Lecture Notes 4: The Frequency Domain

1 Fourier representations

1.1 Complex exponentials

The complex exponential or complex sinusoids is a complex-valued function. Its real part is a cosine and its imaginary part is a sine.

Definition 1.1 (Complex sinusoid).

$$\exp(ix) := \cos(x) + i\sin(x). \tag{1}$$

The argument x of a complex sinusoid is known as its *phase*. For any value of the phase, the magnitude of the complex exponential is equal to one,

$$|\exp(ix)|^2 = \cos(x)^2 + \sin(x)^2 = 1. \tag{2}$$

If we set the phase of a complex sinusoid to equal ft for fixed f, then the sinusoid is periodic with period $1/f$ and the parameter f is known as the *frequency* of the sinusoid.

Definition 1.2 (Frequency). *A complex sinusoid with frequency f is of the form*

$$h_f(t) := \exp(i2\pi ft). \tag{3}$$

Lemma 1.3 (Periodicity). *The complex exponential $h_f(t)$ with frequency f is periodic with period $1/f$.*

Proof.

$$\begin{align} h_f(t+1/f) &= \exp\left(i2\pi f\left(t+\frac{1}{f}\right)\right) \tag{4}\\ &= \exp(i2\pi ft)\exp(i2\pi) \tag{5}\\ &= h_f(t). \tag{6}\end{align}$$

□

Figure 1 shows a complex sinusoid with frequency 1 together with its real and imaginary parts.

If we consider a unit interval, complex sinusoids with integer frequencies form an orthonormal set. The choice of interval is arbitrary, since all these sinusoids are periodic with period $1/k$ for some integer k and hence also with period one.

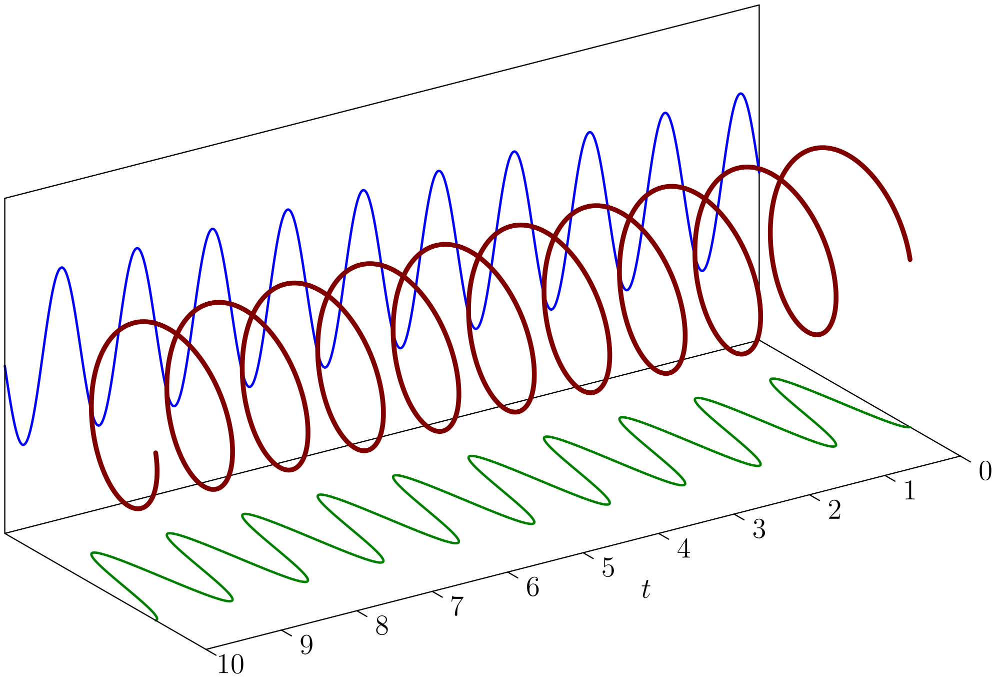


Figure 1: Complex sinusoid h_1 (dark red) plotted between 0 and 10. Its real (green) corresponds to a cosine function and its imaginary part (blue) to a sine function.

Lemma 1.4. *The family of complex sinusoids with integer frequencies*

$$h_k(t) := \exp(i2\pi kt), \qquad k \in \mathbb{Z}, \tag{7}$$

is an orthonormal set of functions on the unit interval $[-1/2, 1/2]$.

Proof. Each function has unit $\mathcal{L}_2$ norm,

$$\begin{aligned} ||h_k||_{\mathcal{L}_2}^2 &= \int_{-1/2}^{1/2} |h_k(t)|^2 \, \mathrm{d}t && (8) \\ &= 1, && (9) \end{aligned}$$

and they are all orthogonal

$$\begin{aligned} \langle h_k, h_j \rangle &= \int_{-1/2}^{1/2} h_k(t) \overline{h_j(t)} \, \mathrm{d}t && (10) \\ &= \int_{-1/2}^{1/2} \exp(i2\pi(k-j)t) \, \mathrm{d}t && (11) \\ &= \frac{\exp(i\pi(k-j)) - \exp(-i\pi(k-j))}{i2\pi} && (12) \\ &= \frac{\cos(\pi(k-j)) - \cos(\pi(k-j))}{i2\pi} && (13) \\ &= 0. && (14) \end{aligned}$$

□

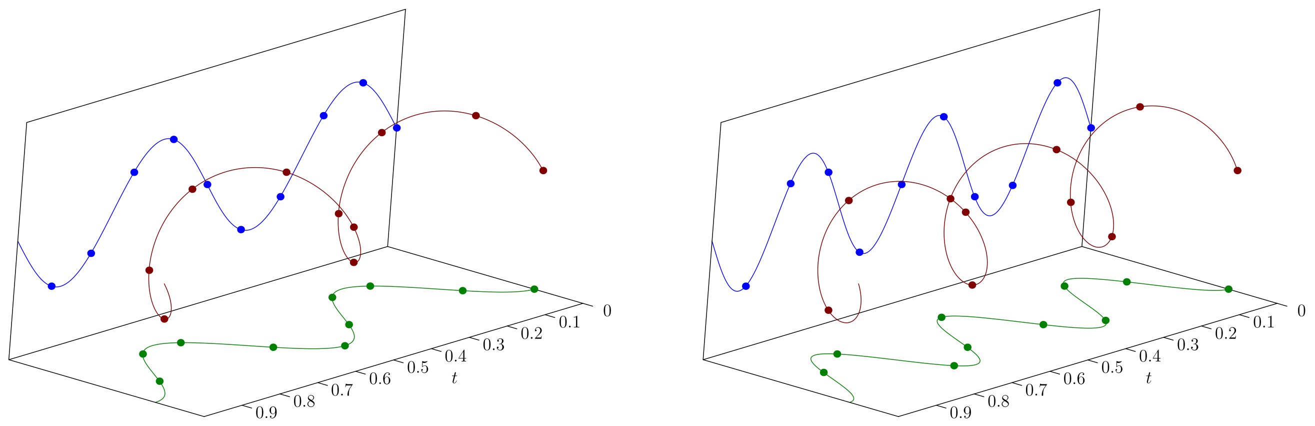


Figure 2: Discrete complex sinusoids $\vec{h}_2^{[10]}$ (left) and $\vec{h}_3^{[10]}$ (right) in red along with their real (green) and imaginary (blue) parts.

Complex sinusoids on the unit interval can be discretized by sampling their values at n equispaced points. This yields a family of discrete complex sinusoids.

Definition 1.5 (Discrete complex exponential)**.** *The discrete complex sinusoid* $\vec{h}_k^{[n]} \in \mathbb{C}^n$ *with integer frequency* k *is defined as*

$$\vec{h}_k^{[n]}[j] := \exp\left(\frac{i2\pi kj}{n}\right), \qquad 0 \leq j, k \leq n-1 \tag{15}$$

The discrete complex sinusoid is periodic in j with period n/k and in k with period n/j. In particular, they are also periodic with period n, so we only consider values in an interval of length n. Here we have fixed the interval to be from 0 to $n-1$ but this is arbitrary. Figure 2 shows $\vec{h}_2^{[10]}$ and $\vec{h}_3^{[10]}$ along with their real and imaginary parts

Discrete complex sinusoids form an orthonormal basis of $\mathbb{C}^n$ if we scale them by $1/\sqrt{n}$.

Lemma 1.6 (Orthonormal sinusoidal basis)**.** *The discrete complex exponentials* $\frac{1}{\sqrt{n}}\vec{h}_0^{[n]}, \ldots, \frac{1}{\sqrt{n}}\vec{h}_{n-1}^{[n]}$ *form an orthonormal basis of* $\mathbb{C}^n$*.*

Proof. Each vector has ℓ_2 norm equal to $\sqrt{n}$,

$$\begin{aligned} \left|\left|\vec{h}_k^{[n]}\right|\right|_2^2 &= \sum_{j=0}^{n-1} \left|\vec{h}_k^{[n]}[j]\right|^2 && (16)\\ &= \sum_{j=0}^{n-1} 1 && (17)\\ &= n, && (18) \end{aligned}$$

and they are all orthogonal

$$\left\langle \vec{h}_k^{[n]}, \vec{h}_l^{[n]} \right\rangle = \sum_{j=0}^{n-1} \vec{h}_k^{[n]}[j] \, \overline{\vec{h}_l^{[n]}[j]} \tag{19}$$

$$= \sum_{j=0}^{n-1} \exp\left(\frac{i2\pi (k-l) j}{n}\right) \tag{20}$$

$$= \frac{1 - \exp\left(\frac{i2\pi(k-l)n}{n}\right)}{1 - \exp\left(\frac{i2\pi(k-l)}{n}\right)} \tag{21}$$

$$= 0. \tag{22}$$

Since there are n vectors in the set and they are linearly independent, they form a basis of $\mathbb{C}^n$. □

1.2 Fourier series

The Fourier series of a function defined in the unit interval is the projection of the function onto the span of the complex sinusoids with integer frequencies, which are orthonormal by Lemma 1.4.

Definition 1.7 (Fourier series)**.** *The partial Fourier series of order N of a function $f \in \mathcal{L}_2[-1/2, 1/2]$ is defined as*

$$S_N\{f\} := \sum_{i=-N}^{N} F[k] \, h_k, \tag{23}$$

$$F[k] := \langle f, h_k \rangle. \tag{24}$$

The Fourier series of a function $f \in \mathcal{L}_2[-1/2, 1/2]$ is defined as

$$S\{f\} := \sum_{i=-\infty}^{\infty} F[k] \, h_k, \tag{25}$$

$$F[k] := \langle f, h_k \rangle. \tag{26}$$

For real functions, the Fourier series is a projection onto an orthonormal basis of cosine and sine functions (we omit the proof that these functions are orthogonal, which is very similar to the proof of Lemma 1.4).

Lemma 1.8. *For a real function $f \in \mathcal{L}_2[-1/2, 1/2]$ we have*

$$S\{f\} = \sum_{i=0}^{\infty} a_k \cos(2\pi k t) + b_k \sin(2\pi k t), \tag{27}$$

$$a_k := 2 \langle f, \cos(2\pi k t) \rangle, \tag{28}$$

$$b_k := 2 \langle f, \sin(2\pi k t) \rangle. \tag{29}$$

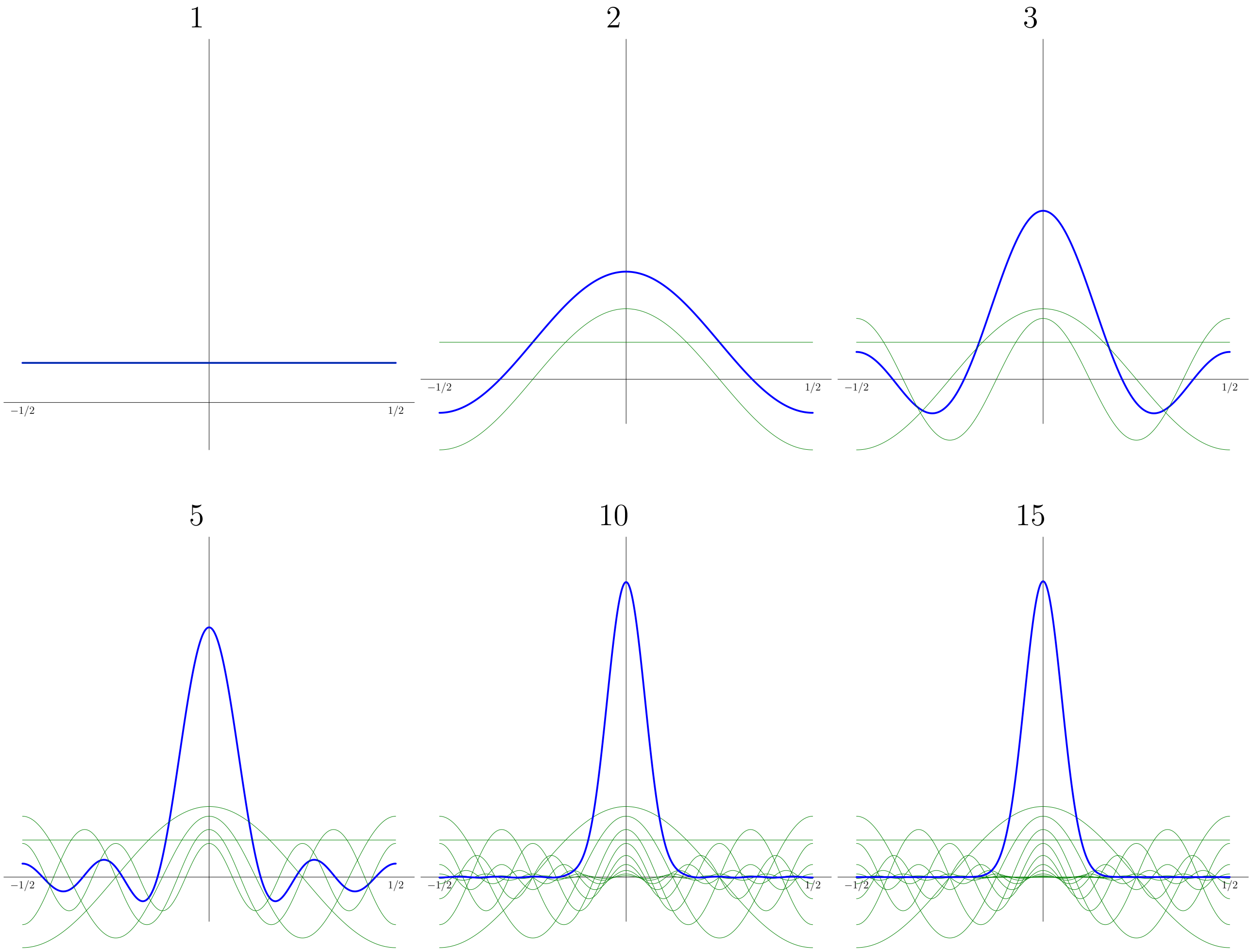


Figure 3: Partial Fourier series (blue) of the Gaussian function in Figure 5 restricted to the unit interval. The complex sinusoids used to produce the approximation are shown in green.

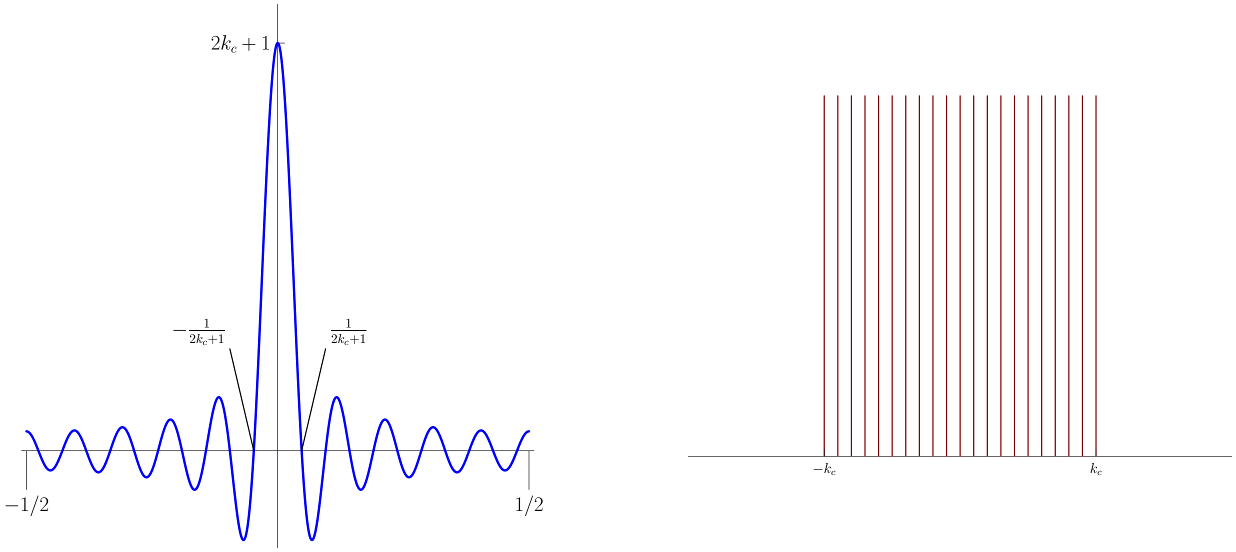


Figure 4: Dirichlet kernel with cut-off frequency k_c (left) and corresponding Fourier coefficients (right).

Figure 3 shows the nth-order partial Fourier series of a Gaussian function restricted to the unit interval for different values of n. As n increases, the approximation obtained by projecting the function onto the span of the complex sinusoids improves. Remarkably, if the function is integrable, the approximation eventually converges to the function. We omit the proof of this result, which is beyond the scope of these notes.

Theorem 1.9 (Convergence of Fourier series)**.** *For any function $f \in \mathcal{L}_2\left[-1/2, 1/2\right]$*

$$\lim_{N\to\infty} \left|\left|f - S_N\left\{f\right\}\right|\right|_{\mathcal{L}_2} = 0. \tag{30}$$

In addition, if f is continuously differentiable, the convergence is uniform and hence also pointwise, so that

$$\lim_{N\to\infty} S_N\left\{f\right\}(t) = f(t), \quad \text{for any } t. \tag{31}$$

By Theorem 1.9 we can represent any square-integrable function defined on an interval by using its Fourier coefficients, which are known as the *spectrum* of the function. It is worth noting that one can generalize this representation to functions defined on the whole real line by considering real-valued frequencies, which yields the Fourier transform. We will not discuss this further in these notes.

The Dirichlet kernel is an example of a *bandlimited* function, for which the higher end of the spectrum is zero, as illustrated in Figure 4.

Definition 1.10 (Dirichlet kernel)**.** *The Fourier coefficients of the Dirichlet kernel with cut-off frequency k_c are equal to*

$$D[k] := \begin{cases} 1 & \text{if } |k| \leq k_c \\ 0 & \text{otherwise.} \end{cases} \tag{32}$$

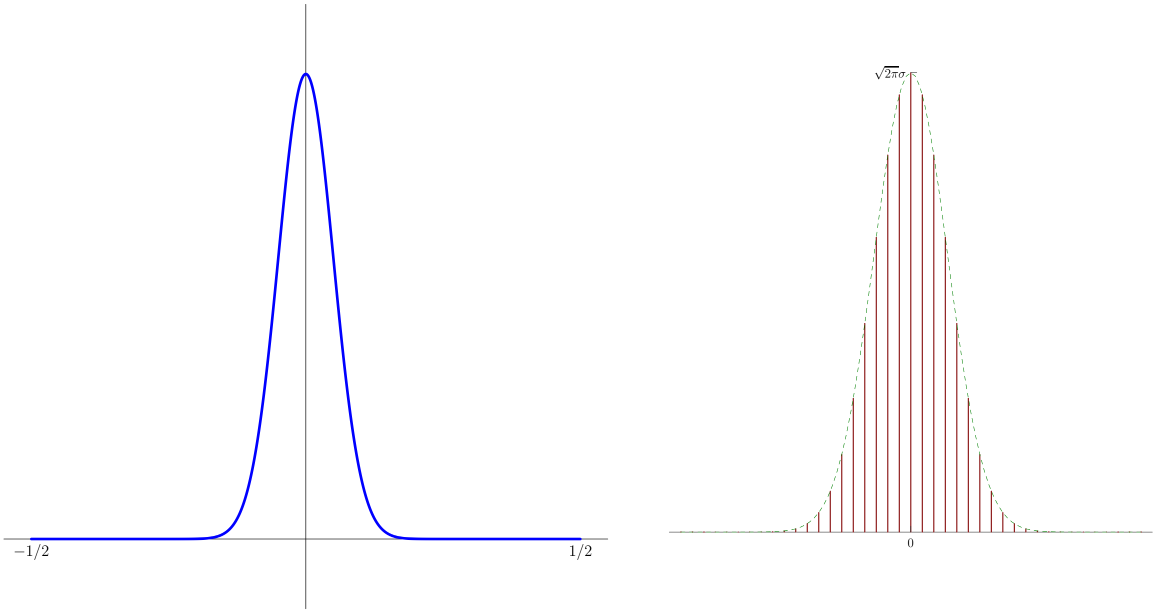


Figure 5: Gaussian function in Example 1.11 and corresponding Fourier coefficients (right).

