Program for calculating total harmonic distortion

Description for program version dated April 1, 2026.

When manufacturing a four-band parametric equalizer (based on model UREI 546 by company UREI), I needed to measure its total harmonic distortion (THD), so I created program described here for its calculation in GNU Octave environment (versions 10.2, 10.3 and 11.1). Program was designed for csv or txt files containing "Wave" data saved by HANTEK DSO4102C and OWON ADS802A digital oscilloscopes, or for the same files created by other oscilloscopes or programs. Octave is free (unlike Matlab) and can be downloaded at: https://octave.org. In order for program to work in an Octave environment, it is also necessary to download and install Signal extension package into Octave environment. Various extension packages for Octave are available for download at: https://gnu-octave.github.io/packages/, Signal package is specifically at: https://gnu-octave.github.io/packages/signal/. After downloading, first install Octave programming environment on your computer by running its downloaded installation file, then install Signal package in Octave environment in "Command Window" using command: `pkg install signal-1.4.7.tar.gz`, where `pkg install` is followed by name of downloaded package, in this case version 1.4.7, it is version dated January 10, 2026. Of course, in order for Octave to find it, it must be located in currently active directory, set in Octave's File Browser environment, or its name including its path would have to be specified.

In order for file to open via UIgetfile, you need to go to main menu of Octave and select: „EDIT/Preferences” and then check „Use native file dialogs” in Generál tab. Otherwise, loading non-default directories takes a long time. You only need to check this once, Octave remembers it.

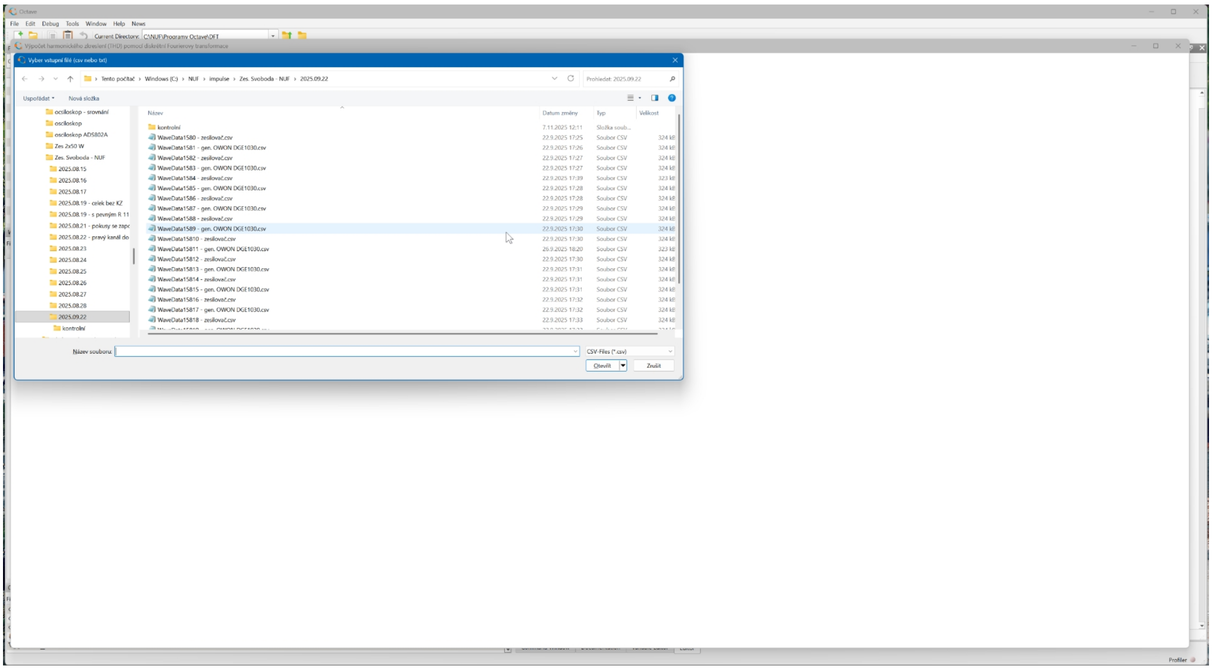
Source file of program is named „**Calculation_DFT.m**“.

Because I could not find an easy way to compile program into an executable form (.exe – it might be possible, but not easy, I don't know) in Octave environment running on Windows, it is provided here for download as a so-called .m file, which is a file type that Octave reads. These are text files that can be read anywhere, which is good on one hand because you can see what's in them, but on other hand, it can cause problems with unwanted, possibly random changes to program in Octave or another editor, and then program might not work at all or might work incorrectly. Therefore, I strongly recommend that you make a backup copy after downloading it, and do not run or open it, but only copy it to another directory if necessary and run it from there.

At first, I had problems with program window size (figure) because I could not set its size on my dual-monitor computer so that it would be suitable when running on a single-monitor computer. This was because I couldn't find a command in Octave that would return monitor resolution in so-called logical pixels, which are pixels of physical monitor resolution converted to the screen magnification scale used in Windows (in Windows settings, magnification scale is found in: System/Display). Over time, with help from so-called artificial intelligence, I made function for Octave that can detect this via PowerShell commands, so there shouldn't be any more problems with figure size, although the program can only do this when Octave is running on Windows (10 or 11). So, if Octave were run on different operating system or if an error occurred during monitor detection, which I was unable to catch, program will be each time it is launched of monitor parameters each time it was launched. I hope that won't happen. In case of incorrect manual resolution settings, figure may be displayed across both monitors, typically with the main menu invisible (outside the monitors). This can be corrected by right-clicking on visible part of window to display context menu and selecting "Close this window" After closing it, enter correct dimensions when restarting program (find them out in meantime).

Because in figure where program output is displayed, text output is also adjusted when changing its size (I don't know how to prevent this, so that instead of changes, scroll bars are added, as Visual Basic does 🤪), changing window size below a certain limit is restricted so that texts do not overlap, and windows automatically redraw to their minimum size after being resized below this limit (this may take a moment and cause flickering).

After running, program looks like this:



and offers a choice of files for THD calculation – see below for file descriptions.

After entering suitable file for calculation, program displays curve of its data and asks for description of measurement, which is only displayed in calculation for better identification. Then it displays complete calculation, which may look like example on following page.

