Sparse Fourier Transform (lecture 2)

Michael Kapralov[1]

[1] IBM Watson → EPFL

St. Petersburg CS Club
November 2015

Given $x \in \mathbb{C}^n$, compute the Discrete Fourier Transform of x:

$$\widehat{x}_f = \frac{1}{n} \sum_{j \in [n]} x_j \omega^{-f \cdot j},$$

where $\omega = e^{2\pi i/n}$ is the n-th root of unity.

Given $x \in \mathbb{C}^n$, compute the Discrete Fourier Transform of x:

$$\widehat{x}_f = \frac{1}{n} \sum_{j \in [n]} x_j \omega^{-f \cdot j},$$

where $\omega = e^{2\pi i/n}$ is the n-th root of unity.

Goal: find the top k coefficients of $\widehat{x}$ approximately

In last lecture:

- 1-sparse noiseless case: two-point sampling

Given $x \in \mathbb{C}^n$, compute the Discrete Fourier Transform of x:

$$\widehat{x}_f = \frac{1}{n} \sum_{j \in [n]} x_j \omega^{-f \cdot j},$$

where $\omega = e^{2\pi i/n}$ is the n-th root of unity.

Goal: find the top k coefficients of $\widehat{x}$ approximately

In last lecture:

- 1-sparse noiseless case: two-point sampling
- 1-sparse noisy case: $O(\log n \log \log n)$ time and samples

Given $x \in \mathbb{C}^n$, compute the Discrete Fourier Transform of x:

$$\widehat{x}_f = \frac{1}{n} \sum_{j \in [n]} x_j \omega^{-f \cdot j},$$

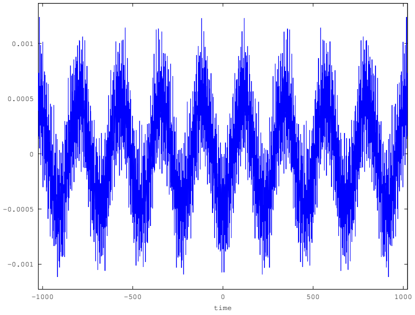
where $\omega = e^{2\pi i/n}$ is the n-th root of unity.

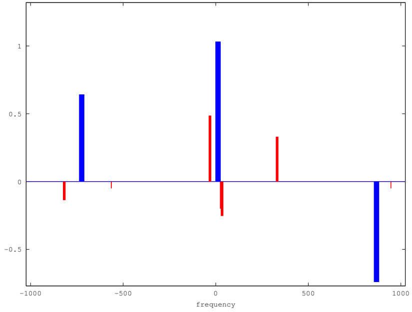
Goal: find the top k coefficients of $\widehat{x}$ approximately

In last lecture:

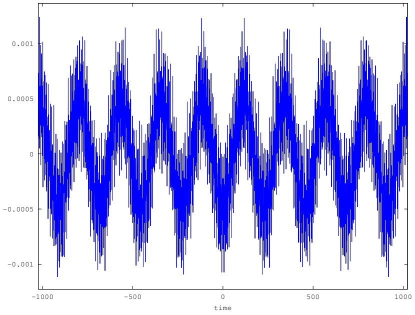
- 1-sparse noiseless case: two-point sampling
- 1-sparse noisy case: $O(\log n \log\log n)$ time and samples
- reduction from k-sparse to 1-sparse case, via filtering

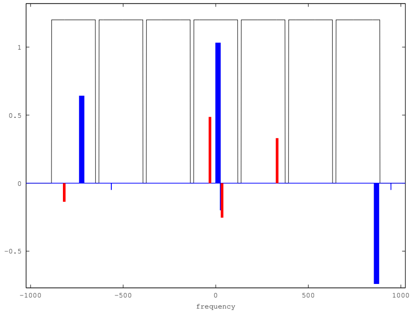
Partition frequency domain into $B \approx k$ buckets



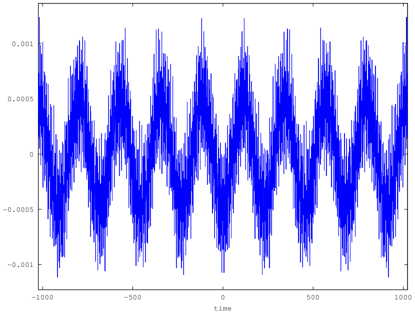


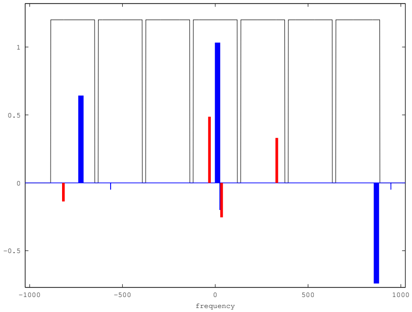
Partition frequency domain into $B \approx k$ buckets





Partition frequency domain into $B \approx k$ buckets



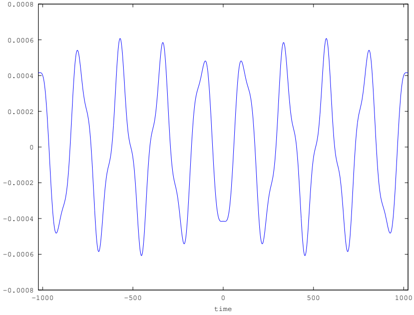


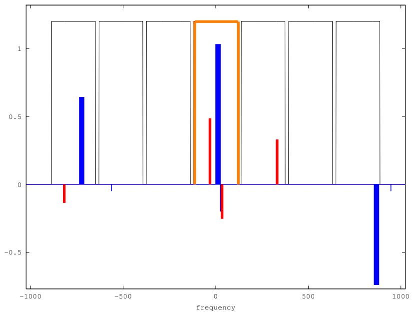
For each $j = 0, \ldots, B-1$ let

$$\widehat{u}^j_f = \begin{cases} \widehat{x}_f, & \text{if } f \in j\text{-th bucket} \\ 0 & \text{o.w.} \end{cases}$$

Restricted to a bucket, signal is likely approximately 1-sparse!

Partition frequency domain into $B \approx k$ buckets



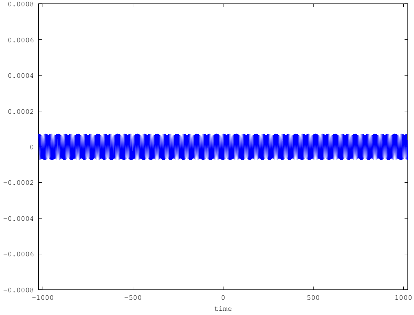


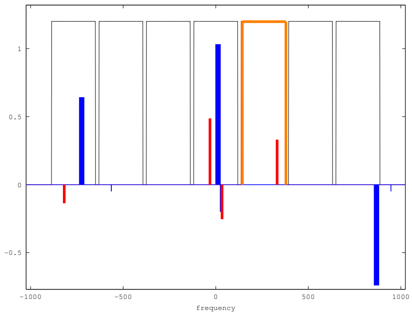
For each $j = 0, \ldots, B-1$ let

$$\widehat{u}^j_f = \begin{cases} \widehat{x}_f, & \text{if } f \in j\text{-th bucket} \\ 0 & \text{o.w.} \end{cases}$$

Restricted to a bucket, signal is likely approximately 1-sparse!

Partition frequency domain into $B \approx k$ buckets



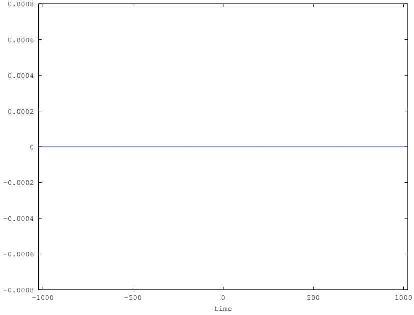


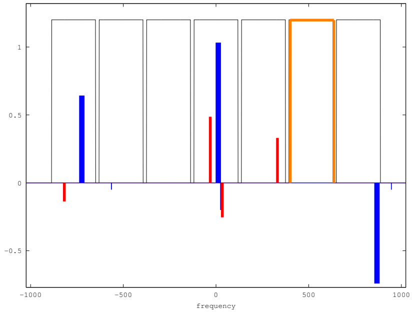
For each $j = 0, \ldots, B-1$ let

$$\widehat{u}^j_f = \begin{cases} \widehat{x}_f, & \text{if } f \in j\text{-th bucket} \\ 0 & \text{o.w.} \end{cases}$$

Restricted to a bucket, signal is likely approximately 1-sparse!

Partition frequency domain into $B \approx k$ buckets



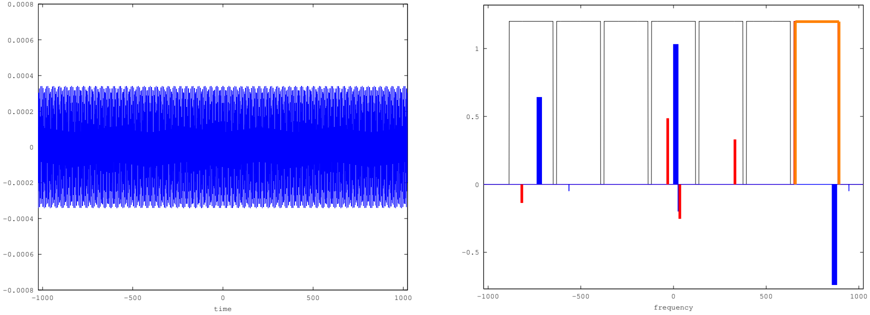


For each $j = 0, \dots, B-1$ let

$$\widehat{u}^j_f = \begin{cases} \widehat{x}_f, & \text{if } f \in j\text{-th bucket} \\ 0 & \text{o.w.} \end{cases}$$

Restricted to a bucket, signal is likely approximately 1-sparse!

Partition frequency domain into $B \approx k$ buckets

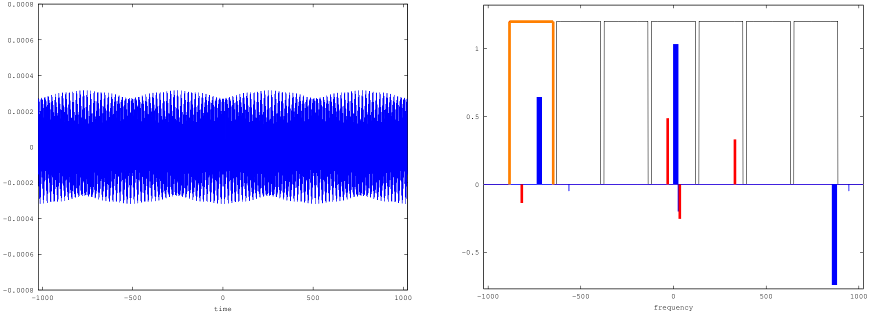


For each $j = 0, \ldots, B-1$ let

$$\widehat{u}^j_f = \begin{cases} \widehat{x}_f, & \text{if } f \in j\text{-th bucket} \\ 0 & \text{o.w.} \end{cases}$$

Restricted to a bucket, signal is likely approximately 1-sparse!

