Reduced order modeling approach for cardiovascular stent design

Berkin Dortdivanlioglu

I. Introduction

A stent is a medical device that holds an artery open and is left inside the artery permanently. When an artery feeding the heart muscles narrows down due to plaque formation, the blood flow is interrupted. This interruption results in chest pain. Further closure of artery preventing blood flow to heart muscles causes heart attack. Today, the disease atherosclerosis (narrowing of arteries) is the leading cause of 52% of deaths and disabilities in Europe and the USA [5].

Stents have been used to reopen the arteries where a plaque formation is observed. Furthermore, structural optimization of these devices greatly improved the stent performance [1]. However, it is difficult, if possible, to do *in-vivo* experiments and testing of a stent. Hence, computational methods such as Finite Element Analysis (FEA) are currently preferred tools to experiment and design the stent structure *in-silico*, and predict its response according to the surrounding mechanical conditions. However, FEA involves solving differential equations at many grid points within a computational domain and this proves to be a computationally expensive task.

Computational modeling of medical devices such as stents provides a tool to evaluate the device under the surrounding mechanical loading during and after the implantation. Furthermore, it can predict the failure location within the device geometry and/or on the artery. Particularly, during a medical operation, a fast and efficient evaluation of these concepts are vital as the performance and damages on the device can signal requirement of further operations and, replacement of the device in some cases. To date, the computational cost of full models (FM) of FEA of stent structure hinders development of real-time, online and patient specific applications [6, 3]. Previous studies have primarily concentrated on reducing the computational time through simplifications of the model and the geometry [7]. A recent work by [6] proposes a method based on spring-mass models to simulate virtual stent deployment in real-time. However, not only the calibration of spring constants with FEA is not straightforward, but their model also has large mean error up to 10%. Therefore, to the best of my knowledge, no work that links the FEA with machine learning tools to provide a framework for real-time analysis of complex stent designs has been proposed.

The objective of this work includes two novel studies. The first part is the prediction of structural failure of a design performance using supervised learning algorithms. The second part is the prediction of failure locations of a stent structure with the ability of real time analysis through reduced order modeling (ROM) of the full FEA, making use of the unsupervised learning algorithms.

II. Part I: Predicting structural failure

A. Features & Dataset

The features for the problem are material parameters (modulus of elasticity, Poisson's ratio, Yield strength), geometry properties (thickness, and weights for non-uniform rational basis splines' (NURBS) control points), design specifications (factor of safety) and loading conditions.

Material properties: Ni-45Ti (Nitinol), used in stent manufacturing, is chosen as the specific material from the *ASM Medical Materials Database* and the information on Modulus of elasticity E, Poisson's ratio ν, and Yield strength (σ_Y) obtained through the same database (Table. I).

Geometry properties: Following the work by [1], NURBS formulation, commonly used by Computer-Aided Design software, is adopted to generate the stent geometry. By modifying the weights of three points (control points) w_1, w_2, and w_3 (see Fig. 1), we can form different stent geometries. In addition, the parameter t_{stent} determines the thickness of the stent.

TABLE I: Features

Modulus of Elasticity E	4.32 - 11.2 $\times 10^6$ psi
Poisson's ratio ν	0.33
Yield Strength σ_Y	54.6 - 184 ksi
weight of first point w_1	0.01 - 360
weight of second point w_2	0.01 - 360
weight of third point w_3	0.01 - 360
thickness t_{stent}	0.02 - 0.08 mm
factor of safety FS	0.5 - 1
crimp - expansion deflection u_x	-0.1 - 0.1 mm
stretching - bending deflection u_y	-0.1 - 0.1 mm

For the computational modeling, we use isogeometric analysis [2] with an elastic large deformation material formulation to account for large deformations of the stent structure. The use of elastic material formulation is a restriction of the model in this work since the plasticity may play an important role in the failure. However, the framework created in this work is easily extendable to plastic models. As a failure indication, we use the Von-Mises stress σ_{VM} criterion to be bounded by the yield strength of the material σ_Y, scaled by factor of safety FS, i.e. $\sigma_Y \times FS$.

We account for two loading cases;

- crimp - expansion deflection (u_x) during deflation-inflation balloon,
- stretching - bending deflection (u_y) during implantation.