In this example, calculation was performed for test file artificially created by calculation in another program as simulation with following parameters:

overall signal is sum of three sinusoidal waveforms:

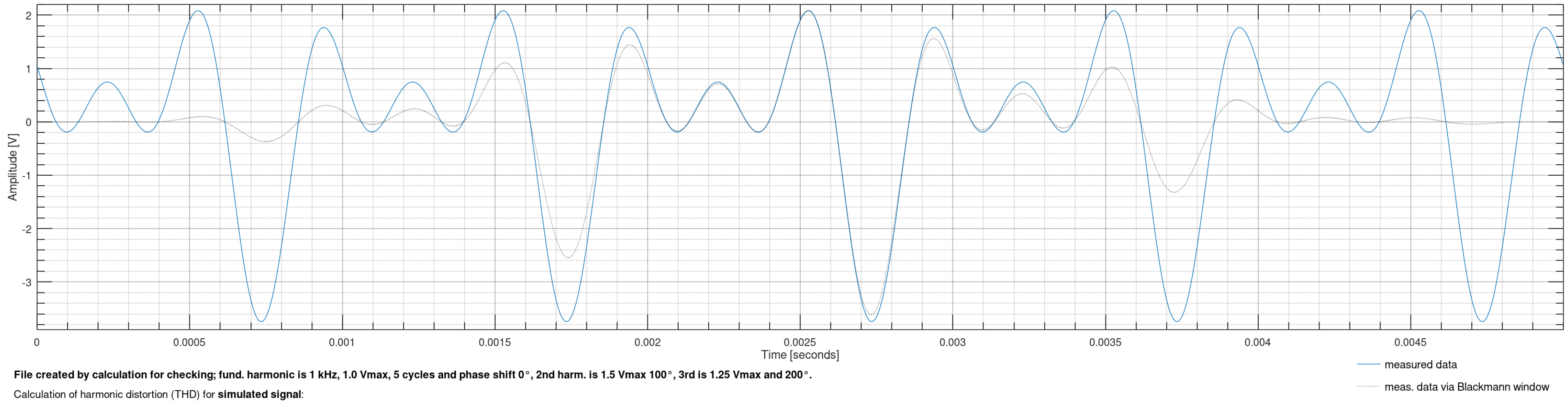
- primary harmonic: 1 kHz sine wave, max. amplitude 1 V, phase shift from beginning of file 0°, 5 cycles
- 2nd harmonic: 2 kHz sine wave, max. amplitude 1,5 V, phase shift from beginning of file 100°
- 3rd harmonic: 3 kHz sine wave, max. amplitude 1,25 V, phase shift from beginning of file 200°

As can be seen, calculated values of individual harmonics in case of processing entire file correspond to values from which it was calculated. If any option of program for selection and processing of entire cycles will be used, results of harmonics will not correspond exactly, because such selection has, among other things, a significant effect on phase of processed signal, see further text.

Calculation of harmonic distortion (THD) using discrete Fourier transform for file: sin 1000 Hz 1V na kmit 0st 5 kmitů a 2 harm 150 proc 100st a 3 harm 125 proc 200st eng.csv

DFT control Settings Editing charts

Measured data from file: C:\NUF\Programy Octave\DFT\sin 1000 Hz 1V na kmit 0st 5 kmitů a 2 harm 150 proc 100st a 3 harm 125 proc 200st eng.csv



File created by calculation for checking; fund. harmonic is 1 kHz, 1.0 Vmax, 5 cycles and phase shift 0°, 2nd harm. is 1.5 Vmax 100°, 3rd is 1.25 Vmax and 200°.

Calculation of harmonic distortion (THD) for **simulated signal**:
File is processed from record 1 do 5120 and has in total 5120 records.
Highest usable F (Nyquist) is 512.0 kHz, is simulation with max. resolution.
Entire file is processed; total 5 cycles.

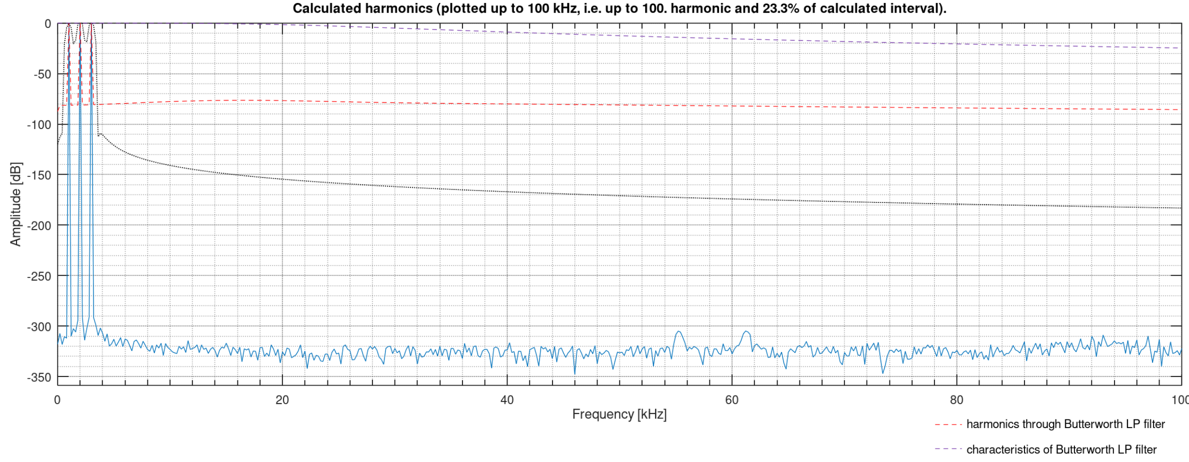
Effective value (RMS) of full processed signal:	1.5512 Vrms
Direct current component:	1.2859e-16 V
THD full proces. signal up to 426th harm., 426 kHz:	**195.26 %**
THD full proces. signal through Blackman window:	**195.26 %**
THD through order 2 Butterworth LP filter, Fo 25 kHz:	195.24 %
THD calc. from 2nd to 10th harmonics (up to 10 kHz):	195.26 %

List of harmonics (here only up to 10. harmonics):

	Frequency	Amplitude	Phase
Primary harmonic:	1000.0 Hz	1.0000 V	0.0000 °
2. harmonic:	2000.0 Hz	1.5000 V	100.000 °
3. harmonic:	3000.0 Hz	1.2500 V	200.00 °
4. harmonic:	4000.0 Hz	4.3959e-16 V	254.01 °
5. harmonic:	5000.0 Hz	2.2163e-16 V	155.35 °
6. harmonic:	6000.0 Hz	2.2879e-16 V	162.31 °
7. harmonic:	7000.0 Hz	2.0579e-16 V	179.16 °
8. harmonic:	8000.0 Hz	6.8526e-17 V	238.24 °
9. harmonic:	9000.0 Hz	8.2080e-17 V	177.08 °
10. harmonic:	10000 Hz	7.0362e-17 V	179.95 °

© Luděk Ruffer lruffer@volny.cz program version dated 1 April 2026

Calculated harmonics (plotted up to 100 kHz, i.e. up to 100. harmonic and 23.3% of calculated interval).



