Gaussian Process for Aerodynamic Pressures Prediction in Fast Fluid Structure Interaction Simulations

Ankit Chiplunkar, Elisa Bosco, Joseph Morlier

12th World Congress on Structural and Multidisciplinary Optimization
5th - 9th, June 2017, Braunschweig, Germany

Gaussian Process for Aerodynamic Pressures Prediction in Fast Fluid Structure Interaction Simulations

Ankit Chiplunkar[1], Elisa Bosco[2], Joseph Morlier[3]

[1] Airbus Operations S.A.S., Toulouse, 31060, France, ankit_chiplunkar@airbus.com
[2] Airbus Operations S.A.S., Toulouse, 31060, France, elisa_bosco@airbus.com
[3] Université de Toulouse, CNRS, ISAE-SUPAERO, Institut Clément Ader (ICA), Toulouse, 31077, France, joseph_morlier@isae-supaero.fr

Abstract
The interaction between inertial, elastic and aerodynamic forces for structures subjected to a fluid flow may cause unstable coupled vibrations that can endanger the structure itself. Predicting these interactions is a time consuming but crucial task in an aircraft design process. In order to reduce the computational time surrogate reduced order models can be used in both structural and aerodynamic models. More over it is possible to avoid launching CFD computations at every time step. A database of aerodynamic pressure distribution on the structural component can be created conveniently sampling the space of the structural model DoF. Starting from the knowledge of the pre-computed data-set a Gaussian Process can be applied to predict the pressure distribution on an unexplored point of the space of DoF. The knowledge of the standard deviation can be used to give indications on where to launch further CFD computations to enrich the database. This technique will be first applied to a database of pressures obtained using the software Xfoil®, later it will be applied to CFD simulations of type RANS launched with elsA® on one Flap track Fairing of an Airbus aircraft.
Keywords: Fluid Structure Interaction, Aerospace, Interpolation, Gaussian Process, Design Of Experiments

1 Introduction

A fluid-structure interaction problem requires the solution of the coupled equations of both the fluid and the structure. This can be computationally highly intensive and extremely time-costly depending on the complexity of the phenomena to be predicted[1]. Such is the case for structures such as the flap track fairing (FTF). Their behavior is geometrically highly nonlinear and depends on the source of excitation. FTFs withhold the mechanisms of deployment of flaps and their shape is conceived not to degrade wing performances. Depending on their position, FTFs can be exposed to high vibrations induced from the engine exhaust for short periods of time during take-off. Failure to predict these vibration problems in the design phase causes the necessity for reinforcement and requires servicing actions with aircraft grounding for repair or replacement of the structure. There is therefore a strong need to develop a method of load prediction that maximizes time efficiency. In the frame of transient fluid structure interaction, at each time step the pressure distribution acting on the structural body has to be recomputed updating the position of the component in the CFD simulation domain.

This can be done either launching a CFD simulation at every time-step or interpolating over a database of precomputed pressure distributions. The second case has the advantage of being a lot faster but is based on the assumptions of the displacement envelope of the component to be analyzed and that the deformations are very small. Interpolating between discrete data points is called as regression in Machine learning literature or Surrogate Modelling in engineering literature. Earlier techniques such as polynomial fitting eg. linear interpolation, quadratic interpolation or splines try to fit a polynomial function on the discrete simulated data-points. [2]. In this paper we will compare two methodologies of performing regression/surrogate modelling. One well tested method is to reduce the dimension of database and rebuild the pressure distribution at a desired point. We would use, Singular Value Decomposition (SVD) to reduce the database and spline to interpolate the reduced database. A second methodology is to build a surrogate model on the whole dataset without performing data reduction, a Gaussian Process or Kriging will be used to interpolate between all data-points. The main advantage of Gaussian Process is that it allows the use of initial information such as smoothness and error distribution into the model in analysis. These kind of assumptions are applicable to physical systems since it is known before hand that the pressure distribution is highly smooth and not necessarily linear or cubic in nature for low Mach numbers.The efficiency of the Gaussian Process will be finally compared to spline interpolation.

