{C}^{n \times \tilde{n}}$ be $B'_j\Lambda$ with its columns of zeros chopped away and apply Theorem 2.1.3 with $\mathcal{B}$ replaced with $\tilde{\mathcal{B}} := \{\tilde{B}_1, \ldots, \tilde{B}_p\}$. Observe that $\kappa(\tilde{\mathcal{B}}) = \kappa(\mathcal{B}'_\Lambda)$, whereas $\lambda(\tilde{\mathcal{B}}) = (\tilde{n}/n)^{1/2}\lambda(\mathcal{B}'_\Lambda)$ because $\left\|\tilde{B}_j\right\|_F = \left\|B'_j\Lambda\right\|_F$ and $\left\|\tilde{B}_j\right\| = \left\|B'_j\Lambda\right\|$ but $\tilde{B}_j$ has $\tilde{n}$ instead of n columns. The proof is complete.

2.3 Probing operators with smooth symbols

2.3.1 Basics and assumptions

We begin by defining what a pseudodifferential symbol is.

Definition 2.3.1. *Every linear operator A is associated with a pseudodifferential symbol $a(x,\xi)$ such that for any $u : \mathbb{R}^d \to \mathbb{R}$,*

$$Au(x) = \int_{\xi \in \mathbb{R}^d} e^{2\pi i \xi \cdot x} a(x,\xi) \hat{u}(\xi) d\xi \tag{2.7}$$

where $\hat{u}$ is the Fourier transform of u, that is $\hat{u}(\xi) = \int_{x \in \mathbb{R}^d} u(x) e^{-2\pi i \xi \cdot x} dx$.

We refrain from calling A a "pseudodifferential operator" at this point because its symbol has to satisfy some additional constraints that will be covered in Section 2.3.5. What is worth noting here is the Schwartz kernel theorem which shows that every linear operator $A : \mathcal{S}(\mathbb{R}^d) \to \mathcal{S}'(\mathbb{R}^d)$ has a symbol representation as in (2.7) and in that integral, $a(x,\xi) \in \mathcal{S}'(\mathbb{R}^d \times \mathbb{R}^d)$ acts as a distribution. Recall that $\mathcal{S}$ is the Schwartz space and $\mathcal{S}'$ is its dual or the space of tempered distributions. The interested reader may refer to [28] or [70] for a deeper discourse.

The term "pseudodifferential" arises from the fact that differential operators have very simple symbols. For example, the Laplacian has the symbol $a(x,\xi) = -4\pi^2 \|\xi\|^2$. Another example is

$$Au(x) = u(x) - \nabla \cdot \alpha(x) ablau(x) \text{ for some } \alpha(x) \in C^1(\mathbb{R}^d).$$

Its symbol is

$$a(x,\xi) = 1 + \alpha(x)(4\pi^2 \|\xi\|^2) - \sum_{k=1}^{d} (2\pi i \xi_k) \partial_{x_k} \alpha(x). \tag{2.8}$$

Clearly, if the media $\alpha(x)$ is smooth, so is the symbol $a(x,\xi)$ smooth in both x and ξ, an important property which will be used in Section 2.3.3.

For practical reasons, we make the following assumptions about $u : \mathbb{R}^d \to \mathbb{R}$ on which symbols are applied.

1. u is periodic with period 1, so only $\xi \in \mathbb{Z}^d$ will be considered in the Fourier domain.

2. u is bandlimited, say $\hat{u}$ is supported on $\Xi := [-\xi_0, \xi_0]^d \subseteq \mathbb{Z}^d$. Any summation over the Fourier domain is by default over Ξ.[4]

3. $a(x,\xi)$ and $u(x)$ are only evaluated at $x \in X \subset [0,1]^d$ which are points uniformly spaced apart. Any summation over x is by default over X.

Subsequently, (2.7) reduces to a discrete and finite form:

$$Au(x) = \sum_{\xi \in \Xi} e^{2\pi i \xi \cdot x} a(x,\xi) \hat{u}(\xi). \tag{2.9}$$

We like to call $a(x,\xi)$ a "discrete symbol." Some tools are already available for manipulating such symbols [24].

2.3.2 User friendly representations of symbols

Given a linear operator A, it is useful to relate its symbol $a(x,\xi)$ to its matrix representation in the Fourier basis. This helps us understand the symbol as a matrix and also exposes easy ways of computing the symbols of A^{-1}, A^* and AB using standard linear algebra software.

By a matrix representation $(A_{\eta\xi})$ in Fourier basis, we mean of course that $\widehat{Au}(\eta) = \sum_\xi A_{\eta\xi}\hat{u}(\xi)$ for any η. We also introduce a more compact form of the symbol: $\hat{a}(j,\xi) = \int_x a(x,\xi)e^{-2\pi i j\cdot x}dx$. The next few results are pedagogical and listed for future reference.

Proposition 2.3.2. *Let A be a linear operator with symbol $a(x,\xi)$. Let $(A_{\eta\xi})$ and $\hat{a}(j,\xi)$ be as defined above. Then*

$$A_{\eta\xi} = \int_x a(x,\xi)e^{-2\pi i(\eta-\xi)x}dx; \quad a(x,\xi) = e^{-2\pi i\xi x}\sum_\eta e^{2\pi i\eta x}A_{\eta\xi};$$

[4] To have an even number of points per dimension, one can use $\Xi = [-\xi_0, \xi_0 - 1]^d$ for example. We leave this generalization to the reader and continue to assume $\xi \in [-\xi_0, \xi_0]^d$.

$$A_{\eta\xi} = \hat{a}(\eta - \xi, \xi); \quad \hat{a}(j, \xi) = A_{j+\xi,\xi}.$$

Proof. Let $\eta = \xi + j$ and apply the definitions. □

Proposition 2.3.3 (Trace). *Let A be a linear operator with symbol $a(x, \xi)$. Then*

$$\mathrm{tr}(A) = \sum_{\xi} \hat{a}(0, \xi) = \sum_{\xi} \int_x a(x, \xi) dx.$$

Proposition 2.3.4 (Adjoint). *Let A and $C = A^*$ be linear operators with symbols $a(x, \xi), c(x, \xi)$. Then*

$$\hat{c}(j, \xi) = \overline{\hat{a}(-j, j + \xi)}; \quad c(x, \xi) = \sum_{\eta} \int_y \overline{a(y, \eta)} e^{2\pi i(\eta - \xi)(x - y)} dy.$$

Proposition 2.3.5 (Composition). *Let A, B and $C = AB$ be linear operators with symbols $a(x, \xi), b(x, \xi), c(x, \xi)$. Then*

$$\hat{c}(j, \xi) = \sum_{\zeta} \hat{a}(j + \xi - \zeta, \zeta) \hat{b}(\zeta - \xi, \xi);$$

$$c(x, \xi) = \sum_{\zeta} \int_y e^{2\pi i(\zeta - \xi)(x - y)} a(x, \zeta) b(y, \xi) dy.$$

We leave it to the reader to verify the above results.

2.3.3 Symbol expansions

The idea is that when a linear operator A has a smooth symbol $a(x, \xi)$, only a few basis functions are needed to approximate a, and correspondingly only a small $\mathcal{B}$ is needed to represent A. This is not new, see for example [24]. In this paper, we consider the separable expansion

$$a(x, \xi) = \sum_{jk} c_{jk} e_j(x) g_k(\xi).$$

This is the same as expanding A as $\sum_{jk} c_{jk} B_{jk}$ where the symbol for B_{jk} is $e_j(x) g_k(\xi)$. With an abuse of notation, let B_{jk} also denote its matrix representa-

procedure APPLYSYMBOL($u(x)$) ▷ Apply the symbol $e_j(x)g_k(\xi)$ to $u(x)$
Perform FFT on u to obtain $\hat{u}(\xi)$.
Multiply $\hat{u}(\xi)$ by $g_k(\xi)$ elementwise.
Perform IFFT on the previous result, obtaining $\sum_\xi e^{2\pi i\xi\cdot x} g_k(\xi)\hat{u}(\xi)$.
Multiply the previous result by $e_j(x)$ elementwise.
end procedure

Figure 2-2: Apply elementary symbol to $u(x)$.

tion *in Fourier basis.* Given our assumption that $\xi \in [-\xi_0, \xi_0]^d$, we have $B_{jk} \in \mathbb{C}^{n\times n}$ where $n = (2\xi_0 + 1)^d$. As its symbol is separable, B_{jk} can be factorized as

$$B_{jk} = \mathcal{F}\,\mathrm{diag}(e_j(x))\mathcal{F}^{-1}\,\mathrm{diag}(g_k(\xi)) \tag{2.10}$$

where $\mathcal{F}$ is the unitary Fourier matrix. An alternative way of viewing B_{jk} is that it takes its input $\hat{u}(\xi)$, multiply by $g_k(\xi)$ and convolve it with $\hat{e}_j(\eta)$, the Fourier transform of $e_j(x)$. There is also an obvious algorithm to apply B_{jk} to $u(x)$ in $\tilde{O}(n)$ time as outlined in Figure 2-2. As mentioned in Section 2.1.4, this speeds up the recovery of the coefficients c and makes matrix probing a cheap operation.

Recall that for L to be well-conditioned with high probability, we need to check whether N, as defined in (2.3), is well-conditioned, or in a rough sense whether $\langle B_j, B_k\rangle \simeq \delta_{jk}$. For separable symbols, this inner product is easy to compute.

Proposition 2.3.6. *Let $B_{jk}, B_{j'k'} \in \mathbb{C}^{n\times n}$ be matrix representations (in Fourier basis) of linear operators with symbols $e_j(x)g_k(\xi)$ and $e_{j'}(x)g_{k'}(\xi)$. Then*

$$\langle B_{jk}, B_{j'k'}\rangle = \langle e_j, e_{j'}\rangle \langle g_k, g_{k'}\rangle$$

where $\langle e_j, e_{j'}\rangle = \frac{1}{n}\sum_{i=1}^n \overline{e_j(x_i)}e_{j'}(x_i)$ and $x_1, \ldots, x_n$ are points in $[0,1]^d$ uniformly spaced, and $\langle g_k, g_{k'}\rangle = \sum_\xi \overline{g_k(\xi)}g_k(\xi)$.

Proof. Apply Propositions 2.3.3, 2.3.4 and 2.3.5 with the symbols in the $\hat{a}(\eta,\xi)$ form. □

To compute $\lambda(\mathcal{B})$ as in Definition 2.1.2, we examine the spectrum of B_{jk} for every

j, k. A simple and relevant result is as follows.

Proposition 2.3.7. *Assume the same set-up as in Proposition 2.3.6. Then*

$$\sigma_{\min}(B_{jk}) \geq \min_x |e_j(x)| \min_\xi |g_k(\xi)|; \quad \sigma_{\max}(B_{jk}) \leq \max_x |e_j(x)| \max_\xi |g_k(\xi)|.$$

Proof. In (2.10), $\mathcal{F}\,\mathrm{diag}(e^j(x))\mathcal{F}^{-1}$ has singular values $|e_j(x)|$ as x varies over X, defined at the end of Section 2.3.1. The result follows from the min-max theorem. □

As an example, suppose $a(x, \xi)$ is smooth and periodic in both x and ξ. It is well-known that a Fourier series is good expansion scheme because the smoother $a(x, \xi)$ is as a periodic function in x, the faster its Fourier coefficients decay, and less is lost when we truncate the Fourier series. Hence, we pick[5]

$$e_j(x) = e^{2\pi i j \cdot x}; \quad g_k(\xi) = e^{2\pi i k \cdot \varphi(\xi)}, \tag{2.11}$$

where $\varphi(\xi) = (\xi + \xi_0)/(2\xi_0 + 1)$ maps ξ into $[0, 1]^d$.

Due to Proposition 2.3.6, $N = \mathbb{E}\,M$ is a multiple of the identity matrix and $\kappa(\mathcal{B}) = 1$ where $\mathcal{B} = \{B_{jk}\}$. It is also immediate from Proposition 2.3.7 that $\lambda(B_{jk}) = 1$ for every j, k, and $\lambda(\mathcal{B}) = 1$. The optimal condition numbers of this $\mathcal{B}$ make it suitable for matrix probing.

2.3.4 Chebyshev expansion of symbols

The symbols of differential operators are polynomials in ξ and nonperiodic. When probing these operators, a Chebyshev expansion in ξ is in principle favored over a Fourier expansion, which may suffer from the Gibbs phenomenon. However, as we shall see, $\kappa(\mathcal{B})$ grows with p and can lead to ill-conditioning.

For simplicity, assume that the symbol is periodic in x and that $e_j(x) = e^{2\pi i j \cdot x}$. Applying Proposition 2.3.2, we see that B_{jk} is a matrix with a displaced diagonal and its singular values are $(g_k(\xi))_{\xi \in \Xi}$. (Recall that we denote the matrix representation (in Fourier basis) of B_{jk} as B_{jk} as well.)

[5] Actually, $\exp(2\pi i k \xi_0/(2\xi_0 + 1))$ does not vary with ξ, and we can use $\varphi(\xi) = \xi/(2\xi_0 + 1)$.

Let T_k be the k-th Chebyshev polynomial. In 1D, we can pick

$$g_k(\xi) = T_k(\xi/\xi_0) \text{ for } k = 1, \ldots, K. \tag{2.12}$$

Define $\|T_k\|_2 = (\int_{z=-1}^{1} T_k(z)^2 dz)^{1/2}$. By approximating sums with integrals, $\lambda(B_{jk}) \simeq \sqrt{2}\,\|T_k\|_2^{-1} = \left(\frac{4k^2-1}{2k^2-1}\right)^{1/2}$. Notice that there is no $(1-z^2)^{-1/2}$ weight factor in the definition of $\|T_k\|_2$ because $e_j(x)T_k(\xi)$ is treated as a pseudodifferential symbol and has to be evaluated on the *uniform* grid. In practice, this approximation becomes very accurate with larger n and we see no need to be rigorous here. As k increases, $\lambda(B_{jk})$ approaches $\sqrt{2}$. More importantly, $\lambda(B_{jk}) \le \lambda(B_{j1})$ for any j, k, so

$$\lambda(\mathcal{B}) = \sqrt{3}.$$

Applying the same technique to approximate the sum $\langle g_k, g_{k'} \rangle$, we find that $\langle g_k, g_{k'} \rangle \propto (1-(k+k')^2)^{-1} + (1-(k-k')^2)^{-1}$ when $k+k'$ is even, and zero otherwise. We then compute $N = \mathbb{E}\,M$ for various K and plot $\kappa(\mathcal{B})$ versus K, the number of Chebyshev polynomials. As shown in Figure 2-3(a), $\kappa(\mathcal{B}) \simeq 1.3K$. This means that if we expect to recover $p = \tilde{O}(n)$ coefficients, we must keep K fixed. Otherwise, if $p = K^2$, only $p = \tilde{O}(n^{1/2})$ are guaranteed to be recovered by Theorem 2.1.3.

In 2D, a plausible expansion is

$$g_k(\xi) = e^{ik_1 \arg \xi} T_{k_2}(\varphi(\|\xi\|)) \text{ for } 1 \le k_2 \le K \tag{2.13}$$

where $k = (k_1, k_2)$ and $\varphi(r) = (\sqrt{2}r/\xi_0) - 1$ maps $\|\xi\|$ into $[-1, 1]$. We call this the "Chebyshev on a disk" expansion.

The quantity $\lambda(B_{jk})$ is approximately $2\left(\int_{x=-1}^{1}\int_{y=-1}^{1} T_k(\psi(x,y))^2 dx\,dy\right)^{-1/2}$ where $\psi(x,y) = (2x^2+2y^2)^{1/2} - 1$. The integral is evaluated numerically and appears to

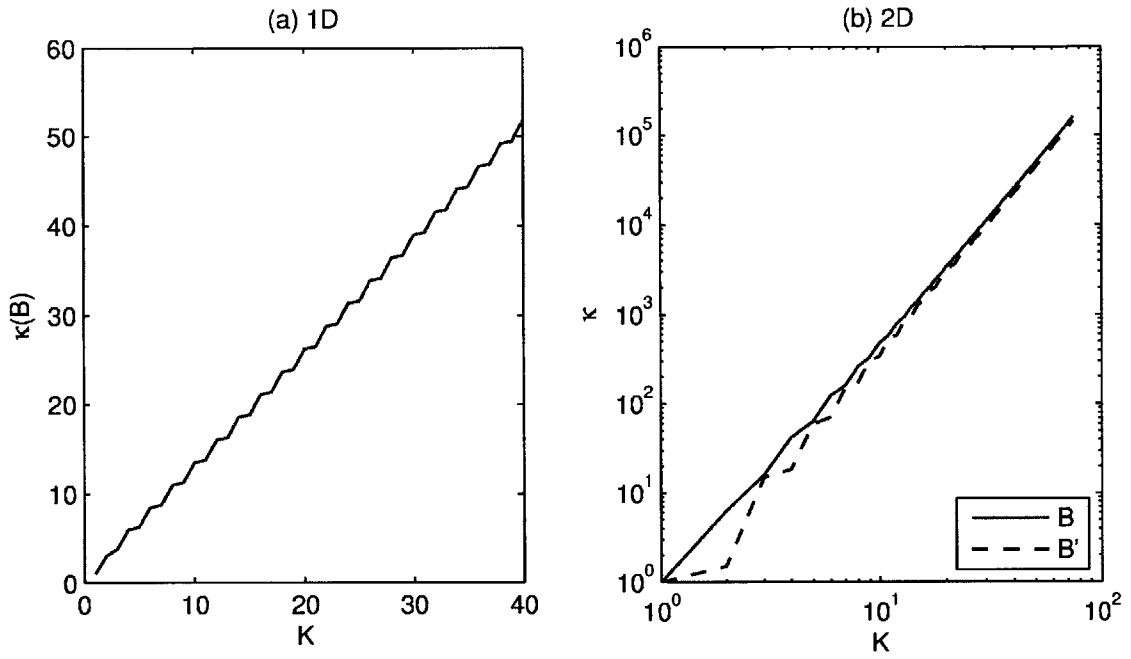


Figure 2-3: Let K be the number of Chebyshev polynomials used in the expansion of the symbol, see (2.12) and (2.13). Observe that in 1D, $\kappa(\mathcal{B}) = O(K)$ while in 2D, $\kappa(\mathcal{B}) = O(K^3)$. These condition numbers mean that we cannot expect to retrieve $p = \tilde{O}(n)$ parameters unless K is fixed and independent of p, n.

converge[6] to $\sqrt{2}$ for large k_2. Also, $k_2 = 1$ again produces the worst $\lambda(B_{jk})$ and

$$\lambda(\mathcal{B}) \leq 2.43.^{7}$$

As for $\kappa(\mathcal{B})$, observe that when $k_1 \neq k_1'$, $\langle g_{k_1 k_2}, g_{k_1' k_2'} \rangle = \pm 1$ due to symmetry[8], whereas when $k_1 = k_1'$, the inner product is proportional to n and is much larger. As a result, the g_k's with different k_1's hardly interact and in studying $\kappa(\mathcal{B})$, one may assume that $k_1 = k_1' = 0$. To improve $\kappa(\mathcal{B})$, we can normalize g_k such that the diagonal entries of N are all ones, that is $g_k'(\xi) = g_k(\xi) / \|g_k(\xi)\|$.

[6]This is because when we truncate the disk of radius $\xi_0\sqrt{2}$ to a square of length $2\xi_0$, most is lost along the vertical axis and away from the diagonals. However, for large k, T_k oscillates very much and the truncation does not matter. If we pretend that the square is a disk, then we are back in the 1D case where the answer approaches $\sqrt{2}$ for large k.

[7]The exact value is $2(4 - \frac{8}{3}\sqrt{2}\sinh^{-1}(1))^{-1/2}$.

[8]The ξ and $-\xi$ terms cancel each other. Only $\xi = 0$ contributes to the sum.

This yields another set of basis matrices $\mathcal{B}'$. Figure 2-3(b) reveals that

$$\kappa(\mathcal{B}) = O(K^3) \text{ and } \kappa(\mathcal{B}') \simeq \kappa(\mathcal{B}).$$

The latter can be explained as follows: we saw earlier that $\langle B_{jk}, B_{jk} \rangle$ converges as k_2 increases, so the diagonal entries of N are about the same and the normalization is only a minor correction.

If $a(x, \xi)$ is expanded using the same number of basis functions in each direction of x and ξ, i.e., $K = p^{1/4}$, then Theorem 2.1.3 suggests that only $p = \tilde{O}(n^{2/5})$ coefficients can be recovered.

To recap, for both 1D and 2D, $\lambda(\mathcal{B})$ is a small number but $\kappa(\mathcal{B})$ increases with K. Fortunately, if we know that the operator being probed is a second order derivative for example, we can fix $K = 2$.

Numerically, we have observed that the Chebyshev expansion can produce dramatically better results than the Fourier expansion of the symbol. More details can be found in Section 2.4.3.

2.3.5 Order of an operator

In standard texts, A is said to be a pseudodifferential operator of order w if its symbol $a(x, \xi)$ is in $C^\infty(\mathbb{R}^d \times \mathbb{R}^d)$ and for any multi-indices α, β, there exists a constant $C_{\alpha\beta}$ such that

$$|\partial_\xi^\alpha \partial_x^\beta a(x, \xi)| \leq C_{\alpha\beta} [\xi]^{w-|\alpha|} \text{ for all } \xi, \text{ where } [\xi] = 1 + \|\xi\| .$$

Letting $\alpha = \beta = 0$, we see that such operators have symbols that grow or decay as $(1 + \|\xi\|)^w$. As an example, the Laplacian is of order 2. The factor 1 prevents $[\xi]$ from blowing up when $\xi = 0$. There is nothing special about it and if we take extra care when evaluating the symbol at $\xi = 0$, we can use

$$[\xi] = \|\xi\| .$$

For forward matrix probing, if it is known a priori that $a(x,\xi)$ behaves like $[\xi]^w$, it makes sense to expand $a(x,\xi)[\xi]^{-w}$ instead. Another way of viewing this is that the symbol of the operator B_{jk} is modified from $e_j(x)g_k(\xi)$ to $e_j(x)g_k(\xi)[\xi]^w$ to suit A better.

For backward matrix probing, if A is of order z, then A^{-1} is of order $-z$ and we should replace the symbol of B_{jk} with $e_j(x)g_k(\xi)[\xi]^{-w}$. We believe that this small correction has an impact on the accuracy of matrix probing, as well as the condition numbers $\kappa(\mathcal{B}_A)$ and $\lambda(\mathcal{B}_A)$.

Recall that an element of $\mathcal{B}_A$ is $B_{jk}A$. If A is of order w and B_{jk} is of order 0, then $B_{jk}A$ is of order w and $\lambda(B_{jk}A)$ will grow with n^w, which will adversely affect the conditioning of matrix probing. However, by multiplying the symbol of B_{jk} by $[\xi]^{-w}$, we can expect $B_{jk}A$ to be order 0 and that $\lambda(B_{jk}A)$ is independent of the size of the problem n. The argument is heuristical but we will support it with some numerical evidence in Section 2.4.3.

2.4 Numerical examples

We carry out four different experiments. The first experiment suggests that Theorem 2.1.4 is not tight. The second experiment presents the output of backward probing in a visual way. In the third experiment, we explore the limitations of backward probing and also tests the Chebyshev expansion of symbols. The last experiment involves the forward probing of the foveation operator, which is related to human vision.

2.4.1 1D statistical study

We are interested in whether the probability bound in Theorem 2.1.3 is tight with respect to p and n, but as the tail probabilities are small and hard to estimate, we opt to study the first moment instead. In particular, if Theorem 2.1.4 captures exactly the dependence of $\mathbb{E}\,\|M-N\|/\|N\|$ on p and n, then we would need n to grow faster than $p\log^2 n$ for $\mathbb{E}\,\|M-N\|/\|N\|$ to vanish, assuming $\lambda(\mathcal{B})$ is fixed.

For simplicity, we use the Fourier expansion of the symbol in 1D so that $\lambda(\mathcal{B}) =$

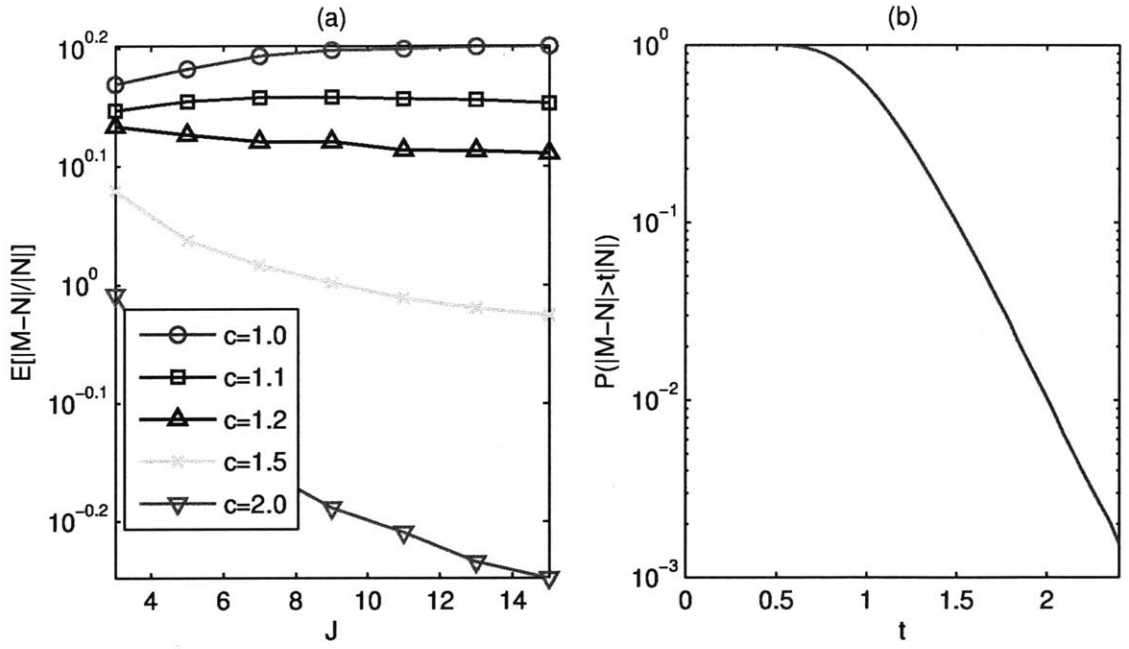


Figure 2-4: Consider the Fourier expansion of the symbol. J is the number of basis functions in x and ξ, so $p = J^2$. Let $n = p \log^c p$. Figure (a) shows that the estimated $\mathbb{E} \|M - N\| / \|N\|$ decays for $c \geq 1.1$ which suggests that Theorem 2.1.4 is not tight. In Figure (b), we estimate $\mathbb{P}(\|M - N\| / \|N\| > t)$ by sampling $\|M - N\| / \|N\|$ 10^5 times. The tail probability appears to be subgaussian for small t and subexponential for larger t.

$\kappa(\mathcal{B}) = 1$. Let J be the number of basis functions in both x and ξ and $p = J^2$. Figure 2-4(a) suggests that $\mathbb{E} \|M - N\| / \|N\|$ decays to zero when $n = p \log^c p$ and $c > 1$. It follows from the previous paragraph that Theorem 2.1.4 cannot be tight.

Nevertheless, Theorem 2.1.4 is optimal in the following sense. Imagine a more general bound

$$\mathbb{E} \frac{\|M - N\|}{\|N\|} \leq (\log^\alpha n) \left(\frac{p}{n}\right)^\beta \text{ for some } \alpha, \beta > 0. \tag{2.14}$$

In Figure 2-5(a), we see that for various values of p/n, $\alpha = 1$ since the graphs are linear. On the other hand, if we fix p and vary n, the log-log graph of Figure 2-5(b) shows that $\beta = 1/2$. Therefore, any bound in the form of (2.14) is no better than Theorem 2.1.4.

Next, we fix $p = 25, n = 51$ and sample $\|M - N\| / \|N\|$ many times to esti-

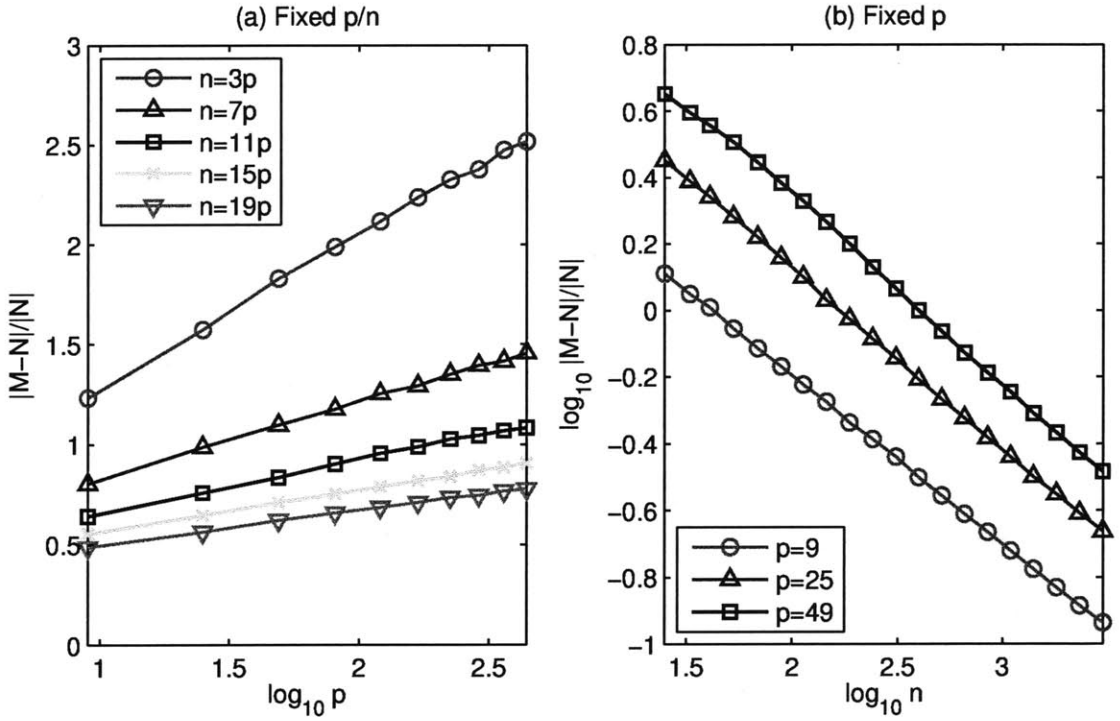


Figure 2-5: Consider bounding $\mathbb{E}\,\|M - N\| \,/\, \|N\|$ by $(\log^{\alpha} n)(p/n)^{\beta}$. There is little loss in replacing $\log n$ with $\log p$ in the simulation. In Figure (a), the estimated $\mathbb{E}\,\|M - N\| \,/\, \|N\|$ depends linearly on $\log p$, so $\alpha \geq 1$. In Figure (b), we fix p and find that for large n, $\beta = 1/2$. The conclusion is that the bound in Theorem 2.1.4 has the best α, β.

mate the tail probabilities. In Figure 2-4(b), we see that the tail probability of $\mathbb{P}(\|M - N\| / \|N\| > t)$ decays as $\exp(-c_1 t)$ when t is big, and as $\exp(-c_2 t^2)$ when t is small, for some positive numbers c_1, c_2. This behavior may be explained by Rauhut and Tropp's yet published result.

2.4.2 Elliptic equation in 1D

We find it instructive to consider a 1D example of matrix probing because it is easy to visualize the symbol $a(x, \xi)$. Consider the operator

$$Au(x) = -\frac{d}{dx}\alpha(x)\frac{du(x)}{dx} \text{ where } \alpha(x) = 1 + 0.4\cos(4\pi x) + 0.2\cos(6\pi x). \quad (2.15)$$

Note that we use periodic boundaries and A is positive semidefinite with a one dimensional nullspace consisting of constant functions.

We probe for A^+ according to Figure 2-1 and the Fourier expansion of its symbol or (2.11). Since A is of order 2, we premultiply $g_k(\xi)$ by $[\xi]^{-2}$ as explained in Section 2.3.5.

In the experiment, $n = 201$ and there are two other parameters J, K which are the number of e_j's and g_k's used in (2.11). To be clear, $-\frac{J-1}{2} \leq j \leq \frac{J-1}{2}$ and $-\frac{K-1}{2} \leq k \leq \frac{K-1}{2}$.

Let C be the output of matrix probing. In Figure 2-6(b), we see that $J = K = 5$ is not enough to represent A^+ properly. This is expected because our media $\alpha(x)$ has a bandwidth of 7. We expect $J = K = 13$ to do better, but the much larger p leads to overfitting and a poor result, as is evident from the wobbles in the symbol of C in Figure 2-6(c). Probing with four random vectors, we obtain a much better result as shown in Figure 2-6(d).

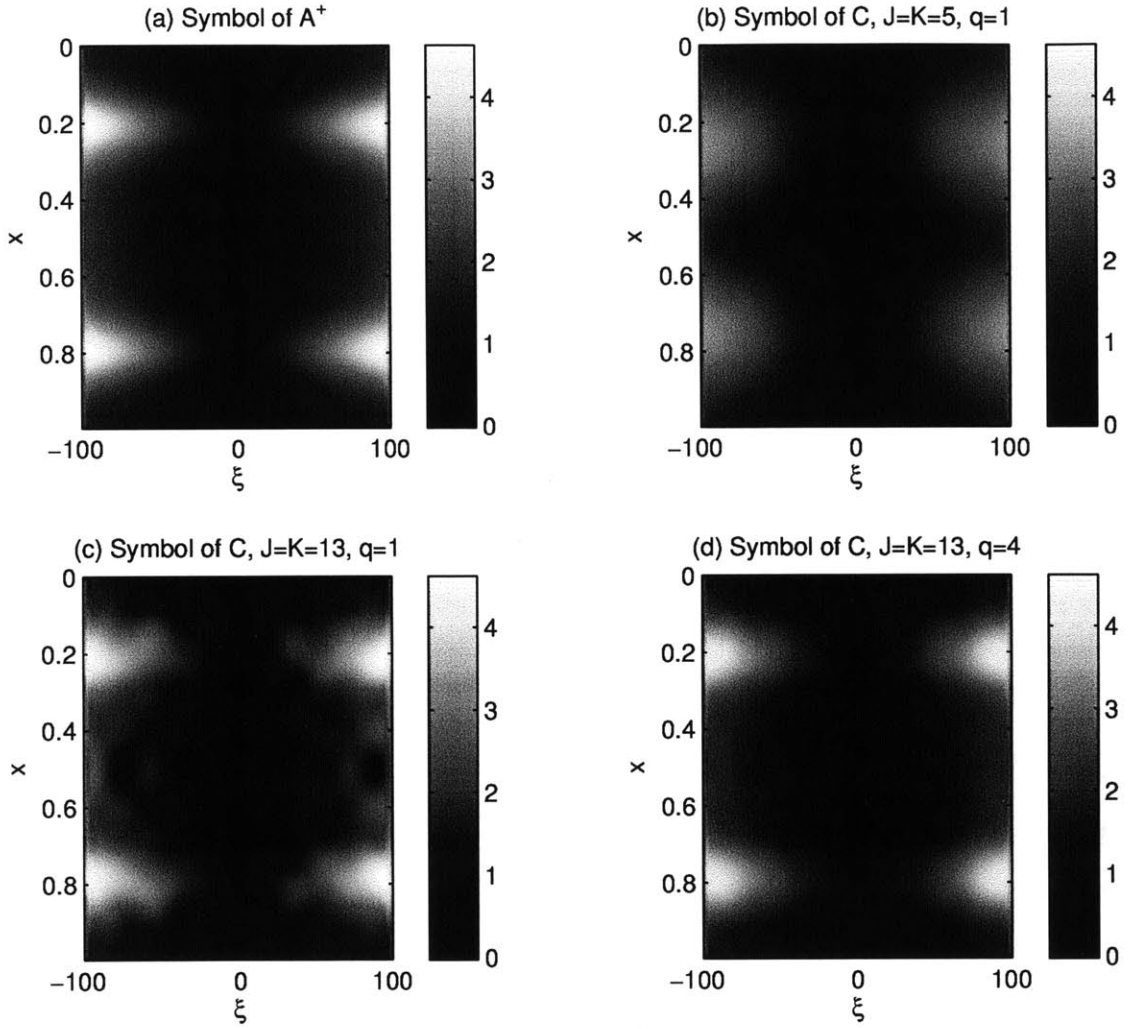


Figure 2-6: Let A be the 1D elliptic operator in (2.15) and A^+ be its pseudoinverse. Let C be the output of backward matrix probing with the following parameters: q is the number of random vectors applied to A^+; J, K are the number of e_j's and g_k's used to expand the symbol of A^+ in (2.11). Figure (a) is the symbol of A^+. Figure (b) is the symbol of C with $J = K = 5$. It lacks the sharp features of Figure (a) because $\mathcal{B}$ is too small to represent A^+ well. With $J = K = 13$, probing with only one random vector leads to ill-conditioning and an inaccurate result in Figure (b). In Figure (c), four random vectors are used and a much better result is obtained. Note that the symbols are multipled by $[\xi]^3$ for better visual contrast.

2.4.3 Elliptic in 2D

In this section, we extend the previous set-up to 2D and address a different set of questions. Consider the operator A defined as

$$Au(x) = -\nabla \cdot \alpha(x)\nabla u(x) \text{ where } \alpha(x) = \frac{1}{T} + \cos^2(\pi\gamma x_1)\sin^2(\pi\gamma x_2). \tag{2.16}$$

The positive value T is called the contrast while the positive integer γ is the roughness of the media, since the bandwidth of $\alpha(x)$ is $2\gamma + 1$. Again, we assume periodic boundary conditions such that A's nullspace is exactly the set of constant functions.

Let C be the output of the backward probing of A. As we shall see, the quality of C drops as we increase the contrast T or the roughness γ.

Fix $n = 101^2$ and expand the symbol using (2.11). Let $J = K$ be the number of basis functions used to expand the symbol in each of its four dimensions, that is $p = J^4$.

In Figure 2-7(b), we see that between $J = 2\gamma - 1$ and $J = 2\gamma + 1$, the bandwidth of the media, there is a marked improvement in the preconditioner, as measured by the ratio $\text{cond}(CA)/\text{cond}(A)$.[9]

On the other hand, Figure 2-7(a) shows that as the contrast increases, the preconditioner C degrades in performance, but the improvement between $J = 2\gamma - 1$ and $2\gamma + 1$ becomes more pronounced.

The error bars in Figure 2-7 are not error margins but $\hat{\sigma}$ where $\hat{\sigma}^2$ is the unbiased estimator of the variance. They indicate that $\text{cond}(CA)/\text{cond}(A)$ is tightly concentrated around its mean, provided J is not too much larger than is necessary. For instance, for $\gamma = 1$, $J = 3$ already works well but pushing to $J = 9$ leads to greater uncertainty.

Next, we consider *forward probing* of A using the "Chebyshev on a disk" expansion or (2.13). Let m be the order correction, that is we multiply $g_k(\xi)$ by $[\xi]^m = \|\xi\|^m$.

[9]Since A has one zero singular value, $\text{cond}(A)$ actually refers to the ratio between its largest singular value and its second smallest singular value. The same applies to CA.

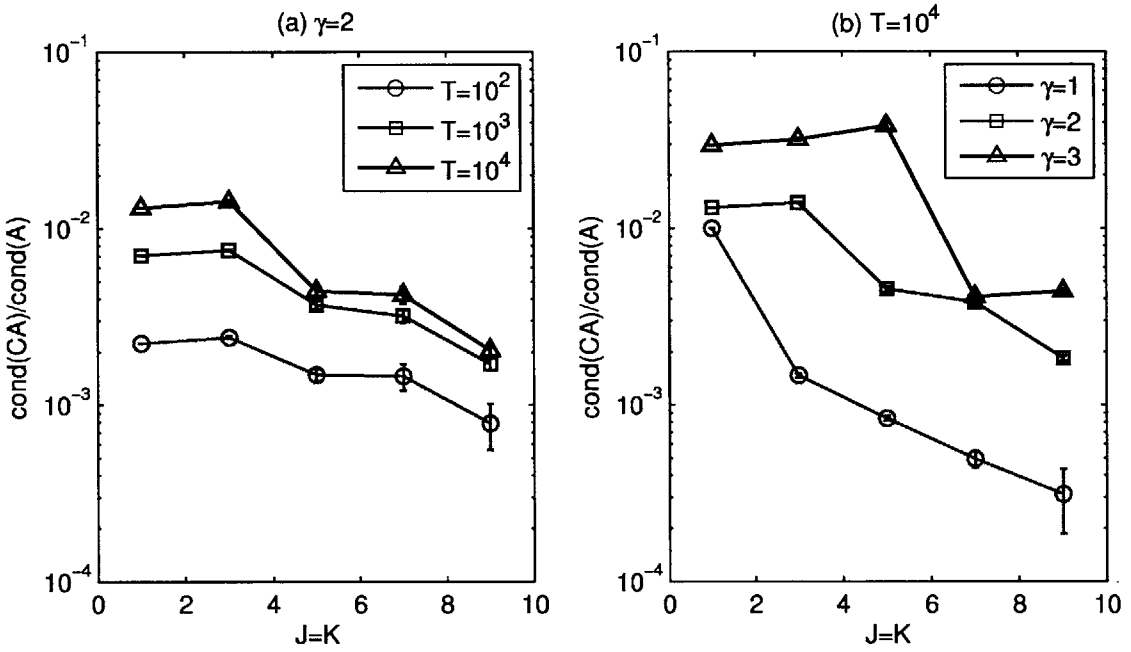


Figure 2-7: Let A be the operator defined in (2.16) and C be the output of backward probing. In Figure (b), we fix $T = 10^4$ and find that as J goes from $2\gamma - 1$ to $2\gamma + 1$, the bandwidth of the media, the quality of the preconditioner C improves by a factor between $10^{0.5}$ and 10. In Figure (a), we fix $\gamma = 2$ and find that increasing the contrast worsens $\text{cond}(CA)/\text{cond}(A)$. Nevertheless, the improvement between $J = 3$ and $J = 5$ becomes more distinct. The error bars correspond to $\hat{\sigma}$ where $\hat{\sigma}^2$ is the estimated variance. They indicate that C is not just good on average, but good with high probability.

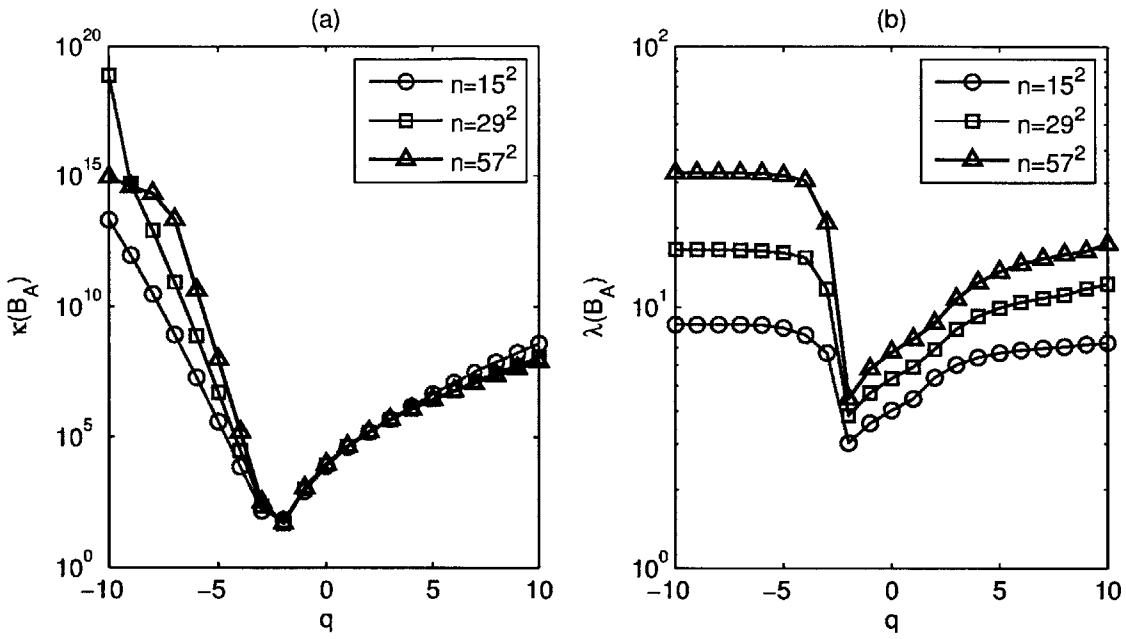


Figure 2-8: Consider the backward probing of A in (2.16), a pseudodifferential oeprator of order 2. Perform order correction by multiplying $g_k(\xi)$ by $[\xi]^q$ in the expansion of the symbol. See Section 2.3.5. Observe that at $q = -2$, the condition numbers $\lambda(\mathcal{B}_A)$ and $\kappa(\mathcal{B}_A)$ are minimized and hardly grow with n.

Let C be the output of the probing and K be the number of Chebyshev polynomials used.

Fix $n = 55^2$, $T = 10$, $\gamma = 2$ and $J = 5$. For $m = 0$ and $K = 3$, i.e., no order correction and using up to quadratic polynomials in ξ, we obtain a relative error $\|C - A\| / \|A\|$ that is less than 10^{-14}. On the other hand, using Fourier expansion, with $K = 5$ in the sense that $-\frac{K-1}{2} \leq k_1, k_2 \leq \frac{K-1}{2}$, the relative error is on the order of 10^{-1}. The point is that in this case, A has an exact "Chebyshev on a disk" representation and probing using the correct $\mathcal{B}$ enables us to retrieve the coefficients with negligible errors.

Finally, we consider backward probing with the Chebyshev expansion. We use $J = 5, \gamma = 2$ and $T = 10$. Figure 2-8 shows that when $m = -2$, the condition numbers $\lambda(\mathcal{B}_A)$ and $\kappa(\mathcal{B}_A)$ are minimized and hardly increases with n. This emphasizes the importance of knowing the order of the operator being probed.

2.4.4 Foveation

In this section, we forward-probe for the foveation operator, a space-variant imaging operator [20], which is particularly interesting as a model for human vision. Formally, we may treat the foveation operator A as a Gaussian blur with a width or standard deviation that varies over space, that is

$$Au(x) = \int_{\mathbb{R}^2} K(x,y)u(y)dy \text{ where } K(x,y) = \frac{1}{w(x)\sqrt{2\pi}} \exp\left(\frac{-\|x-y\|^2}{2w^2(x)}\right), \quad (2.17)$$

where $w(x)$ is the width function which returns only positive real numbers.

The resolution of the output image is highest at the point where $w(x)$ is minimal. Call this point x_0. It is the point of fixation, corresponding to the center of the fovea. For our experiment, the width function takes the form of $w(x) = (\alpha \|x - x_0\|^2 + \beta)^{1/2}$. Our images are 201×201 and treated as functions on the unit square. We choose $x_0 = (0.5, 0.5)$ and $\alpha, \beta > 0$ such that $w(x_0) = 0.003$ and $w(1,1) = 0.012$.

The symbol of A is $a(x,\xi) = \exp(-2\pi^2 w(x)^2 \|\xi\|^2)$, and we choose to use a Fourier series or (2.11) for expanding it. Let C be the output of matrix probing and z be a standard test image. Figure 2-9(c) shows that the relative ℓ^2 error $\|Cz - Az\|_{\ell^2} / \|Az\|_{\ell^2}$ decreases exponentially as p increases. In general, forward probing yields great results like this because we know its symbol well and can choose an appropriate $\mathcal{B}$.

2.4.5 Inverting the wave equation Hessian

In seismology, it is common to recover the model parameters m, which describe the subsurface, by minimizing the least squares misfit between the observed data and $F(m)$ where F, the forward model, predicts data from m.

Methods to solve this problem can be broadly categorized into two classes: steepest descent or Newton's method. The former takes more iterations to converge but each iteration is computationally cheaper. The latter requires the inversion of the Hessian of the objective function, but achieves quadratic convergence near the optimal point.

