

```

1
2
3 #Calculation of harmonic distortion using Fourier transform,
4 #here from the following two types of text files (txt or csv):
5 #
6 #
7 # 1a)Text file from the Hantek DSO4102C oscilloscope and other ↵
  oscilloscopes
8 #   or files created in other ways, whose output file begins as follows:
9 %
10 %   #timebase=400000000(ps)
11 %   ,#voltbase=1000000(mv/100)
12 %   #size=20064
13 %   1.00E-06,-2.560
14 %   2.00E-06,-2.560
15 %   lines with numbers continues and have the same syntax until end ↵
  of file
16 %
17 #   Note: Since octave also uses hash (#) symbol for comments, and first
18 #   three lines of file begin with this symbol, I have used percent ↵
  sign (%)
19 #   to mark the beginning of file for comments. In file, these ↵
  lines begin
20 #   with symbols # and ,# or with numbers, and do not contain any ↵
  percent
21 #   signs with spaces at the beginning.
22 #
23 #   Here, the program assumes ONLY ONE channel recording!!
24 #
25 #   For files from Hantek DSO4102C oscilloscope, program assumes that
26 #   they ALWAYS contain only ONE channel recording, even though ↵
  oscilloscope
27 #   can record up to two (or up to four for a slightly different type)
28 #   channels. !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
29 #
30 #   At the beginning of file, there are three lines beginning with
31 #   characters # and ,# which have different values. I am (currently) ↵
  only
32 #   using second one, beginning with ,#voltbase=... This is vertical ↵
  range
33 #   of oscilloscope (i.e., across entire screen), and after multiplying
34 #   by 1e-5, it is in volts (in file, it is in hundredths of millivolt).
35 #
36 #   Following lines contains data, and there are two lines as an ↵
  example here.
37 #   In file, they are located after spaces with % indicating notes,
38 #   i.e., they begin with digits. Value separator is a comma, and ↵
  decimal
39 #   separator is a dot. First number is sampling time from the start
40 #   [in seconds], and second number (after comma) is measured value.
41 #
42 #   Note on sampling times: since output to the file on Hantek DSO4102C
43 #   oscilloscope was probably done by some clowns or novices or who ↵
  knows
44 #   who, they only did it to 3 valid digits, which is nonsense, ↵
  because

```

```
45 # in that exponential form, those 3 digits will soon be filled up ↵
and
46 # then a bunch of values will have the same time, so graph drawn ↵
from
47 # it is also nonsense !!!!!!
48 # To eliminate this, I calculate interval from the first two times
49 # (it's still okay there, but I don't know what would happen if ↵
interval
50 # was greater than significant 3 digits - it would have to be ↵
expressed
51 # in more digits than 3, and I don't know if that's possible) and ↵
with
52 # calculated interval, I calculate times from first time again and
53 # use the calculated times instead of times read from file.
54 #
55 # Or if someone edited a file from an oscilloscope in Notepad or ↵
similar,
56 # trimming the beginning to where above-described limitation occurs,
57 # then program would not calculate the timing correctly: if first ↵
two
58 # time values are the same, program will incorrectly calculate ↵
sampling
59 # interval (it will be = 0), and that would be big laugh !!!!!!
60 #
61 #
62 #
63 # 1b)A text file generated by another program based on Hantek DSO4102C
64 # oscilloscope files, e.g., with a calculated signal for program ↵
testing.
65 # Such a file begins, for example, as follows:
66 %
67 % #timebase=19000000000(ps)
68 % ,#voltbase=2.5(V); is simulated resolution of 12bit AD ↵
converter from range ±10V
69 % #size=19000
70 % 0,0.002442
71 % 1E-06,0.01709
72 % 2E-06,0.03175
73 % lines with numbers continues and have the same syntax until end ↵
of file
74 %
75 # or like this:
76 %
77 % #timebase=10000000000(ps)
78 % ,#voltbase=10(V); it is simulated by calculation with maximum ↵
possible accuracy of simulation program.
79 % #size=19000
80 % 0,0
81 % 5.26315789473684E-08,0.00330693957508402
82 % 1.05263157894737E-07,0.0066138787885261
83 % lines with numbers continues and have the same syntax until end ↵
of file
84 %
85 # Notes on example are the same as in point 1a), and there may also
86 # be only ONE channel here.
87 #
```

```

88 # File contains three lines beginning with characters # and ,# similar
89 # to point 1a), difference being in the text added after semicolon
90 # in second line. I created a program for generating such a file in
91 # MS Visual Studio, which adds text automatically.
92 #
93 # Program searches for a simulation by keyword "simulated" (without
94 # quotation marks) in second line, and therefore, if it is a signal ↵
that
95 # is not measured by an oscilloscope but generated in another way, ↵
this
96 # word MUST be there, and then program assumes that it is a ↵
simulation.
97 #
98 # There are two examples of simulation here:
99 # - first, where keyword is followed by text containing two numbers:
100 # first defines how many bits calculated signal is quantized ↵
(divided)
101 # into, i.e., what simulated resolution of the AD ↵
(analog-to-digital)
102 # converter is, and second number indicates range from which it
103 # is quantized (the range of the AD converter). Program reads ↵
these
104 # two numbers in the order given in example (their order is a ↵
condition
105 # for correct determination) from file and writes them to the ↵
output.
106 # As can be seen in example, first recorded signal value is ↵
not zero,
107 # and values are recorded with fewer valid digits than in second
108 # example, because in this case a resolution of 12 bits was ↵
simulated,
109 # i.e., the AD converter range is divided into 4096 levels.
110 # Note: With frequently used 8-bit resolution of AD converters
111 # in cheaper oscilloscopes, signal has only 256 levels.
112 #
113 # - second, where there is no number after the keyword, and therefore
114 # program assumes that it is purely a calculated signal at ↵
maximum
115 # accuracy of program that calculated it. Maximum accuracy can ↵
also
116 # be seen in number of digits to which values are written.
117 #
118 #
119 #
120 #
121 # 2) Text file from OWON ADS800/900 series oscilloscopes and other
122 # oscilloscopes or files created in other ways, whose output file
123 # begins as follows:
124 %
125 % Model,ADS802A
126 % Firmware Version,V1.0.1.7.1
127 %
128 % Horizontal Units,s
129 % TimeBase,0.0000100
130 % Horizontal,0.0
131 % Sample Interval,2.0E-7

```

```

132 % Record Length,100000
133 %
134 % Channel,CH1,CH2
135 % Probe attenuation,1.0X,1.0X
136 % Vertical Offset,0.00div,0.00div
137 % vertical Scale,50.00mV,100.0mV
138 % Label,?,?
139 % Frequency,1.000kHz,872.7Hz
140 % Period,1.000ms,1.145ms
141 % PK-PK,316.5mV,409.0mV
142 % Average,335.9µV,375.0µV
143 % Vertical pos,0,0
144 % Per ADC value,50.000000,100.000000
145 %
146 % TIME(Units:s),CH1(Units:V),CH2(Units:V),
147 % -0.0100000,0.003,-0.003
148 % -0.0099998,0.004,-0.003
149 % -0.0099996,0.003,-0.003
150 % -0.0099994,0.002,-0.006
151 % lines with numbers continues and have the same syntax until end ↵
of file
152 %
153 # Note: Here, I have again used the % character for comments at ↵
beginning
154 # of file. In file, lines begin with text or numbers and do not ↵
contain
155 # any % characters with spaces at beginning.
156 #
157 # I can only assume that program is compatible with entire ADS 800/900
158 # series, as I have an ADS802A oscilloscope and have not (yet) had
159 # opportunity to test files from other oscilloscopes in this series,
160 # but hopefully they will be the same.
161 #
162 # There can be one to ten channels here, although this type of ↵
oscilloscope
163 # can have a maximum of four analog channels. Program assumes that ↵
data
164 # is saved to file as "Wave" type and that any combination of ↵
channels has
165 # been set as the source, but NOT MATH, DIR, and FFT types, which I ↵
have
166 # not tried yet. Example shows a file with two channels.
167 #
168 # At beginning, file contains 22 lines of text describing type and ↵
settings
169 # and some measured values of oscilloscope, the meaning of which is ↵
quite
170 # clear from file. I (so far) only use values "Channel,..." which is
171 # identification of recorded channels, "Model," which is model of
172 # oscilloscope, "(Units:" followed by units for individual recorded ↵
data,
173 # and "vertical Scale,...", which is set vertical sensitivity of ↵
oscilloscope
174 # in these units per 1 vertical div of screen, and since screen has ↵
a total
175 # of 8 divs vertically, I multiply this value by 8 and enter it in ↵

```

```
output
176 # of OWON oscilloscope program as oscilloscope range.
177 #
178 # Following lines look like this (there are four examples here, and ↵
again
179 # in file they are after the % symbols indicating comments, i.e., they
180 # start with numbers): value separator is a comma, and decimal ↵
separator
181 # is a dot. First number is time from start [in seconds] and second ↵
and any
182 # subsequent numbers (after the comma) are measured values of ↵
individual
183 # channels.
184 #
185 # Because oscilloscope allows multiple channels to be written to a ↵
file
186 # in form of additional values added to each line
(ADS802A/ADS812A/ADS822A
187 # are dual-channel and ADS804A ADS814A ADS824A are quad-channel), ↵
if program
188 # detects more channels in file, it will ask which one to process, ↵
and that
189 # channel will be processed further.
190 #
191 #
192 # Note on WRITING TO .CSV FILES: When I wrote something about ↵
clowns and
193 # the like in relation to HANTEK oscilloscope, I don't know what ↵
to call
194 # it here, because there were far greater problems with writing ↵
to these
195 # files (but also to .zip and .mat) that I complained about, ↵
namely that
196 # after delivery, oscilloscope only recorded first 2047 data points
197 # correctly in these files (times were correct throughout entire ↵
range),
198 # and then there were almost constant random values (constant in ↵
the sense
199 # of file in question, in another file there were different ↵
ones). This
200 # meant that ONLY recording in the shortest file with 1000 values ↵
was
201 # "usable" because it simply "fit" into this number; longer files ↵
only
202 # had the first 2047 data points "usable," which was useless to me.
203 # So, about a week after delivery (i.e., November 17, 2025), I ↵
complained
204 # about problem in two places (in China, because I bought the ↵
oscilloscope
205 # from Banggood):
206 #
207 # 1) directly from Owon (although I didn't buy it there, but ↵
what if...)
208 # at general address from their website:
209 # info@owon.com.cn
210 # however, as I expected, there was absolutely NO response...
```

211 # (i.e., until February 6, 2026).
212 #
213 # 2) at Banggood on the page where I bought the oscilloscope, after
214 # logging into my account with them, of course:
215 # <https://www.banggood.com/cs/...>
216 # and here the approach was quite accommodating. Basically, if
217 # I disregard some specifics of the people and settings of their
218 # website, thanks to their company, I received a working ↵
upgrade after
219 # more than a month (December 24, 2025). The most important ↵
220 # "peculiarity" is that customer MUST take some action within ↵
7 DAYS,
221 # and if they don't (as in my case, when I was waiting for their
222 # response even though they wrote that I should wait until they
223 # sent upgrade and that it would take a while), they CLOSE
224 # CONVERSATION... Admittedly, my oscilloscope had firmware ↵
version
225 # V1.0.1.0.17 when I bought it (with CSV writing not ↵
working), and in
226 # the first round they sent me an upgrade to version ↵
V1.0.1.7.1, which
227 # I was unable to install immediately due to incorrect naming ↵
of sent
228 # file (as soon as I received file, I tried to install it, ↵
oscilloscope
229 # recognized upgrade, but after launching it, oscilloscope ↵
reported
230 # a "Parse error"). And when I renamed it and successfully ↵
installed
231 # it, the CSV writing still did not work... Finally, Banggood ↵
sent
232 # me email address of a lady from OWON (I don't know if I can ↵
give
233 # it here), and on December 11, 2025, I sent her a ↵
description of
234 # problem again. She replied (on December 16, 2025), although ↵
not
235 # immediately after I sent email, saying that she would look ↵
into
236 # it and that I would have to wait. Well, then on December ↵
24, 2025,
237 # she sent me (as a gift??) an upgrade to version V1.0.1.7.2,
238 # and it WORKS, hooray... (although it still has some "bugs"; ↵
for
239 # example, as can be seen from the above excerpt from the ↵
beginning
240 # of CSV file, it still proudly displays firmware version ↵
V1.0.1.7.1
241 # and not V1.0.1.7.2, which was uploaded...)
242 #
243 #
244 #
245 # Oscilloscope type is determined ONLY by number of text lines at ↵
beginning
246 # of file, partly because HANTEK oscilloscope does not list any model ↵
such

```
247 # as OWON there, so I hope that any upgrade of oscilloscope will not ↵
    # affect
248 # this, but given that in the entire time I have had HANTEK oscilloscope
249 # (about 5 years), I have not seen any upgrades for it anywhere, and ↵
    # given
250 # above-described difficulties with recording to a csv file with OWON
251 # oscilloscope, I do not expect any upgrades to be available for ↵
    # either of
252 # them in the future.
253 #
254 # Program contains all necessary functions within itself, so it does ↵
    # not call
255 # any functions stored in other *.m files.
256 #
257 # For its operation, it needs to have signal package installed ↵
    # because it uses
258 # some of its functions. I have package, i.e., "signal-1.4.6.tar.gz" ↵
    # file,
259 # downloaded from:
260 #     https://gnu-octave.github.io/packages/signal/
261 #
262 # I started developing program in Octave version 10.2.0, i.e. GNU ↵
    # Octave,
263 # which is available at:
264 #     https://octave.org/
265 # it was tested and completed (so far) in February 2026 in Octave ↵
    # version
266 # 10.3.0. According to my testing, it is NOT compatible with MATLAB,
267 # there are quite a few differences...
268 #
269 #
270 #
271 global VerzeProg;
272 VerzeProg = '19 February 2026';   ###   Luděk Ruffer ↵
    lruffer@volny.cz
273
274
275
276
277
278
279
280
281 # If there is no semicolon at end of command, Octave will also display ↵
    # output
282 # in the command window!!!
283
284
285 # Note from NUF: In order for file to open via UIgetfile, you need to ↵
    # go to
286 #     main menu of Octave and select: EDIT/Preferences and then check
287 #     Use native file dialogs in the General tab. Otherwise, loading ↵
    # non-default
288 #     directories takes a long time.
289
290
```

```

291
292
293
294 pkg load signal;           #Without loading package Signal, neither Help nor
295                             #its contents are accessible in Octave, so in order
296                             #to access it, this program must be run at least ↵
                             once....

297
298 close all;                   #erases all figures
299 clear; clc;
300 output_precision(7);       #implicitly is set to 5 digits
301
302
303 # Sorry, but for now, other comments are still in Czech due to time ↵
   constraints.
304 # Only program inputs and outputs are translated.
305
306
307 global Cestafile = ''; # určení typu jde snad jen přiřazením hodnoty, ↵
   cocoti
308                             # a viz poznámka tady u globál UzJePsFile = ...
309
310
311
312 global Iniplk = struct(); #vytvoření prázdné (staré) struktúry jako ↵
   zobální
313 #a obsazení této staré struktúury
314 Iniplk.zpracdat = 'Method of data processing = ';
315 Iniplk.cestafile = 'Last processed path to data file = ';
316 Iniplk.amplharmveV = 'Amplitude of drawn harmonics in V (1 is V, 0 is ↵
   dB) = ';
317 Iniplk.pouzBT = 'Use Butterworth low pass filter (yes, other is no) = ';
318 Iniplk.fsmikBT = 'Butterworth low-pass filter cutoff frequency [Hz] = ';
319 Iniplk.radBT = 'Butterworth low-pass filter order (range 1 to 8, ↵
   otherwise 2nd order will be used) = ';
320 Iniplk.PocKrharm = 'Number of drawn harmonics = ';
321 Iniplk.PocPocharm = 'Number of calculated harmonics = ';
322 Iniplk.PoziceFig = 'Position of figure (window) when program ends = ';
323
324
325
326
327 global Jminifile AmplHarmV Velpistisk;
328
329
330
331
332 # variable ZpracDat can only have following values:
333 #
334 # 1 = "Process data as whole cycles from first zero crossing"
335 # 2 = "Process data as whole cycles from first maximum"
336 # 3 = "Process data as whole cycles from beginning of file"
337 # 4 = "Process data from entire file"
338 #
339 # predefined value (even when reading config file) is 1
340 global ZpracDat = 1;

```

```

341 global VsechnyVybery = false;
342
343
344
345
346 global Fsmik RadBTfiltru JeBTfiltr; #více proměnných v řádku MUSÍ ↵
    být BEZ
347 #oddělovačů (oddělených jen ↵
    mizerami)
348
349
350 global Monitory Hlmonitor;
351
352
353 global Cestafile Jmfile hgrf1 hgrf2 men Velpis;
354
355
356 global rel_pos_leg rel_pos_leg_BT;
357
358
359 global Osciloskop = "";
360 global OscHAN = "HANTEK DSO4102C";
361 global OscOWO = "OWON ADS800/900";
362
363
364 # následuje vytvoření řetězcového vektoru - musí být ve složených ↵
    zátvorkách
365 global plknastpod;
366 plknastpod = {"Process data as whole cycles from first zero crossing" ↵
    , ...
367             "Process data as whole cycles from first maximum" , ...
368             "Process data as whole cycles from beginning of file" , ...
369             "Process data from entire file"};
370
371 global PocKresHarm PocPocHarm PocHarm FposlHarm Fvypis FpocTHD;
372 PocKresHarm = 200; # preparát, dyby v ini filé nebylo
373 PocPocHarm = 2;
374
375 global JednHlWosyX JednHlWosyY JednWY;
376
377
378 global HlavPlk = 'Calculation of harmonic distortion (THD) using ↵
    discrete Fourier transform';
379 global hfig1;
380
381
382 global TlacimFiguru;
383 TlacimFiguru = false;
384
385
386 global ps_file;
387
388 # Gdyš je takto deklarováno a hned v deklaraci je přiřazena hodnota,
389 # tak se hodnota přiřadí JEN při spuštění oktávového prostředí a při
390 # dalším spuštění přes F5 se už tady nepřirazuje, ale kdyby byla hodnota
391 # tady přiřazována až v dalším řádku tak se hodnota přiřadí při každém

```

```

392 # spuštění (třeba přes F5) - tak jako jinde v programu při jeho běhu:
393 global UzJePsFile = false; # takto schválně, páč PS filé se maže při ukončování
394                                     # celé oktáávy, aby zbytečně nepsalo na HD
395
396
397
398 global PocVolTam PocVolVen;
399 PocVolTam = 0;
400 PocVolVen = 0;
401
402
403 global ZaklVelFig = struct(); #vytvoření prázdné (staré) struktúry jako zobální
404 #a obsazení staré struktúúry - násobky pro výpočet zákl. velikosti figúry
405 ZaklVelFig.X = 0.01;
406 ZaklVelFig.Y = 0.06;
407 ZaklVelFig.Sir = 0.97;
408 ZaklVelFig.Vys = 0.87;
409
410
411
412
413
414
415
416
417
418
419
420
421
422 #++++++ funkce volané z událostí
+++++
423
424
425
426 #.....
.....
427 function Oktavasezavira;
428     # Funkce, se volá při zavírání celé oktáávy.
429     # Je tu qůli výmazu filé Póówlšelllu, které se používá k indikaci, jeli
430     # aktiwní vokno maxima lízováno. Filé mazat AŽ PŘI UKONČOVÁNÍ OKTÁÁVY,
431     # protože kdyby se mazalo při ukončování figúúry, bylo by to jednak
432     # zbytečně často, a druhak a HLAVNĚ by se taky musela nějak vypucovat
433     # inicializační proměnná v podprogramu je_maximalizov_okno_s_focusem(),
434     # páč bez výpucu zůstane true a při dalším otevření figúry se filé nevytvoří
435     # a podprogram ho taky nenajde, takže podprogram projde, bude hlásit
436     # chybu a tím pádem se maximalizace okna bude určovat JEN z vodorovného
437     # rozměru, což nemusí vždy odpovídat...
438
439

```

```

440     global ps_file;
441
442
443     try
444         delete(ps_file);
445     end_try_catch
446
447 endfunction#..... ↵
.....
448
449
450
451
452
453
454 #..... ↵
.....
455 function Figurasezavira(src);
456     # Funkce, se zavolá při zavírání objektu s handlem src
457
458
459     #POZOR:
460     # když se tato funkce volala z definice příkazem set(... a měla jako
461     # předávané parametry ještě nějaké jiné než src (byly to Jminifile, ↵
462     # tak se sem předávaly jejich hodnoty které byly v okamžiku volání
463     # příkazu set(... a nebralo to ohled na jejich pozdější změny!!!!
464     # Tady to vadilo u xrozl, páč ta se může měnit, ostatní dvě asi neé.
465     # Takže jsem je udělal všechny preparát zobální a funguje to.
466     #pohof.
467
468
469     global Jminifile;
470
471
472
473     UlozInifile (Jminifile);
474
475
476
477     #POZOR:
478     # jeli volání této udalosti aktivováno, MUSÍ se tady objekt s ↵
479     # taky zavřít, když tu zavření nebude tak zůstane otevřený a v ↵
480     # potom nejde jinak zavřít než s celou Oktávou !!!!!
481     #pohof.
482
483     try
484         delete(src); # Zavření oného, musí být delete, close zde jaksi ↵
485         nefunguje
486     catch
487         # preparát, ale dyš to zblbne tak zmenšenou figuru stejně nesmaže...
488         close all;             #smaže fšecky figuřry
489     end

