NuHAG Numerical Harmonic Analysis Group

Fourier Analysis:
A MATLAB based Approach with Applications

Hans G. Feichtinger
WEBPAGE: **www.nuhag.eu**
http://www.nuhag.eu/talks
http://www.nuhag.eu/bibtex

ENGLISH VERSION, revision of Oct. 2018, Prague

München, May 10th, 2016
MATLAB EXPO 2016

Overview

- Personal background
- Background and historical overview
- MATLAB and Linear Algebra
- From Linear Algebra to DFT/FFT
- Polynomials and the DFT/FFT
- Applications of the FFT, TILS
- Unconventional Applications
- Time-Frequency Analysis and Gabor Analysis
- Links and Internet Resources
- Summary

Short rsum of Hans G. Feichtinger

- Teaching position in Mathematics and Physics, Univ. of Vienna;
- Habilitation in Mathematics 1979 in Vienna;
- Operating at the Univ. of Vienna until retirement (Dec. 2015);
- Visiting professor in many places around the world;
- 1992: Founder of the Numerical Harmonic Analysis Group NuHAG (www.nuhag.eu) in Vienna;
- Since 1979 habilitated (Univ. of Vienna, Mathematics), 27 PhD students in Mathematical Genealogy;
- Since 2000 Chief Editor of J. Fourier Analysis & Appl.;
- Numerous theoretical and applied projects.

HGFei: Research Profile

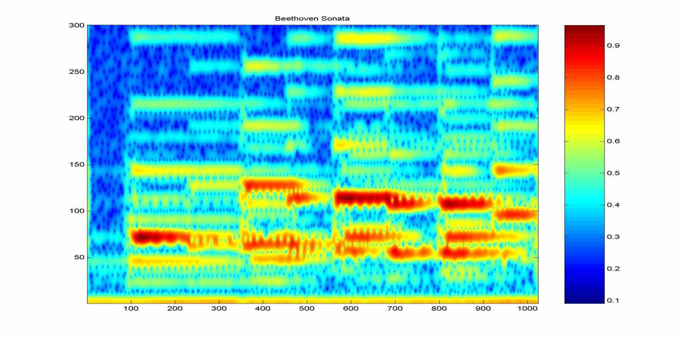
I describe myself as an **application-oriented mathematician** with high interest in the fundamentals of Signal and Image Processing (that are applied in daily life as WAV-files, JPEG-Photos, MP3-Data).

Specifically, I see myself as an **application-oriented representative of Computational Harmonic Analysis**, who develops and shares concepts and algorithms.

I am particularly interested in the notion of Conceptual Harmonic Analysis, that includes distribution theory but also Numerical Fourier Analysis. Gabor Analysis and Functional Spaces are my areas of expertise.

Gabor Analysis: Beethoven Piano Sonata

Gabor Analysis is the mathematical theory of spectrograms: Which window, which possible discretization, etc.. To illustrate: the spectrogram of a Beethoven Piano Sonata!



Where and why is the Fourier Transform important?

The Fourier Transform (FT), in particular the Discrete Fourier Transform (DFT) implemented in the Fast Fourier Transform (FFT), is indispensable in our digital age. Most of the algorithms in digital Signal or Image Processing are based on the FFT and FFT2 respectively. Engineering students are often faced with the following concepts:

1. Linear translation-invariant Systems (LTIS);
2. Impulse response and transfer function;
3. Convolution, Fourier Transform,
4. Sampling and Shannon's Theorem,

just to name a few.

Scholastic Multiplication and System Theory

We want to introduce a form of "translation-invariant systems" that everyone recognizes from school: the MULTIPLICATION with natural numbers, in this case with 123.

Consider the following problem and its obvious solution by a smart pupil, who does not stubbornly follow the general formula and makes the following calculations: 450002 x 123 = 450000 x
123 + 2 x 123
= (45 x 123) x 10000 + 246
= 55350000 + 246 = 55350246
Additive splitting $(450000 + 2)$ and moving the decimal point (before/after) are obviously used. It suffices to know that
1 x 123 = 123 in order to know how to multiply any $n \in \mathbb{N}$.

Multiplication and Convolution

A similar multiplication exercise is:

$$1111110^2 = 1234565432100$$

One can easily see that the structure of multiplication is the same in practice as finding the coefficients of the polynomial $p(x)^2$, where $p(x)$ is given as:

$$p(x) = x^6 + x^5 + x^4 + x^3 + x^2 + x,$$

and we obtain:

$$p(x)^2 = x^{12} + 2x^{11} + 3x^{10} \cdots + 4x^5 + 3x^4 + 2x^3 + x^2.$$

The coefficients of this polynomial are related to the probability of the sum of two dices (expressed in 1/36)!

The Classical Fourier Program

1. J. B. Fourier, 1822: *Every periodic function can be represented as an infinite sum of "harmonic waves"*: $f(t) = \sum_{k \in \mathbb{Z}} c_k e^{2\pi i k t}$ (Fourier Series);
2. The question of convergence is clarified $>>$ Lebesgue Integral;
3. Sufficient theory for the Hilbert Space $\left(\boldsymbol{L}^2(\mathbb{T}), \|\cdot\|_2\right)$;
4. Transition to infinite periods (continuous frequency spectrum) $>>$ *Fourier Transform*;
5. Fourier Transform and Inversion formula for $\boldsymbol{L}^1(\mathbb{R}^d)$; Plancherel's Theorem: $\|f\|_2 = \|\hat{f}\|_2$;
6. L. Schwartz implemented the FT for tempered distributions;
7. DFT = Discrete Fourier Transf., FFT algorithm by Cooley-Tuckey (1965).

Definition of the Discrete Fourier Tranform (DFT)

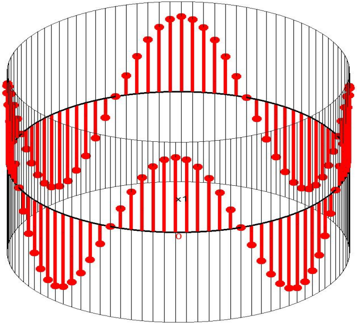
We write $f = [f(0), \cdots, f(L-1)] \in \mathbb{C}^L$ (complex vectors of length L), and we regard f as a **periodic, discrete signal** (of period L), with $i = \sqrt{-1}$ as usual:

$$\widehat{f}(k) = \sum_{l=0}^{L-1} f(l) e^{-\frac{2\pi ik\cdot l}{L}} \qquad k = 0, \cdots, L-1 \tag{1}$$

$$f(n) = \frac{1}{L} \sum_{l=0}^{L-1} \widehat{f}(l) e^{\frac{2\pi in\cdot l}{L}} \qquad n = 0, \cdots, L-1 \tag{2}$$

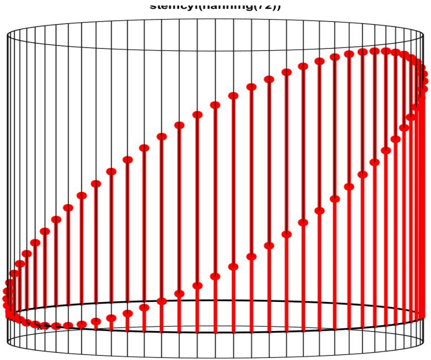
Illustration of a Hanning Window

Representation of a periodic series via STEMCYL (NuHAG M-file).
It is also a reason for using cyclic shifts (rotation!).