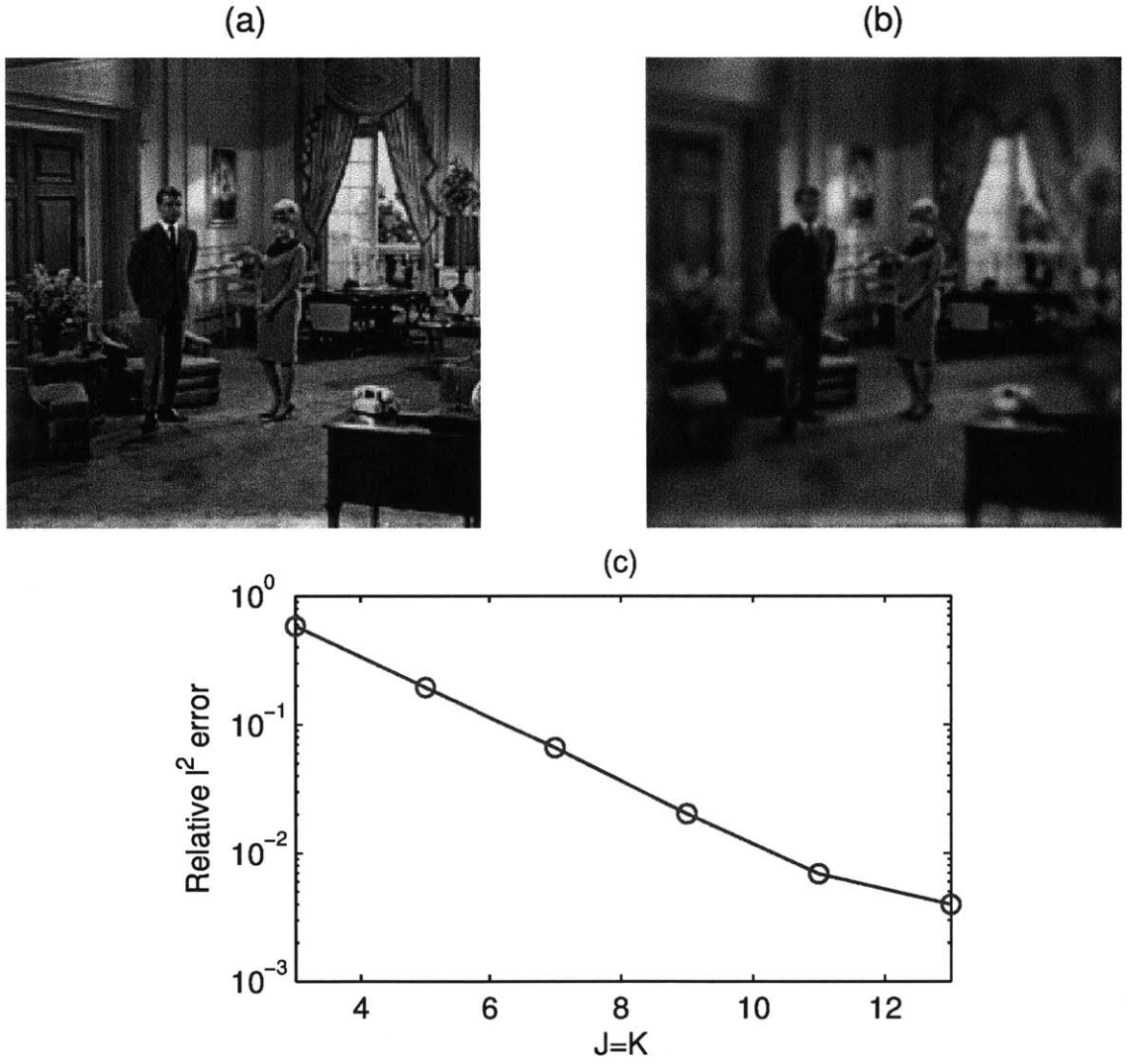


Figure 2-9: Let A be the foveation operator in (2.17) and C be the output of the forward probing of A. Figure (a) is the test image z. Figure (b) is Cz and it shows that C behaves like the foveation operator as expected. Figure (c) shows that the relative ℓ^2 error (see text) decreases exponentially with the number of parameters $p = J^4$.

In another paper, we use matrix probing to precondition the inversion of the Hessian. Removing the nullspace component from the noise vector is more tricky (see Algorithm 2-1) and involves checking whether "a curvelet is visible to any receiver" via raytracing. For details on this more elaborate application, please refer to [23].

2.5 Conclusion and future work

When a matrix A with n columns belongs to a specified p-dimensional subspace, say $A = \sum_{i=1}^{p} c_i B_i$, we can probe it with a few random vectors to recover the coefficient vector c.

Let q be the number of random vectors used, κ be the condition number of the Gram matrix of $B_1, \ldots, B_p$ and λ be the "weak condition number" of each B_i (cf. Definition 2.1.2) which is related to the numerical rank. From Theorem 2.1.3 and Section 2.1.3, we learn that when $nq \propto p(\kappa\lambda \log n)^2$, then the linear system that has to be solved to recover c (cf. (2.1)) will be well-conditioned with high probability. Consequently, the reconstruction error is small by Proposition 2.1.5.

The same technique can be used to compute an approximate A^{-1}, or a preconditioner for inverting A. In [23], we used it to invert the wave equation Hessian — here we demonstrate that it can also be used to invert elliptic operators in smooth media (cf. Sections 2.4.2 and 2.4.3).

Some possible future work include the following.

1. Extend the work of Pfander, Rauhut et. al. [60, 59, 61]. These papers are concerned with sparse signal recovery. They consider the special case where $\mathcal{B}$ contains n^2 matrices each representing a time-frequency shift, but A is an unknown linear combination of only p of them. The task is to identify these p matrices and the associated coefficients by applying A to noise vectors. Our proofs may be used to establish similar recovery results for a more general $\mathcal{B}$. However, note that in [59], Pfander and Rauhut show that $n \propto p \log n$ suffices, whereas our main result requires an additional log factor.

2. Build a framework for probing $f(A)$ interpreted as a Cauchy integral

$$f(A) = \frac{1}{2\pi i} \oint_\Gamma f(z)(zI - A)^{-1} dz,$$

where Γ is a closed curve enclosing the eigenvalues of A. For more on approximating matrix functions, see [38, 42].

3. Consider expansion schemes for symbols that highly oscillate or have singularities that are well-understood.

We conclude the chapter by outlining how better constants (see Remark A.1.3) can be obtained for the Gaussian case. At the start of the proof of Proposition A.1.7, we can split $(\mathbb{E}\,\|M - N\|^s)^{1/s}$ into two parts $\left(\mathbb{E}\left\|\sum_{1\le i\ne j\le n} u_i u_j A_{ij}\right\|^s\right)^{1/s}$ and $\left(\mathbb{E}\,\|\sum_{i=1}^n (u_i^2 - 1)A_{ii}\|^s\right)^{1/s}$. For the first part, decouple using Theorem A.1.1 with $C_2 = 1$, then apply Theorem A.1.5. For the second part, note that every $u_i^2 - 1$ is symmetrically distributed and has zero mean. Thus, we can introduce Rademacher variables, condition on the Gaussians, apply Theorem A.1.4, and pull out the term $\left(\mathbb{E}\max_i |u_i^2 - 1|^s\right)^{1/s}$. Although this log factor is in practice smaller than the constants we have, we prefer to avoid it by decoupling the Gaussian sum right away using [3].

Chapter 3

Sublinear randomized algorithms for skeleton decompositions

3.1 Introduction

3.1.1 Skeleton decompositions

This piece of work is concerned with the decomposition known as the matrix skeleton, pseudo-skeleton [35], or CUR factorization [54, 26].

Throughout this chapter, we adopt the following Matlab-friendly notation. Let R, C be index sets. Given $A \in \mathbb{C}^{m \times n}$, let $A_{:C}$ denote the restriction of A to columns indexed by C, and $A_{R:}$ denote the restriction of A to rows indexed by R. A skeleton decomposition of A is any factorization of the form

$$A_{:C} Z A_{R:} \text{ for some } Z \in \mathbb{C}^{k \times k}.$$

In general, storing a rank-k approximation of A takes up $O((m+n)k)$ space. For skeletons however, only the middle factor Z and the two index sets C and R need to be stored, if we assume that A's entries can be sampled on-demand by an external function. Hence specifying the skeleton decomposition of A only requires $O(k^2)$ space. In addition, row and columns from the original matrix may carry more physical

significance than their linear combinations.

There are important examples where the full matrix itself is not low rank but can be partitioned into blocks each of which has low numerical rank. One example is the Green's function of elliptic operators with mild regularity conditions [7]. Another example is the amplitude factor in certain Fourier integral operators and wave propagators [16, 25]. Algorithms that compute good skeleton representations can be used to manipulate such matrices.

3.1.2 Overview

Our work mostly treats the case of skeleton decompositions with C and R drawn uniformly at random. Denote by A_{RC} the restriction of A to rows in R and columns in C: we compute the middle matrix Z as the pseudoinverse of A_{RC} with some amount of regularization. Algorithm 1 below details the form of this regularization.

Throughout the chapter, we use the letter k to denote the baseline small dimension of the factorization: it is either exactly the rank of A, or, more generally, it is the index of the singular value σ_k that governs the approximation error of the skeleton decomposition. The small dimension of the skeleton decomposition may or may not be k: for instance, Algorithm 1 requires a small oversampling since $\ell = O(k \log \max(m, n))$. Later, we consider two algorithms where ℓ is exactly k.

The situation in which Algorithm 1 works is when A is a priori known to have a factorization of the form $\simeq X_1 A_{11} Y_1^*$ where X_1, Y_1 have k orthonormal columns, and these columns are *incoherent*, or spread, in the sense that their uniform norm is about as small as their normalization allows. In this scenario, our main result in Theorem 3.1.2 states that that the output of Algorithm 1 obeys

$$\|A - A_{:C} Z A_{R:}\| = O\left(\|A - X_1 A_{11} Y_1^*\| \frac{(mn)^{1/2}}{\ell}\right)$$

with high probability, for some adequate choice of the regularization parameter δ.

The drawback of Algorithm 1 is that it requires to set an appropriate regularization parameter in advance. Unfortunately, there is no known way of estimating it fast,

Algorithm 1. $\widetilde{O}(k^3)$-time algorithm where $\widetilde{O}$ is the O notation with log factors dropped.
Input: A matrix $A \in \mathbb{C}^{m \times n}$ that is approximately rank k, and user-defined parameters $\ell = \widetilde{O}(k)$ and δ.
Output: Column index set C of size ℓ, row index set R of size ℓ, center matrix of a matrix skeleton Z. Implicitly, we have the matrix skeleton $A_{:C} Z A_{R:}$.
Steps:

1. Let C be a random index set of size ℓ chosen uniformly from $\{1, \ldots, n\}$. Implicitly, we have $A_{:C}$.
2. Let R be a random index set of size ℓ chosen uniformly from $\{1, \ldots, m\}$. Implicitly, we have $A_{R:}$.
3. Sample A_{RC}, the intersection of $A_{:C}$ and $A_{R:}$.
4. Compute the thin SVD of A_{RC} as $U_1 \Sigma_1 V_1^* + U_2 \Sigma_2 V_2^*$ where Σ_1, Σ_2 are diagonal, Σ_1 contains singular values $\geq \delta$ and Σ_2 contains singular values $< \delta$.
5. Compute $Z = V_1 \Sigma_1^{-1} U_1^*$.

Matlab code:

```
function [C,Z,R]=skeleton1(A,l,delta)
    C=randperm(n,l); R=randperm(m,l); Z=pinv(A(R,C),delta);
end
```

and this regularization step cannot be skipped. In Section 3.4.1, we illustrate with a numerical example that without the regularization, the error in the operator norm can blow up in a way predicted by our main result or Theorem 3.1.2.

Finally, we use our proof framework to establish error estimates for two other algorithms. The goal of these algorithms is to further reduce the small dimension of the skeleton decomposition to exactly k (instead of $\ell = O(k \log \max(m, n))$), with σ_k still providing control over the approximation error. The proposed methods still run in sublinear-time complexity; they use well-known strong rank-revealing QR factorizations applied *after* some amount of pruning via uniform random sampling of rows and columns. This combination of existing ideas is an important part of the discussion of how skeleton factorizations can be computed reliably without visiting all the elements of the original matrix.

3.1.3 Related work

The idea of uniformly sampling rows and columns to build a matrix skeleton is not new. In particular, for the case where A is symmetric, this technique is known as the Nyström method[1]. The skeleton used is $A_{:C}A_{CC}^{+}A_{C:}$, which is symmetric, and the error in the operator norm was recently analyzed by Talwalkar [71] and Gittens [34]. Both papers make the assumption that X_1, Y_1 are incoherent. Gittens obtained relative error bounds that are similar to ours.

Nonetheless, our results are more general. They apply to nonsymmetric matrices that are low rank in a broader sense. Specifically, when we write $A \simeq X_1 A_{11} Y_1^*$, A_{11} is not necessarily diagonal and X_1, Y_1 are not necessarily the singular vectors of A. This relaxes the incoherence requirement on X_1, Y_1. Furthermore, in the physical sciences, it is not uncommon to work with linear operators that are known a priori to be almost (but not fully) diagonalized by the Fourier basis or related bases in harmonic analysis. These bases are often incoherent. One example is an integral operator with a smooth kernel. See Section 3.4 for more details.

[1] In machine learning, the Nystrom method can be used to approximate kernel matrices of support vector machines, or the Laplacian of affinity graphs, for instance.

A factorization that is closely related to matrix skeletons is the interpolative decomposition [21], also called the column subset selection problem [30] or a Rank Revealing QR (RRQR) [17, 36]. An interpolative decomposition of A is the factorization $A_{:C}D$ for some D. It is relevant to our work because algorithms that compute interpolative decompositions can be used to compute matrix skeletons [51]. Algorithms 2 and 3, discussed below, require the computation of interpolative decompositions.

One of the earliest theoretical results concerning matrix skeletons is due to Goreinov et al. [35]. In that paper, it is shown that for any $A \in \mathbb{C}^{m \times n}$, there exists a skeleton $A_{:C}ZA_{R:}$ such that in the operator norm, $\|A - A_{:C}ZA_{R:}\| = O(\sqrt{k}(\sqrt{m} + \sqrt{n})\sigma_{k+1}(A))$. Although the proof is constructive, it requires computing the SVD of A, which takes much more time and space than the algorithms considered in this work. A useful idea in [35] for selecting C and R is to maximize the volume or determinant of submatrices. This idea may date back to interpolating projections [63] and the proof of Auerbach's theorem [68].

A popular method of computing matrix skeletons is cross-approximation. The idea is to iteratively select good rows and columns based on the residual matrix. As processing the entire residual matrix is not practical, there are faster variants that operate on only a small part of the residual, e.g., Adaptive Cross Approximation [5, 6] and Incomplete Cross Approximation [75]. The algorithms considered in this work are non-iterative, arguably easier to implement and analyze, yet possibly less efficient for some applications.

In this work, we compute a matrix skeleton by randomly sampling rows and columns of A. This idea dates back at least to the work of Frieze, Kannan and Vempala [30]. One way of sampling rows and columns of A is called "subspace sampling" [54, 26] by Drineas et al. If we assume that the top k singular vectors of A are incoherent, then a result due to Rudelson, Vershynin [66] implies that *uniform sampling* of rows and columns, a special case of "subspace sampling", will produce a good skeleton representation $A_{:C}(A^{+}_{:C}AA^{+}_{R:})A_{R:}$. However, it is not clear how the middle matrix $A^{+}_{:C}AA^{+}_{R:}$ can be computed in sublinear time.

In the main algorithm analyzed in this work, we *uniformly sample* rows and

columns to produce a skeleton of the form $A_{:C}A_{RC}^{+}A_{R:}$, not $A_{:C}(A_{:C}^{+}AA_{R:}^{+})A_{R:}$. One major difference is that the skeleton $A_{:C}A_{RC}^{+}A_{R:}$ can be computed in $\widetilde{O}(k^3)$ time[2]. Note that the matrix skeleton output by our algorithms is represented by the index sets R, C and matrix Z, not $A_{R:}, A_{:C}$.

Finally, let us mention that the term "skeleton" may refer to other factorizations. Instead of $A \simeq A_{C:}ZA_{:R}$, we can have $A \simeq Z_1 A_{RC} Z_2$ where Z_1, Z_2 are arbitrary $m \times k$ and $k \times n$ matrices [21]. As $O(mk + nk)$ space is needed to store Z_1, Z_2, this representation does not seem as appealing in memory-critical situations where $A_{C:}ZA_{:R}$ is. Nevertheless, it is numerically more stable and has found several applications [43].

Alternatively, when $A = MBN$ where M, B, N are $n \times n$ matrices, we can approximate M as $M_{:C}P$, N as $DN_{R:}$, where M_C has k columns of M and N_R has k rows of N. Thus, $A \simeq M_C(PBD)N_R$, effectively replacing B with the $k \times k$ matrix $\widetilde{B} := PBD$. Bremer calls $\widetilde{B}$ a skeleton and uses it to approximate scattering matrices [11].

3.1.4 Notations

The matrices we consider take the form

$$A = \begin{pmatrix} X_1 & X_2 \end{pmatrix} \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} Y_1^* \\ Y_2^* \end{pmatrix}, \tag{3.1}$$

where $X = \begin{pmatrix} X_1 & X_2 \end{pmatrix}$ and $Y = \begin{pmatrix} Y_1 & Y_2 \end{pmatrix}$ are unitary matrices, with columns being "*spread*", and the blocks A_{12}, A_{21} and A_{22} are in some sense *small*. By "spread", we mean $\widetilde{O}(1)$-coherent.

Definition 3.1.1. *Let $X \in \mathbb{C}^{n \times k}$ be a matrix with k orthonormal columns. Denote $\|X\|_{\max} = \max_{ij} |X_{ij}|$. We say X is μ-coherent if $\|X\|_{\max} \leq (\mu/n)^{1/2}$.*

This notion is well-known in compressed sensing [14] and matrix completion [13, 57].

[2]Note that $\widetilde{O}$ is the O notation with log factors dropped.

Formally, let $\Delta_k := \begin{pmatrix} 0 & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$, and consider that

$$\varepsilon_k := \|\Delta_k\| \text{ is small.} \tag{3.2}$$

That means A can be represented using only $O(k^2)$ data if we allow an ε_k error in the operator norm. Note that ε_k is equivalent to $\max(\|X_2^* A\|, \|AY_2\|)$ up to constants. To prevent clutter, we have suppressed the dependence on k from the definitions of X_1, Y_1, A_{11}, A_{12} etc.

If (3.1) is the SVD of A, then $\varepsilon_k = \sigma_{k+1}(A)$. It is good to keep this example in mind as it simplifies many formulas that we see later.

An alternative to ε_k is

$$\varepsilon'_k := \sum_{i=1}^{m} \sum_{j=1}^{n} |(\Delta_k)_{ij}| . \tag{3.3}$$

In other words, ε'_k is the ℓ^1 norm of Δ_k reshaped into a vector. We know $\varepsilon_k \leq \varepsilon'_k \leq mn\varepsilon_k$. The reason for introducing ε'_k is that it is common for $(\Delta_k)_{ij}$ to decay rapidly such that $\varepsilon'_k \ll mn\varepsilon_k$. For such scenarios, the error guarantee of Algorithm 1 is much stronger in terms of ε'_k than in terms of ε_k as we will see in the next section.

3.1.5 Main result

Random subsets of rows and columns are only representative of the subspaces of the matrix A under the incoherence assumption mentioned earlier, otherwise Algorithm 1 may fail. For example, if $A = X_1 A_{11} Y_1^*$ and $X_1 = \binom{I_{k\times k}}{0}$, then $A_{R:}$ is going to be zero most of the time, and so is $A_{:C} Z A_{R:}$. Hence, it makes sense that we want $X_{1,R:}$ to be "as nonsingular as possible" so that little information is lost. In particular, it is well-known that if X_1, Y_1 are $\widetilde{O}(1)$-coherent, i.e., *spread*, then sampling $\ell = \widetilde{O}(k)$ rows will lead to $X_{1,R:}, Y_{1,C:}$ being well-conditioned[3].

[3]Assume $\ell = \widetilde{O}(k)$. Then $\|Y_1\|_{\max} = \widetilde{O}(n^{-1/2})$ is a sufficient condition for $Y_{1,C:}$ to be well-conditioned with high probability. This condition can be relaxed in at least two ways. First, all we need is that for each row i, $(\sum_j |(Y_1)_{ij}|^2)^{1/2} \leq (\mu k/n)^{1/2}$. This would allow a few entries of each row of Y_1 to be bigger than $O(n^{-1/2})$. Second, we can allow a few rows of Y to violate the previous condition [4].

Here is our main result. It is proved in Section 3.2.

Theorem 3.1.2. *Let A be given by (3.1) for some $k > 0$. Assume $m \geq n$ and X_1, Y_1 are μ-coherent where $\mu = \widetilde{O}(1)$ with respect to m, n. Recall the definitions of $\varepsilon_k, \varepsilon'_k$ in (3.2) and (3.3). Let $\ell \geq 10\mu k \log m$ and $\lambda = \frac{(mn)^{1/2}}{\ell}$. Then with probability at least $1 - 4km^{-2}$, Algorithm 1 returns a skeleton that satisfies*

$$\|A - A_{:C} Z A_{R:}\| = O(\lambda\delta + \lambda\varepsilon_k + \varepsilon_k^2 \lambda/\delta). \tag{3.4}$$

If furthermore the entire X and Y are μ-coherent, then with probability at least $1 - 4m^{-1}$,

$$\|A - A_{:C} Z A_{R:}\| = O(\lambda\delta + \varepsilon'_k + {\varepsilon'_k}^2/(\lambda\delta)). \tag{3.5}$$

The right hand sides of (3.4) and (3.5) can be minimized with respect to δ. For (3.4), pick $\delta = \Theta(\varepsilon_k)$ so that

$$\|A - A_{:C} Z A_{R:}\| = O(\varepsilon_k \lambda) = O(\varepsilon_k (mn)^{1/2}/\ell). \tag{3.6}$$

For (3.5), pick $\delta = \Theta(\varepsilon'_k/\lambda)$ so that

$$\|A - A_{:C} Z A_{R:}\| = O(\varepsilon'_k). \tag{3.7}$$

Here are some possible scenarios where $\varepsilon'_k = o(\varepsilon_k \lambda)$ and (3.7) is strictly stronger than (3.6):

- The entries of Δ_k *decay exponentially* or there are only $O(1)$ nonzero entries as m, n increases. Then $\varepsilon'_k = \Theta(\varepsilon_k)$.
- Say $n = m$ and (3.1) is the SVD of A. Suppose the singular values decay as $m^{-1/2}$. Then $\varepsilon'_k = O(\varepsilon_k m^{1/2})$.

One important question remains: how can we guess ε_k, in order to then chose δ? Unfortunately, we are not aware of any $\widetilde{O}(k^3)$ algorithm that can accurately estimate ε_k. Here is one possible *heuristic* for choosing δ for the case where (3.1) is the SVD.

Imagine $A_{RC} \simeq X_{1,R:} A_{11} Y_{1,C:}^*$. As we will see, the singular values of $X_{1,R:}, Y_{1,C:}$ are likely to be on the order of $(\ell/m)^{1/2}, (\ell/n)^{1/2}$. Therefore, it is not unreasonable to view $\varepsilon_k \simeq \sigma_{k+1}(A) \simeq \lambda \sigma_{k+1}(A_{RC})$.

Another approach is to begin with a big δ, run the $\widetilde{O}(k^3)$ algorithm, check $\|A - A_{:C} Z A_{R:}\|$, divide δ by two and repeat the whole process until the error does not improve. However, calculating $\|A - A_{:C} Z A_{R:}\|$ is expensive and other tricks are needed. This seems to be an open problem.

The $\widetilde{O}(k^3)$ algorithm is among the fastest algorithms for computing skeleton representations that one can expect to have. With more work, can the accuracy be improved? In Section 3.3, we sketch two such algorithms. These two algorithms have for the most part been analyzed in previous work: though their ability to perform in sublinear-time complexity was not explicitly stated in those references, this fact should not come as a surprise. The first algorithm samples $\ell \simeq k \log m$ rows, columns, then reduce it to *exactly* k rows, columns using the a rank-revealing QR decomposition (RRQR), with an operator norm error of $O(\varepsilon_k (mk)^{1/2})$. It is similar to what is done in [9]. In the second algorithm, we uniformly sample $\ell \simeq k \log m$ rows to get $A_{R:}$, then run RRQR on $A_{R:}$ to select k columns of A. The overall error is $O(\varepsilon_k (mn)^{1/2})$. This is similar to the algorithm proposed by Tygert, Rokhlin et al. [51, 77].

Using the proof framework in Section 3.2, we will derive error estimates for the above two algorithms. As mentioned earlier these error guarantees are not new, but (i) they concern provable sublinear-time complexity algorithms, (ii) they work for a more general model (3.1), and (iii) our proofs are also motivated differently. In Section 3.3.3, we compare these three algorithms.

3.1.6 More on incoherence

If either X or Y is not $\widetilde{O}(1)$-coherent, we can use the idea of a randomized Fourier transform [1] to impose incoherence. The idea is to multiply them on the left by the unitary Fourier matrix with randomly rescaled columns. This has the effect of "blending up" the rows of X, Y — at the possible cost of requiring linear-time complexity. The following is a standard result that can be proved using Hoeffding's

inequality.

Proposition 3.1.3. *Let $X \in \mathbb{C}^{n\times k}$ with orthonormal columns. Let $D = \mathrm{diag}(d_1, \ldots, d_n)$ where $d_1, \ldots, d_n$ are independent random variables such that $\mathbb{E}\, d_i = 0$ and $|d_i| = 1$. Let $\mathcal{F}$ be the unitary Fourier matrix and $\mu = \alpha \log n$ for some $\alpha > 0$. Define $U := \mathcal{F}DX$. Then $\|U\|_{\max} \leq (\mu/n)^{1/2}$ with probability at least $1 - 2(nk)n^{-2\alpha}$*

In other words, no matter what X is, $U = \mathcal{F}DX$ would be $\widetilde{O}(1)$-coherent with high probability. Hence, we can write a wrapper around Algorithm 1. Call this Algorithm 1'. Let $\mathcal{F} \in \mathbb{C}^{n\times n}$ and $\mathcal{F}' \in \mathbb{C}^{m\times m}$ be unitary Fourier transforms.

1. Let $B := \mathcal{F}' D_2 A D_1 \mathcal{F}^*$ where D_1, D_2 are diagonal matrices with independent entries that are ± 1 with equal probability.

2. Feed B to the $\widetilde{O}(k^3)$ algorithm and obtain $B \simeq B_{:C} Z B_{R:}$.

3. It follows that $A \simeq (AD_1\mathcal{F}^*_{C:})Z(\mathcal{F}_{R:}D_2A)$.

The output $(AD_1\mathcal{F}^*_{C:})Z(\mathcal{F}_{R:}D_2A)$ is not a matrix skeleton, but the amount of space needed is $O(n) + \widetilde{O}(k^2)$ which is still better than $O(nk)$. Note that we are not storing $AD_1\mathcal{F}^*_{C:}$ just as we do not store $A_{:C}$ in Algorithm 1. Let T_A be the cost of matrix-vector multiplication of A. See that Algorithm 1' runs in $\widetilde{O}(T_A k + mk + k^3)$ time. The most expensive step is computing B_{RC} and it can be carried out as follows.

Compute $D_1(\mathcal{F}^* S_C)$ in $\widetilde{O}(nk)$ time. Multiply the result by A on the left in $\widetilde{O}(T_A k)$ time. Multiply the result by D_2 on the left in $\widetilde{O}(mk)$ time. Multiply the result by $\mathcal{F}'$ on the left in $\widetilde{O}(mk)$ time using FFT. Multiply the result by S_R^T on the left in $\widetilde{O}(k^2)$ time.

3.2 Error estimates for $\widetilde{O}(k^3)$ algorithm

3.2.1 Notation

$S_C, S_R \in \mathbb{R}^{n\times k}$ are both *column* selector matrices. They are column subsets of permutation matrices. The subscripts "R :" and ": C" denote a row subset and a column

subset respectively, e.g., $A_{R:} = S_R^T A$ and $A_{:C} = AS_C$, while A_{RC} is a row and column subset of A. Transposes and pseudoinverses are taken after the subscripts, e.g., $A_{R:}^*$ means $(A_{R:})^*$.

3.2.2 Two principles

Our proofs are built on two principles. The first principle is due to Rudelson [66] in 1999. Intuitively, it says the following.

> Let Y be a $n \times k$ matrix with orthonormal columns. Let $Y_{C:}$ be a *random* row subset of Y. Suppose Y is μ-coherent with $\mu = \widetilde{O}(1)$, and $|C| = \ell \gtrsim \mu k$. Then with high probability, $(\frac{n}{\ell})^{1/2} Y_{C:}$ is like an isometry.

To be precise, we quote [73, Lemma 3.4]. Note that their M is our μk.

Theorem 3.2.1. *Let $Y \in \mathbb{C}^{n \times k}$ with orthonormal columns. Suppose Y is μ-coherent and $\ell \geq \alpha k \mu$ for some $\alpha > 0$. Let $Y_{C:}$ be a random ℓ-row subset of Y. Each row of $Y_{C:}$ is sampled independently, uniformly. Then*

$$\mathbb{P}\left(\left\| Y_{C:}^+ \right\| \geq \sqrt{\frac{n}{(1-\delta)\ell}} \right) \leq k \left(\frac{e^{-\delta}}{(1-\delta)^{1-\delta}} \right)^{\alpha} \text{ for any } \delta \in [0, 1)$$

and

$$\mathbb{P}\left(\|Y_{C:}\| \geq \sqrt{\frac{(1+\delta')\ell}{n}} \right) \leq k \left(\frac{e^{\delta'}}{(1+\delta')^{1+\delta'}} \right)^{\alpha} \text{ for any } \delta' \geq 0.$$

To be concrete, if $\delta = 0.57$ and $\delta' = 0.709$ and $\ell \geq 10 k\mu \log n$, then

$$\mathbb{P}\left(\left\| Y_{C:}^+ \right\| \leq 1.53(n/\ell)^{1/2} \text{ and } \|Y_{C:}\| \leq 1.31(\ell/n)^{1/2} \right) \geq 1 - 2kn^{-2}. \tag{3.8}$$

We will use (3.8) later. Let us proceed to the second principle, which says

> Let C be an arbitrary index set. If $\|A_{:C}\|$ is small , then $\|A\|$ is also small, provided that we have control over $\|AY_2\|$ and $\left\| Y_{1,C:}^+ \right\|$ for some unitary matrix $(Y_1 \quad Y_2)$.

The roadmap is as follows. If we ignore the regularization step, then what we want to show is that $A \simeq A_{:C}A_{RC}^{+}A_{R:}$. But when we take row and column restrictions on both sides, we have trivially $A_{RC} = A_{RC}A_{RC}^{+}A_{RC}$. Hence, we desire a mechanism to go backwards, that is to infer that "$E := A - A_{:C}A_{RC}^{+}A_{R:}$ is small" from "E_{RC} is small." We begin by inferring that "E is small" from "$E_{:C}$ is small".

Lemma 3.2.2. *Let $A \in \mathbb{C}^{m\times n}$ and $Y = \begin{pmatrix} Y_1 & Y_2 \end{pmatrix} \in \mathbb{C}^{n\times n}$ be a unitary matrix such that Y_1 has k columns. Select $\ell \geq k$ rows of Y_1 to form $Y_{1,C:} = S_C^T Y_1 \in \mathbb{C}^{\ell\times k}$. Assume $Y_{1,C:}$ has full column rank. Then*

$$\|A\| \leq \left\|Y_{1,C:}^{+}\right\| \|A_{:C}\| + \left\|Y_{1,C:}^{+}\right\| \left\|AY_2Y_{2,C:}^{*}\right\| + \|AY_2\| .$$

Proof. Note that $Y_{1,C:}^{*}Y_{1,C:}^{*+} = I_{k\times k}$. Now,

$$\begin{aligned}
\|A\| &\leq \|AY_1\| + \|AY_2\| \\
&= \left\|AY_1Y_{1,C:}^{*}Y_{1,C:}^{*+}\right\| + \|AY_2\| \\
&\leq \|AY_1Y_1^{*}S_C\| \left\|Y_{1,C:}^{+}\right\| + \|AY_2\| \\
&\leq \|(A - AY_2Y_2^{*})S_C\| \left\|Y_{1,C:}^{+}\right\| + \|AY_2\| \\
&\leq \|A_{:C}\| \left\|Y_{1,C:}^{+}\right\| + \left\|AY_2Y_{2,C:}^{*}\right\| \left\|Y_{1,C:}^{+}\right\| + \|AY_2\| .
\end{aligned}$$

□

Lemma 3.2.2 can be extended in two obvious ways. First, we can deduce that "A is small if $A_{R:}$ is small." Second, we can deduce that "A is small if A_{RC} is small." This is what the next lemma establishes. (Although its form looks unduly complicated, control over all the terms is *in fine* necessary.)

Lemma 3.2.3. *Let $A \in \mathbb{C}^{m\times n}$ and $X = \begin{pmatrix} X_1 & X_2 \end{pmatrix} \in \mathbb{C}^{m\times m}$ and $Y = \begin{pmatrix} Y_1 & Y_2 \end{pmatrix} \in \mathbb{C}^{n\times n}$ be unitary matrices such that X_1, Y_1 each has k columns. Select $\ell \geq k$ rows and columns indexed by R, C respectively. Assume $X_{1,R:}, Y_{1,C:}$ have full column rank. Then*

$$\|A\| \leq \left\|X_{1,R:}^{+}\right\| \|A_{R:}\| + \left\|X_{1,R:}^{+}\right\| \|X_{2,R:}X_2^{*}A\| + \|X_2^{*}A\|$$

and

$$\begin{aligned}\|A\| \leq & \left\|X_{1,R:}^{+}\right\| \left\|Y_{1,C:}^{+}\right\| \|A_{RC}\| + \\ & \left\|X_{1,R:}^{+}\right\| \left\|Y_{1,C:}^{+}\right\| \left\|X_{2,R:}X_2^*AY_1Y_{1,C:}^*\right\| + \\ & \left\|X_{1,R:}^{+}\right\| \left\|Y_{1,C:}^{+}\right\| \left\|X_{1,R:}X_1^*AY_2Y_{2,C:}^*\right\| + \\ & \left\|X_{1,R:}^{+}\right\| \left\|Y_{1,C:}^{+}\right\| \left\|X_{2,R:}X_2^*AY_2Y_{2,C:}^*\right\| + \\ & \left\|X_{1,R:}^{+}\right\| \|X_{1,R:}X_1^*AY_2\| + \\ & \left\|Y_{1,C:}^{+}\right\| \left\|X_2^*AY_1Y_{1,C:}^*\right\| + \\ & \|X_2^*AY_2\|.\end{aligned}$$

Proof. The top inequality is obtained by applying Lemma 3.2.2 to A^*. The proof of the bottom inequality is similar to the proof of Lemma 3.2.2. For completeness,

$$\begin{aligned}\|A\| &\leq \|X_1^*AY_1\| + \|X_1^*AY_2\| + \|X_2^*AY_1\| + \|X_2^*AY_2\| \\ &= \left\|X_{1,R:}^{+}X_{1,R:}X_1^*AY_1Y_{1,C:}^*Y_{1,C:}^{*+}\right\| + \\ &\quad \left\|X_{1,R:}^{+}X_{1,R:}X_1^*AY_2\right\| + \left\|X_2^*AY_1Y_{1,C:}^*Y_{1,C:}^{*+}\right\| + \|X_2^*AY_2\| \\ &\leq \left\|X_{1,R:}^{+}\right\| \left\|S_R^TX_1X_1^*AY_1Y_1^*S_C\right\| \left\|Y_{1,C:}^{*+}\right\| + \\ &\quad \left\|X_{1,R:}^{+}\right\| \|X_{1,R:}X_1^*AY_2\| + \left\|X_2^*AY_1Y_{1,C:}^*\right\| \left\|Y_{1,C:}^{*+}\right\| + \|X_2^*AY_2\| \\ &= \left\|X_{1,R:}^{+}\right\| \left\|Y_{1,C:}^{+}\right\| \left\|S_R^T(A - X_2X_2^*AY_1Y_1^* - X_1X_1^*AY_2Y_2^* - X_2X_2^*AY_2Y_2^*)S_C\right\| + \\ &\quad \left\|X_{1,R:}^{+}\right\| \|X_{1,R:}X_1^*AY_2\| + \left\|Y_{1,C:}^{+}\right\| \left\|X_2^*AY_1Y_{1,C:}^*\right\| + \|X_2^*AY_2\|.\end{aligned}$$

Split up the term $\left\|S_R^T(A - X_2X_2^*AY_1Y_1^* - X_1X_1^*AY_2Y_2^* - X_2X_2^*AY_2Y_2^*)S_C\right\|$ by the triangle inequality and we are done. □

We conclude this section with a useful corollary. It says that if $PA_{:C}$ is a good low rank approximation of $A_{:C}$ for some $P \in \mathbb{C}^{m\times m}$, then PA may also be a good low rank approximation of A.

Corollary 3.2.4. *Let $A \in \mathbb{C}^{m\times n}$ and $P \in \mathbb{C}^{m\times m}$. Let $Y = \begin{pmatrix} Y_1 & Y_2 \end{pmatrix}$ be a unitary matrix such that Y_1 has k columns. Let $Y_{1,C:} = S_C^TY_1 \in \mathbb{C}^{\ell\times k}$ where $\ell \geq k$. Assume*

$Y_{1,C:}$ has full column rank. Let $I \in \mathbb{C}^{m\times m}$ be the identity. Then

$$\|A - PA\| \leq \left\|Y_{1,C:}^{+}\right\| \|A_{:C} - PA_{:C}\| + \left\|Y_{1,C:}^{+}\right\| \|I - P\| \left\|AY_2Y_{2,C:}^{*}\right\| + \|I - P\| \|AY_2\| .$$

In particular, if P is the orthogonal projection $A_{:C}A_{:C}^{+}$, then

$$\left\|A - A_{:C}A_{:C}^{+}A\right\| \leq \left\|Y_{1,C:}^{+}\right\| \left\|AY_2Y_{2,C:}^{*}\right\| + \|AY_2\| . \tag{3.9}$$

Proof. To get the first inequality, apply Lemma 3.2.2 to $A - PA$. The second inequality is immediate from the first inequality since $\left\|A_{:C} - A_{:C}A_{:C}^{+}A_{:C}\right\| = 0$. □

For the special case where X, Y are singular vectors of A, (3.9) can be proved using the fact that $\left\|A - A_{:C}A_{:C}^{+}A\right\| = \min_D \|A - A_{:C}D\|$ and choosing an appropriate D. See Boutsidis et al. [9].

Note that (3.9) can be strengthened to $\left\|A - A_{:C}A_{:C}^{+}A\right\|^2 \leq \left\|AY_2Y_{2,C:}^{*}Y_{1,C:}^{+}\right\|^2 + \|AY_2\|^2$, by modifying the first step of the proof of Lemma 3.2.2 from $\|A\| \leq \|AY_1\| + \|AY_2\|$ to $\|A\|^2 \leq \|AY_1\|^2 + \|AY_2\|^2$. A similar result for the case where X, Y are singular vectors can be found in Halko et al. [39]. The originality of our results is that they hold for a more general model (3.1).

3.2.3 Proof of Theorem 3.1.2

The proof is split into two main parts. The first part is probabilistic. We will apply the first principle to control the largest and smallest singular values of $Y_{1,C:}, X_{1,R:}$ and other similar quantities. The second part, mainly linear algebra, uses these bounds on $Y_{1,C:}$ and $X_{1,R:}$ to help control the error $\left\|A - A_{:C}B_{RC}^{+}A_{R:}\right\|$.

Probabilistic part

Let $\lambda_X = (\frac{m}{\ell})^{1/2}$ and $\lambda_Y = (\frac{n}{\ell})^{1/2}$. To prove the first part of Theorem 3.1.2, i.e., (3.4), we apply Theorem 3.2.1. From (3.8), it is clear that the assumptions of Theorem 3.1.2 guarantee that $\|Y_{1,C:}\| = O(\lambda_Y^{-1})$, $\left\|Y_{1,C:}^{+}\right\| = O(\lambda_Y)$, $\|X_{1,R:}\| = O(\lambda_X^{-1})$, $\left\|X_{1,R:}^{+}\right\| = O(\lambda_X)$ hold simultaneously with probability at least $1 - 4km^{-2}$.

For the second part of Theorem 3.1.2, i.e., (3.5), we need to refine (3.1) as follows. Let $X = (\widetilde{X}_1, \ldots, \widetilde{X}_{\lceil m/k \rceil})$ and $Y = (\widetilde{Y}_1, \ldots, \widetilde{Y}_{\lceil n/k \rceil})$ where $\widetilde{X}_1, \ldots, \widetilde{X}_{\lceil m/k \rceil - 1}$ and $\widetilde{Y}_1, \ldots, \widetilde{Y}_{\lceil n/k \rceil - 1}$ has k columns, and $\widetilde{X}_{\lceil m/k \rceil}, \widetilde{Y}_{\lceil n/k \rceil}$ have $\leq k$ columns. Note that $\widetilde{X}_1 = X_1$, $\widetilde{Y}_1 = Y_1$, $\widetilde{A}_{11} = A_{11}$ where X_1, Y_1, A_{11} are defined in (3.1). Rewrite (3.1) as

$$A = (\widetilde{X}_1, \ldots, \widetilde{X}_{\lceil m/k \rceil}) \begin{pmatrix} \widetilde{A}_{11} & \cdots & \widetilde{A}_{1,\lceil n/k \rceil} \\ \vdots & \ddots & \vdots \\ \widetilde{A}_{\lceil m/k \rceil,1} & \cdots & \widetilde{A}_{\lceil m/k \rceil,\lceil n/k \rceil} \end{pmatrix} \begin{pmatrix} \widetilde{Y}_1^* \\ \vdots \\ \widetilde{Y}_{\lceil n/k \rceil}^* \end{pmatrix}. \tag{3.10}$$

By applying Theorem 3.2.1 to every $\widetilde{X}_i$, $\widetilde{Y}_j$ and doing a union bound, we see that with probability at least $1 - 4m^{-1}$, we will have $\|Y_{j,C:}\| = O(\lambda_Y^{-1})$, $\left\|Y_{j,C:}^+\right\| = O(\lambda_Y)$, $\|X_{i,R:}\| = O(\lambda_X^{-1})$, $\left\|X_{i,R:}^+\right\| = O(\lambda_X)$ for all i, j.

Deterministic part: Introducing B, an auxillary matrix

Recall that in Algorithm 1, we compute the SVD of A_{RC} as $U_1\Sigma_1V_1^* + U_2\Sigma_2V_2^*$ and invert only $U_1\Sigma_1V_1^*$ to get the center matrix Z.

Define $B \in \mathbb{C}^{m \times n}$ such that $B_{RC} = U_1\Sigma_1V_1^*$ and all other entries of B are the same as A's. In other wrods, define $E \in \mathbb{C}^{m \times n}$ such that $E_{RC} = -U_2\Sigma V_2^*$ and all other entries of E are zeros, then let $B = A + E$.

