Multi-Disciplinary Shape Optimization of Stratospheric Airships

AE 797 M.Tech. Project Stage I Report
by

Manideep Reddy Desham
(Roll No. 163010004)

Supervisor:
Prof. Rajkumar S. Pant

Department of Aerospace Engineering
Indian Institute of Technology Bombay
Mumbai 400076 (India)

6 June 2018

Acceptance Certificate

Department of Aerospace Engineering
Indian Institute of Technology, Bombay

The project report entitled “Multi-Disciplinary Shape Optimization of Stratospheric Airships” submitted by Manideep Reddy Desham (Roll No. 163010004) may be accepted for being evaluated.

Date: 6 June 2018

Prof. Rajkumar S. Pant

Abstract

Stratospheric Airships are being considered as the aerial platform of choice for long endurance applications, especially for mounting next generation of communications payloads. Such airships are generally powered by solar propulsion systems, hence it is preferable for their envelopes to have flat upper surfaces for mounting solar panels.

Most existing studies related to shape optimization of stratospheric airships assume their envelopes to be axisymmetric bodies of revolution, since the drag coefficient of such shapes can be estimated by 2D CFD analyses. But non-axisymmetric envelope shapes need 3D CFD analysis to be performed, which demands more computational effort.

The present study aims at developing a surrogate based design optimization (SBDO) methodology for obtaining minimum drag shapes of stratospheric airship envelopes, which are non-body of revolution. A novel scheme for parameterization of envelope shapes of a given volume is presented, using modified Gertler Series 58 Shape Generator. Latin Hyper-cube Sampling is used to generate 100 shapes, and the volumetric drag coefficient (C DV) of the envelope is determined at these points by carrying out 3-D CFD analysis using OpenFOAM®. A simple Kriging surrogate model was fitted through these points, which predicted C_{DV} **values within** 2% accuracy at 15 uniformly distributed trial shapes. A shape corresponding to minimum C DV of this surrogate function was obtained using Genetic Algorithm optimizer, and was found to have a C_{DV} only slightly worse off than that of a reference GNVR shape, thus establishing the efficacy of this scheme.

Table of Contents

List of Figures

List of Tables

Chapter 1

Background and Introduction

1.1 Stratospheric Airships

In the present climate of growth in broadband-hungry telecommunication applications, wireless infrastructure providers are under continuous pressure to exploit the limited radio spectrum as efficiently as possible. In this context, high altitude platforms (HAPs) are increasingly being cited as having an important role to play in future systems and applications. They have the potential to exploit many of the best aspects of terrestrial and satellite based systems, while offering advantageous propagation characteristics. Such platforms may be airships or aircraft and for environmental considerations would ideally be solar powered.

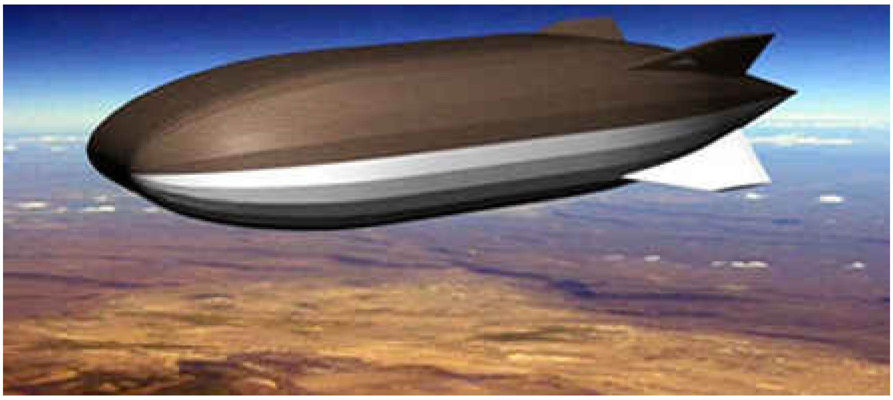

Figure 1.1: A stratospheric airship (Stratellite)

Keeping the above problem in mind, a stratospheric airship can be designed which can float at an altitude of 17-25 km and would be powered by direct solar energy extracted by photo-voltaic cells during day time and energy stored in the regenerative fuel cells during night time. The airship is required to stay aloft at a particular position despite of the wind gusts. So,the position should be corrected from time to time. This demands for the need of propellers using which the position of the airship at a particular altitude is corrected from time to time.

17-25 KM altitude is the place where little or no weather activities like rain, thunderstorms or cyclones takes place. At that altitude, things like pressure, density temperature and wind velocity remains constant or changes from season to season. So, this layer can be utilized to station our stratospheric airship.

There are potentially many advantages of stratospheric airships compared to ground level telecommunication or satellite communication systems. Some of them are:

- Larger ground coverage compared to ground level telecommunication systems like towers.
- They can be used for strategic surveillance purposes by defence and military agencies.
- They can be thought of an alternative to geostationary satellites with two potential advantages. One being low initial and operating costs involved. Second being the ability to update, repair and maintain the payload systems from time to time.
- They can be used for communication purposes during emergency caused because of natural calamities or disasters since ground based telecommunication systems might not work properly at these times.

1.2 Motivation

Unconventional non-axisymmetric shapes provides many advantages over conventional axisymmetric shapes for stratospheric airships because they can have flatter upper surface and is advantageous for capturing more solar irradiance. Existing literature has good amount of work on CFD analysis of axisymmetric shapes. But there is no considerable amount of analysis on non-axisymmetric shapes. The reason is obvious, 2D analysis can be easily done for axisymmetric shapes. But to do CFD analysis on non-axisymmetric shapes, we need to perform 3D CFD analysis. 3D analysis demands more computational effort which might not be available for many researchers.

There are many advantages for OpenFOAM® like availability of source code, small size code (only 200 MB) and complete flexibility in usage. OpenFOAM® also comes with an inbuilt meshing (*Block mesh* and *Snappy Hex Mesh*) and post-processing (*Paraview*) utilities, both of which are known for their unique and advanced method of problem solving capabilities.

1.3 Objective and Aims

The project is divided into two stages namely Stage-1 and Stage-2. In Stage-1, existing CFD results available in the literature for four standard shapes will be reproduced by carrying out 3D CFD analysis using OpenFOAM®. A methodology for carrying out multi-disciplinary optimization will be laid out. The methodology should be such that optimal shape of the envelope should be obtained in minimum amount of time and optimally utilizing computational resources. The objectives for Stage-2 are explained in Chapter 9

1.4 Report layout

After providing a brief introduction to Stratospheric Airships outlining the aims and objectives of this study, Chapter 2 gives the detailed literature survey of various fields required for multi-disciplinary shape optimization. Chapter 3 explains two of the many methods used for the parameterization of geometry. This is also shown in Figure 1.2. Next, short introduction for advantages of using OpenFOAM® is discussed in Chapter 4.

Mesh generation using two utilities namely *Gmsh* and *SnappyHexMesh* are discussed in Chapter 4.2. A brief introduction of methodology for optimization is given in Chapter 5. However optimization will be done in stage -02 along with other considerations as shown in 1.2.

Finally, Chapter 4.3 discusses the results obtained while validating the results present in literature for various standard airship shapes using OpenFOAM®. Chapter 9 gives the future work that will be performed in stage -02. The workflow for the total project is shown in Figure 1.2. The blocks present in transparent oval is the work completed in stage 1.

Next Chapter gives brief literature survey that has been carried out to understand the previous research carried out in the field of High Altitude Airships (HAA's).

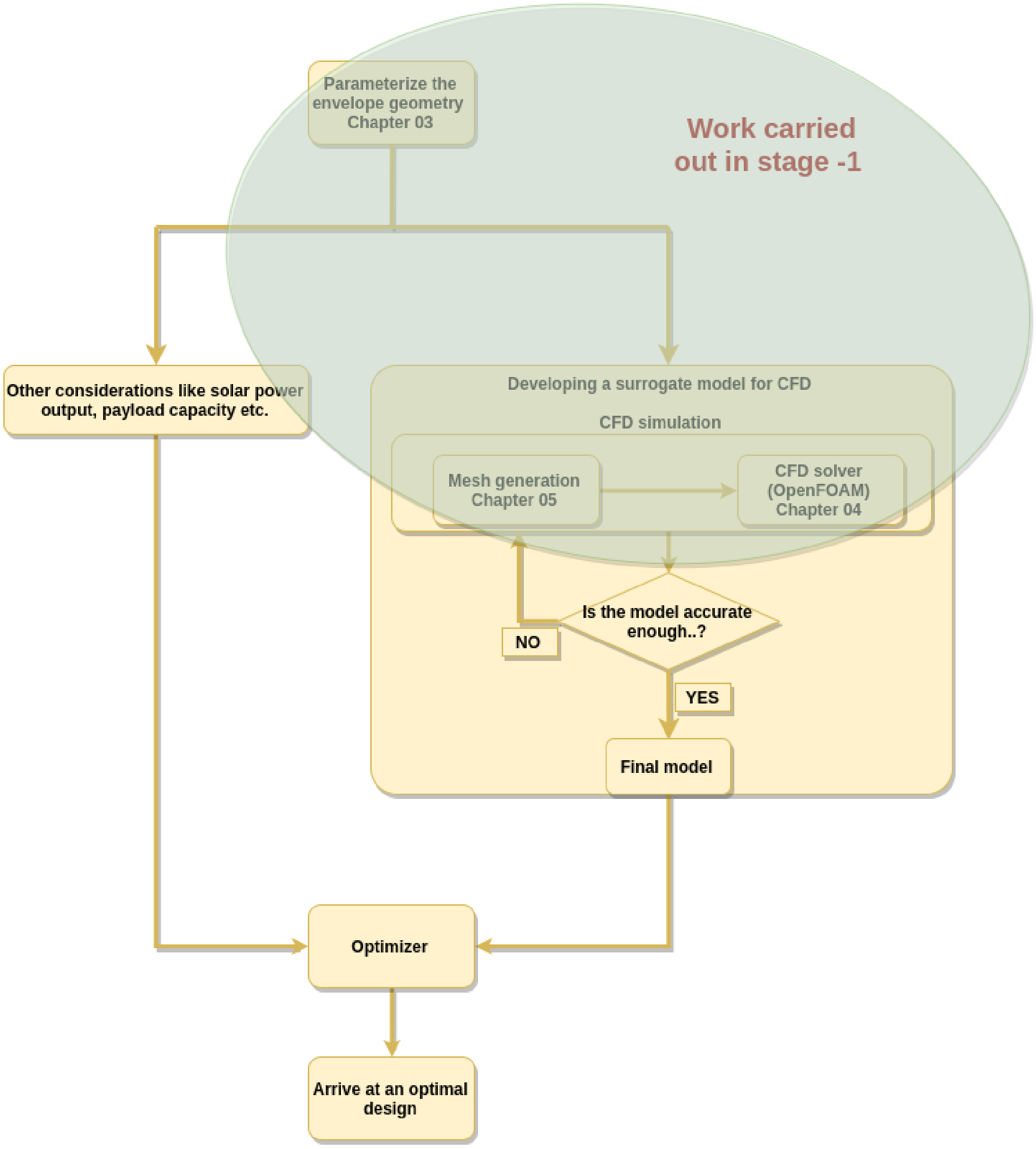
Parameterize the envelope geometry Chapter 03
Work carried out in stage -1
Other considerations like solar power output, payload capacity etc.
Developing a surrogate model for CFD
CFD simulation
Mesh generation Chapter 05
CFD solver (OpenFOAM) Chapter 04
Is the model accurate enough..?
NO
YES
Final model
Optimizer
Arrive at an optimal design

Figure 1.2: Work flow of multi-disciplinary shape optimization

Chapter 2

Literature Survey

As mentioned in Chapter 1.4, This Chapter presents the research carried out by different individuals in the field of High Altitude Airships (HAA's). Over the recent years, there has been increased interest on the topic of stratospheric airships/ High Altitude Airships (HAA's). However interest in this topic has increased exponentially at the beginning of 21st century.

2.1 Initial Sizing

Gawale and Pant (2005) presented how HAA's can be used for navigation purposes. They explained the risks involved in using GPS based navigation systems. They followed the procedure of Rehmet and Kroplin (2000) for initial sizing.

Alam *et al.* (2014) described a methodology that can arrive at the baseline specifications of a stratospheric airship, given the performance and operational requirements. They have made sensitivity analysis like effect of wind speed on the volume required and length of airship. They have also explained how different parametres like length, volume, altitude of operation, lifting gases (hydrogen and helium) are interlinked to one another.

