2.3 Fourier Series in inhomogeneous 1D Heat Equations

Example 6. We now consider an inhomogeneous problem. For the equation $\partial_t U[x,\ t]$ - $\alpha\ \partial_{x,x} U[x,\ t]$=G[x,t] with the right-hand side G we choose a heat flux density that is constant over time, so that G[x,t]= 0.6 (UnitStep[x-L/4]-UnitStep[x-3L/4]) (in C^0/s). We also choose as initial condition U[x,0]=0 for x in [0,L] and as last Neumann boundary conditions. The model describes a uniform heating of our copper rod around the center of the rod, which is otherwise thought to be perfectly insulated. (Temperature now denoted as U, so that - as long as the animation above is still running - there is no naming conflict).

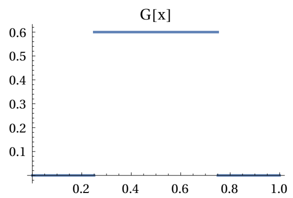
We start with an approach using “variation of the constants”, i.e. we use the solution approach U[x,t] = c_0[t]+ $\sum_{n=1}^{\infty} c_n$[t] Cos[nπx/L], insert this into the equation and use the boundary and initial conditions (please carry out for practice).
This results in c_n' [t] + (α n^2 π^2)/L^2 c_n[t] = g_n for n>0, $c_0[t] = g_0$ t. The constants g_n denote the Fourier cosine coeffi - cients of the inhomogeneity G, g_0 the mean value of G. Calculation for G[x,t] with L=1 m and α:=117 10^(-6) m^2/s as above results in c_0[t] = g_0 t (from c_0' [t] = g_0 and c_0[0]=0)

In[10]:=
```
L = 1;
G[x_] = 6 / 10 (UnitStep [x - L / 4] - UnitStep [x - 3 / 4 L])
g[n_] = 2 / L Integrate[G[x] Cos[n Pi x / L], {x, 0, L}]
g0 = 1 / L Integrate[G[x] , {x, 0, L}]
Plot[G[x], {x, 0, L}, PlotLabel → "G[x]", ImageSize → Small]
```

Out[11]= $\frac{3}{5}\left(-\text{UnitStep}\left[-\frac{3}{4}+x\right]+\text{UnitStep}\left[-\frac{1}{4}+x\right]\right)$

Out[12]= $\frac{6\left(-\text{Sin}\left[\frac{n\pi}{4}\right]+\text{Sin}\left[\frac{3n\pi}{4}\right]\right)}{5\,n\,\pi}$

Out[13]= $\frac{3}{10}$

Out[14]=


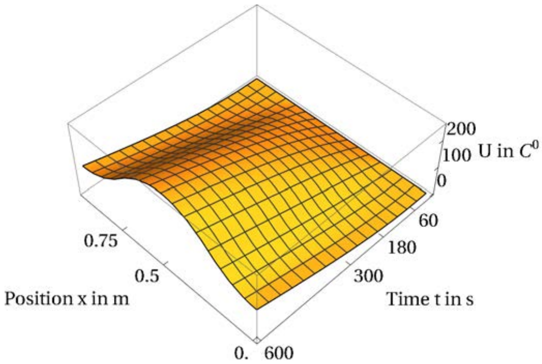
In[15]:=
```
sol := DSolve [{D[c[n, t], {t, 1}] + α n ^ 2 Pi ^ 2 / L ^ 2 c[n, t] == g[n],
   c[n, 0] == 0}, c[n, t], {n, t}]
ccoeff [n_, t_] = sol⟦1, 1, 2⟧
```

Out[16]= $-\frac{1}{n^3}\,330.785\ e^{-0.00115474\, n^2\, t}$
$\left(-1.\ \text{Sin}[0.785398\ n]+1.\ \text{Sin}[2.35619\ n]+1.\ e^{0.00115474\, n^2\, t}\ \text{Sin}\left[\frac{n\,\pi}{4}\right]-1.\ e^{0.00115474\, n^2\, t}\ \text{Sin}\left[\frac{3\,n\,\pi}{4}\right]\right)$

We consider a partial sum of the Fourier series of the exact solution for the problem and plot it. Only every fourth Fourier coefficient is non - zero. In order to reproduce the step - like inhomogeneity well, we choose a higher order (m = 30) of the trigonometric polynomial to approximate the heat flux density and a corresponding order of the trigonometric approxima - tion polynomial for the solution. The "step form" of the initially inhomogeneity remains largely intact in the solution for quite some time before the heat balance takes effect.

