Hydrodynamic Optimization of Marine Propeller Using Gradient and Non-Gradient-based Algorithms

Ramin Taheri, Karim Mazaheri

Laboratory of Hydrodynamic Computation and Optimization Design, Center of excellence in Aerospace Engineering Systems, Sharif University of Technology, Azadi Ave., Tehran, Iran, POBox:11365-8629
E-mail: ramintaheri@ae.sharif.ir; mazaheri@sharif.edu

Abstract: Here a propeller design method based on a vortex lattice algorithm is developed, and two gradient-based and non-gradient-based optimization algorithms are implemented to optimize the shape and efficiency of two propellers. For the analysis of the hydrodynamic performance parameters, a vortex lattice method was used by implementing a computer code. In the first problem, one of the Sequential Unconstraint Minimization Techniques (SUMT) is employed to minimize the torque coefficient as an objective function, while keeping the thrust coefficient constant as a constraint. Also, chord distribution is considered as a design variable, namely 11 design variables. In the second problem, a modified Genetic algorithm is used. The objective function is to maximize efficiency by considering the design variables as non-dimensional blade's chord and thickness distribution along the blade, namely 22 design variables. The hydrodynamic performance analyzer code is modified by a higher order Quasi-Newton scheme. Also, a hybrid function is used to improve the accuracy of the convergence. The solution of the optimization problems showed that a nearly 13% improvement in efficiency and a nearly 15% decrease in torque coefficient for the first propeller, as well as nearly 10% improvement for efficiency of the later propeller, is possible.

Keywords: Marine propeller; gradient-based optimization algorithm; Genetic algorithm; Vortex lattice

1 Introduction

The complexity of the flow field in which the propeller must operate efficiently will lead a designer to lay out a propeller to overcome most of the dilemma. Another difficulty which arises during propeller action is the variation of inflow, which has a great influence on propellers. Hence, the range of design is restricted for designers.

The development of the Momentum theory for marine propellers was the starting point of the aerodynamic analysis of rotary wings. Betz [1] firstly introduced the lifting line theory and Goldstein [2] and Lerbs [3] consequently improved the method. Theodorson extended the vortex theory for highly loaded propeller. Rand and Rosen [4], Chang and Sullivan [5] and Chiu and Peters [6] used the lifting line theory for their works. Later on, Eckhart and Morgan [7] proposed the Lifting-Surface correction factors that were then developed further by Pien [8] and Kerwin [9]. Chord distribution, wing tip shape and twist angle were shown by McVeigh and McHugh [10] and Walsh et al. [11] to be the main factors which control the performance of straightened blade propellers. Lee [12] applied the vortex lattice methods for the prediction of the hydrodynamic performance of marine propellers. Khot and Zweber [13] optimized the structure of a composite wing by using gradient based algorithm. The twist angle distribution and a span wise chord distribution were optimized by Cho and Lee [14] utilizing gradient based optimization with the penalty function method. Also, investigating the possibility of maximizing the efficiency by utilizing Genetic algorithm was done by Lee and Lin [15]. Later on, Plucinski et al. [16] optimized a self-twisting propeller, using a genetic algorithm by considering the orientation angles of the fibers in each layer as the design variables for efficiency improvement. For design optimization, a propeller performance analysis program was developed and integrated into a genetic algorithm by Christoph Burger [17]. The duty of the tool is to produce optimal propeller geometry for a given aim, which includes performance and/or acoustic signature. Using a genetic optimization algorithm, Aykut et al. [30] achieved a more convincible result compared to previous studies. Taheri et al. [18, 19] studied the process of both gradient and non-gradient-based optimization algorithms. Also, the accomplishment of an inverse design as well as the optimization of the propeller were studied and done by Taheri et al. [20, 21].

2 Openprop

OpenProp is a design and analysis tool for propellers and turbines. The code is written in MATLAB M-code and the numerical model is based on vortex lattice lifting line methods. The capability of the code has been tested by validating an experiment results and the numerical method which is used in OpenProp. OpenProp began in 2001 and was further developed by Kerwin [22] in 2007. The code was improved by Stubblefield in 2008 and Epps in 2009, respectively. Development of the OpenProp code suite began at MIT in 2006 under the direction of Kimball as a Matlab version of a Fortran code published by Kerwin called PVL (Kerwin 2007). Subsequent researchers have extended the functionality of the code with the most recent version implemented by Epps (Epps, Stanway and Kimball 2009). An explanation of the theory can be found in (Epps 2010) which also presents validation of the code as presented in Figure 1

below. Further reference to the code and website can be found in (Kimball and Epps, 2010). The good agreement between experimental data and numerical calculations done by OpenProp and a commercial package (Propeller Blade Design, PBD) is shown here.