Partition frequency domain into $B \approx k$ buckets

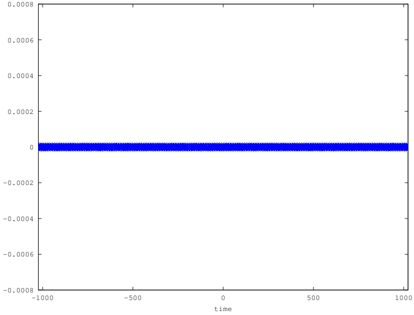


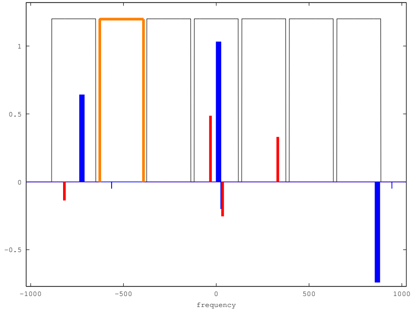
For each $j = 0, \ldots, B-1$ let

$$\hat{u}^j_f = \begin{cases} \hat{x}_f, & \text{if } f \in j\text{-th bucket} \\ 0 & \text{o.w.} \end{cases}$$

Restricted to a bucket, signal is likely approximately 1-sparse!

Partition frequency domain into $B \approx k$ buckets



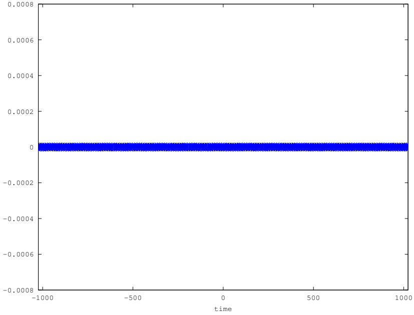


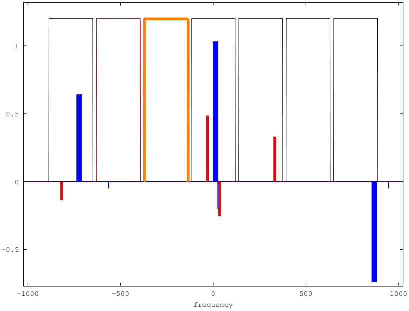
For each $j = 0, \ldots, B-1$ let

$$\widehat{u}_f^j = \begin{cases} \widehat{x}_f, & \text{if } f \in j\text{-th bucket} \\ 0 & \text{o.w.} \end{cases}$$

Restricted to a bucket, signal is likely approximately 1-sparse!

Partition frequency domain into $B \approx k$ buckets





For each $j = 0, \ldots, B-1$ let

$$\widehat{u}^j_f = \begin{cases} \widehat{x}_f, & \text{if } f \in j\text{-th bucket} \\ 0 & \text{o.w.} \end{cases}$$

Restricted to a bucket, signal is likely approximately 1-sparse!

We want time domain access to u^0: for any $a = 0, \ldots, n-1$, compute

$$u^0_a = \sum_{-\frac{n}{2B} \le f \le \frac{n}{2B}} \widehat{x}_f \cdot \omega^{f \cdot a}.$$

Let

$$\widehat{G}_f = \begin{cases} 1, & \text{if } f \in \left[-\frac{n}{2B} : \frac{n}{2B}\right] \\ 0 & \text{o.w.} \end{cases}$$

Then

$$u^0_a = (\widehat{x_{\cdot + a}} * \widehat{G})(0)$$

We want time domain access to u^0: for any $a = 0, \ldots, n-1$, compute

$$u^0_a = \sum_{-\frac{n}{2B} \leq f \leq \frac{n}{2B}} \widehat{x}_f \cdot \omega^{f \cdot a}.$$

Let

$$\widehat{G}_f = \begin{cases} 1, & \text{if } f \in [-\frac{n}{2B} : \frac{n}{2B}] \\ 0 & \text{o.w.} \end{cases}$$

Then

$$u^0_a = (\widehat{x_{\cdot + a}} * \widehat{G})(0)$$

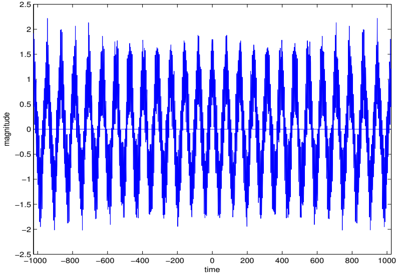
For any $j = 0, \ldots, B-1$

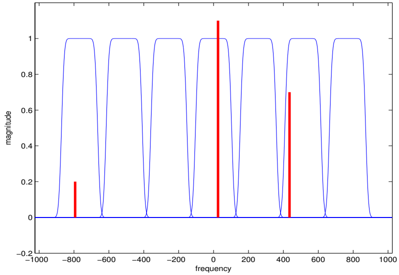
$$u^j_a = (\widehat{x_{\cdot + a}} * \widehat{G})(j \cdot \frac{n}{B})$$

Reducing k-sparse recovery to 1-sparse recovery

For any $j = 0, \ldots, B-1$

$$u_a^j = (\widehat{x_{\cdot+a}} * \widehat{G})(j \cdot \frac{n}{B})$$

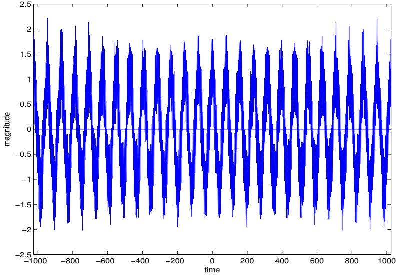


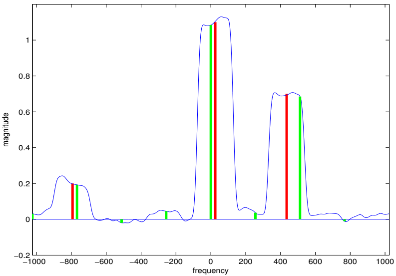


Reducing k-sparse recovery to 1-sparse recovery

For any $j = 0, \ldots, B-1$

$$u_a^j = (\widehat{x_{\cdot+a}} * \widehat{G})(j \cdot \frac{n}{B})$$

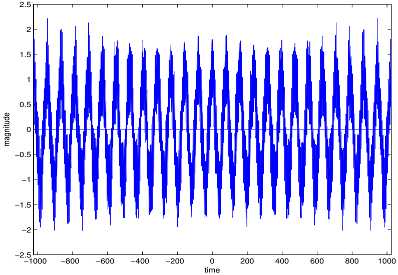


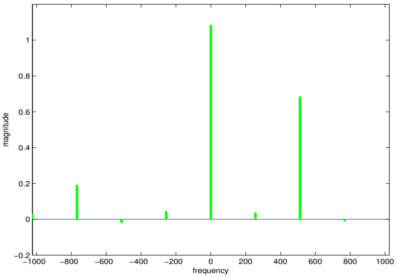


Reducing k-sparse recovery to 1-sparse recovery

For any $j = 0, \ldots, B-1$

$$u_{a}^{j} = (\widehat{x_{\cdot+a}} * \widehat{G})(j \cdot \frac{n}{B})$$





Need to evaluate

$$(\widehat{x}_{\cdot+a} * \widehat{G})\left(j \cdot \frac{n}{B}\right)$$

for $j = 0, \ldots, B-1$.

We have access to x, not $\widehat{x}$...

Need to evaluate

$$(\widehat{x}_{\cdot+a} * \widehat{G})\left(j \cdot \frac{n}{B}\right)$$

for $j = 0, \dots, B-1$.

We have access to x, not $\widehat{x}$...

By the convolution identity

$$\widehat{x}_{\cdot+a} * \widehat{G} = \widehat{(x_{\cdot+a} \cdot G)}$$

Need to evaluate

$$(\widehat{x}_{\cdot+a} * \widehat{G})\left(j \cdot \frac{n}{B}\right)$$

for $j = 0, \ldots, B-1$.

We have access to x, not $\widehat{x}$...

By the convolution identity

$$\widehat{x}_{\cdot+a} * \widehat{G} = (\widehat{x_{\cdot+a} \cdot G})$$

Suffices to compute

$$\widehat{x_{\cdot+a} \cdot G}_{j \cdot \frac{n}{B}}, j = 0, \ldots, B-1$$

Suffices to compute

$$\widehat{x_{\cdot+a} \cdot G}_{j \cdot \frac{n}{B}}, j = 0, \ldots, B-1$$

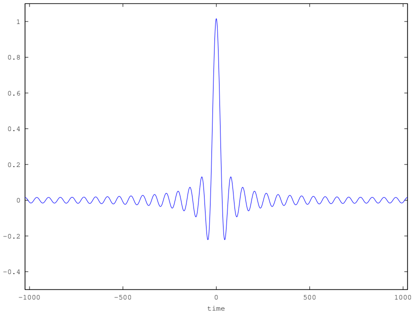
Suffices to compute

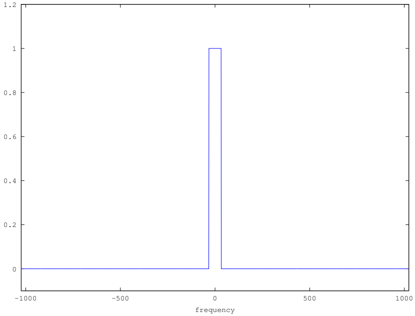
$$\widehat{x \cdot G}_{j \cdot \frac{n}{B}}, j = 0, \ldots, B-1$$

Suffices to compute

$$\widehat{x \cdot G}_{j \cdot \frac{n}{B}}, j = 0, \ldots, B-1$$

Sample complexity? Runtime?

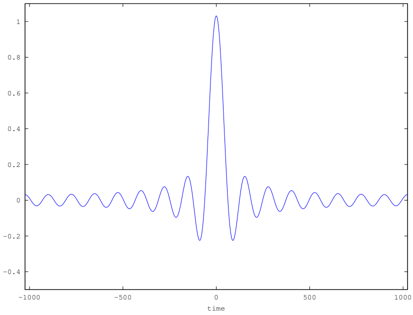


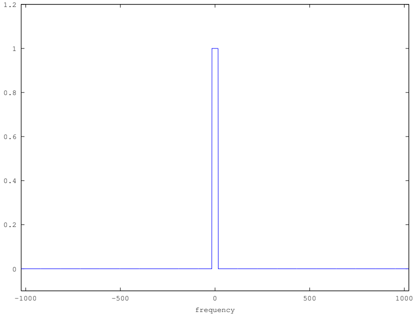


Suffices to compute

$$\widehat{x \cdot G}_{j \cdot \frac{n}{B}}, j = 0, \ldots, B-1$$

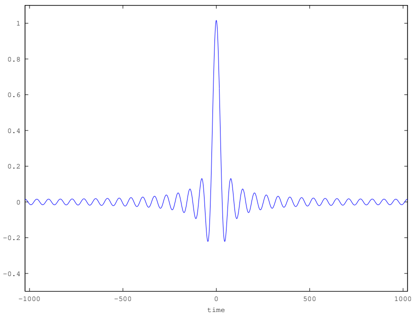
Sample complexity? Runtime?

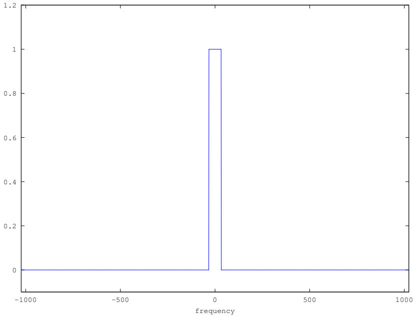




To sample all signals $u^j, j = 0, \ldots, B-1$ in time domain, it suffices to compute

$$\widehat{x \cdot G}_{j \cdot \frac{n}{B}}, j = 0, \ldots, B-1$$





Computing $x \cdot G$ takes $\mathrm{supp}(G)$ samples.

Design G with $\mathrm{supp}(G) \approx k$ that approximates rectangular filter?