What does the Indexing mean (zeroth component?)

Side note: This type of representation, that permits rotation and tilting of the object, allows for interesting demonstrations. For instance, a Hanning window has a simple representation: `hanning(72)` is shown:



The DFT in MATLAB description

Since vectors do not have a "zeroth component", it is better (in case one wants to use the MATLAB formula for the DFT) to utilize the following convention: For $\mathbf{x} = [x[1], \cdots, x[L]] \in \mathbb{C}^L$ the DFT (Discrete Fourier Transform) $\mathbf{y} = DFT(\mathbf{x})$ is defined as:

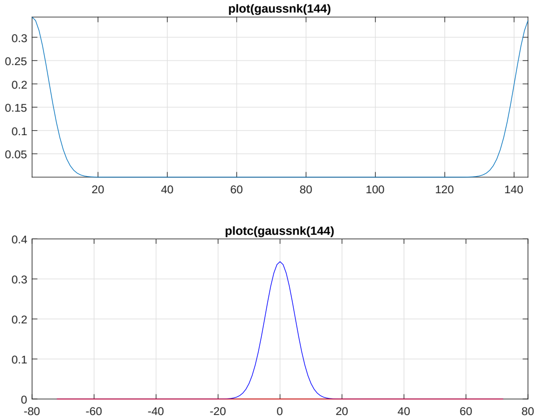
$$y[k] = \sum_{l=1}^{L} x[l] e^{-\frac{2\pi i (k-1)\cdot(l-1)}{L}} \qquad k = 1, \cdots, L. \tag{3}$$

$$x[n] = \frac{1}{L} \sum_{k=1}^{L} y[l] e^{\frac{2\pi i (n-1)\cdot(k-1)}{L}} \qquad n = 1, \cdots, L. \tag{4}$$

ADVANTAGE: Correct indexing, but it is hard to interpret the coefficients! There is also a plotting problem!

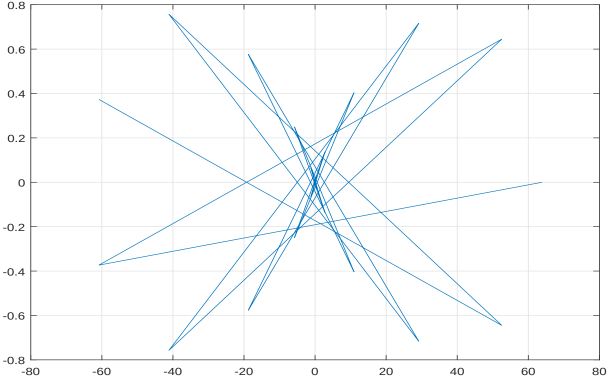
Correct Plotting of a Centered Gaussian Function

The "right" data structure (this discrete Gaussian function is FFT invariant) and its "correct" Plot view (PLOTC.M).



Discrete, Fourier-invariant Gaussian Function

The naive sequence of commands cannot but create confusion:
$bas = linspace(-4, 4, 512); gdisc = exp(-pi * bas.^2); plot(fft(gdisc))$



Alternatives to the Classical Program?

I claim that **MATLAB** offers the possibility *from the side of Linear Algebra* to delve in the main aspects of Fourier Analysis in an application-oriented as well as mathematically correct way.

Just as real numbers can be described as limits of rational numbers, so too should continuous theory be seen as a limit of discrete signals (exactly like pixel images are a good approximation of continuous photos!).

Let us start with **facts from Linear Algebra** (for complex-valued vectors and matrices)!

Linear Algebra and MATLAB I

Let us recall the basic concepts of linear algebra and their implementation using MATLAB, namely the role of matrices and matrix multiplication:

Definition

A vector $\mathbf{b}$ is a linear combination, meaning of the form $\mathbf{b} = x_1\mathbf{a}_1 + \cdots x_n\mathbf{a}_n$ is equivalent with $\mathbf{b} = \boldsymbol{A} * \mathbf{x}$, where $(\mathbf{a}_k)_{k=1}^n$ are the columns of $\boldsymbol{A}$ in $\mathbb{C}^m$ (or $\mathbb{R}^m$).

Accordingly, linear combinations of linear combinations are again linear combinations (the original vectors) and $\boldsymbol{B} = \boldsymbol{A}_2 * \boldsymbol{A}_1$ tells us which coefficients one needs. The explicit coefficients for $\mathbf{b}$ are given by $\mathbf{x} = \boldsymbol{A}^{-1} * \mathbf{b}$.

Linear Algebra and MATLAB II

Definition

A mapping T from $\mathbb{C}^n$ into $\mathbb{C}^m$ is called *linear*, if it preserves linear combinations, namely

$$T\left(\sum_k c_k \mathbf{b}_k\right) = \sum_k c_k T(\mathbf{b}_k).$$

Consequently, a linear mapping is known as soon as it is known for the basis elements and it is easy to describe T via a matrix that contains the image of basis vectors, in other words via a $m \times n$ matrix, with $\mathbf{a}_k = T(\mathbf{e}_k), 1 \leq k \leq n$. **Exactly like matrix composition and inversion!**

Linear Algebra and MATLAB III

There is no command in MATLAB for scalar product (in $\mathbb{C}^n$), because $\langle \mathbf{x}, \mathbf{y} \rangle$ can anyway be obtained by $\mathbf{y}' * \mathbf{x}$. The known formula

$$\langle \boldsymbol{A} * \mathbf{x}, \mathbf{y} \rangle = \langle \mathbf{x}, \boldsymbol{A}' * \mathbf{y} \rangle$$

follows from the associativity of matrix multiplication and the known rule:

$$(\boldsymbol{A} * \boldsymbol{B})' = \boldsymbol{B}' * \boldsymbol{A}'.$$

In particular $\mathbf{y} \mapsto \boldsymbol{A}' * \mathbf{y}$ for $\mathbf{y} \in \mathbb{C}^m$ is equivalent to

$$\mathbf{y} \mapsto (\langle \mathbf{y}, \mathbf{a}_k \rangle)_{k=1}^{n} \, .$$

Linear Algebra and MATLAB IV

Why are unitary matrices useful? By using them we can conclude (with a little theory concerning inverse elements) that an $n \times n$ matrix $\mathbf{U}$, whose columns constitute an orthonormal system (which means $\mathbf{U}' * \mathbf{U} = \mathrm{Id}_{\mathbf{n}}$), also fulfils the following:

$$\mathbf{U} * \mathbf{U}' * \mathbf{x} = \mathbf{U} * (\mathbf{U}' * \mathbf{x}) = \mathbf{x}, \quad \mathbf{x} \in \mathbb{C}^n$$

(respectively $\mathbf{U}^{-1} = \mathbf{U}'$!), therefore it follows that $\mathbf{x} \in \mathbb{C}^n$:

$$\mathbf{x} = \sum_{k=1}^{n} \langle \mathbf{x}, \mathbf{u}_k \rangle \mathbf{u}_k .$$

In other words: The unambiguous coefficients with respect to the base $(\mathbf{u}_k)$ are just the scalar products with these vectors!