The corresponding function is given by

$$d(t) = \sum_{n=-k_c}^{k_c} D[k]\, h_k(t) \tag{33}$$

$$= \sum_{n=-k_c}^{k_c} e^{i2\pi kt} \tag{34}$$

$$= \begin{cases} 2k_c + 1 & \text{if } t = 0, \\ \frac{\sin((2k_c+1)\pi t)}{\sin(\pi t)} & \text{otherwise}\ . \end{cases} \tag{35}$$

Note that the width of the main lobe of the Dirichlet kernel is inversely proportional to its cut-off frequency, which corresponds to the width of its spectrum. The following example shows that this is also the case for the Gaussian function.

Example 1.11 (Gaussian)**.** Consider the Gaussian function

$$g(t) = \exp\left(-\frac{t^2}{\sigma^2}\right), \tag{36}$$

restricted to the interval $[-1/2, 1/2]$. If σ is small with respect to the length of the interval, then we can approximate its spectrum by

$$G[k] = \int_{-1/2}^{1/2} \exp\left(-\frac{t^2}{\sigma^2}\right) \exp\left(-i2\pi kt\right) \, \mathrm{d}t \tag{37}$$

$$= \int_{1/2}^{1/2} \exp\left(-\frac{t^2}{\sigma^2}\right) \cos 2\pi kt \, \mathrm{d}t \qquad \text{because } g \text{ is even and } \sin(2\pi kt) \text{ is odd} \tag{38}$$

$$\approx \int_{-\infty}^{\infty} \exp\left(-\frac{t^2}{\sigma^2}\right) \cos 2\pi kt \, \mathrm{d}t \qquad \text{if } \sigma << 1 \tag{39}$$

$$= \sqrt{\pi}\sigma \exp\left(-\pi^2\sigma^2k^2\right). \tag{40}$$

The envelope of the Fourier coefficients is Gaussian with a standard deviation that is proportional to $1/\sigma$, as shown in Figure 5. △

An important property of the Fourier representation is that shifting a function is equivalent to scaling its Fourier coefficients by a factor that only depends on the shift and the corresponding frequency.

Lemma 1.12 (Time shift)**.** *We define the τ-shifted version of a function $f \in \mathcal{L}_2[-1/2, 1/2]$ by*

$$f_{[\tau]}(t) := f(t - \tau), \tag{41}$$

where the shift is circular *(the function wraps around). For any shift τ we have*

$$F_{[\tau]}[k] = \exp\left(-i2\pi k\tau\right) F[k]. \tag{42}$$

Proof. We interpret f as a periodic function such that $f(t+1) = f(t)$. We have

$$F_{[\tau]}[k] = \int_{-1/2}^{1/2} f(t - \tau) \exp\left(-i2\pi kt\right) \, \mathrm{d}t \tag{43}$$

$$= \int_{-1/2-\tau}^{1/2-\tau} f(u) \exp\left(-i2\pi k(u + \tau)\right) \, \mathrm{d}t \tag{44}$$

$$= \exp\left(-i2\pi k\tau\right) F[k]. \tag{45}$$

□

1.3 Sampling Theorem

In this section we consider the problem of estimating a bandlimited signal from samples taken at regular intervals. Such situations arise when we digitally store or process an analog signal such as music or speech. Figure 6 shows an example of a bandlimited signal, its spectrum and the corresponding samples. Two fundamental questions are:

1. What sampling rate is necessary to preserve all the information in the signal?
2. How can we reconstruct the original signal from the samples?

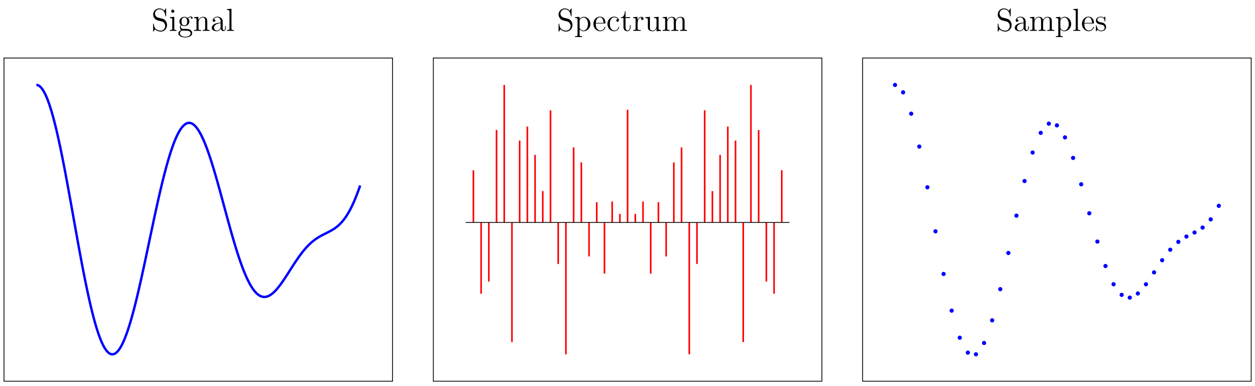


Figure 6: Bandlimited signal (left), corresponding spectrum (center) and regular samples (right).

The answer to the first question is given by the Nyquist-Shannon-Kotelnikov sampling theorem.

Theorem 1.13 (Nyquist-Shannon-Kotelnikov sampling theorem)**.** *Any bandlimited signal* $g \in \mathcal{L}_2\,[0,1]$ *of the form*

$$g\,(t) := \sum_{k=-k_c}^{k_c} G\,[k]\,h_k\,(t) \tag{46}$$

can be recovered exactly from n *uniformly spaced samples* $g\,(0)$*,* $g\,(1/n)$*, ...,* $g\,((n-1)/n)$ *as long as the sampling rate* $f_s := n$ *satisfies*

$$f_s \geq 2k_c + 1, \tag{47}$$

which is known as the Nyquist rate.

Proof. To simplify the proof, we assume that $n = 2k_s + 1$ for some integer $k_s \geq k_c$. We denote the vector of samples by $\vec{g}_n$, which equals

$$\vec{g}_n := \begin{bmatrix} g\,(0) \\ g\left(\frac{2}{n}\right) \\ \cdots \\ g\left(\frac{n-1}{n}\right) \end{bmatrix} = \begin{bmatrix} \sum_{k=-k_c}^{k_c} G\,[k]\,h_k\,(0) \\ \sum_{k=-k_c}^{k_c} G\,[k]\,h_k\left(\frac{1}{n}\right) \\ \cdots \\ \sum_{k=-k_c}^{k_c} G\,[k]\,h_k\left(\frac{n-1}{n}\right) \end{bmatrix} = \sum_{k=-k_c}^{k_c} G\,[k] \begin{bmatrix} h_k\,(0) \\ h_k\left(\frac{1}{n}\right) \\ \cdots \\ h_k\left(\frac{n-1}{n}\right) \end{bmatrix} = \sum_{k=-k_c}^{k_c} G\,[k]\,\vec{h}_k^{[n]}. \tag{48}$$

Now let us define a vector $\vec{G} \in \mathbb{C}^n$ containing the Fourier coefficients of g (padded by zeros if $k_s \geq k_c$),

$$\vec{G}\,[k] := \begin{cases} G\,[k]\,, & \text{if } |k| \leq k_c, \\ 0, & \text{otherwise.} \end{cases} \tag{49}$$

By equation (48) we have

$$\begin{aligned} \vec{g}_n &= \begin{bmatrix} \vec{h}_{-k_s}^{[n]} & \vec{h}_{-k_s+1}^{[n]} & \cdots & \vec{h}_{k_s}^{[n]} \end{bmatrix} \vec{G} && (50) \\ &= F\vec{G}. && (51) \end{aligned}$$

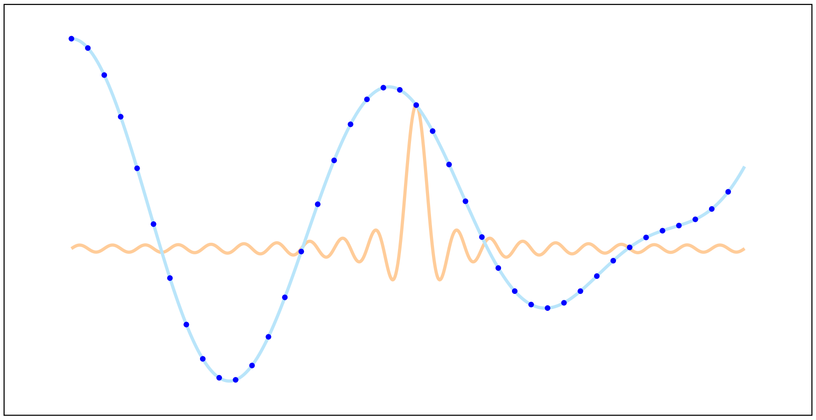


Figure 7: Reconstruction via linear inversion is equivalent to interpolation with a Dirichlet kernel.

By Lemma 1.6 the matrix F has orthogonal columns, so that we can recover $\vec{G}$ and hence g from $\vec{g}_n$. □

To recover the signal we can build its Fourier series from its Fourier series obtained by inverting the system of equations given by (51). Interestingly, this is exactly equivalent to interpolating the samples with shifted Dirichlet kernels, as sketched in Figure 7.

Theorem 1.14 (Dirichlet-kernel interpolation). *Let $n := 2k_s + 1$ for some integer k_s. As long as $k_s \geq k_c$, a bandlimited function of the form (46) is equal to*

$$g(t) = \frac{1}{n} \sum_{j=0}^{n-1} g(j/n)\, d_{[j/n]}(t) \tag{52}$$

where d is a Dirichlet kernel (see Definition 1.10) with cut-off frequency k_s.

Proof. Let us define the vector

$$\vec{a}_t := \begin{bmatrix} \exp(-i2\pi k_s t) & \exp(-i2\pi(k_s-1)t) & \cdots & \exp(i2\pi k_s t) \end{bmatrix}^T, \tag{53}$$

such that the adjoint of the matrix F in (51) can be expressed as

$$F^* = \begin{bmatrix} \vec{a}_0 & \vec{a}_{1/n} & \cdots & \vec{a}_{(n-1)/n} \end{bmatrix}. \tag{54}$$

F scaled by $1/\sqrt{n}$ is orthogonal, so

$$\vec{G} = F^* \vec{g}_n \tag{55}$$

$$= \frac{1}{n} \sum_{j=0}^{n-1} g(j/n)\, \vec{a}_{j/n}. \tag{56}$$

Now, we express g and the Dirichlet kernel in terms of $\vec{a}_t$,

$$g(t) = \frac{1}{n} \sum_{k=-k_c}^{k_c} G[k] \, e^{-i2\pi kt} \tag{57}$$

$$= \frac{1}{n} \vec{a}_t^* \vec{G}, \tag{58}$$

$$d_{[\tau]}(t) = \sum_{k=-k_c}^{k_c} e^{-i2\pi k(t-\tau)} \tag{59}$$

$$= \vec{a}_t^* \vec{a}_\tau . \tag{60}$$

We conclude

$$g(t) = \frac{1}{n} \vec{a}_t^* \vec{G} \qquad \text{by (58)} \tag{61}$$

$$= \frac{1}{n} \sum_{j=0}^{n} g(j/n) \, \vec{a}_t^* \vec{a}_{j/n} \qquad \text{by (56)} \tag{62}$$

$$= \frac{1}{n} \sum_{j=0}^{n} g(j/n) \, d_{[j/n]}(t) \qquad \text{by (60).} \tag{63}$$

□

Finally, another interesting question is what occurs when the cut-off frequency of the function we are sampling k_c is too large with respect to the sampling rate $k_c > k_s$ so that we violate the sampling-rate condition (47). The following example illustrates this.

Example 1.15 (Aliasing)**.** We consider a function

$$g(t) := \sum_{k=-k_c}^{k_c+1} G[k] \, h_k(t) . \tag{64}$$

with cut-off frequency $k_c + 1$ defined on the unit interval. We have set $G[-k_c - 1]$ to simplify notation. Imagine that we underestimate its cut-off frequency and sample it at a rate $k_s := k_c/2$ instead of $k_c + 1$, gathering $n := 2k_s + 1 = k_c + 1$ samples. Let us define $\vec{G}$ as the vector containing the Fourier coefficients of g. By (48) we have

$$\vec{g}_n = \begin{bmatrix} \vec{h}^{[n]}_{-k_c} & \cdots & \vec{h}^{[n]}_{0} & \vec{h}^{[n]}_{1} & \cdots & \vec{h}^{[n]}_{k_c+1} \end{bmatrix} \vec{G} \tag{65}$$

$$= \begin{bmatrix} \vec{h}^{[n]}_{-2k_s} & \cdots & \vec{h}^{[n]}_{0} & \vec{h}^{[n]}_{1} & \cdots & \vec{h}^{[n]}_{2k_s+1} \end{bmatrix} \vec{G} \tag{66}$$

$$= \begin{bmatrix} \vec{h}^{[n]}_{1} & \cdots & \vec{h}^{[n]}_{n} & \vec{h}^{[n]}_{1} & \cdots & \vec{h}^{[n]}_{n} \end{bmatrix} \vec{G} \tag{67}$$

$$= \begin{bmatrix} \widetilde{F} & \widetilde{F} \end{bmatrix} \begin{bmatrix} \vec{G}_1 \\ \vec{G}_2 \end{bmatrix}, \tag{68}$$

since for any k $\vec{h}^{[n]}_k = \vec{h}^{[n]}_{k+n}$ by periodicity of the discrete complex sinusoids. Note that we have separated $\vec{G}_1$ into its first and second half. Now we have a problem because $\begin{bmatrix} \widetilde{F} & \widetilde{F} \end{bmatrix}$ is clearly not invertible!

Compare this to the situation where $n = 2k_c + 1$. As we showed in the proof of Theorem 1.13 (with a different choice of indices that doesn't affect the result) in that case

$$\vec{G} = \frac{1}{n} \widetilde{F}^* \vec{g}_n, \tag{69}$$

since $\widetilde{F}$ has orthogonal columns by Lemma 1.6.

What happens if we assume that this holds even if it doesn't and we try to obtain $\vec{G}$ from $\vec{g}_n$ in the same way? Then *aliasing* occurs:

$$\begin{align} \vec{G}_{\text{aliased}} &= \frac{1}{n} \widetilde{F}^* \vec{g}_n \tag{70} \\ &= \frac{1}{n} \widetilde{F}^* \begin{bmatrix} \widetilde{F} & \widetilde{F} \end{bmatrix} \begin{bmatrix} \vec{G}_1 \\ \vec{G}_2 \end{bmatrix} \tag{71} \\ &= \vec{G}_1 + \vec{G}_2. \tag{72} \end{align}$$

The recovered spectrum is scrambled. We recover a function with cut-off frequency k_c with a spectrum that equals the sum of the spectrum of g and a copy that is shifted by k_c. △

1.4 Discrete Fourier transform

By Lemma 1.6 the discrete complex sinusoids $\frac{1}{\sqrt{n}} \vec{h}_0^{[n]}, \ldots, \frac{1}{\sqrt{n}} \vec{h}_{n-1}^{[n]}$ form an orthonormal basis of $\mathbb{C}^n$. This means that we can express any vector in terms of its coefficients in this representation. The discrete Fourier transform (DFT), which extracts these coefficients, is consequently the discrete counterpart of the Fourier series.

Definition 1.16 (Discrete Fourier transform)**.** *The discrete Fourier transform (DFT) of a vector $\vec{x} \in \mathbb{C}^n$ are given by*

$$\vec{X}[k] := \left\langle \vec{x}, \vec{h}_k^{[n]} \right\rangle, \qquad 0 \leq k \leq n-1. \tag{73}$$

Equivalently,

$$\vec{X} = W \vec{x} \tag{74}$$

where $W^* := \begin{bmatrix} \vec{h}_0^{[n]} & \vec{h}_1^{[n]} & \cdots & \vec{h}_{n-1}^{[n]} \end{bmatrix}^*$. W is known as the DFT matrix.

Recovering a vector from its DFT coefficients only requires inverting the linear transformation.

Lemma 1.17. *If $\vec{X}$ contains the DFT coefficients of $\vec{x} \in \mathbb{C}^n$, then*

$$\begin{align} \vec{x} &= \frac{1}{n} \sum_{k=0}^{n-1} \vec{X}[k] \, \vec{h}_k^{[n]} \tag{75} \\ &= \frac{1}{n} F^* \vec{X}. \tag{76} \end{align}$$

Proof. By Lemma 1.6 $\frac{1}{\sqrt{n}}F$ is an orthogonal matrix, so

$$F^* \vec{X} = F^* F \vec{x} \tag{77}$$
$$= \vec{x}. \tag{78}$$

□

As in the case of the Fourier series, the DFT coefficients of a shifted vector can be computed by scaling linearly the DFT coefficients of the original vector.

Lemma 1.18 (Discrete time shift)**.** *We define the m-shifted version of a vector $\vec{x} \in \mathbb{C}^n$ by*

$$\vec{x}_{[m]}[j] := \vec{x}(j-m), \tag{79}$$

where the shift is circular, *so that $\vec{x}(j+n) = \vec{x}(j)$. For any shift m we have*

$$\vec{X}_{[m]}[k] = \exp(-i2\pi km)\, \vec{X}[k]. \tag{80}$$

Proof.

$$\vec{X}_{[m]}[k] = \sum_{j=0}^{n-1} \vec{x}_{[\tau]}[j] \exp(-i2\pi kj) \tag{81}$$
$$= \sum_{l=-m}^{n-1-m} \vec{x}[l] \exp(-i2\pi k(l+m)) \tag{82}$$
$$= \exp(-i2\pi km)\, \vec{X}[k]. \tag{83}$$

□

Remark 1.19 (Connection between the Fourier series and the DFT)**.** *If $\vec{x}$ corresponds to samples from a bandlimited function, as in Theorem 1.13, which are measured at the Nyquist rate, then the DFT coefficients of $\vec{x}$ are equal to the Fourier series coefficients of the continuous function. This follows from (51), since if we re-index the columns adequately (recall that $\vec{h}^{[n]}_{k+n} = \vec{h}^{[n]}_k$ for any k) we have $W = F^*$. The DFT, combined with the sampling period, consequently makes it possible to compute the spectrum of continuous signals in practice!*

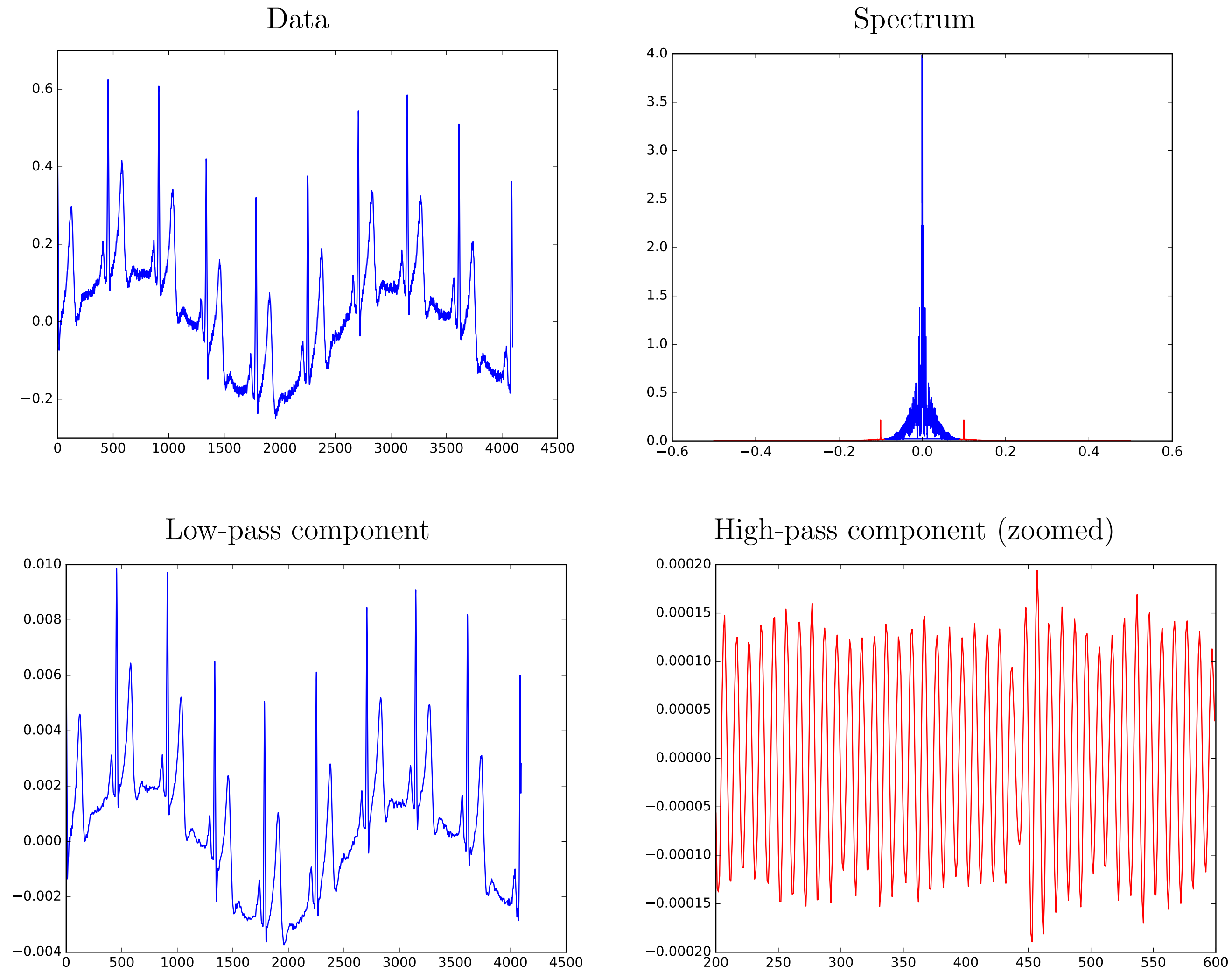
Figure 8 shows the DFT of an electrocardiogram signal. Its frequency representation makes it possible to locate and remove interference caused by the power-line system.