In this lecture:

- permuting frequencies
- filter construction

1. Pseudorandom spectrum permutations
2. Filter construction

1. **Pseudorandom spectrum permutations**
2. Filter construction

Pseudorandom spectrum permutations

Permutation in time domain plus phase shift $\Longrightarrow$ permutation in frequency domain

Pseudorandom spectrum permutations

Permutation in time domain plus phase shift $\Longrightarrow$ permutation in frequency domain

Claim

Let $\sigma, b \in [n]$*,* σ *invertible modulo n. Let* $y_j = x_{\sigma j}\omega^{-jb}$*. Then*

$$\widehat{y}_f = \widehat{x}_{\sigma^{-1}(f+b)}.$$

(proof on next slide; a close relative of time shift theorem)

Pseudorandom spectrum permutations

Permutation in time domain plus phase shift $\Longrightarrow$ permutation in frequency domain

Claim

Let $\sigma, b \in [n]$*,* σ *invertible modulo n. Let* $y_j = x_{\sigma j}\omega^{-jb}$*. Then*

$$\widehat{y}_f = \widehat{x}_{\sigma^{-1}(f+b)}.$$

(proof on next slide; a close relative of time shift theorem)

Pseudorandom permutation:

- select b uniformly at random from $[n]$
- select σ uniformly at random from $\{1,3,5,\ldots,n-1\}$ (invertible numbers modulo n)

Pseudorandom spectrum permutations

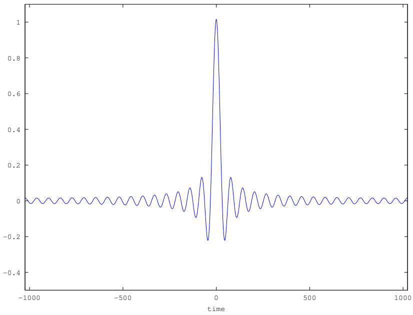
Claim

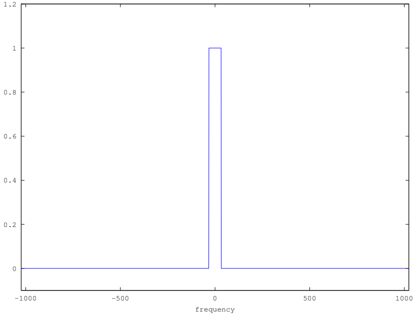
Let $y_j = x_{\sigma j}\omega^{-jb}$*. Then* $\widehat{y}_f = \widehat{x}_{\sigma^{-1}(f+b)}$.

Proof.

$$
\begin{aligned}
\widehat{y}_f &= \frac{1}{n}\sum_{j\in[n]} y_j \omega^{-f\cdot j} \\
&= \frac{1}{n}\sum_{j\in[n]} x_{\sigma j} \omega^{-(f+b)\cdot j} \\
&= \frac{1}{n}\sum_{i\in[n]} x_i \omega^{-(f+b)\cdot \sigma^{-1} i} \quad \text{(change of variables } i = \sigma j\text{)} \\
&= \frac{1}{n}\sum_{i\in[n]} x_i \omega^{-\sigma^{-1}(f+b)\cdot i} \\
&= \widehat{x}_{\sigma^{-1}(f+b)}
\end{aligned}
$$

□

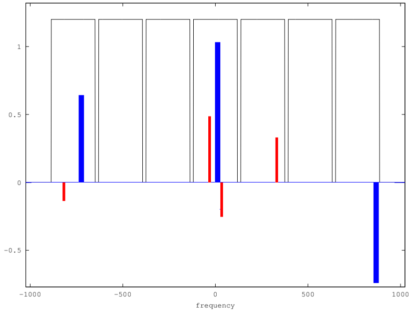




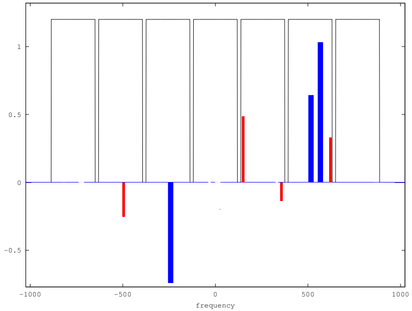
Design G with $\text{supp}(G) \approx k$ that approximates rectangular filter?

Our filter $\widehat{G}$ will approximate the boxcar. Bound collision probability now.

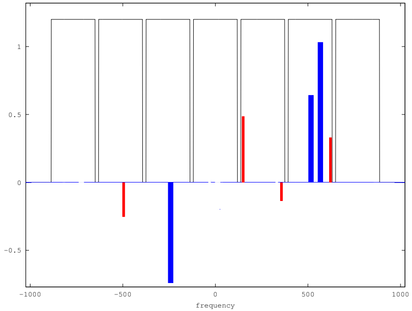
Partition frequency domain into buckets, permute spectrum



Partition frequency domain into buckets, permute spectrum

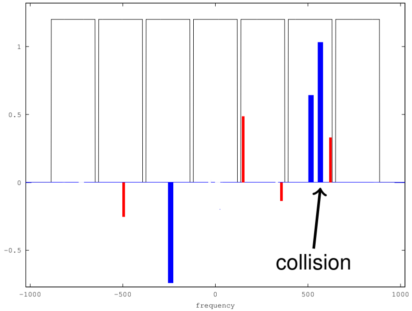


Partition frequency domain into buckets, permute spectrum



Frequency i collides with frequency j only if $|\sigma i - \sigma j| \leq \frac{n}{B}$.

Partition frequency domain into buckets, permute spectrum



Frequency *i* collides with frequency *j* only if $|\sigma i - \sigma j| \leq \frac{n}{B}$.

Collision probability

Lemma

Let σ be a uniformly random odd number in $1, 2, \ldots, n$. Then for any $i, j \in [n], i \neq j$ one has

$$\mathbf{Pr}_{\sigma}\left[|\sigma \cdot i - \sigma j| \leq \frac{n}{B}\right] = O(1/B)$$

Collision probability

Lemma

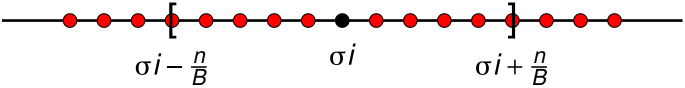
Let σ be a uniformly random odd number in $1, 2, \ldots, n$. Then for any $i, j \in [n], i \neq j$ one has

$$\mathbf{Pr}_{\sigma}\left[|\sigma \cdot i - \sigma j| \leq \frac{n}{B}\right] = O(1/B)$$

Proof.

Let $\Delta := i - j = d2^s$ for some odd d.

The orbit of $\sigma \cdot \Delta$ is $2^s \cdot d'$ for all odd d'.



There are $O(\frac{n}{B2^s})$ values of d' that make $\sigma \cdot \Delta$ fall into $[-\frac{n}{B}, \frac{n}{B}]$, out of $n/2^{s+1}$. □

Collision probability

Lemma

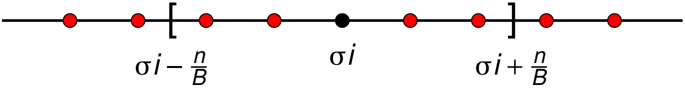
Let σ be a uniformly random odd number in $1, 2, \ldots, n$. Then for any $i, j \in [n], i \neq j$ one has

$$\mathbf{Pr}_\sigma\left[|\sigma \cdot i - \sigma j| \leq \frac{n}{B}\right] = O(1/B)$$

Proof.

Let $\Delta := i - j = d2^s$ for some odd d.

The orbit of $\sigma \cdot \Delta$ is $2^s \cdot d'$ for all odd d'.



There are $O(\frac{n}{B2^s})$ values of d' that make $\sigma \cdot \Delta$ fall into $[-\frac{n}{B}, \frac{n}{B}]$, out of $n/2^{s+1}$. □

Collision probability

Lemma

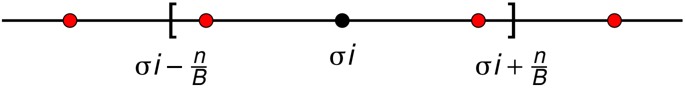
Let σ be a uniformly random odd number in $1,2,\ldots,n$. Then for any $i,j \in [n], i \neq j$ one has

$$\mathbf{Pr}_\sigma\left[|\sigma \cdot i - \sigma j| \leq \frac{n}{B}\right] = O(1/B)$$

Proof.

Let $\Delta := i - j = d2^s$ for some odd d.

The orbit of $\sigma \cdot \Delta$ is $2^s \cdot d'$ for all odd d'.



There are $O(\frac{n}{B2^s})$ values of d' that make $\sigma \cdot \Delta$ fall into $[-\frac{n}{B}, \frac{n}{B}]$, out of $n/2^{s+1}$. □

Collision probability

Lemma

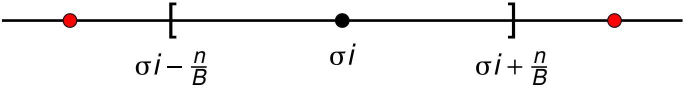
Let σ be a uniformly random odd number in $1,2,\ldots,n$. Then for any $i,j \in [n], i \neq j$ one has

$$\mathbf{Pr}_\sigma \left[|\sigma \cdot i - \sigma j| \leq \frac{n}{B} \right] = O(1/B)$$

Proof.

Let $\Delta := i - j = d2^s$ for some odd d.

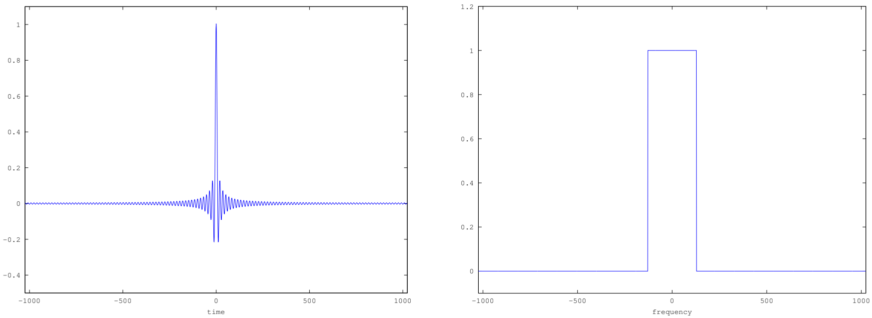
The orbit of $\sigma \cdot \Delta$ is $2^s \cdot d'$ for all odd d'.



There are $O(\frac{n}{B2^s})$ values of d' that make $\sigma \cdot \Delta$ fall into $[-\frac{n}{B}, \frac{n}{B}]$, out of $n/2^{s+1}$. □

1. Pseudorandom spectrum permutations
2. **Filter construction**

Rectangular buckets $\widehat{G}$ have full support in time domain...

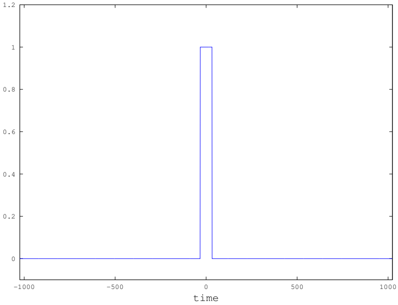


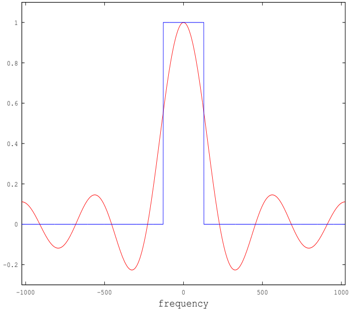
Approximate rectangular filter with a filter G with small support?

Need $\text{supp}(G) \approx k$, so perhaps turn the filter around?