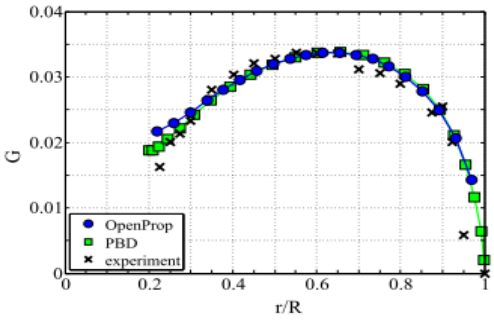

Figure 1

Validation of Circulation results calculated by OpenProp and comparison with experimental data and PBD (Propeller Blade Design), reprinted with permission from Epps [23]

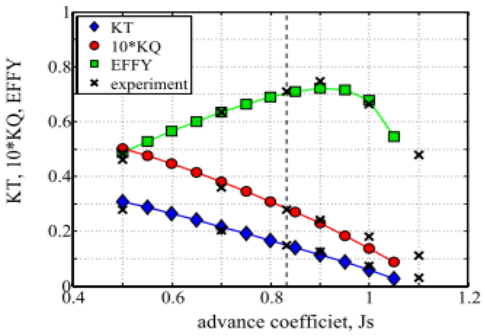

Figure 2

Validation of efficiency calculated by OpenProp and comparison with experimental data, reprinted with permission from Epps [23]

First, the circulation distribution is compared, as is shown in Figure 1. Second, thrust and torque coefficients, as well as final efficiency distribution over the range of propeller performance, has been done and the results are shown in Fig. 2. The above illustrations convinced us to rely on this code and use that as a package to calculate our hydrodynamic performance needs.

3 Propeller Lifting Line Formulation

In the following calculation, based on moderately loaded lifting line theory, the lifting line is the representative of a propeller blade, with trailing vorticity aligned to the local flow velocity (i.e., the vector sum of free-stream plus induced velocity). Using a vortex lattice with helical trailing vortex filaments shed at discrete stations along the blade, induced velocities can be computed. The blade is sectioned discretely, having 2D section properties at each radius. Loads are computed by integrating the 2D sections load over the span of the blade. The velocities and forces (per unit span) on a 2D blade section can be seen in both the axial and tangential directions in "Figure 3". Apparent tangential inflow at radius r is $-\omega r e_t$, while the propeller shaft rotates with angular velocity of ωe_a. Total resultant inflow velocity, V^*, and its orientation pitch angle can be computed by equation (1) and equation (2), respectively.

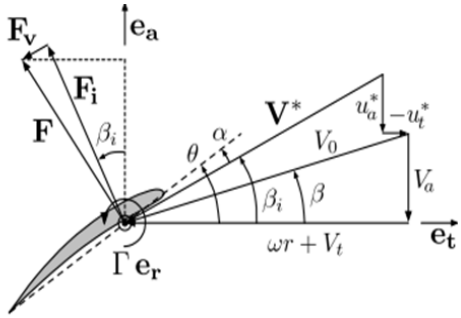

Figure 3

Propeller velocity/force diagram, as viewed from the tip towards the root of the blades. All velocities are relative to a stationary blade section at radius r, reprinted with permission from Epps [23].

This is an equation example:

$$V^* = \sqrt{(V_a + u_a^*)^2 + (\omega r + V_a + u_t^*)^2}. \tag{1}$$

$$\beta_i = \arctan\left(\frac{V_a + u_a^*}{\omega r + V_a + u_t^*}\right), \tag{2}$$

where $V_a = -V_a e_a$ and $V_t = -V_t e_t$ are the axial and tangential inflow velocities, $u_a^* = -u_a^* e_a$ and $u_t^* = -u_t^* e_t$ are induced axial and tangential velocities, α is the angle of attack, $\theta = \alpha + \beta_i$ is blade pitch angle, Γe_r is circulation, $F_i = \rho V^* (\Gamma e_r)$ is (inviscid) Kutta-Joukowski lift force, and F_v is viscous drag force aligned with V^*. Assuming the Z blades are identical, the total thrust and torque on the propeller are

$$T = z\int_{r_h}^{R}(F_i\cos\beta_i - F_v\sin\beta_i)dr(\hat{e}_a), \tag{3}$$

$$Q = z\int_{r_h}^{R}(F_i\sin\beta_i + F_v\cos\beta_i)rdr(-\hat{e}_a), \tag{4}$$

where $F_i = \rho V^* \Gamma$ and $F_v = \frac{1}{2}\rho V^{*2}(C_D)c$ are the magnitude of inviscid and viscous force per unit radius, ρ is the fluid density, C_D is the section drag coefficient, c is the section chord, and r_h and R are the radius of the hub and blade tip, respectively.

