Bio fluid mechanics first project

Provided by:

Amirhossein Mohammadi
Ali Zangenehpour
Mostafa Sanikhani
Saeed Parsa

1. Introduction

The objective of this project was to assess the risk of thrombosis and atherosclerosis in blood vessels by calculating hemodynamic parameters, namely TAWSS, OSI, and HOLMES. Hemodynamic parameters play a crucial role in the measurement of blood pressure, blood flow, and other factors that impact the circulation of blood within the body. Hemodynamic monitoring is used to evaluate cardiovascular function and the response to interventions in critically ill or unstable patients.

TAWSS, or time-averaged wall shear stress, represents the average value of the WSS (wall shear stress) over a cardiac cycle. The WSS is a force exerted by blood flow on the vessel walls. OSI, or oscillatory shear index, quantifies the temporal variation of WSS direction throughout a cardiac cycle. By assessing the reversals in shear stress, OSI helps in understanding the flow patterns that may contribute to the development of thrombosis and atherosclerosis. HOLMES, or highly oscillatory low magnitude shear, combines both the magnitude and direction of the WSS. It is calculated by multiplying TAWSS by (1 - OSI), providing a comprehensive assessment of the hemodynamic forces acting on the blood vessel walls.

Accurately and non-invasively measuring these hemodynamic parameters is a challenge. In this project, we utilized the finite element CFD method to analyze the provided WSS values for 731 time steps, with each time step lasting 0.001 seconds. Unfortunately, we were only provided with the values of WSS at different nodes of one patient's aorta wall. With these limited values, it is not possible to fully analyze and predict the state of thrombosis and atherosclerosis.

To overcome this limitation, we calculated TAWSS, OSI, and HOLMES for each specific node using the available data from the 731 time steps. We employed Python and data science techniques to process the 731 data series, ensuring that each data series

corresponded to one time step. This approach allowed us to perform a node-specific analysis of the hemodynamic parameters, providing valuable insights into the potential risk of thrombosis and atherosclerosis.

This paper aims to present a comprehensive analysis of the calculated hemodynamic parameters and their significance in predicting the risk of thrombosis and atherosclerosis.

2. Method and Data

One of the most important parts of data analysis, data science and machine learning is providing reliable dataset that can also be challenging. In engineering problems these datasets can be extracted from experimental tests or reliable computer simulations. Datasets used for this project are created from a simulation done by Ansys. There are 731 CSV files, which correspond to each time step of the patient's cardiac cycle simulation, which is 0.731 seconds long. These files contain each node's coordinates, wall shear stress along X, Y and Z-axes and also the resultant wall shear stress. However, these datasets must be normalized before being used due to some default indentations and empty columns.