The skeleton returned is $A_{C:}B_{RC}^+A_{R:}$. By construction,

$$\|A - B\| \leq \delta; \quad \left\|B_{RC}^+\right\| \leq \delta^{-1}.$$

Our objective is to bound $\left\|A - A_{:C}B_{RC}^+A_{R:}\right\|$, but it is $\left\|B - B_{:C}B_{RC}^+B_{R:}\right\|$ that we have control over by the second principle. Recall that $B_{RC} = B_{RC}B_{RC}^+B_{RC}$ is to be lifted to $B \simeq B_{:C}B_{RC}^+B_{R:}$ by Lemma 3.2.3. Thus, we shall first relate

$\left\|A - A_{:C}B_{RC}^{+}A_{R:}\right\|$ to quantities involving *only* B by a perturbation argument.

$$\begin{aligned}
\left\|A - A_{:C}B_{RC}^{+}A_{R:}\right\| \leq & \|A - B\| + \left\|B - B_{:C}B_{RC}^{+}B_{R:}\right\| + \\
& \left\|B_{:C}B_{RC}^{+}B_{R:} - A_{:C}B_{RC}^{+}B_{R:}\right\| + \left\|A_{:C}B_{RC}^{+}B_{R:} - A_{:C}B_{RC}^{+}A_{R:}\right\| \\
\leq & \delta + \left\|B - B_{:C}B_{RC}^{+}B_{R:}\right\| + \\
& \|(B-A)S_C\| \left\|B_{RC}^{+}B_{R:}\right\| + \left\|A_{:C}B_{RC}^{+}\right\| \left\|S_R^T(B-A)\right\| \\
\leq & \delta + \left\|B - B_{:C}B_{RC}^{+}B_{R:}\right\| + \\
& \delta \left\|B_{RC}^{+}B_{R:}\right\| + (\left\|B_{:C}B_{RC}^{+}\right\| + \left\|A_{:C}B_{RC}^{+} - B_{:C}B_{RC}^{+}\right\|)\delta \\
\leq & \delta + \left\|B - B_{:C}B_{RC}^{+}B_{R:}\right\| + \\
& \delta \left\|B_{RC}^{+}B_{R:}\right\| + \delta \left\|B_{:C}B_{RC}^{+}\right\| + \|(A-B)S_C\| \, \delta^{-1}\delta \\
\leq & 2\delta + \left\|B - B_{:C}B_{RC}^{+}B_{R:}\right\| + \delta \left\|B_{RC}^{+}B_{R:}\right\| + \delta \left\|B_{:C}B_{RC}^{+}\right\| \quad (3.11)
\end{aligned}$$

Deterministic part: Bounds on $\left\|B_{RC}^{+}B_{R:}\right\|$ and $\left\|B_{:C}B_{RC}^{+}\right\|$

It remains to bound $\left\|B - B_{:C}B_{RC}^{+}B_{R:}\right\|$, $\left\|B_{RC}^{+}B_{R:}\right\|$, and $\left\|B_{:C}B_{RC}^{+}\right\|$. In this subsection, we obtain bounds for the last two quantities. By the second principle, we do not expect $\left\|B_{RC}^{+}B_{R:}\right\|$ to be much bigger than $\left\|B_{RC}^{+}B_{RC}\right\| \leq 1$. Specifically, by Lemma 3.2.2, we have

$$\begin{aligned}
\left\|B_{RC}^{+}B_{R:}\right\| \leq & \left\|Y_{1,C:}^{+}\right\| \left\|B_{RC}^{+}B_{RC}\right\| + \left\|Y_{1,C:}^{+}\right\| \left\|B_{RC}^{+}B_{R:}Y_2Y_{2,C:}^{*}\right\| + \left\|B_{RC}^{+}B_{R:}Y_2\right\| \\
\leq & \left\|Y_{1,C:}^{+}\right\| + \left\|Y_{1,C:}^{+}\right\| \left\|B_{RC}^{+}\right\| (\left\|(B_{R:} - A_{R:})Y_2Y_{2,C:}^{*}\right\| + \left\|A_{R:}Y_2Y_{2,C:}^{*}\right\|) + \\
& \left\|B_{RC}^{+}\right\| (\|(B_{R:} - A_{R:})Y_2\| + \|A_{R:}Y_2\|) \\
\leq & \left\|Y_{1,C:}^{+}\right\| + \left\|Y_{1,C:}^{+}\right\| \delta^{-1}(\delta + \left\|A_{R:}Y_2Y_{2,C:}^{*}\right\|) + \delta^{-1}(\delta + \|A_{R:}Y_2\|) \\
\leq & 1 + 2\left\|Y_{1,C:}^{+}\right\| + \left\|Y_{1,C:}^{+}\right\| \delta^{-1} \left\|A_{R:}Y_2Y_{2,C:}^{*}\right\| + \delta^{-1} \|A_{R:}Y_2\| .
\end{aligned}$$

By the first principle, the following holds with high probability:

$$\left\|B_{RC}^{+}B_{R:}\right\| = O(\lambda_Y + \lambda_Y\delta^{-1} \left\|A_{R:}Y_2Y_{2,C:}^{*}\right\| + \delta^{-1} \|A_{R:}Y_2\|). \quad (3.12)$$

The same argument works for $\left\|B_{:C}B_{RC}^{+}\right\|$. With high probability,

$$\left\|B_{:C}B_{RC}^{+}\right\| = O(\lambda_X + \lambda_X \delta^{-1} \left\|X_{2,R:}X_2^* A_{:C}\right\| + \delta^{-1} \left\|X_2^* A_{:C}\right\|) \tag{3.13}$$

Deterministic part: Bounding $\left\|B - B_{:C}B_{RC}^{+}B_{R:}\right\|$

Bounding the third quantity requires more work, but the basic ideas are the same. Recall that the second principle suggests that $\left\|B - B_{:C}B_{RC}^{+}B_{R:}\right\|$ cannot be too much bigger than $\left\|B_{RC} - B_{RC}B_{RC}^{+}B_{RC}\right\| = 0$. Applying Lemma 3.2.3 with $B - B_{:C}B_{RC}^{+}B_{R:}$ in the role of A yields

$$\begin{aligned}
\left\|B - B_{:C}B_{RC}^{+}B_{R:}\right\| \leq & \left\|X_{1,R:}^{+}\right\| \left\|Y_{1,C:}^{+}\right\| \left\|X_{2,R:}X_2^*(B - B_{:C}B_{RC}^{+}B_{R:})Y_1 Y_{1,C:}^*\right\| + \\
& \left\|X_{1,R:}^{+}\right\| \left\|Y_{1,C:}^{+}\right\| \left\|X_{1,R:}X_1^*(B - B_{:C}B_{RC}^{+}B_{R:})Y_2 Y_{2,:C}^*\right\| + \\
& \left\|X_{1,R:}^{+}\right\| \left\|Y_{1,C:}^{+}\right\| \left\|X_{2,R:}X_2^*(B - B_{:C}B_{RC}^{+}B_{R:})Y_2 Y_{2,C:}^*\right\| + \\
& \left\|X_{1,R:}^{+}\right\| \left\|X_{1,R:}X_1^*(B - B_{:C}B_{RC}^{+}B_{R:})Y_2\right\| + \\
& \left\|Y_{1,C:}^{+}\right\| \left\|X_2^*(B - B_{:C}B_{RC}^{+}B_{R:})Y_1 Y_{1,C:}^*\right\| + \\
& \left\|X_2^*(B - B_{:C}B_{RC}^{+}B_{R:})Y_2\right\|
\end{aligned}$$

which is in turn bounded by

$$\begin{aligned}
& \left\|X_{1,R:}^{+}\right\| \left\|Y_{1,C:}^{+}\right\| \left\|Y_{1,C:}\right\| (\left\|X_{2,R:}X_2^* B\right\| + \left\|X_{2,R:}X_2^* B_{:C}\right\| \left\|B_{RC}^{+}B_{R:}\right\|) + \\
& \left\|X_{1,R:}^{+}\right\| \left\|Y_{1,C:}^{+}\right\| \left\|X_{1,R:}\right\| (\left\|BY_2 Y_{2,C:}^*\right\| + \left\|B_R Y_2 Y_{2,C:}^*\right\| \left\|B_{:C}B_{RC}^{+}\right\|) \\
& \left\|X_{1,R:}^{+}\right\| \left\|Y_{1,C:}^{+}\right\| (\left\|X_{2,R:}X_2^* BY_2 Y_{2,C:}^*\right\| + \left\|X_{2,R:}X_2^* B_{:C}\right\| \delta^{-1} \left\|B_{R:}Y_2 Y_{2,C:}^*\right\|) + \\
& \left\|X_{1,R:}^{+}\right\| \left\|X_{1,R:}\right\| (\left\|BY_2\right\| + \left\|B_{:C}B_{RC}^{+}\right\| \left\|B_{R:}Y_2\right\|) + \\
& \left\|Y_{1,C:}^{+}\right\| \left\|Y_{1,C:}\right\| (\left\|X_2^* B\right\| + \left\|B_{RC}^{+}B_{R:}\right\| \left\|X_2^* B_{:C}\right\|) + \\
& \left\|X_2^* BY_2\right\| + \left\|X_2^* B_{:C}\right\| \delta^{-1} \left\|B_{R:}Y_2\right\| .
\end{aligned}$$

In the expression above, we have paired $\|X_{1,R:}\|$ with $\left\|X_{1,R:}^{+}\right\|$, and $\|Y_{1,C:}\|$ with $\left\|Y_{1,C:}^{+}\right\|$ because the first principle implies that their products are $O(1)$ with high probability.

This implies that $\left\|B - B_{:C}B_{RC}^{+}B_{R:}\right\|$ is less than a constant times

$$\begin{aligned}
&\lambda_X(\|X_{2,R:}X_2^*B\| + \|X_{2,R:}X_2^*B_C\|\left\|B_{RC}^{+}B_{R:}\right\|)+\\
&\lambda_Y(\left\|BY_2Y_{2,C:}^*\right\| + \left\|B_RY_2Y_{2,C:}^*\right\|\left\|B_{:C}B_{RC}^{+}\right\|)+\\
&\lambda_X\lambda_Y(\left\|X_{2,R:}X_2^*BY_2Y_{2,C:}^*\right\| + \|X_{2,R:}X_2^*B_{:C}\|\,\delta^{-1}\left\|B_{R:}Y_2Y_{2,C:}^*\right\|)+\\
&\|BY_2\| + \left\|B_{:C}B_{RC}^{+}\right\|\|B_{R:}Y_2\| +\\
&\|X_2^*B\| + \left\|B_{RC}^{+}B_{R:}\right\|\|X_2^*B_{:C}\| +\\
&\|X_2^*B_{:C}\|\,\delta^{-1}\|B_{R:}Y_2\|\,.
\end{aligned}$$

We have dropped $\|X_2^*BY_2\|$ because it is dominated by $\|X_2^*B\|$. Equations (3.13) and (3.12) can be used to control $\left\|B_{:C}B_{RC}^{+}\right\|$ and $\left\|B_{RC}^{+}B_{R:}\right\|$. Before doing so, we want to replace B with A in all the other terms. This will introduce some extra δ's. For example, $\|X_{2,R:}X_2^*B\| \leq \|X_{2,R:}X_2^*B - X_{2,R:}X_2^*A\| + \|X_{2,R:}X_2^*A\| \leq \delta + \|X_{2,R:}X_2^*A\|$. Doing the same for other terms, we have that $\left\|B - B_{:C}B_{RC}^{+}B_{R:}\right\|$ is with high probability less than a constant times

$$\begin{aligned}
&\lambda_X(\delta + \|X_{2,R:}X_2^*A\| + (\delta + \|X_{2,R:}X_2^*A_C\|)\left\|B_{RC}^{+}B_{R:}\right\|)+\\
&\lambda_Y(\delta + \left\|AY_2Y_{2,C:}^*\right\| + (\delta + \left\|A_{R:}Y_2Y_{2,C:}^*\right\|)\left\|B_{:C}B_{RC}^{+}\right\|)+\\
&\lambda_X\lambda_Y(\delta + \left\|X_{2,R:}X_2^*AY_2Y_{2,C:}^*\right\| + \|X_{2,R:}X_2^*A_{:C}\| +\\
&\left\|A_{R:}Y_2Y_{2,C:}^*\right\| + \|X_{2,R:}X_2^*A_{:C}\|\,\delta^{-1}\left\|A_{R:}Y_2Y_{2,C:}^*\right\|)+\\
&\delta + \|AY_2\| + (\delta + \|A_{R:}Y_2\|)\left\|B_{:C}B_{RC}^{+}\right\| +\\
&\delta + \|X_2^*A\| + (\delta + \|X_2^*A_{:C}\|)\left\|B_{RC}^{+}B_{R:}\right\| +\\
&\delta + \|X_2^*A_{:C}\| + \|A_{R:}Y_2\| + \|X_2^*A_{:C}\|\,\delta^{-1}\|A_{R:}Y_2\|\,.
\end{aligned}$$

Several terms can be simplified by noting that $\delta \leq \lambda_X\delta \leq \lambda_X\lambda_Y\delta$ and $\|X_2^*A_{:C}\| \leq \|X_2^*A\|$. We shall also use the estimates on $\left\|B_{:C}B_{RC}^{+}\right\|$ and $\left\|B_{RC}^{+}B_{R:}\right\|$, from (3.13)

and (3.12). This leads to

$$\lambda_X(\|X_{2,R:}X_2^*A\| + (\delta + \|X_{2,R:}X_2^*A_{:C}\|)(\lambda_Y + \lambda_Y\delta^{-1}\left\|A_{R:}Y_2Y_{2,C:}^*\right\| + \delta^{-1}\|A_{R:}Y_2\|))+$$
$$\lambda_Y(\left\|AY_2Y_{2,C:}^*\right\| + (\delta + \left\|A_{R:}Y_2Y_{2,C:}^*\right\|)(\lambda_X + \lambda_X\delta^{-1}\|X_{2,R:}X_2^*A_{:C}\| + \delta^{-1}\|X_2^*A_{:C}\|))+$$
$$\lambda_X\lambda_Y(\delta + \left\|X_{2,R:}X_2^*AY_2Y_{2,C:}^*\right\| + \|X_{2,R:}X_2^*A_{:C}\| +$$
$$\left\|A_{R:}Y_2Y_{2,C:}^*\right\| + \|X_{2,R:}X_2^*A_{:C}\|\,\delta^{-1}\left\|A_{R:}Y_2Y_{2,C:}^*\right\|)+$$
$$\|AY_2\| + (\delta + \|A_{R:}Y_2\|)(\lambda_X + \lambda_X\delta^{-1}\|X_{2,R:}X_2^*A_{:C}\| + \delta^{-1}\|X_2^*A_{:C}\|)+$$
$$\|X_2^*A\| + (\delta + \|X_2^*A_{:C}\|)(\lambda_Y + \lambda_Y\delta^{-1}\left\|A_{R:}Y_2Y_{2,C:}^*\right\| + \delta^{-1}\|A_{R:}Y_2\|)+$$
$$\|X_2^*A_{:C}\|\,\delta^{-1}\|A_{R:}Y_2\|\,.$$

Collect the terms by their λ_X, λ_Y factors and drop the smaller terms to obtain that with high probability,

$$\begin{aligned}\left\|B - B_{:C}B_{RC}^+B_{R:}\right\| = O(\,&\lambda_X(\|X_{2,R:}X_2^*A\| + \|A_{R:}Y_2\| + \delta^{-1}\|X_{2,R:}X_2^*A_{:C}\|\,\|A_{R:}Y_2\|)+\\ &\lambda_Y(\left\|AY_2Y_{2,C:}^*\right\| + \|X_2^*A_{:C}\| + \delta^{-1}\left\|A_{R:}Y_2Y_{2,C:}^*\right\|\,\|X_2^*A_{:C}\|)+\\ &\lambda_X\lambda_Y(\delta + \|X_{2,R:}X_2^*A_{:C}\| + \left\|A_{R:}Y_2Y_{2,C:}^*\right\| +\\ &\delta^{-1}\|X_{2,R:}X_2^*A_{:C}\|\left\|A_{R:}Y_2Y_{2,C:}^*\right\| + \left\|X_{2,R:}X_2^*AY_2Y_{2,C:}^*\right\|)+\\ &\|X_2^*A\| + \|AY_2\| + \delta^{-1}\|X_2^*A_{:C}\|\,\|A_{R:}Y_2\|\,).\end{aligned} \tag{3.14}$$

Deterministic part: conclusion of the proof

We now have control over all three terms $\left\|B - B_{:C}B_{RC}^+B_{R:}\right\|, \left\|B_{RC}^+B_{R:}\right\|, \left\|B_{:C}B_{RC}^+\right\|$. Substitute (3.12), (3.13), (3.14) into (3.11). As the right hand side of (3.14) dominates δ multiplied by the right hand side of (3.12), (3.13), we conclude that with high probability, $\left\|A - A_{:C}B_{RC}^+A_{R:}\right\|$ is also bounded by the right hand side of (3.14).

To obtain the basic bound, (3.4), we note that all the "normed terms" on the right hand side of (3.14), e.g., $\left\|A_{R:}Y_2Y_{2,C}^*\right\|$ and $\|X_2^*A\|$, are bounded by ε_k. It follows that with high probability, $\left\|A - A_{:C}B_{RC}^+A_{R:}\right\| = O(\lambda_X\lambda_Y(\delta + \varepsilon + \delta^{-1}\varepsilon^2))$.

To obtain the other bound, (3.5), we need to bound each "normed term" of (3.14)

differently. Recall (3.10). Consider $\|X_{2,R:}X_2^*A_{:C}\|$. We have

$$X_{2,R:}X_2^*A_{:C} = (\widetilde{X}_{2,R:}, \ldots, \widetilde{X}_{\lceil m/k \rceil,R:}) \begin{pmatrix} \widetilde{A}_{21} & \ldots & \widetilde{A}_{2,\lceil n/k \rceil} \\ \vdots & \ddots & \vdots \\ \widetilde{A}_{\lceil m/k \rceil,1} & \ldots & \widetilde{A}_{\lceil m/k \rceil,\lceil n/k \rceil} \end{pmatrix} \begin{pmatrix} \widetilde{Y}_{1,C:}^* \\ \vdots \\ \widetilde{Y}_{\lceil n/k \rceil,C:}^* \end{pmatrix}.$$

In Section 3.2.3, we show that with high probability, $\left\|\widetilde{X}_{i,R:}\right\| = O(\lambda_X^{-1})$ and $\left\|\widetilde{Y}_{j,C:}\right\| = O(\lambda_Y^{-1})$ for all i, j. Recall the definition of ε_k' in (3.3). It follows that with high probability,

$$\|X_{2,R:}X_2^*A_{:C}\| \leq \sum_{i=2}^{\lceil m/k \rceil} \sum_{j=1}^{\lceil n/k \rceil} \left\|\widetilde{X}_{i,R:}\right\| \left\|\widetilde{A}_{ij}\right\| \left\|\widetilde{Y}_{j,C:}\right\| \leq \lambda_X^{-1}\lambda_Y^{-1}\varepsilon_k'.$$

Apply the same argument to other terms on the right hand side of (3.14), e.g., $\left\|X_{2,R:}X_2^*AY_2Y_{2,C:}^*\right\| = O(\lambda_X^{-1}\lambda_Y^{-1}\varepsilon_k')$ and $\|X_2^*A_{:C}\| = O(\lambda_Y^{-1}\varepsilon_k')$ with high probability. Mnemonically, a R in the subscript leads to a λ_X^{-1} and a C in the subscript leads to a λ_Y^{-1}.

Recall that $\left\|A_{:C}B_{RC}^+A_{R:}\right\|$ is bounded by the right hand side of (3.14). Upon simplifying, we obtain that $\left\|A - A_{:C}B_{RC}^+A_{R:}\right\| = O(\lambda_X\lambda_Y\delta + \varepsilon_k' + \lambda_X\lambda_Y\delta^{-1}{\varepsilon_k'}^2)$, i.e., (3.5). The proof is complete.

3.3 Alternative sublinear-time algorithms

3.3.1 Second algorithm

In Algorithm 2, uniform random sampling first helps to trim down A to two factors $A_{:C}$ and $A_{R:}^*$ with $|C| = |R| = \ell = \widetilde{O}(k)$, then rank-revealing QR decompositions (RRQR) are used on $A_{:C}$ and $A_{R:}^*$ to further reduce the small dimension to exactly k.

For dense matrices, the most expensive step in Algorithm 2 is the multiplication of A by $A_{R':}^+$. However, for structured matrices, the most expensive steps of Algorithm 2 are likely to be the RRQR factorization of $A_{:C}$ and $A_{R:}^*$ and the inversion of $A_{:C'}, A_{R':}$, which all take $\widetilde{O}(mk^2)$ time. The overall running time is $O(T_Ak) + \widetilde{O}(mk^2)$, where

Algorithm 2. $O(T_A k) + \widetilde{O}(mk^2)$-time algorithm

Input: A matrix $A \in \mathbb{C}^{m \times n}$ that is approximately rank k, and user-defined parameter $\ell = \widetilde{O}(k)$. Assume $m \geq n$.

Output: Column index set C' of size k, row index set R' of size k, center matrix of a matrix skeleton Z. Implicitly, we have the matrix skeleton $A_{:C'} Z A_{R':}$.

Steps:

1. Let C be a random index set of size ℓ chosen uniformly from $\{1, \ldots, n\}$. Explicitly form $A_{:C}$.
2. Let R be a random index set of size ℓ chosen uniformly from $\{1, \ldots, m\}$. Explicitly form $A_{R:}$.
3. Run RRQR on $A_{:C}$ to select k columns of $A_{:C}$. Denote the result as $A_{:C'}$ where $C' \subseteq C$ indexes the k selected columns of A. This takes $\widetilde{O}(mk^2)$ time and $\widetilde{O}(mk)$ space.
4. Run RRQR on $A^*_{R:}$ to select k rows of $A_{R:}$. Denote the result as $A_{R':}$ where $R' \subseteq R$ indexes the k selected rows of A. This takes $\widetilde{O}(nk^2)$ time and $\widetilde{O}(nk)$ space.
5. Compute $Z = A^+_{:C'}(AA^+_{R':})$. This takes $O(T_A k + mk^2)$ time and $O(mk)$ space, where T_A is the time needed to apply A to a vector.

Matlab code:

```
function [Cp,Z,Rp]=skeleton2(A,l)
    C=randperm(n,l); ind=rrqr(A(:,C),k); Cp=C(ind);
    R=randperm(m,l); ind=rrqr(A(R,:)',k); Rp=R(ind);
    Z=pinv(A(:,Cp))*(A*pinv(A(Rp,:)));
end
```

T_A is the cost of a matrix-vector multiplication.

Note that in the Matlab code, the call `rrqr(A,k)` is assumed to return an index set of size k specifying the selected columns. One can use Algorithm 782 [8] or its Matlab port [69].

It can be easily shown [29] that once $A_{:C'}, A_{R':}$ are fixed, the choice of $Z = A_{:C'}^{+} A A_{R':}^{+}$ is optimal in the Frobenius norm (not operator norm), that is

$$Z = \arg_{W \in \mathbb{C}^{\ell \times \ell}} \|A - A_{:C'} W A_{R':}\|_F .$$

Unsurprisingly, the error estimate is better than in Theorem 3.1.2.

Theorem 3.3.1. *Let A be given by (3.1) for some $k > 0$. Assume $m \geq n$ and X_1, Y_1 are μ-coherent where $\mu = \widetilde{O}(1)$ with respect to m, n. Recall the definition of ε_k in (3.2). Let $\ell \geq 10\mu k \log m$. With probability at least $1 - 4km^{-2}$, Algorithm 2 returns a skeleton that satisfies*

$$\|A - A_{:C'} Z A_{R':}\| = O(\varepsilon_k (mk)^{1/2}).$$

Proof. Let $P = A_{:C'} A_{:C'}^{+} \in \mathbb{C}^{m \times m}$. RRQR [36] selects k columns from $A_{:C}$ such that

$$\|A_{:C} - P A_{:C}\| \leq f(k,\ell)\sigma_{k+1}(A_{:C}) \leq f(k,\ell)\sigma_{k+1}(A) \leq f(k,\ell)\varepsilon_k,$$

where $f(k,\ell) := \sqrt{1 + 2k(\ell - k)}$. We have used the fact that

$$\sigma_{k+1}(A) = \sigma_{k+1} \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \leq \sigma_1 \begin{pmatrix} A_{12} \\ A_{22} \end{pmatrix} \leq \varepsilon_k.$$

See interlacing theorems in [44].

Recall from (3.8) that $\left\|Y_{1,C:}^{+}\right\| = O((n/\ell)^{1/2})$ with probability at least $1 - 2km^{-2}$.

Apply Corollary 3.2.4 to obtain that with high probability

$$\begin{aligned}\|A - PA\| &\leq O(\lambda_Y)\,\|A_{:C} - PA_{:C}\| + O(\lambda_Y)\varepsilon_k + \varepsilon_k \\ &= O(\varepsilon_k(n/\ell)^{1/2} f(k,\ell)) = O(\varepsilon_k(nk)^{1/2}).\end{aligned}$$

Let $P' = A^+_{R':}A_{R':}$. By the same argument, $\|A - AP'\| = O(\varepsilon_k(mk)^{1/2})$ with the same failure probability. Combine both estimates. With probability at least $1 - 4km^{-2}$,

$$\begin{aligned}\left\|A - A_{:C'}A^+_{:C'}AA^+_{R':}A_{R':}\right\| &= \|A - PAP'\| \\ &\leq \|A - PA\| + \|PA - PAP'\| \\ &\leq \|A - PA\| + \|A - AP'\| \\ &= O(\varepsilon_k(mk)^{1/2}).\end{aligned}$$

□

Many algorithms that use the skeleton $A_{:C}(A^+_{:C}AA^+_{R:})A_{R:}$, e.g., in [54], seek to select columns indexed by C such that $\left\|A - A_{:C}A^+_{:C}A\right\|$ is small. Here, we further select k out of $\ell = \widetilde{O}(k)$ columns, which is also suggested in [9]. Their estimate on the error in the operator norm is $O(k\log^{1/2}k)\varepsilon_k + O(k^{3/4}\log^{1/4}k)\|A - A_k\|_F$ where A_k is the optimal rank k approximation to A. In general, $\|A - A_k\|_F$ could be as large as $(n-k)^{1/2}\varepsilon_k$, which makes our bound better by a factor of $k^{1/4}$. Nevertheless, we make the extra assumption that X_1, Y_1 are incoherent.

3.3.2 Third algorithm

Consider the case where *only* X_1 is $\widetilde{O}(1)$-coherent. See Algorithm 3. It computes a skeleton with $\widetilde{O}(k)$ rows and k columns in $\widetilde{O}(nk^2+k^3)$ time. Intuitively, the algorithm works as follows. We want to select k columns of A but running RRQR on A is too expensive. Instead, we randomly choose $\widetilde{O}(k)$ rows to form $A_{R:}$, and select our k columns using the much smaller matrix $A_{R:}$. This works because X_1 is assumed to be $\widetilde{O}(1)$-coherent and choosing almost any $\widetilde{O}(k)$ rows will give us a good sketch of A.

Algorithm 3. $\widetilde{O}(nk^2)$-time algorithm

Input: A matrix $A \in \mathbb{C}^{m \times n}$ that is approximately rank k, and user-defined parameter $\ell = \widetilde{O}(k)$.

Output: Column index set C' of size k, row index set R of size ℓ, center matrix of a matrix skeleton Z. Implicitly, we have the matrix skeleton $A_{:C'}ZA_{R:}$.

Steps:

1. Let R be a random index set of size ℓ chosen uniformly from $\{1, \ldots, m\}$. Explicitly form $A_{R:}$.
2. Run RRQR on $A_{R:}$ and obtain a column index set C'. Note that $A_{R:} \simeq A_{RC'}(A^+_{RC'}A_{R:})$ where $A_{RC'}$ contains k columns of $A_{R:}$. This takes $\widetilde{O}(nk^2)$ time and $\widetilde{O}(nk)$ space.
3. Compute $Z = A^+_{RC'}$. This takes $\widetilde{O}(k^3)$ time and $\widetilde{O}(k^2)$ space.

Matlab code:

```
function [Cp,Z,R]=skeleton3(A,l)
    R=randperm(m,l); Cp=rrqr(A(R,:),k); Z=pinv(A(R,Cp));
end
```

Theorem 3.3.2. *Let A be given by (3.1) for some $k > 0$. Assume $m \geq n$ and X_1 is μ-coherent where $\mu = \widetilde{O}(1)$ with respect to m, n. (Y_1 needs not be incoherent.) Recall the definition of ε_k in (3.2). Let $\ell \geq 10\mu k \log m$. Then, with probability at least $1 - 2km^{-2}$, Algorithm 3 returns a skeleton that satisfies*

$$\|A - A_{:C'} Z A_{R:}\| = O(\varepsilon_k (mn)^{1/2}).$$

Proof. We perform RRQR on $A_{R:}$ to obtain $A_{R:} \simeq A_{RC'} D$ where $D = A^{+}_{RC'} A_{R:}$ and C' indexes the selected k columns. We want to use the second principle to "undo the row restriction" and infer that $A \simeq A_{C'} D$, the output of Algorithm 3. The details are as follows.

Strong RRQR [36] guarantees that

$$\|A_{R:} - A_{RC'} D\| \leq \sigma_{k+1}(A_{R:}) f(k,n) \leq \sigma_{k+1}(A) f(k,n) \leq \varepsilon_k f(k,n)$$

and

$$\|D\| \leq f(k,n)$$

where $f(k,n) = \sqrt{1 + 2k(n-k)}$. Prepare to apply a transposed version of Corollary 3.2.4, i.e.,

$$\|A - AP\| \leq \left\|X^{+}_{1,R:}\right\| \|A_{R:} - A_{R:}P\| + \left\|X^{+}_{1,R:}\right\| \|I - P\| \|X_{2,R:} X_2^* A\| + \|I - P\| \|X_2^* A\|. \tag{3.15}$$

Let $P = S_{C'} D$, so that $\|P\| \leq \|D\| \leq f(k,n)$. Note that $AP = A_{:C'} A^{+}_{RC'} A_{R:}$. By (3.8), with probability at least $1 - 2km^{-2}$, $\left\|X^{+}_{1,R:}\right\| = O((m/\ell)^{1/2})$. By (3.15),

$$\begin{aligned} \|A - AP\| &\leq O(\lambda_X) \|A_{R:} - A_{RC'} D\| + O(\lambda_X)(1 + \|P\|)\varepsilon_k + (1 + \|P\|)\varepsilon_k \\ &= O(\varepsilon_k f(k,n)(m/\ell)^{1/2}) = O(\varepsilon_k (mn)^{1/2}). \end{aligned}$$

□

If X_1 is not incoherent and we fix it by multiplying on the left by a randomized

Fourier matrix $\mathcal{F}D$ (cf. Section 3.1.6), then we arrive at the algorithm in [51]. The linear algebraic part of their proof combined with the first principle will lead to similar bounds. What we have done here is to split the proof into three simple parts: (1) show that $\widetilde{X}_1 := \mathcal{F}DX_1$ is incoherent, (2) use the first principle to show that $\widetilde{X}_{1,R:}$ is "sufficiently nonsingular", (3) apply the second principle.

3.3.3 Comparison of three algorithms

Here is a summary of the three algorithms studied in this chapter. Assume $m \geq n$. Recall that $A \simeq X_1 A_{11} Y_1^*$. For Algorithm 1 and Algorithm 2, assume that X_1, Y_1 are both incoherent. For Algorithm 3, assume that X_1 is incoherent.

	No. of rows	No. of columns	Upper bound on error in the operator norm	Running time	Memory
Alg. 1	$\ell = \widetilde{O}(k)$	$\ell = \widetilde{O}(k)$	$O(\varepsilon_k \frac{(mn)^{1/2}}{\ell})$	$\widetilde{O}(k^3)$	$\widetilde{O}(k^2)$
Alg. 2	k	k	$O(\varepsilon_k (mk)^{1/2})$	$O(T_A k) + \widetilde{O}(mk^2)$	$\widetilde{O}(mk)$
Alg. 3	$\widetilde{O}(k)$	k	$O(\varepsilon_k (mn)^{1/2})$	$\widetilde{O}(nk^2)$	$\widetilde{O}(nk)$

Recall that T_A is the cost of applying A to a vector. If $T_A = \widetilde{O}(nk)$ and $m = O(n)$, then the running time of Algorithm 2 and Algorithm 3 are comparable and we would recommend using Algorithm 2 because it has a better error guarantee.

Otherwise, if T_A is on the order of mn, then Algorithm 2 is much slower than Algorithm 1 and Algorithm 3, and is not recommended. Compared to Algorithm 3, and in that scenario, Algorithm 1 is much faster and has better error guarantees, so we view it as the better choice. The advantages of Algorithm 3 are that it selects exactly k columns and does not require Y_1 to be incoherent.

If we cannot afford using $\widetilde{O}(mk)$ memory or having a running time that scales with m, n, then Algorithm 1 is the only possible choice here. Although Theorem 3.1.2 suggests that the error for Algorithm 1 grows with $(mn)^{1/2}$, we believe that in practice, the error usually increases with $m^{1/2}$. See Section 3.4 for some numerical

results.

Finally, we remind the reader that these recommendations are made based on error guarantees which are not always tight.

3.4 Examples

3.4.1 First toy example: convolution

This first example shows that in Algorithm 1, it is crucial to regularize when inverting A_{RC} because otherwise, the error in the operator norm can blow up. In fact, even when A is positive definite and we pick $C = R$ as in the work of Gittens [34], we encounter the same need to regularize. The reason is that due to numerical errors, A_{RC} tends to be ill-conditioned when A_{RC} has more rows and columns than the rank of A. In other words, numerical errors introduce spurious small but nonzero singular values in A_{RC} and inverting the components corresponding to these small singular values leads to large errors.

The experiment is set up as follows. Let $A = X\Sigma X^* \in \mathbb{C}^{n\times n}$ where X is the unitary Fourier matrix and Σ is a diagonal matrix of singular values. Note that every entry of X is of magnitude $n^{-1/2}$ and X is 1-coherent. Fix $n = 301$, $\ell = 100$ and $k = 10, 30, 50$. Pick $\varepsilon = \varepsilon_k = \sigma_{k+1} = \ldots = \sigma_n = 10^{-15}$. Pick the largest k singular values to be *logarithmically spaced* between 1 and ε. Note that A is Hermitian and positive definite. In each random trial, we randomly shuffle the singular values, pick ℓ random rows and columns and measure $\|A - A_{:C}ZA_{R:}\|$. The only parameters being varied are k and δ. Note that although $R \neq C$ in this experiment, similar results are obtained when $R = C$.

From (3.4) in Theorem 3.1.2, we expect that when variables such as n, m, ℓ, k are fixed,

$$\log \|A - A_{:C}ZA_{R:}\| \sim \log(\delta^{-1}(\varepsilon_k + \delta)^2) = -\log\delta + 2\log(\varepsilon_k + \delta). \tag{3.16}$$

Consider a plot of $\|A - A_{:C}ZA_{R:}\|$ versus δ on a log-log scale. According to the above

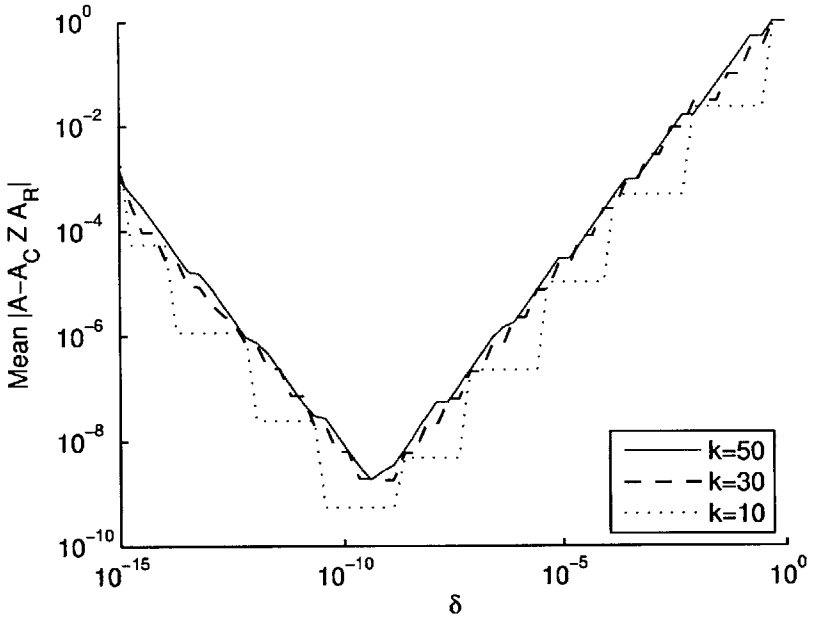


Figure 3-1: Loglog plot of the empirical mean of the error in operator norm by the $\widetilde{O}(k^3)$ algorithm versus δ, a regularization parameter. This relationship between the error and δ agrees with Theorem 3.1.2. See (3.16). More importantly, the error blows up for small δ, which implies that the regularization step should not be omitted.

equation, when $\delta \ll \varepsilon_k$, the first term dominates and we expect to see a line of slope -1, and when $\delta \gg \varepsilon_k$, $\log(\varepsilon_k + \delta) \simeq \log \delta$ and we expect to see a line of slope $+1$. Indeed, when we plot the experimental results in Figure 3-1, we see a right-angled V-curve.

The point here is that the error in the operator norm can blow up as $\delta \to 0$.

A curious feature of Figure 3-1 is that the error curves resemble staircases. As we decrease k, the number of distinct error levels seems to decrease proportionally. A possible explanation for this behavior is that the top singular vectors of $A_{:C}$ match those of A, and as δ increases from $\sigma_i(A)$ to $\sigma_{i-1}(A)$ for some small i, smaller components will not be inverted and the error is all on the order of $\sigma_i(A)$.

3.4.2 Second toy example

For the second experiment, we consider $A = X\Sigma Y^*$ where X, Y are unitary Fourier matrices with randomly permuted columns and Σ is the diagonal matrix of singular

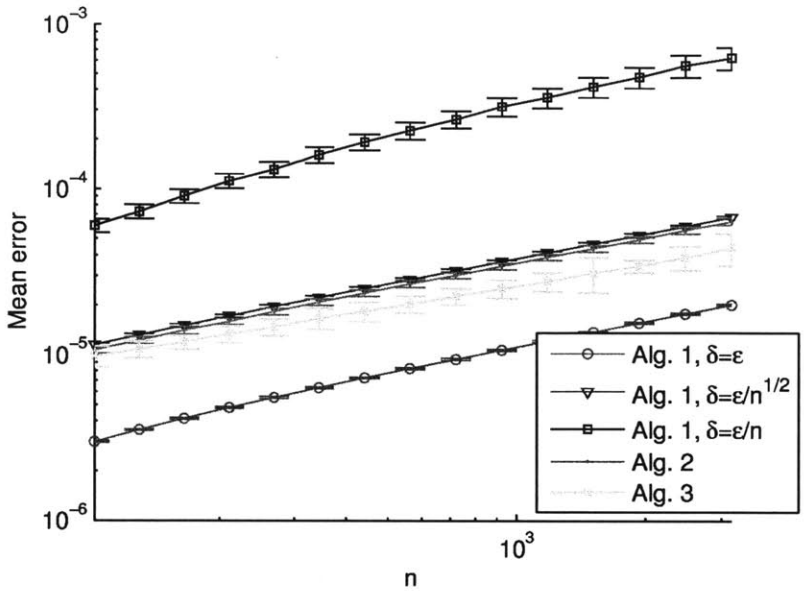


Figure 3-2: The above is a loglog plot of the empirical mean of the error in operator norm versus n, the size of square matrix A. Fix $k = 10$, $\ell = 40$ and $A = X\Sigma Y^*$ where X, Y are unitary Fourier matrices with randomly permuted columns and Σ is the diagonal matrix of singular values. The top k singular values are set to 1 and the others are set to $\varepsilon = 10^{-6}$. When we run Algorithm 1 with $\delta = \varepsilon, \varepsilon/\sqrt{n}, \varepsilon/n$, the expected errors seem to grow with $n^{0.55}, n^{0.51}, n^{0.69}$ respectively. For Algorithm 2 and 3, the expected errors seem to grow with $n^{0.52}, n^{0.43}$ respectively. The errorbars correspond to $\frac{1}{5}$ of the standard deviations obtained empirically. Observe that the error in Algorithm 3 fluctuates much more than Algorithm 1 with $\delta = \varepsilon, \varepsilon/\sqrt{n}$.

values. Fix $k = 10$, $\ell = 40$. The singular values are set such that the largest k singular values are all 1 and the other singular values are all $\varepsilon = 10^{-6}$. We consider all three algorithms. For Algorithm 1, we set δ in three different ways: $\delta = \varepsilon$, $\delta = \varepsilon/\sqrt{n}$ and $\delta = \varepsilon/n$.

We plot the error $\|A - A_{:C}ZA_{R:}\|$ versus n in Figure 3-2. The numerical results show that if we pick $\delta = \varepsilon/\sqrt{n}$ for Algorithm 1, then the estimated mean error is almost the same as that of Algorithm 2 — they both scale with $n^{0.51}$, with k, ℓ fixed. On the other hand, if we pick $\delta = \varepsilon$ as suggested by (3.4) of Theorem 3.1.2, the expected error seems to grow with $n^{0.55}$ which is slightly worse than Algorithm 2 but much better than described in (3.6).

The expected error of Algorithm 3 seems to grow with $n^{0.43}$ which is the best in this experiment. However, its error is not as concentrated around the mean as

Algorithm 2 and Algorithm 1 with $\delta = \varepsilon, \varepsilon/\sqrt{n}$.

3.4.3 Smooth kernel

Consider a 1D integral operator with a kernel K that is analytic on $[-1,1]^2$. Define A as $(A)_{ij} = cK(x_i, y_j)$ where the nodes $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$ are *uniformly spaced* in $[-1,1]$. First, suppose $K = \sum_{1 \le i,j \le 6} c_{ij} T_i(x) T_j(y) + 10^{-3} T_{10}(x) T_{10}(y) + 10^{-9} N$ where $T_i(x)$ is the i-th Chebyshev polynomial and N is the random Gaussian matrix, i.e., noise. The coefficients c_{ij}'s are chosen such that $\|A\| \simeq 1$. Pick $n = m = 10^3$ and slowly increase ℓ, the number of rows and columns sampled by the $\widetilde{O}(k^3)$ algorithm. As shown in Figure 3-3, the skeleton representation $A_{:C} Z A_{R:}$ converges rapidly to A as we increase ℓ.

Next, consider $K(x,y) = c\exp(xy)$. Let $n = 900$ and pick c to normalize $\|A\| = 1$. We then plot the empirical mean of the error of the $\widetilde{O}(k^3)$ algorithm against ℓ on the left of Figure 3-4. Notice that the error decreases exponentially with ℓ.

To understand what is happening, imagine that the grid is infinitely fine. Let $\varphi_1, \varphi_2, \ldots$ be Legendre polynomials. Recall that these polynomials are orthogonal on $[-1,1]$. Define the matrices X, Y as $(X)_{ij} = \varphi_j(x_i)$ and $(Y)_{ij} = \varphi_j(y_i)$. Assume the φ_i's are scaled such that X, Y are unitary. It is well-known that if we expand K in terms of Chebyshev polynomials or Legendre polynomials [10] or prolate spheroidal wave functions [78], the expansion coefficients will decay exponentially. This means that the entries of X^*AY should decay exponentially away from the topleft corner and $\varepsilon'_k = \Theta(\varepsilon_k)$ (cf. (3.2) and (3.3)). We confirm this by plotting $\varepsilon_k, \varepsilon'_k$ versus k on the right of Figure 3-4. The actual X, Y used to obtain this plot are obtained by evaluating the Legendre polynomials on the uniform grid and orthonormalizing. It can be verified that the entries of X, Y are of magnitude $O((k/n)^{1/2})$ which implies a coherence $\mu \simeq k$, independent of n. The implication is that the algorithm will continue to perform well as n increases.

As ℓ increases, we can apply Theorem 3.1.2 with a larger k. Since $\varepsilon_k, \varepsilon'_k$ decrease exponentially, the error should also decrease exponentially. However, as k increases beyond $\simeq 15$, ε_k stagnates and nothing can be gained from increasing ℓ. In general,

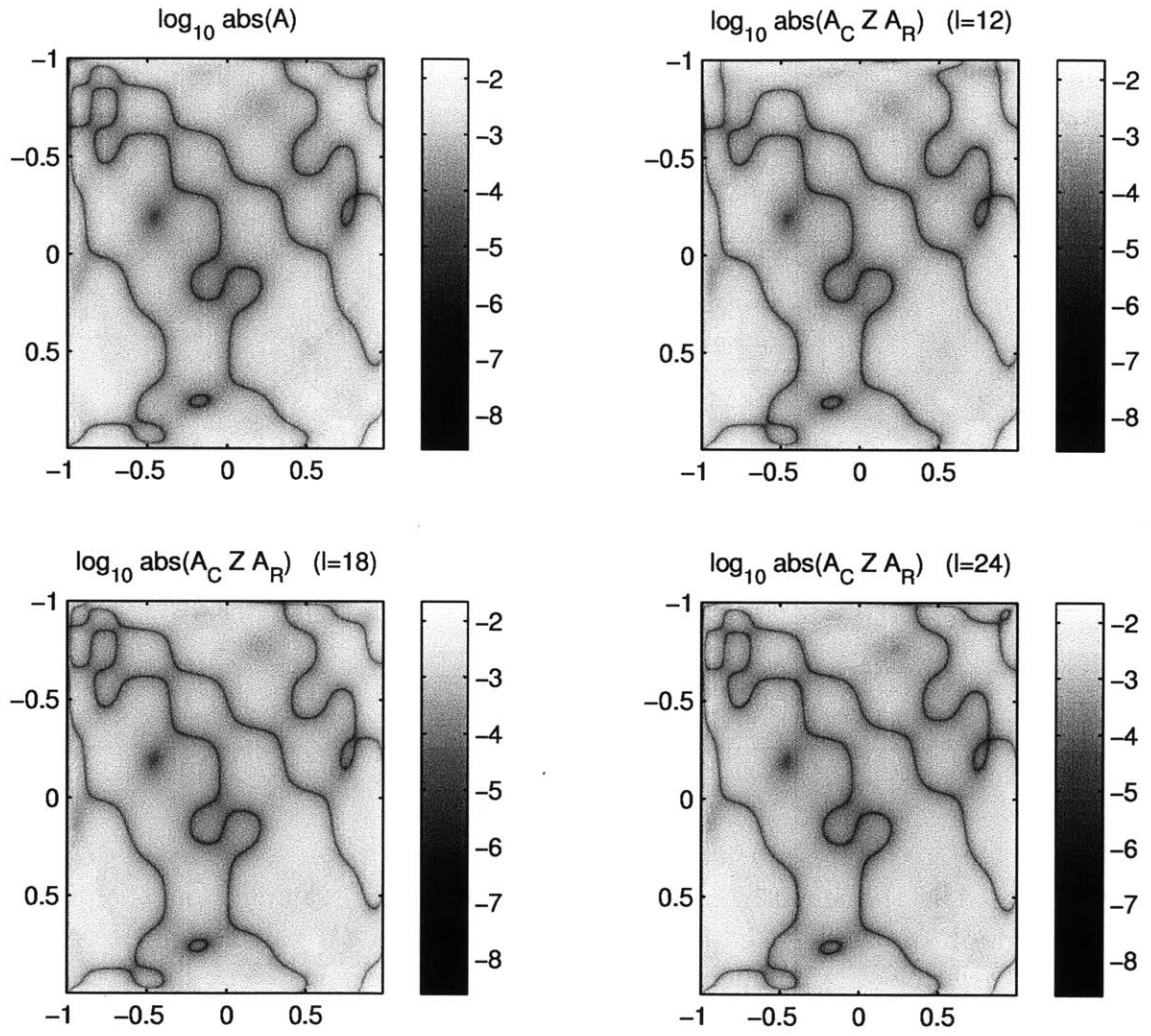


Figure 3-3: A is the smooth kernel $K(x, y)$ where K is the sum of 6^2 low degree Chebyshev polynomials evaluated on a $10^3 \times 10^3$ uniform grid. The topleft figure is A while the other figures show that the more intricate features of A start to appear as we increase ℓ from 12 to 18 to 24. Recall that we sample ℓ rows and ℓ columns in the $\widetilde{O}(k^3)$ algorithm.

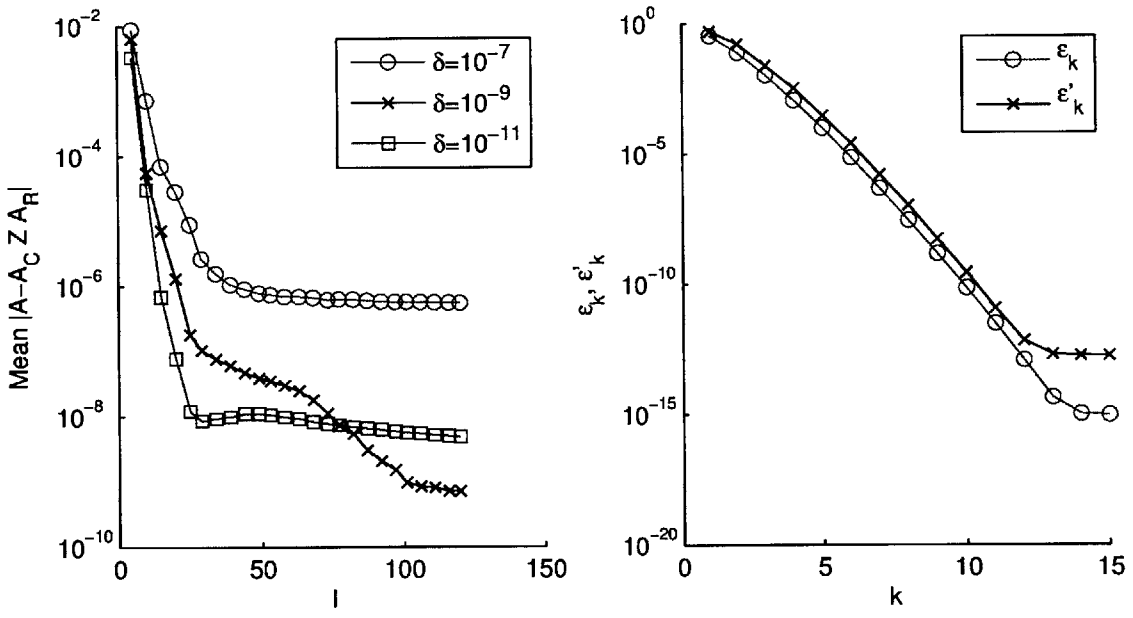


Figure 3-4: A is the smooth kernel $K(x,y) = \exp(-xy)$ sampled on a uniform grid. The graph on the left shows that the error of the $\widetilde{O}(k^3)$ algorithm decreases exponentially with ℓ, the number of sampled rows and columns. The figure on the right shows that if we expand A in terms of Legendre polynomials, the coefficients (and therefore $\varepsilon_k, \varepsilon'_k$) decay exponentially. See (3.1), (3.2) and (3.3) for the definitions of ε_k and ε'_k.

as ε_k decreases, we should pick a smaller δ. But when $k \gtrsim 15$, choosing a smaller δ does not help and may lead to worse results due to the instability of pseudoinverses. This is evident from Figure 3-4.

A recent paper by Platte et al. [62] states that we cannot have an exponential decrease of the error without a condition number that grows exponentially. In our case, the random selection of columns and rows correspond to selecting interpolation points randomly, and δ serves as a regularization parameter of the interpolation method. Due to the regularization, we can only expect an exponential decrease of the error up to a limit dependent on δ.

3.4.4 Fourier integral operators

In [16], Candes et al. consider how to efficiently apply 2D Fourier integral operators of the form

$$Lf(x) = \int_{\xi} a(x,\xi) e^{2\pi i \Phi(x,\xi)} \hat{f}(\xi) d\xi$$

where $\hat{f}(\xi)$ is the Fourier transform of f, $a(x,\xi)$ is a smooth amplitude function and Φ is a smooth phase function that is homogeneous, i.e., $\Phi(x,\lambda\xi) = \lambda\Phi(x,\xi)$ for any $\lambda > 0$. Say there are N^2 points in the space domain and also the frequency domain.