The propeller power consumption is the product of torque and angular velocity

$$P = Q\omega, \tag{5}$$

The propeller puts power into the fluid when, $P \succ 0$ (i.e. the torque resists the motion). As the useful power produced by the propeller is TV_s, where V_s is the ship speed (i.e. free stream velocity), the efficiency of propeller is defined by [21]

$$\eta = \frac{TV_s}{Q\omega}. \tag{6}$$

After the above calculations, thrust and torque coefficients as well as advanced ratio are calculated as follow

$$K_T = \frac{T}{\rho n^2 D^4}. \tag{7}$$

$$K_Q = \frac{Q}{\rho n^2 D^5}. \tag{8}$$

$$J = \frac{V_s}{nD}. \tag{9}$$

4 Inverse Design

First, to show the capability of the hydrodynamic analyzer code, the inverse design was done by a nearly ill-posed initial guess. This part was only done to prove the validity of the results that had been calculated by OpenProp code. For this purpose, the circulation distribution along the blade was chosen to reach our desired circulation distribution. Following is the function with which we explored the validation of the code

$$I = \int \left| G - G_{desired} \right|. \tag{10}$$

The result of this calculation is shown in "Figure 4".

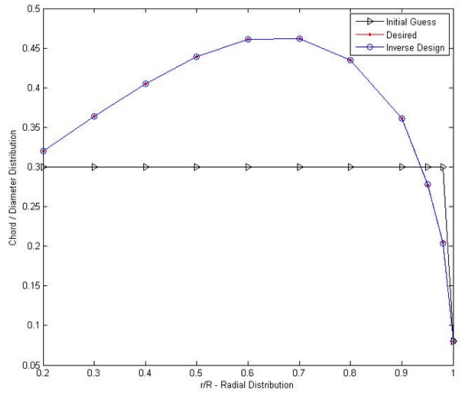

Figure 4
Results of propeller inverse design

4 Optimization Algorithms

Generally speaking, optimization algorithms are categorized as two major sets. The first one is gradient-based algorithms and the other one is non-gradient-based algorithms. The act of choosing each set depends on the pros and cons related to each category, as well as to the specific conditions for a problem.

4.1 Gradient-based Algorithm

For Gradient-based algorithms, the simplest way of calculating derivatives for functions is the Finite Difference method. If the round-off error is important, the Complex method will be beneficial. There are many methods to calculate the derivatives, such as sequential quadratic programming (SQP), which is the most popular method for constraint optimization problems. For unconstraint optimization problems, Quasi-Newton methods play an important role. Also, by adding penalty terms to constraint formulation one can obtain an unconstraint approach. One of the sequential unconstraint optimization techniques, called the extended linear interior penalty function method, (EIPM) was used here. One of the most considerable differences between gradient-based and non-gradient-based algorithms is the existence of linear trend associated with number of design

variables, while in the latter methods the cost of calculations is increased drastically by an increase in number of design variables.

The Extended linear Interior Penalty function Method (EIPM), one of the sequential Unconstrained Minimization Techniques (SUMT), is employed as one of the optimization techniques. The aforementioned technique transforms a constrained optimization problem into a series of unconstrained optimization problems and constructs a pseudo-objective function using penalty functions.

A constrained optimization problem is stated as [24]:

$$\begin{aligned} &\text{minimize } f(x) \\ &\text{subject to } g_j(x) \le 0,\ \text{j=1,m} \\ &\mathrm{h}_k(x) = 0,\ k = 1, l \end{aligned} \tag{11}$$

where $f(x)$ is objective function. $g_j(x)$ and $h_k(x)$ are inequality and equality constraints, respectively. The transformed unconstrained optimization problem is also stated as:

$$pseudo-objective\ \phi(x, r_p, \dot{r_p}) = f(x) + r_p P(x) \tag{12}$$

where r_p is a multiplier and will increase in each iteration until it reaches to a pre-defined value, and $P(x)$ is a penalty function consists of equality and inequality constraints. The final form of the transformed constrained optimization problem is

$$S^k = -H^k \nabla F(x^k) \tag{13}$$

$$H^{k+1} = H^k + D^k \tag{14}$$

where D^k is defined as follows [25]:

$$\left[D^k\right] = \left[M_i\right] + \left[N_i\right] \tag{15}$$

$$\left[M_i\right] = \lambda_i^* \frac{S_i S_i^T}{S_i^T g_i} \tag{16}$$

$$\left[N_i\right] = -\frac{\left(\left[H_i\right] g_i\right)\left(\left[H_i\right] g_i\right)^T}{g_i^T \left[H_i\right] g_i} \tag{17}$$

where $g_i = \nabla f\left(X_{i+1}\right) - \nabla f\left(X_i\right) = \nabla f_{i+1} - \nabla f_i$.