	A	B	C	D	E	F	G	H	I	J	K
1											
2	[Name]										
3	WALL										
4											
5	[Data]										
6	Node Number	X [mm]	Y [mm]	Z [mm]	Wall Shear [k	Wall Shear X [	Wall Shear Y [	Wall Shear Z [	X [mm]	Y [mm]	Z [mm]
7	0	-2.89E+00	5.43E+01	-4.08E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	-2.89E+00	5.43E+01	-4.08E+00
8	1	-2.35E+00	5.30E+01	-4.41E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	-2.35E+00	5.30E+01	-4.41E+00
9	2	-3.47E+00	5.32E+01	-3.60E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	-3.47E+00	5.32E+01	-3.60E+00
10	3	-7.21E-01	6.86E+01	4.95E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	-7.21E-01	6.86E+01	4.95E+00
11	4	6.12E-16	7.00E+01	5.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	6.12E-16	7.00E+01	5.00E+00
12	5	-1.43E+00	7.00E+01	4.79E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	-1.43E+00	7.00E+01	4.79E+00
13	6	4.95E+00	5.50E+01	-6.72E-01	0.00E+00	0.00E+00	0.00E+00	0.00E+00	4.95E+00	5.50E+01	-6.72E-01
14	7	5.00E+00	5.35E+01	1.71E-01	0.00E+00	0.00E+00	0.00E+00	0.00E+00	5.00E+00	5.35E+01	1.71E-01
15	8	4.83E+00	5.35E+01	-1.29E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	4.83E+00	5.35E+01	-1.29E+00
16	9	-1.98E-01	5.41E+01	-5.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	-1.98E-01	5.41E+01	-5.00E+00
17	10	5.25E-01	5.28E+01	-4.97E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	5.25E-01	5.28E+01	-4.97E+00
18	11	-9.49E-01	5.28E+01	-4.91E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	-9.49E-01	5.28E+01	-4.91E+00
19	12	4.90E+00	5.48E+01	9.71E-01	0.00E+00	0.00E+00	0.00E+00	0.00E+00	4.90E+00	5.48E+01	9.71E-01
20	13	4.73E+00	5.34E+01	1.62E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	4.73E+00	5.34E+01	1.62E+00
21	14	-5.29E-01	5.42E+01	4.97E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	-5.29E-01	5.42E+01	4.97E+00
22	15	-1.30E+00	5.29E+01	4.83E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	-1.30E+00	5.29E+01	4.83E+00
23	16	1.69E-01	5.28E+01	5.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.69E-01	5.28E+01	5.00E+00
24	17	-4.99E+00	5.48E+01	-2.66E-01	0.00E+00	0.00E+00	0.00E+00	0.00E+00	-4.99E+00	5.48E+01	-2.66E-01
25	18	-4.90E+00	5.35E+01	-9.88E-01	0.00E+00	0.00E+00	0.00E+00	0.00E+00	-4.90E+00	5.35E+01	-9.88E-01
26	19	-4.98E+00	5.35E+01	4.97E-01	0.00E+00	0.00E+00	0.00E+00	0.00E+00	-4.98E+00	5.35E+01	4.97E-01
27	20	8.41E-01	5.42E+01	4.93E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	8.41E-01	5.42E+01	4.93E+00
28	21	1.63E+00	5.29E+01	4.73E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.63E+00	5.29E+01	4.73E+00
29	22	-1.55E+00	5.42E+01	-4.75E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	-1.55E+00	5.42E+01	-4.75E+00
30	23	1.17E+00	5.42E+01	-4.86E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.17E+00	5.42E+01	-4.86E+00
31	24	1.96E+00	5.29E+01	-4.60E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.96E+00	5.29E+01	-4.60E+00
32	25	-4.82E+00	5.48E+01	1.32E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	-4.82E+00	5.48E+01	1.32E+00
33	26	-4.61E+00	5.34E+01	1.94E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	-4.61E+00	5.34E+01	1.94E+00
34	27	4.50E+00	5.46E+01	-2.18E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	4.50E+00	5.46E+01	-2.18E+00

Figure 1 - Primary dataset

Python programing language and pandas library are used for data manipulation and final results calculations.
The code below shows a function that removes extra, empty and unwanted text from 1st to 5th line.

```
In [7]: import pandas as pd

def normalizeCSV(fileName):
    with open(fileName , "r") as file:
        inputFilelines = file.readlines()
        with open(fileName , "w") as file:
            for line in inputFilelines:
                if line != "\n" and line != "[Name]\n" and line != "WALL\n"
                    file.write(line)
    file.close()

for i in range(732):
    normalizeCSV(f"Wall/export_{i}.csv")
```

The function argument is the name of the CSV file and it iterates through the file and keeps the lines that are not the same as unwanted characters and indentations. This function is called 731 times to normalize each dataset. Therefore they are ready to be analyzed.