The remaining paper is divided as follows; section 2 describes various methods of calculating pressure distributions on a given structure under specified aerodynamic conditions, section 3 explains the basic principles of model

reduction through the Singular Value Decomposition technique, section 4 gives the formulation of gaussian process and how to launch it in presence of large number of datapoints. Section 5 compares the results of the Gaussian methodology with respect to earlier techniques. Finally, section 6 concludes the paper with current understanding and future remarks.

2 Pressure database

The way of building the pressure database depends on the number of degrees of freedom (DoF) activated on the component under analysis. In both the aerodynamic models described in this paper the component will have two degrees of freedom, therefore the database will be three dimensional (third dimension representing pressure).

2.1 Xfoil - A simple example

Xfoil® is a program that can be used for the analysis of subsonic isolated airfoils[3]. Various parameters can be changed simultaneously such as properties of the flow or geometric properties of the airfoil itself. Inviscid analysis of the airfoil NACA 3320 is launched for varying angle of attach and Mach number. The space of variables is sampled as in the following vectors shown in figure 1, angle of attack (α) is expressed in [°].

$$[\alpha] = [-2,-1,0,1,2]; [M] = [0.125, 0.25, 0.375, 0.4]; \tag{1}$$

The C_p distribution for a fixed Mach number and varying angle of attack (α) can be seen in the following figure 1.

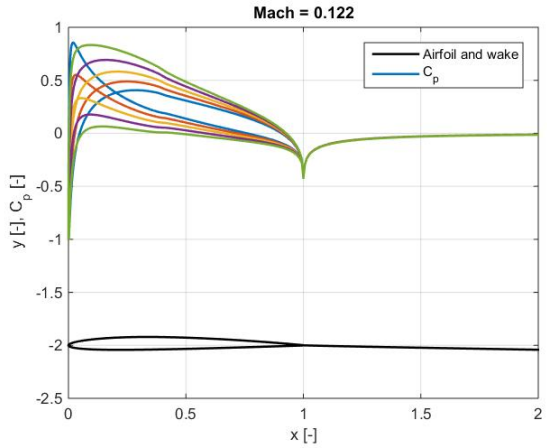


Figure 1: Pressure distribution along the chord of NACA 3320 airfoil. The solid black line at the bottom represents the shape of the airfoil. The colored lines on the top represent various pressure distribution on this airfoil upon varying α and Mach(M) independently.

The assembled database will have the dimensions specified in *table 1*.

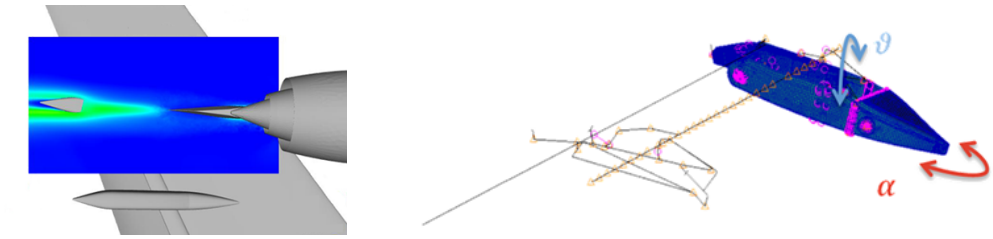
Table 1: Xfoil Pressure Database

	N°[-]
Xfoil Nodes	182
Sampling in α	5
Sampling in Mach	4

The database so obtained has very limited dimensions, allowing fast computations for testing the method, while having the suitable characteristics for this application.

2.2 elsA RANS - Application to a Flap Tack Fairing

elsA® [4] is a pluri-function CFD simulation platform that allows representation of internal and external aerodynamics from the low subsonic to the high supersonic flow regime. Several formulations of the 3D Navier-Stokes equations can be chosen for arbitrary moving bodies. In this particular case Reynolds Averaged Navier-Stokes, *RANS*, equations are used with Spalart-Allmaras turbulence model to simulate the flow around the flap track fairing.