Linear Algebra and MATLAB V

Example: We consider the mapping, which assigns to a given sequence of coefficients $\vec{a}$ of a polynomial (in accordance with the MATLAB conventions) the values at a sequence of points z_k in $\mathbb{C}$ via the command `polyval(a,z)` which realizes

$$p(z) = a_1 z^{n-1} + a_2 z^{n-2} \ldots a_n.$$

This mapping is invertible only if we have exactly n distinct points $z_1, ..., z_n$. Since the matrix of this mapping is the Vandermonde Matrix `vander(z)` for $\vec{z} \in \mathbb{C}^n$

$$\vec{d} = polyval(a, z) = vander(z) * a$$

and consequently

$$a = inv(vander(z)) * d.$$

From Linear Algebra to the Fourier Transform

Our main point now will be to illustrate how one can explain the basic properties of the Fourier Transform using *only experiments involving the* FFT (Fast FT) and some basic linear algebra arguments (mainly matrix calculations).

Because the *output* of the FFT-Routine is a complex sequence, we should regard the FFT as a mapping from $\mathbb{C}^N$ to $\mathbb{C}^N$ (clearly the output format is the same format as the input!), as this easy TEST shows: `fft(rand(1,5))`, `fft(rand(4,1))`,

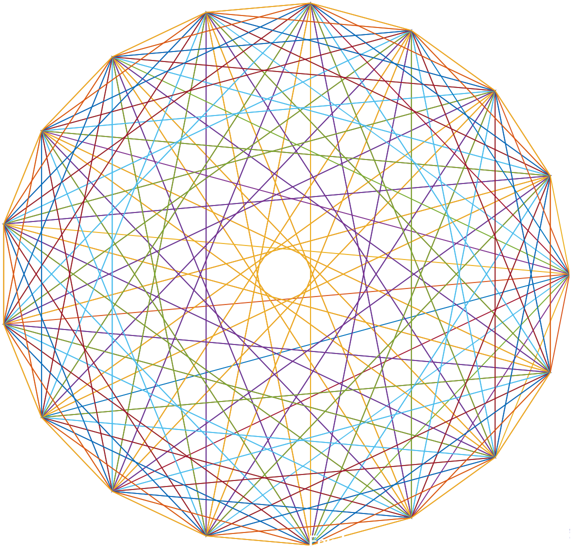
`fft(rand(3,4))`,

From Linear Algebra to the Fourier Transform II

1. **The FFT preserves the (Euclidean) norm of vectors**, let alone a factor $\sqrt{N}$ that depends on the length.
   TEST: `x = rand(N,1);`
   `norm(fft(x))/(sqrt(N)*norm(x)),`
2. **the FFT is linear**, which is "verified" by the following test: `A = rand(3,4); x = rand(4,1);`
   `norm(fft(A)*x - fft(A*x)),`
   In accordance with the conventions **the FFT operates on a matrix column by column**. That means the test above shows that the FFT routine respects random linear combinations!
3. `F = fft(eye(4)), norm(fft(x) - F * x)`
   shows that the matrix multiplication with F truly equates to the application of the FFT routine.

Illustration of the FFT Matrix of Size 17

FFT-Matrix of size 17



The DFT Matrix as a Vandermonde Matrix

The previous picture suggests that the Nth roots of unity have a special role, and MATLAB helps us to verify that the mapping $\mathbf{x} \mapsto \mathrm{fft}(\mathbf{x})$ can be interpreted as the evaluation of a polynomial with coefficients $\vec{a}$ at the unity roots of order N.

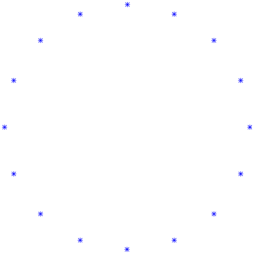
Still one has to ask according to which convention the polynomial (of degree $L-1$) from the coefficient sequence $\vec{a}$ is generated (of length N), and in which order the Nth roots of unity are to be specified to achieve exact (numerical) agreement. MATLAB is using the coefficients in decreasing order, while standard mathematical terminology starts from the constant term (usually called a_0).

The DFT Matrix as a Vandermonde Matrix II

This statement is easily verifiable in MATLAB:

```
N = 16; bas = 0 : 1/N : (N-1)/N; u = exp(2 * pi * i * bas)
plot(u,'b*'); axis square; axis off; shg; norm( polyval(fl
- fft(aa))  = approx. zero!
```

For illustration:



The DFT Matrix as a Vandermonde Matrix III

It should be noted: The Vandermonde routine is adjusted to the MATLAB convention for the command polyval. Therefore, the monomials are listed in ascending order and not in traditional mathematical mapping of polynomials.

Accordingly one can identify the associated matrices:

```
>> u5 = exp(2 * pi * i * (0:1/5:4/5)); >> V5 = vander(u5);
V5g = V5(:,[5,1:4]); >> norm(V5g - (fft(eye(5))) ) ans
= 1.9239e-15
```

SUMMARY: The FFT-Matrix is in fact a Vandermonde Matrix corresponding to the sequence of unit roots of order N, starting at $1 = \omega_N^0$ (then $\omega_N^{-k}, k \geq 1$), with $\omega = e^{-2\pi i/N}$, thus taken in the CLOCKWISE sense.

The DFT Matrix as a Vandermonde Matrix IV

Alternatively, we take a look at the Vandermonde Matrix for $N = 5$, with the unity roots arranged clockwise:

```
>> u5 = exp(-2 * pi * i * (0:1/5:4/5));
```

Adapted due to the MATLAB conventions regarding polyval (adjusted using fliplr) the following can be stated:

```
>> VC5 = vander(u5); F5 = fft(eye(5));
>> norm(VC5 - fliplr(F5))    ans = 1.9239e-15
```

That means: After reversing the arrangement of the columns the FFT Matrix equates to the Vandermonde Matrix of the unity roots, starting at $1 = \omega_N^0$, taken in the clockwisely sense.