2.2 Envelope geometry parameterization

Ghani (2013) used NURBS for envelope geometry parameterization of passenger car and constructed a response surface model on the geometries to obtain drag coefficients. Finally, the design exploration was performed using the response surface model instead of actual CFD simulations.

Alam (2016) selected *Gertler Series-58* 11 shape generation for optimization studies, in which the envelope shape is driven by four shape parameters and fineness ratio.

Wang *et al.* (2010) proposed a geometry parameterization algorithm for 2D bodies of revolution shapes four shape parameters namely *a,b,c,d* and length *l*.

2.3 Multi disciplinary shape optimization

Kanikdale *et al.* (2004) has taken GNVR shape 23 as baseline reference and validated their ANSYS® *Fluent* results with it. They then parameterized the geometry using six design parameters and response surface based on CFD results was fitted. With low fidelity analysis using the formula by Gillett and Khoury 5. An improvement of 1.3 % was observed in C_{DV}. Whereas using CFD analysis, they found that the drag for optimized shape is 27 % greater than GNVR hull shape. By this they concluded that the formula given by Gillet and Khoury cannot be used for optimization purpose.

Kale *et al.* (2005) proposed a generic methodology for determination of drag coefficient of an aerostat envelope using CFD. The envelope was parameterized in terms of six geometric coefficients, and a shape generation algorithm was developed. They have studied around 600 feasible shaped using ANSYS® *Fluent* CFD package and fitted a quadratic response surface using Design-Expert package. They taken a composite objective function involving aerodynamic drag, structural stress and surface area (so weight) of the envelope. They have observed that the location of maximum thickness affects the drag mostly.

Ram and Pant (2010) parameterized the shape using the formulation proposed by Kanikdale *et al.* (2004) and arrived at optimum shape using Hybrid Genetic Algorithm (GA). The objective function was to maximize its payload while incorporating considerations of aerodynamics, structures and flight mechanics.

Wang *et al.* (2010) have tested a scaled down model of Zhiyuan - 1 airship in a ϕ 3.2 m wind tunnel. They had installed roughness strips on the surface of hull to trip the flow from laminar to turbulent. They studied the effect of lengthwise location of these strip on the its aerodynamic characteristics at different angles of attack and side slip. They have reported large sets of experimental data for bare hull, hull with fins and hull with fins, gondola. They concluded that the drag almost doubled because of change of flow condition from laminar to turbulent.

Suman *et al.* (2011) tried to reproduce the experimental results of Wang *et al.* (2010) using computational fluid dynamics. To simulate the turbulence strips installed by Wang *et al.* (2010) on the surface of the hull, Suman *et al.* (2011) specified the lengthwise location of transition from laminar to turbulent. They observed that they could not recover the drag coefficient results are not matching. They concluded that although Wang *et al.*

(2010) provided strips at the leading edge, the flow is not tripped to turbulent because of favourable pressure gradient.

Comprehensive results for the effect on size and payload capability of airship of various factors like, altitude, latitude, pressure difference, Helium purity etc. have been published by Chen *et al.* (2010) The calculation is carried out for NPL shape.

Liu *et al.* (2013) carried out numerical calculations about the test model to investigate the aerodynamics behaviour. They confirmed the lower drag behaviour of *Zhiyuan-1* airship. They discussed the influence of gondola and fins on the pressure distribution.

Ceruti *et al.* (2013) defined a shape which is non body of revolution using five design parameters. They have found the drag coefficient for 10 such combinations of design parameters and used interpolation based approach for estimation of drag coefficient of any given shape. They used different optimization techniques to arrive at a configuration of minimum drag and weight.

2.4 Recent work in the field

In a recently completed doctoral research work by Alam **?** in the field of multi disciplinary optimization has been carried out by Alam (2016). A stage-wise summary of this research work is listed below:

Surrogate model for CFD analysis

For CFD analysis, 60 simulations have been carried out and fitted into the *Kriging* surrogate model toolbox developed by Viana *et al.* (2014). This has been coupled to optimizer without having the need to do CFD routine every time.

Shape generation algorithm:

The shape generation algorithm proposed by Gertler (Jan. 1950) called *Gertler Series 58 Shape Generator* instead of Wang *et al.* (2010) that was previously used. Capabilities of this generator has been demonstrated by capturing different standard airship shapes.

Thermal modelling

A new thermal model by considering Four nodes namely Outer layer, Solar cells, substrate and Envelope has been developed. It is then validated against the experimental data by Harada *et al.* (2003) and numerical results quoted by Liu *et al.* (2014).

Wind and solar model Initially, Global Ir-radiance model by Ran *et al.* (2007) has been used and total solar energy has been calculated assuming the surface of airship as cylinder. It is then replaced by his new model and eliminated the simplification of assuming airship as cylinder. Initially, wind velocity was assumed to be constant. It is then replaced by NASA Horizontal Wind Model (HWM07).

Sizing and Optimization: A new sizing methodology has been developed Alam *et al.* (2014) and optimization has been carried out using MATLAB Global optimization Toolbox and SIMANN optimizer.

The next chapter investigates envelope shape generation algorithms for non-axisymmetric shapes because the existing literature has only algorithms for axisymmetric shapes.

Chapter 3

Envelope geometry parameterization

Geometry parameterization is an essential part of the design exploration and shape optimization process and there are several methods of generating parametric geometries. Since our ultimate aim of the project is to study different configurations of the non-body-of-revolution shapes and select the best one satisfying our needs. Thus there is a need to parameterize the geometry. Initial attempts were made to parameterize a shape using unconventional methods like polynomial fitting, ruled surface between splines, NURBS (Non Uniform Rational B Splines) etc. The Idea was to define any shape using design variables of not more than 10 because the complexity (time taken, number of function evaluations and memory consumption) of an optimizing problem increases exponentially with number of design variables.

So, the idea was to keep as many less number of design variables as possible provided every combination of them (design variables) gives a meaningful airship shape.

3.1 Modified Gertlers parametre technique

$$z^2 = a_1 y + a_2 y^2 + a_3 y^3 + ... + a_n y^n \tag{3.1}$$

where a_1, a_2, a_3,..., a_n are the shape coefficients, whose values are driven by geometrical parameters, viz., m (location of maximum diameter), r_o (Nose radius), r_1 (tail radius) and C_p (prismatic coefficient), shown in Fig. 3.1.

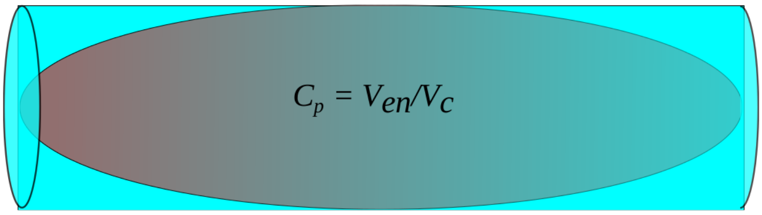

Figure 3.1: Definition of Cp

By satisfying the various constraints applicable for the shape of a stratospheric airship envelope, the final equations can be obtained in terms of these shape coefficients. To satisfy the constraints applicable for an airship envelope, we apply the conditions $z = 0$, when $y = 1$; $z = 1/2$, when $y = m$ and $\frac{dz}{dy} = 0$, when $y = m$, to obtain:

$$a_1 + a_2 + a_3 + ... + a_n = 0 \tag{3.2}$$

$$a_1 m + a_2 m^2 + a_3 m^3 + ... + a_n m^n = \frac{1}{4} \tag{3.3}$$

$$a_1 + 2a_2 m + 3a_3 m^2 + ... + n a_n m^{n-1} = 0 \tag{3.4}$$

Radius of curvature R for any curve can be given as:

$$R = \pm \frac{1}{\frac{d^2Y}{dZ^2}} \left[1 + \left(\frac{dY}{dZ} \right)^2 \right]^{3/2} \tag{3.5}$$

Eq. 3.5 can be expressed in a dimensionless form as:

$$r = \pm \frac{1}{\frac{d^2y}{dz^2}} \left[1 + \frac{L^2}{D^2} \left(\frac{dy}{dz} \right)^2 \right]^{3/2} \tag{3.6}$$

Differentiating Eq. 3.1 successively with respect to y we get:

$$2z = \left(a_1 + 2a_2 y + ... + n a_n^{n-1} \right) \frac{dy}{dz} \tag{3.7}$$

and

$$2 = \left(a_1 + 2a_2 y + ... + n a_n^{n-1} \right) \frac{d^2y}{dz^2} + \left[2a_2 + ... + n(n-1) a_n^{n-2} \right] \left(\frac{dy}{dz} \right)^2 \tag{3.8}$$

In Eq. 3.7, if $a_1 \neq 0$ then it can be said that when $x = 0$, $\frac{dy}{dz} = 0$, and hence from the Eq. 3.8, that $\frac{d^2y}{dz^2} = \frac{2}{a_1}$. Consequently substituting these values in Eq. 3.6 we can get:

$$a_1 = 2r_o \tag{3.9}$$

where r_o is the radius of curvature at the nose. If, on the other hand $a_1 = 0$, then $r_o = 0$, i.e., the body has pointed nose, which is not possible for an inflated envelope. Hence, Eq.

3.9 is valid for both the cases.
Similarly, when $y = 1$, $z = 0$ and from Eq. 3.7, $\dfrac{dy}{dz} = 0$, unless:

$$a_1 + 2a_2 + 3a_3 + ... + na_n = 0 \tag{3.10}$$

Hence, Eq. 3.6 and Eq. 3.8 give:

$$a_1 + 2a_2 + 3a_3 + ... + na_n = -2r_1 \tag{3.11}$$

where r_1 is the radius of curvature at the tail. Different signs are taken in equation of r_o and r_1 because of the nature of slope at the points $y = 0$ and $y = 1$ respectively.

Volume of the envelope (V_{env}) can be expressed as:

$$V_{env} = \int_0^l \pi Z^2 dY = \pi d^2 L \int_0^1 z^2 dy \tag{3.12}$$

substituting for z^2 from Eq. 3.1 we get,

$$\frac{1}{2}a_1 + \frac{1}{3}a_2 + \frac{1}{4}a_3 + ... + \frac{1}{n+1}a_n = \frac{1}{4}C_p \tag{3.13}$$

The aforementioned six constraints can be expressed in the form of a six degree polynomial as follows:

$$z^2(0) = 0 \tag{3.14}$$

$$z^2(1) \implies a_1 + 2a_2 + 3a_3 + ... + na_n = 0 \tag{3.15}$$

$$\left.\frac{dz}{dy}\right|_0 \implies a_1 = 2r_o \tag{3.16}$$

$$\left.\frac{dz}{dy}\right|_1 \implies a_1 + 2a_2 + 2a_3 + ... + na_n = -2r_1 \tag{3.17}$$

$$z^2(m) \implies a_1 m + a_2 m^2 + a_3 m^3 + ... + a_n m^n = \frac{1}{4} \tag{3.18}$$

$$\left.\frac{dz}{dy}\right|_m \implies a_1 + 2a_2 m + 3a_3 m^2 + ... + na_n m^{n-1} = 0 \tag{3.19}$$

$$\int_0^1 z^2(y)dy \implies \frac{1}{2}a_1 + \frac{1}{3}a_2 + \frac{1}{4}a_3 + ... + \frac{1}{n+1}a_n = \frac{1}{4}C_p \tag{3.20}$$

The above-mentioned linear equations can be represented in a Matrix form as:

$$AY = B \tag{3.21}$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 3 & 4 & 5 & 6 \\ m & m^2 & m^3 & m^4 & m^5 & m^6 \\ 1 & 2m & 3m^2 & 4m^3 & 5m^4 & 6m^5 \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} = \begin{bmatrix} 0 \\ 2r_o \\ -2r_1 \\ \frac{1}{4} \\ 0 \\ \frac{1}{4}C_p \end{bmatrix}$$

There are six unknowns in these six linear equations, hence $\boldsymbol{Y}$ can be obtained for a set of geometrical parameters by solving these equations simultaneously. The final equation for airship envelope shape can be rewritten as:

$$z(y) = D\sqrt{a_1\left(\frac{y}{L}\right) + a_2\left(\frac{y}{L}\right)^2 + a_3\left(\frac{y}{L}\right)^3 + a_4\left(\frac{y}{L}\right)^4 + a_5\left(\frac{y}{L}\right)^5 + a_6\left(\frac{y}{L}\right)^6} \tag{3.22}$$

The obtained curve is then rotated about X-axis to get axisymmetric shape. However to get a non axisymmetric shape, we introduce a new variable called $scale_y$ which when multiplied with the y-coordinates of the body changes it into non-axisymmetric shape. The below figure shows the effect of scaling.