Input data

Program assumes input of a signal measured by a digital oscilloscope and stored (in it) in text file with either .csv or .txt extension, preferably on a USB flash drive. File can also be created by some program, provided that it will be the same as files from oscilloscope

Over time, I purchased two digital oscilloscopes because I found the first one, HANTEK DSO 4102 C with its 8-bit AD converter, to be insufficient, and because about five years after purchasing it, oscilloscopes with a 12-bit AD converter were available at an acceptable price, I bought a second oscilloscope with a 12-bit AD converter, OWON ADS802A. So, program can process files from both oscilloscopes and probably also from other oscilloscopes that create the same files. I don't know exactly what types these could be for HANTEK, but for OWON it could be entire ADS800/900 series, but I don't have opportunity to test this. However, saving to CSV files was not without problems for either of oscilloscopes mentioned, as I will describe below.

Oscilloscope HANTEK DSO 4102 C

My first HANTEK DSO 4102 C digital oscilloscope, which wasn't (relatively) very expensive when I bought it, which is visible in its features: firstly, it only has an 8-bit AD converter, which seemed acceptable to me at time of purchase, because oscilloscopes with better AD converters (10–14 bits) were significantly more expensive at the time. It was only when I started using it that I realized that 8 bits means only 256 output levels, which is (probably) quite a bit. That is why I think it is better to record multiple signal cycles (around 20) for waveforms because, for example, THD calculation from 4 oscillations is higher than from 20. Secondly, and more importantly, this oscilloscope adds some kind of "nonsenses" (which is, of course, an error of this type) to output files, which obviously cannot be present in measured signal. In my experience, this happens when writing to a file using a memory depth of 20 K, i.e. when 20 064 measurements are stored, when last 16 values at end of file almost always contain "bullshit" see end of one measured file copied here from "Notepad" (only first three faulty values from end are shown here, but there are always 16):

```
1.00E-03,0.240
1.00E-03,0.400
1.00E-03,0.320
1.00E-03,-2506.480
1.00E-03,2501.200
1.00E-03,2506.720
```

It is clearly visible that values differ by several orders of magnitude from previous three acceptable values. A sine wave with maxima of approximately 4 V was measured, not around 2500. However, such a difference is not always present; false values can also be around zero, but they are never related to measured signals. To make matters even "better", oscilloscope sometimes (but not often) saves faulty values at file's beginning. I don't do anything about this in THD calculation program; I just draw data, so I edit it manually in Notepad.

Another error in HANTEK DSO 4120 C oscilloscope recordings can be seen in previous example: each measurement is always stored in single line with dot as decimal separator and comma as value separator. First item is always time in seconds that has passed since start of measurement, second is measured value in V. Recording to file was probably done at Hantek by some "novices" and "clumsy fingers" who did not realize that expressing time to three valid digits is too little, it will soon overflow, and then a bunch of data will have the same time, which is nonsense. Program corrects this by calculating measurement intervals as difference between first two values in a file and corrects (calculates) time values accordingly, assuming that sampling interval is constant. This means that first two values MUST be different, increasing and containing correct oscilloscope sampling interval for given record. In the case of a difference of 0 (with identical values), program would not be able to calculate interval.

These are shortcomings I have found in csv recording on Hantek oscilloscope, and I no longer expect to be able to fix them by replaying oscilloscope firmware, which would help if it were fixed in new version. But in the roughly five years I've had oscilloscope; company's website still lists the same firmware version that I have on my oscilloscope. Since I bought this oscilloscope through Alibaba (https://www.aliexpress.com/) and defects can be eliminated (e.g., faulty signal values can be removed in notepad, even though it is irritating for each measurement), I did not submit complaint about them to seller.

Beginning of file looks like this (here are only first 10 lines):

```
#timebase=20000000(ps)
,#voltbase=2000000(mv/100)
#size=20064
5.00E-08,0.240
1.00E-07,0.240
1.50E-07,0.240
2.00E-07,0.320
2.50E-07,0.240
3.00E-07,0.320
3.50E-07,0.240
lines with numbers continue and have the same syntax until end of file
```

File contains 3 lines at beginning starting with characters # and ,# which have different values and I use (so far) only second one, starting with ,#voltbase=... which is vertical range of oscilloscope (i.e. across entire screen) and after multiplying by 1e-5 it is in Volts (in file it is in hundredths of a millivolt). Following lines are individual measurements stored in single line with dot as decimal separator and comma as value separator, and they repeat until end. Their number depends on oscilloscope sampling settings, in example, third line shows that there were 20064 measurements, while I count number of measurements (records) in program and do not use third line due to the above–described insertion of nonsense at end and beginning – after their manual removal there will be fewer records.

Oscilloscope OWON ADS802A

I bought this oscilloscope from company Banggood (https://www.banggood.com/cs/) on October 22, 2025, because it has 12-bit AD converter and I liked their price, even though they were shipping it from China. Oscilloscope was delivered in good condition on November 11, 2025. However, after testing it, I discovered a significant defect: when saving measured wave data to a .csv file, data was only saved correctly to file with the smallest number of data, i.e. for Acquire/Depth 1k setting, which saves 1000 data to file. For setting 10k, it only saved well a short distance from beginning, and rest of values were either close to zero or nonsense. After more detailed testing, I found that for greater depths I tried, it only saved first 2047 data points well, which is of course only acceptable at smallest depth of 1000 points, and that was not enough for me. It saved times well at all set depths.

That's why, about a week after delivery (i.e., November 17, 2025), I filed a complaint about defects at two locations in China, because I bought oscilloscope from Banggood in China:

1) Directly at company Owon (even though I didn't buy it there, but what if...) at general address from their website: info@owon.com.cn however, as I expected, there was absolutely NO response (i.e., until February 11, 2026 and I no longer expect to receive any reply).
2) At company Banggood, on page where I bought oscilloscope, of course after logging into my account with them: https://www.banggood.com/cs/ and here their approach was quite accommodating. Basically, if I disregard some "peculiarities" of people and "peculiarities" of their website, thanks to their company, I received working upgrade after more than a month (December 24, 2025). From those "peculiarities" the most important one is probably that customers MUST take some action within 7 DAYS, and if they do not (as in my case, when I was waiting for their response even though they wrote that I should wait until they sent upgrade and that it would take a while), they will CLOSE CONVERSATION... When I bought my oscilloscope, it had firmware version V1.0.1.0.17 (with CSV writing not working), and after about a week, Banggood sent me upgrade to version V1.0. 1.7.1, which I was unsuccessful in installing immediately due to incorrectly named sent file (as soon as I received file, I attempted to install it, oscilloscope recognized upgrade, but after launching it, it reported "Parse error"). After further discussion, Banggood sent me a video showing how to upgrade oscilloscope, which briefly showed correct file name. I renamed file accordingly and upgrade was successful, but writing to csv file did not work again... Finally, Banggood sent me email address for a specific lady from company OWON (I don't know if I can give her address here), and on December 11, 2025, I sent her a description of defect including video again, and she replied, although not immediately after sending email (December 16, 2025) that she would look into it and that I would have to wait. Then, on December 24, 2025, she sent me (as a gift??) an upgrade to version V1.0.1.7.2, which WORKS, hooray... (although it has some "bugs," for example, as can be seen from excerpt from beginning of csv file below, it still proudly displays firmware version V1.0.1.7.1 and not V1.0.1.7.2, which was the version used to upload file...).

