Quantitative and Functional Imaging

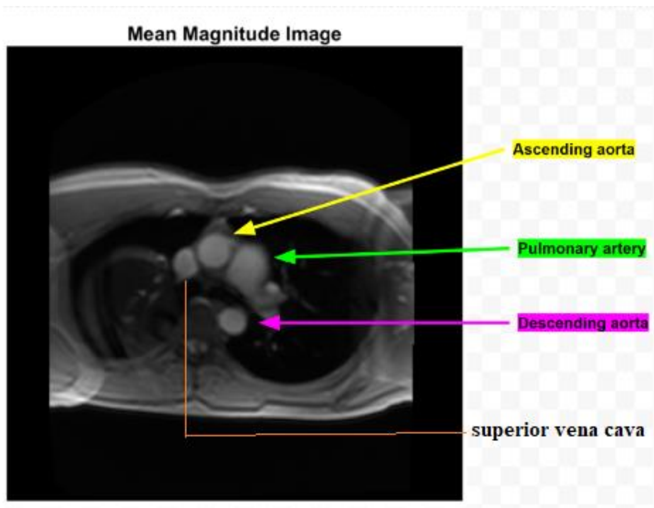
BME 4420/7450

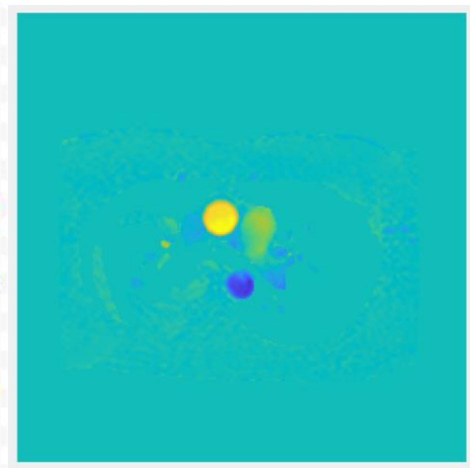
Project #6

<u>Measuring Arterial Blood Flow</u>

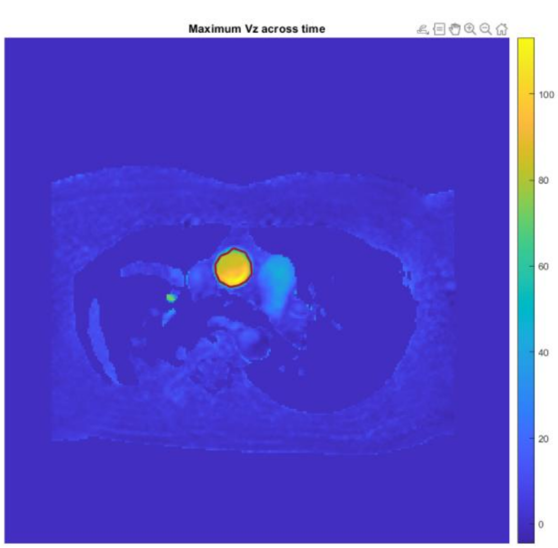
<u>Deliverables:</u>

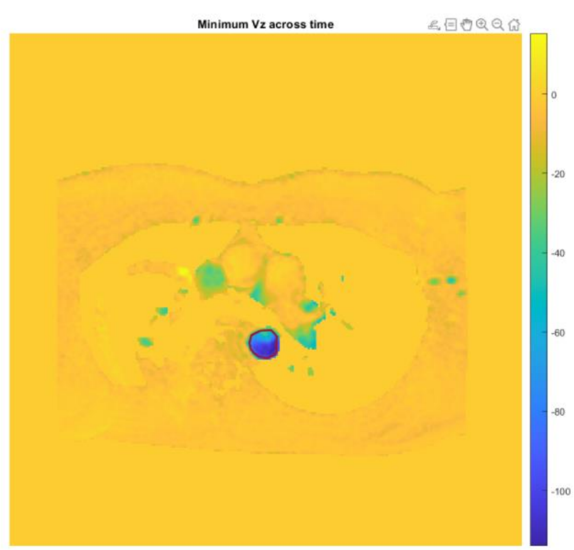
1. A figure showing the mean magnitude image. Label the superior vena cava, the trunk of the pulmonary arteries, and the ascending and descending aorta.