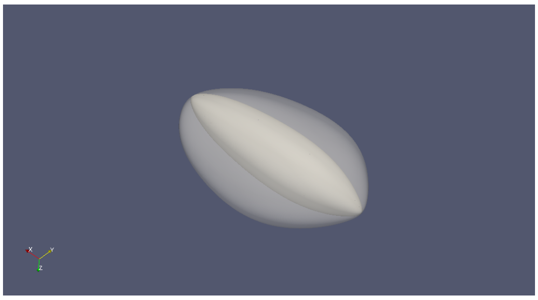

Figure 3.2: Effect of scaling

3.2 Methodology to generate surface using *Octave*

Since the analysis work is being carried out using OpenFOAM®, we need to input the geometry file to *SnappyHexMesh* utility. OpenFOAM® don't have any geometry creation utility. We need to create the geometry using any commercial designing software and export the file to .stl format and carryout the analysis. During optimization, Calling the commercial CAD software every time to define simple mathematical shape is not a good practise. Instead we can define the geometry in Octave itself and carryout simulations. So, an existing MATLAB® script has been edited which takes the input of surface data (X,Y and Z vectors) and writes them into .stl file. This .stl file can then be used in OpenFOAM®. So, the need for commercial designing software has been eliminated positively.

3.2.1 STL (file format)

An ASCII STL file begins with the line

```
solid name
```

where name is an optional string (though if name is omitted there must still be a space after solid). The file continues with any number of triangles, each represented as follows:

```
facet normal ni nj nk
        outer loop
                vertex v1x v1y v1z
                vertex v2x v2y v2z
                vertex v3x v3y v3z
        endloop
endfacet
```

where each n or v is a floating-point number in sign-mantissa-"e"-sign-exponent format, e.g., "2.648000e-002". The file concludes with the line,

```
endsolid name
```

Thus the *surf* data consisting of x,y,z coordinates of all the points are written into a text file in the above format and can be imported into OpenFOAM® to carryout CFD simulation.

Next chapter discusses why OpenFOAM® has been used for CFD analysis when we have industry standard commercial CFD softwares.

Chapter 4

Introduction to OpenFOAM®

OpenFOAM® is a Free Open Source Software (FOSS) where we can get the source code along with binary executable for all the solvers. The FOAM project (precursor to OpenFOAM®) was created at Imperial College London in the late 80s and early 90s by Henry Weller, along with Hrvoje Jasak, using the new object-oriented programming language C++. Succeeding the commercial software FOAM, OpenFOAM® was released in 2004 as free and open-source software under the GNU General Public License (GPL), a widely used free software license, which guarantees end users the freedom to run, study, share and modify the software.

OpenFOAM® has been used instead of ANSYS® *Fluent* because it has many potential advantages. Here we can modify the existing solvers/models to suit our particular needs, compile them and add them to our library. One distinguishing feature of OpenFOAM® is its syntax for tensor operations and partial differential equations (PDEs) that closely resembles the equations being solved. For example, the equation:

$$\frac{\partial(\rho U)}{\partial t} + \nabla \cdot \phi U - \nabla \cdot \mu \nabla U = -\nabla p \tag{4.1}$$

is represented by the code:

```
Solve
(
fvm :: ddt(rho,U)
+ fvm :: div(phi,U)
- fvm :: laplacian(mu,U)
==
-fvc :: grad(p)
);
```

Thus we can see that all the mathematical operators like divergence, curl, gradient, laplacian etc. are well defined in OpenFOAM®. OpenFOAM® treats fluid, structural, thermal or acoustic or any other field problems in the exactly same way as long as there are governing equations in them and they can be written in conservative form. It uses finite volume discretization method. Whenever we are programming any new solver, 90 percent of the code remains same with baseline solver and changes to be made are less than 10 percent. Thus using OpenFOAM®, There is no need to write the entire code right from scratch every time. We can just edit the existing solver and save lot of time.

Automatic mesh generation utility called *SnappyHexMesh* comes along with OpenFOAM where in we have to give .stl or .obj file of geometry and specify the level of refinement needed. The rest is taken care by it. This is particularly useful when we have to couple CFD analysis to the optimizer. *SnappyHexMesh* is still in development phase and is getting better with every release.

The absence of Graphical User Interface (GUI) can be taken as an advantage because we come to know the actual physics of flow happening behind the scene. If you become moderately expert in OpenFOAM® then it saves a lot of time because we can programatically control all the inputs/outputs and we can automate many processes right from mesh generation to post processing.

OpenFOAM® comes with a Open Source CFD post processing utility known as *ParaView.ParaView* accepts both point field data as well as volume field data. Since we solve Navier Stokes equations using Finite Volume Approach, at the end we are provided with data at the centre of cells. In this case, *ParaView* is very useful. On the other hand, *Tecplot*® (commercially available CFD post software) accepts data only at the vertices.

4.1 OpenFOAM® for CFD researchers

The main difficulty for engineers in using OpenFOAM® is that it is not user friendly. Learning curve for openFOAM® is very steep. But learning it adds value to one's skill set. OpenFOAM® is used in the cases where ANSYS® *Fluent* fails. Because it has the transparency within it. It is arguably equally powerful as ANSYS® *Fluent*. To do any simulation using OpenFOAM®, We have to basically think all the way back from vector Calculus to advanced topological errors. This will not only make us think like a computational scientist but will create awareness about how do we go about solving a problem from its inception to execution.

There are also many graphical user interfaces available for OpenFOAM®. However using gui, we can do only limited tasks. OpenFOAM® also comes with many utilities using which we can convert mesh generated by third party meshing tools like ANSYS® *Gambit*, *Gmsh*, etc to OpenFOAM® format.

OpenFOAM® also comes with parallel processing utilities using which we can decompose, reconstruct and redistribute the computational case to perform parallel calculations, at times better than other CFD software. All these things makes OpenFOAM® suitable for general engineers because we come to know whats happening behind the screen rather than just clicking the buttons.

4.2 Computational mesh generation using *SnappyHexMesh*

SnappyHexMesh is a hexahedral unstructured mesh generation tool included in the distribution of OpenFOAM®. *SnappyHexMesh* can use a 3-D .stl surface and iteratively build a mesh upon it. Some options allow construction of several cell layers of a controllable height above the 3D surface.This feature makes it possible to have a refined mesh in the region of high velocity gradients, close to the surface. Near the surface, the mesh is refined and cells are splitted acoordingly. The process is explained in Figure 4.1 .Several regions can be selected to be refined to a desired level, and this method creates mesh of a cell aspect ratio close to 1.This is done in order to ensure that OpenFOAM® can solve the numerical problems with the highest efficiency and accuracy using meshes generated with *SnappyHexMesh*. The meshing tool *SnappyHexMesh* can be run in parallel on a computer cluster or a PC.

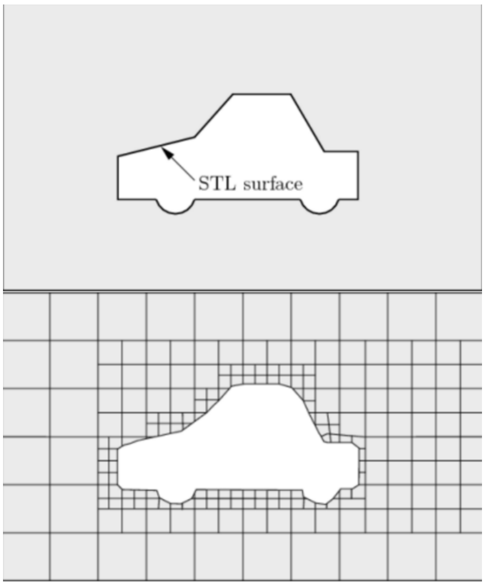

Figure 4.1: Surface snapping process in *snappyHexMesh*

Although in the Figure 4.1, it is shown for 2D, *SnappyHexMesh* is a 3D meshing utility.

SnappyHexMesh also provides options to add viscous layers to resolve boundary layer. We can also control the thickness of first layer. This thickness of first layer is usually determined by Yplus value which is a dimensionless wall thickness. A non-dimensional wall distance for a wall-bounded flow can be defined in the following way:

$$y^+ \equiv \frac{u_* y}{\nu} \tag{4.2}$$

Where u_* is the friction velocity at the nearest wall, y is the distance to the nearest wall and ν is the local kinematic viscosity of the fluid.

y^+ is often referred to simply as y plus and is commonly used in boundary layer theory and in defining the law of the wall.

Fig. 4.2 shows the mesh near the surface of an arbitrary airship shape to resolve the boundary layer.

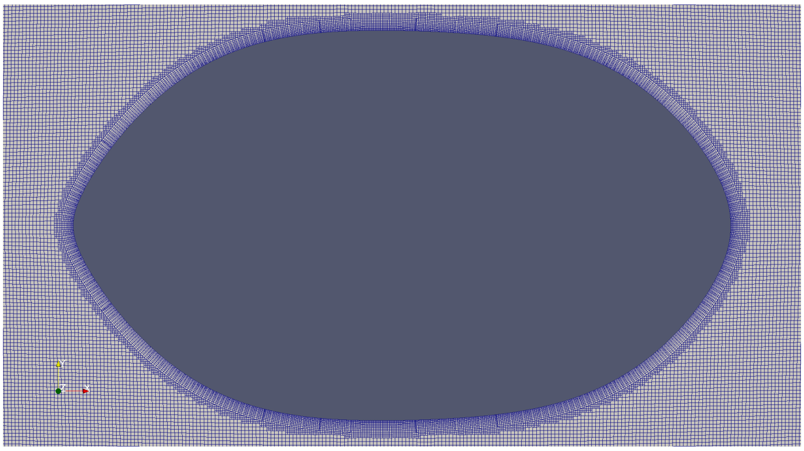

Figure 4.2: Prism layers on the surface of GNVR to capture boundary layer

4.3 Validation of OpenFOAM® CFD results

This section describes the values of C_{DV} obtained for four standard axisymmetric LTA envelope shapes, viz., GNVR Ram and Pant (2010), NPL Cheeseman *et al.* (1999) Zhiyuan-1 Wang *et al.* (2010), and Wang Wang *et al.* (2009), using OpenFOAM®, and a comparison with the quoted values. Fig. 4.3 shows these four profiles. All the simulations were carried at zero degree angle of attack.

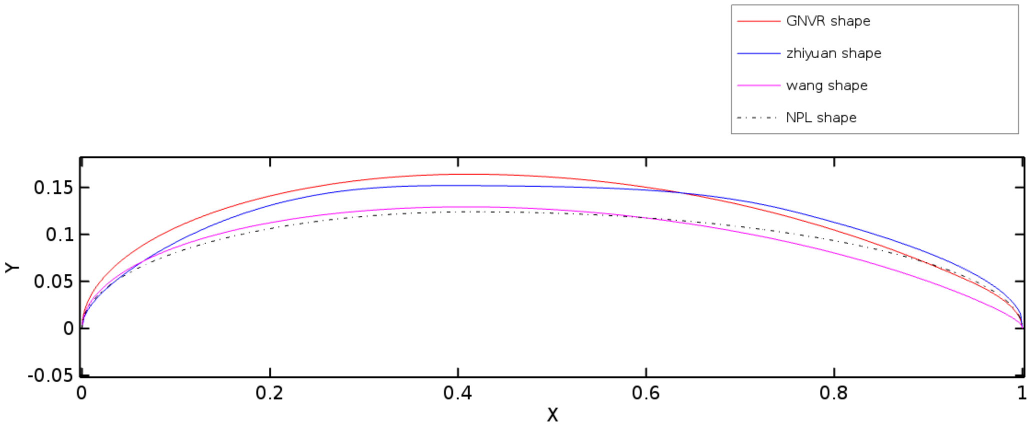

Figure 4.3: Four standard shapes available in literature

The specifications of machine, operating system, and software used in this study are as follows:

- Model name = Intel(R) Xeon(R) CPU E5-2650
- Number of physical cores = 40
- Processor speed = 2.8 GHz
- Operating system = Cent OS 7.4
- OpenFOAM® version = 4.1

Following subsections shows flow conditions, solver parametres and results of CFD simulation carried on the above mentioned standard shapes.

