Bahcesehir University-
Capstone 4992- 1010214

DEVELOPMENT OF A NUMERICAL MODEL FOR THE HUMAN RED BLOOD CELL

Presented by
Nageen Asif (CME)
Arda Acıksoz (CME)
Tasnim Yaseen Hammoodi (BME)
Sarah Jamal Abdullah (BME)

Presentation Outline

Overview

Goal of this project

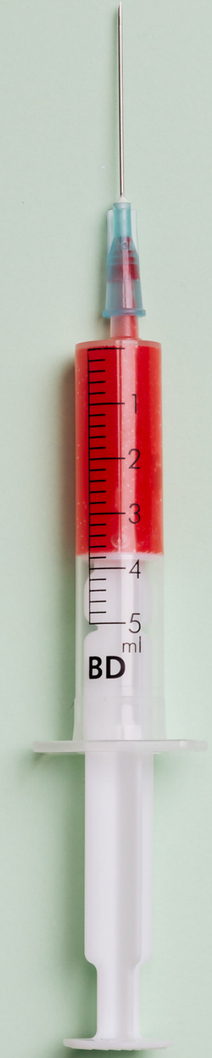
Current methods

Gantt Chart

Subsytems CME
Physical Architecture

Project Network

Results



NEXT

Overview

This capstone project intends to develop a numerical model of a human red blood cell (RBC), also known as (erythrocytes), that can be used for examination of the Red Blood Cell (RBC) rheological properties, characteristics, and kinematics in accordance with empirical results.

Did you know?

Goal of the project

NEXT

Here we will discuss the Numerical model of red blood cell. The issues we are facing in these times for modeling Red blood cells. The technologies and methods which are being used in current time for numerical or any other kind of modeling of red blood cells. The limitations they have and an overview of the processes and finally coming up with a computational approach to solve the problem. And discuss the reasons for that method being authentic and the outputs of the method.

Current methods?