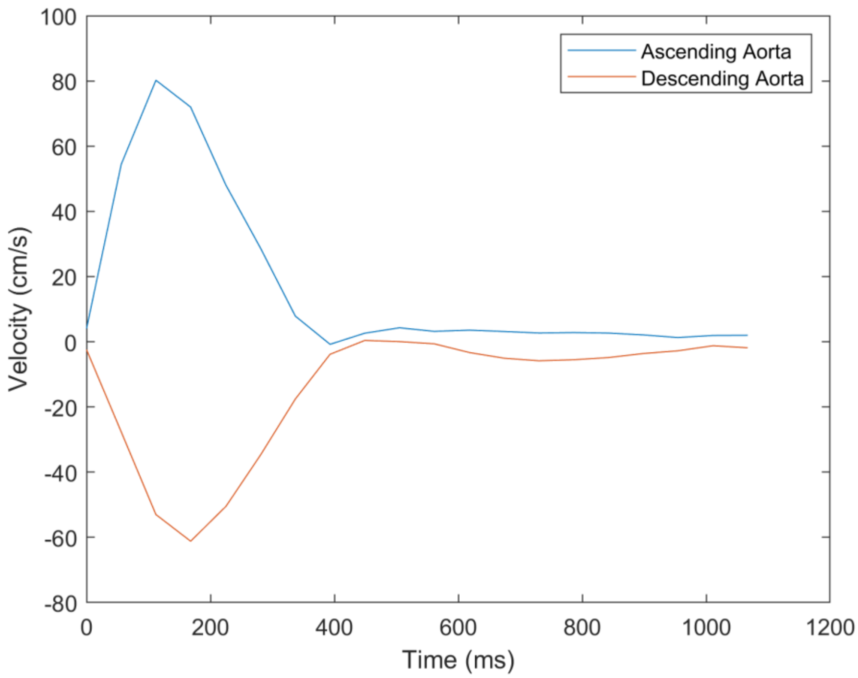


2. Both velocity maps showing the outlines of the ascending and descending aorta.

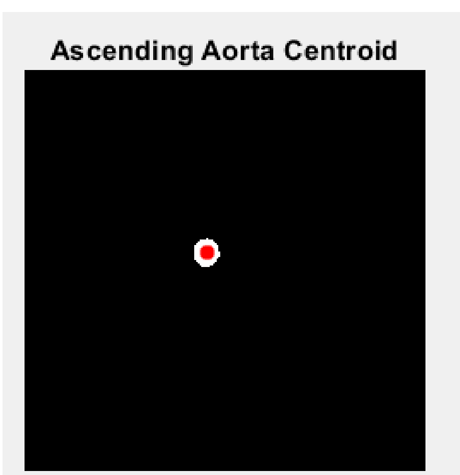


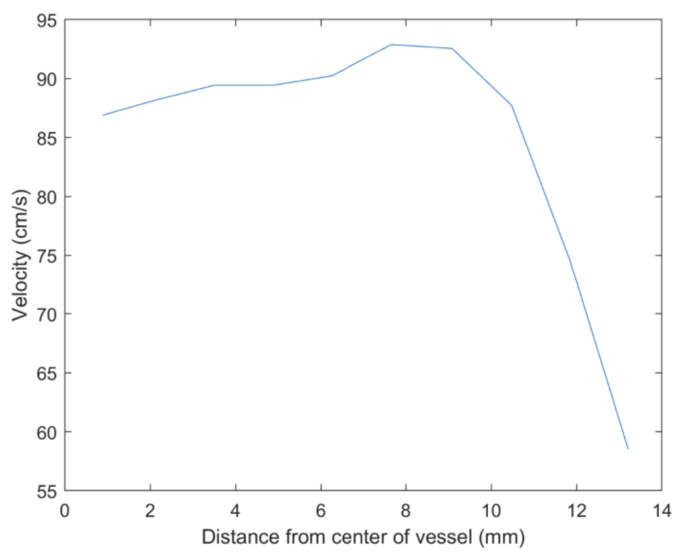


3. Your plot of mean velocity versus time for the ascending and descending aorta.



4. Your plot of mean velocity versus radius in the ascending aorta (for graduate credit/undergrad extra credit).





<u>Questions:</u>

1. What is the total volume of blood pumped through the ascending aorta in one cardiac cycle? What is the total volume passing through the descending aorta?
    - Depending on the ROIs we draw there are variations in the volume obtained
    - Ascending aorta: 103.5368 mL