4.3.1 GNVR shape Ram and Pant (2010)

The GNVR shape Ram and Pant (2010) is named after late Prof. G.N.V Rao of Indian Institute of Science, Bengaluru. It consists of three standard geometrical constructs, viz, ellipse, circle and parabola. The entire envelope shape is parametrized in terms of its maximum diameter (*D*), as follows:

$$Y = \begin{cases} 0.5D\sqrt{1-\left(\frac{X-1.25D}{1.25D}\right)^2} & 0.0 < X \leq 1.25D \\ \sqrt{16D^2-(X-1.25D)^2}-3.5D & 1.25D < X \leq 2.875D \\ \sqrt{0.1373D(1.8D-(X-1.25D))} & 2.875D < X \leq 3.05D \end{cases}$$

The flow conditions used for the simulation are similar to that of Kanikdale et al. Kanikdale *et al.* (2004). Taking advantage of the axisymmetric nature of GNVR shape, its 3D CFD analysis was carried out only for 1/4 of its shape. The flow conditions and solver parameters are shown in Table 4.1

Table 4.1: Flow conditions and solver parametres for GNVR shape

Flow Conditions	Solver parametres
$Velocity = 50.339\ m/s$	$Number\ of\ cells = 646,639$
$Pressure = 101325\ Pa$	$Number\ of\ parallel\ cores = 20$
$Density = 1.225\ kg/m^3$	$Time\ taken\ for\ solution = 475\ s$
$Reynolds\ number = 34.783 * 10^6$	
$Turbulence\ model = k - \epsilon$	

The variation of Pressure coefficient (C_P) along the envelope length obtained using OpenFOAM® is compared with that of using ANSYS® *Fluent* by Kanikdale et al. Kanikdale *et al.* (2004) and Panel method by Narayana and Srilatha Narayana and Srilatha (July 2000).

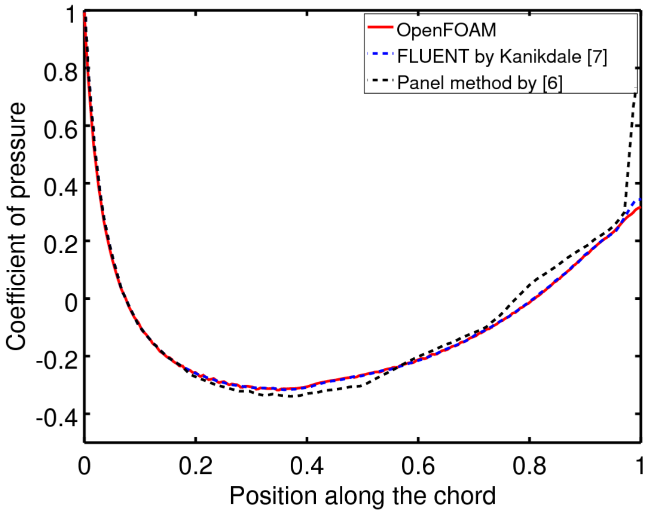

Figure 4.4: C_P distribution for GNVR profile obtained by OpenFOAM®, Source Panel Method Narayana and Srilatha (July 2000) and ANSYS® *Fluent* Kanikdale *et al.* (2004)

It can be observed from Fig. 4.4 that the C_P distribution obtained by OpenFOAM® matches well with that obtained by ANSYS® by Kanikdale et al. Kanikdale *et al.* (2004) along the entire envelope length, except at the trailing edge, where the flow is separated. The distribution obtained by Panel Method Narayana and Srilatha (July 2000) shows very high pressure recovery at trailing edge because attached flow is assumed, and the *Kutta* condition is satisfied.

4.3.2 Zhiyuan-1 shape Wang *et al.* (2010)

The equation that define the Zhiyuan-1 shape is:

$$
y = \begin{cases}
f_r'[r_n F_1(z) + k_1 F_2(z) + G(z)]^{\frac{1}{2}} & \begin{array}{l} For: 0 < x \leq x_m \\ F_1(z) = -2z(z-1)^3 \\ F_2(z) = -z^2(z-1)^2 \\ G(z) = z^2(3z^2 - 8z + 6) \end{array} & z = \frac{x}{x_m} \\
\hline
f_r'[s_t^2 F_1(z) + \left(\frac{1-x_m}{x_m}\right)^2 k_1 F_2(z) + G(z)]^{\frac{1}{2}} & \begin{array}{l} For: x_m < x \leq x_p \\ F_1(z) = -z^2(z-1)^3 \\ F_2(z) = -z^3(z-1)^2 \\ G(z) = z^3(6z^2 - 15z + 10) \end{array} & z = \frac{1-x}{1-x_m} \\
\hline
f_r'[c_p(1-z)]^{\frac{1}{2}} & For: x_p < x \leq 1 & z = x
\end{cases}
$$

Values of the constants mentioned above are as follows:

$x_m = 0.393591$, $x_p = 0.7570323$, $r_n = 0.5070992$, $k_1 = 0.291256$, $c_p = 2.735107$, $f_r' = 0.1515518$ *and* $s_t = 3.236105$

Suman et al. Suman *et al.* (2011) carried out a Reynolds-Averaged Navier - Stokes (RANS) simulation to reproduce the results obtained by Wang et al. Wang *et al.* (2010) for transition assumed at the leading edge itself, i.e., for a fully turbulent flow. They have also obtained results for various assumed locations of the transition points, resulting in pockets of laminar and turbulent regions in the computational domain.

In the present study, a 3-D CFD analysis for Zhiyuan-1 shape was carried out and pressure distribution is shown in Fig. 4.5.

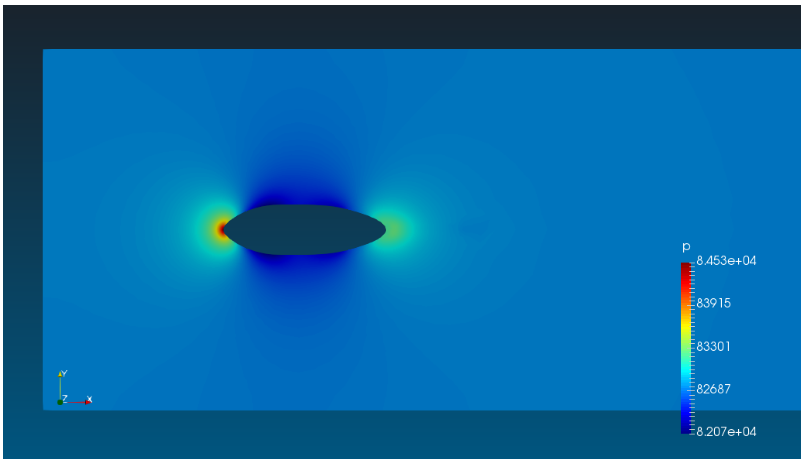

Figure 4.5: Pressure distribution on Zhiyuan-1 airship shape

The results were obtained only for one half of the envelope and ‘SymmetryPlane’ boundary condition was incorporated for the other half to reduce the computational time.

The flow conditions used for the simulation were same as used by Suman et al. Suman *et al.* (2011) and Wang et al. Wang *et al.* (2010) and is given in Table 4.2

Table 4.2: Flow conditions and solver parameters for Zhiyuan-1 shape

Flow Conditions	Solver parameters
$Velocity = 60.39\ m/s$	$Number\ of\ cells = 2,982,639$
$Pressure = 101325\ Pa$	$Number\ of\ parallel\ cores = 8$
$Density = 1.225\ kg/m^3$	$Time\ taken\ for\ solution = 4534\ s$
$Reynolds\ number = 2.4 * 10^6$	
$Turbulence\ model = \kappa\omega\ SST$	

Fig. 4.6 shows a comparison between the C_P distribution for Zhiyuan-1 envelope shape using OpenFOAM® with ANSYS® *Fluent* and experimental results quoted by Wang et al. Wang *et al.* (2010)

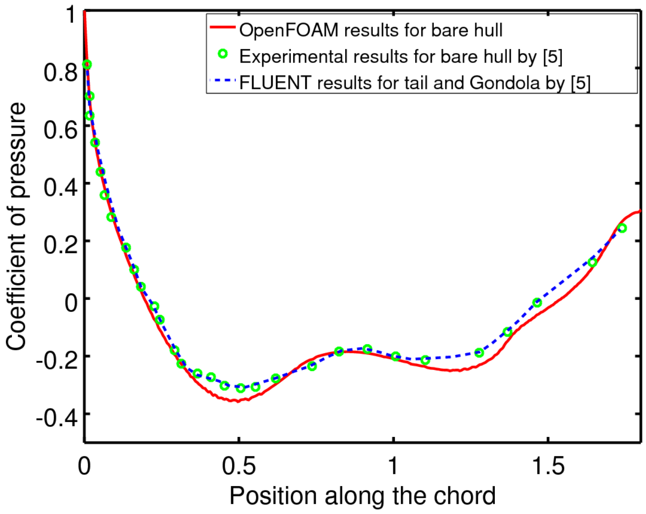

Figure 4.6: C_P distribution for Zhiyuan-1 shape using OpenFOAM®, Experiments Wang *et al.* (2010) and ANSYS® *Fluent* Wang *et al.* (2010)

Table 4.3 compares the components of C_{DV} obtained using OpenFOAM® and RANS Suman *et al.* (2011).

Table 4.3: Comparison of OpenFOAM® results with RANS Suman *et al.* (2011) for Zhiyuan -1 shape

	OpenFOAM®	RANS Suman *et al.* (2011)	% error
$C_{DV}(Viscous)$	0.01989	0.01890	5.3
$C_{DV}(Pressure)$	0.00583	0.00568	2.8
$C_{DV}(Total)$	0.02573	0.02456	4.8

4.3.3 Wang Shape Wang *et al.* (2009)

Wang et al. Wang *et al.* (2009) proposed a new shape exploring the better shapes in the view of multi-disciplinary optimization. The geometry is defined by five shape parameters namely *a, b, c, d* and length *l*. The original equation proposed by Wang et al.Wang *et al.* (2009) is given by Eq. 4.3:

$$64(y^2 + z^2) = a(l - x)\left(bx - l\sqrt{c} + \sqrt{cl^2 - dlx}\right) \tag{4.3}$$

The values taken for the parameters of Wang shape are as follows:
a = 7.447 *b = 2.072* *c = 9.010* *d = 7.981* and *l = 194.0*

For the sake of convenience, geometry is scaled down by length (*l*) in all directions. Wang shape defined by Eq. 4.3 is a body of revolution and also can be obtained by rotating the profile given by Eq. 4.4 through 360^o.

$$y = \frac{\sqrt{a(l - x)\left(bx - l\sqrt{c} + \sqrt{cl^2 - dlx}\right)}}{8} \tag{4.4}$$

The flow conditions and solver parameters used for the simulation are shown in Table 4.4:

Table 4.4: Flow conditions and solver parameters for Wang shape

Flow Conditions	Solver parameters
$Velocity = 51\ m/s$	$Number\ of\ cells = 1,060,202$
$Pressure = 8750\ Pa$	$Number\ of\ parallel\ cores = 8$
$Density = 1.225\ kg/m^3$	$Time\ taken\ for\ solution = 2865\ s$
$Reynolds\ number = 3.01 * 10^6$	
$Turbulence\ model = SpalartAllmaras$	

The values obtained for finer grid case using OpenFOAM® are compared with those of using ANSYS® *Fluent*. Fig. 4.7 shows the comparison of C_P distribution.

