Thesis for the Degree of Master of Engineering

Ship Hull Form Modification using Shift Method Based on Hydrodynamic Performances Simulated by WAVIS

By

Thandar Aung

Department of Naval Architecture and Marine System Engineering

The Graduate School

Pukyong National University

August 2019

Ship Hull Form Modification using Shift Method Based on Hydrodynamic Performances Simulated by WAVIS

Shift 법과 WAVIS 를 이용한 선형 최적화

Advisor: Prof. Dong Joon Kim

By

Thandar Aung

A thesis submitted in partial fulfillment of the requirements

for the degree of Master of Engineering

In Department of Naval Architecture and Marine Systems Engineering,

The Graduate School

Pukyong National University

August 2019

Ship Hull form Modification using Shift Method Based on Hydrodynamic Performances Simulated by WAVIS

A dissertation

by

Thandar Aung

Approved by:

Professor Yong-Jig Kim

(Chairman)

Professor Jong-Yong Park

(Member)

Professor Dong-Joon Kim

(Member)

August 2019

Contents

List of Figures

List of Tables

Ship Hull Form Modification using Shift Method Based on Hydrodynamic Performances Simulated by WAVIS

Thandar Aung

Department of Naval Architecture and Marine Systems Engineering,
The Graduate School,
Pukyong National University

Abstract

The field of optimization is attracting widespread interest due to the fact that an optimum design provides economical and performance benefits. Consequently, the demand for optimum ship design based on computational fluid dynamics has increased since the ships that undergo an optimization process are more energy saving and competitive. Designers are making enormous efforts to find optimum ship design based on various constraints with different solvers. Generally, finding the optimum ship design is a process consisting of three modules; Computer-Aided Design/ Manufacturing (CAD/CAM) and modification module, computational fluid dynamic (CFD) module and optimization module.

CAD/CAM module generates the geometry definition of ship hull. The modification module adjusts the ship hull according to the necessary conditions of the optimization process. The CFD module analyzes the flow around the ship and hydrodynamic performances of a ship hull with different solvers. The optimization module finds the minimum or maximum design under various constraints which are dependent on the designer's preferences. Optimization can be considered as an iterative process that continuously modifies variables until a 'perfect' optimum achieved in other words, the condition which meet all the constraints.

To get the reasonable geometry of a ship during an optimization process, efficient and effective modification methods play a crucial role. Among all the available modification methods this paper will focus on use of the shift method. The shift method is a technique extended from Lackenby's approach and deals with the sectional area curve, modifying the ship hull geometry by moving the sections in a longitudinal direction. In describing the ship hull form, the sectional area curve is one of the most important aspects that influences on the geometry of hull. The author of this study selected the shift method for modification of ship hull form because this method has demonstrated to be simple and effective. Nowadays, most of the ship designers use this method because of its practicality.

As the use for ships optimization based on CFD has increased, various CFD solvers such as OpenFOAM, ANSYS and SHIPFLOW have been developed despite the fact that each solver has limitations when connecting to the CAD software. Among these choices for CFD solvers, it is necessary to select which one is suitable to solve specific problems and for the required modifications.

To reach to a continuous process of optimization, it is necessary to link the previously mentioned modules to achieve a computational efficiency. Since each one of the CFD solvers have some limitation on linking, there is a possibility for difficulties to occur such as compatibility and functionality problems. For example, OpenFOAM has its particular format to link with certain CAD modules. Based on this, the range of software options are limited.

The aim of this study is to develop a program to link the modification module and CFD module to reduce the limitations that can appear during the optimization process of hull form. For the modification module, a shift method will be used and the CFD calculation will be simulated by WAVIS (version 1.3) as a potential flow solver in this study. The reason of using a potential solver lies in the fact that computation times are lower than viscous solvers. As an application example, KCS ship hull form is used for validity tests and KCS hull form was modified by the integrated program developed by the author of this paper.

Keywords: WAVIS, shift method, programming language C++, curve modeling, optimization, SQP (Sequential Quadratic Programming)

1. Introduction

1.1 Background

Modern Industry has become a highly competitive field. This constant competition has led companies to make efforts in order to produce the best products within the shortest periods of time and obtaining the highest profits. Similarly, in the ship building area, companies have the purpose of building the most efficient and greener ships, since the global emission standards and international competitive pressures have increased year by year. Moreover, companies are also exposed to the market pressures pushing ship builders to deliver ships in a shorter time-scale, with increasing complexity and modularity, and to comply with environmental rules, while lowering initial build cost and operational costs. This is the reason for the increasing demand for optimization techniques for ship hull forms. In addition, the following facts can be the cause for the increased demand for optimization.

First, the traditional ship design process, which is highly dependent on both; data base and naval architect's experiences, has already reached its limits. In this regard, naval architects need to develop eco-friendly and efficient ship technologies according to the recent market demands. Thus, it is necessary to modify the old traditional practice in which basic hull form is obtained from a data base or experience. In that case, several questions arise: how to modify the hull form? Which part of a hull should be changed? Which constraints need to be accounted in for a successful modification? Which is the optimal method for a hull modification? Etc. Several researchers have tried to find appropriate and modern techniques to change the traditional method of design process, however, modification of ship hull form is a process that requires to consider numerous variables like labor cost, material cost, and maximum capacity, among others. Consequently, optimization has become a commonly-discussed topic among all the industry fields since optimum designs can offer better solutions in terms of costs and time. The following examples show the reasons why optimization is a necessary technique for problems with several considerations.

In the case of designing a LNG-Fueled propulsion engine by replacing the existing heavy oil fuel propulsion engine, one of the most difficult tasks is to select the type of engine because installation equipment, space and potential risks need to be taken into account during this process.

Recently, there has been an announcement about the Panama Canal to extend its length in order to retain more market share. Consequently, a new class of ships was developed to adjust the increment of canal. In this case, optimization techniques are applied to provide the optimum hull form with the maximum breadth.

Another example is the nesting problem in the ship building field. Designers are trying to develop the most effective nesting algorithm to reduce costs based on material and computation time and path.

Regarding the previous examples, optimization can provide specific solution when confronting these and many other similar challenges in the development and building of new types of vessels or other type of fields with diverse physical features and characteristics like SWATH (Small Water plane Area Twin Hull), planning hull, air cushion vessels, among others. For each of the mentioned problems, optimization techniques can provide suitable solutions. Generally speaking, most researchers recommend the use of optimization in such complicated conditions.

Second, ship design is an iterative process composed of an early design, concept design and detailed design stages. Since ship design is complex and requires a successful coordination between many different fields, decisions made at the early stages of the process can influence the performance of the ship hull, building costs or any other function of the final product. In other words, if the optimization technique is correctly implemented, its benefits can be plausible since the very first step of the early design stage.

In CAD, as computer systems rapidly developed throughout the second half of the twentieth century, the performances and geometric representations were also updated thanks to Bezier curve, B-spline, and NURBS (Non-uniform Rational B-spline) representations. This development represented not only the implementation of arbitrary geometry, but also the manipulation of geometry became much more intuitive than the old polynomial representations. That helped the CAD-designers to perform work more simple, easy and efficient ways in comparison to the past.

If we look back to the ship design process history, after the CAD system was developed, the demand for optimization exponentially increased in the same degree. Researchers have explored ways to develop the optimum ship hull form under various considerations such as production costs, time constraint and ship characteristics. Since the ship design process is a complex activity and needs to deal with several decisions, researchers tried to involve the CFD-based optimization process to solve the problems that can occur in complex ship design process as mentioned before. When using the optimization technique, the final hull form needs to be substantially better than its initial form in terms of time-money costs for the process to be considered successful. Additionally, it is expected for numerous hull form modifications to occur in order to achieve the best possible version of ship hull form. CFD-based optimization was initially introduced by the aerospace community for the design of 2-dimensional airfoil sections (H. Kim & Yang, 2010). Since then, these techniques has been spread widely in many other engineering design applications, being one of them the ship design process.

1.2 State of the Art

In general, CFD-based ship hull form optimization is a combination of three processes which are the modification of the initial hull form, hydrodynamic performance analysis of a modified hull form and the selection of the optimum hull form. In other words, this process is composed of three main modules; CAD and Modification module, CFD module and Optimization module shown in Fig. 1.1. Each one of these modules were developed to be applied under different conditions (Park, Choi, & Chun, 2015).

Various modification of hull form techniques have been developed such as vertex control, data point manipulation and other types of modification of form parameters. Vertex control method expresses the hull form with B-spline curve surface and modifies the hull form by shifting the vertices, which are used as the design parameters(Choi, 2015; H. . Kim & Chun, 2000). This technique has good flexibility for the hull form modification but might get modified with poor fairness since it is controlled by the vertex. Another drawback of this technique is the number of design variables that are required to be controlled is larger than in other methods. Fairness and flexibility are both important for the ship hull form but it is difficult for the designer to get the reasonable portion for both. The ship hull form is modified by high entities which are form parameters in geometry (S. Harries, Abt, & Hochkirch, 2004; S. Harries & Systems, 2014). This technique is called surface modeling, which is considered a parametric modeling. Parametric modeling gives a greater flexibility and fairness; moreover, it is easy to use. For this technique, the hull form requires to be expressed in advance as form parameters. The hull form modification is controlled by the modification functions, so there can be alternative modification functions.

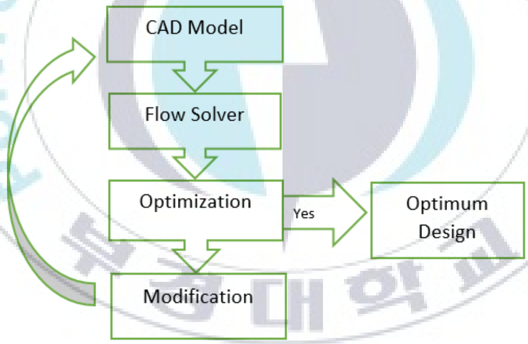


Fig. 1.1 Process of CFD-based ship hull form optimization

CFD-based optimization has become a popular trend in the ship building industry. Alternative CFD-solvers are also being developed to analyze the flow around the ship. There are two common methods for flow analysis; Potential and Viscous flow analyses. Numerous flow analyzers such as ANYSIS, WAVIS, OpenFOAM, StarCCM+ and many others have been developed in the past few years. Potential flow analyzer has the advantage of requiring less time to compute hydrodynamic performances compared to viscous one because it focuses on calculating waves making resistance. Unlike potential flow solver, viscous flow analyzer uses RANS solver and tends to consume more time because it analyses the ship hull boundary surface, free surface and flow domain, which is recommended for parts of the ship hull which has an important influence of viscous flow such as the stern and propeller parts.

Nowadays, several optimization methods have been developed. Some of them are based on deterministic or stochastic algorithms are currently used for ship hull form optimization. In general, we can speak of global and local optimization. A global optimization algorithm for multi-objective problems has been developed for both commercial container ship and a destroyer ship (Campana, Peri, & Rossetti, 2001). SQP method was employed to optimize the DTMB Model-5415 (Tahara et al., 2012; Tahara, Tohyama, & Katsui, 2006) and to obtain the optimum hull form of KSUEZMAX (Park et al., 2015). (Bagheri & Ghassemi, 2014) applied Genetic Algorithm to find the optimum hull form Series 60 and Wigley hulls with respect to seakeeping performance. There is various types of optimization such as gradient-based method, global search method, direct search, Genetic algorithm (GA) and Particle Swarm Optimization (PSO) and among others.

1.3 Motivation

Ship design is a complicated matter since ship systems are complex and its processes are multi-level relationships that depend on each other: people, tools, organization, etc. Based on this, in order to obtain a better process of ship design, those processes should be in harmony with each other just as a group of people should work in harmony. Since ship building market requirements have exponentially increased in the past few years and companies have also invested in team building for the success ship building process, it is expected for the outcome of the harmonized team and synthesized efforts to be more than the sum of its individual parts. In this regard, FRIENDSHIP SYSTEMS has introduced a new CAE (Computer Aided Engineering) environment to allow for better use of CAD and CFD (C. Abt & Harries, 2007; C. Abt & Harries, 2007b, 2007a).

Today, there exist several alternatives to commercial software such as CAD and CFD that have been developed in the practical field and since optimization has become the trend in diverse fields, it is normal for this trend to be gaining popularity in the ship building industry in order to find CFD based optimum hull. The systems of finding the optimum ship hull and research of ship characteristics are standalone most of the time but they are connected via files. For example, geometry is given by offsets, iges files or some other files format and exported and converted as some legacy formats to be able to read in CFD. This is process is called 'pre-processing for CFD'. In this step, the conversion of file format might differ from tool to tool. In fact, it is common to have several conversions and preparations for the same geometry in a single optimization process. If after these pre- and post-preparations are done, then we can finally start the numerical simulations and get results. Despite the fact that there are diverse commercial softwares, few of them are integrated software for CAD and CFD for the calculation convenience. FRIENDSHIP-framework is one of the few integrated CAD and CFD software available right now. With this in mind, the aim of this dissertation is to take into account existing codes rather than to supply the maritime community with a new system. This program will modify the geometry by using a Shift Method to get several shapes of hull form. For CFD code, WAVIS version 1.3 is used. For optimization, SQP (Sequential Quadratic Programming) is applied. This is an integrated program for CAD and CFD for effective and easy calculation of CFD-based optimum ship hull form.