The iterative procedure is,

I. Start with initial design variables as a vector x^0 and initiate Hessian matrix H^0. (commonly identity matrix, $H^0 = I$)

II. Finding search direction, $S^k = -H^k \nabla F(x^k)$

III. Finding step length, λ by a univariate optimization process which determines the amount of change in the search direction.

IV. Updating design variables and then calculating gradient and Hessian of the function,

$$x^{k+1} = x^k + \lambda S^k$$

V. Checking convergence criteria and going to step II.

Here, non-dimensional chord distribution is considered as design variables, in fact 11 variables.

The test case which has been used was a DTMB 4119 propeller of which the geometry characteristics are listed in "Table 1".

Table 1
Geometry definition of DTMB 4119 propeller [26]

r / R	c / D	p / D	qr	IT / D	tm / C	fm / C
0.2	0.32	1.105	0	0	0.2055	0.01429
0.3	0.3635	1.102	0	0	0.1553	0.02318
0.4	0.4048	1.098	0	0	0.1180	0.02303
0.5	0.4392	1.093	0	0	0.09016	0.02182
0.6	0.4610	1.088	0	0	0.0696	0.02072
0.7	0.4622	1.084	0	0	0.05814	0.02003
0.8	0.4347	1.081	0	0	0.04206	0.01967
0.9	0.3613	1.079	0	0	0.03321	0.01817
0.95	0.2775	1.077	0	0	0.03228	0.01631
1.0	0.0	1.075	0	0	0.0316	0.01175

Other characteristics that should be considered are:

1. The propeller inflow is uniform.
2. The propeller has 3 blades, i.e. N = 3.
3. The hub-to-diameter ratio is 0.2.
4. The propeller has no skew and no rake.
5. The blade sections are designed with NACA 66 modified profiles and a camber line of a = 0.8.
6. The propeller advanced ratio is $j = 0.833$.
7. The direction of rotation is right-handed.

4.1.1 Objective Function

For the abovementioned gradient-based optimization algorithm, the objective function would be the torque, and the equality constraint function would be given in terms of thrust as below:

$$\begin{aligned} &Minimize\ f(x) = \mathrm{k}_Q(x) \\ &Subject\ to\ h(x) = \frac{k_t - k_{t0}}{k_{t0}} \end{aligned} \tag{18}$$

Where X represents design variables, f(x) is objective function, and h(x) is equality constraint.

4.2 Genetic Algorithm

Mitchel [27] states that "Genetic Algorithms were invented by John Holland in the 1960s and were developed by Holland and his colleagues at the University of Michigan in the 1960s and 1970s". Genetic algorithms are a subset of stochastic optimization methods, methods in which statistical data play an important role [28]. Literally speaking, genetic algorithms are specifically modeled after the process of natural selection. In the natural selection process it is obvious that the fittest individuals, which have a higher probability for regeneration, also have a higher probability to be chosen and being sent for the next generation.

In an optimization problem, design variables are put into chromosomes and then they undergo every chromosomal operation, such as cross-over, the process by which chromosomes exchange genes in real-world genetics, mutation and so on.

The creation of the initial population of random individuals is what the genetic algorithm begins with, each of which represents a probable solution in term of parameters. The way of well-satisfying the objective function is called fitness, which is assigned to each individual consisting of a particular set of design variables. In GA terminology, the objective function is the function that determines the performance of a particular chromosome (i.e., member of the population).

The fitter the individual, the higher chance of it having children and reproducing for the next population. This should be certified by genetic algorithm for modeling the process of natural selection efficiently. Pairing off individuals for mating and having children will occur for subsequent populations.

The simplicity of making the chance of having children proportional to the relative fitness of an individuals, and mating individuals which are competing in a domain of diverse fitness, in which higher fitness means a higher probability of winning, both can be equal.