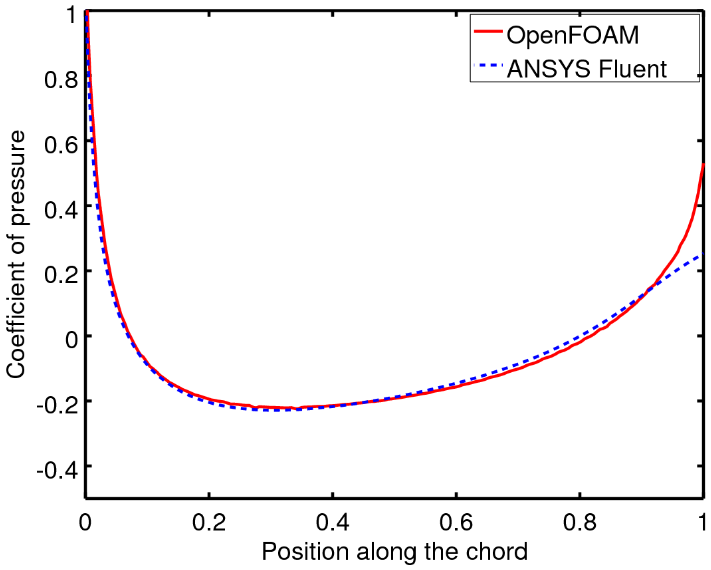

Figure 4.7: Comparison of OpenFOAM® results with ANSYS® *Fluent* results for pressure distribution of Wang shape

Table 4.5 shows the comparison of C_{DV} obtained using OpenFOAM® and ANSYS® *Fluent*

Table 4.5: Comparison of OpenFOAM® results with ANSYS® *Fluent* for Wang shape

	OpenFOAM®	ANSYS® *Fluent*	% error
C_{DV}	0.02730	0.02610	4.6

4.3.4 NPL Shape Cheeseman (2012)

NPL shaped envelope is basically the combination of two half prolate joint at maximum diameter. It is known for its lower drag characteristics. The mathematical representation of prolate is:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1 \tag{4.5}$$

The same surface given by Eq.4.5 can also be obtained by rotating the curve given by Eq.4.6 along X axis

$$y = \begin{cases} \pm b\sqrt{1 - \dfrac{(x-a)^2}{a^2}} & When\ x \leq a \\ \pm b\sqrt{1 - \dfrac{(x-a)^2}{2a^2}} & When\ x \leq a \end{cases} \tag{4.6}$$

Values of constants mentioned above are: *a = 82.485* *and* *b = 24.681*

For the sake of convenience, The surface generated is scaled down by length (l) in all directions.The values obtained for finer grid case using OpenFOAM® are compared with those of using ANSYS® *Fluent*.

The flow conditions and solver parameters used for the simulation are shown in Table 4.6

Table 4.6: Flow conditions and solver parameters for NPL shape

Flow Conditions	Solver parameters
$Velocity = 51\ m/s$	$Number\ of\ cells = 1,063,736$
$Pressure = 87500\ Pa$	$Number\ of\ parallel\ cores = 8$
$Density = 1.057\ kg/m^3$	$Time\ taken\ for\ solution = 2701\ s$
$Reynolds\ number = 3.01 * 10^6$	
$Turbulence\ model = Spalart\ Allmaras$	

Fig. 4.8 shows the comparison of C_P distribution using OpenFOAM® and ANSYS® *Fluent*

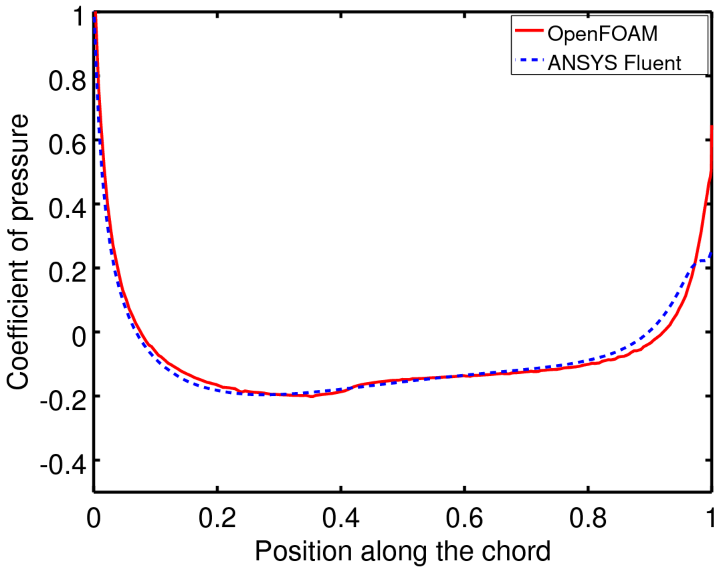

Figure 4.8: Comparison of OpenFOAM® results with ANSYS® *Fluent* results for pressure distribution of NPL profile Cheeseman (2012)

Table 4.7 shows the comparison of C_{DV} obtained using OpenFOAM® and ANSYS® *Fluent*.

Table 4.7: Comparison of OpenFOAM® results with ANSYS® *Fluent* for NPL shape

	OpenFOAM®	ANSYS® *Fluent*	% error
C_{DV}	0.02598	0.02660	2.3

4.3.5 Grid Dependence study

Since the solution is influenced by mesh size, three self similar grids with increasing number of grid points were considered. Table 4.8, 4.9 and 4.10 compares the value for C_{DV} obtained using OpenFOAM® for GNVR Ram and Pant (2010), Wang Wang *et al.* (2009) and NPLCheeseman (2012) shapes respectively.

Table 4.8: Grid Dependence study for GNVR shape

	No. of cells	C_{DV} using OpenFOAM®	% error change
$Coarse$	198315	0.01565	NA
$Fine$	646543	0.01555	0.6
$Veryfine$	1444016	0.01572	1.1

Table 4.9: Grid Dependence study for Wang shape

	No. of cells	C_{DV} using OpenFOAM®	% error change
$Coarse$	1516763	0.02709	NA
$Fine$	2494963	0.02597	10.2
$Veryfine$	2808841	0.02496	7.9

Table 4.10: Grid Dependence study for NPL shape

	No. of cells	C_{DV} using OpenFOAM®	% error change
$Coarse$	220813	0.02709	NA
$Fine$	1520195	0.02597	4.1
$Veryfine$	2521622	0.02496	3.8

According to Suman et al. Suman *et al.* (2011) if percentage changes compared to previous grid is $\leq$ 5% then the value obtained from latest grid can be considered as the most accurate. From the above tables we may observe that for GNVR and NPL profiles, the percentage changes are < 5 %, however for Wang profile, they are >5 %. Thus, we can observe that the solution is nearly independent of grid size in all the cases.

4.4 Observations & Conclusions

From the above results we can observe that OpenFOAM® results match quite well with those obtained using commercial softwares like ANSYS® *Fluent* and other proprietary software like RANS Suman *et al.* (2011). The likely reasons for the minor differences in the values are as follows:

- The results available in literature are 2D simulations using ANSYS® *Fluent* whereas in our case, they are 3D simulations using OpenFOAM®.
- The mesh is different in the case of ANSYS® *Fluent* and OpenFOAM®. OpenFOAM®'s *SnappyHexMesh* generates the mesh by splitting the cells of structured mesh.
- The values of some parameters (e.g., wall functions and wall boundary conditions) had to be assumed because they were not listed in the literature.

Apart from the above reasons, OpenFOAM® solvers are also different from those of ANSYS® *Fluent*. Only a more detailed and systematic computational study and comparison with reliable experimental data can totally establish the efficacy of OpenFOAM® w.r.t. ANSYS® *Fluent* and other proprietary codes. In last three Chapters, we have discussed about geometry, mesh and solver. Now we have essential tools to carry out shape optimization. The methodology for optimization is discussed in next Chapter.

Chapter 5

Surrogate based shape optimization

The usual way of doing any CFD geometry optimization is shown in 5.1. But following this procedure is not feasible because it takes a lots of computational effort and time. Also, grid influences the solution considerably. So, using this method we can no longer guarantee that the changes in the solution of objective function value is because of the changes in the parameters of CAD or if they are coming from changes in the grid. The solution for the problems can be answered by developing a surrogate model.

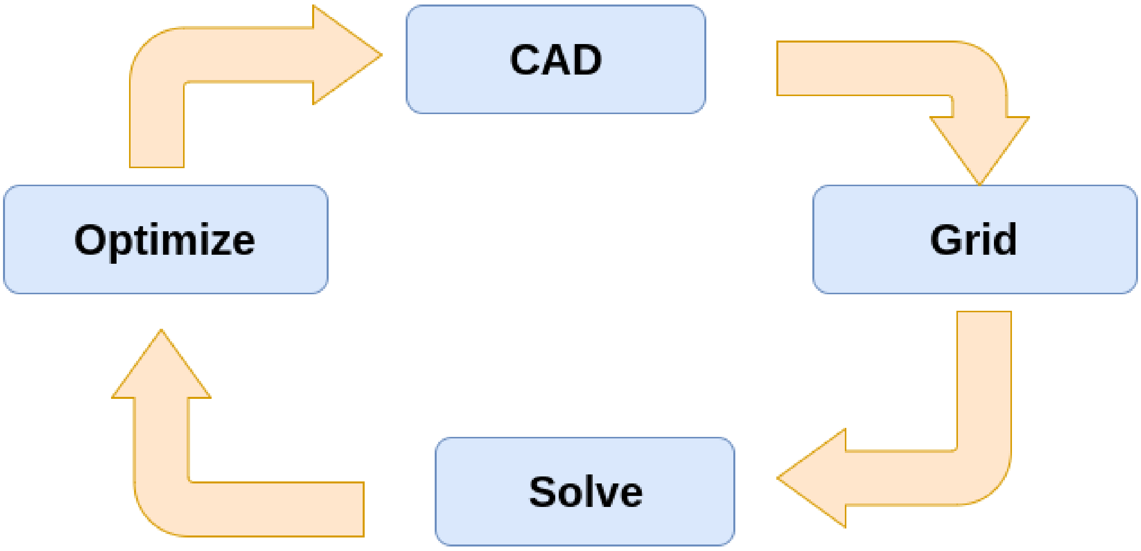

Figure 5.1: Closed Optimization loop

5.1 Surrogate model

Surrogate model is one of the mathematical and statistical techniques used to develop adequate functional relationship between an objective function y(x) and the control or design variables $x_1, x_2....x_k$. In our case, The model acts like a black box for aerodynamic parameters. Given the set of design variables, the model should give volumetric drag coefficient as shown in Figure 5.2 . The work flow associated with the development of a surrogate model is shown in Figure 5.3 . To develop an accurate surrogate, we need to carry out large number of CFD simulations. Creating geometry every time and meshing them is possible but takes a lot of time. Instead we can automate all the processes right from geometry creation to meshing and solving. This has become possible with OpenFOAM®.

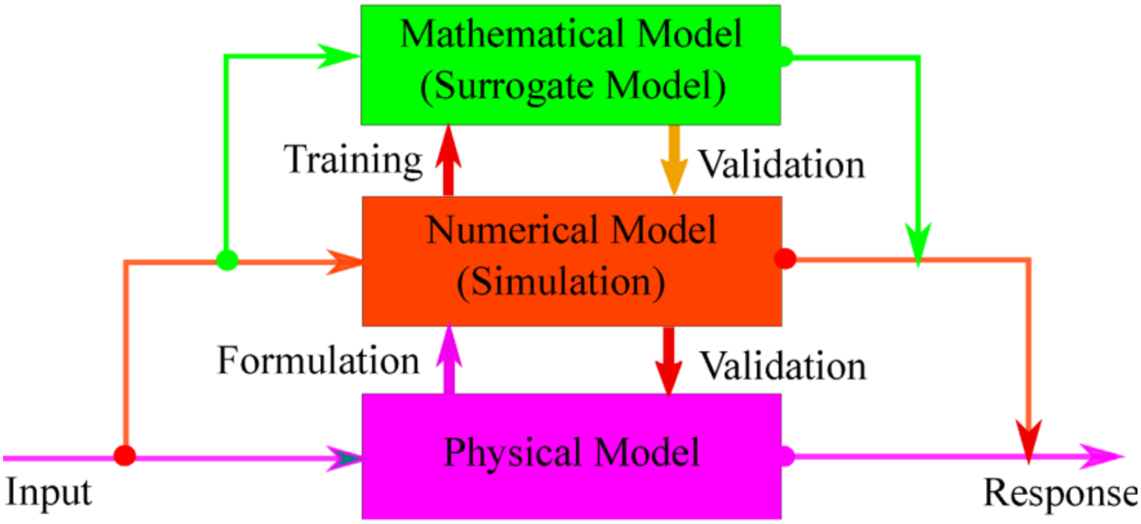

Figure 5.2: Surrogate model definition

5.2 Design of Experiments

Design of Experiments is a technique used to extract maximum possible information from minimum number of computational experiments. This can be accomplished by the selection of a proper and robust scheme for DOE (Design of Experiments). In the present study, an open-source software named Surrogates Toolbox Version 3.0 (Viana, 2011) was used. Apart from many other important features, the toolbox provides an access to several varieties of widely used DOE models. Of the many DOE models available in the toolbox, optimal Latin Hypercube Sampling (OLHS) generated using translational propagation algorithm (TPA) **cite Viana et. al** was chosen, mainly due to its orthogonal structure. Fig. 5.4 shows sampling distribution of points in a unit square.

