

Jan 7, 2015

MATLAB PROGRAMS FOR CHEBYSHEV

In the two previous app note, AN-419 and AN420 [1, 2], we looked at the Chebyshev polynomials and the way they are used in filter design. Here we will be giving a Matlab program that does some additional calculations. Actually, it is quite simple in implementing the recursion relation:

$$C_N(\omega) = 2\omega C_{N-1}(\omega) - C_{N-2}(\omega) \quad (1)$$

Using the defining equation:

$$C_N(\omega) = \cos(N\cos^{-1}\omega) \quad \text{for } |\omega| \leq 1 \quad (2)$$

we get the first two C_N [equations (3) and (4)] and the recursion equation does the rest.

$$C_0(\omega) = \cos(0) = 1 \quad (3)$$

$$C_1(\omega) = \cos(\cos^{-1}(\omega)) = \omega \quad (4)$$

The Matlab program ***cheby*** that is below (page 2) has a number of functions which may be beyond anything anyone needs but does not hog many resources in doing more than asked. The most basic function is computing the Chebyshev polynomial coefficients for orders beyond those normally tabulated. As set up, the coefficients for the first 30 Chebyshev polynomials for orders 0 to 29 are calculated in a matrix C. This can be easily modified for larger orders if desired. We see from the table in [2] that the numerical values of the numbers were already getting quite large, and this was extended from published tables to order 12 by hand using equation (1). Because the coefficients get so large (above N=25), Matlab may print the results in exponential form. You can use ***format long*** to extend the retention of the full number of digits. So much for the matrix forming the extended table.

