Quantitative and Functional Imaging

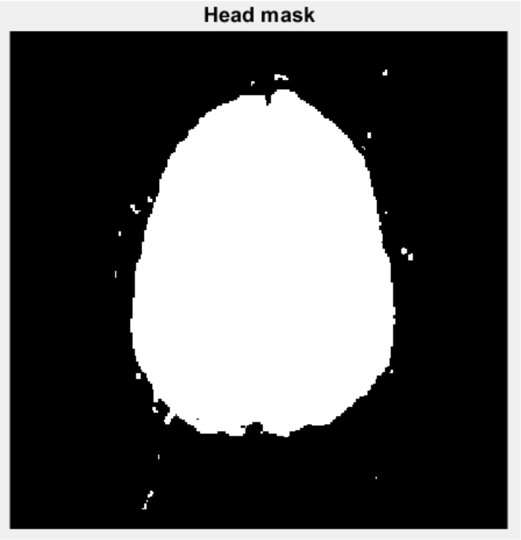
BME 4420/7450

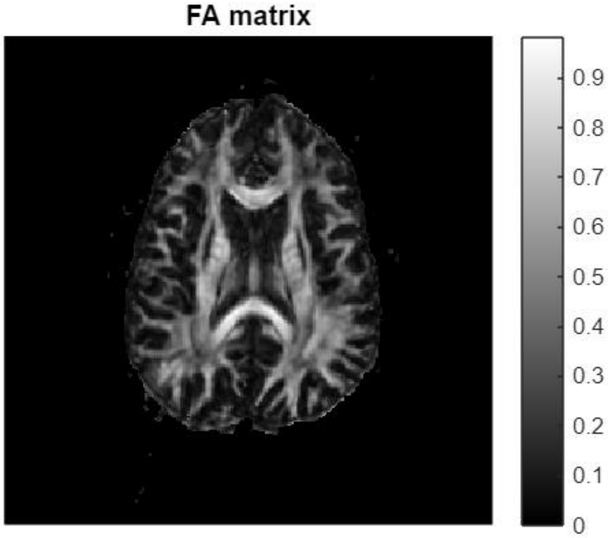
Project #4

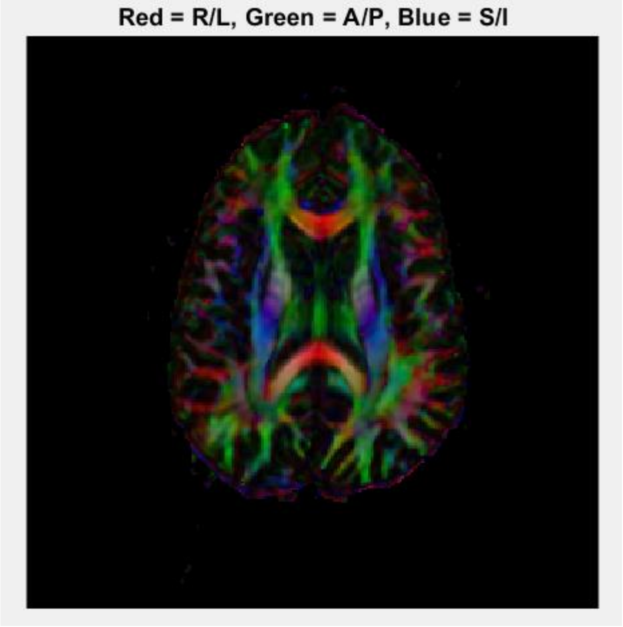
Imaging Tissue Microstructure with Diffusion Tensor Imaging

Project:

