Algovize

Luděk Kučera

January 12, 2008

1 Discrete Fourier transform

Scene: *Function drawing*

The aim of several introductory scenes is to learn how to work with the applet. The first scene shows a window displaying a graph of a function. The window is called a *function window*. The graph of a function can be re-drawn using a mouse; try it.

Scene: *Function defining*

A function in the function window can also be described by a formula written in the upper field of the control bar. A formula is written in a way similar to many pocket calculators, e.g. `sin x`, `cos x`, `exp x`, `sgn x`, and their combinations like `2*sin x`, `1.5*cos x + sgn x`, `exp(-x)*(sin x + cos x)` etc. The full syntax is described in the appendix [the appendix in not yet finished]. For simplicity if the expression is not defined for some x, the calculator uses 0 as a result.

In many case the x- and y-scale must be changed. A displayed interval on the x-axis spans between values $x1$ and $x2$ as set in the control bar, similarly for $y1$ and $y2$ and the y-axis..

Scene: *Sinusoids*

The most important functions of the Fourier analysis are functions derived from the function sinus; some of them will be displayed in the scene. Let us quickly revise their properties.

The standard functions sinus a cosinus are periodical with the period 2π. We will use modified functions $\sin(2\pi x)$ and $\cos(2\pi x)$ that have the period 1 and can be considered as the functions sinus and cosinus scaled along the x-axis. Therefore the function window is adapted to display $x \in [0, 1]$.

The screen shows another window, called a *spectral window* for reasons that will become clear later. In the present scene the window will be used in a very restricted way:

Click the blue square in the spectral window. The function window will show a graph of a constant function that has a value 1 for all x. This function will be used quite often.

Click the red square labeled by 1 to display the function $\sin(2\pi x)$ and click the green square labeled by 1 to display the function $\cos(2\pi x)$, in both cases for values x in the range displayed in the function window. Both functions have the period equal to the width of the function window.

Now click first the red and then the green square labeled by 2. This displays functions $\sin(2 \cdot 2\pi x)$ and $\cos(2 \cdot 2\pi x)$. They are similar to the two preceeding functions, but their period is two times smaller, in other words their frequency is two times higher. The full width of the functional window displays two full periods.

Remaining squares serve to display sinusoidal and cosinusoidal functions of higher frequencies. If a square is labeled by k, the displayed function is either $\sin(2\pi kx)$ or $\cos(2\pi kx)$. They have similar smooth periodic form, but their period is k-time smaller than the width of the functional window; in other words the have k-times higher frequency than the functions $\sin(2\pi x)$ and $\cos(2\pi x)$, that were displayed at the beginning.

Now try to drag the top of a red, green or blue bar in the spectral window up or down. The amplitude of the corresponding sinusoidal function in the functiow window changes and remains proportional to the heigth of the bar. Try again choosing different periods of the sinusoids and changing their amplitudes.

Scene: *Linear combinations of sinusoids*

This scene is a simple but very important exercise in using the spectral window. In the previous scene only one sinusoidal function could be selected and its amplitude changed. In this scene we can select several different sinusoids (all of them if you wish), choose their amplitudes, and the functional window displays their *sum*.

In fact, when entering the scene, *all* boxes in the spectral window are checked, but the amplitudes for all but one sinusoids are zero (which is equivalent to non-selection).

Change amplitudes in different columns in the spectral window in a way similar to audio mixing and observe how the corresponding linear combination of the sinusoids in the functional window changes.

Any box in the spectral window can be deselected by a mouse click; after that the corresponding sinusoid does not enter in the sum displayed in the function window. Note that unselecting a sinusoid is equivalent to selecting with the amplitude 0.

In many cases it will be necessary to change the y-scale to see the whole function (the x-scale can not be changed in this scene).

Do not leave this scene too early, experiment with the tool to create "weird" functions with sudden and sharp changes of their values.

Scene: *Sampling of a continuous function*

Let us forget for a moment a limited resolution of the screen and imagine that the x-axis represents a continuum of real numbers. A description of a function over a continuous axis would have needed an infinite amount of data and if the function represented a physical signal, each mesurement must have been performed in an infinitesimaly short time interval, which is, of course, not possible.