1.4 Outline of This Dissertation

This paper is divided into six chapters. The general contents of each chapter can be summarized as follows:

Chapter I Introduction: This chapter describes the background of the ship design process and motivation of this thesis.

Chapter II Geometry Modeling: In this chapter, there is an explanation about ship building process, brief overview of geometry modeling and discussion on different kinds of modeling techniques. Additionally, an explanation about the curve models and curve fitting are offered.

Chapter III Brief Overview of WAVIS: This chapter describes the explanation about WAVIS version 1.3.

Chapter IV Optimization Solver: This chapter describes brief summary of the optimization and SQP (Sequential Quadratic Programming) optimization.

Chapter V About the Integrated Program of CAD, CFD and Optimization: The program developed for CAD and CFD is presented and explained. As an addition, the application example with KCS hull is presented.

Chapter VI Conclusion and Further Study: This chapter describes the conclusions of this paper and suggestions for future studies.

2. Geometry Modeling

2.1 Ship Design Process

Formerly, the ship design process was treated more like an art than science, mostly depending on the designer's experiences and numerous databases. The process was used to be heuristic method, namely method derived from the knowledge gained through a process of trial and error.

Ship design is a complex activity that requires a successful coordination between different disciplines of both; technical and non-technical nature, with the objective of creating a valuable and optimum design solution. Ship design must be considered as a sequential process, increasing its detail level step by step, until a single design that satisfies all constraints, balancing all considerations is obtained. Fig. 2.1 shows the traditional design spiral process and indicates several refinements that have been made to get a detailed ship design. The process combines both synthesis and analysis in a sequential process. The traditional work flow has the objective of studying one issue at a time, advancing the design step by step, undertaking modifications and establishing refinements iteratively, allowing an increase in complexity and precision across the design cycle. Several improvements have been made to the design spiral, since it is an inefficient method for handling complex and simultaneous design changes, especially when doing late changes on variables that affect the ship's performances characteristics. Although the spiral approach may result in satisfactory designs, the time-related costs are tremendous.

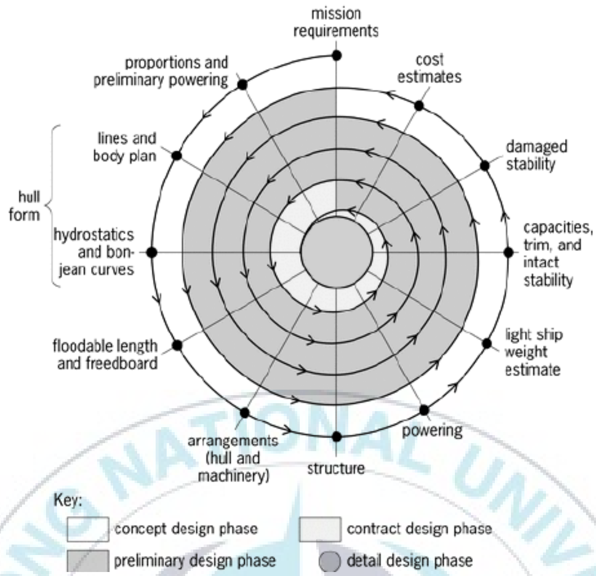


Fig. 2.1 Design Spiral (traditional) (Internet Source)

Nowadays, with evolution of computer hardware and software, more parts of the design process are virtually approached, especially, the heavy calculations and drafting elements of the ship design. Moreover, CAD/CAM systems are largely promoted with curves and surfaces. The non-uniform rational B-spline (NURBS) curves and surfaces have been used in many CAD/CAM systems. Since NURBS can provide a common mathematical expression for analytical and free-form curves and surface, they have been mainly used in geometry modeling. The followings are some reasons for the use of computers in ship building process.

- **Aesthetic Characteristics:** Since modern computers used the mathematical expression for curves and surfaces, smooth linear curve and curved surfaces can be generated effectively with the given conditions.

- **Accuracy and Consistency:** Since the early design stage characteristics can be produced within the allowance error without loss of information, the accuracy and consistency increases.
- **Evaluation of Hydrodynamic Properties:** Since ships are computationally analyzed, shape and structural analysis function can be analyzed in advance.
- **Time reduction**: During the process of ship design, several modifications require less time if they are computationally analyzed.

2.2 Brief Overview of Geometry Modeling Techniques

With the strictly definition of environment care in these days, the optimum ship hull need to consider the so-called 'green' characteristics. Consequently, the hull form optimization has set to become a vital factor in ship building industry. Since the ship hull form is a platform that supports the entire ship systems, and is one of the most important factors in ship hydrodynamic and structural behaviors. The success of the entire ship systems, from production to the final design stage, mainly depends on the hull form definition of the ship. Ship hull form is usually designed by existing hull. The main criteria of hull form definition is fairness of waterline, section lines, etc., which are met with an acceptable shape and with relevant physical and geometric characteristics. This fact makes the geometric modeling techniques become a relevant feature in ship hull definition.

In the 1980s, the ship hull design tremendously advanced helped by computational technologies and by combining traditional hull design methods and parametric surface representation such as NURBS representation for the ship building process. As a result, NURBS curves and surfaces have become the most popular technique for describing the shape of hull. In NURBS representation, control vertices can be interactively manipulated to get the new geometry easily.

There are numerous alternatives hull geometry modeling techniques that have been developed. (S. Harries et al., 2004) analyzed various modeling techniques and divided them into two categories; conventional and parametric modeling techniques (See in Fig. 2.2). Most of the modeling techniques are compared by in terms of flexibility, efficiency and effectiveness. Nowadays modeling techniques are diversifying; the reason of their existence is their ability to stand for diverse purposes. Each method can modify the geometry in different ways respectively. So, here the question will arise, "What are the basic properties to let the modeling techniques to become diverse?" The next are some of the basic abilities for each modeling techniques to exist.

- **Flexibility:** The ability to cope with any possible shape. In other words, the ability to be easily modified for any types of shape.
- **Efficiency:** The swiftness with which information (geometry) is generated.
- **Effectiveness:** The quality of the outcome like fairness, completeness, and correctness.
- **Cost:** The value of resources used for producing something new.

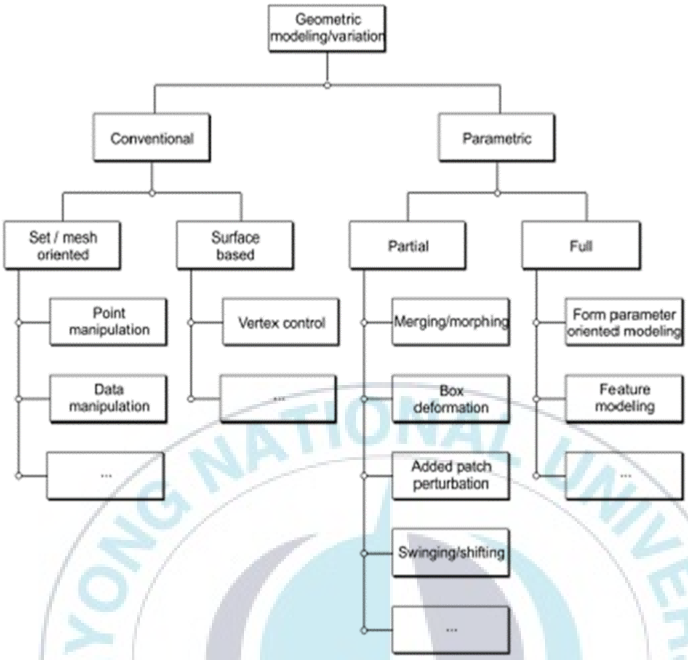


Fig. 2.2 Modeling Techniques (S. Harries et al., 2004)

2.3 Conventional Modeling Techniques

The conventional modeling techniques build on low definition of geometry. As an example, points are used to define the curves and curves are used to define the surfaces. This characteristic gives great flexibility even though it requires special care to ensure the production of a reasonable geometry. NURBS representation can be regarded as one of the conventional modeling techniques. Some of the NURBS control points can be used as the design variables in modification geometry. Some other examples are point manipulation and vertex control in which the designers can manipulate the geometry by moving the control points and or vertex points, directly modifying the design from the lines.

In this technique, the designer can describe the geometry at will, but requires great amount of time, which is one of its weak points. Another drawback of this technique is the need of large number of design variables to modify the geometry that can lead to unreasonable geometry if the control points are wrongly manipulated, requiring special care for each point.

2.4 Parametric Modeling Techniques

Unlike the former techniques, instead of generating a lot of points to define curves and hull form, parametric modeling used high entities like form parameters to generate the hull form. The main advantage of this method is that those entities can be used as design variables so the number of variables to consider can be reduced than conventional modeling techniques. This method can also balance between the simplicity and variability or in other words, has the ability to balance between freedom and restriction. There are various methods of parametric modeling, for example, merging or morphing, box deformation, swinging or shifting and so on.

The main advantage of parametric modeling techniques is the consideration of small numbers of design variables to modify the geometry. In parametric modeling, the diversity of possible hull forms is confined by the topology and the modified design is rules by the form parameter sets.

- **Merging or Morphing:** This method combines the two or more shapes to produce a new shape. (See in Fig. 2.3)
- **Box Deformation**: In this method, a parent shape is placed into a box and instead of varying the parent shape itself, the box is distorted, dragging and squeezing the original shape. (See in Fig. 2.4)
- **Added Patch Perturbation**: In here, a patch is placed on the top of a given shape and used to perturb the original geometry which itself is left untouched. (See in Fig 2.5).
- **Swinging or Shifting**: This method modifies the geometry by shifting some part of curves or by adding the shape function to the original curve. For example, in ship hull form definition, the sectional area curve of a parent hull is systematically changed and the new hull is determined by moving the entire sections longitudinally to match the new sectional area curve.

From the ship building point of view, this technique can modify ships by easily modifying some form parameters of parent hull, taking into account the hydro properties of existing hull. (S. Harries et al., 2004) researched about parametric hull representation to improve efficiency to define the realistic hull forms. (Markov & Suzuki, 2011) used the B-spline to define the ship hull and used the free surface geometry and free-surface boundary element method for steady ship wave problem. (Choi, 2015) presented the approach of modification function with the used of bell-shaped modification function. (ZHANG, ZHU, & LENG, 2008) used the NURBS to present the hull form and studied the parametric approach to design the hull form which provides quick generation and variation of hull form for hydrodynamic optimization of hull form.

Since there exist plenty of modification methods, the choice of method among thousands of modifications is also an important matter. In ship design process, since the hull form definition plays a crucial role in optimization process, special care is required when choosing the modification method.

The following are considerations regarding modification methods:

- Design variables should be small numbers of parameters.
- Larger variation of hull form can be obtained by modifying the given parameters (design variables)
- Modified region can be joined with the original one without discontinuities,
- Various geometrical constraints can easily be implemented in the optimization process.

In the present dissertation parametric modeling of hull form, Shift method is adopted.

Fig. 2.3 Merging or Morphing (internet source)

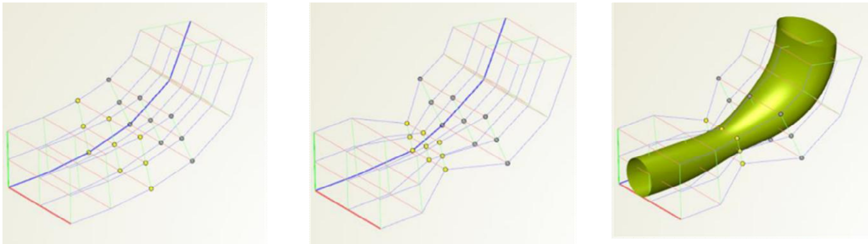


Fig. 2.4 Free Form Deformation

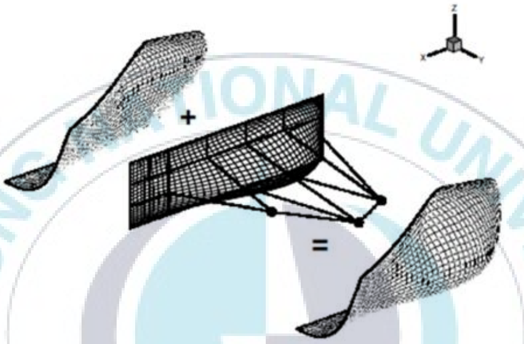


Fig. 2.5 Perturbation (Campana et al., 2001)

2.5 Brief Overview of Shift Method

Shift method is based on the Lackenby's approach and deals with the sectional area curve of the hull. Various researchers have extended the Lackenby method in various forms. Shift method is one of the extended versions of Lackenby. It modifies the hull form by shifting the sections in a longitudinal direction and make the change in longitudinal center of buoyancy, prismatic coefficient and sectional area curve. This method is widely used in ship building process since it is efficient, simple and has practically benefits.

In shifting method, hull forms are represented by using the sectional area curves. A new hull shape can be derived by adjusting the longitudinal spacing of the transverse sections to suit the new curve of sectional area, which is obtained by modifying the initial one through a shape function.