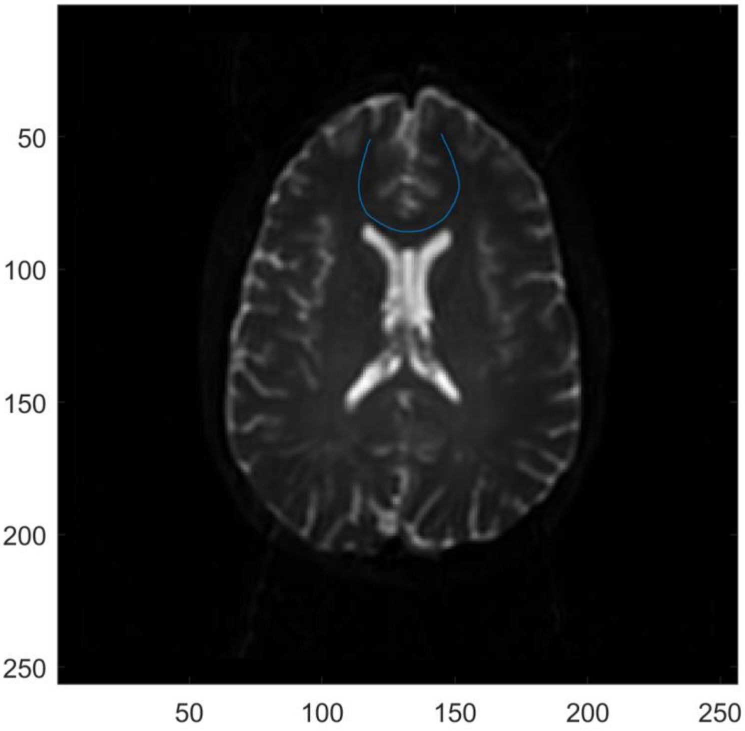
1. Your gray-scale and color-coded FA maps



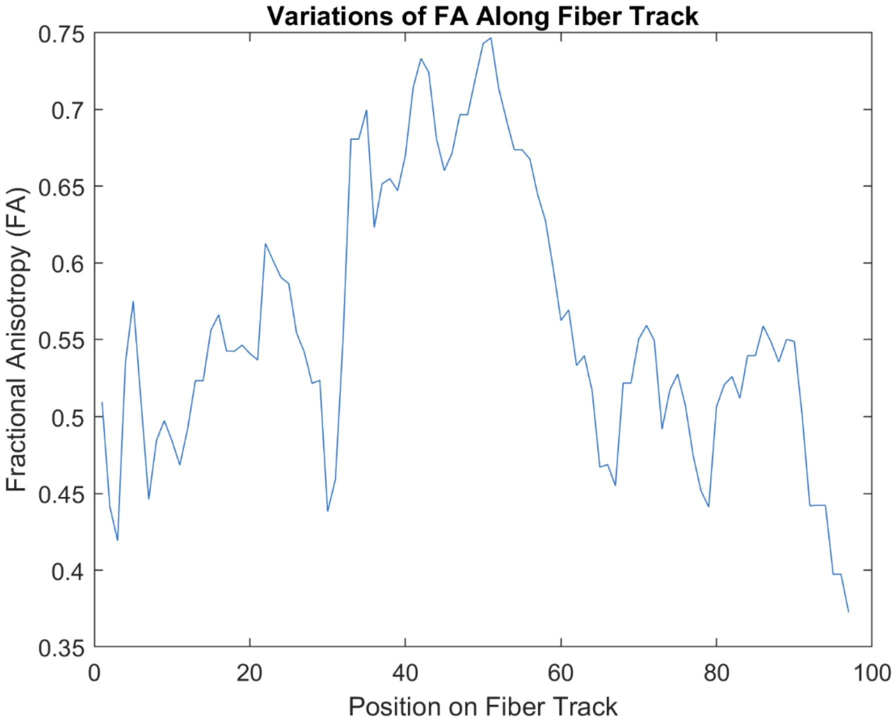




2. A figure showing the anatomical image with a fiber path connecting gray matter regions in the two hemispheres



3. Your plot of FA as a function of position along the fiber



Questions:

1. How accurate do you think your fiber path is? If there are points where the path took a wrong turn, indicate these on your figure.

- The path appears to be smooth and accurate; there do not appear to be any points where the path took a wrong turn.

2. Can you think of simple ways to improve the tracking algorithm? If so, briefly describe them.

- You could select multiple points on the corpus callosum and take an average of the fiber directions. This may give you a more representative fiber path and give you standard deviation to calculate the error propagation of the algorithm.
- You could use several images as we did in project one to look at the fiber path as a function of T1 or T2.
- Additionally, decreasing our step size in the while loop of our algorithm would also allow us to improve the accuracy of our fiber tracking.