The DFT of a signal of length n can be obtained with complexity $\mathcal{O}(n \log n)$ by applying the fast-Fourier transform (FFT) algorithm described below. It is difficult to overstate the importance of this method, which makes it extremely efficient to compute the DFT. The algorithm relies on the following lemma, which you will prove in the homework.

Lemma 1.20. *Let $W^{[n]}$ denote the $n \times n$ DFT matrix and assume that n is even. Then for $k < n/2$*

$$W^{[n]}\vec{x}[k] = W^{[n/2]}\vec{x}_{\text{even}}[k] + \exp\left(-\frac{i2\pi k}{n}\right) W^{[n/2]}\vec{x}_{\text{odd}}[k], \tag{84}$$

$$W^{[n]}\vec{x}[k+n/2] = W^{[n/2]}\vec{x}_{\text{even}}[k] - \exp\left(-\frac{i2\pi k}{n}\right) W^{[n/2]}\vec{x}_{\text{odd}}[k]. \tag{85}$$



Data

Spectrum

Low-pass component

High-pass component (zoomed)

Figure 8: Electrocardiogram data (top left), along with its spectrum (top right). We separate the spectrum into a low-pass (blue) and a high-pass (red) region. The low-pass region corresponds to the signal of interest (bottom left), whereas the high-pass component corresponds to power-line interference (bottom right).

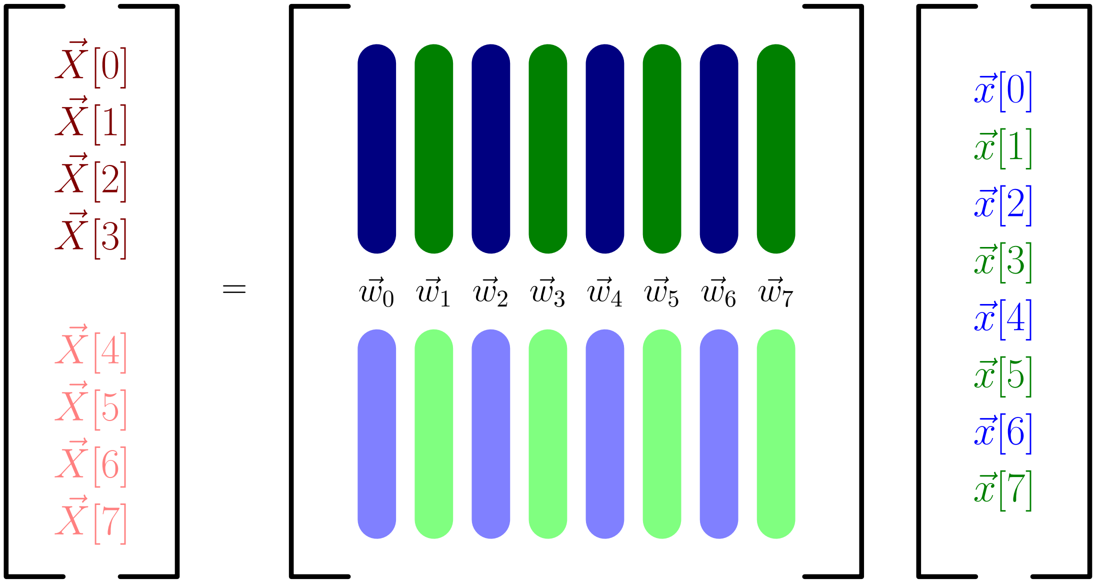


Figure 9: $W^{(8)}$ can be decomposed into even columns (blue) and odd columns (green). As we will see, it is also useful to distinguish the top half (dark) from the bottom half (light).

Algorithm 1.21 (Fast Fourier transform)**.** *If $n = 1$, then set $W^1\vec{x} := \vec{x}$. Otherwise apply the following steps:*

1. *Compute $W^{[n/2]}\vec{x}_{\text{even}}$.*
2. *Compute $W^{[n/2]}\vec{x}_{\text{odd}}$.*
3. *For $k = 1, 2, \ldots, n/2$ set*

$$W^{[n]}\vec{x}\,[k] := W^{[n/2]}\vec{x}_{\text{even}}\,[k] + \exp\left(-\frac{i2\pi k}{n}\right) W^{[n/2]}\vec{x}_{\text{odd}}\,[k]\,, \tag{86}$$

$$W^{[n]}\vec{x}_{k+n/2} := W^{[n/2]}\vec{x}_{\text{even}}\,[k] - \exp\left(-\frac{i2\pi k}{n}\right) W^{[n/2]}\vec{x}_{\text{odd}}\,[k]\,. \tag{87}$$

In Figures 9 ,10, 11, 12 we visually depict the FFT algorithm for $n = 8$. Figure 13 shows the recursion tree, which illustrates why the runtime is $O(n \lg n)$.

The DFT can be extended to two dimensions by considering two-dimensional sinusoidal atoms obtained by taking the outer product of the one-dimensional discrete complex sinusoids of different frequencies.

Definition 1.22 (Two-dimensional DFT)**.** *The 2D DFT $\widehat{M}$ of a matrix $M \in \mathbb{C}^{n\times n}$ is given by*

$$\widehat{M}\,[k] := \left\langle M, \vec{h}^{\,\text{2D}}_{k_1,k_2} \right\rangle, \qquad 0 \le k_1, k_2 \le n-1, \tag{88}$$

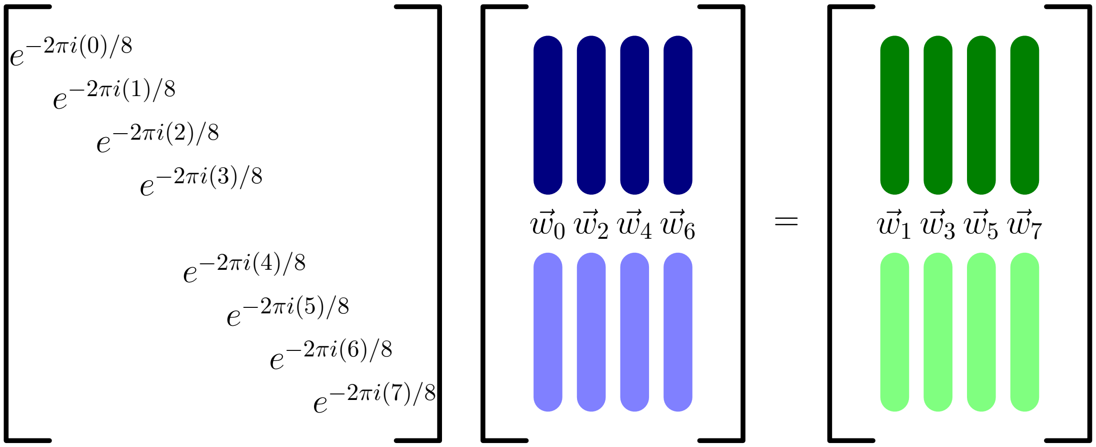


Figure 10: The odd columns of $W^{(8)}$ can be calculated from the evens by scaling the rows.

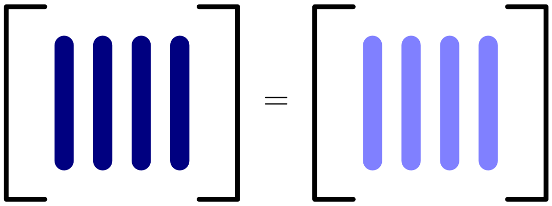


Figure 11: The bottom and top half of the even columns are the same. Both are a DFT matrix of size $4 = n/2$.

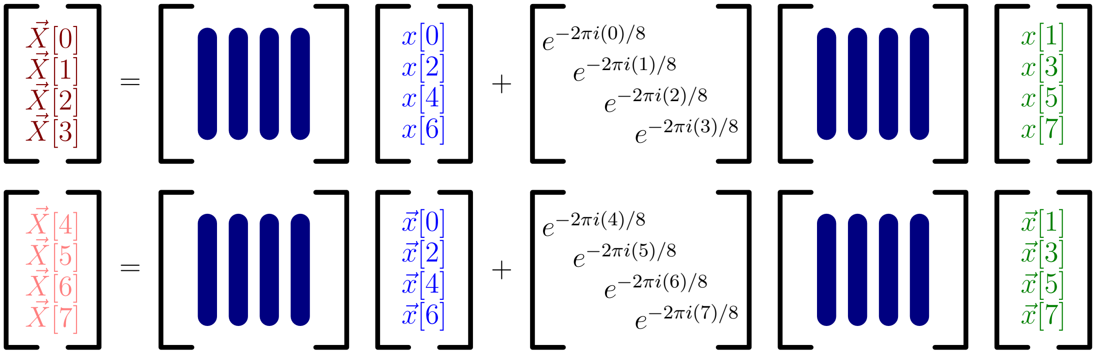


Figure 12: The $n = 8$ FFT calculation represented using only the dark blue portion of the matrix. The reduces the $n = 8$ FFT to two calls ot the $n = 4$ FFT and then $O(n)$ work to combine the results.

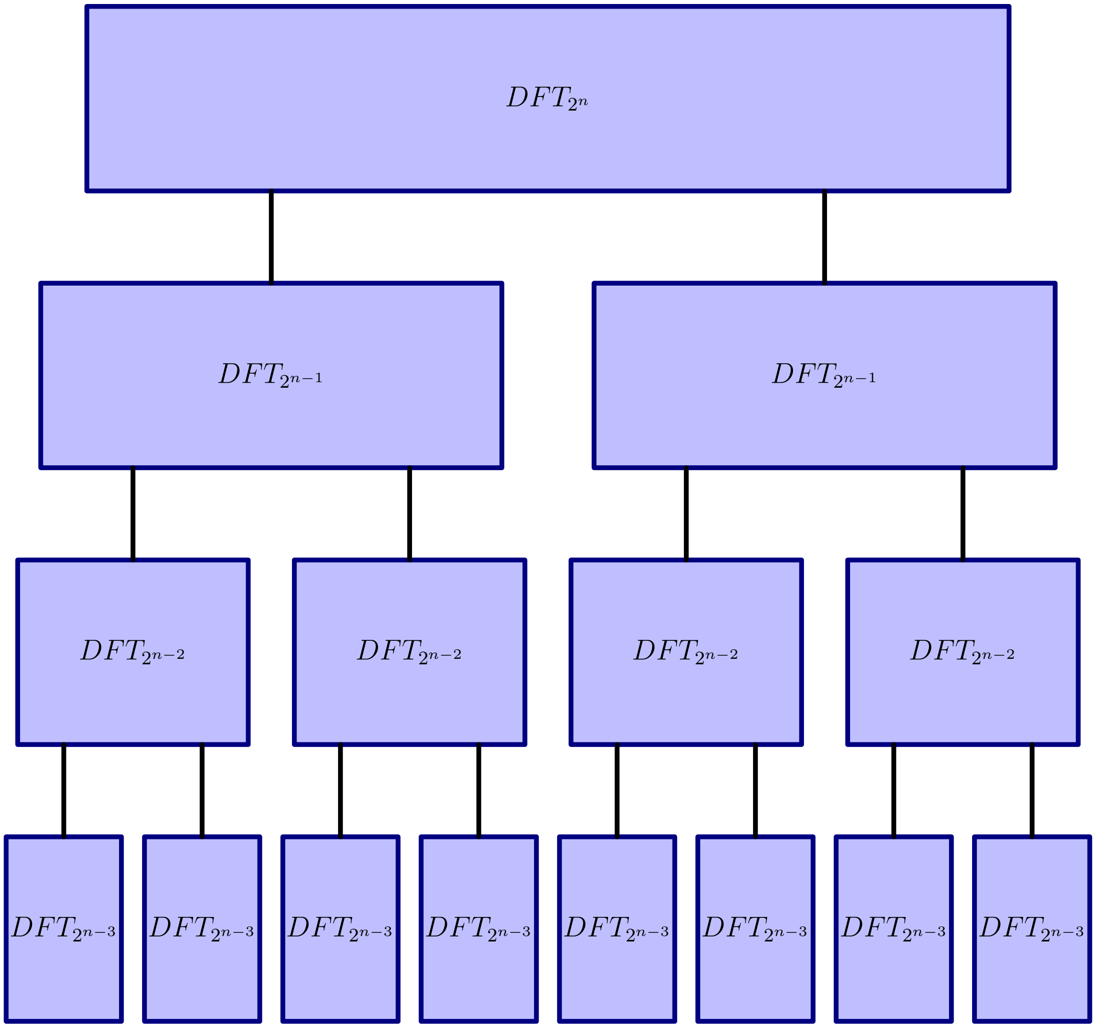


Figure 13: The recursion tree for the FFT.

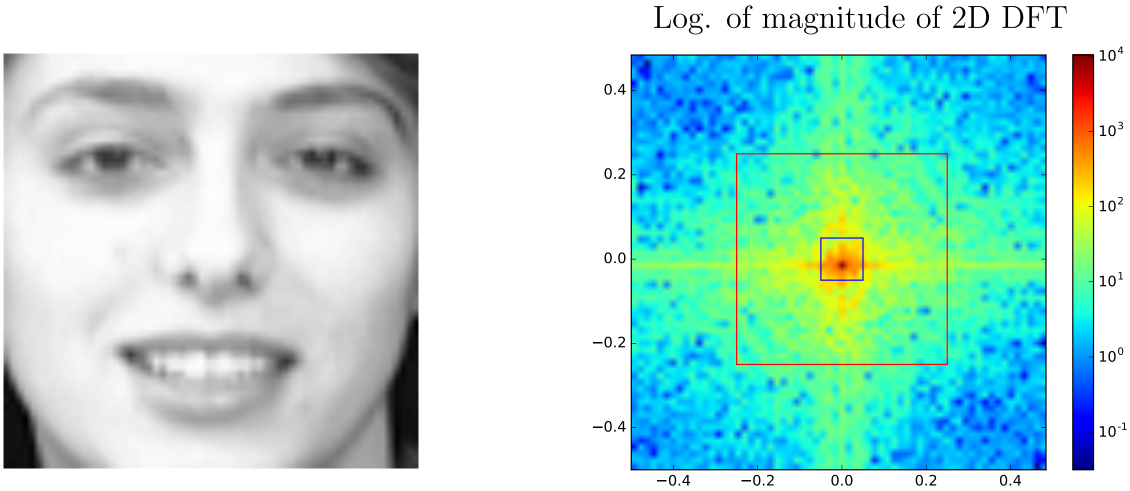


Figure 14: The logarithm of the magnitudes of the 2D DFT coefficients of the image on the left are shown on the right.

where

$$\vec{h}^{2\text{D}}_{k_1,k_2} := \vec{h}^{[n]}_{k_1} \left(\vec{h}^{[n]}_{k_2}\right)^T \tag{89}$$

$$= \begin{bmatrix} 1 & e^{\frac{i2\pi k_2}{n}} & \cdots & e^{\frac{i2\pi k_2 (n-1)}{n}} \\ e^{\frac{i2\pi k_1}{n}} & e^{\frac{i2\pi (k_1+k_2)}{n}} & \cdots & e^{\frac{i2\pi (k_1+k_2(n-1))}{n}} \\ & & \cdots & \\ e^{\frac{i2\pi k_1 (n-1)}{n}} & e^{\frac{i2\pi (k_1(n-1)+k_2)}{n}} & \cdots & e^{\frac{i2\pi (k_1(n-1)+k_2(n-1))}{n}} \end{bmatrix}. \tag{90}$$

Equivalently,

$$\widehat{M} = W M W, \tag{91}$$

where $W := \begin{bmatrix} \vec{h}^{[n]}_0 & \vec{h}^{[n]}_1 & \cdots & \vec{h}^{[n]}_{n-1} \end{bmatrix}^*$ *is the 1D DFT matrix.*

Figure 14 shows an image along with the magnitudes of its 2D DFT coefficients. As is often the case for natural images, most of the energy is concentrated in the lower end of the spectrum. Figure 15 shows the result of projecting different components of the spectrum of the image (see the right image of Figure 14) onto the image domain. The low-pass component captures low-resolution variations, the band-pass component higher-resolution details and the high-pass component high-frequency fluctuations.

1.5 Compression

The discrete cosine transform (DCT) is a variant of the discrete Fourier transform for real discrete signals. The DCT is obtained from the DFT by assuming that the signal is symmetric (but only one half is observed). In that case the DFT components correspond to discrete cosines (the

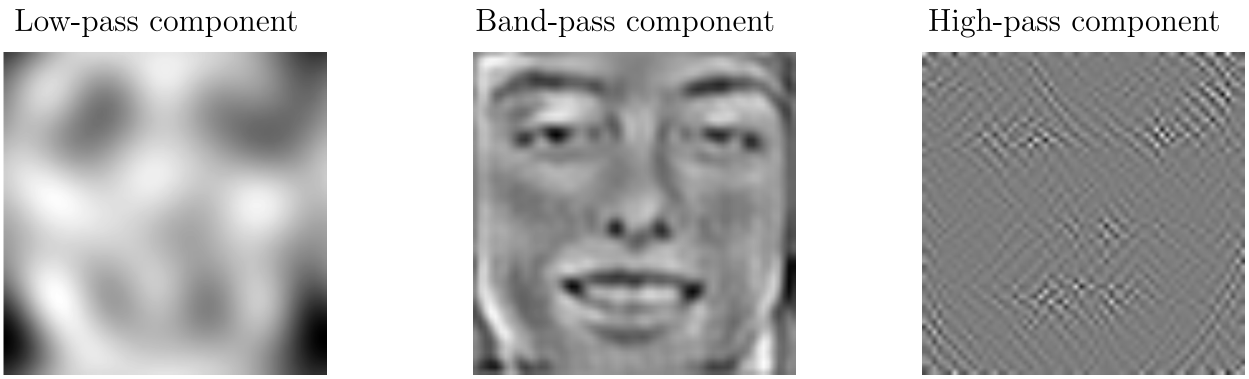


Figure 15: Image corresponding to the low-pass (left), band-pass (center) and high-pass (right) components of the image in Figure 14. The corresponding regions of the spectrum are shown in Figure 14.

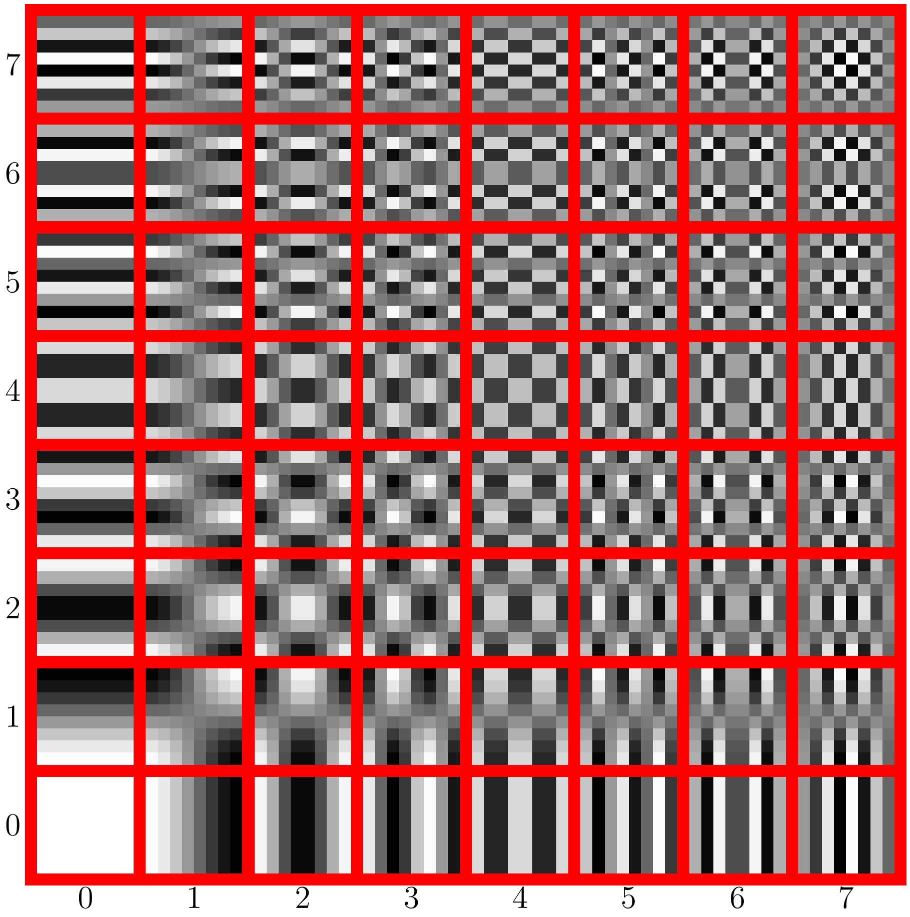


Figure 16: 8×8 DCT basis vectors.