```

```

490
491 endfunction#..... ↵
.....
492
493
494
495
496 #..... ↵
.....
497 function volani_tlac_plkboxu(~, ~) #prázdný vstupy tam musí být, ↵
jinak řve botu
498
499     #Zabudovaná fuňkce gcbf ():
500     #Vrátí handle k figure obsahují objekt, jehož zpětné volání se ↵
právě provádí.
501
502
503     # Zavření figuury při stisknutí tlačítka
504     uiresume(gcbf); # Pokračuje v provádění po stisknutí tlačítka
505
506 endfunction#..... ↵
.....
507
508
509
510
511
512 #..... ↵
.....
513 function omezeni_rozmeru(hzdroj,~)
514
515     # Omezení rozměrů figúry s handle hzdroj tak aby byly v nějakých mezích
516     # kdy ještě nejsou plky přes sebe (mimum) anebo aby nebylo přes dva
517     # monitory, dyš sou (maximumo). Funkce se volá následovně:
518     #     set(hFig, 'sizechangedfcn', @(h,ev) omezeni_rozmeru(h));
519     # kde hFig je handle dotyčné figúry (tady je to hzdroj) a to způsobí,
520     # že je volána (většinou vícekrát) když probíhá změna velikosti oné ↵
figúry.
521     # L. Ruffer
522
523
524
525     global TlacimFiguru Hlmonitor;
526     global PocVolTam PocVolVen;
527
528
529     PocVolTam++;
530
531     # Funkce volá sama sebe (rekurze) třeba při provedení příkazu k ↵
nastavení
532     # velikosti figúry set(hzdroj,'position',pos) a možná i při změně ↵
jednotek
533     # a při takové rekurzi hrozí zacyklování, takže aby se funkce ↵
nezacyklovala
534     # používám tady (po pokusech s více všelijakými možnostmi) počítání
535     # vstupů do funkce a výstupů z funkce do dvou proměnných PocVolTam

```

```

536 # a PocVolVen deklarovaných jako global v hlavním programu (aby byly ↵
    jen
537 # jednou v paměti a viděly na ně všechny instance volané funkce)
538 # a vyskakuje se až při rovnosti vstupů a výstupů (u výstupu se ↵
    proměnná
539 # samozřejmě zvýší když už je jasné, že bude následovat výskok a ↵
    poté se
540 # rovnost vyhodnocuje). K provedení více čekajících instancí ↵
    podprogramu
541 # používám příkaz pause(jeho popis je kousek dáál), ten způsobí mj. ↵
    provedení
542 # nějakých čekajících volání tohoto podprogramu, ale nevím jak zjistit
543 # příkazem oktávy kolik volání čeká, a kolik už vyskočilo na pause a ↵
    pod.
544 # a když jsem tady začal vyskakovat pro zbytečná volání dříf, ↵
    nedařilo se
545 # mě dopočítat stejného počtu výskoků protože jsem měl příkaz pause ↵
    ještě
546 # před přidáním do proměnné počítající vstupy. Takže přidávání počtu ↵
    vstupů
547 # (proměnná PocVolTam) MUSÍ být před voláním příkazu pause() a ↵
    jiných na
548 # které to vyskakuje (mj. taky drawnow), páč na ně to vyskakuje což
549 # samozřejmě způsobí další zavolání tohoto podprogramu, ve kterém by se
550 # počet vstupů už nepřičetl a byl by v tom boordéél...
551 if (PocVolTam - PocVolVen) > 1
552     PocVolVen++;
553     return;
554 endif
555
556
557 # Protože výstup (tisknutí) figúúry do .jpg volá taky změnu rozměrů
558 # a provedení oného dál dělalo boordéél (nevytiskla se figura do ↵
    obrazového
559 # filé ve fuňkci Ulozdofile - tamější příkaz print), tak je tu ↵
    následující
560 # výskok přes TlacimFiguru když je tisk do .jpg
561 if TlacimFiguru
562     PocVolVen++;
563     return;
564 endif
565
566
567
568
569
570 # Dyš je maxima líízace tak véén bez ničeho a hned, aby maxima blízace
571 # zústala a funkce nechala figúúru maximalizovanou.
572 [Bota, JeMaximaBliizace] = je_maximalizov_okno_s_focusem();
573 if length(Bota)>0
574     # Ošetření případné obuvi při zjišťování maxima blízace ↵
    PowerodeŠelem.
575     #
576     # Tady aspoň zjištěním jeli šířka wokna zhruba stejná jako šířka ↵
    monitoru,
577     # oboje v logických pixelech a jen šířka je tu proto, že aspoň u W11

```

```

578 # jakštakš sedí (W11 by už neměly mít možnost přemísťovat hlavní
579 # panel), ale u W10 to bude fungovat jen když bude hl. panel doole,
580 # no aspoň něco...
581 posmaxim = round(get(hzdroj, 'position'));
582
583 # Zdá se, že stačí tole rance 7 pikslí:
584 JeMaxSir = abs(posmaxim(3) - Hlmonitor.sirkaLogic) <= 7;
585 JeMaximaBliizace = JeMaxSir;
586
587
588 # a výstup chybových plků aspoň do Commmand okna
589 printf('\n\nAn error occurred while determining whether figure was ↵
maximized:');
590 printf('\n\n%s', Bota);
591 printf('\n\nEnd of error text -----%s', ↵
newline);
592 endif
593 if JeMaximaBliizace
594     PocVolTam = 0;
595     PocVolVen = 0;
596     return;
597 endif
598
599
600
601
602
603
604
605 # Protože jsem nikde nenašel možnost (ani přes IÁÁ) říct oktávě, že ↵
má fčííl
606 # udělat všechny příkazy ve frontě (stack?) je to tu uděláno přes ↵
fuňkci
607 # pause(sevteřiny), protože v dokumentaci oktávy se k pause praví: ↵
"I když
608 # je program pozastaven, Octave stále zpracovává vykreslování obrázků
609 # a provádění grafických zpětných volání". Ovšem podle mých pokusů ↵
musí mít
610 # pauza zadaný nějaký čas aby se to provedlo, při kratším neudělá nic.
611 # A otázka je, jestli ten čas není závislý taky na rychlosti daného
612 # počítače, to by pak někde můj čas tady použitý nemusel stačit, ale ↵
zase
613 # to nechci mít zapauzované dlouho, aby to zbytečně neotravovalo...
614 pause(0.2);
615
616
617
618
619
620
621
622 # Načtení limitů uložených v figure v ÚserData.
623 # Je to v PIKSLÍCH , protože v programu jsou jednotky figúry
624 # nastaveny na PIKSLE.
625 # Tedy kromě přepnutí normalized - piksle při posledním volání, jehož
626 # důvod je popsán u něj.

```

```

627 s = get (hzdroj, 'UserData');
628
629
630 # Pozici pro nastavení zjistiti až tady (a znovu), aby zjištění bylo
631 # až po výskocích - může se lišit od zjištění na začátku podprogramu...
632 pos = round(get(hzdroj,'position'));
633 sir = pos(3); vys = pos(4);
634
635
636 # Omezení hodnot na povolený rozsah (round proto, že pixsely jsou celá čísla
637 # a nejsou možné desetiny, pak by se s celým číslem v pos() dále nerovnálo)
638 Omez_sir = round(min(max(sir, s.minSir), s.maxSir));
639 Omez_vys = round(min(max(vys, s.minVys), s.maxVys));
640
641
642 # kvůli níže popisovanému blbnutí oktáávy (nevím proč, možná blbnu já)
643 # je přidání a vynulování následujících oných tady, ocaď to už stejně
644 # vyskočí
645 PocVolVen++;
646 if PocVolTam == PocVolVen
647     PocVolTam = 0;
648     PocVolVen = 0;
649
650 # Někdy se stane (asi když je myš při tažení pro změnu rozměrů mimo
651 # miminální rozměry figuůry - nahoře a v levo) že se figuůra
652 # nenastaví i když příkaz set má pole rozměrů pos() jedlé a zůstane
653 # malá (někdy v obou rozměrech a někdy jen v jednom podle pozice myši)
654 # no a zdálo se mě (i když jsem nespal) že by mohla pomoct následující
655 # změna jednotek (asi musí být obě změny), ale nepomůže vždy...
656 set (hzdroj, 'units','normalized'); # normalizace
657 set (hzdroj, 'units','pixels');
658 endif
659
660
661
662 MimoVobrazofku = false;
663 # korekce začátku na výšku dyš by wrch byl někde mimo vobražofku nebo dyby
664 # bylo víc jak 3/4 wokna pod wobra žofkou
665 if (pos(2) + pos(4) > s.maxVys) || (pos(2) + pos(4)*0.75 < s.zacY)
666     MimoVobrazofku = true;
667 endif
668
669 #korekce začátku na šířku dyš by bylo víc jak 3/4 okna mimo vobražofku
670 if (pos(1) + pos(3)*0.25 > s.maxSir) || (pos(1) + pos(3)*0.75 < s.zacX)
671     MimoVobrazofku = true;
672 endif
673
674
675
676 if Omez_sir ~= sir || Omez_vys ~= vys || MimoVobrazofku
677

```

```

678     if Omez_sir ~= sir || Omez_vys ~= vys
679         pos(3) = Omez_sir;
680         pos(4) = Omez_vys;
681     endif
682
683
684
685     if pos(2) + pos(4)*0.75 < s.zacY
686         pos(2) = s.zacY;
687     elseif (pos(2) + pos(4) > s.maxVys)
688         pos(2) = s.maxVys - pos(4);
689     endif
690
691     if pos(1) + pos(3) * 0.75 < s.zacX
692         pos(1) = s.zacX;
693     elseif (pos(1) + pos(3) * 0.25 > s.maxSir)
694         pos(1) = s.maxSir - pos(3);
695     endif
696
697
698     set(hzdroj,'position',pos);
699     drawnow ("expose");
700
701
702     # protože změna rozměrů figuury někdy umístí legendy grafu doprčic ↵
703     # (někam
704     # mimo původní umístění nebo i mimo graf a pak nejsou viděti) je tu
705     # volání následující fuňkce aby se umístění legend předělalo
706     try
707         umisti_legendy(hzdroj)
708     end
709
710     endif
711
712     endfunction#..... ↵
713     .....
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731

```

```

732
733
734
735
736
737
738
739
740
741
742 #+++++
+++++
743 #+++++ funkce volané ocaď
+++++
744
745
746
747
748
749
750 #.....
.....
751 function [Bota, vystup] = je_maximalizov_okno_s_focusem();
752     # od IÁÁ Claude Haiku 4.5 ze dne 29.1.2026 - upr. ŇUF
753
754
755     % Funkce zjišťuje, zda je AKTIVNÍ okno (s fokusem) maximalizováno
756     % (optimalizovaná verze - soubor se vytvoří jen jednou)
757     %
758     % Syntaxe:
759     % [Bota, vystup] = je_maximalizov_okno_s_focusem()
760     %
761     % Výstup:
762     %   vystup - logická hodnota (true/false)
763     %           true = okno je maximalizováno
764     %           false = okno není maximalizováno
765     #   Bota - v případě průběhu bez chyby je prázdné (length(Bota) = 0)
766     #           v případě chyby je tam chybové hlášení (text)
767     %
768     % Poznámka:
769     %   Soubor se vytvoří jen při prvním volání a zůstane v tempdir
770     %   Vhodné pro časté volání (např. z resize eventů)
771     #
772     # Aby v temp adresáři soubor nesmrďeél tak na konci hl. programu
se má
773     # smazat, tady je to uděláno na začátku hl. programu definicí funkce,
774     # která se spustí při ukončování celé oktáavy, a to příkazem:
775     #   atexit('jméno oné funkce')
776
777     global ps_file UzJePsFile;
778
779
780
781     Bota = '';
782
783     # Inicializace jen při prvním volání po spuštění Oktááve

```

```

784 # (je zajištěno globálně deklarací na začátku v hl. programu)
785 if UzJePsFile == false
786     % Vytvoření dočasného souboru (jen jednou)
787     temp_dir = tempdir();
788     ps_file = fullfile(temp_dir,
789         'Octave-je_maximalizov_okno_s_focusem.ps1');
790
791     % PowerShell skript
792     ps_code = sprintf([...
793         'Add-Type @"\\n' ...
794         'using System;\\n' ...
795         'using System.Runtime.InteropServices;\\n' ...
796         'public class WindowState {\\n' ...
797         '    [DllImport("user32.dll", SetLastError = true)]\\n' ...
798         '    public static extern IntPtr GetForegroundWindow();\\n'
799         '    [DllImport("user32.dll", SetLastError = true)]\\n' ...
800         '    public static extern bool IsZoomed(IntPtr hWnd);\\n' ...
801         '\\n' ...
802         '@\\n' ...
803         '$hWnd = [WindowState]::GetForegroundWindow()\\n' ...
804         'if ($hWnd -ne 0) {\\n' ...
805         '    if ([WindowState]::IsZoomed($hWnd)) {\\n' ...
806         '        Write-Host "1"\\n' ...
807         '    } else {\\n' ...
808         '        Write-Host "0"\\n' ...
809         '    }\\n' ...
810         '} else {\\n' ...
811         '    Write-Host "0"\\n' ...
812         '\\n'}]);
813
814     % Zápis skriptu do souboru (jen jednou)
815     try
816         fid = fopen(ps_file, 'w');
817         fprintf(fid, '%s', ps_code);
818         fclose(fid);
819         UzJePsFile = true;
820     catch exception
821         Bota = strcat('When creating file, there is a function
822         error: je_maximalizov_okno_s_focusem %s',
823         exception.message);
824         UzJePsFile = false;
825         vystup = false;
826         return;
827     end
828 end
829
830 try
831     % Spuštění PowerShell skriptu (bez zápisu - jen čtení a spuštění)
832     cmd = sprintf('powershell.exe -NoProfile -ExecutionPolicy
833     Bypass -File "%s"', ps_file);
834     [status, output] = system(cmd);
835
836     % Zpracování výstupu
837     if length(strtrim(output)) > 1
838         # je obuf, jedlá odpověď je 0 nebo 1, při chybě je tam

```

```

835     chybový plk
836     Bota = strtrim(output);
837     vystup = false;
838     else
839     output = strtrim(output);
840     vystup = strcmp(output, '1');
841     endif
842     catch exception
843     % V případě chyby vrátit false
844     Bota = strcat('Function error: je_maximalizov_okno_s_focusem ↵
845     %s', exception.message);
846     vystup = false;
847     end
848     endfunction#..... ↵
849     .....
850
851
852
853
854
855
856
857
858
859
860     #..... ↵
861     .....
862     function [Bota, Monitory] = Zjisti_monitory()
863     # načtení hodnot o monitourech jinak, dyš to oktááva neumí, ale jen ↵
864     we Voknech
865     # deklarace prázdného oného
866     Monitory = struct('primarni', {}, 'sirkaSkut', {}, 'vyskaSkut', {}, ...
867     'zvetseni_proc', {}, 'sirkaLogic', {}, ↵
868     'vyskaLogic', {},...
869     'rucni_vstup', {});
870
871     Bota='';
872     [Bota, monLogicke] = Zjisti_monitory_PS();
873     if length(Bota)>0, return; end #je obuf
874
875     [Bota, DPI_ef] = Zjisti_DPI_ef();
876     if length(Bota)>0, return; end #je obuf
877
878
879     for i=1 : numel(monLogicke)
880     Monitory(i).sirkaLogic = monLogicke(i).widthLogic;
881     Monitory(i).vyskaLogic = monLogicke(i).heightLogic;
882     Monitory(i).zvetseni_proc = (DPI_ef(i,1)/96)*100;
883     Monitory(i).primarni = monLogicke(i).primary;
884

```

```

885     # oboje zaokrouhlit podle pravidel zaokrouhlování, round kladné ↵
      hodnoty
886     # okrouhlí takto: 2.4 = 2; 2.6 = 3, záporné tu snad nebudou
887     Monitor(i).sirkaSkut = round(Monitor(i).sirkaLogic * ...
888                               Monitor(i).zvetseni_proc/100);
889     #výšku preparát brát z Y
890     Monitor(i).vyskaSkut = round(Monitor(i).vyskaLogic * ↵
      (DPI_ef(i,2)/96));
891
892     Monitor(i).rucni_vstup = 0;
893     endfor
894
895 endfunction#..... ↵
      .....
896
897
898
899
900
901 #..... ↵
      .....
902 function [Bota, monitors] = Zjist_i_monitory_PS()
903
904     # Funkce pro přečtení rozlišení displejů počítače vytvořená IÁÁ, éé ↵
      pardon AI
905     # GPT-5 mimi a voláním příkazu Powershellu 11.1.2026.
906
907     # Vrátí pole struktur s poli: deviceName, widthLogic, heightLogic, ↵
      x, y, primary
908
909     # Vrací rozlišení v tzv. "logických" pikselech, a pokud je ve woknech
910     # na oném (nebo oných) displejích nastaveno nějaké zvětšení, tedy
911     # měřítko větší jak 100% bude rozlišení v logických pikselech ↵
      přiměřeně menší.
912     # Wokenice používají jako 100% rozlišení 96 DPI, tedy mělo by být:
913     # 96 DPI = 100%, 120 DPI = 125%, 144 DPI = 150% a to tady čte funkce
914     # Zjist_i_DPI_ef().
915     # Já mám na hl. monitoru ve woknech nastaveno zvětšení 150%, takže ↵
      pro tento
916     # monitor tato funkce vrátí X rozlišení 3840/1,5 = 2560 logických ↵
      pikslí
917     # (což vrací) a pro druhý monitor mám zvětšení 125%, takže 2560/1,25 ↵
      = 2048
918     # log. pikslí atd.
919
920
921     # Výpis z podprogramu do Command wokna woktávy se tam, odkud se volá
922     # (NEE TADY) dá udělat třeba takto (tady MUSÍ být zapoznámkováno):
923     #     [Bota, mon] = Zjist_i_monitory_PS();
924     #     for i = 1:length(mon)
925     #         fprintf('Monitor %d (%s): %dx%d at (%d,%d) primary=%d\n', ↵
      i, ...
926     #             mon(i).deviceName, mon(i).widthLogic, ↵
      mon(i).heightLogic, ...
927     #             mon(i).x, mon(i).y, mon(i).primary);
928     #     end

```

```

929
930
931 Bota = '';
932
933 monitors = struct('deviceName', {}, 'widthLogic', {}, 'heightLogic', ↵
    {}, ...
934                 'x', {}, 'y', {}, 'primary', {});
935
936 % PowerShell příkaz: použijte System.Windows.Forms.Screen
937 ps_cmd = [
938     'powershell -NoProfile -Command "Add-Type -AssemblyName ↵
    System.Windows.Forms; ', ...
939     '$s = [System.Windows.Forms.Screen]::AllScreens | ForEach-Object { ↵
    ', ...
940     ' [PSCustomObject]@{ deviceName=$_DeviceName; ↵
    widthLogic=$_Bounds.Width; heightLogic=$_Bounds.Height; ↵
    x=$_Bounds.X; y=$_Bounds.Y; primary=$_Primary } ', ...
941     '}; $s | ConvertTo-Json -Compress"'
942 ];
943 [st, out] = system(ps_cmd);
944 if st ~= 0
945     Bota = sprintf('Subprogram Zjistí_monitory_PS returns:\n ↵
    PowerShell error: %s', out);
946 end
947 if length(Bota)>0, return; end #je obuf
948
949
950
951 % Parse JSON (handle single object vs array)
952 try
953     j = jsondecode(out);
954 catch
955     % Pokud jsondecode selže, vypiš surový výstup pro diagnostiku
956     Bota = sprintf('Subprogram Zjistí_monitory_PS returns:\n Unable to ↵
    decode JSON from PowerShell: %s', out);
957 end
958 if length(Bota)>0, return; end #je obuf
959
960
961
962
963 % Normalize to array
964 if isstruct(j) && ~isscalar(j)
965     arr = j;
966 elseif isstruct(j) && isscalar(j)
967     arr = j; % ConvertTo-Json returns single object for 1 screen; keep ↵
    as scalar
968 else
969     Bota = sprintf('Subprogram Zjistí_monitory_PS returns:\n ↵
    Unexpected JSON format: %s', out);
970 end
971 if length(Bota)>0, return; end #je obuf
972
973
974 % Build monitors structure array
975 if isscalar(arr)

```

```

976     n = 1;
977 else
978     n = numel(arr);
979 end
980 monitors(1:n) = struct('deviceName', '', 'widthLogic', 0,
    'heightLogic', ...
981                        0, 'x', 0, 'y', 0, 'primary', false);
982 for k = 1:n
983     if isscalar(arr)
984         item = arr;
985     else
986         item = arr(k);
987     end
988     monitors(k).deviceName = item.deviceName;
989     monitors(k).widthLogic = double(item.widthLogic);
990     monitors(k).heightLogic = double(item.heightLogic);
991     monitors(k).x = double(item.x);
992     monitors(k).y = double(item.y);
993     monitors(k).primary = logical(item.primary);
994 end
995
996 endfunction# .....
    .....
997
998
999
1000
1001
1002
1003 # .....
    .....
1004 function [Bota, dpi] = Zjistí_DPI_ef()
1005
1006     # Funkce pro přečtení efektivního DPI odpovídajícího Windows škálování
1007     # pro všechny displeje (monitory) počítače vytvořená IÁÁ, éé pardon AI
1008     # GPT-5 mimi a voláním příkazů Powershellu 11.1.2026.
1009
1010     # Stručný popis podprogramu od IAA:
1011     #   Podprogram před čtením DPI nastaví proces na per-monitor DPI aware
1012     #   (pokud API dostupné) a pak volá GetDpiForMonitor pro každý monitor
1013     #   - vrací efektivní DPI odpovídající Windows škálování,
1014     #   např. 144 pro 150%, 120 pro 125%.
1015     # Podprogram volá SetProcessDpiAwareness(2) (per-monitor) pokud je
    funkce
1016     # dostupná. Pokud není (velmi staré Windows), volání je ignorováno.
1017
1018     # Pro každý monitor vystoupí dvě hodnoty:
1019     #   dpi(i,1)   je DPI ve směru X (horizontální)
1020     #   dpi(i,2)   je DPI ve směru Y (vertikální)
1021     # Hodnoty budou většinou stejné, rozdíl může být např. když by
    monitor měl
1022     # různý počet bodů na palec horizontálně a vertikálně.
1023
1024     # Podprogram tedy např. pro nastavené wokenní zvětšení (škálování) 150%
1025     # vrátí hodnotu 144. To vychází z toho, že ve vokenicích se jako
    výchozí