3. What is the total length of the fiber path (the path that you show in your report)? The width of each pixel is 1 mm.

- 101 pixels * 1 mm/pixel = 101 mm = 10.1 cm total length of fiber path

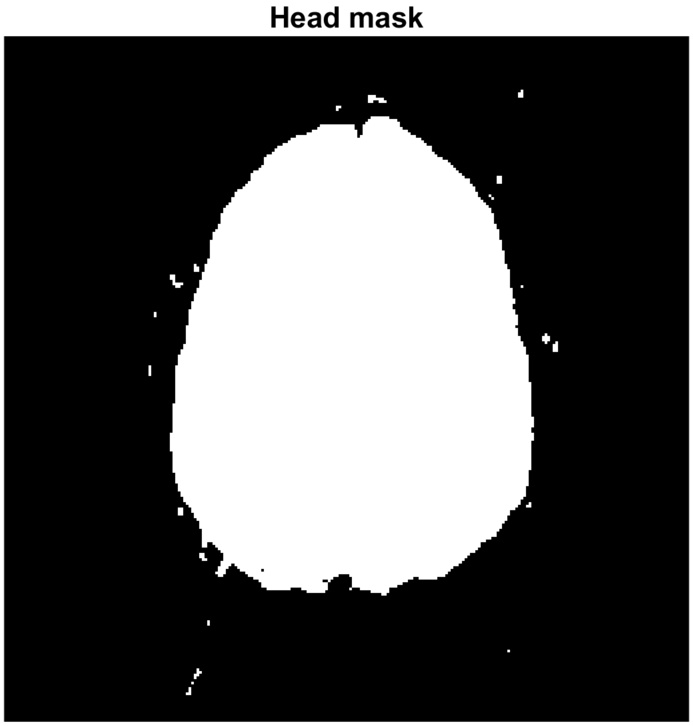
4. What tissue properties might account for the variation of FA along the fiber?

- Since diffusion depends primarily on the movement of water molecules, tissue compartments with water moving strongly in a specific direction (anisotropic) will increase FA closer to 1.
- From our plot, we have 3 peaks (one tall peak in the center of the fiber track and two smaller peaks on either side). Those likely represent the parts of the imaged brain along that path where axons are strongly anterior/posterior-oriented (Green), then left/right-oriented Red) then Green again.
- Low FA areas of the plot correspond to areas where axon orientation in the tissue changes direction from left/right to anterior/posterior.
- In disease cases, low FA may represent fewer interhemispheric fibers, reduced axonal myelination, or a higher proportion of crossing fibers.

Code:

Project 4: Fiber Tracking

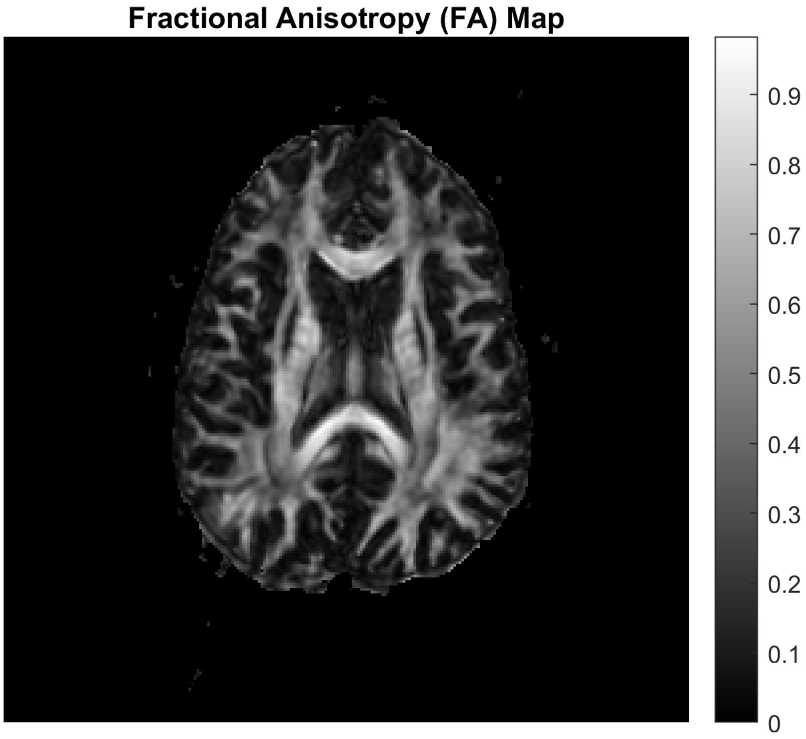
```
clc
clear
load("proj4data.mat")

[nRows, nCols] = size(anat_m) ;

% make binary mask:

image_1 = squeeze(anat_m(:,:,1)) ;

mask_1 = (image_1 > 0.05 * max(image_1(:))) ;
figure
imagesc(mask_1)
colormap(gray)
axis image
axis off
title("Head mask")
```