The number of design experiments depends on the number of shape parameters used for defining geometry. A rule of thumb is that the number of design experiments should

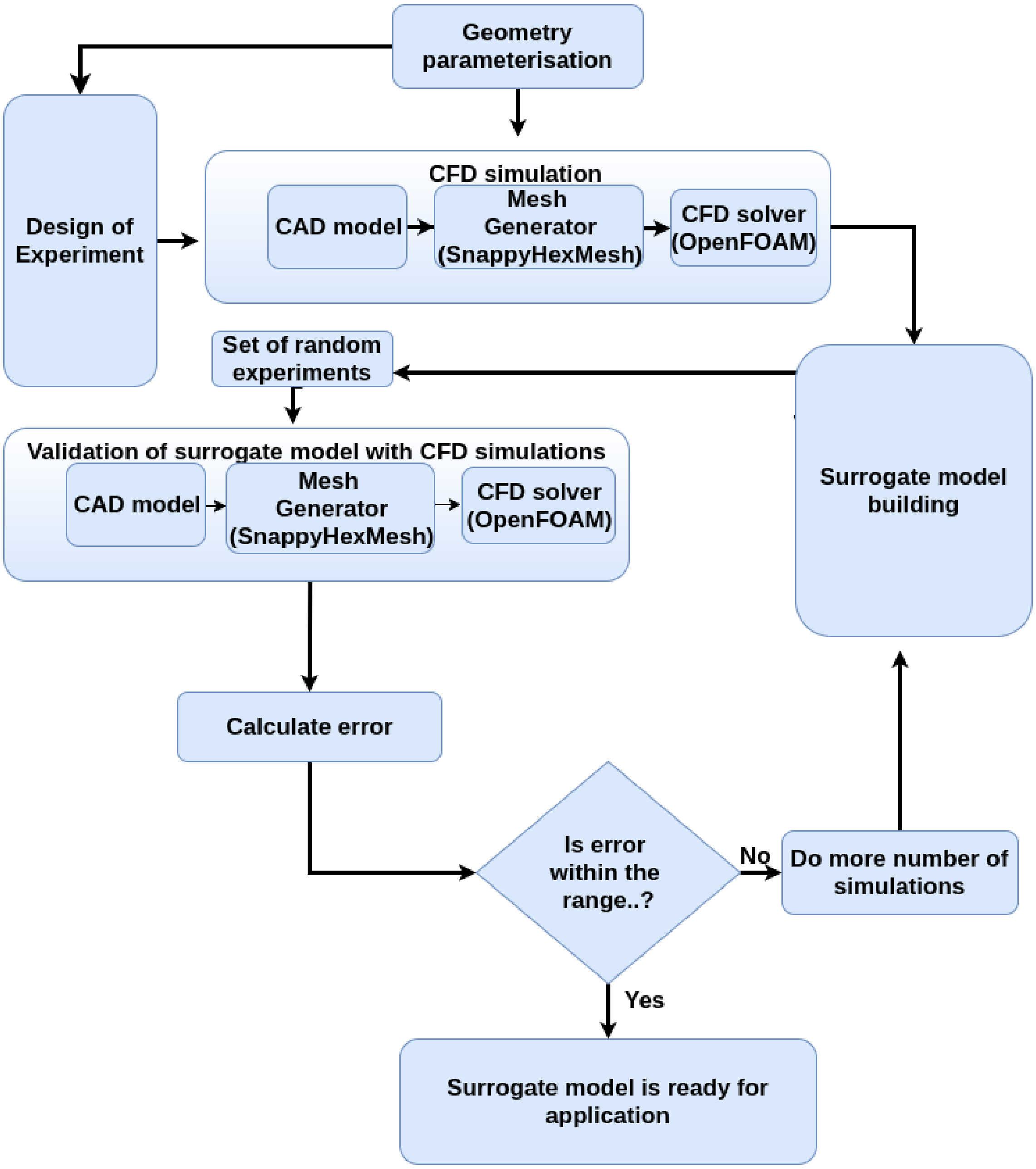
Geometry parameterisation
Design of Experiment
CFD simulation
CAD model
Mesh Generator (SnappyHexMesh)
CFD solver (OpenFOAM)
Set of random experiments
Surrogate model building
Validation of surrogate model with CFD simulations
CAD model
Mesh Generator (SnappyHexMesh)
CFD solver (OpenFOAM)
Calculate error
Is error within the range..?
No
Do more number of simulations
Yes
Surrogate model is ready for application

Figure 5.3: work flow for the development of a surrogate model

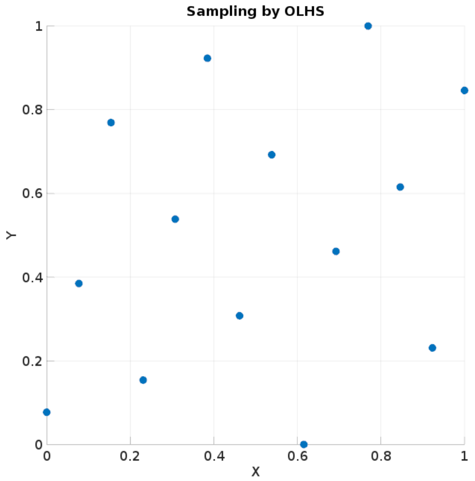

Figure 5.4: Sampling of unit square using OLHS

be ten times the number of shape parametres. However the actual number is decided by obtaining error between surrogate model results and actual CFD results. For the surrogate model to be acceptable, the error should be less than 2 percent. Alam (2016) reported to have made 80 design experiments while developing his surrogate model for 2D bodies of revolution shapes. Kale *et al.* (2005) reported to have studied about 600 feasible shapes using ANSYS® *Fluent* CFD package while developing quadratic response surface using Design Expert package.

For numerical experiments, Kriging surrogate model is reported to be best 32. Kriging is named after the pioneering work of the South African mining engineer D. G. Krige. It is an interpolating method modelled by a Gaussian process governed by prior co-variances, which features the observed data at all design points. Kriging provides a statistic prediction of an unknown function by minimizing its Mean Squared Error (MSE).

The Kriging method in its basic formulation estimates the value of a function at some un sampled location as the sum of two components: the linear regression model $f_i(x)$ (e.g., polynomial trend) of the data with p regressors modelling the drift of the process mean, i.e., the trend, over the domain, and a systematic departure representing low (large scale) and high frequency (small scale) variation components, respectively.

5.3 Test Function for Kriging Surrogate Model

The Himmelblau function is taken as a test function. It has one local maxima and four local minima in the domain x = [-6,6] and y = [-6,6]. The function is defined as

$$f(x,y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2 \tag{5.1}$$

$$x \in [-6, 6]; \quad y \in [-6, 6] \tag{5.2}$$

It has one local maximum at x=-0.270845 and y=-0.923039 where f(x,y)=181.617, and four identical local minima:

$$f(3.0000, 2.0000) = 0.0 \tag{5.3}$$

$$f(-2.8051, 3.1313) = 0.0 \tag{5.4}$$

$$f(-3.7793, -3.2831) = 0.0 \tag{5.5}$$

$$f(3.5844, -1.8481) = 0.0 \tag{5.6}$$

$$\tag{5.7}$$

If a small number of design points are used to create a surrogate model, then the approximate model created is prone to large errors at the trial points. However, the prediction accuracy of a surrogate model cannot be improved merely by taking larger number of design points; that is a function of its behaviour, design space and the required accuracy. Fig. 5.5 shows the effect of increase in number of design points on the Root Mean Square Error (RMSE) of this test function. It is seen that the prediction accuracy of the model improves till 120 design points, after which addition of more design points does not approximate model for surrogate on the right with all the design points and test points. The actual value fo the function and predicted value from the surrogate model at test points are seen to match with each other within 1%.