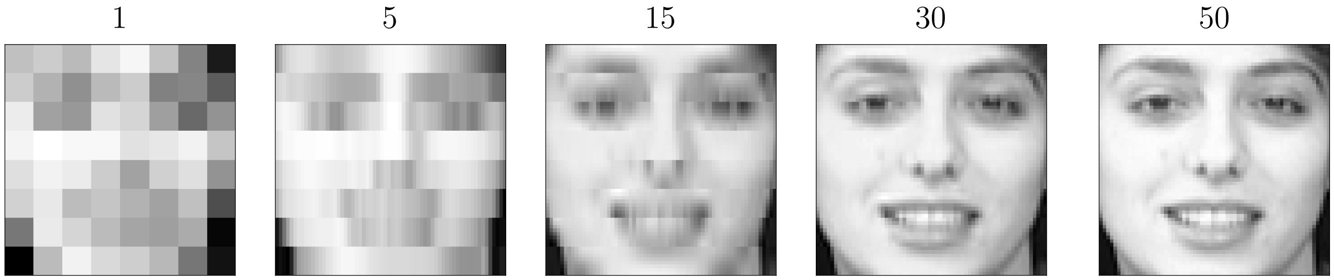


Figure 17: Result of projecting each 64-pixel patch from the natural image in Figure 14 onto the lowest 1, 5, 15, 30 and 50 2D DCT basis functions.

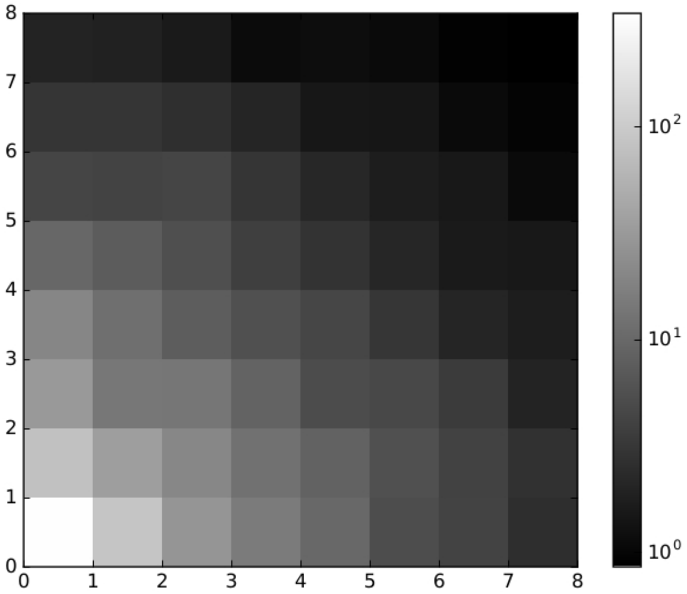


Figure 18: Average magnitudes of each 2D DCT coefficient in a database of patches extracted from natural images.

coefficients corresponding to the sines are zero due to symmetry). The two-dimensional DCT is a very important tool in image processing. Figure 16 shows the basis vectors of the 8×8 2D DCT.

One of the main reasons that the 2D DCT is so useful is that the energy of natural images are highly concentrated in their low-frequency components. Figure 18 shows the average magnitudes of each 2D DCT coefficient in a database of patches extracted from natural images. Figure 17 shows the result of dividing the image into 64-pixel patches and projecting it onto the span of the low-frequency DCT basis functions. Ignoring some of the high-frequency components is almost imperceptible. This is the main insight behind the JPEG method for lossy compression of digital images. JPEG divides the image in 8×8 patches and then quantizes each band differently, using more bits for lower-frequency bands where differences are more apparent.

Algorithm 1.23 (JPEG compression (for grayscale images))**.** *1. Choose a quality setting* $Q \in (0, 100)$.

2. Divide image into a collection of 8×8 *pixel patches.*

$$M = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

Figure 19: JPEG DCT Quantization Matrix

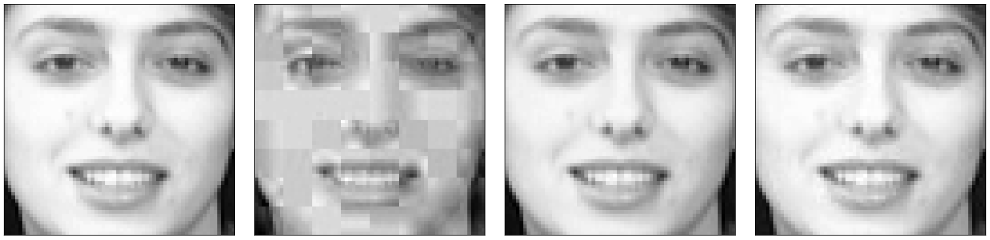


Figure 20: Natural image compared to the result of quantizing the lowest 16 DCT coefficients of each patch, the highest 16 DCT coefficients of each patch and applying JPEG Compression with $Q = 90$ (which yields a compression factor of between 4 and 5).

3. *Compute the 2D DCT of each patch.*
4. *Quantize the 2D DCT of each patch separately. Let $\widehat{P} \in \mathbb{R}^{8\times 8}$ denote the 2D DCT of a patch and let M denote the JPEG quantization matrix shown in Figure ??. Set*

$$\widehat{P}'_{ij} = \text{round}\left(\frac{\widehat{P}_{ij}}{S(Q)M_{ij}}\right) S(Q)M_{ij}, \tag{92}$$

where $S(Q)$ is the quality scaling factor:

$$S(Q) := \begin{cases} \frac{100-Q}{50} & \textit{if } Q > 50, \\ \frac{50}{Q} & \textit{otherwise.} \end{cases} \tag{93}$$

5. *Encode into a file. When viewing, the decoder will compute the inverse 2D DCT of each quantized patch $\widehat{P}'$ to display the image.*

2 Convolution

2.1 Continuous convolution

Convolution between functions is a fundamental operation in signal and image processing.

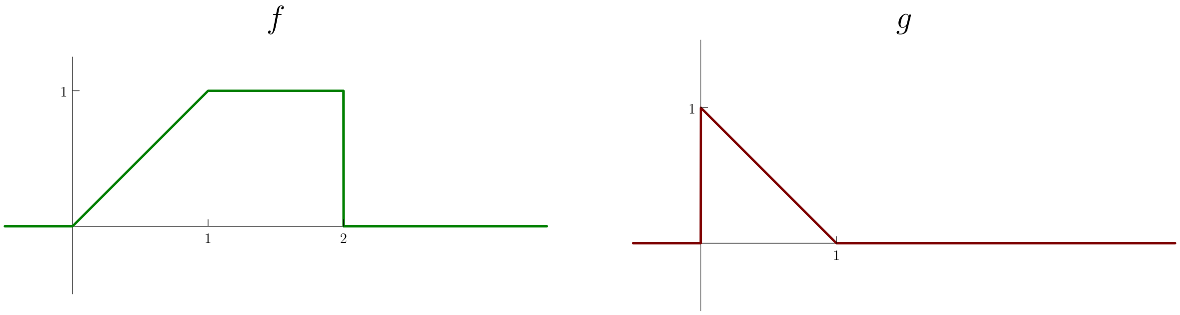


Figure 21: Two functions f and g.

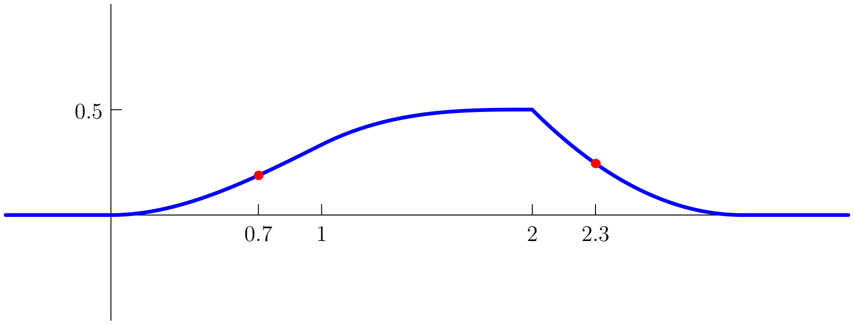


Figure 22: The result of convolving functions f and g from Figure 21.

Definition 2.1 (Convolution)**.** *The convolution of two functions* $f, g \in \mathcal{L}_2\left[-1/2, 1/2\right]$ *is defined as*

$$f * g\left(t\right) := \int_{-1/2}^{1/2} f\left(u\right) g\left(t-u\right)\, du. \tag{94}$$

Figures 21, 22 and 23 illustrate the operation with a simple example. To compute the value of the convolution between two functions at point t, we (1) fix one of them ($f\left(u\right)$), (2) flip and shift the other by t ($g\left(t-u\right)$) and (3) integrate their product.

Convolution in time (or space) is equivalent to multiplication in frequency. This implies that we can compute convolutions very efficiently using the FFT.

Theorem 2.2 (Convolution in time is multiplication in frequency)**.** *Let* $r := f * g$ *for* $f, g \in \mathcal{L}_2\left[-1/2, 1/2\right]$*. Then*

$$R\left[k\right] = F\left[k\right] G\left[k\right]. \tag{95}$$

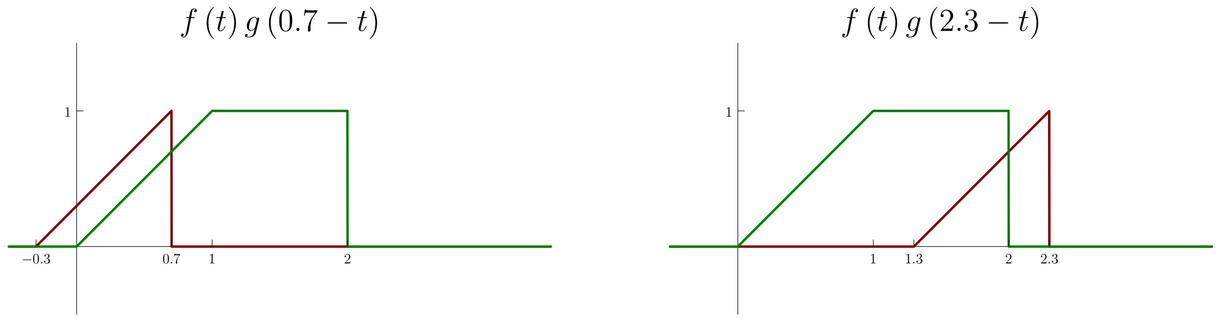


Figure 23: Position of the two functions in Figure 21 when they are multiplied to compute the value of their convolution at 0.7 (left) and 2.3 (right).

Proof.

$$
\begin{aligned}
R[k] &:= \int_{-1/2}^{1/2} \exp(-i2\pi kt)\, f * g(t)\, \mathrm{d}t && (96)\\
&= \int_{-1/2}^{1/2} f(u) \int_{-1/2}^{1/2} \exp(-i2\pi kt)\, g(t-u)\, \mathrm{d}t\, \mathrm{d}u && (97)\\
&= \int_{-1/2}^{1/2} f(u)\, G[k] \exp(-i2\pi ku)\, \mathrm{d}t\, \mathrm{d}u \qquad \text{by Lemma 1.12} && (98)\\
&= F[k]\, G[k]. && (99)
\end{aligned}
$$

□

The convolution theorem shows that we can compute convolutions between continuous functions by just multiplying their Fourier coefficients. It can also be used to prove the central limit theorem.

Example 2.3 (Sketch of a proof of the central limit theorem)**.** The convolution theorem provides insight into why the distribution of sums of independent random variables become Gaussian in the limit.

Theorem 2.4 (Pdf of the sum of two independent random variables)**.** *The pdf of* $\mathbf{z} = \mathbf{x} + \mathbf{y}$*, where* $\mathbf{x}$ *and* $\mathbf{y}$ *are independent random variables is equal to the* ***convolution*** *of their respective pdfs* $f_{\mathbf{x}}$ *and* $f_{\mathbf{y}}$*,*

$$f_{\mathbf{z}}(z) = \int_{u=-\infty}^{\infty} f_{\mathbf{x}}(z-u)\, f_{\mathbf{y}}(u)\, du. \tag{100}$$

Proof. Note that

$$
\begin{aligned}
F_{\mathbf{z}}(z) &= \mathrm{P}(\mathbf{x} + \mathbf{y} \le z) && (101)\\
&= \int_{y=-\infty}^{\infty} \int_{x=-\infty}^{z-y} f_{\mathbf{x},\mathbf{y}}(x,y)\, \mathrm{d}x\, \mathrm{d}y && (102)\\
&= \int_{y=-\infty}^{\infty} \int_{u=-\infty}^{z} f_{\mathbf{x},\mathbf{y}}(u-y,y)\, \mathrm{d}u\, \mathrm{d}y \qquad (u = x+y) && (103)\\
&= \int_{u=-\infty}^{z} \int_{y=-\infty}^{\infty} f_{\mathbf{x},\mathbf{y}}(u-y,y)\, \mathrm{d}y\, \mathrm{d}u, && (104)
\end{aligned}
$$

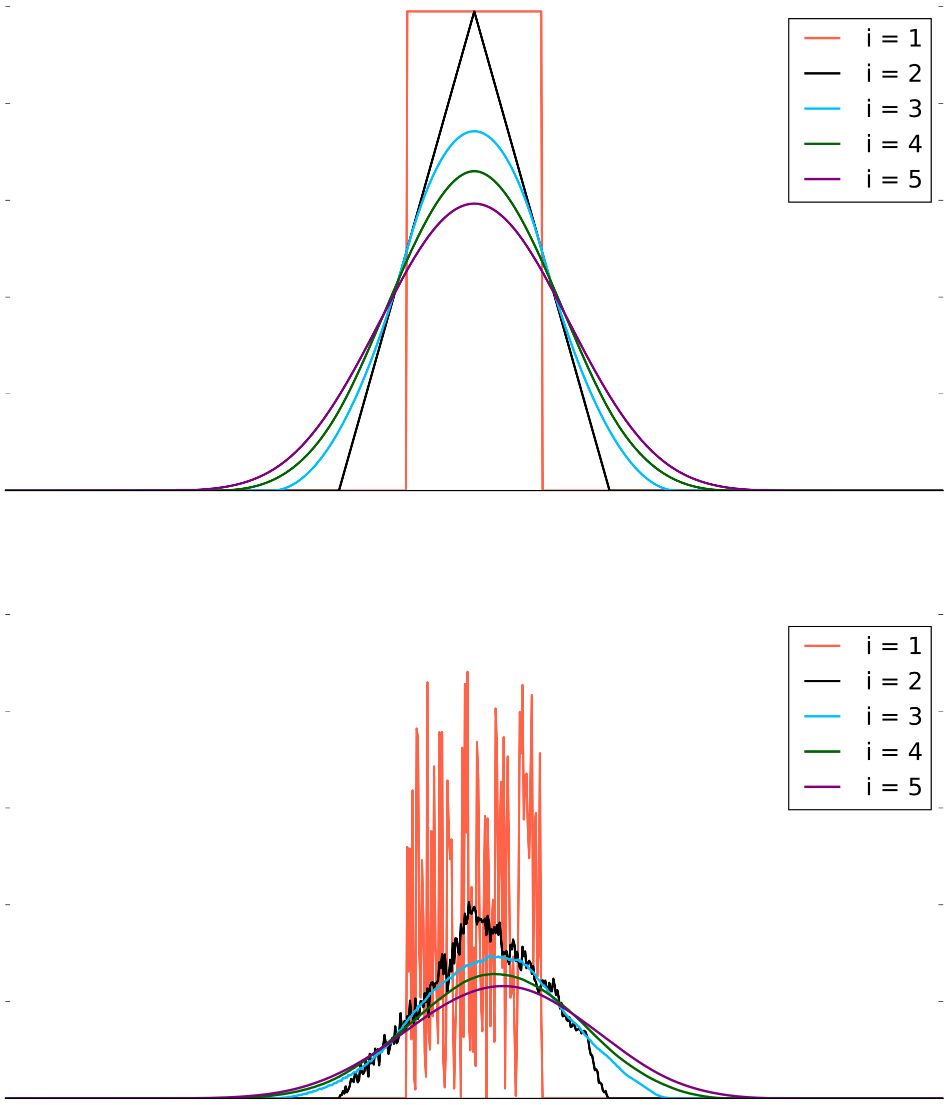


Figure 24: Result of convolving two different distributions with themselves several times. The shapes quickly become Gaussian-like.

where swapping the integrals is justified since the pdfs are nonnegative. Applying independence shows

$$f_{\mathbf{z}}(z) = \int_{y=-\infty}^{\infty} f_{\mathbf{x},\mathbf{y}}(z-y,y)\,\mathrm{d}y = \int_{y=-\infty}^{\infty} f_{\mathbf{x}}(z-y) f_{\mathbf{y}}(y)\,\mathrm{d}y. \tag{105}$$

△

Now let us consider a sequence of iid random variables $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$, ... with pdf f. The pdf of their sum is given by

$$f_{\sum_{j=1}^{\infty} \mathbf{x}_j}(x) = (f * f * \cdots)(x)\,. \tag{106}$$

Convolutions have a smoothing effect, which eventually transforms the pmf/pdf into a Gaussian! We show this numerically in Figure 24 for two very different distributions: a uniform distribution and a very irregular one. Both converge to Gaussian-like shapes after just 3 or 4 convolutions. The central limit theorem makes this precise, establishing that the shape of the pmf or pdf does indeed become Gaussian asymptotically. △

2.2 Discrete convolution

Definition 2.5 (Discrete convolution)**.** *The circular convolution of two vectors $\vec{x}, \vec{y} \in \mathbb{C}^n$ is defined as*

$$\vec{x} * \vec{y}\,[j] := \sum_{m=0}^{n-1} \vec{x}\,[m]\,\vec{y}\,[j-m]\,, \qquad 0 \leq j \leq n-1, \tag{107}$$

$$\tag{108}$$

where the shifts are circular*, so that $\vec{x}\,[j] = \vec{x}\,[j+n]$ and $\vec{y}\,[j] = \vec{y}\,[j+n]$.*

Circular convolution can be expressed as multiplication with a convolution matrix.

Definition 2.6 (Convolution matrix)**.** *The convolution matrix corresponding to a vector $\vec{x} \in \mathbb{C}^n$ contains every possible shift of the entries of $\vec{y}$ in its rows*

$$C_{\vec{y}} := \begin{bmatrix} \vec{y}[0] & \vec{y}[n-1] & \cdots & \vec{y}[2] & \vec{y}[1] \\ \vec{y}[1] & \vec{y}[0] & \cdots & \vec{y}[3] & \vec{y}[2] \\ & & \cdots & & \\ \vec{y}[n-1] & \vec{y}[n-2] & \cdots & \vec{y}[1] & \vec{y}[0] \end{bmatrix}. \tag{109}$$

*Matrices with this structure are called **circulant** matrices. Assuming that the vectors entries are numbered from 0 to $n-1$, the convolution between $\vec{y}$ and any other vector $\vec{x} \in \mathbb{C}^n$ can be expressed as*

$$\vec{x} * \vec{y} = C_{\vec{y}}\vec{x}. \tag{110}$$

Theorem 2.7 (Convolution in time is multiplication in frequency)**.** *Let $\vec{r} := \vec{x} * \vec{y}$ for $\vec{x}, \vec{y} \in \mathbb{C}^n$. Then*

$$\vec{R}\,[k] = \vec{X}\,[k]\,\vec{Y}\,[k]\,. \tag{111}$$

Proof.

$$
\begin{aligned}
R\left[k\right] &:= \sum_{j=0}^{n-1} \exp\left(-i2\pi kj\right) \sum_{m=0}^{n-1} \vec{x}\left[m\right] \vec{y}\left[j-m\right] && (112)\\
&= \sum_{m=0}^{n-1} \vec{x}\left[m\right] \sum_{j=0}^{n-1} \exp\left(-i2\pi kj\right) \vec{y}\left[j-m\right] && (113)\\
&= \sum_{m=0}^{n-1} \vec{x}\left[m\right] \exp\left(-i2\pi km\right) \vec{Y}\left[k\right] \qquad \text{by Lemma 1.18} && (114)\\
&= \vec{X}\left[k\right] \vec{Y}\left[k\right]. && (115)
\end{aligned}
$$

□

Let $\Lambda_{\vec{Y}}$ contain the DFT of an arbitrary vector $\vec{y} \in \mathbb{C}^n$. We can express the convolution of $\vec{y}$ with any vector $\vec{x} \in \mathbb{C}^n$ as

$$
\begin{aligned}
C_{\vec{y}} &= \vec{x} * \vec{y} && (116)\\
&= \frac{1}{n} W^* \Lambda_{\vec{Y}} \vec{X} && (117)\\
&= \frac{1}{n} W^* \Lambda_{\vec{Y}} W \vec{x}. && (118)
\end{aligned}
$$

This immediately implies that the discrete sinusoids in the columns of the DFT matrix are eigenvectors of the convolution matrix.

Corollary 2.8 (Eigendecomposition of circulant matrices)**.** *A circulant matrix $C_{\vec{y}}$ corresponding to a vector $\vec{y}$ has an eigendecomposition of the form*

$$C_{\vec{y}} = \frac{1}{n} W^* \Lambda_{\vec{Y}} W. \tag{119}$$

Convolution can be extended to two dimensions by using 2D shifts.