Dataset generation for training: Given the feature vector $x^T = [E, \nu, \sigma_Y, w_1, w_2, w_3, t_{stent}, FS, u_x, u_y]$, we simulate

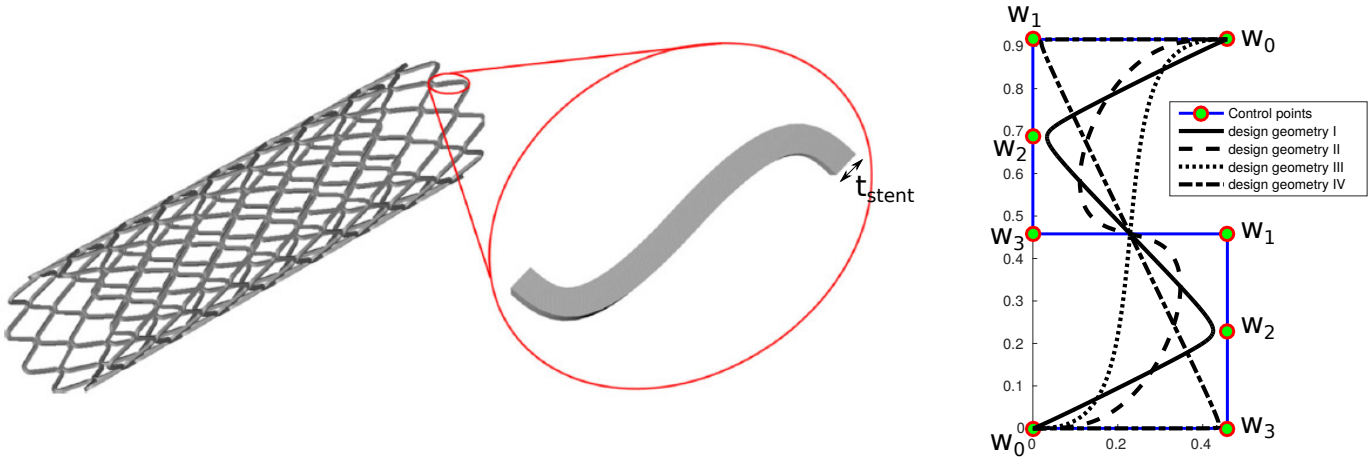


Fig. 1: *Left:* the whole geometry of a stent taken from [1]. Making use of symmetry and periodicity of the structure, we use a single segment (inside red circle) for full model analysis. *Right:* few potential design geometries by varying the weights of the control points.

the stent model and compute σ_{VM} at each point (gauss points) in the design domain. This creates a very large matrix to analyze. However, in the first part of this project, we are only interested in the maximum magnitude of the values in this matrix since this information is enough to determine if the structure failed. By analyzing this information, we determine if the stent fails or not under the loading conditions;

$$y = \begin{cases} 1 \text{ (safe)} & \sigma_{VM,max} < \sigma_Y \times FS \\ 0 \text{ (fail)} & otherwise. \end{cases} \tag{1}$$

Since we simplified the analysis by designing just a single segment (Fig. 1) (due to time consideration of this project), we generated a dataset with a large number of examples for the first part. We saved each example $\{(x^{(i)}, y^{(i)}); i = 1, \ldots, m\}$ in rows of the matrix as the output of simulations.

B. Methods

In this section, we list the supervised learning algorithms (binary classifications) used in the first part of the project (documentation on algorithms [4]).

- Logistic Regression (LR)
- Naive Bayes (NB)
- Support Vector Machine (SVM)
- SVM with kernels such as Gaussion/Radial Basis Function (RBF) and polynomial function with different polynomial orders.

The kernels used are defined by

$$\begin{aligned} K_{rbf}(x, z) &= \exp\big(-\frac{1}{2\tau^2}||x - z||_2^2\big) \\ K_{poly}(x, z) &= \big(1 + x^T z\big)^p, \end{aligned} \tag{2}$$

