RESEARCH ARTICLE OPEN ACCESS

Improving the Hydraulic Efficiency of Centrifugal Pumps through Computational Fluid Dynamics Based Design optimization

Abdellah Ait moussa*, Lin Yunhao*
*(Department of Engineering and Physics, University of central Oklahoma, 100 University Drive, Edmond, Oklahoma, USA 73034.)

ABSTRACT
The design and optimization of turbo machine impellers such as those in pumps and turbines is a highly complicated task due to the complex three-dimensional shape of the impeller blades and surrounding devices. Small differences in geometry can lead to significant changes in the performance of these machines. We report here an efficient numerical technique that automatically optimizes the geometry of these blades for maximum performance. The technique combines, mathematical modeling of the impeller blades using non-uniform rational B-spline (NURBS), Computational fluid dynamics (CFD) with Geometry Parameterizations in turbulent flow simulation and the Globalized and bounded Nelder-Mead (GBNM) algorithm in geometry optimization.
Keywords-Fluid dynamics, Optimization, Golobalized and bounded Nelder-Mead, CFD based optimization

I. INTRODUCTION

Pumps are mechanical devices that add energy to a fluid as a result of the dynamic interactionbetween the device and the fluid.Two types of pumps can be distinguished, centrifugal and positive displacement pumps. In centrifugal pumps energy is transferred to the fluid through contact with a set of rotating blades. In positive displacement pumps a portion of the fluid is trapped and moved in a given direction. The overall efficiency of a pump is affected by losses in the pump; distinction is made betweentwo primary types of losses. Mechanical losses, such as those in the bearing and shaft seal and hydraulic losses which may incorporate flow separation, mixing, recirculation, leakage, and cavitation [1-2]. The hydraulic efficiency of a centrifugal pump depends significantly on the impeller and casing geometries and small changes in geometrical details can lead to large changes in performance [3].

Designers have been challenged to provide centrifugal pumps that can operate more efficientlyand quietly, especially for large-scale pump operation in industrial plants where energy savingis a significant issue.Techniques ranging from the traditional trial and error design approach based on the one or two dimensional theory and semi-empirical equations [4], in addition to the use of analytical functions to parameterize the surface geometry of impeller channels [5-6] were all employed. The inverse based design methods by far are the most accurate [7]. The inverse method, involves determining the corresponding blade contour for a given set of aerodynamic properties,such as surface-velocity distribution, and surface-pressure distribution. The final product is an inversely-designed blade which has the prescribed performance in terms of the prescribed distribution. Different distributions will produce different blade contours and the challenge is to identify the most beneficial target distribution to be optimized. To over compensate the drawback in the inverse design method due to the prescribed target distribution, we proceed by using a different approach. In the present development, we use a parametric equation to describe the blade angle, then proceed with an optimization algorithm to identify these parameters under prescribed operating conditions.

The complexity of the flow in a turbo machine is primarily due to the three dimensional developed structures involving turbulence, secondary flows, unsteadiness and others. Tremendous advances however have been made to accurately simulate the flow field inside these devices for given blade geometries [8-10]. Many of these algorithms are available in most Finite Element Simulation (FEA) commercial packages such as ANSYS-CFX and Fluent. In the same sense, several geometrical parameters are usually involved in the design process of centrifugal pumps and to accurately select the most optimal configuration, a hydraulic designer must take into account the local flow field inside the pump during on and off design operations. Accordingly, a direct connection must be established between the design and flow simulation to systematically and accurately improve the performance of these turbo machines. The present development is based on this perception and will be discussed in detail in the following sections.

This paper is divided into three major parts. In the first, we introduce the technique of optimization and components used in the process of computing and maximizing the hydraulic efficiency of a centrifugal pump. Next, we apply this technique to a test model of a centrifugal pump. In the final section we conclude with a summary and a discussion of future work.

II. THECOMPUTATIONAL TECHNIQUE

In this section, we shall describe the general structure of the computational technique used in the optimization that we apply to a specific system in section III. We shall present this general case first and then indicate briefly how the results will simplify for our special cases.