The main idea is to split the frequency domain into $\sqrt{N}$ wedges, perform a Taylor expansion of $\Phi(x,\cdot)$ about $|\xi|\,\hat{\xi}_j$ where j indexes a wedge, and observe that the residual phase $\Phi_j(x,\xi) := \Phi(x,\xi) - \Phi(x,|\xi|\,\hat{\xi}_j)\cdot\xi$ is nonoscillatory. Hence, the matrix $A^{(j)}_{st} := \exp(2\pi i \Phi_j(x_s,\xi_t))$ can be approximated by a low rank matrix, i.e., $\exp(2\pi i\Phi_j(x,\xi))$ can be written as $\sum_{q=1}^{r} f_q(x) g_q(\xi)$ where r, the separation rank, is independent of N. By switching order of summations, the authors arrive at $\widetilde{O}(N^{2.5})$ algorithms for both the preprocessing and the evaluation steps. See [16] for further details.

What we are concerned here is the approximate factorization of $A^{(j)}$. This is a N^2 by $N^{1.5}$ matrix since there are N^2 points in the space domain and $N^2/\sqrt{N}$ points in one wedge in the frequency domain. In [16], a slightly different algorithm is proposed.

1. Uniformly and randomly select ℓ rows and columns to form $A_{R:}$ and $A_{:C}$.

2. Perform SVD on $A_{:C}$. Say $A_{:C} = U_1\Lambda_1 V_1^* + U_2\Lambda_2 V_2^*$ where U, V are unitary and $\|\Lambda_2\| \leq \delta$, a user specified parameter.

3. Return the low rank representation $U_1 U_{1,R:}^+ A_{R:}$.

In the words of the authors, "this randomized approach works well in practice although we are not able to offer a rigorous proof of its accuracy, and expect one to be non-trivial" [16].

We are now in a position to explain why this randomized approach works well. Consider equations (3.1) and (3.2). Let B be a perturbation of A such that $B_C = U_1\Lambda_1 V_1^*$ and $\|A - B\| \leq \delta$. Since Λ_1 is invertible, the output can be rewritten as

$$U_1 U_{1R}^+ A_{R:} = B_{:C} B_{RC}^+ A_{R:}.$$

By following the proof of Theorem 3.1.2, we see that

$$\left\|A - B_{:C} B_{RC}^+ A_{R:}\right\| = O(\left\|B - B_{:C} B_{RC}^+ B_{R:}\right\|)$$

and that all the estimates in Theorem 3.1.2 must continue to hold.

The analysis presented here therefore answers the questions posed in [16]. We believe that the assumption of incoherence of the generating vectors is precisely the right framework to express the error guarantees of the skeleton in such situations.

An important subclass of Fourier integral operators is pseudodifferential operators. These are linear operators with pseudodifferential symbols that obey certain smoothness conditions [70]. In Discrete Symbol Calculus [24], a similar randomized algorithm is used to derive low rank factorizations of such smooth symbols. It is likely that the method works well here in the same way as it works well for a smooth kernel as discussed in the previous section.

Chapter 4

Sparse Fourier transform using the matrix pencil method

4.1 Introduction

Frequency-sparse signals are ubiquitous. We are interested in computing the discrete Fourier transform (DFT) of such signals much faster than using standard FFT algorithms.

Before we proceed, we establish some notation for the rest of the paper. For any positive integer T, let $[T] = \{0, 1, \ldots, T-1\}$. If T is odd, let $[\![T]\!] = \{-\frac{T-1}{2}, \ldots, \frac{T-1}{2}\}$. Let $\|\cdot\|$ be the ℓ^2 norm, $\|\cdot\|_p$ be the ℓ^p norm and $\sigma_j(\cdot)$ be the j-th largest singular value. The overline denotes complex conjugation or set complement. When a set is used as a subscript of a vector, we refer to the vector restricted to coordinates indexed by the set. Let $\widetilde{O}(\cdot)$ be the $O(\cdot)$ notation with log factors dropped. For any $b > 0$, let $a\%b$ denote $a \bmod b$ with the result being in $[0, b)$. Define $\mathrm{dist} : \mathbb{R} \times \mathbb{R} \to [0, \frac{1}{2}]$ as the wraparound distance in $[0, 1)$, i.e., $\mathrm{dist}(\xi_1, \xi_2) = \min_{k \in \mathbb{Z}} |k + \xi_1 - \xi_2|$.

Let N be a *large prime*[1]. Given the signal $x \in \mathbb{C}^N$, we want to compute its DFT

[1]Our algorithm MPFFT works even when N is not prime. However, we would have to be more careful when analyzing the random shuffling of modes in Proposition 4.5.5. We envision that the overall running time will be worsened by a factor of $N/\varphi(N) = O(\log \log N)$ where $\varphi(N)$ is the Euler totient function, as suggested by [41, Lemma 3.6].

$\hat{x} \in \mathbb{C}^N$. They are related by

$$\hat{x}_k = \frac{1}{N} \sum_{t \in [N]} x_t e^{-2\pi i k t/N}; \quad x_t = \sum_{k \in [N]} \hat{x}_k e^{2\pi i k t/N}.$$

Assume that $\hat{x}$ is S-sparse with some additive noise. Traditional FFT algorithms can compute $\hat{x}$ in $O(N \log N)$ time. However, since there are only $\binom{N}{S}$ possible solutions, the ideal algorithm should run in $O\left(\log_2 \binom{N}{S}\right) = O(S \log N)$ time, which is much superior to $O(N \log N)$.

Some existing sparse Fourier transform (SFT) algorithms *already* achieve a running time of $\widetilde{O}(S)$. We believe the fastest implemented and published *robust* $\widetilde{O}(S)$-*time* SFT algorithm is currently the AAFFT (Ann Arbor FFT) [47]. In this paper, we present a robust $\widetilde{O}(S)$-time SFT algorithm called MPFFT (Matrix Pencil FFT) that runs many times faster than AAFFT. The major new ingredient is a *mode collision detector* based on the *matrix pencil method.* This mechanism enables us to use fewer samples of the input signal.

To facilitate the discussion and the analysis of MPFFT, we assume that every heavy mode of x stands out against the noise in the following sense:

Assumption 4.1.1. *For any* $0 < \rho < 1$, *define the set of* ρ-heavy modes *as*

$$\Lambda_\rho(x) = \{k \in [N] : |\hat{x}_k| \geq 1 - \rho\}.$$

Assume that there exists a $0 < \rho \ll 1$ *such that* $|\Lambda_\rho(x)| \leq S$ *and* $\left\| x_{\overline{\Lambda_\rho(x)}} \right\| \ll 1$.

We emphasize that MPFFT does work for a wide variety of inputs as demonstrated numerically in Section 4.6. We impose the above assumption on x so that we can provide a formal analysis of MPFFT in Section 4.5. Throughout the paper, ρ should be regarded as very small, and for clarity, we will often drop ρ from the discussion. For example, we may write Λ_ρ as Λ and refer to ρ-heavy modes as heavy modes. Modes not in $\Lambda_\rho(x)$ are referred to as *nonheavy modes* of x.

Assumption 4.1.1 can be interpreted as follows. Suppose we have an underlying signal X that is exactly S-sparse in frequency space. Assume by rescaling that its

nonzero Fourier coefficients have magnitude at least 1. In our notation, $|\Lambda_0(X)| = S$ and $\left\|X_{\overline{\Lambda_0(X)}}\right\| = 0$. Sample X in time to obtain x. This introduces errors that can be modelled as Gaussian random variables, i.e., for some $\sigma \ll 1$,

$$x_t - X_t \sim N(0, \sigma^2).$$

Note that $\hat{x}_k - \hat{X}_k \sim N(0, \sigma^2/N)$ and $\mathbb{E}\left\|\hat{x} - \hat{X}\right\|^2 = \sigma^2$. Let $\rho = \left\|\hat{x} - \hat{X}\right\|_\infty$. It is well-known that with high probability, $\left\|\hat{x} - \hat{X}\right\| = O(\sigma)$ and $\rho = \widetilde{O}\left(\frac{\sigma}{\sqrt{N}}\right) \ll 1$. Observe that $\Lambda_\rho(x) = \Lambda_0(X)$ and the noise energy is bounded as $\left\|\hat{x}_{\overline{\Lambda_\rho(x)}}\right\|^2 = \sum_{k \notin \Lambda_0(X)} \left|\hat{x}_k - \hat{X}_k\right|^2 \leq \left\|\hat{x} - \hat{X}\right\|^2 = O(\sigma^2) \ll 1$. We have verified that x satisfies Assumption 4.1.1.

In previous work on SFT, almost all numerical examples use an input signal that satisfies Assumption 4.1.1. It is unfortunate that even for this simple test case, existing SFT algorithms seem hardly more appealing than FFTW [31], the fastest implementation of standard FFT algorithms. The problem is that large constants are hidden in their $\widetilde{O}(S)$ running time and for a fixed N, most existing SFT algorithms are faster than FFTW only when $S \ll N$. Otherwise, they face other major problems that are summarized in Figure 4-1.

Our algorithm MPFFT is presented in two forms. The first form requires Assumption 4.1.1 and is analyzed in this paper. The second form is implemented and seems to work well without Assumption 4.1.1. The numerical results in Section 4.6 show that the second form of MPFFT runs much faster than AAFFT and we encourage the reader to try out the publicly available code.

4.1.1 Review of sFFT3.0

Our algorithm MPFFT is an extension of sFFT3.0 [40]. To understand the improvements we have made to sFFT3.0, it is imperative to understand how sFFT3.0 works. The goal of this section is to introduce the reader to the main ideas of sFFT3.0. We begin by listing its pseudocode in Figure 4-2.

First and foremost, sFFT3.0 is an iterative algorithm. At the beginning of it-

Method	Faster when $S <?$	What are the issues?	What is appealing?
AAFFT0.9	$\simeq 150$	Requires S to be too small relative to N	Fastest published $\widetilde{O}(S)$-time algorithm.
sFFT1.0	$\simeq 1000$	Running time scales with $N^{1/2}$ which is nonoptimal.	Faster than FFTW over a respectable range of S.
sFFT2.0	$\simeq 1200$	Running time scales with $N^{1/3}$ which is nonoptimal.	Faster than FFTW over a respectable range of S.
sFFT3.0	$\simeq 95000$	Nonheavy modes remain in the solution, leading to a final error that is not acceptable. *Nonrobust* because mode identification fails when there is too much noise. See Section 4.1.2 for more.	Simple, elegant, very fast.
sFFT4.0	Unknown	Not implemented. Not likely to be much faster than AAFFT. See Section 4.1.2 for more.	Offer new insights on the analysis of SFT algorithms.

Figure 4-1: List of some SFT algorithms. The second column shows the range of S where the SFT algorithm is faster than FFTW for a fixed $N = 2^{22}$. The values for sFFT1.0, sFFT2.0 are derived from our own numerical tests (cf. Section 4.6.1) and differ from [41]. We believe the reason is that in our tests, FFTW is compiled with hardware acceleration on the same machine as it is run. The value for AAFFT is obtained from [47]. The value for sFFT3.0 is obtained from [41].

eration r, our approximate of $\hat{x}$ is $\hat{z}^r$, and we strive to recover at least a constant proportion of the heavy modes of the residual signal $x^r = x - z^r$. We claim that with good probability, this objective is achieved in each iteration, such that the number of heavy modes left *decays exponentially* and the total running time is dominated by the running time of the first iteration, which is $O(S \log N)$.

sFFT3.0 and many other SFT algorithms rely on a basic but important operation called *"binning"*. This is also the most computationally expensive step of these SFT algorithms. As sFFT3.0 bins only two signals in each outer iteration, which is much fewer than AAFFT or sFFT4.0, it is not surprising that sFFT3.0 runs much faster in comparison.

Binning is carried out on the signal y, which is the residual signal x^r randomly transformed such that

$$y_t = x^r_{\alpha t+\gamma} e^{2\pi i \beta t/N}; \quad \hat{y}_{\varphi(k)} = \hat{x}^r_k e^{2\pi i \gamma k/N} \tag{4.1}$$

where $\varphi(k) = \alpha k + \beta$ is a random permutation with α, β uniformly chosen from $[N]\backslash\{0\}$ and $[N]$ respectively. Binning of y requires only samples of $y, \hat{y}$ which can be obtained as samples of $x^r, \hat{x}^r$ by (4.1). This means that the random transform of x^r into y is *implicit* and we do not compute or store y or $\hat{y}$ in full.

Think of the spectrum of y as being supported on the grid $[N]/N \subset [0,1)$. Split $[0,1)$ evenly into B_r intervals $[\frac{b}{B_r}, \frac{b+1}{B_r})$. We say *mode k_* lands in bin b* if $\varphi(k_*)/N \in [\frac{b}{B_r}, \frac{b+1}{B_r})$. Define $h : [N] \to [B_r]$ as $h(k) = \left\lfloor \frac{\varphi(k) B_r}{N} \right\rfloor$. Observe that $h(k)$ is the bin that mode k lands in. When we bin signal y, we produce B_r *bin coefficients* which *ideally* satisfies the following: for $b \in [B_r]$,

$$Y^b_0 := \sum_{k \in h^{-1}(b)} \hat{y}_{\varphi(k)} = \sum_{k \in h^{-1}(b)} \hat{x}^r_k e^{2\pi i \gamma k/N}.$$

In reality, (4.2) is not correct because realizing it is computationally infeasible. Instead, (4.2) is only approximately realized in $O\left(\frac{B_r}{\kappa_r} \log \frac{1}{\delta} + |\text{supp}\, \hat{z}^r|\right)$ time where κ_r, δ controls the quality of the approximation. More details on binning is found in Section

procedure sFFT3($x \in \mathbb{C}^N, S$)
  $\hat{z}^0 \leftarrow 0$
  for $r \leftarrow 0, 1, \ldots, R-1 = O(\log S)$ **do** ▷ Start of an outer iteration
    Let $\varphi(k) = \alpha k + \beta$ be uniformly chosen permutation of $[N]$
    Let γ be uniformly chosen from $[N]$ ▷ $\hat{y}_{\varphi(k)} = \hat{x}^r_k e^{2\pi i \gamma k/N}$
    $B_r \leftarrow B a_B^r$ ▷ $0 < a_B < 1, B = O(S)$
    $\kappa_r \leftarrow \kappa a_\kappa^r$ ▷ $a_B < a_\kappa < 1$
    $\mathcal{H} \leftarrow \{0, 1\}$ ▷ $\delta = O(\log N)$
    $Y' \leftarrow \texttt{BinInTime}(x, \alpha, \beta, \gamma, \mathcal{H}, B_r, \delta, \kappa_r)$
    $Y'' \leftarrow \texttt{BinInFrequency}(\hat{z}^r, \alpha, \beta, \gamma, \mathcal{H}, B_r, \delta, \kappa_r)$
    $Y \leftarrow Y' - Y''$ ▷ Obtain B_r sub-signals Y^b
    for $b \in [B_r]$ such that $\left|Y_0^b\right| \geq 1/2$ **do** ▷ Do not process every bin
      Identify one mode k_0 using $\{Y_\tau^b : \tau \in \mathcal{H}\}$:
        $\xi_0 \leftarrow \arg(Y_1^b / Y_0^b)$
        $k_0 \leftarrow \text{round}\left(\frac{N}{2\pi}\xi_0\right)$
      Estimate $\hat{y}_{k_0}$ as $\hat{y}'_{k_0} \leftarrow Y_0^b$
      $k_* \leftarrow \varphi^{-1}(k_0)$
      Update our solution by $\hat{z}^{r+1}_{k_*} \leftarrow \hat{z}^r_{k_*} + \hat{y}'_{k_0} e^{-2\pi i \gamma k_*/N}$
    end for
  end for
  return $\hat{z}^R$
end procedure

Figure 4-2: sFFT3.0 [40] runs in $O(S \log N)$-time. It is fast in theory and in practice, but faces two limitations as described in Section 4.1.2. Firstly, mode collision can create modes whose coefficients are of magnitude between 0 and 1/2. These spurious modes are unlikely to be found in subsequent iterations. Secondly, the mode identification is very unstable to noise. The parameters κ_r, δ controls how well (4.2) is approximated and will be covered later in Section 4.4.

4.4. Nevertheless, to simplify the explanation of sFFT3.0, we shall assume (4.2) holds exactly for the rest of Section 4.1.1.

Translation in time corresponds to modulation in frequency. Thus, the binning of signal y^τ, which is y translated by τ, yields another set of B_r bin coefficients: for $b \in [B_r]$,

$$Y_\tau^b := \sum_{k \in h^{-1}(b)} \hat{y}^\tau_{\varphi(k)} = \sum_{k \in h^{-1}(b)} \hat{y}_{\varphi(k)} e^{2\pi i \varphi(k)\tau/N} = \sum_{k \in h^{-1}(b)} \hat{x}^r_k e^{2\pi i \gamma k/N} e^{2\pi i \varphi(k)\tau/N}. \quad (4.2)$$

Treat Y_τ^b as the τ-th time-sample of a signal $Y^b \in \mathbb{C}^N$ where Y^b is the transformed signal y with frequency components outside $[\frac{b}{B}, \frac{b+1}{B})$ zeroed out. We like to call Y^b a *sub-signal.* To reiterate, the set of bin coefficients $\left\{Y_\tau^b : b \in [B_r], \tau \in \mathcal{H}\right\}$ are simply the B_r sub-signals sampled at $\mathcal{H}$.

We say mode k_* of signal x is *isolated* if its bin contains no heavy modes other than k_*, i.e., $(h^{-1}(h(k_*)) \cap \Lambda(x)) \backslash \{k_*\} = \phi$. The objective of binning is to identify isolated heavy modes. Suppose bin b contains an isolated heavy mode and there is *no noise*, i.e., $\|x_{\overline{\Lambda}}\| = 0$. Say $h^{-1}(b) = \{k_*\}$ and $\varphi(k_*) = k_0$. Then our sub-signal Y^b is a *pure sinusoid*: for any $\tau \in [N]$, $Y_\tau^b = \hat{y}_{k_0} e^{2\pi i \xi_0 \tau}$ where $\xi_0 = k_0/N$. It is easy to decipher a pure sinusoid. Observe that k_0, $\hat{y}_{k_0}$ can be obtained as

$$\xi_0 = \arg(Y_1^b/Y_0^b); \quad k_0 = \text{round}\left(\frac{N}{2\pi}\xi_0\right); \quad \hat{y}_{k_0} = Y_0^b. \quad (4.3)$$

Finally, undo the random transformation to obtain k_* and $\hat{x}_{k_*}$ as in Figure 4-2.

Next, consider why the number of heavy modes decays exponentially with good probability. Our argument differs slightly from [40]'s. Suppose we are at the beginning of iteration r. Let $\Lambda^r = \Lambda(x^r)$ be the set of heavy modes in the residual x^r. Assume for now that $|\Lambda^r| \leq S_r := Sa_S^r$ for some $0 < a_S < 1$. Instead of letting B_r be proportional to S_r like in [40], we let $B_r = Ba_B^r$ decay exponentially slower than S_r. Fix a heavy mode k_*. By Markov's inequality, the probability that k_* is not isolated

is bounded by $\frac{S_r - 1}{B_r} \leq \frac{S}{B}(a_S/a_B)^r$. By Markov's inequality,

$$\mathbb{P}\left(\text{no. of heavy modes not isolated} \geq \frac{a_S}{2}|\Lambda^r|\right) \leq \frac{2S}{a_S}(a_S/a_B)^r.$$

Assuming no noise and that our bin coefficients are ideal and satisfy (4.2), we see that an isolated heavy mode will be identified and eliminated in the residual, while a non-isolated heavy mode will at worst create a new heavy mode in the next iteration. Hence, with probability at least $1 - \frac{2S}{a_S B}(a_S/a_B)^r$, the number of heavy modes in the iteration $r+1$ does not exceed $2(a_S S_r/2) = a_S S_r = S_{r+1}$. By union bound over all iterations,

$$\mathbb{P}\left(|\Lambda^r| > S_r \text{ for some } r\right) \leq \frac{2S}{a_S B}\sum_{r=0}^{\infty}(a_S/a_B)^r = \frac{2S}{a_S B}\frac{1}{1 - a_S/a_B}.$$

Pick $a_B = 3/4$, $a_S = a_B/2$, and $B = 24S/a_B = 32S$ and conclude that with probability at least $2/3$, the number of heavy modes in x^r decay exponentially with r and is bounded by $S_r = S a_S^r$.

In practice, the bin coefficients are not ideal. Let k_* be an isolated heavy mode and $b = h(k_*)$. First, $\hat{x}_{k_*}$ may be highly attenuated before it is added to its sub-signal Y^b. Second, heavy modes landing outside bin b can contribute to the sub-signal Y^b and act as noise. Both of these imperfections of (4.2) can make the recovery of k_* by (4.3) unstable. Nonetheless, these problems can be mitigated with a proper choice of δ, κ_r. There are however two other problems of sFFT3.0 that are more serious and they are the focus of the next section.

4.1.2 Two limitations of sFFT3.0

The *first limitation* of sFFT3.0 is that its mode identification step is *nonrobust*. Fix a heavy mode k_*. Let $b = h(k_*)$ and $\Lambda^r = \Lambda(x^r)$. Continuing from (4.2), let the perturbation in Y_τ^b due to nonheavy modes be $\Delta Y_\tau^b := \sum_{k \in \overline{\Lambda^r} \cap h^{-1}(b)} \hat{x}_k^r e^{2\pi i \gamma k/N} e^{2\pi i k \tau/N}$.

Taking expectation with respect to the random permutation φ and integer γ, we have

$$\mathbb{E}^{\varphi}\mathbb{E}^{\gamma}\left|\Delta Y_{\tau}^{b}\right|^{2}=\mathbb{E}^{\varphi}\sum_{k\in\overline{\Lambda^{r}}\cap h^{-1}(b)}\left|\hat{x}_{k}^{r}\right|^{2}=\sum_{k\in\overline{\Lambda^{r}}}\left|\hat{x}_{k}\right|^{2}\mathbb{E}^{\varphi}\mathbb{1}_{\{k\in h^{-1}(b)\}}\leq\left\|\hat{x}_{\overline{\Lambda^{r}}}^{r}\right\|^{2}/B_{r}\quad(4.4)$$

Suppose mode k_* is isolated. Note that the perturbation in $\arg Y_{\tau}^{b}$ due to the perturbation in Y_{τ}^{b} is $O\left(\left|\Delta Y_{\tau}^{b}\right|/\left|\hat{x}_{k_*}\right|\right)=O\left(\left|\Delta Y_{\tau}^{b}\right|\right)$. For the rounding in (4.3) to correct ΔY_{τ}^{b} so that we can recover mode $k_0=\varphi(k_*)$, we need $\Delta Y_{\tau}^{b}=O\left(1/N\right)$ for each $\tau=0,1$. Now (4.4) suggests that to have a good chance of identifying the heavy modes, we need $B_r=\Omega\left(N^2\left\|\hat{x}_{\overline{\Lambda^{r}}}^{r}\right\|^{2}\right)$. Unless $\left\|\hat{x}_{\overline{\Lambda^{r}}}^{r}\right\|=O\left(1/N\right)$, we will need B_r to grow with a power of N which is undesirable. For example, if $\left\|\hat{x}_{\overline{\Lambda^{r}}}^{r}\right\|=\Omega\left(N^{-1/2}\right)$, we will need $B_r=\Omega\left(N\right)$, which means that sFFT3.0 runs in $O\left(N\log N\right)$ time and is no faster than FFT even in theory.

The way to fix this is to *identify* k_0 *bit by bit.* This idea is not new and has been employed in AAFFT, sFFT4.0, etc. In AAFFT, we identify the least significant bit of k_0, implicitly bitshift k_0 to the right and repeat. In sFFT4.0, we identify groups of bits at a time, starting from the most significant bits instead. We will use a simplified version of the mode identification procedure in sFFT4.0. The details are postponed to Section 4.2.3.

The *second limitation* of sFFT3.0 is that when two or more heavy modes land in a bin, i.e., mode collision, they may cancel one another partially and create a mode with coefficient 1/4 for instance. Such a mode will remain in the residual signal because sFFT3.0 processes a bin b only if $\left|Y_{0}^{b}\right|\geq 1/2$, and whenever this mode is isolated in a bin, its bin coefficient Y_{0}^{b} will have magnitude $< 1/2$. We call such nonheavy unidentifiable modes *ghost modes.*

It is tempting to fix this problem by reducing the threshold value 1/2 to a small value μ_r. As far as we are concerned, this modification alone does not solve the problem. The inherent difficulty is that if μ_r is too small, then too many bins with no heavy modes will be processed and too many spurious modes will be created. On the other hand, if μ_r is too big, we run into the same problem of ghost modes having too much energy.

Require: Chance of getting a good estimate must exceed $1/2$

procedure COEFFICIENTESTIMATIONLOOP($x \in \mathbb{C}^N, \hat{z}, B, \mathcal{L}$)

$R_e \leftarrow O(\log B)$, number of random shuffles

Create a table A of size $|\mathcal{L}| \times R_e$

for $i \leftarrow 0, 1, \ldots, R_e - 1$ **do**

Let $\varphi(k) = \alpha k + \beta$ be uniformly chosen permutation of $[N]$

Let γ be uniformly chosen from $[N]$ ▷ $\hat{y}_{\varphi(k)} = \hat{x}^r_k e^{2\pi i \gamma k/N}$

$\mathcal{H} \leftarrow \{0\}$

$Y' \leftarrow \texttt{BinInTime}(x, \alpha, \beta, \gamma, \mathcal{H}, B, \delta, \kappa)$

$Y'' \leftarrow \texttt{BinInFrequency}(\hat{z}^r, \alpha, \beta, \gamma, \mathcal{H}, B, \delta, \kappa)$

$Y \leftarrow Y' - Y''$

for $j = 0, \ldots, |\mathcal{L}| - 1$ **do**

$k_0 \leftarrow \varphi(k_*)$ where k_* is the j-th mode in $\mathcal{L}$

$b \leftarrow \lfloor \frac{k_0 B}{N} \rceil$

$A_{j,i} \leftarrow Y^b_0 e^{-2\pi i \gamma k_*/N}$

end for

end for

Create a list D of size $|\mathcal{L}|$

for $j = 0, \ldots, |\mathcal{L}| - 1$ **do**

$D_j \leftarrow \text{median}\{A_{j,i} : i \in [R]\}$

end for

return D

end procedure

Figure 4-3: Coefficient estimation loop used in AAFFT and sFFT4.0 requires us to bin the residual signal $O(\log B)$ times, which is computationally very expensive.

SFT algorithms such as AAFFT, sFFT4.0 fix this problem by *processing all bins*, adding up to B_r modes to a temporary list $\mathcal{L}$ and using a separate *coefficient estimation loop* to estimate the coefficients of *all modes* in $\mathcal{L}$ to the desired accuracy. After that, the largest $O(S_r)$ coefficients are kept. See Figure 4-3. We think that this coefficient estimation loop should not be used for the following reasons.

Firstly, if k_* is an isolated heavy mode, then the bin coefficient Y_0^b used for mode identification is most likely a good estimate of $\hat{y}_{\varphi(k_*)}$. It seems unnecessary to estimate its coefficient in a separate loop. The difficulty lies in distinguishing between a bin with an isolated mode and a bin with more than one heavy mode.

Secondly, binning is an *extremely costly operation* and the running time of many SFT algorithms is very much determined by the number of times binning is performed. As the coefficient estimation loop requires us to bin the residual signal $O(\log B_r)$ more times in iteration r, it will slow down the SFT algorithm considerably.

Thirdly, for the coefficient estimation loop to work, B_r has to be relatively large compared to S_r, which is *not optimal*. Taking the median of estimates only works if the probability that we get a good estimate of a mode coefficient per random shuffle happens with sufficiently high probability, say at least a 3/4 chance. That means the chance of mode collision has to be less than 1/4 and we need $B_r \gtrsim 4S_r$. However, the optimal B_r is S_r not $4S_r$ by the following heuristic argument. In practice, the modes of y appear to be fully randomly shuffled, and the chance that a mode is isolated is $(1-1/B_r)^{S_r-1} \simeq e^{-S_r/B_r}$. Suppose we fix $B_r = C_{\text{mul}} S_r$ for some $C_{\text{mul}} \geq 1$. Suppose $e^{-S_r/B_r} = e^{-1/C_{\text{mul}}}$ of the heavy modes in x^r is removed in iteration r. Then the total time taken by binning is proportional to

$$\sum_{r=0}^{\infty} B_r = \sum_{r=0}^{\infty} B\left(1-e^{-1/C_{\text{mul}}}\right)^r = SC_{\text{mul}}e^{1/C_{\text{mul}}}. \tag{4.5}$$

The above is minimized when $C_{\text{mul}} = 1$, and whenever we use a larger C_{mul}, our algorithm will be slowed down by roughly a factor of C_{mul}. That is not all. The chance that a mode cannot be recovered because it lands too far away from the center of its bin [40] is κ_r. For the median-taking to work, we need $\kappa_r \leq 1/4$. The

time taken by binning scales with $1/\kappa_r$ which further slows down the algorithm by a factor of 4.

Lastly, out of B_r bins where B_r is unreasonably large compared to S_r, at most S_r of them contain useful information. It is wasteful to process every single bin, estimate the coefficients of up to B_r modes in the list $\mathcal{L}$ and then discard most of these coefficients at the end of the iteration.

In the next section, we present our algorithm MPFFT and describe how it fixes the second limitation of sFFT3.0 in a more efficient way.

4.1.3 MPFFT and main results

See Figure 4-4 for the pseudocode of MPFFT. Compare MPFFT with sFFT3.0 in Figure 4-2. The main difference is that in each bin b, we run the "multiscale matrix pencil method" on the sub-signal Y^b and skip to the next bin if the subroutine returns a $\mu_{\max}$ that is too large. This is the mode collision test. The basic idea is that if there is more than one heavy mode in the bin, then $\mu_{\max}$ is unlikely to be small and we will not attempt to recover any mode in the bin. In this way, we avoid creating ghost modes and overcome the second limitation of sFFT3.0 without resorting to the costly coefficient estimation loop in AAFFT.

The matrix pencil method [45] is a classical method for spectral estimation in signal processing. Given a signal $X' = X + \Delta X$ where $X_j = \sum_{q\in[Q]} c_q e^{2\pi i j \xi_q}$ and ΔX_j is noise, the matrix pencil method aims to recover the frequencies ξ_q's and the coefficients c_q's from $2J-1$ samples $(X'_j)_{|j|\le J-1}$ with $J \ge Q+1$. We do not apply the matrix pencil method to the input signal $x \in \mathbb{C}^N$ with $Q = S$ because too many samples or a large J will be needed to resolve the frequencies to a precision of $1/N$. This will be elaborated in Section 4.2.3. Instead, the matrix pencil method is applied to a few frequency-dilated copies of the *sub-signal* Y^b with $Q = 1$ so as to recover the permuted mode location $k_0 = \varphi(k_*)$. More precisely, from `MatrixPencilMultiscale` in Figure 4-6, we see that in MPFFT, the matrix pencil method is applied to $(Y^b_{j2^{\ell M}B_r})_{|j|\le J-1}$ for some ℓ to recover the ℓ-th group of M bits of the frequency $\frac{k_0 B_r}{N} \% 1$ where M is an input parameter to `MPFFT`. For example, if $\ell = 3$ and $M = 2$, then the matrix pencil

procedure MPFFT($x \in \mathbb{C}^N, S, J, M, \mathcal{E}, \varepsilon$)
  $\hat{z}^0 \leftarrow 0$
  $\rho_0 \leftarrow 0$
  for $r \leftarrow 0, 1, \ldots, R-1 = O(\log S)$ **do**
    Let $\varphi(k) = \alpha k + \beta$ be uniformly chosen permutation of $[N]$
    Let γ be uniformly chosen from $[N]$ ▷ $\hat{y}_{\varphi(k)} = \hat{x}^r_k e^{2\pi i \gamma k/N}$
    $B_r \leftarrow B(r+1)^{-2(1+p)}$ where $p = 0.01$
    $S_r \leftarrow Se^{-r}$
    $\kappa_r \leftarrow \kappa(r+1)^{-(1+p)}$
    $L_r \leftarrow \lfloor \log_{2^M}(N/B_r) \rfloor + 1 = O\left(\frac{1}{M} \log \frac{N}{B_r}\right)$
    $\mathcal{H} \leftarrow \{j 2^{M\ell} B_r : |j| \leq J-1, \ell \in [L_r]\}$
    $Y' \leftarrow$ BinInTime$(x, \alpha, \beta, \gamma, \mathcal{H}, B_r, \delta, \kappa_r)$
    $Y'' \leftarrow$ BinInFrequency$(\hat{z}^r, \alpha, \beta, \gamma, \mathcal{H}, B_r, \delta, \kappa_r)$
    $Y \leftarrow Y' - Y''$ ▷ Obtain B_r sub-signals Y^b
    for $b \in [B_r]$ such that $\left|Y^b_0\right| \geq 1 - \rho_r - 2\sqrt{f_r \mathcal{E}_r / B_r}$ **do**
      Identify one mode k_0 using $\{Y^b_\tau : \tau \in \mathcal{H}\}$:
        $(\xi_0, \mu_{\max}) \leftarrow$ MatrixPencilMultiscale$(L_r, J, M, (Y^b_{j2^{M\ell}B_r})_{|j| \leq J-1, \ell \in [L_r]})$
        $k_0 \leftarrow \text{round}\left(N\left(\frac{b+\xi_0}{B_r}\right)\right)$
      if $\mu^2_{\max} > C_{\text{mp}} f_r \mathcal{E}_r / B_r$ where C_{mp} is defined in (4.6) **then** mp
        continue to next bin
      if $\left|\frac{k_0}{N} - \frac{b+1/2}{B_r}\right| \geq \frac{1-\kappa}{2B}$ **then**
        continue to next bin
      Estimate $\hat{y}_{k_0}$ as $\hat{y}'_{k_0} \leftarrow Y^b_0$
      $k_* \leftarrow \varphi^{-1}(k_0)$
      Update our solution by $\hat{z}^{r+1}_{k_*} \leftarrow \hat{z}^r_{k_*} + \hat{y}'_{k_0} e^{-2\pi i \gamma k_*/N}$
    end for
    $\rho_{r+1} \leftarrow \rho_r + 4\sqrt{f_r \mathcal{E}_r / B_r}$
    $\mathcal{E}_{r+1} \leftarrow \mathcal{E}_r(1 + 4 f_r S_r / B_r)$
  end for
  return $\hat{z}^R$
end procedure

Figure 4-4: First form of MPFFT is analyzed in Section 4.5.

method is used to recover the 7-th and 8-th most significant bits of $\frac{k_0 B_r}{N} \% 1$.

The user needs to supply $\mathcal{E}$ as an upper bound on the energy of the nonheavy modes in x, i.e., $\mathcal{E} \geq \|\hat{x}_{\overline{\Lambda}}\|^2$. The parameter ε is a precision parameter and `MPFFT` aims to return a $\hat{z}^R$ such that $\left\|\hat{z}^R - \hat{x}\right\|^2 \leq (1+\varepsilon)\mathcal{E}$. Numerical experiments suggest that MPFFT runs in $\widetilde{O}(S)$ time, but developing a rigorous proof is tricky for a few reasons.

Firstly, without using the coefficient estimation loop in Figure 4-3 to boost the success probability of estimating a mode coefficient well, we cannot use a union bound to say that up to B_r modes are estimated well, which is a crucial step in the analysis of sFFT4.0 and AAFFT. In practice, the mode collision detector serves a similar purpose: it ensures that for all bins, we will not make too much error in the estimation of any mode coefficient.

Secondly, despite the usefulness of the mode collision detector in practice, providing theoretical guarantees of its effectiveness seems difficult unless some assumptions are made about the signal model. In Section 4.3, we assume that the frequencies ξ_q's are *independently* and *uniformly distributed* and establish some lower bounds on $\mu_{\max}$. This assumption seems reasonable for the randomly shuffled modes in the context of MPFFT, but it unfortunately does not hold formally. The reason is that noise in the second iteration can arise from errors in the first iteration — even if noise in the first iteration is Gaussian, it is no longer Gaussian in subsequent iterations — and the random shuffling of these subdominant modes is only pairwise independent, not fully independent.

To bypass these technical difficulties so that we can provide a formal analysis of MPFFT, we make the following assumption about the matrix pencil method:

Assumption 4.1.2. *There exists a* $0 < C_{mp} < 1$ *such that the following holds. Let* $X'_j = \sum_{s\in[P]} c_s e^{2\pi i j \xi_s}$ *where* $|c_0| \geq \ldots \geq |c_{P-1}|$. *When* `MatrixPencil` *in Figure 4-5 is run on* $(X'_j)_{|j|\leq J-1}$ *with* $Q = 1$, *it will return a* μ *that satisfies*

$$\mu^2 \geq C_{mp} \left| \sum_{s=1}^{P-1} c_s \right|^2 + C_{mp} \sum_{s=1}^{P-1} |c_s|^2 . \tag{4.6}$$

The above assumption is motivated by some theoretical results in Section 4.3 on the matrix pencil method, which will be briefly discussed after we present the main result. In addition to Assumption 4.1.2, we also assume that the perturbation in the single frequency obtained by the matrix pencil method due to noise can be obtained by first order approximation. This is Assumption 4.2.2. It has an impact in the proof of the main result only when the perturbation in the input to the matrix pencil method due to noise is small, which means that first order perturbation theory is well-justified. More on Assumption 4.2.2 can be found at the end of Section 4.5.1. Now, we are ready to present the main result about MPFFT.

Theorem 4.1.3 (Main result). *Assume* $\|\hat{x}_\Lambda\|^2 \leq \mathcal{E}$ *and for some* $c > 0$, $S \log^c S = O(N)$. *Let* $0 < \varepsilon < 1$. *Suppose* $\delta = N^{-\Theta(1)}$ *is sufficiently small as required by Lemma 4.5.1. Suppose* $\varepsilon, \mathcal{E}$ *are small such that* $(\varepsilon\mathcal{E})^{1/2} \leq \frac{1}{8} \frac{S}{\log^{2.5+1.5p} S}$ *where* $p = 0.01$. *Pick* $f = \Theta(\log \frac{N}{S})$, $\kappa = \Theta(1/M)$ *and* $B = \Theta(\frac{S}{\varepsilon} \log \frac{N}{S})$ *such that* $B \geq \frac{100fS}{\varepsilon}$. *Under Assumption 4.1.2 and Assumption 4.2.2, we have that with probability at least*

$$1 - O\left(\mathcal{E} \log S + \frac{1}{\log \frac{N}{S}} + \frac{1}{M} + \left(\frac{2^{2M} \mathcal{E} \varepsilon \log^{2(1+p)+2} S}{J^2 S}\right)^{1/3} \frac{\log \frac{N}{S}}{M} + \frac{2^M \log \frac{N}{S} \log S}{MN}\right),$$

`MPFFT` *in Figure 4-4 runs in*

$$O\left(\frac{S}{M\varepsilon} \log^2 \frac{N}{S} \left(J^3 + \log N\right)\right)$$

time and outputs a z^R *such that* $\left\|\hat{z}^R - \hat{x}\right\|^2 \leq (1+\varepsilon)\mathcal{E}$ *and* $\left|\operatorname{supp} \hat{z}^R\right| = O(S)$.

Typically, we pick a small M such that $N \gg 2^M$ and the $\frac{2^M \log \frac{N}{S} \log S}{MN}$ term in the failure probability is negligible. Moreover, J is usually very small such that the bound on the running time reads as $O\left(\frac{S}{M\varepsilon} \log^2 \frac{N}{S} \log N\right)$. Note that p can be arbitrarily close to 0 but this will increase the constants in the bounds on the running time and failure probability. The proof of Theorem 4.1.3 can be found in Section 4.5.

Now, let us motivate Assumption 4.1.2 by giving an overview of the results in Section 4.3. Assume that the ξ_s's in Assumption 4.1.2 are independently, uniformly

chosen from $[0,1)$. Refer to ξ_0 as the dominant mode and $\sum_{s=1}^{P-1}|c_s|^2$ as the subdominant energy. Proposition 4.3.1 states that it is unlikely that μ is much bigger than $\left|\sum_{s=1}^{P-1} c_s\right|^2+\sum_{s=1}^{P-1}|c_s|^2$, so the right hand side of (4.6) is at the least not unreasonably large.

Theorem 4.3.3 says that if the total energy is comparable to $|c_0|^2$, which is inspired by the case where there are several heavy modes in the bin with roughly the same magnitude, then with high probability, $\mu^2 \gtrsim \sum_{s\in[P]}|c_s|^2$. A similar result is Corollary 4.3.7. It says that if the subdominant energy is comparable to $|c_1|^2$, which is inspired by the case where there is an isolated heavy mode in the bin with nonheavy modes of roughly the same magnitude, then $\mu^2 \gtrsim \left|\sum_{s=1}^{P-1} c_s\right|^2+\sum_{s=1}^{P-1}|c_s|^2$.

Theorem 4.3.9 is of a different nature compared to Theorem 4.3.3 and Corollary 4.3.7. It says that if there are T heavy modes in the bin and $T^2 \lesssim J$, then $\mu^2 \gtrsim T-1$ with good probability. While this lower bound is weaker than $\sum_{s=1}^{P-1}|c_s|^2$, it hints at why the collision detector is good at detecting the presence of a few heavy modes. The $T=2$ case is especially important because it is the most common case encountered by MPFFT as mentioned in Section 4.3.5. For this case, Proposition 4.3.10 tells us that for some scale level ℓ, the μ^2 returned by the matrix pencil method on the input $(Y^b_{j2^M\ell B_r})_{|j|\le J-1}$ must be $\gtrsim |c_1|\left(1-\frac{2^M+1}{2J}\right)$. Note that this is a deterministic result.

4.2 Matrix pencil method

We first present the matrix pencil method [45, 46], then use it to identify just one mode. Then we discuss in Section 4.3 how to detect whether the subdominant modes are too energetic.

4.2.1 Introduction

Suppose we have a signal with Q modes, i.e., $X_j=\sum_{q\in[Q]} c_q\omega_q^j$ where $\omega_q=e^{2\pi i\xi_q}$ and $\xi_q\in[0,1)$. This is an undamped signal as $|\omega_q|=1$. Let $J\ge Q+1$. Our objective is to recover the frequencies $\xi_0,\dots,\xi_{Q-1}$ and the coefficients $c_0,\dots,c_{Q-1}$ from $2J-1$ noisy measurements $X'=(X'_{-(J-1)},\dots,X'_{J-1})^T$ where each $X'_j=X_j+\Delta X_j$ and ΔX_j

is a random perturbation. Define the *Toeplitz* matrix on X as

$$\mathbb{T}X = \frac{1}{J}\begin{pmatrix} X_0 & X_{-1} & X_{-2} & \dots & X_{-J+1} \\ X_1 & X_0 & X_{-1} & \dots & X_{-J+2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ X_{J-1} & X_{J-2} & X_{J-3} & \dots & X_0 \end{pmatrix}.$$

The $1/J$ normalization factor is non-standard. We use it out of convenience in Section 4.3, a major part of the paper. Consider the matrices $A', A, \Delta A \in \mathbb{C}^{J\times J}$:

$$A' = \mathbb{T}X'; \quad A = \mathbb{T}X; \quad \Delta A = \mathbb{T}(\Delta X). \tag{4.7}$$

Let A_1 be A with the rightmost column removed and A_2 be A with the leftmost column removed. Let $\boldsymbol{\lambda}(A_2, A_1)$ be the set of *generalized eigenvalues* of $A_2 - \lambda A_1$. It is equal to $\boldsymbol{\lambda}(A_1^+ A_2)$, the set of *nonzero*[2] ordinary eigenvalues of $A_1^+ A_2$.

For any T, denote $\mathbb{v}_J(\xi) = (1, e^{2\pi i\xi}, \dots, e^{2\pi i(J-1)\xi})^T$. Define $U_T \in \mathbb{C}^{T\times Q}$ as

$$U_T = (\mathbb{v}_T(\xi_0), \dots, \mathbb{v}_T(\xi_{Q-1})). \tag{4.8}$$

Note that U_T's columns are in general not orthogonal. Let $C = \mathrm{diag}(c_q)_{q\in[Q]}$. Observe that $\mathrm{rank}(A) = Q$ and A has a *Vandermonde decomposition.* We can write $A = \frac{1}{J}U_J C U_J^*$, $A_1 = \frac{1}{J}U_J C U_{J-1}^*$, $A_2 = \frac{1}{J}U_J C\,\mathrm{diag}(\omega)^* U_{J-1}^*$. This suggests that $\mathrm{range}(A_1) = \mathrm{range}(A_2)$ and $\mathrm{range}(A_1^*) = \mathrm{range}(A_2^*)$. Furthermore, the generalized eigenvalues of $A_2 - \lambda A_1$ are exactly the ω_q's we seek, conjugated, i.e., $\overline{\boldsymbol{\lambda}(A_2, A_1)} = \{\omega_0, \dots, \omega_{Q-1}\}$ because $A_2 - \lambda A_1$ has a nullspace whenever $\lambda = \overline{\omega_q}$ for some q:

$$A_2 - \lambda A_1 = \frac{1}{J}U_J C(\mathrm{diag}(\omega)^* - \lambda I)U_{J-1}^*.$$

Once ω is found, we can solve a Vandermonde system to find the coefficients $c_0, \dots, c_{Q-1}$.

Now, consider the noisy version of A, i.e., A'. Let A_1' be A' with the rightmost column removed and A_2' be A' with the leftmost column removed. Let $\boldsymbol{\lambda}(A_2', A_1') =$

[2] $A_1^+ A_2$ contains Q nonzero eigenvalues and $J - 1 - Q$ zero eigenvalues.

procedure MATRIXPENCIL$(J, Q, (X'_j)_{|j|\le J-1})$
  Form the matrix $A' = \mathbb{T}(X'_j)_{|j|\le J-1} \in \mathbb{C}^{J\times J}$ according to (4.7)
  Compute the SVD of A'
  $\mu^2 \leftarrow \sum_{j=Q+1}^{J} \sigma_j^2(A') = \|A'\|_F^2 - \|A'\|^2$ ▷ Used for collision detection
  Let $V \in \mathbb{C}^{J\times Q}$ be the top Q *right* singular vectors of A'
  Let $V_1 \in \mathbb{C}^{(J-1)\times Q}$ be V with its bottom row removed
  Let $V_2 \in \mathbb{C}^{(J-1)\times Q}$ be V with its top row removed
  $\omega' \leftarrow \boldsymbol{\lambda}(V_2, V_1) = \boldsymbol{\lambda}(V_1^+ V_2)$
  Obtain Q frequencies by $\xi'_q \leftarrow \frac{1}{2\pi} \arg \omega'_q$ for $q \in [Q]$
  return $(\xi'_0, \ldots, \xi'_{Q-1})^T, \mu > 0$
end procedure

Figure 4-5: Matrix pencil method.

$\{\omega'_0, \ldots, \omega'_{Q-1}\}$ be the generalized eigenvalues of $A'_2 - \lambda A'_1$. Let $\Delta\omega_q = \omega'_q - \omega_q$. The hope is that $\Delta\omega$ is small and we can estimate the frequencies $\xi_0, \ldots, \xi_{Q-1}$ by computing $\boldsymbol{\lambda}(A'_2, A'_1)$. In Section 4.2.2, we will bound $\Delta\omega$ in terms of ΔX to first order accuracy.

One way to obtain $\boldsymbol{\lambda}(A'_2, A'_1)$ is to first compute the pseudoinverse $(A'_1)^+$ and then the ordinary eigenvalues of $(A'_1)^+ A'_2$. The problem is that due to the perturbation ΔX, A' is very likely to have rank $> Q$. To avoid inverting the components corresponding to these small spurious singular values, we *truncate* A' to rank Q using SVD, obtain A'_1, A'_2 as column subsets of the truncated A', and then compute the eigenvalues of $(A'_1)^+ A'_2$.