As a matter of fact, the existence of the probability which says that a child may have a mutation in its genetic code, as well as the randomness of chromosomes which are received by father or mother, prevents the algorithm from being trapped in a globally non-optimal local minimum or maximum. Before reaching the determined number of generations or some other stopping criteria, the process of mating, mutation and cross-over will be repeated. After that, by comparing fittest individuals iteratively, the global optimum will be found. At the same time, GA has some disadvantages. As has been mentioned, in using GA there is a greater likelihood that a global optimum solution will be found. However, finding this global optimum is not guaranteed. Even if GA is in the neighborhood of the global optimum, there is the possibility that through crossover and mutation the global optimum may not be achieved. Also, GA does not address the robustness of the individual design solutions it creates. GA simply attempts to meet the desired goals and will adjust the design parameters accordingly. Thus, it is up to the user to ensure the proper operation of GA and to verify that the results are genuine. Finally, the satisfactory operation of GA relies on the accuracy of the system models that make up the objective function.

Here, in order to create the genetic algorithm being modified, a hybrid function is used. The hybrid function enables us to specify another minimization function that runs after the genetic algorithm terminates. Also, the hybrid function is a function by which we can cause the genetic algorithm to be converged faster. For instance, after reaching to the region of a highly likely fitter population for the next generation, the hybrid function, by using a simple gradient-based algorithm (Steepest Descent), helps us to put the design variables in a more efficient direction towards an optimum point, instead of just producing a large number of populations. Hence, using this scheme decreases the time needed for convergence. To implement this method, a gradient-based algorithm is used to lower the computational costs. Since in some manners of using genetic algorithm it is obvious to find better population for the next generation near the solution point, using a gradient-based algorithm can be considered very helpful and reliable.

In the GA algorithm, the population type specifies the type of input to the fitness function. Types and their restrictions here is a double vector, and the population size is 20.

A scaling function is the function that converts raw fitness scores returned by the fitness function to values in a range that is suitable for the selection function. Here we have used ranked scaling, which scales the raw scores based on the rank of each individual, rather than its score. The rank of an individual is its position in the sorted scores. The rank of the fittest individual is 1, the next fittest is 2, and so on. Rank fitness scaling removes the effect of the spread of the raw scores.

The selection function chooses parents for the next generation based on their scaled values from the fitness scaling function. Here we have used Stochastic Uniform, which lays out a line in which each parent corresponds to a section of

the line of length proportional to its expectation. The algorithm moves along the line in steps of equal size, one step for each parent. At each step, the algorithm allocates a parent from the section it lands on. The first step is a uniform random number less than the step size.

Crossover combines two individuals, or parents, to form a new individual, or child, for the next generation. Here we have used Scattered, which creates a random binary vector. It then selects the genes where the vector is a 1 from the first parent, and the genes where the vector is a 0 from the second parent, and combines the genes to form the child.

4.2.1 Objective Function

$$\begin{aligned} &\text{minimize } f(x) \\ &\text{subject to } g_j(x) \le 0,\ \text{j=1,m} \\ &\mathrm{h}_k(x) = 0,\ k = 1, l \end{aligned} \tag{19}$$

where h_k and g_j are equality and inequality constraints, respectively. Herein, thickness distribution and non-dimensional chord distribution are considered as design variables, in fact 22 design variables, under the following conditions as an inequality constraint:

- For structural requirements, minimum foil section thickness should be considered.
- To avoid cavitation, the minimum pressure coefficient should be negative and the cavitation is also depth dependent.

The test case used was a DTRC 4119 propeller, the geometry characteristics of which are listed in Table 2.

Table 2
Geometry definition of DTRC 4119 propeller [26]

r / R	c / D	p / D	qr	IT / D	tm / C	fm / C
0.2	0.32	1.105	0	0	0.2055	0.01429
0.3	0.3635	1.102	0	0	0.1553	0.02318
0.4	0.4048	1.098	0	0	0.1180	0.02303
0.5	0.4392	1.093	0	0	0.09016	0.02182
0.6	0.4610	1.088	0	0	0.0696	0.02072
0.7	0.4622	1.084	0	0	0.05814	0.02003
0.8	0.4347	1.081	0	0	0.04206	0.01967
0.9	0.3613	1.079	0	0	0.03321	0.01817
0.95	0.2775	1.077	0	0	0.03228	0.01631
1.0	0.0	1.075	0	0	0.0316	0.01175

where r/R is non-dimensional radius distribution, c/D is non-dimensional chord distribution, p/D is non-dimensional pitch distribution, qr is the rake of propeller, IT/D is the non-dimensional skew of propeller, tm/C is the non-dimensional maximum thickness distribution, and fm/C is the non-dimensional camber distribution.

Other characteristics that should be considered are:

1. The propeller inflow is uniform.
2. The propeller has 3 blades, i.e. N = 3.
3. The hub-to-tip diameter ratio is 0.2.
4. The propeller has no skew and no rake.
5. The blade sections are designed with NACA 66 modified profiles and a camber line of a = 0.8.
6. The propeller advanced ratio is $J = 0.833$.
7. The direction of rotation is right-handed.