The expression of the new sectional area curve is developed as follows:

$$f^{n}(x) = f^{0}(x) + g(x, \alpha_1, \alpha_2) \qquad (2.1)$$

where the shape function ***g*** is defined as:

$$g(x) = \begin{cases} \alpha_1 \, [0.5(1 - \cos 2\pi \frac{x-\alpha_2}{\alpha_2 - x_1}]^{0.5}, x_1 \le x \le \alpha_2 \\ -\alpha_1 \, [0.5(1 - \cos 2\pi \frac{x-\alpha_2}{\alpha_2 - x_2}]^{0.5}, \alpha_2 \le x \le x_2 \\ 0, other\ wise \end{cases} \qquad (2.2)$$

in which $\boldsymbol{f^{n}(x)}$ represents the expression of new sectional area curve while $\boldsymbol{f^{0}(x)}$ is corresponding to the initial one; ***g(x)*** is the shape function with four arguments; $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ control the range of area needed to be modified; $\boldsymbol{\alpha_1}$ determines the slope of the new sectional area curve, and $\boldsymbol{\alpha_2}$ denotes the location of the fixed station. An application of shape function ***g*** is carried out and the initial curve of sectional area is modified partly. That is shown in Fig. 2.6.

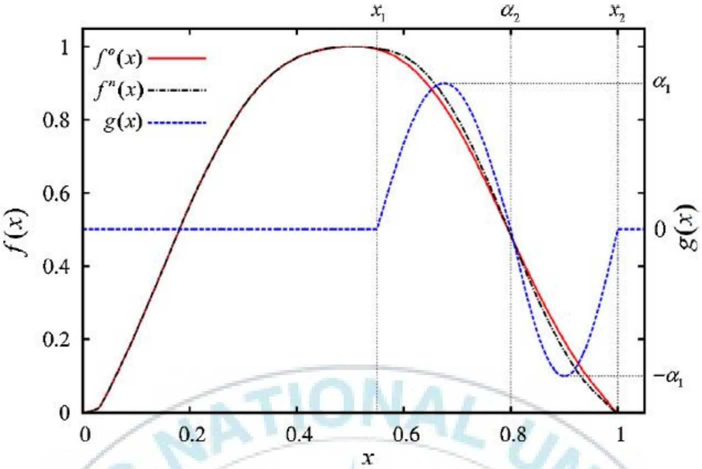


Fig. 2.6 Shift Function (H. Kim & Yang, 2010)

Therefore, if the shift method is employed in optimization process, there will only be four design variables. Once these variables are determined, a new sectional area curve is obtained and the corresponding hull shape also generated. Several hull forms can be generated by changing the previously four variables which are controlling the hull form not to produce unrealistic hull forms. The original sectional area curve can be represented by B-spline, Bezier Curve and NURBS etc. In this study, the NUBS (Non-Uniform B-spline) was used to represent the sectional area curve because of its three continuity properties and convenience to use.

Every process has its own advantages and disadvantages and shift methods are not the exception. As strong points, shift method is simple and effective in changing the ship sectional area curve with only four variables and small numbers of design variables. On the other side, shift method used the cosine, sine function to change the sectional area curve. Since there is a sudden change from zero to one which is the nature of cosine and sine curve, that nature can generate some knuckles in the Cp-curve. Despite this, shift method is still widely used among ship designers because of its practicality.

2.6 Brief Overview of NUBS Curves

With the evolution of computational technologies, CAD/CAM systems used curves and surfaces which are expressed in mathematical expression. So designers need to know about curve models, how can be expressed the hull in mathematical forms and how can be useful in ship building industry.

Moreover, optimization based ship hull design is popular in these days; therefore the modification of hull form play an effective and efficient role during the optimization process, making the role of curve models evenly important. We can consider Bezier curve, Ferguson curve, Uniform and Non B-spline curve and NURBS (Non-Uniform Rational B-spline curve) as the predominant curve forms in the field. Among them, NUBS will be used in this study.

In general, a B-spline curve is a numerically defined function. In order to help its understanding, in here the properties of cubic B-spline curve are presented. B-spline has minimal support with respect to a given degree, smoothness and domain partition.

NUBS is defined in mathematical form by using cubic B-spline basis function $L_j^4(t)$ as

$$r(t) = \sum L_j^4(t) P_j \quad : \quad t \in [t_i, t_{i+1}]. \qquad (2.3)$$

B-spline basis function $L_i^n(t)$ in general, can be called as recursive scalar function defined with respect to a non-decreasing sequence of knot points $\{t_i\}$:

$$L_i^n(t) = \frac{(t-t_i)}{(t_{i+n-1}-t_i)}\ L_i^{n-1}(t) + \frac{(t_{i+n}-t)}{(t_{i+n}-t_{i+1})}\ L_{i+1}^{n-1}(t) \qquad (2.4)$$

where $$L_i^1(t) = \begin{cases} 1, & t \in [t_i, t_{i+1}] \text{ and } t_i < t_{i+1} \\ 0, & \text{otherwise} \end{cases}$$

Let n =2;

$$L_i^2(t) = \frac{(t-t_i)}{(t_{i+1}-t_i)}\ L_i^1(t) + \frac{(t_{i+2}-t)}{(t_{i+2}-t_{i+1})}\ L_{i+1}^1(t) \qquad (2.5)$$

This is the standard method for defining B-spline basis function of degree ***n-1***. In order to simplify the algebraic formula, difference operator ∇ is introduced as a knot span. After introducing the knot spans, *(2.5)* become *(2.6)* with ***n* =*2***.

$$L_i^2(t) = \frac{(t-t_i)}{(\nabla_i)}\ L_i^1(t) + \frac{(t_{i+2}-t)}{(\nabla_{i+1})}\ L_{i+1}^1(t) \qquad (2.6)$$

The composite NUBS-curve is supported by ***n*** knot spans, ∇_0 through ∇_{n-1}, which is called supporting knot spans. The remaining knot spans are called extended knot spans. Depending on the supporting knot span, there can be different shape of curves (See Fig. 2.7) and can also be distinguished like Uniform B-spline, Bezier and Non-uniform B-spline. If the supporting knot spans are identical, this can be called Uniform B-spline, otherwise Non-uniform B-spline. If relative weight is added to NUBS became NURBS (Non-uniform Rational B-spline). These are the special case of NUBS.

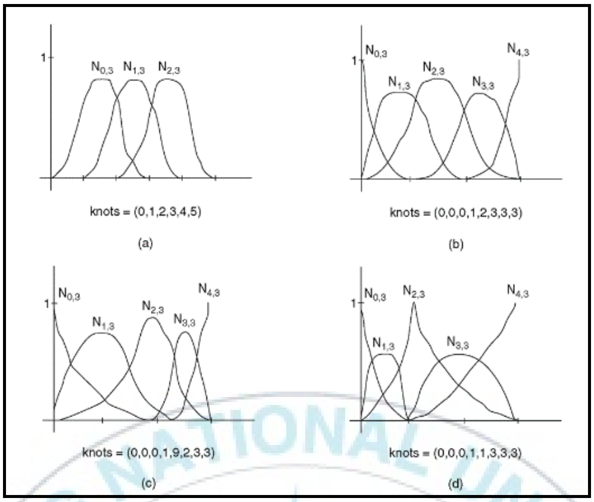


Fig. 2.7 Different B-spline depending on knot spans (internet source)

From a geometrical point of view, NUBS has three continuity conditions:

- **Position Continuity (C^0)**: this means that the end point of curve segment is the same as the starting point of another curve segment or in other words end postions of curves are concide or same position.
- **Tangential Continuity (C^1)**: this means that no abrupt change in slope occurs at the transition between one segment and one segment. In other words, the first derivatives of both curves at the common join are equal.
- **Curvature Continuity (C^2)**: this means that there is no polarity changes in slope at the transition between one segment and one segment. This can be visually recognized as “perfectly smooth”.

Because of these continuities, NUBS will not occur local flatness or discontinuities in defining curves or surfaces. That is the reasons of why

NURBS or NUBS has been used in most of the CAD/CAE systems and also in this study.

2.7 Composite NUBS Curves Fitting

The previous section was about the NUBS curve model and how to express the NUBS curves in mathematical form. In here, how to construct the composite curve from a sequence of data points will be presented. This can be called curve fitting problem. Curve fitting problem is one of the problems to solve a system of linear equations. In here the important part is to connect the curve segments as the smooth composite curve passing through sequence data points $\{P_i: i = 0, 1, \ldots, n\}$.

The choice of a specific curve models depend on the following facts:

- The implementation of curve models should be easy.
- The efficiency of computation should be high.
- The mathematical continuity
- Aesthetic smoothness and
- The flow rate should be uniform which means no flatness or knuckles.

Since the curve fitting problem is basically the problem of solving a system of linear equations, if a second order continuity condition is imposed at each point P_i, a linear equation system for the unknown coefficients of a cubic curve model is obtained. Thus, for curve fitting problem, it is necessary to have suitable set of continuity conditions and to be able to solve a system of linear equations. There can be various types of data points such as evenly spaced data points or unevenly spaced data points. Depending on how the data points spaced, the curve fitting methods will be different respectively.

1) Curve fitting of evenly spaced data points
    - Cubic spline fitting
    - Uniform B-spline fitting
2) Curve fitting of unevenly spaced data points
    - Chord-length spline fitting
    - Non-uniform B-spline fitting

For the methods of curve fitting with evenly spaced data points used Ferguson curve and B-spline curve model for constructing smooth composite curves. If Ferguson curve and B-spline is used for unevenly spaced data point, the drawback is generating the local flatness or kinks. Then, for the unevenly spaced data points, non-uniform B-spline curve model is used. In this method, each chord length is accounted for as a knot span for the respective curve segment. By using non-uniform B-spline and chord length spline fitting, it can avoid of generating local flatness and kinks. The author will focus on the NUBS curve fitting for unevenly spaced data points in here.

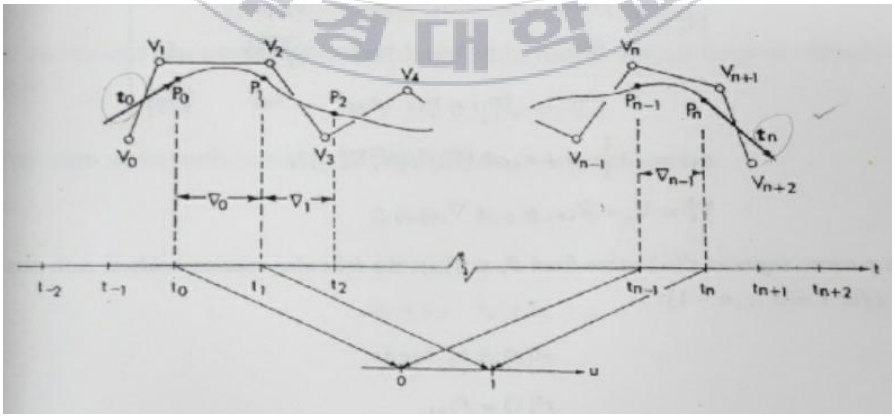


Fig. 2.8 Non-uniform Cubic B-spline curve fitting (Byoung K.choi, 1991)

Fig. 2.8 displays a composite curve of non-uniform B-spline curve segments. In here, there is

- n + 3 control vertices $\{V_i: i = 0,1,\ldots, n+2\}$ and
- n + 4 knot spans $\{\nabla_i = (t_{i+1} - t_i): i = -2,\ldots, n+1\}$.

Input data will be the unevenly data points and end tangents respectively:

$\{P_i: i = 0, 1\ldots n\}$, t_0 and t_n: end tangents vectors of the composite curve.

But need to determine knot spans and control vertices which is the basis of non-uniform B-spline curves.

- $\{\nabla_i = (t_{i+1} - t_i) : i = -2,\ldots, n+1\}$: knot spans and
- $\{V_i: i = 0, 1,\ldots, n+2\}$: control vertices.

The curve fitting procedure is similar to the construction of uniform B-spline models. The process of fitting B-spline curve from data points is called inversion process. In here, determination of end tangents is one of the important tasks in curve fitting procedure and also end tangents are not given in real field. Because of that, end tangents vectors are estimated by various means. In estimating end tangent vectors there are three different methods: free end condition, polynomial end conditions and / or circular end condition.

1) **Free end condition:** This condition is setting the curvatures at the end of the points as zero.
2) **Polynomial end condition:** This is the one by fitting with standard polynomial curve at each ends.
3) **Circular end condition:** This is estimating the end tangents vectors by fitting a circle through the three points at the end.

After estimating the end of tangents vectors, there is a need to determine knot spans $\{\nabla_i\}$. According to Fig. 2.8, the composite curve is supported by $\boldsymbol{n}$ knots, ∇_0 through ∇_{n-1} which are supporting knots and the rest are the extended knot spans. As mentioned before, according to knot spans, there can be different types of curves. In here, knot spans will be equal to the chord length for the sake of reasonable choices. But the choice of extended knot spans does not effect on the NUBS curve smoothness, so in here they are given as zero. Zero knot spans are called multiple knots.