Suppose the SVD of the truncated A' is $\tilde{V}\Sigma V^*$ where $\Sigma \in \mathbb{R}^{Q\times Q}$, $V \in \mathbb{C}^{J\times Q}$. Let V_1 be V with the bottom row removed and V_2 be V with the top row removed. Then $\boldsymbol{\lambda}(A'_2, A'_1) = \boldsymbol{\lambda}((A'_1)^+ A'_2) = \boldsymbol{\lambda}(V_1^{*+} V_2^*) = \boldsymbol{\lambda}(V_2^* V_1^{*+}) = \overline{\boldsymbol{\lambda}(V_1^+ V_2)} = \overline{\boldsymbol{\lambda}(V_2, V_1)}$. Hence, it suffices to compute $\boldsymbol{\lambda}(V_2, V_1)$ and there is no need to compute A'_1, A'_2. See `MatrixPencil` in Figure 4-5 for the pseudocode.

In `MatrixPencil` in Figure 4-5, we compute the quantity $\mu^2 = \sum_{q=Q+1}^{J} \sigma_q^2(A')$. This will be used in MPFFT to decide whether there is too much noise energy in the bin. We will discuss its role in mode collision detection in Section 4.3.

4.2.2 Identifying one mode and first order perturbations

For the rest of Section 4.2, we focus on the case where there is only one mode, i.e., $Q = 1$.

In our algorithm MPFFT, we apply the matrix pencil method to detect one mode in each sub-signal. Fortunately, it is much easier to study the perturbation in ω due to the perturbation in X for the case where there is only one mode. The bounds we present here can be obtained by adapting the arguments from [45]. The difference is that they take samples $X_0, \ldots, X_{2J-1}$ whereas we take $X_{-(J-1)}, \ldots, X_{J-1}$. This seems like a trivial change as we can always modulate the coefficients c_q, but it turns out that our proof looks simpler. In particular, the variable Γ in (4.10) takes a much simpler form than its counterpart in [45].

Proposition 4.2.1. *Let* $X'_j = c_0\omega_0^j + \Delta X_j$ *where* $\omega_0 = e^{2\pi i\xi_0}$ *and* $\xi_0 \in [0, 1)$. *Run* `MatrixPencil` *in Figure 4-5 with* $Q = 1$ *on* $(X'_j)_{|j|\le J-1}$ *which returns* ξ'_0. *Let* $\Delta\xi_0 = \xi'_0 - \xi_0$. To first order,

$$|\Delta\xi_0|^2 \le \frac{\|\Delta X\|^2}{\pi^2 |c_0|^2 J(J-1)^2}.$$

Assumption 4.2.2. *In this paper, the entries of* ΔX *are very small and it is reasonable to assume that* $\Delta\xi_0$ *is indeed bounded by* $\frac{\|\Delta X\|^2}{\pi^2|c_0|^2 J(J-1)^2}$ *even though the inequality is only first order accurate.*

See the end of Section 4.5.1 for a more detailed justification of Assumption 4.2.2. Let us proceed with the proof of Proposition 4.2.1.

Proof of Proposition 4.2.1. Recall this fact about first order perturbations in generalized eigenvalues. Suppose $(A_2 - \lambda A_1)v = 0$ and $u^*(A_2 - \lambda A_1) = 0$. Then to first order,

$$\Delta\lambda = \frac{u^*(\Delta A_2 - \lambda\Delta A_1)v}{u^* A_1 v}. \tag{4.9}$$

In our case, $\lambda = \overline{\omega_0}$, $u = U_J = \mathbb{v}_J(\xi_0)$ defined in (4.8), $v = U_{J-1} = \mathbb{v}_{J-1}(\xi_0)$ and $u^*A_1v = c_0(J-1)$. Observe that $(\Delta A_2 - \lambda\Delta A_1)v$ is *linear* in Δx. Let $D_j \in \mathbb{R}^{J\times(2J-1)}$ such that its j-th diagonal is 1. For example, the topleft 3×3 corners of D_1, D_{-1} are

$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$ respectively. With some algebra, we find that

$$(\Delta A_2 - \lambda \Delta A_1)v = \frac{1}{J}\Gamma(\Delta X) \text{ where } \Gamma = -\lambda D_{J-1} + \lambda^{-J+2} D_0. \tag{4.10}$$

By (4.9) and (4.10), we know $\Delta\lambda = \frac{1}{|c_0|J(J-1)} U_J^* \Gamma(\Delta X)$. Thus,

$$|\Delta\omega_0|^2 = \frac{1}{|c_0|^2 J^2(J-1)^2} U_J^* \Gamma(\Delta X)(\Delta X)^* \Gamma^T U_J \leq \frac{1}{|c_0|^2 J^2(J-1)^2} \|\Delta X\|^2 \|\Gamma\|^2 \|U_J\|^2.$$

Apply the fact that $\|\Gamma\| \leq \|D_J\| + \|D_{2J-1}\| \leq 2$ and $\|U_J\|^2 = J$. Note that to first order, $|\Delta\xi_0| = \frac{1}{2\pi}|\Delta\omega_0|$. □

If ΔX is random while c_0, ξ_0 are fixed, then the proof from Proposition 4.2.1 also implies that $\mathbb{E}|\Delta\xi_0|^2 \leq \frac{\|\mathbb{E}(\Delta X)(\Delta X)^*\|}{\pi^2 |c_0|^2 J(J-1)^2}$, which is a tighter result than Proposition 4.2.1. However, in MPFFT, the entries of ΔX are highly correlated and we do not lose much when we bound $\|\mathbb{E}(\Delta X)(\Delta X)^*\|$ by $\mathbb{E}\|\Delta X\|^2$.

In our application, the frequency ξ_0 lies on a grid and by rounding off to the nearest grid point, there is a chance of obtaining the exact ξ_0. If we do obtain the correct ξ_0 and estimate the coefficient c_0 as $c_0' = \frac{1}{|\mathcal{H}|}\sum_{\tau\in\mathcal{H}} X_\tau' \omega_0^{-\tau}$, then the estimation error can be bounded as follows.

Proposition 4.2.3. *Let $X_j' = c_0\omega_0^j + \Delta X_j$ where $\omega_0 = e^{2\pi i\xi_0}$ and $\xi_0 \in [0,1)$. Assume the same set-up as Proposition 4.2.1. Suppose we sample the signal at a set of points $\mathcal{H}$ and estimate c_0 as $c_0' = \frac{1}{|\mathcal{H}|}\sum_{\tau\in\mathcal{H}} X_\tau' \omega_0^{-\tau}$. Then $|c_0' - c_0|^2 \leq \frac{1}{|\mathcal{H}|}\sum_{\tau\in\mathcal{H}} |\Delta X_\tau|^2$.*

Proof. Let $F = \text{diag}(\omega_0^\tau)_{\tau\in\mathcal{H}}$ and $z = (1,\ldots,1)^T \in \mathbb{C}^{|\mathcal{H}|\times 1}$. Then

$$c_0' - c_0 = \frac{1}{|\mathcal{H}|}\sum_{\tau\in\mathcal{H}} \Delta X_\tau \omega_0^\tau = \frac{1}{|\mathcal{H}|} z^T F(\Delta X).$$

Therefore, $|c_0' - c_0|^2 = \frac{1}{|\mathcal{H}|^2}\left|z^T F(\Delta X)(\Delta X)^* F^* z\right| \leq \frac{1}{|\mathcal{H}|^2}\|z\|^2\|F\|^2\|\Delta X\|^2$. Apply the fact that $\|z\|^2 = |\mathcal{H}|$ and $\|F\| = 1$ and we are done. □

The above proof also implies the tighter bound $\mathbb{E}\,|c_0' - c_0|^2 \le \frac{1}{|\mathcal{H}|}\,\|\mathbb{E}(\Delta X)(\Delta X)^*\|$.

4.2.3 Multiscale matrix pencil method

Suppose our signal is $X_\tau' = X_\tau + \Delta X_\tau$ where $X_\tau = c_0 e^{2\pi i \tau k_0/N}$ and $k_0 \in [N]$ is what we want to recover exactly. Suppose the frequency space $[0,1)$ is split into B bins and k_0 is in bin b, i.e., $\frac{k_0}{N} \in [\frac{b}{B}, \frac{b+1}{B})$. Define $\xi_0 = \left(\frac{k_0 B}{N}\right) \% 1 = \frac{(k_0 B)\%N}{N}$. Apply `MatrixPencil` in Figure 4-5 with $Q = 1$ to $(X'_{jB})_{|j|\le J-1}$ and obtain ξ_0' as an estimate of ξ_0. Since $k_0 = \frac{N}{B}(b + \xi_0)$, we shall estimate k_0 as

$$k_0' = \text{round}\left(\frac{N}{B}(b + \xi_0')\right) \in [N]. \tag{4.11}$$

Unfortunately, this method is unstable for large N. This is related to the first limitation of sFFT3.0 in Section 4.1.2. The reason is that Proposition 4.2.1 suggests that $\Delta\xi_0$ may be on the order of $\frac{\|\Delta X\|}{\pi J^{3/2}}$. For the rounding in (4.11) to correct this perturbation, we need $\Delta\xi_0 \le \frac{B}{2N}$. This means we need $J = \Omega\left(\frac{N\|\Delta X\|}{B}\right)^{2/3}$ which grows too fast with N.

It is not surprising that a direct application of the matrix pencil method does not work well. If we can only access the first few time samples, there is no hope of distinguishing between two pure sinusoids with *very close frequencies*. Say $\Delta\xi_0 = \frac{B}{N}$. Then for any small j, $\left|e^{2\pi i j(\xi_0+\Delta\xi_0)} - e^{2\pi i j\xi_0}\right| = |2\sin(2\pi(\Delta\xi_0)j/2)| \simeq 2\pi(\Delta\xi_0)j$ which is very small. But if we can skip the earlier samples and jump to $j \simeq \frac{1}{2\Delta\xi_0} = \frac{N}{2B}$, then $|2\sin(2\pi(\Delta\xi_0)j/2)| \simeq 2$ and we would have noticed the difference between a signal with frequency ξ_0 and a signal with frequency $\xi_0 + \Delta\xi_0$.

In our application, we do have access to any time samples. Exploit this by uncovering ξ_0 M bits at a time for some small $M \ge 1$. For example, if $M = 1$, we will be solving a *low resolution* problem of whether $2^0\xi_0, 2^1\xi_0, 2^2\xi_0, \ldots$ is in $[0, \frac{1}{2})$ or $[\frac{1}{2}, 1)$. Consequently, we can tolerate much larger errors in estimating $2^0\xi_0, 2^1\xi_0, 2^2\xi_0$ and so on.

Suppose $\xi_0 = 0.d_0 d_1 \ldots$ in base 2^M where each $d_\ell \in [2^M]$. Define

$$\xi_0^\ell = (2^{\ell M}\xi_0)\,\%1 = \left(\frac{k_0 2^{\ell M} B}{N}\right)\%1 = \frac{(k_0 2^{\ell M} B)\%N}{N} = 0.d_\ell d_{\ell+1}\ldots \text{ in base } 2^M.$$

As $d_\ell = \lfloor \xi_0^\ell 2^M \rfloor$, we shall run `MatrixPencil` in Figure 4-5 on $(X'_{j2^{\ell M}B})_{|j|\le J-1}$ which returns $\xi_0'^\ell$ as an estimate of ξ_0^ℓ, and then estimate d_ℓ as $d'_\ell = \lfloor \xi_0'^\ell 2^M \rfloor$. Notice that to get the less significant bits of ξ_0, we sample X much further in time as an earlier paragraph suggested.

Next, consider how to calculate ξ_0 if we manage to compute $d_0, \ldots, d_{L-1}$. Let $\xi_L = 0.d_0 \ldots d_{L-1}$ and $\xi_R = \xi_L + 2^{-LM}$. Recall that $\xi_0 = 0.d_0 \ldots d_{L-1} d_L \ldots \in [\xi_L, \xi_R)$. If $2^{-LM} < \frac{B}{N}$, or equivalently, $LM > \log_2 \frac{N}{B}$, then there is at most one integer k' such that $k' \in [\frac{N}{B}(b+\xi_L), \frac{N}{B}(b+\xi_R))$, and it must be that $k' = k_0$. This implies that whenever we correctly identify the first L digits of ξ_0 in base 2^M and $LM > \log_2 \frac{N}{B}$, we will be able to obtain k_0 as $\lceil \frac{N}{B}(b+\xi_L) \rceil = \lfloor \frac{N}{B}(b+\xi_R) \rfloor$.

Here is another way to estimate k_0 which we find more intuitive. Let $\xi'_0 = \frac{1}{2}(\xi_L + \xi_R)$ be an estimate of ξ_0. Since $\xi_R - \xi_L = 2^{-LM}$ and $\xi_0 \in [\xi_L, \xi_R)$, we have $|\xi'_0 - \xi_0| < 2^{-LM-1} < \frac{N}{2B}$. Consequently, k_0 can be recovered by the rounding in (4.11).

Refer to Figure 4-6 for the pseudocode. Notice that the algorithm does not require knowledge of N or B and can be applied to any discrete signal with a dominant frequency $\xi_0 \in \mathbb{R}$. It returns ξ'_0 as an estimate of ξ_0 and if our parameters are reasonably chosen, we can expect to recover the first LM bits of ξ_0. We leave it to the caller to estimate k_0 from ξ'_0 using (4.11). Note that we use integer arithmetic as much as possible to avoid floating point errors. Finally, the quantity $\mu^2_{\max}$ is used to determine if there is too much noise in the signal and will be analyzed in Section 4.3.

The point of finding ξ_0 M bits at a time for a small M is that we can tolerate a larger $\Delta\xi_0^\ell$ for each $\ell \in [L]$. For example, if $M = 1$ and $\xi_0^\ell = \frac{1}{5}$, then $d'_\ell = \lfloor (\xi_0^\ell + \Delta\xi_0^\ell)\cdot 2 \rfloor$ is equal to the correct d_ℓ for any $|\Delta\xi_0^\ell| \le \min\{\frac{1}{5}, \frac{1}{2} - \frac{1}{5}\} = \frac{1}{5}$. Let us formalize this argument.

Proposition 4.2.4. *Suppose* ξ_0 *is* uniformly distributed *in* $[0,1)$. *Let* $X'_j = X_j + \Delta X_j$ *where* $X_j = c_0 e^{2\pi i j \xi_0}$ *and* ΔX_j *is random and not necessarily independent from* ξ_0.

procedure MATRIXPENCILMULTISCALE$(L, J, M, (X'_{j2^{\ell M}})_{|j|\le J-1,\ell\in[L]})$
  $s \leftarrow 0$
  for each scale level $\ell \in [L]$ **do**
    $(\xi_0'^{\ell}, \mu_\ell) \leftarrow$ MatrixPencil$(J, 1, (X'_{j2^{\ell M}})_{|j|\le J-1})$ ▷ $d'_\ell = \lfloor \xi_0'^{\ell} 2^M \rfloor$
    $s \leftarrow s2^M + \lfloor \xi_0'^{\ell} 2^M \rfloor$ ▷ s is an integer
  end for
  $\xi'_0 \leftarrow 2^{-LM-1}(2s+1)$ ▷ $2^{-LM} s = \xi_L = 0.d'_0 \dots d'_{L-1}$ in base 2^M
  $\mu_{\max} \leftarrow \max_{\ell\in[L]} \mu_\ell$
  return $\xi'_0, \mu_{\max}$ ▷ $\xi'_0 = \frac{1}{2}(\xi_L + \xi_R)$
end procedure

Figure 4-6: Multiscale matrix pencil method.

Run MatrixPencilMultiscale *in Figure 4-6 on* $(X'_{j2^{\ell M}})_{|j|\le J-1,\ell\in[L]}$ *which returns an estimate* ξ'_0. *Suppose* $\xi_0 = 0.d_0 \dots d_{L-1}$ *in base* 2^M. *Then* $\mathbb{P}\left(|\xi'_0 - \xi_0| < 2^{-LM-1}\right) \le 3 \cdot 2^{2M/3} \sum_{\ell\in[L]} \left(\mathbb{E}\left|\Delta\xi_0^\ell\right|^2\right)^{1/3}$. *By Proposition 4.2.1 and Assumption 4.2.2,*

$$\mathbb{P}\left(|\xi'_0 - \xi_0| < 2^{-LM-1}\right) \le 3 \cdot 2^{2M/3} \sum_{\ell\in[L]} \left(\frac{\sum_{|j|\le J-1} \mathbb{E}\left|\Delta X_{j2^{\ell M}}\right|^2}{\pi^2 |c_0|^2 J(J-1)^2} \right)^{1/3}.$$

Note that Proposition 4.2.4 is not useful if $\left(\mathbb{E}\left|\Delta\xi_0^\ell\right|^2\right)^{1/2} \ge 3^{-3/2}2^{-M}$ for any $\ell \in [L]$.

Proof of Proposition 4.2.4. It suffices to bound $\mathbb{P}(d'_\ell \ne d_\ell \text{ for some } \ell \in [L])$. Fix a $\ell \in [L]$. Note that ξ_0^ℓ is also uniformly distributed in $[0,1)$. Let $0 < \theta < 2^{-M-1}$. Split $[0,1)$ into 2^M parts. Define the "*decision edges*" as $\text{III} = [2^M]/2^M$. Suppose $\left|\Delta\xi_0^\ell\right| \le \theta$. Then as long as ξ_0^ℓ is more than θ away from all the decision edges, i.e., $\min_{\eta\in\text{III}} \text{dist}(\xi_0^\ell, \eta) > \theta$, d_ℓ will be identified correctly as $\lfloor \xi_0'^{\ell} 2^M \rfloor$. For example, if $M = 1$, we can obtain the correct d_ℓ whenever $\xi_0^\ell \notin [0,\theta] \cup [\frac{1}{2} - \theta, \frac{1}{2} + \theta] \cup [1-\theta, 1]$. Hence, by union bound and Markov's inequality,

$$\mathbb{P}(d'_\ell \ne d_\ell) \le 2^{M+1}\theta + \frac{\mathbb{E}\left|\Delta\xi_0^\ell\right|^2}{\theta^2}.$$

Pick $\theta = 2^{-M/3}(\mathbb{E}\left|\Delta\xi_0^\ell\right|^2)^{1/3}$ to obtain $\mathbb{P}(d'_\ell \ne d_\ell) \le 3 \cdot 2^{2M/3} \left(\mathbb{E}\left|\Delta\xi_0^\ell\right|^2\right)^{1/3}$. Union

bound over all $\ell \in [L]$ to complete the proof. $\square$

Suppose we fix J and the desired accuracy, i.e., fix LM, the number of bits of ξ_0 we want. Then we should pick $M = 1$ to minimize the chance of failure according to Proposition 4.2.4. However, if ΔX is very small, then Proposition 4.2.4 says that the failure probability can be acceptable even if we choose a larger M. The advantage of a larger M is that L can be smaller (since LM is fixed) and the number of samples, $\mathcal{N}_{J,M} := \left|\{j2^{\ell M} : |j| \leq J-1, \ell \in [L]\}\right|$ can be smaller. In MPFFT, taking a sample of our sub-signal corresponds to one expensive binning operation, so choosing a larger M can speed up MPFFT significantly.

Nevertheless, for maximum robustness, we recommend choosing $M = 1$. For this case, instead of computing $\xi_0'^{\ell} = \frac{1}{2\pi} \arg \omega_0'$ when running `MatrixPencil` in Figure 4-5, it suffices to check if $\mathrm{Im}(\omega_0) > 0$. In addition, if J is even, then

$$\mathcal{N}_{J,1} = J(L+1) - 1; \quad \mathcal{N}_{J+1,1} = \mathcal{N}_{J,1} + 2. \tag{4.12}$$

This suggests that we should always pick an odd J because for two extra samples, J can be incremented which implies a better error bound by Proposition 4.2.4 and also a better collision detector as we will see in Section 4.6.2. In our numerical experiments (cf. Section 4.6.1), however, MPFFT seems to be just as robust when $J = 2$ as $J = 3$.

Finally, if the Fourier coefficients of X' are all real, then A is a Hermitian matrix and the number of samples needed N_S can be halved because $X_j' = \overline{X_{-j}'}$.

4.3 Collision detector

Let us return to studying `MatrixPencil` in Figure 4-5 with $Q = 1$. Consider the following probability model for the noisy signal X'. Fix P coefficients $c_0, \ldots, c_{P-1}$ such that $|c_0| \geq \ldots \geq |c_{P-1}|$. Our true signal is $X_j = c_0 e^{2\pi i j \xi_0}$ and ξ_0 is referred to as the *dominant mode*. Our noise vector ΔX is composed of $P-1$ subdominant modes, i.e., $\Delta X_j = \sum_{s=1}^{P-1} c_s e^{2\pi i j \xi_s}$. Assume the ξ_s's to be independently and *uniformly chosen* from $[0, 1)$. The results of this section also hold if the frequencies lie on a grid, i.e.,

$\xi_s = k_s/N$ with k_s independently, uniformly chosen from $[N]$. Be warned that these results cannot be applied directly in MPFFT because these modes can be thought of as being fully randomly shuffled whereas in MPFFT, the random shuffling of modes is only pairwise independent.

Let $v_s = \frac{1}{\sqrt{J}}\mathbf{v}_J(\xi_s)$. Note that $A = c_0 v_0 v_0^*$ and $\Delta A = \sum_{s=1}^{P-1} c_s v_s v_s^*$. In MPFFT, we stop processing a bin if μ^2 exceeds a certain threshold. This is our mode collision test. The problem of false negatives, i.e., not rejecting a bin when we should, is tricky. Let us first deal with the problem of *false positives*. For any vector u, let

$$\mathcal{R}[u] = \frac{1}{J}\left|\sum_j u_j\right|^2 + \|u\|^2\left(1 - \frac{1}{J}\right).$$

Proposition 4.3.1. *Assume that* $X'_j = \sum_{s\in[P]} c_s e^{2\pi i j \xi_s}$ *where each* ξ_s *is independently, uniformly chosen from* $[0,1)$. *Run* `MatrixPencil` *in Figure 4-5 on* $(X'_j)_{|j|\le J-1}$ *which returns some* μ. *Then*

$$\mathbb{P}\left(\mu^2 \ge t\mathcal{R}[(c_1,\ldots,c_{P-1})]\right) \le 1/t.$$

Proposition 4.3.1 suggests that if $\mathcal{R}[(c_1,\ldots,c_{P-1})]$ is small, then it is unlikely that μ^2 is large and the bin is rejected by the collision test. To prove Proposition 4.3.1, we compute some basic quantities that will be useful later as well.

Lemma 4.3.2. *For any* $s \in [P]$, $\mathbb{E}\, v_s v_s^* = I/J$ *where* I *is the identity. Let* $K \subseteq [P]$ *be an index set. Then*

$$\left\|\mathbb{E}\sum_{s\in K} c_s v_s v_s^*\right\| = \frac{\left|\sum_{s\in K} c_s\right|}{J}; \quad \mathbb{E}\left\|\sum_{s\in K} c_s v_s v_s^*\right\|_F^2 = \mathcal{R}[(c_s)_{s\in K}].$$

Proof. We leave it to the reader to verify that $\mathbb{E}\, v_s v_s^* = I/J$. Now, $\mathbb{E}\sum_s c_s v_s v_s^* =$

$\sum_s c_s \mathbb{E}\, v_s v_q^* = \frac{\sum_s c_s}{J} I$. On the other hand,

$$\begin{aligned}
\mathbb{E}\left\|\sum_s c_s v_s v_s^*\right\|_F^2 &= \mathbb{E}\,\mathrm{tr}\left((\sum_r c_r v_r v_r^*)^*(\sum_s c_s v_s v_s^*)\right) \\
&= \sum_s |c_s|^2 \,\mathrm{tr}(\mathbb{E}\, v_s v_s^* v_s v_s^*) + \sum_{r\neq s} c_s \overline{c_r}\,\mathrm{tr}(\mathbb{E}\, v_s v_s^* v_r v_r^*) \\
&= \sum_s |c_s|^2 \,\mathrm{tr}(\mathbb{E}\, v_s v_s^*) + \sum_{r\neq s} c_s \overline{c_r}\,\mathrm{tr}(I/J^2) \\
&= \sum_s |c_s|^2 + \sum_{\substack{r,s \\ r\neq s}} c_s \overline{c_r}/J \\
&= \left|\sum_s c_s\right|^2 \frac{1}{J} + \sum_s |c_s|^2 \left(1 - \frac{1}{J}\right).
\end{aligned}$$

□

Proof of Proposition 4.3.1. Recall that $A' = A + \Delta A$ where A has rank one. Write

$$\mu^2 = \sum_{j=2}^{J} \sigma_j(A')^2 \leq \sum_{j=1}^{J} (\sigma_j(A') - \sigma_j(A))^2 \leq \|\Delta A\|_F^2 . \tag{4.13}$$

The second inequality is due to Wielandt-Hoffman. By Lemma 4.3.2, $\mathbb{E}\,\|\Delta A\|_F^2 = \mathcal{R}[(c_1, \ldots, c_{P-1})]$. The result follows from Markov's inequality. □

Now, consider *false negatives*. The claim is that if μ^2 is small, then most likely, the noise energy $\sum_{s=1}^{P-1} |c_s|^2$ and the sum of the coefficients $\left|\sum_{s=1}^{P-1} c_s\right|$ will be both small. In the context of MPFFT, X' is the sub-signal of a bin dilated in frequency by a factor of $2^{\ell M} B_r$. If the noise energy in the bin is small, then our dominant mode must be isolated and if $\left|\sum_{s=1}^{P-1} c_s\right|$ is small, then the coefficient estimation error must be small. This will be useful in Section 4.5 where MPFFT is analyzed.

For the rest of Section 4.3, we establish the claim or weaker form of the claim for some special cases, e.g., multiple heavy modes of roughly the same magnitude, or one heavy mode with many smaller modes of roughly the same magnitude. Lastly, Section 4.6.2 contains some supporting numerical evidence.

4.3.1 Total energy comparable to energy of dominant mode

The first result pertains to the case where there is substantial amount of subdominant energy relative to our dominant mode, i.e.,

$$\sum_{s\in[P]} |c_s|^2 \gtrsim |c_0|^2 J.$$

Theorem 4.3.3. *Without loss of generality, assume that* $|c_0| = 1$. *Let* $\alpha = \sum_{s\in[P]} c_s$ *and* $\beta^2 = \|c\|^2 \geq 1$ *where* $c = (c_0, \ldots, c_{P-1})^T$. *Assume that* $X'_j = \sum_{s\in[P]} c_s e^{2\pi i j \xi_s}$ *where each* ξ_s *is independently, uniformly chosen from* $[0, 1)$. *Let* $A' = \mathbb{T}X'$ *as in (4.7). For any* $0 < t < \frac{\beta^2}{C_1 J^{1/2}}$ *and* $0 < u < C_6 \frac{\beta^2}{J}$, *we have with probability at least* $1 - C_2^2 e^{-t} - 2Je^{-u}$,

$$\|A'\|_F^2 - \|A'\|^2 \geq \frac{|\alpha|^2}{J}\left(1 - \frac{2}{J}\right) + \beta^2\left(1 - \frac{1}{J} - \frac{C_3 t}{J^{1/2}} - \frac{2C_5^2 u}{J}\right). \tag{4.14}$$

The constants C_1, C_2, C_3, C_5, C_6 *are defined in Lemma 4.3.5 and Corollary A.2.2.*

For the bound on the failure probability $C_2^2 e^{-t} + 2Je^{-u}$ to be nontrivial, t, u cannot be too small. For example, u has to be $\Omega(\log J)$. But for the lower bound in (4.14) to be useful, J has to be sufficiently large relative to t, u . Recall our assumption about β^2. It has to be comparable to J for the proof to go through. Therefore, the theorem is applicable only for a sufficiently large J and a sufficiently large β^2 relative to J.

Now, let us prove Theorem 4.3.3. The idea is to use concentration inequalities to control $\|A'\|_F^2 - \mathbb{E}\|A'\|_F^2$ and $\|A' - \mathbb{E}A'\|$. First, we check that $\|\mathbb{E}A'\|^2$ and $\mathbb{E}\|A'\|_F^2$ are sufficiently far apart. The gap between them is $\frac{|\alpha|^2}{J}(1 - \frac{1}{J}) + \beta^2(1 - \frac{1}{J})$.

Later, we also need the second and fourth moments of $|v_r^* v_s|$.

Lemma 4.3.4. *Let* $r, s \in [P]$ *such that* $r \neq s$. *Then*

$$\mathbb{E}|v_r^* v_s|^2 = \frac{1}{J}; \quad \mathbb{E}|v_r^* v_s|^4 = \frac{1 + 2J^2}{3J^3}.$$

Proof. Note that v_s are random vectors in isotropic position [56, 66] because for all $u \in \mathbb{C}^J$, $\mathbb{E}|u^* v_s|^2 = u^*(\mathbb{E}v_s v_s^*)u = \|u\|^2/J$. Let $u = v_r$ which is independent of v_s to

obtain the second moment of $|v_r^* v_s|$.

For the fourth moment, we condition on v_r and take expectation over v_s. Let $B_r = \mathrm{diag}(v_r)^*$. Write $|v_r^* v_q|^4 = |(B_r v_r)^*(B_r v_s)|^4 = |z^* v'_s|^4$ where $v'_s = B_r v_s$ and $z = (1, \ldots, 1)^T/\sqrt{J}$. Observe that conditioned on v_r, v'_s has the same distribution as v_s, so $\mathbb{E}\,|v_r^* v_s|^4 = \mathbb{E}\,|z^* v_s|^4$. Take expectation over v_r to undo the conditioning and obtain $\mathbb{E}\,|v_r^* v_s|^4 = \mathbb{E}\,|z^* v_s|^4$. The latter can be evaluated directly. Let $\delta(x) = 1$ if $x = 0$, zero otherwise.

$$\begin{aligned}
\mathbb{E}\,|z^* v_q| &= \frac{1}{J^4} \int_{\xi=0}^{1} \left| \sum_{j \in [J]} e^{-2\pi i j \xi} \right|^4 d\xi \\
&= \frac{1}{J^4} \int_{\xi=0}^{1} \sum_{j_1, j_2, j_3, j_4 \in [J]} e^{2\pi i (j_1 - j_2 + j_3 - j_4)\xi} d\xi \\
&= \frac{1}{J^4} \sum_{j_1, j_2, j_3, j_4 \in [J]} \delta(j_1 - j_2 + j_3 - j_4).
\end{aligned}$$

We want to count the total number of 4-tuples $(j_1, j_2, j_3, j_4) \subseteq [J]^4$ such that $j_1 - j_2 + j_3 - j_4 = 0$. If $j_1 - j_2 = k$ for some $-(J-1) \le k \le J-1$, then there are $J - |k|$ pairs of (j_3, j_4) such that $j_3 - j_4 = -k$. Deduce that the total number of such 4-tuples is $\sum_{k=-(J-1)}^{J-1} (J - |k|)^2 = J^2 + 2\sum_{k=1}^{J-1} J - k = \frac{J}{3}(1 + 2J^2)$. It follows that $\mathbb{E}\,|z^* v_s| = \frac{1}{J^4}\frac{J}{3}(1 + 2J^2) = \frac{1+2J^2}{3J^3}$. □

Next, we bound the deviation of $\|A'\|_F^2$ from its mean using *moments*. Let $\mathbb{E}_m X$ denote $(\mathbb{E}\,|X|^m)^{1/m}$ for any real random variable X. We will use standard techniques such as symmetrization, decoupling and Khintchine inequalities. Let $\tilde{X}$ denote an independent copy of random variable X.

Lemma 4.3.5. *Let $C_1 = 4$, $C_2 = 12$. For any $t > 0$,*

$$\mathbb{P}\left(\left| \|A'\|_F^2 - \mathbb{E}\,\|A'\|_F^2 \right| \ge \frac{eC_1\beta^2}{J^{1/2}} t + \frac{eC_1^{3/2}\beta}{J^{1/4}} t^{3/2} + eC_1^2 t^2 \right) \le C_2^2 e^{-t}.$$

Let $C_3 = 3eC_1$. If $0 < t < \frac{\beta^2}{C_1 J^{1/2}}$, then $\mathbb{P}\left(\left| \|A'\|_F^2 - \mathbb{E}\,\|A'\|_F^2 \right| \ge t\frac{C_3\beta^2}{J^{1/2}} \right) \le C_2^2 e^{-t}$.

Proof. Let $D_m = C_1 C_2^{1/m}$. Let $c_{rs} = c_r c_s$ if $r \ne s$ and zero otherwise, i.e., it is

diagonal-free. Now,

$$\begin{aligned}
\mathbb{E}_m\left(\|A'\|_F^2 - \mathbb{E}\,\|A'\|_F^2\right) &= \mathbb{E}_m \sum_{r,s\in[P]} c_{rs}\left(|v_r^* v_s|^2 - \frac{1}{J}\right) \\
&\le D_m\,\mathbb{E}_m \sum_{r,s\in[P]} c_{rs}\varepsilon_r\tilde{\varepsilon}_s\,|v_r^*\tilde{v}_s|^2 \quad \text{by [22, Theorem 3.5.3]} \\
&\le D_m m\,\mathbb{E}_m\left(\sum_{r,s\in[P]} |c_{rs}|^2\,|v_r^* v_s|^4\right)^{1/2} \quad \text{by [22, Theorem 3.2.2]} \\
&\le D_m m\left(\mathbb{E}_m \sum_{r,s\in[P]} |c_{rs}|^2\,|v_r^* v_s|^4\right)^{1/2}. \qquad (4.15)
\end{aligned}$$

The randomization step between the first and second line is justified by the fact that the sum $\sum_{r,s\in[P]} c_{rs}(|v_r^* v_s|^2 - \frac{1}{J})$ is centered and degenerate of order 1 [22, Chapter 3]. This means that for any $r \neq s$, when we fix v_r, we have $\mathbb{E}\,c_{rs}(|v_r^* v_s|^2 - \frac{1}{J}) = 0$, due to Lemma 4.3.4.

Next, we bound $\mathbb{E}_m \sum_{r,s\in[P]} |c_{rs}|^2\,|v_r^* v_s|^4$ in the same way as in [67]. Let $g = \mathbb{E}\sum_{r,s\in[P]} |c_{rs}|^2\,|v_r^* v_s|^4$. By Lemma 4.3.4, $g = \sum_{r,s\in[P]} |c_{rs}|^2 \frac{1+2J^2}{3J^3} \le \|c\|^4 \frac{1+2J^2}{3J^3} \le \frac{\|c\|^4}{J}$. Furthermore,

$$\begin{aligned}
E &:= \mathbb{E}_m\left(\sum_{r,s\in[P]} |c_{rs}|^2\,|v_r^* v_s|^4 - g\right) \\
&\le D_m\,\mathbb{E}_m \sum_{r,s\in[P]} |c_{rs}|^2\,\varepsilon_p\tilde{\varepsilon}_q\,|v_r^*\tilde{v}_q|^4 \quad \text{by [22, Theorem 3.5.3]} \\
&\le D_m m\left(\mathbb{E}_m \sum_{r,s\in[P]} |c_{rs}|^4\,|v_r^*\tilde{v}_q|^8\right)^{1/2} \quad \text{by [22, Theorem 3.2.2]} \\
&\le D_m m\left(\mathbb{E}_m \sum_{r,s\in[P]} |c_{rs}|^2\,|v_r^*\tilde{v}_q|^4\right)^{1/2} \\
&\le D_m m(E+g)^{1/2}.
\end{aligned}$$

The sum $\mathbb{E}_m \sum_{r,s\in[P]} |c_{rs}|^2\,(|v_r^* v_s|^4 - \frac{1+2J^2}{3J^3})$ is also centered and degenerate of order 1 by Lemma 4.3.4. This justifies the randomization in the second line. We have

obtained $E \le D_m m(E+g)^{1/2}$. It follows that $E \le \frac{1}{2}(D_m^2 m^2 + \sqrt{D_m^4 m^4 + 4D_m^2 m^2 g}) \le D_m^2 m^2 + D_m m g^{1/2}$ and

$$\mathbb{E}_m \sum_{r,s\in[P]} |c_{rs}|^2 |v_r^* v_s|^4 \le g + D_m m g^{1/2} + D_m^2 m^2 \le \frac{\|c\|^4}{J} + \frac{D_m m \|c\|^2}{J^{1/2}} + D_m^2 m^2.$$

Substitute this back into (4.15) to obtain $\mathbb{E}_m \left(\|A'\|_F^2 - \mathbb{E}\|A'\|_F^2 \right) \le \frac{D_m m \|c\|^2}{J^{1/2}} + \frac{D_m^{3/2} m^{3/2} \|c\|}{J^{1/4}} + D_m^2 m^2$. Recall $D_m = C_1 C_2^{1/m}$. Let $\Delta = \frac{C_1 t \|c\|^2}{J^{1/2}} + \frac{C_1^{3/2} t^{3/2} \|c\|}{J^{1/4}} + C_1^2 t^2$. Let $m = t$. By Markov's inequality, $\mathbb{P}\left(\left| \|A'\|_F^2 - \mathbb{E}\|A'\|_F^2 \right| \ge e\Delta \right) \le \frac{\left(\mathbf{E}_t \left| \|A'\|_F^2 - \mathbf{E}\|A'\|_F^2 \right| \right)^t}{(e\Delta)^t} \le C_2^2 e^{-t}$. □

As for the deviation in the spectral norm, we use the Matrix Bernstein inequality [58, 74].

Lemma 4.3.6. *Let C_6 be as defined in Corollary A.2.2. For any $0 < u < C_6 \frac{\beta^2}{J}$,*

$$\mathbb{P}\left(\|A' - \mathbb{E}A'\| \ge C_5 u^{1/2} \frac{\beta}{J^{1/2}} \right) \le 2Je^{-u}.$$

Proof. Note that $A' - \mathbb{E}A' = \sum_{s\in[P]} c_s (v_s v_s^* - \frac{I}{J})$. Apply Theorem A.2.2 with $G_k = c_k(v_k v_k^* - \frac{I}{J})$, $R = 1$ and $\sigma^2 = \frac{\beta^2}{J}(1 - \frac{1}{J}) \le \frac{\beta^2}{J}$. □

Finally, we apply our bounds on $\|A' - \mathbb{E}A'\|$ and $\left| \|A'\|_F^2 - \mathbb{E}\|A'\|_F^2 \right|$ to prove Theorem 4.3.3.

Proof of Theorem 4.3.3. By Lemma 4.3.5 and Lemma 4.3.2, with probability at least $1 - C_2^2 e^{-t}$, $\|A'\|_F^2 \ge \frac{|\alpha|^2}{J} + \beta^2 \left(1 - \frac{1}{J}\right) - \frac{C_3 \beta^2}{J^{1/2}} t$. By Lemma 4.3.6 and Lemma 4.3.2, with probability at least $1 - 2Je^{-u}$, $\|A'\|^2 \le \left(\frac{|\alpha|}{J} + \frac{C_5 \beta}{J^{1/2}} u^{1/2} \right)^2 \le \frac{2|\alpha|^2}{J^2} + \frac{2C_5^2 \beta^2}{J} u$. Take the difference between the two bounds and we are done. □

4.3.2 Subdominant energy comparable to energy of second mode

Suppose $\sum_{s=1}^{P-1} |c_s|^2 \gtrsim |c_1|^2 J$. This models the case where there is an isolated heavy mode and many small nonheavy modes of roughly the same magnitude in the sub-

signal of a bin. Such nonheavy modes typically arise from coefficient estimation errors made in previous iterations. As a simple consequence of Theorem 4.3.3, we have:

Corollary 4.3.7. *Without loss of generality, assume that* $|c_1| = 1$. *Let* $\alpha = \sum_{s=1}^{P-1} c_s$ *(which excludes* c_0*) and* $\beta^2 = \sum_{s=1}^{P-1} |c_s|^2 \geq 1$ *(which excludes* $|c_0|^2$*). Assume that* $X'_j = \sum_{s\in[P]} c_s e^{2\pi i j \xi_s}$ *where each* ξ_s *is independently, uniformly chosen from* $[0,1)$. *Let* $A' = \mathbb{T}X'$ *as in (4.7). For any* $0 < t < \frac{\beta^2}{C_1 J^{1/2}}$ *and* $0 < u < C_6 \frac{\beta^2}{J}$,

$$\|A'\|_F^2 - \|A'\|^2 \geq \frac{|\alpha|^2}{J}\left(1 - \frac{4}{J}\right) + \beta^2\left(1 - \frac{1}{J} - \frac{C_3 t}{J^{1/2}} - \frac{4C_5^2 u}{J}\right).$$

The constants C_1, C_2, C_3, C_5, C_6 *are defined in Lemma 4.3.5 and Corollary A.2.2.*

Proof. As $A' - \Delta A$ is rank one, we have $\sum_{j=2}^{J} \sigma_j^2(A') \geq \sum_{j=3}^{J} \sigma_j^2(\Delta A) \geq \|\Delta A\|_F^2 - 2\|\Delta A\|^2$. Compute a lower bound for $\|\Delta A\|_F$ and an upper bound for $\|\Delta A\|$. Specifically, apply the proof of Theorem 4.3.3 with X'_j replaced with $\Delta X_j = \sum_{s=1}^{P-1} c_s e^{2\pi i j \xi_s}$ and A replaced with ΔA to conclude that with probability at least $1 - C_2^2 e^{-t} - 2Je^{-u}$, $\|\Delta A\|_F^2 \geq \frac{|\alpha|^2}{J} + \beta^2(1 - \frac{1}{J}) - \frac{C_3\beta^2}{J^{1/2}}t$ and $\|\Delta A\|^2 \leq \frac{2|\alpha|^2}{J^2} + \frac{2C_5^2\beta^2}{J}u$. Take the difference between the two bounds to complete the proof. □

4.3.3 Subdominant modes do not cancel one another

Suppose $\sum_{s=1}^{P-1} |c_s|^2 \gtrsim |c_1|^2 J$ like in the previous section. In addition, assume that the subdominant modes do not cancel one another and satisfies $\frac{1}{|c_1|}\left|\sum_{s=1}^{P-1} c_s\right| \gtrsim \frac{1}{|c_1|^2}\sum_{s=1}^{P-1} |c_s|^2$. Then $\|A'\|_F^2 - \|A'\|_F^2 \gtrsim \beta^2$ according to Proposition 4.3.8 below. It is a weaker result than Corollary 4.3.7 as it requires an additional assumption, but its proof is much shorter and may be of interest to the reader. An example of such a signal is $c_s = s^{-1/2}$ for $s \geq 1$. In this case, $\sum_{s=1}^{P-1} c_s = \Theta(P^{1/2})$ which is much bigger than $\sum_{s=1}^{P-1} |c_s|^2 = \Theta(\log P)$.

Proposition 4.3.8. *Without loss of generality, assume that* $|c_1| = 1$. *Let* $\alpha = \sum_{s=1}^{P-1} c_s$ *and* $\beta^2 = \sum_{s=1}^{P-1} |c_s|^2 \geq 1$. *Assume that* $X'_j = \sum_{s\in[P]} c_s e^{2\pi i j \xi_s}$ *where each* ξ_s *is independently, uniformly chosen from* $[0,1)$. *Let* $A' = \mathbb{T}X'$ *as in (4.7). Assume that for some* $\kappa > 1$, $|\alpha| \geq \kappa C_5 C_6^{1/2} \beta^2$. *For any* $0 < u < C_6 \frac{\beta^2}{J}$, *we have with*

probability at least $1 - 2Je^{-u}$,

$$\|A'\|_F^2 - \|A'\|^2 \geq \beta^2(\kappa-1)^2 C_5^2 \left(\frac{J-2}{J}\right)\left(\frac{C_6^{1/2}\beta}{J^{1/2}} - u^{1/2}\right)^2.$$

The constants C_5, C_6 *are defined in Corollary A.2.2.*

Proof. Like in Lemma 4.3.6, apply matrix Bernstein or Theorem A.2.2 to ΔA and deduce that with probability at least $1 - 2Je^{-u}$, $\|\Delta A - \mathbb{E}\Delta A\| \leq \frac{C_5\beta}{J^{1/2}}u^{1/2}$. Since $\mathbb{E}\Delta A = \frac{\alpha I}{J}$, we have that for $1 \leq j \leq J$,

$$\sigma_j(\Delta A) \geq \frac{|\alpha|}{J} - \frac{C_5\beta}{J^{1/2}}u^{1/2} \geq \frac{\kappa C_5 C_6^{1/2}\beta^2}{J} - \frac{C_5\beta}{J^{1/2}}u^{1/2} \geq (\kappa-1)C_5\frac{\beta}{J^{1/2}}\left(\frac{C_6^{1/2}\beta}{J^{1/2}} - u^{1/2}\right). \tag{4.16}$$

As $A' - \Delta A$ is rank one, $\sum_{j=2}^{J}\sigma_j^2(A') \geq \sum_{j=3}^{J}\sigma_j(\Delta A)^2$. Substitute in (4.16) to complete the proof. □

4.3.4 A few heavy modes with little noise

In MPFFT, most of the time, very few heavy modes will land in the same bin. Let $T < J$ be a small integer, say $T = 2$ or $T = 3$. In Theorem 4.3.9 below, we shall treat the case where there are T heavy modes with little noise energy $\sum_{s=T}^{P-1}|c_s|^2$.

Theorem 4.3.9. *Let* $T < J$. *Without loss of generality, assume that* $|c_0| \geq \ldots \geq |c_{T-1}| = 1 \gg |c_T| \geq \ldots \geq |c_{P-1}|$. *Assume that* $X'_j = \sum_{s\in[P]} c_s e^{2\pi i j\xi_s}$ *where each* ξ_s *is independently, uniformly chosen from* $[0,1)$. *Let* $A' = \mathbb{T}(X')$ *as in (4.7). For any* $t > 0$ *and* $0 < u < \sqrt{T-1}(1-t)$, *we have*

$$\mathbb{P}\left(\|A'\|_F^2 - \|A'\|^2 \geq \left(\sqrt{T-1}(1-t) - u\right)^2\right) \leq \frac{T^2}{tJ} + \frac{\mathcal{R}[(c_T,\ldots,c_{P-1})]}{u^2}.$$

Before we prove Theorem 4.3.9, we remark that unlike Theorem 4.3.3, Theorem 4.3.9 works for an arbitrarily large $|c_0|^2$ relative to the energy of the subdominant modes $\sum_{s=1}^{P-1}|c_s|^2$. Also, in our application, $\sum_{s=T}^{P-1}c_s v_s v_s^*$ is due to nonheavy modes and $\left\|\sum_{s=T}^{P-1}c_s v_s v_s^*\right\|_F$ is very small. Hence, Theorem 4.3.9 should be interpreted as

$\mathbb{P}\left(\|A'\|_F^2 - \|A'\|^2 \gtrsim T(1-t)^2\right) \lesssim \frac{T^2}{tJ}$.