II.1 Mathematical Modeling of the Surface Geometry of the Blade

Impeller blades usually have complicated curved shapes and a common way to describethis shape is to define an impeller blade angle. The blade angle β is the angle between the blade contour and the circumferential direction, i.e., a circular arc around the axis of rotation as depicted in Fig.1. In general the blade angle depends on the meridional distance (x_m) defined along the meridional line from the leading edge of the blade.

To mathematically model the impeller blade, we begin with a given meridional projection on the (r, z) plane as shown on the inset of Fig.1. A line on the meridional plane from leading to trailing edge is spun about the axis of rotation of the pump (z-axis) to form a surface of revolution. The final blade contour over this surface is formed by moving each point on the meridional line over the surface of revolution with an angle θ around the axis of rotation, and along a direction defined by the blade angle β. Mathematically, this transformation is described by the following equation.

$$\tan(\beta) = \frac{dx_m}{r d\theta} \tag{1}$$

wheredx_m is an infinitesimal arc length in the meridional direction such that

$$dx_m = \sqrt{dr^2 + dz^2} \tag{2}$$

To successfully incorporate this transformation, we introduce a new variable η such that

$$d\eta = \frac{dx_m}{r} = \frac{dr}{r}\sqrt{1 + \left(\frac{dz}{dr}\right)^2} \tag{3}$$

With this variable equation (1) reduces to

$$d\theta = \tan\left(\frac{\pi}{2} - \beta\right) d\eta \tag{4}$$

To integrate this equation, we need an explicit expression of the blade angle as a function of η. To this end, we approximate $\tan(\pi/2 - \beta)$ with a polynomial of the variable η as emphasized in (5), where a_i are constants to be determined later by the optimization algorithm. We then integrate the resulting equations to get θ as a function of η (6). Note that when all the a_i buta_o are zero, we recover the logarithmic blade where the blade angle is constant over the blade length. The additional terms when different from zero, are corrective terms devised to change the curvature over the logarithmic blade and along the blade length.

$$\tan\left(\frac{\pi}{2} - \beta\right) = a_o + a_1\eta + a_2\eta^2 + \cdots \tag{5}$$

$$\theta = \theta_o + a_o\eta + \frac{a_1}{2}\eta^2 + \frac{a_2}{3}\eta^3 + \cdots \tag{6}$$

whereθ_o is an integration constant

The variable η depends on r and z; however, since we started from a known meridional line, we can directly integrate (3) to get η as a function of r.

In practice, the surface of revolution can be mapped onto a two dimensional plane (R, α) using the Prasil transformation [11] in (7), where R is the distance from the axis of rotation (z-axis).

$$\frac{d\alpha}{d\theta} = \frac{\alpha}{\theta} = 1, \qquad \frac{d\eta}{d\alpha} = \frac{1}{r}\frac{dx_m}{d\theta} = \frac{1}{R}\frac{dR}{d\alpha} \tag{7}$$

Using this transformation and the definition in (3), we find

$$R = R_o\, e^{\eta} \tag{8}$$

where R_o is an integration constant.

Accordingly, the final blade contour can be formed from a point on the meridional line defined by the coordinates (r, z), as it transforms to a point on the surface of revolution defined by the coordinates (R, θ).

To numerically implement these transformations, we proceed from a given meridional projection. A number of control points are selected over the surface and NURBS curves from leading to trailing edge are constructed. These curves constitute the meridional lines that are then transformed according to the steps described in this section to produce the final blade surface geometry. The diagram in Fig.2 describes the steps used in the numerical application of this transformation. This algorithm with preselected values of the constants a_i, will be called several times during the optimization that we introduce in section II.5.

