This is the file: http://www.univie.ac.at/NuHAG/FEICOURS/ss11/DigSigBook.pdf

1 Digital Signal Processing Template

Summary. **Keywords.** test

1.1 Subsection 1

1.2 Subsection 2

1.3 MATLAB CODE

```
VERBATIM PART
```

1.4 Exercises

1. first
2. second

2 Brieflet!? on Computational Harmonic Analysis?

Summary. Comments on Signal Processing, also preparing material for the MACHA11 course (August 2011 in Marburg), by Hans G. Feichtinger an co-authors.

Format model: ([3]): Shlomo Engelberg: *Digital signal processing. An experimental approach.* In the book series “Signals and Communication Technology”. London: Springer. xv,
Keywords. test

2.1 Subsection 2

2.2 MATLAB CODE

```
VERBATIM PART
```

2.3 Exercises

1. first
2. second

3 Complex numbers, unit roots and plotting

Summary. It is the purpose of this section to show how to generate point sequences in the complex domain and how to plot complex-valued functions.
Keywords. test

3.1 Subsection 1

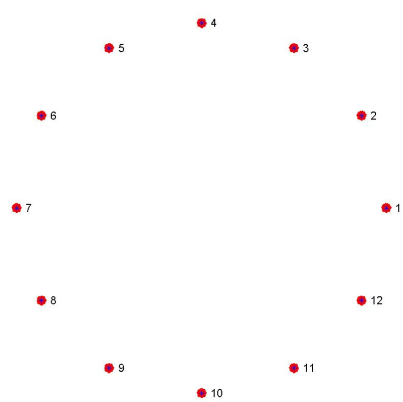
```
function    u = uroots(N,l);

if nargin == 0;  N = 8;  end;
bas = 0 : 1/N : (N-1)/N;
u =  exp(2 * pi * i * bas);
% i.e. unit roots are generated in the
% mathematical positive sense.
```

Using the sequence of MATLAB commands (NuHAG tools):

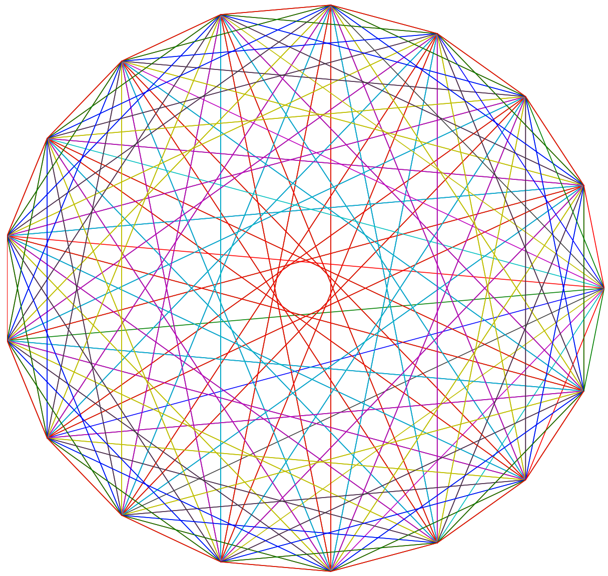
```
plotnum(uroots(12)); axis square; axis off; figure(gcf);
```

one obtains the following plot



3.2 Subsection 2

Plotting the columns of the FFT-matrix in the complex plane



3.3 MATLAB CODE

```
VERBATIM PART
```

3.4 Exercises

1. Try: `>> plot(fft(eye(n)));` , e.g. for $n = 13$.

2. `F = fft(eye(256)); plot(real(F(2,:)),imag(F(2,:)));`

3. `n=256; plot(1:n,real(F(2,:)),'k',1:n,imag(F(2,:)),'r'); axis tight; '`

4 DigSig FFT as a Vandermonde Matrix

Summary. TEXT
Keywords. test

4.1 Subsection 1

4.2 Subsection 2

```
norm( fft(eye(4)) - fliplr(vander(conj(uroots(4)))))
ans =    7.7079e-016
```

shows that essentially the Fourier transform is Vandermonde-matrix. It uses the unit-roots in the clockwise sense: `conj(uroots(9))` . The `fliplr` is needed due to the convention of MATLAB to convert a sequence of coefficients to a polynomial of the form $a_{n-1}x^{n-1} + \cdots + a_n$, and one has

```
>> vander(1:4)
ans =
     1     1     1     1
     8     4     2     1
    27     9     3     1
    64    16     4     1

    >> a = rand(1,3), x = 1 : 3;
a =     0.9080    0.8758    0.6048
>> polyval(a,x)
ans =      2.3887    5.9887   11.4047
>> vander(x) * a'
ans =      2.3887
           5.9887
          11.4047
```

4.3 MATLAB CODE

```
VERBATIM PART
```

4.4 Exercises

1. first
2. second

5 Functions on the Unit Circle

Summary. Although the FFT (resp. DFT) is just a linear mapping and hence representable (not realized!) by a matrix, but it is very helpful to interpret it not just as a linear mapping from $\mathbb{C}^n$ to $\mathbb{C}^n$, but from function on the set of unit roots of order N, i.e. $\mathbf{Z}_N$. Although in the finite dimensional setting any pair of norms is equivalent to each other (hence one could always choose the Euclidian norm $\|\vec{\mathbf{z}}\|_2 = \sqrt{\sum_{k=1}^N |z_k|^2}$). We express it by viewing vectors of length N (alternatively later on n) as elements of $\boldsymbol{\ell}^2(\mathbf{Z}_N)$, sometimes also as elements of $\boldsymbol{\ell}^1(\mathbf{Z}_N)$ of $\boldsymbol{\ell}^\infty(\mathbf{Z}_N)$, with the endowed with the $\boldsymbol{\ell}^1$ or sup- (= max-norm) respectively.

Theorem 1. *Since the Fourier matrix F_N is, up to scaling (by the factor $\sqrt{N}$) a unitary linear mapping, one finds that $\mathcal{F}\colon \vec{\mathbf{x}} \mapsto \vec{\mathbf{y}} = \mathcal{F}(\vec{\mathbf{x}})$ satisfies*

$$\|\vec{\mathbf{y}}\|_2 = N \cdot \|\vec{\mathbf{x}}\|_2. \tag{1}$$

Keywords. test

5.1 Plotting signals of finite length

The interpretation of finite sequences as functions on $\mathbf{Z}_N$ is nothing else but viewing a finite sequence of length N as a *periodic* (infinite) sequence

5.2 Subsection 2

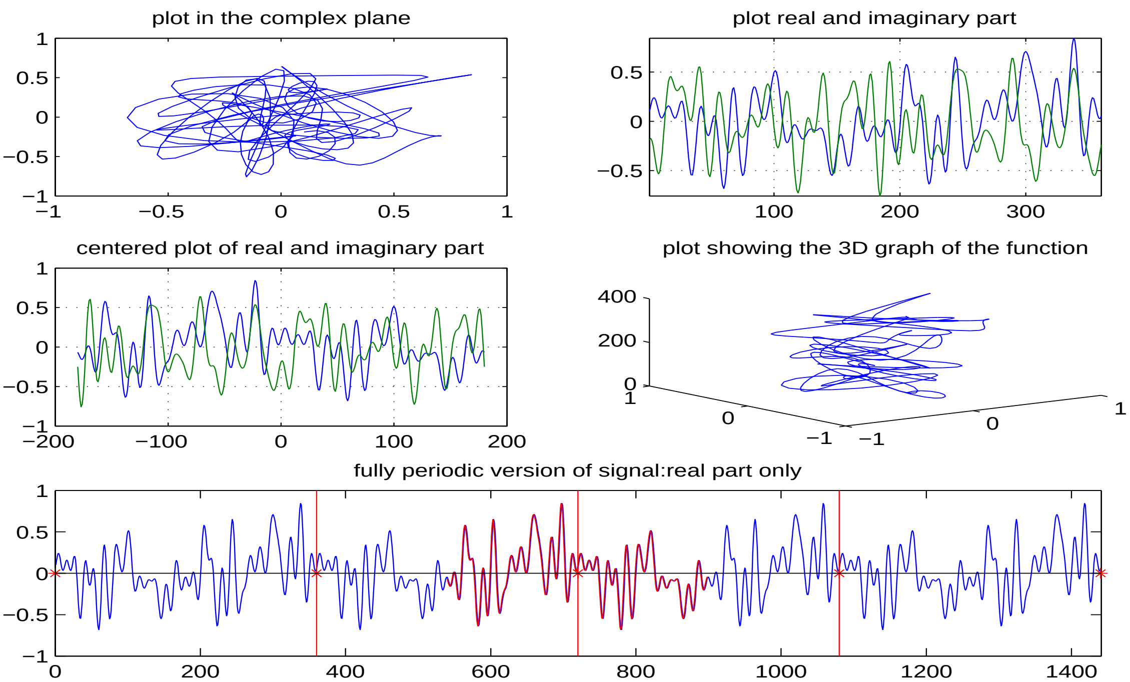
5.3 MATLAB CODE

```
VERBATIM PART
```

5.4 Exercises

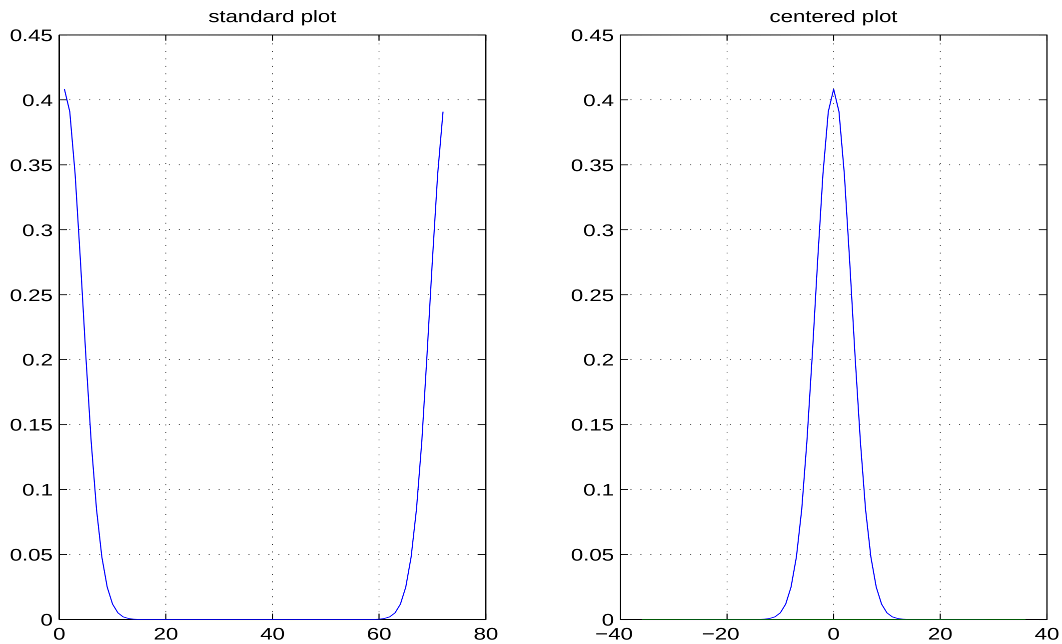
1. first

2. second

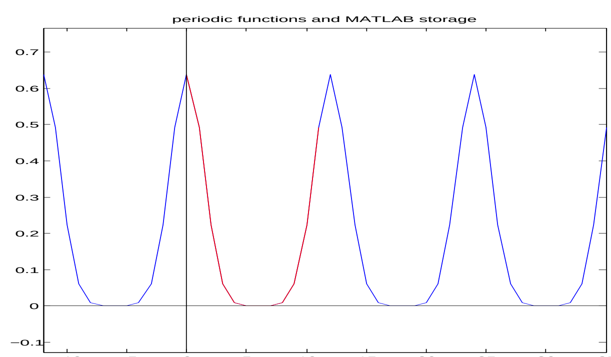


TOP left: standard plot in the complex plane; TOP right: plotri (real/imag plot); MIDDLE left: plotc(xx) (centered plot); MIDDLE right: $3D-$plot; BOTTOM: plotting the periodic version of (for simplicity) the real part.

5.5 Subsection 1



Left: standard plot `plot(g);` versus right `plotc(g);`



How the different plot show *natural sections* of a periodic signal.