Proof of Theorem 4.3.9. Let $W = (v_0, \ldots, v_{T-1}) \in \mathbb{C}^{J\times T}$. Write

$$W^*W = I + \begin{pmatrix} 0 & v_0^*v_1 & \ldots & v_0^*v_{T-1} \\ v_1^*v_0 & 0 & \ldots & v_1^*v_{T-1} \\ \vdots & \vdots & \ddots & \vdots \\ v_{T-1}^*v_0 & v_{T-1}^*v_1 & \ldots & 0 \end{pmatrix}.$$

By Lemma 4.3.4, $\mathbb{E}\,\|W^*W - I\|_F^2 = (T^2 - T)/J \leq T^2/J$. Therefore, for any $t > 0$, $\mathbb{P}\left(\|W^*W - I\| \geq t\right) \leq \frac{T^2}{tJ}$. If $\|W^*W - I\| \leq t$, then $\sigma_j(W) \geq \sqrt{1-t}$ for $1 \leq j \leq T$ and by [76], we have

$$\begin{aligned}
\sum_{j=1}^{T-1} \sigma_{j+1}^2(W \operatorname{diag}(c_0, \ldots, c_{T-1})W^*) &\geq \sum_{j=1}^{T-1} \sigma_{j+1}^2(W)\sigma_{T-j+1}^2\left(\operatorname{diag}(c_0, \ldots, c_{T-1})W^*\right) \\
&\geq \sum_{j=1}^{T-1} \sigma_{j+1}^2(W)\sigma_{T-j+1}^2\left(\operatorname{diag}(c_0, \ldots, c_{T-1})\right)\sigma_T^2(W) \\
&\geq (T-1)(1-t)^2.
\end{aligned}$$

Let $H = \sum_{s=T}^{P-1} c_s e^{2\pi i j \xi_s}$. By Lemma 4.3.2, $\mathbb{E}\,\|H\|_F^2 = \mathcal{R}[(c_T, \ldots, c_{P-1})]$. By Markov, for any $u > 0$, $\mathbb{P}\left(\|H\|_F \geq u\right) \leq \mathcal{R}[(c_T, \ldots, c_{P-1})]/u^2$. Conclude that with probability at least $1 - \frac{T^2}{tJ} - \frac{\beta^2}{u^2}$, we have

$$\begin{aligned}
\left(\|A'\|_F^2 - \|A'\|^2\right)^{1/2} &\geq \left(\sum_{j=1}^{T-1} \sigma_{j+1}^2\left(W \operatorname{diag}(c_0, \ldots, c_{T-1})W^* + H\right)\right)^{1/2} \\
&\geq \left(\sum_{j=1}^{T-1} \sigma_{j+1}^2(W \operatorname{diag}(c_0, \ldots, c_{T-1})W^*)\right)^{1/2} - \left(\sum_{j=1}^{T-1} \sigma_j^2(H)\right)^{1/2} \\
&\geq \sqrt{T-1}(1-t) - u.
\end{aligned}$$

Between the first line and second line, we applied Wielandt-Hoffman-Lidskii in the following way. Let $\boldsymbol{\sigma}(\cdot)$ denote the vector of singular values of a given matrix. It is known [50, 55] that for any matrices A, E, $|\boldsymbol{\sigma}(A+E) - \boldsymbol{\sigma}(A)|$ is weakly majorized

from below[3] by $\boldsymbol{\sigma}(E)$. Thus, for any $i_1 > \ldots > i_k$, $\sum_{j=1}^{k} \left|\sigma_{i_j}(A+E) - \sigma_{i_j}(A)\right|^2 \leq \sum_{j=1}^{k} \sigma_j^2(E)$, which implies by triangle inequality that

$$\left(\sum_{j=1}^{k} \sigma_{i_j}^2(A+E)\right)^{1/2} \geq \left(\sum_{j=1}^{k} \sigma_{i_j}^2(A)\right)^{1/2} - \left(\sum_{j=1}^{k} \sigma_j^2(E)\right)^{1/2}.$$

□

4.3.5 Two heavy modes

The $T = 2$ case is especially important to MPFFT. The reason is that it is very unlikely for more than a few heavy modes to land in a bin. Let S_r be the number of heavy modes in the residual signal and B_r be the number of bins in iteration r. Condition on a fixed heavy mode landing in a bin b. Let X be the number of *other* heavy modes in bin b. In practice, the shuffling of the heavy modes appears to be *fully random* such that X is a binomial random variable $X \sim \mathrm{Bin}(S_r - 1, 1/B_r)$. Hence, $\mathbb{P}(X = T-1) = \frac{1}{B_r^{T-1}} \left(1 - \frac{1}{B_r}\right)^{S_r - T} = B_r \left(1 - \frac{1}{B_r}\right)^{S_r} (B_r - 1)^{-T}$ decreases exponentially with T. Clearly, the most common case that needs to be detected and rejected by the collision detector is $T = 2$.

Fix the frequencies of the two heavy modes, ξ_0, ξ_1. Let $|c_0| \geq |c_1| = 1 \geq |c_2| \geq \ldots \geq |c_{P-1}|$ and $H = \sum_{s=T}^{P-1} c_s e^{2\pi i j \xi_s}$. Specialize the proof of Theorem 4.3.9 to $T = 2$. Let $W = (v_0 \quad v_1)$. By applying Gershgorin to W^*W, $\sigma_2(W) \geq \sqrt{1 - |v_0^* v_1|}$. It follows that

$$\begin{aligned} \sigma_2(A') &\geq \sigma_2(W \operatorname{diag}(c_0, c_1) W^*) - \|H\| \\ &\geq 1 - |v_0^* v_1| - \|H\|. \end{aligned}$$

Observe that if v_0, v_1 are *incoherent*, i.e., $|v_0^* v_1|$ is small, then $\sigma_2(A')$ should be close to 1 and the collision detector should detect that there are two heavy modes in the bin. On the other hand, if ξ_0, ξ_1 are very close, then v_0, v_1 will be coherent and the

[3]For more on the topic of majorization, see Marshall, Olkin and Arnold [55] or Horn and Johnson [44].

collision detector may fail. Fortunately, when we try to identify the next group of M bits of the dominant mode in `MatrixPencilMultiscale` in Figure 4-6, the distance between the two frequencies of the signal supplied to `MatrixPencil` will be dilated by 2^M, and as we continue this dilation, this distance between the two frequencies will become sufficiently large such that v_0, v_1 become incoherent.

Proposition 4.3.10. *Let $M \geq 1$. Let $\xi_0, \xi_1 \in [0,1)$ where $\xi_0 \neq \xi_1$. Let $L \geq \log_{2^M} \frac{1}{dist(\xi_0,\xi_1)}$. Let $\xi_s^\ell = \xi_s 2^{\ell M}$. Let $v_s^\ell = \frac{1}{\sqrt{J}} \mathbb{V}_J(\xi_s^\ell)$ for $s = 0, 1$. Then there exists some $\ell \in [L]$ such that*

$$dist(\xi_0^\ell, \xi_1^\ell) \geq \frac{1}{2^M + 1}; \quad \left| v_0^{\ell *} v_1^\ell \right| \leq \frac{2^M + 1}{2J}.$$

For instance, for $M = 1$, we can expect $\sigma_2(H) \gtrsim 1 - \frac{3}{2J}$ for some $\ell \in [L]$. The result is *deterministic* but is useful for only $T = 2$ heavy modes.

Proof of Proposition 4.3.10. Without loss of generality, assume that $\xi_0 = 0$ and $\xi_1 = \text{dist}(\xi_0, \xi_1) < \frac{1}{2^M+1}$. Let

$$\ell_{\max} = \max\{\ell : \xi_1^\ell < 1\}.$$

In other words, $\ell = \ell_{\max} + 1$ is the first time $2^{\ell M} \text{dist}(\xi_0, \xi_1)$ exceeds or is equal to 1. The hypothesis $L \geq \log_{2^M} \frac{1}{\text{dist}(\xi_0, xi_1)}$ guarantees that $\xi_1^L \geq 1$, so $\ell_{\max} \leq L - 1$. Also, $\ell_{\max} \geq 1$ because $\xi_1 < \frac{1}{2^M+1}$ and $\xi_1^1 < 1$. To summarize, $1 \leq \ell_{\max} \leq L - 1$. Observe that $\xi_1^{\ell_{\max}}$ is away from 0, i.e.,

$$\xi_1^{\ell_{\max}} \geq 2^{-M} > (2^M + 1)^{-1}.$$

Otherwise, $\xi_1^{\ell_{\max}+1} < 1$, which contradicts the maximality of $\ell_{\max}$. If $\xi_1^{\ell_{\max}}$ is also away from 1, i.e., $\xi_1^{\ell_{\max}} \leq 1 - \frac{1}{2^M+1}$, then $\text{dist}(\xi_0^{\ell_{\max}}, \xi_1^{\ell_{\max}}) \geq (2^M + 1)^{-1}$ as desired. For the rest of the proof, suppose $\xi_1^{\ell_{\max}} > 1 - (2^M + 1)^{-1}$. The claim is that $\text{dist}(\xi_0^{\ell_{\max}-1}, \xi_1^{\ell_{\max}-1}) \geq \frac{1}{2^M+1}$. First, we see that $\xi_1^{\ell_{\max}-1}$ is away from 0:

$$\xi_1^{\ell_{\max}-1} > 2^{-M} \left(1 - (2^M + 1)^{-1}\right) = (2^M + 1)^{-1}.$$

Second, we see that $\xi_1^{\ell_{\max}-1}$ is away from 1: as $M \geq 1$,

$$\xi_1^{\ell_{\max}-1} \leq 2^{-M}\xi_1^{\ell_{\max}} < 1/2 \leq 1 - (2^M+1)^{-1}.$$

We have shown the first inequality of Proposition 4.3.10. For the second inequality, assume we have a ℓ such that $\zeta := \operatorname{dist}(\xi_0^\ell, \xi_1^\ell) \in [\frac{1}{2^M+1}, \frac{1}{2}]$. It follows that $\left|v_0^{\ell *} v_1^\ell\right| = \frac{1}{J}\left|\frac{\sin(J\pi\zeta)}{\sin(\pi\zeta)}\right| \leq \frac{1}{2J\zeta} \leq \frac{2^M+1}{2J}$. We have used the fact that $\sin(\pi\zeta) \geq 2\zeta$ on $[0, \frac{1}{2}]$. □

4.4 Binning

Binning is the most costly operation in MPFFT. In this section, we explain how binning is done, discuss how to speed up binning, and establish some elementary lemmas which will be needed in Section 4.5. As hinted by (4.2 in Section 4.1, binning is achieved by convolving our signal with a boxcar-like filter in frequency space. The basic design of this filter is the same as in [40], but we make some small improvements. Most notably, our analysis does not require N to be divisible by the number of bins B_r unlike Lemma 3.3 of [40] or Claim 3.7 of [41]. As a result, we do not need N to be powers of 2 in order to have more choices for B_r. In fact, we are allowed to work with a prime N which simplifies slightly the analysis of the random shuffling of modes in Section 4.5.

4.4.1 How binning works

Let B be the number of bins. For simplicity, assume that B is odd. For any odd T, let $[\![T]\!]$ denote $\{-\frac{T-1}{2}, \ldots, \frac{T-1}{2}\}$. For any $0 < w < \frac{1}{2}$, let $\hat{\chi}^w(\xi) : [-\frac{1}{2}, \frac{1}{2}) \to \mathbb{R}$ be the indicator function on $[-\frac{w}{2}, \frac{w}{2}]$, i.e., $\hat{\chi}^w(\xi) = 1$ if $|\xi| \leq w/2$, zero otherwise. Its semidiscrete Fourier transform $\chi^w \in \mathbb{R}^{\mathbb{Z}}$ is the sinc function: $\chi_t^w = \int_{-1/2}^{1/2} \hat{\chi}^w(\xi) e^{2\pi i \xi t} d\xi = \frac{\sin(\pi t w)}{\pi t}$.

Suppose we want to bin a signal $x \in \mathbb{C}^N$. Extend it to a $X \in \mathbb{C}^{\mathbb{Z}}$ by $X_t = x_{t\%N}$. The semidiscrete Fourier transform of X is a series of spikes supported on $\frac{[N]}{N}$, i.e., $\hat{X}(\xi) = \sum_{t\in\mathbb{Z}} X_t e^{2\pi i\xi t} = \sum_{t\in\mathbb{Z}}(\sum_{k\in[N]} \hat{x}_k e^{2\pi i k t/N}) e^{-2\pi i\xi t} = \sum_{k\in[N]} \hat{x}_k \delta(\xi - \frac{k}{N})$ where δ is the Dirac delta function.

procedure BININTIME($x \in \mathbb{C}^N, \alpha, \beta, \gamma, \mathcal{H}, B, \delta, \kappa$) ▷ $\mathcal{H} = \{j2^{\ell M}B : |j| \le J-1, \ell \in [L]\}$
  Compute W_t by (4.17) for all $t \in [\![P]\!]$
  for $\tau \in \mathcal{H}$ **do**
    for $t \in [\![P]\!]$ **do**
      $c \leftarrow e^{2\pi i(\Delta t + \beta(t+\tau)/N)} W_t$ where $\Delta = -\frac{1}{2B}$ ▷ $y_t = x_{\alpha t+\gamma} e^{2\pi i \beta t/N}$
      $d \leftarrow x_{(\alpha(t+\tau)+\gamma)\%N}$ ▷ $y_t^\tau = y_{t+\tau}$
      $u_{t\%P} \leftarrow cd$ ▷ $u_t = y_t^\tau W_t e^{2\pi i \Delta t}$
    end for
    for $t \in [B]$ **do**
      $v_t \leftarrow \sum_{j \in [P/B]} u_{jB+t}$
    end for
    $\hat{v} \leftarrow$ StandardFFT(v) ▷ $\hat{v}_b = \sum_{t \in [B]} v_t e^{2\pi i t b/B}$
    $Y_\tau^b \leftarrow \hat{v}_b$ for all $b \in [B]$
  end for
  return Y
end procedure
procedure BININFREQUENCY($\hat{x} \in \mathbb{C}^N, \alpha, \beta, \gamma, \mathcal{H}, B, \delta, \kappa$) ▷
$\mathcal{H} = \{j2^{\ell M}B : |j| \le J-1, \ell \in [L]\}$
  Zero out $Y \in \mathbb{C}^{B \times |\mathcal{H}|}$
  for $k_* \in \operatorname{supp} \hat{x}$ **do**
    $k_0 \leftarrow \varphi(k_*)$ where $\varphi(k) = \alpha k + \beta$
    $b \leftarrow \lfloor \frac{k_0 B}{N} \rceil$ ▷ $b \in [B]$ is the bin k_* lands in
    $c \leftarrow \hat{x}_{k_*} \hat{W}'_{b,k_0}$ where $\hat{W}'_{b,k}$ is defined in (4.21)
    for $\tau \in \mathcal{H}$ **do**
      $Y_\tau^b \leftarrow Y_\tau^b + ce^{2\pi i(\gamma k_*/N + k_0 \tau/N)}$
    end for
  end for
  return Y
end procedure

Figure 4-7: BinInTime runs in $O\left(\frac{B}{\kappa} \log \frac{1}{\delta}\right)$ time. As for BinInFrequency, we use the erf routine to approximate $\hat{W}_{b,k_0}$. In practice, it makes sense to assume that erf takes $O(1)$ time with respect to δ because if δ is too small, we will run into floating point precision issues *anyway*. With this assumption, BinFrequency runs in $O(|\operatorname{supp} \hat{x}|)$ time.

We want to *convolve* $\hat{X}(\xi)$ with the rectangular window $\hat{\chi}^{1/B}(\xi)$ and sample the result $\hat{X} \star \hat{\chi}^{1/B}(\xi)$ at B uniformly spaced points in $[0,1)$. Each of the B samples will be the sum of all the modes landing in a bin as desired. To implement the convolution in frequency space, we will need to multiply X with $\chi^{1/B}$. The problem is that the sinc function $\chi^{1/B}$ decays too slowly and we will need too many time samples of x. The solution is to *smooth* $\hat{\chi}^{1/B}(\xi)$ by convolving it with a periodized Gaussian, so that in the time domain, $\chi^{1/B}$ is multiplied by a Gaussian and will decay exponentially with $|t|$ and can be truncated with negligible loss of accuracy. Let us be more specific.

Let $0 < \kappa \leq 1$ and $\delta > 0$. Let $c_\delta = \log \frac{1}{\delta}$. Let $I_\xi = \cup_{j \in \mathbb{Z}} [\xi + j - \frac{1-\kappa/2}{2B}, \xi + j + \frac{1-\kappa/2}{2B}]$. Define our window function $W \in \mathbb{C}^{\mathbb{Z}}$ or $\hat{W}(\xi) : [0,1) \to \mathbb{R}$ as

$$W_t = e^{-t^2/2\sigma_t^2} \chi_t^{\frac{1-\kappa/2}{B}}; \quad \hat{W}(\xi) = \frac{1}{\sigma_f \sqrt{2\pi}} \int_{I_\xi} e^{-\eta^2/2\sigma_f^2} d\eta \text{ where} \tag{4.17}$$

$$\sigma_f = \frac{\kappa}{4B\sqrt{2c_\delta}}; \quad \sigma_t = \frac{1}{2\pi\sigma_f} = \frac{2B\sqrt{2c_\delta}}{\pi\kappa} = O\left(\frac{B\sqrt{c_\delta}}{\kappa}\right).$$

The parameters σ_t, σ_f are carefully chosen so that we apply the ideal amount of smoothing to the boxcar filter. This will be made precise later. Note that $\hat{W}(\xi)$ is real-valued, infinitely differentiable and is *periodic* with period 1. Next, we verify that W_t and $\hat{W}(\xi)$ are indeed Fourier transform of each other.

Proposition 4.4.1.

$$W_t = \int_{-1/2}^{1/2} \hat{W}(\xi) e^{2\pi i \xi t} d\xi; \quad \hat{W}(\xi) = \sum_{t \in \mathbb{Z}} W_t e^{-2\pi i \xi t}.$$

Proof. Let $G_t = e^{-t^2/2\sigma_t^2}$. Its Fourier transform is the periodized Gaussian

$$\begin{aligned}
\hat{G}(\xi) &= \sum_{j \in \mathbb{Z}} G_j e^{-2\pi i \xi j} \\
&= \sum_{j \in \mathbb{Z}} \left(\int_{\eta \in \mathbb{R}} \frac{1}{\sigma_f \sqrt{2\pi}} e^{-\eta^2/2\sigma_f^2} e^{2\pi i \eta j} d\eta \right) e^{-2\pi i \xi j} \\
&= \frac{1}{\sigma_f \sqrt{2\pi}} \int_{\eta \in \mathbb{R}} e^{-\eta^2/2\sigma_f^2} \sum_{j \in \mathbb{Z}} \delta(\eta - \xi - j) d\eta.
\end{aligned}$$

By the convolution theorem, the Fourier transform of $G_t \chi_t^{1/2B}$ is

$$\begin{aligned}
\hat{G} \star \hat{\chi}^{\frac{1-\kappa/2}{B}}(\xi) &= \int_{-\frac{1-\kappa/2}{2B}}^{\frac{1-\kappa/2}{2B}} \hat{G}(\xi - \zeta) d\zeta \quad \text{because } \hat{G} \text{ is even} \\
&= \frac{1}{\sigma_f \sqrt{2\pi}} \int_{-\frac{1-\kappa/2}{2B}}^{\frac{1-\kappa/2}{2B}} \int_{\eta \in \mathbb{R}} e^{-\eta^2/2\sigma_f^2} \sum_{j \in \mathbb{Z}} \delta(\eta - \xi + \zeta - j) d\eta \, d\zeta \\
&= \frac{1}{\sigma_f \sqrt{2\pi}} \sum_{j \in \mathbb{Z}} \int_{-\frac{1-\kappa/2}{2B}}^{\frac{1-\kappa/2}{2B}} e^{-(\xi - \zeta + j)^2/2\sigma_f^2} \, d\zeta \\
&= \frac{1}{\sigma_f \sqrt{2\pi}} \int_{I_\xi} e^{-\eta^2/2\sigma_f^2} d\eta.
\end{aligned}$$

□

In the time domain, we *truncate* $W \in \mathbb{R}^{\mathbb{Z}}$ to have support $[\![P]\!]$ where P is an odd integer that is divisible by B. As we will see later in Section 4.4.3, the truncation error will be $O(\delta)$ if P is sufficiently large relative to σ_t:

$$P \geq 2\sqrt{2c_\delta}\sigma_t + 1 = \Omega\left(Bc_\delta/\kappa\right). \tag{4.18}$$

Let $F \in \mathbb{R}^{\mathbb{Z}}$ be the indicator on $[\![P]\!]$. Its Fourier transform is $\hat{F}(\xi) = \frac{\sin(P\pi\xi)}{\sin(\pi\xi)}$, the periodic sinc function. Define our truncated window $\widetilde{W} \in \mathbb{C}^{\mathbb{Z}}$ or $\hat{\widetilde{W}} : [0,1) \to \mathbb{R}$ as

$$\widetilde{W}_t = W_t F_t; \quad \hat{\widetilde{W}}(\eta) = \int_{-1/2}^{1/2} \hat{W}(\xi) \hat{F}(\eta - \xi) d\xi. \tag{4.19}$$

Multiply our signal X with $\widetilde{W}$ and obtain a P-vector u such that for $t \in [\![P]\!]$, $u_{t\%P} = X_t W_t e^{2\pi i \Delta t}$ for some Δ to be decided later. Let $\hat{u}$ be the unscaled DFT of u, i.e., $\hat{u}_p = \sum_{t \in [P]} u_t e^{-2\pi i t p/P}$. The following proposition says that $\hat{u}_p$ corresponds to a bin with center $\frac{p}{P} - \Delta$.

Proposition 4.4.2. *For any $p \in [P]$,*

$$\hat{u}_p = \sum_{k \in [N]} \hat{x}_k \hat{\widetilde{W}}\left(\frac{p}{P} - \Delta - \frac{k}{N}\right).$$

Proof. Recall that for any t, $x_t = \sum_{k\in[N]} \hat{x}_k e^{2\pi ikt/N}$ and $W_t = \int_{-1/2}^{1/2} \hat{W}(\xi) e^{2\pi i\xi t} d\xi$. Thus,

$$\begin{aligned}
\hat{u}_p &= \sum_{t\in[\![P]\!]} x_t W_t e^{2\pi i\Delta t} e^{-2\pi itp/P} \\
&= \sum_{k\in[N]} \hat{x}_k \int_{-1/2}^{1/2} \hat{W}(\xi) \sum_{t\in[\![P]\!]} e^{-2\pi it(p/P-k/N-\Delta-\xi)} d\xi \\
&= \sum_{k\in[N]} \hat{x}_k \int_{-1/2}^{1/2} \hat{W}(\xi) \hat{F}\left(\frac{p}{P} - \frac{k}{N} - \Delta - \xi\right) d\xi.
\end{aligned}$$

From (4.19), we see that above integral is $\widehat{\widetilde{W}}(\frac{p}{P} - \Delta - \frac{k}{N})$. □

These bins *overlap* because their centers are $\frac{1}{P}$ apart but their width is about $\frac{1}{B}$. We only want B of the P bins. In other words, we want to *subsample* $\hat{u} \in \mathbb{C}^P$. This corresponds to periodizing u to obtain a B-vector v in the time domain. Formally, let $v_t = \sum_{j\in[P/B]} u_{jB+t}$ and $\hat{v}_k = \sum_{t\in[B]} v_t e^{2\pi itb/B}$ be the unscaled DFT of v. Define

$$\widehat{\widetilde{W}}_{b,k} = \widehat{\widetilde{W}}\left(\frac{b}{B} - \Delta - \frac{k}{N}\right); \quad \hat{W}_{b,k} = \hat{W}\left(\frac{b}{B} - \Delta - \frac{k}{N}\right). \tag{4.20}$$

The following proposition says that $\hat{v}_b$ corresponds to a bin with center $\frac{b}{B} - \Delta$. By setting $\Delta = -\frac{1}{2B}$, the center of bin b will be the center of the interval $[\frac{b}{B}, \frac{b+1}{B}]$. This is convenient because the bin that a given mode k lands in will be simply $\lfloor \frac{kB}{N} \rfloor$.

Proposition 4.4.3. *For $b \in [B]$, $\hat{v}_b = \hat{u}_{bP/B} = \sum_{k\in[N]} \hat{x}_k \widehat{\widetilde{W}}_{b,k}$.*

Proof. Apply the Poisson summation formula and Proposition 4.4.2. We omit the details. □

In `MPFFT` of Figure 4-4, the residual signal $x^r = x - z^r$ is implicitly randomly transformed before we bin it. As binning is a linear operation, it is carried out *separately* on the transformed x and the transformed $\hat{z}^r$. To bin x, we follow the aforementioned steps which are summarized in `BinInTime` of Figure 4-7, i.e., multiply by Gaussian and sinc function on $[\![P]\!]$, periodize, perform B-point FFT.

To bin $\hat{z}^r$, we apply Proposition 4.4.3 with $\hat{x}$ being $\hat{z}^r$. As $\widehat{\widetilde{W}}_{b,k}$ is hard to compute, we replace it with

$$\hat{W}'_{b,k} = \hat{W}'\left(\frac{b}{B} - \Delta - \frac{k}{N}\right); \quad \hat{W}'(\xi) = \frac{1}{\sigma_f\sqrt{2\pi}}\int_{I'_\xi} e^{-\eta^2/2\sigma_f^2}d\eta \tag{4.21}$$

where $I'_\xi = [\xi - \frac{1-\kappa/2}{2B}, \xi + \frac{1-\kappa/2}{2B}]$. Also, we incorporate the technique of updating the bins instead of the signal from [40]. Specifically, given any nonzero mode k_* in z^r, we update only the b-th bin coefficient where b is the bin that k_* lands in. The pseudocode for binning $\hat{z}^r$ is found in `BinInFrequency` of Figure 4-7. To be clear, during iteration r of `MPFFT` in Figure 4-4, Y_τ^b is equal to $Y_\tau^b[[N]]$ where for any $K \subseteq [N]$,

$$Y_\tau^b[K] := \sum_{k\in K} \hat{x}_k e^{2\pi i(\gamma k+\varphi(k)\tau)/N}\widehat{\widetilde{W}}_{b,\varphi(k)} - \sum_{k\in h^{-1}(b)\cap K} \hat{z}_k^r e^{2\pi i(\gamma k+\varphi(k)\tau)/N}\hat{W}'_{b,\varphi(k)}. \tag{4.22}$$

In Proposition 4.4.9 of Section 4.4.3, we see that $\hat{W}_{b,k}, \hat{W}'_{b,k}, \widehat{\widetilde{W}}_{b,k}$ are all close to one another such that (4.22) is approximately

$$\widetilde{Y}_\tau^b[K] := \sum_{k\in K} \hat{x}_k^r e^{2\pi i(\gamma k+\varphi(k)\tau)/N}\hat{W}_{b,\varphi(k)}. \tag{4.23}$$

We introduce a $K \subseteq [N]$ because in Section 4.5, it is more convenient to bound the contributions to Y_τ^b by different subsets of $[N]$, e.g., the nonheavy modes of x^r. If $K = \{k_*\}$ is a singleton set, we abuse notation by denoting $Y_\tau^b[\{k_*\}]$ as $Y_\tau^b[k_*]$. Similarly, denote $\widetilde{Y}_\tau^b[\{k_*\}]$ as $\widetilde{Y}_\tau^b[k_*]$.

4.4.2 Faster binning

The speed bottleneck of MPFFT lies in binning. It is therefore worthwhile to optimize the binning procedure as much as possible. Below, we list two changes that speed up binning significantly in our implementation of MPFFT.

Firstly, in `BinInTime` of Figure 4-7, instead of running a single inner loop over

procedure BINTIMEFAST($x \in \mathbb{C}^N, \alpha, \beta, \gamma, \mathcal{H}, B, \delta, \kappa$) ▷ $\mathcal{H} = \{j2^{\ell M}B : j \in [J], \ell \in [L]\}$
  Compute W_t by (4.17) for all $t \in [\![P]\!]$
  for $\tau \in \mathcal{H}$ **do**
    for $t \in [\![P]\!]$ **do**
      $c \leftarrow e^{2\pi i(\Delta t+\beta(t+\tau)/N)}W_t$ where $\Delta = -\frac{1}{2B}$
      $d_1 \leftarrow x_{(\alpha(t+\tau)+\gamma)\%N}$ ▷ $y_t^R = \frac{1}{2}(y_t + y_{-t})$; $y_t^I = \frac{1}{2i}(y_t - y_{-t})$
      $d_2 \leftarrow x_{(-\alpha(t+\tau)+\gamma)\%N}$ ▷ $y_t^{R,\tau} = y_{t+\tau}^R$; $y_t^{R,\tau} = y_{t+\tau}^R$
      $u_{t\%P}^R \leftarrow \frac{1}{2}c(d_1 + d_2)$ ▷ $u_t^R = y_t^{R,\tau}W_t e^{2\pi i\Delta t}$
      $u_{t\%P}^I \leftarrow \frac{1}{2i}c(d_1 - d_2)$ ▷ $u_t^I = y_t^{I,\tau}W_t e^{2\pi i\Delta t}$
    end for
    for $t \in [B]$ **do**
      $v_t^R \leftarrow \sum_{j\in[P/B]} u_{jB+t}^R$
      $v_t^I \leftarrow \sum_{j\in[P/B]} u_{jB+t}^I$
    end for
    $\hat{v}^R \leftarrow$ StandardFFT(v^R) ▷ $\hat{v}_b^R = \sum_{t\in[B]} v_t^R e^{2\pi itb/B}$
    $\hat{v}^I \leftarrow$ StandardFFT(v^I) ▷ $\hat{v}_b^I = \sum_{t\in[B]} v_t^I e^{2\pi itb/B}$
    $Y_\tau^{R,b} \leftarrow \hat{v}_b^R$ for all $b \in [B]$
    $Y_\tau^{I,b} \leftarrow \hat{v}_b^I$ for all $b \in [B]$
  end for
  return Y^R, Y^I
end procedure
procedure BININFREQUENCYFAST($\hat{x} \in \mathbb{C}^N, \alpha, \beta, \gamma, \mathcal{H}, B, \delta$) ▷
$\mathcal{H} = \{j2^{\ell M}B : j \in [J], \ell \in [L]\}$
  Zero out $Y^R, Y^I \in \mathbb{C}^{B\times|\mathcal{H}|}$
  for $k_0 \in \operatorname{supp}\hat{x}$ **do**
    $k_1 \leftarrow \varphi(k_0)$ where $\varphi(k) = \alpha k + \beta$
    $b \leftarrow \lfloor \frac{k_1 B}{N} \rceil$ ▷ $b \in [B]$ is the bin k_0 lands in
    $c \leftarrow \hat{x}_{k_0}\hat{W}'_{b,k_1}e^{2\pi i\gamma k_0/N}$ where $\hat{W}'_{b,k}$ is defined in (4.21)
    for $\tau \in \mathcal{H}$ **do**
      $d \leftarrow e^{2\pi ik_1\tau/N}$
      $Y_\tau^{R,b} \leftarrow Y_\tau^{R,b} + \mathrm{Re}(c)d$
      $Y_{-\tau}^{I,b} \leftarrow Y_\tau^{I,b} + \mathrm{Im}(c)d$
    end for
  end for
  return Y^R, Y^I
end procedure

Figure 4-8: Faster binning by splitting y into y^R and y^I and halving the number of trigonometric evaluations. We also recommend splitting the loop over $[\![P]\!]$ in BinInTimeFast into three simpler loops as mentioned in the text.

$[\![P]\!]$, it is surprisingly much better to run three simpler loops over $[\![P]\!]$. In the first loop, compute and store all the indices $\alpha(t+\tau)+\gamma$ modulo N. In the second loop, sample x at all the indices computed in the first loop. In the third loop, multiple by the phase factor $e^{2\pi i(\Delta+\beta(t+\tau)/N)}$ and W_t. We believe by splitting into three loops, the code becomes more cache-friendly and can be unrolled more by the compiler.

Secondly, the most expensive step within binning is the computation of phase factors in `BinInTime` because evaluating sines and cosines is costly. It turns out that we can *halve* the number of trigonometric computations by exploiting the symmetry of $\mathcal{H}$ and splitting our transformed signal y into two signals with real Fourier coefficients. Let x^r be the residual signal. Here is the original schematic:

$$x^r \xrightarrow{\alpha,\beta,\gamma} y \xrightarrow{\text{bin}} Y^b \text{ for each } b \in [B_r].$$

Recall that we sample Y^b at $\mathcal{H} = \{j2^{M\ell}B_r : |j| \leq J-1, \ell \in [L]\}$. The number of phase factors that need to be computed is $(2J-1)L(P+|\operatorname{supp}\hat{z}^r|)$. Consider the following new schematic:

$$x^r \xrightarrow{\alpha,\beta,\gamma} y^R, y^I \xrightarrow{\text{bin}} Y^{R,b}, Y^{I,b} \text{ for each } b \in [B_r].$$

The signals y^R, y^I satisfy $\hat{y}^R_k = \operatorname{Re}(\hat{y}_k)$ and $\hat{y}^I_k = \operatorname{Im}(\hat{y}_k)$. Let $Y^{R,b}, Y^{I,b}$ be sub-signals obtained from binning y^R, y^I respectively. Since y^R, y^I as well as $Y^{R,b}, Y^{I,b}$ are *even* in time domain, we only need to sample them at nonnegative τ's. This essentially halves the size of $\mathcal{H}$. After sampling $Y^{R,b}, Y^{I,b}$ at nonnegative τ's, we can rebuild the original sub-signal Y^b by $Y^b_\tau = Y^{R,b}_\tau + iY^{I,b}_\tau$ if $\tau \geq 0$ and $Y^b_\tau = \overline{Y^{R,b}_{-\tau}} + i\overline{Y^{I,b}_{-\tau}}$ if $\tau < 0$. In addition, observe that y^R, y^I can be binned simultaneously using the same phase factors. The number of phase factors computed is in fact reduced to $JLP + |\operatorname{supp}\hat{z}^r|(1+JL)$, which is roughly half as before. The pseudocode is found in Figure 4-8.

4.4.3 Binning-related estimates

In this section, we derive some bounds that will be used in the analysis of MPFFT in Section 4.5. The following fact about the normal distribution (cf. Proposition A.2.3) is useful for the rest of the section. For any $z > 0$,

$$\int_z^\infty e^{-t^2/2\sigma^2} dt \leq \frac{\sigma^2}{z} e^{-z^2/2\sigma^2}. \tag{4.24}$$

Let $c_\delta = \log \frac{1}{\delta}$ which is bounded by δ when $0 < \delta < e^{-1/\pi}$. The first result controls the error made in frequency space when our window W is truncated to $\widetilde{W}$.

Proposition 4.4.4 (Truncation in time). *Let $0 < \delta < e^{-1/\pi}$. Then*

$$\left\| \widehat{\widetilde{W}} - \hat{W} \right\|_\infty \leq \frac{\delta}{\pi c_\delta} \leq \delta.$$

Proof. For any ξ, we have

$$\begin{aligned}
\left| \widehat{\widetilde{W}}(\xi) - \hat{W}(\xi) \right| &\leq \sum_{t \in \mathbb{Z}} \left| (W_t \chi_t - W_t) e^{-2\pi i \xi t} \right| \leq \sum_{|t| \geq (P+1)/2} |W_t| \\
&\leq \sum_{|t| \geq (P+1)/2} \frac{e^{-t^2/2\sigma_t^2}}{\pi |t|} \\
&\leq \frac{1}{\pi (P-1)/2} \int_{|t| \geq (P-1)/2} e^{-t^2/2\sigma_t^2} dt \\
&\leq \frac{1}{\pi (P-1)/2} \frac{2\sigma_t^2}{(P-1)/2} e^{-(P-1)^2/8\sigma_t^2} \quad \text{due to (4.24).}
\end{aligned}$$

The result follows from (4.18) which guarantees that $\frac{(P-1)/2}{\sigma_t} \geq \sqrt{2c_\delta}$. □

Next, we have a simple lemma that will be used for the next few results. Let $I_{\text{out}} = \{\eta : |\eta| \geq \frac{\kappa}{4B}\}$.

Lemma 4.4.5. *Suppose $0 < \delta < e^{-1/\pi}$. Then $\int_{I_{out}} \frac{1}{\sigma_f \sqrt{2\pi}} e^{-\eta^2/2\sigma_f^2} d\eta \leq \frac{\delta}{\sqrt{\pi c_\delta}} \leq \delta$.*

Proof. By (4.24), $\int_{I_{\text{out}}} \frac{1}{\sigma_f \sqrt{2\pi}} e^{-\eta^2/2\sigma_f^2} d\eta \leq \frac{1}{\sigma_f \sqrt{2\pi}} \frac{2\sigma_f^2}{\kappa/4B} e^{-(\kappa/4B)^2/2\sigma_f^2}$. The latter is bounded by $\frac{\delta}{\sqrt{\pi c_\delta}}$ because our choice of σ_f in (4.17) guarantees that $\frac{\kappa/4B}{\sigma_f} \geq \sqrt{2c_\delta}$. □

The next result says that $\hat{W}(\xi)$ is very small when $|\xi| \geq \frac{1}{2B}$. This is to ensure that heavy modes that land outside a bin b do not contribute much to the sub-signal Y^b.

Proposition 4.4.6 (Outside bin). *Let* $0 < \delta < e^{-1/\pi}$ *and* $c_\delta = \log \frac{1}{\delta}$. *For any* ξ *such that* $\frac{1}{2B} \leq |\xi| \leq \frac{1}{2}$, *we have* $\hat{W}(\xi) \leq \frac{\delta}{\sqrt{\pi c_\delta}} \leq \delta$.

Proof. Observe that for $\frac{1}{2B} \leq \xi \leq \frac{1}{2}$, we have $I_\xi = \cup_{j\in\mathbb{Z}}[\xi + j - \frac{1-\kappa/2}{2B}, \xi + j + \frac{1-\kappa/2}{2B}] \subseteq I_{\text{out}}$. This is because for $j = 0$, $\xi + j - \frac{1-\kappa/2}{2B} \geq \frac{1}{2B} - \frac{1-\kappa/2}{2B} \geq \frac{\kappa}{4B}$ and for $j = -1$, $\xi + j + \frac{1-\kappa/2}{2B} \leq -\frac{1}{2} + \frac{1-\kappa/2}{2B} \leq -\frac{\kappa}{4B}$ as $B \geq 1$. Apply Lemma 4.4.5 to complete the proof. □

Define the passband and *relaxed passband* as

$$\mathcal{P} = \left\{\eta : |\eta| \leq \frac{1-\kappa}{2B}\right\}; \quad \mathcal{P}'[C_{\text{win}}] = \left\{\eta : \left|\hat{W}_{b,k}\right| \geq C_{\text{win}}\right\} \text{ for some } 0 < C_{\text{win}} < 1/2. \tag{4.25}$$

The next result says that $\hat{W}(\xi)$ is very close to 1 when ξ is in the passband $\mathcal{P}$ and that $\mathcal{P}'[C_{\text{win}}]$ contains $\{\eta : |\eta| \leq \frac{1-\kappa/2}{2B}\}$ for any $\delta < 1 - 2C_{\text{win}}$. We use the passband for analysis but in our implementation of MPFFT, we use the relaxed passband. The reason is that for computational speed, we prefer using $\kappa = 1$, but when $\kappa = 1$, $\mathcal{P}$ is empty which means that we have to reject every mode.

Proposition 4.4.7 (Inside passband). *Let* $0 < \delta < e^{-1/\pi}$. *For any* $\xi \in \mathcal{P}$, $\hat{W}(\xi) \geq 1 - \frac{\delta}{\sqrt{\pi c_\delta}} \geq 1 - \delta$. *For any* $|\xi| \leq \frac{1-\kappa/2}{2B}$, $\hat{W}(\xi) \geq \frac{1}{2} - \frac{\delta}{2\sqrt{\pi c_\delta}} \geq \frac{1}{2} - \frac{\delta}{2}$.

Proof. Observe that for $0 \leq \xi \leq \frac{1-\kappa}{2B}$, we have $I_\xi \supset [\xi - \frac{1-\kappa/2}{2B}, \xi + \frac{1-\kappa/2}{2B}] \supset [-\frac{\kappa}{4B}, \frac{1-\kappa/2}{2B}] \supset \overline{I_{\text{out}}}$. The last containment is because $\kappa \leq 1$ implies $\frac{1-\kappa/2}{2B} \geq \frac{\kappa}{4B}$. It follows from $\frac{1}{\sigma_f\sqrt{2\pi}} \int_{\mathbb{R}} e^{-\eta^2/2\sigma_f^2} d\eta = 1$ and Lemma 4.4.5 that $\hat{W}(\xi) \geq 1 - \frac{\delta}{\sqrt{\pi c_\delta}} \geq 1 - \delta$.

Now, consider the second inequality of Proposition 4.4.7. When $0 \leq \xi \leq \frac{1-\kappa/2}{2B}$, we have $I_\xi \supseteq [\xi - \frac{1-\kappa/2}{2B}, \xi + \frac{1-\kappa/2}{2B}] \supseteq [0, \frac{1-\kappa/2}{2B}] \supseteq [0, \frac{\kappa}{4B}]$. Thus, $\hat{W}(\xi) \geq \frac{1}{2} - \frac{1}{2}\int_{I_{\text{out}}} \frac{1}{\sigma_f\sqrt{2\pi}} e^{-\eta^2/2\sigma_f^2} d\eta$. Apply Lemma 4.4.5. □

Next, we show that when $|\xi| \leq \frac{1}{2B}$, then $\hat{W}'(\xi)$ is close to $\hat{W}(\xi)$, where $\hat{W}'(\xi)$ is defined in (4.21).

Proposition 4.4.8 (Truncation in frequency). *Let* $0 < \delta < e^{-1/\pi}$ *and* $|\xi| \le \frac{1}{2B}$. *Then* $\left|\hat{W}'(\xi) - \hat{W}(\xi)\right| \le \frac{\delta}{\sqrt{\pi c_\delta}} \le \delta$.

Proof. Suppose $0 \le \xi \le \frac{1}{2B}$. Observe that $I_\xi \backslash I'_\xi = \cup_{j\neq 0}[\xi + j - \frac{1-\kappa/2}{2B}, \xi + j + \frac{1-\kappa/2}{2B}] \subseteq I_{\text{out}}$. This is because for $j = 1$, $\xi + j - \frac{1-\kappa/2}{2B} \ge 1 - \frac{1-\kappa/2}{2B} \ge \frac{\kappa}{4B}$ as $B \ge \frac{1}{2}$, and for $j = -1$, $\xi + j + \frac{1-\kappa/2}{2B} \le \frac{1}{2B} - 1 + \frac{1-\kappa/2}{2B} \le -\frac{\kappa}{4B}$ as $B \ge 1$. □

Combining the previous results, we can control how much (4.22) deviates from (4.23) when we restrict the residual signal to some $K \subseteq [N]$ in frequency space.

Proposition 4.4.9. *Let* $K \subseteq [N]$. *Suppose we are in iteration* r *of* **MPFFT**, *and have just run* **BinInTime** *on* x, **BinInFrequency** *on* $\hat{z}^r$ *and obtained the bin coefficients* Y. *Then for any* $b \in [B_r]$, $\tau \in \mathcal{H}$, *we have* $\left|Y_\tau^b[K] - \widetilde{Y}_\tau^b[K]\right| \le 3\delta \max(\|\hat{x}\|_1, \|\hat{z}^r\|_1)$. *If* K *is a singleton set, then* $\left|Y_\tau^b[K] - \widetilde{Y}_\tau^b[K]\right| \le 3\delta \max(\|\hat{x}\|_\infty, \|\hat{z}^r\|_\infty)$.

Proof. Write

$$\left|Y_\tau^b[K] - \widetilde{Y}_\tau^b[K]\right| \le \sum_{k\in K} |\hat{x}_k| \left|\hat{W}_{b,\varphi(k)} - \widehat{\widetilde{W}}_{b,\varphi(k)}\right| + \sum_{k\in K\cap h^{-1}(b)} |\hat{z}_k^r| \left|\hat{W}_{b,\varphi(k)} - \hat{W}'_{b,\varphi(k)}\right| + \sum_{k\in K\backslash h^{-1}(b)} |\hat{z}_k^r| \, \hat{W}_{b,k}.$$

Consider each of the three terms on the right hand side of the equation above. The first term is bounded by $\delta \|\hat{x}\|_1$ by Proposition 4.4.4. The second term is bounded by $\delta \|\hat{z}_K^r\|_1$ by Proposition 4.4.8. The third term is bounded by $\delta \|\hat{z}_K^r\|_1$ by Proposition 4.4.6. Therefore, $\left|Y_\tau^b[K] - \widetilde{Y}_\tau^b[K]\right| \le 3\delta \max(\|\hat{x}_K\|_1, \|\hat{z}_K^r\|_1)$. Trivially, $\|\hat{x}_K\|_1 \le \|\hat{x}\|_1$, $\|\hat{z}_K^r\|_1 \le \|\hat{z}^r\|_1$ and if K is singleton, $\|\hat{x}_K\|_1 \le \|\hat{x}\|_\infty$, $\|\hat{z}_K^r\|_1 \le \|\hat{z}^r\|_\infty$. □

The next result provides a bound on $\sum_{k\in[N]} \left|\widehat{\widetilde{W}}_{b,k}\right|^2$, the energy of the window. This result will be used later to control the perturbation of bin coefficients due to nonheavy modes in a bin in Proposition 4.5.5. In practice, $\frac{1}{N}\sum_{k\in[N]} \left|\widehat{\widetilde{W}}_{b,k}\right|^2 \simeq \int_{\xi=0}^1 \left|\hat{W}(\xi)\right|^2 \simeq \frac{1}{2B}$, but for the analysis of MPFFT, we use the weaker bounds below.

Proposition 4.4.10 (Energy of window). *Let* $0 < \delta < e^{-1/\pi}$ *and* $b \in [B]$. *Then* $\sum_{k\in[N]} \left|\hat{W}_{b,k}\right|^2 \le \frac{3N}{B}$.

Proof. The idea is that $\hat{W}(\xi)$ is very small outside $[-\frac{1}{2B}, \frac{1}{2B}]$ and bounded by 1 inside $[-\frac{1}{2B}, \frac{1}{2B}]$. Let $K = \left\{k \in [N] : \text{dist}(\frac{k}{N}, \frac{b}{B} - \Delta) \leq \frac{1}{2B}\right\}$. Note that $|K| \leq \lfloor \frac{1/B}{1/N} \rfloor + 1 \leq \frac{N}{B} + 1$. By Proposition 4.4.6, $\sum_{k \in K} \left|\hat{W}_{b,k}\right|^2 + \sum_{k \notin K} \left|\hat{W}_{b,k}\right|^2 \leq |K| + (N - |K|)\delta^2 \leq |K| + N\delta^2 \leq \left(\frac{N}{B} + 1\right) + N\delta^2 \leq \frac{3N}{B}$. □

For the analysis of MPFFT, we use a κ_r that decreases with r. Although this simplifies our proofs, we do not recommend using a small κ_r. The reason is that in practice, MPFFT performs just as stably even when $\kappa_r = 1$, and using a smaller κ_r will slow down `BinInTime` and the overall algorithm significantly. We will revisit this point in Section 4.6.

4.5 Analysis of MPFFT

4.5.1 Chance that a mode is identified and estimated well

Our analysis of MPFFT is adapted from [40]. The main difference is that we have to do without guarantees related to the coefficient estimation loop.

Fix an iteration r and a heavy mode k_* of the residual signal. Let $k_0 = \varphi(k_*)$ and $b = h(k_*)$. With an abuse of notation, let $S = S_r$, $B = B_r$, $\kappa = \kappa_r$, $\Lambda = \Lambda^r$, $\mathcal{E} = \mathcal{E}_r$, $f = f_r$ and $L = L_r$ just for Section 4.5.1. Here is the main result of Section 4.5.1.

Lemma 4.5.1. *Assume* $|\hat{x}_{k_*}| \geq 1/2$, $\left\|\hat{x}^r_\Lambda\right\| \leq \mathcal{E} = N^{-O(1)}$, $\max(\|\hat{x}\|_1, \|\hat{z}^r\|_1) = N^{O(1)}$, $|\Lambda| \leq S$ *and* $\delta = N^{-\Theta(1)}$ *is sufficiently small, e.g.,* $3\delta \max(\|\hat{x}\|_\infty, \|\hat{z}^r\|_\infty) \leq \frac{1}{2}\sqrt{\frac{f\mathcal{E}}{B}}$. *Run* `MPFFT` *in Figure 4-4. The probability that* k_* *is identified and* $\hat{x}_{k_*}$ *is estimated with an additive error no greater than* $\sqrt{\frac{f\mathcal{E}}{B}}$ *is at least*

$$1 - \frac{2S}{B} - \kappa - 3L\left(\frac{2^{2M+5}\mathcal{E}}{J^2}\right)^{1/3} - \frac{8L}{C_{mp}f} - \frac{8}{f} - \frac{2^M L + 1}{N}.$$

The $\frac{2^M L+1}{N}$ term is due to the frequencies being discretized and is usually unimportant because $N \gg 2^M, L$ in practice.

