Homework #5, EECS 556, W21. Due **Thu. Mar. 04**, by 9:00AM

Skills and Concepts

- DFS, DFT, FFT, DSFT, FT, DCT

Problems

1. [10] Two real $M \times N$ images $x[m,n]$ and $h[m,n]$ are combined to form a complex image via

$$y[m,n] \triangleq x[m,n] + \imath\, h[m,n],$$

and the result is passed to a **FFT** routine yielding complex **DFT** coefficients $Y[k,l]$.
Find expressions for the DFT coefficients $X[k,l]$ and $H[k,l]$ in terms of the $Y[k,l]$ values.
Hint: $x[m,n] = \text{real}\{y[m,n]\} = \frac{1}{2}\,(y[m,n] + y^*[m,n])$.
These relationships are at the heart of how one can compute two DFTs (of real images) using one FFT call.

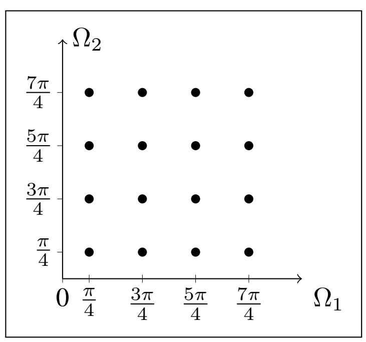
2. [20] A $M \times N$ image $x[m,n]$ is **up-sampled** to form a $2M \times 2N$ image $y[m,n]$.
Relate the $(2M, 2N)$-point **DFT** $Y[k,l]$ to the (M,N)-point DFT $X[k,l]$ for the following two cases.
(a) [10] The up-sampling uses zero insertion, *i.e.*, [1 2 3] $\to$ [1 0 2 0 3 0] in 1D.
(b) [10] The up-sampling is done by replacing each pixel value with a 2×2 block of pixels with the same value.
(The analogous operation in 1D is [1 2 3] $\to$ [1 1 2 2 3 3].)

3. [5] We want to convolve a 512×512 image $x[m,n]$ with a filter $h[m,n]$ that is nonzero only for $0 \le m \le 10$ and $0 \le n \le 20$. Let signals $y[m,n]$ and $w[m,n]$ be defined by

$$y[m,n] = h[m,n] ** x[m,n]$$
$$w[m,n] = \text{IDFT}\left(\text{DFT}\left(h[m,n]\right) \;.\!\star\; \text{DFT}\left(x[m,n]\right)\right),$$

where **DFT** and IDFT are 512×512 points in size. For what values of $[m,n]$ does $w[m,n]$ always equal $y[m,n]$?

4. [10] We are given a 4×4 image $g[m,n]$ that is zero except for $0 \le n \le 3$ and $0 \le m \le 3$. Suppose that we would like to evaluate the **DSFT** $G(\Omega_1, \Omega_2)$ of $g[m,n]$ at the following 16 points in the (Ω_1, Ω_2) plane.



We would like to do this efficiently, ultimately by using a 2D **FFT**. Explain how you could use a single 4×4-point DFT (along with other elementary operations) to compute the 16 desired values of $G(\Omega_1, \Omega_2)$.

5. [25] This problem explores how to use the **DFT** (and thus the **FFT**) to compute the **DCT** efficiently (in 1D for simplicity). If $x[n]$ is a N-point sequence with support $0, \ldots, N-1$, then we define the $2N$-point sequence $y[n] = \begin{cases} x[n], & n = 0, \ldots, N-1 \\ x[2N-1-n] & n = N, \ldots, 2N-1. \end{cases}$
You will show that the $2N$-point DFT of $y[n]$ can be computed by one N-point DFT.
(a) [0] First define two N-point sequences $v[n]$ and $w[n]$ in terms of $y[n]$ as follows:

$$v[n] \triangleq y[2n], \qquad n = 0, \ldots, N-1$$
$$w[n] \triangleq y[2n+1], \quad n = 0, \ldots, N-1.$$