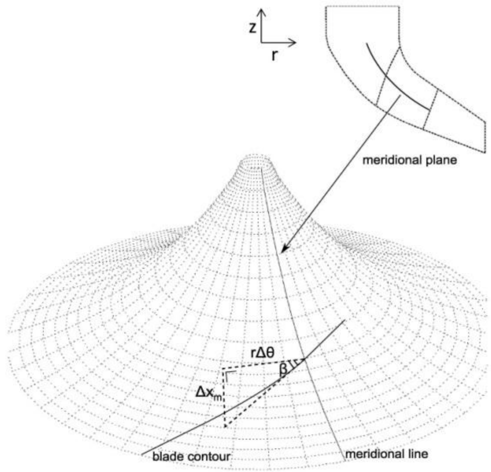


Figure 1: Blade angle

II.2 Graphical Modeling of the Blade, rotor and volute

From the mathematical model introduced in section II.1, we exported a file.out that contained the coordinates of numerous control points over the final surface geometry of the blade. In the current step, we import these coordinates using computer-aided engineering (CAE) tools and plot the vertices on a graphical interface as shown in Fig.3 (left). Most sophisticated graphic creation tools such as SolidWorksprovide an interfacefor using splines; these tools in conjunction with the imported data file were used to sketch the different contours of the blade surface. The physical surface of the blade is then mapped over the splines, as shown in Fig.3 (middle). Finally, the blade surface is offset and filled to produce the final blade geometry as depicted in Fig.3 (right). The fluid domain is then added and the volume of the blades subtracted as illustrated in Fig.4 for one of the blade configurations. The volute shown in the same figure was kept in a separate file. Reasonable lengths of inlet and outlet extensions were added to the physical model to reduce the unavoidable effect of inlet and outlet boundaries on the final flow solutions.

The steps described in this section will later be automated to work with different blade configurations and produce the computational domain needed for the finite element simulation (FEA) that we introduce in section II.3.

II.3 Finite Element Simulation

We used the ANSYS-CFX analysis system in ANSYS Workbench. The impeller blades and volute models were imported and meshed. ANSYS Workbench offers a robust and easy to use set of meshing tools. These tools have the benefit of being highly automated along with having a moderate to high degree of user control. Based on the analysis system utilized, the Mesher in ANSYS Workbench uploads a set of default parameters that will result in a mesh that is more favorable to the solver used. By means of global and local mesh controls, the user can easily modify the mesh parameters. In this paper we adopted a physics based meshing, the physics preference was set to CFD and solver to CFX. An unstructured mesh with tetrahedralcells was used for the zones of the impeller and volute.

The mesh was refined in the near tongue region of thevolute as well as in the regions close to the leadingand trailing edge of the blades. An inflation layer wasadded over the surfaces of the blades; the prisms were grown with a first aspect ratio of 10and a growth factor of 1.2 extruding 5 layers. The grids generated for one of the blade configurations are shown in Fig.5. Note that the blades did not extend to the edge of the impeller, this is because we constrained the blade to maintain a fixed length throughout the analysis and this can only be achieved if the trailing edge is kept strictly within the impeller. The impeller and volute meshes were then imported to the CFX solver and a steady state analysis was conducted in conjunction with the following boundary conditions. At the inlet of the computational domain, the mass flow rate, the turbulence intensity, and the total pressure were specified. At the outlet (end of volute diffuser), the mass flow rate was specified. The volute casing and intake section walls were in stationary frame and modeled using no-slip boundary condition. The meshes of the impeller and volute casing were connected by means of frozen rotor interface. A blend factor of 0.90 was used in the advection scheme, and CFX default convergence criteria of 10^{-4} were adopted. The three dimensional incompressible Navier-Stokes equations with a standard $k-\varepsilon$ turbulence model were then solved and relevant physical properties such as the head rise,break power and total pressure at the inlet of the impeller were calculated and exported to a file.out.

The steps described in this section will later be automated and used several times in the optimization that we introduce in the next section.