After having determined the knot spans, the next step to consider is building the linear system for the unknown vertices. For this step, we need to recall the NUBS curve equations from early section.

$$r^i(u) = U N_c^i R^i; \quad 0 \le u \le 1; \quad i = 0,1,\ldots\ldots,n-1 \qquad (2.7)$$

where, $U = [1\ u\ u^2\ u^3]$,

$R = [V_i\ V_{i+1}\ V_{i+2}\ V_{i+3}]^T$,

$$N_c^i = \begin{bmatrix} \dfrac{(\nabla_i)^2}{\nabla_{i-1}^2 - \nabla_{i-2}^3} & (1-n_{11}-n_{13}) & \dfrac{(\nabla_{i-1})^2}{\nabla_{i-1}^3\nabla_{i-1}^2} & 0 \\ -3n_{11} & (3n_{11}-n_{23}) & \dfrac{3\nabla_i\nabla_{i-1}}{\nabla_{i-1}^3\nabla_{i-1}^2} & 0 \\ 3n_{11} & -(3n_{11}+n_{33}) & \dfrac{3(\nabla_i)^2}{\nabla_{i-1}^3\nabla_{i-1}^2} & 0 \\ -n_{11} & (n_{11}-n_{43}-n_{44}) & n_{43} & \dfrac{(\nabla_i)^2}{\nabla_i^3\nabla_i^2} \end{bmatrix}$$

$$n_{43} = -\left\{\frac{1}{3}n_{33} + n_{44} + \frac{(\nabla_i)^2}{\nabla_i^2\nabla_{i-1}^3}\right\},$$

$$\nabla_i^k = \nabla_i + \nabla_{i+1} + \cdots + \nabla_{i+k-1}$$

Since a curve segment $\boldsymbol{r}^i$ ***(u)*** spans from $\boldsymbol{P}_i$ to $\boldsymbol{P}_{i+1}$, this

$\boldsymbol{r}^i(\mathbf{0}) = \boldsymbol{P}_i$ and $\boldsymbol{r}^i(\mathbf{1}) = \boldsymbol{P}_{i+1}$ will be the relative ***(for i =0, 1,…., n-1)***.

After evaluating all above equations, we can get the next linear equations:

$$f_i V_i + h_i V_{i+1} + g_i V_{i+2} = P_i \ ; i = 0,1, \dots n. \qquad (2.8)$$

where,

$$h_i = (1 - f_i - g_i)$$

$$f_i = \frac{(\nabla_i)^2}{\nabla_{i-1}^2 \nabla_{i-2}^3},$$

$$g_i = \frac{(\nabla_{i-1})^2}{\nabla_{i-1}^2 \nabla_{i-1}^3}.$$

Then, the end tangents vectors will be evaluated as follows:

$$\tilde{t}_0 = \dot{r}^{\,0}(0) \qquad (2.9\text{-}a)$$

$$= a_0 V_2 + (b_0 - a_0) V_1 - b_0 V_0$$

Where, $a_0 = 3\frac{\nabla_0 \nabla_{-1}}{\nabla_{-1}^2 \nabla_{-1}^3}$,

$b_0 = 3\frac{(\nabla_0)^2}{\nabla_{-1}^2 \nabla_{-2}^3}$,

$$\tilde{t}_n = \dot{r}^{\,n-1}(1) \qquad (2.9\text{-}b)$$

$$= a_1 V_{n+2} + (b_1 - a_1) V_{n+2} - b_1 V_n$$

where, $a_1 = 3\frac{(\nabla_{n-1})^2}{\nabla_{n-1}^2 \nabla_{n-1}^3}$,

$b_0 = 3\frac{\nabla_n \nabla_{n-1}}{\nabla_{n-1}^2 \nabla_{n-2}^3}$, This is the evaluation for the both end tangents vectors. Moreover, the extended knot spans will be defined as zero.

After all the evaluations from equations *(2.8)* and *(2.9)* the linear equation system for NUB curve fitting is obtained.

$$\begin{bmatrix} -3 & 3 & 0 & 0 & . & . & . & . \\ 1 & 0 & 0 & 0 & . & . & . & . \\ 0 & f_1 & h_1 & g_1 & . & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & f_{n-1} & h_{n-1} & g_{n-1} \\ . & . & . & . & . & 0 & 0 & 1 \\ . & . & . & . & . & 0 & -3 & 3 \end{bmatrix} \begin{bmatrix} V_0 \\ V_1 \\ V_2 \\ . \\ . \\ . \\ V_{n+1} \\ V_{n+2} \end{bmatrix} = \begin{bmatrix} t_0 \\ P_0 \\ P_1 \\ . \\ . \\ . \\ P_n \\ t_n \end{bmatrix} \qquad (2.10)$$

This linear system matrix becomes the tri-diagonal matrix and can be solved by Appendix C. While solving the tri-diagonal matrix, care is needed since some of the elements are zero. Since each NUBS curve model has a second order continuity, naturally the composite NUBS curve satisfied the second order continuity. So, it automatically avoid the local flatness and unwanted discontinuities.

3. Brief Overview of WAVIS

3.1 Computational Fluid Dynamic Flow solver

Computation of fluid dynamic is important in ship hull form analyzing process. If hydrodynamic calculation can be involved from the preliminary ship design stage to final design, the ship design would become an efficient and successful system. Moreover, the demand for CFD based hull form optimization is trending these days. Many researchers have studied CFD with different solvers. (Park et al., 2015) presented hull form optimization by minimizing the wave making resistance obtained via Rankine source panel method. (Wu et al., 2017) also implemented the Neumann-Michell (NM) theory to predict the wave drag. FLOWPACK version 2004d, a Reynolds-averaged Navier-Strokes (RANS) solver was developed by (Tahara et al., 2006). During the past few years, CFD solvers have been developed by great quantities of researchers. Nowadays, CFD solvers are getting diverse and using with different flows. (Yu et al., 2017) also optimized the 66,000 DWT bulk carrier in calm water and in waves using WAVIS version 1.3. In this study also, WAVIS version 1.3 will be used. The following is a brief summary about WAVIS version 1.3.

3.2 What is WAVIS?

Since CFD-based optimization gained a lot of popularity these days, several solvers for CFD have been developed. In fact, CFD-based optimization was firstly introduced by the aero-space community for the design of 2-dimensional foil sections. Consequently, in ship building process, CFD-based hull form optimization became a commonly discussed topic. There are several commercial softwares for CFD solver. WAVIS (WAve and VIScous flow analysis system for hull form development) is one of the CFD solvers and construed by KRISO (Korean Research Institute of Ships & Ocean Engineering).

WAVIS is a system for evaluating of ship hull resistance. The main goal of WAVIS is to obtain the resistance of propulsion performance by predicting the wave resistance, wave form, viscous resistance and turbulence via numerical calculation without the need of experimental research which tends to be time and resource-consuming.

3.3 Structure of WAVIS

WAVIS is a standalone launcher which is implemented in Pc's WindowsXP/98/2000/ME environment and input the file form FOTRAN console programs, execute the calculation and analyze the calculation results using Tecplot macro. This is one of the convenience facts about WAVIS. The working procedure of WAVIS to calculate the resistance of ship hull form is simple, first, it requires to define the hull form using offset table first and then generate the meshing of ship hull form using WAVIS's module. After all the preliminary setting is done, we can start the calculation for wave resistance, wave form and friction resistance. WAVIS version 1.3 doesn't have the viscous flow solver and only for potential solver. So the calculation time becomes even shorter, requiring around one or two minutes to finish the potential flow solver.

To calculate the wave resistance or friction resistance in WAVIS, first, it is required to define the ship hull form. Generally, the offset given by 25 to 30 stations cannot accurately represent the surface of bow and stern. Therefore, we need data from stations that are densely distributed in the forward and aft sections. WAVIS's X station module allows to get data from these densely distributed stations. This kind of dense station data can be used data generated by other CAD program. Currently, among the CAD output types, format of HCAD output (.pnl, .prf) or the format of input data of SHIPFLOW can be used without any modification.

In WAVIS there are 9 important programs for performance (See in Fig.3.2), offset view, x-station, surface mesh, field grid O-O, field O-H, FS panel gen, potential solver, viscous solver and wake analysis. See reference figures as follows:

1) **Offset View:** It displays the body plan and the centerline contour by reading the original offset table in WAVIS standard format or the converted offset which is done in Offset view program. (See in Fig.3.3)

2) **X station**: By reading the converted offset from the original offset table of WAVIS standard format or from the Offset view or X-station, this program creates the body plan and centerline contour, and define the end of each waterline mathematically (See in Fig.3.5). This creates a more dense stations at bow and stern to generate station data points and centerline contour points output files at certain longitudinal positions. (See in Fig.3.4)

3) **Surface Mesh:** This program read the output files from X-station program (X-station's fine x-station file, side profile file) or HCAD output (*.pnl,*.prf), or SHIPFLOW input data file. Then generate a hull surface grid. (See in Fig.3.6)

4) **Field Grid O-O:** The fore body surface mesh and the aft body surface mesh, which are the outputs of WAVIS Surface Mesh module are read to create a three-dimensional space grid for viscous flow calculation with O-O topology. This module accepts only a single-block hull surface grid for the fore body and aft body each. (See in Fig.3.7)

5) **Field Grid O-H:** Same with Field Grid O-O, but in here used with O-H topology and it accepts not only single block but also multi-blocks hull surface grid.

6) **FS Panel Gen:** This program generates free surface panel by using information of Froude number and given surface mesh when solving potential flow to calculate wave resistance and waveform. (See in Fig.3.8)

7) **Potential solver:** This is for calculation of the linear and non-linear harmonic problem of monotonic line (ship hull) using Raised Panel Approach based on the method of Rankine Source Panel. (See in Fig.3.9)

8) **Viscous Solver:** This is for simulation of turbulent flow phenomenon around a single line by using finite volume method and SIMPLEC method to solve the Reynolds-averaged Navier Stokes equations and the continuity equations which are the governing equations of turbulent flow. In here, the effect of free surface effect is not taken into account of consideration. Instead of this, summary of condition is shown.

9) **Wake Analysis:** This is a program to perform Nominal Wake, Harmonic Analysis., etc. from the velocity flow field at the section of the propeller obtained from the viscous flow calculation.