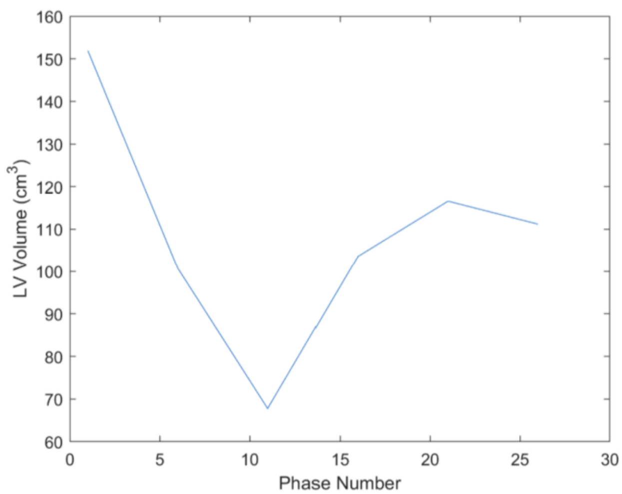
- Descending aorta: 81.4321 mL

Ascending aorta (mL)	103.5368	100.52	97.1480
Descending aorta (mL)	81.4321	65.68	68.7868
Fractional Difference (%)	21.35	34.6598	29.1938

- This makes sense that the descending aorta has less volume than the ascending aorta because some blood leaves the aorta to go to the arms and head after passing through the ascending aorta, and therefore does not reach the descending aorta.

2. How does the total volume pumped through the ascending aorta compare to your results in Project 2? How should these quantities be related?

Results from project 2:



- Results from Project 2 indicated that roughly 84 mL was ejected from the left ventricle into the ascending aorta during the cardiac cycle. This is roughly similar to the total ascending aorta volumes (for example 103.5 mL) that we calculated here in Project 6. The difference in the two values is due to this being a different patient and/or the image being taken of the patient at a different time when the heart was pumping more forcefully.

3. If the total volumes of blood flowing in the ascending and descending aorta are not the same, why do you think they are different? Where is the 'missing' blood flowing?
    - The "missing" blood is flowing to the head/arms via arteries that branch off of the arch of the aorta.

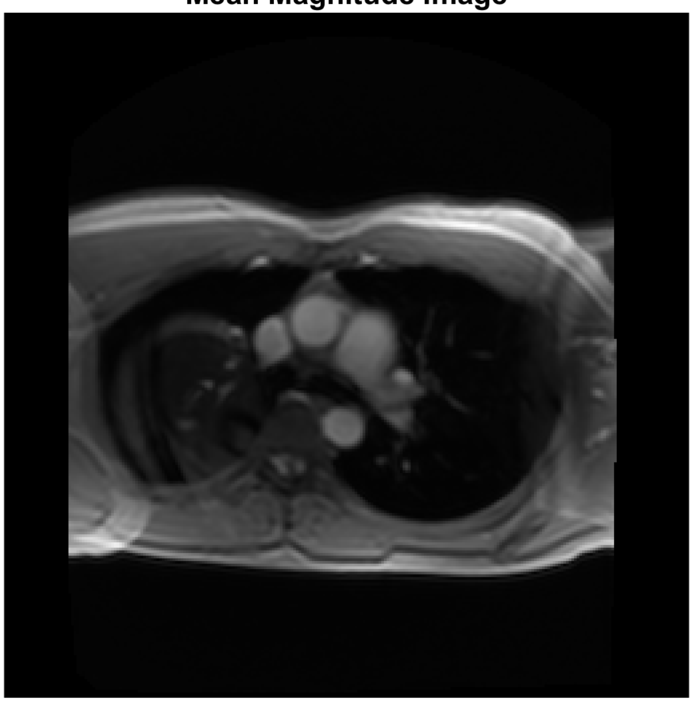
4. Is the velocity profile parabolic at high flow rates? <u>For Graduate Credit (undergrad extra credit)</u>

- No, the velocity profile is not parabolic. This is likely because, at high flow rates, the flow will not be laminar. The ascending aorta is also curved, so the velocity distribution ends up being skewed towards the outer wall, as shown in the vessel radius vs. velocity plot (the max velocity occurs not at the center of the vessel as one would expect for parabolic flow, but instead the max velocity occurs around 7 mm from the center of the vessel).