Head mask

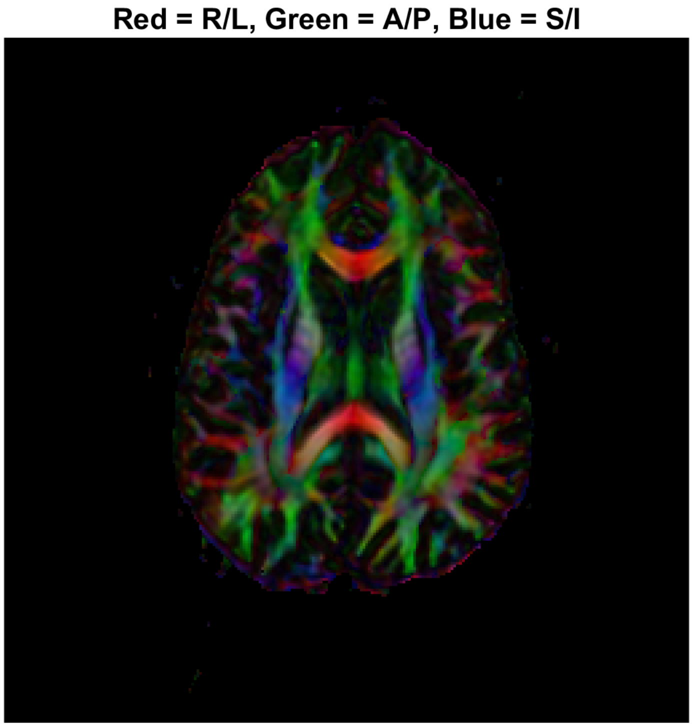
```
% calculate the fractional anisotropy (FA) in every pixel in head mask

fa_m = zeros(nRows,nCols) ;

for row = 1:nRows
    for col = 1:nCols
        if (mask_1(row,col)==1)
```

```
            gam1 = eigValues_3d(row,col,1) ;
            gam2 = eigValues_3d(row,col,2) ;
            gam3 = eigValues_3d(row,col,3) ;
            gam_avg = (gam1+gam2+gam3)./3 ;

            res1 = (gam1-gam_avg).^2 ;
            res2 = (gam2-gam_avg).^2 ;
            res3 = (gam3-gam_avg).^2 ;
            denom = gam1.^2+gam2.^2+gam3.^2 ;

            fa_m(row,col) = sqrt((1.5 .*(res1+res2+res3))./denom) ;
        end
    end
end

figure
imagesc(fa_m)
colormap(gray)
hc=colorbar;
axis image
axis off
title("Fractional Anisotropy (FA) Map")
```



```
% Display color-coded FA map:
red_m = fa_m .* abs(fastDiffVector_3d(:, :, 1));
green_m = fa_m .* abs(fastDiffVector_3d(:, :, 2));
blue_m = fa_m .* abs(fastDiffVector_3d(:, :, 3));
```

```
color_3d = cat(3, red_m, green_m, blue_m);
imagesc(color_3d)
axis image
axis off
title('Red = R/L, Green = A/P, Blue = S/I')
```



```
% Prompt user to identify a seed point in a white matter fiber that passes
% through the corpus callosum
% Find the coordinates of the seed point

imagesc(anat_m)
axis image
colormap(gray)
hold on
disp('Define seed point...')
```

```
Define seed point...
```

```
[x0, y0] = ginput(1);

% create variables to hold the current x and y positions of the fiber path

% initialize variables:
x = x0;
y = y0;
x_v = [x0];
y_v = [y0];
stepFlag = 1;
cosAngle = 1;
```

```
stepSize = 1;
count = 1;

while (stepFlag == 1)
    % Insert code here to find the fast diffusion direction at the nearest
    % integer values of x and y:

    fast_3d = fastDiffVector_3d(round(y),round(x),:) ;

    % We want to track fibers in the plane of the image, so insert code
    % here to find the component of the fast diffusion direction in the
    % image plane. Call this fast_v (it should have just two elements,
    % the x and y components of the fast diffusion direction):

    fast_v = fast_3d(:,:,1:2) ;

    % Break out of the while loop if the in-plane component of the
    % vector is too small:
    if (sum(fast_v.^2) < 0.5)
        disp('In-plane component of fast_v is too small')
        break
    end
    % If this is not the first step away from the seed point, calculate
    % cosAngle, the cosine of the angle between the previous step and
    % fast_v. If cosAngle is negative, reverse the direction of fast_v
    % (add code here):

    if count ~= 1
        dot_prod = dot(fast_v,fast_v_old) ;
        abs_prod = abs(fast_v) .* abs(fast_v_old) ;
        cosAngle = dot_prod ./ abs_prod ;
        if cosAngle < 0
            fast_v = -fast_v ;
        end
    end

    % Step a distance stepSize in the direction of fast_v
    % (i.e., update x_v and y_v):

    x= x + stepSize .* fast_v(1) ;
    y = y + stepSize .* fast_v(2) ;

    x_v(end+1) = x ;
    y_v(end+1) = y ;

    % Add a line segment to the image to show the current step:
    line([x_v(length(x_v)-1), x], [y_v(length(y_v)-1), y])
    drawnow

    % Add code here to set stepFlag = 0 if abs(cosAngle) is too small:

    if abs(cosAngle) < 0.1          % arbitrary threshold
        stepFlag = 0 ;
    end
    fast_v_old = fast_v ;
    count = count +1;
end
```

```
In-plane component of fast_v is too small
```