Definition 2.9 (Two-dimensional discrete convolution)**.** *The circular convolution of two matrices $M_1, M_2 \in \mathbb{C}^{n \times n}$ is defined as*

$$M_1 * M_2\left[j,l\right] := \sum_{m=0}^{n-1} \sum_{u=0}^{n-1} M_1\left[m,u\right] M_2\left[j-m, l-u\right] \tag{120}$$

where the shifts are circular, so that $M_1\left[j,l\right] = M_1\left[j+n, l+n\right]$.

In two dimensions, convolution is again equivalent to multiplication in the 2D DFT domain. We omit the proof, which is similar to the one for the 1D case.

Theorem 2.10 (Convolution in space is multiplication in frequency)**.** *Let $R := M_1 * M_2$ for $M_1, M_2 \in \mathbb{C}^{n \times n}$. Then*

$$\widehat{R}\left[k_1,k_2\right] = \widehat{M_1}\left[k_1,k_2\right] \widehat{M_2}\left[k_1,k_2\right]. \tag{121}$$

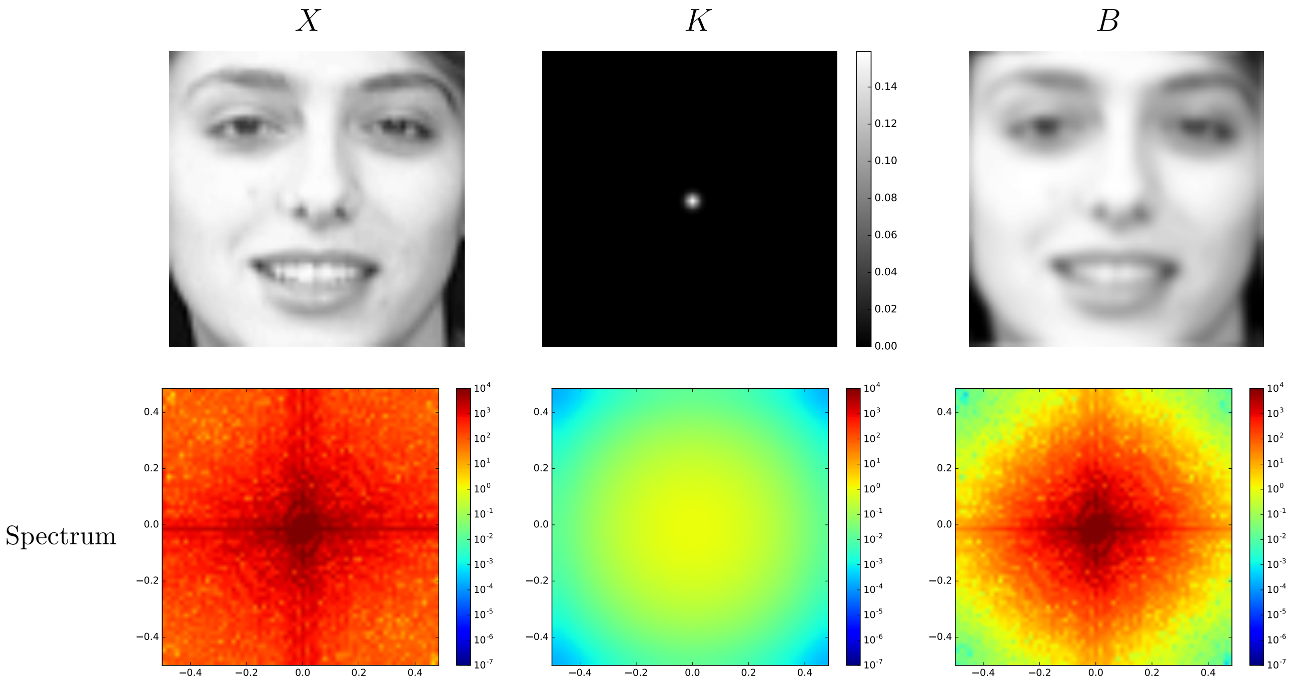


Figure 25: The top row shows a natural image (left), a blurring kernel (center) and the corresponding blurred image (right). The bottom row shows the magnitudes of the 2D DFT coefficients of the three images on a logarithmic scale.

2.3 Wiener deconvolution

In imaging, the resolution of lenses is limited by diffraction. If the resolution is low with respect to the number of pixels in the image, this is perceived as blur. A simplified model for a blurred image B is the convolution of the high-resolution images, which we represent by a matrix $X \in \mathbb{R}^{n\times n}$, and a convolution kernel $K \in \mathbb{R}^{n\times n}$ that depends on the optical system

$$B = K * X. \tag{122}$$

In the frequency domain, the 2D DFT of the image equals the product between the DFTs of the high resolution image and the convolution kernel

$$\widehat{B} = \widehat{K} \circ \widehat{X}, \tag{123}$$

where $\circ$ denotes the Hadamard or entry-wise product. Figure 25 illustrates this model with an example. In the frequency domain, we can see how the high-end of the spectrum of the image is suppressed by the filter. Recovering the high-resolution image from these data is easy. We just need to invert the action of the kernel,

$$\widehat{X}_{\text{est}} = \widehat{K}_{\text{dec}} \circ \widehat{Y} \tag{124}$$

where the spectrum of the *deconvolution* kernel is the inverse of the spectrum of the convolution kernel

$$\widehat{K}_{\text{dec}}[k_1, k_2] := \frac{1}{\widehat{K}_{\text{dec}}[k_1, k_2]}, \qquad 0 \leq k_1, k_2 \leq n-1, \tag{125}$$

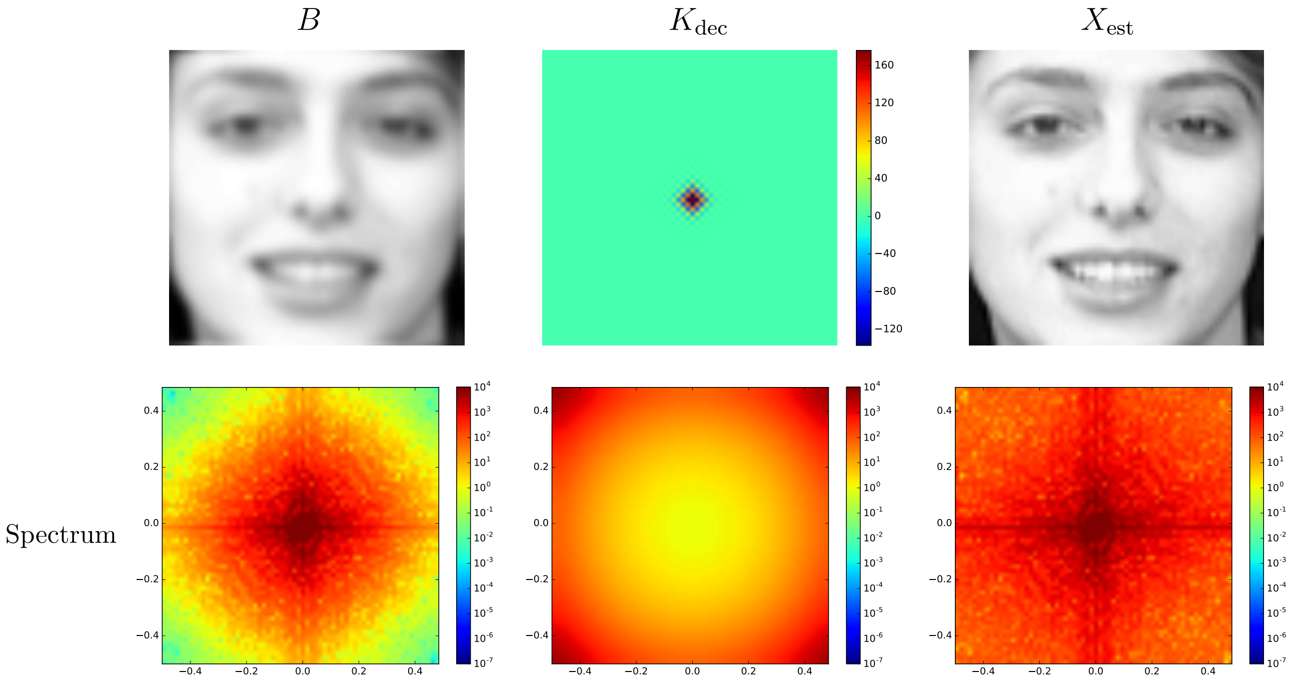


Figure 26: The top row shows a blurred image (left), a deconvolution kernel obtained by inverting the spectrum of the convolution kernel K in Figure 25 (center) and the corresponding deconvolved image (right). The bottom row shows the magnitudes of the 2D DFT coefficients of the three images on a logarithmic scale.

assuming that it is nonzero everywhere. Figure 26 shows that this scheme works perfectly on our simulated data. The deconvolution filter amplifies the high-frequency components to undo the effect of the convolution kernel.

Unfortunately, real data always contain noise. We take this into account by incorporating a noise term to our model

$$B_{\text{noisy}} = K * X + Z, \tag{126}$$

where $Z \in \mathbb{R}^{n\times n}$ represents the noise. Figure 27 shows the noisy data. As opposed to the image, which has most of its energy concentrated in the low frequencies, the energy in the noise is distributed uniformly across all the spectrum. This is not surprising, since the noise is iid Gaussian and the 2D DFT is an orthogonal transformation (up to a constant). By Theorem 2.3 in Lecture Notes 3, the DFT of the noise is consequently also Gaussian with a unit covariance matrix, so that the variance of each corresponding frequency coefficient is the same.

What happens if we apply our deconvolution scheme to the noisy blurred image? The result is catastrophic, as you can see in Figure 28. The reason is obvious when we look at this in the frequency domain. The high-end of the spectrum of the noisy blurred image is dominated by the noise. The deconvolution filter amplifies this high-frequency noise drowning out the actual image! In order to avoid this effect while deconvolving it is necessary to take into account the ratio between the noise level and the signal level at each frequency. Wiener filtering is a principled way of doing this if we can have a prior estimate of the spectral statistics of the signal and the noise.

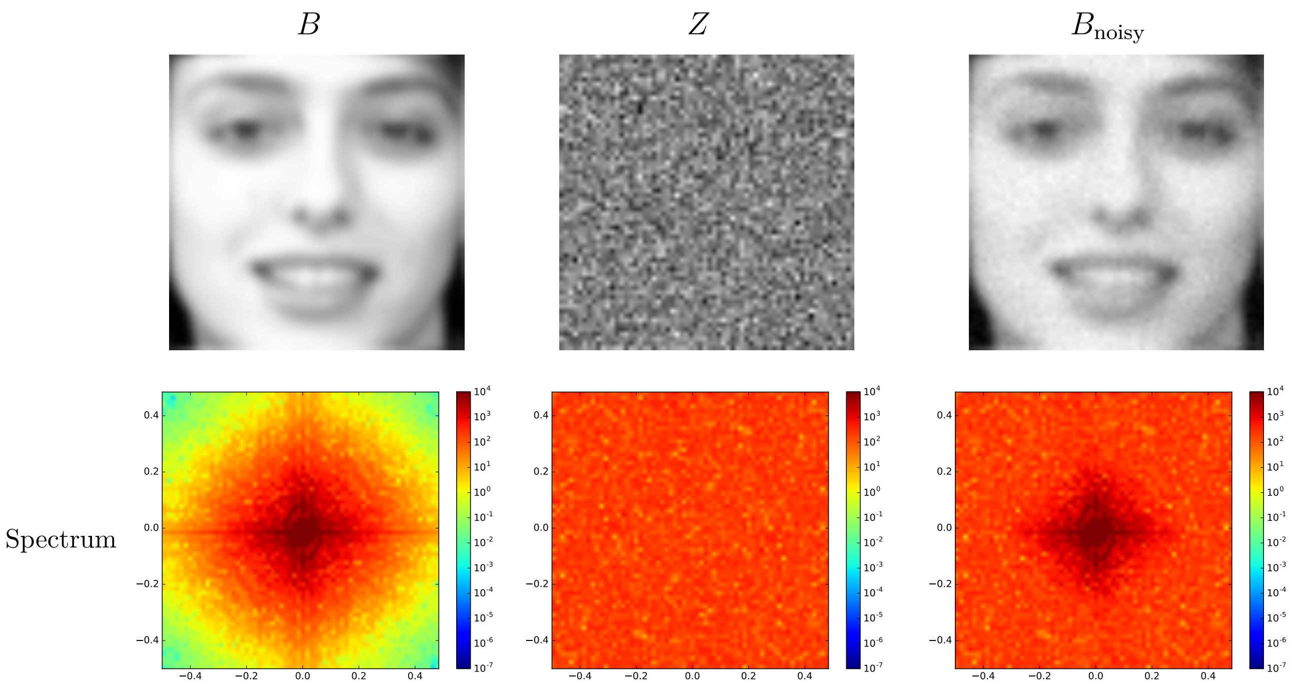


Figure 27: The top row shows a blurred image (left), additive Gaussian noise (center) and the corresponding noisy blurred image (right). The bottom row shows the magnitudes of the 2D DFT coefficients of the three images on a logarithmic scale.

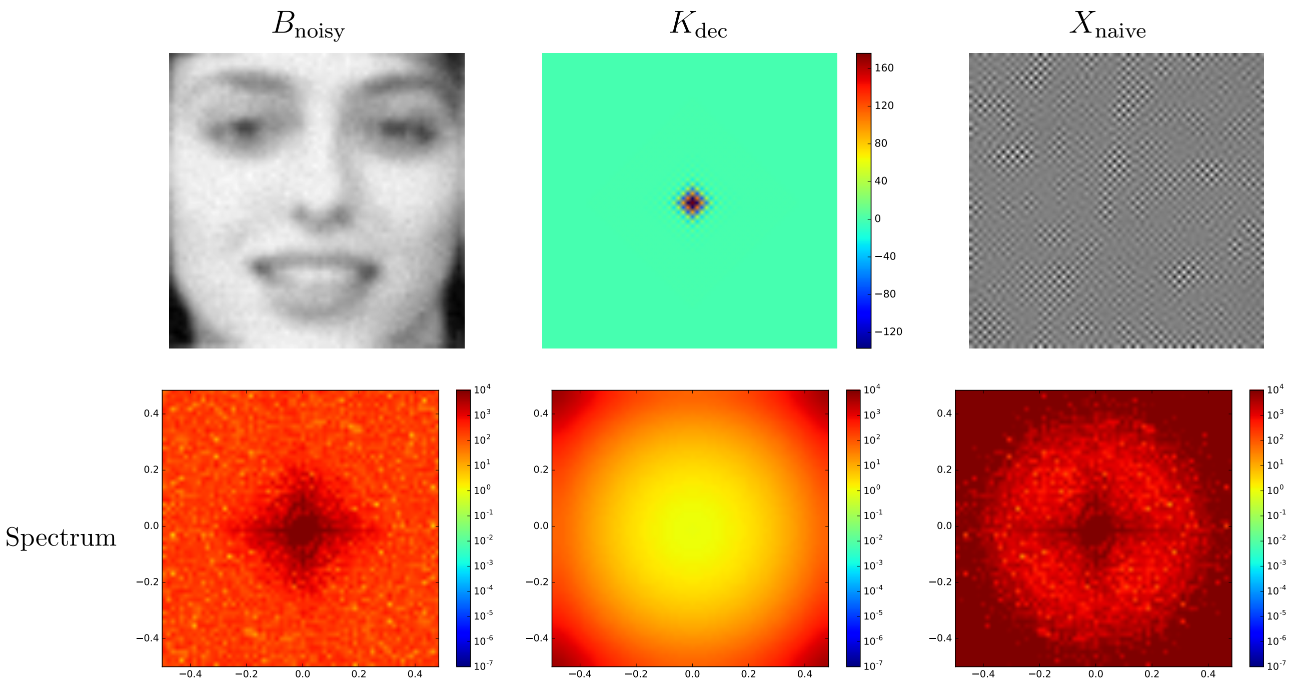


Figure 28: The top row shows a noisy blurred image (left), a deconvolution kernel obtained by inverting the spectrum of the convolution kernel K in Figure 25 (center) and the corresponding deconvolved image (right). The bottom row shows the magnitudes of the 2D DFT coefficients of the three images on a logarithmic scale.

The version of Wiener filtering that we present here makes the assumption that the DFT coefficients of the image and of the noise are uncorrelated. However, the method can be adapted to the case where the correlations are known. Let us model each DFT coefficient of the signal and the noise as random variables with known means and variances. For simplicity, we subtract their respective means, so that both random variables have zero mean. This makes it possible to interpret the random variables as vectors in the vector space of zero-mean random variables.

Theorem 2.11 (Vector space of zero-mean random variables)**.** *Zero-mean complex-valued random variables form a vector space with the usual summing and multiplication operations. The zero vector is the random variable that equals one with probability one.*

Proof. If two random variables have zero mean, any linear combination of the variables also has zero mean because of linearity of expectation. □

The covariance between two random variables is a valid inner product in this vector space, which means that uncorrelated random variables are orthogonal.

Theorem 2.12 (Inner product for zero-mean random variables)**.** *The covariance*

$$\langle \mathbf{x}, \mathbf{y} \rangle := \operatorname{Cov}(\mathbf{x}, \mathbf{y}) \tag{127}$$

$$= \operatorname{E}(\mathbf{x}\overline{\mathbf{y}}) \tag{128}$$

is a valid inner product in the vector space of zero-mean random variables. The corresponding inner-product norm is the variance,

$$||\mathbf{x}||_{\langle\cdot,\cdot\rangle} = \operatorname{Var}(\mathbf{x}) \tag{129}$$

Proof. This follows directly from linearity of expectation and the assumption that all the random variables are zero-mean. The fact that $\langle \mathbf{x}, \mathbf{x} \rangle = 0$ implies $\mathbf{x} = 0$ with probability one follows from Chebyshev's inequality. In more detail, if $||\mathbf{x}||_{\langle\cdot,\cdot\rangle} = 0$, for any $\epsilon > 0$

$$\operatorname{P}(|\mathbf{x}| > \epsilon) \leq \frac{\operatorname{Var}(\mathbf{x})}{\epsilon^2} = 0 \tag{130}$$

so $\mathbf{x}$ with probability one. □

The problem of estimating each DFT coefficient from the measured DFT coefficients now boils down to estimating a zero-mean random variable $\mathbf{x}$ given a measurement of the form

$$\mathbf{y} = a\mathbf{x} + \mathbf{z} \tag{131}$$

where a represents the value of the DFT coefficient of the convolution kernel. The following theorem derives the best linear estimate of $\mathbf{x}$ given $\mathbf{y}$ assuming that the signal and the noise are uncorrelated.

Theorem 2.13 (Linear estimation)**.** *Let*

$$\mathbf{y} = a\mathbf{x} + \mathbf{z} \tag{132}$$

where a is a known constant and $\mathbf{x}$ and $\mathbf{z}$ are uncorrelated zero-mean random variables with variances $\sigma_{\mathbf{x}}^2$ and $\sigma_{\mathbf{z}}^2$ respectively. The linear estimate of $\mathbf{x}$ given $\mathbf{y}$

$$\mathbf{x}_{\text{MMSE}} := w\mathbf{y} \tag{133}$$

that minimizes the mean square error

$$\mathrm{E}\left((\mathbf{x} - \mathbf{x}_{\text{MMSE}})^2 \right) \tag{134}$$

is given by

$$\mathbf{x}_{\text{MMSE}} = \frac{\overline{a}\sigma_{\mathbf{x}}^2 \mathbf{y}}{|a|^2 \sigma_{\mathbf{x}}^2 + \sigma_{\mathbf{z}}^2}. \tag{135}$$

Proof. We need to find the vector $\tilde{\mathbf{x}}$ in the span of $\mathbf{y}$ that minimizes

$$||\mathbf{x} - \tilde{\mathbf{x}}||_{\langle \cdot, \cdot \rangle} = \sqrt{\mathrm{E}\left((\mathbf{x} - \tilde{\mathbf{x}})^2 \right)}. \tag{136}$$

By basic linear algebra, this is just the orthogonal projection of $\mathbf{y}$ onto $\mathbf{x}$! The projection is given by

$$\mathcal{P}_{\text{span}(\mathbf{y})}\, \mathbf{x} = \left\langle \mathbf{x}, \frac{\mathbf{y}}{||\mathbf{y}||_{\langle \cdot, \cdot \rangle}} \right\rangle \frac{\mathbf{y}}{||\mathbf{y}||_{\langle \cdot, \cdot \rangle}} \tag{137}$$

This implies that

$$\begin{aligned}
w &= \frac{1}{||\mathbf{y}||_{\langle \cdot, \cdot \rangle}} \left\langle \mathbf{x}, \frac{\mathbf{y}}{||\mathbf{y}||_{\langle \cdot, \cdot \rangle}} \right\rangle && (138) \\
&= \frac{\langle \mathbf{x}, a\mathbf{x} + \mathbf{z} \rangle}{||a\mathbf{x} + \mathbf{z}||_{\langle \cdot, \cdot \rangle}^2} && (139) \\
&= \frac{\overline{a}\, ||\mathbf{x}||_{\langle \cdot, \cdot \rangle}^2 + \overline{a} \langle \mathbf{x}, \mathbf{z} \rangle}{||a\mathbf{x} + \mathbf{z}||_{\langle \cdot, \cdot \rangle}^2} && (140) \\
&= \frac{\overline{a}\, ||\mathbf{x}||_{\langle \cdot, \cdot \rangle}^2}{|a|^2\, ||\mathbf{x}||_{\langle \cdot, \cdot \rangle}^2 + ||\mathbf{z}||_{\langle \cdot, \cdot \rangle}^2} \quad \text{by orthogonality and the Pythagorean theorem} && (141) \\
&= \frac{\overline{a}\sigma_{\mathbf{x}}^2}{|a|^2 \sigma_{\mathbf{x}}^2 + \sigma_{\mathbf{z}}^2}. && (142)
\end{aligned}$$

□

Wiener filtering consists of estimating the statistics of the signal and the noise, and then applying the best linear estimate at each frequency coefficient. In this version of the method we assume that the DFT coefficients of the signal and noise are all uncorrelated and that the noise has zero mean.