Project 6

```
clc
clear

load("proj6_arterialFlowData_qfi.mat")

% calculate the mean magnitude image (over time) and display

nRows = 256;
nCols = 256;
mean_mag_M = zeros(nRows,nCols) ;

for ii = 1:nRows
    for jj = 1:nCols
        mean_mag_M(ii,jj) = mean(mag_3d(ii,jj,:)) ;
    end
end

figure
imagesc(mean_mag_M)
colormap(gray)
axis image
axis off
title("Mean Magnitude Image")
```

Mean Magnitude Image



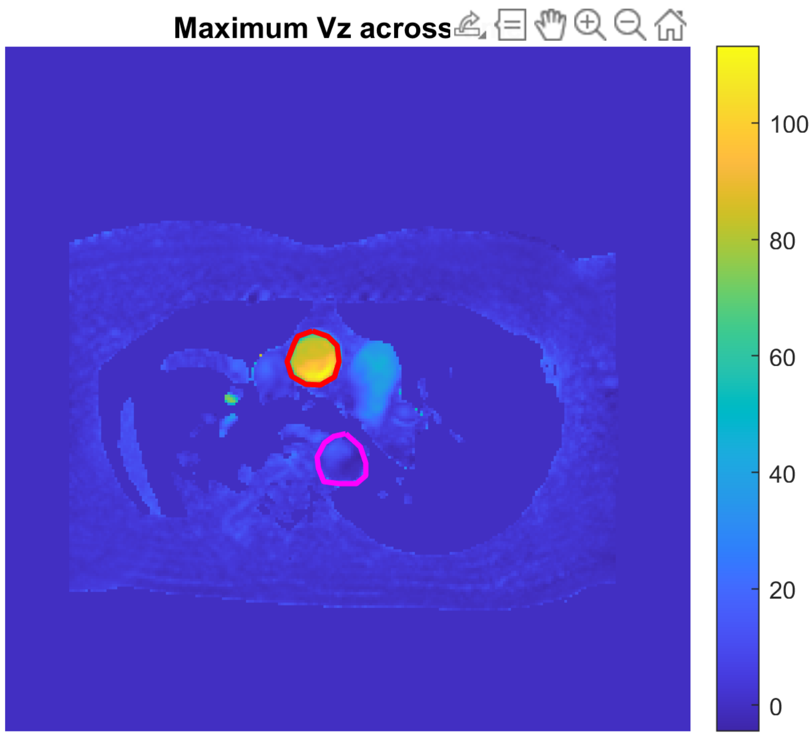
```
% make a binary mask showing where mean over time is >= 10% max pixel intensity
```

```
bodyMask_m = (mean_mag_M >= 0.1 * max(max(mean_mag_M))) ;
figure
imagesc(bodyMask_m)
colormap(gray)
axis image
axis off
title("Body Mask")
```