For the example $x[n] = [\underline{0}\ 1\ 2\ 3]$, sketch $y[n]$, $v[n]$, and $w[n]$ to illustrate that $w[n] = v[N-1-n]$ for $n = 0, \ldots, N-1$.
(b) [10] Let $Y[k]$ denote the $2N$-point DFT of $y[n]$, and define $W_N = \mathrm{e}^{-\imath 2\pi/N}$. Show that

$$Y[k] = \sum_{n=0}^{N-1} v[n]\, W_N^{kn} + W_{2N}^{k} \sum_{n=0}^{N-1} w[n]\, W_N^{kn}, \qquad k = 0, \ldots, 2N-1.$$

(c) [5] Combine (a) and (b) to show that

$$Y[k] = \sum_{n=0}^{N-1} v[n]\, W_N^{kn} + W_{2N}^{-k} \sum_{n=0}^{N-1} v[n]\, W_N^{-kn}, \qquad k = 0, \ldots, 2N-1.$$

(d) [10] Let $V[k]$ denote the N-point DFT of $v[n]$. Express $Y[k]$ for $k = 0, \ldots, N-1$ in terms of $V[k]$.
This shows that we can find the values of $Y[k]$ for $k = 0, \ldots, N-1$ with one N-point DFT. (The remaining $Y[k]$ values are not needed for a N-point DCT.)

6. [100] Consider an imaging system with frequency response $H(\rho) = \exp\left(-\pi(\rho/16)^2\right) - \exp\left(-\pi(\rho/4)^2\right)$. We would like to determine what the output image would be if the input image were $f(x, y) = \mathrm{rect}_2(x/3, 3y) + \mathrm{rect}_2(2x, y)$. One fairly painful way to do this would be to use convolution via integration. For an easier method, use JULIA's `FFTW.fft` or MATLAB's `fft2` command to compute the output image $g(x, y)$ at suitable (x, y) locations, using the convolution property that $g(x, y) = h(x, y) ** f(x, y) \overset{\mathcal{F}_2}{\longleftrightarrow} G(\nu_X, \nu_Y) = H(\nu_X, \nu_Y)\, F(\nu_X, \nu_Y)$.
(a) [20] Display (sampled versions of) $f(x, y)$ and $H(\nu_X, \nu_Y)$ as grayscale images. Make sure that you choose sampling parameters that are reasonable for both domains. Include a `colorbar` for *every* image you make in this class.
(b) [50] Display the real part and the imaginary part of your $g(x, y)$ result as two distinct images using subplots.
(c) [20] Also show $|F(\nu_X, \nu_Y)|$, $\mathrm{real}\{F(\nu_X, \nu_Y)\}$, and $\mathrm{imag}\{F(\nu_X, \nu_Y)\}$ using subplots.
(d) [10] Is your resulting $g(x, y)$ exact or approximate at the sample locations?
Caution: you must be very careful with `fftshift` and your sample locations to get a correct answer.
To work on this problem, download the template file `conv_via_fft_template.{jl,m}` from the web site.

7. [10] The image used in Fig. 1a of this compressed sensing MRI paper originated from a radiology web site.
Download the image from
`http://web.eecs.umich.edu/~fessler/course/556/r/ravishankar-11-mir,fig1a,t2axialbrain-09.jpg`
Load it into JULIA or MATLAB and examine its 2D magnitude spectrum (using an **FFT**) on a log-scale.
By what factor could you compress this image in a lossless (or nearly lossless) way using the spectral properties of this particular image?

8. [10] Please complete the anonymous midterm course evaluation online by the deadline on **Mar. 02, 11:59PM**. The "Teaching Evaluations" link on Canvas takes you take to `https://umich.bluera.com/umich`. Write a note in your gradescope solutions attesting that you completed it. (The Honor Code applies.) This is the first time the course is being taught online so your feedback is especially important to me.