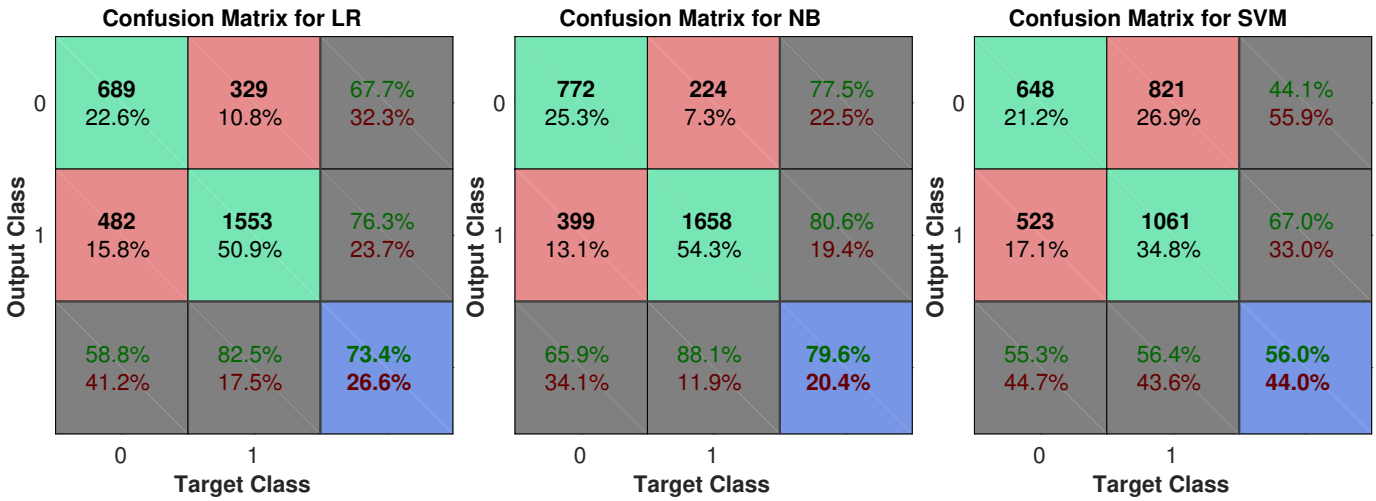
where p is the polynomial order and τ is the bandwidth of kernel function.

C. Experiments/Results

Using the feature vector $x^T = [E, \nu, \sigma_Y, w_1, w_2, w_3, t_{stent}, FS, u_x, u_y]$, we generated a dataset with $m = 10175$ examples. Unless otherwise stated, we used the whole dataset for both training (70% of the data) and testing (30% of the data).

First, we investigate the learning curve of our learning algorithm using LR. We compute the test error for changing example set sizes (see Fig. 3). Fig. 3 shows the learning curve for LR with a threshold of 0.5. We obtain a test error $\approx 26\%$, which is unexpectedly high than the desired performance. Furthermore, the gap between test error and the training error is vanishing as the size of training set is increased. In addition, both the training and test error are reaching a plateau. This also suggests that the model exhibits a high bias, and we are clearly underfitting the data. It is important to note that, in the preliminary results, we had a similar trend in the learning curve obtained using less number of features. Although, in the final report, we tried to eliminate this problem by increasing our feature vector, the results show no improvements over the preliminary results. A plausible explanation for these findings is that the data is noisy and/or it is not linear.

Secondly, we used LR, SVM and NB classification algorithms and compute the Receiver Operating Characteristic Curve (ROC) for each method. This method enables us to summarize the performance of the algorithms for varying thresholds over a range of trade-offs between true positive and false positive error rates. By looking at the trend and the area under the curves, we can comment on the prediction accuracy of the methods. Specifically, that the curve has a high slope indicates the increase in the number of correctly determined safe structures (as safe) without making wrong prediction of failed structures (as safe). Fig. 4 shows that all the methods predicts the results better than random guessing. Surprisingly, the accuracy of NB proves to be better than other methods including SVM. One important observation is that the test errors are $\approx$ 19%, 26% and 43% for NB, LR and SVM, respectively. (These values obtained by a model selection procedure where we find the minimum test error by conducting hold-out cross validation (30% of data is used for cross validation) for 10 times shuffled data). We further



(a) Confusion matrix for LR (b) Confusion matrix for NB (c) Confusion matrix for SVM

Fig. 2: Confusion matrices for LR,SVM and NB. The rows show the predicted label and the columns show the true labels.The left bottom entries of the matrices shows the overall performance of the model.