Gantt chart

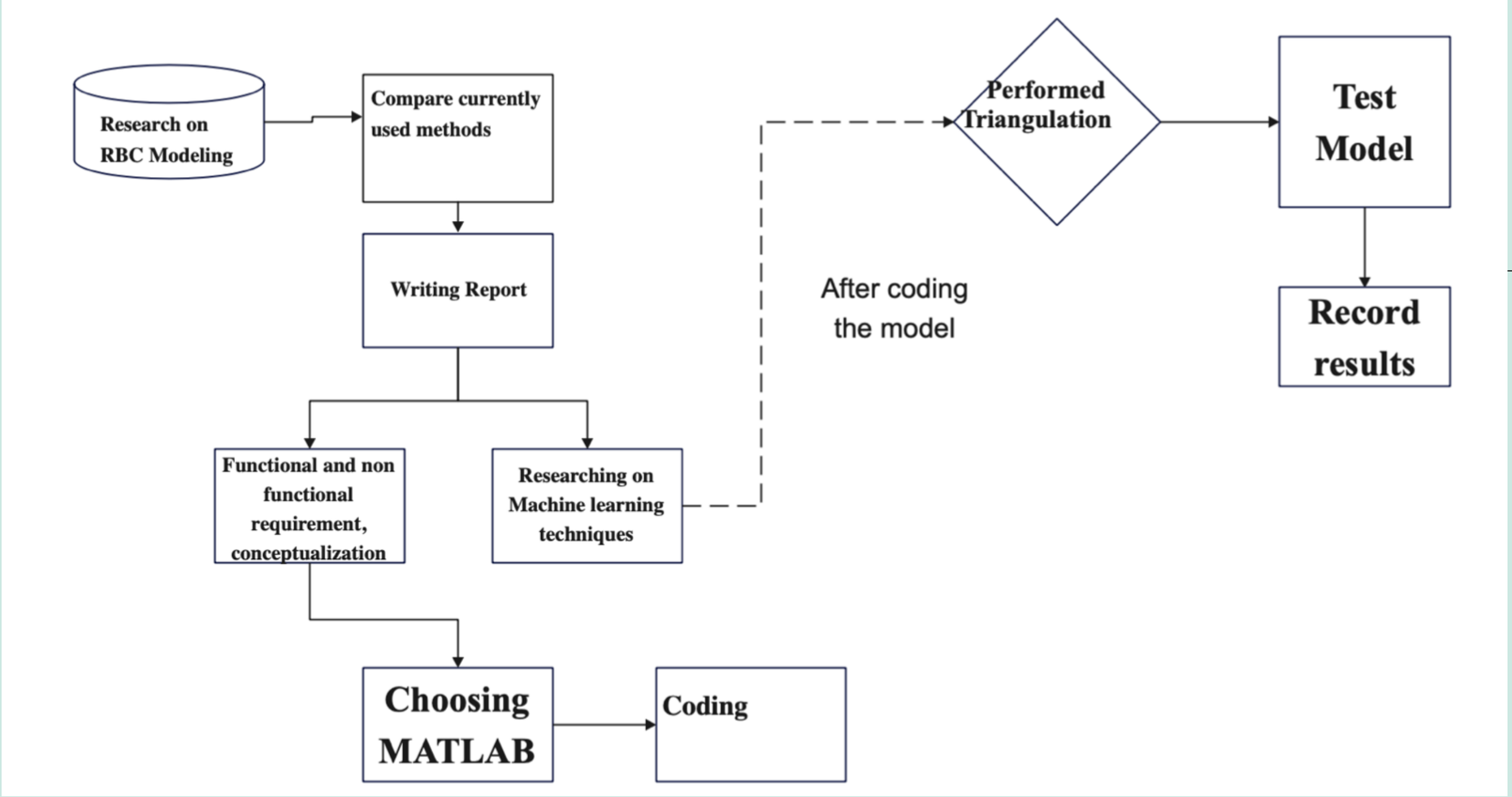
Activity / Month	Feb	Mar	Apr	May	June
Research on characteristics red blood cell					
Research on the methodology of red blood cell					
Studying the numerical model techniques and determine project requirements and numerical model techniques					
Comparing with the current models being used					
Propose a numerical model after					
Simulation, Applying numerical methods and using computing techniques					
Calculation and Analysis					
Stating the result					
Writing final report					
Preparing for presentation					