6 Pure Frequencies: Eigenvectors for Cyclic Shift

Summary. TEXT The interpretation of finite signals functions on the cyclic group of order n (resp. N), i.e. as elements of $\boldsymbol{\ell}^2(\mathbf{Z}_N)$, resp. as periodic functions, implies that we have the shift operators $T_k, k \in \mathbb{Z}$ (for the periodic case) resp. its analogue on $\boldsymbol{\ell}^2(\mathbf{Z}_N)$ which is simple the cyclic shift. Instead of a formal definition let us give an example of use in MATLAB (code is given below)[1]:

```
rot(1:8,2) ==  [7,8, 1,2,3,4,5,6];
rot(1:7,-2) == [3,4,5,6,7,1,2];
rot(1:12,35) == [12,1,2,3,4,5,6,7,8,9,10,11];
```


Keywords. test

6.1 Subsection 1: Cyclic Shift operators

It is obvious that we have

$$T_r \circ T_s = T_{r+s} = T_s \circ T_r, \quad r, s \in \mathbb{Z} \tag{2}$$

and also

$$T_r = T_s \quad \text{if and only if} \quad r - s == 0(\texttt{mod}\ n). \tag{3}$$

Hence we have the simple Lemma:

Lemma 1. *The mapping $k \mapsto T_k$ is an isomorphism from Z_n (viewed as $\mathbb{Z}/(n\mathbb{Z})$) onto a commutative group of unitary operators on $\boldsymbol{\ell}^2(\mathbb{Z}_n)$.*

Proof. Note only that T_k preserves norms and is invertible, hence it is unitary operator on $\boldsymbol{\ell}^2(\mathbb{Z}_n)$. □

Corollary 1. *The linear span of all the shift operators consists exactly of all operators which can be written (uniquely) in the form*

$$T = \sum_{k=0}^{N-1} c_k T_k, \quad \vec{\mathbf{c}} = (c_k)_{k=0}^n \in \boldsymbol{\ell}^1(\mathbb{Z}_N). \tag{4}$$

They form a commutative Banach algebra of complex $N \times N$-matrices.

Proof. Using the composition law (3) one easily finds that the set of matrices of the from (4) are closed under composition, and that this composition is a commutative one.

It is also clear that the representation of T in the form (4) is *unique*, and that, endowed with the norm

$$\|T\|_{\mathbf{1}} := \sum_{k=0}^{N-1} |c_k| = \|\vec{\mathbf{c}}\|_1 \tag{5}$$

the composition satisfies

$$\|T_1 \circ T_2\|_{\mathbf{1}} \leq \|T_1\|_{\mathbf{1}} \|T_2\|_{\mathbf{1}}, \tag{6}$$

In other words, the composition of these operators endowed with the $\boldsymbol{\ell}^1$-norm (5) turns $\boldsymbol{\ell}^1(\mathbb{Z}_N)$ into a commutative Banach algebra of matrices of *dimension* N. □

[1] the code is typically acting on row vectors

We can also take a look at the corresponding matrices: The matrix for T_1 (right shift) is of course (for $N = 5$):

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \tag{7}$$

and hence the operator corresponding to the sequence $\vec{\mathbf{c}} = [1, 2, 3, 4, 5, 6]$ is the matrix

$$\begin{pmatrix} 1 & 6 & 5 & 4 & 3 & 2 \\ 2 & 1 & 6 & 5 & 4 & 3 \\ 3 & 2 & 1 & 6 & 5 & 4 \\ 4 & 3 & 2 & 1 & 6 & 5 \\ 5 & 4 & 3 & 2 & 1 & 6 \\ 6 & 5 & 4 & 3 & 2 & 1 \end{pmatrix} \tag{8}$$

By transfer of the multiplication law we can thus turn $\boldsymbol{\ell}^1(\mathbf{Z}_N)$ into a commutative Banach algebra, called the *group algebra* of $G = \mathbf{Z}_N$. The corresponding multiplication is called *convolution*. Each convolution matrix is also corresponding to a unique bounded operator on $\boldsymbol{\ell}^2(\mathbf{Z}_N)$, i.e. we have a representation of the Banach algebra $\boldsymbol{\ell}^1(\mathbf{Z}_N)$ on the Hilbert space $\mathcal{H} = \boldsymbol{\ell}^2(\mathbf{Z}_N)$.

Theorem 2. *Given two sequences $\vec{\mathbf{a}}, \vec{\mathbf{b}} \in \boldsymbol{\ell}^1(\mathbb{Z}_N)$, both indexed in the sense of the corresponding operators, i.e. indexed with $0, \ldots, N-1$, then the resulting convolution product, denoted by $\vec{\mathbf{c}}$ is given by the coordinates, following the* **Cauchy Product Rules***:*

$$c_k = \sum_{k=0}^{N-1} a_j b_{N-j} = \sum_{r+s=k} a_r b_s. \tag{9}$$

Note that the product of convolution matrices can be characterized by the effect of the circulant matrix of the first factor applied just to the first column of the second factor, hence one can say, that the convolution can be described directly at the level of generators of these convolution matrices, i.e. we have the following statement:

Lemma 2. *For $\vec{\mathbf{c}}, \vec{\mathbf{d}} \in \boldsymbol{\ell}^1(\mathbb{Z}_N)$ gilt:*

$$\vec{\mathbf{a}} := \vec{\mathbf{c}} * \vec{\mathbf{d}} = \sum_{k=1}^{n} c_k T_{k-1} \vec{\mathbf{d}} \tag{10}$$

or in a coordinate description

$$a_k = \sum_{r=0}^{N-1} b_{k-r} c_r. \tag{11}$$

CHECK THE DETAILS! *This can also be reinterpreted as a scalar product*

$$a_k = \langle \vec{\mathbf{c}}, T_k \vec{\mathbf{b}}^* \rangle. \tag{12}$$

To make it more concrete let us do it on $\mathbb{Z}_8$: Given the sequence $[a_0, a_1, \ldots, a_7]$ and correspondingly $[b_0, \ldots b_7]$ we have (e.g.) $c_2 = a_0 b_2 + a_1 b_1 + a_2 b_0 + a_3 b_{-1} + \ldots a_7 b_{-5}$ which of course has to be interpreted modulo 8, hence as

$$c_2 = a_0 b_2 + a_1 b_1 + a_2 b_0 + a_3 b_7 + \ldots a_7 b_3$$

Recall that the involution applied to $\vec{\mathbf{b}}$ (up to conjugation) is just $[b_7, b_6, \ldots, b_0]$ and its translate by 2 is just

Noting that MATLAB does not allow for zero-indexing and rewriting the sequences $\mathbf{a}, \mathbf{b}, \mathbf{c}$ as $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{C}^N$ we find that the convolution product $\mathbf{z}$ has as a typical coordinate

$$z_2 = x_1 y_3 + x_2 y_2 + x_3 y_1 + x_4 y_8 + \ldots x_8 y_4.$$

Such a some can be translated into a scalarproduct, involving the flipped version of $\vec{\mathbf{b}}$ resp. $\mathbf{y}$, which is $\check{\mathbf{y}} = [y_1, y_8, \ldots, y_2]$ which essentially means reading out the data from the group Z_N not in the clockwise sense, but instead in the mathematical positive sense.

There is also a natural involution on the set of circulant matrices, namely taking the adjoint. The adjoint of a circulant matrix whose *generator* (i.e. first row) is $\vec{\mathbf{a}}$ is just

$$\mathbf{a}^* := [\overline{a_1}, \overline{a_8}, \ldots, \overline{a_2}]$$

By looking at the matrix corresponding to the adjoint of the circulant matrix generated by $\mathbf{a}$ we find out that the adjointness relation $\mathbf{a} \mapsto \mathbf{a}^*$ is just the natural analogue of the involution, taken at the matrix level (or equivalently in the operator algebra on the Hilbert space $\boldsymbol{\ell}^2(\mathbf{Z}_N)$).

This involution is compatible with the usual adjointness relation: The adjoint of the matrix given above is:

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 6 & 1 & 2 & 3 & 4 & 5 \\ 5 & 6 & 1 & 2 & 3 & 4 \\ 4 & 5 & 6 & 1 & 2 & 3 \\ 3 & 4 & 5 & 6 & 1 & 2 \\ 2 & 3 & 4 & 5 & 6 & 1 \end{pmatrix} \tag{13}$$

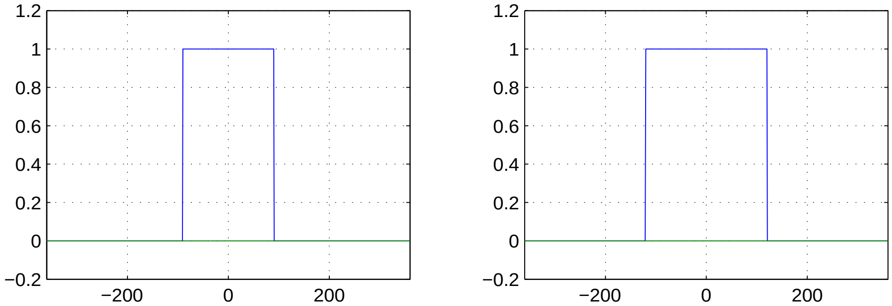
The involution, which is quite different from the flip-operator (in MATLAB we have `flipud, fliplr`), and can be understood better if we interpret the sequences $\vec{\mathbf{c}}$ not just as elements of $\mathbb{C}^n$ but rather as elements in $\boldsymbol{\ell}^1(\mathbb{Z}_N)$ (with $f(0) = c_1, f(1) = c_2$, etc.). So in the continuous domain the involution (at the signal level) is of the form

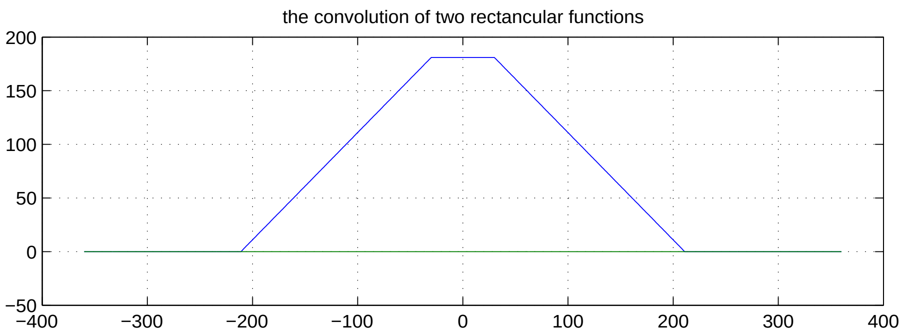
$$f \mapsto f^* : f^*(x) = \overline{f(-x)}.$$

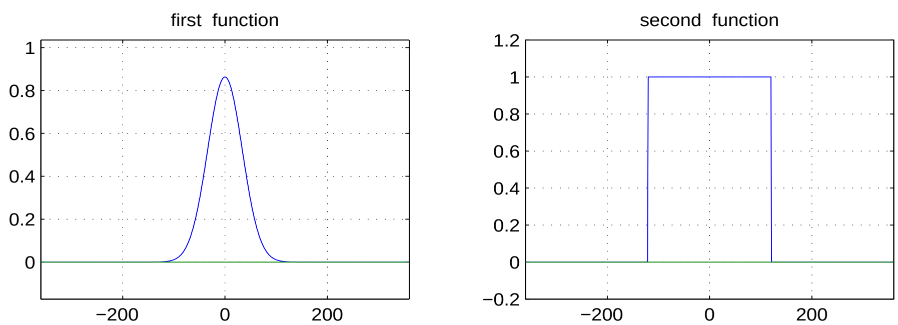
Clearly

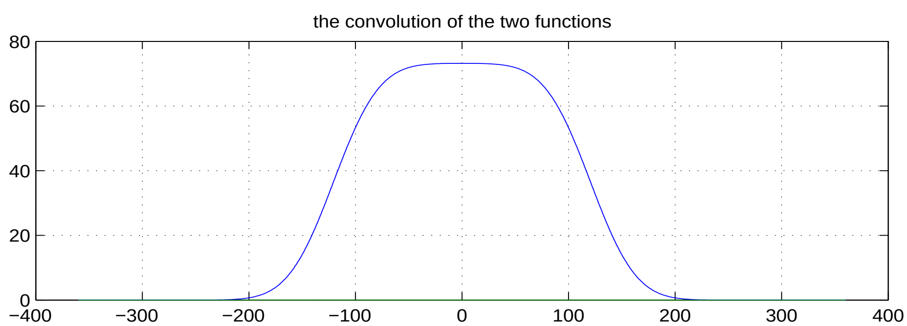
$$\|\mathbf{x}^*\|_1 = \|\mathbf{x}\|_1, \quad (T_k\mathbf{x})^* = T_{-k}\mathbf{x}^*, \quad \text{and} \quad \mathbf{x}^{**} = \mathbf{x}.$$

A few concrete demos for convolution


1.2
1
0.8
0.6
0.4
0.2
0
−0.2
−200
0
200


the convolution of two rectancular functions
200
150
100
50
0
−50
−400
−300
−200
−100
0
100
200
300
400


first function
second function
1.2
1
0.8
0.6
0.4
0.2
0
−0.2
−200
0
200


the convolution of the two functions
80
60
40
20
0
−400
−300
−200
−100
0
100
200
300
400

6.2 Subsection 2: Pure Frequencies as eigenvectors

Here it will be shown that the *pure frequencies*, i.e. the columns (or rows!) of the DFT Matrix (`fft(eye(n))`) are exactly the eigenvectors for (the commutative family) of cyclic shift operators.