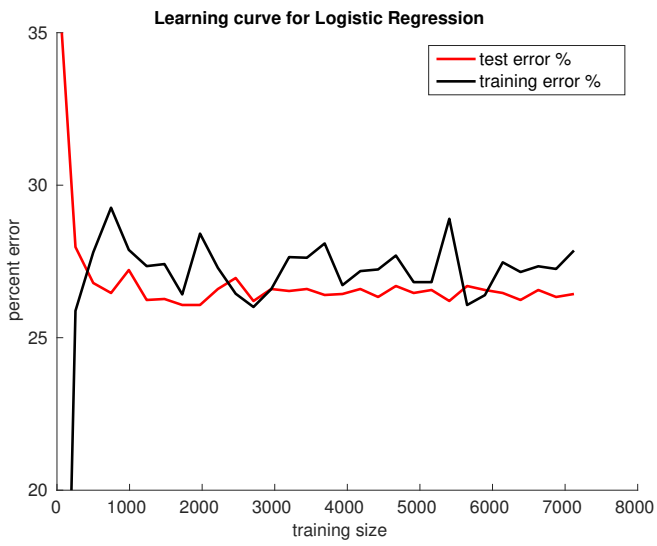


Fig. 3: Graph of test/training error for increasing size of example set. Firstly, the learning algorithm test error $\approx 26\%$, which is unacceptable high and also the training error is very high, which is well above desired performance. We can make several inferences on bias/variance trade-off of our model by looking at this curve and a typical learning curve for high bias (see lecture notes). This tells us we have high bias in our learning algorithm. We explain this trend by the fact that the data is noisy and/or it is not linear.

compare the confusion matrices for each model, as illustrated in Figure. 2. Although NB has an improved performance overall, it still has a very high test error ($\approx 20\%$).

Observing that the linear classifiers work unsatisfactory, we try one more attempt to find a better model by using SVM with kernels such as Gaussion/Radial Basis Function (RBF) and polynomial function with different polynomial orders. The confusion matrix (5) shows that, in fact, the SVM with nonlinear features produced good results at predicting

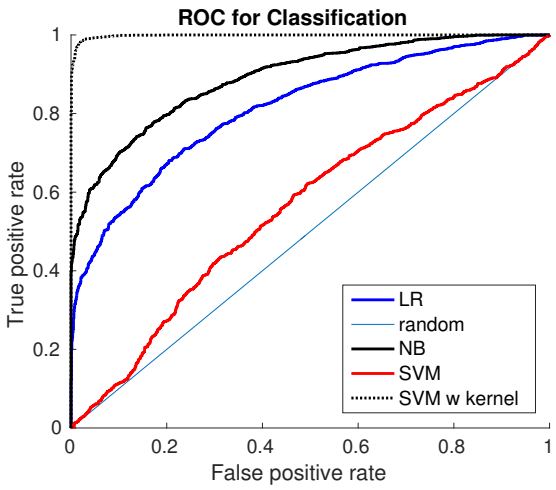


Fig. 4: ROC curves for NB, SVM and LR compared with random guessing.

the structural failure of stents. Table II illustrates the results for RBF and polynomial functions of different degree. Several inferences can be made by investigating Table II.

TABLE II: SVM with kernel results

kernel function	parameters	training error %	test error %
RBF	$\tau = 1$	0.0421	10.8090
Polynomial	$p = 2$	4.2685	4.9787
Polynomial	$p = 3$	0.8705	**1.8015**
Polynomial	$p = 4$	0	4.8149

Firstly, the model with nonlinear features improved the results significantly, suggesting that the data linearly not separable. Secondly, for the polynomial function, the training error is reduced as the polynomial order is increased ($p = 2, 3, 4$). However, we can clearly observe the bias/variance trade off in the results for polynomial functions. Quadratic polynomial