```

```

1026 # DPI používá hodnota 96 DPI, tedy 96 DPI = 100%. Vrácené hodnoty se ↵
    tedy
1027 # na jedlá procenta musí přepočítat, např. pro vrácených 144 v ↵
    případě 150%:
1028 # (144 / 96) * 100 = 150, pro vrácených 120: (120 / 96) * 100 = 125 ↵
    atp.
1029
1030
1031 Bota = '';
1032
1033 if ispc == 0
1034     Bota = sprintf('Subprogram Zjistí_DPI_ef returns:\n This ↵
        subprogram only works on Windows.');
```

```

1035 end
1036 if length(Bota)>0, return; end     #je obuf
1037
1038
1039
1040 ps_script = [
1041     'Add-Type -TypeDefinition @"' char(10) ...
1042     'using System; using System.Runtime.InteropServices; using ↵
        System.Collections.Generic;' char(10) ...
1043     'public static class DPIHelper {' char(10) ...
1044     '     private const int MDT_EFFECTIVE_DPI = 0;' char(10) ...
1045     '     [DllImport("shcore.dll")] public static extern int ↵
        GetDpiForMonitor(IntPtr hmonitor, int dpiType, out uint dpiX, out ↵
        uint dpiY);' char(10) ...
1046     '     [DllImport("shcore.dll")] public static extern int ↵
        SetProcessDpiAwareness(int value);' char(10) ... %
        0=Unaware,1=System,2=PerMonitor
1047     '     [DllImport("user32.dll")] public static extern bool ↵
        EnumDisplayMonitors(IntPtr hdc, IntPtr lprcClip, ↵
        MonitorEnumDelegate lpfnEnum, IntPtr dwData);' char(10) ...
1048     '     public delegate bool MonitorEnumDelegate(IntPtr hMonitor, ↵
        IntPtr hdcMonitor, ref RECT lprcMonitor, IntPtr dwData);' char(10) ↵
        ...
1049     '     [StructLayout(LayoutKind.Sequential)] public struct RECT { ↵
        public int left, top, right, bottom; }' char(10) ...
1050     '     public static List<string> GetAllDpi()' char(10) ...
1051     '     try { SetProcessDpiAwareness(2); } catch { /* ignore if not ↵
        available */ }' char(10) ...
1052     '     var list = new List<string>();' char(10) ...
1053     '     MonitorEnumDelegate d = (IntPtr hMon, IntPtr hdc, ref RECT r, ↵
        IntPtr dd) => {' char(10) ...
1054     '         uint x=0,y=0; int res = GetDpiForMonitor(hMon, ↵
        MDT_EFFECTIVE_DPI, out x, out y);' char(10) ...
1055     '         if (res == 0) list.Add(string.Format("({0},{1})", x, y));' ↵
        char(10) ...
1056     '         return true; };' char(10) ...
1057     '     EnumDisplayMonitors(IntPtr.Zero, IntPtr.Zero, d, ↵
        IntPtr.Zero);' char(10) ...
1058     '     return list; }' char(10) ...
1059     '}' char(10) ...
1060     '"@" char(10) ...
1061     '$list = [DPIHelper]::GetAllDpi();' char(10) ...
1062     'if ($list -eq $null -or $list.Count -eq 0) { Write-Output "" ; ↵
```

```

1063     exit }' char(10) ...
1064     'Write-Output ($list -join ";" )" ]';
1065 tf = [tempname '.ps1'];
1066 fid = fopen(tf,'w');
1067 if fid == -1
1068     Bota = sprintf('Subprogram Zjisti_DPI_ef returns:\n Unable to
1069     create temporary PS program. ');
1070 endif
1071 if length(Bota)>0, return; end #je obuf
1072 fwrite(fid, ps_script);
1073 fclose(fid);
1074
1075 cmd = ['powershell -NoProfile -ExecutionPolicy Bypass -File "' tf '"'];
1076 [status, out] = system(cmd);
1077
1078 delete(tf);
1079
1080 if status ~= 0
1081     Bota = sprintf('Subprogram Zjisti_DPI_ef returns:\n PowerShell
1082     call failed: %s', out);
1083 endif
1084 if length(Bota)>0, return; end #je obuf
1085
1086 out = strtrim(out);
1087 if isempty(out)
1088     dpi = zeros(0,2);
1089     return;
1090 end
1091
1092 parts = strsplit(out, ';');
1093 n = numel(parts);
1094 dpi = zeros(n,2);
1095 for i = 1:n
1096     s = regexp(parts{i}, '[\(\)\s]', ''); % odstranit zavoroky/mezery
1097     v = strsplit(s, ',');
1098     dpi(i,1) = str2double(v{1});
1099     dpi(i,2) = str2double(v{2});
1100 end
1101
1102 endfunction#.....
1103 .....
1104
1105
1106
1107
1108
1109
1110
1111
1112 #.....
1113 .....
1114
1115 function VobsadDilkyWos (handl, Mimumum, Maximum, Wosa)

```

```

1114
1115 # obsluha kreslení wos
1116 # ve Wosa je, pro kterou wosu se budou dílky a rozsah předělávat:
1117 #     když první znak bude "x" nebo "X" (s výjimkou předchozích
mizer, ty se
1118 #     ignorují) bude to wosa X (vodorovná),
1119 #     musí tedy Mimum, Maximum být hodnoty vykreslované na wosu X
1120 #     když první znak bude jiný bude to wosa Y (svislá)
1121 #     musí tedy Mimum, Maximum být hodnoty vykreslované na wosu Y
1122 #     (třetí wosu Z to zatím neobhospodařuje)
1123 #
1124 # handl je handle na objekt wos (gca) pro který se to dělá
1125
1126
1127
1128
1129 Wosa = upper(strtrim(Wosa)); # může se upravovat vstupní prom.
Wosa, protože
1130
# tady neleze veeen
1131
1132 KratkaWosa = false;
1133 if Cmuchret(Wosa(1:1), "Y")==1
1134     KratkaWosa = true;
1135 endif
1136
1137 [Hod] = IntervOs (Mimum , Maximum, KratkaWosa);
1138
1139
1140 # Pořadí následujících příkazů MUSÍ být takovéto:
1141 #     1. nastavení rozsahu,
1142 #     2. mood maňuál
1143 #     3. nastavení velkých dílků
1144 #     4. nastavení malých dílků
1145 # jinak si může některé hodnoty přeonačit (třeba dyš se nastaví malé
dílky
1146 # dříf jak velké...
1147
1148
1149 if Cmuchret(Wosa, "X")==1; #takto nemusí být ve Wosa jen
jeden znak
1150
# wosa X (vodorovná)
1151
xlim(handl, [Hod(6), Hod(7)]); #nastavení vypočteného rozsahu
wosy handl
1152
set(handl, 'xtickmode', 'manual');
1153
set(handl, "xtick", (Hod(1):Hod(4):Hod(2)));
1154
set(handl, "xminortickvalues", (Hod(6):Hod(5):Hod(7)));
1155
1156
else
1157
# wosa Y (svislá)
1158
1159
# Případné trochu zvětšení rozsahu wosy dyš se kryje s max nebo
1160
min dat
1161
# zatím jen u Y wosy, aby byl líp vyděti nasnímaný průběh.
1162
# A u takto nastavovaných wos MUSÍ být tady a takto, aby se použilo
1163
# a už nezměnilo, poáč dyš jsem je měl za čtveřicí příkazů pro

```

```

nastavení
1164 # tak tam provedený ylim() způsobuje nové přepočítání dílků podle
Octave
1165 # a to tady nechcu...
1166 MaxGr = Hod(7);
1167 MinGr = Hod(6);
1168 if MaxGr <= Maximum*1.025
1169     MaxGr = MaxGr*1.025;
1170     if MaxGr > Hod(2) + Hod(4)
1171         #ještě enev. úprava maxima podle vel. dílků wos
1172         Hod(2) = Hod(2) + Hod(4);
1173     endif
1174 endif
1175 if MinGr >= Mimumum*1.025
1176     MinGr = MinGr*1.025;
1177     if MinGr < Hod(1) - Hod(4)
1178         #ještě enev. úprava mimina podle vel. dílků wos
1179         Hod(1) = Hod(1) - Hod(4);
1180     endif
1181 endif
1182
1183
1184 ylim(handl, [MinGr, MaxGr]); #nastavení vypočteného rozsahu wosy
handl
1185 set(handl, 'ytickmode', 'manual');
1186 set(handl, "ytick", (Hod(1):Hod(4):Hod(2)));
1187 set(handl, "yminortickvalues", (Hod(6):Hod(5):Hod(7)));
1188
1189 endif
1190 drawnow;
1191
1192
1193
1194 endfunction#.....
.....
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204 #.....
.....
1205 function [Hod] = IntervOs (Min, Max, KratkaWosa)
1206 # Převzatý a upravený podprogram z programu Zpracování měření z VB6.
1207 #
1208 # Vypočte z minima a maxima hodnoty pro wosy, Max a Min by měly být
skutečné
1209 # maximální a miminální hodnoty z kreslených dat dotyčné wosy.
1210 #
1211 # Když bude KratkaWosa = true, tak se malé dílky udělají s větším
odstupem,
1212 # (4 nebo 5 na velký dílek místo 10 na velký dílek) aby na kratší wose

```

```

1213 # nebylo dílkování moc husté...
1214 #
1215 # Ve výstupu jsou výsledky uloženy takto:
1216 #     Hod(1)   minimum  ("kulaté" podle velkých dílků)
1217 #     Hod(2)   maximum  ("kulaté" podle velkých dílků)
1218 #     Hod(3)   průsečík (podle velkých dílků)
1219 #     Hod(4)   velký dílek
1220 #     Hod(5)   malý dílek
1221 #     Hod(6)   minimum  ("kulaté" podle malých dílků)
1222 #     Hod(7)   maximum  ("kulaté" podle malých dílků)
1223 #     Hod(8)   průsečík (podle malých dílků)
1224 #
1225 # Tady spočítá všechny možnosti najednou a vybrat z nich se musí jinde
1226
1227
1228
1229
1230 if Max == Min
1231     Max = Max + 0.9;      # aby osy byly rozdílné o 1
1232 endif
1233
1234
1235 P = 10 ^ (floor(log10(abs(Max - Min))));
1236
1237 Hod(1) = floor(Min / P) * P;
1238 if Min > 0
1239     if Hod(1) > Min
1240         Hod(1) = (floor(Min / P) - 1) * P;
1241     endif
1242 endif
1243
1244 Hod(2) = (floor(Max / P)) * P;
1245 if Max > 0
1246     if Hod(2) < Max
1247         # Toto je tady proto, že funkce floor() (ve VB Int()) funguje tak,
1248         # že např. z -2.7 udělá -3, ale z +2.7 udělá 2, takže by se to tam
1249         # pokud bylo Max trochu větší než Hod(2) nevlezlo, a proto tam ↵
1250         # bylo +1 furt, ale v tom případě, pokud bylo Max = Hod(2) tak to
1251         # zbytečně přidávalo velký dílek (u mimina výše je to kladných ↵
1252         # obdobně, jen by se mělo ubírat).
1253         Hod(2) = (floor(Max / P) + 1) * P;
1254     endif
1255 endif
1256
1257
1258
1259
1260 P = log10(abs(Hod(2) - Hod(1)));
1261
1262 if (P - floor(P)) < 0.30102999567
1263     #10^(0.30102999567)=2.000000000027717
1264     Hod(4) = 0.2;
1265     if KratkaWosa
1266         Hod(5) = 0.05;

```

```

1267     else
1268         #toto tady dát jen pro delší wosu páč na krátké wose je to moc
            jemné...
1269         Hod(5) = 0.02;
1270     endif
1271 elseif (P - floor(P)) > 0.69897000434
1272     #10^(0.69897000434)=5.000000000045835
1273     Hod(4) = 1;
1274     if KratkaWosa
1275         Hod(5) = 0.2;
1276     else
1277         #toto tady dát jen pro delší wosu páč na krátké wose je to moc
            jemné...
1278         Hod(5) = 0.1;
1279     endif
1280 else
1281     Hod(4) = 0.5;
1282     Hod(5) = 0.1;
1283 endif
1284
1285 P = floor(P);
1286 Hod(4) = Hod(4) * 10 ^ P;
1287 Hod(5) = Hod(5) * 10 ^ P;
1288
1289
1290
1291
1292
1293 # & je and    | je or    ~ je not    ~= je není rovno    cocoti
1294 if (Min <= 0) && (Max >= 0) #chce to zkratku &&
1295     # úprava mimina a maxima tak, aby dyš je wosa přes nulu tam nula
        byla dycky,
1296     # v tomto případě se to musí podle malých dílků upravit
1297     # tam, kde se v Oktááve nastavuje ylim, páč je potřeba nastavit
        limity
1298     # gřáfu, a to bez změny zde spočítaného jde jen tam a v tam
        uvedeném pořadí
1299     Hod(1) = 0;
1300     while Min <= Hod(1) - Hod(4)
1301         Hod(1) = Hod(1) - Hod(4);
1302     endwhile
1303
1304     Hod(2) = 0;
1305     while Max >= Hod(2) + Hod(4)
1306         Hod(2) = Hod(2) + Hod(4);
1307     endwhile
1308
1309     # miminum, maximum a prusečik pro malé dílky
1310     Hod(6) = Hod(1);
1311     while Min < Hod(6)                # uprava minima na malý dílek
1312         Hod(6) = Hod(6) - Hod(5);
1313     endwhile
1314     Hod(7) = Hod(2);
1315     while Max > Hod(7)                # uprava maxima na malý dílek
1316         Hod(7) = Hod(7) + Hod(5);
1317     endwhile

```

```

1318
1319     else
1320
1321         # miminum, maximum a prusečik pro velké dílky
1322         while Min > Hod(1) + Hod(4) # uprava minima na interval popisu
1323             Hod(1) = Hod(1) + Hod(4);
1324         endwhile
1325
1326         while Max < Hod(2) - Hod(4) # uprava maxima na interval popisu
1327             Hod(2) = Hod(2) - Hod(4);
1328         endwhile
1329
1330         # miminum, maximum a prusečik pro malé dílky
1331         Hod(6) = Hod(1);
1332         while Min > Hod(6) + Hod(5) # uprava minima na malý dílek
1333             Hod(6) = Hod(6) + Hod(5);
1334         endwhile
1335         Hod(7) = Hod(2);
1336         while Max < Hod(7) - Hod(5) # uprava maxima na malý dílek
1337             Hod(7) = Hod(7) - Hod(5);
1338         endwhile
1339
1340     endif
1341     Hod(3) = Hod(1);
1342     Hod(8) = Hod(6);
1343
1344
1345
1346     endfunction#..... ↵
1347     .....
1348
1349
1350
1351
1352
1353     #..... ↵
1354     .....
1355     function indeks = Cmuchret(vcem, co) ↵
1356         # fuňkce by měla vrátiti první výskyt řetězce co v retezci vcem,
1357         nebo nulu
1358         # gdyš tam nejni
1359         indeksy = strfind(vcem, co);
1360         if isempty(indeksy)
1361             indeks = 0;
1362         else
1363             indeks = indeksy(1);
1364         end
1365     endfunction;#..... ↵
1366     .....
1367
1368
1369

```

```

1370
1371 #..... ↵
.....
1372 function aktivuj_omez_rozm(hFig)
1373     # Připojí ResizeFcn k figuře, aby volala funkci omezeni_rozmeru
1374     # při každé změně velikosti.
1375
1376     # Pokud není hFig zadána, použije se(gcf).
1377     if nargin < 1 || isempty(hFig)
1378         hFig = gcf;
1379     end
1380
1381     set(hFig, 'sizechangedfcn', @(h,ev) omezeni_rozmeru(hFig));
1382
1383 endfunction;#..... ↵
.....
1384
1385
1386
1387
1388 #..... ↵
.....
1389 function nastavPozLegendyRelatkOse(hOsy, hleg, rel_pos);
1390
1391     # Funkce pro nastavení pozice legendy tak, jak chci jááá, tedy aby byla
1392     # napravo (east) vevnitř ale pod popisem wosy. Původně to bylo podle
1393     # IÁ GPT-5 mimi relativně k ose, což měnilo všechny parametry pozice,
1394     # ovšem to blbě umísťovalo pozici vodorovně páč se mě asi nedařilo ↵
    najít
1395     # správnou relativní pozici zde v rel_pos (obsazuje se natwrdo po ↵
    spuštění
1396     # programu), tak jsem vodorovnou pozici legendy nechal na automatizaci
1397     # oktávy, ta to dělá celkem dobře. Ofšem aby fungovala, JE NUTNÉ v ↵
    každém
1398     # průchodu před zjištěním pozice legendy znovu určit její pozici AŽ ↵
    po použití
1399     # příkazu legend(hleg,'location','southeast') (zde používám pozici ↵
    vlevo dole
1400     # v gřafu - podle terminologie map southeast) protože jakékoliv ↵
    nastavení
1401     # pozice legendy přes 'position' logicky způsobí změnu 'location' na ↵
    'none'
1402     # a v tom případě se už pozice automaticky nepřepočítává.
1403
1404
1405     # kde je:
1406     # hOsy:     handle oné osy
1407     # hleg:     handle oné legendy
1408     # rel_pos:  [x y sirka výška] v normalizovaných osových jednotkách
1409
1410
1411     # Získat pozice v normalizovaných jednotkách vůči figuře pro osu a ↵
    pro legendu
1412     puv_jedn_osy = get(hOsy, "Units");
1413     puv_jedn_legendy = get(hleg, "Units");
1414     set(hOsy, "Units", "normalized");

```

```

1415  set(hleg, "Units", "normalized");
1416
1417  # zjištění parametrů legendy pro lokaci 'southwest' legendu
NESCHOVÁVAT,
1418  # 'hide' páč pak blbne a neobnoví pozici jedle (asi dyš je schovaná
to pozici
1419  # nepřepočítává čico...)
1420  legend(hleg, 'location', 'southeast');
1421
1422  PosOsy = get(hOsy, "Position");
1423  PosLeg = get(hleg, "Position");
1424
1425
1426  # Původní přepočet podle IÁÁÁ, necháno preparát na památku
1427  # Spočítat legend pozici v normalizovaných (vzhledem k figúře)
souřadnicích
1428  # pos_leg_prepoc = [ PosOsy(1) + rel_pos(1)*PosOsy(3), ...
1429  #                   PosOsy(2) + rel_pos(2)*PosOsy(4), ...
1430  #                   rel_pos(3)*PosOsy(3), ...
1431  #                   rel_pos(4)*PosOsy(4) ];
1432
1433
1434  # moje úprava, z přepočtu IÁÁÁ používá jen korekci svislého
umístění, aby
1435  # legenda byla pod popisem wosy, na ostatní používá automat oktávy
1436  pos_leg_prepoc = [ PosLeg(1), ...
1437  #                 PosOsy(2) + rel_pos(2)*PosOsy(4), ...
1438  #                 PosLeg(3), ...
1439  #                 PosLeg(4) ];
1440
1441
1442  set(hleg, "Position", pos_leg_prepoc);
1443
1444
1445  # vrátit původní jednotky a vidi telnost
1446  set(hOsy, "Units", puv_jedn_osy);
1447  set(hleg, "Units", puv_jedn_legendy);
1448
1449
1450  endfunction;#.....
.....
1451
1452
1453
1454
1455  #.....
.....
1456  function umisti_legendy(fig);
1457  # znovu umístí obě legendy tady použité podle proměnných uložených
ve figúře s handlem fig
1458
1459
1460  # legenda hl. gřafu
1461  hleg = getappdata(fig, "hLegHL");
1462  ax = getappdata(fig, "hOsyXHL");
1463  rel_pos = getappdata(fig, "RelPosLegHL");

```

```

1464     if isempty(hleg) || ~ishandle(hleg) || isempty(ax) || ~ishandle(ax)
1465         return;
1466     end
1467
1468     % a umístiti
1469     nastavPozLegendyRelatkOse(ax, hleg, rel_pos);
1470
1471
1472
1473     # legenda BT gřafu
1474     JeKresBT = getappdata(fig, "KresSmikHarm");
1475
1476     if JeKresBT
1477         hleg = getappdata(fig, "hLegBT");
1478         ax = getappdata(fig, "hOslyXBT");
1479         rel_pos = getappdata(fig, "RelPosLegBT");
1480         if isempty(hleg) || ~ishandle(hleg) || isempty(ax) || ~ishandle(ax)
1481             return;
1482         end
1483
1484         % a umístiti
1485         nastavPozLegendyRelatkOse(ax, hleg, rel_pos);
1486     endif
1487
1488     drawnow;
1489
1490 endfunction;#..... ↵
1491 .....
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501 #..... ↵
1502 .....
1503 function [y] = rms(x);
1504     #výpočet efektivní hovnoty, RMS - root mean square
1505
1506     n = length(x);
1507     y = sqrt(sum(x.^2)/n);
1508
1509 endfunction;#..... ↵
1510 .....
1511
1512
1513
1514
1515 #..... ↵
1516 .....