(a) elsA - Reynolds Average Navier-Stokes

(b) Flap Track Fairing DoF

Figure 2: Details of the Flap track Fairing

The flap track fairing DoF chosen for this analysis are θ, the rotation around the longitudinal axis of the flap track fairing itself, and α the rotation around the *spigot* axis, main connection between the track and the wing structure figure 2(b).

The space of variable is regularly sampled and the assembled database will have the dimensions specified in *table 2*.

Table 2: RANS Pressure Database

	N°[-]
RANS Nodes	36802
Sampling in α	9
Sampling in θ	9

For a fixed value of θthe pressure distribution, in Pa, as a function of the number of the flap track fairing aerodynamic grids and the angle alpha is shown in the following *figure 3* whereas the pressure distribution on the flap track fairing at a fixed value of α and θ is shown in *figure 3*.

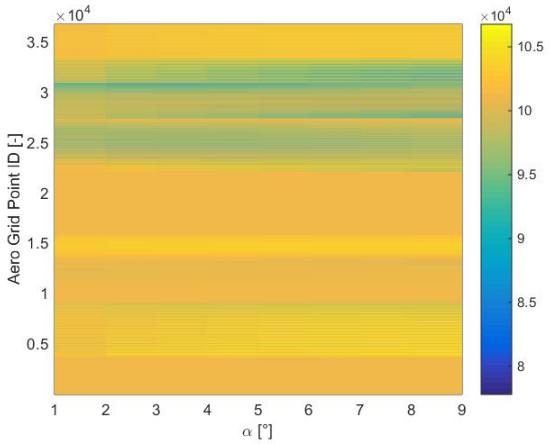


Figure 3: The following figure shows the evolution of pressure distribution for a constant θ and varying values of α. Y-axes represents the 36802 nodes available on the aero-mesh, whereas the X-axes represents the evolution of α

3 Singular Value Decomposition and Interpolation

Singular Value Decomposition is a method for remodeling a set of correlated variables into a set of orthogonal variables [5]. Basically, SVD transforms the original input space into a new space with independent dimensions. This technique gives us more clearly the idea about which dimensions contribute significantly in the database. Hence it can be used to remove dimensions with low energy content and thereby giving a reduced order model. SVD is regularly used in reducing order of structural FEM's [6] and aerodynamic meshes [7] while performing dynamical simulations. In mathematical terms an SVD is the factorisation of a rectangular matrix of real or complex data of dimension *m x n*, defined as $[A]$, of the form $[U]$ $[\Sigma]$ $[V]^*$.

$$[A] = [U][\Sigma][V]^T \tag{2}$$

The matrix $[U]$ is an *m x m* unitary matrix and its columns are the left singular vectors, $[\Sigma]$ is an *m x n* rectangular diagonal matrix containing the singular values and $[V]$ is a *n x n* unitary matrix whose transpose has rows that are the right singular vectors. The SVD represents an expansion of the original data in a coordinate system where the covariance matrix is diagonal.
In this paper SVD is used to reduce the dimension of the pressure database by neglecting the contribution of the singular values whose energetic content is lower than a certain fixed threshold. After reducing the database, the weights are interpolated to recreate the pressure distribution on the arifoil. Interpolation through data is realized with the built-in Matlab® function *interpn* that returns interpolated values of a function of n variables at specific points using, in this specific case, spline interpolation.

4 Gaussian Process Regression

A gaussian process is an infinite dimensional multi-variate gaussian. Such that any subset of the process is a multi-variate gaussian distribution. A gaussian process can be fully parametrized by a mean and covariance function Eq. 3.

$$y(x) = GP(m(x,\theta), k(x,x',\theta)) \tag{3}$$

A random draw from a Gaussian Process yields a random function around the mean function $m(x,\theta)$ and of the shape as defined by covariance function $k(x,x',\theta)$. Hence, Gaussian Process constitutes a method to define a family of functions whose shape is defined by its covariance function. A popular choice of covariance function is a squared exponential function Eq. 4, because it defines a family of highly smooth (infinitely differentiable) non-linear functions as shown in Fig. 4(a).

$$k(x,x',\theta) = \theta_1^2 exp[-\frac{(x-x')^2}{2\theta_2^2}] \tag{4}$$

A covariance function is fully parametrized by its hyperparameters θ_i's. For the case of Squared exponential kernel the hyperparameters are its amplitude θ_1 and length scale θ_2.