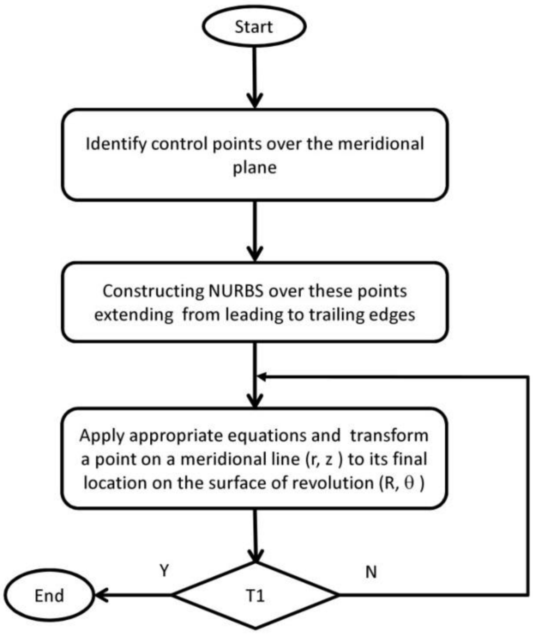


Figure 2: Diagram for generating control points over the transformed blade geometry. (T1: last vertex)

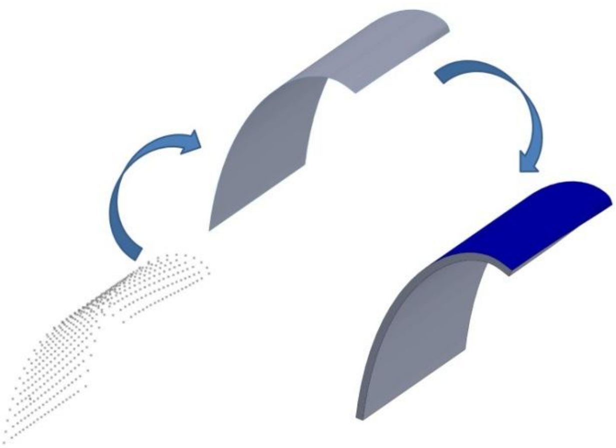


Figure 3: Graphical design of a blade

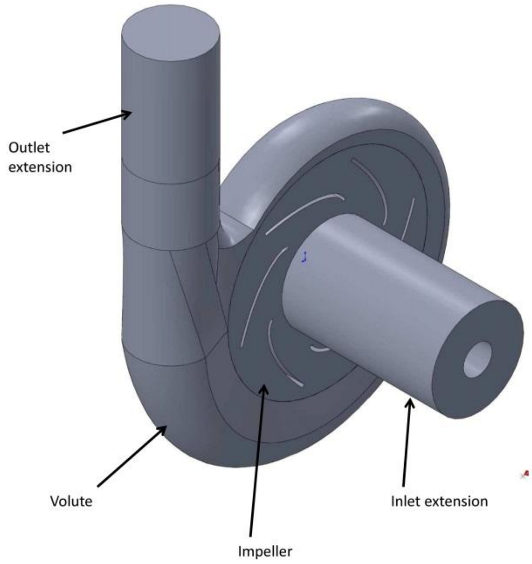


Figure 4: Computational fluid domain

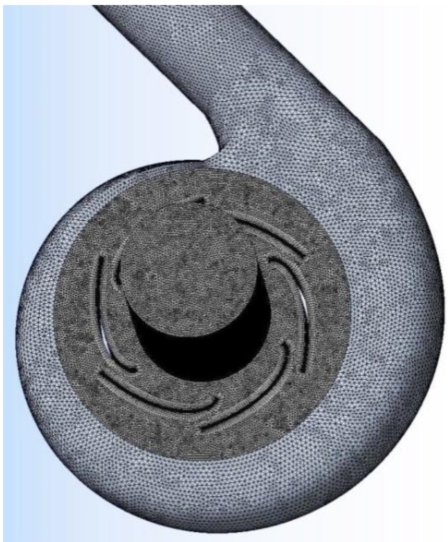


Figure 5: Grid of the fluid domain

II.4 The optimization Technique

We devised a direct method in sections II.1,2, and 3that allowed the straight forward and self-regulated modeling, simulation and calculation of relevant physical properties pertaining to the flow within the pump. In what is next, we use this information alongwith a direct optimization technique known as the Globalized and Bounded Nelder-Mead (GBNM) [12],to optimize the geometry of the blades for maximum hydraulic efficiency.