```



```

maximum',newline );
1564 fprintf(fid, plk);
1565 plk = strcat('3 = Process data as whole cycles from beginning of
file',newline );
1566 fprintf(fid, plk);
1567 plk = strcat('4 = Process data from entire file',newline );
1568 fprintf(fid, plk);
1569 plk = strcat('When another value or no value is loaded, following
is set = 1',newline,newline );
1570 fprintf(fid, plk);
1571
1572
1573 plk = strcat(Iniplk.zpracdat, num2str(ZpracDat,
"%d"),newline,newline);
1574 fprintf(fid, plk);
1575
1576
1577 plk = strcat(newline, Iniplk.PocKrharm, num2str(PocKresHarm,
"%d"),newline );
1578 fprintf(fid, plk);
1579
1580 plk = strcat(newline, Iniplk.PocPocHarm, num2str(PocPocHarm,
"%d"),newline );
1581 fprintf(fid, plk);
1582
1583 plk = strcat(newline, Iniplk.amplharmveV, num2str(AmplHarmV,
"%d"),newline );
1584 fprintf(fid, plk);
1585
1586
1587
1588 plk = strcat(newline, Iniplk.cestafile);
1589 fprintf(fid, plk);
1590 #cesta se zpět. lomítka MUSÍ být uložena jako řetěz pomocí
následujícího
1591 #formátování, jinak z ní oktáva ty zpětná lomítka vyháže!!!!!!
1592 fprintf(fid, '%s\n', Cestafile);
1593
1594
1595 plk = strcat(newline, Iniplk.PoziceFig);
1596 fprintf(fid, plk);
1597 fprintf(fid, '%d %d %d %d\n', round(get(hfig1, 'position')));
1598
1599
1600 if JeBTfiltr
1601     plk1 = 'yes';
1602 else
1603     plk1 = 'no';
1604 endif
1605 plk = strcat(newline, Iniplk.pouzBT, plk1, newline);
1606 fprintf(fid, plk);
1607
1608 plk = strcat(Iniplk.fsmikBT, num2str(Fsmik, "%d"),newline );
1609 fprintf(fid, plk);
1610
1611 plk = strcat(Iniplk.radBT, num2str(RadBTfiltru, "%d"),newline );

```



```

.....
1663
1664
1665
1666
1667
1668
1669 #..... ↵
.....
1670 function Jemriz = FunNufmriz (Jemriz, hgrf1, hgrf2); ↵
1671     #V Oktááavě jsou v zátvorce za jménem funkce JEN vstupní parametry do ↵
1672     funkce,
1673     #výstupní musí být před rovnítkem a dyš je jich víc tak v hranatejch
1674     #zátvorkách odděleny čárkama.
1675
1676     if (Jemriz == true);
1677         grid (hgrf1, 'off');
1678         grid (hgrf2, 'off');
1679         Jemriz = false;
1680     else
1681         #u obouch gřafúú je to děláno takto, tak tady to tak musí býti taky
1682         #aby to obnovovalo mřížky u malejch dílků
1683         set (hgrf1, "xgrid", "on", "ygrid", "on");
1684         set (hgrf1, "xminorgrid", "on", "yminorgrid", "on");
1685
1686         set (hgrf2, "xgrid", "on", "ygrid", "on");
1687         set (hgrf2, "xminorgrid", "on", "yminorgrid", "on");
1688
1689         Jemriz = true;
1690     endif
1691
1692 endfunction #..... ↵
.....
1693
1694
1695
1696
1697
1698
1699
1700 #..... ↵
.....
1701 function plkbox(message, title) ↵
1702     #uživatelský msg box, ale velikost by byla potřeba upravit podle ↵
1703     vloženého plku...
1704
1705     figuury = findobj('Type', 'figure');           #získání všech ↵
1706     otevřených figur
1707     pocet_figur = length(figuury);                 #počet otevřených figur
1708
1709     if pocet_figur > 0
1710         cisla_figur = get(figuury, 'Number');     #čísla otevřených figuur
1711     else

```

```

1712     maximumo = 98;      # nějaké číslo které snad hned nebude
1713 endif
1714 Cfig = maximumo + 1;
1715
1716 sirplku = 600;
1717
1718 h = figure (Cfig, 'Name', title, 'NumberTitle', 'off', 'MenuBar',
1719             'none', ...
1720             'position', [500, 500, sirplku, 150]);
1721
1722 uicontrol('parent',h , 'Style', 'text', 'String', message,
1723           'FontSize', 12, ...
1724           'FontWeight', 'normal', 'position', [20, 70, sirplku -
1725           25, 50], ...
1726           'HorizontalAlignment', 'center', 'backgroundcolor', [1 1
1727           1]);
1728
1729 sirtlac = 100;
1730 uicontrol('parent',h , 'Style', 'pushbutton', 'String', 'OK',
1731           'position', ...
1732           [sirplku/2 - sirtlac/2, 20, sirtlac, 30], ...
1733           'Callback', @volani_tlac_plkboxu);
1734
1735 #Čekání na stisk tlačítka
1736 uiwait(h); # Pozastaví provádění, dokud není figure zavřena nebo
1737            # tlačítko stisknuto
1738
1739 try
1740     # dyš se figure s plkem zavře křížkem v rohu je tady už zavřená
1741     # a close(h) nemá co zavírat tak udělá botu
1742     close(h);
1743 end
1744
1745 end#.....
1746
1747
1748
1749
1750
1751
1752 #.....
1753
1754 function [hgrf1, hgrf2] = UlozdofileMensi (Cestafile, Jmfile, hgrf1,
1755 hgrf2, men, Velpis);
1756
1757 [hgrf1, hgrf2] = Ulozdofile (Cestafile, Jmfile, hgrf1, hgrf2, men,
1758 Velpis, '-r300', ' ', lower resolution');
1759
1760 endfunction#.....

```

```

.....
1758
1759
1760
1761 #..... ↵
.....
1762 function [hgrf1, hgrf2] = UlozdofileVetsi (Cestafile, Jmfile, hgrf1, ↵
hgrf2, men, Velpis);
1763
1764 [hgrf1, hgrf2] = Ulozdofile (Cestafile, Jmfile, hgrf1, hgrf2, men, ↵
Velpis, '-r600', ', higher resolution');
1765
1766 endfunction#..... ↵
.....
1767
1768
1769
1770
1771
1772 #..... ↵
.....
1773 function [hgrf1, hgrf2] = Ulozdofile (Cestafile, Jmfile, hgrf1, hgrf2, ↵
men, Velpis, Rozlis, plkroz);
1774
1775     #nakreslí se znovu s písmem jedlým pro tisk, vytiskne se a překreslí se
1776     #s písmem pro vobra žofku
1777
1778
1779     global Velpistisk;
1780     global TlacimFiguru;
1781
1782
1783     plk = cstrcat("Enter name for saving to file (as image formats)", ...
1784                 plkroz);
1785     [fjmeno, fcesta, fpocet] = uinputfile ({"*.jpg","Image jpg"; "*.gif", ↵
"Image gif"; ...
1786                 "*.tif","Image tif"; "*.png", "Image png" }, ...
1787                 plk, Cestafile);
1788
1789
1790     if fpocet < 1
1791         odp = msgbox('No file has been selected for saving, nothing will ↵
be saved.' , ...
1792                 'Saving to file');
1793         return;
1794     endif
1795
1796
1797     mysisko = get (gcf, 'pointer');
1798     set (gcf, 'pointer', 'watch'); ↵
#čekatiti!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
1799     set (men.zaklovl, "enable", 'off');
1800     set (men.zakledit, "enable", 'off');
1801
1802     Tlacenice = cstrcat(fcesta , fjmeno);
1803

```

```

1804 plkfig = get(gcf, 'name');
1805 plk = strcat(plkfig, ' - Wait a minute, pushing into file:
      ',Tlacenice);
1806 set (gcf, 'name', plk);
1807
1808
1809
1810
1811 TlacimFiguru = true;
1812
1813 ZaklPosFig = getappdata(gcf,'ZaklPosFig');
1814 VcilJednFig = get (gcf, 'units');
1815 set (gcf, 'units', 'normalized');
1816 VcilPosFig = get (gcf, 'position');
1817
1818 # úprava pozic legend pro výstup do obrazového filé
1819 ZaklPosLegHL = getappdata(gcf,'RelPosLegHL');
1820 ZaklPosLegBT = getappdata(gcf,'RelPosLegBT');
1821 ZaklPosLegHLup = ZaklPosLegHL;
1822 ZaklPosLegBTup = ZaklPosLegBT;
1823 ZaklPosLegHLup(1) = ZaklPosLegHL(1) - 0.008; # úprava počátku
      výšky umístění
1824 ZaklPosLegHLup(2) = ZaklPosLegHL(2) - 0.035; # úprava počátku
      výšky umístění
1825 ZaklPosLegBTup(1) = ZaklPosLegBT(1) - 0.045 # úprava počátku
      šířky umístění
1826 ZaklPosLegBTup(2) = ZaklPosLegBT(2) - 0.075; # úprava počátku
      výšky umístění
1827 setappdata(gcf,'RelPosLegHL', ZaklPosLegHLup);
1828 setappdata(gcf,'RelPosLegBT', ZaklPosLegBTup);
1829
1830
1831 MeniloTuRozmery = false;
1832 if all(ZaklPosFig == VcilPosFig) == false
1833     # při takto udělané podmínce se sem dostane když nebudou všechny
      prvky
1834     # dotyčných polí stejné (stačí jeden různý)
1835
1836     MeniloTuRozmery = true;
1837     set (gcf, 'position', ZaklPosFig);
1838 endif
1839
1840
1841
1842 #některé výstupní proměnné tady nejsou potřeba
1843 [ ~, ~, ~, hgrf1, hgrf2] = VypDFT (Cestafile, Jmfile, hgrf1, hgrf2,
      men, ...
1844                                     Velpistisk, [], false, true, true);
1845
1846 print (gcf,Tlacenice , Rozlis);
1847
1848 # vrácení původních pozic tady, aby v následujícím VypDFT překreslilo
1849 setappdata(gcf,'RelPosLegHL', ZaklPosLegHL);
1850 setappdata(gcf,'RelPosLegBT', ZaklPosLegBT);
1851
1852 [ ~, ~, ~, hgrf1, hgrf2] = VypDFT (Cestafile, Jmfile, hgrf1, hgrf2,

```



```

1903
1904 mysisko = get (gcf, 'pointer');
1905 set (gcf, 'pointer', 'watch');
#čekatiti'.....
1906 set (men.zaklovl, "enable", 'off');
1907 set (men.zakledit, "enable", 'off');
1908
1909
1910
1911 plkfig = get(gcf, 'name');
1912 plk = strcat(plkfig, ' - One moment, reprocessing data');
1913 set (gcf, 'name', plk);
1914
1915
1916 ZpracDat = NoveZprac;
1917
1918
1919 # zafajfkování aktuálního menu pro zpracování dat z filé
1920 for i = 1: length(men.nastpod)
1921     set(men.nastpod(i), "checked", "off")
1922 endfor
1923 set(men.nastpod(ZpracDat), "checked", "on")
1924
1925
1926
1927
1928 #přeježení pro nové ZpracDat - některé výstupní proměnné tady nejsou
potřeba
1929 [ ~, ~, ~, hgrf1, hgrf2] = VypDFT (Cestafile, Jmfile, hgrf1, hgrf2,
men, ...
Velpis , [], false, false, false);
1930
1931
1932
1933
1934
1935 set (gcf, 'name', plkfig);
1936 set (men.zaklovl, "enable", 'on');
1937 set (men.zakledit, "enable", 'on');
1938 set (gcf, 'pointer', mysisko);
#nečekatiti'.....
1939
1940
1941 endfunction#.....
.....
1942
1943
1944
1945
1946
1947
1948
1949
1950 #.....
.....
1951 function ZmenKreslHarm (Cestafile, Jmfile, hgrf1, hgrf2, men, Velpis);
1952

```

```

1953
1954 global AmplHarmV;
1955
1956
1957 mysisko = get (gcf, 'pointer');
1958 set (gcf, 'pointer', 'watch');
#čekatiti'.....

1959
1960 plk = cstrcat("Switch [dB - V] for drawing Y-axis of graph of
harmonics: now it is ");
1961 if AmplHarmV == 0
1962     AmplHarmV = 1;
1963     plk = cstrcat(plk, 'in [V]');
1964 else
1965     AmplHarmV = 0;
1966     plk = cstrcat(plk, 'in [dB]');
1967 end
1968
1969
1970 set (men.nastpodharm, 'text' , plk);
1971 set (men.kontmenu.nastharm, 'text' , plk);
1972
1973
1974 #přeježení pro nové kreslení harmonických - některé výst. prom. nejsou
potřeba
1975 [ ~, ~, ~, hgrf1, hgrf2] = VypDFT (Cestafile, Jmfile, hgrf1, hgrf2,
men, ...
Velpis , [], false, true, false);
1976
1977
1978 set (gcf, 'pointer', mysisko); #ne
čekatiti'.....

1979
1980
1981 endfunction#.....
.....

1982
1983
1984
1985
1986
1987
1988
1989
1990 #.....
.....
1991 function ZmenPocKreslHarm (Cestafile, Jmfile, hgrf1, hgrf2, men, Velpis);
1992
1993
1994 global PocKresHarm PocHarm FposlHarm Fvypis;
1995
1996
1997 mysisko = get (gcf, 'pointer');
1998 set (gcf, 'pointer', 'watch');
#čekatiti'.....

1999
2000

```

```

2001
2002
2003 plk = strcat('Enter last harmonic, after which harmonics will be
drawn in their graph,' ...
2004     ,newline, 'now is last plotted harmonic:'...
2005     ,newline, '    ', num2str(PocKresHarm + 1, '%d'), char(9), char(9)...
2006     , 'i.e. up to frequency ', num2str(sigfig(Fvypis/1000,4)), ' kHz'...
2007     ,newline, 'last possible harmonic for processed file is:'...
2008     ,newline, '    ', num2str(PocHarm + 1, '%d'), char(9), char(9)...
2009     , 'i.e. up to frequency ', num2str(FposlHarm/1000, "%d"), ' kHz'...
2010     ,newline, newline, 'If a value less than 20 is entered, input
will be adjusted to 20,'...
2011     ,newline, 'when entering a value greater than that contained in
calculation from processed'...
2012     ,newline, 'file, input will be adjusted to calculated number of
harmonics.');
```

```

2013
2014
2015 plktit = 'Entering of last harmonic in graph.';
2016 defaults = {PocKresHarm + 1};
2017 rowscols = [1,10];
2018 vlozeno = inputdlg (plk ,plktit ,rowscols , defaults);
2019
2020 if isempty(vlozeno)
2021     # vstup zrušen Cancel => véééén
2022     set (gcf, 'pointer', mysisko); #ne
čekatiti'.....
2023     return;
2024 endif
2025
2026 if isnan(str2double(vlozeno{1}))
2027     #je to NaN (Not a Number)
2028     vstup = PocKresHarm;
2029 else
2030     #je to muňkové pole(může obsahovat v muňkách různé
2031     #typy dat, tady řítěz), musí se převést na číslo...
2032     vstup = fix(str2double(vlozeno{1}));
2033 endif
2034
2035
2036 if vstup - 1 == PocKresHarm
2037     set (gcf, 'pointer', mysisko); #ne
čekatiti'.....
2038     return; #nic se nezměnilo => véééén
2039 endif
2040
2041
2042 if vstup < 20
2043     vstup = 20; #mimumum pro kreslení
2044 endif
2045
2046 if vstup - 1 > PocHarm
2047     vstup = PocHarm + 1;
2048 endif
2049
2050 PocKresHarm = vstup - 1;
```



```
2144
2145 endfunction#..... ↵
.....
2146
2147
2148
2149
2150
2151
2152
2153 #..... ↵
.....
2154 function [Velpis] = NastavVelPis (monitor);
2155
2156
2157     global Velpistisk;
2158
2159
2160     Velpis = 0.00438 * monitor.sirkaLogic + 4.32;
2161     Velpistisk = 0.00078 * monitor.sirkaLogic + 5.98;
2162
2163
2164 endfunction#..... ↵
.....
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177 #..... ↵
.....
2178 function [signalfilt] = DPButerworth (rad_filtru, signal, Fvzorkovani, ↵
kmitocet_zlomu);
2179
2180 #dolní Butterworthova propust řádu rad_filtru:
2181 # první řád je -6 dB/oktáávu
2182 # druhý řád je -12 dB/oktáávu
2183 # třetí řád je -18 dB/oktáávu atd.
2184
2185
2186 if kmitocet_zlomu == 0
2187     smikfrekv = Fvzorkovani/100;
2188 else
2189     smikfrekv = kmitocet_zlomu;
2190 end
2191
2192
2193
2194 if smikfrekv/(Fvzorkovani/2) > 1;
```

```

2195     signalfilt = signal; # smikfrekv je mimo povolený interval, nefiltrovat
2196 else
2197     % návrh Butterworthova filtru:
2198     [b, a] = butter(rad_filtru, smikfrekv/(Fvzorkovani/2), 'low');
2199
2200     signalfilt = filtfilt(b, a, signal); # aplikace oného filtru bez
    zkreslení
2201 end
2202
2203
2204
2205 endfunction#.....
    .....
2206
2207
2208
2209
2210
2211
2212 #.....
    .....
2213 function NastavButt (Cestafile, Jmfile, hgrf1, hgrf2, men, Velpis);
2214
2215
2216     global Fsmik RadBTfiltru JeBTfiltr;
2217
2218
2219     RFtu = RadBTfiltru;
2220     KZtu = Fsmik;
2221     JFtu = JeBTfiltr;
2222
2223
2224
2225     plk1 = strcat('Processed signal will be modified by passing through
    a low-pass filter, and then THD will be calculated' ...
2226                 ,newline, '      (displayed on a separate line). This is
    to suppress influence of unwanted higher harmonics' ...
2227                 ,newline, '      on THD. THD calculated after signal
    passes through a suitably selected low-pass filter' ...
2228                 ,newline, '      should be lower. Butterworth filter was
    chosen because of its maximally flat (within' ...
2229                 ,newline, '      possibilities of filters) characteristic
    in passband.'...
2230                 ,newline, newline, 'Use low pass filter (1 = yes, other
    = no):');
2231
2232
2233
2234
2235     plk2 = strcat(newline, newline, 'Filter order, here it can be in
    range of 1 to 8,' ...
2236                 ,newline, '      for 1st order, low-pass filter has slope
    of -6 dB/octave (-20 dB/decade)'...
2237                 ,newline, '      for 2nd order, low-pass filter has slope
    of -12 dB/octave (-40 dB/decade)'...
2238                 ,newline, '      for 3rd order, low-pass filter has slope

```

```

2239         of -18 dB/octave (-60 dB/decade), etc.'...
2240         ,newline, 'Filter order (range 1 to 8):');
2241
2242
2243 plk3 = strcat(newline, newline, 'Cutoff frequency (Fo) is frequency
at which there is an approximate -3 dB drop. It is entered here' ...
2244         ,newline, '    in Hz and is not limited in any way, but
in order for THD calculations via low-pass filter to be'
2245         ...
,newline, '    acceptable, it should be higher than or
equal to at least 3rd harmonic, i.e., for example, for'
2246         ...
,newline, '    measured frequency of 20 kHz, it should
be at least 60 kHz. For lower Fo, calculated THD will
be' ...
2247         ,newline, '    higher, which may not meet requirement
for suppression of unwanted higher harmonics.
Conversely,' ...
2248         ,newline, '    if selected Fo is higher than highest
harmonic that can be calculated from given signal,
low-pass' ...
2249         ,newline, '    filter will not apply and THD calculated
through it will be the same as THD of the signal
without' ...
2250         ,newline, '    low-pass filter.'...
2251         ,newline, 'Cutoff frequency [Hz]:');
2252
2253
2254
2255
2256 plkcel = {plk1, plk2, plk3};
2257 plktit = 'Entering parameters for Butterworth low-pass filter.';
2258
2259 defaults = {JeBTfiltr, RadBTfiltru, Fsmik};
2260 rowscols = [1,10; 1,10; 1,10];
2261 vlozeno = inputdlg (plkcel ,plktit ,rowscols , defaults);
2262
2263
2264 Prekreslit = false;
2265 Blbyvstup = false;
2266 if isempty(vlozeno)
2267     #vstup byl zrušen
2268     Blbyvstup = true;
2269 endif
2270
2271
2272
2273 if Blbyvstup
2274     # když byl vstup zrušen vrátiti původní hodnoty
2275     RadBTfiltru = RFtu ;
2276     Fsmik = KZtu;
2277     JeBTfiltr = JFtu;
2278 else
2279     if isnan(str2double(vlozeno{1}))
2280         JeBTfiltr = JFtu;

```

```

2281 else
2282     vstup = fix(str2double(vlozeno{1}));
2283
2284     if vstup == 1
2285         JeBTfiltr = true;
2286     else
2287         JeBTfiltr = false;
2288     endif
2289
2290     if JeBTfiltr ~= JFtu # neni ho... pardon, rovno
2291         Prekreslit = true;
2292     endif
2293 endif
2294
2295
2296
2297 if isnan(str2double(vlozeno{2}))
2298     RadBTfiltru = RFtu;
2299 else
2300     vstup = fix(str2double(vlozeno{2}));
2301
2302     if (vstup > 0) & (vstup < 9)
2303         RadBTfiltru = vstup;
2304     else
2305         RadBTfiltru = RFtu;
2306     endif
2307
2308     if RadBTfiltru ~= RFtu
2309         Prekreslit = true;
2310     endif
2311 endif
2312
2313
2314
2315 if isnan(str2double(vlozeno{3}))
2316     Fsmik = KZtu;
2317 else
2318     Fsmik = str2double(vlozeno{3});
2319 endif
2320
2321 if Fsmik ~= KZtu
2322     Prekreslit = true;
2323 endif
2324 endif
2325
2326
2327
2328 if Prekreslit
2329     mysisko = get(gcf, 'pointer');
2330     set(gcf, 'pointer', 'watch');
2331     #čekatiti!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
2332     set(men.zaklovl, "enable", 'off');
2333     set(men.zakledit, "enable", 'off');
2334     set(men.hlnast, "enable", 'off');
2335     #některé výstupní proměnné tady nejsou potřeba

```



```

.....
2385 function [formplk] = Formnamista(cislo, pocmist)
2386
2387
2388 #jeli pocmist <=0 tak véén
2389 if pocmist <= 0
2390     formplk = '%4.2f';
2391     return
2392 endif
2393
2394
2395
2396 #jeli cislo = 0 tak véén
2397 if (sigfig(cislo, pocmist) == 0)
2398     formplk = strcat('%1.', num2str(pocmist-1, '%d'), 'f');
2399     return
2400 endif
2401
2402
2403
2404 mista = log10(abs(cislo));
2405 znaminko = sign(mista);
2406 mista = fix(mista);
2407
2408
2409 if mista >= 0
2410     #cislo je >= 0,1
2411
2412     if mista >= pocmist
2413         pred = mista;
2414         za = 0;
2415     else
2416         if znaminko >= 0
2417             za = pocmist - mista - 1; #pred musí být +1 místo pro x,xx
2418         else
2419             za = pocmist - mista; #ale né pro 0,xx
2420         endif
2421         pred = pocmist - za;
2422
2423     endif
2424 else
2425     #cislo je < 0,1
2426
2427     pred = 1; #nula celá, (0,xx)
2428     za = abs(mista) + pocmist;
2429
2430     if za > 7 #pro víc jak zde daný počet míst za des. tečkou dát ee
2431         formplk = strcat('%1.', num2str(pocmist-1, '%d'), 'e');
2432         return
2433     endif
2434
2435 endif
2436
2437
2438 formplk = strcat('%', num2str(pred, '%d'), '.', num2str(za, '%d'), 'f');

```

```

2439
2440
2441 endfunction#..... ↵
      .....
2442
2443
2444
2445
2446
2447
2448 #..... ↵
      .....
2449 function y = sigfig(x,n,do_round=1)
2450     #NUF:
2451     #I took liberty of using this here (I hope I didn't I took liberty ↵
      too much..)
2452
2453
      %----- ↵
      ----- ↵
2454     % Jonathan R. Senning <senning@gordon.edu>
2455     % Gordon College
2456     % October 22, 2000
2457     % January 2015: Revised and renamed from rnd()
2458     %
2459     % Usage: y = sigfig( x, n [, do_round] )
2460     %
2461     % Returns:
2462     %     y:     The value of x rounded or chopped to n significant ↵
      figures
2463     %
2464     % Parameters:
2465     %     x:     The value to round or chop
2466     %     n:     The number of significant decimal places to keep.
2467     %     do_round: non-zero for rounding (default), 0 for truncation
2468     %
2469     % Rounds or chops x to n significant decimal places
2470
      %----- ↵
      ----- ↵
2471
2472     % Since we can't take the log of a negative number, we need to
2473     % strip any negative sign from x and reapply it at the end. In
2474     % the event that x is zero, we just want to return zero.
2475
2476     s = 1;
2477     if ( x < 0 )
2478         x = -x;
2479         s = -1;
2480     elseif ( x == 0 )
2481         y = x;
2482         return
2483     end
2484
2485     % Compute the proper power of 10 to multiply x by so that n decimal
2486     % places are left to the left of the decimal point.