Algorithm 2.14 (Wiener filtering)**.** *Given a noisy blurred image $B_{\text{noisy}} \in \mathbb{R}^{n \times n}$ and a kernel $K \in \mathbb{R}^{n \times n}$, apply the following steps:*

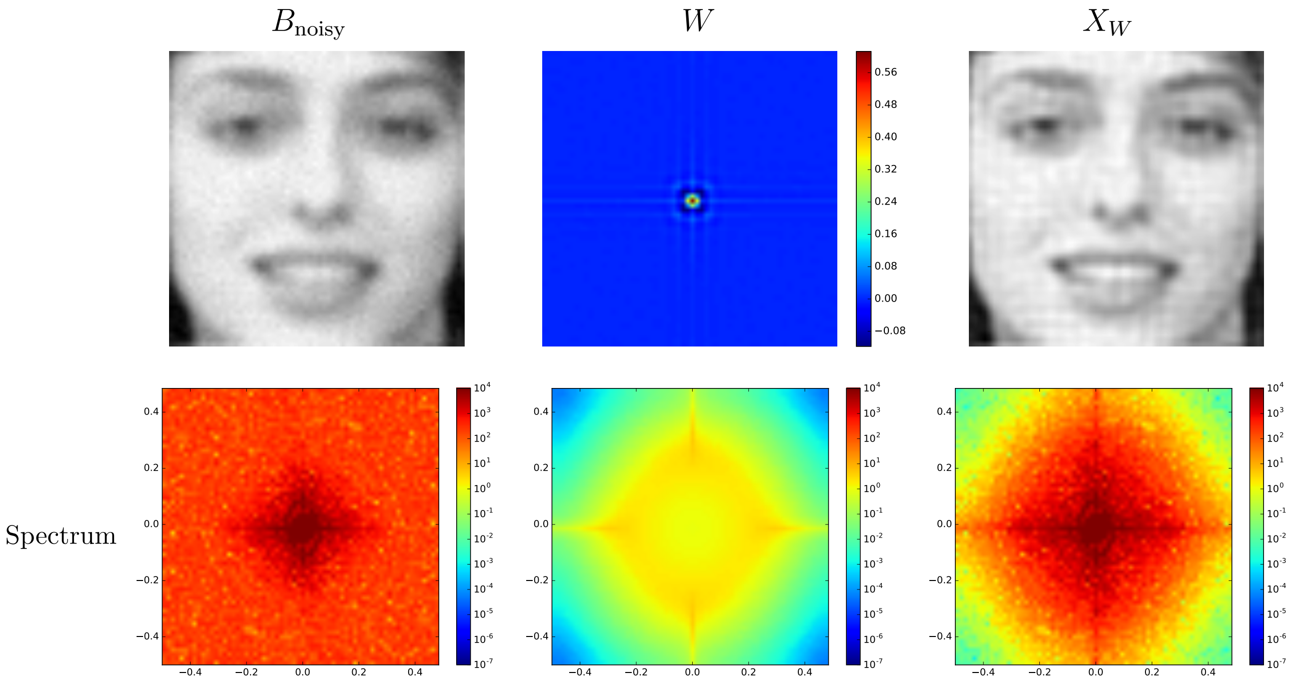


Figure 29: The top row shows a blurred image (left), the Wiener filter obtained by applying Algorithm 2.14 (center) and the corresponding deconvolved image (right). The bottom row shows the magnitudes of the 2D DFT coefficients of the three images on a logarithmic scale.

1. *Estimate the variance of each 2D DFT coefficient of the noise* $\sigma_Z\left[k_1,k_2\right]^2$.

2. *Estimate the mean* $\mu_X\left[k_1,k_2\right]$ *and variance* $\sigma_X\left[k_1,k_2\right]^2$ *of each 2D DFT coefficient* $\widehat{X}\left[k_1,k_2\right]$ *of the image using a database of images.*

3. *Compute the 2D DFT coefficients of the noisy blurred image* $\widehat{B}_{\text{noisy}}$ *and the kernel* $\widehat{K}$.

4. *For* $0 \le k_1, k_2 \le n-1$

   - *Center* $\widehat{B}_{\text{noisy}}[k_1,k_2]$ *by subtracting* $\widehat{K}[k_1,k_2]\mu_X\left[k_1,k_2\right]$.
   - *Set*

$$W[k_1,k_2] := \frac{\overline{\widehat{K}[k_1,k_2]}\sigma_X\left[k_1,k_2\right]^2}{|\widehat{K}[k_1,k_2]|^2\sigma_X\left[k_1,k_2\right]^2+\sigma_Z\left[k_1,k_2\right]^2} \tag{143}$$

$$X_W := \mu_X\left[k_1,k_2\right] + W[k_1,k_2]\widehat{B}_{\text{noisy}}\left[k_1,k_2\right]. \tag{144}$$

5. *Compute the inverse 2D DFT of* X_W.

Figure 29 shows the result of applying Wiener filtering to our running example.

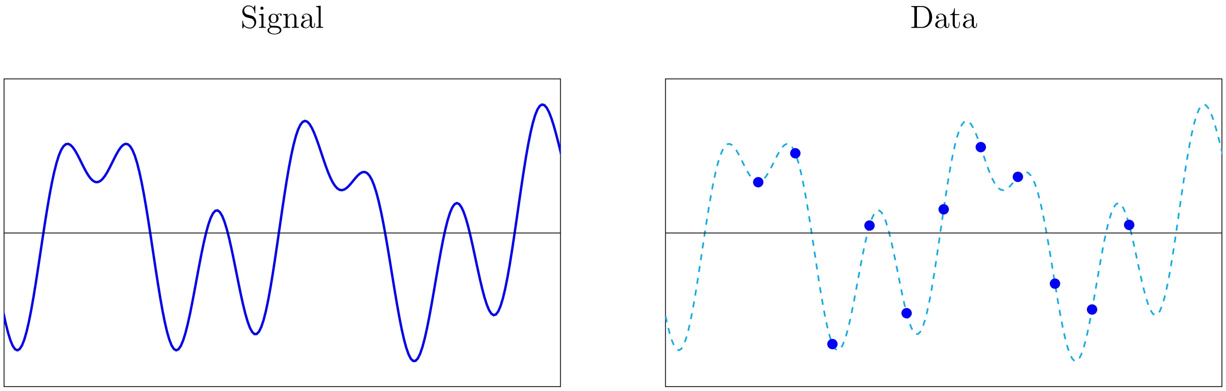


Figure 30: The goal of spectral super-resolution is to estimate the frequencies of a multisinusoidal signal like the one on the left from the finite samples shown on the right.

3 Spectral Super-resolution

3.1 The spectral super-resolution problem

The aim of spectral super-resolution is to estimate the components of a multisinusoidal signal from finite data. This problem arises for example in radar, radio telescopy and other applications where we want to determine the direction of arrival of a propagating wave using an array of sensors. Here we will look at a basic version of the problem. Consider a multisinusoidal signal of the form

$$g(t) := \sum_{j=1}^{s} \vec{c}\,[j] \exp\left(-i 2\pi f_j t\right), \tag{145}$$

which is the weighted sum of s complex sinusoids. The goal of spectral super-resolution is to estimate the frequencies f_1, f_2, ..., f_s and the corresponding complex-valued amplitudes $\vec{c}\,[1]$, $\vec{c}\,[2]$, ..., $\vec{c}\,[s]$ from a finite number of samples of g, as illustrated in Figure 30 (in the figure the amplitudes are real for ease of visualization). We assume that the frequencies lie in a unit interval, for example in $[-1/2, 1/2]$ and the samples are measured at integer values from $-(n-1)/2$ to $(n-1)/2$ (n is assumed to be odd to simplify the exposition).

One can model the *spectrum* of g as spikes with amplitudes c_1, c_2, ..., c_s located at the frequencies f_1, f_2, ..., f_s. To make this mathematically precise, we introduce the Dirac measure.

Definition 3.1 (Dirac measure)**.** *A Dirac measure $\delta_{[\tau]}$ is a measure associated to a point τ, which assigns a value of one to sets that contain τ and zero to sets that do not,*

$$\int_{\mathcal{S}} \delta_{[\tau]}(du) = \begin{cases} 1 & \text{if } \tau \in \mathcal{S}, \\ 0 & \text{otherwise.} \end{cases} \tag{146}$$

Even though it is often called a Dirac delta function*, the Dirac measure is not a function. It can however be interpreted as a distribution or a generalized function such that for any function h we have*

$$\int_{\mathcal{S}} h(u)\, \delta_{[\tau]}(du) = \begin{cases} h(\tau) & \text{if } \tau \in \mathcal{S}, \\ 0 & \text{otherwise.} \end{cases} \tag{147}$$

Although a rigorous justification is beyond the scope of these notes, one can define the Fourier series of a distribution and in particular of a Dirac measure. The coefficients are obtained by integrating the complex sinusoidal atoms h_k, $k \in \mathbb{Z}$, against the measure. We can now interpret the spectral super-resolution problem as that of estimating the support of the measure

$$\mu_g := \sum_{j=1}^{s} \vec{c}\,[j]\delta_{[f_j]} \tag{148}$$

from a finite subset of Fourier coefficients

$$\begin{aligned}
\int_{-1/2}^{1/2} \overline{h_k\,(u)}\mu_g\,(\,\mathrm{d}u) &= \sum_{j=1}^{s} \vec{c}\,[j] \int_{-1/2}^{1/2} \overline{h_k\,(u)}\delta_{[f_j]}\,(\,\mathrm{d}u) && (149)\\
&= \sum_{j=1}^{s} \vec{c}\,[j] \exp\left(-i2\pi k f_j\right) && (150)\\
&= g\,(k)\,, \qquad -\frac{n-1}{2} \leq k \leq \frac{n-1}{2}. && (151)
\end{aligned}$$

In the next section it will become apparent why this is a super-resolution problem.

3.2 The periodogram

In order to estimate the frequencies of g from the available samples, we can compute the truncated Fourier series of the measure μ_g. This is known as the periodogram in the signal processing literature.

Definition 3.2 (Periodogram)**.** *The periodogram of a vector of data $\vec{y} \in \mathbb{C}^n$ is defined as*

$$P_{\vec{y}}\,(u) := \sum_{k=-\frac{n-1}{2}}^{\frac{n-1}{2}} \vec{y}\,[k]\,h_k\,(u)\,, \tag{152}$$

where h_k is a complex sinusoid with integer frequency k and the entries of the data are numbered from $-\,(n-1)\,/2$ to $(n-1)\,/2$ for ease of exposition.

The periodogram can be interpreted as the convolution between a Dirichlet kernel with cut-off frequency $(n-1)\,/2$ and the measure μ_g.

Lemma 3.3. *The periodogram of $g\,(-\,(n-1)\,/2)$, $\ldots$, $g\,((n-1)\,/2)$ where g is the multisinusoidal function defined by equation (145) equals*

$$P_g\,(u) = \sum_{j=1}^{s} \vec{c}\,[j] d_{[f_j]}\,(u)\,. \tag{153}$$

Proof. By Definition 1.10 and Lemma 1.12 the Fourier coefficients of a Dirichlet kernel with cut-off frequency $(n-1)/2$ shifted by f are equal to

$$D_{[f]}\,[k] := \begin{cases} \exp\left(-i2\pi k f\right) & \text{if } |k| \leq (n-1)/2 \\ 0 & \text{otherwise.} \end{cases} \tag{154}$$

This implies that the samples

$$g\left(k\right)=\sum_{j=1}^{s}\vec{c}[j]D_{[f_j]}\left[k\right], \qquad -\frac{n-1}{2}\leq k\leq\frac{n-1}{2}, \tag{155}$$

are exactly equal to the Fourier coefficients of the periodogram, so that

$$P_g\left(u\right)=\sum_{k=-\frac{n-1}{2}}^{\frac{n-1}{2}}g\left(k\right)h_k\left(u\right) \tag{156}$$

$$=\sum_{j=1}^{s}\vec{c}[j]\sum_{k=-\frac{n-1}{2}}^{\frac{n-1}{2}}D_{[f_j]}\left[k\right]h_k\left(u\right) \tag{157}$$

$$=\sum_{j=1}^{s}\vec{c}[j]d_{[f_j]}\left(u\right). \tag{158}$$

□

Recall that the width of the main lobe of a Dirichlet kernel is inversely proportional to its cut-off frequency, which in this case equals $(n-1)/2$. As we gather more samples, the widths become narrower and narrower, revealing the location of the frequencies of interest. The *resolution* at which we observe the spectrum μ_g is consequentially tied to the number of data. This is illustrated in Figure 31.

Computing the periodogram does not solve the super-resolution problem, it just allows to visualize a lower resolution version of μ_g. If the frequencies $f_1, \ldots, f_s$ are far apart, the local maxima of P are a good indication of their location. The problem with this approach is that the side lobes corresponding to large amplitudes may mask the presence of smaller spikes. As a result, the periodogram is not very useful if the spikes are not far enough from each other or if their amplitudes differ substantially, *even if no noise is present in the data.* The image at the center of Figure 32 illustrates this: detecting some of the lower-amplitude spikes from the periodogram is impossible.

In order to alleviate the interference caused by the side lobes of the Dirichlet kernel, one can *window* the data before computing the periodogram.

Definition 3.4 (Windowed periodogram)**.** *Let us define a bandlimited window function w with Fourier coefficients $W\left[k\right]$ that are only nonzero if $|k|\leq(n-1)/2$. The windowed periodogram of a vector of data $\vec{y}\in\mathbb{C}^n$ is defined as*

$$P_{w,\vec{y}}\left(u\right):=\sum_{k=-\frac{n-1}{2}}^{\frac{n-1}{2}}W\left[k\right]g\left(k\right)h_k\left(u\right). \tag{159}$$

Following the exact same reasoning as in the proof of Lemma 3.3,

$$P_{w,\vec{y}}\left(u\right):=\sum_{j=1}^{s}\vec{c}[j]w_{[f_j]}\left[u\right]. \tag{160}$$

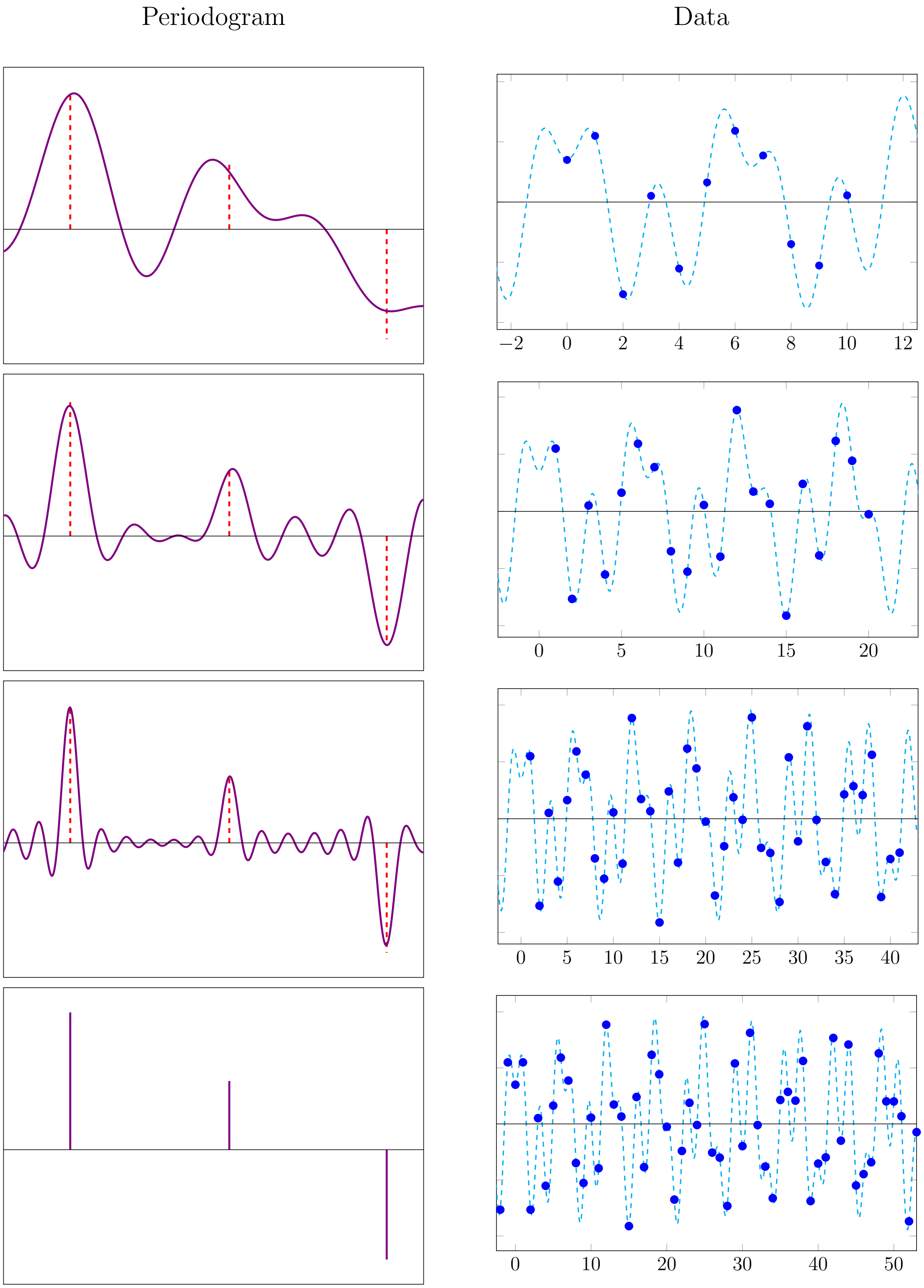


Figure 31: Periodogram (left) and corresponding samples (right) for different numbers of measurements (increasing downwards). In the limit where $n \times \infty$ the frequencies are completely resolved.

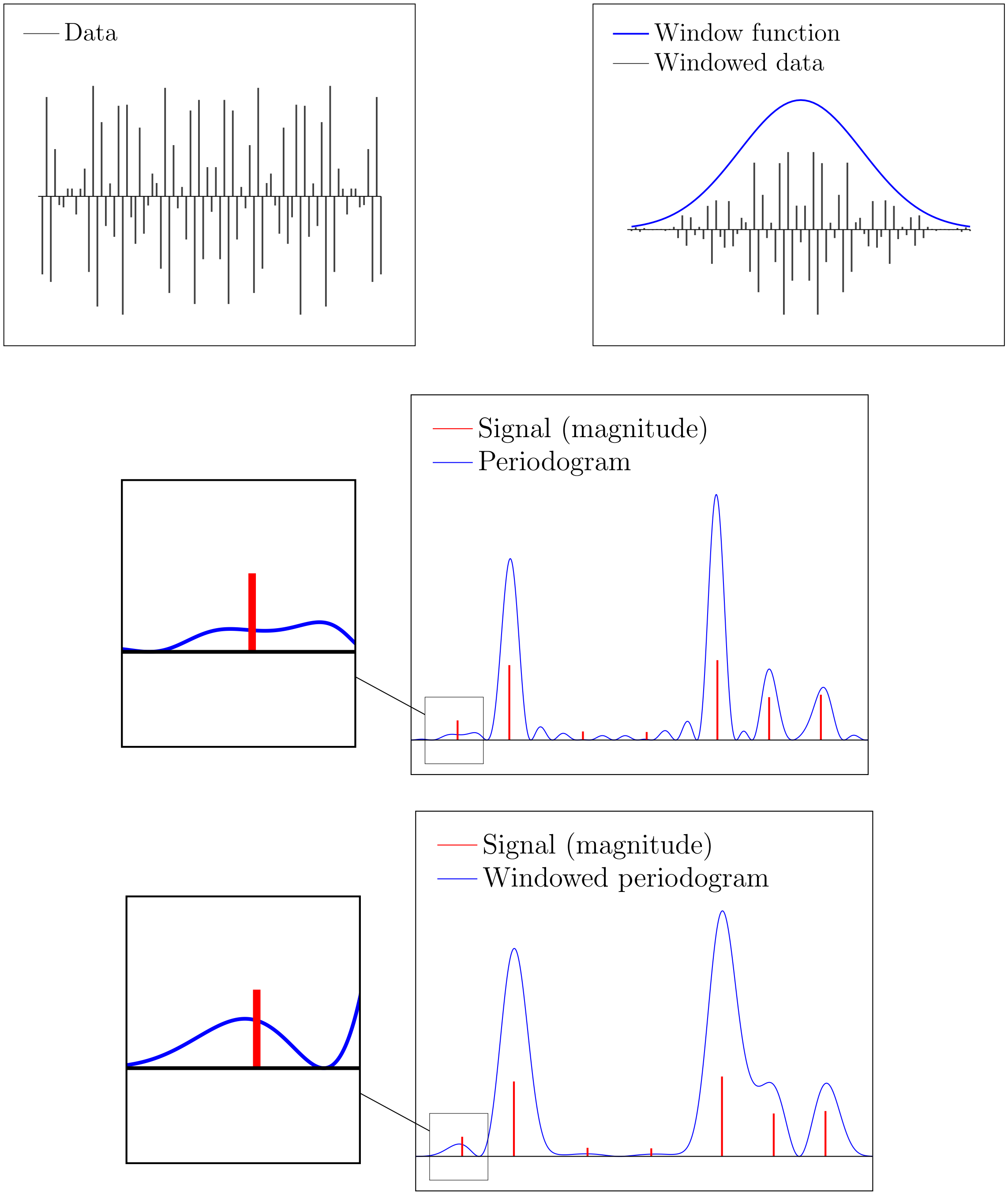


Figure 32: Data before (top left) and after applying a window function (top right). No noise is added to the data. Below we see the periodogram (center) and windowed periodogram (bottom) computed from the data. The scaling of the periodograms is set so that both the large and small peaks can be seen on the plot.