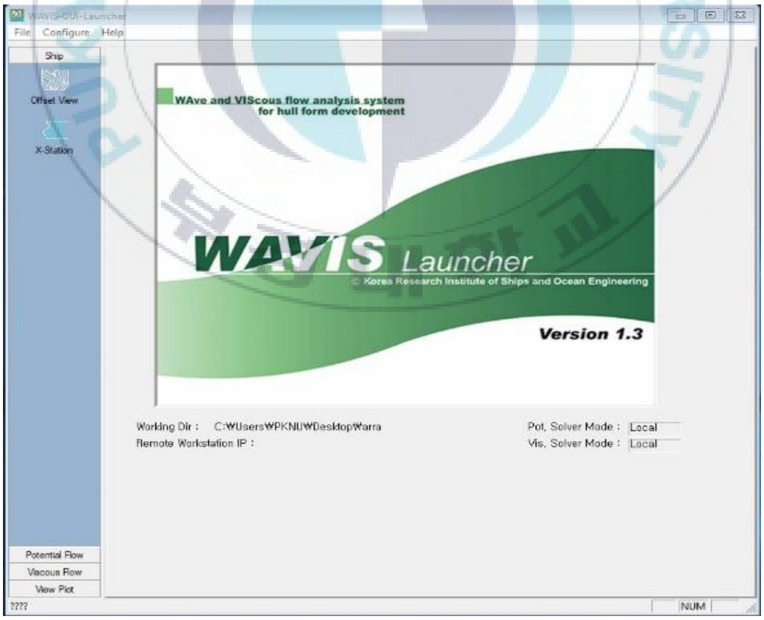


Fig. 3.1 Start Page of WAVIS

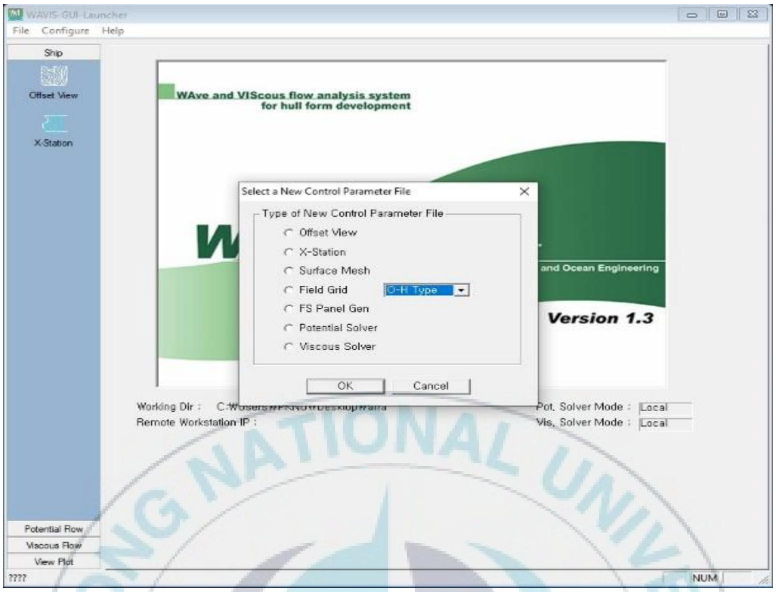


Fig. 3.2 Important 9 programs of WAVIS

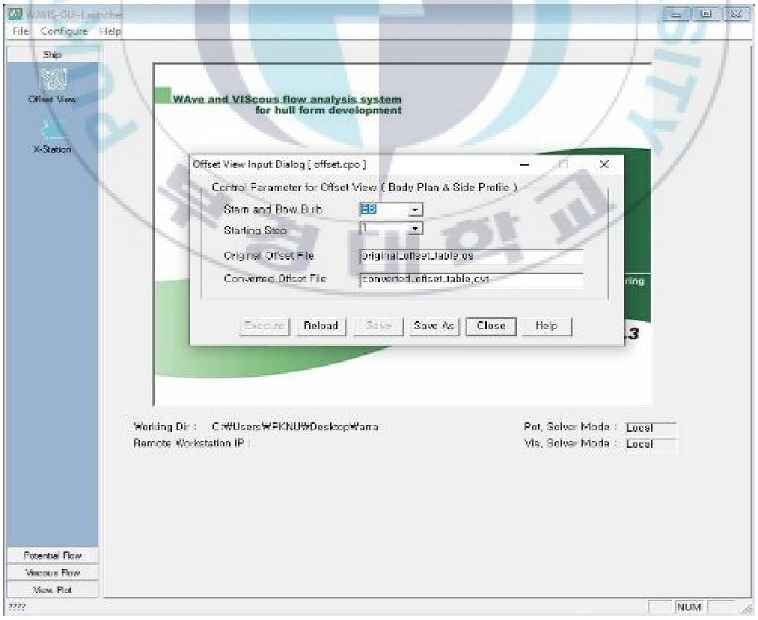


Fig. 3.3 Offset View module of WAVIS

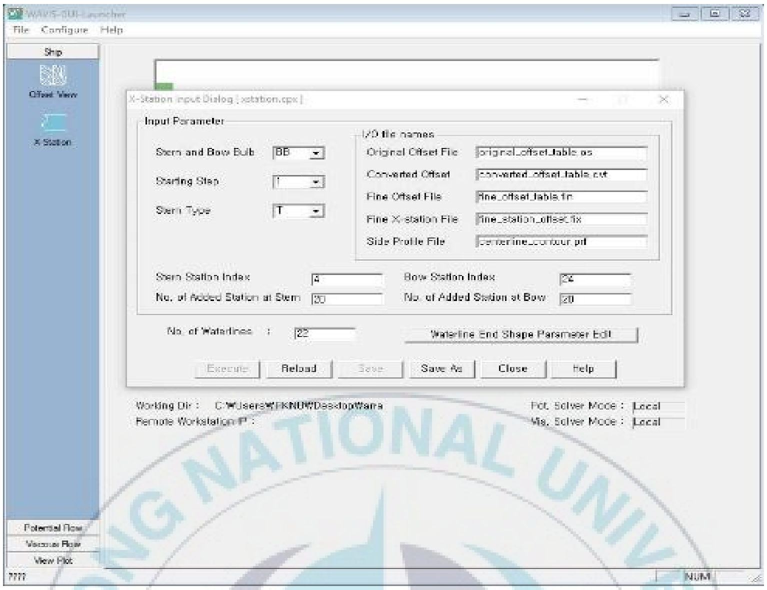
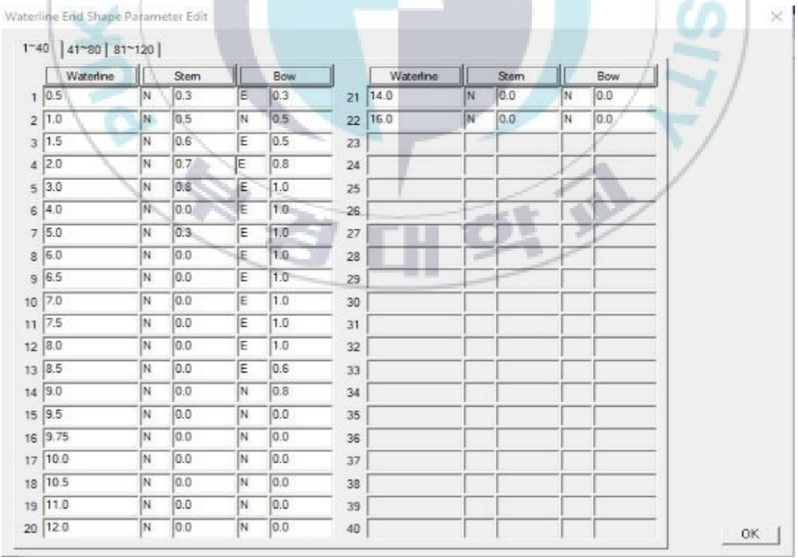


Fig. 3.4 X-station module of WAVIS



	Waterline	Stem		Bow	
1	0.5	N	0.3	E	0.3
2	1.0	N	0.5	N	0.5
3	1.5	N	0.6	E	0.5
4	2.0	N	0.7	E	0.8
5	3.0	N	0.8	E	1.0
6	4.0	N	0.0	E	1.0
7	5.0	N	0.3	E	1.0
8	6.0	N	0.0	E	1.0
9	6.5	N	0.0	E	1.0
10	7.0	N	0.0	E	1.0
11	7.5	N	0.0	E	1.0
12	8.0	N	0.0	E	1.0
13	8.5	N	0.0	E	0.6
14	9.0	N	0.0	N	0.8
15	9.5	N	0.0	N	0.0
16	9.75	N	0.0	N	0.0
17	10.0	N	0.0	N	0.0
18	10.5	N	0.0	N	0.0
19	11.0	N	0.0	N	0.0
20	12.0	N	0.0	N	0.0
21	14.0	N	0.0	N	0.0
22	16.0	N	0.0	N	0.0
23					
24					
25					
26					
27					
28					
29					
30					
31					
32					
33					
34					
35					
36					
37					
38					
39					
40					

Fig. 3.5 Waterline Definition of WAVIS

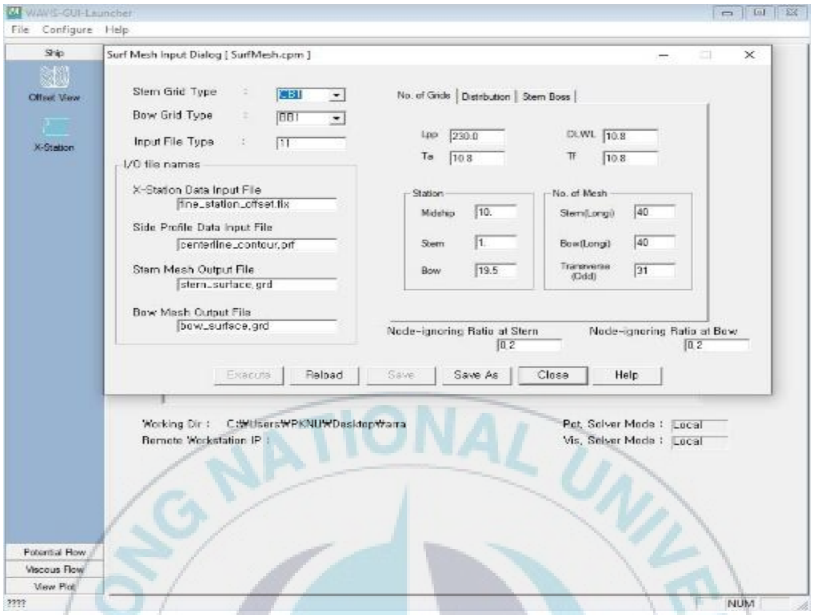


Fig. 3.6 Surface Mesh module of WAVIS

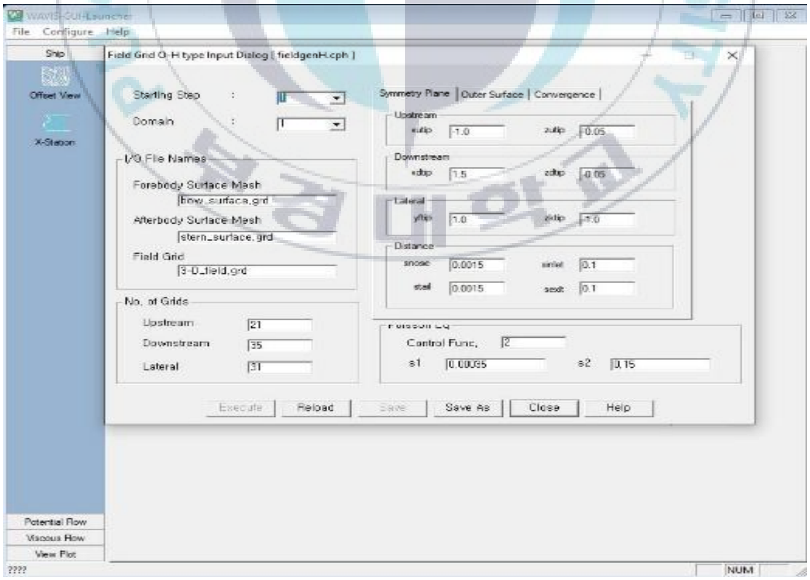


Fig. 3.7 Field Grid module of WAVIS

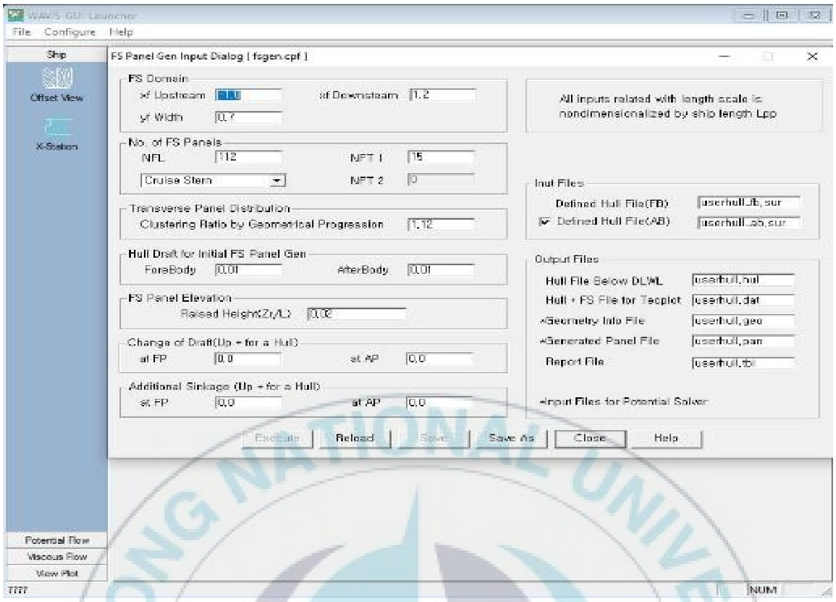


Fig. 3.8 FS panel module of WAVIS

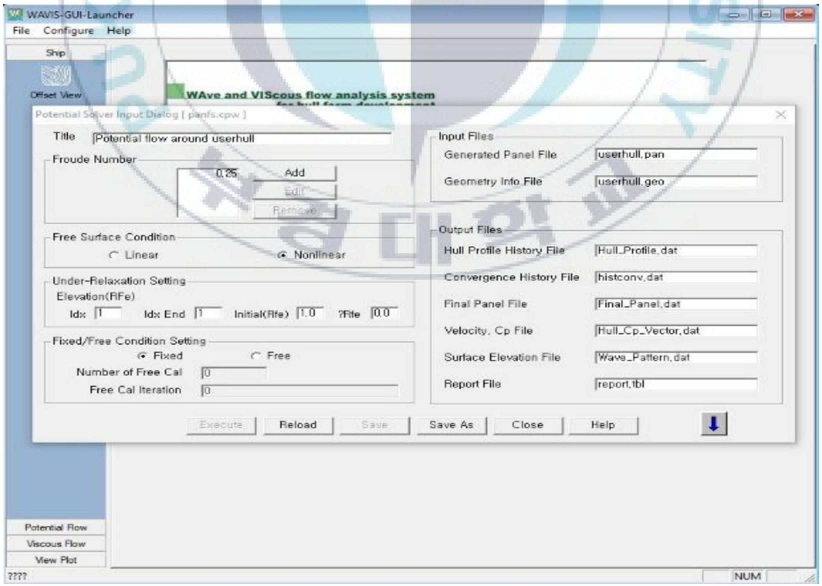


Fig. 3.9 Potential solver of WAVIS

If all the setting are done in each program for potential flow calculation, the process can be finished by one click of execute button. This is one of the simple facts of WAVIS.

For the calculation, first, we need to choose the working directory which is under the configure menu (See in Fig. 3.1). Then, click the potential solver tab from the side tool bar for potential flow calculation. And put the related data for the ship in the programs which are required for potential flow solver, the calculation will be done by few clicks of WAVIS command buttons. Moreover, the calculation time for potential flow solver can be done in around one or two minutes. There will be different output files from each program respectively and they can be analyzed via Tecplot 360Ex. The followings are the output files of WAVIS of potential flow solver.

- Hull Profile.dat: which is the wave profile along the sides of the ship. (See in Fig.3.10)
- histconv.dat: which is the convergence history file with iteration number and tolerance. (See in Fig.3.11)
- Final Panel.dat: which is the free surface panel output file. (See in Fig.3.12)
- Hull Cp Vector.dat: which is the output file of speed and pressure of ship.
- Wave Pattern.dat: which is the surface evaluation file. (See in Fig.3.13)
- report.tbl: which is the output of important results from potential solver.