In[17]:=
```
U[m_, x_, t_] := g0 t + Sum[ccoeff[n, t] Cos[n Pi x / L], {n, 1, m}]
Plot3D[N[U[30, x, t]], {x, 0, L}, {t, 0, 600},
 PlotRange → All, AxesLabel → {"Position  x in m ", " Time  t in s ", "U in C^0 "},
 AxesStyle → Directive[Black, 12], ViewPoint → {-2, 2, 3},
 Ticks → {{0.0, 0.5, 0.75}, {60, 180, 300, 600}, {0, 100, 200, 300}}]
```

Out[18]=

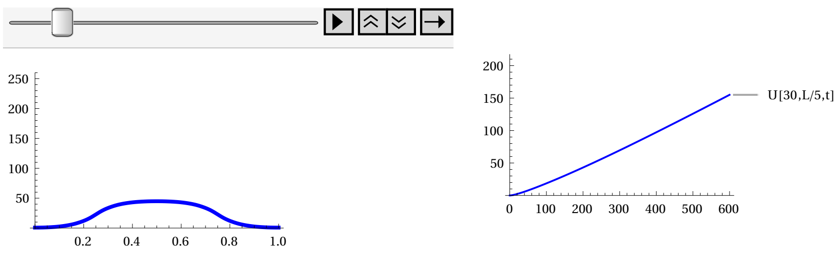


Here again the temperature evolution as an animation (due to the approximation of approximation of G by a trigonometric polynomial, we have negative temperatures U at the edge and a "ripple". Both effects are due to the Gibbs phenomenon.
A mathematically precise treatment of differential equations with discontinuous right-hand sides, such as G here, is possible within the framework of distribution theory (see[1]).

In the right graphics the temperature development is shown, smoothed by arithmetic averaging of the partial sums of the result at some distance from the directly heated interval [L/4,3L/4]. We see an approximately linear increase in the tempera - ture, which becomes stronger as you move closer to the interval that is heated (Test it yourself.) Here we look at x=L/5:

In[ ]:=
```
a1 = Animate[Plot[N[U[30, x, t]], {x, 0, 1}, PlotRange → {-2, 260},
    PlotStyle → Directive[Blue, Thickness[0.015]], ImageSize → Small], {t, 0, 600},
   AnimationRepetitions → 1, AnimationRate → 5, RefreshRate → 50];
Usmoothed[m_, x_, t_] := g0 t + Sum[ccoeff[n, t] (1 - n / (m + 1)) Cos[n Pi x / L], {n, 1, m}]
a2 = Plot[N[Usmoothed[30, L / 5, t]], {t, 0, 600}, PlotRange → {0, 200},
   PlotStyle → Directive[Blue, Thickness[0.008]], PlotLabels → {"U[30,L/5,t]"}];
GraphicsRow[{a1, a2}]
```

Out[ ]=



And the **final test** that everything is correct: The mean temperature after 600 s corresponds to the heat supplied corresponding to 0.3 C^0/s on average for the whole rod with the assumed perfect insulation and the differential equation as well as the initial and the boundary conditions are fulfilled:

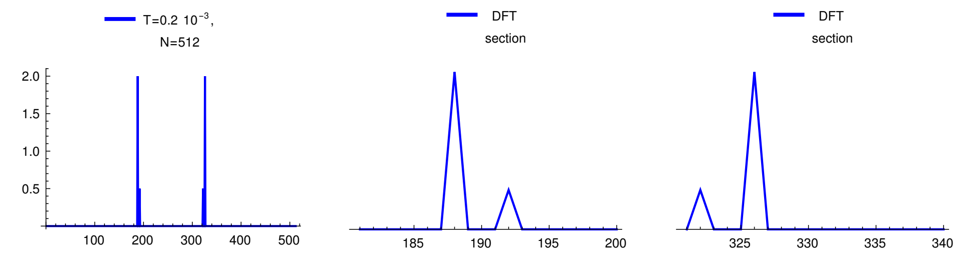
Example 4. For EMC radiation measurements in the **GHz range**, you simply **avoid sampling rates of several gigasamples per second** by using bandpass filters. For example, bandpass filters with a bandwidth of 1 MHz in subbands are available as analog circuits to achieve the filtering in advance to a DFT.
Then, with **only a 512 - point DFT** and an observation time of **T=0.2 ms** per frequency band, i.e. approximately 2.5 MHz sampling frequency N/T, you achieve a frequency resolution 1/T of about 5 kHz. The analysis of the subbands in a measurement lab can then be put together to form an overall picture ("Undersampling Solution for High Frequency FFT Analysis"). This saves a lot of time and costs for such radiation measurements.
We look at the DFT magnitude spectrum of such an example in a single subband of width 1 MHz, which shows that the alias effect must be carefully considered when to make statements with correct frequency assignments. It is assumed that the signal is in the frequency band [1GHz, 1GHz + 1MHz], e.g. generated at the output of a corresponding bandpass filter.

```
In[ ]:= T = 0.2 × 10^(-3);
NN = 512; (* T observation time, NN number of samples
 (NN instead of N here, since N is protected by Mathematica) *)
expml2 = Table[Cos[2 Pi (10^9 + 5000) n T / NN] + 4 Cos[2 Pi (10^9 + 25 000) n T / NN],
   {n, 0, 511}];(* high frequency signal *)
```

We are therefore looking at the superposition of two high - frequency oscillations in the GHz range. We then plot the entire DFT magnitude spectrum as a polygonal curve as well as the relevant parts of it and see how the frequency assignment in the example has to be done. We have taken only 512 samples of the signal in the time T = 0.2 ms.

```
In[ ]:= absdft = Abs[Fourier[expml2, FourierParameters → {-1, -1}]];
p1 = ListLinePlot[%, PlotRange → All, PlotStyle →
    Directive[Blue, Thickness[0.008]], PlotLegends → Placed[{"T=0.2 10^-3,
N=512"}, Above]];
list1 = Table[absdft[[n]], {n, 181, 200}];  (* Extraction of part of the DFT list*)
list2 = Table[absdft[[n]], {n, 321, 340}];
p2 = ListLinePlot[list1,
   PlotStyle → Directive[Blue, Thickness[0.008]], PlotRange → All,
   DataRange → {181, 200}, Axes → {True, False}, PlotLegends → Placed[{"DFT
section"}, Above]];
p3 = ListLinePlot[list2,
   PlotStyle → Directive[Blue, Thickness[0.008]], PlotRange → All,
   DataRange → {321, 340}, Axes → {True, False}, PlotLegends → Placed[{"DFT
section"}, Above]];
GraphicsRow[{p1, p2, p3}, ImageSize → Full]
```