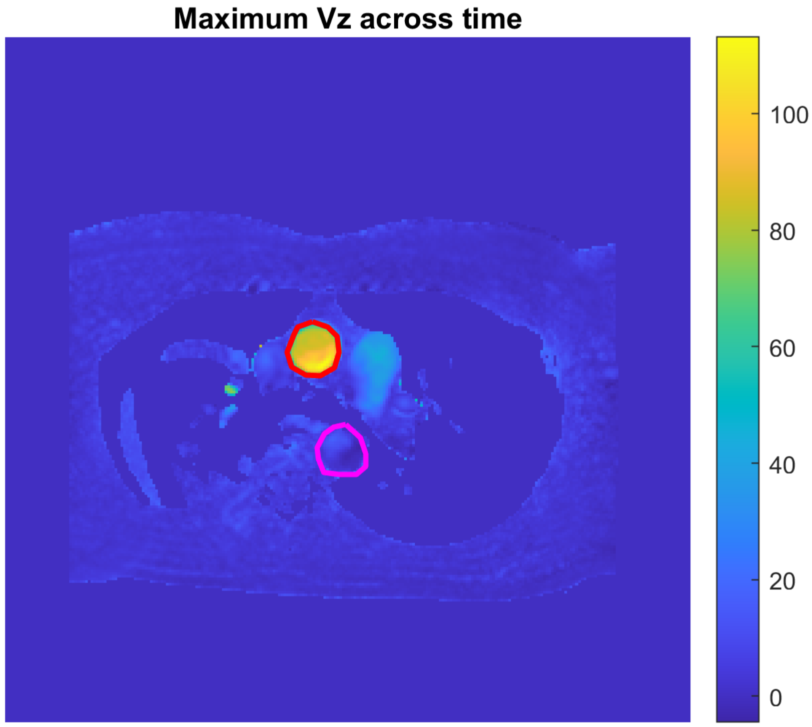

```
% convert the magnetization phase angle to velocity

vz_3d = (venc .* phase_3d .* 2) ./ pi ;
```

```
% Play pro6_movie.m

% Calculate the max vz value at each pixel and display:
maxVz_m = max(vz_3d, [], 3);
figure;
imagesc(maxVz_m .* bodyMask_m)
axis image
axis off
colorbar
title('Maximum Vz across time')
% define a mask for the ascending aorta ("aa")
% draw the boundary of your ROI
[aaMask_m, aaX_v, aaY_v] = roipoly ;
hold on;
line(aaX_v,aaY_v,'Color','r','LineWidth',2)
```

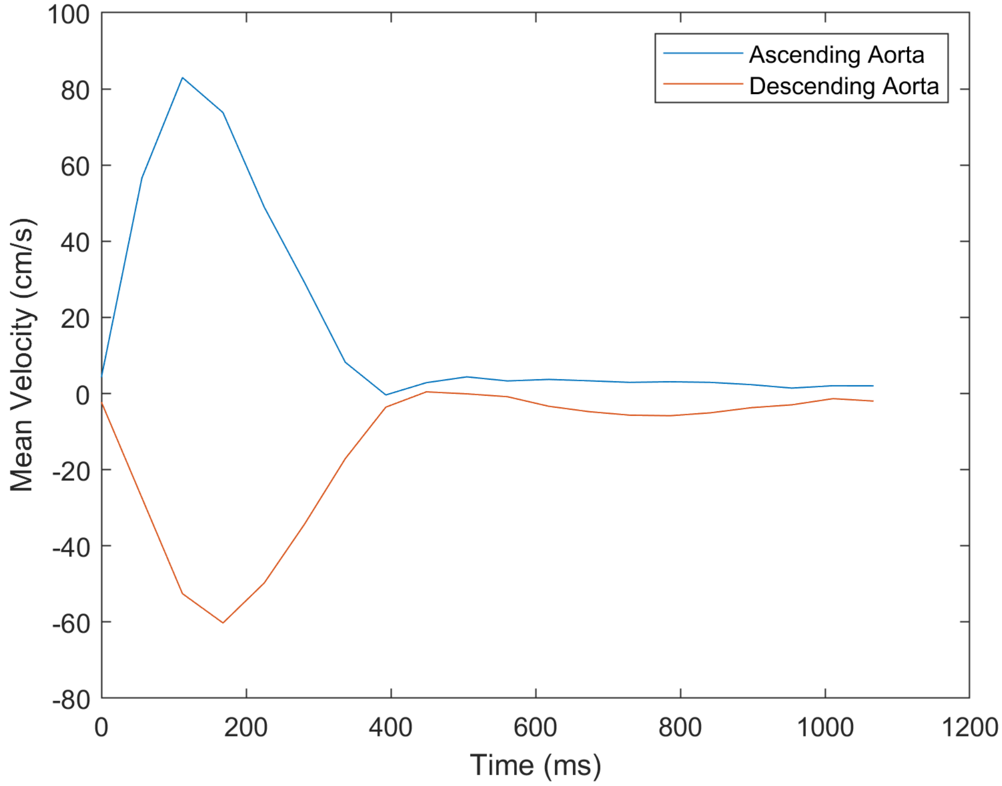
```
% repeat for descending aorta:

% define a mask for the desc. aorta ("da")
% draw the boundary of your ROI
[daMask_m, daX_v, daY_v] = roipoly ;
hold on;
line(daX_v,daY_v,'Color','m','LineWidth',2)
```




```
% calculate the mean blood velocity in the aa and da as a function of time:
hold off;
```




```
nTimes = length(time_v);
aaVz_v = zeros(1, nTimes);
daVz_v = zeros(1, nTimes);
for timeIndex = 1:nTimes
