Midterm Project

Name: Jiping Zhang
StudentID: 020082910066
all the code can be downloaded from https://github.com/zhangjipinggom/MidtermNeuroimaging.

1. Create 10000 different arrays of 1000 normally-distributed random numbers with a mean of 10 and a standard deviation of 2 (be sure to type a semi-colon at the end of the line or your screen will become cluttered with random numbers), Named as matrix NA. Create another array at the same size, NB, with normally distributed numbers having a mean of 10.5 and a standard deviation of 2. Find the mean and standard deviation of each row in NA and NB (10000 rows in all), named as NAM, NBM, NAstd and NBstd. Plot and send me the histograms of NAM, NBM, NAstd and NBstd. Notice that there is a range of both standard deviations and means. Calculate 10000 t statistics of the difference of means between NA and NB (saved as array tN). Plot and send me the histogram of tN. What is the mean of your experimentally determined tN? What is the standard deviation of tN? Approximately what fraction of the time would your experimentally measured t statistic be greater than 7? (25points)
   Hint: Programming with MATLAB.
   Answer:

   1. Plot the histograms of NAM, NBM,NAstd and NBstd.

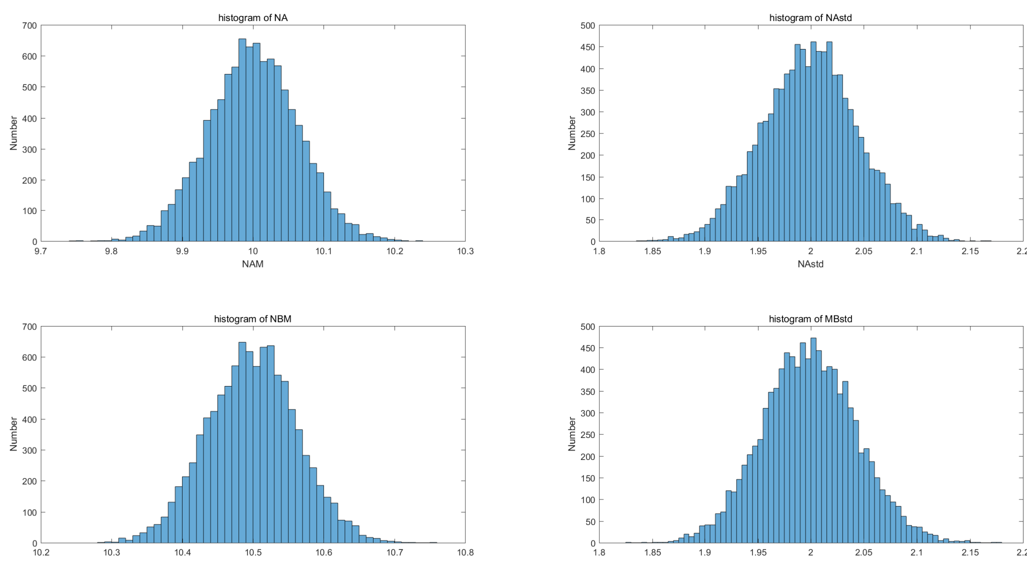

Figure 1: Histograms of NAM, NBM,NAstd and NBstd

2. Plot the histogram of tN

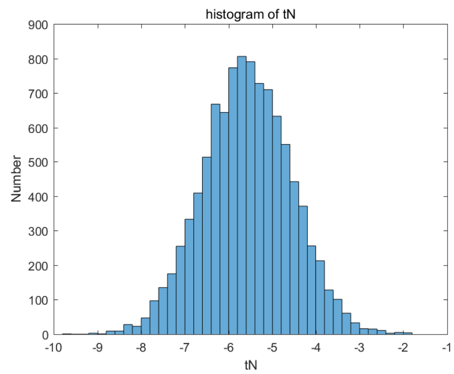

Figure 2: Histogram of tN

3. Attributes about tN
   Mean: 5.60
   std: 1.00
   the ratio of t statistic greater than 7: 0.08

Code:

```
NB = normrnd(10.5,2,[10000,1000]);
NAM = mean(NA, 2);
NBM = mean(NB, 2);
NAstd = std(NA, 0, 2);
NBstd = std(NB, 0, 2);
[h,p,ci,tN_tmp] = ttest2(NB',NA');
tN = tN_tmp.tstat;
tNM = mean(tN);
tNstd = std(tN);
tN_7_ratio = sum((tN > 7))/size(tN, 2);

grid
subplot(2, 2, 1), histogram(NAM)
xlabel('NAM'), ylabel('Number')
title('histogram of NA')
subplot(2, 2, 2), histogram(NAstd)
xlabel('NAstd'), ylabel('Number')
title('histogram of NAstd')