Out[ ]=



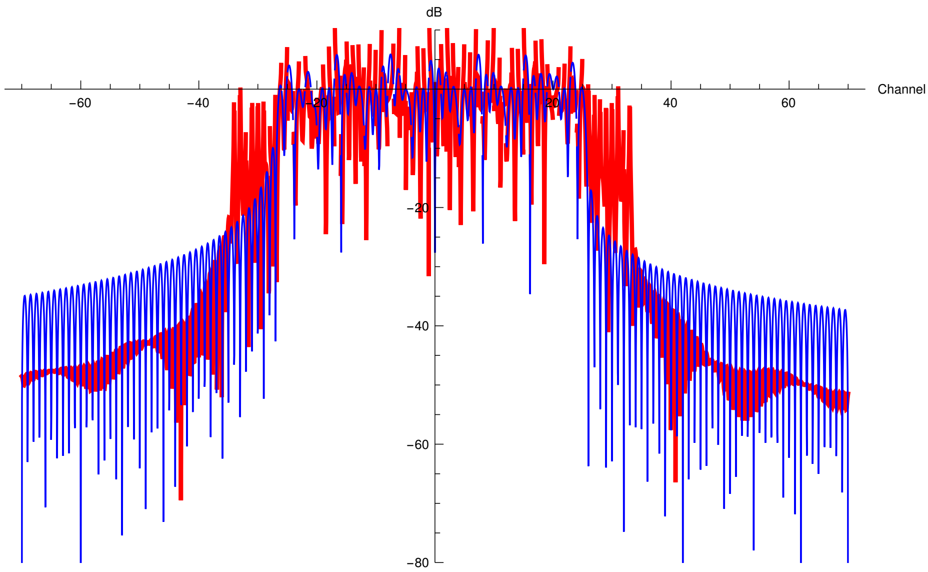
Frequency assignment: The two "peaks" of height 2 belong to the oscillation with frequency 10^9 + 25000 Hz and amplitude 4.
In Mathematica, compared to the notation in[1] they have a number increased by 1. The peaks with numbers 188 and 326 therefore belong - this is where the alias effect comes into play - to 4 Cos[2 π (10^9 + 25000) t] = 4 Cos[(325 + 390*N) ω_0t] with $\omega_0 = 2\,\pi/T = 2\,\pi$*5*10^3 rad/s, T = 0.2*10^(-3)s observation time as above, because 187 = -325 + 512 and (325 + 390*512)*5*(10^3) = 1000025000. The peak with the number 188 in Mathematica then belongs as an alias to the oscillation component 2 Exp[-I (325 + 390 N) ω_0t] = 2 Exp[+I (187 - 391 N) ω_0t] in the Fourier series of the T - periodically extended signal .
In the same way, you can assign the other oscillation frequency corresponding to the two peaks with the numbers 192 and 322 and the height 1/2. Please carry out the small analog calculation yourself. In particular, we note that the values associated with the positive signal frequencies with the numbers 322 and 326 lie in the upper half of the DFT spectrum, while those with the numbers 188 and 192 are "alias values" of parts with negative frequencies.

The alias effect therefore sometimes requires a little thought in order to assign the DFT line spectra to the correct frequencies. However, if necessary, you can write a small program for this.

Exercise: Consider (with pencil and paper if necessary) which numbers have the peaks of a 512-point DFT with Mathematica for an observation time T=0.2*10^(-3) s of an oscillation with the frequency 1001.06 MHz?

Example 5. (Subsampling in digital transmission systems) **A decisive advantage of the alias effect** with a DFT is found in any kind of digital transmission (WLAN, mobile phones, DVB etc.). The point is that the transmissions take place in very high frequency bands outside the bandwidth of the used digital devices like phones et al. Since the transmission bands are known at the receiver side, the *signals are accordingly undersampled*, what automati - cally can generate the signal spectrum in a lower frequency band by aliasing. For example, 5G transmissions can use frequency bands up to 26 GHz, while mobile phones at present have 2.2-2.6 GHz CPU's. For the digital signal processing **direct IF subsampling receivers** (IF, intermediate frequency) can be used to shift the signal spectrum without analog mixers by the alias effect from a high to a low frequency band, where the phone signal processing works. This reduces considerably receiver complexity, power consumption and costs of hardware.
We have seen that the bandwidth of a segment of a signal spectrum by a DFT is determined by N and T , and thus a segment of the spectrum is representable by a DFT without aliasing. **Not a priori determined is the position** of such a spectral part on the frequency axis. Its position can be determined from a priori knowledge or deliberately. This has disadvantages in observing unknown signals, but also **enormous advantages in signal processing** for technical systems as for example in communications engineering. Because there the signals and the allocation of signal frequencies in the spectrum can be chosen intentionally. Thus, the **DFT with subsampling offers** the opportunity to bring **a signal spectrum automatically into a frequency band where device processing works** . This is one of the reasons, why digital transmission nowadays is so successful and cheap, because otherwise with analog technique you would need expensive mixers to achieve the same by amplitude modulations. Modern digital transmission with multi-carrier methods like OFDM transmission in high frequency bands the information in spectra of trigonometric polynomials, which can be reconstructed with a DFT by aliasing in a desired lower frequency band. This is a cornerstone in modern communication systems. We can see more on this in [1], 12.3 or in a later booklet on Fourier transforms and the principle of OFDM transmissions with Mathematica.