The GBNM uses the Nelder-Mead (N-M) or simplex scheme [13] along with multiplerestarts and a projection procedure on a box constrained variables to identify the differentoptima of the function. The box projection procedure in (9) assures that the variables arealways selected over the domain of the analysis. Multiple restarts are needed, since the N-Mcan only lead to a local optimum that is dependent on the initial simplex. To avoid findingthe same local optima, the new initial vertex should be different and preferably far fromprevious initial vertices and already known local solutions. To this end, we use a variablevariance probability density (VVP)[14] to identify a starting vertex that is reasonably farfrom the known local optima and initial starting points then construct a simplex from itand restart the N-M for the next optimum.

$$x_i = \begin{cases} x_i^{\text{lower bound}} & \text{if } x < x_i^{\text{lower bound}} \\ x_i^{\text{upper bound}} & \text{if } x > x_i^{\text{upper bound}} \end{cases} \tag{9}$$

where xi is a point sampled during the optimization.

The diagram in Fig.6 represents the scheme used in the implementation of the Globalizedand Bounded Nelder-Mead algorithm (GBNM) and the repetitive restarts needed to reachglobal optimum. This is the same restart scheme used by Luersen[12]. We start with a fixed number of random vertices; these are the initial points. We thenidentify the vertex with the largest probability density; this is the vertex with the largestdistance to the closest neighbor. At this point we use a probabilistic restart by constructingan initial simplex from this point of size equal to 20% the domain size. We then proceedwith the bounded Nelder-Mead optimizer and identify the first local optimum. We stop theN-M algorithm when the simplex is small, or flat.

A simplex is small when

$$\max\left(\left|\frac{x_i^{k+1} - x_i^k}{x_i^u - x_i^l}\right|\right) < \epsilon_1 \tag{10}$$

where k is the number of iterations, subscripts u and l represent the upper and lower bound on variable xi, and$\epsilon_1$is a predetermined small number.

Similarly, a simplex is flat when

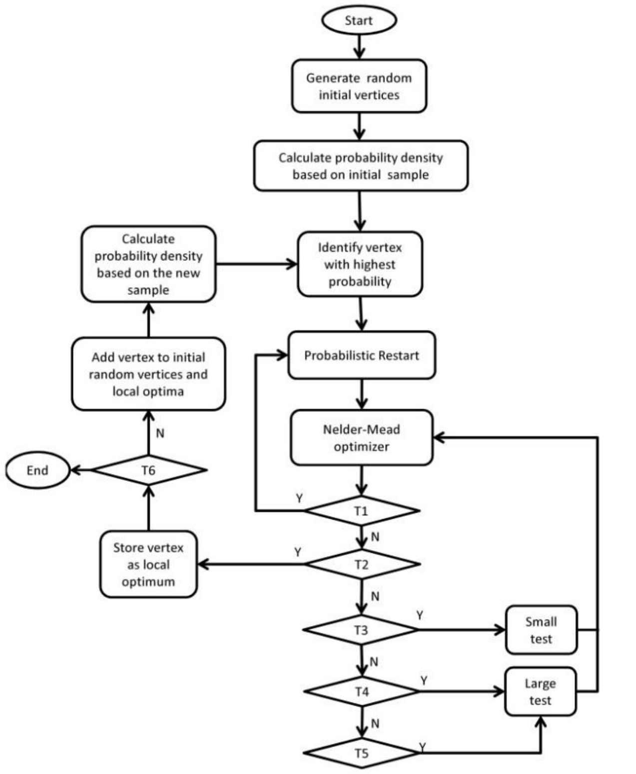
$$|f_H - f_L| < \epsilon_2 \tag{11}$$

wheref_H and f_L are the highest and lowest function values at the current simplex, and ϵ_2 is a given small number.