subplot(2, 2, 3), histogram(NBM)
xlabel('NBM'), ylabel('Number')
title('histogram of NBM')
subplot(2, 2, 4), histogram(NBstd)
xlabel('MBstd'), ylabel('Number')
title('histogram of MBstd')

```

```
figure(2)
grid
histogram(tN')
xlabel('tN'), ylabel('Number')
title('histogram of tN')
```

2. Construct a synthesized magnetic resonance spectroscopy (MRS) time series using the Lorentzian model:

$$y_n = \sum_{k=1}^{K} a_k e^{j\phi_k} e^{(-d_k + j2\pi f_k)t_n} + e_n, \quad t = 0, 1..., N-1 \tag{1}$$

where y_n is the nth measured data point, (a_k, ϕ_k, d_k, f_k) are the amplitude, phase, damping factor and frequency of the kth component, $t_n = 1s$ as the data sampling interval, and e_n is the circular complex white Gaussian noise. Here the data length is assumed as N=1024. The metabolite component parameters were assigned as in Table 1.
Using various noise level $\sigma^2 = 2$ and 5, respectively: Plot the corresponding absolute

Table 1: Parameters for MRS signal construction

k	Metablolite	f_k(Hz)	d_k(Hz)	a_k(a.u)	ϕ_k(rad)
1	NAA	0.8285	0.025	10.3	0
2	Cr	0.8925	0.02	4.8	π
3	Cho	0.9053	0.015	3.2	$\pi/2$
4	MI	0.9232	0.015	1.5	0
5	Lipid	0.7504	0.01	0.8	$\pi/6$

value of MRS signal in time domain. Perform a fast Fourier transform (FFT) to obtain the frequency domain representation of MRS signal (i.e. the spectrum). Plot the absolute value, real value and imaginary value of MRS spectrum separately. Send me the codes and plots. (35points)
Hint: Programming with MATLAB. Might use "fft" function for transformation.
Answer:
Figure 3 (in the next page) shows the absolute value of MRS signal in time domain, and the absolute value, real value and imaginary value of MRS spectrum, when the circular complex white Gaussian noise $\sigma^2 = 2$ and $\sigma^2 = 5$.

Code
code in Lorentzian function:

```
function Y = Q2Lorentzian(a, phi, d, f)
% return the time serious shape as [1x1024]
t_n = 1;
N = 1024;
T_n = 1:t_n:N;
amp = exp((-d+1i*2*pi*f)*T_n);
Y = a*exp(phi*1i)*amp;
```