Theorem 3. *The pure frequencies are* **joint eigenvectors** *to the (commutative Banach) algebra of circulant matrices, and the eigenvalue for a given character χ_s is just the value of the corresponding Fourier transform, which is*

$$\langle \vec{\mathbf{c}}, \vec{\chi_{\mathbf{s}}} \rangle = \sum_{l=0}^{N-1} c_l \, \omega^{-sl} = d_s, \quad \textit{for } \vec{\mathbf{d}} = \mathrm{fft}(\vec{\mathbf{c}}). \tag{14}$$

Proof. We have seen that the homomorphism property of χ_s (due to the exponential law)

$$[T_z \chi](x) = \chi(x - z) = \chi(x) \cdot \chi(-z) = \overline{\chi(z)} \chi(x) \tag{15}$$

implies that χ_s is an eigenvector to all the translation operators T_z, with eigenvalue $\chi(-z)$, or in argument-free form:

$$T_z \chi_s = \chi(-s) \chi_s.$$

Hence with the above convention of putting $\vec{\mathbf{d}} = \mathrm{fft}(\vec{\mathbf{c}})$ we have:

$$T \chi_s = (\sum_l c_l T_l) \chi_s = [\sum_l c_l \chi_s(-l)] \chi_s = d_s \chi_s \tag{16}$$

Summarizing these facts in matrix format we find out that the circulant matrix $\mathbf{C}$ representing the operator C (convolution by $\vec{\mathbf{c}}$) is the diagonal matrix generated from the sequence $\vec{\mathbf{d}}$, the Fourier transform of $\vec{\mathbf{c}}$:

$$C = F' * D * F/N, \quad \text{with} \quad D = diag(\vec{\mathbf{d}}). \tag{17}$$

□

Note that for the concrete choice of $G = \mathbf{Z}_N$ one easily finds, that the set of all possible homomorphism (nontrivial, from $\mathbf{Z}_N$ into $\mathbb{C} \setminus \{0\}$) can be identified with the set

$$\{\chi_s, 0 \leq s \leq N - 1\}.$$

Since it is clear that the product (in the pointwise sense) of two characters of any group is again a character, due to the identity

$$[\chi_1 \cdot \chi_2](x+y) := \chi_1(x+y) \cdot \chi_2(x+y) = \chi_1(x) \chi_1(y) \chi_2(x) \chi_2(y) = [\chi_1 \chi 2](x) [\chi_1 \chi_2](y). \tag{18}$$

It is also not difficult to verify that an operator which can be diagonalized via the Fourier transform, i.e. a matrix, which is of the form $\boldsymbol{A} = F * \mathbf{D} * F$ for some diagonal matrix $\mathbf{D}$ has to be a *circulant matrix*. This follows from the fact that a diagonal matrix is just a multiplication operator, which obviously implies that it commutes with the pointwise multiplication with pure frequencies. But multiplication with characters on the Fourier transform side is exactly commutation with translation operators.

So it remains to verify that a matrix commuting with cyclic shift operators (it is enough to focus on cyclic shift by one sample, the rest follows therefrom!) is a circulant matrix. Thus we have to inspect matrices which satisfy the invariance property

$$\boldsymbol{A} = T_{-1} * \boldsymbol{A} * T_1. \tag{19}$$

This matrix has the same entries, but both in the row direction and the column direction cyclically shifted, thus e.g. turning

$$\begin{pmatrix} 1 & 5 & 9 & 13 \\ 2 & 6 & 10 & 14 \\ 3 & 7 & 11 & 15 \\ 4 & 8 & 12 & 16 \end{pmatrix} \tag{20}$$

into the matrix

$$\begin{pmatrix} 6 & 10 & 14 & 2 \\ 7 & 11 & 15 & 3 \\ 8 & 12 & 16 & 4 \\ 5 & 9 & 13 & 1 \end{pmatrix} \tag{21}$$

Observe that the block

$$\begin{pmatrix} 6 & 10 & 14 \\ 7 & 11 & 15 \\ 8 & 12 & 16 \end{pmatrix} \tag{22}$$

in the lower right corner of the matrix is moved up along the main diagonal!

A formal verification results from the observation that the matrix entries ...