This shows that it is possible to complain in China (maybe sometimes?), but it takes patience, not succumbing to their "peculiarities," and writing to them in English, as I did, because I don't know how they would cope with my native Czech, although they might translate it somehow. In this reclamation, I also had advantage of not having to request a physical replacement of device or any monetary compensation, since everything could be resolved online by sending a fixed firmware update – as I had suggested to them from start (I don’t know if they would have suggested sending update on their own).

Beginning of file of OWON ADS800/900 series oscilloscopes looks like this:

```
Model,ADS802A
Firmware Version,V1.0.1.7.1

Horizontal Units,s
TimeBase,0.0000100
Horizontal,0.0
Sample Interval,2.0E-7
Record Length,100000

Channel,CH1,CH2
Probe attenuation,1.0X,1.0X
Vertical Offset,0.00div,0.00div
vertical Scale,50.00mV,100.0mV
Label,?,?
Frequency,1.000kHz,872.7Hz
Period,1.000ms,1.145ms
PK-PK,316.5mV,409.0mV
Average,335.9µV,375.0µV
Vertical pos,0,0
Per ADC value,50.000000,100.000000

TIME(Units:s),CH1(Units:V),CH2(Units:V),
-0.0100000,0.003,-0.003
-0.0099998,0.004,-0.003
-0.0099996,0.003,-0.003
-0.0099994,0.002,-0.006
lines with numbers continue and have the same syntax until end of file
```

I can only assume that program is compatible with entire ADS 800/900 series, as I have ADS802A oscilloscope and have not (yet) had opportunity to try files from other oscilloscopes in this series, but hopefully they will be the same.

Program can process one to ten channels (for OWON oscilloscopes), even though this type of oscilloscope can have a maximum of four analog channels. Program assumes that data is saved to file as "Wave" type and that any combination of channels has been set as source, but NOT MATH, DIR, and FFT types, which I have not tried yet. Example shows file with two channels.

File begins with 22 lines of text describing type, settings and some measured values of oscilloscope, whose meaning is quite clear from file. I (so far) only use values "Channel, ..." which is identification of recorded channels, "Model," which is oscilloscope model, "(Units:" followed by units for individual recorded data, and "vertical Scale, ...", which is setting of vertical sensitivity of oscilloscope in specified units per 1 vertical div of screen, and because screen has 8 dives vertically, so after multiplying this value by 8, I use it in output for OWON oscilloscopes as oscilloscope range.

Rows containing measured data have a comma as separator between values, and dot as decimal separator. First number is sampling time from start [in seconds], oscilloscope stores time data such that 0 is at the middle of recording, so times at beginning of file are negative; program converts this them so that time in recording starts by zero. Second and any further numbers (after commas) are measured values of individual channels.

Because oscilloscopes allow multiple channels to be written to file in form of additional values added to each line (ADS802A/ADS812A/ADS822A are dual-channel and ADS804A

ADS814A ADS824A are four-channel) if program detects multiple channels in file, it will ask which one to process, and that channel will be processed further.

Text file generated by another program

Text file generated by another program based on Hantek DSO4102C oscilloscope files, e.g., with calculated signal for testing this program.
Such a file begins, for example, as follows:

```
#timebase=19000000000(ps)
,#voltbase=2.5(V); is simulated resolution of 12bit AD converter from range ±10V
#size=19000
0,0.002442
1E-06,0.01709
2E-06,0.03175
lines with numbers continue and have the same syntax until end of file
```

or like this:

```
#timebase=1000000000(ps)
,#voltbase=10(V); it is simulated by calculation with maximum possible accuracy of
              simulation program.
#size=19000
0,0
5.26315789473684E-08,0.00330693957508402
1.05263157894737E-07,0.0066138787885261
lines with numbers continue and have the same syntax until end of file
```

Notes on sample are the same as in description of Hantek DSO4102C oscilloscope file, and there can be only one channel here as well.

File begins with three lines beginning with characters # and ,# similar to Hantek DSO4102C oscilloscope file, the difference being in text added after semicolon in second line. I created program for generating such file in MS Visual Studio, which text adds automatically.

Program searches for simulation according to keyword "simulated" (without quotation marks) in second line after semicolon, and so if file is not measured by oscilloscope but otherwise generated, MUST be there this keyword, and then program assumes that it is simulation.

There are two examples of simulation here:

- the first, where keyword is followed by text containing two numbers: first defines how many bits calculated signal is quantized (divided), i.e. what is simulated resolution of AD (analog-to-digital) converter, and second number indicates range from which it is quantized (range of AD converter). Program reads these two numbers in the order given in example (their order is a condition for correct determination) from file and prints them to output. As can be seen in example, first recorded signal value is not zero, and values are recorded with fewer valid digits than in second example, because in this case 12-bit resolution was simulated, i.e. AD converter range is divided into 4096 levels. Note: with frequently used cheaper oscilloscopes uses 8-bit resolution of AD converters, signal only has 256 levels.
- the second, where there is no number after keyword, and therefore program assumes that it is purely a calculated signal at max. precision of program that calculated it. Maximum precision is also visible in number of digits, to which values are written.

Modification of measured data and calculations

Program calculates total harmonic distortion THD for all higher harmonics (which it is able to detect) and last harmonic used, including its frequency, is shown in output in line "THD full process. signal up to xxxth harm., yyy kHz:".