Code in main function

```
variance = [2, 5];
L = 1024;
Fs = 1;
```

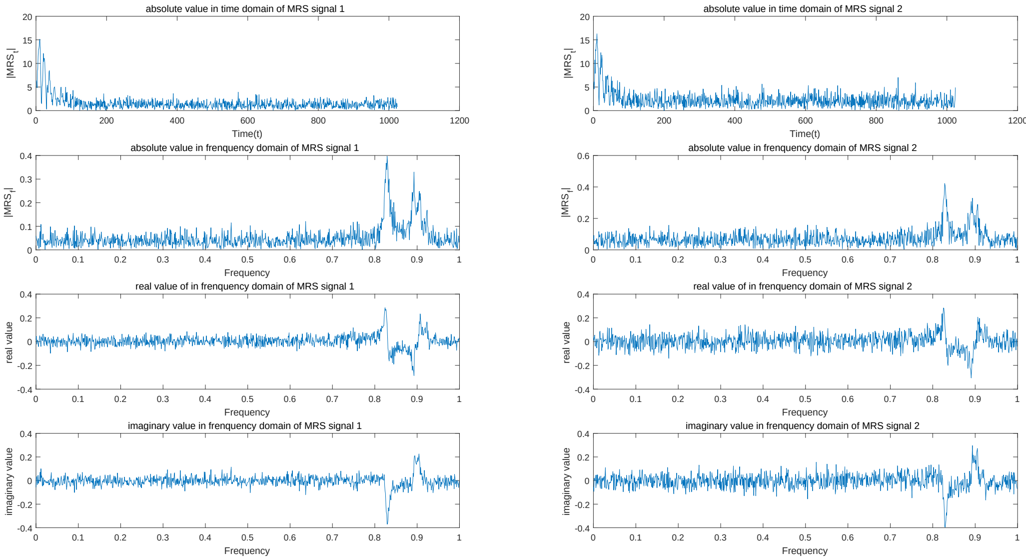

Figure 3: Absolute value of MRS signal in time domain, absolute value, real value and imaginary value of MRS spectrum separately from the 1st row to the 4th row. The 1st is when $\sigma^2 = 2$ and 2nd is when $\sigma^2 = 5$.

```
f = Fs*(0:L-1)/L;

y_all = [[zeros(1, 1024), zeros(1, 1024)]; [zeros(1, 1024), ...
    zeros(1, 1024)]];
for i_fig = 1:2
    E_n = sqrt(variance(i_fig) / 2) * (randn(L, 1) + (1i * randn(L, ...
        1)));
    NAA_t = Q2Lorentzian(10.3, 0, 0.025, 0.8285);
    Cr_t = Q2Lorentzian(4.8, pi, 0.02, 0.8925);
    Cho_t = Q2Lorentzian(3.2, pi/2, 0.015, 0.9053);
    MI_t = Q2Lorentzian(1.5, 0, 0.015, 0.9232);
    Lipid_t = Q2Lorentzian(0.8, pi/6, 0.01, 0.7504);
    MRS_t = NAA_t + Cr_t + Cho_t + MI_t + Lipid_t + E_n';
    MRS_f = fft(MRS_t)/L;
    y_all(i_fig, :) = [MRS_t, MRS_f];
end

grid
for i_fig = 1: 2
    subplot(4, 2, 1+floor(i_fig/2)), plot(abs(y_all(i_fig, 1: 1024)))
    xlabel('Time(t)'), ylabel('|MRS_t|')
    title('absolute value in time domain of MRS signal ' + ...
        string(i_fig)')
    subplot(4, 2, 3+floor(i_fig/2)), plot(f, abs(y_all(i_fig, ...
        1025:2048)))
    xlabel('Frequency'), ylabel('|MRS_f|')
    title('absolute value in frenquency domain of MRS signal ' + ...
        string(i_fig))
```

```
    subplot(4, 2, 5+floor(i_fig/2)), plot(f, real(y_all(i_fig, ...
        1025:2048)))
    xlabel('Frequency'), ylabel('real value')
    title('real value of in frenquency domain of MRS signal ' + ...
        string(i_fig))
    subplot(4, 2, 7+floor(i_fig/2)), plot(f, imag(y_all(i_fig, ...
        1025:2048)))
    xlabel('Frequency'), ylabel('imaginary value')
    title('imaginary value in frenquency domain of MRS signal ' + ...
        string(i_fig))
end
```

3. Given the raw speckle image files in the data folder 'Dat' (80 frames altogether), calculate the spatial LSI (using a 3*3 window) contrast for each frame. By averaging all the contrast frames, draw the final sLASCA image. Implement temporal LSI (using a 20-frame window) too and draw the final tLASCA image. Apply eLASCA to both images and plot the new figures (2 figures). Send me the codes and plots. All images in gray map. (40 points)
Hint: Programming with MATLAB. Might use imread and imagesc functions for image loading and plotting. Try to improve the computational efficiency by intelligent matrix manipulation.
Answer:

1) **Draw the sLASCA and corresponding eLASCA image**
Calculate the spatial LSI (using a 3*3 window, $n = 2$ in the following equation) contrast for each frame according to the equation:

$$K_s(i,j) = \frac{\sqrt{\frac{1}{(n+1)^2}\sum_{x=i-\frac{n}{2}}^{i+\frac{n}{2}}\sum_{y=j-\frac{n}{2}}^{j+\frac{n}{2}} I_{x,y}^2 - \left(\frac{1}{(n+1)^2}\sum_{x=i-\frac{n}{2}}^{i+\frac{n}{2}}\sum_{y=j-\frac{n}{2}}^{j+\frac{n}{2}} I_{x,y}\right)^2}}{\frac{1}{(n+1)^2}\sum_{x=i-\frac{n}{2}}^{i+\frac{n}{2}}\sum_{y=j-\frac{n}{2}}^{j+\frac{n}{2}} I_{x,y}} \tag{2}$$

where $n + 1$ is the kernel size. Then average all the contrast frames and draw the final sLASCA image.