The whole procedure is depicted in Figure 5. First, the code is started from an initial guess which is our so-called propeller. Then, Openprop is used to calculate the hydrodynamic parameters that are compulsory for the objective function. The next step is to find better geometry by implementing the modified genetic algorithm. Finally, checking the convergence criteria is done iteratively to quit the program.

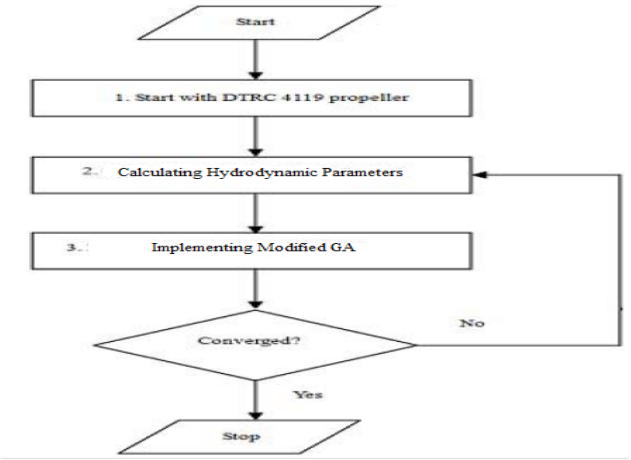

Figure 5
Flowchart for optimization process

5 Results and Discussion

The results for the circulation distribution as well as the torque coefficient for the first optimization problem are as follows, where higher circulation keeps the efficiency higher (Figure 6 and Table 3):

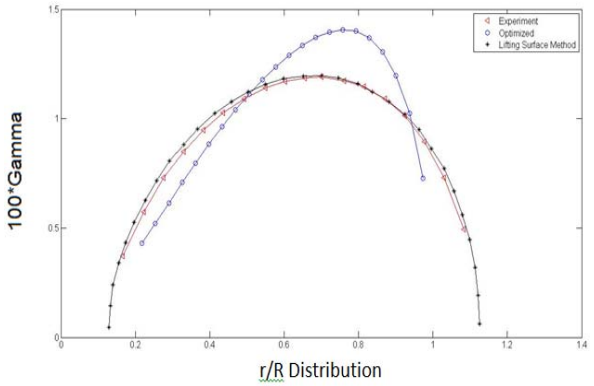

Figure 6
Radial Circulation distribution. Initial (Lifting Surface Method), Experiment, and Optimized

Table 3
Optimized propeller in comparison to the original one

Type of propeller	K_t	K_Q	η
DTMB 4119	0.1468	0.0264	0.7375
Present (Optimized)	0.147	0.0227	0.8589

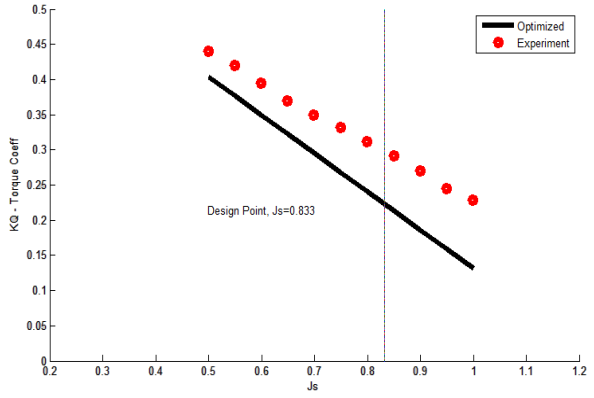

Figure 7
Reduction in torque coefficient distribution with respect to advanced ratio. Initial (Lifting Surface Method), and Optimized.

For the second problem, the combination of both hydrodynamic analyzer code and modified Genetic algorithm has been done elaborately and the results of any improvement are shown in the next consequent figures. The shape of the cross sectional airfoils is depicted in Figure 7. Furthermore, the elongated chord distribution for optimized propeller in comparison with initial one can be seen in Figure 9. A little ripple at the top most of blade's tip is also demonstrated, which is similar to the behavior which was shown by Karim [29]. The time needed for calculation was approximately 10 minutes for 113 generations on a personal computer with a CPU speed of 3.1 GHz. The history of the convergence is shown in Figure 10. Scrutinizing Figure 11, it is obvious that efficiency at the predetermined advanced ratio is higher than those for which calculation was done for the initial and redesigned ones [29]. In order to reach higher efficiency, as shown in both Figure 12 and Figure 13, the thinner camber distribution, whether non-dimensional distribution or just the very distribution, should accommodate with the length of the blade. Although the new design for the propeller has a lower thrust coefficient, it is clear that this penalty can be compensated for by higher efficiency, by which the performance of propeller are put considered. Table 3 shows the thrust and torque coefficients as well as efficiency over time. It is very clear that this efficiency is high enough to convince a propeller designer not only of the reliability of the method, but also the usefulness of the tool.