Therefore a digital signal is *sampled*, which means that its values are periodically measured in pre-defined time instants. If the signal is periodic and if the period is 1 and n samples are taken during this time, then the values of the signal are determined in times $x_0, \ldots, x_{n-1}$, where $x_\ell = \ell/n$ for $\ell = 0, \ldots, n-1$.

The sample points are displayed as vertical lines in the function window and values of the function (signal) are emphasized by small circles. The number of sample points can be changed using the control bar field [**Value** N] (which should be a reasonable even number). Taking into account their regular placement, this number is sufficient to describe the sample points completely.

Scene: *Spectral analysis*

In this scene two functions are generally displayed in the function window. The black one (originally a sinusoid) can be redrawn by a mouse or replaced by a function specified in the formula field of the control bar. It is also possible to change the frequency of sampling and the scale both the x- and y-axis.

The second function displayed in the function window, the red one, corresponds to bars in the spectral window. It can not be re-drawn in the function window, but it is computed by Algovision in a way described below.

Fix a particular black function. The task is to match the black function by the red one using the bars of the spectral window. Of course, it would not be possible to overlap black and red functions completely, because the black graph has much larger number of degrees of freedom than n degrees of freedom of the spectral window, but the main result of the Discrete Fourier Analysis says that the bars in the spectral window are sufficient to exactly match values of any black function by the values of the red function *in all sampling points*.

In this scene the amplitudes of particular sinusoids to match the black function in sample points are computed by Algovision. Change the initial black function and observe how the amplitudes change to keep matching the black function. The collection of the amplitudes is called a *spectrum* of the input function (which explains the name of the spectral window).

Keep changing the black function and observe its spectrum, computed by Algovision, and the corresponding red function. Try functions that suddenly change their value ("jump"). The red function reacts to a jump in a way that is known from electrical circuits - it "oscilates" for some time around values of the black function in such a way that it matches the black function in sample points, but the amplitude of the oscilation decreases in time. However, the analogy is not complete, the behavior of the red function is "time reversible", and the red fuction oscilates also in preparation to a jump before it occurs.

Scene: *Spectral search*

Now we come to the most difficult exercise. Try to do what was done by Algovision in the previous scene: find the exact match of the black sampled points by the red function, i.e., find the spectrum of the black function. This is another scene, where you should spent as much time as possible and look for approximation of different types and shapes of the black function.

You will learn that the change of the constant component of the spectrum moves a graph of the black function up and down, the component $\sin x$ shifts the graph up in the first quarter of the width of the functional window and pushes it down the same amount in the third quarter while keeping its value essentially unchanged on the ends and in the middle of the window, etc.

I must admit that the exercise is in fact extremely difficult and it would be surprising if a beginner succeeds completely. This is why the scene provides help for a user. If a question mark below a column in the spectral window is clicked, the correct value of the bar in the column is set by Algovision, leaving the other bars unchanged.

Most instructive is the Fourier approximation of the signum function given when the scene in entered (note that there is a discontinuity - a jump of the value by 2 - for $x = 0$).

Scene: *Spectral compresssion*

The scene is similar to the previous one, but it difers in the goal. Now, the spectrum of a black function is shown automatically, but it is possible to switch off selected columns in the spectral window. As a result, the inactive columns do not contribute to the linear combination described in the previous scene and the approximating red function differs from the original black function even in the sample points. It can be seen that switching off higher harmonic components (functions with high frequency) usually preserves the global character of the approximated function. If, e.g., the function describes an acoustic signal, removal of higher harmonic components can change the timbre of the sound, but the signal is apprehended in a similar way: speach can still be understood and music gives the same artistic impression. This is a basis for signal compression, used for example in wireless communication: higher harmonic components are suppressed to limit amount of information carried by the signal.

Similar principle is used for JPEG image compression: spectral decomposition of visual data is performed, less important components of the spectrum are eleminated or supressed and modified spectral information is stored. Reconstruction of an image from the modified spectrum gives an image that can not usually be distinguished from the original one. On the other hand it is known that in some cases, e.g., if two bright monochromatic areas of different colors meet, JPEG could create artefacts that are caused by the effects of the same nature as oscillation in response to a jump behavior, that was mentioned in the previous scene.

It should be mentioned that the JPEG compression uses the cosine transformation rather than the discrete Fourier transformation, but the difference is not substantial, the cosine transformation can be considered as the Fourier transformation modified to avoid using complex numbers and working only with reals.