# A MATLAB quick start into using the Confined Gaussian Window

D. Hägele
*Arbeitsgruppe Spektroskopie der kondensierten Materie,*
*Ruhr-Universität Bochum, Universitätsstraße 150, D-44780 Bochum, Germany*
(Dated: September 15, 2015)

We provide easy to use MATLAB code (CGWindow.zip) for calculating the Confined Gaussian Window (CGW) and Approximate Confined Gaussian Window (ACGW) which posses an optimal RMS time-bandwidth product. A single command `w = replacewinbyCG(w);` replaces a suboptimal window `w` from existing code by the Confined Gaussian window. The new window conserves the RMS temporal width and the square norm w'*w of the initial window.

PACS numbers:

## REPLACING OLD WINDOWS

The Confined Gaussian Window (CGW) offers provably the best RMS time-bandwidth product of all window functions [1]. This property is especially attractive in FFT based time-frequency analysis, where both, temporal and spectral localization is important. The original 2014 paper included a number of MATLAB functions that in principle could be used to calculate the windows. However, the heavy use of global variables has made a direct integration of the functions in user-code highly unattractive.

The MATLAB functions in CGWindow.zip solve this problem by providing an easy quick start into getting the Confined Gaussian Window to work in existing code. Suppose you created a Truncated Gaussian Window by

```
w = gausswin(128,2.8);
```

you will obtain a Confined Gaussian Window `u` just from

```
u = replacewinbyCG(w);
```

where `u` and `w` will have the same RMS temporal width but with `u` having optimal RMS frequency width (use command `wvtool(u,w)` to compare the two). The square norm is also conserved by the new window `u`, i.e. `w'*w` and `u'*u` will give the same result. Conservation of the square norm is an extremely important property. The values of a power spectrum calculated with windows of different square norm would spoil some of the spectrum's properties. For instance, the area under the spectrum would vary proportional to the value of `u'*u`.

The command for the Approximate Confined Gaussian window (ACGW)

```
u = replacewinbyACG(w);
```

performs much faster as in contrast to the CGW no matrix diagonalization needs to be envied for its calculation.

Figure 1(a-f) show a number of common windows created by the corresponding MATLAB function in comparison with the CG and ACG replacement windows. The replacement windows for the triangular window shows both, an improved main-lobe width and stronger attenuation of the side lobes. In case of the Hamming window, the main lobe width and the height of the forth and higher side lobes is improved while side lobes one to three are better in the Hamming window. The Hann window shows basically the same main lobe but worse first and second side lobe while all other side lobes are better than that of the CGW and ACGW. However, suppression of higher side lobes by more than 35dB is overall very good. Surprisingly, the Blackman-Harris window is seemingly superior to the ACGW and CGW in all aspects. However, a magnification of the main lobe in the interval 0.014 to 0.02 (not shown) reveals that the ACGW and CGW main lobe is slightly more narrow than that of the Blackman-Harris window. This explains why the Confined Gaussian windows still show a superior RMS time-bandwidth product. Examples (e) and (f) are somewhat pathological. The window of the Tukey family with parameter value 0.5 exhibits a RMS temporal width that is larger than the sensible value of $\sigma = 0.181N$ discussed in [1]. The Cosine window has provably the best possible RMS frequency width and a $\sigma$ very close to $0.181N$, i.e. all windows with either wider or narrower RMS temporal width will have a worse RMS frequency width. The CGW family contains the Cosine window for the window parameter $\alpha = 0$. While the windows (a) to (d) were all replaced by CGWs with $\alpha > 0$, the Tukey window required a negative parameter $\alpha$. The case (f) of the rectangular window is even more pathological where $\alpha$ is more negative and the window shape shows a minimum in the middle and two maxima outside. Clearly, in case (f) replacement with a CGW is not recommended. Therefore, the function `u = replacewinbyCG(w)` returns the inital window `u` instead of a CGW in case `alpha` $< 0$.