From Linear Algebra to the Fourier Transform III

Because $x \mapsto fft(x)$ is a linear function, we can further analyze the affiliated matrix:

```
N = 17; F17 = fft(eye(N)); plot(F17); axis square;
```

This matrix is symmetrical: `>> norm(F - F.', 'fro'), ans = 1.4316e-14`

On the other hand, it is interesting to look at the entries of this matrix one column at a time. One example: $N = 144$: `>> N = 144; F = fft(eye(N));`

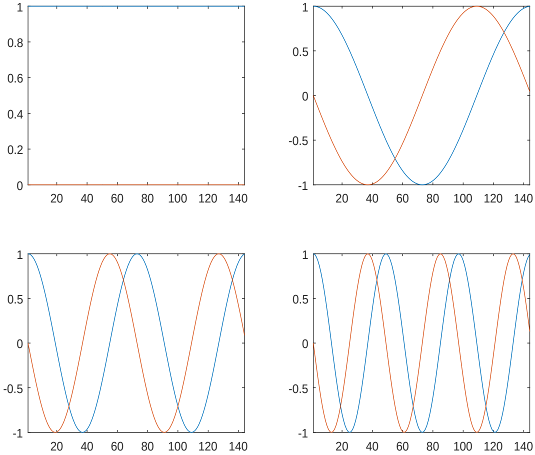
```
>> for jj=1:6; plot(1:N,real(F(:,jj)),1:N,
imag(F(:,jj))); shg; pause; end;
```

The columns of the matrix all have the same length (: $\sqrt{N}$)

```
for jj=1:N; lF(jj)=norm(F(:,jj)); end;
norm(ones(1,N)*sqrt(N) - lF),
```

The first columns of the Fourier Matrix

The columns of the Fourier Matrix: The pure frequencies!



Pure Frequencies and Fourier Synthesis

The columns of the Fourier Matrix (as seen above) only contain the so called ”pure frequencies”.
According to the **Eulerian formula** they can be expressed as follows (with $j = \sqrt{-1}$ afor engineers!):

$$\chi_k(t) := e^{2\pi jkt} = \cos(2\pi kt) + j\sin(2\pi kt).$$

These are “natural building blocks”, because they are **Eigen vectors for the translation operators** with complex eigenvalues, due to the exponential law:

$$[T_x\chi_k](t) := \chi_k(t-x) = e^{2\pi jk(t-x)} = e^{-2\pi jkx} \cdot \chi_k(t).$$

In other words: On the so-called characters χ_k the action of T_x is simply multiplication by the complex number $e^{-2\pi jkx}$.

Orthogonality of the Fourier Matrix columns

One can hope that the FFT-Routine is an **orthogonal change of basis**. As a reminder: `A = rand(N); U = orth(A);`

```
norm(U'*U - eye(N)), norm(U*U' - eye(N)),
```

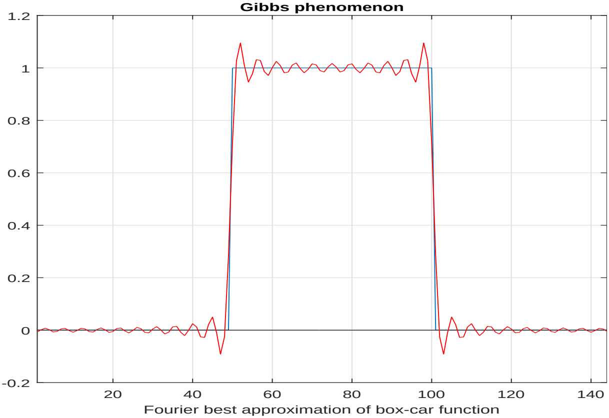
Because the columns of F (and therefore also the columns of $F.' = F$) all have the same length of $\sqrt{N}$, one can expect that **except for the scaling factor N the matrix F' is the inverse of** F: `norm(F*F'/N - eye(N))`

The best approximation of certain frequencies, where J is a subset of $1:N$:

```
bx = zeros(1,N); bx(50:100) = 1; plot(bx);
ax20; grid; plotax;
```

Fourier Approximation of Box-car function

The best approximation of the boxcar function, using 61 Fourier coefficients (in the Euclidean sense) (max. frequency 30):



Pure Frequencies and Fourier Synthesis

$$\mathbf{x} = F' * (F * \mathbf{x}/N), \quad \text{with} \quad F = \mathit{fft}(\mathit{eye}(N)),$$

shows that every signal can be written as the sum of the pure frequencies, and that (except for the factor $1/N$) the Fourier coefficients are exactly the distinct coefficients.

$$\mathbf{x} = \sum_{k=0}^{N-1} c_k \chi_k$$

When the function is written like this, one can refer to it as the **Fourier Synthesis**. The unique coefficients are $\mathbf{c} = \mathit{fft}(\mathbf{x})/N$. Therefore, $\mathbf{x} \mapsto \mathit{fft}(\mathbf{x})$ is the **Fourier analysis**.

Relevance for the System Theory

Some of the most important properties of the Fourier transformation arise from the fact that the *convolution* is implemented in pointwise multiplicaion. Thus, it serves to illustrate *translation-invariant linear systems* as multiplication operators (*transfer function*). To do so, we observe the effect of a *cyclic rotation*, in matrix format ($N = 7$, shift of 2 samples):

$$S2 := \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{5}$$

Relevance for the System theory II

If one conjugates the standard matrix (each position lists its coordinate number from 1 to 49) with $S2$ (in other terms: $\mathrm{inv}(S2) * T7 * S2$) one gets the following matrix:

$$\begin{pmatrix} 41 & 48 & 6 & 13 & 20 & 27 & 34 \\ 42 & 49 & 7 & 14 & 21 & 28 & 35 \\ 36 & 43 & 1 & 8 & 15 & 22 & 29 \\ 37 & 44 & 2 & 9 & 16 & 23 & 30 \\ 38 & 45 & 3 & 10 & 17 & 24 & 31 \\ 39 & 46 & 4 & 11 & 18 & 25 & 32 \\ 40 & 47 & 5 & 12 & 19 & 26 & 33 \end{pmatrix} \tag{6}$$

All entries move down & right two coordinates parallel to the main diagonal.

Relevance for the System Theory III

Correspondingly, a 7 × 7 matrix commutes with all (cyclic) shifts if and only if (in cyclic sense) it is constant on the (cyclic) side diagonals. We use our own [NuHAG] command:

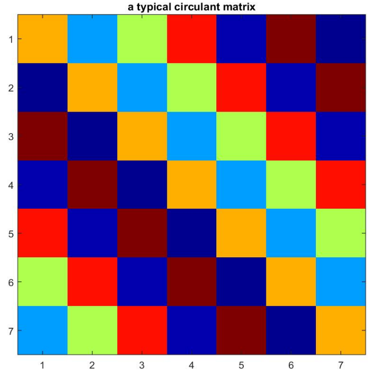
```
c = rand(1,7); CONVC = convmat(c),
```

$$\begin{pmatrix} 0.7133 & 0.5072 & 0.6387 & 0.7903 & 0.3971 & 0.8663 & 0.3760 \\ 0.3760 & 0.7133 & 0.5072 & 0.6387 & 0.7903 & 0.3971 & 0.8663 \\ 0.8663 & 0.3760 & 0.7133 & 0.5072 & 0.6387 & 0.7903 & 0.3971 \\ 0.3971 & 0.8663 & 0.3760 & 0.7133 & 0.5072 & 0.6387 & 0.7903 \\ 0.7903 & 0.3971 & 0.8663 & 0.3760 & 0.7133 & 0.5072 & 0.6387 \\ 0.6387 & 0.7903 & 0.3971 & 0.8663 & 0.3760 & 0.7133 & 0.5072 \\ 0.5072 & 0.6387 & 0.7903 & 0.3971 & 0.8663 & 0.3760 & 0.7133 \end{pmatrix} \quad (7)$$

Notice the correlation between the first column (*impulse response*) and the first row (2nd to 7th value in inverted order, BUT NOT simply transposed using FLIPUP!).