All harmonics are calculated using Fourier transform. Octave has function fft(..) for this purpose, which (following is taken from Octave documentation) "Compute the discrete Fourier transform of x using a Fast Fourier Transform (FFT) algorithm".)“. Fourier transform is based on Fourier's theory, which states that continuous signal can be decomposed into trigonometric series (sines and cosines) with amplitudes a_n (even harmonics), b_n (odd harmonics) and DC component $\frac{a_0}{2}$, phase shifts φ_n and integer multiples of fundamental angular frequency ω, where „n“ are individual harmonics:

$$f(t) \approx \frac{a_0}{2} + \sum_{n=1}^{\infty} \left(a_n \cos(n\omega t + \varphi_n) + b_n \sin(n\omega t + \varphi_n) \right); \quad where\ \omega = \frac{2\pi}{T} = 2\pi f$$

In order for Fourier transform (expansion) to be applicable, function must satisfy following conditions (Dirichlet conditions): **1. *f(t)* is periodic**, **2.** within specified interval (one period), *f(t)* must has only finite number of discontinuities, **3.** within a given interval, function must have a finite number of extrema, and **4.** function must be defined at endpoints of interval (i.e., it must take finite values at them). Since processed signal is not continuous but is sampled by oscilloscope with certain sampling frequency f_V, program can determine harmonics only up to frequency $f_V/2$ (Nyquist frequency). Here, Fourier transform converts signals from time domain to frequency domain. For example, oscilloscope recording, where measured values are stored as they were measured, i.e. with a constant time step and time is independent variable, converts them to values depending on frequency, where frequency is independent variable. And because signals from oscilloscopes, from which FFT is calculated here, are voltages, calculated harmonics are also voltages and THD is:

$$THD_V = \frac{\sqrt{V_2^2 + V_3^2 + V_4^2 + \cdots + V_n^2}}{V_1} * 100\ [\%]$$

Well, I had quite a bit of trouble with periodicity of signal measured by HANTEK oscilloscope from very beginning, because I couldn't find a way to tell oscilloscope to record only full cycles – probably it can't do that, it's not as simple as it might seem, but OWON ADS802A oscilloscope stores data with fairly acceptable periods, although not always. If signal processed using FFT contains a non-repeating segment at beginning, end, or both, calculated harmonics and THD from harmonics will not correspond exactly to measured signal, because this „tail” will generate additional harmonics that are not present in original signal, or will affect harmonics already present in signal.

In order to eliminate these unwanted "tails," or at least suppress their significance, oscilloscopes that can display FFT waveforms of the scanned signal commonly use so-called windows, which process measured signals (before FFT calculation) is processed by passing it through mathematical filters that attenuate the measured signal at both beginning and end of waveform in different ways, either to zero or significantly reducing it, while preserving original values in the middle of captured waveform. There are several types of windows, each with certain properties that make them more suitable for some purposes and less suitable for others. For example, my oscilloscopes use Hanning, Flattop, and Rectangular windows. Hanning and Flattop windows are better suited for processing periodic waveforms – Hanning is better for frequencies and Flattop for amplitudes – while a Rectangular window is better for pulse waveforms. Octave has quite a few options for calculating these windows in the Signal package. For comparison, I built into program option to modify input signal via Blackman window. Modified signal and harmonics calculated from it are drawn in graphs with black dotted line, THD calculation from it is shown in one line in results.

Another option for removing "tails" is to adjust of measured signal to full cycles, whereby it does not matter where waveform begins and ends for THD calculation (however, beginning and end must be at the same point on the waveform) but start of signal has significant effect on phase of all components. Because I designed this program for use in THD calculations from measurements taken on acoustic devices (amplifiers, equalizers, etc.), and THD in tens of percent and higher is not expected here, I used as main method in the program limitation of waveform to whole cycles from first zero crossing of signal. Program also includes two more methods of selecting whole cycles: from first signal maximum and from beginning of file. These methods can achieve slightly lower THD than calculations using windows, but they also have their disadvantages: even though I have made selections on whole oscillations in the program as best I was capable to doing, in results, whole cycles may not always be well selected, so it is necessary to check this from graph of measured signal. This is because selections work for more or less sinusoidal signals up to a certain level of distortion, or in other words, they work when signal has a predictable shape and direction (decreasing/increasing) that can be determined in some way, and this is the case even after signals have passed through some filter, e.g., a noise suppression filter. For example, in waveforms from a HANTEK oscilloscope (which only has an 8-bit AD converter), noise manifests itself in that waveforms are quite "hairy", i.e., values fluctuate around a certain level several times depending on steepness of waveform, but it seems that program can handle selection for such a signal. I also recommend that waveforms measured with oscilloscopes with "hairy" waveform (8-bit AD converter) have about 20 cycles for processing by this program, because for such a number of cycles THD is slightly smaller than for example for 4 cycles.

Program cannot detect whole cycles, for example in signal shown in example at beginning of this document, which consists of three frequencies with comparable amplitudes, because it contains more signal changes than correct number of cycles. In such cases, program will not allow these selections in menu, and these items will begin with "NOT POSSIBLE –" Other conditions for program to allow these selections are that one cycle must have at least 300 measured points, or signal must have at least 3 cycles, or calculated THD must be less than 25%. If someone wants to process a waveform that does not meet these conditions and does not start and end in the same place so that processing corresponds to its periodicity, they would have to edit this file in another program (Excel, Notepad), as program is not (yet) able to manually select data interval for processing.

As regards resulting THD, I recommend following procedure: since THD is calculated from a waveform measured by an oscilloscope, which is obtained by passing a signal generated by a generator, which also has its own harmonic distortion, through measured device, it is quite logical that resulting THD of signal measured in this way will be sum of THD of generator signal and THD of measuring device, and disturbance induced into connecting wires may also be added to THD (although it should not really be the case). I assume that the best procedure is to measure THD for the entire generator system with the device under the same conditions (connect oscilloscope input to generator output and leave grounding connected to powered device) and also calculate THD for signal from generator. Repeat this for each point at which THD of device is measured. Then resulting THD for each point will be:

$$THD_{device} = THD_{whole\ chain} - THD_{generator}$$

Described procedure does not take into account errors that may be introduced into calculation by oscilloscope and calculations in this program, but I assume that they will be very similar in both measurements (oscilloscope should have vertical gain set so that signal-to-total vertical gain ratio is approximately the same for measuring generator itself and entire chain), so they should be eliminated by subtraction and should not have a significant effect on result.

When measuring THD in this way, however, it is necessary that frequency response of measured device is affected only by what is to be measured, i.e. for example that there are no limiting filters connected in device, which would be undesirable for THD. I had this experience when measuring THD of parametric equalizer consisting of four controllable parametric filters and upper and lower pass filters. Although it was possible to control breakpoint of pass filters, but they could not be disconnected. Well, since only higher harmonics are used to calculate THD, it was problematic that the low-pass filter had the highest adjustable breakpoint at approximately 25 kHz with a slope of 12 dB/octave, and when measuring entire system (equalizer with this filter + generator), THD of whole system was lower than THD of generator itself (generator was not limited by any low-pass filter), so THD of equalizer was negative, which is obviously nonsense. The reason was that signal passing through corrector had its highs limited by its low-pass filter, while signal from generator had no such limitation, its signal contained higher harmonics with greater amplitude, and therefore had greater THD. The solution was to create a passive low-pass filter of the same order and response as one in a corrector and measure signal from generator through it.