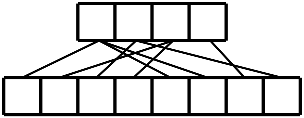
Let

$$G_j := \begin{cases} 1/(B+1) & \text{if } j \in [-B/2, B/2] \\ 0 & \text{o.w.} \end{cases}$$





Have $\text{supp}(G) = B \approx k$, but buckets leak



$\Longrightarrow$

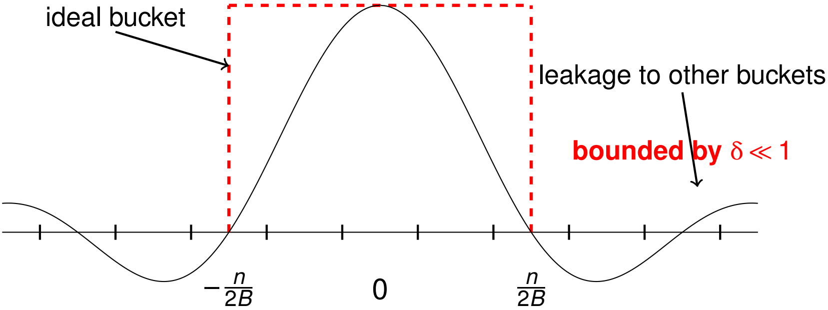
In what follows: reduce leakage at the expense of increasing supp(G)

Window functions

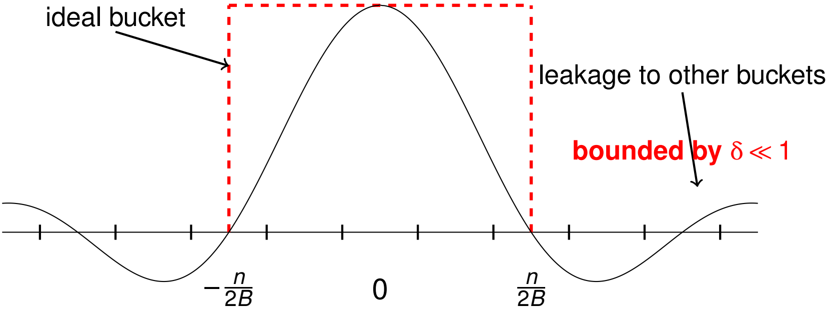
Definition

A symmetric filter G is a (B, δ)-*standard window function* if

1. $\widehat{G}_0 = 1$
2. $\widehat{G}_f \geq 0$
3. $|\widehat{G}_f| \leq \delta$ for $f \notin [-\frac{n}{2B}, \frac{n}{2B}]$



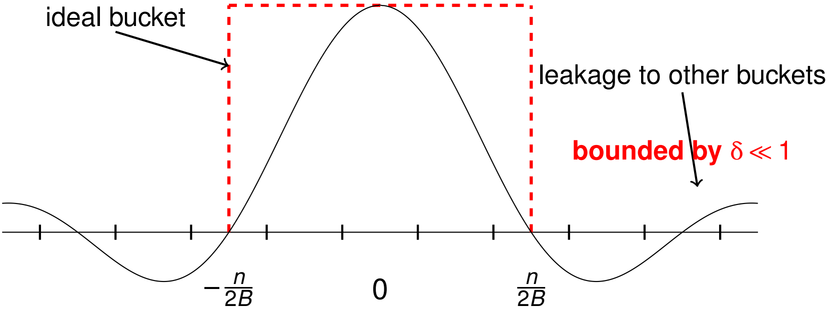
Window functions



Start with the sinc function:

$$\widehat{G}_f := \frac{\sin(\pi(B+1)f/n)}{(B+1)\cdot \pi f/n}$$

Window functions



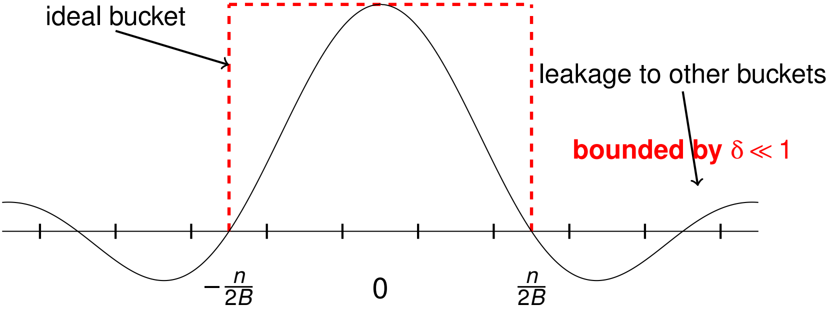
Start with the sinc function:

$$\widehat{G}_f := \frac{\sin(\pi(B+1)f/n)}{(B+1)\cdot\pi f/n}$$

For all $|f| > \frac{n}{2B}$ we have

$$|\widehat{G}_f| \leq \frac{1}{(B+1)\pi f/n} \leq \frac{1}{\pi/2} \leq 2/\pi \leq 0.9$$

Window functions



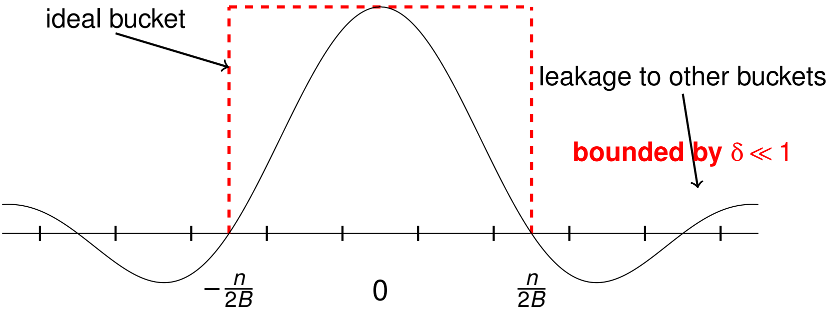
Consider **powers of the sinc function**:

$$\widehat{G}_f^r := \left(\frac{\sin(\pi(B+1)f/n)}{(B+1)\cdot\pi f/n}\right)^r$$

For all $|f| > \frac{n}{2B}$ we have

$$|\widehat{G}_f|^r \leq (0.9)^r$$

Window functions



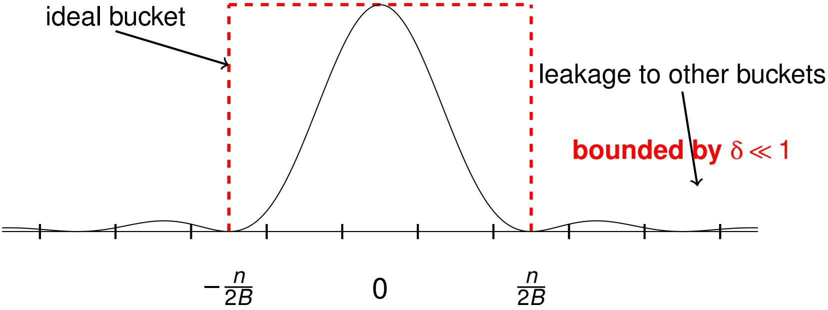
Consider **powers of the sinc function**:

$$\widehat{G}_f^r := \left(\frac{\sin(\pi(B+1)f/n)}{(B+1)\cdot\pi f/n}\right)^r$$

For all $|f| > \frac{n}{2B}$ we have

$$|\widehat{G}_f|^r \leq (0.9)^r$$

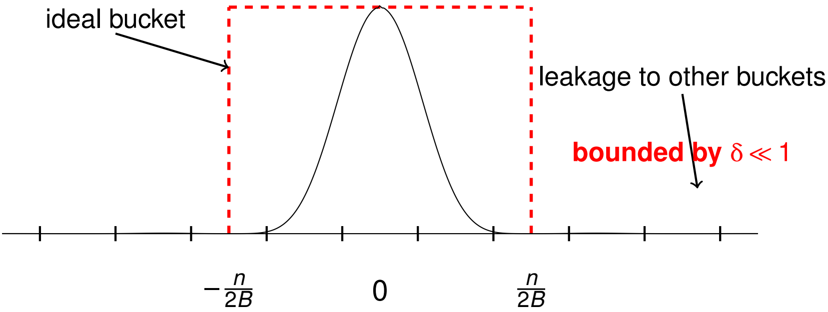
Window functions



Consider **powers of the sinc function**: $\widehat{G}_f^r$
For all $|f| > \frac{n}{2B}$ we have

$$|\widehat{G}_f|^r \leq (0.9)^r$$

Window functions



Consider **powers of the sinc function**: $\widehat{G}_f^r$
For all $|f| > \frac{n}{2B}$ we have

$$|\widehat{G}_f|^r \leq (0.9)^r$$

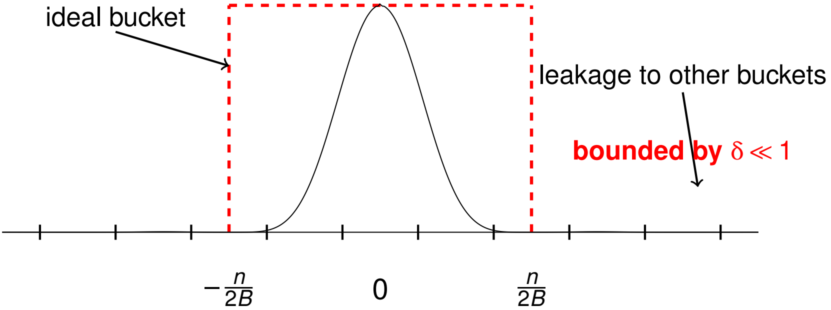
So setting $r = O(\log(1/\delta))$ is sufficient!

Window functions

Definition

A symmetric filter G is a (B, δ)-*standard window function* if

1. $\widehat{G}_0 = 1$
2. $\widehat{G}_f \geq 0$
3. $|\widehat{G}_f| \leq \delta$ for $f \notin [-\frac{n}{2B}, \frac{n}{2B}]$

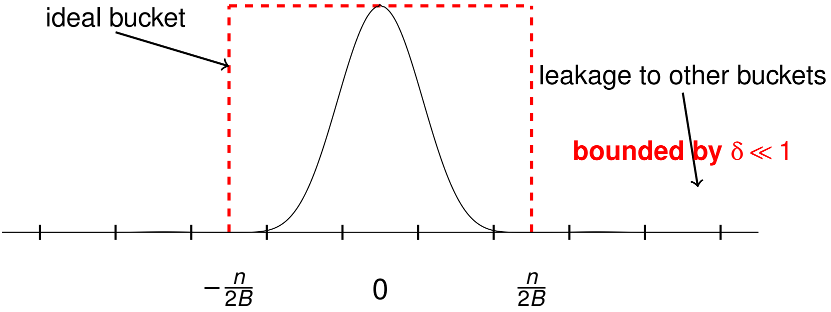


Window functions

Definition

A symmetric filter G is a (B, δ)-*standard window function* if

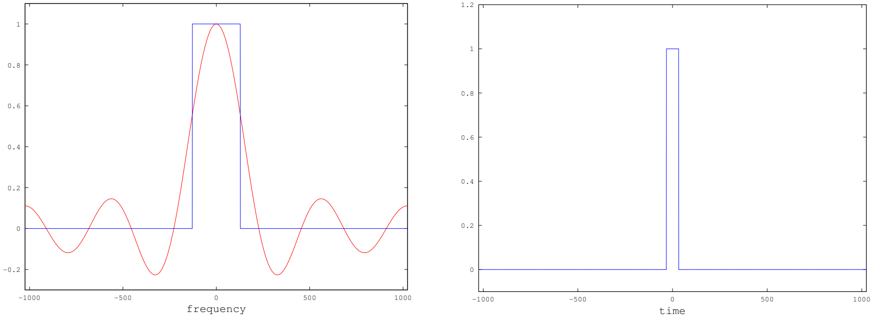
1. $\widehat{G}_0 = 1$
2. $\widehat{G}_f \geq 0$
3. $|\widehat{G}_f| \leq \delta$ for $f \notin [-\frac{n}{2B}, \frac{n}{2B}]$



How large is supp$(G) \subseteq [-T, T]$**?**

Let

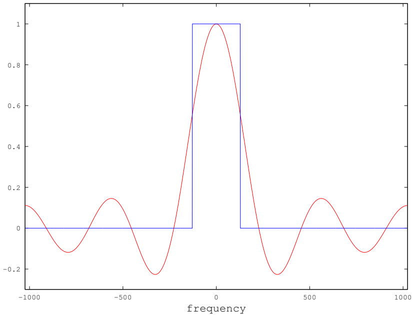
$$G_j := \begin{cases} 1/(B+1) & \text{if } j \in [-B/2, B/2] \\ 0 & \text{o.w.} \end{cases}$$

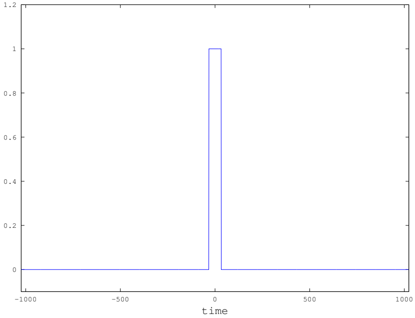


Let $\widehat{G}^r := (\widehat{G}^0)^r$. How large is the support of G^r?