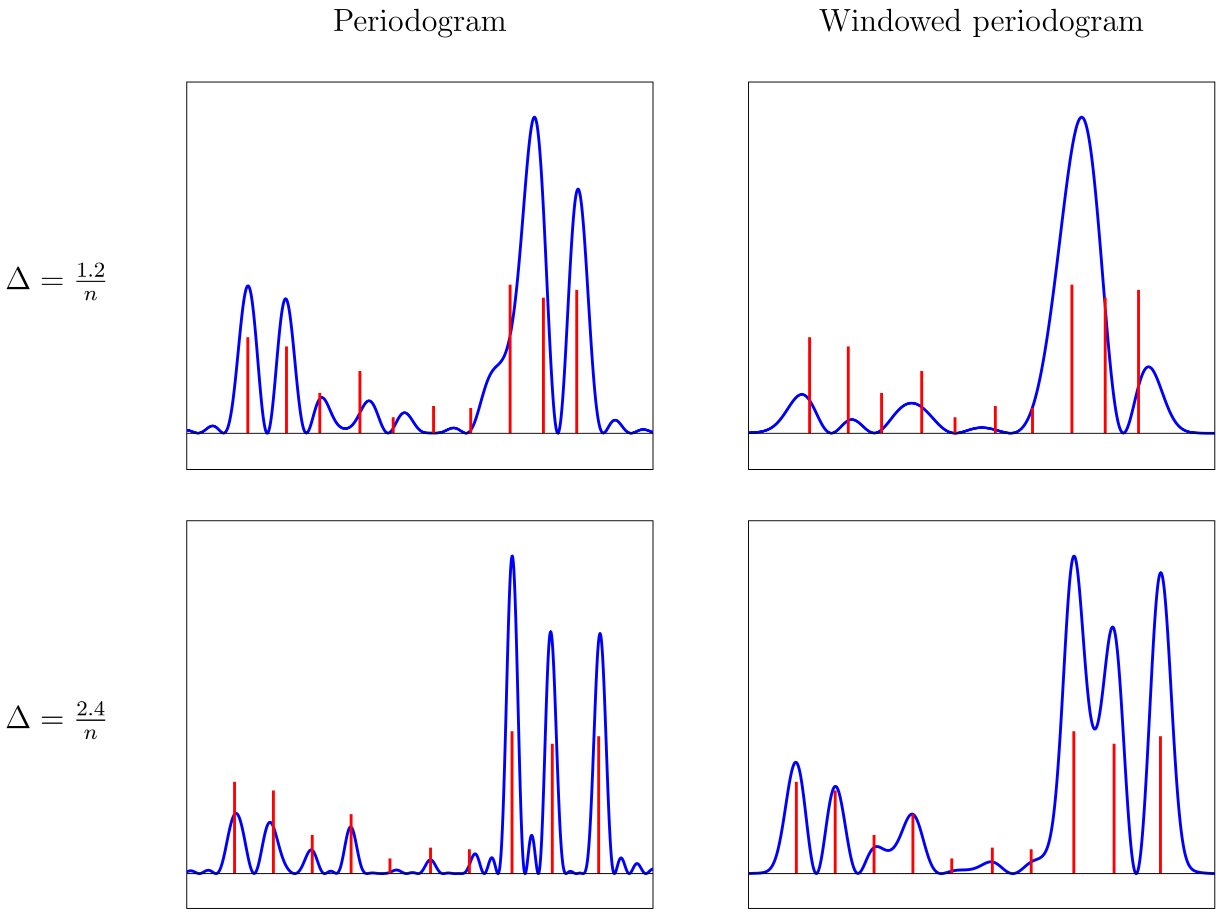


Figure 33: Periodogram (left) and windowed periodogram (right) for two signals with identical amplitudes but different minimum separations.

where w denotes the inverse Fourier transform of the window function. Ideally, w should be as *spiky* as possible to make it easier to locate the frequency locations from the windowed periodogram. However, this is challenging due to the constraint that $\widehat{w}$ has finite support and hence w is a low-pass function.

In the image on the top right of Figure 32 we apply a Gaussian window to the data. To be more precise, we set W to be a truncated Gaussian, so that w is also approximately Gaussian. The resulting periodogram, shown at the center of Figure 32, has much less *spectral leakage* from the largest signal components, due to the fact that the Gaussian window has lower side lobes than the periodized sinc. However, the latter is spikier at the origin, which allows to better distinguish neighboring spikes with similar amplitudes. In general, designing an adequate window implies finding a good tradeoff between the width of the main lobe and the height of the side lobes. We refer the reader to [2] for a detailed account of design considerations and types of window function.

3.3 Prony's method

Prony's method solves the spectral super-resolution problem exactly in the absence of noise by encoding the position of the s frequencies of interest as the zeros of a trigonometric polynomial of order s. The following theorem shows that such a polynomial always exists.

Theorem 3.5 (Prony polynomial)**.** *Given any set of frequencies f_1, f_2, ..., f_s in the unit interval, there exists a nonzero complex polynomial of order s*

$$p\left(z\right) := \sum_{k=0}^{s} P[k] z^k, \tag{161}$$

which we call a Prony polynomial, such that its s roots are equal to $\exp\left(2\pi f_1\right)$, $\exp\left(2\pi f_2\right)$, ..., $\exp\left(2\pi f_s\right)$.

Proof. Consider the polynomial

$$p(z) := \prod_{j=1}^{s} \left(1 - \exp\left(-i2\pi f_j\right) z\right). \tag{162}$$

If we expand the product, we have

$$p(z) = 1 + \sum_{k=1}^{s} P\left[k\right] z^k \tag{163}$$

for some $P\left[1\right]$, ..., $P\left[k\right]$. Since $p\left(0\right) = 1$, the polynomial is nonzero and by the fundamental theorem of algebra it has at most s roots. By construction,

$$p(\exp\left(i2\pi f_j\right)) = 0 \tag{164}$$

for $1 \leq j \leq s$, which establishes the result. □

If we are able to compute such a polynomial, then finding its roots immediately reveals the frequencies of interest. Prony's method consists of building a system of linear equations from the available data, such that its solution is equal to the coefficients of the Prony polynomial. The following theorem shows how to build the system.

Theorem 3.6 (Prony system)**.** *Let $P[k]$ denote the kth coefficient of the Prony polynomial defined in Theorem 3.5 and let g be the multisinusoidal function defined by equation (145), for any integer b*

$$\sum_{l=0}^{s} P[l] g[l-b] = 0. \tag{165}$$

Proof.

$$\begin{aligned}
\sum_{l=0}^{s} P[l]g[l-b] &= \sum_{l=0}^{s} P[l] \int_{-1/2}^{1/2} \overline{h_{l-b}\,(u)}\mu_g\,(\,\mathrm{d}u) \qquad \text{by (151)} && (166)\\
&= \int_{-1/2}^{1/2} \exp\,(i2\pi bu) \sum_{l=0}^{s} P[l] \exp\,(-i2\pi lu)\,\mu_g\,(\,\mathrm{d}u) && (167)\\
&= \int_{-1/2}^{1/2} \exp\,(i2\pi bu)\,p\,(\exp\,(-i2\pi u))\,\mu_g\,(\,\mathrm{d}u) && (168)\\
&= \sum_{j=1}^{s} \vec{c}[j] \exp\,(i2\pi bf_j)\,p\,(\exp\,(-i2\pi f_j)) && (169)\\
&= 0. && (170)
\end{aligned}$$

□

Notice that equation (165) only involves samples of g between $-b$ and $s-b$. Prony's method consists of setting up enough such equations in a system so that the solution yields the coefficients of the Prony polynomial, which can then be used to estimate the frequencies.

Algorithm 3.7 (Prony's method)**.** *The input is the number of frequencies s that we aim to estimate and $2s+1$ uniform samples of multisinusoidal function defined by equation* (145)*.*

1. *Form the system of equations*

$$\begin{bmatrix} g(1) & g(2) & \cdots & g(s) \\ g(0) & g(1) & \cdots & g(s-1) \\ \cdots & \cdots & \cdots & \cdots \\ g(-s+2) & g(-s+3) & \cdots & g(1) \end{bmatrix} \vec{P} = - \begin{bmatrix} g(0) \\ g(-1) \\ \cdots \\ g(-s+1) \end{bmatrix}, \tag{171}$$

   where $\vec{P} \in \mathbb{C}^s$.

2. *Solve the system to obtain $\vec{P}$.*

3. *Root the polynomial*

$$p\,(z) := 1 + \sum_{k=1}^{s} \vec{P}[k] z^k, \tag{172}$$

   to obtain its s roots $z_1, \ldots, z_s$.

4. *For every root on the unit circle $z_j = \exp\,(i2\pi\tau)$ include τ in the set of estimated frequencies.*

This procedure is guaranteed to achieve exact recovery of the original signal. This implies that in a noiseless scenario spectral super-resolution is achieved using only $n = 2s+1$ measurements, which is essentially optimal since we need to estimate $2s$ free parameters (the s frequencies and the corresponding amplitudes).

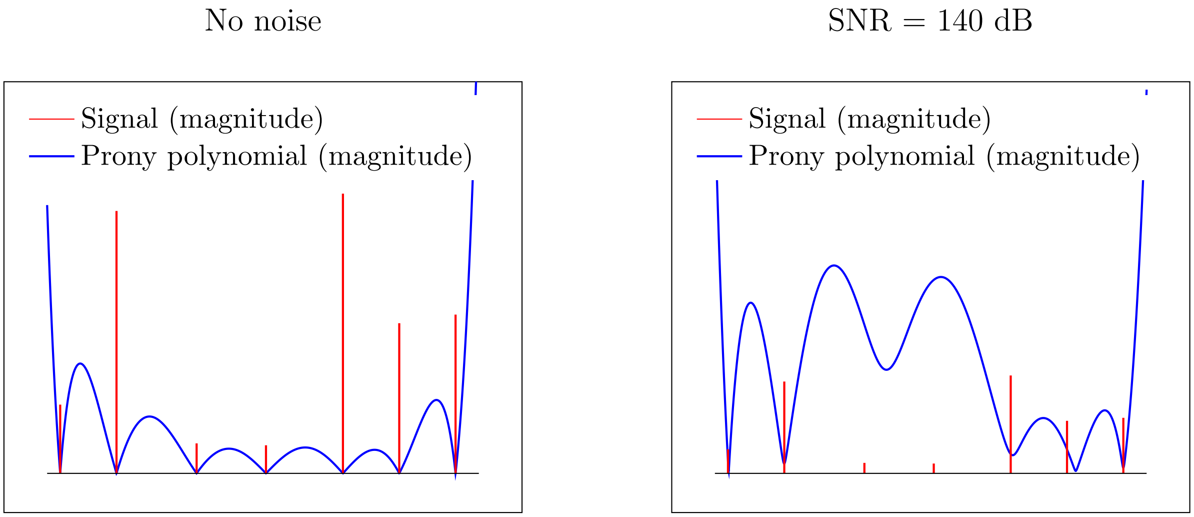


Figure 34: Prony polynomial applied on noiseless data (left). The image on the right shows the effect of adding a very small quantity of noise to the data. The roots of the polynomial no longer coincide with the frequencies of the original signal. Note that the vertical axis is scaled differently in the two images.

Lemma 3.8. *In the absence of noise, Prony's method recovers the frequencies of a multisinusoidal function of the form* (145) *exactly.*

Proof. The coefficients of the polynomial (161) are a feasible solution for the system of equations (171). In fact, they are the unique solution. To show this we compute the factorization

$$\begin{bmatrix} g(1) & g(2) & \cdots & g(s) \\ g(0) & g(1) & \cdots & g(s-1) \\ & & \cdots & \\ g(-s+2) & g(-s+3) & \cdots & g(1) \end{bmatrix} = \tag{173}$$

$$\begin{bmatrix} e^{-i2\pi f_1} & e^{-i2\pi f_2} & \cdots & e^{-i2\pi f_s} \\ 1 & 1 & \cdots & 1 \\ & & \cdots & \\ e^{-i2\pi(2-s)f_1} & e^{-i2\pi(2-s)f_2} & \cdots & e^{-i2\pi(2-s)f_s} \end{bmatrix} \begin{bmatrix} \vec{c}[1] & 0 & \cdots & 0 \\ 0 & \vec{c}[2] & \cdots & 0 \\ & & \cdots & \\ 0 & 0 & \cdots & \vec{c}[s] \end{bmatrix} \begin{bmatrix} 1 & e^{-i2\pi f_1} & \cdots & e^{-i2\pi(s-1)f_1} \\ 1 & e^{-i2\pi f_2} & \cdots & e^{-i2\pi(s-1)f_2} \\ & & \cdots & \\ 1 & e^{-i2\pi f_s} & \cdots & e^{-i2\pi(s-1)f_s} \end{bmatrix}.$$

The diagonal matrix is full rank as long as all the coefficients $\vec{c}[j]$ are nonzero, whereas the two remaining matrices are full rank by the following lemma, proved in Section 4.1 of the appendix.

Lemma 3.9 (Vandermonde matrix)**.** *For any distinct set of s nonzero complex numbers $z_1, z_2, \ldots, z_s$*

and any positive integers m_1, m_2, s such that $m_2 - m_1 + 1 \geq s$ the Vandermonde matrix

$$\begin{bmatrix} z_1^{m_1} & z_2^{m_1} & \cdots & z_s^{m_1} \\ z_1^{m_1+1} & z_2^{m_1+1} & \cdots & z_s^{m_1+1} \\ z_1^{m_1+2} & z_2^{m_1+2} & \cdots & z_s^{m_1+2} \\ & & \cdots & \\ z_1^{m_2} & z_2^{m_2} & \cdots & z_s^{m_2} \end{bmatrix} \tag{174}$$

is full rank.

As a result, the matrix in (171) is full rank, so the system of equations has a unique solution equal to (161). This completes the proof. □

Unfortunately, Prony's method as presented above cannot be applied to real data even if the signal-to-noise ratio is exceptionally high. The image on the left of Figure 34 shows how the Prony polynomial allows to super-resolve the spectrum to very high accuracy from noiseless data. However, on the right we see the result of applying the method to data that have a very small amount of noise (the ratio between the ℓ_2 norm of the noise and the noiseless data is around 10^{-8}!). The roots of the Prony polynomial are perturbed away from the points of the unit circle that correspond to the true frequencies, so that it is no longer possible to achieve accurate spectral super-resolution. It is consequently necessary to adapt the method to deal with noisy data if there is to be any hope of applying it in any realistic scenario. This is the subject of the following section.

3.4 Subspace methods

In this section we consider the spectral super-resolution problem when noise is added to the data. First, we will generalize Prony's method so that it can use more data than just $2s + 1$ samples. Applying Prony's method is equivalent to finding a nonzero vector in the null space of $Y(s+1)^T$, where $Y(m)$ is defined for any integer m as the Hankel matrix

$$Y(m) := \begin{bmatrix} \vec{y}[0] & \vec{y}[1] & \cdots & \vec{y}[n-m] \\ \vec{y}[1] & \vec{y}[2] & \cdots & \vec{y}[n-m+1] \\ \cdots & \cdots & \cdots & \cdots \\ \vec{y}[m-1] & \vec{y}[m] & \cdots & \vec{y}[n-1] \end{bmatrix}, \tag{175}$$

where

$$\vec{y}[k] = g(-k+1) \qquad 0 \leq k \leq n. \tag{176}$$

The vector in the null space of $Y(s+1)^T$ corresponds to the coefficients of the Prony polynomial, which we can root to find the frequencies of g.

In the absence of noise, $Y(m)$ can be decomposed in the following way

$$\begin{aligned} Y(m) &= \begin{bmatrix} \vec{a}_{0:m-1}(f_1) & \vec{a}_{0:m-1}(f_2) & \cdots & \vec{a}_{0:m-1}(f_s) \end{bmatrix} \begin{bmatrix} c_1 & 0 & \cdots & 0 \\ 0 & c_2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & c_s \end{bmatrix} \begin{bmatrix} \vec{a}_{0:n-m}(f_1)^T \\ \vec{a}_{0:n-m}(f_2)^T \\ \cdots \\ \vec{a}_{0:n-m}(f_s)^T \end{bmatrix} \\ &= A_{0:m-1}\, C\, A_{0:m}^T, \end{aligned} \tag{177}$$

where for $k > 0$ we define

$$\vec{a}_{0:k}(u) := \begin{bmatrix} 1 & \exp(-i2\pi u) & \exp(-i2\pi 2u) & \cdots & \exp(-i2\pi k u) \end{bmatrix}^T, \tag{178}$$

$$A_{0:k} := \begin{bmatrix} \vec{a}_{0:k}(f_1) & \vec{a}_{0:k}(f_2) & \cdots & \vec{a}_{0:k}(f_s) \end{bmatrix}. \tag{179}$$

This decomposition suggests an alternative way of estimating the spectrum from $Y(m)$: finding sinusoidal atoms $\vec{a}_{0:m-1}(u)$ that are in the column space of $Y(m+1)$. Lemma 3.10 below proves that the only atoms of this form that belong to the column space of $Y(m)$ are precisely $\vec{a}_{0:m-1}(f_1)$, $\vec{a}_{0:m-1}(f_2)$, ..., $\vec{a}_{0:m-1}(f_s)$.

In order to apply the same idea in the presence of noise, where

$$\vec{y}[k] = g(-k+1) + \vec{z}[k] \qquad 0 \le k \le n, \tag{180}$$

for some additive perturbation $\vec{z}$, we check what atoms are *close* to the column space of $Y(m)$. To quantify this we compute the orthogonal complement $\mathcal{N}$ of the column space of $Y(m)$ and construct the *pseudospectrum*

$$P_{\mathcal{N}}(u) = \log \frac{1}{\left|\mathcal{P}_{\mathcal{N}}\left(\vec{a}_{0:m-1}(u)\right)\right|^2}, \tag{181}$$

where $\mathcal{P}_{\mathcal{N}}$ denotes a projection onto $\mathcal{N}$. If an atom is almost orthogonal to $\mathcal{N}$ then $P_{\mathcal{N}}$ will have a very large value at that point.

The following lemma shows that the maxima of the pseudospectrum reveal the frequencies of interest of the signal in the noiseless case.

Lemma 3.10. *Let $\mathcal{N}$ be the null space of the empirical covariance matrix $\Sigma(m)$ for $m \ge s$. Then*

$$P_{\mathcal{N}}(f_j) = \infty, \qquad 1 \le j \le s, \tag{182}$$

$$P_{\mathcal{N}}(u) < \infty, \qquad \text{for } u \notin \{f_1, \ldots, f_s\}. \tag{183}$$

Proof. By (177) the atoms $\vec{a}_{0:m-1}(f_1)$, ..., $\vec{a}_{0:m-1}(f_s)$ span the column space of $Y(m)$ and $\Sigma(m)$. As a result they are orthogonal to the null space $\mathcal{N}$ of the empirical covariance matrix, which proves (182).