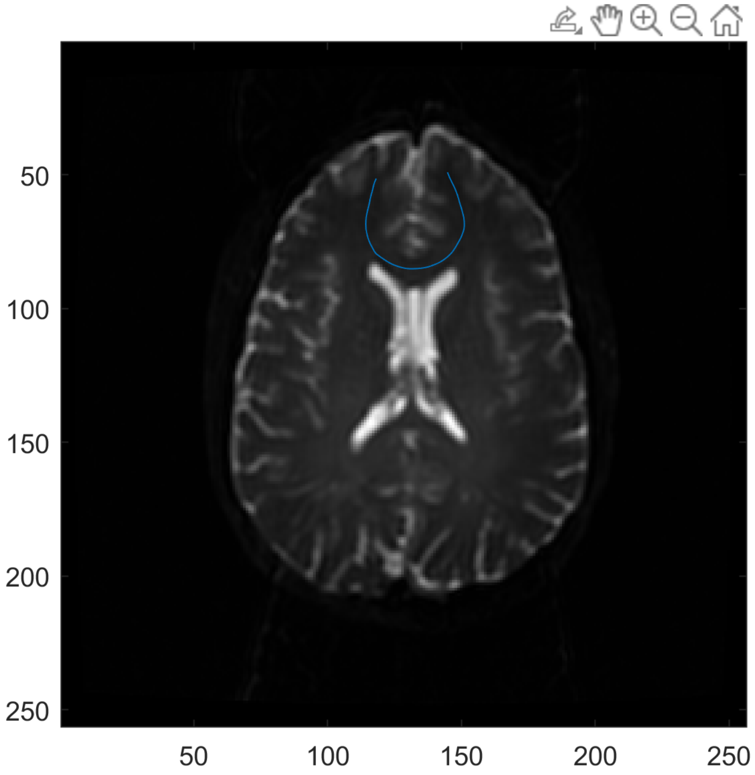
```

% repeat parts 5 and 6 to track in the opposite direction from the seed
% point

% Prompt user to identify a seed point in a white matter fiber that passes
% through the corpus callosum
% Find the coordinates of the seed point

% imagesc(anat_m)
% axis image
% colormap(gray)
% hold on
disp('Define seed point...')
```

```
Define seed point...
```