    % Insert your code to calculate the mean velocity
    % in each vessel at the current time point:
    aaM = aaMask_m .* vz_3d(:,:,timeIndex) ;
    daM = daMask_m .* vz_3d(:,:,timeIndex) ;

    aa_numel = sum(sum(aaMask_m)) ;
    da_numel = sum(sum(daMask_m)) ;
    aaRoi_sum = sum(sum(aaM)) ;
    daRoi_sum = sum(sum(daM)) ;

    aaVz_v(timeIndex) = aaRoi_sum ./ aa_numel ;
    daVz_v(timeIndex) = daRoi_sum ./ da_numel ;
end

plot(time_v, aaVz_v, time_v, daVz_v)
xlabel("Time (ms)")
ylabel("Mean Velocity (cm/s)")
legend("Ascending Aorta", "Descending Aorta")
```




```
% calculate the total volume passing through image plane over cycle for aa
% and da

aa_area = aa_numel .* dx .* dy ./ 100 ; % dx and dy in mm so convert to cm
da_area = da_numel .* dx .* dy ./ 100 ; % final units: cm squared

aaVol= aa_area .* trapz(time_v ./ 1000,aaVz_v) % convert time from ms to s
```

```
aaVol = 104.7505
```

```
daVol = abs(da_area .* trapz(time_v ./ 1000,daVz_v)) % final units: cm cubed
```

```
daVol = 77.6514
```