The convergence criteria used in WAVIS are as follows:

Eps_k: The equation residual of Kinematic Free Surface Condition (0.002)

Eps_d: The equation residual of Dynamic Free Surface Condition (0.005)

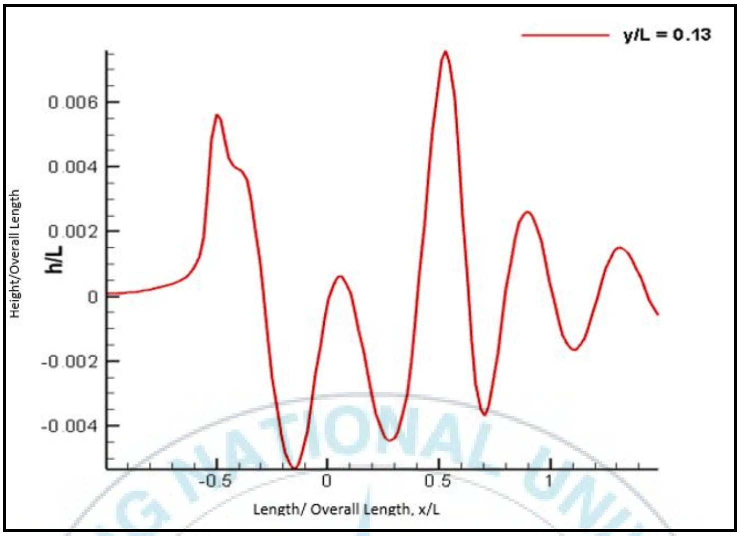


Fig. 3.10 Hull profile history file at y/L (breadth/length overall) = 0.13

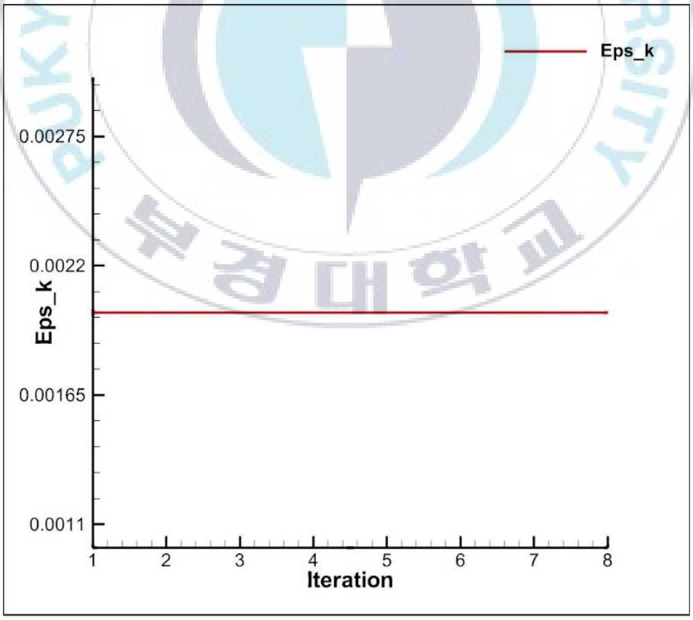


Fig. 3.11 History Convergence file

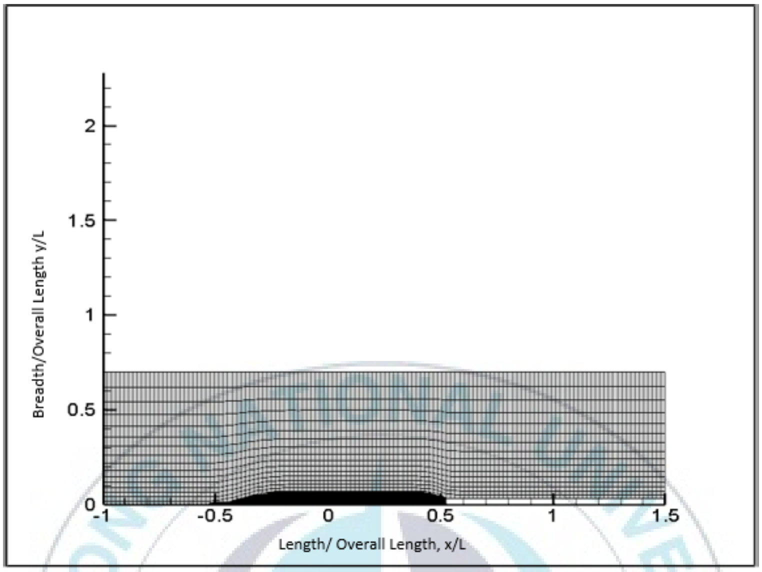


Fig. 3.12 Final panel output file

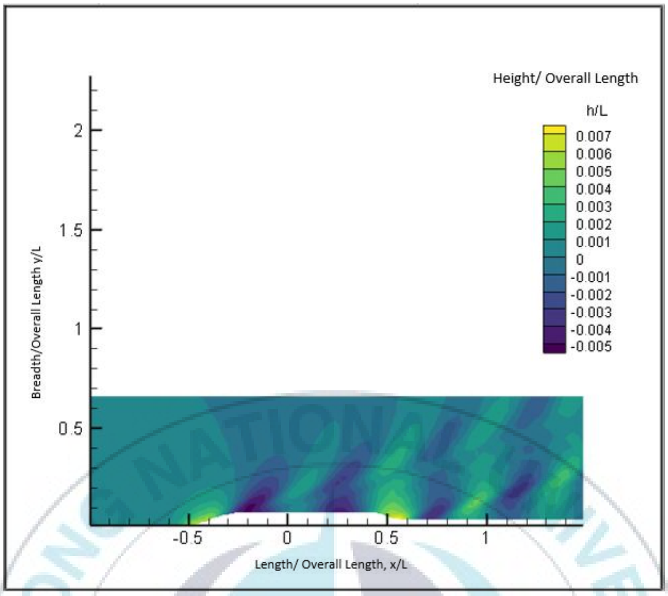


Fig. 3.13 Wave pattern file 2D

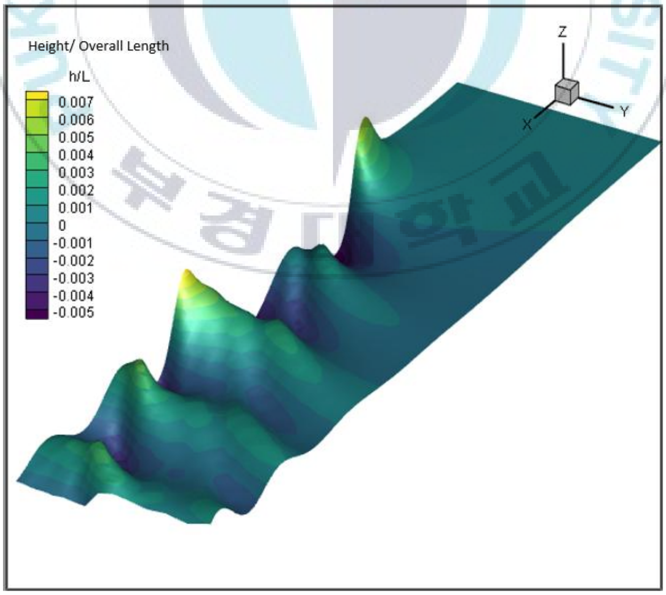


Fig. 3.14 Wave pattern file 3D

3.4 Input files of WAVIS and Format

In this section, the input files formats of WAVIS will be shown. We can conclude that the required file formats needed to evaluate the potential flow solver will be five files, basically; .cpm file (See in Fig.3.15) for the surface grid, two surface grid files for aft and fore part, .cpf file (See in Fig.3.16) for the free surface panel generation and .cpw file (See in Fig.3.17) for the potential flow solver.


```
TB3 BB2
11
KRISO_3600f1a1.sta
KRISO_3600f1a1.prf
grid_aft.grd
grid_fwd.grd
230 10.8 15.6 15.6
10 9 11
31 31 31
0.0 0.0
1.08 1.08 1.01
0.0 3.5
5.4 3.45 4.75 1.0
10 0.5
11 21 8.0
1.10 1.10
9 21 9.5 7.7
1.08 1.01
0.2 0.2
```

Fig. 3.15 kcs.cpm file

```
-1.0 1.5 0.7
129 16 0
1.1033
0.005 0.005 0.0154
0 0.0
0.0 0.0 -0.003 -0.003
2
grid_fwd.grd
grid_aft.grd
out_dlwl.hul
out_hull_fs.dat
out_geometry.geo
out_panel.pan
out_report.tbl
nocreate
```

Fig. 3.16 kcs.cpf file

```
"kcs"
1 0.2600
nonlinear1
15
0.002 0.005
1 100 %
1 25 %
1 1 1.0 0.0
1 5 0.5 0.1
0 0
4 2 4 6 8
0.13
out_panel.pan
out_geometry.geo
out_hproh.dat
out_convl.dat
out_F_Panel.dat
out_CpVect.dat
out_Wavepattern.dat
out_report.out
```

Fig. 3.17 kcs.cpw file

Fig. 3.18 shows how the above mentioned five basis files connected and working to evaluate the wave resistance.

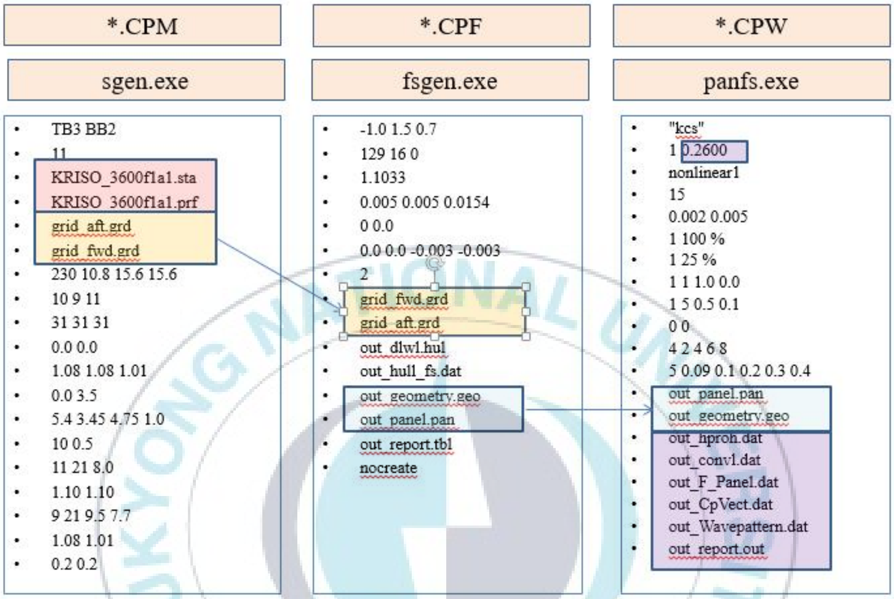


Fig. 3.18 Connection between cpm, cpf and cpw files

4. Optimization

4.1 Optimization Methods

Each engineering field contains a number of well-established activities, including analysis, design, experimenting and researching for systems development. In this regard, most of the processes of engineering research are complicated, time-consuming, and costly, requiring substantial human and material resources. Additionally, these processes are somehow related to each other, making it necessary to be carefully handled from beginning to end. Since system design can be seen from the perspective of optimization, systems can be improved and their performance can be optimized while all other requirements are met. Consequently, many numerical methods of optimization have been developed. The concept of general optimization means to find the best solution with different methods among numerous solutions. Some optimization methods are based on graph and some based on the concept of nature such as Particle Swarm Optimization (PSO) and Ant Colony Algorithm while others are based on evolution algorithm such as Genetic Algorithm. Any problem in which certain parameters need to be determined to satisfy constraints can be formulated as one optimization problem. Once the problem has been defined, one of the optimization algorithms can be used to solve the issue, leading optimization algorithms to be widely applicable in diverse fields and leaving to designers and researchers the difficult task to determine which optimization algorithm to be used in each situation. In this paper, SQP (sequential quadratic programming) was used for optimization.

4.2 Sequential Quadratic Programming

SQP is one of the non-linear programming (NLP) techniques and has been widely used these days. The method has a theoretical basis that is related to (1) the solution of a set of nonlinear equations using Newton's method and (2) the derivation of simultaneous nonlinear equation using Kuhn-Tucker conditions to the Lagrangian of the constrained optimization problem.