The procedures of eLASCA can be summarized as (P. Miao et al.):

- Reshape 2-D contrast K^2 to 1-D vector f.
- Sort the vector f by ascending order and save it as g.
- Estimate f_e.
- Reshape the vector fe back to 3-D K_e^2.

The images of sLASCA and corresponding eLASCA result are shown in Figure 4:

2) **Draw the tLASCA and corresponding eLASCA image**
Implement temporal LSI (using a 20- frame window, n=20 in the following equation) according to equation:

$$K_t(i,j) = \frac{\sqrt{\frac{1}{n+1}\sum_{l=t-\frac{n}{2}}^{t+\frac{n}{2}} I_{x,y,l}^2 - \left(\frac{1}{n+1}\sum_{x=t-\frac{n}{2}}^{t+\frac{n}{2}} I_{x,y,l}\right)^2}}{\frac{1}{n+1}\sum_{l=t-\frac{n}{2}}^{t+\frac{n}{2}} I_{x,y,l}} \tag{3}$$

Then draw the final tLASCA image. eLASCA procedure is the same as 1). Results are presented in Figure 5.

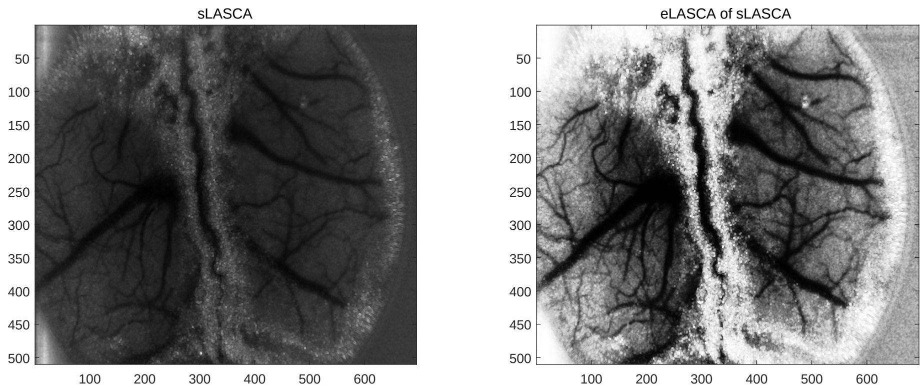

Figure 4: sLASCA and corresponding eLASCA image

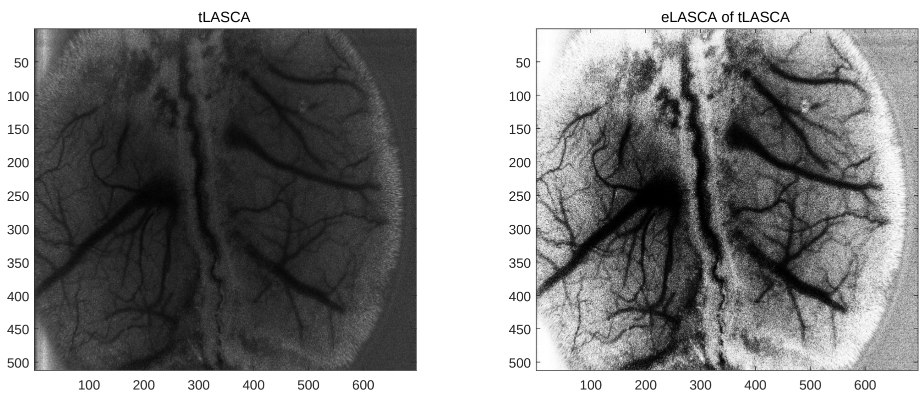

Figure 5: tLASCA and corresponding eLASCA image

Miao P, Li M, Fontenelle H, et al. Imaging the cerebral blood flow with enhanced laser speckle contrast analysis (eLASCA) by monotonic point transformation[J]. IEEE Transactions on Biomedical Engineering, 2008, 56(4): 1127-1133.