	A	B	C	D	E	F	G	H	I	J	K
1	Node Number	X [mm]	Y [mm]	Z [mm]	Wall Shear [k	Wall Shear X	Wall Shear Y	Wall Shear Z [	X [mm]	Y [mm]	Z [mm]
2	0	-2.89E+00	5.43E+01	-4.08E+00	1.90E-03	1.50E-04	1.89E-03	-1.12E-04	-2.89E+00	5.43E+01	-4.08E+00
3	1	-2.35E+00	5.30E+01	-4.41E+00	2.43E-03	9.43E-04	1.98E-03	1.04E-03	-2.35E+00	5.30E+01	-4.41E+00
4	2	-3.47E+00	5.32E+01	-3.60E+00	2.62E-03	1.35E-03	2.07E-03	8.65E-04	-3.47E+00	5.32E+01	-3.60E+00
5	3	-7.21E-01	6.86E+01	4.95E+00	1.30E-03	-2.50E-06	1.30E-03	3.09E-07	-7.21E-01	6.86E+01	4.95E+00
6	4	6.12E-16	7.00E+01	5.00E+00	1.42E-03	4.35E-07	1.42E-03	-1.84E-05	6.12E-16	7.00E+01	5.00E+00
7	5	-1.43E+00	7.00E+01	4.79E+00	1.41E-03	3.24E-06	1.41E-03	-1.81E-05	-1.43E+00	7.00E+01	4.79E+00
8	6	4.95E+00	5.50E+01	-6.72E-01	1.67E-03	1.35E-05	1.67E-03	-4.65E-07	4.95E+00	5.50E+01	-6.72E-01
9	7	5.00E+00	5.35E+01	1.71E-01	2.04E-03	-1.24E-03	1.62E-03	-3.02E-05	5.00E+00	5.35E+01	1.71E-01
10	8	4.83E+00	5.35E+01	-1.29E+00	2.02E-03	-1.19E-03	1.60E-03	2.86E-04	4.83E+00	5.35E+01	-1.29E+00
11	9	-1.98E-01	5.41E+01	-5.00E+00	1.74E-03	1.09E-04	1.73E-03	-1.24E-05	-1.98E-01	5.41E+01	-5.00E+00
12	10	5.25E-01	5.28E+01	-4.97E+00	2.08E-03	7.15E-05	1.76E-03	1.09E-03	5.25E-01	5.28E+01	-4.97E+00
13	11	-9.49E-01	5.28E+01	-4.91E+00	2.22E-03	5.24E-04	1.86E-03	1.10E-03	-9.49E-01	5.28E+01	-4.91E+00
14	12	4.90E+00	5.48E+01	9.71E-01	1.70E-03	-1.66E-06	1.70E-03	6.03E-08	4.90E+00	5.48E+01	9.71E-01
15	13	4.73E+00	5.34E+01	1.62E+00	2.03E-03	-1.17E-03	1.61E-03	-3.82E-04	4.73E+00	5.34E+01	1.62E+00
16	14	-5.29E-01	5.42E+01	4.97E+00	1.74E-03	1.27E-04	1.73E-03	1.90E-05	-5.29E-01	5.42E+01	4.97E+00
17	15	-1.30E+00	5.29E+01	4.83E+00	2.26E-03	6.23E-04	1.88E-03	-1.09E-03	-1.30E+00	5.29E+01	4.83E+00
18	16	1.69E-01	5.28E+01	5.00E+00	2.11E-03	1.78E-04	1.79E-03	-1.11E-03	1.69E-01	5.28E+01	5.00E+00
19	17	-4.99E+00	5.48E+01	-2.66E-01	2.04E-03	-9.33E-06	2.04E-03	-2.57E-05	-4.99E+00	5.48E+01	-2.66E-01
20	18	-4.90E+00	5.35E+01	-9.88E-01	2.87E-03	1.84E-03	2.19E-03	2.58E-04	-4.90E+00	5.35E+01	-9.88E-01
21	19	-4.98E+00	5.35E+01	4.97E-01	2.92E-03	1.89E-03	2.22E-03	-1.42E-04	-4.98E+00	5.35E+01	4.97E-01
22	20	8.41E-01	5.42E+01	4.93E+00	1.69E-03	8.44E-05	1.69E-03	-1.17E-05	8.41E-01	5.42E+01	4.93E+00
23	21	1.63E+00	5.29E+01	4.73E+00	2.00E-03	-2.20E-04	1.69E-03	-1.04E-03	1.63E+00	5.29E+01	4.73E+00
24	22	-1.55E+00	5.42E+01	-4.75E+00	1.79E-03	1.48E-04	1.78E-03	-5.37E-05	-1.55E+00	5.42E+01	-4.75E+00
25	23	1.17E+00	5.42E+01	-4.86E+00	1.69E-03	6.47E-05	1.68E-03	1.26E-05	1.17E+00	5.42E+01	-4.86E+00
26	24	1.96E+00	5.29E+01	-4.60E+00	1.97E-03	-2.99E-04	1.67E-03	1.01E-03	1.96E+00	5.29E+01	-4.60E+00
27	25	-4.82E+00	5.48E+01	1.32E+00	2.05E-03	7.46E-06	2.04E-03	6.69E-05	-4.82E+00	5.48E+01	1.32E+00
28	26	-4.61E+00	5.34E+01	1.94E+00	2.79E-03	1.74E-03	2.14E-03	-4.76E-04	-4.61E+00	5.34E+01	1.94E+00
29	27	4.50E+00	5.46E+01	-2.18E+00	1.71E-03	8.89E-06	1.71E-03	-2.30E-06	4.50E+00	5.46E+01	-2.18E+00
30	28	4.24E+00	5.33E+01	-2.66E+00	1.97E-03	-9.74E-04	1.59E-03	6.34E-04	4.24E+00	5.33E+01	-2.66E+00
31	29	-4.68E+00	5.49E+01	-1.75E+00	1.98E-03	7.91E-06	1.98E-03	-7.84E-05	-4.68E+00	5.49E+01	-1.75E+00
32	30	-4.39E+00	5.34E+01	-2.39E+00	2.78E-03	1.69E-03	2.13E-03	5.57E-04	-4.39E+00	5.34E+01	-2.39E+00
33	31	-2.00E+00	5.41E+01	4.58E+00	1.85E-03	1.57E-04	1.84E-03	7.40E-05	-2.00E+00	5.41E+01	4.58E+00