```

```

2487
2488     a = -( floor( log10( x ) ) - ( n - 1 ) );
2489
2490     % Now multiply x by 10^a, round to the nearest integer, and then
2491     % multiply by 10^(-a). Also, don't forget to reapply the sign of x.
2492
2493     if ( do_round ~= 0 )
2494         y = s * round( x * 10^a ) * 10^(-a);
2495     else
2496         y = s * fix( x * 10^a ) * 10^(-a);
2497     end
2498
2499 endfunction#..... ↵
.....
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509 #..... ↵
.....
2510 function [Indprvmax, Indposmax, pockmitu] = UprnaPruchNulou (signal, ↵
Fvzorkovani);
2511
2512     # úprava na celé kmity od 1. průchodu nulou ↵
    \\\repeated backslashes
2513
2514
2515     # odfiltrování vyšších kmitočtů, zde dolní Butterworthova propust
2516     [signalfilt] = DPButerworth (4, signal, Fvzorkovani,0);
2517
2518
2519
2520
2521     # podle gpt-4o mimi
2522     # Detekce křížení nuly
2523     pruch_nulou = find(diff(sign(signalfilt)));
2524
2525
2526
2527     if length(pruch_nulou) > 3
2528         #pokus o úpravu na průchody nulou
2529
2530         Indprvmax = pruch_nulou(1);
2531         if mod(length(pruch_nulou), 2) == 0
2532             #sudy
2533             Indposmax = pruch_nulou(length(pruch_nulou) - 1);
2534             pockmitu = length(pruch_nulou)/2 - 1;
2535         else
2536             #lichy
2537             Indposmax = pruch_nulou(length(pruch_nulou));
2538             pockmitu = (length(pruch_nulou) - 1)/2;

```

```

2539     end
2540
2541     else
2542         #preparát zařadit celé filé
2543
2544         [Indprvmax, Indposmax, pockmitu] = UprnaCeleFile (signal,
2545         Fvzorkovani)
2546     end
2547
2548     endfunction#.....
2549     .....
2550
2551
2552
2553
2554
2555
2556     #.....
2557     .....
2558     function [Indprvmax, Indposmax, pockmitu] = UprnaCeleodZac (signal,
2559     Fvzorkovani);
2560
2561     #úprava na celé kmity od začátku filé
2562     \\\
2563
2564     #odfiltrování vyšších kmitočtů, zde dolní Butterworthova propust
2565     [signalfilt] = DPButerworth (4, signal, Fvzorkovani,0);
2566
2567     poc_bodu_sign = length(signal); #z celého signálu
2568     Trvani_signalu = poc_bodu_sign / Fvzorkovani; # v sevrteřinách
2569     # Detekce kmitů, tadý z filtrovaného signálu
2570     praach=0.15*abs(max(signalfilt)); # prahová hodnota pro detekci, zde
2571     15% maxima
2572
2573     # indeksy nulových průsečíků (takto celých cyklů)
2574     Ind_0prus = find(signalfilt(1:end-1) < praach & signalfilt(2:end) >=
2575     praach);
2576
2577     Prum_vzd_ind = mean(diff(Ind_0prus)); #průměrný rozdíl indeksů průsečíků
2578     Cas_1kmitu = Prum_vzd_ind * (1/Fvzorkovani); # v sevrteřinách
2579     pockmitu_skut = Trvani_signalu / Cas_1kmitu;
2580
2581     wocas_kmitu = pockmitu_skut - fix(pockmitu_skut); #fix ušmikne celé
2582     číslo
2583     if abs(1 - wocas_kmitu) < 0.0075
2584         wocas_kmitu = 0;
2585         pockmitu_skut = fix(pockmitu_skut) + 1;
2586     end
2587
2588     wocas_v_indexech = fix((Cas_1kmitu * wocas_kmitu) / (1/Fvzorkovani));
2589

```



```

////////////////////////////////////
2686 # částečně s pomocí gpt-4o mimi
2687
2688 poc_bodu_sign = length(signal); # ze zpracovávané části
2689 Trvani_signalu = poc_bodu_sign / Fvzorkovani; # v sešteřinách
2690
2691
2692 fft_signalu = fft(signal);
2693
2694
2695 dvoustr_spektrum = abs(fft_signalu/poc_bodu_sign); # dvoustr. ↵
2696 # spektrum
2697 jednostr_spektrum = dvoustr_spektrum(1:poc_bodu_sign/2+1); # jednostr. ↵
2698 # spektrum
2699 jednostr_spektrum(2:end-1) = 2 * jednostr_spektrum(2:end-1); # úprava ↵
2700 # amplitudy
2701 fft_signalu_jednostr = fft_signalu(1:poc_bodu_sign/2+1); #jednostr. ↵
2702 # komplexní
2703
2704
2705 # Frekvenční vektor
2706 f_vektor = Fvzorkovani*(0:(poc_bodu_sign/2))/poc_bodu_sign;
2707
2708
2709 # Identifikace dominantních frekvencí, prej jsou > 10%, ale tady je 30%
2710 praach = 0.3 * max(jednostr_spektrum); % nastavení prahu pro identifikaci
2711
2712 # najít vrcholy:
2713 [s_picky, ind_s_picek] = findpeaks(jednostr_spektrum, 'MinPeakHeight', ↵
2714 praach);
2715
2716 index_zakl_harm = ind_s_picek(1);
2717
2718
2719 zakladni_frekvence = f_vektor(index_zakl_harm); # základní frekvence
2720
2721 # výpočet počtu kmitů tady, sedí líp pro hodně zkreslené signály,
2722 # pro ně se bude víc lišit od kmity_z_fun
2723 pockmitu = zakladni_frekvence * Trvani_signalu;
2724
2725 if ~ isempty(kmity_z_fun)
2726     if (kmity_z_fun > 1) & (abs(pockmitu - kmity_z_fun) < 1)
2727         pockmitu = kmity_z_fun;
2728     end
2729 endif
2730
2731
2732
2733
2734 # amplituuda základní harmonické
2735 zakl_harmonicka = jednostr_spektrum(index_zakl_harm);

```

```

2736
2737 #ss slozka
2738 ss_slozka = jednostr_spektrum(1);
2739
2740
2741 # Výpočet z ostatních harmonických
2742 % obsažené vyšší harmonické:
2743 Indexy_harmonickych = 2:(floor(poc_bodu_sign/2/index_zakl_harm));
2744 Indexy_harmonickych = Indexy_harmonickych .* (index_zakl_harm-1)+1;
2745
2746
2747 souc_harm_na2 = sum(jednostr_spektrum(Indexy_harmonickych).^2);
2748
2749 # Výpočet THD v %
2750 THD = 100 * sqrt(souc_harm_na2) / zakl_harmonicka;
2751
2752
2753 #konec výpočtu FFT
2754 ///////////////////////////////////////////////////
endfunction#.....
.....
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767 #*****
*****
2768 #*****Hl. program jako funkce qůli volání z
menu*****
2769 function [Cestafile, Jmfile, pocfile, hgrf1, hgrf2] = VypDFT
(Cestafile, ...
2770     Jmfile, hgrf1, hgrf2, men, Velpis, pocfile, VolitFile,
JenPrekres, Jedojpg);
2771
2772 #POZOR: pokud se dá VolitFile = true a filé bude zvoleno (= vybráno a
pou žito)
2773 #     bude programem nastavena proměnná JenPrekres na false, aby se
změny
2774 #     provedly. (Nezáleží tedy na její hodnotě při vstupu do funkce.)
2775
2776
2777
2778 global ZpracDat VsechnyVybery;
2779 global AmplHarmV;
2780
2781 global Fsmik RadBTfiltru JeBTfiltr;
2782
2783 global Osciloskop OsCHAN OscOWO;

```

```

2784 global plknastpod;
2785 global PocKresHarm PocPocHarm PocHarm FposlHarm Fvypis FpocTHD;
2786 global VerzeProg;
2787 global JednHlWosyX JednHlWosyY JednWY;
2788
2789 global hfig1 HlavPlk;
2790
2791
2792
2793
2794 persistent THD ss_slozka pockmitu index_zakl_harm f_vektor;
2795 persistent jednostr_spektrum fft_signalu_jednostr Indexy_harm;
2796 persistent OknoBlack signBlack thdnufBlack jednostr_spektrum_black;
2797 persistent Indprvmax Indposmax kmity_z_fun;
2798
2799 persistent Popisfile;
2800
2801 persistent THDF jednostr_spektrumF signalsmik THDZadPoc;
2802 persistent KresSmikHarm PrubButter KresPrubBT;
2803
2804 persistent procint IndVypis;
2805
2806
2807
2808
2809
2810 if VolitFile
2811
2812
2813     if length(Cestafile) == 0
2814         Cestafile = pwd();           #nynější pracovní adresář
2815     else
2816         #zkusiti, jestli oná cesta ještě je
2817         if exist(Cestafile,'dir') ~= 7;     #dyš vrátí 7 tak je to adresář
2818         Cestafile = pwd();           #nejni, tak dát nyníjší pracovní adresář
2819     endif
2820 endif
2821
2822
2823
2824
2825     # dobrý popis uuuiigetfilé je na (sice pro matlab...)
2826     # https://uk.mathworks.com/help/matlab/ref/uiigetfile.html
2827
2828
2829
2830     if compare_versions(version, '10.2.0','==')
2831         # takle to chodilo ve verzi oktávy 10.2.0 a jeto v ní potřeba,
2832         # uíigetfilé po spuštění oktávy občas předvolí adresář prostředí
2833         # a NĚĚ ten, co tomu posílám v Cestafile....
2834         [Jmfile, Cestafilevyb, pocfile] = uiigetfile ( {'*.csv'; '*.txt'}, ...
2835             "Choose input file (csv or txt)",
2836             cstrcat(Cestafile, '\'));
2837     else

```



```

2884     bota= true;
2885 end_try_catch
2886
2887
2888 if bota
2889     bota = false;
2890     try
2891         datata = importdata (JmfilesCest, ', ',22); # takto to (snad) načte ↵
                jedle pro
2892                                 # dvoukanálový záznam OWOUN, ale taky pro ↵
                simulaci
2893     if Cmuchret(datata.textdata{1}, 'Model,ADS802A') == 0
2894         bota= true;
2895     endif
2896
2897     catch
2898         bota= true;
2899     end_try_catch
2900 endif
2901
2902
2903
2904
2905
2906 if bota
2907     # záznam OWOUN nebo HANTEK filé to nejni, jestli to nejni simulace
2908
2909     bota=false;
2910     try
2911         datata = importdata (JmfilesCest,', ');
2912     catch
2913         bota= true;
2914     end_try_catch
2915 endif
2916
2917
2918
2919
2920 Osciloskop = "";
2921 if bota == false
2922     if length(datata.textdata) == 3
2923         # Hantek, má jen 3 řádky palice
2924         Osciloskop = OscCHAN;
2925     elseif length(datata.textdata) == 22
2926         # OWOUN má 22 řádků palice
2927         Osciloskop = OscOWO;
2928     endif
2929 endif
2930
2931
2932 if length(Osciloskop) == 0;
2933     plk = strcat('Selected file does not correspond to either of ↵
                two,',newline );
2934     plk = strcat(plk, 'oscilloscopes, whose files this program can ↵
                read.',newline);
2935     plk = strcat(plk, '( ',OscCHAN,' a ', OscOWO, ' )',newline);

```



```

2987     catch
2988         RozsachOscProKan(1) = str2double(regex(PlkyOsc{i}, ...
2989             '[-+]?\\d*\\.?.?\\d+([eE][-+]?\\d+)?', 'match', 'once'));
2990         Jeprekodovani = false;
2991     end
2992     PozRoz = i;
2993     break;
2994 endif
2995 endfor
2996
2997 if RozsachOscProKan(1) > 0
2998     if Jeprekodovani
2999         PlkOscUTF8 = sprintf('%c', unicode2native(PlkyOsc{PozRoz},
3000             'cp1250'));
3001         Cmuchplk = 'simulated';
3002     else
3003         PlkOscUTF8 = PlkyOsc{PozRoz};
3004         Cmuchplk = 'simulated';
3005     endif
3006     if Cmuchret(PlkOscUTF8, Cmuchplk) > 0
3007         # je simulace
3008         Hausnumera = str2double(regex(PlkOscUTF8,...
3009             '[-+]?\\d*\\.?.?\\d+([eE][-+]?\\d+)?', 'match'));
3010         JeSimulace = true;
3011         if length(Hausnumera) >= 3
3012             #je simulováno qantování podle bitů AD převodníku
3013             RozsachOscProKan(1) = Hausnumera(3);
3014             bityAD = Hausnumera(2);
3015         endif
3016     else
3017         # převod ze setin mV Hanteku na V:
3018         RozsachOscProKan(1) = RozsachOscProKan(1) * 1e-5;
3019     endif
3020 endif
3021
3022
3023
3024 # čtení palice osciloskopů OWOUN
3025 *****
3026 UzJeObs = 0; # spočtení kolik oných obsadilo, zatím se dáál nepoužívá
3027 if Cmuchret(Osciloskop, OscOWO) > 0
3028     for i = 1:length(PlkyOsc);
3029         PlkOscUTF8 = PlkyOsc{i};
3030
3031         if Cmuchret(PlkOscUTF8, 'Model,') > 0
3032             #přečtení modelu osciloskopu
3033             *****
3034             OsciloskopModel = strcat('OWON
3035                 ',PlkOscUTF8(Cmuchret(PlkOscUTF8, ',')+1:end));
3036             UzJeObs = UzJeObs + 1;
3037         endif
3038     if Cmuchret(PlkOscUTF8, 'vertical Scale,') > 0

```

```

3039 #přečtení vertikálního rozsahu
*****
3040
3041 RozsachOscProKan = str2double(regexpi(PlkOscUTF8, ...
3042     '[-+]?\d*\.\d+([eE] [-+]?\d+)?',
     'match'));
3043
3044
3045 for i = 1:length(RozsachOscProKan);
3046     #můžou být volty nebo mV nebo µV a nic jinšího ?? - byl by
     rozsach = 0!!
3047     Nasrozs(i) = 0;
3048     if Cmuchret(PlkOscUTF8(Cmuchret(PlkOscUTF8,
     ',')+1:length(PlkOscUTF8)), 'mV') > 0
3049         # mV musí být dříf, páč V je tam taky
3050         Nasrozs(i) = 0.001;     #násobitel na V z mV
3051     elseif Cmuchret(PlkOscUTF8(Cmuchret(PlkOscUTF8,
     ',')+1:length(PlkOscUTF8)), 'µV') > 0
3052         # µV musí být dříf, páč V je tam taky
3053         Nasrozs(i) = 0.000001; #násobitel na V z µV
3054     elseif Cmuchret(PlkOscUTF8(Cmuchret(PlkOscUTF8,
     ',')+1:length(PlkOscUTF8)), 'V') > 0
3055         Nasrozs(i) = 1;     #násobitel na V z V
3056     endif
3057     #stínítka má 8 dílků:
3058     RozsachOscProKan(i) = RozsachOscProKan(i) * 8 * Nasrozs(i);
3059 endfor
3060 UzJeObs = UzJeObs + 1;
3061 endif
3062
3063
3064 if Cmuchret(PlkOscUTF8, 'Channel,') > 0
3065     #přečtení počtu zaznamenaných kanalizací
     *****
3066
3067     CislaKanalů = str2double(regexpi(PlkOscUTF8, ...
3068         '[-+]?\d*\.\d+([eE] [-+]?\d+)?', 'match'));
3069
3070     KanalyVeFileVse = PlkOscUTF8(Cmuchret(PlkOscUTF8, ',')+1:end);
3071     KanalyVeFile = KanalyVeFileVse;
3072
3073     Carka = Cmuchret(KanalyVeFile, ",");
3074     i = 1;
3075     if Carka > 0
3076         KanalyVeFilePole{i} = KanalyVeFile(1:Carka-1);
3077         do
3078             i = i + 1;
3079             if i > 10     # podle deklarace pole KanalyVeFilePole(1,x)
3080                 break;
3081             endif
3082             KanalyVeFile = KanalyVeFile(Carka+1:end);
3083             Carka = Cmuchret(KanalyVeFile, ",");
3084             if Carka > 0
3085                 KanalyVeFilePole{i} = KanalyVeFile(1:Carka-1);
3086             else
3087                 KanalyVeFilePole{i} = KanalyVeFile;

```

```

3088         endif
3089         until (Carka == 0) | (Carka == length(KanalyVeFile))
3090     else
3091         KanalyVeFilePole{i} = KanalyVeFile;
3092     endif
3093     UzJeObs = UzJeObs + 1;
3094 endif

3095
3096
3097 if Cmuchret(PlkOscUTF8, '(Units:') > 0
3098     #přečtení jednotek zaznamenaných dat
3099     *****

3100     JednVeFileVse = PlkOscUTF8;
3101     JednVeFile = JednVeFileVse;
3102
3103     Carka = Cmuchret(JednVeFile, ",");
3104     i = 1;
3105     if Carka > 0
3106         JednVeFilePole{i} = JednVeFile(1:Carka-1);
3107         do
3108             i = i + 1;
3109             if i > 10 # podle deklarace pole JednVeFilePole(1,x)
3110                 break;
3111             endif
3112             JednVeFile = JednVeFile(Carka+1:end);
3113             Carka = Cmuchret(JednVeFile, ",");
3114             if Carka > 0
3115                 JednVeFilePole{i} = JednVeFile(1:Carka-1);
3116             else
3117                 JednVeFilePole{i} = JednVeFile;
3118             endif
3119             until (Carka == 0) | (Carka == length(JednVeFile))
3120         else
3121             JednVeFilePole{i} = JednVeFile;
3122         endif
3123         UzJeObs = UzJeObs + 1;
3124     endif

3125
3126
3127 endifor
3128 endif

3129
3130
3131
3132
3133
3134 ZprcKanal = 1;
3135 RozsachOsc = RozsachOscProKan(1);
3136 if length(CislaKanal) > 1
3137     #je víc kana lízací, dát dotas kterou zpracovat
3138
3139     CelkPocKan = length(CislaKanal);
3140     plk = strcat('In processed file:' ...
3141         ,newline, ' ', JmfilesCest...
3142         ,newline, 'are channels ', KanalyVeFileVse ...