A typical circulant matrix

The colored table below represents a convolution matrix (where the first column represents the impulse response):



Diagonalization of circulant matrices

```
F7 = fft(eye(7)); D7 = F7' * CONVC * F7;
imagesc(abs(D7)); shg
```

The above illustrates the fact that the Fourier Matrix as a base change diagonalizes a circulant matrix!

```
norm(D7 - diag(diag(D7))),
```

This verifies that *D7* is indeed a diagonal matrix.
`diag(diag(A))` is the pure part of the diagonal matrix!

Additionally, the entries on the diagonal is the exact Fourier Transform of the generated vector *c*:

```
norm(fft(c(:))*7 - diag(D7))
```

Are there other Fourier Transforms?

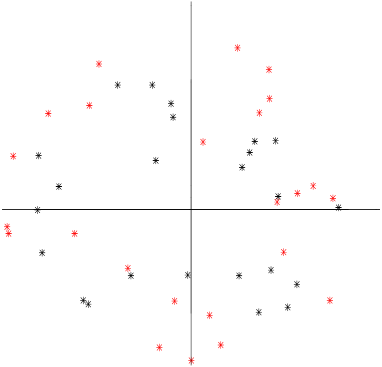
The better question to ask would be: Is it possible that there are other orthonormal bases which allow a diagonalization of such systems? If so, what could they look like? If not (right answer!), why do they look the way they look?

In detail this question cannot be answered with a MATLAB experiment. Nonetheless, it can at least be made plausible by illustrating what the multiplication with complex numbers means, and - deriving thereof - how a limited function, which has a complex number with an absolute value of 1 (on the unity circle) as eigenvalue concerning the translation operator T_x, CAN ONLY look like: It has to be a HELIX, i.e. a PURE FREQUENCY. We can illustrate that!

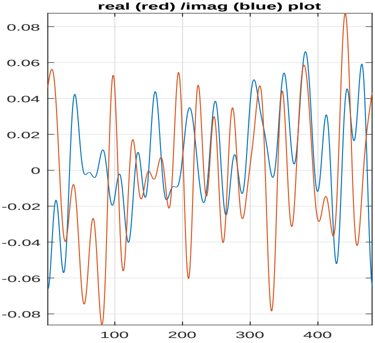
Multiplication of Complex Numbers

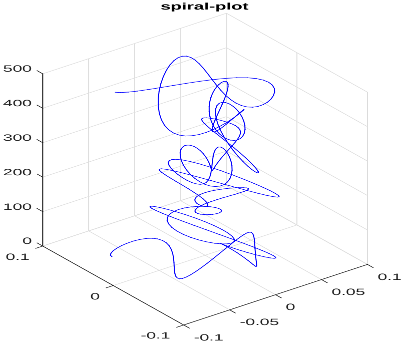
The red set of points is formed by a complex multiplication of the black set. With which factor in $\mathbb{C}$?
Reminder: The multiplication corresponds to a rotary extension (polar representation of complex numbers!).

a set of 24 points in the complex plane (black)

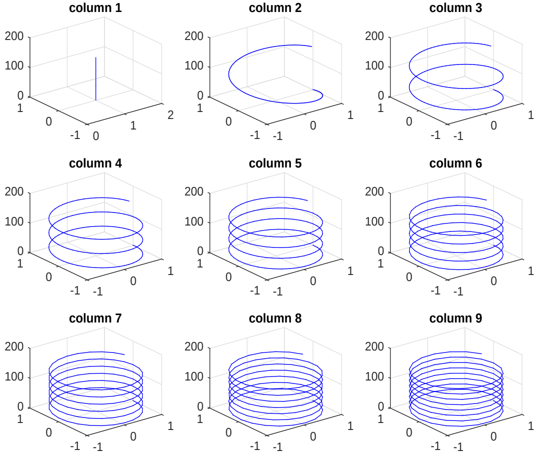


Plot of complex Function of Time





Columns of the Fourier Matrix: Spirals



Fourier Basis is not a Coincidence

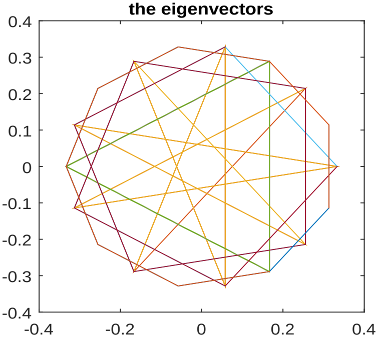
A random experiment leads to the same insight!

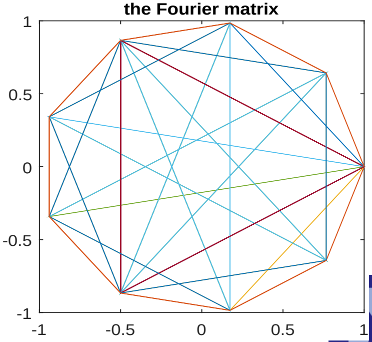
Not only does the Fourier basis transform all cyclic translation operators into multiplication operators (because of the exponential law), it rather diagonalizes *all translation-invariant linear systems*! The reasoning: the system matrices are sums of pure translation matrices.

The inversion is true as well. More or less every (random) translation-invariant linear system (in other words: every positively defined circulant matrix) has the pure frequencies as an orthonormal basis of eigenvectors
(!!except for the algebraic sign)!

Eigenvectors of Random Systems

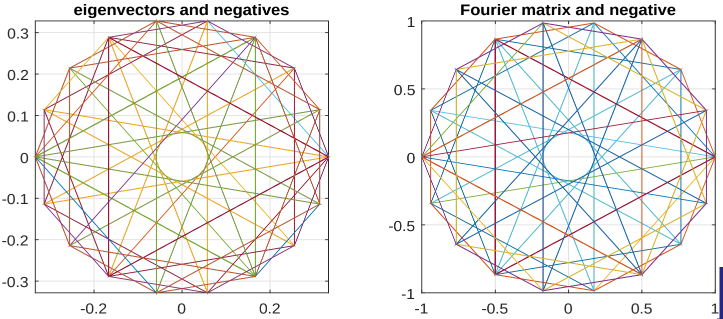
The following are images (plots in the complex plane) of the eigenvectors of a random, symmetric circulant matrix (to keep it simple $N = 9$):





Eigenvectors of Random Systems II

Since eigenvectors are each uniquely determined only by the sign, we plot these (in $\mathbb{C}$) together with their negative variants! (below for $N = 17$)