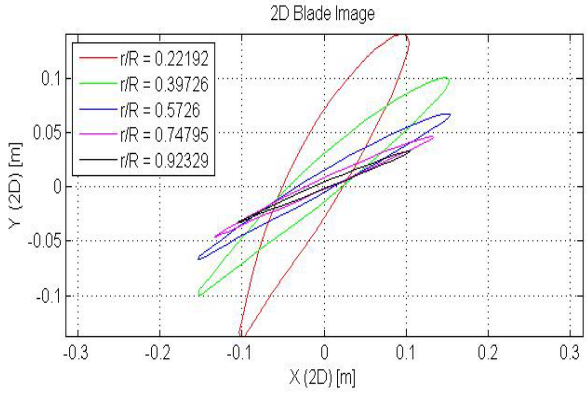

Figure 8
Two dimensional cross section distribution for optimized propeller

Conclusions

In this article, to validate our scheme, first we note that our physical analysis tool (OpenProp) is clearly validated, as shown in Figure 1, Figure 2 and "Figure 4" for our propellers. The result of OpenProp has been compared with experimental data.

Here we applied two optimization algorithms to a given propeller geometry. For gradient based algorithm we have used non-dimensional chord distribution as

design variables and have kept thrust coefficient constant as an equality constraint. Moreover, we have applied an inverse design scheme to an initial guess for our propeller geometry. Figure 4 shows, as expected, very good agreement between our results and experimental data. Table 3 compares hydrodynamic shape parameters achieved by optimization algorithms with the initial values. The efficiency improvement (nearly 13%) shows that the optimized circulation distribution was higher than the experimental values; higher circulation leads to a higher lift force and, consequently, higher efficiency can be achieved. Also, the torque coefficient reduction shown in Figure 7 demonstrates that a nearly 15% improvement can be considered possible.

The second optimization problem presented results from the application of a modified genetic algorithm technique to the design optimization of marine propeller incorporating vortex lattice method (VLM). In this research, the hybrid function was used in order to modify the genetic algorithm modified and to reach the final solution more quickly than usual. The performance of the modified genetic algorithm coupled with hydrodynamic performance analyzer code seemed to be better than one which was used by Karim [29]. Chord distribution and non-dimensional thickness distribution at twenty two sections have been optimized as design variables for efficiency improvement. Achieving a lower torque coefficient as well as higher efficiency were clearly possible. Another improvement has been applied to the hydrodynamic performance analyzer code, that is, OpenProp, to increase the accuracy of the results one order of magnitude higher than the previously best one by setting a Quasi-Newton method instead of Newton method to reach to the abovementioned purpose.

References

[1] Betz, A., “SchraubenPropeller mit geringstem Energieverlust”. K. Ges. Wiss. Gottingen Nachr. Math. Phys. Klasse, 1919

[2] Goldstein, S., “On the Vortex Theory of Screw Propellers”. Proc. R. Soc. London Ser. A 123 :440-465, 1929

[3] Lerbs, H. W., “Moderately Loaded Propellers with a Finite Number of Blades and an Arbitrary Distribution of Circulation”. SNAME Trans, 60, 1952

[4] Rand, A., and Rosen, A., “A Lifting Line Theory for Curved Helicopter Blades in Hovering and Axial Flight”. The 8th European Rotorcraft Forum, Aix-en Provence, France, 1982

[5] Chang, L. K., and Sullivan, J. P., “Optimization of Propeller Blade Twist by an Analytical Method”. AIAA Journal, 22(2), 22, 1982

[6] Chiu, Y. D., and Peters, D. A., “Numerical Solution of Induced Velocities by Semi-Infinite Tip Vortex Lines”. Journal of Aircraft, 25(8), 684, 1987

[7] Eckhart, M. K., and Morgan, W. B., "A Propeller Design Method". SNAME Trans. 1955, 63, 1955

[8] Pein, P. C., "The Calculation of Marine Propellers Based on Lifting Surface Theory". J. of Ship Research, Vol. 5, No. 2

[9] Kerwin, J. E., "The Solution of Propeller Lifting Surface Problems by Vortex Lattice Methods". Report, Dep. of Ocean Eng., MIT, 2001