Subsystem CME

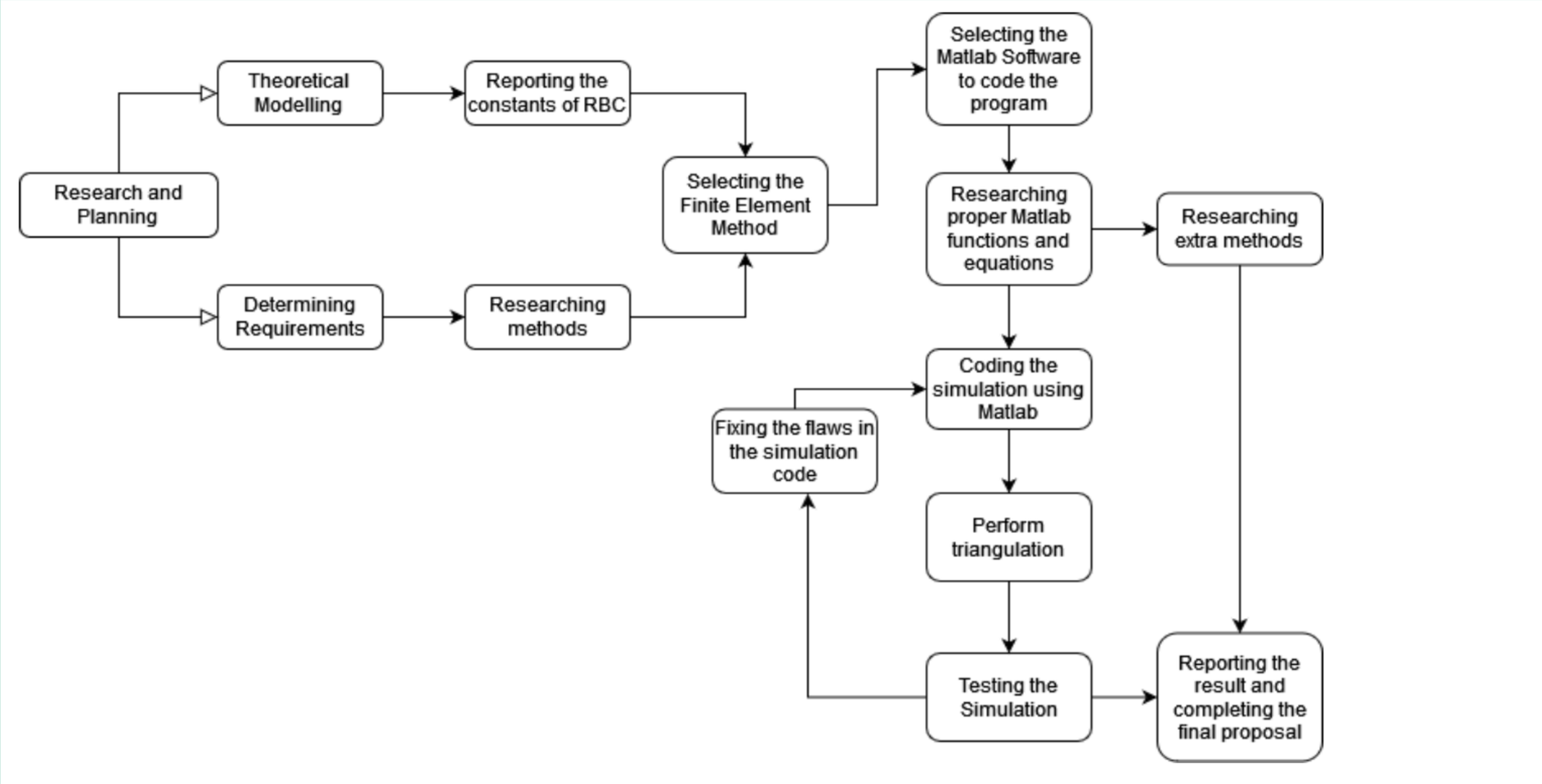
We were responsible for bringing the numerical model into existence using computational expertise. By doing research on the current RBC models and finding what are they lacking and what can be improved and coming up with a unique approach to find an accurate model to do numerical modeling of RBCs.

MATLAB

In order to visualize our numerical model, we selected MATLAB. MATLAB is a tool to analyze data, develop algorithms, and build models. After determining the methods and the technology, we coded our method using MATLAB.



Physical Architecture



Project Network

We have calculated the surface for the form of an unstressed RBC using Evans and Skalak's equation in MATLAB.

```
D0 = 7.82; % diameter of the RBC according to evans and skalak
a0 = 0.0518;
a1 = 2.0026;
a2 = -4.491; % variables of the equation
nRP = 10; % number of points along the radius

rMax = D0/2;
rvec = linspace( 0,rMax, nRP )'; % Create a vector of 10(nRP) evenly spaced points
f = 0.8;
    rad = (1-f) * rvec(end-1) + f * rvec(end);
    rvec = [rvec(1:end-1); rad; rvec(end)]; %Add an extra point
    dr = rvec(2) - rvec(1);

vecRho = [];
vecTheta = [];
for k=1:nRP
L = 2 * pi * rvec(k);
nTheta = round( L / dr );
theta = linspace( 0, 2 * pi * (nTheta - 1)/ nTheta, nTheta )';
vecRho = [vecRho; rvec(k) * ones( nTheta, 1) ];
vecTheta = [vecTheta; theta];
end;
```