Code for eLASCA function

```
function eLASCARec=eLASCA(stLASCA)
[M,N]=size(stLASCA);
[g, g_idx] = sort(stLASCA(:));

%% get f_e
f_e = zeros(size(g));
left=1;max_tmp=g(1);
for righ=1:M*N
    if righ==M*N
        f_e(left:righ) = righ/(M*N);
    elseif g(righ)>max_tmp
        max_tmp=g(righ);
        f_e(left:righ-1)=(righ-1)/(M*N);
        left=righ;
    end
end

%% return to M*N matrix
Result=zeros(2,M*N);
Result(1,:)=g_idx';
Result(2,:)=f_e';
GIm=sortrows(Result');
ReIm=GIm(:,2);
eLASCARec=reshape(ReIm,M,N);
end
```

code for Question 3-1, to get the sLASCA and its eLASCA image

```
%% configuration
file_path =  '.\Dat\';
img_path_list = dir(strcat(file_path,'*.tif'));
img_num = length(img_path_list);
kernel_size = 3;
par = 1/(kernel_size^2); % parameter
img0 = imread(strcat(file_path,img_path_list(1).name));
size_img = size(img0);

%% calcute the contrast for each frame and sum them in matrix sLASCA
sLASCA = zeros(size_img(1)-2, size_img(2)-2);
for i_img = 1: img_num
    img0 = imread(strcat(file_path,img_path_list(i_img).name));
    img0 = double(img0); % attention here

    Pa = zeros(size_img(1)-2, size_img(2)-2);
    Pb = zeros(size_img(1)-2, size_img(2)-2);
    for i_y_shift = 1: kernel_size
        for i_x_shift = 1: kernel_size
            img_roi = img0(i_y_shift:size_img(1)+i_y_shift-3, ...
                i_x_shift: size_img(2)+i_x_shift-3);
            Pa = Pa + img_roi.*img_roi; % matrix manipulation
```

```
            Pb = Pb + img_roi;
        end
    end
    sLASCA0 = sqrt(par*Pa - par^2*(Pb.*Pb))./(par*Pb);
    sLASCA = sLASCA + sLASCA0;
end

 %% sLASCA and  eLASCA
sLASCA = sLASCA / img_num;
esLASCA = eLASCA(sLASCA);
subplot(121),colormap(gray(256));imagesc(sLASCA);title('sLASCA');
subplot(122),colormap(gray(256));imagesc(esLASCA);title('eLASCA of ...
    sLASCA');
```

code for Question 3-2, get the tLASCA and its eLASCA image

```

%% configuration and store all the frames in I1 and their square in I2
file_path =  '.\Dat\';  I = dir(fullfile(file_path,'*.tif'));
filename = fullfile(file_path,I(1).name); img0 = imread(filename);
size_img = size(img0); img_num = numel(I);

I1 = zeros(size_img(1), size_img(2), img_num);
I2 = zeros(size_img(1), size_img(2), img_num);
for  k= 1:img_num
  filename = fullfile(file_path,I(k).name);img0 = ...
      double(imread(filename));
  I1(:, :, k) = img0;
  I2(:, :, k) = img0.*img0;
end
temp_length = 21; temp_range = (temp_length-1)/2; par=1/temp_length;

%% calcute the contrast for each frame and sum them in matrix sLASCA
tLASCA = zeros(size_img);
for i_frame = temp_range+1: img_num - temp_range
    Pa = sum(I2(:,:, i_frame-temp_range: i_frame+temp_range), 3);
    Pb = sum(I1(:,:, i_frame-temp_range: i_frame+temp_range), 3);
    tLASCA0 = sqrt(par*Pa - par^2*(Pb.*Pb))./(par*Pb);
    tLASCA = tLASCA + tLASCA0;
end

 %% tLASCA and  eLASCA
tLASCA = tLASCA / (img_num-2*temp_range);
etLASCA = eLASCA(tLASCA);
subplot(121),colormap(gray(256));imagesc(tLASCA);title('tLASCA');
subplot(122),colormap(gray(256));imagesc(etLASCA);title('eLASCA of ...
    tLASCA');
```