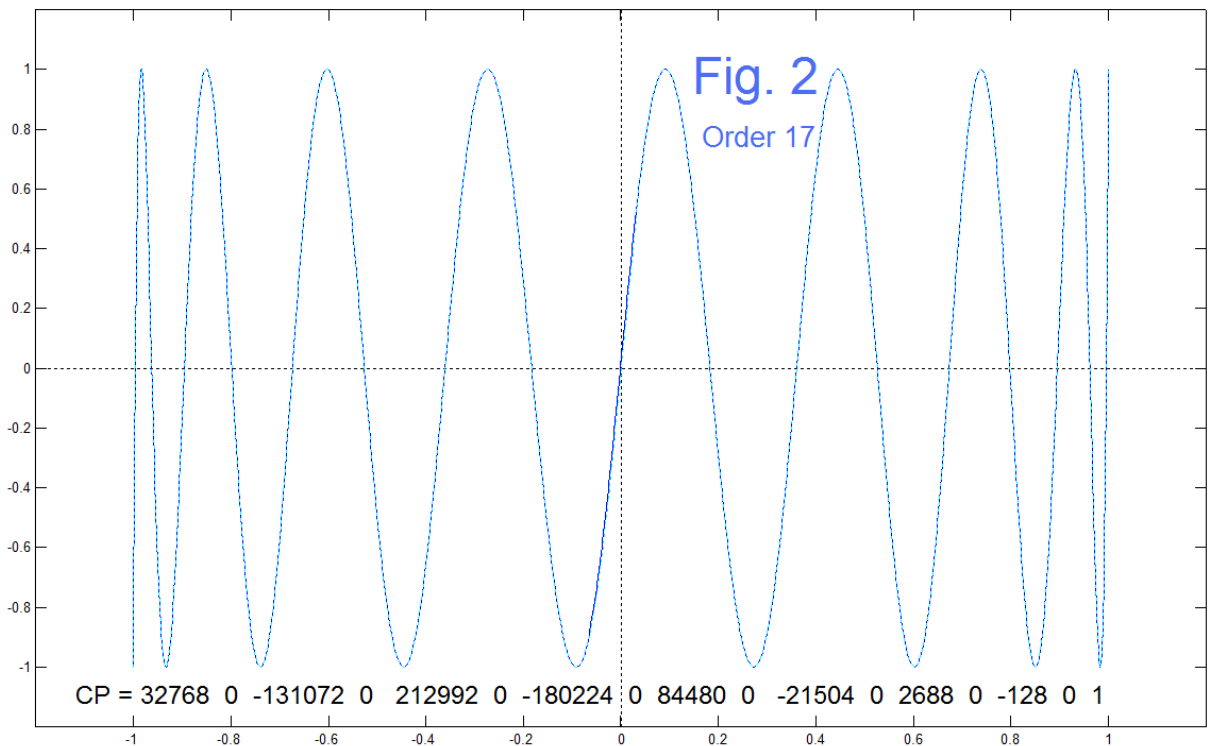
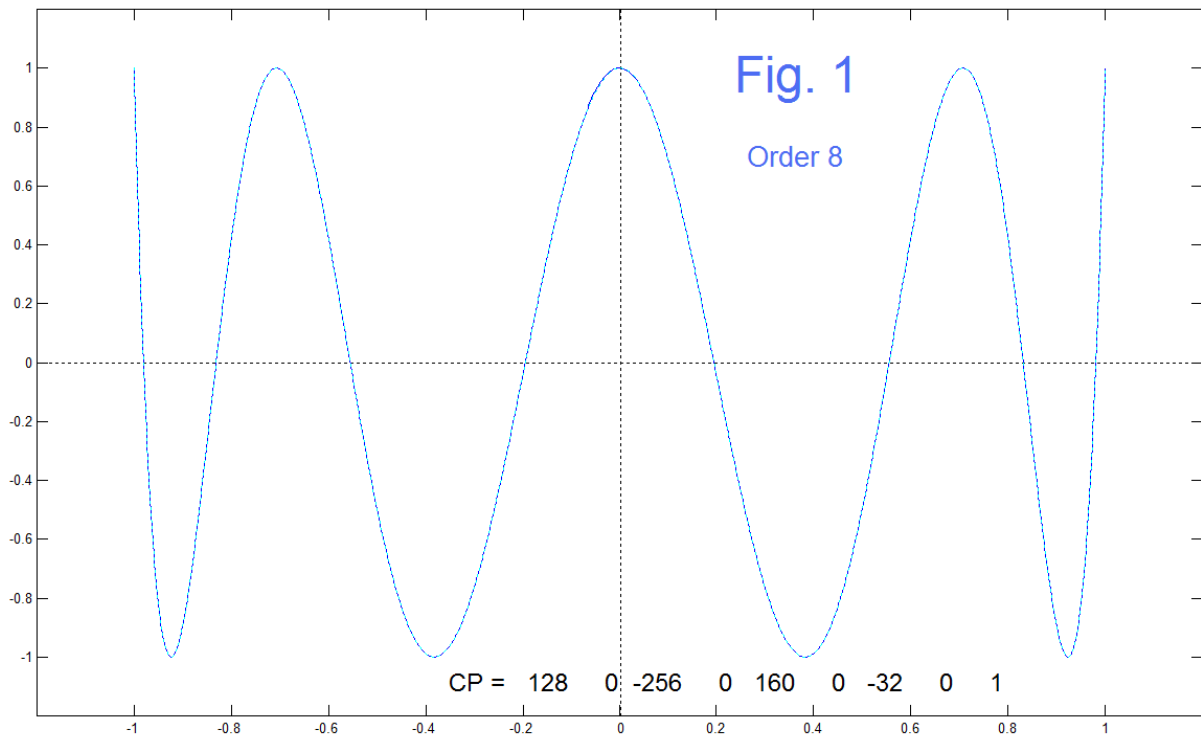
The user specifies a particular order N – here for N=0 to N=29. We have already calculated all the coefficients, and it takes so little work to get these all we do it every time even if we have low orders. The coefficients for the chosen order are printed, flipping left/right so that the highest order coefficient (N) is first. Thus we can print out any particular C_N . Note that every other coefficient is zero, but we print them anyway. Thus the first coefficient is for ω^N and the second is 0 and the third is for ω^{N-2} , and so on. This is the reversal of the matrix C – which was set up to simplify the recursion.

```

% [CN,C]=cheby(N)
%
% N = order of cheby poly
% CN = poly coefficients - Nth power on down
% C = matrix of first 30 polynomials
function [CN,C]=cheby(N)
C=zeros(30:30); % clear space for coefficients
R=[1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
C(1,:)=R; % poly C0
R=[0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
C(2,:)=R; % poly C1
% recursion fills matrix
for k = 3:30
R=C(k-1,:);
R=[0 2*R(1:29)]-C(k-2,:);
C(k,:)=R;
end
%
CS=C(1:26,1:26) % smaller matrix avoids exponential output
%
% Now concentrate on particular N
%
N
e=1;
if floor(N/2)==N/2; e=0; end
CP=C(N+1,:)
CP=fliplr(CP(e+1:N+1))
CN=zeros(1,2001); % vector of polynomial curve
w=-1:.001:1;
% Sum polynomial
for kk=1:N+1
    CN=CN+C(N+1,kk)*w.^(kk-1);
end
figure(1)
plot([0 0],[-1.2 1.2],'k:')
hold on
plot([-1.2 1.2],[0 0],'k:')
plot(w,CN)
% check by original calculation
CC=cos(N*acos(w));
plot(w,CC,'c:')
hold off
axis([-1.2 1.2 -1.2 1.2])
figure(1)