Optional problems

9. [0] Consider the signal $g[m,n]$ with spectrum $G(\Omega_1,\Omega_2) = \frac{1}{1-0.5\,e^{-\imath\Omega_1}} \frac{1}{1-0.25\,e^{-\imath\Omega_2}}$.
   (a) Find $g[m,n]$ and $g[0,0]$.
   (b) Suppose the inverse DFT is applied to samples of $G(\Omega_1,\Omega_2)$ at $N \times M$ points over the $[-\pi,\pi) \times [-\pi,\pi)$ box. Find *analytically* the computed value at $[0,0]$.
   (c) For $N = M = 8$, will the computed value be within 0.5% of the correct value?

10. [0] State and prove duality properties for the DFS and for the DFT.
Hint: consider what happens if you take the DFT of an image, and then take the DFT again of the result.

11. [0] The notes for Ch. 5 have two different definitions of **strong-sense circular symmetric** (**SSCS**) DS functions.
Def. 1: $g[m,n]$ is SSCS iff there exists a function $g_R(r)$ defined on $[0,\infty)$ whose Hankel transform is zero for $\rho > 1/2$ for which $g[m,n] = g_R\left(\sqrt{m^2+n^2}\right)$.
Def. 2: $g[m,n]$ is SSCS iff for all θ and for all $m,n \in \mathbb{Z}$:

$$g[m,n] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} g[k,l] \operatorname{jinc}\left(\sqrt{(x-k)^2+(y-l)^2}\right)\Bigg|_{\substack{x = m\cos\theta + n\sin\theta \\ y = -m\sin\theta + n\cos\theta}}$$

$$= \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} g[k,l] \operatorname{jinc}\left(\sqrt{(m\cos\theta + n\sin\theta - k)^2 + (-m\sin\theta + n\cos\theta - l)^2}\right).$$

Prove that Def. 1 and Def. 2 are equivalent.

The DFT in the context of continuous-space problems

DSP courses typically present the DFT as a transform for discrete-time signals. But the DFT is also useful as a computational replacement for the 2D FT in continuous-space. Suppose we would like to visualize the image $g_a(x, y)$ that has the spectrum

$$G_a(\nu_X, \nu_Y) = e^{-4|(\nu_X+\nu_Y)^2-0.5|} \, e^{-3|\nu_X-\nu_Y|} \, e^{-\imath 2\pi 4 \nu_X} .$$

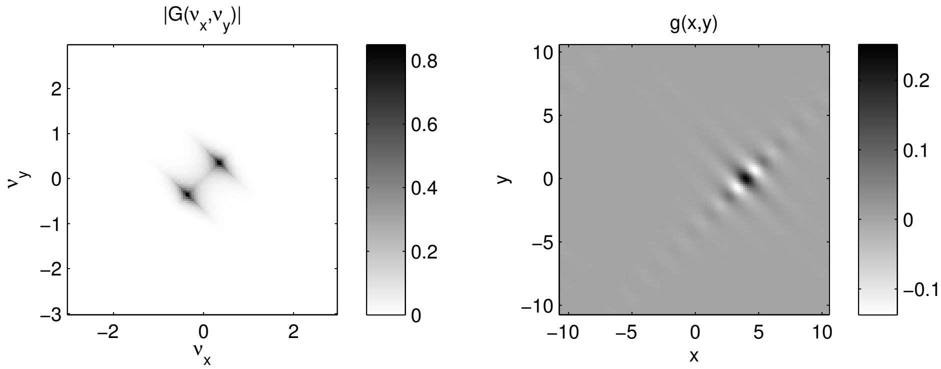
We could attempt to find the inverse 2D FT analytically via integration, but that would be painful. A faster and easier approach is just to sample $G_a(\nu_X, \nu_Y)$, take the inverse 2D FFT, and display the results. The subtle parts about doing this are choosing the samples reasonably (because this $g_a(x, y)$ is neither space limited nor band-limited), figuring out `fftshift`, and making sure to get all the dimensions correct for the axes labels. Note from the lecture notes that $\Delta_\nu \Delta_X = 1/N$.