Let

$$G_j := \begin{cases} 1/(B+1) & \text{if } j \in [-B/2, B/2] \\ 0 & \text{o.w.} \end{cases}$$



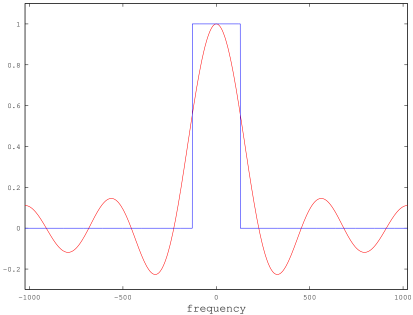


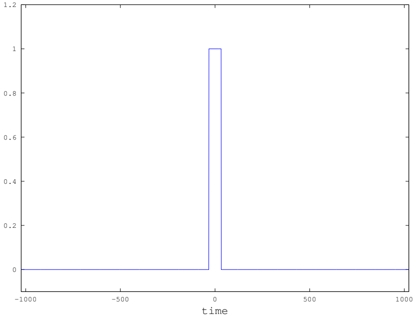
Let $\widehat{G}^r := (\widehat{G}^0)^r$. How large is the support of G^r?

By the convolution identity $G^r = G^0 * G^0 * \ldots * G^0$

Let

$$G_j := \begin{cases} 1/(B+1) & \text{if } j \in [-B/2, B/2] \\ 0 & \text{o.w.} \end{cases}$$





Let $\widehat{G}^r := (\widehat{G}^0)^r$. How large is the support of G^r?

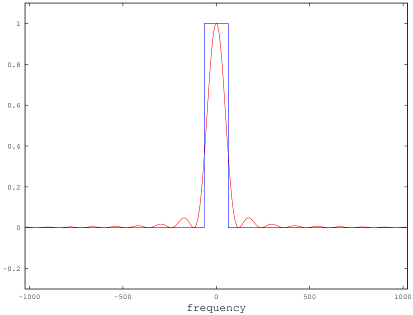
By the convolution identity $G^r = G^0 * G^0 * \ldots * G^0$

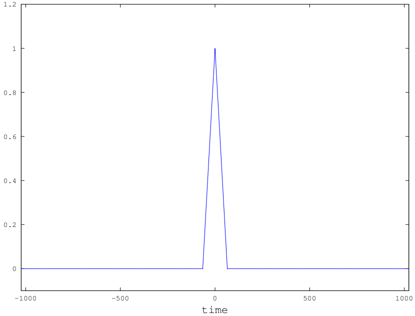
Support of G^0 is in $[-B/2, B/2]$, so

$$\text{supp}(G * \ldots * G) \subseteq [-r \cdot B/2, r \cdot B/2]$$

Let

$$G_j := \begin{cases} 1/(B+1) & \text{if } j \in [-B/2, B/2] \\ 0 & \text{o.w.} \end{cases}$$





Let $\widehat{G}^r := (\widehat{G}^0)^r$. How large is the support of G^r?

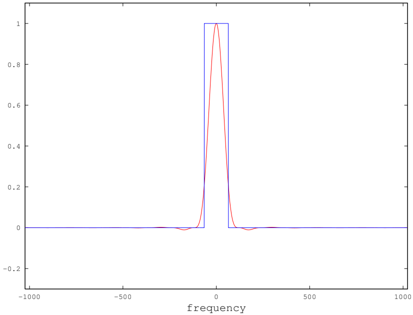
By the convolution identity $G^r = G^0 * G^0 * \ldots * G^0$

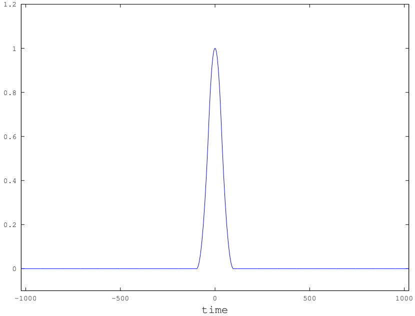
Support of G^0 is in $[-B/2, B/2]$, so

$$\text{supp}(G * \ldots * G) \subseteq [-r \cdot B/2, r \cdot B/2]$$

Let

$$G_j := \begin{cases} 1/(B+1) & \text{if } j \in [-B/2, B/2] \\ 0 & \text{o.w.} \end{cases}$$





Let $\widehat{G}^r := (\widehat{G}^0)^r$. How large is the support of G^r?

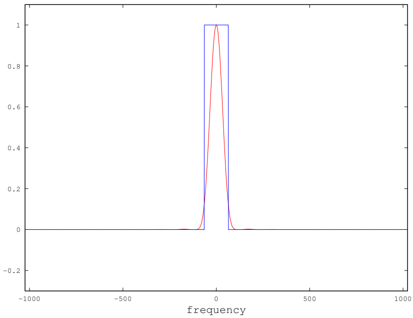
By the convolution identity $G^r = G^0 * G^0 * \ldots * G^0$

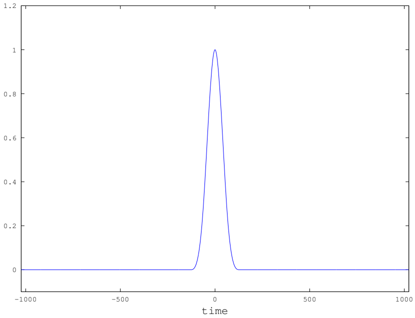
Support of G^0 is in $[-B/2, B/2]$, so

$$\text{supp}(G * \ldots * G) \subseteq [-r \cdot B/2, r \cdot B/2]$$

Let

$$G_j := \begin{cases} 1/(B+1) & \text{if } j \in [-B/2, B/2] \\ 0 & \text{o.w.} \end{cases}$$





Let $\widehat{G}^r := (\widehat{G}^0)^r$. How large is the support of G^r?

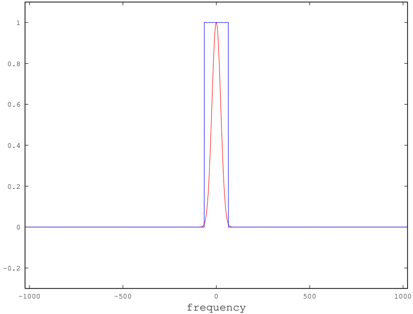
By the convolution identity $G^r = G^0 * G^0 * \ldots * G^0$

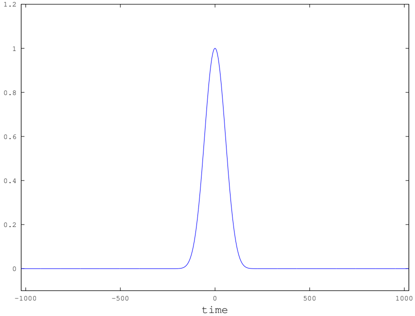
Support of G^0 is in $[-B/2, B/2]$, so

$$\text{supp}(G * \ldots * G) \subseteq [-r \cdot B/2, r \cdot B/2]$$

Let

$$G_j := \begin{cases} 1/(B+1) & \text{if } j \in [-B/2, B/2] \\ 0 & \text{o.w.} \end{cases}$$





Let $\widehat{G}^r := (\widehat{G}^0)^r$. How large is the support of G^r?

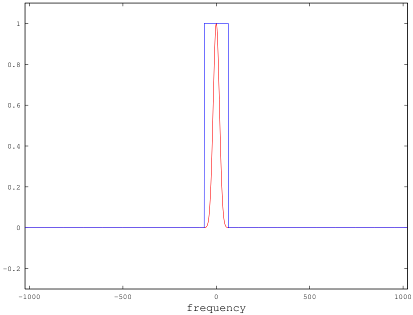
By the convolution identity $G^r = G^0 * G^0 * \ldots * G^0$

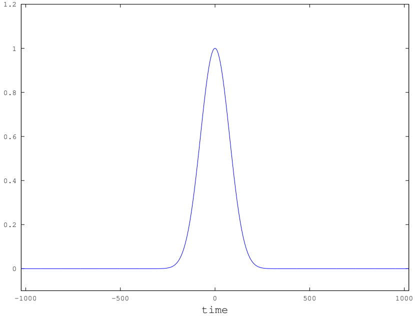
Support of G^0 is in $[-B/2, B/2]$, so

$$\text{supp}(G * \ldots * G) \subseteq [-r \cdot B/2, r \cdot B/2]$$

Let

$$G_j := \begin{cases} 1/(B+1) & \text{if } j \in [-B/2, B/2] \\ 0 & \text{o.w.} \end{cases}$$





Let $\widehat{G}^r := (\widehat{G}^0)^r$. How large is the support of G^r?

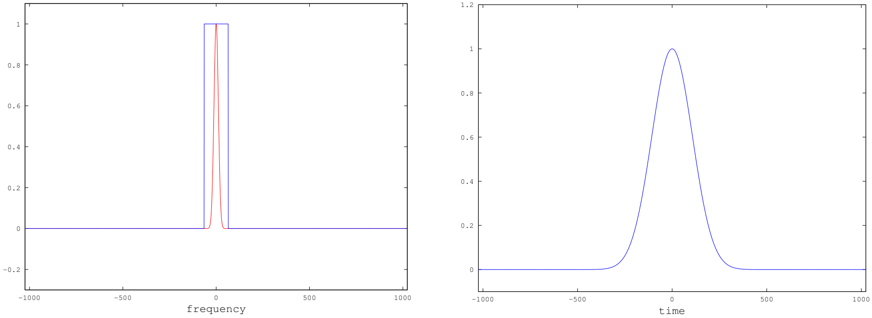
By the convolution identity $G^r = G^0 * G^0 * \ldots * G^0$

Support of G^0 is in $[-B/2, B/2]$, so

$$\text{supp}(G * \ldots * G) \subseteq [-r \cdot B/2, r \cdot B/2]$$

Let

$$G_j := \begin{cases} 1/(B+1) & \text{if } j \in [-B/2, B/2] \\ 0 & \text{o.w.} \end{cases}$$



Let $\widehat{G}^r := (\widehat{G}^0)^r$. How large is the support of G^r?