## DIRECT CALCULATION OF THE CGW AND ACGW

The parameter $\alpha$ for the CGW is obtained form

```
[u,alpha] = replacewinbyCG(w);
```

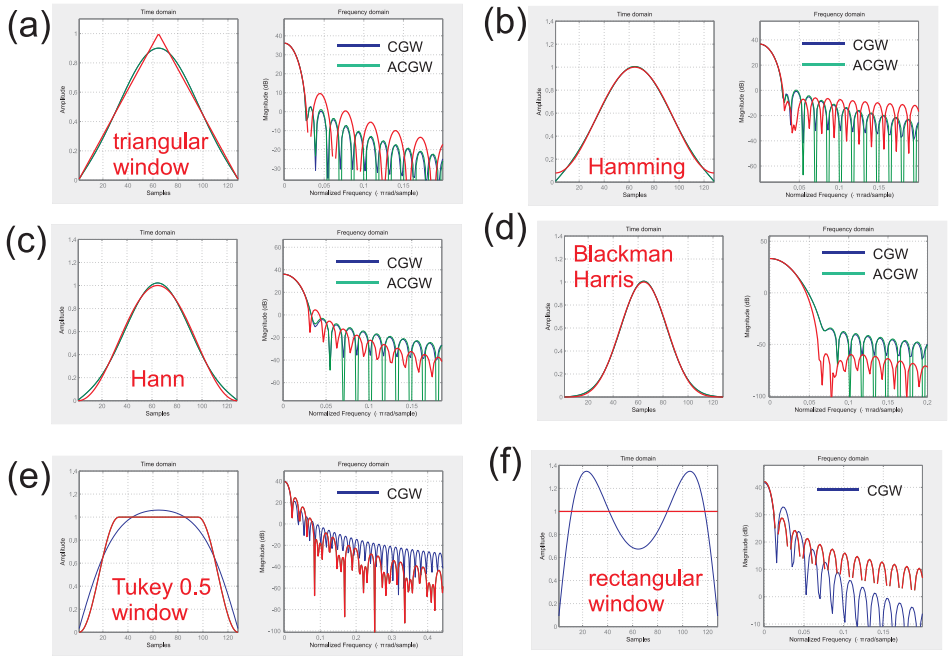


FIG. 1: Six popular windows of lengths 128 (red lines) in comparison with CGWs (blue lines) and ACGWs (green lines) obtained from the MATLAB functions `replacewinbyCG()` and `replacewinbyAC()`, respectively.

TABLE I: Parameters `alpha` for Confined Gaussian window.

| $L$ | triang | hamming | barthannwin | hann | parzenwin | nuttallwin | blackmanharris |
|---|---|---|---|---|---|---|---|
| 32 | 0.1157 | 0.1747 | 0.2378 | 0.2655 | 0.7834 | 1.0226 | 1.0785 |
| 64 | 0.1112 | 0.1561 | 0.2162 | 0.2429 | 0.7831 | 0.9589 | 1.0114 |
| 128 | 0.1087 | 0.1468 | 0.2055 | 0.2317 | 0.7830 | 0.9289 | 0.9798 |
| 256 | 0.1073 | 0.1421 | 0.2002 | 0.2262 | 0.7829 | 0.9143 | 0.9644 |
| 512 | 0.1066 | 0.1397 | 0.1975 | 0.2234 | 0.7828 | 0.9071 | 0.9568 |
| 1024 | 0.1063 | 0.1385 | 0.1962 | 0.2220 | 0.7828 | 0.9035 | 0.9531 |

where `alpha` is in units of $\bar{\alpha}$, see [1]. The commands

```
L = length(w);
u = Cgausswin(L, alpha);
```

yields then directly the CGW by a function analogous to the usual MATLAB window function commands (gausswin(L,alf), triang(L), ...). Please note that the parameter `alpha` of `Cgausswin()` and `alf` of `gausswin()` are two completely different quantities and must not be confused with each other! Similarly, the ACGW is obtained from

```
[u,alfACG] = replacewinbyACG(w);
L = length(w);
u = ACgausswin(L, alfACG);
```

Here, the parameter `alfACG` and `alf` are similar quantities as `alf` governs the width of the truncated Gausswindow and `alfACG` governs the widths of threes Gaussian functions whose (weighted) sum constitute the ACGW. Note, that in the original definition of the ACGW the parameter $s$ was proportional to the widths of the three Gaussians. The relation of $s$ and `alfACG` is `alfACG` $\propto s^{-2}$.