```

↩

```

3143     , '    in total ', num2str(CelkPocKan, '%d'), ' channels.'...
3144     , newline, 'Enter channel number (without CH, just number),'...
3145     , ' that will be processed from this list.'...
3146     , newline, newline, 'If you enter a value that is not in channel ↵
    list,'...
3147     , ' FIRST CHANNEL LISTED will be processed.'...
3148     , newline, 'When there are more channels in file than ↵
    ', num2str(MaxPocKanal, '%d')...
3149     , ' (which I do not expect), only first '...
3150     , num2str(MaxPocKanal, '%d'), ' channels are listed here.', newline);
3151
3152
3153
3154     plkTit = 'Entering of measured channel for processing.';
3155     defaults = {CislaKanal(1)};
3156     rowscols = [1,5];
3157     vlozeno = inputdlg (plk ,plkTit ,rowscols , defaults);
3158
3159     if isempty(vlozeno)
3160         #vstup byl zrušen
3161         ZprcKanal = 1;
3162
3163     elseif (unicode2native(upper(strtrim(vlozeno{1}))(1:1))<48) | ...
3164             (unicode2native(upper(strtrim(vlozeno{1}))(1:1))>57)
3165         #první znak nejní číslo
3166         ZprcKanal = 1;
3167
3168     else
3169         ZprcKanal = fix(str2double(vlozeno{1})); #číslo zprc. kanálu
3170         if (ZprcKanal < CislaKanal(1)) | (ZprcKanal > ↵
            CislaKanal(length(CislaKanal)))
3171             ZprcKanal = 1;
3172         else
3173             Priradilo = false;
3174             for i = 1:length(CislaKanal);
3175                 if CislaKanal(i) == ZprcKanal
3176                     ZprcKanal = i; #toto je pořadí oného v poli CislaKanal
3177                     Priradilo = true;
3178                     break; #výskok z Do, While nebo For smyček
3179                 endif
3180             endfor
3181             if ~ Priradilo # to je NOT Priradilo
3182                 ZprcKanal = 1;
3183             endif
3184         endif
3185         RozsachOsc = RozsachOscProKan(ZprcKanal);
3186     endif
3187 endif
3188
3189
3190     signal = datata.data(:,ZprcKanal+1);
3191     casovaniOsc = datata.data(:,1);
3192
3193
3194
3195     if strcmp(Osciloskop, OscOWO)

```

```

3196 # doplnění jednotek do popisu vos hl. gřafu podle filé
3197
3198 PorDwojtecky = Cmuchret(JednVeFilePole{1},':');
3199 JednWY = JednVeFilePole{1}(PorDwojtecky + 1:PorDwojtecky + 1);
3200 if Cmuchret(upper(JednWY), 'S') > 0
3201     JednHlWosyX = strcat(JednHlWosyX, 'seconds');
3202 else
3203     JednHlWosyX = strcat(JednHlWosyX, JednWY, ' ');
3204 endif
3205
3206 PorDwojtecky = Cmuchret(JednVeFilePole{ZprcKanal+1},':');
3207 JednWY = JednVeFilePole{ZprcKanal+1}(PorDwojtecky+1:PorDwojtecky+1);
3208 JednHlWosyY = strcat(JednHlWosyY, JednWY, ' ');
3209 else
3210 # doplnění jednotek do popisu vos hl. gřafu natvrdo
3211
3212     JednHlWosyX = strcat(JednHlWosyX, 'seconds');
3213     JednHlWosyY = strcat(JednHlWosyY, 'V');
3214     JednWY = 'V';
3215 endif
3216
3217
3218
3219
3220 CasKrok = casovaniOsc(2)-casovaniOsc(1); %v sevteřinách
3221 Fvzorkovani = 1/CasKrok; %kmitočet vzorkování
3222
3223
3224 if strcmp(Osciloskop, OscCHAN)
3225 # osciloskop hantek
3226     casovani(1,:) = 0;
3227     for i = 2:length(signal);
3228         casovani(i,:) = casovani(i - 1,:) + CasKrok;
3229     endfor;
3230
3231 elseif strcmp(Osciloskop, OscOWO)
3232 # osciloskop OWOUN - jeho časování má nulu u prostředí, takže použití
3233 # úpravy pro hantek i pro OWOUN to udělá od nuly...
3234     casovani = casovaniOsc;
3235 endif
3236
3237
3238
3239
3240
3241
3242
3243 if ~
JenPrekres#-----
--
3244
3245
3246 #zjištění, jeli možnost použití všechny výběry pro zpracování filé
3247
3248 #zmenšit na miminální velikost řekněme 2048 dat
3249 vel_cel_sig = length(signal);

```

```

3250 koesmik = 2048 / vel_cel_sig;
3251 VsechnyVybery = false;
3252
3253 if koesmik < 1
3254     #redukovat
3255
3256     redukrok = fix(vel_cel_sig / (vel_cel_sig * koesmik));
3257     redukovany_signal = signal(1:redukrok:end); % každá redukrok hodnota
3258
3259     if abs(max(redukovany_signal)) == 0
3260         # BOTA!!!!!!!!!!
3261         plk = strcat('When processing the file:' ...
3262             ,newline, ' ', JmfilesCest...
3263             ,newline, 'bug occurred (the reduced signal contains only
3264                 zeros)!!'...
3265             ,newline,newline, 'This file cannot be processed at this time;
3266                 it probably does not' ...
3267             ,newline, 'have a suitable waveform (pulses, rectangle, etc.)
3268                 for processing...' ...
3269             ,newline, newline, 'After closing this warning, the program
3270                 will be TERMINATED !!');
3271
3272         plktit = 'PROCESSING ERROR';
3273         handlplk = msgbox(plk , plktit, 'modal');
3274         uiwait(handlplk); # čeká na zavření
3275
3276         close all; # smaže fšecky figůůry => ukončí i program
3277     endif
3278
3279     redukovane_casovani = casovani(1:redukrok:end);
3280     Fvzred = 1/(redukovane_casovani(2)-redukovane_casovani(1));
3281
3282     [THDred , ~, pockmitured, ~, ~, ~, ~, ~] = FFTNufsTHD
3283     (redukovany_signal, Fvzred, []);
3284
3285 else
3286     #je málo dat, redukce nemá smysl, THD spočítat z celého signálu
3287
3288     [THDred , ~, pockmitured, ~, ~, ~, ~, ~] = FFTNufsTHD (signal,
3289     Fvzorkovani, []);
3290 end
3291
3292 if THDred < 25 #jen při takovém THD mají výběry průběhů smysl,
3293 jinak kecají
3294     if pockmitured > 2 #a jen dyš je víc kmitů jak aspoň 2
3295         if (vel_cel_sig / pockmitured) > 300
3296             #a jen dyš je víc jak 300 bodů na kmit, jinak neumí jedle vybrat
3297             začátky
3298             VsechnyVybery = true;
3299         endif
3300     endif
3301 endif

```



```

3396 # "-" plná, "--" čárk., ":" tečk., "-." čerch., "" nic:
3397 set (gca, "gridlinestyle", "-");
3398 grid minor on;
3399 grid on;
3400 ti_tulek = strcat('Measured data from file: ', JmfilesCest);
3401
3402
3403 # z IÁÁ: Octave (při vykreslování textu v grafech) interpretuje určité ↵
3404 # jako formátovací sekvence pro subscripts/superscripts. ↵
3405 # Podtržítka "_" je v Octave/gnuplot/Qt interpretováno jako začátek dolního ↵
3406 # indexu.
3407 # Takže aby se zobrazilo jedle vypnout interpretaci formátování.
3408 title(ti_tulek, 'Interpreter', 'none');
3409
3410 xlabel(hgrf1, 'JednHlWosyX');
3411 ylabel(hgrf1, 'JednHlWosyY');
3412 hold on # udržení grafu pro další kreslení do něj
3413
3414
3415
3416
3417 Efhodn = rms(signalcast); #efektivní hovnota, RMS - root mean square
3418
3419
3420
3421
3422
3423
3424 if ~ ↵
3425 JenPrekres#----- ↵
3426 --
3427 #je to: if NOT JenPrekres
3428
3429 # Výpočty THD a Fourierovy transformace ↵
3430 ////////////////////////////////////////////////////////////////////
3431 [THD, ss_slozka, pockmitu, index_zakl_harm, f_vektor, ↵
3432 jednostr_spektrum, ...
3433 fft_signalu_jednostr, Indexy_harm] = FFTNufsTHD (signalcast, ↵
3434 ...
3435 Fvzorkovani, kmity_z_fun);
3436
3437 PocHarm = length(Indexy_harm);
3438
3439 FposlHarm = f_vektor(Indexy_harm(PocHarm));
3440
3441
3442 if PocHarm < PocPocHarm
3443     PocPocHarm = PocHarm;
3444 endif

```

```

3444
3445 THDZadPoc = 100 *
sqrt(sumsq(jednostr_spektrum(Indexy_harm(1:PocPocHarm))))...
3446 / jednostr_spektrum(index_zakl_harm)
3447
3448
3449 if PocHarm < PocKresHarm
3450     PocKresHarm = PocHarm;
3451 endif
3452 IndVypis = round(Indexy_harm(PocKresHarm));
3453 procint = (PocKresHarm/PocHarm) * 100;
3454 Fvypis = round( f_vektor(IndVypis));
3455 FpocTHD = round(f_vektor(round(Indexy_harm(PocPocHarm))));
3456
3457 if ~ isempty(men)
3458 # změnit taky plky v menu pro zadání počtu kreslených počítaných
harmounických
3459     plk = cstrcat("Change in number of plotted harmonics ");
3460     plk = cstrcat(plk, '(now it is ',num2str(PocKresHarm, "%d"), '
i.e. up to ');
3461     plk = cstrcat(plk, num2str(sigfig(Fvypis/1000,4)), ' kHz)');
3462     set (men.nastPocHarm, 'text' , plk);
3463     set (men.kontmenu.nastPocHarm, 'text' , plk);
3464
3465
3466     plk = cstrcat("Change in last calculated harmonics ");
3467     plk = cstrcat(plk, '(now it is ',num2str(PocPocHarm + 1, "%d"), '
i.e. up to ');
3468     plk = cstrcat(plk, num2str(sigfig(FpocTHD/1000,4)), ' kHz)');
3469     set (men.nastPocPocHarm, 'text' , plk);
3470     set (men.kontmenu.nastPocPocHarm, 'text' , plk);
3471 endif
3472
3473
3474
3475
3476
3477 KresPrubBT = false;
3478 KresSmikHarm = false;
3479 if JeBTfiltr
3480
3481     signalsmik = DPButerworth (RadBTfiltru, signalcast, Fvzorkovani,
Fsmik);
3482
3483
3484     [THDF, ~, ~, ~, ~, jednostr_spektrumF, ...
3485     ~, Indexy_harmF] = FFTNufsTHD (signalsmik, Fvzorkovani,
kmity_z_fun);
3486
3487
3488
3489     Fnorm = Fsmik/(Fvzorkovani/2);
3490
3491     if (Fnorm <= 1) && (Fnorm >= 0); #zdvojení operátoru na && je
zkratka
3492

```

```

3493     % Výpočet frekvenční odezvy Butterwortha, PrubButter je komplexní
3494     [b, a] = butter(RadBTfiltru, Fsmik/(Fvzorkovani/2), 'low');
3495     [PrubButter, we] = freqz(b, a, f_vektor, Fvzorkovani);
3496
3497     KresPrubBT = true;
3498
3499     endif
3500
3501
3502
3503
3504
3505     #kresliti když je průběh harmonických přes filtr Butterworth aspoň ↵
    trochu
3506     #jiný než průběh harmonických bez filtru, páč dyš se blíží nebo je ↵
    stejné
3507     #tak kreslení průběhu přes filtr přerazí průběh bez filtru a ↵
    vypadá to,
3508     #jakodyby tam nebyl...
3509     KresSmikHarm = false;
3510
3511     prumkrfilt = mean(jednostr_spektrumF(1:IndVypis));
3512     prumkr = mean(jednostr_spektrum(1:IndVypis));
3513     if abs(1 - prumkrfilt/prumkr) * 100 >= 1 #rozdíl v %
3514         KresSmikHarm = true;
3515     endif
3516 endif
3517
3518
3519
3520
3521
3522
3523
3524     #..... ↵
    ..... ↵
3525     #výpočet THD pro vstupní signál přes Blackmanovo okno (jako příklad)
3526
3527     poc_bodu_sign = length(signalcast);
3528
3529     #výpočet koe ficyjentů wokenic
3530     OknoBlack = blackman(poc_bodu_sign);
3531
3532     #přepoččet fft koe ficyjentama wokenic
3533     #s tečkou (.* ) je to násobení element * element, BEZ (jen *) MATICOVÉ
3534     signBlack = signalcast .* OknoBlack;
3535     [thdnufBlack, ~, ~, ~, ~, jednostr_spektrum_black, ~, ~] = ↵
    FFTNufsTHD ...
3536
3537
3538
3539
3540     #konec výpočtů FFT ↵
    //////////////////////////////////////
3541

```

```

3542 endif#-----↵
-----↵
3543
3544
3545
3546
3547
3548
3549
3550
3551
3552
3553 plot(casovanicast, signBlack, 'linewidth', 0.4, 'linestyle', ':', ↵
    'color','black');
3554
3555
3556 # přidání ↵
legendy+++++++
3557 hlegendy = legend('measured data' , 'meas. data via Blackmann window'); ↵
3558
3559
3560 set(hlegendy, 'Box', 'off'); # vypnout rámeček a možná i pozadí
3561
3562
3563 if ~ Jedojpg
3564     # výstup na vobra žofku
3565     set(hlegendy, 'fontsize', Velpis*1.04); # 0.97);
3566 else
3567     # výstup do obrazového filé
3568     set(hlegendy, 'fontsize', Velpis*0.95);
3569 endif
3570
3571 # nastavení pozice legendy relativně k X wose:
3572 nastavPozLegendyRelatkOse(gca, hlegendy, getappdata(gcf,'RelPosLegHL'));
3573 drawnow;
3574
3575 # uložení do appdat figúry potřebných handlů pro HL gřaf
3576 setappdata(gcf, "hLegHL", hlegendy);
3577 setappdata(gcf, "hOsyXHL", gca);
3578
3579
3580 hold off; #zrušení udržení pro další kreslení
3581 # ↵
+++++++ ↵
+++++++
3582
3583
3584 VobsadDilkyWos (hgrf1, min(casovanicast), max(casovanicast),"X");
3585 VobsadDilkyWos (hgrf1, min(signalcast), max(signalcast),"Y");
3586
3587
3588
3589
3590
3591
3592

```

```

3593
3594
3595
3596
3597 if
VolitFile#-----
-
3598     #vymazání starého grafu harmonických, aby tam při dotazu nesmrđeél:
3599     delete(hgrf2);
3600
3601     plk = cstrcat('Enter a description (identification) of processed
file:');
3602     PopisfileVst = inputdlg (plk , 'Entering description of this
calculation', [1,60], {Popisfile});
3603
3604     if length(PopisfileVst)>0
3605         Popisfile = PopisfileVst{1};
3606     else
3607         if length(Popisfile)==0
3608             Popisfile = "";
3609         endif
3610     end
3611 end
#-----
---
3612
3613
3614
3615
3616
3617
3618
3619
3620
3621 #přidání plků
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
3622 #jsou umístovány v souřadnicích prvního obrázku, musí být za ním
3623
3624
3625
3626
3627
3628 krajeos = axis(); #krajní body os grafu
3629 odstrad = 0.05; #odstup řádků v násobiteli y souřadnice
3630 prvrad = 0.15; #první řádek v násobiteli y souřadnice qůli nyqvistovi
3631 tabnuf = 0.25; #něco jako tabulátor - posunuje plk
3632 pocharvyp = 10; #počet harmonických na výpis
3633 pocplatcislic = 5; #počet platných číslic ve výpise
3634
3635
3636
3637 # trochu posun plků do leva qůli jejich šířce, MUSÍ být uděláno jako
poměr
3638 # rozsahů x, páč krajeos(2) se u každého kreslení jaksi mění, čert ví
proč..
3639 krajeos(1) = krajeos(1) - 0.015*(krajeos(2)-krajeos(1));

```

```

3640
3641
3642
3643 # první řádek
3644 radek = 0;           #číslo vystupovaného řádku (první řádek je 0)
3645
3646 #násl. příkazy text vystoupí text do aktivní figury, v souřadnicích
3647 #os právě kresleného grafu
3648
3649
3650 plk = cstrcat(Popisfile);
3651 # prvrad-0.01 je zvýšení popisu aby byl trochu oddělen aspoň mizerou
3652 # dyš tu nejde podtržení
3653 text
3654     (krajeos(1),krajeos(3)-(krajeos(4)-krajeos(3))*(prvrad-0.01+odstrad*radek),
3655     plk, 'fontsize', Velpis, 'fontweight', 'bold');
3656
3657 radek++;
3658 if JeSimulace
3659     plk = cstrcat('Calculation of harmonic distortion (THD) for
3660     {\bfsimulated signal}:');
3661 else
3662     if strcmp(Osciloskop, OscOWO)
3663         plk = cstrcat('Calculation of harmonic distortion (THD) for
3664         oscilloscope ', '{\bf',OsciloskopModel,':');
3665     else
3666         plk = cstrcat('Calculation of harmonic distortion (THD) for
3667         ', '{\bf',OsciloskopModel,}', ' oscilloscope data type:');
3668     endif
3669 endif
3670 text
3671     (krajeos(1),krajeos(3)-(krajeos(4)-krajeos(3))*(prvrad+odstrad*radek),
3672     plk, 'fontsize', Velpis);
3673
3674
3675 radek++;
3676 plk = cstrcat('File is processed from record ', num2str(Indprvmax,
3677     "%d"),...
3678     ' do ', num2str(round(Indposmax), "%d"), ' and has in total ',...
3679     num2str(length(signal), "%7.0i"), ' records. ');
3680 text
3681     (krajeos(1),krajeos(3)-(krajeos(4)-krajeos(3))*(prvrad+odstrad*radek),
3682     plk, 'fontsize', Velpis);
3683
3684 radek++;
3685 plk = cstrcat('Highest usable F (Nyquist) is ');
3686 plk = cstrcat(plk, num2str(sigfig(Fvzorkovani/2000,4),

```

```

Formnamista(Fvzorkovani/2000,4)), ' kHz');
3686 if JeSimulace
3687     if bityAD <= 0
3688         plk = cstrcat(plk, ', is simulation with max. resolution. ');
3689     else
3690         plk = cstrcat(plk, ', is simulation ');
3691         plk = cstrcat(plk, num2str(bityAD, "%d"), ' bits from ');
3692         plk = cstrcat(plk, num2str(RozsachOsc, "%d"), ' V. ');
3693     endif
3694 else
3695     plk = cstrcat(plk, ', oscilloscope range ');
3696     plk = cstrcat(plk, num2str(RozsachOsc, "%d"), ' V ');
3697     if strcmp(Osciloskop, OscOWO)
3698         plk = cstrcat(plk, ', channel ', KanalyVeFilePole{ZprcKanal});
3699     endif
3700 endif
3701 text
(krajeos(1), krajeos(3)-(krajeos(4)-krajeos(3))*(prvrad+odstrad*radek),
plk, 'fontsize', Velpis);
3702
3703
3704 radek++;
3705 switch ZpracDat
3706     case 1
3707         plk = cstrcat('Whole cycles from first zero crossing are
processed; ');
3708     case 2
3709         plk = cstrcat('Whole cycles from first maximum are processed; ');
3710     case 3
3711         plk = cstrcat('Whole cycles from beginning of file are processed; ');
3712     case 4
3713         plk = cstrcat('Entire file is processed; ');
3714 endswitch
3715 plk = cstrcat('\bf', plk, '); #tučný výstup plku
3716
3717
3718
3719 if (pockmitu == 1)
3720     #plkzac = 'Je zpracován ';
3721     plkzac = 'total ';
3722     plkkon = ' cycle.';
3723 elseif ((pockmitu > 1) & (pockmitu < 5)) # & je and
3724     plkzac = 'total ';
3725     plkkon = ' cycles.';
3726 else
3727     plkzac = 'total ';
3728     plkkon = ' cycles.';
3729 endif
3730 plk = cstrcat(plk, ' ', plkzac, num2str(pockmitu, "%d"), plkkon);
3731 text
(krajeos(1), krajeos(3)-(krajeos(4)-krajeos(3))*(prvrad+odstrad*radek),
plk, 'fontsize', Velpis);
3732
3733
3734
3735

```

```

3736
3737
3738
3739
3740
3741
3742 radek++;radek++;
3743 plk = cstrcat('Effective value (RMS) of full processed signal:');
3744 text
(krajeos(1),krajeos(3)-(krajeos(4)-krajeos(3))*(prvrad+odstrad*radek),
plk, 'fontsize', Velpis);
3745 plk = cstrcat(num2str(sigfig(Efhodn,pocplatcislic),
Formnamista(Efhodn,pocplatcislic)), ' ',JednWY,'rms');
3746 text (krajeos(1) +
(krajeos(2)-krajeos(1))*tabnuf,krajeos(3)-(krajeos(4)-krajeos(3))*(prvra
d+odstrad*radek), plk, 'fontsize', Velpis);
3747
3748
3749
3750 radek++;
3751 plk = cstrcat('Direct current component:');
3752 text
(krajeos(1),krajeos(3)-(krajeos(4)-krajeos(3))*(prvrad+odstrad*radek),
plk, 'fontsize', Velpis);
3753 plk = cstrcat(num2str(sigfig(ss_slozka,pocplatcislic),
Formnamista(ss_slozka,pocplatcislic)), ' ', JednWY);
3754 text (krajeos(1) +
(krajeos(2)-krajeos(1))*tabnuf,krajeos(3)-(krajeos(4)-krajeos(3))*(prvra
d+odstrad*radek), plk, 'fontsize', Velpis);
3755
3756
3757
3758 radek++;
3759 plk = cstrcat('THD full proces. signal up to ',num2str(sigfig(PocHarm
+ 1,8)), 'th harm.', ' ',num2str(sigfig(FposlHarm/1000,4)), ' kHz:');
3760 text
(krajeos(1),krajeos(3)-(krajeos(4)-krajeos(3))*(prvrad+odstrad*radek),
plk, 'fontsize', Velpis);
3761 plk = cstrcat(num2str(sigfig(THD,pocplatcislic),
Formnamista(THD,pocplatcislic)), ' %');
3762 if sigfig(THD, pocplatcislic) <= sigfig(thdnufBlack, pocplatcislic)
3763 text (krajeos(1) +
(krajeos(2)-krajeos(1))*tabnuf,krajeos(3)-(krajeos(4)-krajeos(3))*(prv
rad+odstrad*radek), plk, 'fontsize', Velpis, 'fontweight', 'bold');
3764 else
3765 text (krajeos(1) +
(krajeos(2)-krajeos(1))*tabnuf,krajeos(3)-(krajeos(4)-krajeos(3))*(prv
rad+odstrad*radek), plk, 'fontsize', Velpis);
3766 endif
3767
3768
3769
3770 radek++;
3771 plk = cstrcat('THD full proces. signal through Blackman window:');
3772 text
(krajeos(1),krajeos(3)-(krajeos(4)-krajeos(3))*(prvrad+odstrad*radek),