The local optimum is then stored and used with the initial random points and any priorstored optima to update the probability density from which we identify the next best vertex and use the same probabilistic restart with a polyhedron of size equal to 20% the domain size. There may be cases however, when the new optimum is identical to one of the stored optima; that the maximum number of iterations in the N-M is reached; or that one or moreof the simplex parameters are on the edge of thebox constraint. In cases like these we proceed as indicated in the diagram. A small and large test are used to restart the Nelder-Mead from the best point of the current simplex with a polyhedron of size 5% and 10% thedomain size respectively.

II.5 Program Structure

To achieve optimal values of the hydraulic efficiency, we will be facing four parts of work; Mathematical and geometric modelling, finiteelement analysis (FEA) and mathematical programming. Different program files were developed for each part, and communication between these parts is manipulated by an interface. One of the most interesting features of the ANSYS Workbench software is the possibility to use it as a mere subroutine of any other external program. Parameters can be either directly passed or exchanged through external files. This flexibility allows us to build an interface between ANSYS and our external optimization algorithm, written in Visual Basics for application (VBA), where ANSYS is a finite element package used to calculate the objective function and constraints. The diagram in Fig.7 describes thescheme used to reach global optimization. Commands for implementing the mathematicalmodel and for generating the coordinates of the transformed blade geometry wereincorporated in a Mathematica parametric file. Design parameters are exchanged withthe VBA interface through an external file. The Mathematica file is called from the VBA interface and the inputparameters are updated several times during the optimization. Commands for importing thesurface coordinates, for graphical design of the blades and the rest of the fluid domain,were automated using the SolidWorks Application Programming Interface (API). Thesecommands were implemented directly in the VBA interface. Commands for uploading theparasolid model, for meshing, for adding inflation, and match control among the bladesurfaces as well as mesh refinement and inflation were incorporated in a command fileusing the Java Python language for the ANSYS Mesher. Commands for adding boundaryconditions, for running the CFX-Solver in parallel and for exporting the relevant physical properties in a file.out were incorporated in a Workbench script. The Globalized and BoundedNelder-Mead communicates parametric updates through the VBA interface.



\

Fig 6. GBNM restarts schematics.T1: (this point is already known as a local optimum). T2: (vertex ix a local optimum). T3: (large test or probabilistic and not return to the same point and point on the bound). T4: (small test and not return to the same point and not on the bunds). T5: (N-M stopped by maximum number of iterations). T6: (maximum number of analyses is reached.

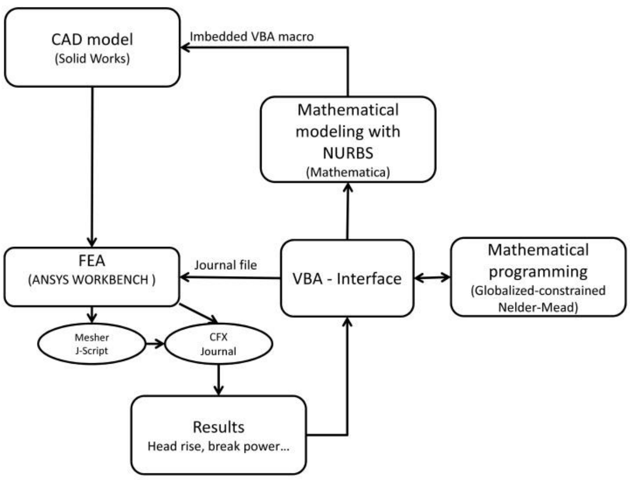


Figure 7: Methodology for geometry optimization

III. APPLICATION

In this section, we shall apply the technique introduced in section II to a simple model of a centrifugal pump.