6.3 MATLAB CODE

```
VERBATIM PART
```

6.4 Exercises

1. first

2. second

7 Fourier Basics

Summary. TEXT **Keywords.** test

7.1 Subsection 2

7.2 MATLAB CODE

```
VERBATIM PART
```

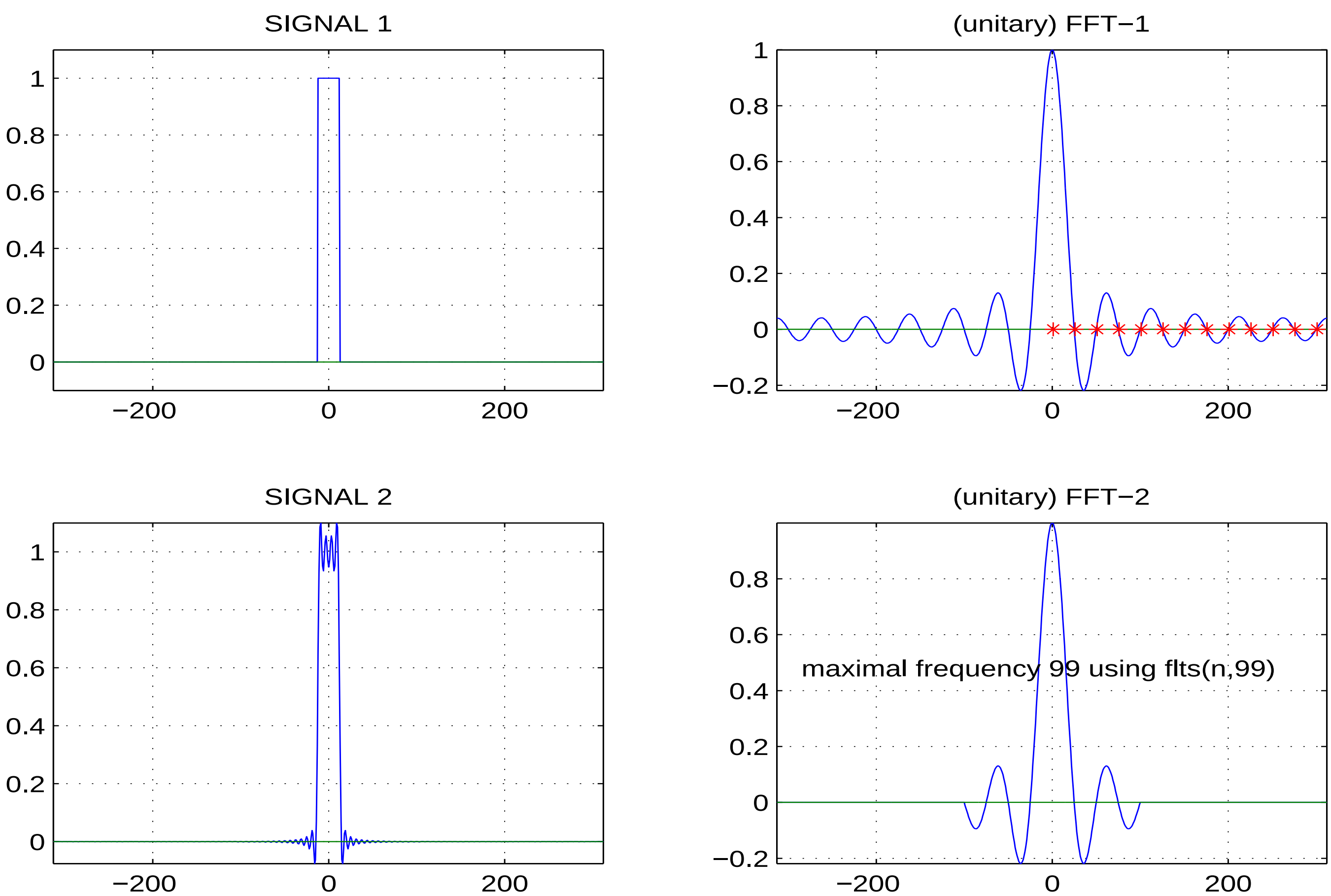
7.3 Exercises

1. first
2. second

8 Fourier Inversion and Gibbs Phenomenon

Summary. Keywords. test

8.1 Gibbs Phenomenon



8.2 Subsection 2

8.3 MATLAB CODE

```
VERBATIM PART
```

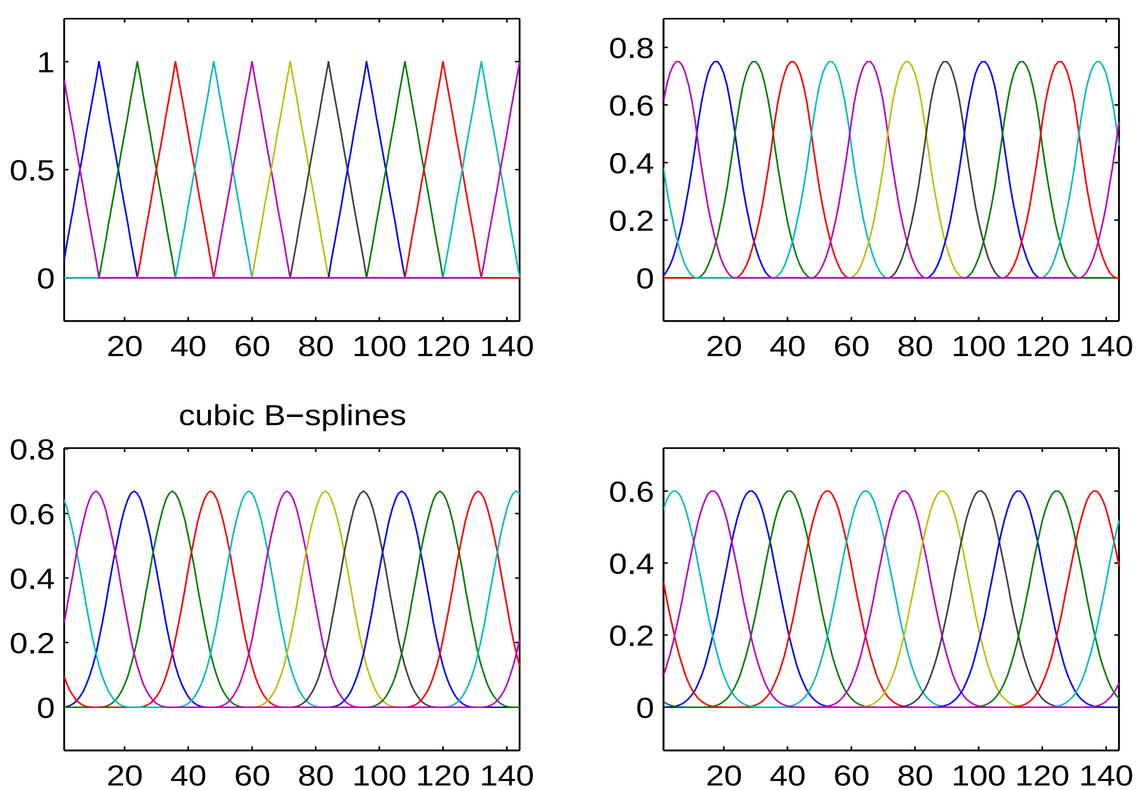
8.4 Exercises

1. first

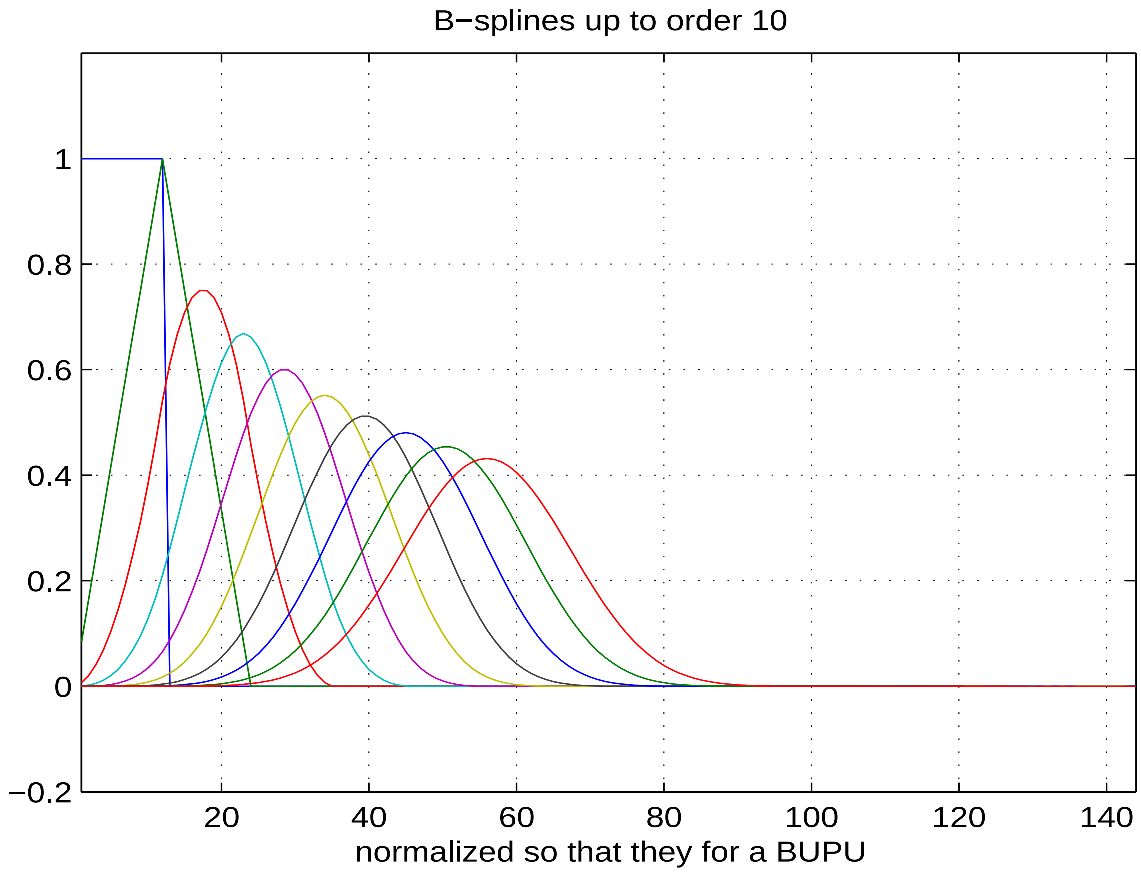
2. second

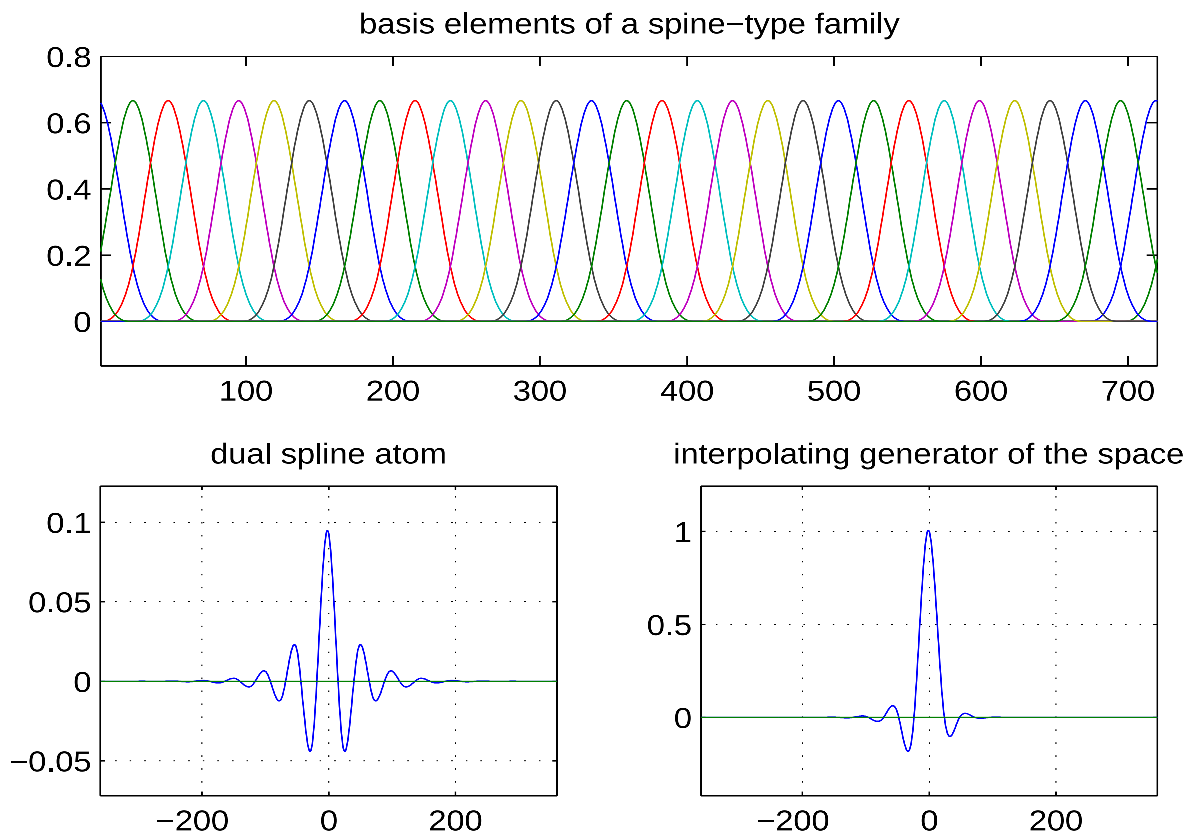
9 Spline Function Basics

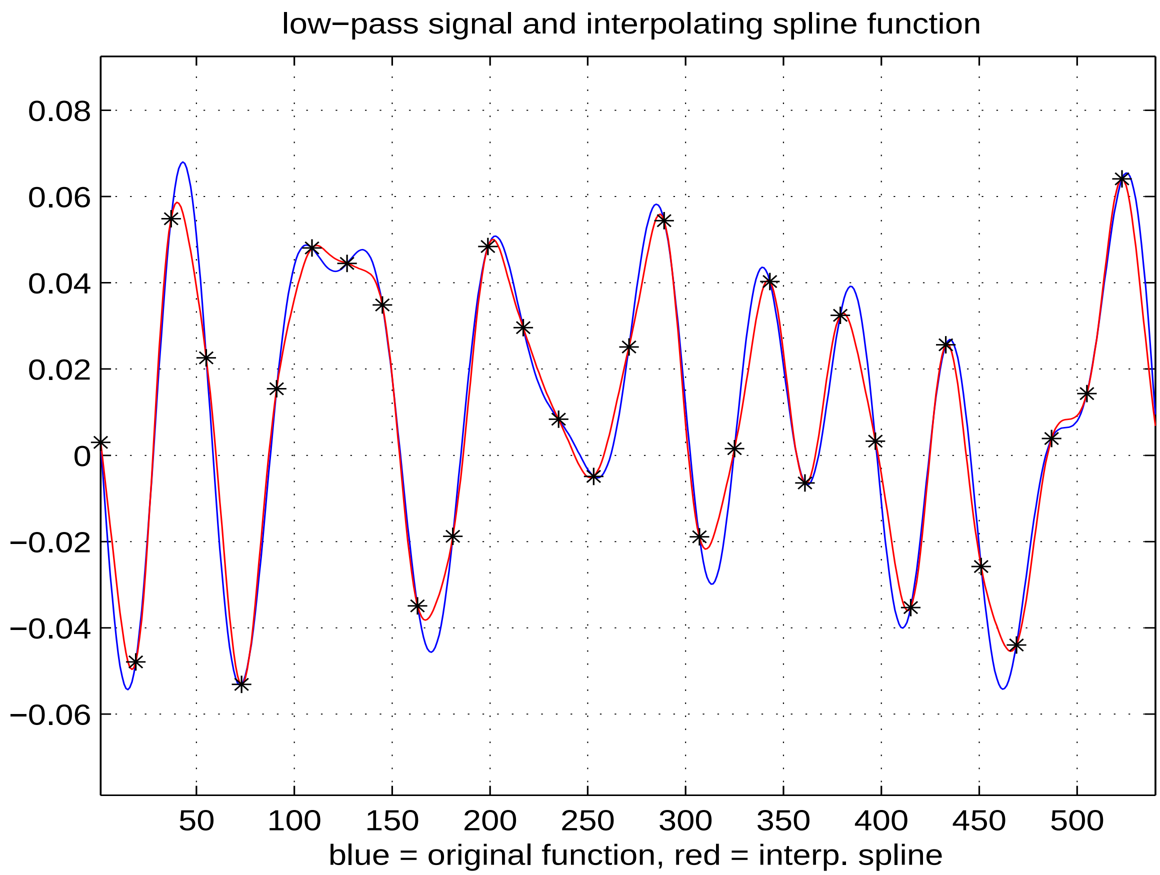
Summary. Spline functions are obtained as convolution powers of box-functions. Therefore the collection of spline-functions has the same translative structure, and also constitutes (for each degree of smoothness) a (bounded uniform) partition of unity.



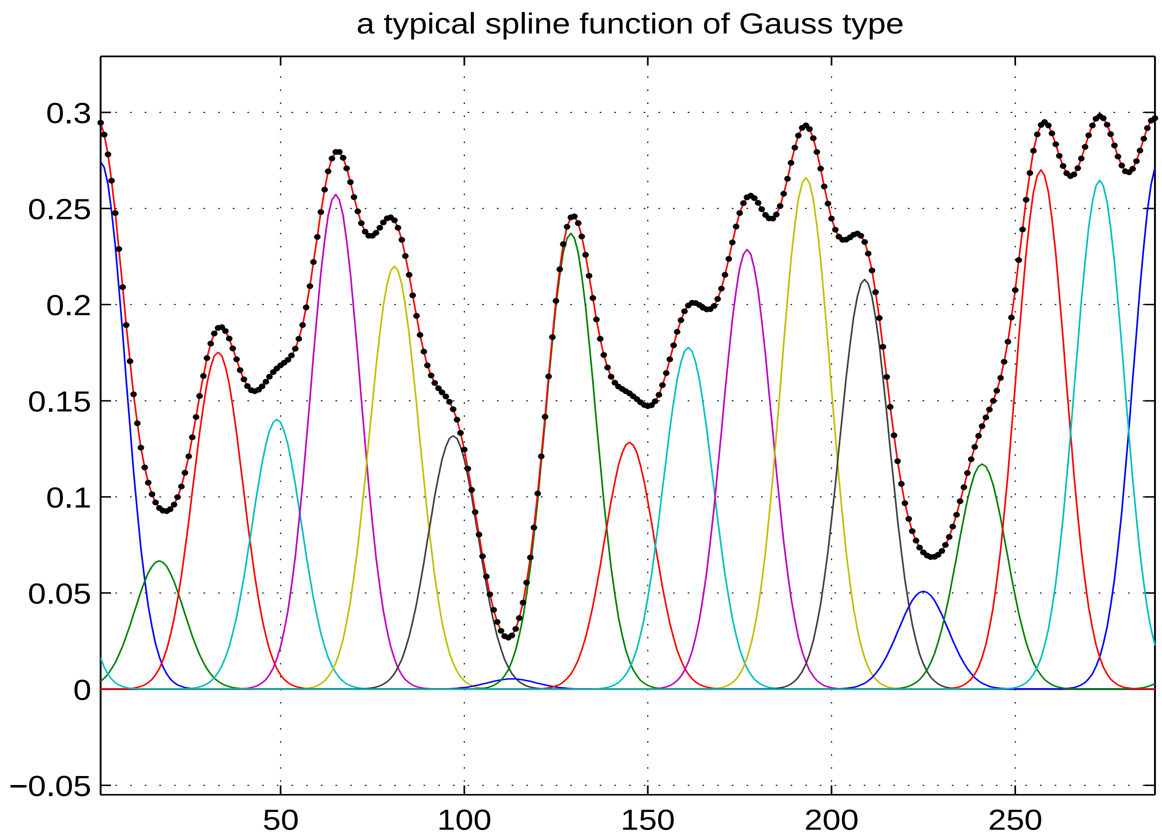
Spline of various, i.e. higher order


B−splines up to order 10
1
0.8
0.6
0.4
0.2
0
−0.2
20
40
60
80
100
120
140
normalized so that they for a BUPU


basis elements of a spine−type family
0.8
0.6
0.4
0.2
0
100
200
300
400
500
600
700
dual spline atom
0.1
0.05
0
−0.05
−200
0
200
interpolating generator of the space
1
0.5
0
−200
0
200



low−pass signal and interpolating spline function
0.08
0.06
0.04
0.02
0
−0.02
−0.04
−0.06
50
100
150
200
250
300
350
400
450
500
blue = original function, red = interp. spline



Keywords. test

9.1 Subsection 2

9.2 MATLAB CODE

```
VERBATIM PART
```

9.3 Exercises

1. first
2. second

10 Scalar Products, Orthonormal Systems

Summary. For the description of signal spaces (linear manifolds of signals) and the operators between them the most important subclass is the family of Hilbert spaces, such as $\boldsymbol{L}^2(\mathbb{R}^d)$, known as the space of Lebesgue square integrable complex-valued functions (>

separate chapter) or *signals of finite energy* (interpreting $\int_{\mathbb{R}^d} |f(x)|^2 dx$ as the "energy" contained in the signal f). While infinite-dimensional (mostly separable) Hilbert spaces $\mathcal{H}$, which can be shown to be isomorphic to $\ell^2(I)$ for some countable index set I (via Gram-Schmidt, which allows to built complete orthonormal bases in such spaces), the situation is much easier in the finite dimensional context. For this reason we will work with finite dimensional Hilbert spaces (signals of finite length resp. periodic, discrete signals)

Keywords. test

10.1 Subsection 1

10.2 Subsection 2

10.3 MATLAB CODE

```
VERBATIM PART
```

10.4 Exercises

1. first

2. second

11 Sampling and Shannon's Sampling Theorem

The core statement of this section is the Whittaker-Kotelnikov-Shannon Theorem, which states that an $\boldsymbol{L}^2(\mathbb{R})$-function whose Fourier transform is contained in the symmetric interval $I = [-1/2, 1/2]$ around zero (i.e. $\mathrm{supp}(\hat{f}) \subseteq I$) can be completely recovered from regular samples of the form $(f(\alpha n)_{n\in\mathbb{Z}}$ as long as $\alpha \leq 1$.