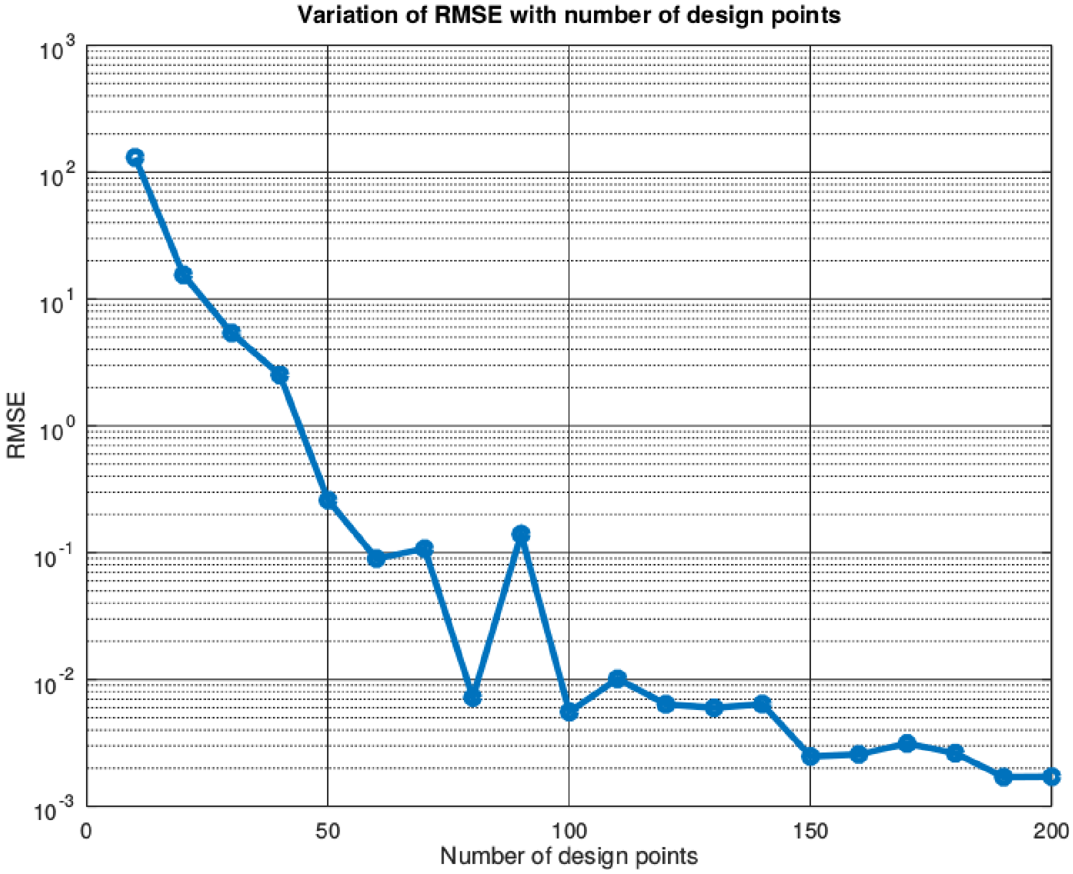

Figure 5.5: Root Mean Square Error with Design Points considered

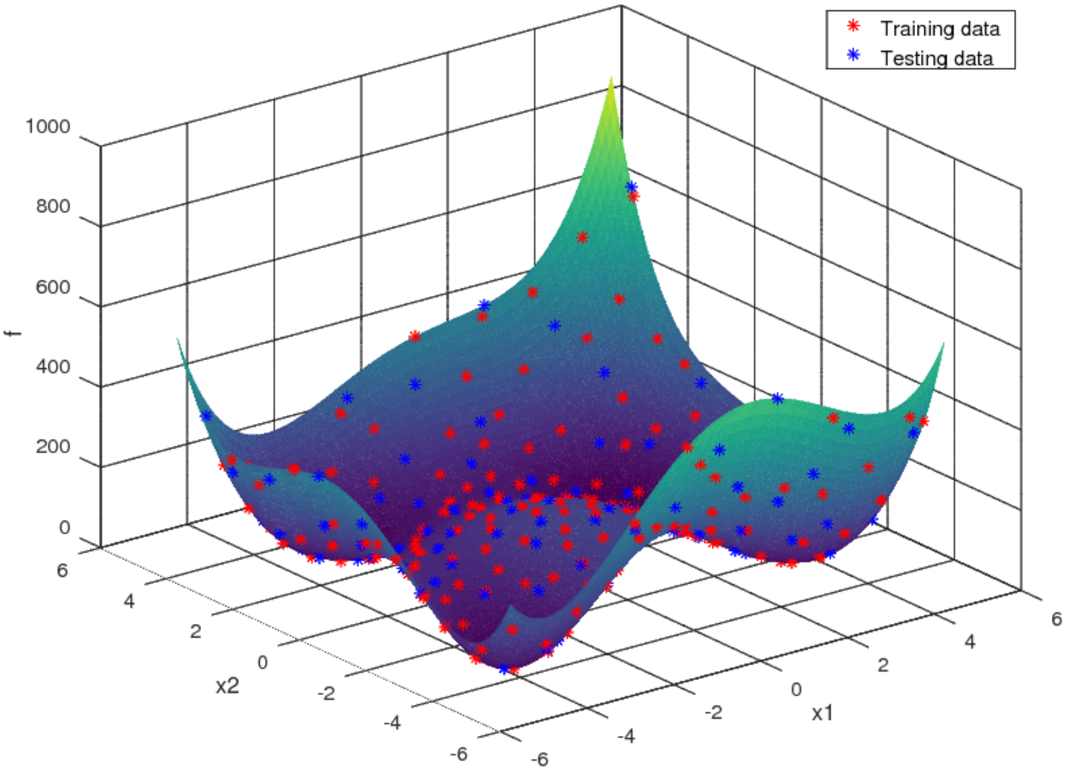

Figure 5.6: Comparison of Himmelblau function with its Surrogate Model

5.4 Coupling of Genetic Algorithm, Surrogate model and testing for modified Himmelblau functions

It is a well known fact that evolutionary algorithms like Genetic algorithm searches for global minimum and outputs that global minimum as final answer. Unlike gradient based methods, it should not get struck in local minimum. In order to check this, we modify the existing Himmelblau function. This is because actual Himmelblau function has four identical local minimum and don't have any global mimima. So, we modify it by adding the term $0.01*(x+3.7793)^2+(y+3.2832)^2$. This term is added to the function value evaluated at all points except (-3.7793,-3.2832). Hence we obtained modified Himmelblau function with one global minima at (-3.7793,-3.2832) and three local minima at other three points. If we optimise this modified Himmelblau function using Genetic algorithm, we have to get the final answer as (-3.7793,-3.2832). The same answer is expected using surrogate model for the modified Himmelblau function. The obtained results are shown in Table.

Table 5.1: Results obtained for modified Himmelblau function

	X	Y	Function value
Actual Function	-3.7793	-3.2832	1.6725e-11
Surrogate model	-3.7791	-3.2828	2.5643e-03
% error	0.006	0.013	2.5643e-03

Chapter 6

Surrogate model for CFD

To develop a surrogate model which mimics the behaviour of CFD in finding the drag coefficient at zero degree angle of attack, it is essential to identify the limits of design parameters, perform Design Of Experiments (DOE) on them, perform CFD analysis at these DOE points and fit a surrogate model. The whole processes is described in the subsections below.

6.1 Mapping design variables

The first task before creation of DOE was the proper definition of the upper and lower limits for the six design variables (viz., $m, r_0, r_1, C_p, \frac{l}{d}$ *and scale_y*). The upper and lower limits of design variables are calculated using the initial sizing done by Alam (2017). The design space is confined as mentioned in Table 6.1

Table 6.1: Design Space

Design Parameters	Min.	Max.
$Point\ of\ Max.\ Dia., m$	0.35	0.50
$Nose\ Radius, r_o$	0.20	0.80
$Tail\ Radius, r_1$	0.1	0.5
$Prismatic\ Coeff., C_p$	0.55	0.70
$Fineness\ Ratio, \frac{l}{d}$	2.50	6.00
$Scaling\ in\ Y\ direction,\ scale\_y$	1.00	5.00

Eighty candidate points were first obtained by Surrogates Toolbox **cite surrogate toolbox** using the Optimal Latin Hypercube Sampling (OLHS) method. The generated points were then mapped to the shapes corresponding to envelope volume (V_{env}) of 250000 m^3 .

6.2 Design of Experiments study

As mentioned previously in section 5.2, DOE study is used to extract maximum amount of information from every experiment conducted. This is especially needed when we are carrying out expensive computer or real life experiments. While developing surrogate model for CFD, each CFD analysis will take about 3 hours to complete. Hence to optimally use the computation resources, the total number experiments needs to be as low as possible but extracting maximum information about function behaviour. So, initially the number of experiments are taken as 80 and the design points are generated using the Optimal Latin Hypercube Sampling (OLHS) technique present in SURROGATES Toolbox by Viana *et al.* (2014). The obtained points are given in Appendix.

6.3 CFD analysis on the points obtained from DOE

All the pre-processing required to perform CFD analysis like creating the geometry, deciding the size of domain, meshing the domain has been automated using Octave and C++ scripting. The scripts are shared in Appendix.

6.4 Building the surrogate model

A brief overview of different surogate models by Luo and Lu (2014) is given below.

6.4.1 Polynomial response surface

Polynomial response surface is the simplest approximation method to build surrogate models Forrester and Keane (2009). The most widely used polynomial response surface model is the second-order polynomial model which has the following form.

$$y = \beta_0 + \sum_{i=1}^{n} \beta_i x_i + \sum_{i=1}^{n} \sum_{j \geq i}^{n} \beta_{ij} x_i x_j + .. \tag{6.1}$$

where β_0, β_i, β_{ii}, and β_{ij} are the regression coefficients, n is the number of variables, xi and xj are the variables. Using least square method (LSM), the regression coefficients can be solved.

6.4.2 Radial basis function

RBF is a 3-layer feed forward neural network consisting of an input layer, a hidden layer, and an output layer Shen *et al.* (2011). X is an N dimensional input vector. The output of the neurons in the RBF hidden layer is assumed as:

$$q_i = \Phi(||X - c_i||) \tag{6.2}$$

where c_i is the center associated with the ith neuron in the radial basis function hidden layer, i = 1, 2,...,H, where H is the number of hidden units, $||X - c_i||$ is the norm of Xc_i, $\Phi(.)$ is a radial basis function Chen *et al.* (1991). Outputs of the kth neuron in RBF output layer are linear combinations of the hidden layer neuron outputs as:

$$y_k = \sum_{i=1}^{H} w_{ki} q_i - \theta_k \quad (k = 1, 2, ..., M) \tag{6.3}$$

where w_{ki} is the connecting weights from the i^{th} hidden layer neuron to the k^{th} output layer, θ_k is the threshold value of the k^{th} output layer neuron.

6.4.3 Kriging

The kriging method was developed by the French mathematician Georges Matheron based on the Master′s thesis of Daniel Gerhardus Krige Matheron (1963), it was first used as a geostatistical method. Sacks *et al.* (1989) firstly introduced kriging method as a surrogate modelling method, in the paper of Sacks *et al.* (1989), kriging surrogate model was also called design and analysis of computer experiment (DACE). The kriging model is a combination of two components Queipo *et al.* (2005): deterministic functions and localized deviations.

$$Y(x) = \sum_{i=1}^{k} f_i(x)\beta_i + z(x) \tag{6.4}$$

where $\sum_{i=1}^{k} f_i(x)\beta_i$ is the term of deterministic functions, β are coefficients of deterministic functions, fi(x) are k known regression functions, which are usually polynomial functions. z(x) is term of localized deviations with mean zero, variance ÏČ2, and covariance expressed as:

$$Cov[z(x_i), z(x_j)] = \sigma^2 R(x_i, x_j) \tag{6.5}$$

where $R(x_i, x_j)$ is the correlation function between any two of the ns samples The common types of correlation functions are linear function, exponential function, Gauss function, spline function, etc. (Ryu et al. 2002). The prediction of unsampled points response y(x) can be expressed as:

$$\hat{y}(x) = f(x)^T\beta + r^T R^{-1}(Y - F\beta) \tag{6.6}$$

where Y is the vector of ns samples response, r is the correlation vector between samples and prediction points.

$$r = [R(x, x_1), R(x, x_1), ..., R(x, x_1)]^T \tag{6.7}$$

$$F = [f(x_1), f(x_2), ...f(x_{n_x})]^T. \tag{6.8}$$

6.5 Toolbox used for different surrogate models

Table 6.2 details the different surrogates used during this investigation. The SURROGATES toolbox was also used for easy manipulation of the surrogates.

Table 6.2: Setup for the set of used surrogates. The DACE 20, RBF 15 and SURROGATES 31 toolboxes were used to run the kriging, radial basis function and polynomial response surface respectively.

Table 6.2: Different surrogates used during this investigation

Surrogate	Details
KRG	Kriging model: constant trend function and Gaussian correlation
PRS	Polynomial response surface: Second degree polynomial
RBF	Radial basis function: Multiquadric basis function

Chapter 7

Training data of CFD results to train surrogate model

In our case, 80 DOEs are generated and steady state 3D CFD analysis has been performed for these 80 shapes. The solver parameters used are as follows:

Table 7.1: Flow conditions and solver parametres for all DOE shapes

Parameters	Value
Reynolds number based on volume Re	$3.01 * 10^6$
Pressure, N/m2	87500
Density, Kg/m3	1.057
Speed, m/s	51
Mesh	Hexahedral mesh using blockMesh and snappyHexMesh
Solver	SimpleFOAM (Incompressible steady state solver)
Turbulent..?	Yes
Turbulence model	K-Omega SST

7.1 Grid Convergence study

Effect of grid size on the overall results by changing the number of cells in self similar mesh has been carried out. Below is the plot of variation of pressure and viscous drag with number of cells for a simulation. As mentioned by Suman *et al.* (2011) , when the percentage change in the values is less than 5 %, then the solution is reported to have grid convergence.

Table 7.2: Grid convergence study

Cells	Pressure Drag (N)	Viscous Drag (N)
376380	1.1	11.02
604375	2.2	11.68
840565	2.02	12.1
994864	2.09	12.24
1563471	2.12	12.5

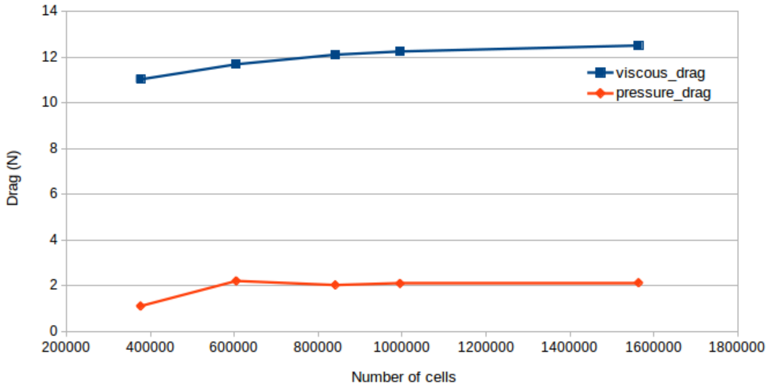

Figure 7.1: Variation of pressure and viscous drag with number of cells

Chapter 8

Calculation of Hoop Stress

In order to maintain positive pressure inside the envelope, three main loadings are considered to estimate the internal over pressure (ΔP) viz., the loading due to dynamic pressure, aerodynamic loading, and hydro-static pressure as suggested by Gupta & Malik Gupta and Malik (2002) . A brief description of these three components of the module is as given below.