III.1Optimization Setup

The pump considered in this application is similar in aspect to the model shown in Fig.4.A detailed description including designconditions is shown in Table.1. Here we specify an inlet total pressure and an outlet mass flow rate and allow the blade curvature to adjust toward maximum hydraulic efficiency. We began with 10 random initial vertices overthe box constrained variables described in (12); each vertex encompassesa value for $a_o$through a_3 since all other constants are assumed to hold the value of zero. Appropriate transformations were implemented in the CAD model to change the blade angle to $\beta' = \pi/2 - \beta$. The numberof analyses in the GBNM was set to 30 and the maximum number of iterations in theN-M was set to 30. The stopping criteria for the Nelder-Mead were $\epsilon_1 = \epsilon_2 = 10^{-3}$ for small and flat simplex respectively. The optimum points were rounded off to 10^{-2}.

$$a_i = \begin{cases} -1.0 \le a_o \le 0 \\ -1.0 \le a_1 \le 0 \\ -1.0 \le a_2 \le 1.0 \\ -1.0 \le a_3 \le 1.0 \\ 0 \text{ for } i \ge 4 \end{cases} \tag{12}$$

TABLEI: PUMP CHARACTERISTICS AND DESIGN CONDITIONS

Blade width	0.005 (m)
Number of blades	6
Inlet diameter	0.15 (m)
Outlet diameter (volute diffuser)	0.35 (m)
Fluid	Water
Angular speed of impeller	1500 (rpm)
Inlet turbulence intensity	≤ 10%
Inlet Total pressure	1 atm
Mass flow rate outlet	75 kg/s

A grid independence test was performed on several configurations of the impeller bladesand the hydraulic efficiency and convergence time were selected as the criteria. Forbrevity, we discuss results pertaining to a blade configuration identified by $a_o = -0.57 \; and \; a_1 = a_2 = a_3 = 0$. The analysis was conducted according to the characteristics and design conditions listed in Table.1. Six nodes were used in parallel computation to conduct the FEA simulation. Referring to the results in Table.2, as the mesh became finer; the hydraulicefficiency reached an asymptotic value. Balance between calculation, time and the accuracy order of the simulation has been made and the setting for the "Fine1" grid is considered to be sufficiently reliable.

TABLE II: GRID INDEPENDENCE TEST

Total number of cells	Hydraulic efficiency	Convergence time
Coarse (1260781 million)	67.93 %	28 minutes
Medium (1829397 million)	68.4438 %	37 minutes
$Fine_1$ (2207039 million)	68.3520 %	46 minutes
$Fine_2$ (2734539 million)	68.3871 %	53 minutes

III.2 Results

Table.3 shows four local optima, although there are around twelve other local solutions found during the optimization process that a designer can select from. In thisrespect, the optimization is comparable to an evolutionaryprocedure that provides a family of optimal solutions insteadof just one specific solution. This feature is importantespecially for multi-objective optimization.

TABLE III: OPTIMUM DESIGN OBTAINED BY GBNM

	a_o	a_1	a_2	a_3	Hydraulic Efficiency
Case 1	-0.57	-0.8	0.17	0.35	71.35 %
Case 2	-0.67	-0.93	0.17	0.38	73.83 %
Case 3	-0.77	-1.03	0.18	0.4	75.44%
Case 4	-0.87	-1.08	0.20	0.40	73.68 %

The results of table.3suggest that under the design conditions specified in Table.1, improved hydraulic efficiency isachievable at different values of (a_o, a_1, a_2, a_3) with slightly better results at relatively larger magnitudes. In addition, from the blade angle depicted in Fig.8, it is clear that improved efficiency is attainable with smaller inlet blade angles followed by an increasing blade curvature with the maximum being achieved at specific values of the constants (a_o, a_1, a_2, a_3).