The reconstruction can be achieved using the so-called SINC-function, with $SINC(t) = sin(\pi t)/\pi t$, the *sinus cardinales*[2], which can be characterized as the inverse Fourier transform of the box-function $\mathbf{1}_I$, the indicator function of I.

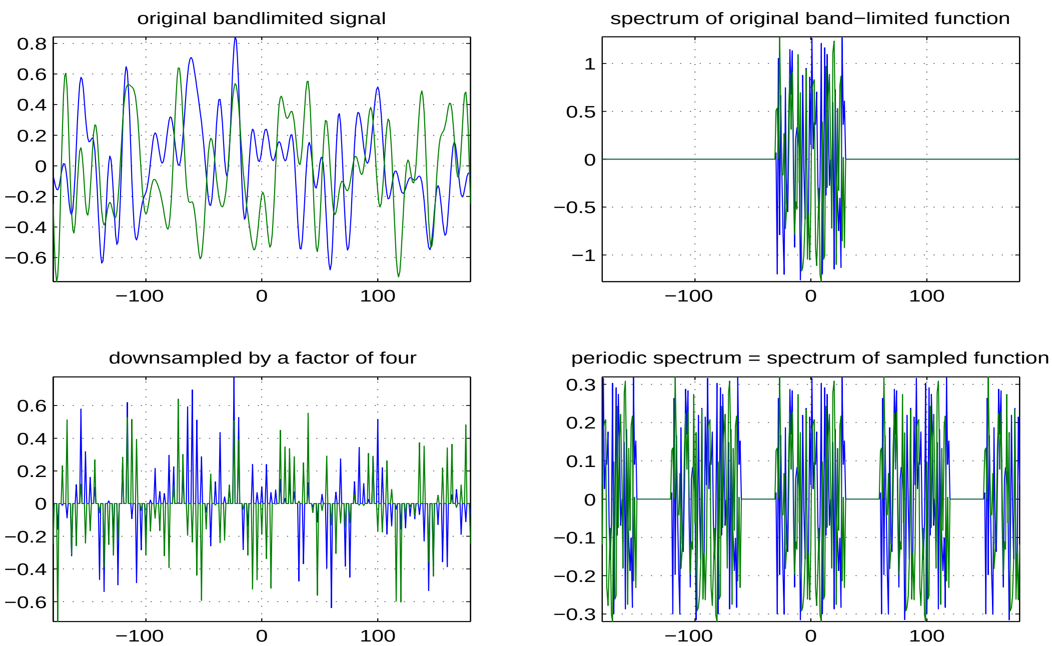
It is convenient to apply the following notation:

$$\boldsymbol{B}_I := \{f \mid f \in \boldsymbol{L}^2(\mathbb{R}),\, \mathrm{supp}(\hat{f}) \subseteq I\}, \tag{23}$$

Due to the fact that $\mathcal{F}(f * g) = \mathcal{F}f \cdot \mathcal{F}g$ it is clear that we have for every $f \in \boldsymbol{B}_I$:

$$f * \mathrm{sinc} = f. \tag{24}$$

Let us first try to understand the effect of sampling:

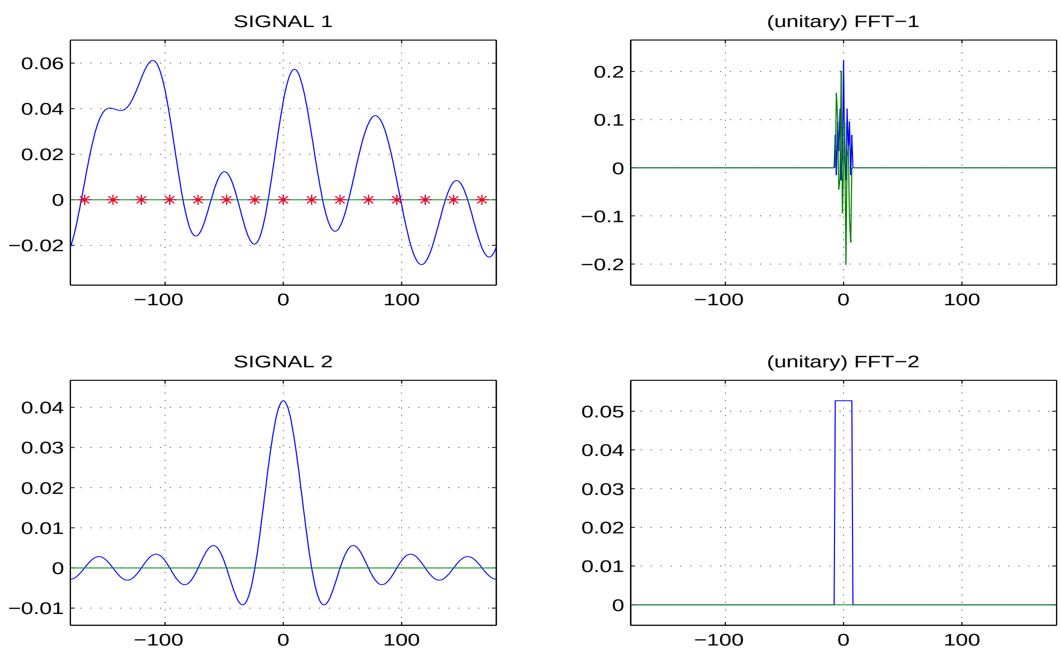


The image provided above shows the following situation. In the left upper corner we see a band-limited complex-valued function, whose real part is plotted in blue, the imaginary part is plotted in green. In the right upper corner one sees the Fourier transform. In fact, these Fourier coefficients have been obtained using a random number generator, with both real and imaginary part uniformly distributed with the interval $[-\beta, \beta]$, and normalized in the $\boldsymbol{L}^2$-norm. Clearly the length of the interval is (clearly smaller) than one quarter of the signal length, which appears to be 360 in this example.

[2]The word "cardinal" comes into the picture because of the *Lagrange type* interpolation property of the function $SINC$: $SINC(k) = \delta k, 0$.

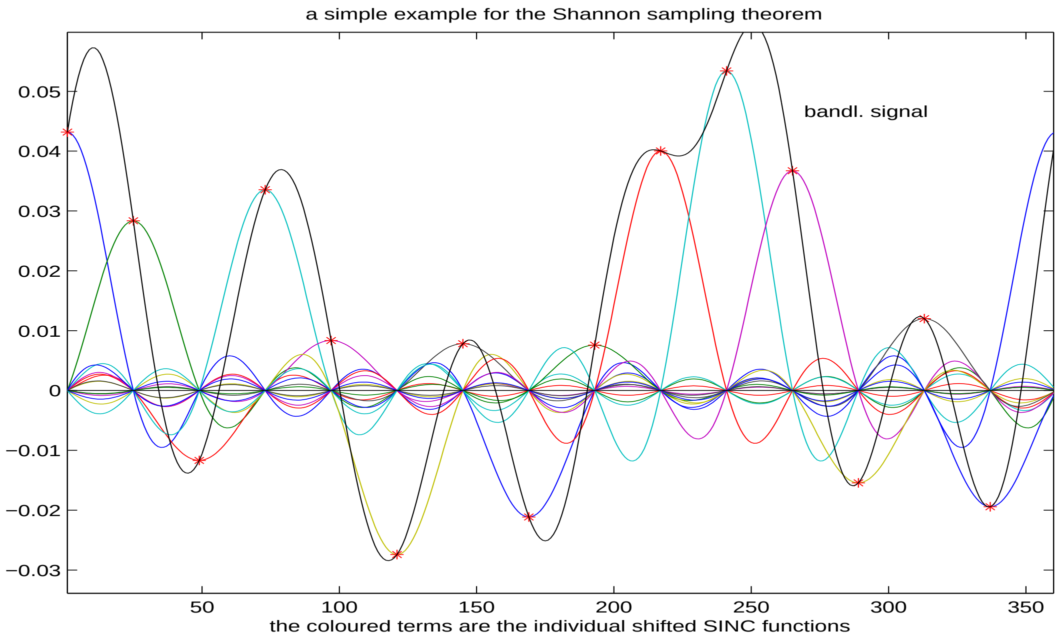
The sampling is carried out at a rate of 1/4, i.e. starting from the first coordinate (corresponding to the zero-point in $\mathbb{Z}_N$) we preserve every fourth value, while the intermediate (3 out of 4 values) are discarded, or more correctly, are put to zero. The effect can easily be seen on the Fourier transform, which is just the 4-periodic version of the original spectrum (Fourier transform). Also, in the given situation it is clear, that we have no overlap between the periodic copies of the spectrum.

The next plot shows a similar situation. Top row: The signal and its spectrum, concentrated on some interval. Also the regular sampling points are marked with red stars, and obviously one needs at least as many sampling points (point-evaluations) as one has unknown (complex) Fourier coefficients in the signal in order to be able to recover arbitrary signals with the given spectrum. The lower row shows the *ideal low-pass filter*, i.e. simply a box-function of appropriate size, symmetrically around zero in the frequency domain, and on the left side its inverse Fourier transform, which is a SINC-like (discrete) function.

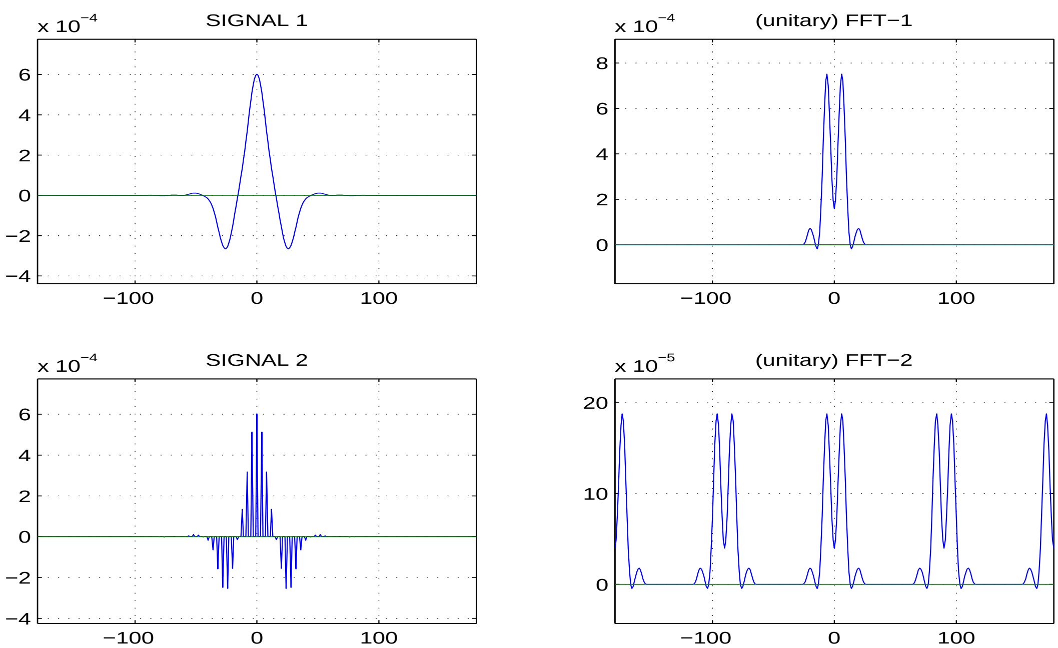


The next plot demonstrates how the band-limited (real-valued) functions is composed from shifted SINC-functions (positioned at exactly the regular sampling points, and with amplitudes which are identical with the sampling values).

Next we display the graph of some real-valued, band-limited function (in black, continuous line), which is sampled regularly, and with the corresponding



The following example is showing a similar example, but with some "natural/smooth" Fourier-transform, for better visualization on the FT-side.