Dynamic Pressure: Dynamic pressure loading acts on the front portion of the aerostat envelope and tries to make a depression on the envelope surface, its diameter depends on the region of highly stressed area which is normally 7 to 9 % of the envelope length. To maintain the shape of the envelope, the internal pressure is kept slightly higher than dynamic pressure, normally 15% as suggested by Gupta & Malik Gupta and Malik (2002) as per Eq. 8.1

$$\Delta P_{dyn} = 1.15\left(\frac{1}{2}\rho_{air}v^2\right) \tag{8.1}$$

Aerodynamic Pressure: Aerodynamic pressure loading results from applying the stability conditions to the airship. Coefficient of pressure, as observed normally for airship profiles is in the range of 0.30 to 0.35 from the leading edge, but normally for such shapes, it is assumed at the maximum diameter. is used for calculating the pressure due to this loading.

$$\Delta P_{aer} = c_p\left(\frac{1}{2}\rho_{air}v^2\right) \tag{8.2}$$

Hydrostatic Pressure: Hydrostatic pressure loading is due to the difference between the height of the top and bottom of aerostat and could be quite substantial of the aerostat diameter in large. The hydrostatic pressure as observed at mean centerline axis of the envelope is calculated at maximum diameter using:

$$\Delta P_{hyd} = \frac{1}{2}(\rho_{air} - \rho_{He})gD_c \tag{8.3}$$

The sum of pressure due to these three loadings gives the total required internal pressure. Since the aerostat envelope diameter is more than twenty times of the thickness of the material; it can be considered as a very thin shell and hence hoop stress is calculated in terms of the circumferential unit load as shown in Eq. 8.4

$$\Delta P = \Delta P_{dyn} + \Delta P_{aer} + \Delta P_{hyd} \tag{8.4}$$

To estimate the longitudinal stress (σ_l) and the circumferential hoop stress (σ_h) developed in an envelope, an equivalent cylindrical envelope is considered, having the same surface area and volume as the actual airship envelope, as shown in Fig 8.1

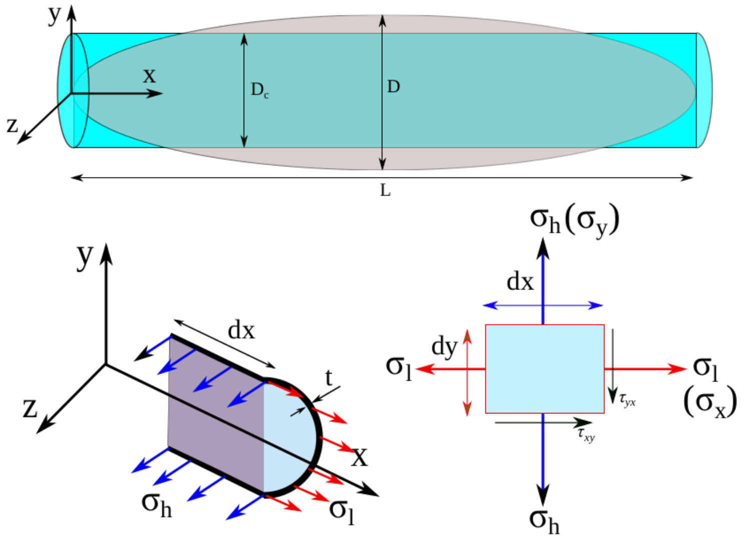

Figure 8.1: Stress representation in envelope

Hence, σ_l and σ_h developed in an envelope are estimated as:

$$\sigma_h = \frac{\Delta P D_c}{2} \tag{8.5}$$

$$\sigma_l = \frac{\Delta P D_c}{4} \tag{8.6}$$

To maintain the aerodynamic shape and rigidity of an airship envelope some pressure difference ($\Delta P'$) is also needed, which can be determined by bending moment estimation.

8.0.1 Bending Moment Calculation

An empirical relation for maximum bending moment given by FAA 5 can be expressed as:

$$M = 0.029[1 + (f - 4)(0.5624L^{0.02} - 0.5)]\, \rho_{air}.u.v.V_{env}L^{0.25} \tag{8.7}$$

Where D, is the diameter of airship envelope with the where u and v are the gust and wind velocity respectively. Eq. 8.7 is valid for fineness ration between 4 to 6. For fineness ratio less than 4, f can be assumed as 4. The point at which the maximum bending moment occurs, i.e., near the center of gravity, σ_l at the top of this must be greater than or equal to zero. Therefore, the equation for σ_l can be written as:

$$\sigma_l = \frac{\Delta P' \pi R^2}{2\pi R} - \frac{MRT}{I} \geq 0 \tag{8.8}$$

Where R is the radius of airship envelope at the point of maximum bending. t is the thickness the envelope material. I is the second moment of area, which has the value of $\pi R^3 t$. Minimum pressure required $\Delta P'$ to maintain the shape of the envelope can be calculated as:

$$\Delta P' = \frac{16M}{\pi D^3} \tag{8.9}$$

Where D is the diameter of airship envelope with the assumption that, the location of maximum bending is same as the location of maximum diameter. Total differential pressure âĹĘP t can be obtained as:

$$\Delta P_t = max(\Delta P, \Delta P') \tag{8.10}$$

In that case, σ_h and σ_l can be calculated as:

$$\sigma_h = \frac{\Delta P D_c}{2} \tag{8.11}$$

$$\sigma_l = \frac{\Delta P D}{4} + \frac{4M}{\pi D^2} \tag{8.12}$$

von-Mises Stress Calculation: From the principle stresses, von-Mises stress can be expressed as:

$$\sigma_v = \left[\frac{(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2}{2}\right]^{\frac{1}{2}} \tag{8.13}$$

Stresses are considered only in 2-D planes due to very low thickness of envelope compared to its diameter. Therefore ÏČ 3 = 0, which simplifies the equation as:

$$\sigma_v = \sqrt{\frac{(\sigma_1 - \sigma_2)^2 + \sigma_1^2 + \sigma_2^2}{2}} \tag{8.14}$$

Stresses are considered only in 2-D planes due to very low thickness of envelope compared to its diameter. Therefore $\sigma_3 = 0$, which simplifies the equation as:

$$\sigma_v = \sqrt{\sigma_1^2 + \sigma_2^2 - \sigma_1 \sigma_2} \tag{8.15}$$

Chapter 9

Future work

Appendix A

Table of design points obtained from OLHS

References

Alam, M. I., 2016, "Multidisciplinary design optimization of stratospheric airship," in *International Council of the Aeronautical Sciences (ICAS 2016)*

Alam, M. I., 2017, "Multi-objective multidisciplinary design optimization of high altitude airships,"

Alam, M. I., Subhani, S., and Pant, R. S., 2014, "Multidisciplinary shape optimization of stratospheric airships," in *International Conference on Theoretical, Applied, Computational and Experimental Mechanics (ICTACEM-2014)*

Ceruti, A., Voloshin, V., and Marzocca, P., 2013, "Multi-Disciplinary Design Optimization of Unconventional Airship Configurations with Heuristic Algorithms," *54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference* **51**, 1–20.

Cheeseman, I., Aerodynamics, Chapter-02, Khoury, G. A., and Gillett, J. D., 1999, "Airship technology," (Cambridge Aerospace Series: 10, Cambridge University Press, ISBN: 0 521 430 747).

Cheeseman, I., 2012, "Airship technology," Chap. 2:Aerodynamics (Cambridge University Press). pp. 29–30.

Chen, Q., Zhu, M., and Sun, K., 2010, "Analysis to Effects on Conceptual Parameters of Stratospheric Airship with Specified Factors," *Journal of Computers* **6**, 1055–1062.

Chen, S., Cowan, C. F., and Grant, P. M., 1991, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Transactions on neural networks* **2**, 302–309.

Forrester, A. I., and Keane, A. J., 2009, "Recent advances in surrogate-based optimization," *Progress in Aerospace Sciences* **45**, 50–79.

Gawale, A., and Pant, R., 2005 sep, “Initial Sizing and Sensitivity Analyses of Stratospheric Airships for Psuedolite Based Precision Navigation System,” *AIAA 5th ATIO and16th Lighter-Than-Air Sys Tech. & Balloon Systems Conferences*, 1–15.

Key: Gawale2005

Annotation: \cite{Gawale2005}.

Gertler, M., Jan. 1950, “Resistance Experiments on a Systematic Series of Streamlined Bodies of Revolution For Application to the Design of High-Speed Submarines,” in *David Taylor Model Basin* (Washington DC).

Ghani, O. A., 2013, “Design optimization of aerodynamic drag at the rear of generic passenger cars using nurbs representation,” , 30 – 36.

Gupta, S., and Malik, S., 2002, “Envelope details for demo airship,” *Aerial Delivery Research and Development Establishment (ADRDE), Agra*

Harada, K., Eguchi, K., Sano, M., and Sasa, S., 2003, “Experimental study of thermal modeling for stratospheric platform airships,” *AIAA*

Jekabsons, G., 2009, “ RBF: Radial Basis Function interpolation for MATLAB/OCTAVE, Riga Technical University, Latvia, version 1.1 ed.,,”

Kale, S. S., Joshi, P., and Pant, R. R., 2005, “A generic methodology for determination of drag coefficient of an aerostat envelope using CFD,” *AIAA 5th ATIO and16th Lighter-Than-Air Sys Tech. and Balloon Systems Conferences*, 1–16.

Kanikdale, T., Marathe, A., and Pant, R., 2004, “Multi-disciplinary optimization of airship envelope shape,” in *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, p. 4411.

Liu, P., Fu, G.-y. Y., Zhu, L.-j. J., and Wang, X.-l. L., 2013 sep, “Aerodynamic characteristics of airship Zhiyuan-1,” *Journal of Shanghai Jiaotong University (Science)*, 679–687.

Liu, Q., Wu, Z., Zhu, M., and Xu, W., 2014, “A comprehensive numerical model investigating the thermal-dynamic performance of scientific balloon,” *Advances in Space Research* **53**, 325–338.

Lophaven, S. N., Nielsen, H. B., and Søndergaard, J., 2002, *DACE: a Matlab kriging toolbox*, Vol. 2 (Citeseer).

Luo, J., and Lu, W., 2014, "Comparison of surrogate models with different methods in groundwater remediation process," *Journal of earth system science* **123**, 1579–1589.

Matheron, G., 1963, "Principles of geostatistics," *Economic geology* **58**, 1246–1266.

Narayana, C. L., and Srilatha, K. R., July 2000, "Analysis of aerostat configurations by panel methods," in *BLISS Project Document CF 0010* (National Aerospace Laboratories, Bangalore, India).

Queipo, N. V., Haftka, R. T., Shyy, W., Goel, T., Vaidyanathan, R., and Tucker, P. K., 2005, "Surrogate-based analysis and optimization," *Progress in aerospace sciences* **41**, 1–28.

Ram, C. V., and Pant, R. S., 2010 may, "Multidisciplinary Shape Optimization of Aerostat Envelopes," *Journal of Aircraft* **47**, 1073–1076.

Key: Ram2010
Annotation: From Duplicate 2 (Multidisciplinary Shape Optimization of Aerostat Envelopes - Ram, C. Vijay; Pant, Rajkumar S.) [1] C. V. Ram and R. S. Pant, âĂIJMultidisciplinary Shape Optimization of Aerostat Envelopes,âĂİ J. Aircr., vol. 47, no. 3, pp. 1073âĂŞ1076, May 2010.

Ran, H., Thomas, R., and Mavris, D., 2007, """A Comprehensive Global Model of Broadband Direct Solar Radiation for Solar Cell Simulation"," in *45th AIAA Aerospace Sciences Meeting and Exhibit*, January, pp. 1–16.

Key: Ran2007
Annotation: [1] H. Ran, R. Thomas, and D. Mavris, âĂIJA comprehensive global model of broadband direct solar radiation for solar cell simulation,âĂİ AIAA2007-33, no. January, pp. 1âĂŞ16, 2007.

Rehmet, M. A., and Kroplin, B. H., 2000 1st-5th July, "Recent Developments on High Altitude Platforms," in *3rd International Airship Convention and Exhibition* (Friedrichshafen, Germany). pp. Paper B–08.

Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P., 1989, "Design and analysis of computer experiments," *Statistical science*, 409–423.

Shen, W., Guo, X., Wu, C., and Wu, D., 2011, "Forecasting stock indices using radial basis function neural networks optimized by artificial fish swarm algorithm," *Knowledge-Based Systems* **24**, 378–385.

Suman, S., Lakshmipathy, S., and Pant, R., 2011, "Evaluation of the assumed-transition-point criterion in context of RANS simulations around Lighter-Than-Air vehicles," , 1–10.

Viana, F. A. C., 2011, "SURROGATES Toolbox User ' s Guide," , 0–27.

Viana, F. A., Simpson, T. W., Balabanov, V., and Toropov, V., 2014, "Metamodeling in multidisciplinary design optimization: How far have we really come?." *AIAA JOURNAL* **52**

Wang, Q., Chen, J., Fu, G., and Duan, D., 2009 Nov., "An Approach for Shape Optimization of Stratosphere Airships based on Multidisciplinary Design Optimization," *Journal of Zhejiang University SCIENCE A* **10**, 1609–1616.

Wang, X.-L., Fu, G.-Y., Duan, D.-P., and Shan, X.-X., 2010 jul, "Experimental Investigations on Aerodynamic Characteristics of the ZHIYUAN-1 Airship," *Journal of Aircraft* **47**, 1463–1468.