Figure 2 - Normalized dataset

One of the demands of the problem is calculation of Time Average Wall Shear Stress (TAWSS) for each node. The integral form of TAWSS is presented in the project proposal. However, integration is only possible in continuous mathematics when there is a continuous function. In this case datasets contain discrete data. Hence, ' Σ ' symbol is used instead of ' $\int$ ' symbol. Thus the TAWSS equation is as follows.

$$TAWSS = \frac{1}{T}\int_0^T |\overrightarrow{wss}|dt \xrightarrow{\text{Due to discrete data}} TAWSS = \frac{1}{T}\sum |\overrightarrow{wss}|\,\Delta t$$

Then a dictionary is defined in python called wss that contains all the wall shears as the keys and 2 dimensional arrays as values that contain every node's corresponded wall shear stress from every dataset (every related column is extracted).

```
In [8]: wss = {"Wall Shear" : [] , "Wall Shear x" : [] , "Wall Shear y" : [] , "Wal
        for i in range(732):
            df = pd.read_csv(f"Wall/export_{i}.csv")
            wss["Wall Shear"].append(df[" Wall Shear [ kPa ]"][0:2888].values)
            wss["Wall Shear x"].append(df[" Wall Shear X [ kPa ]"][0:2888].values)
            wss["Wall Shear y"].append(df[" Wall Shear Y [ kPa ]"][0:2888].values)
            wss["Wall Shear z"].append(df[" Wall Shear Z [ kPa ]"][0:2888].values)
        wss
```

```
Out[8]: {'Wall Shear': [array([0., 0., 0., ..., 0., 0., 0.]),
          array([0.00190247, 0.00243131, 0.00262095, ..., 0.00014239, 0.00014447,
                 0.00014359]),
          array([0.00268859, 0.0034825 , 0.0037652 , ..., 0.00022601, 0.00022778,
                 0.00022756]),
          array([0.00325921, 0.00434161, 0.00471621, ..., 0.00028548, 0.00028699,
                 0.00028736]),
          array([0.0037748 , 0.00518851, 0.0056605 , ..., 0.00033424, 0.00033572,
                 0.00033652]),
          array([0.00425341, 0.00602506, 0.00659622, ..., 0.00037684, 0.00037834,
                 0.00037947]),
          array([0.00469633, 0.00685295, 0.00752378, ..., 0.00041541, 0.00041692,
                 0.00041829]),
          array([0.00510928, 0.00768675, 0.00845784, ..., 0.00045116, 0.00045268,
                 0.00045426]),
          array([0.00549961, 0.00854297, 0.00941557, ..., 0.00048489, 0.00048643,
                 0.00048821]),
          array([0.00586859, 0.00942354, 0.01039625, ..., 0.00051688, 0.00051841,
                 0.0005204 ]),
```

