# fft_shift

May 18, 2018

```
In [1]: %plot -f png
        pkg load signal
```

```
In [2]: a = [1 2 3 4]
        a(1:2)
```

```
a =

   1   2   3   4

ans =

   1   2
```

```
In [226]: fftshift([1 2 3 4 5])
          fftshift([1 2 3 4 5 6])
```

```
ans =

   4   5   1   2   3

ans =

   4   5   6   1   2   3
```

```
In [3]: function ret = fftsft(array)

            siz = size(array)(2);
            len = size(array)(1);
            pos = floor(siz/2);
            ret = zeros(len,siz);

            ret(:,1:pos) = array(:,siz-pos+1:end);
            ret(:,pos+1:end) = array(:,1:siz-pos);
            tret = transpose(ret);
            siz = size(tret)(2);
            len = size(tret)(1);
            pos = floor(siz/2);
            ret = zeros(len,siz);
            ret(:,1:pos) = tret(:,siz-pos+1:end);
            ret(:,pos+1:end) = tret(:,1:siz-pos);

            ret = transpose(ret);


        endfunction

        fftsft([1 2 3 4 5 6; 7 8 9 10 11 12])
        fftsft([1 2 3 4 5 6])
```

```
ans =

   10   11   12    7    8    9
    4    5    6    1    2    3

ans =

   4   5   6   1   2   3
```

```
In [4]: fftsft([1 2 3 4 5 6; 7 8 9 10 11 12])
        t = fftshift([1 2 3 4 5 6; 7 8 9 10 11 12])
```

```
ans =

   10   11   12    7    8    9
    4    5    6    1    2    3

t =

   10   11   12    7    8    9
    4    5    6    1    2    3
```

```
In [5]: function ret = logfft(t,c=0.1)
            t = fftsft(t);
            ret = log(fft(t).+ c);

        endfunction
```

```
In [6]: logfft(t)

ans =

 Columns 1 through 4:

   2.0919 + 0.0000i   2.3125 + 0.0000i   2.4932 + 0.0000i   2.6462 + 0.0000i
   1.7750 + 3.1416i   1.7750 + 3.1416i   1.7750 + 3.1416i   1.7750 + 3.1416i

 Columns 5 and 6:

   2.7788 + 0.0000i   2.8959 + 0.0000i
   1.7750 + 3.1416i   1.7750 + 3.1416i
```

```
In [7]: function ret = medflt(m)
            A = reshape(m,1,9);
            A = sort(A);
            len = length(A);
            %median in case M is not odd
            %ret = (A(floor((len-1)/2)) + A(ceil((len-1)/2)))/2;
            ret = A(5);
        endfunction


        medflt([1 1 8 ;9 1 2;3 4 3 ])

ans =  3
```

```
In [10]: function ret = saltnoise(A,p)

             ret = rand(size(A));

             %% Make black with whites
             ret1 = arrayfun(@(x) 1-le(x,p/2),ret);

             %% Make white with blacks
             ret2 = arrayfun(@(x) 1-ge(x,1-p/2),ret);

             %% Invert black array
             MAX_COLOR = 255;
             ret3 = (ones(size(A)).-ret2) * MAX_COLOR;
             ret = A.*ret1.*ret2.+ret3;
         endfunction

         saltnoise([3,3,3,1;1,2,6,7],0.2)
```

```
ans =

     3     3     3     1
   255     0     6     7
```

```
In [11]: A = [1 1 8 ;9 1 2;3 4 3 ]
         p = 0.1
         arrayfun(@(x)x-x*le(x,p*10),A)
```

```
A =

   1   1   8
   9   1   2
   3   4   3

p =  0.10000
ans =

   0   0   8
   9   0   2
   3   4   3
```

```
In [21]: function ret = addgaussnoise(A,m,v)

             noise = randn(size(A))*sqrt(v) + m;
             ret = A + noise;
         endfunction

         addgaussnoise(A,1,4)
```

```
ans =

    0.99191    2.84580    8.75789
   10.73025    0.71578    7.83942
    2.27815    6.34115    2.48240
```

```
In [22]: function ret = dct1d(x)
             N = length(x);
             y = x + x(end:-1:1);

             Y = fft(x);
             k = 1:N;
             C = e.^(-j*pi/2/N*k).*Y(k);
             %ret = [C zeros(1,N)];
             ret = C;
         endfunction

         dct1d([1,2,4,1,2,6,7])
         length(dct1d([1,2,4,1,2,6,7]))
```

```
ans =

 Columns 1 through 3:

   22.42334 -   5.11798i    4.24698 +   4.93935i   -4.35482 +   7.59783i

 Columns 4 through 6:

    0.27479 +   2.19064i   -2.09293 +   0.70291i   -4.95593 +   7.22013i

 Column 7:

   -6.29290 -   1.68329i

ans =  7
```

```
In [23]: function ret = dct2d(x)

             %%DCT per row
             M = num2cell(x,2);
             C = cellfun(@dct1d,M,"UniformOutput", false);
             B = cell2mat(C);

             %%DCT per collumn
             M = num2cell(B,1);
             %Transpose to make rows collums and back
             C = cellfun(@transpose,M,"UniformOutput", false);
             C = cellfun(@dct1d,C,"UniformOutput", false);
             C = cellfun(@transpose,C,"UniformOutput", false);

             %return Matrix
             ret = cell2mat(C);
         endfunction

         dct2d(([1,1,1,1;2,2,2,2;3,3,3,3]))
         dct2d(transpose([1,1,1,1;2,2,2,2;3,3,3,3]))
```

```
ans =
 Columns 1 through 3:

   14.61027 - 19.04048i    0.00000 +  0.00000i    0.00000 +  0.00000i
    2.65131 +  6.40083i    0.00000 +  0.00000i    0.00000 +  0.00000i
   -0.90431 +  6.86893i    0.00000 +  0.00000i    0.00000 +  0.00000i

 Column 4:

    0.00000 +  0.00000i
    0.00000 +  0.00000i
    0.00000 +  0.00000i

ans =

   14.61027 - 19.04048i    2.65131 +  6.40083i   -0.90431 +  6.86893i
    0.00000 +  0.00000i    0.00000 +  0.00000i    0.00000 +  0.00000i
    0.00000 +  0.00000i    0.00000 +  0.00000i    0.00000 +  0.00000i
    0.00000 +  0.00000i    0.00000 +  0.00000i    0.00000 +  0.00000i
```

```
In [ ]:
```