12 Pure Mathematics and Engineering Applications

Summary. It is the purpose of this booklet to serve a dual purpose. On the one hand mathematicians should get to see how abstract concepts from harmonic analysis, combined with methods from numerical linear algebra allow to compute and visualize these concepts, to obtain numbers, to produce plots supporting the (geometric) intuition and further a deeper understanding of the abstract theory. On the other hand engineers should see, that one can arrive at the most important abstract concepts starting from ideas in signal processing and linear algebra (which at least communication engineers are studying quite carefully anyway). The abstract view-point helps to avoid discussions about continuous versus discrete, periodic versus non-periodic, finite versus infinite, or
Keywords. test

12.1 Subsection 1

See the books of Walker91 [12], or [13, 14] other citations

Literatur

[1] B. Bekka. Square integrable representations, von Neumann algebras and an application to Gabor analysis. *J. Fourier Anal. Appl.*, 10(4):325–349, 2004.

[2] H. Bölcskei and A. J. E. M. Janssen. Equivalence of two methods for constructing tight Gabor frames. *IEEE Signal Processing Letters*, 7(4):79–82, April 2000.

[3] S. Engelberg. *Digital signal processing. An experimental approach.* Signals and Communication Technology. London: Springer. xv, 212 p. EUR 79.95net, 2008.

[4] J.-P. Gabardo. Tight Gabor frames associated with non-separable lattices and the hyperbolic secant. *Acta Appl. Math.*, 107(1-3):49–73, 2009.

[5] J.-P. Gabardo and D. Han. Frames associated with measurable spaces. *Adv. Comput. Math.*, 18(2-4):127–147, 2003.

[6] D. Han. Tight frame approximation for multi-frames and super-frames. *J. Approximation Theory*, 129(1):78–93, 2004.

[7] D. Han. The existence of tight Gabor duals for Gabor frames and subspace Gabor frames. *J. Funct. Anal.*, 256(1):129–148, 2009.

[8] A. J. E. M. Janssen. On generating tight Gabor frames at critical density. *J. Fourier Anal. Appl.*, 9(2):175–214, 2003.

[9] A. J. E. M. Janssen and T. Strohmer. Characterization and computation of canonical tight windows for Gabor frames. *J. Fourier Anal. Appl.*, 8(1):1–28, January 2002.

[10] S. Qiu. Super-fast Computations of dual and tight Gabor atoms. In F. Laine, M. A. Unser, and V. Wickershauser, editors, *Proc. SPIE Wavelet Applications in Signal and Image Processing III*, volume 2569, pages 464–475, 1995.

[11] S. Qiu. Gabor-type matrix algebra and fast computations of dual and tight Gabor wavelets. *Optical Engineering*, 36(1):276–282, 1997.

[12] J. S. Walker. *Fast Fourier transforms. With floppy disc.* CRC Press, Boca Raton, FL, 1991.

[13] H. J. Weaver. *Applications of discrete and continuous Fourier analysis.* Wiley-Interscience, 1983.

[14] H. J. Weaver. *Theory of discrete and continuous Fourier analysis.* Wiley-Interscience, 1989.

12.2 Subsection 2

12.3 MATLAB CODE

```
VERBATIM PART
```

12.4 Exercises

1. first
2. second

13 DigSig II Material

Summary. TEXT
Keywords. test

13.1 Subsection 1

13.2 Subsection 2

13.3 MATLAB CODE

```
VERBATIM PART
```

13.4 Exercises

1. first
2. second

14 DigSig II Material

Summary. TEXT
Keywords. test

14.1 Subsection 1

14.2 Subsection 2

14.3 MATLAB CODE

A few short MATLAB codes

finding all proper divisors

```
function d=propdiv(n)

% PROPDIV   - Proper divisors of integer
% N. Kaiblinger, 2000
% Usage:      d=propdiv(n)
% Input:      n  integer
% Output:     d  vector of proper divisors of n

d = n./(n-1:-1:2);  d = d(d==round(d));
```

'cyclic shift operator'

```
function  RM = rot(M,a);
% a integer, M matrix, which is rotated in cyclic way

[hig,wid] = size(M);
RM  = M(:, rem( (wid:2*wid-1)-a ,wid) +1);
```

'Centralized adjusted plotting'

```
function plotc(xx);

% centered plot of single, complex-valued signal

 xx = xx(:).';  xx = rot(xx,round(u/2)-1);
 bas = 0:u ;  bas =  bas - round(u/2);
% adjusting the axes:
v3 = 1.1 * min(min(real(xx)),min(imag(xx)));
v4 = 1.2 * max(max(real(xx)),max(imag(xx)));
plot(bas,real(xx),bas,imag(xx));  grid;
axis([ bas(1) bas(u) v3 v4 ]);  % adjust axes
figure(gcf);  % get current figure: activates plot
```

14.4 Exercises

1. Use the M-file PROPDIV.M to find the number of divisors of n, for $n = 1, \ldots, 2520$.

2. Check catalogues of flatscreens, computer screens, notebook displays, mobile phones etc. for the format of the display and the corresponding number;

3. Find the prime factorization and number of divisors of the sampling rate for CDs, which is 44100 samples per second (Answer: 79 proper divisors); also in connection with the Shannon sampling theorem we will come to the conclusion that something in the order of 40.000 + 10% is a reasonable *sampling rate* in case one wants to reproduce band-limited signals of maximal frequency of 20 kHz (the maximal frequency a person ever can hear), plus some oversampling (to allow alternative reconstructions aside from the classical Shannon-sampling theorem).

15 Recalling concepts from linear algebra

Summary. TEXT
Keywords. test

15.1 Subsection 1

15.2 Subsection 2

15.3 MATLAB CODE

```
VERBATIM PART
```

15.4 Exercises

1. first

2. second

16 DigSig II Material

Summary. TEXT **Keywords.** test

16.1 Subsection 1

16.2 Subsection 2

16.3 MATLAB CODE

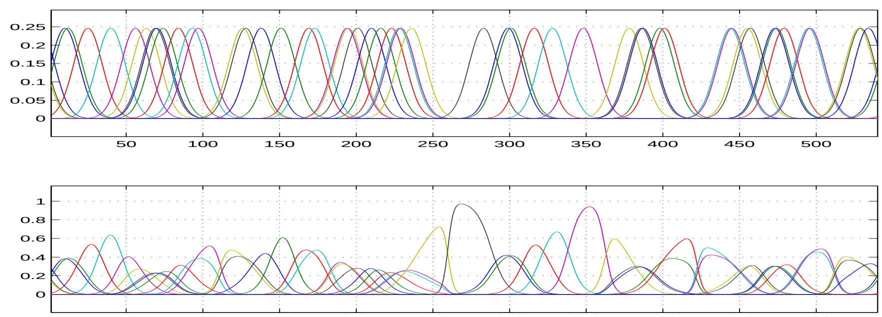
```
VERBATIM PART
```

16.4 Exercises

1. first
2. second

17 Summability kernels

Summary. The functions in the Segal algebra $\boldsymbol{S}_0(\mathbb{R})$ are all suitable summability kernels in the sense of classical Fourier Transform theory. This means, that one can use them to recover the function $f(t)$ from its Fourier transform $\hat{f}(s)$, which is known to be continuous and bounded. By pointwise multiplication with a dilated version $D_\rho h$ of such a summability kernel $h(s)$, which has to satisfy $h(0) = 1$ (and typically with $h(x) \geq 0$) allows to guarantee the applicability of the Fourier inversion formula to $D_\rho h \cdot \hat{f}$, which obviously tends to $\hat{f}$ (in some sense). Since each good summability kernel (all the ones listed below) are of the form $h = \hat{g}$, for some other "nice" summability kernel $g \in \boldsymbol{L}^1(\mathbb{R})$, with $\int_{\mathbb{R}} g(t)dt = 1$, this can be seen from the *Dirac property* of the dilated family $(St_\rho g)_{\rho \to 0}$, which forms an *approximate unit* with respect to convolution. Mathematical/technical details will be given separately.



17.1 Subsection 1

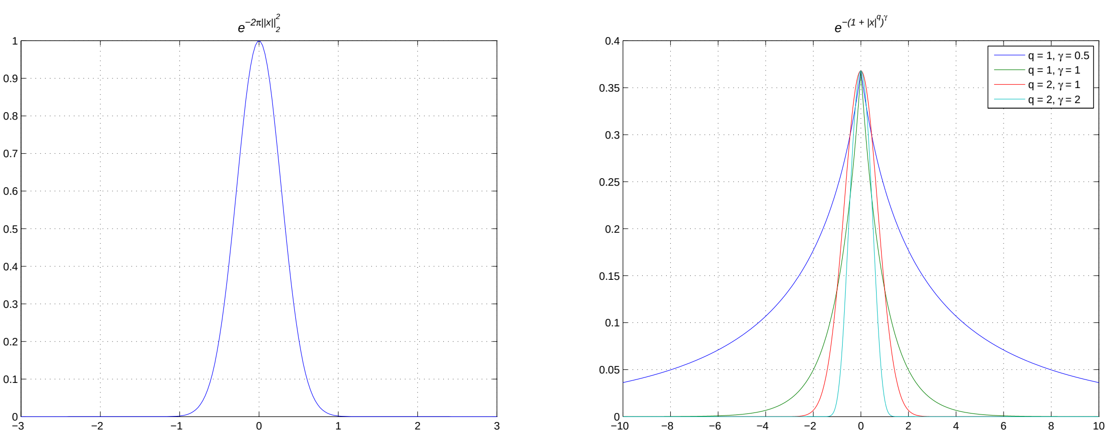


Abbildung 1: two figures

Beethoven Piano Sonata:

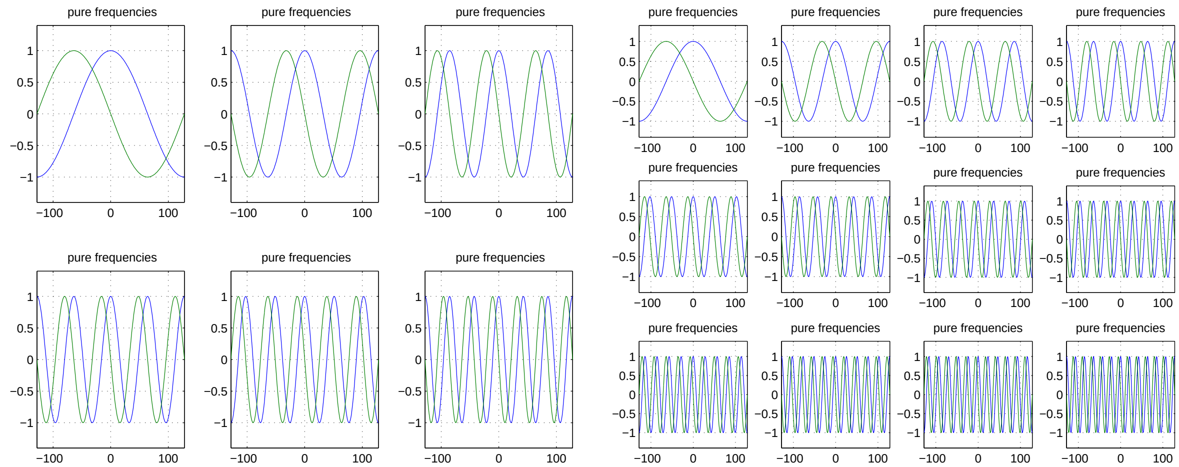
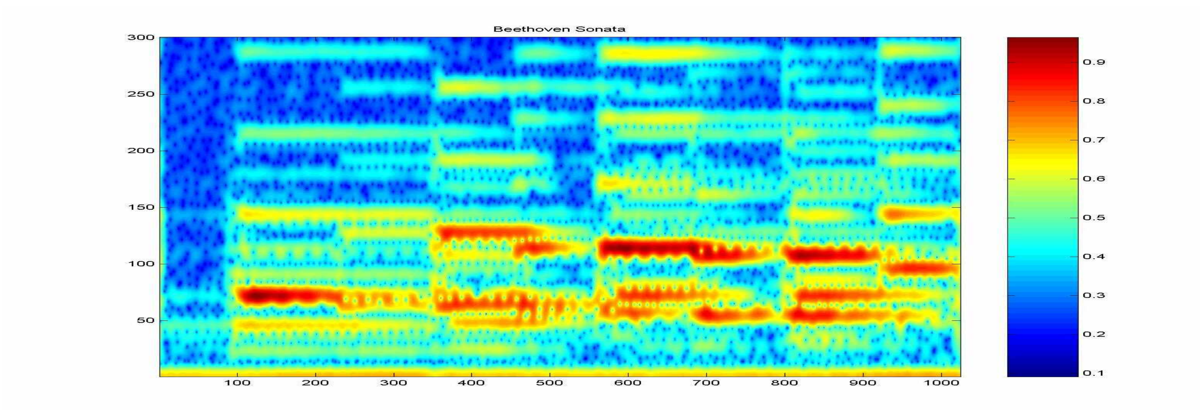
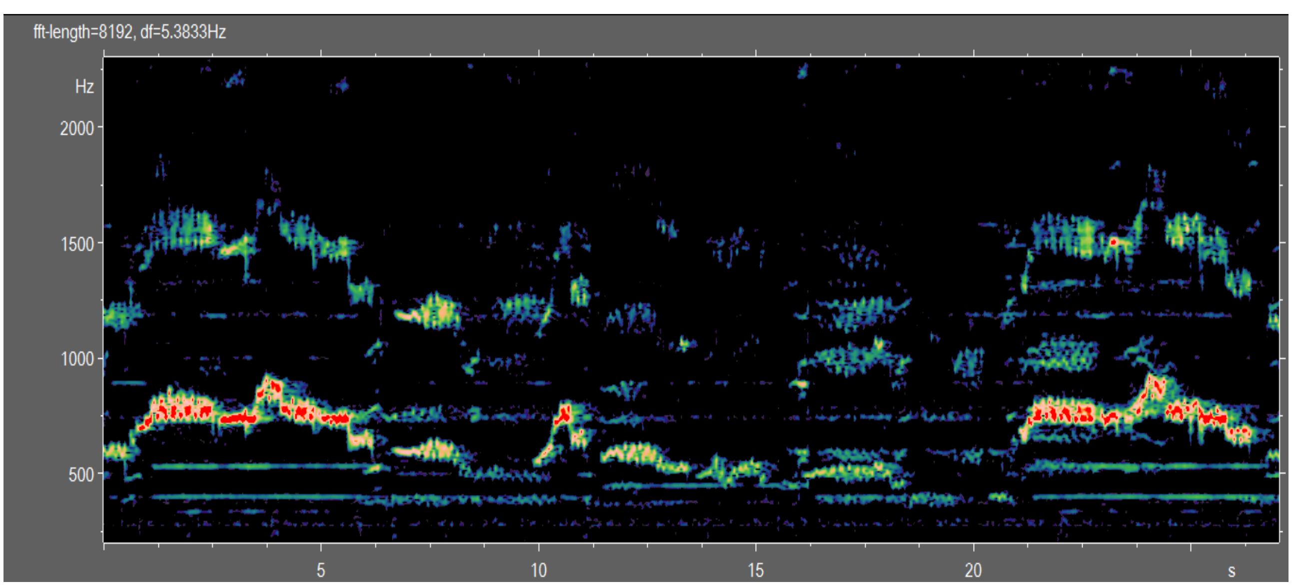
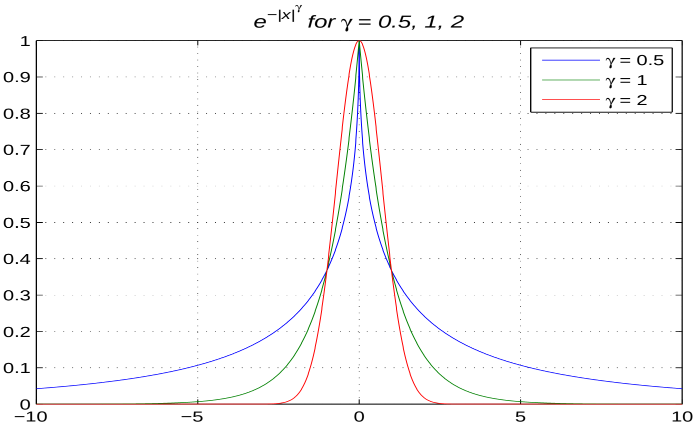