We prove (183) by contradiction. The atoms $\vec{a}_{0:m-1}(f_1)$, ..., $\vec{a}_{0:m-1}(f_s)$ span the orthogonal complement to $\mathcal{N}$. As a result, if $\vec{a}_{0:m-1}(u)$ is orthogonal to $\mathcal{N}$ for some u then $\vec{a}_{0:m-1}(u)$ is in the span of $\vec{a}_{0:m-1}(f_1)$, ..., $\vec{a}_{0:m-1}(f_s)$. This would imply that $A_{0:m-1}^T(T \cup \{u\})$ is not full rank, which can only hold if $u \in \{f_1, \ldots, f_s\}$ by Lemma 3.9. □

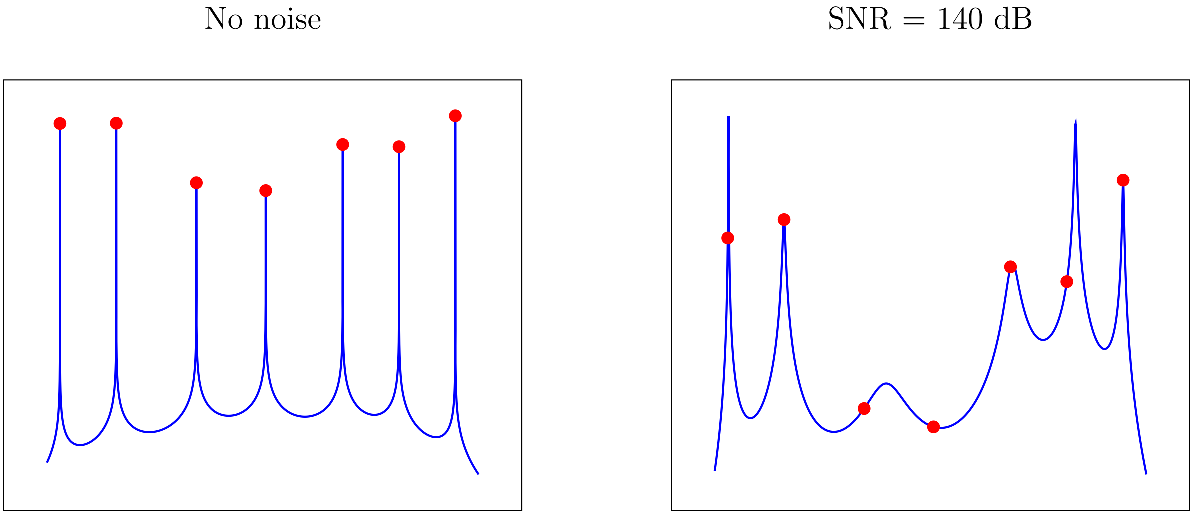


Figure 35: Pseudospectrum corresponding to the data used to construct the Prony polynomials in Figure 34. The true frequencies of interest are marked with red circles.

Figure 35 shows the pseudospectrum corresponding to the data used to construct the Prony polynomials in Figure 34 when $n = 2s + 1$. In the noiseless case, the pseudospectrum achieves exact super-resolution. Unfortunately, the locations of the local maxima are severely perturbed by even a very small amount of noise.

The subspace $\mathcal{N}$ is the null space of the empirical covariance matrix

$$\Sigma\left(m\right) = \frac{1}{n-m+1} Y Y^{*} \tag{184}$$

$$= \frac{1}{n-m+1} \sum_{j=0}^{n-m} \begin{bmatrix} \vec{y}\left[j\right] \\ \vec{y}\left[j+1\right] \\ \cdots \\ \vec{y}\left[j+m-1\right] \end{bmatrix} \begin{bmatrix} \overline{\vec{y}\left[j\right]} & \overline{\vec{y}\left[j+1\right]} & \cdots & \overline{\vec{y}\left[j+m-1\right]} \end{bmatrix}. \tag{185}$$

In order to obtain an estimate that is more robust to noise, we can average over more data. If we fix m and increase n, the column space of $\Sigma\left(m\right)$ remains the same, but the averaging process will cancel out the noise to some extent. This is the principle underlying the multiple-signal classification (MUSIC) [1, 4].

Algorithm 3.11 (Multiple-signal classification (MUSIC))**.** *The input is the number of frequencies s, the data y, which are assumed to be of the form* (180) *and the value of the parameter $m \geq s$.*

1. *Build the empirical covariance matrix $\Sigma\left(m\right)$ defined in* (184)*.*
2. *Compute the eigendecomposition of $\Sigma\left(m\right)$ to select the subspace $\mathcal{N}$ corresponding to the $m-s$ smallest singular values.*
3. *Output an estimate of s estimated frequencies by locating the s highest peaks of the pseudospectrum*

$$P_{\mathcal{N}}(u) = \log \frac{1}{\left|\mathcal{P}_{\mathcal{N}}\left(\vec{a}_{0:m-1}\right)\right|^{2}}, \tag{186}$$

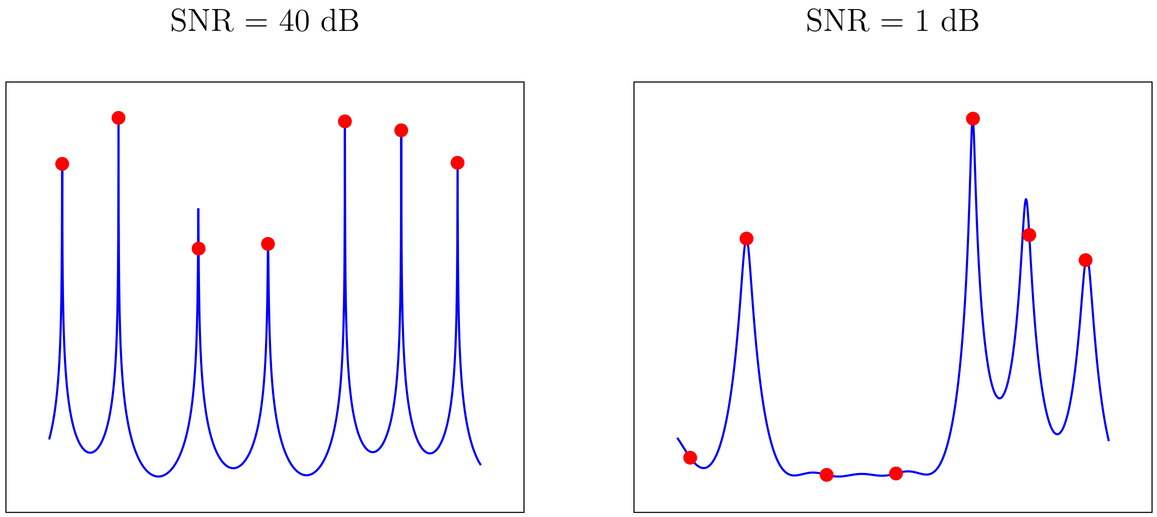


Figure 36: Pseudospectrum constructed by the MUSIC algorithm with $n = 81$ and $m = 30$ for the same signal used in Figure 34 and different noise levels. The true frequencies are marked with red circles.

By Lemma 3.10, in the absence of noise MUSIC achieves perfect super-resolution. When additive Gaussian noise is present in the data, MUSIC is much more robust than Prony's method. Figure 36 shows the result of applying MUSIC algorithm with $n = 81$ and $m = 30$ to the same data used in Figure 34 and different noise levels. The method is able to super-resolve the frequencies at a noise level of 40 dB. At 1 dB the pseudospectrum does not detect the smaller spikes (the true magnitudes are shown in Figure 34), but the estimate for the rest is still rather accurate.

In order to provide a theoretical justification of why MUSIC is stable we study the method in an asymptotic regime where the two following assumptions on the signal and the noise are met:

- **Assumption 1:** Consider the amplitude and phase of each coefficient in the multisinusoidal signal g:

$$\vec{c}[j] = \alpha_j \exp(i\phi_j), \qquad 1 \leq j \leq s, \tag{187}$$

  We model the phases $\boldsymbol{\phi_1}, \ldots, \boldsymbol{\phi_s}$ as independent random variables that are uniformly distributed in $[0, 2\pi]$. The amplitudes $\alpha_1, \ldots, \alpha_s$ can be arbitrary and deterministic. This implies that $\mathrm{E}(\vec{\mathbf{c}}[j]) = 0$ and that the covariance matrix equals

$$\mathrm{E}[\vec{\mathbf{c}}\,\vec{\mathbf{c}}^*] = S_{\vec{c}} := \begin{bmatrix} \alpha_1^2 & 0 & \cdots & 0 \\ 0 & \alpha_2^2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \alpha_s^2 \end{bmatrix}. \tag{188}$$

- **Assumption 2:** The noise in the measurements (180) is Gaussian with zero mean and covariance matrix $\sigma^2 I$, which is also independent from the signal.

In summary, we model the data as an m-dimensional random vector of the form

$$\vec{\mathbf{y}} = A_{0:m-1}\,\vec{\mathbf{c}} + \vec{\mathbf{z}}, \tag{189}$$

where $\vec{\mathbf{z}}$ is Gaussian with mean zero and covariance matrix $\sigma^2 I$. The following theorem, proved in Section 4.2 of the appendix, derives the covariance matrix of $\vec{\mathbf{y}}$.

Theorem 3.12. *Let $\vec{\mathbf{y}}$ be an m dimensional vector of data satisfying Assumptions 1 and 2. The SVD of the covariance matrix of $\vec{\mathbf{y}}$ is equal to*

$$\mathrm{E}\left[\vec{\mathbf{y}}\vec{\mathbf{y}}^*\right] = \begin{bmatrix} U_{\mathcal{S}} & U_{\mathcal{N}} \end{bmatrix} \begin{bmatrix} S + \sigma^2 I_s & 0 \\ 0 & \sigma^2 I_{n-s} \end{bmatrix} \begin{bmatrix} U_{\mathcal{S}}^* \\ U_{\mathcal{N}}^* \end{bmatrix}, \tag{190}$$

where S is a diagonal matrix containing the singular values of The singular vectors are divided into two unitary matrices.

- *$U_{\mathcal{S}} \in \mathbb{C}^{m \times s}$ contains an orthonormal basis of the* signal subspace *which corresponds to the span of $\vec{a}_{0:m-1}(f_1), \ldots, \vec{a}_{0:m-1}(f_s)$.*
- *$U_{\mathcal{N}} \in \mathbb{C}^{m \times (m-s)}$ is a unitary matrix spanning the* noise subspace*, which is the orthogonal complement of the signal subspace.*

This theorem provides a rather intuitive interpretation of the MUSIC algorithm. The SVD of the covariance matrix of the data allows to estimate a signal subspace and a noise subspace. As a result, the term *subspace methods* is often used to describe MUSIC and related algorithms. Computing the pseudospectrum from these subspaces allows us to locate the support of the signal.

In practice, we approximate the covariance matrix using the empirical covariance matrix $\Sigma(m)$ defined in (184). Asymptotically, if we fix s and m and let $n \to \infty$, $\Sigma(m)$ converges to the true covariance matrix (see Section 4.9.1 in [5]). However that this does *not* necessarily imply that MUSIC will allow to find the support! To ensure that we can actually identify the noise subspace correctly, the singular values in the matrix S must all be large with respect to the variance of the noise σ^2. In the case of signals that have a small separation between the frequencies (with respect to n), some of these singular values may be small due to the correlation between $\vec{a}_{0:m-1}(f_1), \ldots, \vec{a}_{0:m-1}(f_s)$.

Figure 37 compares the performance of MUSIC for different noise levels and different values of the parameter m. On the left column, we see the decay of the singular values of the empirical covariance matrix. At high signal-to-noise ratios (SNR) there is a clear transition between the singular values corresponding to the signal subspace (in this case $s = 7$) and the noise subspace, but this is no longer necessarily the case when the noise is increased. On the right column, we see the performance of the algorithm for different values of the SNR and the parameter m. At relatively high SNRs MUSIC is an effective algorithm as long as the assumptions on the signal (random phases), noise (Gaussian) and measurement model (equispaced time samples) are satisfied. In Figure 38 we show the result of running the algorithm for the wrong value of the parameter s. If the value is not too different to s and the SNR not too low, the method is still capable of approximately locating the frequencies.

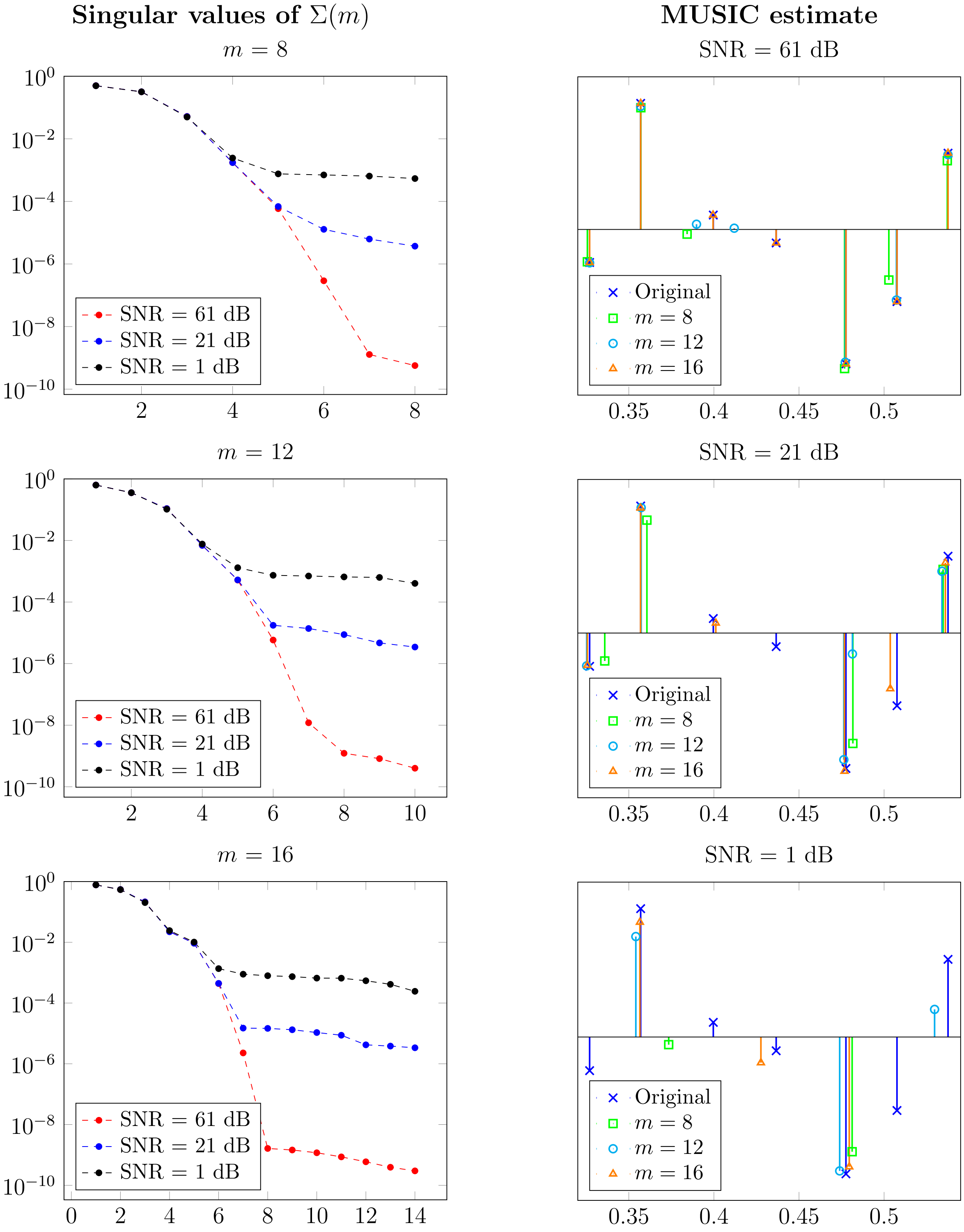


Figure 37: Singular values of the empirical covariance matrix $\Sigma(m)$ used by MUSIC (left) and corresponding estimates (right) for different values of the parameter m and of the SNR. The cardinality of the true support is $s = 7$.

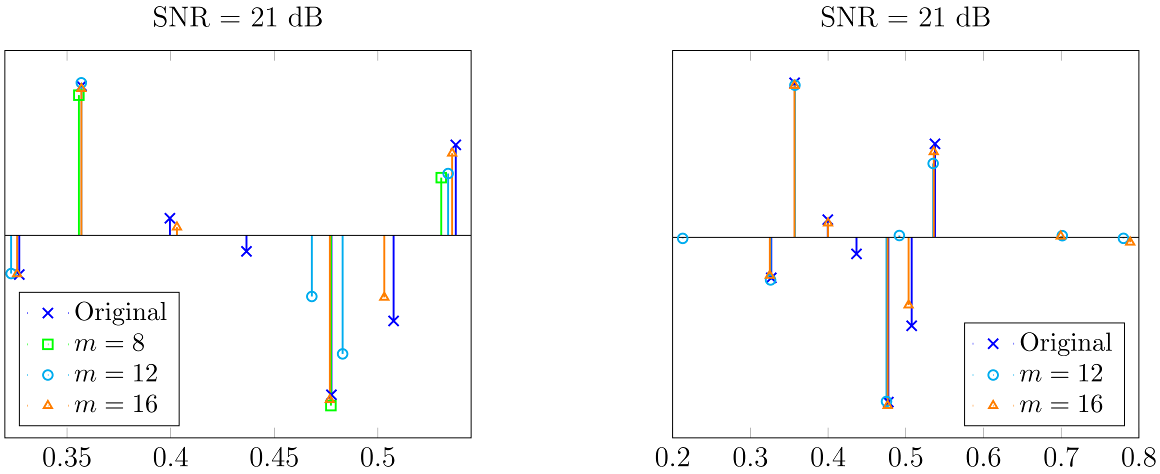


Figure 38: Line-spectra estimates obtained by Root MUSIC when the estimated number of sources is equal to $s-1$ (left) and $s+1$ (right) for the same data as in Figure 37.

4 Proofs

4.1 Proof of Lemma 3.9

Let us define

$$Z := \begin{bmatrix} 1 & 1 & \cdots & 1 \\ z_1 & z_2 & \cdots & z_s \\ z_1^2 & z_2^2 & \cdots & z_s^2 \\ & & \cdots & \\ z_1^{s-1} & z_2^{s-1} & \cdots & z_s^{s-1} \end{bmatrix}. \tag{191}$$

The determinant of the first s rows of our matrix of interest is equal to

$$|Z| \prod_{1\leq i\leq s} z_i = \prod_{1\leq j<k\leq s} (z_j - z_k) \prod_{1\leq i\leq s} z_i \neq 0 \tag{192}$$

This implies that the first s rows are linearly independent and consequently that the whole matrix is full rank.

4.2 Proof of Theorem 3.12

Due to the assumptions,

$$\begin{aligned} \mathrm{E}\left[\vec{\mathbf{y}}\vec{\mathbf{y}}^*\right] &= \mathrm{E}\left[A_{0:m-1}\vec{\mathbf{c}}\vec{\mathbf{c}}^* A_{0:m-1}^* + A_{0:m-1}\vec{\mathbf{c}}\vec{\mathbf{z}}^* + \vec{\mathbf{z}}\vec{\mathbf{c}}^* A_{0:m-1}^* + \vec{\mathbf{z}}\vec{\mathbf{z}}^*\right] & (193)\\ &= A_{0:m-1}\mathrm{E}\left[\vec{\mathbf{c}}\vec{\mathbf{c}}^*\right] A_{0:m-1}^* + A_{0:m-1}\mathrm{E}\left[\vec{\mathbf{c}}\right]\mathrm{E}\left[\vec{\mathbf{z}}^*\right] + \mathrm{E}\left[\vec{\mathbf{z}}\right]\mathrm{E}\left[\vec{\mathbf{c}}^*\right] A_{0:m-1}^* + \mathrm{E}\left[\vec{\mathbf{z}}\vec{\mathbf{z}}^*\right] & (194)\\ &= A_{0:m-1} S_{\vec{c}} A_{0:m-1}^* + \sigma^2 I. & (195) \end{aligned}$$

The matrix $A_{0:m-1} S_{\vec{c}} A^*_{0:m-1}$ is symmetric and has rank s. It therefore has a singular value decomposition of the form

$$A_{0:m-1} S_{\vec{c}} A^*_{0:m-1} = \begin{bmatrix} U_{\mathcal{S}} & U_{\mathcal{N}} \end{bmatrix} \begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U^*_{\mathcal{S}} \\ U^*_{\mathcal{N}} \end{bmatrix}, \tag{196}$$

where $U_{\mathcal{S}}$ and $U_{\mathcal{N}}$ are as defined in the statement of the theorem.

To complete the proof, we decompose the identity matrix using $U_{\mathcal{S}}$ and $U_{\mathcal{N}}$ to obtain

$$\mathrm{E}\left[yy^*\right] = A_{0:m-1} S_{\vec{c}} A^*_{0:m-1} + \sigma^2 I \tag{197}$$

$$= \begin{bmatrix} U_{\mathcal{S}} & U_{\mathcal{N}} \end{bmatrix} \begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U^*_{\mathcal{S}} \\ U^*_{\mathcal{N}} \end{bmatrix} + \begin{bmatrix} U_{\mathcal{S}} & U_{\mathcal{N}} \end{bmatrix} \begin{bmatrix} \sigma^2 I_s & 0 \\ 0 & \sigma^2 I_{n-s} \end{bmatrix} \begin{bmatrix} U^*_{\mathcal{S}} \\ U^*_{\mathcal{N}} \end{bmatrix}. \tag{198}$$

References

We recommend the excellent notes [3] for more background on frequency representations. For more information on spectral super-resolution using the periodogram and subspace methods we refer the reader to [5] and references therein.

[1] G. Bienvenu. Influence of the spatial coherence of the background noise on high resolution passive methods. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, volume 4, pages 306 – 309, 1979.

[2] F. Harris. On the use of windows for harmonic analysis with the discrete Fourier transform. *Proceedings of the IEEE*, 66(1):51 – 83, 1978.

[3] B. Osgood. The Fourier transform and its applications. *Lecture notes for EE261*, 2009.

[4] R. Schmidt. Multiple emitter location and signal parameter estimation. *IEEE Transactions on Antennas and Propagation*, 34(3):276 – 280, 1986.

[5] P. Stoica and R. L. Moses. *Spectral Analysis of Signals*. Prentice Hall, 2005.