[10] McVeigh, M. A., and McHugh, F. J., "Influence of Tip Shape Chord, Blade Number and Airfoil on Advanced Rotor Performance". The 38th Annual Forum of the American Helicopter Society, Anaheim, CA, 1982

[11] Walsh, J. L., and Bingham, G. J., and Riley, M. F., "Optimization Method Applied to the Aerodynamic Design of Helicopter Rotor Blades". The 26th AIAA/ASME/ASCE/AHS structures, Structural Dynamic and Material Conference, Orlando, Florida, 1985

[12] Lee, C. S., Prediction of Steady and Unsteady Performance of Marine Propellers with or without Cavitation by Numerical Lifting Surface Theory. Ph. D. Thesis, MIT., USA, 1979

[13] Khot, N. S., and Zweber, J. V., "Design of Flutter Characteristics of Composite Wings Using Frequency Constraint Optimization". Journal of Aerospace Engineering, Vol. 16, pp 19-30, 2003

[14] Cho, J., Lee, S-C., "Propeller Blade Shape Optimization for Efficiency Improvement". Computers and Fluids, Vol. 27, No. 3, pp. 407-419, 1998

[15] Lee, Y-J., and Lin, C-C., "Optimized Design of Composite Propeller". Mechanics of Advanced Materials and Structures. Vol. 11, pp. 17-30, 2004

[16] Plucinski, M. M., and Young, Y. L., and Liu, Z., "Optimization of a Self-Twisting Composite Marine Propeller Using Genetic Algorithms". 16th International Conference on Composite Materials, Kyoto, Japan, 2007

[17] Burger, C., "Propeller Performance Analysis and Multidisciplinary Optimization Using a Genetic Algorithm". ProQuest Publisher, PHD Thesis, Auburn University, 2007

[18] Taheri, R., Mazaheri, k., "*Comparison of Gradiend-based and Genetic Optimization Algorithms Applied to wing-body Configuration***"**. 20th Annual International Conference on Mechanical Engineering-ISME2012, School of Mechanical Eng., Shiraz University, Shiraz, Iran, 16-18 May, 2012

[19] Taheri, R., Mazaheri, k., "*Rapid Aerodynamic Optimization of Wing and Body ConfigurationUsing Gradient Based Approach* **"**. 11th Annual International Conference on Aerospace Engineering, Shahid Sattary University of Science and Technologies, Tehran, Iran, 2-4 March, 2012

[20] Taheri, R., Mazaheri, K., "*Blade Shape Optimization of Marine Propeller via Genetic Algorithm for Efficiency Improvement*". Proceedings of ASME Turbo Expo 2012, Copenhagen, Denmark, June 11-15, 2012

[21] Taheri, R., Mazaheri, k., "*Hydrodynamic optimization of propeller using Gradient based approach***"**. 14th Conference on Fluid Dynamics, Birjand, Iran, May 1-3, 2012

[22] Kerwin, J. E., "Hydorofoils and Propellers". MIT course, 13.04 lecture notes. http://ocw.mit.edu/courses/mechanical-engineering/2-23-hydrofoils-and-propeller-spring-2007

[23] Kimball, R. W., and Epps, B., "OpenProp v2.4 propeller/turbine design code", http://openprop.mit.edu, 2010

[24] S. S. Rao, Engineering Optimization, Theory and Practice. J, Wiely & Sons Inc, Fourth Edition, School Of Mechanical Engineering, University of Purdue, 2009

[25] G. N. Vanderplaats, Numerical Optimization Techniques for Engineering Design, 1984

[26] Brizzolara, S, Villa, D, and Gaggero, S. "A Systematic Comparison between RANS and Panel Method for Propeller Analysis". Proc. of 8th International Conference on Hydrodynamics, Nantes, France, 2008

[27] Mitchel, M., "An introduction to Genetic Algorithms". MIT Press, Cambridge, MA, 1998

[28] Coley, D. A., "An Introduction to Genetic Algorithms for Scientists and Engineers". World Scientific Publishing Co., Singapore, 1999

[29] Karim, M. M., Suzuki, K., and Kai, H., "Optimal Design of Hydrofoil and Marine Propeller Using Micro-Genetic Algorithm". Journal of Naval Architecture and Marine Engineering, December, 2004

[30] Aykut, S., Kentli, A., Gulmez, S., and Yazicioglu, S., "Robust Multiobjective Optimization of Cutting Parameters in Face Milling". Journal of Applied Science, Acta Polytechnica Hungarica, Vol. 9, No. 4, 2012