By the convolution identity $G^r = G^0 * G^0 * \ldots * G^0$

Support of G^0 is in $[-B/2, B/2]$, so

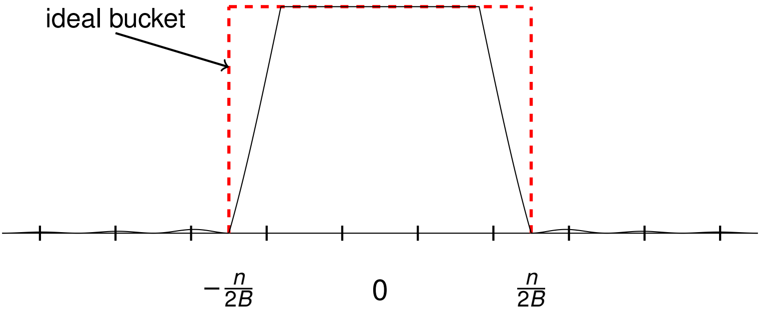
$$\text{supp}(G * \ldots * G) \subseteq [-r \cdot B/2, r \cdot B/2]$$

Flat window function

Definition

A symmetric filter G is a (B, δ, γ)-**flat** *window function* if

1. $\widehat{G}_j \geq 1 - \delta$ for all $j \in [-(1-\gamma)\frac{n}{2B}, (1-\gamma)\frac{n}{2B}]$
2. $\widehat{G}_j \in [0, 1]$ for all j
3. $|\widehat{G}_f| \leq \delta$ for $f \notin [-\frac{n}{2B}, \frac{n}{2B}]$

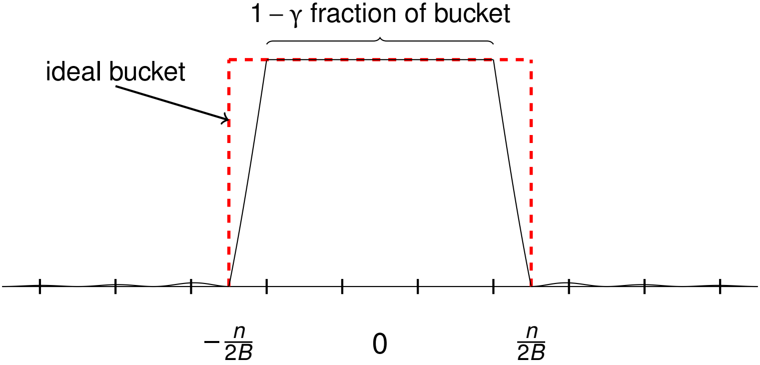


Flat window function

Definition

A symmetric filter G is a (B, δ, γ)-**flat** *window function* if

1. $\widehat{G}_j \geq 1 - \delta$ for all $j \in [-(\mathbf{1} - \gamma)\frac{n}{2B}, (\mathbf{1} - \gamma)\frac{n}{2B}]$
2. $\widehat{G}_j \in [0, 1]$ for all j
3. $|\widehat{G}_f| \leq \delta$ for $f \notin [-\frac{n}{2B}, \frac{n}{2B}]$

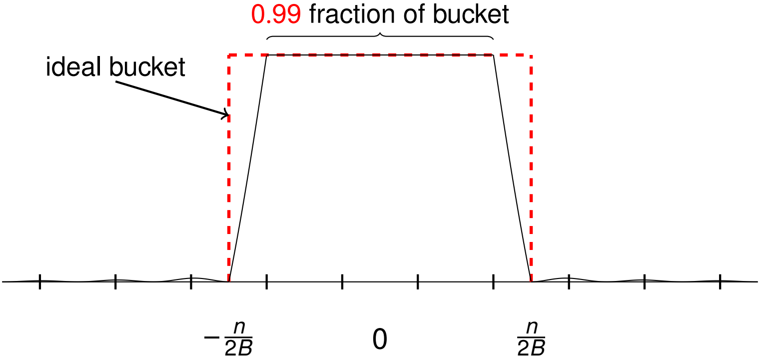


Flat window function

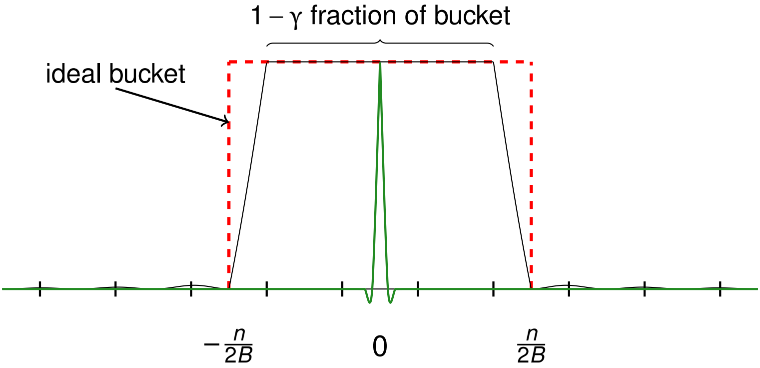
Definition

A symmetric filter G is a (B, δ, γ)-**flat** *window function* if

1. $\widehat{G}_j \geq 1 - \delta$ for all $j \in [-(\mathbf{1} - \gamma)\frac{n}{2B}, (\mathbf{1} - \gamma)\frac{n}{2B}]$
2. $\widehat{G}_j \in [0, 1]$ for all j
3. $|\widehat{G}_f| \leq \delta$ for $f \notin [-\frac{n}{2B}, \frac{n}{2B}]$



Flat window function – construction



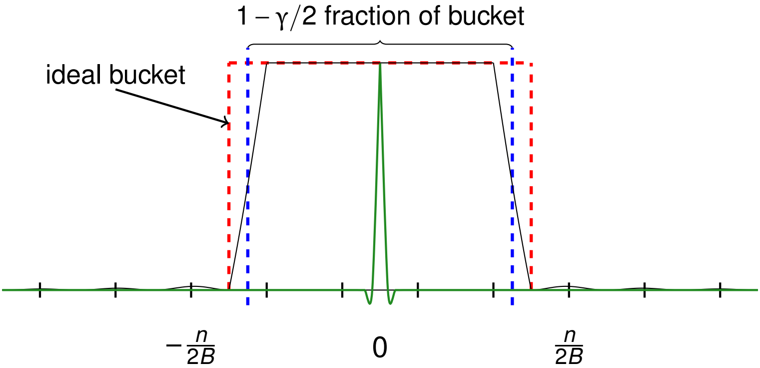
Let H be a $(2B/\gamma, \delta/n)$-standard window function. Note that

$$|\widehat{H}_f| \leq \delta/n$$

for all f outside of

$$\left[-\gamma\frac{n}{4B}, \gamma\frac{n}{4B}\right].$$

Flat window function – construction



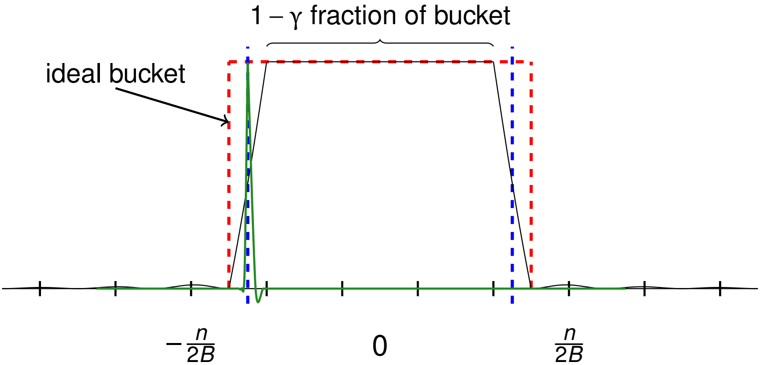
Let H be a $(2B/\gamma, \delta/n)$-standard window function. Note that

$$|\widehat{H}_f| \le \delta/n$$

for all f outside of

$$\left[-\gamma\frac{n}{4B}, \gamma\frac{n}{4B}\right].$$

Flat window function – construction



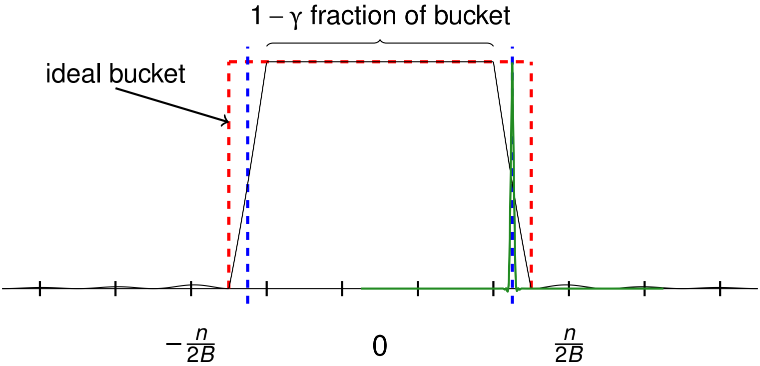
Let H be a $(2B/\gamma, \delta/n)$-standard window function. Note that

$$|\widehat{H}_f| \leq \delta/n$$

for all f outside of

$$\left[-\gamma\frac{n}{4B}, \gamma\frac{n}{4B}\right].$$

Flat window function – construction



Let H be a $(2B/\gamma, \delta/n)$-standard window function. Note that

$$|\widehat{H}_f| \leq \delta/n$$

for all f outside of

$$\left[-\gamma\frac{n}{4B}, \gamma\frac{n}{4B}\right].$$

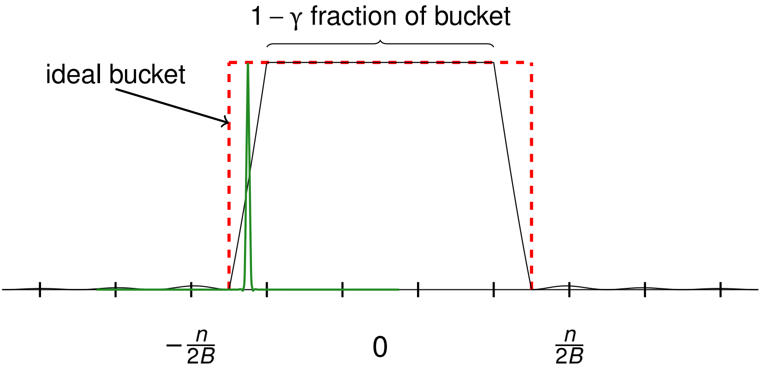
Flat window function – construction

To construct $\widehat{G}$:

1. sum up shifts $\widehat{H}_{\cdot-\Delta}$ over all $\Delta \in [-U, U]$, where

$$U = (1 - \gamma/2)\frac{n}{2B}$$

2. normalize so that $\widehat{G}_0 = 1 \pm \delta$



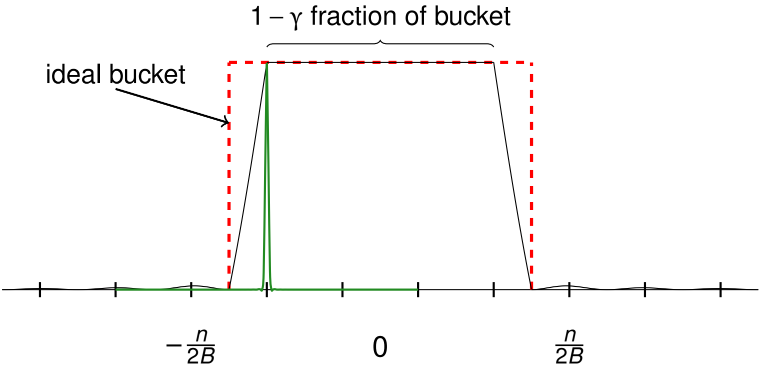
Flat window function – construction

To construct $\widehat{G}$:

1. sum up shifts $\widehat{H}_{\cdot - \Delta}$ over all $\Delta \in [-U, U]$, where

$$U = (1 - \gamma/2)\frac{n}{2B}$$

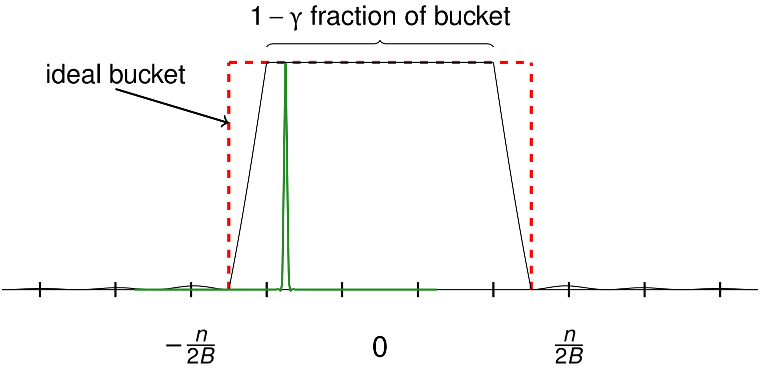
2. normalize so that $\widehat{G}_0 = 1 \pm \delta$



Flat window function – construction

To construct $\widehat{G}$:

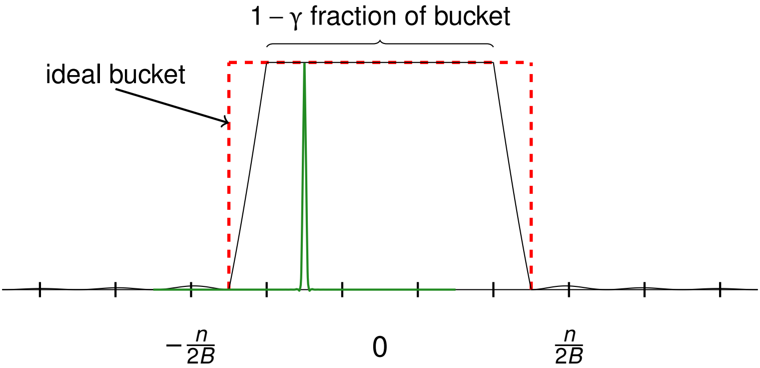
1. sum up shifts $\widehat{H}_{\cdot - \Delta}$ over all $\Delta \in [-U, U]$, where
$$U = (1 - \gamma/2)\frac{n}{2B}$$
2. normalize so that $\widehat{G}_0 = 1 \pm \delta$



Flat window function – construction

To construct $\widehat{G}$:

1. sum up shifts $\widehat{H}_{\cdot - \Delta}$ over all $\Delta \in [-U, U]$, where
$$U = (1 - \gamma/2)\frac{n}{2B}$$
2. normalize so that $\widehat{G}_0 = 1 \pm \delta$



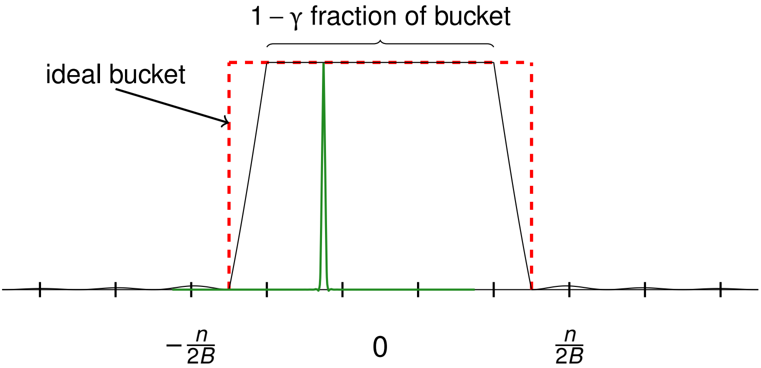
Flat window function – construction

To construct $\widehat{G}$:

1. sum up shifts $\widehat{H}_{\cdot - \Delta}$ over all $\Delta \in [-U, U]$, where

$$U = (1 - \gamma/2)\frac{n}{2B}$$

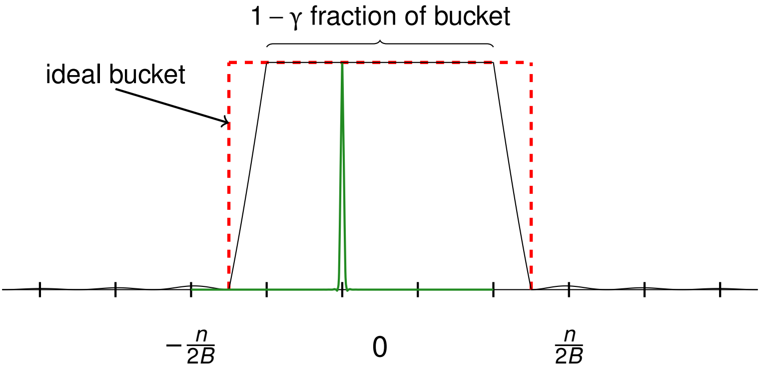
2. normalize so that $\widehat{G}_0 = 1 \pm \delta$



Flat window function – construction

To construct $\widehat{G}$:

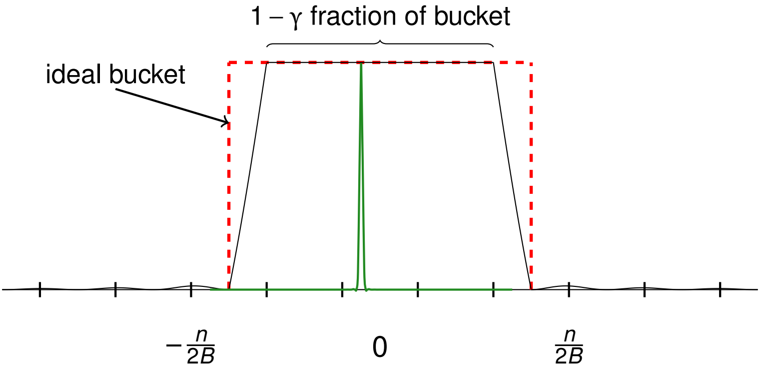
1. sum up shifts $\widehat{H}_{\cdot - \Delta}$ over all $\Delta \in [-U, U]$, where
$$U = (1 - \gamma/2)\frac{n}{2B}$$
2. normalize so that $\widehat{G}_0 = 1 \pm \delta$



Flat window function – construction

To construct $\widehat{G}$:

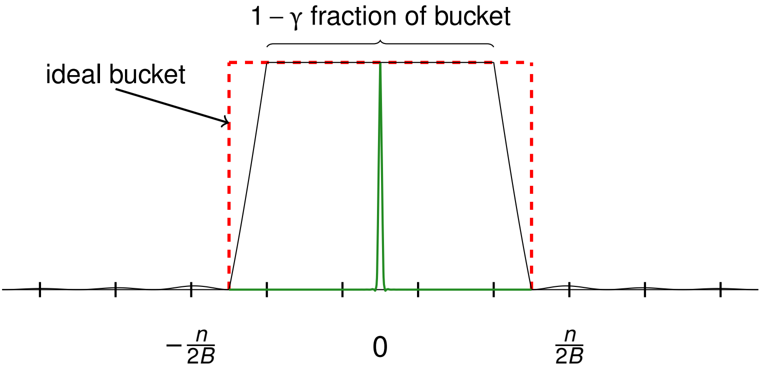
1. sum up shifts $\widehat{H}_{\cdot - \Delta}$ over all $\Delta \in [-U, U]$, where
$$U = (1 - \gamma/2)\frac{n}{2B}$$
2. normalize so that $\widehat{G}_0 = 1 \pm \delta$



Flat window function – construction

To construct $\widehat{G}$:

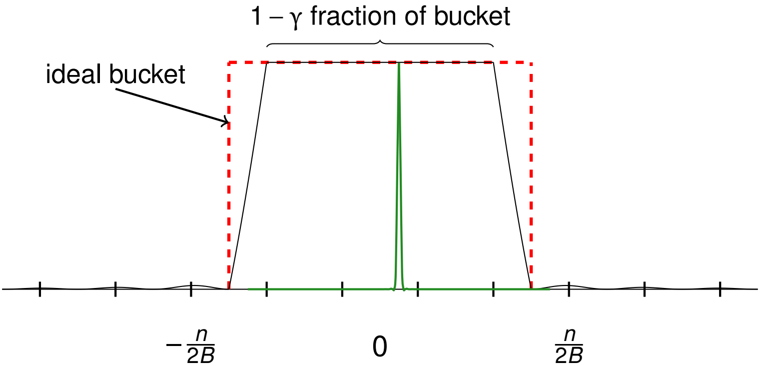
1. sum up shifts $\widehat{H}_{\cdot-\Delta}$ over all $\Delta \in [-U, U]$, where
$$U = (1-\gamma/2)\frac{n}{2B}$$
2. normalize so that $\widehat{G}_0 = 1 \pm \delta$



Flat window function – construction

To construct $\widehat{G}$:

1. sum up shifts $\widehat{H}_{\cdot - \Delta}$ over all $\Delta \in [-U, U]$, where
$$U = (1 - \gamma/2)\frac{n}{2B}$$
2. normalize so that $\widehat{G}_0 = 1 \pm \delta$



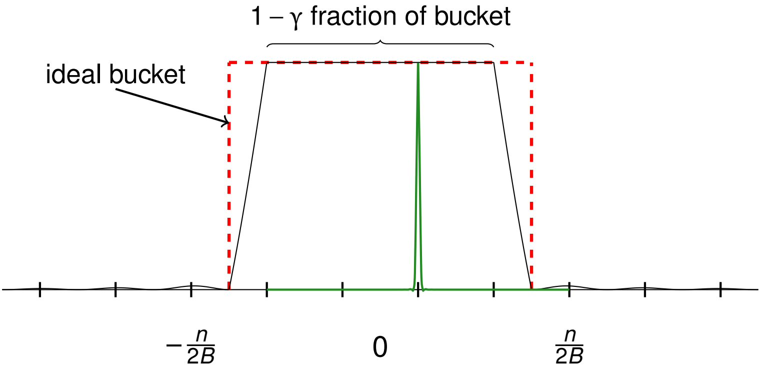
Flat window function – construction

To construct $\widehat{G}$:

1. sum up shifts $\widehat{H}_{\cdot - \Delta}$ over all $\Delta \in [-U, U]$, where

$$U = (1 - \gamma/2)\frac{n}{2B}$$

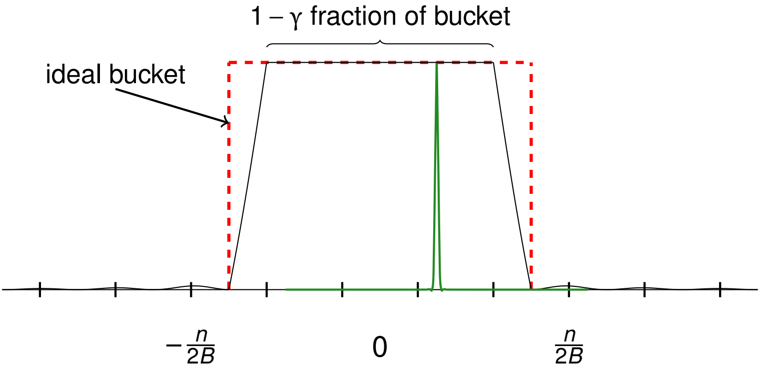
2. normalize so that $\widehat{G}_0 = 1 \pm \delta$