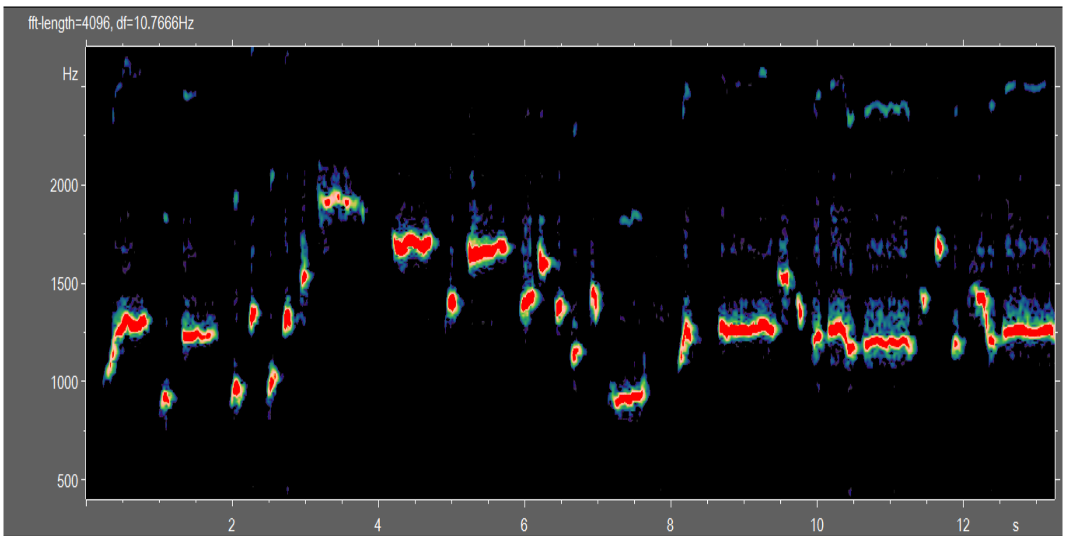
Abbildung 2: The collection of first pure frequencies: blue shows real part = cosine [even function] and green curves = sine-component = imaginary part: odd function.



Recording: Anna Bolena Premiere, Vienna State Opera




$e^{-|x|^{\gamma}}$ for γ = 0.5, 1, 2
γ = 0.5
γ = 1
γ = 2
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
−10
−5
0
5
10


fft-length=4096, df=10.7666Hz
Hz
2000
1500
1000
500
2
4
6
8
10
12
s

17.2 Subsection 2

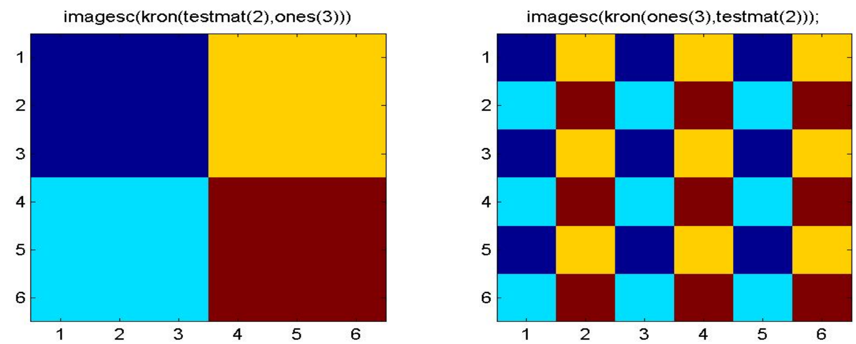
17.3 MATLAB CODE

```
VERBATIM PART
```

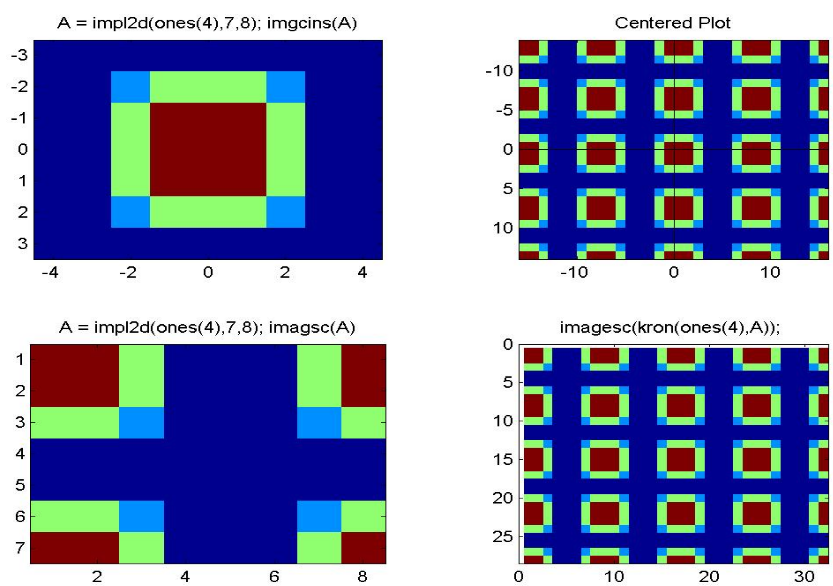
17.4 Exercises

1. first
2. second

Demo for simple Kronecker product of matrices



Demo for `imgc` and `imgcins` $2D-$display options



My favorite Test image

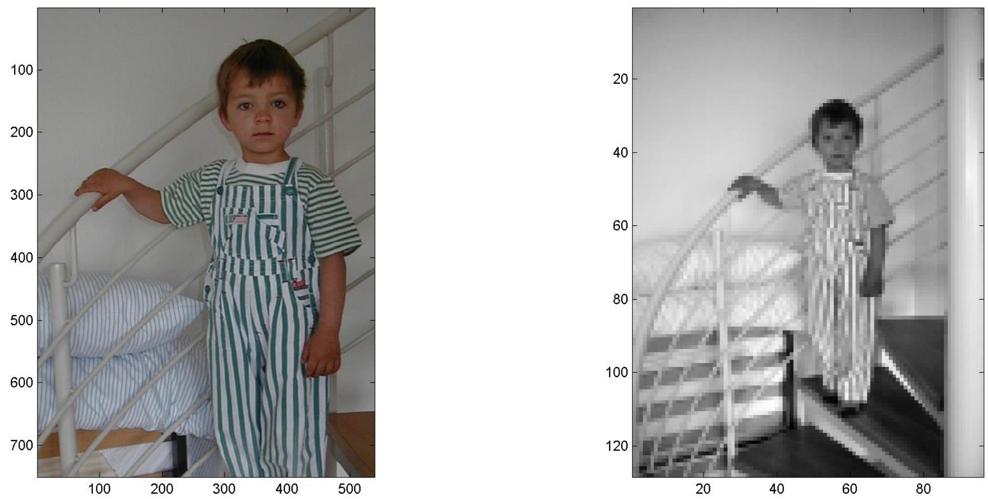
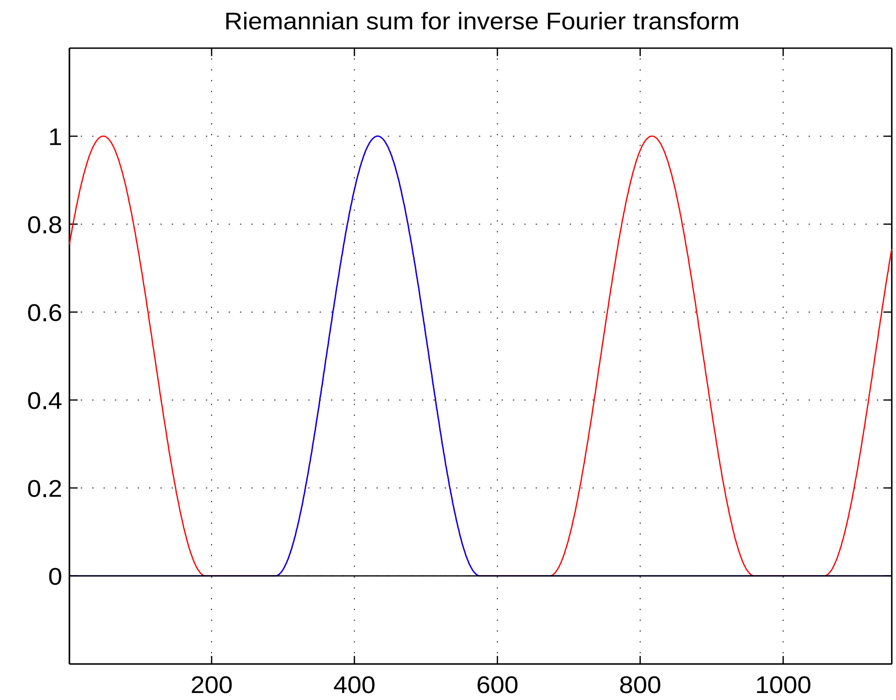
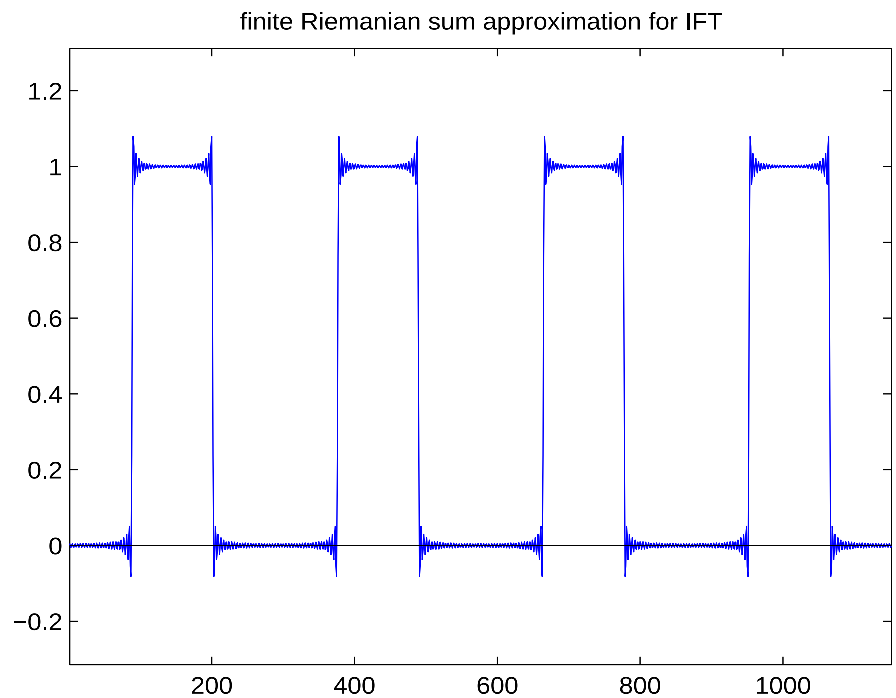


Abbildung 3: Original image and coarse resolution approximation: “pixelized”

Fourier Approximation and Gibb's phenomenon



High order Fourier approximation to box-function (periodic version)



18 Tight Gabor Frames: featured articles

Material related to the construction of tight Gabor frames:

Featured articles for SS 2011: (Monika, Maurice, etc.)