Practical Issues

The different variants show an **enourmous difference in time to complete!**

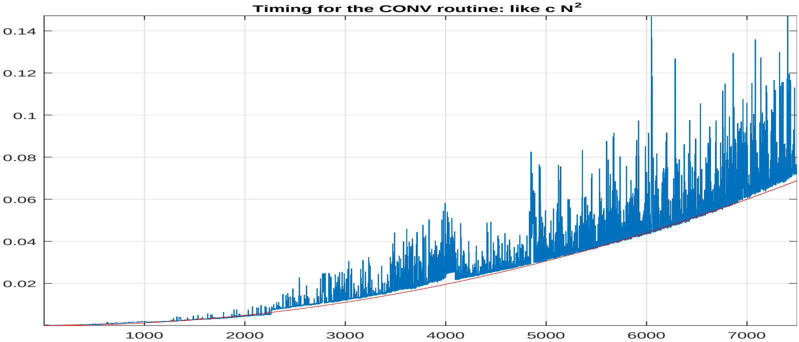
```
>> xx = rand(1,4410); 1/10 of a second audio signal
>> F = fft(eye(4410));
>> tic; myDFT(xx); toc; :NAIVE IMPLEMENTATION
>> tic; F*xx(:); toc :MATRIX MULTIPLIKATION
>> tic; fft(xx); toc :FFT ROUTINE
```

```
Elapsed time is 130.442276 seconds.
Elapsed time is 0.038570 seconds.
Elapsed time is 0.000194 seconds.
```

To check:

```
>> yx = myDFT(xx); norm(yx - fft(xx))
ans = 1.8122e-09
```

Duration of the CONV command for increasing N



Important Properties of the FFT/DFT

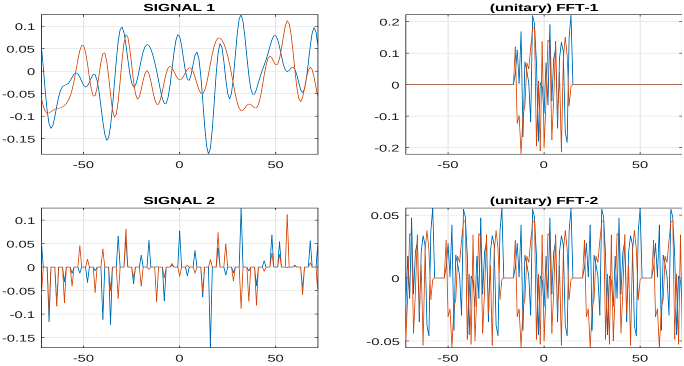
The shift of the coefficients means an increase in the power. This is the same as a multiplication of the Fourier coefficients with the pure frequencies.

The procedure of the so called *downsampling* can easily be explained. If one replaces the following sequence $[c_1, \ldots, c_{2n}]$ with the progression

$$[c_1, 0, c_3, 0, c_5, ...c_{2n-1}, 0],$$

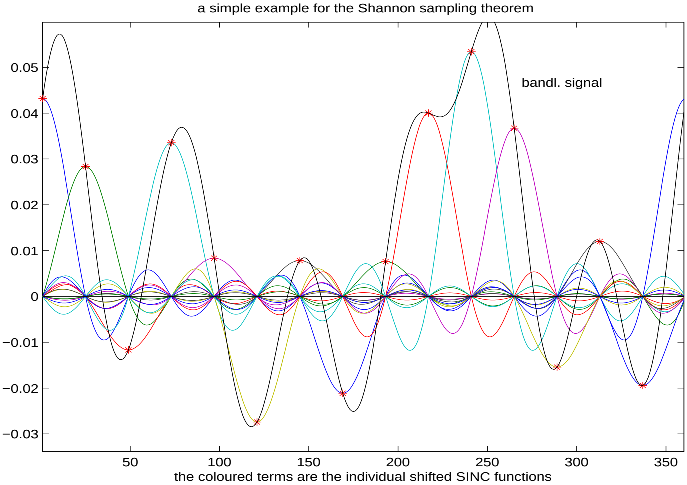
the application of the FFT routine means the evaluation of a polynomial $q(z)$. Given $d_k := c_{2k-1}, 1 \leq k \leq n$, the following holds: $p(z) = q(z^2)$, with $q(y) = d_1 + d_2 y + ...d_n y^{n-1}$.

Shannon Sampling Theorem



As seen above, one can win back the central block (around zero) by multiplication with a filter.

Reconstruction using a Shifted SINC Function



Common mistakes and tricks

There are books and even some mathematical literature that refer to some problems and special tricks in order to go from a contentious FT to a discrete FT

"Apparently" there are different conditions for the discretization. For example, one can ask the question, in what way the important property of the standardized Gauss function $g_0(t) := e^{-\pi|t|^2}$, that is being invariant under the (integral form of the) Fourier Transform, i.e. to fulfil $\mathcal{F}(g_0) = g_0$, has an analogy in the finite-discrete situations.

Correct Sampling and Approximation through Vectors

We have seen that the "naive application" of the FFT *does not lead to an FFT-invariant* discrete Gauss progression (we see a plot in the complex plane!).

The principal that symmetrical series have real FFTs needs to be addressed with care. What does it mean for a series, $(c_k)_{k=1}^N$, to be symmetrical or Hermitian?

A circulant matrix is only Hermitian if:
$c_1 \geq 0$, and $c_{k+1} = conj(c(N+1-k))$ for $1 \leq k \leq N/2$.

Recipe: **Sampling and dividing into periods**, ideally:
sampling rate $1/k$ and period k, with $k \in \mathbb{N}$.

CONV and Cauchy Product of the Coefficients

Not only do the polynomial functions create a vector space, which is furthermore invariant with respect to translations and dilation, they also create a *commutative algebra*.

The meaning of the *degree of the polynomial* lies in the fact that when multiplying the polynomial functions the degrees add up!

Cubic polynomials are obviously of the degree $= 3$, and of the order (number of coefficients) $= 4$.

In the description on the plane of the coefficients (CONV!) the multiplication can be characterized by means of the so-called **Cauchy product** (expand and rearrange!)

$$c_k = \sum_{k=0}^{r+s} a_k b_{r+s-k}.$$

The CONV Command and the PASCAL Triangle

```
>> format compact >> aa = [1,1]; bb = aa;
>> for jj=1:5; bb = conv(bb,aa), end
bb = 1 2 1
bb = 1 3 3 1
bb = 1 4 6 4 1
bb = 1 5 10 10 5 1
bb = 1 6 15 20 15 6 1
```

Provides the coefficient of a polynomial with the coefficients $[1,1]$, so $p(x) = 1 \cdot x + 1 = x + 1$ for $\mathrm{polyval}(aa, x)$, therefore $(x+1)^k, k = 2, \ldots, 6$.

Unconventional Applications

1. Multiplication of very long numbers
2. Sums of random variables with discrete distribution;
3. Demonstration of the central limit theorem; binomial distribution exact;
4. Multiplication and division of polynomials in two variables (using FFT2);

Unconventional Applications Ia

Since the result of the MATLAB command

$$\mathrm{polyval}([1,2,3],10)$$

is obviously the natural number 123, it is clear **that we can assign to every natural number (numerical sequence) a sequence of coefficients so that the respective polynomial gives exactly that number at $t = 10$.**

In this case, the MULTIPLICATION of POLYNOMIALS equates exactly to the multiplication of the respective integers, but can at the same time be easily materialized with the FFT! Merely at the end we have to ”clean up”. 17×10^1 is equal to 7×10^1 and 1×10^2!