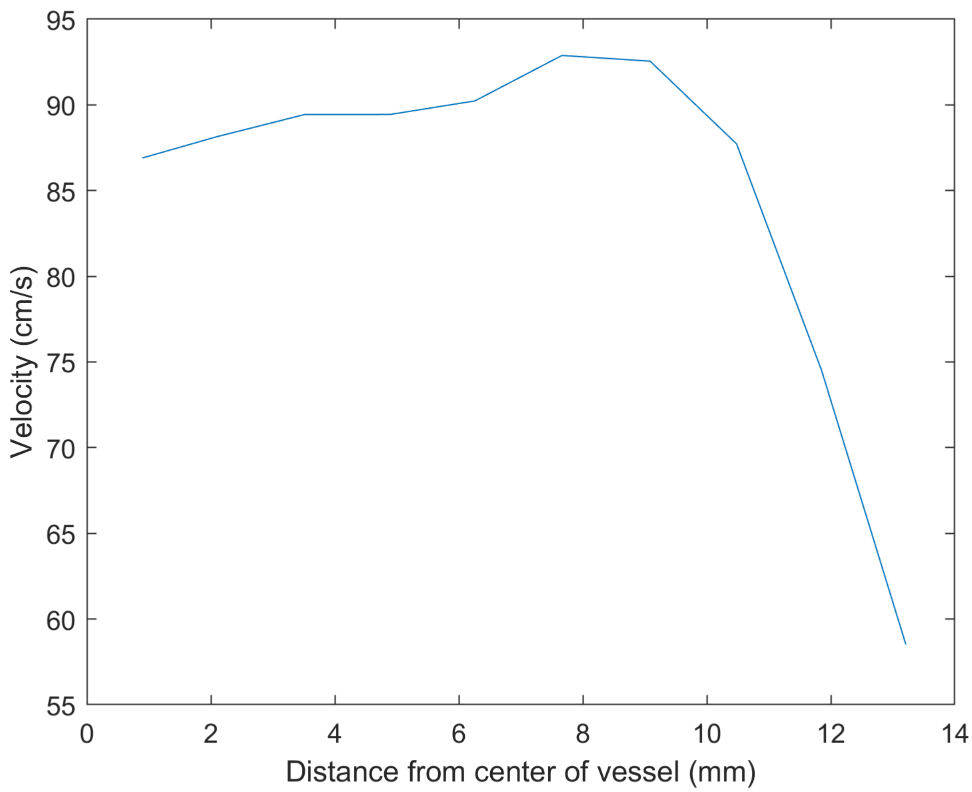
```
% calculate the fractional difference between the ascending and descending
% aorta blood volumes

fracDiff = 100*(aaVol-daVol) ./ aaVol
```

```
fracDiff = 25.8702
```

```
% Measure blood velocity in ascending aorta as a function of distance from
% center of vessel.
% Use the data from the time point with maximum mean velocity in the aorta.
% One way to do this is to calculate the coordinates of the
% center of the vessel, then find the distance from each pixel in the vessel to the
% center. Average the velocities of pixels at similar distances to find the mean
```

```
% velocity profile as a function of radius from the center.

% Find maximum mean velocity in aorta:
[max_vel, max_ind] = max(aaVz_v) ;
time_of_max = time_v(max_ind) ;
maxvel_M = aaMask_m .* vz_3d(:,:,max_ind) ;

% calculate coordinates of center of the vessel:
    % find where distance between any two points is at a max, then find
    % midpoint between those two points

    max_dist = 0;
    max_x = 0;
    max_y = 0;
    x1 = aaX_v(1) ;
    y1 = aaY_v(1) ;

    for ii = 2:length(aaX_v)
        x2 = aaX_v(ii);
        y2 = aaY_v(ii);
        dist = sqrt((x1-x2).^2 + (y1-y2).^2) ;
        if dist > max_dist
            max_dist = dist ;
            max_x = x2;
            max_y = y2;
        end
    end

    radius = dx.* max_dist ./2 ;

    % calculate coordinates of center of vessel:
    midx = (x1+max_x)./2 ;
    midy = (y1+max_y)./2 ;

% Then find the distance from each pixel to center
    dist_M = zeros(256,256);  % dist in mm

    for ii = 1:nRows
        for jj = 1:nCols
            if (aaMask_m(ii,jj)==1)
                x = jj ;
                y = ii ;
                dist_M(ii,jj) = sqrt((x-midx).^2+(y-midy).^2) ; % distance in pixels not mm!
            end
        end
    end

% Average the velocities of pixels at similar distances to find the mean
% velocity profile as a function of radius from the center.

% break up radius into 5 bins for example
bin_num = 10;
deltaX = radius ./ bin_num ;
```

```
avgdist_v = zeros(1,bin_num);        % mm
avgvel_v = zeros(1,bin_num);         % cm/s

maskInd = find(aaMask_m) ;
dist_v = dx.* dist_M(maskInd) ;
maxvel_v = maxvel_M(maskInd) ;

for bin = 1:bin_num
   smallDist = (bin-1).* deltaX ;
   largeDist = bin.* deltaX ;
   index_v = find(dist_v>=smallDist & dist_v<largeDist);
   avgdist_v(bin) = sum(dist_v(index_v)) ./ (length(index_v));
   avgvel_v(bin) = sum(maxvel_v(index_v)) ./ (length(index_v));
end

plot(avgdist_v, avgvel_v)
xlabel("Distance from center of vessel (mm)")
ylabel("Velocity (cm/s)")
```



```
% Show a movie of velocity versus time. Red is flow toward head,
% blue is toward feet:
nTimes = length(time_v);
maxVz = max(vz_3d(:));
minVz = min(vz_3d(:));
figure
for timeIndex = 1:nTimes
    imagesc(bodyMask_m .* vz_3d(:, :, timeIndex))
    axis image
    axis off
    % Set color limits to visualize slow and fast flow:
    set(gca, 'CLim', [minVz, maxVz]/2)
    drawnow
    m(timeIndex) = getframe;
end
nLoops = 2;
fps = 2; % Frames per second.
movie(m, nLoops, fps)
```