The key relationship being used here is the following important approximation from the lecture notes:

$$G[k, l] \approx \frac{1}{\Delta_X} \frac{1}{\Delta_Y} G_a\left(\frac{k}{N\Delta_X}, \frac{l}{M\Delta_Y}\right), \quad -\frac{M}{2} \le k \le \frac{M}{2} - 1, \quad -\frac{N}{2} \le l \le \frac{N}{2} - 1,$$

where $G[k, l]$ is the 2D DFT of appropriate samples of $g_a(x, y)$. The code and figure below illustrates a complete example.

```
# ft_example1.m    Illustrate approximate inverse 2D FT via FFT
using MIRT: jim
using Plots
using Plots.PlotMeasures; default(top_margin=-50px, bottom_margin=0px) # kludge
using LaTeXStrings
using FFTW: ifft, fftshift, ifftshift

nu = 128; nv = nu
du = 6/nu; dv = 6/nv
u = (-nu/2:nu/2-1) * du # this "integer" way is safer than LinRange!
v = (-nv/2:nv/2-1) * dv # frequency domain sample locations
Guv = (u,v) -> exp(-4*abs((u+v)^2-0.5)) * exp(-3*abs(u-v)) * exp(-2im*π*4*u)
G = Guv.(u, v') # broadcast

p1 = jim(u, v, abs.(G), title=L"|G(\nu_x,\nu_y)|",
    xlabel=L"\nu_X", ylabel=L"\nu_Y")

nx = nu; ny = nv
dx = 1/(nx*du); dy = 1/(ny*dv)
x = (-nx/2:nx/2-1) * dx # note these object domain sample locations!
y = (-ny/2:ny/2-1) * dy
g = fftshift(ifft(ifftshift(G/dx/dy))) # dx dy essential to be quantitative

@show frac = maximum(abs.(imag.(g))) / maximum(abs.(g))
frac > 1e-7 && throw("non-negligible imaginary part")
g = real.(g) # discard any (almost) negligible imaginary part

p2 = jim(x, y, g, title=L"g(x,y)", xlabel=L"x", ylabel=L"y")
plot(p1, p2)
#savefig("ft_example1.pdf')
```




```
% ft_example1.m Illustrate approximate inverse 2D FT via FFT

nu = 128;       nv = nu;
du = 6/nu;      dv = 6/nv;
u = [-nu/2:nu/2-1] * du; % this "integer" way is safer than linspace!
v = [-nv/2:nv/2-1] * dv;
[uu vv] = ndgrid(u,v); % grid of frequency domain sample locations
G = exp(-4*abs((uu+vv).^2-0.5)) .* exp(-3*abs(uu-vv)) .* exp(-2i*pi*4*uu);

clf, subplot(221), imagesc(u, v, abs(G)'), axis image, axis xy
xlabel \nu_x, ylabel \nu_y, title |G(\nu_x,\nu_y)|, colormap(1-gray(256)), colorbar

nx = nu;        ny = nv;
dx = 1/(nx*du); dy = 1/(ny*dv);
x = [-nx/2:nx/2-1] * dx; % note these object domain sample locations!
y = [-ny/2:ny/2-1] * dy;
g = fftshift(ifft2(ifftshift(G/dx/dy))); % dx dy essential to be quantitative

frac = max(abs(imag(g(:)))) / max(abs(g(:)))
if frac > 1e-7, error 'non-negligible imaginary part', end
g = real(g); % discard (almost) negligible imaginary part

subplot(222), imagesc(x,y,g'), axis image, axis xy
xlabel x, ylabel y, title g(x,y), colorbar
%print('ft_example1', '-deps')
```

Caution: a key difference between the JULIA and MATLAB versions is that in JULIA we use `ifft` which performs an appropriate inverse 2D FFT when the input is a 2D array, whereas in MATLAB one must use `ifft2`, otherwise the result will be incorrect.