Proof of Lemma 4.5.1. Recall (4.22) and that $Y^b_\tau = Y^b_\tau[[N]]$. Write the sub-signal Y^b

as $Y^b = X'^b + U^b$ and $X'^b = X^b + \Delta X^b$ and $\Delta X^b = \Delta X^{b,1} + \Delta X^{b,2}$ where

$$\begin{aligned}
X_\tau^b &= Y_\tau^b[k_*], \quad \text{(true signal)} \\
\Delta X_\tau^{b,1} &= Y_\tau^b[(\Lambda \backslash (\{k_*\}) \backslash h^{-1}(b)], \quad \text{(heavy modes outside bin)} \\
\Delta X_\tau^{b,2} &= Y_\tau^b[\overline{\Lambda}], \quad \text{(nonheavy modes)} \\
U_\tau^b &= Y_\tau^b[(\Lambda \backslash \{k_*\}) \cap h^{-1}(b)]. \quad \text{(heavy modes in bin)}
\end{aligned}$$

X^b can be thought of as the *true signal* and it is contaminated by ΔX^b. The signal $\Delta X^{b,1}$ is due to heavy modes landing *outside* bin b, while the signal $\Delta X^{b,2}$ is due to nonheavy modes. On the other hand, U^b is due to heavy modes landing in the bin. We will see that with good probability, mode k_* is isolated, which means that $U_\tau^b = 0$ and `MatrixPencilMultiscale` is effectively run on X'^b.

Recall the definition of the passband in (4.25). Note that $\frac{b+1/2}{B}$ is the center of bin b. Let $\xi_0 = \frac{k_0 B}{N} \% 1$. Recall that `MatrixPencilMultiscale` attempts to identify the first LM bits of ξ_0. Let $\xi_0^\ell = (\xi_0 2^{\ell M}) \% 1$. Let $\text{III} = [2^M]/2^M$ be the "decision edges". Define the following bad events.

$$\begin{aligned}
\mathscr{E}^I &= \{k_* \text{ not isolated}\} = \left\{\left|h^{-1}(b)\right| \geq 2\right\}, \\
\mathscr{E}^O &= \left\{\frac{k_0}{N} \text{ too near to bin edges, i.e., large offset}\right\} = \left\{\frac{k_0}{N} - \frac{b+1/2}{B} \notin \mathcal{P}\right\}, \\
\mathscr{E}_\ell^D(\theta) &= \left\{\xi_0^\ell \text{ within } \theta \text{ of decision edges}\right\} = \left\{\min_{\eta \in \text{III}} \text{dist}(\xi_0^\ell, \eta) \leq \theta\right\}, \\
\mathscr{E}_\ell^M(t) &= \{\text{perturbation in input to } \texttt{MatrixPencil} \text{ at scale } \ell \text{ too large}\} \\
&= \left\{\frac{1}{2J-1} \sum_{|j| \leq J-1} \left|\Delta X_{j2^{\ell M} B}^b\right|^2 \geq t\right\}, \\
\mathscr{E}_\ell^A(t) &= \{\text{perturbation in } \|A\|_F \text{ at scale } \ell \text{ too large}\} \\
&= \left\{\left\|\mathbb{T}(\Delta X_{j2^{\ell M} B})_{|j| \leq J-1}\right\|_F^2 \geq t\right\} \\
\mathscr{E}^E(t) &= \{\text{estimation error too large}\} = \left\{\left|\Delta X_0^b\right|^2 \geq t\right\}.
\end{aligned}$$

We first bound the probability of each of the above bad events, then infer that if they

all do not hold, then k_* will be identified and estimated well.

Proposition 4.5.2 (No isolation). *Given that* $|\Lambda| \le S$, *we have* $\mathbb{P}\left(\mathscr{E}^I\right) \le 2S/B$.

Proof. Condition on $b = h(k_*)$. For any mode k, we have

$$\mathbb{P}\left(h(k) = b\right) \le \frac{1}{N}\left(\left\lfloor \frac{1/B}{1/N} \right\rfloor + 1\right) \le 1/B + 1/N.$$

Thus, the expected number of heavy modes landing in the same bin as k_* is $(|\Lambda| - 1)(1/B + 1/N) \le S/B + S/N$. Apply Markov's inequality to complete the proof. □

Proposition 4.5.3 (Large offset). *The chance that* k_0 *misses the passband region is* $\mathbb{P}\left(\mathscr{E}^O\right) \le \kappa + 1/N$.

Proof. Let $k' = (k_0 B)\%N$ which has the same distribution as k_0. By definition of $\mathcal{P}$ in (4.25), $\mathbb{P}\left(\mathscr{E}^O\right) = \mathbb{P}\left(\left|\frac{k_0}{N} - \frac{b+1/2}{B}\right| \ge \frac{1-\kappa}{2B}\right)$. Observe that by taking modulo 1, $\left|\frac{k_0 B}{N} - (b + 1/2)\right| \ge \frac{1}{2}(1-\kappa)$ is equivalent to $\left|\frac{k'}{N} - 1/2\right| \ge \frac{1}{2}(1-\kappa)$. The probability that this happens is bounded by $\frac{1}{N}\left(\lfloor \frac{\kappa}{1/N} \rfloor + 1\right) \le \kappa + 1/N$. □

Proposition 4.5.4 (Near decision edges). *For any* $\ell \in [L]$, $\mathbb{P}\left(\mathscr{E}_\ell^D(\theta)\right) \le 2^{M+1}\theta + 2^M/N$.

Proof. The proof is very similar to the proof for the previous lemma. Let $k' = (k_0 2^{\ell M} B)\%N$. Instead of avoiding an interval of width κ, we have to avoid 2^M intervals of width 2θ. Therefore, the probability that ξ_0^ℓ is too close to any decision edge is bounded by $\frac{2^M}{N}\left(\lfloor \frac{2\theta}{1/N} \rfloor + 1\right) \le 2^{M+1}\theta + 2^M/N$. Compare this with the proof of Proposition 4.2.4 and see that we have an additional $2^M/N$ term due to the discretization. □

The following is analogous to [41, Lemma 4.1]. Define

$$\mathcal{E}_* = 6\mathcal{E}/B + 52\delta^2 \max(\|\hat{x}\|_1, \|\hat{z}^r\|_1)^2. \tag{4.26}$$

Assume $\delta = N^{-\Theta(1)}$ is sufficiently small such that $\mathcal{E}_* \le 8\mathcal{E}/B$.

Proposition 4.5.5. *For any $\tau \in \mathcal{H}$, $\mathbb{E}\left|\Delta X_\tau^b\right|^2 \le \mathcal{E}_*$. Thus, for any $t > 0$,*

$$\max\left(\mathbb{P}\left(\mathscr{E}_\ell^M(t)\right), \mathbb{P}\left(\mathscr{E}_\ell^A(t)\right), \mathbb{P}\left(\mathscr{E}^E(t)\right)\right) \le \mathcal{E}_*/t.$$

Proof. Condition on $b = h(k_*)$. Let $K = \overline{\Lambda} \cup K'$ where $K' = (\Lambda \backslash \{k_*\}) \backslash h^{-1}(b)$. Note that $\Delta X_\tau^b = Y_\tau^b[K] = (Y_\tau^b[K] - \widetilde{Y}_\tau^b[K]) + \widetilde{Y}_\tau^b[\overline{\Lambda}] + \widetilde{Y}_\tau^b[K']$. By Cauchy-Schwarz, $\mathbb{E}\left|\widetilde{Y}_\tau^b[K]\right|^2 \le 4\,\mathbb{E}(Y_\tau^b[K] - \widetilde{Y}_\tau^b[K])^2 + 4\,\mathbb{E}\,\widetilde{Y}_\tau^b[K']^2 + 2\,\mathbb{E}\,\widetilde{Y}_\tau^b[\overline{\Lambda}]^2$. The first term is bounded by $4(3\delta \max(\|\hat{x}\|_1, \|\hat{z}^r\|_1))^2$ by Proposition 4.4.9. The second term is bounded by $4(\delta\,\|\hat{x}^r\|_1)^2 \le 4\delta^2(2\max(\|\hat{x}\|_1, \|\hat{z}^r\|_1))^2$ by Proposition 4.4.6. Bound the third term:

$$\begin{aligned}
\mathbb{E}^\varphi\,\mathbb{E}^\gamma\left|\widetilde{Y}_\tau^b[\overline{\Lambda}]\right|^2 &= \mathbb{E}^\varphi\,\mathbb{E}^\gamma\left|\sum_{k\in\overline{\Lambda}} \hat{x}_k^r e^{2\pi i \gamma k/N} \hat{W}_{b,\varphi(k)} e^{2\pi i \tau \varphi(k)/N}\right|^2 \\
&= \mathbb{E}^\varphi \sum_{k\in\overline{\Lambda}} |\hat{x}_k^r|^2 \left|\hat{W}_{b,\varphi(k)}\right|^2 = \sum_{k\in\overline{\Lambda}} |\hat{x}_k^r|^2\,\mathbb{E}^\varphi\left|\hat{W}_{b,\varphi(k)}\right|^2 \\
&= \sum_{k\in\overline{\Lambda}} |\hat{x}_k^r|^2 \left(\frac{1}{N}\sum_{\tilde{k}\in[N]} \left|\hat{W}_{b,\tilde{k}}\right|^2\right) \le \mathcal{E}(3/B).
\end{aligned}$$

The last inequality is due to Proposition 4.4.10. Undo the conditioning by taking expectation over $b = h(k_*)$. We have shown that $\mathbb{E}\left|\Delta X_\tau^b\right|^2 \le \mathcal{E}_*$ for any $\tau \in \mathcal{H}$. Complete the proof by applying the linearity of expectation and Markov's inequality. □

Consider the chance that k_* is identified. Recall the proof of Proposition 4.2.4. Suppose $\mathscr{E}^I$ does not happen. Then `MatrixPencilMultiscale` is effectively run on $(X'^b_{j2^\ell M B})_{|j|\le J-1, \ell\in[L]}$. Think of $(X'^b_{j2^\ell M B})_{j\in[N]}$ as a single sinusoid perturbed by noise and the single mode has coefficient $c_0 = Y_0^b[k_*]$. Suppose $\mathscr{E}^O$ does not happen. Write $|c_0| \ge \left|\widetilde{Y}^B[k_*]\right| - \left|Y_0^b[k_*] - \widetilde{Y}^B[k_*]\right|$. The first term is at least $|\hat{x}^r|_{k_*}(1-\delta)$ by Proposition 4.4.7 because k_* lands in the passband. The second term is bounded by $3\delta\max(\|\hat{x}\|_\infty, \|\hat{z}^r\|_\infty)$ due to Proposition 4.4.9. Assume $\delta = N^{-\Theta(1)}$ is sufficiently small such that $3\delta\max(\|\hat{x}\|_\infty, \|\hat{z}^r\|_\infty) \le \frac{1}{100}\left|\hat{x}_{k_*}^r\right|$ and $\delta \le 1 - \frac{1}{100} - \frac{3}{\pi}$. As a result,

$$|c_0| \ge \left|\hat{x}_{k_*}^r\right|(1-\delta) - 3\delta\max(\|\hat{x}\|_\infty, \|\hat{z}^r\|_\infty) \ge \left|\hat{x}_{k_*}^r\right|(3/\pi). \tag{4.27}$$

By Proposition 4.2.1, $\left|\Delta\xi_0^\ell\right|^2 \le \frac{\frac{1}{2J-1}\sum_{|j|\le J-1}\left|\Delta X_{j2^\ell M_B}\right|^2}{J^2\left|\hat{x}^r_{k_*}\right|^2}$ because $\frac{2J-1}{J(J-1)^2} \le \frac{8}{J^2}$ as $J \ge 2$. By Proposition 4.5.5,

$$\mathbb{P}\left(\left|\Delta\xi_0^\ell\right| \ge \theta\right) \le \mathbb{P}\left(\mathscr{E}^M(J^2\left|\hat{x}^r_{k_*}\right|^2\theta^2)\right) \le \frac{\mathcal{E}_*}{J^2\left|\hat{x}^r_{k_*}\right|^2\theta^2}.$$

Recall that $\mathbb{P}\left(\mathscr{E}_\ell^D(\theta)\right) \le 2^{M+1}\theta + 2^M/N$ by Proposition 4.5.4. Pick $\theta = \left(\frac{\mathcal{E}_*}{J^2 2^M\left|\hat{x}^r_{k_*}\right|^2}\right)^{1/3}$ and obtain

$$\mathbb{P}\left(\mathscr{E}_\ell^D(\theta)\right) + \mathbb{P}\left(\mathscr{E}^M\left(J^2\left|\hat{x}_{k_*}\right|^2\theta^2\right)\right) \le 3\cdot\left(\frac{2^{2M}\mathcal{E}_*}{J^2\left|\hat{x}^r_{k_*}\right|^2}\right)^{1/3} + 2^M/N. \tag{4.28}$$

To identify k_*, we also need to pass the collision test, i.e., we need $\mu_\ell^2 \le \frac{C_{\mathrm{mp}}f\mathcal{E}}{B}$ for all ℓ. By (4.13) in the proof of Proposition 4.3.1, $\mu_\ell^2 \le \left\|\mathbb{T}(\Delta X_{j2^\ell M_B})_{|j|\le J-1}\right\|_F^2$. Thus, the chance that μ_ℓ^2 exceeds $\frac{C_{\mathrm{mp}}f\mathcal{E}}{B}$ is bounded by

$$\mathbb{P}\left(\mathscr{E}_\ell^A\left(\frac{C_{\mathrm{mp}}f\mathcal{E}}{B}\right)\right) \le \frac{\mathcal{E}_*B}{C_{\mathrm{mp}}f\mathcal{E}} \le \frac{8}{C_{\mathrm{mp}}f}. \tag{4.29}$$

When there is mode isolation, $\hat{x}^r_{k_*}$ will be estimated as $X'^B e^{-2\pi i\gamma k_*/N}$. Control the estimation error by

$$\begin{aligned}\left|\hat{X}'^B e^{-2\pi i\gamma k_*/N} - \hat{x}^r_{k_*}\right| &\le \left|\hat{X}'^B - \hat{x}^r_{k_*}e^{2\pi i\gamma k_*/N}\right| \\ &\le \left|Y_0^b[k_*] - \hat{x}^r_{k_*}e^{2\pi i\gamma k_*/N}\right| + \left|\Delta X^B\right| \\ &\le \left|Y_0^b[k_*] - \widetilde{Y}^B[k_*]\right| + \left|\hat{x}^r_{k_*}\hat{W}_{b,k_0} - \hat{x}^r_{k_*}\right| + \left|\Delta X^B\right|. \end{aligned} \tag{4.30}$$

By Proposition 4.4.9, the first term on the right hand side of the equation above is bounded by $3\delta\max(\|\hat{x}\|_\infty, \|\hat{z}^r\|_\infty) \le \frac{1}{2}\sqrt{\frac{f\mathcal{E}}{B}}$ assuming that $\delta = N^{-\Theta(1)}$ is sufficiently small. By Proposition 4.4.7, the second term is bounded by $\left|\hat{x}^r_{k_*}\right|\delta \le 2\delta\max(\|\hat{x}\|_\infty, \|\hat{z}^r\|_\infty) \le \frac{1}{2}\sqrt{\frac{f\mathcal{E}}{B}}$ assuming that $\delta = N^{-\Theta(1)}$ is sufficiently small. Conclude that to estimate $\hat{x}^r_{k_*}$ with an error less than $2\sqrt{\frac{f\mathcal{E}}{B}}$, it suffices that $\mathscr{E}^E(\frac{f\mathcal{E}}{B})$ does

not hold. By Proposition 4.5.5,

$$\mathbb{P}\left(\mathscr{E}^E\left(\frac{f\mathcal{E}}{B}\right)\right) \leq \frac{B\mathcal{E}_*}{f\mathcal{E}} \leq \frac{8}{f}. \tag{4.31}$$

We also need to ensure that we do not reject the bin containing k_* because $\left|Y_0^b\right|$ is too small. Note that when $\mathscr{E}^E(\frac{f\mathcal{E}}{B})$ does not hold, we have $\left|Y_0^b\right| \geq \left|Y_0^b[k_*]\right| - \sqrt{\frac{f\mathcal{E}}{B}}$. Recall from (4.27) that

$$|c_0| = \left|Y_0^b[k_*]\right| \geq \left|\hat{x}^r_{k_*}\right|(1-\delta) - 3\delta\max(\|\hat{x}\|_\infty, \|\hat{z}^r\|_\infty) \geq (1-\rho)(1-\delta) - \frac{1}{2}\sqrt{\frac{f\mathcal{E}}{B}}.$$

Assume $\delta = N^{-\Theta(1)}$ is sufficiently small such that $2\delta \leq \sqrt{\frac{f\mathcal{E}}{B}}$. Hence, $|c_0| \geq 1-\rho-\sqrt{\frac{f\mathcal{E}}{B}}$ and $\left|Y_0^b\right| \geq 1-\rho-2\sqrt{\frac{f\mathcal{E}}{B}}$. Indeed, in Figure 4-4, a bin is processed only if $\left|Y_0^b\right| \geq 1-\rho-2\sqrt{\frac{f\mathcal{E}}{B}}$.

Complete the proof by doing a union bound over scale levels and over all the bad events listed earlier. This involves summing (4.28), (4.29), (4.31) and applying Proposition 4.5.2 and Proposition 4.5.3.

□

We end the section with a justification of Assumption 4.2.2. In the proof of Lemma 4.5.1, we argue that mode k_* is identified and well-estimated if a series of conditions hold *simultaneously*. In particular, $\mathscr{E}^I$ and $\mathscr{E}^A_\ell\left(\frac{C_{\text{mp}}f\mathcal{E}}{B}\right)$ must not happen. This guarantees that the perturbation in the matrix used by the matrix pencil method, or ΔA in (4.7), is small. In other words, if the first order perturbation theory is inaccurate and Assumption 4.2.2 is invalid because ΔA is too big, we would not claim that k_* is identified anyway.

4.5.2 Overall analysis of MPFFT

We now use the same notation as the rest of the paper, not Section 4.5.1. For example, S is the *initial* number of heavy modes and B is the *initial* number of bins.

Before we begin the proof of Theorem 4.1.3, we remark why we think that the $O\left(S\log N\log^2\frac{N}{S}\right)$ bound on the running time is tight. Consider the running time

of the first iteration. According to Lemma 4.5.1, the chance of failure includes a $O(L_0/f)$ term which suggests that we need $f = \Omega(L_0) = \Omega\left(\log \frac{N}{S}\right)$. The total error energy made in the first iteration is on the order of $\frac{fS}{B}$ which suggests that we need $B = \Omega\left(S \log \frac{N}{S}\right)$. As we need to bin $L_0 = \Omega\left(\log \frac{N}{S}\right)$ times and each binning takes $\Omega\left(B \log \frac{1}{\delta}\right) = \Omega\left(S \log \frac{N}{S} \log N\right)$ time, the total time is expected to be $\Omega\left(S \log N \log^2 \frac{N}{S}\right)$. Now, let us proceed with the proof of Theorem 4.1.3.

Proof of Theorem 4.1.3. Let $\Lambda^r = \Lambda_{\rho_r}(x^r)$ be the set of heavy modes at the start of iteration r. We say iteration j is successful is $|\Lambda^{j+1}| \leq |\Lambda^j| e^{-1}$ and $\left\|\hat{x}^{j+1}_{\Lambda^{j+1}}\right\|^2 \leq \mathcal{E}_{j+1}$. Define $\mathscr{E}_r$ as the event that iteration j is unsuccessful for some $j = 0, \ldots, r-1$:

$$\mathscr{E}_r = \left\{ \left|\Lambda^{j+1}\right| > \left|\Lambda^j\right| e^{-1} \text{ or } \left\|\hat{x}^{j+1}_{\Lambda^{j+1}}\right\|^2 > \mathcal{E}_{j+1} \text{ for some } j = 0, \ldots, r-1 \right\}.$$

Read $\overline{\mathscr{E}_r}$ as "iteration $0, \ldots, r-1$ are all successful". Observe that $\overline{\mathscr{E}_r}$ implies $|\Lambda^j| \leq Se^{-j} = S_j$ for $j = 1, \ldots, r$. In particular, for $R = \lfloor \log S \rfloor + 1 > \log S$, $\overline{\mathscr{E}_R}$ implies that $\left|\Lambda^R\right| \leq Se^{-R} < 1$, i.e, all heavy modes are found after R iterations. Next, we establish a few intermediate results. Let $p = 0.01$.

Proposition 4.5.6. *Suppose* $0 < \varepsilon < 1$ *and* $\overline{\mathscr{E}_r}$ *holds. Then* $\left\|\hat{x}^r_{\Lambda^r}\right\|^2 \leq \mathcal{E}(1+\varepsilon) \leq 2\mathcal{E}$.

Proof. Recall that $B_r = B(r+1)^{-2-2p}$ and $f_r = f(r+1)^{1+p}$ and $\frac{fS}{B} \leq \frac{\varepsilon}{100}$. After R iterations,

$$\begin{aligned}
\left\|\hat{x}^r_{\Lambda^r}\right\|^2 \leq \mathcal{E}_r &\leq \mathcal{E} \prod_{s=0}^{\infty} \left(1 + \frac{4f_s S_s}{B_s}\right) \leq \mathcal{E} \exp\left(\sum_{r=0}^{\infty} \frac{4f_s S_s}{B_s}\right) \\
&= \mathcal{E} \exp\left(\frac{4fS}{B} \sum_{s=0}^{\infty} (s+1)^{3+3p} e^{-s}\right) \leq \mathcal{E} \exp\left(\frac{68fS}{B}\right) \\
&\leq \mathcal{E} e^{\varepsilon(\log 2)} \leq \mathcal{E}(1+\varepsilon).
\end{aligned}$$

The last inequality is because $0 < \varepsilon < 1$. □

This is a good time to check the growth of ρ_r. We need ρ_r to be bounded by $\frac{1}{2}$ so that all heavy modes are sufficiently heavy and can be easily identified by the matrix

pencil method. Note that $R+1 \le \log S + 2 \le 3\log S$ assuming $S \ge e$. Note that $\mathcal{E}_r, \rho_r$ are predefined and independent of how well the algorithm performs.

Proposition 4.5.7. *Let $0 \le r \le R$. Suppose $S \ge e$ and $4(\varepsilon\mathcal{E})^{1/2}\frac{\log^{2.5+1.5p} S}{S} \le \frac{1}{2}$. Then $\rho_r \le \frac{1}{2}$.*

Proof. Expand our ρ_r as follows:

$$\begin{aligned}
\rho_r &= 4\sum_{s=0}^{r-1}\sqrt{\frac{f_s\mathcal{E}_s}{B_s}} \le 4\sqrt{2}\left(\frac{f\mathcal{E}}{B}\right)^{1/2}\sum_{s=0}^{R-1}(s+1)^{1.5(1+p)} \\
&< 4\sqrt{2}\left(\frac{\varepsilon\mathcal{E}}{100S}\right)^{1/2}\int_1^{R+1} x^{1.5(1+p)}dx \\
&\le 4\sqrt{2}\left(\frac{\varepsilon\mathcal{E}}{100S}\right)^{1/2}\frac{1}{2.5+1.5p}(3\log S)^{2.5+1.5p} \\
&\le 4(\varepsilon\mathcal{E})^{1/2}\frac{\log^{2.5+1.5p} S}{S} \le \frac{1}{2}.
\end{aligned}$$

□

A bin b is processed only if $\left|Y_0^b\right| \ge 1 - \rho_r - 2\sqrt{\frac{f_r\mathcal{E}_r}{B_r}}$. If the bin contains no heavy modes, this is unlikely to happen.

Proposition 4.5.8. *Consider iteration r where $0 \le r \le R-1$. Suppose $\overline{\mathcal{E}_r}$. The probability that some bin with no heavy modes landing in it is processed by MPFFT is bounded by $64\mathcal{E}$.*

Proof. By Proposition 4.5.7, $1 - \rho_r - 2\sqrt{\frac{f_r\mathcal{E}_r}{B_r}} \ge 1 - \rho_{r+1} \ge \frac{1}{2}$. Some bin with no heavy modes is processed only if $\left|\Delta X^B\right| \ge 1 - \rho_r - 2\sqrt{\frac{f_r\mathcal{E}_r}{B_r}} \ge \frac{1}{2}$ for some $b \in [B_r]$. By Proposition 4.5.5, $\mathbb{E}\left|\Delta X^B\right|^2 \le 8\mathcal{E}_r/B_r$ for any b. The latter is bounded by $16\mathcal{E}/B_r$ by Proposition 4.5.6. Apply Markov's inequality and union bound over B_r bins. □

Suppose mode k' has been *added* to $\hat{z}^r$ in iteration r. The previous paragraph rules out probabilistically the case where the mode comes from a bin with no heavy modes. Say k' comes from bin b. Let the dominant mode of bin b be $k_* = \operatorname{argmax}_k \left|Y_0^b[k]\right|$ (cf. (4.22)).

Suppose $k_* = k'$, i.e., k_* is correctly identified. The explicit check for whether k' is in the passband in Figure 4-4 ensures that $\hat{W}_{b,\varphi(k)} \geq 1 - \delta$ (cf. Proposition 4.4.7). Moreover, the bin has passed the collision test and satisfies (4.6) by Assumption 4.1.2. This ensures that $\left|Y_0^b - Y_0^b[k_*]\right| \leq \sqrt{\frac{f_r \mathcal{E}_r}{B_r}}$ whether there is mode isolation or not. As $\hat{x}^r$ is approximated as $Y_0^b e^{-2\pi i \gamma k_*/N}$, the error on mode k_* is reduced to the following in the next iteration:

$$\begin{aligned}
\left|\hat{x}_{k_*}^{r+1}\right| &= \left|\hat{x}_{k_*}^r e^{2\pi i \gamma k_*/N} - Y_0^b\right| \\
&\leq \left|\hat{x}_{k_*}^r e^{2\pi i \gamma k_*/N} - \widetilde{Y}_0^b[k_*]\right| + \left|\widetilde{Y}_0^b[k_*] - Y_0^b[k_*]\right| + \sqrt{\frac{f_r \mathcal{E}_r}{B_r}} \\
&\leq \left|\hat{x}_{k_*}^r\right| \delta + 3\delta \max(\|\hat{x}\|_\infty, \|\hat{z}^r\|_\infty) + \sqrt{\frac{f_r \mathcal{E}_r}{B_r}} \leq 2\sqrt{\frac{f_r \mathcal{E}_r}{B_r}}. \qquad (4.32)
\end{aligned}$$

The above equation is technically very similar to (4.30): apply Proposition 4.4.9 to bound $\left|\widetilde{Y}_0^b[k_*] - Y_0^b[k_*]\right|$ and Proposition 4.4.7 to bound $\left|\hat{x}_{k_*}^r e^{2\pi i \gamma k_*/N} - \widetilde{Y}_0^b[k_*]\right|$. The difference between (4.32) and (4.30) is that (4.32) is deterministic and the $Y_0^b - Y_0^b[k_*]$ term above can be due to *heavy* modes that land out of the passband region *inside* bin b, whereas the ΔX_τ^b term in (4.30) can only come from *nonheavy* modes or *heavy* modes that fall *outside* bin b.

Now, suppose $k' \neq k_*$, i.e., k' is a subdominant mode. The collision test ensures that each of the subdominant modes cannot have too much energy. In particular, (4.6) guarantees that $\left|Y_0^b[k']\right| \leq \sqrt{\frac{f_r \mathcal{E}_r}{B_r}}$. The check for k' being in the passband ensures that $\left|\hat{W}_{b,\varphi(k')}\right| \geq 1 - \delta$ by Proposition 4.4.7. By Proposition 4.4.9, $\left|\widetilde{Y}^B[k']\right| \leq \left|Y_0^b[k']\right| + 3\delta \max(\|\hat{x}\|_\infty, \|\hat{z}^r\|_\infty) \leq \frac{3}{2}\sqrt{\frac{f_r \mathcal{E}_r}{B_r}}$. But $\left|\widetilde{Y}^B[k']\right| = |\hat{x}_{k'}^r| \hat{W}_{b,\varphi(k')}$ and $\delta \leq \frac{1}{4}$ implies

$$|\hat{x}_{k'}^r| \leq \frac{1}{1-\delta}\left|\widetilde{Y}^B[k']\right| \leq 2\sqrt{\frac{f_r \mathcal{E}_r}{B_r}}. \qquad (4.33)$$

Since bin b is processed, we know that $\left|Y_0^b\right| \geq 1 - \rho_r - 2\sqrt{\frac{f_r \mathcal{E}_r}{B_r}}$. By (4.33), this wrong update must create a mode that is heavy enough for the next iteration, i.e.,

$k' \in \Lambda_{r+1}$ because

$$\left|\hat{x}^{r+1}_{k'}\right| = \left|\hat{x}^{r}_{k'} - Y^b_0 e^{-2\pi i \gamma k'/N}\right| \geq \left|Y^b_0\right| - \left|\hat{x}^{r}_{k'}\right| \geq 1 - \rho_r - 4\sqrt{\frac{f_r \mathcal{E}_r}{B_r}} = 1 - \rho_{r+1}.$$

To summarize, whenever a mode k' is added, it is either correctly identified and estimated well, or wrongly identified and reappear as a heavy mode in the next iteration. Observe that as a result,

$$\begin{aligned}\left\|\hat{x}^{r+1}_{\Lambda_{r+1}}\right\|^2 &= \left\|\hat{x}^{r+1}_{\Lambda_{r+1}\cap\Lambda^r}\right\|^2 + \left\|\hat{x}^{r+1}_{\Lambda_{r+1}\cap\overline{\Lambda^r}}\right\|^2 \\ &\leq S_r\left(\frac{4f_r\mathcal{E}_r}{B_r}\right) + \left\|\hat{x}^{r}_{\overline{\Lambda^r}}\right\|^2 \leq \left(1 + \frac{4f_r S_r}{B_r}\right)\mathcal{E}_r = \mathcal{E}_{r+1}.\end{aligned}$$

We have used (4.32) to bound the $\left\|\hat{x}^{r+1}_{\Lambda_{r+1}\cap\Lambda^r}\right\|^2$ term above.

Next, we want to show that with good probability, iteration r is successful. Assume $|\Lambda^r| \leq S_r$. Suppose less than $\frac{1}{2}(e^{-1}|\Lambda^r|)$ of the heavy modes in iteration r are not identified or estimated up to an error of $\sqrt{\frac{f_r\mathcal{E}_r}{B_r}}$. Then the number of heavy modes in iteration $r+1$ is at most $e^{-1}|\Lambda^r|$. Note that $\mathbb{P}(\mathscr{E}_0) = 0$. By Lemma 4.5.1, Proposition (4.5.8), Proposition 4.5.7,

$$\begin{aligned}\mathbb{P}(\mathscr{E}_R) &\leq \mathbb{P}\left(\mathscr{E}_1 \cap \overline{\mathscr{E}_0}\right) + \mathbb{P}\left(\mathscr{E}_2 \cap \overline{\mathscr{E}_1}\right) + \ldots + \mathbb{P}\left(\mathscr{E}_R \cap \overline{\mathscr{E}_{R-1}}\right) \leq \sum_{r=1}^{R}\mathbb{P}\left(\mathscr{E}_r \mid \overline{\mathscr{E}_{r-1}}\right) \\ &\leq R(64\mathcal{E}) \\ &+ \frac{2}{e^{-1}}\sum_{r=0}^{R-1}\left(\frac{2S_r}{B_r} + \kappa_r + 3L_r\cdot\left(\frac{2^{2M+5}\mathcal{E}_r}{J^2 B_r}\right)^{1/3} + \frac{8L_r}{C_{\mathrm{mp}}f_r} + \frac{8}{f_r} + \frac{2^M L_r + 1}{N}\right).\end{aligned}$$

As $\kappa_r = \kappa(r+1)^{-1-p}$ and $B_r = B(r+1)^{-2-2p}$, the above is the big O of

$$\begin{aligned}&\mathcal{E}\log S + \frac{S}{B} + \kappa + \left(\frac{2^{2M}\mathcal{E}}{J^2 B}\right)^{1/3}\sum_{r=0}^{R-1} L_r(r+1)^{(2+2p)/3} \\ &+ \left(\frac{1}{f}\sum_{r=0}^{R-1} L_r(r+1)^{-1-p}\right) + \frac{1}{f} + \frac{2^M}{N}\sum_{r=0}^{R-1} L_r. \qquad (4.34)\end{aligned}$$

Pick $\kappa = \Theta(1/M)$ and $f = \Theta(\log\frac{N}{S})$ and $B = \Theta\left(\frac{S}{\varepsilon}\log\frac{N}{S}\right)$ such that $B \geq \frac{100fS}{\varepsilon}$.

Three of the terms in (4.34) require further work.

First, we show that for any $p > 0$,

$$\sum_{r=0}^{R-1} L_r (r+1)^{-1-p} = O\left(\frac{1}{M} \log \frac{N}{S}\right). \tag{4.35}$$

Note that $L_r = O\left(\frac{1}{M} \log \frac{N}{B_r}\right) = O\left(L_0 + \frac{1}{M} \log r\right)$ and $f = O(L_0)$. Write the sum on the left hand side of (4.35) as $\sum_{r=1}^{R} O\left(L_0 + \frac{1}{M} \log r\right) r^{-1-p} = O(L_0) = O\left(\frac{1}{M} \log \frac{N}{S}\right)$. It follows that $\frac{1}{f} \sum_{r=0}^{R-1} L_r (r+1)^{-1-p} = O(1/M)$. Second, we show that

$$\sum_{r=0}^{R-1} L_r (r+1)^{(2+2p)/3} = O\left(\frac{1}{M} \left(\log^{2(1+p)/3+1} S\right) \left(\log \frac{N}{S}\right)\right). \tag{4.36}$$

Write the left hand side of (4.36) as $\sum_{r=1}^{R} O\left(L_0 + \frac{1}{M} \log r\right) r^{(2+2p)/3}$. Note that $L_0 \sum_{r=1}^{R} r^{(2+2p)/3} = O\left(\frac{1}{M} (\log \frac{N}{S}) R^{(2+2p)/3+1}\right)$ while $\sum_{r=1}^{R} (\log r) r^{2(1+p)/3}$ is bounded by $\left(\sum_{r=1}^{R} \log r\right) R^{2(1+p)/3} = O\left((\log R) R^{2(1+p)/3+1}\right)$. Therefore, $\sum_{r=0}^{R-1} L_r (r+1)^{(2+2p)/3}$ is on the order of $\frac{1}{M} (\log^{2(1+p)/3+1} S)(\log \frac{N}{S} + \log \log S)$. In Theorem 4.1.3, it is assumed that for some $c > 0$, $S \log^c S = O(N)$. This implies that $\log^c S = O(N/S)$ and $\log \log S = O\left(\log \frac{N}{S}\right)$ and (4.36) follows. Third, we show that

$$\sum_{r=0}^{R-1} L_r = O\left(\frac{1}{M} \log S \log \frac{N}{S}\right). \tag{4.37}$$

Like before, $\sum_{r=0}^{R-1} L_r = O\left(\sum_{r=1}^{R} L_0 + \frac{1}{M} \log r\right) = O\left(L_0 R + \frac{R \log R}{M}\right)$ which we know is $O\left(\frac{1}{M} \log S \left(\log \frac{N}{S} + \log \log S\right)\right) = O\left(\frac{1}{M} \log S \log \frac{N}{S}\right)$ as $\log \log S = O\left(\log \frac{N}{S}\right)$.

Substitute (4.35), (4.36), (4.37) and $B = \Theta(\frac{S}{\varepsilon} \log \frac{N}{S})$ into (4.34). Obtain that the failure probability is the big O of

$$\mathcal{E} \log S + \frac{1}{\log \frac{N}{S}} + \frac{1}{M} + \left(\frac{2^{2M} \mathcal{E} \varepsilon \log^{2(1+p)+2} S}{J^2 S}\right)^{1/3} \frac{1}{M} \left(\log \frac{N}{S}\right) + \frac{2^M (\log \frac{N}{S})(\log S)}{MN}.$$

The constants hidden in the O notation above can be reduced by increasing κ, f, B. However, this comes at the expense of increasing the running time. The total running

time of MPFFT is dominated by the time spent on `BinInTime`, `BinInFrequency` and `MatrixPencil`. By (4.35), the total time spent on `BinInTime` is

$$O\left(\sum_{r=0}^{R-1}\frac{L_r B_r}{\kappa_r}\log\frac{1}{\delta}\right)=O\left((\log N)\sum_{r=0}^{\infty}\frac{B}{\kappa}L_r(r+1)^{-1-p}\right)=O\left(\frac{S}{M\varepsilon}\log^2\frac{N}{S}\log N\right).$$

To be conservative, suppose each call to `MatrixPencil` runs in $O(J^3)$ time. By (4.35), the total time spent on `MatrixPencil` is

$$O\left(\sum_{r=0}^{R-1}J^3L_rB_r\right)=O\left(J^3B\sum_{r=0}^{R-1}L_r(r+1)^{-2(1+p)}\right)=O\left(\frac{J^3S}{M\varepsilon}\log^2\frac{N}{S}\right).$$

Assume `erf` takes $O(1)$ time such that `BinInFrequency` takes $O\left(\left|\operatorname{supp}\hat{z}^r\right|\right)$ time. As $\left|\operatorname{supp}\hat{z}^r\right|\leq\sum_{j=0}^{r-1}S_j=O(S)$, the total time MPFFT spends on `BinInFrequency` is $O\left(\sum_{r=0}^{R-1}L_r\left|\operatorname{supp}\hat{z}^r\right|\right)=O\left(S\sum_{r=0}^{R-1}L_r\right)=O\left(\frac{S}{M}\log S\log\frac{N}{S}\right)$ by (4.37). Sum up the running time of `BinInTime`, `MatrixPencil`, `BinInFrequency` to obtain the desired bound on MPFFT's running time.

□

4.6 Implementation and numerical results

The second form of MPFFT is listed in Figure 4-9. It differs from the first form of MPFFT in Figure 4-4 in some minor ways listed below. Note that the first form of MPFFT is not implemented but analyzed and shown to run in $O(S\log\frac{N}{S}\log^2 N)$ under certain assumptions. The second form of MPFFT is implemented. Although we do not provide any theoretical guarantees for the second form of MPFFT, numerical evidence suggests that it does run in $\widetilde{O}(S)$ time.

Firstly, we estimate $\hat{y}_{k_0}$ as $\hat{W}_{b,k_0}^{-1}\frac{1}{|\mathcal{H}|}\sum_{\tau\in\mathcal{H}}Y_\tau^b e^{-2\pi i k_0\tau/N}$ instead of Y_0^b. Assuming that $\hat{W}_{b,k_0}\simeq 1$, Proposition 4.2.3 suggests that the error in this new estimate of $\hat{y}_{k_0}$ is on the order of $(\frac{1}{|\mathcal{H}|}\sum_{\tau\in\mathcal{H}}\mathbb{E}\left|\Delta Y_\tau^b\right|^2)^{1/2}$ where ΔY_τ^b is the perturbation due to non-heavy modes. Although $\mathbb{E}\frac{1}{|\mathcal{H}|}\sum_{\tau\in\mathcal{H}}\left|\Delta Y_\tau^b\right|^2$ is the same as $\mathbb{E}\left|\Delta Y_0^b\right|^2$ (cf. Proposition 4.5.5), the *averaging* over $|\mathcal{H}|$ tends to "denoise" and improve our estimate of $\hat{y}_{k_0}$

procedure MPFFT-IMPLEMENTED($x \in \mathbb{C}^N, S, \delta, J, M, \mathcal{E}, B_{\min}, C_{\text{win}}, C_{\text{mul}}, C_{\text{bin}}, C_{\text{collide}}, C_{\text{iters}}$)

$\hat{z}^0 \leftarrow 0$

$r \leftarrow 0$

iters $\leftarrow 0$ ▷ No. of iterations where no mode is found

while iters $< C_{\text{iters}}$ **do** ▷ Start of an outer iteration

Let $\varphi(k) = \alpha k + \beta$ be uniformly chosen permutation of $[N]$ ▷ $x^r = x - z^r$

Let γ be uniformly chosen from $[N]$ ▷ $\hat{y}_{\varphi(k)} = \hat{x}^r_k e^{2\pi i \gamma k/N}$

$S_r \leftarrow S - |\operatorname{supp} \hat{z}^r|$

$B_r \leftarrow \max(B_{\min}, C_{\text{mul}} S_r)$

$\kappa_r \leftarrow 1$

$L_r \leftarrow \lfloor \log_{2M}(N/B_r) \rfloor + 1$

$\mathcal{H} \leftarrow \{j 2^{M\ell} B_r : |j| \le J - 1, \ell \in [L_r]\}$

$Y' \leftarrow \texttt{BinInTime}(x, \alpha, \beta, \gamma, \mathcal{H}, B_r, \delta, \kappa_r)$ ▷ $y^\tau_t = y_{t+\tau}$; $\hat{y}^\tau_k = \hat{y}_k e^{2\pi i k\tau/N}$

$Y'' \leftarrow \texttt{BinInFrequency}(\hat{z}^r, \alpha, \beta, \gamma, \mathcal{H}, B_r, \delta, \kappa_r)$ ▷ Bin y^τ for each $\tau \in \mathcal{H}$

$Y \leftarrow Y' - Y''$ ▷ Obtain B_r sub-signals Y^b

foundNothing $\leftarrow$ true

for $b \in [B_r]$ such that $\left|Y^b_0\right| \ge C_{\text{bin}} \sqrt{\mathcal{E}/B_r}$ **do**

Identify one mode k_0 using $\{Y^b_\tau : \tau \in \mathcal{H}\}$:

$(\xi_0, \mu_{\max}) \leftarrow \texttt{MatrixPencilMultiscale}(L_r, J, M, (Y^b_{j2^{M\ell}B_r})_{|j|\le J-1, \ell\in[L_r]})$

$k_0 \leftarrow \operatorname{round}\left(N\left(\frac{b+\xi_0}{B_r}\right)\right)$

if $\mu_{\max} > C_{\text{collide}} \sqrt{\mathcal{E}/B_r}$ **then continue** to next bin

if $\hat{W}_{b,k_0} < C_{\text{win}}$ **then continue** to next bin

Estimate $\hat{y}_{k_0}$ as $c'_0 \leftarrow \frac{1}{|\mathcal{H}|} Y^b_\tau e^{-2\pi i k_0 \tau/N}$ and $\hat{y}'_{k_0} \leftarrow \hat{W}'^{-1}_{b,k_0} c'_0$ (cf. (4.21))

$k_* \leftarrow \varphi^{-1}(k_0)$

Update our solution by $\hat{z}^{r+1}_{k_*} \leftarrow \hat{z}^r_{k_*} + \hat{y}'_{k_0} e^{-2\pi i \gamma k_*/N}$

foundNothing $\leftarrow$ false

end for

if foundNothing **then** iters $\leftarrow$ iters $+ 1$

$r \leftarrow r + 1$

end while

return $\hat{z}^R$

end procedure

Figure 4-9: The second form of MPFFT is implemented. Refer to Section 4.6.1 for its numerical tests.

significantly.

Secondly, we use the *relaxed passband* $\mathcal{P}'[C_{\text{win}}]$ in (4.25) for some $0 < C_{\text{win}} < 1/2$ and $\kappa_r = 1$ instead of a κ_r that decays exponentially. Recall that $0 < \kappa_r \leq 1$ and the cost of binning is $O\left(\frac{B_r}{\kappa_r} \log \frac{1}{\delta}\right)$, so $\kappa_r = 1$ is the best κ_r in terms of computational cost. However, when $\kappa_r = 1$ is used, Proposition 4.4.7 only guarantees us $|\mathcal{P}| \geq \frac{1-\kappa_r}{B} = 0$, i.e., no modes will be passed. On the other hand, if $\delta < 1 - 2C_{\text{win}}$, then $|\mathcal{P}'[C_{\text{win}}]| \geq \frac{1-\kappa/2}{B} = \frac{1}{2B}$, which means that at least half of the modes will be "passed" if we use the relaxed passband. Below, we see that using a relaxed passband may worsen the error by a multiplicative factor C_{win}^{-1}. To see this, repeat the steps in 4.30:

$$\begin{aligned}
&\left| \frac{Y_0^b}{W'_{b,k_0}} - \hat{x}^r_{k_*} e^{2\pi i \gamma k_* / N} \right| \\
&\leq \frac{1}{W'_{b,k_0}} \left(\left| Y_0^b - Y_0^b[k_*] \right| + \left| Y_0^b[k_*] - \widetilde{Y}^B[k_*] \right| + \left| \hat{W}_{b,k} - \hat{W}'_{b,k} \right| \left| \hat{x}^r_{k_*} \right| \right).
\end{aligned}$$

The right hand side of the equation above is very similar to (4.30). The first term is due to noise energy. The second term is small by Proposition 4.4.9. The third term is small by Proposition 4.4.8. The main difference is that the error is blown up by $W'^{-1}_{b,k_0} \gtrsim C_{\text{win}}^{-1}$. We like picking $C_{\text{win}} \simeq 0.1$ because $\mathcal{P}'[C_{\text{win}}]$ passes many more modes than $\mathcal{P}$ (cf. (4.25)) at the expense of a slight loss of accuracy.

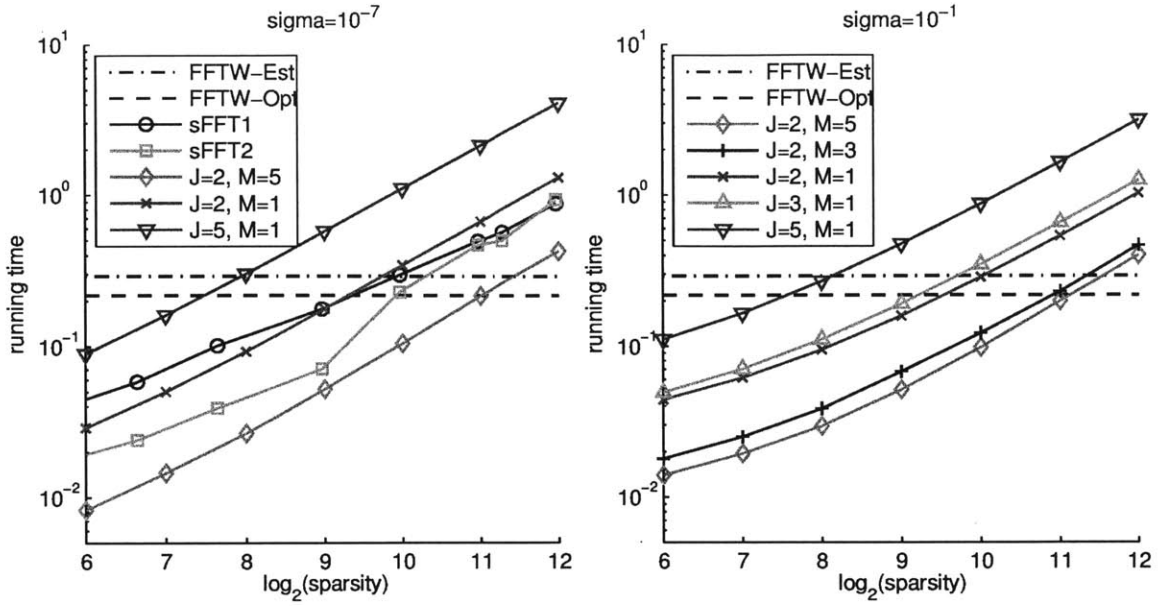
The third change is that we use *much smaller thresholds* for Y_0^b and $\mu_{\max}$. In practice, we do not know a priori that the S heavy modes have magnitude greater than 1. If there are modes with coefficients 1/3, then MPFFT will suffer from the same problem of *ghost modes* as sFFT3.0. See Section 4.1.2. In practice, the error or noise energy of iteration r is $O(\mathcal{E})$ and a bin b with no heavy modes will have $\left|Y_0^b\right| = O\left(\sqrt{\mathcal{E}/B_r}\right)$ by Proposition 4.5.5. To avoid processing such bins and adding a lot of small spurious modes, it suffices to reject bins with $\left|Y_0^b\right| \leq C_{\text{bin}}\sqrt{\mathcal{E}/B_r}$ for some $C_{\text{bin}} > 0$, say $C_{\text{bin}} = 10$. Also, from the proof of Lemma 4.5.1, we see that $\mathbb{E}\,\mu^2 \leq O(\mathcal{E}/B_r) = O(\mathcal{E}/B_r)$. In practice, μ^2 is seldom much bigger than $O(\mathcal{E}/B_r)$ when there are ≤ 1 heavy mode in the bin. Hence, it makes sense to impose a smaller threshold on $\mu_{\max}$, i.e., $\mu_{\max} \leq C_{\text{collide}}\sqrt{\mathcal{E}/B_r}$.