Consider a nonlinear optimization problem with only equality constraints as:

Find $\boldsymbol{X}$ which minimized $\boldsymbol{f(X)}$

Subjects to $h_k(\boldsymbol{X}) = 0;\ k = 1, 2 \ldots p;$ *(4.1)*

The Lagrangian function, $\boldsymbol{L\ (X, \lambda)}$, corresponding to the equation *(4.1)* is given by

$$L = f(\boldsymbol{X}) + \sum_{k=1}^{p} \lambda_k h_k(\boldsymbol{X}) \tag{4.2}$$

where $\boldsymbol{\lambda_k}$ is the Lagrange multiplier for the ***kth*** equality constraint. The Kuhn-Tucker necessary conditions can be stated as

$$\nabla L = \boldsymbol{0} \text{ or } \nabla f + \sum_{k=1}^{p} \lambda_k h_k = \boldsymbol{0} \text{ or } \nabla f + [A]^T h = \boldsymbol{0} \tag{4.3}$$

$$h_k(\boldsymbol{X}) = 0, k = 1, 2 \ldots p \tag{4.4}$$

where $\boldsymbol{[A]}$ is an $\boldsymbol{n \times p}$ matrix whose ***kth*** column denotes the gradient of the function $\boldsymbol{h_k}$. Equations *(4.3) and (4.4)* represent a set of $\boldsymbol{n + p}$ nonlinear equations in $\boldsymbol{n + p}$ unknowns $\boldsymbol{(x_i,\ i = 1, \ldots, n}$ and $\boldsymbol{\lambda_k,\ k = 1, \ldots, p)}$. These nonlinear equations can be solved using Newton's method. For convenience, can rewrite equations *(4.3) and (4.4)* as

$$\mathbf{F(Y) = 0} \qquad (4.5)$$

where

$$\mathbf{F} = \begin{Bmatrix} \nabla L \\ h \end{Bmatrix}_{(n+p)x1}, \quad \mathbf{Y} = \begin{Bmatrix} X \\ \lambda \end{Bmatrix}_{(n+p)x1}, \quad \mathbf{0} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}_{(n+p)x1} \qquad (4.6)$$

According to Newton's Method, the solution of equations *(4.5)* can be found iteratively as

$$Y_{j+1} = Y_j + \Delta Y_j \qquad (4.7)$$

with

$$[\nabla F]_j^T \, \Delta \, Y_j = -F\,(Y_j) \qquad (4.8)$$

where $\boldsymbol{Y_j}$ is the solution at the start of ***jth*** iteration and $\Delta \boldsymbol{Y_j}$ is the change in $\boldsymbol{Y_j}$ necessary to generate the improved solution, $\boldsymbol{Y_{j+1}}$, and $\boldsymbol{[\nabla F]_j = [\nabla F\,(Y_j)]}$ is the ***(n+p) x (n+p)*** Jacobian matrix of the nonlinear equations whose ***ith*** column denotes the gradient of the function $\boldsymbol{F_i(Y)}$ with respect to the vector $\boldsymbol{Y}$. By substituting equations (4.5) and (4.6) into equation (4.8),

$$\begin{bmatrix} [\nabla^2 L] & [H] \\ [H]^T & [0] \end{bmatrix}_j \begin{Bmatrix} \Delta X \\ \Delta \lambda \end{Bmatrix}_j = -\begin{Bmatrix} \nabla L \\ h \end{Bmatrix}_j \qquad (4.9)$$

$$\Delta X_j = X_{j+1} - X_j \qquad (4.10)$$

$$\Delta \lambda_j = \lambda_{j+1} - \lambda_j \qquad (4.11)$$

where $[\boldsymbol{\nabla}\,^2\boldsymbol{L}]_{nxn}$ denotes the Hessian matrix of the Lagrangian function. The first set of equations in *(4.9)* can be written separately as

$$[\nabla^2 L]_j \Delta X_j + [H]_j \Delta\lambda_j = -\nabla L_j \qquad (4.12)$$

Using equations *(4.11)* for $\Delta\lambda_j$ and equation *(4.3)* for ∇L_j, equation *(4.12)* can be expressed as

$$[\nabla^2 L]_j \Delta X_j + [H]_j(\lambda_{j+1} - \lambda_j) = -\nabla f_j - [H]_j^T \lambda_j \qquad (4.13)$$

which can be simplified to obtain

$$[\nabla^2 L]_j \Delta X_j + [H]_j \lambda_{j+1} = -\nabla f_j \qquad (4.14)$$

Equation *(4.14)* and the second set of equations in *(4.9)* can now be combined as:

$$\begin{bmatrix} [\nabla^2 L] & [H] \\ [H]^T & [0] \end{bmatrix}_j \begin{Bmatrix} \Delta X_j \\ \lambda_{j+1} \end{Bmatrix} = -\begin{Bmatrix} \nabla f_j \\ \boldsymbol{h}_j \end{Bmatrix} \qquad (4.15)$$

Equation *(4.15)* can be solved to find the change in the design vector $\Delta \boldsymbol{X}_j$ and the new values of the Lagrangian multipliers, $\boldsymbol{\lambda}_{j+1}$. The iterative process indicated by equation *(4.15)* can be continued until convergence is achieved.

Now consider the following quadratic programming problem:

Find $\Delta \boldsymbol{X}$ that minimizes the quadratic programming objective function

$$Q = \nabla f^T \Delta X + \frac{1}{2} \Delta X^T [\nabla^2 L] \Delta X \qquad (4.16\text{-}a)$$

Subjects to the linear equality constraints

$$h_k + \nabla h_k^T \Delta X = 0,\ k = 1, 2, \ldots, p \text{ Or } \boldsymbol{h} + [H]^T \Delta X = 0 \qquad (4.16\text{-}b)$$

The Lagrangian function, $\boldsymbol{L}^{\sim}$, corresponding to the problem of equation *(4.16)* is given by

$$\tilde{L} = \nabla f^T \Delta X + \frac{1}{2} \Delta X^T [\nabla^2 L] \Delta X + \sum_{k=1}^{p} \lambda_k (h_k + \nabla h_k^T \Delta X) \qquad (4.17)$$

where λ_k is the Lagrangian multiplier associated with the ***kth*** equality constraint. The Kuhn-Tucker necessary conditions can be stated as

$$\nabla f + [\nabla^2 L]\Delta X + [H]\lambda = 0 \qquad (4.18)$$

$$h_k + \nabla h_k^T \Delta X = 0, \; k = 1,2, \dots, p \qquad (4.19)$$

Equations *(4.18)* and *(4.19)* can be identified to be same as equation *(4.15)* in matrix form. This shows that the original problem of equation *(4.1)* can be solved iteratively by solving the quadratic programming problem defined by equation *(4.16)*. In fact, when inequality constraints are added to the original problem, the quadratic programming problem of equation *(4.16)* becomes:

Find ***X*** which minimizes $Q = \nabla f^T \Delta X + \frac{1}{2} \Delta X^T [\nabla^2 L] \Delta X$

Subjects to

$$\begin{aligned} g_j + \nabla g_j^T \Delta X &\le 0, & j &= 1,2, \dots, m \\ h_k + \nabla h_k^T \Delta X &= 0, & k &= 1,2, \dots, p \end{aligned} \qquad (4.20)$$

with the Lagrange Function given by

$$\tilde{L} = f(X) + \sum_{j=1}^{m} \lambda_j g_j(X) + \sum_{k=1}^{p} \lambda_{m+k} h_k(X) \qquad (4.21)$$

Since the minimum of the augmented Lagrange function is involved, the sequential quadratic programming method is also known as the **projected Lagrangian method.**

5. About the Integrated Program of CAD, CFD and Optimization

5.1 Programming Language

According to the aim of this study, which is to provide an integrated program to find the optimum ship hull form based on CAD, CFD and optimization algorithm, the author used WAVIS and Shift method which are widely used in ship building field and combined them as a one window program with C++ language for convenience purposes. Since the demand for CFD-based optimization has rapidly increased, the limitation of the currently available CFD solvers to connect with CAD represents an opportunity to develop an integrated program. The author of this paper tried to combine CAD, CFA and optimization modules into one program inspired on the previous success of CAESES, developed by FRIENDSHIP system and having particular interest in potential flow solver which can be simply calculated using WAVIS. Additionally, WAVIS can also provide simple solutions for evaluation of wave making resistance. For the modification module, Shift method can be the best option for the study because of its simplicity when it comes to modify the ship hull by dealing with sectional area curves. The window of the integrated program is presented in Fig. 5.1. As mentioned before, for WAVIS to calculate the potential flow solver, it needs a total of five files, being this is the basis for the program. An explanation about the functions of the buttons of the developed program is presented next.

When clicking on the “READ CPM” button, it will load the *.cpm file and two surface mesh files. Similarly, “READ CPF” and “READ CPW”, will bring out *.cpf file for free surface panel generation and *.cpw file for potential flow solver respectively.

"RUN WAVIS" button will start WAVIS and begin the potential flow solver for wave making resistance.

The "TECPLOT" button will generate the readable files for Tecplot to analyze the results and ship hull form.

"WAVIS OPTIMIZE" will perform the optimization by using SQP (Sequential Quadratic Programming). In optimization procedure, the variables will be the four parameters from Shift method to modify the sectional area curve. By recalling the Shift method, x_1, x_2, α_1 and α_2 will be the design variables. x_1 and x_2 controls the range of Cp-curve modification. α_1 will be the slope of sectional area curve so it controls the longitudinal movement of the Cp-curve. And the final one, α_2 determines the fixed station which doesn't change by the Shift method.

While developing this integrated program, the author found out that the values of x_1 and x_2 are not required to be used as design variables in optimization because the variation of their parameters did not significantly affect the modification of the fore part of ship hull form which is the focus of this dissertation. The modification of the fore part of the ship hull form was limited to the range of x_1 and x_2 because increasing the width's range can affect the midship length and the desire results of optimization process, leaving only two variables for modification. The objective function of the optimization will be to find the minimum C_w produced by WAVIS. For constraint condition, α_1 will be in the range of -0.5 to 0.5 and for α_2, the range will be station number 13 to 18.

"WAVIS MODIFY" button will modify the ship hull form by changing the four variables of Shift method.

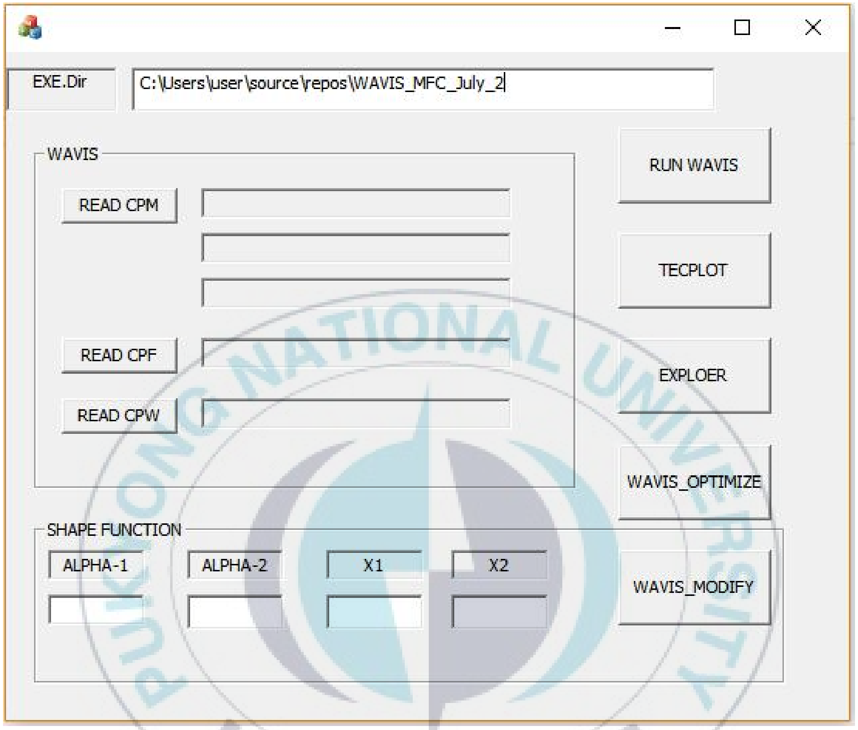


Fig. 5.1 The integrated program

Fig. 5.2 and Fig. 5.3 display the modification of Cp-curve and shift curve with the change of α_2 from station 13 to station 17. Table 5.1 is a sample of the code used for the modification shown in Fig. 5.2 and Fig. 5.3. The procedure of making this integrated program is presented in Fig. 5.4.

"WAVIS_OPTIMIZE" button will automatically modify the Cp-curve according to the Shift method and then run WAVIS for calculation of the C_w and find the optimum hull with the SQP algorithm. Finally, the three modules for finding the optimum ship hull form are now in a combined program without connecting limitations.

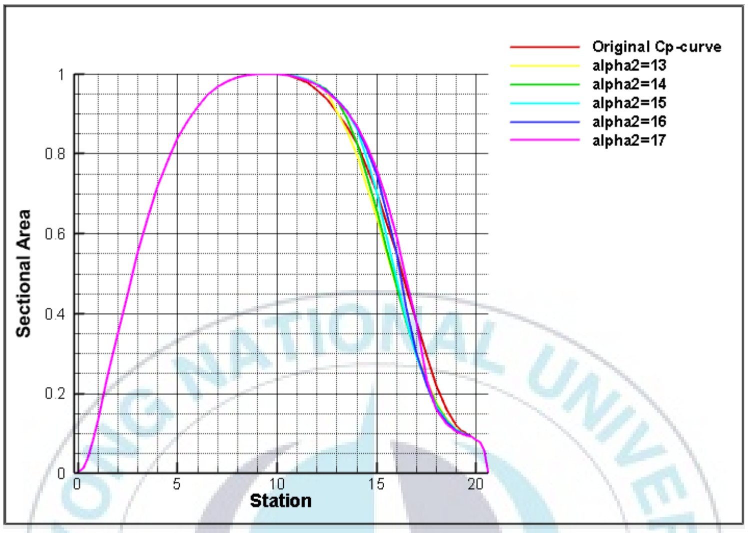


Fig. 5.2 Change of Cp-curve

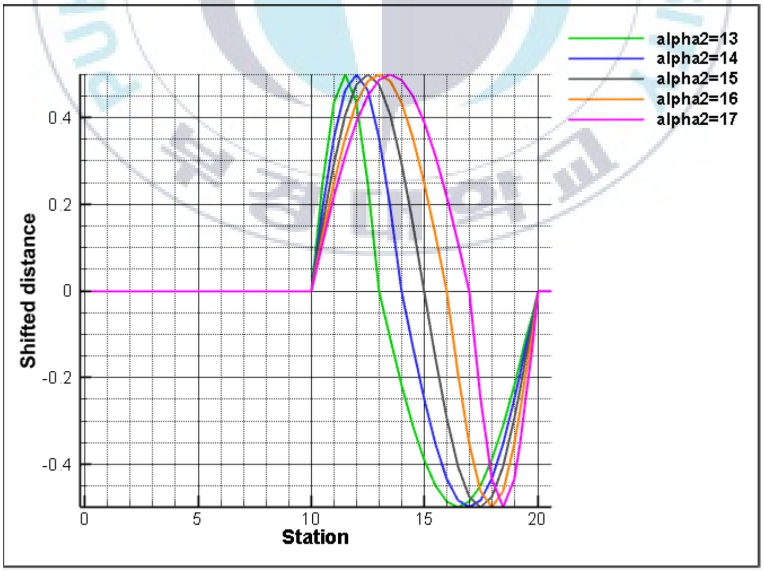


Fig. 5.3 Change of Shift function

Table 5.1. Code for modification

```
void CWAVISMFCDlg::OnBnClickedButtonRunWavisModify()
        for (x = 13.0f; x <= 17.0 + 0.01; x += 1.0f)
{
                x1 = 10.0f;
                x2 = 20.0f;
                a1 = 0.5f;
                a2 = x;

                twavis.get_shape_curveG(x1, x2, a1, a2, ncp, g);
                twavis.modify_cp_with_Shape(ncp, cp, g, cp2);

                add_tecplot_2d("000_cp_curve_old_new.dat", ncp, cp2);
                add_tecplot_2d("000_cp_curve_dx.dat", ncp, g);

                twavis.modify_GRD_cpcurve_fwd(ncp, cp2);

                sprintf(str_section, "000_section_%03d.dat", nn);
                twavis.tecplot_section(str_section);

                twavis.grd_mesh_copy_results_to_origin();


                twavis.runCPFCPW(iexe_wavis,        nn,        str_dir_output,
str_fn_cpw);
                twavis.write_object_function_result(twavis.dir_work_opt,
"[000]-Results.out", str_fn_cpw, nn, "Results");
                nn++;
        }
}
```