```

```

3773 plk, 'fontsize', Velpis);
3773 plk = cstrcat(num2str(sigfig(thdnufBlack,pocplatcislic),
Formnamista(thdnufBlack,pocplatcislic)), ' %');
3774 if sigfig(THD, pocplatcislic) >= sigfig(thdnufBlack, pocplatcislic)
3775     text (krajeos(1) +
           (krajeos(2)-krajeos(1))*tabnuf,krajeos(3)-(krajeos(4)-krajeos(3))*(prv
           rad+odstrad*radek), plk, 'fontsize', Velpis, 'fontweight', 'bold');
3776 else
3777     text (krajeos(1) +
           (krajeos(2)-krajeos(1))*tabnuf,krajeos(3)-(krajeos(4)-krajeos(3))*(prv
           rad+odstrad*radek), plk, 'fontsize', Velpis);
3778 endif
3779
3780
3781
3782
3783 if JeBTfiltr
3784     radek++;
3785     plk = cstrcat(' THD through order ',num2str(sigfig(RadBTfiltru,8)), '
Butterworth LP filter, Fo ',num2str(sigfig(Fsmik/1000,4)), ' kHz:');
3786     text
           (krajeos(1),krajeos(3)-(krajeos(4)-krajeos(3))*(prvrad+odstrad*radek),
           plk, 'fontsize', Velpis);
3787     plk = cstrcat(num2str(sigfig(THDF,pocplatcislic),
Formnamista(THDF,pocplatcislic)), ' %');
3788     text (krajeos(1) +
           (krajeos(2)-krajeos(1))*tabnuf,krajeos(3)-(krajeos(4)-krajeos(3))*(prv
           rad+odstrad*radek), plk, 'fontsize', Velpis);
3789 endif
3790
3791
3792
3793 radek++;
3794 plk = cstrcat(' THD calc. from 2nd to ',num2str(PocPocHarm + 1,"%d"));
3795 if PocPocHarm == 2
3796     plk = cstrcat(plk, 'rd harmonic ');
3797 else
3798     plk = cstrcat(plk, 'th harmonics ');
3799 endif
3800 plk = cstrcat(plk, '(up to ',num2str(sigfig(FpocTHD/1000,4)), ' kHz):');
3801 text
           (krajeos(1),krajeos(3)-(krajeos(4)-krajeos(3))*(prvrad+odstrad*radek),
           plk, 'fontsize', Velpis);
3802 plk = cstrcat(num2str(sigfig(THDZadPoc,pocplatcislic),
Formnamista(THDZadPoc,pocplatcislic)), ' %');
3803 text (krajeos(1) +
           (krajeos(2)-krajeos(1))*tabnuf,krajeos(3)-(krajeos(4)-krajeos(3))*(prvra
           d+odstrad*radek), plk, 'fontsize', Velpis);
3804
3805
3806
3807
3808
3809
3810
3811 radek++;radek++;

```

```

3812 plk = cstrcat('{\bfList of harmonics} (here only up to           ↵
      ',num2str(pocharvyp,"%d"),'. harmonics):');
3813 text                                                         ↵
      (krajeos(1),krajeos(3)-(krajeos(4)-krajeos(3))*(prvrad+odstrad*radek), ↵
      plk, 'fontsize', Velpis) #, 'fontweight', 'bold');
3814
3815
3816
3817
3818 tabnuf1 = 0.1;
3819 tabnuf2 = 0.175;
3820 tabnuf3 = 0.25;
3821
3822
3823
3824
3825
3826 radek++;
3827 text (krajeos(1) +                                           ↵
      (krajeos(2)-krajeos(1))*tabnuf1,krajeos(3)-(krajeos(4)-krajeos(3))*(prvr ↵
      ad+odstrad*radek), 'Frequency', 'fontsize', Velpis, 'fontweight', ↵
      'bold');
3828 text (krajeos(1) +                                           ↵
      (krajeos(2)-krajeos(1))*tabnuf2,krajeos(3)-(krajeos(4)-krajeos(3))*(prvr ↵
      ad+odstrad*radek), 'Amplitude', 'fontsize', Velpis, 'fontweight', ↵
      'bold');
3829 text (krajeos(1) +                                           ↵
      (krajeos(2)-krajeos(1))*tabnuf3,krajeos(3)-(krajeos(4)-krajeos(3))*(prvr ↵
      ad+odstrad*radek), 'Phase' , 'fontsize', Velpis, 'fontweight', 'bold');
3830
3831
3832
3833
3834
3835
3836 radek++;
3837 text                                                         ↵
      (krajeos(1),krajeos(3)-(krajeos(4)-krajeos(3))*(prvrad+odstrad*radek), ↵
      'Primary harmonic:', 'fontsize',Velpis);
3838 # freqence
3839 plk = cstrcat(num2str(sigfig(f_vektor(index_zakl_harm),pocplatcislic), ↵
      Formnamista(f_vektor(index_zakl_harm),pocplatcislic)), ' Hz');
3840 text (krajeos(1) +                                           ↵
      (krajeos(2)-krajeos(1))*tabnuf1,krajeos(3)-(krajeos(4)-krajeos(3))*(prvr ↵
      ad+odstrad*radek), plk, 'fontsize', Velpis)
3841 # ampli túúda
3842 plk =                                                         ↵
      cstrcat(num2str(sigfig(jednostr_spektrum(index_zakl_harm),pocplatcislic) ↵
      , Formnamista(jednostr_spektrum(index_zakl_harm),pocplatcislic)), ' ↵
      ',JednWY);
3843 text (krajeos(1) +                                           ↵
      (krajeos(2)-krajeos(1))*tabnuf2,krajeos(3)-(krajeos(4)-krajeos(3))*(prvr ↵
      ad+odstrad*radek), plk, 'fontsize', Velpis)
3844 # fááze
3845 fazev = faze(real(fft_signalu_jednostr(index_zakl_harm)), ↵
      imag(fft_signalu_jednostr(index_zakl_harm)));

```

```

3846 plk = cstrcat(num2str(sigfig(fazev, pocplatcislic),
Formnamista(fazev,pocplatcislic)), ' °');
3847 text (krajeos(1) +
(krajeos(2)-krajeos(1))*tabnuf3,krajeos(3)-(krajeos(4)-krajeos(3))*(prvr
ad+odstrad*radek), plk , 'fontsize', Velpis)
3848
3849
3850
3851
3852
3853
3854
3855
3856 for i = 2:pocharvyp
3857     radek++;
3858     plk = cstrcat(num2str(i,"%d"),'. harmonic:');
3859     text
(krajeos(1),krajeos(3)-(krajeos(4)-krajeos(3))*(prvrad+odstrad*radek),
plk, 'fontsize',Velpis);
3860 # frequence
3861     plk =
cstrcat(num2str(sigfig(f_vektor((index_zakl_harm-1)*i+1),pocplatcislic
), Formnamista(f_vektor((index_zakl_harm-1)*i+1),pocplatcislic)), '
Hz');
3862     text (krajeos(1) +
(krajeos(2)-krajeos(1))*tabnuf1,krajeos(3)-(krajeos(4)-krajeos(3))*(pr
vrad+odstrad*radek), plk, 'fontsize', Velpis)
3863 # ampli túúda
3864     plk =
cstrcat(num2str(sigfig(jednostr_spektrum((index_zakl_harm-1)*i+1),pocp
latcislic),
Formnamista(jednostr_spektrum((index_zakl_harm-1)*i+1),pocplatcislic))
,' ',JednWY);
3865     text (krajeos(1) +
(krajeos(2)-krajeos(1))*tabnuf2,krajeos(3)-(krajeos(4)-krajeos(3))*(pr
vrad+odstrad*radek), plk, 'fontsize', Velpis)
3866 # fááze
3867     fazev = faze(real(fft_signalu_jednostr((index_zakl_harm-1)*i+1)),
imag(fft_signalu_jednostr((index_zakl_harm-1)*i+1)));
3868     plk = cstrcat(num2str(sigfig(fazev, pocplatcislic),
Formnamista(fazev,pocplatcislic)), ' °');
3869     text (krajeos(1) +
(krajeos(2)-krajeos(1))*tabnuf3,krajeos(3)-(krajeos(4)-krajeos(3))*(pr
vrad+odstrad*radek), plk , 'fontsize', Velpis)
3870 endfor
3871
3872
3873
3874
3875
3876 radek++;
3877 plk = '@ Luděk Ruffer          lruffer@volny.cz';
3878 plk = cstrcat(plk, '          program version dated ', VerzeProg);
3879 # v příkazu color [r,g,b] se míchají světla: třeba 0,0,0, je černá..
3880 # třeba [0, 0.35, 0.65] dál je tmafší modrá
3881 text

```



```

3930 set (gca, "xgrid", "on", "ygrid", "on");
3931 set (gca, "gridcolor", "black");
3932 set (gca, "gridlinestyle", "-");
3933
3934 # vlastnost xlimmode MUSÍ být "auto"
3935 set (gca, "xlimmode", "auto");
3936 set (gca, "xlimitmethod", "tight");
3937
3938 grid minor on;
3939 grid on;           #když je až po malé mřížce, udělá velký mřížkový
čáry plný
3940
3941
3942
3943
3944 plk = strcat('Calculated harmonics (plotted up to ', ...
3945     num2str(sigfig(Fvypis/1000,4)), ' kHz', ', i.e. up to ', ...
3946     num2str(PocKresHarm + 1, '%d'), '. harmonic and ', ...
3947     num2str(sigfig(procint,3), '%d'), '% of calculated interval).');
3948 title(plk);
3949
3950
3951
3952
3953 xlabel(hgrf2, 'Frequency [kHz]');
3954
3955 if AmplHarmV > 0
3956     ylabel(hgrf2, 'Amplitude [V]');
3957 else
3958     ylabel(hgrf2, 'Amplitude [dB]');
3959 end
3960 hold on # udržení grafu pro další kreslení do něj
.....
3961
3962
3963
3964
3965
3966 #přidání Blackmannna a harm. přes DP Butterworth, dyš se kreslí
3967
3968 barvaprubBT = [0.45, 0.2, 0.666];
3969
3970 if AmplHarmV > 0
3971     # ve V
3972
3973     plot(f_vektor(1:IndVypis)/1000,
abs(jednostr_spektrum_black(1:IndVypis)), ...
'linewidth', 0.5, 'linestyle', ':', 'color', 'black');
3974
3975
3976 if KresSmikHarm
3977     handlBT = semilogy(f_vektor(1:IndVypis)/1000,
(jednostr_spektrumF(1:IndVypis)), ...
'linewidth', 0.35, 'linestyle', '-.', 'color', 'red');
3978
3979
3980 Ywosyrozsz = ylim; # zapamatování si rozsachu wosy y teď
3981

```

```

3982
3983     hlegendyBT = legend(handlBT, 'harmonics through Butterworth LP
    filter');
3984
3985     if KresPrubBT
3986         hav = abs(PrubButter); #prevod na vektor z komplexu a ve V
3987
3988         #přidání další proměnné do legendy takto:
3989         plot(f_vektor(1:IndVypis)/1000, hav(1:IndVypis), ...
3990             'linewidth', 0.35, 'linestyle', '--', 'color',
    barvaprubBT, ...
3991             "displayname", "characteristics of Butterworth LP filter");
3992     endif
3993
3994
3995     ylim(Ywosyrozs); # vrácení rozsahu wosy y na hodnoty před
    kreslením průběhu,
3996         # aby ho průběh nezvětšoval
3997
3998 endif
3999
4000
4001 else
4002     # v dB
4003
4004     jednostr_spektrum_black_dB = 20 *
    log10(jednostr_spektrum_black/jednostr_spektrum_black(index_zakl_harm)
    );
4005     plot(f_vektor(1:IndVypis)/1000,
    (jednostr_spektrum_black_dB(1:IndVypis)), ...
4006         'linewidth', 0.5, 'linestyle', ':', 'color','black');
4007
4008
4009     if KresSmikHarm
4010
4011         jednostr_spektrum_F_dB = 20 *
    log10(jednostr_spektrumF/jednostr_spektrumF(index_zakl_harm));
4012     handlBT = plot(f_vektor(1:IndVypis)/1000,
    (jednostr_spektrum_F_dB(1:IndVypis)), ...
4013         'linewidth', 0.35, 'linestyle', '--', 'color','red');
4014
4015     Ywosyrozs = ylim; # zapamatování si rozsachu wosy y teď
4016
4017
4018     hlegendyBT = legend(handlBT, 'harmonics through Butterworth LP
    filter');
4019
4020     if KresPrubBT
4021         hav = 20*log10(abs(PrubButter)); #prevod na vektor z komplexu
    a v dB
4022
4023         #přidání další proměnné do legendy takto:
4024         plot(f_vektor(1:IndVypis)/1000, hav(1:IndVypis), ...
4025             'linewidth', 0.35, 'linestyle', '--', 'color', barvaprubBT,
    "displayname", "characteristics of Butterworth LP filter");
4026
4027     endif

```

```

4028
4029     ylim(Ywosyrozsz); # vrácení rozsahu wosy y na hodnoty před
      kreslením průběhu,
4030                                     # aby ho průběh nezvětšoval
4031     endif
4032
4033 end
4034
4035
4036
4037 # a nastavení legendy pro Butterwortha dyš je, jen jednou tu
4038 if KresSmikHarm
4039
4040     set(hlegendyBT, 'Box', 'off');
4041     if Jedojpg
4042         set(hlegendyBT, 'fontsize', Velpis*1.04); # 0.97);
4043     else
4044         set(hlegendyBT, 'fontsize', Velpis*0.97);
4045     endif
4046
4047     nastavPozLegendyRelatKOse(gca, hlegendyBT, getappdata(gcf,
      'RelPosLegBT'));
4048     drawnow;
4049
4050     # uložení do appdat figúry potřebných handlů pro BT gřaf
4051     setappdata(gcf, "hLegBT", hlegendyBT);
4052     setappdata(gcf, "hOsyXBT", gca);
4053
4054     endif
4055
4056 # uložení do appdat figúry proměnné KresSmikHarm pro BT gřaf (jestli
      je legenda)
4057 setappdata(gcf, "KresSmikHarm", KresSmikHarm);
4058
4059
4060 hold off # konec udržení gřafu pro další kreslení do něj
      .....
4061
4062
4063 #frequenční wosa by měla být dycky od 0
4064 VobsadDilkyWos (hgrf2, 0, max(f_vektor(1:IndVypis)/1000), "X");
4065 if AmplHarmV == 0
4066     # jen pro dB, zde maximumo by vždy měla být 0
4067     # a miminumo vybrat, je tam víc hodnot
4068
4069     # následující 2 hodnoty se kreslí vždycky, další (snad) miminum
      nepřesáhnou
4070     minBlack = min(jednostr_spektrum_black_dB(1:IndVypis));
4071     minSig = min(jednostr_spektrum_dB(1:IndVypis));
4072     if KresSmikHarm
4073         minHarm = min(jednostr_spektrum_F_dB(1:IndVypis));
4074     else
4075         minHarm = 9999.9; #nějaké maximumo které miminumo nemá šanci
      přelést
4076     endif
4077

```



```
4130
4131
4132
4133
4134
4135 #*****
*****
4136 # hlavní
program-----
4137 #*****
*****
4138
4139
4140
4141 # Tady, rači hned na začátku (i když to tak nevypadá) je definovaná
fuňkce,
4142 # která se zavolá při ukončení celé oktávy - je to qůli výmazu filé
4143 # Póówlšelllu, které se používá k indikaci, jeli aktiwní vokno maxima
lízováno.
4144 atexit ("Oktavasezavira");
4145
4146
4147
4148
4149
4150
4151 #velikost vobra žofky 0 v pikslích
4152 set (0, 'units','pixels')
4153
4154 #Oktaava takto monitoruje jen PRIMÁRNÍ monitor, no je zadarmo...
4155 [velobr] = get(0,"monitorpositions");
4156 dpi = get(0, 'screenpixelsperinch'); #toto celkem odpovídá a je to
fyzické DPI
4157 obrcmvypsir = 2.54* (velobr(3)-velobr(1))/dpi; #skut. rozměr šířky
monitoru v cm
4158 obrcmvypvys = 2.54* (velobr(4)-velobr(2))/dpi; #skut. rozměr výšky
monitoru v cm
4159
4160
4161
4162
4163
4164
4165
4166
4167 #*****
*****
4168 #pokus o načtení ini filé s stejným jménem jako toto spuštěné .m filé
4169 #a s rozšířením .NUF (to snad ještě není)
4170
4171
4172 Cestaulozena = '';
4173 AmplHarmV = 0;
4174 Fsmik = 50000;
4175 RadBTfiltru = 2;
4176 JeBTfiltr = true;
```

```

4177 PoziceMinula = [-999, -999, -999, -999];
4178
4179
4180 Jminifile = strcat(mfilename, '.nuf');
4181
4182 Pocradfile = 0;
4183 radkyfile = [''];
4184
4185 # Otevření souboru pro čtení
4186 fid = fopen(Jminifile, 'r');
4187
4188
4189 # Kontrola, zda se soubor otevřel správně, kdyby neé, tak fid bude = -1
4190 # a když se neotevřel tak nedělat nic
4191 if fid ~= -1 # není rovno
4192
4193     # Čtení z filé po řádcích
4194     Pocradfile = 0;
4195     while ~feof(fid)
4196         Pocradfile++;
4197         % Načtení jednoho řádku, takto celý řádek (s :):
4198         radkyfile {Pocradfile,1} = fgetl(fid);
4199     end
4200
4201     # Zavření filé
4202     fclose(fid);
4203
4204
4205     # zpracování filé
4206     Cestaulozena = ''; #dyby ve filé nebylo
4207     if Pocradfile > 0
4208         for i = 1 : Pocradfile
4209
4210
4211             pozice = strfind(radkyfile {i,1}, Iniplk.amplharmveV);
4212             if pozice > 0
4213                 pozice = strfind(radkyfile {i,1}, '=');
4214                 AmplHarmV = str2num(radkyfile {i,1}(pozice + 1:end));
4215
4216                 if AmplHarmV ~ 0
4217                     AmplHarmV = 1; # V
4218                 end
4219             end
4220
4221
4222             pozice = strfind(radkyfile {i,1}, Iniplk.PocKrHarm);
4223             if pozice > 0
4224                 pozice = strfind(radkyfile {i,1}, '=');
4225                 PocKresHarm = str2num(radkyfile {i,1}(pozice + 1:end));
4226                 if isempty(PocKresHarm)
4227                     PocKresHarm = 200;
4228                 endif
4229             endif
4230
4231
4232             pozice = strfind(radkyfile {i,1}, Iniplk.PocPocHarm);

```

```

4233     if pozice > 0
4234         pozice = strfind(radkyfile {i,1}, '=');
4235         PocPocHarm = str2num(radkyfile {i,1}(pozice + 1:end));
4236         if isempty(PocPocHarm)
4237             PocPocHarm = 2;
4238         endif
4239     endif
4240
4241
4242     pozice = strfind(radkyfile {i,1}, Iniplk.zpracdat);
4243     if pozice > 0
4244         pozice = strfind(radkyfile {i,1}, '=');
4245         ZpracDat = str2num(radkyfile {i,1}(pozice + 1:end));
4246         if isempty(ZpracDat)
4247             ZpracDat = 1;
4248         else
4249             ZpracDat = fix(ZpracDat);
4250             if ZpracDat < 1 | ZpracDat > 4 # & je and; | je or; ~ je ↵
4251                 not, cocoti
4252                 ZpracDat = 1;
4253             endif
4254         endif
4255     endif
4256
4257     pozice = strfind(radkyfile {i,1}, Iniplk.cestafile);
4258     if pozice > 0
4259         pozice = strfind(radkyfile {i,1}, '=');
4260         Cestaulozena = strtrim (radkyfile {i,1}(pozice + 1:end));
4261     endif
4262
4263
4264     pozice = strfind(radkyfile {i,1}, Iniplk.PoziceFig);
4265     if pozice > 0
4266         pozice = strfind(radkyfile {i,1}, '=');
4267         PoziceMinula = str2num (radkyfile {i,1}(pozice + 1:end));
4268     endif
4269
4270
4271     pozice = strfind(radkyfile {i,1}, Iniplk.pouzBT);
4272     if pozice > 0
4273         pozice = strfind(radkyfile {i,1}, '=');
4274         plk = strtrim (radkyfile {i,1}(pozice + 1:end));
4275         if strcmp(upper(plk), 'YES')
4276             JeBTfiltr = true;
4277         else
4278             JeBTfiltr = false;
4279         endif
4280     endif
4281
4282     pozice = strfind(radkyfile {i,1}, Iniplk.fsmikBT);
4283     if pozice > 0
4284         pozice = strfind(radkyfile {i,1}, '=');
4285         Fsmik = str2num (radkyfile {i,1}(pozice + 1:end));
4286         if isempty(Fsmik)
4287             Fsmik = 50000;

```

```

4288         else
4289             Fsmik = fix(Fsmik);
4290         endif
4291     endif
4292
4293     pozice = strfind(radkyfile {i,1}, Iniplk.radBT);
4294     if pozice > 0
4295         pozice = strfind(radkyfile {i,1}, '=');
4296         RadBTfiltru = str2num (radkyfile {i,1}(pozice + 1:end));
4297         if isempty(RadBTfiltru)
4298             RadBTfiltru = 2;
4299         else
4300             RadBTfiltru = fix(RadBTfiltru);
4301             if RadBTfiltru < 1 | RadBTfiltru > 8
4302                 RadBTfiltru = 2;
4303             endif
4304         endif
4305     endif
4306
4307
4308
4309     endfor
4310
4311     endif
4312
4313 endif
4314 #xonec čtení filé
4315 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
4316
4317 if exist(Cestaulozena,'dir') == 7    #dyš vrátí 7 tak je to adresář
4318     Cestafile = Cestaulozena;
4319 endif
4320
4321
4322
4323
4324
4325
4326
4327
4328
4329
4330 # načtení hodnot o všech monitourech jinak, dyš to oktááva neumí,
4331 # až po přečtení ini filé
4332 # ale funguje to jen ve Voknech
4333 [Bota, Monitory] = Zjisti_monitory();
4334
4335
4336 if length(Bota)>0
4337     #ošetření případné obuvi při zjišťování monitourů PowerodeŠelem
4338
4339     plk1 = strcat('Error occurred while detecting connected monitors
4340                 ,newline, Bota ...
4341                 ,newline,newline, 'Correct following values according to