Unconventional Applications Ib

A student's work carried out the following routines (addition, multiplication, division, etc.):

```
>> bigmul('343454353467885','579879234342934871242587')
ans = 199162047520704871884305106362548818495
>> bigpow('34343',9)
>> ans = 66457510529646949647733745761756398845543
```

Demonstration of routine "cleanvec":

```
>> bin = zeros(1,9); bin(1:2) = 1;
>> bin8 = round(real(ifft(fft(bin).^ 8)))
bin8 = 1 8 28 56 70 56 28 8 1
>> cleanvec(bin8) ans = 2 1 4 3 5 8 8 8 1
```

Unconventional Applications II

We already implied above that one can express the probability distribution of the dice experiment with a polynomial.

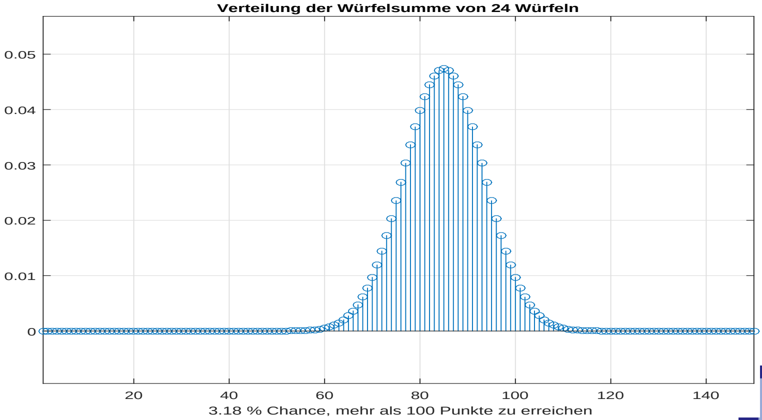
$$q(x) = (x^6 + x^5 + x^4 + x^3 + x^2 + x)/6$$

We already saw the probability distribution for the sum of 2 dice: $q(x)^2 = p(x)^2/36$ (look above).

The distribution for 20 dice: `w = zeros(1,150); w(2:7) = 1/6; w24 = real(ifft(fft(w).^ 24)); sum(w24(101:150)), stem(w24)`

The results show that the probability to score at least 100 points with 24 dice is $3,18\%$

Unconventional Applications IIb



Unconventional Applications III

The squaring (multiplication) of polynomials in two variables can be conducted by means of the FFT2: Say
$p(x, y) = 2 + x - y - 2xy + x^2$;

The respective coefficients can easily be rendered into a matrix M. Since it is clear that the highest power of $p^2(x, y)$ is without a doubt x^4, it suffices to embed the data in a 5×5 matrix.

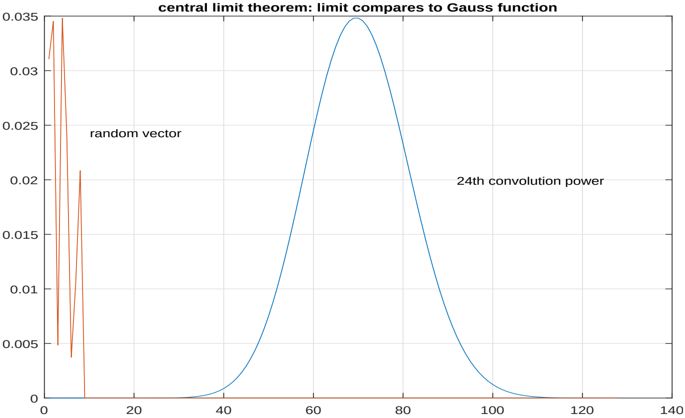
$$M = zeros(5); M(1,1) = 2; M(1,2) = 1; M(2,1) = -1;$$

$$M(2,2) = -2; M(1,3) = 1; round(real(ifft2(fft2(M).^2)))$$

Thereof we can deduce the coefficients of $p^2(x, y)$.
The factor of x^2y is -6, the one of x^3 is 2.
Equally, the division of polynomials can be conducted in the same manner!

Unconventional Applications IV

The central limit theorem states that a (high) convolution power will always look like a Gauss function:



Non-periodic Functions: the STFT

Normally, people argue that when working with non-periodic signals, the signals must at least be integrable or integrable as their respective squares. The normal argument uses Fourier Series Development with a very dense spectrum (the basic period gets larger).

The result is a continuous Fourier Transform, defined by its integral (defined for $f \in \boldsymbol{L}^1(\mathbb{R}^d)$):

$$\hat{f}(s) = \int_{\mathbb{R}^d} f(t)\, e^{2\pi i\, s \bullet t}\, dt, \quad s \in \mathbb{R}^d.$$

In this scope the FT can be seen as a unitary automorphism of $\left(\boldsymbol{L}^2(\mathbb{R}^d), \|\cdot\|_2\right)$ (Plancherel theorem), with

$$\|f\|_2 = \|\hat{f}\|_2, \quad f \in \boldsymbol{L}^2(\mathbb{R}^d).$$

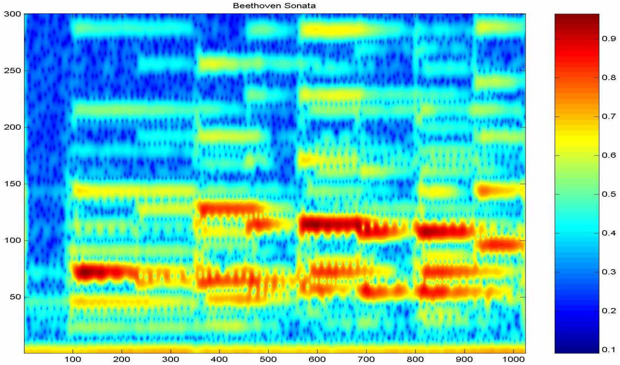
Short Term Transformation and Applications

This generalization of the finite Fourier Transform does not contain any information about *when* each of the sound events happen. In other terms: It does not contain any information about the melody of the signal.

This can only be resolved using STFT or Sliding Window Fourier Transform. This can also easily be illustrated in MATLAB.

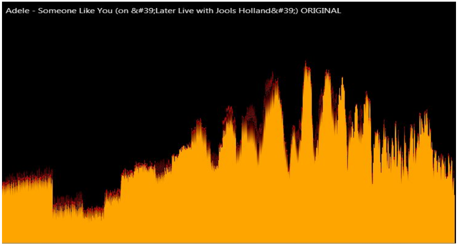
There are two interpretations of the Gabor Analysis: Either one wants to reconstruct a STFT from a discrete FT, or one wants to portray a signal atomically, using time frequency shifted variants of a Gabor Atom.