Given a dataset (x,y) regression deals with finding latent function f between the inputs x and outputs y. While performing polynomial regression it is assumed that the above-mentioned function f comes from a family of polynomial functions. Since Gaussian Processes are such handy tools to define a family of non-linear functions. In a Gaussian Process Regression (GPR) one starts with an initial family of functions defined by a GP called prior Fig. 4(a).

$$\mathbb{P}(f \mid x,\theta) = GP(y|0,K_{xx}) \tag{5}$$

Due to the bayesian setting of GP it is possible to calculate the posterior mean and variance as shown in Eq. 6 and Eq. 7. This means to effectively eliminate all the functions in the prior that do not pass through the specified data points Fig. 4(c).

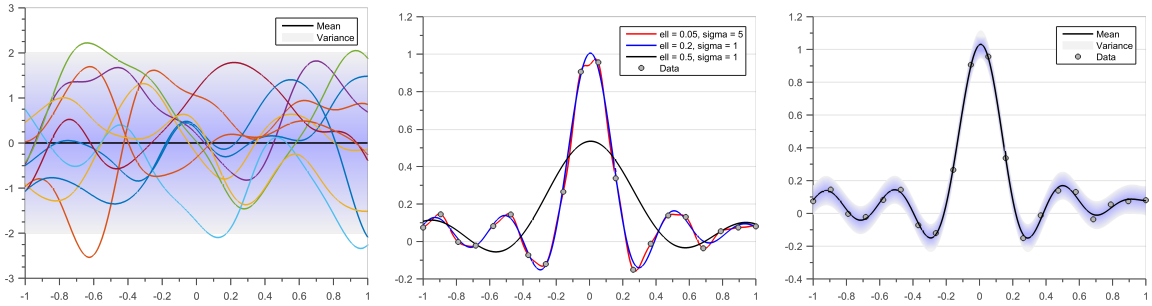
$$m(y_*) = k_{x_*x}(K_{xx})^{-1}y \tag{6}$$

$$Cov(y_*) = k_{x_*x_*} - k_{x_*x}\,(K_{xx})^{-1}k_{xx_*} \tag{7}$$

More over predictions can be improved by choosing a better prior. This involves optimizing the Marginal Likelihood (ML) $\mathbb{P}(y \mid x,\theta)$ calculated as Eq. 8. The probability that the dataset of study (x,y) comes from a

family of functions defined by the prior is called the ML [8]. Hence, when the ML is optimized the optimal θ or family of functions that describe the data are actually found Fig. 4(b).

$$\mathbb{P}(y \mid x, \theta) = GP(y|0, K_{xx} + \sigma^2 I) \tag{8}$$



(a) Prior (b) Marginal Likelihood (c) Posterior

Figure 4: Gaussian Process Regression

4.1 Distributed Gaussian Process

The above GP approach is intractable for large datasets. For a GP as defined in section 4 the covariance matrix is of size N, where $\mathcal{O}\left(N^3\right)$ time is needed for inference and $\mathcal{O}\left(N^2\right)$ memory for storage. Thus, it is necessary to consider approximate methods in order to deal with large datasets.

Inverting the covariance matrix takes considerable amount of time and memory during the process. Hence, almost all techniques to approximate inference try and approximate the inversion of covariance matrix K_{xx}.

A way to approximate covariance matrix is to learn local experts on subset of data. Traditionally, each subset of data learns a different model from another, this is done to increase the expressiveness in the model [9]. The final predictions are then made by combining the predictions of local experts [10].

An alternative way is to tie all the different experts using one single set of hyperparameters [11]. This is equivalent to assuming one single GP on the whole data-set such that there is no correlation across experts. This tying of experts acts as a regularization and inhibits overfitting. Although ignoring correlation among experts is a strong assumption, it can be justified if the experts are chosen randomly and with enough overlap.