```

```

reality, because'...
4342 , ' Octave probably cannot read (or I do not know how to ↵
find out) display'...
4343 ,newline, ' magnification in percent specified in ↵
Windows settings,'...
4344 , ' it is in Settings/System/Display,'...
4345 ,newline, 'and check resolution of main monitor to see ↵
if it is read correctly.'...
4346 ,newline, ' (If entered values do not correspond to ↵
reality, main program'...
4347 , ' window may not display correctly.)'...
4348 ,newline, newline, 'Horizontal resolution of main ↵
display in pixels:');
4349
4350 plk2 = strcat(newline, newline, 'Vertical resolution of main ↵
display in pixels:');
4351
4352 plk3 = strcat(newline, newline, 'Display magnification in percent:');
4353
4354 plkcel = {plk1, plk2, plk3};
4355 plkitit = 'Manual entry of parameters for size of base window';
4356
4357
4358 defaults = {velobr(3), velobr(4), 100};
4359 rowscols = [1,10; 1,10; 1,10];
4360 vlozeno = inputdlg (plkcel ,plkitit ,rowscols , defaults);
4361
4362
4363 Blbyvstup = false;
4364 if isempty(vlozeno)
4365     #vstup byl zrušen
4366     Blbyvstup = true;
4367 endif
4368
4369
4370 # deklarace pole Hlmonitor
4371 Hlmonitor = struct('sirkaSkut', -999, 'vyskaSkut', -999, ↵
'zvetseni_proc', ...
4372                 -999, 'sirkaLogic', -999, 'vyskaLogic', -999);
4373
4374
4375 if Blbyvstup
4376     # když byl vstup zrušen dáti načtené
4377     Xrozlmon = velobr(3);
4378     Yrozlmon = velobr(4);
4379     ZvetsMon = 100;
4380 else
4381     if isnan(str2double(vlozeno{1}))
4382         Xrozlmon = velobr(3);
4383     else
4384         Xrozlmon = str2double(vlozeno{1});
4385     endif
4386
4387
4388     if isnan(str2double(vlozeno{2}))
4389         Yrozlmon = velobr(4);

```

```

4390     else
4391         Yrozlmon = str2double(vlozeno{2});
4392     endif
4393
4394
4395     if isnan(str2double(vlozeno{3}))
4396         ZvetsMon = 100;
4397     else
4398         ZvetsMon = str2double(vlozeno{3});
4399     endif
4400
4401     endif
4402
4403     # no a nastavení pole Hlmonitor podle vstupu:
4404     Hlmonitor.sirkaLogic = Xrozlmon/ZvetsMon*100;
4405     Hlmonitor.vyskaLogic = Yrozlmon/ZvetsMon*100;
4406     Hlmonitor.zvetseni_proc = ZvetsMon;
4407     Hlmonitor.sirkaSkut = Xrozlmon;
4408     Hlmonitor.vyskaSkut = Yrozlmon;
4409
4410     # a taky pole Monitory pro jeden monitor podle vstupu:
4411     Monitory(1).primarni = 1;
4412     Monitory(1).sirkaLogic = Hlmonitor.sirkaLogic;
4413     Monitory(1).vyskaLogic = Hlmonitor.vyskaLogic;
4414     Monitory(1).zvetseni_proc = Hlmonitor.zvetseni_proc;
4415     Monitory(1).sirkaSkut = Hlmonitor.sirkaSkut;
4416     Monitory(1).vyskaSkut = Hlmonitor.vyskaSkut;
4417     Monitory(1).rucni_vstup = 1;
4418
4419     else
4420         # dyš není bota tak vybrání hlavního monitoru, aby se nemusel dále ↵
         hledati
4421
4422         for i=1 : numel(Monitory)
4423             if Monitory(i).primarni
4424                 Hlmonitor.sirkaLogic = Monitory(i).sirkaLogic;
4425                 Hlmonitor.vyskaLogic = Monitory(i).vyskaLogic;
4426                 Hlmonitor.zvetseni_proc = Monitory(i).zvetseni_proc;
4427                 Hlmonitor.sirkaSkut = Monitory(i).sirkaSkut;
4428                 Hlmonitor.vyskaSkut = Monitory(i).vyskaSkut;
4429                 break;
4430             endif
4431         endfor
4432
4433         #konec ošetření případné obuvi při zjišťování monitourů PowerodeŠelem
4434     endif
4435
4436
4437
4438
4439
4440
4441
4442
4443
4444

```

```

4445 [Velpis] = NastavVelPis (Hlmonitor);
4446
4447
4448
4449 hfig1 = figure (1, 'numbertitle', 'off' , 'name', HlavPlk, 'visible',
'off');
4450
4451
4452
4453 # nastavení velikosti figúúry v pikslích podle logické velikosti
displayjeje
4454 # a zaokrrouhlit na celá čísla, páč piksle jsou celé
4455 set (hfig1, 'units', 'pixels' , 'position', [ ...
4456     round(Hlmonitor.sirkaLogic * ZaklVelFig.X), ...
4457     round(Hlmonitor.vyskaLogic * ZaklVelFig.Y), ...
4458     round(Hlmonitor.sirkaLogic * ZaklVelFig.Sir), ...
4459     round(Hlmonitor.vyskaLogic * ZaklVelFig.Vys)]);
4460 PozfigPiksle = get (hfig1, 'position');
4461
4462
4463 #nastavení normalizace qůli zjištění normalizačních jednotek
4464 set (hfig1, 'units', 'normalized');
4465 Pozfig = get (hfig1, 'position');
4466
4467
4468 # Uložení základní normalizované pozice figuury po spuštění do appdat
figúry
4469 # (tady aby byla normálized)
4470 setappdata(hfig1, 'ZaklPosFig', Pozfig);
4471
4472
4473 #a dál je to zase v PIKSLÍCH:
4474 set (hfig1,
'units', 'pixels', 'papertype', 'a4', 'paperorientation', 'landscape');
4475 set (hfig1, 'paperunits', 'centimeters');
4476 #menubar none zruší jejich menu, bude jen moje:
4477 set (hfig1, 'resize', 'on', 'menubar', 'none');
4478
4479 set (hfig1, 'visible', 'on');
4480
4481
4482
4483
4484
4485 # nastavení min a max rozměrů figúry tady V PIKSLÍCH a zaokr. na celá
čísla:
4486 # nasobSirky je úprava podle mého zkoušení aby se plky v min. okně
nepřekrývaly
4487 nasobSirky = 2.2189E-08 * Hlmonitor.sirkaLogic^2 - 1.7899E-04 ...
4488             * Hlmonitor.sirkaLogic + 1.0591E+00;
4489 minSir = round(PozfigPiksle(3)*nasobSirky);
4490 maxSir = round(PozfigPiksle(3));
4491 minVys = round(PozfigPiksle(4)*0.82);
4492 maxVys = round(PozfigPiksle(4));
4493
4494 % a jejich uložení do UserData od figúry

```

```

4495 set (hfig1, 'UserData', struct('zacX',PozfigPiksle(1),'minSir',minSir, ↵
...
4496                                     ↵
    'maxSir',maxSir,'zacY',PozfigPiksle(2),'minVys',minVys,'maxVys',max ↵
    Vys));
4497
4498
4499
4500
4501 # Z relativních pozic legend vzhledem k x ose (v rel. jednotkách [0 až ↵
1]) se
4502 # tu od 11.2.2026 používá jen pozice (2) - výškové umístění, protože ↵
se mě
4503 # asi nepodařilo najít jedlou relativní pozici pro umístění vodorovné ↵
a to
4504 # pak nesedělo jedle, takže jsem vodorovnou pozici nechal na octááve
4505 # bližší k tomu je ve funkci nastavPozLegendyRelatKose(). Hodnoty jsou tu
4506 # ty, se kterými to jaksi nejlíp pasovalo pro anglickou verzi.
4507 #
4508 # Relativní pozice legendy v rámci osy pro graf hlav. průběhu
4509 rel_pos_leg = [0.82, -0.23, 0.19, 0.15];
4510 setappdata(hfig1,'RelPosLegHL',rel_pos_leg); #a její uložení do ↵
appdat figúry
4511
4512 # Relativní pozice legendy v rámci osy pro graf harmonických (je menší)
4513 rel_pos_leg_BT = [0.66, -0.225, 0.4, 0.15];
4514 setappdata(hfig1,'RelPosLegBT',rel_pos_leg_BT); #a její uložení do ↵
appdat figúry
4515
4516
4517
4518
4519
4520 # a nastavení pozice figúry jaká byla při minulém ukončení programu, jeli
4521 if all(PoziceMinula > 0)
4522     if PoziceMinula(3) <= PozfigPiksle(3) && PoziceMinula(4) <= ↵
PozfigPiksle(4)
4523         # Minulou pozici nastavit jen dyš program minule neskončil s ↵
maximalizovaným
4524         # woknem, páč příkaz k maksima lizaci wokna wotáve nemá, a ↵
uložené rozměry
4525         # jsou pro okno maximalizované, a kdyš by se tu nastavily, bude ↵
wokno větší
4526         # a nepújde se dostat na horní lištu (nebude viděti)
4527         #
4528         # Ale taky kontrolovat počátek Y a výšku figúúry, páč nevím čím, ↵
ale i tak
4529         # se vobčas zobrazí s horní lištou nahoře mimo vobražofku, což je ↵
blbé
4530
4531
4532
4533     if PoziceMinula(4) + PoziceMinula(2) > maxVys
4534
4535         if PoziceMinula(4) > maxVys
4536             PoziceMinula(4) = maxVys;

```

```

4537         endif
4538         if PoziceMinula(4) + PoziceMinula(2) > PozfigPiksle(4) +
PozfigPiksle(2)
4539             PoziceMinula(2) = PozfigPiksle(2);
4540         endif
4541
4542     endif
4543
4544     set (hfig1, 'position', PoziceMinula);
4545
4546     endif
4547 endif
4548
4549
4550
4551
4552
4553
4554 # groot je default graphics systém
4555 # třeba: get(groot, 'ScreenSize')
4556 #
4557 # gcf   je current figúre
4558 # gca   je curren axes
4559 # gco   je current objekt
4560
4561 # třeba nastavení defaultního písma:
4562 # set(groot, 'DefaultTextFontName', 'Times New Roman')
4563 # set(groot, 'DefaultTextFontName', 'Arial')
4564 # set(groot, 'DefaultTextFontSize', Velpis)
4565
4566
4567
4568
4569
4570
4571 Jemriz = true; #po spuštění mřížky jsou, tak takto
4572
4573
4574 # první zavolání ocaď
4575 # při prvním volání ignorovat zatím neobsazene a nefinované vst.
proměnné pomocí
4576 # prázdného argumentu []
4577 [Cestafile, Jmfile, pocfile, hgrf1, hgrf2] = VypDFT (Cestafile, [],
[], [], ...
4578                                     [], Velpis, [], true, false,
false);
4579
4580
4581
4582
4583
4584
4585 #*****
*****
4586 #menu, až nakonec, aby už byly figuury a jejich handleové
-----

```

```

4587
4588 #hl menu - !! v hl menu akcelerátory nějak nefungujou !!!
4589 men.zaklovl = uimenu ("text", "DFT control");
4590 men.hlnast = uimenu ("text", "Settings");
4591 men.zakledit = uimenu ("text", "Editing charts");
4592
4593
4594
4595
4596
4597
4598
4599 #podmenu
4600 plk = cstrcat(plknastpod{1});
4601 men.nastpod(1) = uimenu (men.hlnast, "text", plk, "menuselectedfcn", ...
4602     "PrejezZprac (Cestafile, Jmfile, hgrf1, hgrf2, men, Velpis, ↵
4603     1)");
4604 plk = cstrcat(plknastpod{2});
4605 men.nastpod(2) = uimenu (men.hlnast, "text", plk, "menuselectedfcn", ...
4606     "PrejezZprac (Cestafile, Jmfile, hgrf1, hgrf2, men, Velpis, ↵
4607     2)");
4608 plk = cstrcat(plknastpod{3});
4609 men.nastpod(3) = uimenu (men.hlnast, "text", plk, "menuselectedfcn", ...
4610     "PrejezZprac (Cestafile, Jmfile, hgrf1, hgrf2, men, Velpis, ↵
4611     3)");
4612 plk = cstrcat(plknastpod{4});
4613 men.nastpod(4) = uimenu (men.hlnast, "text", plk, "menuselectedfcn", ...
4614     "PrejezZprac (Cestafile, Jmfile, hgrf1, hgrf2, men, Velpis, ↵
4615     4)");
4616
4617
4618
4619
4620 plk = cstrcat("Choice of using Butterworth LP filter calculation and ↵
4621     its parameters");
4622 men.nastDPButt = uimenu (men.hlnast, "text", plk, "separator", "on", ...
4623     "menuselectedfcn", "NastavButt (Cestafile, Jmfile, hgrf1, ↵
4624     hgrf2, men, Velpis)");
4625
4626
4627
4628 plk = cstrcat("Switch [dB - V] for drawing Y-axis of graph of ↵
4629     harmonics: now it is ");
4630 if AmplHarmV == 0
4631     plk = cstrcat(plk, 'in [dB]');
4632 else
4633     plk = cstrcat(plk, 'in [V]');
4634 end
4635 men.nastpodharm = uimenu (men.hlnast, "text", plk, "separator", "on", ...
4636     "menuselectedfcn", "ZmenKreslHarm (Cestafile, Jmfile, hgrf1, ↵

```

```
hgrf2, men, Velpis)");
```

4636

4637

4638

```
plk = cstrcat("Change in number of plotted harmonics ");
```

```
plk = cstrcat(plk, '(now it is ', num2str(PocKresHarm, "%d"), ' i.e. up  
to ');
```

```
plk = cstrcat(plk, num2str(sigfig(Fvypis/1000,4)), ' kHz)');
```

```
men.nastPocHarm = uimenu (men.hlnast, "text", plk, ...
```

```
"menuselectedfcn", "ZmenPocKreslHarm (Cestafile, Jmfile,  
hgrf1, hgrf2, men, Velpis)");
```

4644

4645

```
plk = cstrcat("Change in last calculated harmonics ");
```

```
plk = cstrcat(plk, '(now it is ', num2str(PocPocHarm + 1, "%d"), '.  
i.e. up to ');
```

```
plk = cstrcat(plk, num2str(sigfig(FpocTHD/1000,4)), ' kHz)');
```

```
men.nastPocPocHarm = uimenu (men.hlnast, "text", plk, ...
```

```
"menuselectedfcn", "ZmenPocPocitHarm (Cestafile, Jmfile,  
hgrf1, hgrf2, men, Velpis)");
```

4651

4652

4653

4654

4655

4656

```
men.ovlpod1 = uimenu (men.zaklovl, "label", "Load input file", ...
```

```
"accelerator", "b", "menuselectedfcn",...
```

```
"[Cestafile, Jmfile, pocfile, hgrf1, hgrf2] = VypDFT  
(Cestafile, Jmfile, hgrf1, hgrf2, men, Velpis, pocfile,  
true, false, false)");
```

4660

```
#vono to de i bez parenta, třeba viz men.nastpod1
```

```
men.ovlpod2 = uimenu ("parent", men.zaklovl, "label", "Save as image  
with &lower resolution", ...
```

```
"accelerator", "l", "separator", "on" , "menuselectedfcn", ...
```

```
"[hgrf1, hgrf2] = UlozdofileMensi (Cestafile, Jmfile,  
hgrf1, hgrf2, men, Velpis)");
```

4665

```
men.ovlpod3 = uimenu ("parent", men.zaklovl, "label", "Save as image  
with &higher resolution", ...
```

```
"accelerator", "h", "menuselectedfcn", "[hgrf1, hgrf2] =
```

```
UlozdofileVetsi (Cestafile, Jmfile, hgrf1, hgrf2, men,  
Velpis)");
```

4668

```
men.ovlpod4 = uimenu ("parent", men.zaklovl, "label", "&Close this  
window", ...
```

```
"accelerator", "c", "separator", "on" , "menuselectedfcn",  
"close (hfig1)");
```

4671

4672

4673

```
men.editpod1 = uimenu (men.zakledit, "text", "&Grids in graphs",  
"accelerator", "g", ...
```

```
"menuselectedfcn", "[Jemriz] = FunNufmriz (Jemriz, hgrf1,  
hgrf2)");
```

```

4676
4677
4678
4679
4680
4681
4682
4683
4684 # a kontextové menu
4685
4686 # Vytvoření kontextového menu
4687 men.kontmenu.hlav = uicontextmenu(hfig1);
4688
4689 # Přidání položek do kontextového menu
4690 men.kontmenu.pod1 = uimenu(men.kontmenu.hlav, 'Label', 'Load input file', ...
4691     "menuselectedfcn", ...
4692     "[Cestafile, Jmfile, pocfile, hgrf1, hgrf2] = VypDFT (Cestafile, Jmfile, hgrf1, hgrf2, men, Velpis, pocfile, true, false, false)");
4693
4694 men.kontmenu.pod2 = uimenu(men.kontmenu.hlav, 'Label', 'Save as image with &lower resolution', ...
4695     "separator", "on", "menuselectedfcn", ...
4696     "[hgrf1, hgrf2] = UlozdofileMensi (Cestafile, Jmfile, hgrf1, hgrf2, men, Velpis)");
4697
4698 men.kontmenu.pod3 = uimenu(men.kontmenu.hlav, 'Label', 'Save as image with &higher resolution', ...
4699     "menuselectedfcn", ...
4700     "[hgrf1, hgrf2] = UlozdofileVetsi (Cestafile, Jmfile, hgrf1, hgrf2, men, Velpis)");
4701
4702
4703
4704 plk = cstrcat("Choice of using Butterworth LP filter calculation and its parameters");
4705 men.kontmenu.nastDPButt = uimenu (men.kontmenu.hlav, "text", plk, "separator", "on", ...
4706     "menuselectedfcn", "NastavButt (Cestafile, Jmfile, hgrf1, hgrf2, men, Velpis)");
4707
4708
4709
4710 plk = cstrcat("Switch [dB - V] for drawing Y-axis of graph of harmonics: now it is ");
4711 if AmplHarmV == 0
4712     plk = cstrcat(plk, 'in [dB]');
4713 else
4714     plk = cstrcat(plk, 'in [V]');
4715 end
4716 men.kontmenu.nastharm = uimenu (men.kontmenu.hlav, "text", plk, "separator", "on", ...
4717     "menuselectedfcn", "ZmenKreslHarm (Cestafile, Jmfile, hgrf1, hgrf2, men, Velpis)");
4718

```

```

4719
4720
4721 plk = cstrcat("Change in number of plotted harmonics ");
4722 plk = cstrcat(plk, '(now it is ', num2str(PocKresHarm, "%d"), ' i.e. up ↵
to ');
4723 plk = cstrcat(plk, num2str(sigfig(Fvypis/1000,4)), ' kHz)');
4724 men.kontmenu.nastPocHarm = uimenu (men.kontmenu.hlav, "text", plk, ...
4725 "menuselectedfcn", "ZmenPocKreslHarm (Cestafile, Jmfile, ↵
hgrf1, hgrf2, men, Velpis)");
4726
4727 plk = cstrcat("Change in last calculated harmonics ");
4728 plk = cstrcat(plk, '(now it is ', num2str(PocPocHarm + 1, "%d"), '. ↵
i.e. up to ');
4729 plk = cstrcat(plk, num2str(sigfig(FpocTHD/1000,4)), ' kHz)');
4730 men.kontmenu.nastPocPocHarm = uimenu (men.kontmenu.hlav, "text", plk, ...
4731 "menuselectedfcn", "ZmenPocPocitHarm (Cestafile, Jmfile, ↵
hgrf1, hgrf2, men, Velpis)");
4732
4733
4734
4735 men.kontmenu.ZavrWokno = uimenu("parent", men.kontmenu.hlav, 'Label', ↵
'&Close this window', ...
4736 "separator", "on" , "menuselectedfcn", "close (hfig1)");
4737
4738
4739 # Přiřazení kontextového menu k figúúře a ke FŠEM jejím onýšům
4740 set (findall(hfig1), "uicontextmenu", men.kontmenu.hlav);
4741
4742
4743
4744
4745
4746
4747
4748
4749
4750
4751
4752 #přednastavení menu pro zpracování dat z filé
4753 set(men.nastpod(ZpracDat), "checked", "on")
4754 if ~ VsechnyVybery #to je (if not VsechnyVybery)
4755 #znepřístupnění ostatních výběrů, kromě celého filé
4756 #tady taky, páč kdýš se obsazuje VsechnyVybery tak menu "men" ještě ↵
neexistuje
4757 set (men.nastpod(1), "text", cstrcat('NOT POSSIBLE - ', ↵
plknastpod{1}));
4758 set (men.nastpod(2), "text", cstrcat('NOT POSSIBLE - ', ↵
plknastpod{2}));
4759 set (men.nastpod(3), "text", cstrcat('NOT POSSIBLE - ', ↵
plknastpod{3}));
4760
4761 set (men.nastpod(1), "enable", 'off');
4762 set (men.nastpod(2), "enable", 'off');
4763 set (men.nastpod(3), "enable", 'off');
4764
4765 endif

```

```
4766
4767
4768
4769
4770 # následující poslouchač pomocí addlistener (od Duck.ia) nějak nefunguje
4771 # addlistener(hFig, 'CloseRequestFcn', @(src, event) closeFigure(src));
4772
4773 # Přidání poslouchače události zavření figure s hfig1, kvůli uložení ↵
nastavení
4774 # do filé k předávání proměnných viz plk ve funkci !!!!!!!
4775 set(hfig1, 'CloseRequestFcn', @(src, event) Figurasezavira(src));
4776
4777
4778
4779
4780
4781 # Aktivace omezení rozměrů figúry MUSÍ být až tady (dál od nastavení ↵
pozice
4782 # figúry podle minula), páč dyš je těsně za nastavením oné pozice tak
4783 # se program s takto nastavenou figúrou spustí, ALE při první změně
4784 # rozměrů ZDECHNE celá Oktáva !!!!!!!!!!!!!!! a čert ví proč...
4785 aktivuj_omez_rozm(hfig1)
4786
4787
4788
4789
4790
4791
4792
4793
4794 if (pocfile < 1); # znepřístupnění menu pro hotový výstup
4795 # (třeba editace mříšky) dyš poprvé nejni filé
4796
4797     set (men.editpod1, "enable", 'off');
4798     set (men.ovlpod2, "enable", 'off');
4799     set (men.ovlpod3, "enable", 'off');
4800     set (men.nastPocHarm, "enable", 'off');
4801     set (men.nastPocPocHarm, "enable", 'off');
4802
4803     #a z kontext. menu taky
4804     set (men.kontmenu.pod2, "enable", 'off');
4805     set (men.kontmenu.pod3, "enable", 'off');
4806     set (men.kontmenu.nastPocHarm, "enable", 'off');
4807     set (men.kontmenu.nastPocPocHarm, "enable", 'off');
4808
4809 endif
4810
4811
4812
4813 #konec ↵
menu-----
4814
4815
4816
4817
4818
```

4819
4820
4821
4822
4823
4824
4825
4826
4827
4828
4829
4830
4831
4832
4833