## PORTING CODE TO OTHER LANGUAGES

When porting the MATLAB code to other languages like Octave, SciLab, Fortran, C++, MATHEMATICA, MuPad, etc., the matrix diagonalization required in Cgausswin [and also `replacewinbyCG()` which calls `Cgausswin()`] may pose a problem in very basic work environments. In that case, `ACgausswin()` and `replacewinbyAC()` may alternatively be ported without the need for matrix diagonalization and, by the way, much faster running time. `ACgausswin()` calls the MATLAB function `findmin()` to perform a minimization with respect to one variable. If porting the minimization is still to much a hassle we may recommend to use values for `alfACG` from table II calculated for the most common windows. With these values, only the undemand-

TABLE II: Parameters `alfACG` for the Approximate Confined Gaussian window.

| $L$ | triang | hamming | barthannwin | hann | parzenwin | nuttallwin | blackmanharris |
|---|---|---|---|---|---|---|---|
| 32 | 2.0440 | 2.2408 | 2.4071 | 2.4708 | 3.2226 | 3.4443 | 3.4904 |
| 64 | 2.0782 | 2.2325 | 2.4000 | 2.4648 | 3.2746 | 3.4442 | 3.4904 |
| 128 | 2.0950 | 2.2277 | 2.3960 | 2.4615 | 3.3006 | 3.4442 | 3.4903 |
| 256 | 2.1033 | 2.2251 | 2.3939 | 2.4597 | 3.3136 | 3.4442 | 3.4903 |
| 512 | 2.1075 | 2.2237 | 2.3928 | 2.4588 | 3.3201 | 3.4441 | 3.4903 |
| 1024 | 2.1096 | 2.2231 | 2.3922 | 2.4584 | 3.3233 | 3.4441 | 3.4903 |

ing function `ACgausswin()` needs adaption to other programming languages. For the sake of completeness, the values `alpha` for the CGW are shown in table I. The very small dependence of `alpha` on the window length $L$ is a consequence of the scaling behavior of its unit $\bar{\alpha}$ with $L$ discussed in the appendix of [1].

## FUTURE OF CGWINDOW.ZIP

I am aware that the code is in no way close to being optimal in terms of performance and error handling. Nevertheless, it's something most of you will easily be able to work with. Feel free to improve the code, I will be happy to put new versions on suitable places in the internet. This short manual and the code itself were written in my precious spare time because window functions are no longer a research topic in my group. When you use the code in your own work, please cite the original publication [1]. Using the code is completely free of charge. If you like it, please consider donating to the bank account of my university work group. It was them who brought the Confined Gaussian window into being. We may in the future be able to fund bright students who can put more of our signal processing ideas into practise. IMPORTANT: include the exact reference number (as shown below), otherwise the money will end up at the university administration without a link to my group. The information below includes the BIC-Code which is necessary for international wire transfers.

| | |
|---|---|
| Account holder: | `Ruhr-Universitaet Bochum` |
| IBAN: | `DE53 43050001 000 1486828` |
| BIC-Code: | `WELADED1BOC` |
| reference: | `473 703 0011` |
| bank institute: | `Sparkasse Bochum, Germany.` |

[1] S. Starosielec and D. Hägele, "Discrete-time windows with minimal RMS bandwidth for given RMS temporal width," *Signal Processing (Elsevier)*, vol. 102, pp. 240, 2014.