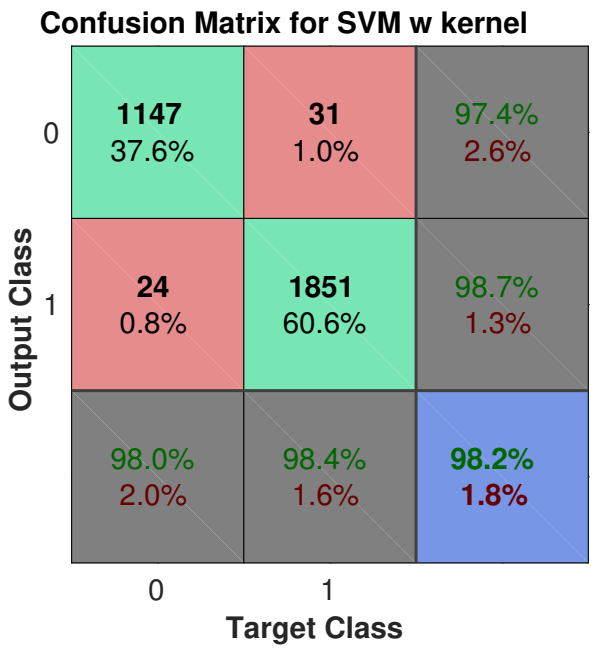


Fig. 5: Confusion matrix obtained by SVM with polynomial kernel function of order 3.

function has higher bias compared with cubic and quartic functions whereas quartic function has higher variance. we can conclude that the cubic polynomial function obtains a test error of 1.8% and does better than second and fourth degree polynomials. Furthermore, the confusion matrix for SVM with cubic polynomial function (5) shows that only 0.8% of the failing structures are labeled as safe, and 1% of safe structures are predicted to fail. Overall, our model can predict with an accuracy of $\geq 98\%$.

III. Part II: Reduced Order Modeling (ROM)

In *Part I*, we presented our attempt to predict the stent performance under environmental loading using binary classification algorithms. Now, we are also interested in prediction of failure locations at the overall structure since this plays a vital role for real-time monitoring of the stent during and after clinical operations. Notice that the need for a reduced order model approach emerges due to the high computational cost of full models. Hence, the purpose of ROM is to significantly lower the computational cost of numerical simulations, enabling us to provide real-time analysis results. As a primary step, we need a full scale analysis tools (in this project, I will use my in-house developed IGA (finite element analysis tool) code). The IGA code takes parameters $[E, \nu, w_1, w_2, w_3, t_{stent}, u_x, u_y]$ as input and forms the the stiffness matrix $\mathbf{K} \in \mathbb{R}^{n\times n}$ and force vector $\mathbf{f} \in \mathbb{R}^n$ of the system. Then, it solves the system for the deformation $\mathbf{d} \in \mathbb{R}^n$ (displacement field) using $\mathbf{K}\ \mathbf{d} = \mathbf{f}$. Herein, $\mathbf{K}$ is positive definite, banded and symmetric. The computational cost for the solution of this system constitutes an important part in the overall finite element analysis. Note that the size of the system can be $n \approx 1000$ to 100000, even the systems with $n \approx 10^6$ are common.

A. Features & Dataset

Different from *Part 1*, to generate the dataset, the displacement field $\mathbf{d}$ at every control point in our computational domain mesh is saved in a vector called *snapshot*. Each full model simulation requires a parameter set as input (from the training set), then it outputs the *snapshots*. We repeat this procedure as many as our sampling points S. Then, all snapshots are saved in one global snapshot matrix. Note that the dataset has been generated prior to reduced order modeling (off-line).

B. Methods

In this section, we summarize the steps to create a ROM. By k–times *random* sampling from our parameter space (training set), we save each *snaphot* into the global snapshot matrix obtained from full model simulations. Using PCA, we obtain the first k principle components of the model. Basically, we compute the SVD of the global snapshot matrix. These components form a k–dimensional subspace ($\mathbf{V} \in \mathbb{R}^{n\times k}$ and $k <<< n$). Now, we are interested in the solution of a much smaller system $\bar{\mathbf{K}}\ \bar{\mathbf{d}} = \bar{\mathbf{f}}$, where $\bar{\mathbf{K}} = \mathbf{V}^T\mathbf{K}\mathbf{V}$, $\bar{\mathbf{K}} \in \mathbb{R}^{k\times k}$, $\bar{\mathbf{f}} = \mathbf{V}^T\mathbf{f}$, $\bar{\mathbf{f}} \in \mathbb{R}^k$ and, also $\bar{\mathbf{d}} \in \mathbb{R}^k$.

Finally, projecting the reduced solution $\bar{\mathbf{d}}$ back to initial space $\mathbf{d} = \mathbf{V}\bar{\mathbf{d}}$, we predict the full model solution with a significantly lowered computational cost.