Now, we consider the **typical spectral shape of a WLAN pulse**, i.e., a single OFDM symbol, which transmits with 16 QAM an information package. The 802.11a/g standard uses 48 data subcarriers and 4 pilot subcarriers. Thus, for the pulse a trigonometric polynomial of degree 52 is used to transmit the data in complex amplitudes c_k. Each subcarrier is a trigonometric function of the form $c_k\, e^{i\, k\, \omega_0\, t}$ of a given frequency known at the receiver. Therefore, the receiver can reconstruct the amplitudes by discrete Fourier transforms of the sampled signal. The total channel bandwidth is 20 MHz with an occupied bandwidth of 16.6 MHz. You clearly see that the blue spectrum of a transmission, when a rectangle time window is used, has much more out-of-band transmission than the red spectrum, where an RC pulse shaping is used. Thus, pulse shaping is a relevant topic in communication engineering. **More on this in [1], 12.3.**



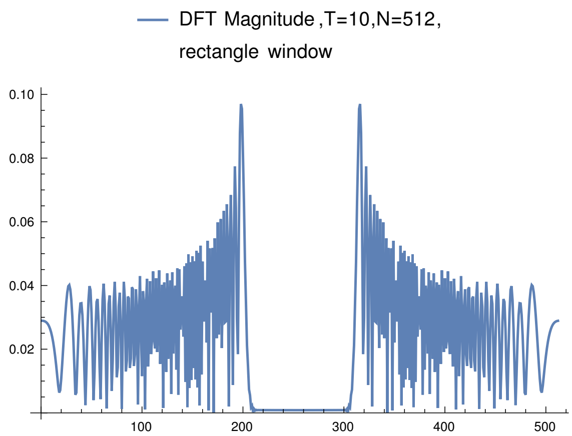
Example 14. Windowed Fourier transform with a DFT, Spectrograms

In its classical form, the Fourier transform $\mathcal{F}$ does not allow for simultaneous time-frequency analysis. For example, speech or a piece of music in our everyday experience has a specific "time pattern" and at the same time a specific "frequency pattern". However, the spectral function of a signal does not show at what times and with what respective amplitudes a specific angular frequency ω occurs in a signal f, but rather accumulates contributions of the same angular frequency ω over the entire time course of f in $\mathcal{F}$ f(ω). Dennis Gabor (1900-1979) already noticed these disadvantages for signal processing purposes, and in 1946 in his work "Theory of Communication," he proposed time-frequency localization through Fourier transforms with window functions.
To obtain information about the "time-frequency pattern" of a signal, one determines not the spectral function of the entire signal, but the spectral functions for time segments of f. Time segments of a signal f are obtained by multiplying f with functions of finite effective duration. Such functions are referred to as window functions or time windows as considered above. We consider the following example.

A short-term model for a siren is approximately the function or chirp f(t)=A sin(g(t)) with g(t)=2πt (αt+βt^2) for 0⩽ t⩽10 s and constants A, α, β. The derivative of the argument g'(t)=2πt(2α+ 3βt) can be considered as the instantaneous angular frequency at time t. The magnitude spectrum, approximately calculated with a DFT for paramters A=1, $\alpha = 4[1/s^2]$, $\beta = -4/15\ [1/s^3]$ over T = 10 s, shows a multitude of frequencies up to the maximum frequency 20 Hz, but not the parabolic frequency modulation and not the instantaneous frequencies at different times (left image below). The graph of an approximation for the windowed Fourier transform of f with the "Hann window" w(t)=0.5-0.5 cos(2 πt/T) for 0 ⩽t⩽ T=1 s, on the other hand, clearly shows the rise and fall of the instantaneous frequencies and corresponds to our usual impression of the variable frequency of the siren tone (right image). The calculations used a 512-point DFT over a total of T =10 s, with the DFT coefficients $C_k T$ plotted as approximations for $\mathcal{F}$ f(2πk/T) in the left image. In the second case, 50 Hann windows of duration 1 s were used at intervals of 0.2 s each. Per time segment, a 128-point DFT was performed and the resulting (single-sided) DFT magnitude spectra were combined to form the second image. Neither representation shows the constant amplitude A=1. One reason is the strong aliasing effects due to the frequency modulation. The sum of the $|C_k|^2$ of the left image agrees numerically very well with the quadratic mean of f in [0, T] (in both cases, the value is about 0.5). Numerical integration to calculate the windowed Fourier transform for 20 Hz at t_0 = 5 s results in approximately 0.24, as shown in the following spectrogram on the right. The signal values (and thus A) can only be approximately recovered from the DFT using an interpolation polynomial or the formula for discrete reconstruction from the data (for more details please see [1], 12.5). Now to the images:

In[ ]:=
```
B = 1;
M = 128;
NN = 50;
h[x_] = UnitStep[x];
w[x_] = (0.5 - 0.5 Cos[2 Pi B x]) * (h[x] - h[x - 1 / B]);
f[x_] = Sin[2 Pi 20 × (2 / 10 x ^ 2 - 1 / 75 x ^ 3)]
data0 = Table[f[10 n / 512], {n, 0, 511}];
dft = Chop[Fourier[data0, FourierParameters → {-1, -1}]];
pdft = ListLinePlot[Abs[dft], PlotRange → All,
  PlotLegends → Placed[{"DFT Magnitude ,T=10,N=512,
rectangle window"}, Above]]
data1[k_, j_] = N[w[j / (B M)] × f[k * 0.2 + j / (B M)]];
FT1[k_, n_] := N[1 / M Sum[data1[k, j] Exp[-2 Pi I n j / M], {j, 0, M - 1}]];
z[k_, n_] := N[Abs[FT1[k, n]]];
data2 = Table[z[k , n], {n, 1, 25}, {k, 0, NN - 1}];
pwindowed =
  ListPlot3D[data2, PlotRange → All, Mesh → 100, Axes → {True, True, True},
   Boxed → False, AxesLabel → {"Window No.,Time in s =
Window No. x 0.2s", "Hz", "Magnitude"},
   AxesStyle → Directive[Black, Plain, 10],
   PlotStyle → Directive[PlotPoints → 100], ViewPoint → {30, -40, 50},
   AxesEdge → {{-1, -1}, {1, -1}, {-1, -1}}, Ticks → {{10, 40}, {10, 20, 25}, {0.0, 0.2}},
   PlotLegends → Placed[{"3D Spectrogram , Windowed Fourier Transform ,
50 Hann Windows in T=10s"}, Above]];
```

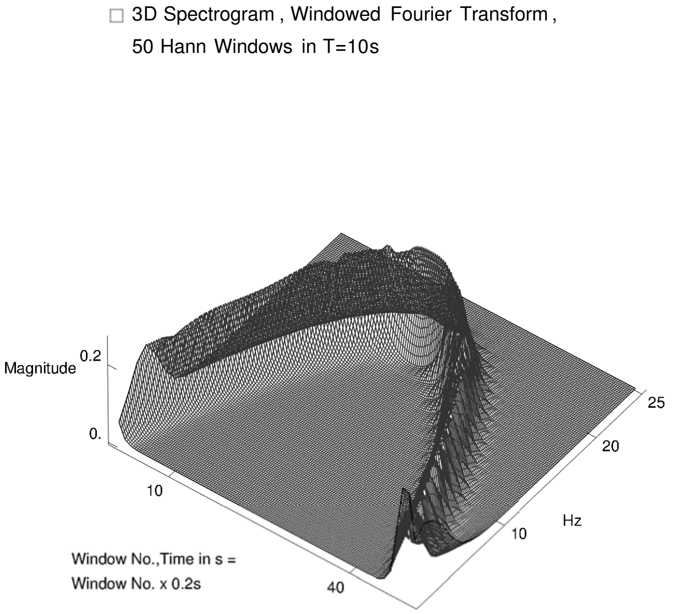
Out[ ]= $\text{Sin}\left[40\,\pi\left(\frac{x^2}{5}-\frac{x^3}{75}\right)\right]$

Out[ ]=


Now the 3D illustration of the windowed Fourier transform showing the time-frequency pattern of the signal. **The illustration is also called a spectrogram.**

In[ ]:=
```
Show[pwindowed]
```

Out[ ]=



In Mathematica you can illustrate spectrograms in a 2D image with the **command Spectrogram for a list of sampled data**. For tests this is left to the reader. Instead we make a 2D representation ourselves with MatrixPlot for our list data2:

In[ ]:=
```
mplot = MatrixPlot[data2 , PlotLegends → True ,
    Axes → True , FrameLabel → {"Hz", "Window No."},
    DataReversed → {True , False}, ColorFunction → "CMYKColors "];
```

2 D Spectrogram of the siren signal, time t = window number x 0.2s.
It could be sharpened with more sampling points.

In[ ]:=
```
Show[mplot]
```

Out[ ]=