D0 = 7.82 UM : cell diameter

nRP : number of points along the radius

a0,a1,a2 are the constants defined by Evans and Skalak

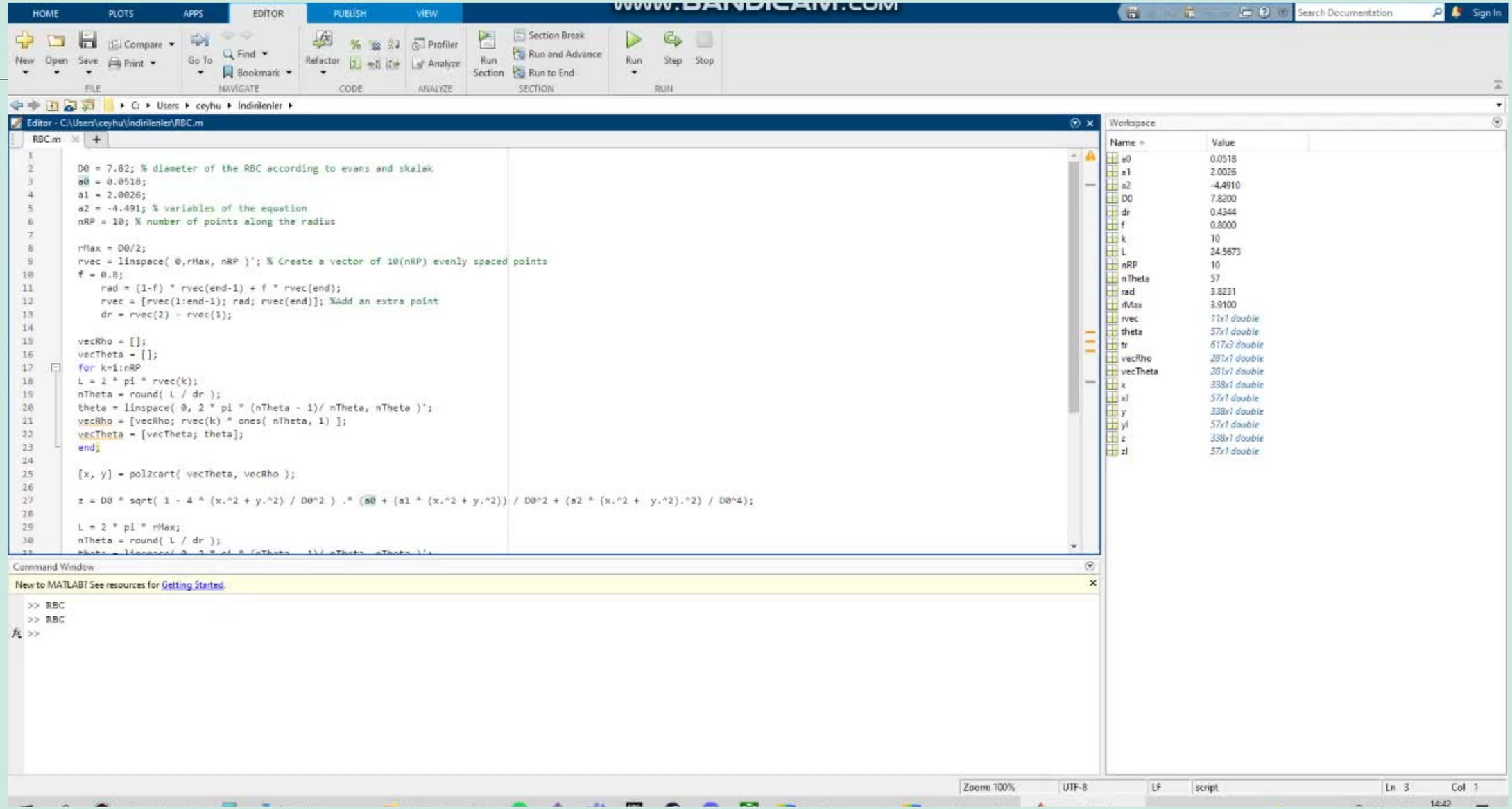
Then we generate a collection of relatively evenly distributed x and y coordinates on a disk in polar coordinates. In order to optimize the surface computation, we add an extra point before the last radius. After adding the extra point, we compute all points except those exactly at the maximum radius:

- rvec : vector of points
- dr : Difference between the points on the circle
- vecRho : Vector of rho
- vecTheta : Vector of theta
- theta : Needed to find the polar coordinates

```
[x, y] = pol2cart( vecTheta, vecRho );

z = D0 * sqrt( 1 - 4 * (x.^2 + y.^2) / D0^2 ) .* (a0 + (a1 * (x.^2 + y.^2)) / D0^2 + (a2 * (x.^2 +  y.^2).^2) / D0^4);

L = 2 * pi * rMax;
nTheta = round( L / dr );
theta = linspace( 0, 2 * pi * (nTheta - 1)/ nTheta, nTheta )';
[xl, yl] = pol2cart(theta, rMax * ones( nTheta, 1));
zl = zeros( nTheta, 1);

x = [x; xl];
y = [y; yl];
z = [z; zl];
tr = delaunay( x, y );

trisurf( tr, x, y, z, 'FaceColor', 'r' );
material dull;
hold on;
trisurf( tr, x, y, -z, 'FaceColor', 'r' );
material dull;
axis off;
axis equal;
```

Then we convert polar coordinates to cartesian coordinates.After that we apply the algorithm to find the z coordinates.At last we compute the maximum radius point.We link the points and perform the triangulation.Finally we plot the 3-D triangular surface based on the points in vectors x, y, z and merge it with the triangular surface of x,y and -z.

Result


Editor - C:\Users\ceyhu\Indirilenler\RBC.m
RBC.m
D0 = 7.82; % diameter of the RBC according to evans and skalak
a0 = 0.0518;
a1 = 2.0026;
a2 = -4.491; % variables of the equation
nRP = 10; % number of points along the radius
rMax = D0/2;
rvec = linspace(0,rMax, nRP)'; % Create a vector of 10(nRP) evenly spaced points
f = 0.8;
rad = (1-f) * rvec(end-1) + f * rvec(end);
rvec = [rvec(1:end-1); rad; rvec(end)]; %Add an extra point
dr = rvec(2) - rvec(1);
vecRho = [];
vecTheta = [];
for k=1:nRP
L = 2 * pi * rvec(k);
nTheta = round(L / dr);
theta = linspace(0, 2 * pi * (nTheta - 1)/ nTheta, nTheta)';
vecRho = [vecRho; rvec(k) * ones(nTheta, 1)];
vecTheta = [vecTheta; theta];
end;
[x, y] = pol2cart(vecTheta, vecRho);
z = D0 * sqrt(1 - 4 * (x.^2 + y.^2) / D0^2) .* (a0 + (a1 * (x.^2 + y.^2)) / D0^2 + (a2 * (x.^2 + y.^2).^2) / D0^4);
L = 2 * pi * rMax;
nTheta = round(L / dr);
Command Window
New to MATLAB? See resources for Getting Started.
>> RBC
>> RBC
>>
Workspace
Name
Value
a0 0.0518
a1 2.0026
a2 -4.4910
D0 7.8200
dr 0.4344
f 0.8000
k 10
L 24.5673
nRP 10
nTheta 57
rad 3.8231
rMax 3.9100
rvec 11x1 double
theta 57x1 double
tr 617x3 double
vecRho 281x1 double
vecTheta 281x1 double
x 338x1 double
xl 57x1 double
y 338x1 double
yl 57x1 double
z 338x1 double
zl 57x1 double

Any Questions?

THANK YOU!