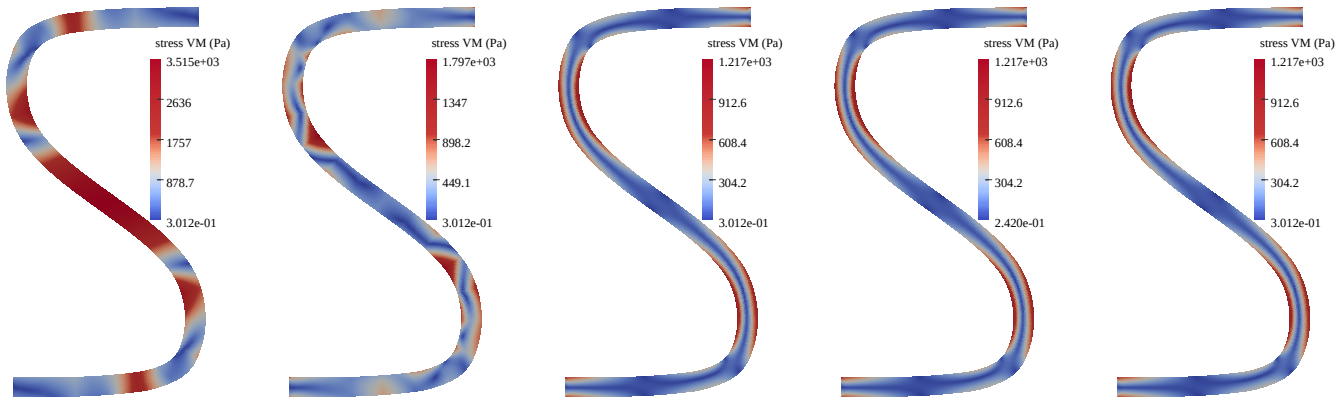
C. Experiments & Results

Initially, to perform error analysis, we randomly select grid points in our parameter space and ran the full model simulation. Hence, we have the exact displacement field $\mathbf{d}$ for these randomly selected parameters, i.e. $[E, \nu, w_1, w_2, w_3, t_{stent}, u_x, u_y]$. Then, we conduct several detailed studies to test the performance of our ROM. First, we vary the sampling size S, and compute the error for each test grid point using

$$E = \frac{||\mathbf{d} - \mathbf{V}\bar{\mathbf{d}}||_2^2}{||\mathbf{d}||_2^2} \ . \tag{3}$$

We repeat these computations several times and compute the maximum and average error for every test and grid points. Both the maximum and the average for each test grid point exhibit a decaying performance as we increase the sampling size, as shown in Fig. 7. We explain this expected behavior as, introducing more computational points (information) for the PCA from the parameter space enhances the reduced order basis $\mathbf{V}$, leading to an improved error performance. Figure 6 illustrates that the solutions obtained using ROMs with $S = 5$ and $S = 10$ not only overestimates Von Mises stress values compared to the true solution, but they also fail to predict the location of the failure. By investigating the Fig.7, the sampling size should be greater than $S > 40$ for a good prediction of the problem at hand.

To compare computational cost, we only focus on the solver subroutine of the overall analysis. The solution time for a full model is normalized to unity, whereas it takes 10^{-3} for a reduced order model to solve the system. In other words, if it takes one minute to solve the system for a full model, we



(a) S=5 (b) S=10 (c) S=50 (d) S=100 (e) Full model

Fig. 6: The contour plots of the Von Mises stress field obtained using ROMs and the full model. The high Von Mises stress indicates the locations that the structure is prone to failure.

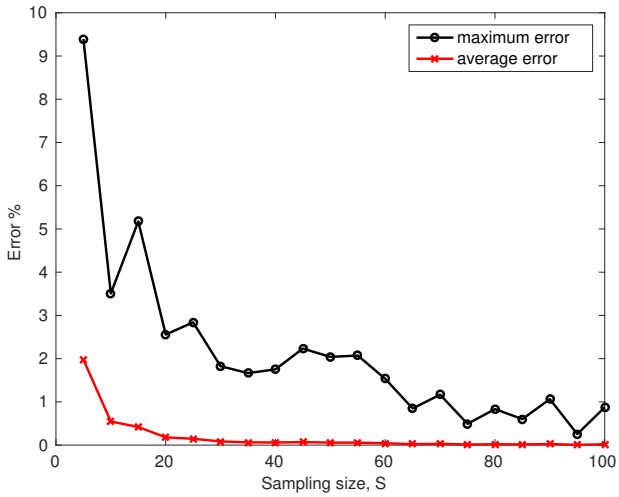
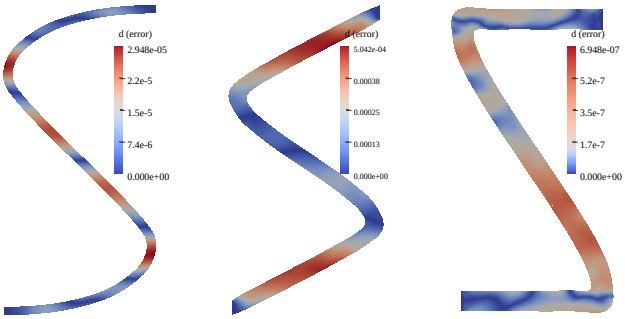