Flat window function – construction

To construct $\widehat{G}$:

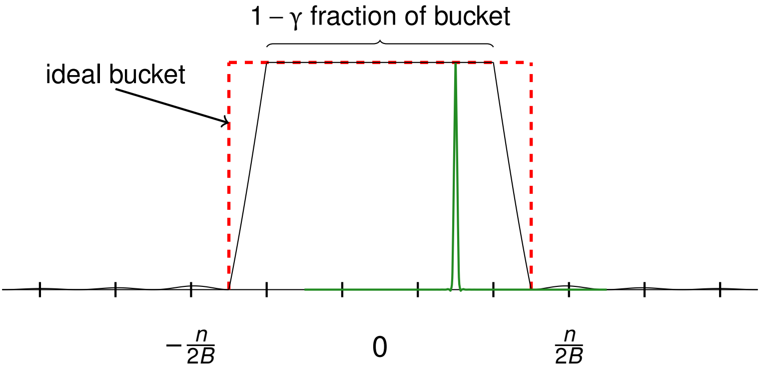
1. sum up shifts $\widehat{H}_{\cdot - \Delta}$ over all $\Delta \in [-U, U]$, where
$$U = (1 - \gamma/2)\frac{n}{2B}$$
2. normalize so that $\widehat{G}_0 = 1 \pm \delta$



Flat window function – construction

To construct $\widehat{G}$:

1. sum up shifts $\widehat{H}_{\cdot - \Delta}$ over all $\Delta \in [-U, U]$, where
$$U = (1 - \gamma/2)\frac{n}{2B}$$
2. normalize so that $\widehat{G}_0 = 1 \pm \delta$



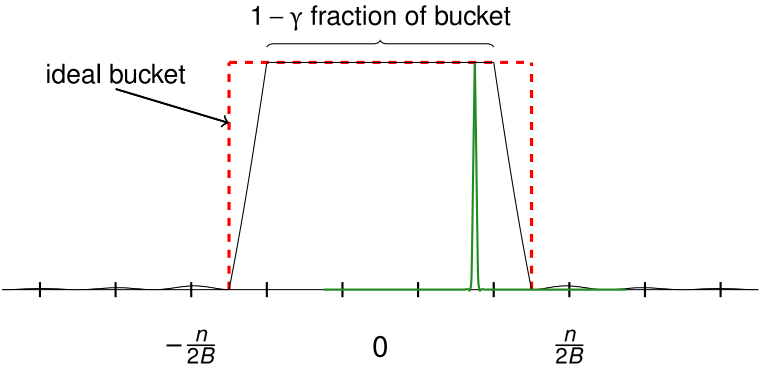
Flat window function – construction

To construct $\widehat{G}$:

1. sum up shifts $\widehat{H}_{\cdot - \Delta}$ over all $\Delta \in [-U, U]$, where

$$U = (1 - \gamma/2)\frac{n}{2B}$$

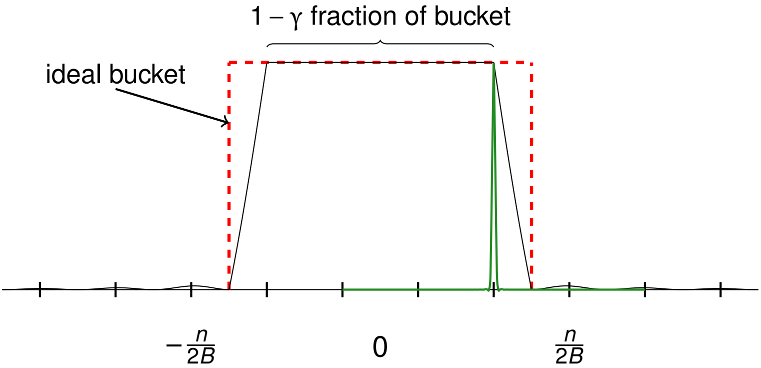
2. normalize so that $\widehat{G}_0 = 1 \pm \delta$



Flat window function – construction

To construct $\widehat{G}$:

1. sum up shifts $\widehat{H}_{\cdot-\Delta}$ over all $\Delta \in [-U, U]$, where
$$U = (1 - \gamma/2)\frac{n}{2B}$$
2. normalize so that $\widehat{G}_0 = 1 \pm \delta$



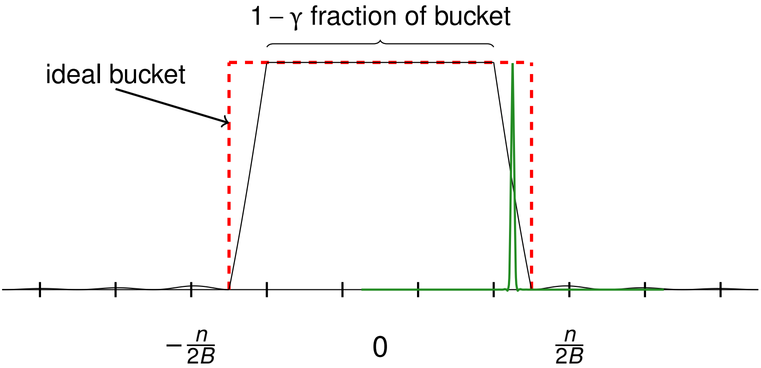
Flat window function – construction

To construct $\widehat{G}$:

1. sum up shifts $\widehat{H}_{\cdot - \Delta}$ over all $\Delta \in [-U, U]$, where

$$U = (1 - \gamma/2)\frac{n}{2B}$$

2. normalize so that $\widehat{G}_0 = 1 \pm \delta$



To construct $\widehat{G}$:

1. sum up shifts $\widehat{H}_{\cdot - \Delta}$ over all $\Delta \in [-U, U]$, where

$$U = (1 - \gamma/2)\frac{n}{2B}$$

2. normalize so that $\widehat{G}_0 = 1 \pm \delta$

To construct $\widehat{G}$:

1. sum up shifts $\widehat{H}_{\cdot - \Delta}$ over all $\Delta \in [-U, U]$, where
$$U = (1 - \gamma/2)\frac{n}{2B}$$
2. normalize so that $\widehat{G}_0 = 1 \pm \delta$

Formally:
$$\widehat{G}_f := \frac{1}{Z}\left(\widehat{H}_{f-U} + \widehat{H}_{f+1-U} + \ldots + \widehat{H}_{f+U}\right)$$
where Z is a normalization factor.

To construct $\widehat{G}$:

1. sum up shifts $\widehat{H}_{\cdot-\Delta}$ over all $\Delta \in [-U, U]$, where

$$U = (1-\gamma/2)\frac{n}{2B}$$

2. normalize so that $\widehat{G}_0 = 1 \pm \delta$

Formally:

$$\widehat{G}_f := \frac{1}{Z}\left(\widehat{H}_{f-U} + \widehat{H}_{f+1-U} + \ldots + \widehat{H}_{f+U}\right)$$

where Z is a normalization factor.

Upon inspection, $Z = \sum_{f \in [n]} \widehat{H}_f$ works.

Formally:

$$\widehat{G}_f := \frac{1}{Z}\left(\widehat{H}_{f-U} + \widehat{H}_{f+1-U} + \ldots + \widehat{H}_{f+U}\right)$$

where Z is a normalization factor.

Upon inspection, $Z = \sum_{f\in[n]} \widehat{H}_f$ works.

(Flat region) For any $f \in [-(1-\gamma)\frac{n}{2B}, (1-\gamma)\frac{n}{2B}]$ (flat region) one has

$$\begin{aligned}\widehat{H}_{f-U} + \widehat{H}_{f+1-U} + \ldots + \widehat{H}_{f+U} &\geq \sum_{f\in[-\gamma\frac{n}{4B}, \gamma\frac{n}{4B}]} \widehat{H}_f \\ &\geq Z - \text{tail of } \widehat{H} \\ &\geq Z - (\delta/n)n \geq Z - \delta\end{aligned}$$

Formally:

$$\widehat{G}_f := \frac{1}{Z}\left(\widehat{H}_{f-U} + \widehat{H}_{f+1-U} + \ldots + \widehat{H}_{f+U}\right)$$

where Z is a normalization factor.

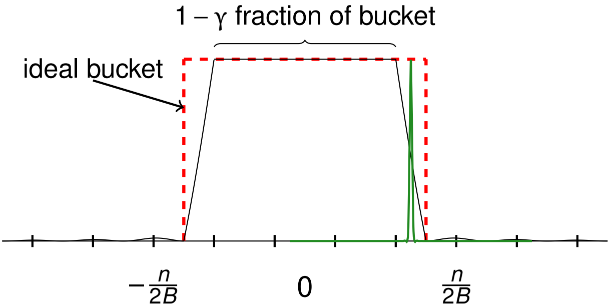
Upon inspection, $Z = \sum_{f\in[n]} \widehat{H}_f$ works.

Indeed, for any $f \notin [-\frac{n}{2B}, \frac{n}{2B}]$ (zero region) one has

$$\widehat{H}_{f-U} + \widehat{H}_{f+1-U} + \ldots + \widehat{H}_{f+U} \leq \sum_{f > \gamma \frac{n}{4B}} \widehat{H}_f$$

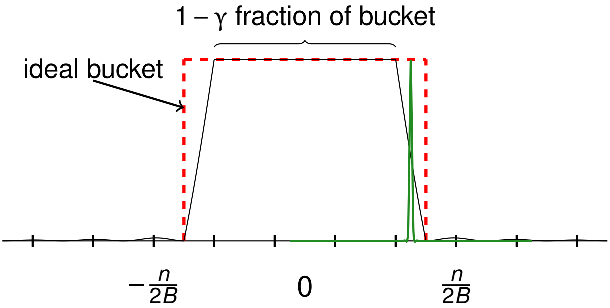
$$\leq \text{tail of } \widehat{H} \leq (\delta/n)n \leq \delta$$

Flat window function



How large is support of $\widehat{G} := \frac{1}{Z}\left(\widehat{H}_{\cdot - U} + \ldots + \widehat{H}_{\cdot + U}\right)$?

Flat window function

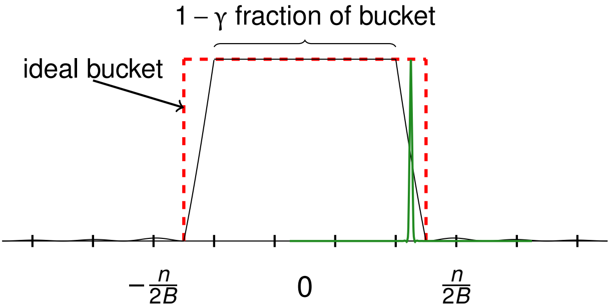


How large is support of $\widehat{G} := \frac{1}{Z}\left(\widehat{H}_{\cdot - U} + \ldots + \widehat{H}_{\cdot + U}\right)$?

By time shift theorem for every $q \in [n]$

$$G_q := H_q \cdot \frac{1}{Z} \sum_{j=-U}^{U} \omega^{qj}$$

Flat window function



How large is support of $\widehat{G} := \frac{1}{Z}\left(\widehat{H}_{\cdot - U} + \ldots + \widehat{H}_{\cdot + U}\right)$?

By time shift theorem for every $q \in [n]$

$$G_q := H_q \cdot \frac{1}{Z} \sum_{j=-U}^{U} \omega^{qj}$$

Support of G a subset of support of H!

Flat window functions – construction