If the dataset is partitioned into M subsets such as $\mathscr{D}^{(i)} = X^{(i)}, y^{(i)}, i = 1, \ldots M$.

$$\log p(y|X, \theta) \approx \sum_{k=1}^{M} \log p_k(y^{(i)}|X^{(i)}, \theta) \tag{9}$$

The above equation 9 describes the formulation for marginal likelihood. Due to the independence assumption the marginal likelihood can be written as a sum of individual likelihoods and then can be optimized to find the best-fit hyperparameters. After learning the hyperparameters the predictions of local experts can be combined to give mean and variance predictions. The robust Bayesian Committee Machine (rBCM) model combines the various experts using their confidence on the prediction point [11]. In such manner experts which have high confidence at the prediction points get more weight when compared to experts with low confidence.

$$m(Y_*) = (Cov(X_*))^{-2} \sum \beta_k \sigma_k^{-2} m_k(X_*) \tag{10}$$

$$(Cov_(Y_*))^{-2} = \sum_k \beta_k \sigma_k^{-2} + (1 - \sum_k \beta_k)\sigma_{**}^{-2} \tag{11}$$

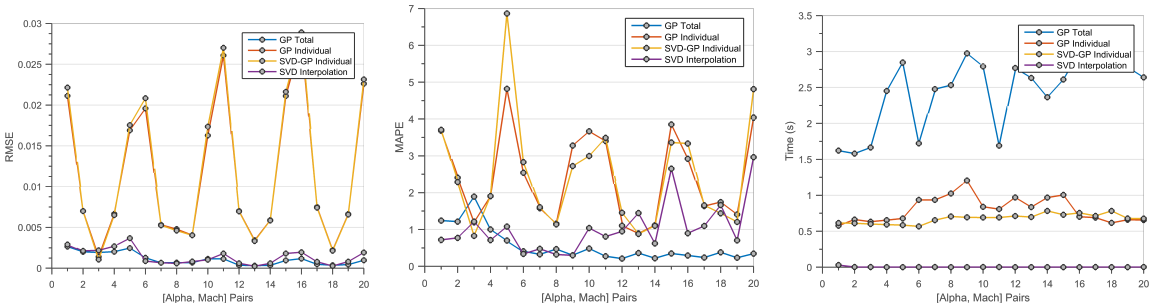
In the above equations $m_k(X_*)$ and σ_k are the mean and covariance predictions from expert k at point X_*. σ_{**} is the auto-covariance of the prior at prediction points X_*. β_k determines the influence of experts on the final predictions [12] and is given as $\beta_k = \frac{1}{2}(\log \sigma_{**}^{-2} - \log \sigma_k^{-2})$.

5 Results

The performance of SVD interpolation, Gaussian Process interpolation performed on individual nodes and distributed Gaussian Process on all the aero nodes together is empirically assessed. The XFoil data-set as described in section 2.1 is the first to be tested. Then the performance on the elsA® data-set 2.2 is evaluated.

Toolbox GPML provided with [8] is used to perform Gaussian Process regression and distributed GP is inspired from [11]. The SVD and interpolation analysis is done using built-in matlab functions[13]. All experiments were performed on an Intel quad-core processor with 4Gb RAM.

5.0.1 Xfoil



(a) Root Mean Square Error (RMSE) for the four different model types. The x axes denotes indexes of removed doublets. The performance of SVD interpolation is comparable to that of distributed GP interpolation. The peaks in the RMSE plots denote cases when the removed doublets are on the edges of database. Hence extrapolation is effectively performed during these experiments.

(b) Mean absolute percentage error (MAPE) for the four different model types. The x axes denotes indexes of removed doublets. The performance of SVD interpolation is comparable to that of distributed GP interpolation. The peaks in the RMSE plots denote cases when the removed doublets are on the edges of database. Hence extrapolation is effectively performed during these experiments.

(c) Time taken to perform prediction for the four different model types. The x axes denotes indexes of removed doublets. The time taken by Large scale GP is highest when compared to SVD and individual interpolation techniques. The SVD interpolation is fastest sometimes performing 30 times faster than large scale GP.