Based on this experience, I added an optional feature into program that allows signal to be run through a programmed Butterworth low-pass filter, whose order and cutoff frequency at 3 dB can be adjusted within a specific range. THD calculated using this filter is shown in separate row for sake of comparison. In a graph of harmonics, waveforms of harmonics passing through this low-pass filter are drawn in red dotted lines, and his characteristic (frequency response) are drawn in purple dotted lines. Waveforms are only drawn here when a calculated harmonic from base signal (blue line) differs from harmonics through low-pass filter at least enough so that waveforms do not partially overlap and do not "hide" behind each other.

According to my testing of low-pass filter in program (and also according to THD calculation logic), its cutoff frequency should be selected to be greater than at least 3rd harmonic of basic measured signal, because if it will be smaller, corresponding harmonics (3rd and 2nd) will also be smaller and THD calculated this way will differ significantly from reality, but question is what is required from measurement. Interestingly, however, when measuring generator through a hardware low-pass filter with a cutoff frequency of approximately 25 kHz, I was measure THD of generator even at base frequency 20 kHz, and calculated values seemed realistic, so I don't know what difference was, and use of a low-pass filter in program is up to everyone.

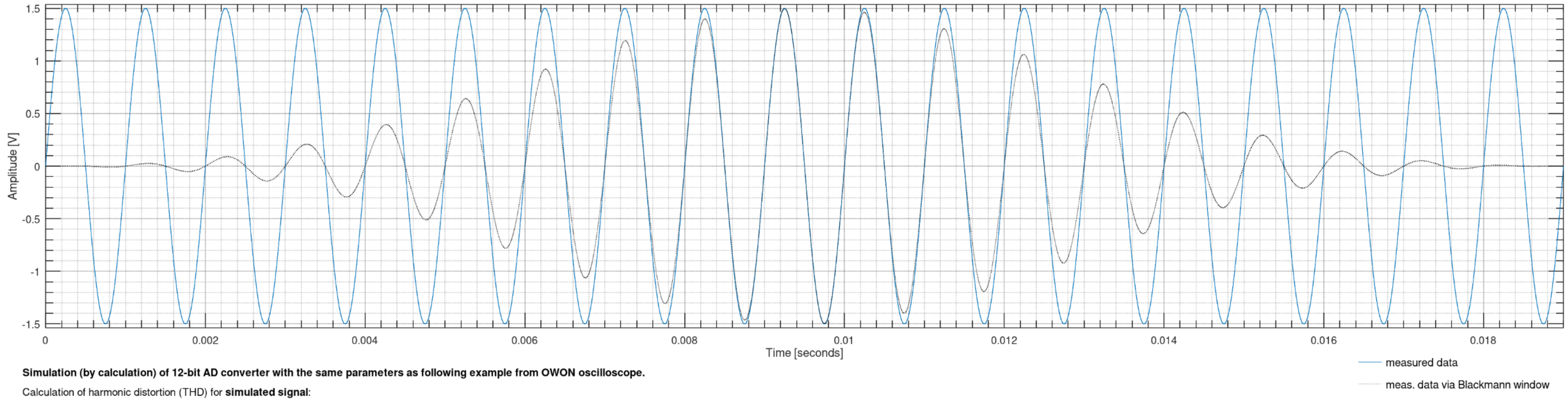
Difference among oscilloscopes with 8 and 12-bit converters

Although THD calculation method described above should eliminate distortion described below, for the sake of interest, I am presenting both THD of calculated signal with simulated 12-bit AD converter and THD of the same signal measured by oscilloscopes with 12-bit and 8-bit AD converters. Because 8-bit converter divides (quantizes) analog signal into (only) 256 levels and 12-bit converter into 4096 levels, there is, of course, a difference in accuracy between signals measured using these two methods. The number of levels always relates to set total vertical range of oscilloscope and measured signal will always be smaller than vertical range, so uncertainty related to measured signal will be greater. Below are examples of simulated signal calculations as a 12-bit converter and signals captured by oscilloscopes with 12-bit and 8-bit converters. In both cases of measured signal, signal from OWON DGE1030 generator was used, which according to documentation has harmonic distortion for range from DC to 1 MHz < 65 dBc, converted to percentage is 0.05623% (according to website https://sengpielaudio.com/calculator-thd.htm).

Calculation of harmonic distortion (THD) using discrete Fourier transform for file: simulation 1 kHz 19 cycles 1,5 V max from 4 V 12 bit.csv

DFT control Settings Editing charts

Measured data from file: C:\NUF\impulse\ocsiloskop - srovnání\simulation 1 kHz 19 cycles 1,5 V max from 4 V 12 bit.csv



Simulation (by calculation) of 12-bit AD converter with the same parameters as following example from OWON oscilloscope.

Calculation of harmonic distortion (THD) for **simulated signal**:
File is processed from record 1 do 95000 and has in total 95000 records.
Highest usable F (Nyquist) is 2500 kHz, is simulation 12 bits from 4 V.
Entire file is processed; total 19 cycles.

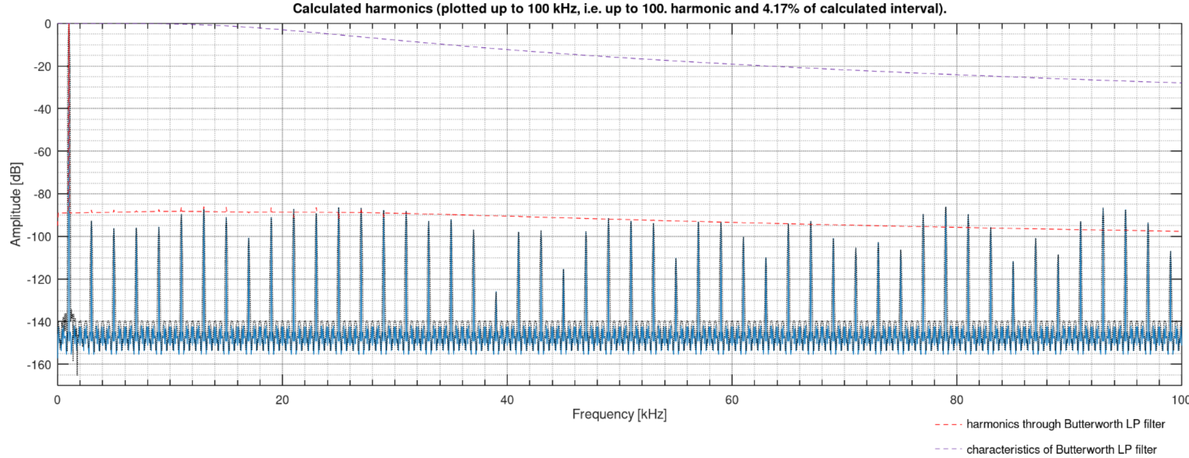
Effective value (RMS) of full processed signal:	1.0606 Vrms
Direct current component:	2.0564e-08 V
THD full proces. signal up to 2375th harm., 2375 kHz:	0.055648 %
THD full proces. signal through Blackman window:	**0.055647 %**
THD through order 2 Butterworth LP filter, Fo 20 kHz:	0.031044 %
THD calc. from 2nd to 10th harmonics (up to 10 kHz):	0.0035986 %