```

Besides calculating the coefficients, the program calculates and plots the polynomial on the interval -1 to +1. This range can be expanded to show the run-away on the ends if desired [2]. The polynomial curve is calculated in two ways. The evaluation of the polynomial is done directly (dark blue), and then using equation (2), light-blue dashed, as a check. A third way would be to use Matlab's *polyval*. Running the program a few times is probably enough to believe it all checks without doing it every time, but the plots are cheap. The figures show two examples.



Note that these plots show the traditional view of Chebyshev polynomials, the equiripple part defined on the interval -1 to +1. The ends rapidly rush off to infinity as we have noted [1,2]. We also note that very rapidly the polynomials become far too high an order to be useful for even digital realizability, let alone analog realizability. As

N	$C_N(\omega)$
0	1
1	ω
2	$2\omega^2 - 1$
3	$4\omega^3 - 3\omega$
4	$8\omega^4 - 8\omega^2 + 1$
5	$16\omega^5 - 20\omega^3 + 5\omega$
6	$32\omega^6 - 48\omega^4 + 18\omega^2 - 1$
7	$64\omega^7 - 112\omega^5 + 56\omega^3 - 7\omega$
8	$128\omega^8 - 256\omega^6 + 160\omega^4 - 32\omega^2 + 1$
9	$256\omega^9 - 576\omega^7 + 432\omega^5 - 120\omega^3 + 9\omega$
10	$512\omega^{10} - 1280\omega^8 + 1120\omega^6 - 400\omega^4 + 50\omega^2 - 1$
11	$1024\omega^{11} - 2816\omega^9 + 2816\omega^7 - 1232\omega^5 + 220\omega^3 - 11\omega$
12	$2048\omega^{12} - 6144\omega^{10} + 6912\omega^8 - 3584\omega^6 + 840\omega^4 - 72\omega^2 + 1$
13	$4096\omega^{13} - 13312\omega^{11} + 16640\omega^9 - 9984\omega^7 + 2912\omega^5 - 364\omega^3 + 13\omega$
14	$8192\omega^{14} - 28672\omega^{12} + 39424\omega^{10} - 26880\omega^8 + 9408\omega^6 - 1568\omega^4$ $+ 98\omega^2 - 1$
15	$16384\omega^{15} - 61440\omega^{13} + 92160\omega^{11} - 70400\omega^9 + 28800\omega^7 - 6048\omega^5$ $+ 560\omega^3 - 15\omega$
16	$32768\omega^{16} - 131072\omega^{14} + 212992\omega^{12} - 180224\omega^{10} + 84480\omega^8 - 21504\omega^6$ $+ 2688\omega^4 - 128\omega^2 + 1$
17	$65536\omega^{17} - 278528\omega^{15} + 487424\omega^{13} - 452608\omega^{11} + 239360\omega^9$ $- 71808\omega^7 + 11424\omega^5 - 816\omega^3 + 17\omega$
18	$131072\omega^{18} - 589824\omega^{16} + 1105920\omega^{14} - 1118208\omega^{12} + 658944\omega^{10}$ $- 228096\omega^8 + 44352\omega^6 - 4320\omega^4 + 162\omega^2 - 1$
19	$262144\omega^{19} - 1245184\omega^{17} + 2490368\omega^{15} - 2723840\omega^{13} + 1770496\omega^{11}$ $- 695552\omega^9 + 160512\omega^7 - 20064\omega^5 + 1140\omega^3 - 19\omega$
20	$524288\omega^{20} - 2621440\omega^{18} + 5570560\omega^{16} - 6553600\omega^{14} + 4659200\omega^{12}$ $- 2050048\omega^{10} + 549120\omega^8 - 84480\omega^6 + 6600\omega^4 - 200\omega^2 + 1$
21	$1048576\omega^{21} - 5505024\omega^{19} + 12386304\omega^{17} - 15597568\omega^{15}$ $+ 12042240\omega^{13} - 5870592\omega^{11} + 1793792\omega^9 - 329472\omega^7 + 33264\omega^5$ $- 1540\omega^3 + 21\omega$
22	$2097152\omega^{22} - 11534336\omega^{20} + 27394048\omega^{18} - 36765696\omega^{16}$ $+ 30638080\omega^{14} - 16400384\omega^{12} + 5637632\omega^{10} - 1208064\omega^8$ $+ 151008\omega^6 - 9680\omega^4 + 242\omega^2 - 1$
23	$4194304\omega^{23} - 24117248\omega^{21} + 60293120\omega^{19} - 85917696\omega^{17}$ $+ 76873728\omega^{15} - 44843008\omega^{13} + 17145856\omega^{11} - 4209920\omega^9$ $+ 631488\omega^7 - 52624\omega^5 + 2024\omega^3 - 23\omega$
24	$8388608\omega^{24} - 50331648\omega^{22} + 132120576\omega^{20} - 199229440\omega^{18}$ $+ 190513152\omega^{16} - 120324096\omega^{14} + 50692096\omega^{12} - 14057472\omega^{10}$ $+ 2471040\omega^8 - 256256\omega^6 + 13728\omega^4 - 288\omega^2 + 1$

such, what value here is it to have a program to generate them? Well – we are curious, and as such, we can expand our table from the 12 in AN-420 [2] to 24 here, mainly to make the point about how large the coefficients become. The larger table is above. Note that by the time we get to order 24, there is a 2-billion-to-one range in coefficient values. Here we have constructed the table by cut-and-paste of the numbers from Matlab to this text, so we expect we have avoided any transcription errors. A cute way to check the coefficients is to see if they must sum to +1. Equation (2) shows why this must be so. Further, we can check the table. Or, note the fact that the polynomial is equal to 1 when $\omega=1$, which is also suggested in the two figures, and in previous plots [2].

REFERENCES

[1] B. Hutchins, “Finding Chebyshev Polynomials,” Electronotes Application Note 419, Dec. 26, 2014

<http://electronotes.netfirms.com/AN419.pdf>

[2] B. Hutchins, “Chebyshev as a Filter Characteristic,” Electronotes Application Note No. 420, Dec 31, 2014

<http://electronotes.netfirms.com/AN420.pdf>