Figure 5: Results for XFOIL interpolation

The data of XFoil® is generated for a set of $\alpha$and M variations as described in section 2.1. $[\alpha, M]$ doublets are then progressively removed from the database. The new database is used to learn the interpolation model according to various techniques. Finally, performance is measured by comparing the Root Mean Square Error (RMSE) and Mean absolute percentage error (MAPE) between C_P of removed $[\alpha, M]$ doublets and the predicted models.

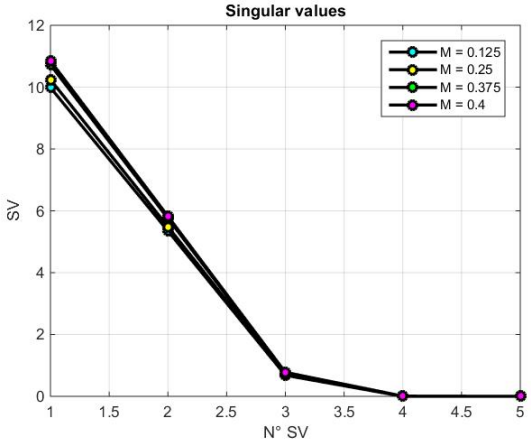


Figure 6: Singular Values for different Mach numbers

SVD and spline interpolation Transforming the full database with SVD allows to identify the parameters of the system with the highest energetic content. Form figure 6 it can be observed that most of the energetic content of the system is contained in the first two singular values. Therefore the system can be cropped accordingly. Then the prediction in removed doublets is made through spline interpolation.

Gaussian Process - individual nodes Since the Gaussian Process model scales poorly with increasing amount of data. Building one single model for all the mesh nodes is not advisable since the number of datapoints will tend to the order of $\mathcal{O}^4$. Hence in this case individual models were built for all the 182 nodes of the XFoil mesh. This method does not learn the correlations between individual nodes of a mesh and hence some information is lost while building the model.

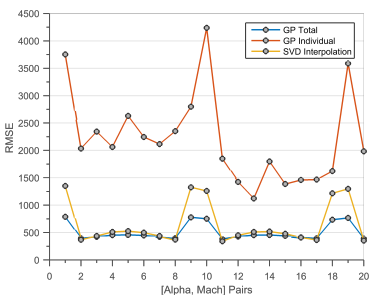
SVD and GP interpolation Using the reduced model created earlier, weights are interpolated through Gaussian process regression for individual nodes. This is done to see how the Gaussian process can handle the lack of information after model reduction.

Gaussian Process - large scale To overcome the huge amount of learning and prediction time a distributed Gaussian Process model is built. The dataset was distributed into different experts randomly and each expert had 512 datapoints. Then like earlier cases the performance was verified by calculating the RMSE and MAPE error.

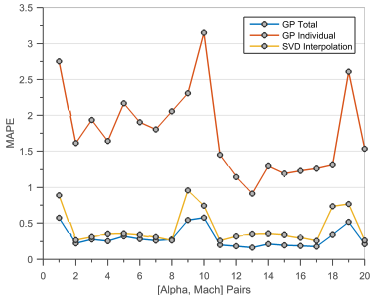
Figures 5(a) and 5(b) denote the two error estimates for different doublets. The SVD and large scale GP perform significantly better than models where individual models are constructed. Large scale GP performs marginally better than SVD. The peaks in the RMSE plots denote cases when the removed doublets are on the edges of database. Hence extrapolation is effectively performed during these experiments. Figure 5(c) shows the time taken to perform prediction for the four different model types. The x axes denotes indexes of removed doublets. The time taken by Large scale GP is highest when compared to SVD and individual interpolation techniques. The SVD interpolation is fastest sometimes performing 30 times faster than large scale GP.