List of harmonics (here only up to 10. harmonics):

	Frequency	Amplitude	Phase
Primary harmonic:	1000.0 Hz	1.5000 V	0.0000 °
2. harmonic:	2000.0 Hz	4.1128e-08 V	270.00 °
3. harmonic:	3000.0 Hz	3.4613e-05 V	358.91 °
4. harmonic:	4000.0 Hz	4.1128e-08 V	270.00 °
5. harmonic:	5000.0 Hz	2.3167e-05 V	181.63 °
6. harmonic:	6000.0 Hz	4.1128e-08 V	270.00 °
7. harmonic:	7000.0 Hz	2.3575e-05 V	358.40 °
8. harmonic:	8000.0 Hz	4.1128e-08 V	270.00 °
9. harmonic:	9000.0 Hz	2.4960e-05 V	181.51 °
10. harmonic:	10000 Hz	4.1128e-08 V	270.00 °

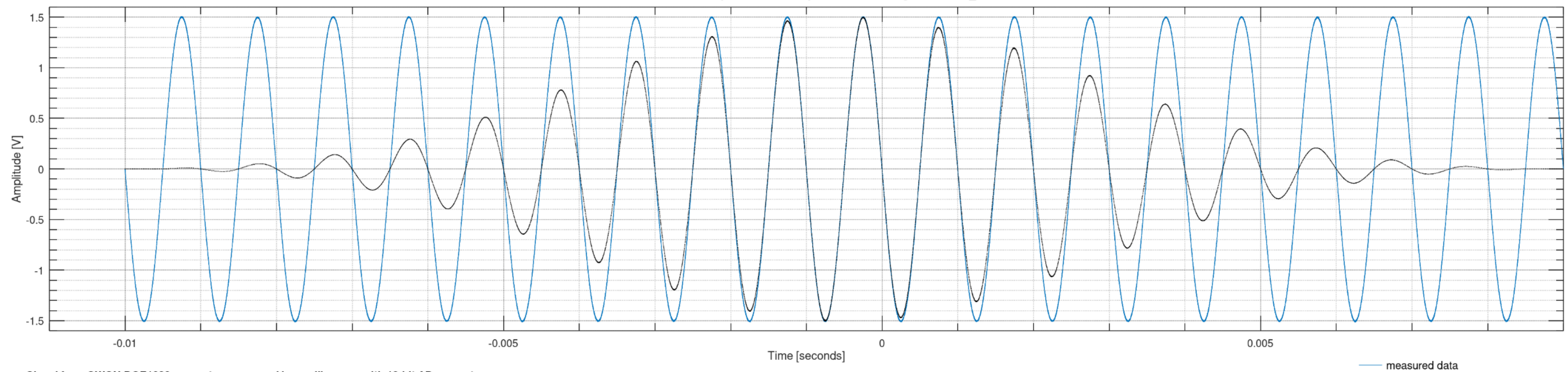
© Luděk Ruffer   lruffer@volny.cz   program version dated 1 April 2026

Calculated harmonics (plotted up to 100 kHz, i.e. up to 100. harmonic and 4.17% of calculated interval).



Calculation of harmonic distortion (THD) using discrete Fourier transform for file: OSCgen 1000 Hz_20260131132727.csv

DFT control Settings Editing charts



Signal from OWON DGE1030 generetor, measured by oscilloscope with 12-bit AD converter.

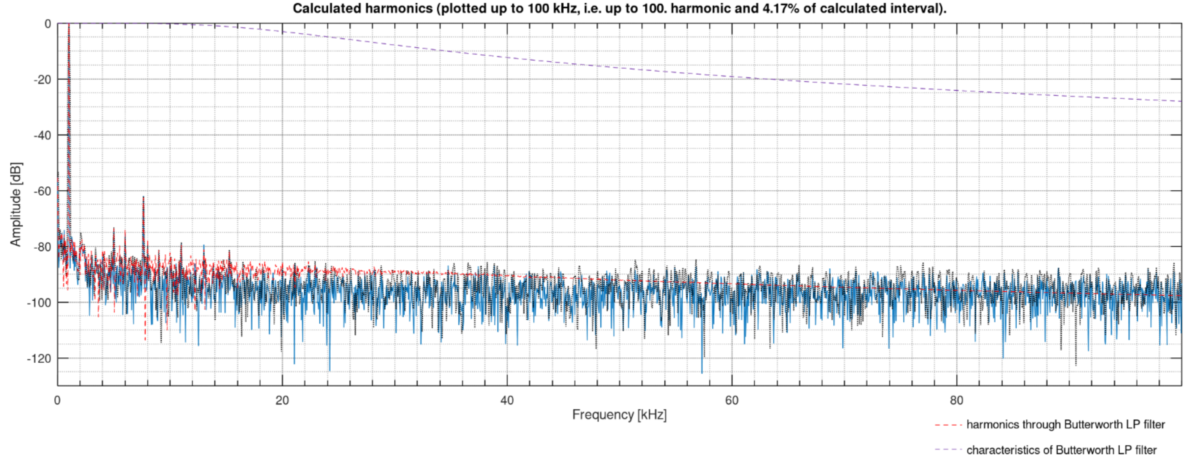
Calculation of harmonic distortion (THD) for oscilloscope **OWON ADS802A**:
File is processed from record 592 do 95591 and has in total 100000 records.
Highest usable F (Nyquist) is 2500 kHz, oscilloscope range 4 V, channel CH1
Whole cycles from first zero crossing are processed; total 19 cycles.