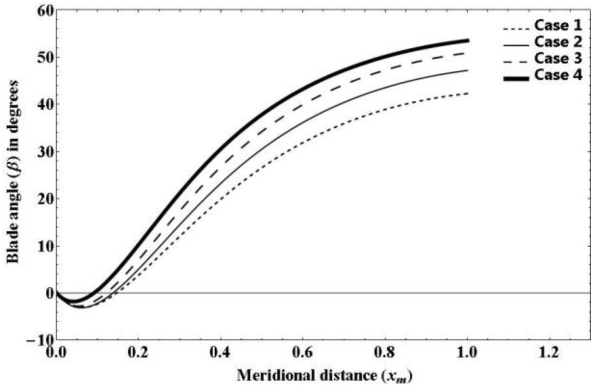


Fig 8.Blade angle versus meridional distance

IV. Conclusion

In this paper, we introduced a robust technique thatcombines mathematical and geometrical modeling, programmingand finite element analysis to optimize the geometry of the impeller blades of a test model of a centrifugal pump under preselected boundary conditions. As an application we specified the total inlet pressure and outlet mass flow rate then allowed the blade curvature to adjust toward maximizing the hydraulic efficiency. The evolutionary andconstrained aspect of the technique produced a family ofoptimal solutions that a hydraulic designer can choose from.

The technique could also be extended to include additional terms in the expansion of the blade angle. It can also be improved if parameterization of the meridional projection was also incorporated. Furthermore, the user is free to adjust the boundary conditions to fit the design requirement, yet the current methodology is guaranteed to render optimal designs.

V. Acknowledgments

The first author would like to express his gratitude to the office of research and grants at the University of Central Oklahoma (UCO) for the financial support during this research.

References

[1] E.T. Pak,J.C. Lee,"Performance and pressure distribution changes in a centrifugalpump under two-phase flow". Proceedings of the Institution of Mechanical Engineers. PartA, Journal of Power and Energy, Vol.212(3), pp.165-171 (1998).

[2] R.B.Medvitz, R.F. Kunz, D.A. Boger, J.W. Lindau, A.M. Yocum, " Performance analysis of cavitating flow in centrifugal pumps using multiphase CFD", Journal of FluidsEngineering Vol.124(2), p.377(7) (2002).

[3] S. Yedidiah, "An alternate method for calculating the head developed by a centrifugal impeller". ASME/JSME Fluids Engineering Conference 107:131138.

[4] J.M.Lighthill, " A new method of two-dimensional aerodynamicdesign". ARCR&M 2112.13 (1945).

[5] P.M.Came, "The Development, Application and Experimental Evaluation of a DesignProcedure for Centrifugal Compressors", Proceedings of Institution of Mechanical Engineers,Vol. 192.

[6] A.Whitfield, M.H.Patel, F.J.Wallace, "Design and Testing of Two Radial Flow Backward Swept Turbocharger Compressors",Institution of Mechanical Engineers ConferencePublication, 1978-2.

[7] R.I.Lewis, " Turbomachinery Performance Analysis". John Wiley, New York.12.(1996)

[8] B.Lakshminarayana, "An assessment of computational fluid dynamic techniques in theanalysis and design of turbomachinery", ASME Journal of Fluids Engineering, vol. 113,pp. 315-352, 1991.

[9] W. Rodi, S. Majumdar, and B. Schonung, "Finite volume methods for two-dimensionalincompressible flows with complex boundaries", Computer Methods in Applied Mechanicsand Engineering, vol. 75, pp. 369-392, (1989).

[10] S. Thakur, J. Wright, W. Shyy, J. Liu, H. Ouyang, and T. Vu, "Development ofpressure-based composite multigrid methods for complex fluid flows", Program in AerospaceScience, vol. 32, pp. 313-375, 1996.

[11] F.Prasil," Techniche Hydrodynamik", Appendix 4, J. Springer,Berlin, 1926.

[12] M.A.Luersen, and R.L.Riche, "Globalized Nelder-Mead Methodfor Engineering Optimization", Computers and Structures, 82:22512260 (2004)

[13] J.A.Nelder, and R.Mead, "A Simplex Method for FunctionMinimization”, The Computer Journal :308313 7 (1965)

[14] H.Ghiasi, D. Pasini, L.Lessard, "Constrained globalized Nelder-Mead method for simultaneous structural and manufacturingoptimization of a composite bracket". J.Compos Mater 42(7):717736,2008