5.0.2 elsA RANS - Application to a Flap Tack Fairing

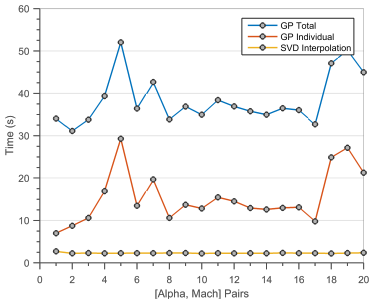
The data obtained with elsA® are calculated for a set of $\alpha$and θ. Refer to 2.2 for a general description of the database. The procedure applied during the analysis is similar to the one described in the previous section. $[\alpha, \theta]$ doublets are removed one by one from the database to create a new one. Pressure is calculated for the missing $[\alpha, \theta]$ doublets by interpolation, Gaussian procces on the individual nodes and large scale Gaussian process. The methods are finally compared by evaluating the RMSE and MAPE values for each doublet case.



(a) Root Mean Square Error (RMSE) for the three different model types. The x axes denotes indexes of removed doublets. The performance of SVD interpolation is comparable to that of distributed GP interpolation. The peaks in the RMSE plots denote cases when the removed doublets are on the edges of database. Hence extrapolation is effectively performed during these experiments.



(b) Mean absolute percentage error (MAPE) for the three different model types. The x axes denotes indexes of removed doublets. The performance of SVD interpolation is comparable to that of distributed GP interpolation. The peaks in the RMSE plots denote cases when the removed doublets are on the edges of database. Hence extrapolation is effectively performed during these experiments.



(c) Time taken to perform prediction for the three different model types. The x axes denotes indexes of removed doublets. The time taken by Large scale GP is highest when compared to SVD and individual interpolation techniques. The SVD interpolation is fastest sometimes performing 30 times faster than large scale GP.

Figure 7: Results for elsA interpolation

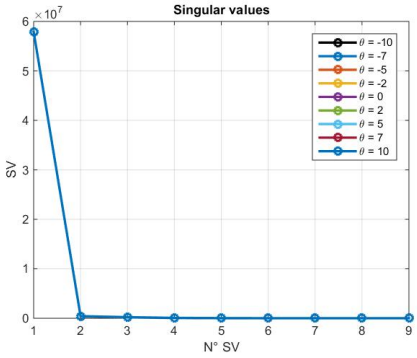


Figure 8: Singular Values for different θvalues for elsA RANS simulations

SVD and interpolation After transforming the database through SVD we find that most of the energy content of the database was contained in the first two singular values figure8. When compared to the XFOIL case figure 6 the energy is more concentrated to the first two dimensions in the elsA RANS simulations figure 8. Finally, the prediction from the removed doublets is performed through spline interpolation as described in section 3.

Gaussian Process - individual nodes The Gaussian Process Regression model as described in section 4 is performed on the elsa RANS data in this section. The elsA RANS simulation has a total of 36802 nodes in the mesh, this combined with the 81 α and θ combinations $(9\alpha \times 9\theta)$ will result in a staggering $N = 2.9$ million data-points. Even if we are willing to wait $\mathcal{O}\left(N^2\right)$ time to perform the prediction our current RAM capacity cannot hold a matrix of size N^2. Hence we build individual GP models for each node in the mesh. This greatly reduces the memory needed for prediction, while the time for building and prediction of the model increases linearly with number of nodes. But, on the downside we cannot learn the correlation between individual nodes of the mesh.

Gaussian Process - large scale Since the GP model on individual nodes cannot capture inter-node correlations. A distributed GP model as described in section 4.1 is used for prediction. The data-set was distributed into different experts randomly and each expert had 512 data-points. In this case the problem of memory overload is bypassed by distributing the model building to several experts. Then like earlier cases the performance was measured by calculating the RMSE and MAPE error.

Figures 7(a) and 7(b) denote the two error estimates for different doublets. The SVD and large scale GP perform significantly better than GP regression on individual nodes. Large scale GP performs marginally better than SVD. The peaks in the RMSE plots denote cases where the removed doublets are on the edges of database. Hence extrapolation is effectively performed during these experiments. Figure 5(c) shows the time taken to perform prediction for the three different model types. The x axes denotes indexes of removed doublets. The time taken by Large scale GP is highest when compared to SVD and individual interpolation techniques. The SVD interpolation is the fastest sometimes performing 30 times faster than large scale GP.