Fig. 7: Increasing the number of samples S, we observe that the mean error is decaying. Satisfactory results are obtained for $S > 30$.



(a) Design 1 (b) Design 2 (c) Design 3

Fig. 8: The contour plots of the error between the solutions $\mathbf{d}$ and $\bar{\mathbf{d}}$ for different parameters. The absolute error is $\approx 10^{-7}$ to 10^{-4}.

obtain the solution in less than 0.1 sec. using ROM, which paves the way for real time analysis.

Lastly, contours of error in the solution fields, i.e. $\mathbf{d}$ and $\bar{\mathbf{d}}$, are shown in Fig. 8 for three different designs (sampling size $S = 100$). It is observed that the ROM does better for some designs, for example *Design 3* in Fig.8 compared to *Design 1 & 2*. We tried to understand the reason by looking at the samples used in the Principle Component Analysis. We observed that the *random* sampling produces self-similar geometries for the stent structure. Although this problem could be eliminated by increasing the sampling size, it is not desired due to the computational cost of the full model simulations. Hence, another solution would be better sampling algorithms such as *latin-hypercube* sampling.

IV. Conclusion

In this project, we first presented our attempt to predict the structural failure of stents, and later on, we also proposed an efficient and accurate framework for realtime analysis of stent deployment. For the prediction of structural failure, we obtained the best result using SVM with third order polynomial function kernel ($> 98\%$). In the second part, the model based on PCA provided almost as accurate results as the full model simulation for $k > 30$. Furthermore, the computational time is significantly reduced using the ROM framework proposed in this study.

V. Future work

For *Part I*, a detailed parameter study is required. We expect the results to improve after a proper parameter study of τ bandwidth in SVM with Gaussian function kernel. In *Part II*, we will implement different sampling algorithms to better span the design space and reduce the maximum error in the model. Of course, in this project, the mechanical loading and application of the loading on the domain includes many simplifications, that needs to be improved in the future works through a collaboration with relevant disciplines. Furthermore, a 3D geometry with the full stent structure will be considered to account for local and global instabilities in the structure, and also the curvature effects in the model.

REFERENCES

[1] Rory Clune, Denis Kelliher, James C. Robinson, and John S. Campbell. Nurbs modeling and structural shape optimization of cardiovascular stents. *Structural and Multidisciplinary Optimization*, 50(1):159–168, 2014.

[2] T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs. Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194(3941):4135 – 4195, 2005.

[3] Kumaran Kolandaivelu, Caroline C. O'Brien, Tarek Shazly, Elazer R. Edelman, and Vijaya B. Kolachalama. Enhancing physiologic simulations using supervised learning on coarse mesh solutions. *Journal of The Royal Society Interface*, 12(104), 2015.

[4] MATLAB. Stats toolbox. https://www.mathworks.com/help/stats/.

[5] Awani Rai, Suresh Chandra, Shashi Singh, and Asfa Parveen. Atherosclerosis: a life changing phenomenon. *Pharmaceutical and Biological Evaluations*, 3(2):154–164, 2016.

[6] K Spranger, C Capelli, G M Bosi, S Schievano, and Y Ventikos. Comparison and calibration of a real-time virtual stenting algorithm using Finite Element Analysis and Genetic Algorithms. *Computer Methods in Applied Mechanics and Engineering*, 293:462–480, aug 2015.

[7] L. Flrez Valencia, J. Montagnat, and M. Orkisz. 3d models forvascular lumen segmentation inmra images andforartery-stenting simulation. {*IRBM*}, 28(2):65 – 71, 2007.