In the next step a function is defined to calculate the TAWSS value based on the name argument (wall shear, wall shear x, wall shear y or wall shear z) and the node number (0 to 2887).

In [9]:
```
def tawssCalculation(name , nodeNumber):
    result = 0
    for i in range(732):
        result += wss[name][i][nodeNumber]
    return result / 731
```

The other demands of the problem are OSI and HOLMES values that are explained in the proposal. As mentioned before, the integral form in the OSI formula is converted to the summation of discrete data. The code below is written based on simple iteration and summation algorithms to calculate TAWSS, OSI and HOLMES values of each node. These values are saved as dictionary variables with the related node as the key.

In [10]:
```
tawssValues = {}
osiValues = {}
holmesValues = {}
for nodeIndex in range(2888):
    tawssValues[f"Node {nodeIndex}"] = tawssCalculation("Wall Shear" , node

    wss_x = tawssCalculation("Wall Shear x" , nodeIndex)
    wss_y = tawssCalculation("Wall Shear y" , nodeIndex)
    wss_z = tawssCalculation("Wall Shear z" , nodeIndex)

    osiValues[f"Node {nodeIndex}"] = (1 - ((wss_x ** 2 + wss_y ** 2 + wss_z

    holmesValues[f"Node {nodeIndex}"] = tawssValues[f"Node {nodeIndex}"] *
```

These dictionaries are used to make the final data frame and CSV file as the output of the problem.

3. Results and Conclusion

In conclusion, this paper highlights the significance of hemodynamic parameters in assessing the risk of thrombosis and atherosclerosis. By calculating TAWSS, OSI, and HOLMES for each node of a patient's aorta wall using finite element CFD analysis and data science techniques, valuable insights can be gained. The limitations and challenges faced in this project underscore the need for further research and advancements in non-

invasive measurement techniques to enhance the accuracy and reliability of hemodynamic parameter assessment.

Finally all three dictionaries are converted to a data frame and also written into a new CSV file called output.csv.

In [24]:
```
output = pd.DataFrame({"TAWSS" : tawssValues.values() , "OSI" : osiValues.v
index = []
for nodeIndex in range(len(wss["Wall Shear"][0])):
    index.append(f"Node {nodeIndex}")
output.index = index
output
```

Out[24]:

	TAWSS	OSI	HOLMES
Node 0	0.005596	0.491104	0.000050
Node 1	0.011177	0.161577	0.003782
Node 2	0.011296	0.146239	0.003996
Node 3	0.005166	0.212339	0.001486
Node 4	0.006912	0.170655	0.002276
...	...	...	...
Node 2883	0.000674	0.000031	0.000337
Node 2884	0.000683	0.000005	0.000341
Node 2885	0.000689	0.000002	0.000344
Node 2886	0.000659	0.000006	0.000329
Node 2887	0.000604	0.002403	0.000301

2888 rows × 3 columns

In [29]:
```
outputFile = open("output.csv" , "a")
output.to_csv(outputFile)
outputFile.close()
```

The notebook of this project is attached as both Jupyter (ipynb) and PDF files alongside this article. Also the python code and output.csv are attached.

Genius is 1 percent inspiration and 99 percent perspiration.
-Thomas Edison-