6 Conclusion

This paper presents a comparison between three different surrogate model building methods: time-tested surrogate modelling methods such as SVD coupled with spline interpolation and upcoming machine learning methods such as distributed Gaussian Process Regression in simple and large scale mode.

Section 55.0.1 presents a comparison on an XFOIL data-set. Distributed GP performs marginally better when compared to SVD with interpolation, while GP on individual nodes performs the worst. On the contrary SVD with interpolation is significantly faster when compared to the two GP techniques. Section 5.0.2 presents a comparison on elsA RANS simulation comprising of 36802 nodes on the mesh. Again the distributed GP performs marginally better than SVD technique but is a minimum 30 times slower than SVD. Although the GP technique can more efficiently capture non-linearities we see a diminished improvement in performance for the amount of time invested. We leave the weight of deciding between simple and time-tested SVD interpolation vs costly and accurate distributed GP interpolation on the utility function of the final user.

If we keep aside the error induced due to the performing the Gaussian Process interpolation on distributed architecture, the Gaussian Process should give better results when compared to SVD interpolation. This is mostly

due to the GP's capability to learn non-linear models. For the current two cases we have shown that the marginal increase in efficiency given by distributed GP is of not the same order of the time spent in learning. In the final manuscript we will also demonstrate the capabilities of interpolation of GP vs SVD in presence of shocks on the CRM in transonic regime [14]. Shocks create a mathematical discontinuity in the pressure distribution. In our initial calculations we have found that capability of performing non-linear interpolation and hence distributed GP becomes indispensable for such scenarios.

References

[1] Sandboge, R., "Fluid–structure interaction with OpenFSITM and MD NastranTM structural solver," *Ann. Arbor*, Vol. 1001, 2010.

[2] Montgomery, D. C., Peck, E. A., and Vining, G. G., *Introduction to linear regression analysis*, John Wiley and Sons, 2015.

[3] Drela, M., "XFOIL: An analysis and design system for low Reynolds number airfoils," *Low Reynolds number aerodynamics*, Springer, 1989, pp. 1–12.

[4] Cambier, L. and Veuillot, J., "Status of the elsA CFD software for flow simulation and multidisciplinary applications," *AIAA paper*, Vol. 664, 2008, pp. 2008.

[5] Sadek, R. A., "SVD based image processing applications: state of the art, contributions and research challenges," *arXiv preprint arXiv:1211.7102*, 2012.

[6] Friswell, M. and Mottershead, J. E., *Finite element model updating in structural dynamics*, Vol. 38, Springer Science & Business Media, 1995.

[7] Schmid, P. J., "Dynamic mode decomposition of numerical and experimental data," *Journal of Fluid Mechanics*, Vol. 656, 2010, pp. 5–28.

[8] Rasmussen, C. E. and Williams, C. K. I., *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*, The MIT Press, 2005.

[9] Rasmussen, C. E. and Ghahramani, Z., "Infinite mixtures of Gaussian process experts," *Advances in neural information processing systems*, Vol. 2, 2002, pp. 881–888.

[10] Chen, T. and Ren, J., "Bagging for Gaussian process regression," *Neurocomputing*, Vol. 72, No. 7, 2009, pp. 1605–1610.

[11] Deisenroth, M. P. and Ng, J. W., "Distributed Gaussian Processes," *arXiv preprint arXiv:1502.02843*, 2015.

[12] Cao, Y. and Fleet, D. J., "Generalized Product of Experts for Automatic and Principled Fusion of Gaussian Process Predictions," *CoRR*, Vol. abs/1410.7827, 2014.

[13] MathWorks, I., *MATLAB: the language of technical computing. Desktop tools and development environment, version 7*, Vol. 9, MathWorks, 2005.

[14] Vassberg, J. C., DeHaan, M. A., Rivers, S. M., and Wahls, R. A., "Development of a common research model for applied CFD validation studies," 2008.