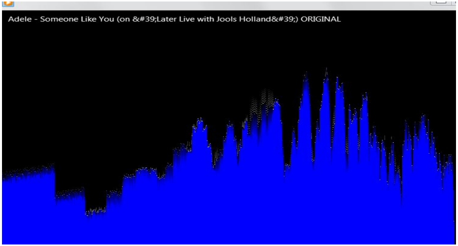
Gabor Analysis: Beethoven Piano Sonata



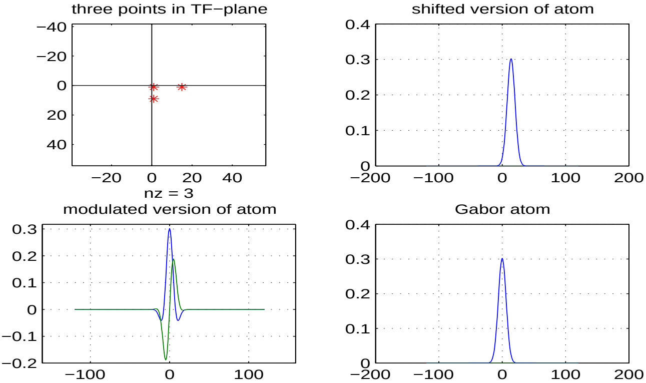
The spectrogram of the Beethoven sonata

Visualization using Windows Media Player

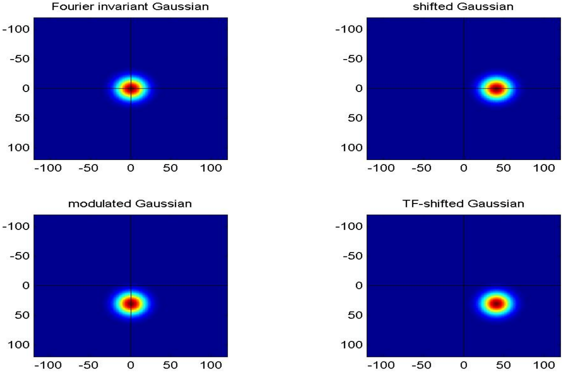

Adele - Someone Like You (on &#39;Later Live with Jools Holland&#39;) ORIGINAL


Adele - Someone Like You (on &#39;Later Live with Jools Holland&#39;) ORIGINAL

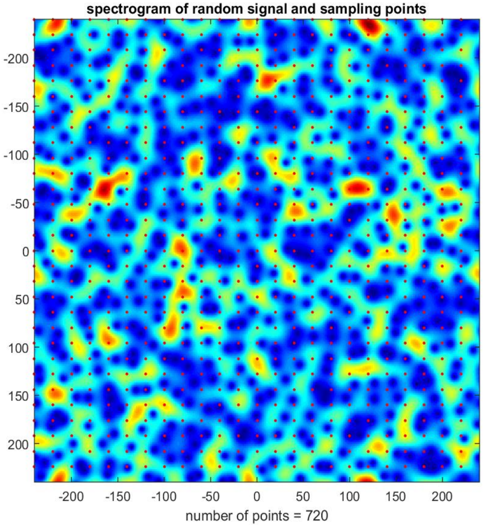
Time-Frequency Shifts



Time-Frequency Shifts in the Spectrogram

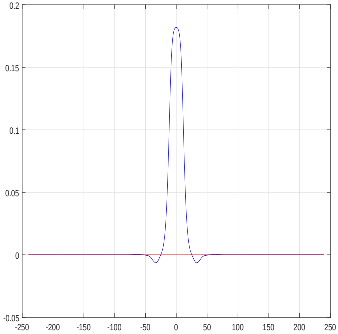


Spectrogram and Sampling Points



Spectrogram and Gabor Series Representation

Mathematically one can describe the situation like this: If one was to choose an atom g, that:



Tight Gabor Frames, cont.

If one was to choose such an atom, then the Gabor Family $(\pi(\lambda)g)_{\lambda\in\Lambda}$ is a "tight Gabor frame", meaning that one can write it in the following notation: $g_\lambda := \pi(\lambda)g$, which leads to

$$f = \sum_{\lambda\in\Lambda} \langle f, g_\lambda\rangle g_\lambda, \quad f \in \mathbb{C}^n.$$

Except for the constant $C = C(\Lambda, g)$ the following is true: The length of the coefficient vector $(\langle f, g_\lambda\rangle)$ is the norm $\|f\|_2$. These "tight frames" are not the same thing as an orthonormal system as they are linearly dependent (they are in many ways similar though!).

The *Gabor Multiplier* operates (by multiplying) on the Gabor coefficients, and is the basis of MP3 transformation (combined with the psychoacoustic masking effect).

Internet Resources and Links

We will publish all the commands used in the presentation as well as the used NuHAG M-Files. Please direct any questions towards the author:

`hans.feichtinger@univie.ac.at`

They will also be available online:

www.nuhag.eu/EXPO10

MATLAB materials of the NuHAG group can generally be found here:

http://www.univie.ac.at/nuhag-php/mmodule/

LTFAT, the **The Large Time-Frequency Analysis Toolbox**, can be found on this page:

http://ltfat.sourceforge.net/

Acknowledgement

This presentation was created in early 2016, while the author was staying at the Institute of Mathematics of the University of Salzburg. During this time the author got financially aided by the FWF Projekt (Fond zur Förderung der Wissenschaftlichen Forschung) **F5504-N26**, as the project was part of the "Quasi-Monte Carlo Methods: Theory and Applications" project.

The author thanks his host, Prof. Peter Hellekalek. Some ideas, which are presented in this presentation, are part of discussions with my student Bayram Ülgen, whose dissertation, with the title *"Signal- und Bildverarbeitung im Kontext Höherer Schulen. Grundlegende Vorstellungen und Chancen aus der Sicht der Mathematikdidaktik"*, will be defended in a few weeks (2016!).

Distribution Theory and Banach Gelfand Triple

The properties of the Fourier Transform explained in this presentation can easily be applied to the continuous signal analysis, if one looks at suitable pieces of the function only. The classic approach uses the *nuclear Frchet space* $\boldsymbol{\mathcal{S}}(\mathbb{R}^d)$ of rapid descending functions which is a special type of topological vector spaces, with a a countable family of seminorms. However, there is also a variant, where a single norm suffices, namely the (special) Segal Algebra with the following norm:

$$\|f\|_{\boldsymbol{S}_0} := \|STFT_{g_0} f\|_{\boldsymbol{L}^1(\mathbb{R}^{2d})} < \infty,$$

for which the Fourier inversion formula can easily be materialized with Riemann integrals, for example.

The Banach Gelfand Triple

Together the $\boldsymbol{L}^2(\mathbb{R}^d)$ and the dual space $\boldsymbol{S}_0'(\mathbb{R}^d)$ create three spaces, also called the *Banach Gelfand Triple* $(\boldsymbol{S}_0, \boldsymbol{L}^2, \boldsymbol{S}_0')(\mathbb{R}^d)$, which is a helpful, mathematically accurate tool for engineers. But this is not to be discussed today.

For example, one can characterize a Fourier Transform as a *unitary Banach Gelfand Triple automorphism*, which (like in the Linear Algebra example) maps the "pure frequencies" on the corresponding *Dirac measures* on the frequency side.