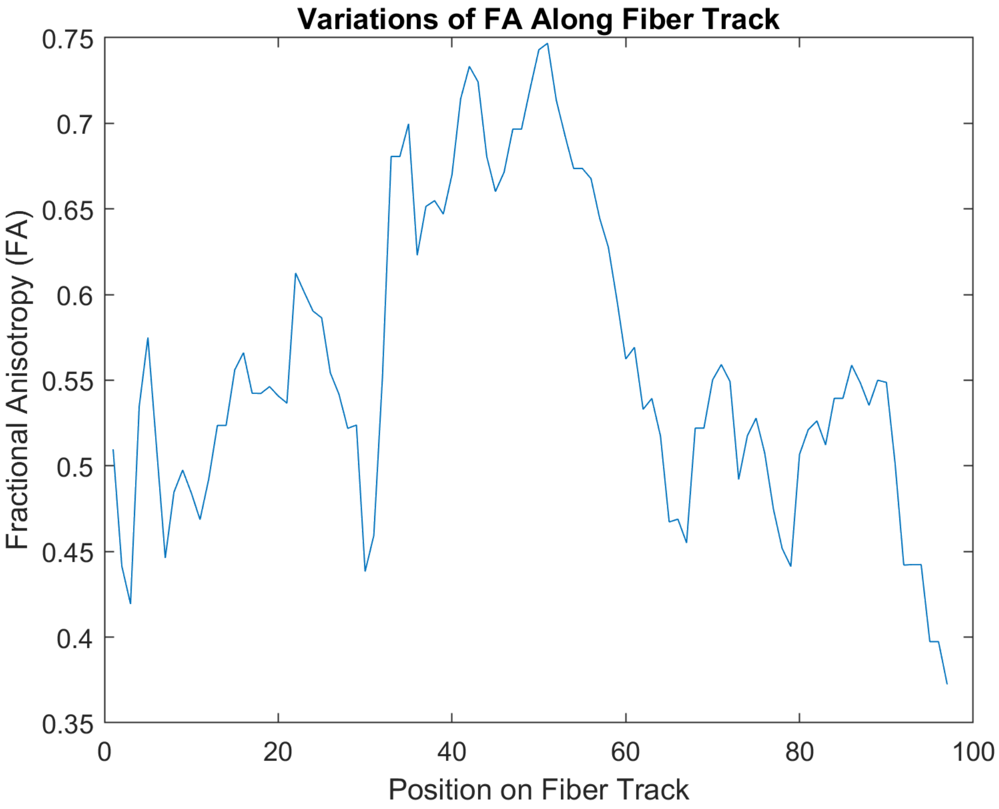
```
[x0, y0] = ginput(1);

% initialize variables:
x = x0;
y = y0;
x_v2 = [x0];
y_v2 = [y0];
stepFlag = 1;
cosAngle = 1;
stepSize = -1;          % only change from previous code (opposite direction)
count = 1;

while (stepFlag == 1)
    % Insert code here to find the fast diffusion direction at the nearest
    % integer values of x and y:

    fast_3d = fastDiffVector_3d(round(y),round(x),:) ;

    % We want to track fibers in the plane of the image, so insert code
    % here to find the component of the fast diffusion direction in the
    % image plane. Call this fast_v (it should have just two elements,
    % the x and y components of the fast diffusion direction):

    fast_v = fast_3d(:,:,1:2) ;

    % Break out of the while loop if the in-plane component of the
    % vector is too small:
    if (sum(fast_v.^2) < 0.5)
        disp('In-plane component of fast_v is too small')
        break
    end
    % If this is not the first step away from the seed point, calculate
    % cosAngle, the cosine of the angle between the previous step and
    % fast_v. If cosAngle is negative, reverse the direction of fast_v
    % (add code here):

    if count ~= 1
        dot_prod = dot(fast_v,fast_v_old2) ;
        abs_prod = abs(fast_v) .* abs(fast_v_old2) ;
        cosAngle = dot_prod ./ abs_prod ;
        if cosAngle < 0
            fast_v = -fast_v ;
        end
    end

    % Step a distance stepSize in the direction of fast_v
    % (i.e., update x_v and y_v):

    x= x + stepSize .* fast_v(1) ;
    y = y + stepSize .* fast_v(2) ;

    x_v2(end+1) = x ;
    y_v2(end+1) = y ;

    % Add a line segment to the image to show the current step:
    line([x_v2(length(x_v2)-1), x], [y_v2(length(y_v2)-1), y])
    drawnow

    % Add code here to set stepFlag = 0 if abs(cosAngle) is too small:

    if abs(cosAngle) < 0.1          % arbitrary threshold
        stepFlag = 0 ;
    end

    fast_v_old2 = fast_v ;
    count = count +1;
end
```

```
In-plane component of fast_v is too small
```

```

% free the figure for the next graphics command:
hold off
```



```
% Plot the FA as a function of position along the fiber track you just
% created in the step above. Be sure to order your data to show FA starting
% at one end of the track and ending at the other. Note the variations of
% FA along the fiber.

y_v = fliplr(y_v) ;
x_v = fliplr(x_v) ;

y_inds = [round(y_v), round(y_v2)] ;
x_inds = [round(x_v), round(x_v2)] ;
FA_vect = zeros(1,length(y_inds)) ;

for ii = 1:length(y_inds)
    FA_vect(ii) = fa_m(y_inds(ii),x_inds(ii)) ;
end

plot(FA_vect)
ylabel("Fractional Anisotropy (FA)")
xlabel("Position on Fiber Track")
title("Variations of FA Along Fiber Track")
```



```
% What is the total length of the fiber path?
% The width of each pixel is 1 mm.

(length(y_inds)) .* 1 % mm
```

```
ans = 101
```