Let us elaborate on why we expect $\mathcal{E}_r$ to be $O(\mathcal{E})$. Suppose iteration r is successful. Then up to S_r heavy modes may be estimated within an error of $O\left(\sqrt{\mathcal{E}/B_r}\right)$ and up to $a_S S_r = S_{r+1}$ modes may be left in x^{r+1}. If these leftover modes have magnitude $O\left(\sqrt{\mathcal{E}/B_{r+1}}\right)$, which is the bin threshold for iteration $r+1$, then they will remain as ghost modes and contribute to the error energy of iteration $r+1$. Thus, the error energy in iteration $r+1$ is bounded by $\mathcal{E}_{r+1} \leq \mathcal{E}_r + O\left(\frac{S_r \mathcal{E}}{B_r}\right) + O\left(\frac{S_{r+1}\mathcal{E}}{B_{r+1}}\right)$. It follows that $\mathcal{E}_r \leq \mathcal{E}\left(1 + \sum_{r=0}^{\infty} O\left(\frac{S_r}{B_r} + \frac{S_{r+1}}{B_{r+1}}\right)\right) = O(\mathcal{E})$ for a sufficiently large B.

The last change we make is that instead of setting a conservative decay rate for the S_r, B_r's, we adopt an *adaptive strategy*: let $S_r = S - |\operatorname{supp} \hat{z}^r|$ and $B_r = C_{\text{mul}} S_r$ for some $C_{\text{mul}} \geq 1$. This is because in practice, if the parameters are set appropriately, then almost all of the modes added to z^r in iteration r are correctly identified and well-estimated. This means that $S - |\operatorname{supp} \hat{z}^r|$ is an excellent estimate of how many heavy modes are left in the residual. In case we do find a few wrong modes and S_r turns negative, we set $B_r = \max(C_s S_r, B_{\min})$ for some small $B_{\min}$, e.g., $B_{\min} = 8$. We stop the algorithm when no mode is added for C_{iters} iterations.

We recommend picking C_{mul} slightly bigger than 1. Although we argued that $C_{\text{mul}} = 1$ is optimal in Section 4.1.2, it is still safer to use a slightly larger C_{mul} in case we add some wrong modes and underestimate the number of heavy modes left in x^r. We like to remark that in our experience, if the collision detector is turned off for a C_{mul} close to 1, then too many wrong modes tend to be created, which leads to catastrophic failure.

The main reason for using a bigger C_{mul} is to reduce the effect of noise on the bin coefficients. From Proposition 4.5.5, we see that each bin coefficient is perturbed by $O\left(\sqrt{\mathcal{E}/B_r}\right)$. From (4.28) in the proof of Lemma 4.5.1, we see that the chance that we fail to identify a mode k_* grows with $O\left(\frac{\mathcal{E}}{B_r |\hat{x}_{k_*}|^2}\right)^{1/3}$. If $|\hat{x}_{k_*}|$ is too close to $\sqrt{\mathcal{E}}$, we simply have to use a larger B_r or C_{mul} so that mode identification can succeed with good probability. However, using any C_{mul} larger than 1 means a slowdown by a factor of C_{mul} as mentioned in Section 4.1.2. For this reason, we do not recommend trying to find any mode coefficient with magnitude much smaller than $\sqrt{\mathcal{E}/B_{\min}}$.



(a) (b)

Figure 4-10: Results of our first and second experiments. MPFFT is run with N fixed as the closest prime to 2^{22} while FFTW is run with $N = 2^{22}$. MPFFT with $J = 2, M = 5$ is faster than FFTW-Est when $S \lesssim 2500$ while MPFFT with $J = 2, M = 1$ is faster than FFTW-Est when $S \lesssim 1000$. Each x_t is perturbed by Gaussian noise $N(0, \sigma^2)$ where $\sigma = 10^{-7}, 10^{-1}$ in the first and second experiments respectively. The average L^1 error is on the order of $10^{-7}, 10^{-3}$ respectively.

4.6.1 Numerical tests

MPFFT is implemented using FFTW [31] for the binning and the Eigen library [37] for the matrix pencil method. It is benchmarked against FFTW's in-place complex 1D FFT routine. The size of the input signal to MPFFT are primes closest to powers of 2, whereas the size of the input signal to FFTW are *exact powers of* 2. This is to ensure a fair comparison because FFTW tends to run slower when N is prime. MPFFT, FFTW, sFFT1.0, sFFT2.0 are all compiled using the same flags, e.g., `-O3`, `-mtune=native`, `-ffast-math`. They are compiled and run on 2.67GHz Intel Xeon X5550 processors with 8Mb cache.

FFTW is run with two different options, `FFTW_ESTIMATE` or `FFTW_MEASURE`. They will be referred to as FFTW-Est and FFTW-Opt respectively. FFTW-Opt requires heavy preprocessing and always outperforms FFTW-Est. On our machines, FFTW seems to run much faster relative to sFFT1.0, sFFT2.0 than in [41]. For example, for

$N = 2^{22}$, sFFT2 is faster than FFTW-Est when $S \lesssim 1000$ here instead of $\lesssim 2000$ in [41]. Our results seem consistent with [47] which says that AAFFT0.9 is faster than FFTW when $S \lesssim 135$ instead of $S \lesssim 250$ according to [41].

In the first four experiments, we use the following random signal model. Construct a signal with S modes independently and uniformly distributed in $[N]$. Each of these S coefficients have magnitude 1 and a phase independently and uniformly distributed in $[0, 2\pi]$. In the time domain, each x_t is perturbed by Gaussians with variance σ^2.

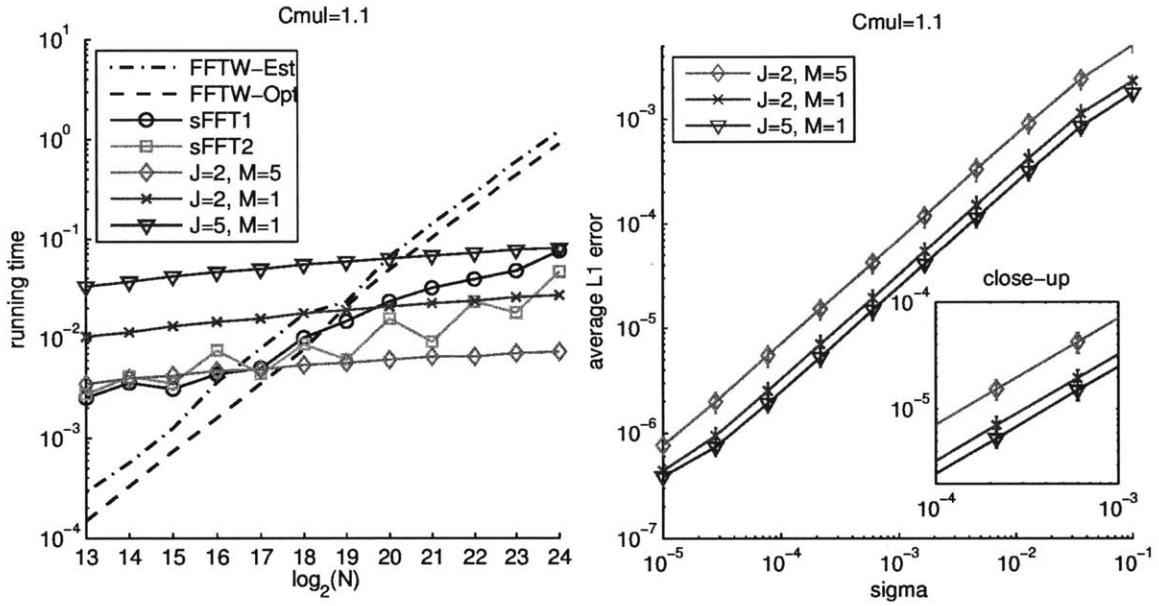
For the *first experiment*, we fix $N \simeq 2^{22}$ and vary S. Pick $\sigma = 10^{-7}$ and run MPFFT with $\delta = 3 \times 10^{-5}$, $\mathcal{E} = (3 \times 10^{-5})^2$, $B_{\text{min}} = 8$, $C_{\text{bin}} = 20$, $C_{\text{collide}} = 4$, $C_{\text{mul}} = 1.1$, $C_{\text{iters}} = 10$ and $C_{\text{win}} = 0.1$. The only variables being varied in the first experiment are J, M, S. Recall that MPFFT outputs $\hat{z}^R$ after R iterations. Define the average L^1 error as

$$\frac{1}{S} \left\| \hat{x}_\Lambda - \hat{z}^R \right\|_1 .$$

The parameter $\mathcal{E}$ is set as $(3 \times 10^{-5})^2$ instead of σ^2 because it is empirically observed that it always yields an average L^1 error on the order of 10^{-7}. In [41, 47], a similar input signal is used and the parameters of their algorithms are set such that the average L^1 error is also on the order of 10^{-7}.

Each data point of Figure 4-10a is the average running time of MPFFT over $100 \times 2^{14-\log_2 S}$ independent runs. Observe that in Figure 4-10a, MPFFT with $J = 2, M = 5$ is faster than FFTW-Est when $S \lesssim 3500$. This is more than 10 times faster than AAFFT, and hardly slower than sFFT1.0, sFFT2.0 for any N. Meanwhile, MPFFT with $J = 2, M = 1$ is faster than FFTW-Est when $S \lesssim 1000$ and its running time is comparable to that of sFFT1.0, sFFT2.0.

Our *second experiment* is similar to our first experiment except that $\sigma = 10^{-1}$, $\mathcal{E} = \sigma^2$, $\delta = 10^{-4}, C_{\text{mul}} = 1.5$ and $B_{\text{min}} = 32$. To deal with the higher level of noise compared to the first experiment, we use a larger number of bins, i.e., a larger B_{min} and C_{mul}. The average L^1 error this time is on the order of 10^{-3}. Using more bins slows down the algorithm, but as our desired accuracy is degraded from 10^{-7} to 10^{-3}, we can use a larger δ, which leads to MPFFT running at about the same speed as in



(a) (b)

Figure 4-11: The left figure is generated by our third experiment where S is fixed at 50 and N is varied. Each x_t is perturbed by Gaussian noise $N(0, \sigma^2)$ where $\sigma = 10^{-7}$. The average L^1 error is on the order of 10^{-7}. The left figure suggests that MPFFT with $J = 2, M = 5$ is faster than FFTW-Est when $N \gtrsim 70000$ while MPFFT with $J = 2, M = 1$ is faster than FFTW-Est when $N \gtrsim 260000$. The figure also suggests that MPFFT runs in $\widetilde{O}(S)$ time. The right figure is generated by our fourth experiment where $S = 50$, $N \simeq 2^{22}$ and σ is varied. The figure shows that MPFFT is robust. The errorbars indicate the square root of the empirical variance of the average L^1 error.

the first experiment.

In Figure 4-10b, we see that there is little difference in the running time for $M = 5, J = 2$ and $M = 3, J = 2$. This is due to two counterbalancing effects. On one hand, each outer iteration of MPFFT is cheaper for $M = 5$ than $M = 3$ because L_r is smaller and we need to bin fewer times. On the other hand, the chance that mode identification fails is higher for $M = 5$ than $M = 3$, which means that fewer heavy modes are found per iteration and more iterations are needed. In fact, when $M \geq 6$, mode identification fails too often and MPFFT no longer finds all the heavy modes consistently. This is not surprising because Theorem 4.1.3 or Proposition 4.2.4 suggest that the chance that mode identification fails grows exponentially with M.

Figure 4-10b also shows that for $M = 1$, the running time for $J = 2$ and $J = 3$ are about the same. This is because the number of times binning is performed is about the same for $J = 2$ and $J = 3$. Specifically, from (4.12), $\mathcal{N}_{2,1} = 2L_r - 1 \simeq 2L_r + 1 = \mathcal{N}_{3,1}$. Similarly, the running time for $J = 5$ is about twice that of $J = 2$ because $\mathcal{N}_{5,1} = 4L_r + 5 \simeq 2\mathcal{N}_{2,1}$. This underscores the fact that the binning step is the bottleneck of iterative SFT algorithms such as AAFFT and sFFT4.0.

For our *third experiment*, we fix $S = 50$ and vary N. All other parameters are set the same as the first experiment. The results are displayed in Figure 4-11a. It shows that the running time of MPFFT is $\widetilde{O}(S)$ and does not grow with $N^{1/2}$ or $N^{1/3}$ like sFFT1.0 or sFFT2.0.

For the *fourth experiment*, we fix $S = 50$, $N \simeq 2^{22}$, $\delta = 10^{-5}$, $B_{\min} = 32$, $C_{\text{mul}} = 1.1$ and vary σ, the amount of Gaussian noise. Pick $\mathcal{E} = \sigma^2$. Figure 4-11b shows that with these settings, the average L^1 error of MPFFT scales almost linearly with σ, i.e., MPFFT is robust. Observe that $J = 5, M = 1$ produces the smallest errors because MPFFT averages over more samples when estimating the mode coefficients. The average L^1 error does not decrease beyond 10^{-7} as σ decreases because $\delta = 10^{-5}$ is not small enough for the desired accuracy.

The objective of the fifth and sixth experiments is to demonstrate that the second form of MPFFT in Figure 4-9 works even when its input signal does not satisfy Assumption 4.1.1. In both experiments, we set $\mathcal{E}$ to be roughly proportional to the

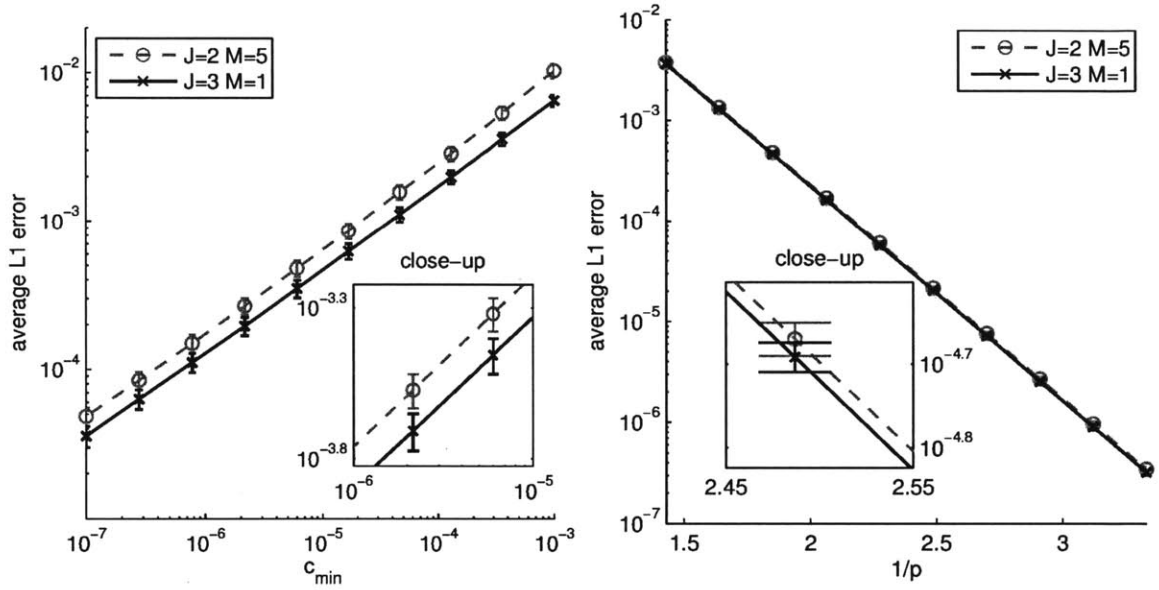


Figure 4-12: In our fifth experiment, the input signal has $2S$ modes with coefficients spaced logarithmically between $c_{\min}$ and 1 and MPFFT is asked to find $S = 250$ modes with $\mathcal{E} = c_{\min}$, which is the energy of the less energetic S modes. In our sixth experiment, the input signal is p-compressible, which means that the k-th largest coefficient has magnitude $k^{-1/p}$. MPFFT is asked to find $S = 100$ modes with $\mathcal{E} = \frac{1}{2/p-1} S^{1-2/p}$. This is the total energy minus the energy of the largest S modes. Both figures show that the average L^1 error varies linearly with $\sqrt{\mathcal{E}}$. The errorbars are obtained from the square roots of the empirical variance.

total energy of the signal minus the energy of its top S modes. We also pick $N \simeq 2^{22}$, $C_{\text{iters}} = 20$, $C_{\text{mul}} = 1.5$ and $B_{\min} = 32$ and add no Gaussian noise to the signal. Numerically, we observe that MPFFT almost always terminates with less than S modes found, but the average L^1 error will be roughly proportional to $\sqrt{\mathcal{E}}$. Their running times are consistent with the first experiment for $S = 100, 250$.

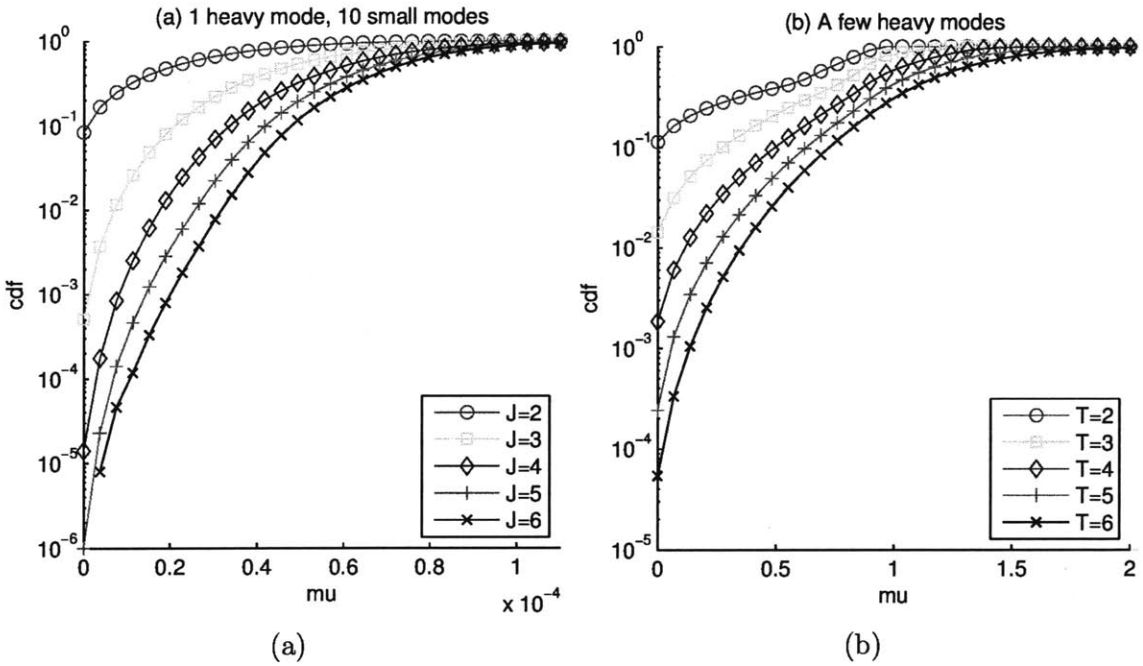
For the *fifth experiment*, the locations of $2S$ modes are independently and uniformly chosen from $[N]$, and their magnitudes are spaced logarithmically between $c_{\min}$ and 1 where $c_{\min}$ is varied between 10^{-7} and 10^{-3}. MPFFT is asked to recover only $S = 250$ modes with $\mathcal{E} = c_{\min}$. We run MPFFT 1000 times and plot the empirical mean of the average L^1 error versus $c_{\min}$ in Figure 4-12a.

For the *sixth experiment*, we consider a p-compressible signal. The k-th largest Fourier coefficient of a p-compressible signal has magnitude $k^{-1/p}$ for $1 \leq k \leq N$. In our experiment, these N modes are fully randomly permuted. Fix $S = 100$ and pick $\mathcal{E} = \frac{1}{2/p-1} S^{1-2/p}$. We run MPFFT 1000 times and plot the empirical mean of the average L^1 error versus $1/p$ in Figure 4-12b.

4.6.2 Collision detection

The collision detector plays a crucial role when C_{mul} is close to 1 and the chance of mode collision is high. In Section 4.3, we show that under some circumstances, `MatrixPencil` in Figure 4-5 will return a μ that reflects the energy of the subdominant modes in its input signal. In this section, we perform two numerical experiments to check this claim.

In the first experiment, we simulate what happens in a bin with an isolated mode after a few iterations of MPFFT: there is one heavy mode and many small modes which come from well-estimated modes in previous iterations. Here, we have 10 small modes with total energy 10^{-4}. To be concrete, our signal is $x_t = \sum_{s=0}^{10} c_s e^{2\pi i \xi_s t}$ where $c_0 = 1$ and for $1 \leq s \leq 10$, $c_s = (-1)^s \cdot 10^{-4}/\sqrt{10}$. The frequencies ξ_s's are independently and uniformly distributed in $[0, 1)$. We pick alternating signs for the small modes because Proposition 4.3.1 suggests that $\sum_{s=1}^{10} c_s = 0$ is the worst case scenario for our collision detector. We slowly increase J and plot the empirical cdf



(a) (b)

Figure 4-13: On the left, we fix 1 heavy mode and generate 10 small modes with frequencies uniformly chosen from $[0, 1)$. The total energy of the 10 small modes is 10^{-4} and they have alternating signs. From 10^6 trials, we obtain an empirical cdf of μ returned by `MatrixPencil` in Figure 4-5 with $Q = 1$. The left figure shows that it is very unlikely for μ to be much smaller than the total energy of the small modes. For the right figure, we fix $J = 3$ and consider a signal with T heavy modes. Each of these modes have magnitude 1. Again, we perform 10^6 trials and obtain an empirical cdf of μ. The plot shows that as T increases, it becomes very unlikely for μ to be much smaller than 1.

of μ in Figure 4-13a. The plot suggests that as J increases, it becomes extremely unlikely that μ^2 is much smaller than the total energy of the small modes. This agrees with Corollary 4.3.7.

For the second experiment, we simulate what happens in a bin with more than one heavy mode. The input signal to `MatrixPencil` is $x_t = \sum_{s=0}^{T-1} c_s e^{2\pi i \xi_s t}$ where c_s has magnitude 1 and a random phase so that $\mathbb{E} \sum_{s=0}^{T-1} c_s = 0$ and ξ_s is uniformly chosen from $[0, 1)$. Fix $J = 3$. Observe in Figure 4-13b that as T increases, it becomes very unlikely for μ to be much less than 1. This is consistent with Theorem 4.3.3 and Proposition 4.3.9.

Appendix A

A.1 Khintchine inequalities

In this section, we present some probabilistic results used in our proofs. The first theorem is used to decouple homogeneous Rademacher chaos of order 2 and can be found in [22, 64] for example.

Theorem A.1.1. *Let (u_i) and $(\tilde{u}_i)$ be two iid sequences of real-valued random variables and A_{ij} be in a Banach space where $1 \leq i, j \leq n$. There exists universal constants $C_1, C_2 > 0$ such that for any $s \geq 1$,*

$$\left(\mathbb{E} \left\| \sum_{1 \leq i \neq j \leq n} u_i u_j A_{ij} \right\|^s \right)^{1/s} \leq C_1 C_2^{1/s} \left(\mathbb{E} \left\| \sum_{1 \leq i,j \leq n} u_i \tilde{u}_j A_{ij} \right\|^s \right)^{1/s}. \tag{A.1}$$

A homogeneous *Gaussian* chaos is one that involves only products of Hermite polynomials with the same total degree. For instance, a homogeneous Gaussian chaos of order 2 takes the form $\sum_{1 \leq i \neq j \leq n} g_i g_j A_{ij} + \sum_{i=1}^n (g_i^2 - 1) A_{ii}$. It can be decoupled according to Arcones and Giné [3].

Theorem A.1.2. *Let (u_i) and $(\tilde{u}_i)$ be two iid Gaussian sequences and A_{ij} be in a Banach space where $1 \leq i, j \leq n$. There exists universal constants $C_1, C_2 > 0$ such that for any $s \geq 1$,*

$$\left(\mathbb{E} \left\| \sum_{1 \leq i \neq j \leq n} u_i u_j A_{ij} + \sum_{i=1}^n (u_i^2 - 1) A_{ii} \right\|^s \right)^{1/s} \leq C_1 C_2^{1/s} \left(\mathbb{E} \left\| \sum_{1 \leq i,j \leq s} u_i \tilde{u}_j A_{ij} \right\|^s \right)^{1/s}.$$

Remark A.1.3. *For Rademacher chaos, $C_1 = 4$ and $C_2 = 1$. For Gaussian chaos, we can integrate Equation (2.6) of [3] (with $m = 2$) to obtain $C_1 = 2^{1/2}$ and $C_2 = 2^{14}$. Better constants may be available.*

We now proceed to the Khintchine inequalties. Let $\|\cdot\|_{C_s}$ denote the s-Schatten norm. Recall that $\|A\|_{C_s} = (\sum_i |\sigma_i|^s)^{1/s}$ where σ_i is a singular value of A. The following is due to Lust-Piquard and Pisier [52, 53].

Theorem A.1.4. *Let $s \geq 2$ and (u_i) be a Rademacher or Gaussian sequence. Then for any set of matrices $\{A_i\}_{1 \leq i \leq n}$,*

$$\left(\mathbb{E} \left\| \sum_{i=1}^n u_i A_i \right\|_{C_s}^s \right)^{1/s} \leq s^{1/2} \max \left(\left\| (\sum_{i=1}^n A_i^* A_i)^{1/2} \right\|_{C_s}, \left\| (\sum_{i=1}^n A_i A_i^*)^{1/2} \right\|_{C_s} \right).$$

The factor $s^{1/2}$ above is not optimal. See for example [12] by Buchholz, or [65, 72].

In [64], Theorem A.1.4 is applied twice in a clever way to obtain a Khintchine inequality for a decoupled chaos of order 2.

Theorem A.1.5. *Let $s \geq 2$ and (u_i) and $(\tilde{u}_i)$ be two independent Rademacher or Gaussian sequences. For any set of matrices $\{A_{ij}\}_{1 \leq i,j \leq n}$,*

$$\left(\mathbb{E} \left\| \sum_{1 \leq i,j \leq n} u_i \tilde{u}_j A_{ij} \right\|_{C_s}^s \right)^{1/s} \leq 2^{1/s} s \max \left(\left\| Q^{1/2} \right\|_{C_s}, \left\| R^{1/2} \right\|_{C_s}, \|F\|_{C_s}, \|G\|_{C_s} \right)$$

where $Q = \sum_{1 \leq i,j \leq n} A_{ij}^* A_{ij}$ and $R = \sum_{1 \leq i,j \leq n} A_{ij} A_{ij}^*$ and F, G are the block matrices $(A_{ij})_{1 \leq i,j \leq n}$, $(A_{ij}^*)_{1 \leq i,j \leq n}$ respectively.

For Rademacher and Gaussian chaos, higher moments are controlled by lower moments, a property known as "hypercontractivity" [3, 22]. This leads to exponential tail bounds by Markov's inequality as we illustrate below. The same result appears as Proposition 6.5 of [65].

Proposition A.1.6. *Let X be a nonnegative random variable. Let $\sigma, c, \alpha > 0$. Sup-*

pose $(\mathbb{E}\, X^s)^{1/s} \leq \sigma c^{1/s} s^{1/\alpha}$ *for all* $s_0 \leq s < \infty$. *Then for any* $k > 0$ *and* $u \geq s_0^{1/\alpha}$,

$$\mathbb{P}\left(X \geq e^k \sigma u\right) \leq c \exp(-k u^\alpha).$$

Proof. By Markov's inequality, for any $s > 0$, $\mathbb{P}\left(X \geq e^k \sigma u\right) \leq \frac{\mathbb{E}\, X^s}{(e^k \sigma u)^s} \leq c \left(\frac{\sigma s^{1/\alpha}}{e^k \sigma u}\right)^s$. Pick $s = u^\alpha \geq s_0$ to complete the proof. □

Proposition A.1.7. *Let* (u_i) *be a Rademacher or Gaussian sequence and* C_1, C_2 *be constants obtained from Theorem A.1.1 or A.1.2. Let* $\{A_{ij}\}_{1 \leq i,j \leq n}$ *be a set of* p *by* p *matrices, and assume that the diagonal entries* A_{ii} *are positive semidefinite. Define* $M = \sum_i u_i u_j A_{ij}$ *and* $\sigma = C_1 \max(\|Q\|^{1/2}, \|R\|^{1/2}, \|F\|, \|G\|)$ *where* Q, R, F, G *are as defined in Theorem A.1.5. Then*

$$\mathbb{P}\left(\|M - \mathbb{E}\, M\| \geq e\sigma u\right) \leq (2C_2 np) \exp(-u).$$

Proof. We will prove the Gaussian case first. Recall that the s-Schatten and spectral norms are equivalent: for any $A \in \mathbb{C}^{r \times r}$, $\|A\| \leq \|A\|_{C_s} \leq r^{1/s} \|A\|$. Apply the decoupling inequality, that is Theorem A.1.2, and deduce that for any $s \geq 2$,

$$(\mathbb{E}\, \|M - N\|^s)^{1/s} \leq C_1 C_2^{1/s} \left(\mathbb{E} \left\| \sum_{1 \leq i,j \leq n} u_i \tilde{u}_j A_{ij} \right\|_{C_s}^s \right)^{1/s}.$$

Invoke Khintchine's inequality, that is Theorem A.1.5, and obtain

$$\begin{aligned}
(\mathbb{E}\, \|M - N\|^s)^{1/s} &\leq C_1 (2C_2)^{1/s} s \max(\left\|Q^{1/2}\right\|_{C_s}, \left\|R^{1/2}\right\|_{C_s}, \|F\|_{C_s}, \|G\|_{C_s}) \\
&\leq C_1 (2C_2 np)^{1/s} s \max(\|Q\|^{1/2}, \|R\|^{1/2}, \|F\|, \|G\|) \\
&\leq \sigma (2C_2 np)^{1/s} s.
\end{aligned}$$

Apply Proposition A.1.6 with $c = 2C_2 np$ and $k = \alpha = 1$ to complete the proof for the Gaussian case. For the Rademacher case, we take similar steps. First, decouple $(\mathbb{E}\, \|M - N\|^s)^{1/s}$ using Theorem A.1.1. This leaves us a sum that excludes the A_{ii}'s. Apply Khintchine's inequality with the A_{ii}'s zeroed. Of course, Q, R, F, G in Propo-

sition A.1.5 will not contain any A_{ii}'s, but this does not matter because $A_{ii}^* A_{ii}$ and $A_{ii} A_{ii}^*$ and A_{ii} are all positive semidefinite for any $1 \le i \le n$ and we can add them back. For example, $\|(A_{ij})_{1 \le i \ne j \le n}\| \le \|(A_{ij})_{1 \le i,j \le n}\|$ as block matrices. □

A.2 Other probabilistic inequalities

Theorem A.2.1 (Theorem 1.6 of [74]). *Consider a finite sequence $\{G_k\}$ of independent, random, Hermitian matrices with dimension d. Assume $\mathbb{E} G_k = 0$ and $\|G_k\| \le R$. Let $\sigma^2 = \|\sum_k \mathbb{E} G_k^2\|$. For any $t > 0$,*

$$\mathbb{P}\left(\left\|\sum_k G_k\right\| \ge T\right) \le 2d \exp\left(-\frac{T^2/2}{\sigma^2 + RT/3}\right).$$

Corollary A.2.2. *Assume the same set-up as Theorem A.2.1. Let $C_4 = 4$, $C_5 = \sqrt{2(1 + \frac{C_4}{3})}$ and $C_6 = (C_4/C_5)^2 = 24/7$. For any $0 < t < C_6 \frac{\sigma^2}{R^2}$,*

$$\mathbb{P}\left(\left\|\sum_k G_k\right\| \ge C_5 \sigma t^{1/2}\right) \le 2de^{-t}.$$

Proof. Apply Theorem A.2.1 with $T = \sigma t^{1/2} C_5$. Our upper bound on t ensures that $T \le C_4 \frac{\sigma^2}{R}$. Thus, the exponent $\frac{T^2/2}{\sigma^2 + RT/3}$ is at least $\frac{T^2/2}{\sigma^2(1+C_4/3)} = \frac{\sigma^2 t C_5^2/2}{\sigma^2(1+C_4/3)} = t$.

□

Here is an elementary result about the tail of a Gaussian distribution.

Proposition A.2.3. *For any $z > 0$, $\int_z^\infty e^{-t^2/2\sigma^2} dt \le \frac{\sigma^2}{z} e^{-z^2/2\sigma^2}$.*

Proof. This is proved by integration by parts. For simplicity, let $\sigma = 1$. Write $\int_z^\infty \frac{1}{t}(te^{-t^2/2})dt = -[\frac{1}{t}e^{-t^2/2}]_{t=z}^{t=\infty} - \int_z^\infty \frac{1}{t^2} e^{-t^2/2} dt \le \frac{1}{z} e^{-z^2/2}$. □

A.3 Linear algebra

Recall the definitions of $\kappa(\mathcal{B})$ and $\lambda(\mathcal{B})$ at the beginning of the paper. The following concerns probing with multiple vectors (cf. Section 2.1.3).

Proposition A.3.1. *Let $I_q \in \mathbb{C}^{q\times q}$ be the identity. Let $\mathcal{B} = \{B_1, \ldots, B_p\}$. Let $B'_j = I_q \otimes B_j$ and $\mathcal{B}' = \{B'_1, \ldots, B'_p\}$. Then $\kappa(\mathcal{B}) = \kappa(\mathcal{B}')$ and $\lambda(\mathcal{B}) = \lambda(\mathcal{B}')$.*

Proof. Define $N \in \mathbb{C}^{p\times p}$ such that $N_{jk} = \langle B_j, B_k\rangle$. Define $N' \in \mathbb{C}^{p\times p}$ such that $N'_{jk} = \langle B'_j, B'_k\rangle$. Clearly, $N' = qN$, so their condition numbers are the same and $\kappa(\mathcal{B}) = \kappa(\mathcal{B}')$.

For any $A = B_j \in \mathbb{C}^{m\times n}$ and $A' = B'_j$, we have $\frac{\|A'\|(nq)^{1/2}}{\|A'\|_F} = \frac{\|A\|(nq)^{1/2}}{\|A\|_F q^{1/2}} = \frac{\|A\|n^{1/2}}{\|A\|_F}$. Hence, $\lambda(\mathcal{B}) = \lambda(\mathcal{B}')$. □

Bibliography

[1] N. Ailon and B. Chazelle. Approximate nearest neighbors and the fast Johnson-Lindenstrauss transform. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 557–563. ACM, 2006.

[2] C. Andrieu, N. De Freitas, A. Doucet, and M. Jordan. An introduction to MCMC for machine learning. *Machine learning*, 50(1):5–43, 2003.

[3] M. A. Arconcs and E. Giné. On decoupling, series expansions, and tail behavior of chaos processes. *Journal of Theoretical Probability*, 6(1):101–122, 1993.

[4] H. Avron, P. Maymounkov, and S. Toledo. Blendenpik: Supercharging LAPACK's least-squares solver. *SIAM Journal on Scientific Computing*, 32(3):1217–1236, 2010.

[5] M. Bebendorf. Approximation of boundary element matrices. *Numerische Mathematik*, 86(4):565–589, 2000.

[6] M. Bebendorf and R. Grzhibovskis. Accelerating Galerkin BEM for linear elasticity using adaptive cross approximation. *Mathematical Methods in the Applied Sciences*, 29(14):1721–1747, 2006.

[7] M. Bebendorf and W. Hackbusch. Existence of H-matrix approximants to the inverse FE-matrix of elliptic operators with ℓ^∞-coefficients. *Numerische Mathematik*, 95(1):1–28, 2003.

[8] C. Bischof and G. Quintana-Ortí. Algorithm 782: codes for rank-revealing QR factorizations of dense matrices. *ACM Transactions on Mathematical Software (TOMS)*, 24(2):254–257, 1998.

[9] C. Boutsidis, M. Mahoney, and P. Drineas. An improved approximation algorithm for the column subset selection problem. In *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 968–977. Society for Industrial and Applied Mathematics, 2009.

[10] J. Boyd. *Chebyshev and Fourier spectral methods.* Dover Pubns, 2001.

[11] J. Bremer. A fast direct solver for the integral equations of scattering theory on planar curves with corners. *Journal of Computational Physics*, 231(4):1879–1899, 2012.

[12] A. Buchholz. Operator Khintchine inequality in non-commutative probability. *Mathematische Annalen*, 319(1):1–16, 2001.

[13] E. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.

[14] E. Candes and J. Romberg. Sparsity and incoherence in compressive sampling. *Inverse problems*, 23(3):969, 2007.

[15] E. Candes, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on pure and applied mathematics*, 59(8):1207–1223, 2006.

[16] E. J. Candès, L. Demanet, and L. Ying. Fast computation of Fourier integral operators. *SIAM Journal on Scientific Computing*, 29(6):2464–2493, 2007.

[17] T. Chan. Rank revealing QR factorizations. *Linear Algebra and Its Applications*, 88:67–82, 1987.

[18] T. Chan and T. Mathew. The interface probing technique in domain decomposition. *SIAM Journal on Matrix Analysis and Applications*, 13:212, 1992.

[19] T. Chan and D. Resasco. A survey of preconditioners for domain decomposition. Technical report, DTIC Document, 1985.

[20] E. Chang, S. Mallat, and C. Yap. Wavelet foveation. *Applied and Computational Harmonic Analysis*, 9(3):312–335, 2000.

[21] H. Cheng, Z. Gimbutas, P. Martinsson, and V. Rokhlin. On the compression of low rank matrices. *SIAM Journal on Scientific Computing*, 26(4):1389–1404, 2005.

[22] V. De la Peña and E. Giné. *Decoupling: from dependence to independence.* Springer Verlag, 1999.

[23] L. Demanet, P. Létourneau, N. Boumal, H. Calandra, J. Chiu, and S. Snelson. Matrix probing: a randomized preconditioner for the wave-equation Hessian. *Applied and Computational Harmonic Analysis*, 2011.

[24] L. Demanet and L. Ying. Discrete symbol calculus. *SIAM Review*, 53(1):71–104, 2011.

[25] L. Demanet and L. Ying. Fast wave computation via Fourier integral operators. *Mathematics of Computation*, 81(279):1455–1486, 2012.

[26] P. Drineas, M. Mahoney, and S. Muthukrishnan. Relative-error CUR matrix decompositions. *SIAM Journal on Matrix Analysis and Applications*, 30(2):844–881, 2008.

[27] A. Edelman. *Eigenvalues and condition numbers of random matrices.* PhD thesis, Massachusetts Institute of Technology, 1989.

[28] G. B. Folland. *Introduction to partial differential equations.* Princeton University Press, 1995.

[29] S. Friedland and A. Torokhti. Generalized rank-constrained matrix approximations. *SIAM Journal on Matrix Analysis and Applications*, 29(2):656–659, 2007.

[30] A. Frieze, R. Kannan, and S. Vempala. Fast Monte-Carlo algorithms for finding low-rank approximations. *Journal of the ACM (JACM)*, 51(6):1025–1041, 2004.

[31] M. Frigo and S. G. Johnson. The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231, 2005. Special issue on "Program Generation, Optimization, and Platform Adaptation".

[32] A. Gilbert, S. Muthukrishnan, and M. Strauss. Improved time bounds for near-optimal sparse Fourier representations. In *Proceedings of SPIE*, volume 5914, page 59141A, 2005.

[33] A. C. Gilbert, S. Guha, P. Indyk, S. Muthukrishnan, and M. Strauss. Near-optimal sparse Fourier representations via sampling. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 152–161. ACM, 2002.

[34] A. Gittens. The spectral norm error of the naive Nystrom extension. *Arxiv preprint arXiv:1110.5305*, 2011.

[35] S. Goreinov, E. Tyrtyshnikov, and N. Zamarashkin. A theory of pseudoskeleton approximations. *Linear Algebra and its Applications*, 261(1-3):1–21, 1997.

[36] M. Gu and S. Eisenstat. Efficient algorithms for computing a strong rank-revealing QR factorization. *SIAM Journal on Scientific Computing*, 17(4):848–869, 1996.

[37] G. Guennebaud, B. Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010.

[38] N. Hale, N. Higham, L. Trefethen, et al. Computing a^{α}, log (a), and related matrix functions by contour integrals. *SIAM Journal on Numerical Analysis*, 46(5):2505–2523, 2008.

[39] N. Halko, P.-G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

[40] H. Hassanieh, P. Indyk, D. Katabi, and E. Price. Nearly optimal sparse Fourier transform. In *Proceedings of the 44th symposium on Theory of Computing*, pages 563–578. ACM, 2012.

[41] H. Hassanieh, P. Indyk, D. Katabi, and E. Price. Simple and practical algorithm for sparse Fourier transform. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1183–1194. SIAM, 2012.

[42] N. Higham. *Functions of matrices: theory and computation.* Society for Industrial Mathematics, 2008.

[43] K. L. Ho and L. Greengard. A fast direct solver for structured linear systems by recursive skeletonization. *SIAM Journal on Scientific Computing*, 34(5):A2507–A2532, 2012.

[44] R. A. Horn and C. R. Johnson. *Matrix analysis.* Cambridge university press, 1990.

[45] Y. Hua and T. Sarkar. Matrix pencil method for estimating parameters of exponentially damped / undamped sinusoids in noise. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 38(5):814–824, 1990.

[46] Y. Hua and T. Sarkar. On SVD for estimating generalized eigenvalues of singular matrix pencil in noise. *IEEE Transactions on Signal Processing*, 39(4):892–900, 1991.

[47] M. Iwen, A. Gilbert, and M. Strauss. Empirical evaluation of a sub-linear time sparse DFT algorithm. *Communications in Mathematical Sciences*, 5(4):981–998, 2007.

[48] D. Karger and C. Stein. A new approach to the minimum cut problem. *Journal of the ACM*, 43(4):601–640, 1996.

[49] R. Karp and M. Rabin. Efficient randomized pattern-matching algorithms. *IBM Journal of Research and Development*, 31(2):249–260, 1987.

[50] C.-K. Li and R. Mathias. The Lidskii-Mirsky-Wielandt theorem–additive and multiplicative versions. *Numerische Mathematik*, 81(3):377–413, 1999.

[51] E. Liberty, F. Woolfe, P. Martinsson, V. Rokhlin, and M. Tygert. Randomized algorithms for the low-rank approximation of matrices. *Proceedings of the National Academy of Sciences*, 104(51):20167, 2007.

[52] F. Lust-Piquard. Inégalités de Khintchine dans Cp. *CR Acad. Sci. Paris*, 303:289–292, 1986.

[53] F. Lust-Piquard and G. Pisier. Non commutative khintchine and paley inequalities. *Arkiv för Matematik*, 29(1):241–260, 1991.

[54] M. Mahoney and P. Drineas. CUR matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences*, 106(3):697, 2009.

[55] A. W. Marshall, I. Olkin, and B. C. Arnold. *Inequalities: theory of majorization and its applications.* Springer Science+ Business Media, 2011.

[56] V. D. Milman and A. Pajor. Isotropic position and inertia ellipsoids and zonoids of the unit ball of a normed n-dimensional space. In *Geometric aspects of functional analysis*, pages 64–104. Springer, 1989.

[57] S. Negahban and M. J. Wainwright. Restricted strong convexity and weighted matrix completion: Optimal bounds with noise. *The Journal of Machine Learning Research*, 98888:1665–1697, 2012.

[58] R. Oliveira. Concentration of the adjacency matrix and of the Laplacian in random graphs with independent edges. *arXiv preprint arXiv:0911.0600*, 2009.

[59] G. Pfander and H. Rauhut. Sparsity in time-frequency representations. *Journal of Fourier Analysis and Applications*, 16(2):233–260, 2010.

[60] G. Pfander, H. Rauhut, and J. Tanner. Identification of matrices having a sparse representation. *Signal Processing, IEEE Transactions on*, 56(11):5376–5388, 2008.

[61] G. E. Pfander, H. Rauhut, and J. A. Tropp. The restricted isometry property for time–frequency structured random matrices. *Probability Theory and Related Fields*, pages 1–31, 2011.

[62] R. Platte, L. Trefethen, and A. Kuijlaars. Impossibility of fast stable approximation of analytic functions from equispaced samples. *SIAM Rev*, 53(2):308–318, 2011.

[63] K. Price and E. Cheney. Minimal projections. *Approximation Theory, A. Talbot., ed., Academic Press, NewYork*, pages 261–289, 1970.

[64] H. Rauhut. Circulant and Toeplitz matrices in compressed sensing. *Proc. SPARS*, 9, 2009.

[65] H. Rauhut. Compressive sensing and structured random matrices. *Theoretical Foundations and Numerical Methods for Sparse Recovery*, 9:1–92, 2010.

[66] M. Rudelson. Random vectors in the isotropic position. *Journal of Functional Analysis*, 164(1):60 – 72, 1999.

[67] M. Rudelson and R. Vershynin. Sampling from large matrices: An approach through geometric functional analysis. *Journal of the ACM (JACM)*, 54(4):21, 2007.

[68] A. Ruston. Auerbach's theorem and tensor products of banach spaces. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 58, pages 476–480. Cambridge Univ Press, 1962.

[69] J. Saak and P. Benner. Efficient solution of large scale Lyapunov and Riccati equations arising in model order reduction problems. *Proceedings in Applied Mathematics and Mechanics*, 8(1):10085–10088, 2008.

[70] M. Shubin. *Pseudodifferential operators and spectral theory.* Springer Verlag, 2001.

[71] A. Talwalkar and A. Rostamizadeh. Matrix coherence and the Nystrom method. In *UAI*, pages 572–579, 2010.

[72] J. Tropp. On the conditioning of random subdictionaries. *Applied and Computational Harmonic Analysis*, 25(1):1–24, 2008.

[73] J. Tropp. Improved analysis of the subsampled randomized hadamard transform. *Advances in Adaptive Data Analysis*, 3(1-2):115–126, 2011.

[74] J. Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434, 2012.

[75] E. Tyrtyshnikov. Incomplete cross approximation in the mosaic-skeleton method. *Computing*, 64(4):367–380, 2000.

[76] B.-Y. Wang and B.-Y. Xi. Some inequalities for singular values of matrix products. *Linear algebra and its applications*, 264:109–115, 1997.

[77] F. Woolfe, E. Liberty, V. Rokhlin, and M. Tygert. A fast randomized algorithm for the approximation of matrices. *Applied and Computational Harmonic Analysis*, 25(3):335–366, 2008.

[78] H. Xiao, V. Rokhlin, and N. Yarvin. Prolate spheroidal wavefunctions, quadrature and interpolation. *Inverse Problems*, 17:805, 2001.