[Gabardo, Jean-Pierre;Han, Deguang] Frames associated with measurable spaces. [5]

D.Han [Han, Deguang] The existence of tight Gabor duals for Gabor frames and subspace Gabor frames [7]

[Han, Deguang] Tight frame approximation for multi-frames and super-frames. [6]

Bekka [1],

[Gabardo, Jean-Pierre] Tight Gabor frames associated with non-separable lattices and the hyperbolic secant. [4]

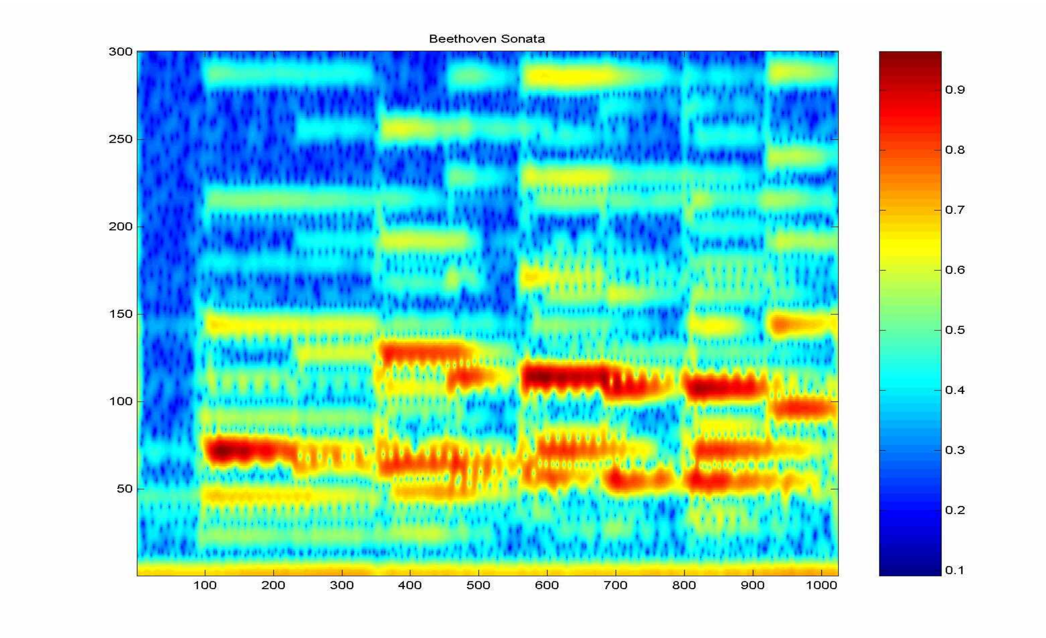
[2] as a justification of the double-preconditioning method (still not verified theoretically)

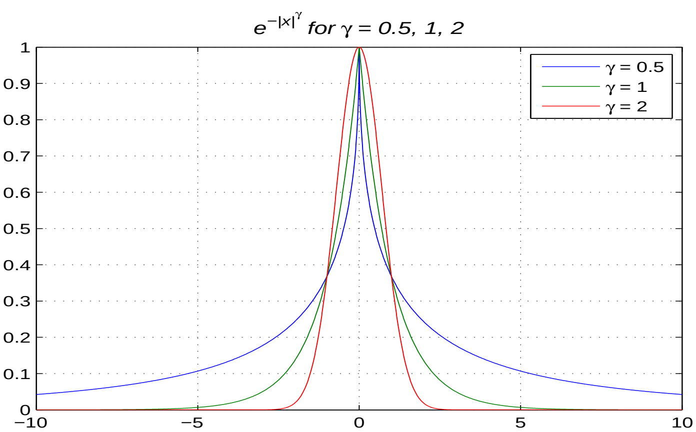
[Janssen, A. J. E. M.;Strohmer, Thomas] Characterization and computation of canonical tight windows for Gabor frames. [9] [Qiu, Sigang] Gabor-type matrix algebra and fast computations of dual and tight Gabor wavelets [11]

[Qiu, Sigang;] Super-fast Computations of dual and tight Gabor atoms [10] [Janssen, A. J. E. M.;Strohmer, Thomas] Characterization and computation of canonical tight windows for Gabor frames. [9]

[Janssen, A. J. E. M.] On generating tight Gabor frames at critical density. [8]

Are there multiple tight frames?? (i.e. for two different TF-lattices $\Lambda \lhd \mathbb{R}^{2d}$)





ENDE !!!

TEST EPS-file

19 Hermite functions and their TF-concentration

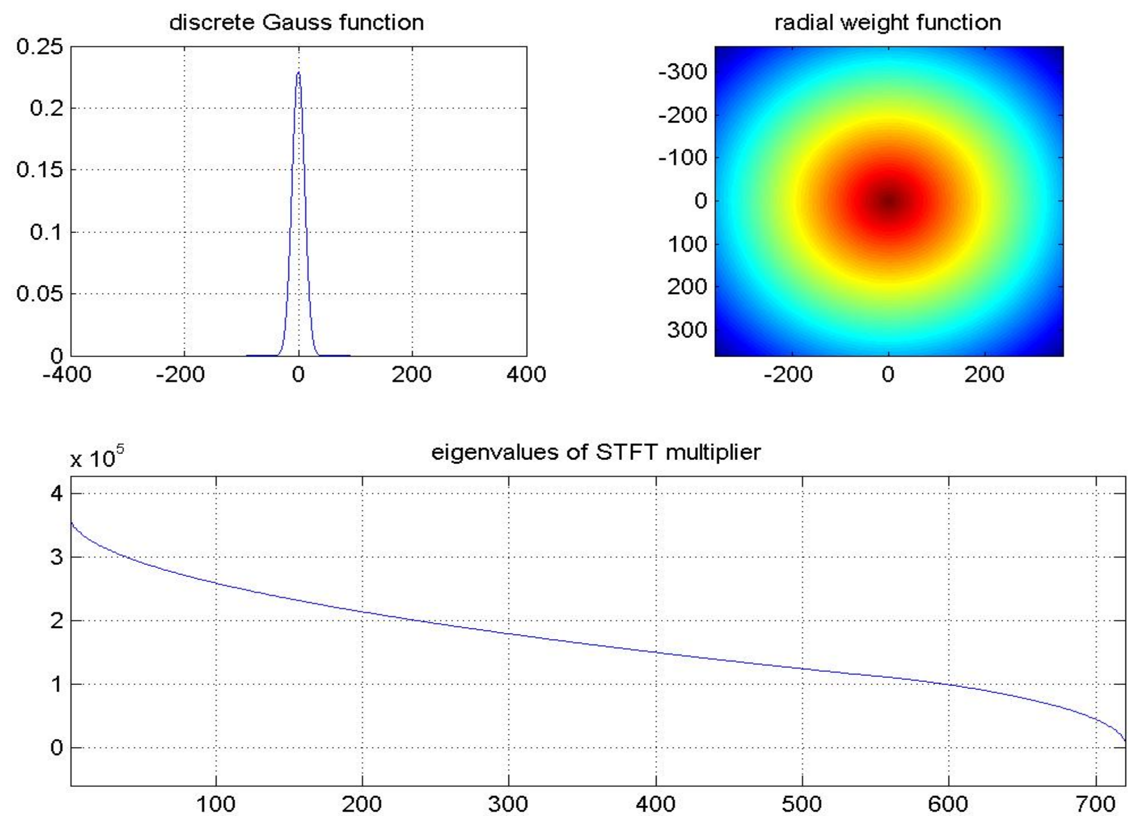
Hermite function are eigenvectors for the Fourier transform. They can be generated numerically as eigenvectors of certain STFT-multipliers, with Gaussian analysis and synthesis window, and (more or less) an arbitrary radial symmetric weight function over (the discrete) TF-plane.

The NuHAG toolbox contains the following functions

`radwgh.m : creating a the radial distance function`

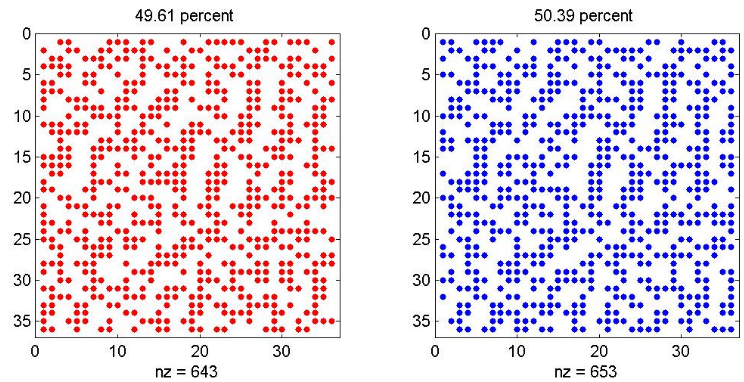
`gabmulmh: gabmulmh(W,g); Gabor multiplier, with weight` W `and window` g

`gaussnk: Gauss-function (discrete, sampled and periodized)`



20 LEFT over Material

gameliv4



THE BIBLIOGRAPHY is NOT RELEVANT for the content of the previous manuscript so far!

Literatur

[1] B. Bekka. Square integrable representations, von Neumann algebras and an application to Gabor analysis. *J. Fourier Anal. Appl.*, 10(4):325–349, 2004.

[2] H. Bölcskei and A. J. E. M. Janssen. Equivalence of two methods for constructing tight Gabor frames. *IEEE Signal Processing Letters*, 7(4):79–82, April 2000.

[3] S. Engelberg. *Digital signal processing. An experimental approach.* Signals and Communication Technology. London: Springer. xv, 212 p. EUR 79.95net, 2008.

[4] J.-P. Gabardo. Tight Gabor frames associated with non-separable lattices and the hyperbolic secant. *Acta Appl. Math.*, 107(1-3):49–73, 2009.

[5] J.-P. Gabardo and D. Han. Frames associated with measurable spaces. *Adv. Comput. Math.*, 18(2-4):127–147, 2003.

[6] D. Han. Tight frame approximation for multi-frames and super-frames. *J. Approximation Theory*, 129(1):78–93, 2004.

[7] D. Han. The existence of tight Gabor duals for Gabor frames and subspace Gabor frames. *J. Funct. Anal.*, 256(1):129–148, 2009.

[8] A. J. E. M. Janssen. On generating tight Gabor frames at critical density. *J. Fourier Anal. Appl.*, 9(2):175–214, 2003.

[9] A. J. E. M. Janssen and T. Strohmer. Characterization and computation of canonical tight windows for Gabor frames. *J. Fourier Anal. Appl.*, 8(1):1–28, January 2002.

[10] S. Qiu. Super-fast Computations of dual and tight Gabor atoms. In F. Laine, M. A. Unser, and V. Wickershauser, editors, *Proc. SPIE Wavelet Applications in Signal and Image Processing III*, volume 2569, pages 464–475, 1995.

[11] S. Qiu. Gabor-type matrix algebra and fast computations of dual and tight Gabor wavelets. *Optical Engineering*, 36(1):276–282, 1997.

[12] J. S. Walker. *Fast Fourier transforms. With floppy disc.* CRC Press, Boca Raton, FL, 1991.

[13] H. J. Weaver. *Applications of discrete and continuous Fourier analysis.* Wiley-Interscience, 1983.

[14] H. J. Weaver. *Theory of discrete and continuous Fourier analysis.* Wiley-Interscience, 1989.