Effective value (RMS) of full processed signal:	1.0624 Vrms
Direct current component:	0.0025975 V
THD full proces. signal up to 2375th harm., 2375 kHz:	**0.082133 %**
THD full proces. signal through Blackman window:	0.10377 %
THD through order 2 Butterworth LP filter, Fo 20 kHz:	0.047965 %
THD calc. from 2nd to 10th harmonics (up to 10 kHz):	0.031829 %

List of harmonics (here only up to 10. harmonics):

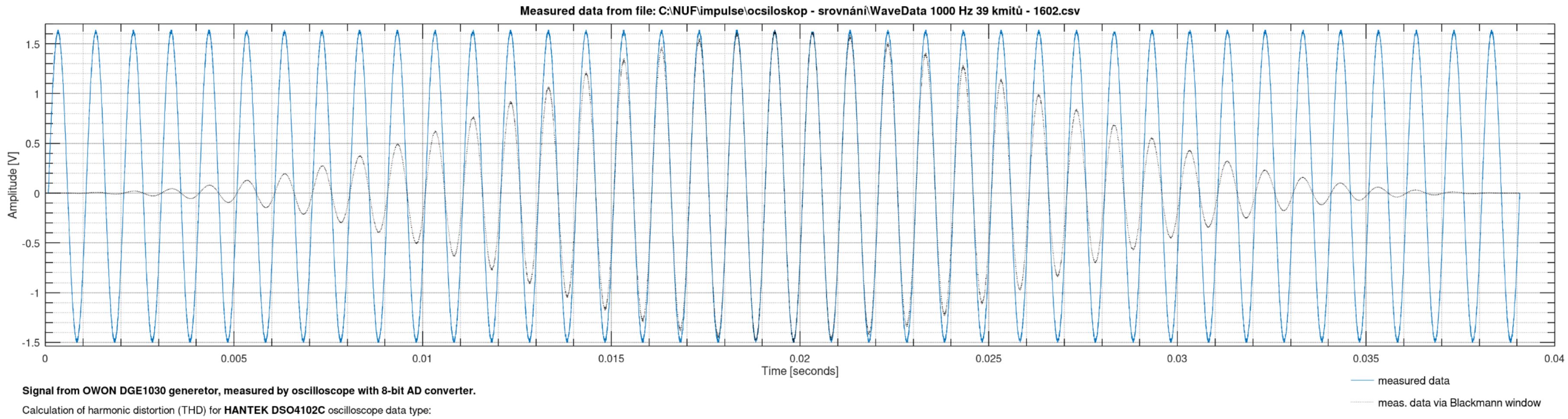
	Frequency	Amplitude	Phase
Primary harmonic:	1000.00 Hz	1.5025 V	179.85 °
2. harmonic:	2000.0 Hz	9.5570e-05 V	279.49 °
3. harmonic:	3000.0 Hz	8.6200e-05 V	140.79 °
4. harmonic:	4000.0 Hz	9.7583e-05 V	19.765 °
5. harmonic:	5000.0 Hz	3.0538e-04 V	356.64 °
6. harmonic:	6000.0 Hz	2.5127e-04 V	77.606 °
7. harmonic:	7000.0 Hz	5.5003e-05 V	13.774 °
8. harmonic:	8000.0 Hz	1.5822e-04 V	116.39 °
9. harmonic:	9000.0 Hz	1.1750e-04 V	358.41 °
10. harmonic:	10000.0 Hz	6.5950e-05 V	113.92 °

© Luděk Ruffer lruffer@volny.cz program version dated 1 April 2026



Calculation of harmonic distortion (THD) using discrete Fourier transform for file: WaveData 1000 Hz 39 kmitů - 1602.csv

DFT control   Settings   Editing charts



Signal from OWON DGE1030 generetor, measured by oscilloscope with 8-bit AD converter.

Calculation of harmonic distortion (THD) for **HANTEK DSO4102C** oscilloscope data type:

File is processed from record 40 do 19535 and has in total 20048 records.

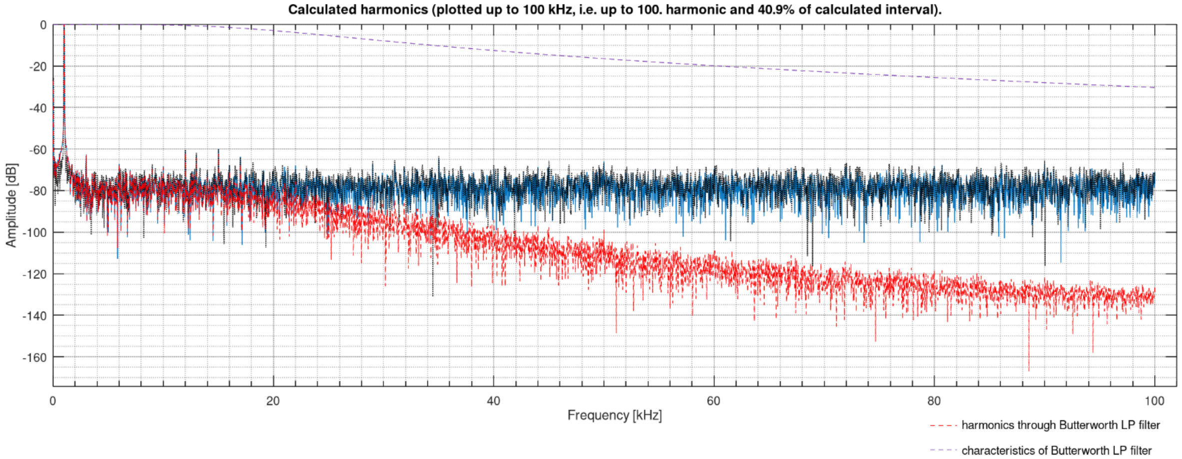
Highest usable F (Nyquist) is 250.0 kHz, oscilloscope range 5 V

Whole cycles from first zero crossing are processed; total 39 cycles.

Effective value (RMS) of full processed signal:	1.0981 Vrms
Direct current component:	0.067686 V
THD full proces. signal up to 243th harm., 243 kHz:	**0.32300 %**
THD full proces. signal through Blackman window:	0.38494 %
THD through order 2 Butterworth LP filter, Fo 20 kHz:	0.17982 %
THD calc. from 2nd to 10th harmonics (up to 10 kHz):	0.10766 %

List of harmonics (here only up to 10. harmonics):

	Frequency	Amplitude	Phase
Primary harmonic:	1000.2 Hz	1.5498 V	358.57 °
2. harmonic:	2000.4 Hz	1.8805e-04 V	276.13 °
3. harmonic:	3000.6 Hz	0.0011417 V	175.26 °
4. harmonic:	4000.8 Hz	1.6507e-04 V	272.33 °
5. harmonic:	5001.0 Hz	1.5307e-05 V	146.61 °
6. harmonic:	6001.2 Hz	6.7198e-04 V	248.32 °
7. harmonic:	7001.4 Hz	4.8375e-04 V	147.42 °
8. harmonic:	8001.6 Hz	2.2400e-04 V	248.64 °
9. harmonic:	9001.8 Hz	5.7219e-04 V	160.93 °
10. harmonic:	10002 Hz	5.9534e-04 V	70.872 °



© Luděk Ruffer   lruffer@volny.cz   program version dated 1 April 2026