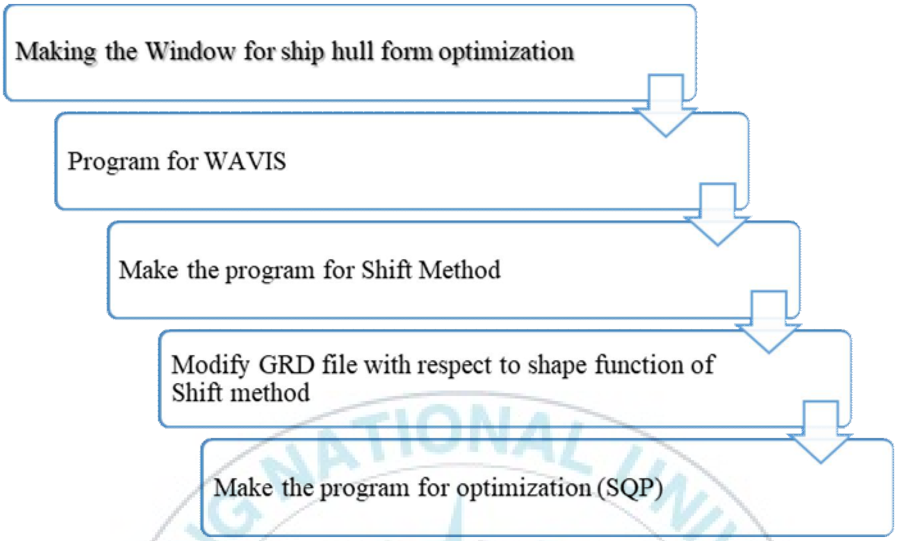


Fig. 5.4 The integrated program's process

5.2 Comparison of Results between WAVIS and the Integrated Program

The author compared the results obtained with WAVIS and the integrated program developed in this study for KCS ship hull form to check whether or not the integrated program's functionality was acceptable. Table 5.1 shows the main dimensions of KCS. The resulting C_w output of the integrated program equally compared with results obtained using WAVIS. Since the integrated program was based on WAVIS for CFD, the output files format and input files format will be the same. The procedure of WAVIS for KCS hull calculation is presented in Fig. 5.5 ~ 5.8.

Table 5.2 Main dimensions of KCS hull

Main Particulars	
LBP (m)	230
LWL (m)	232.5
BWL (m)	32.2
D (m)	19
T (m)	10.8
Displacement (m3)	52030
CB	0.651

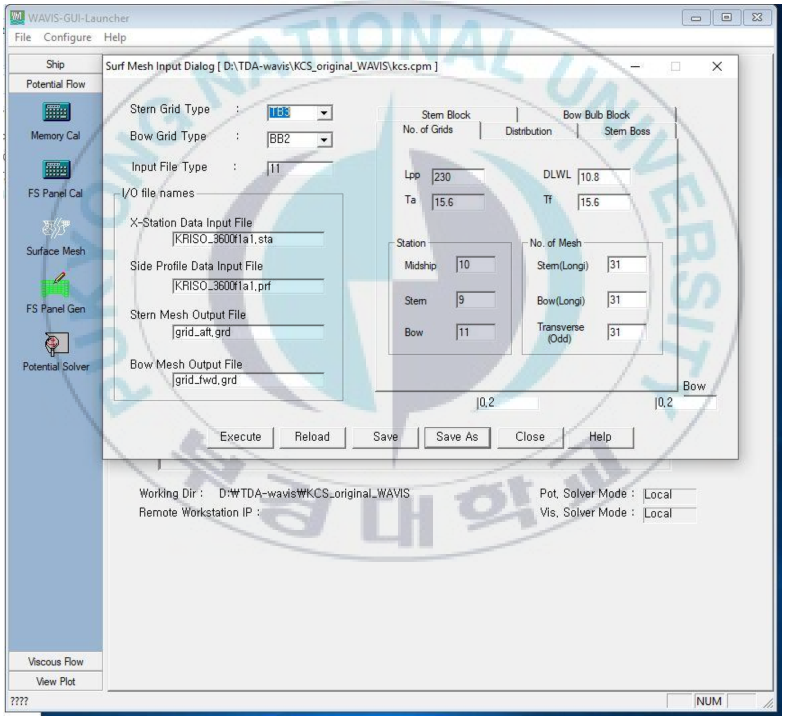


Fig. 5.5 Surface Mesh setting for KCS in WAVIS

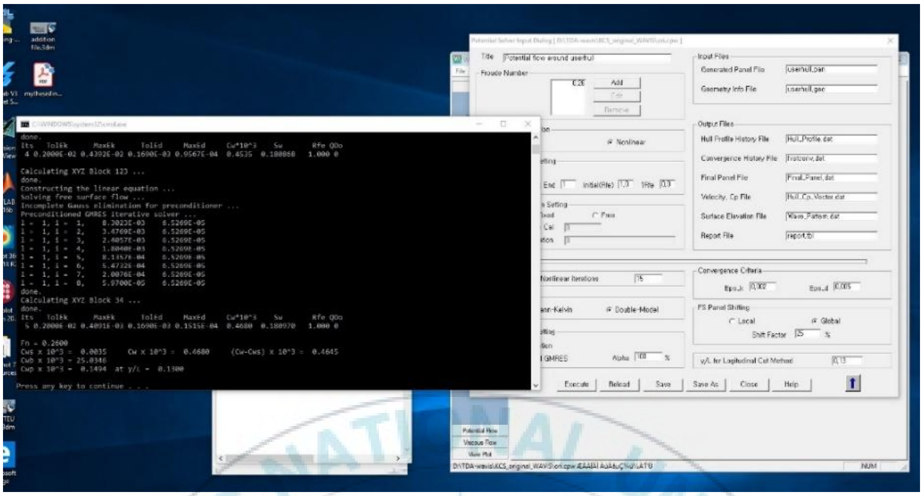


Fig. 5.6 Running in WAVIS

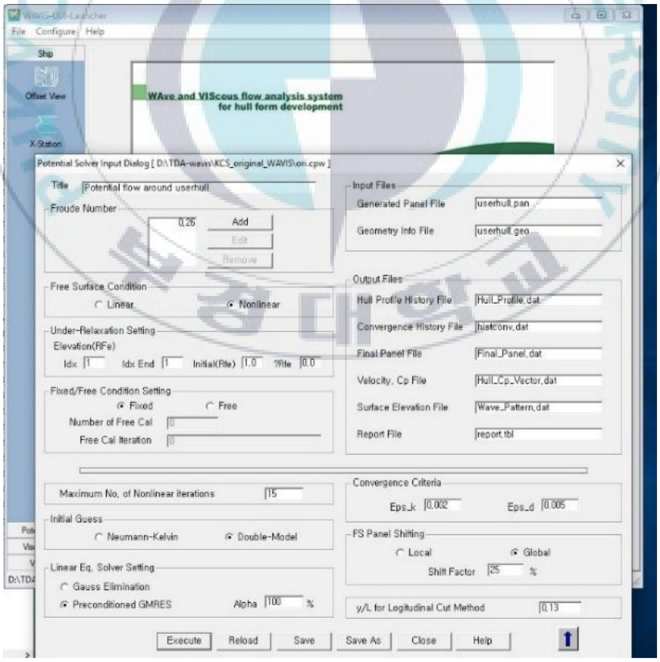


Fig. 5.7 Potential flow setting in WAVIS

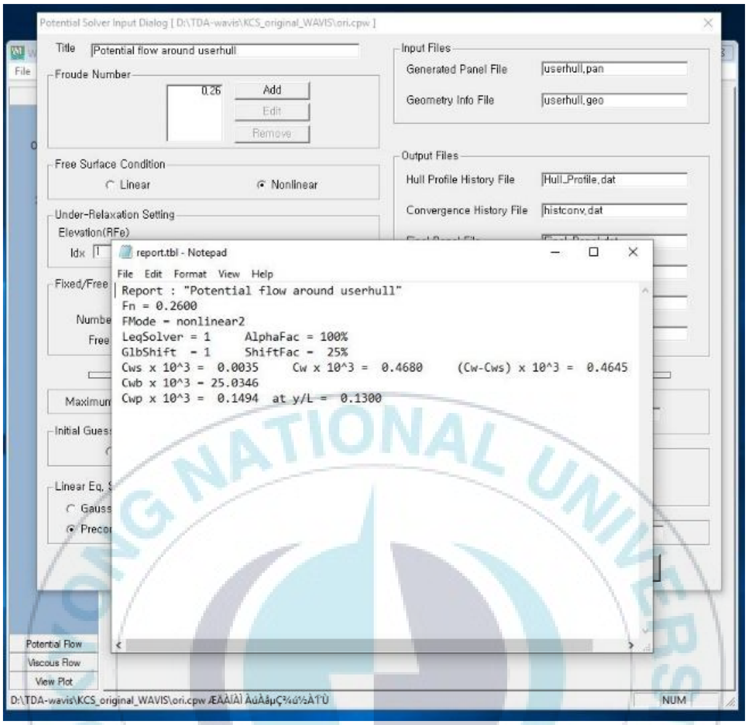


Fig. 5.8 Results from WAVIS

The KCS hull form calculation procedure of the integrated program is displayed in Fig. 5.9 ~ Fig. 5.10. Fig. 5.11 shows the result files management of the integrated program. The output file for KCS hull calculation is shown in Fig. 5.8 for both WAVIS and the integrated program developed in this study.

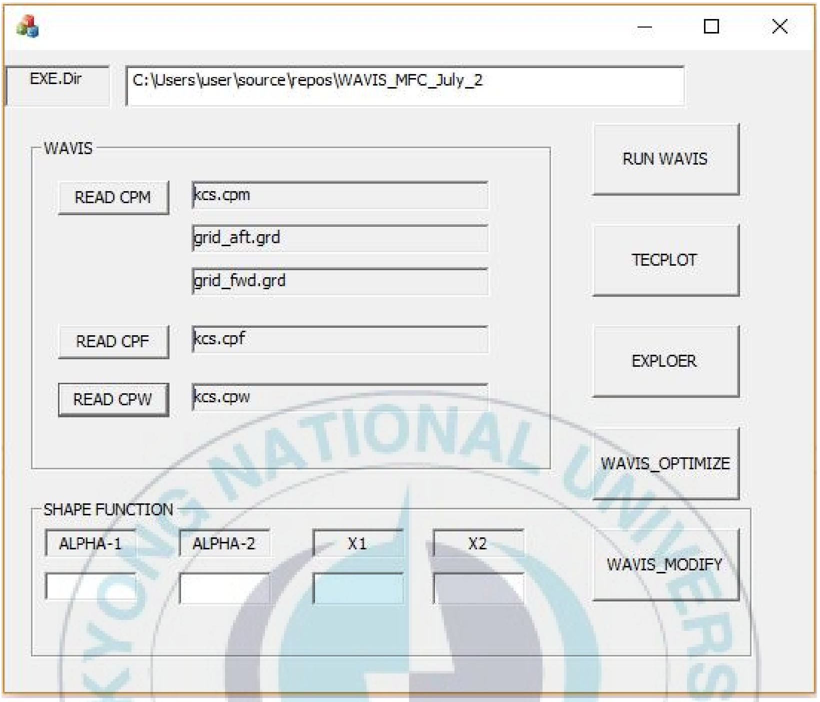


Fig. 5.9 Preparation for run WAVIS in the integrated program

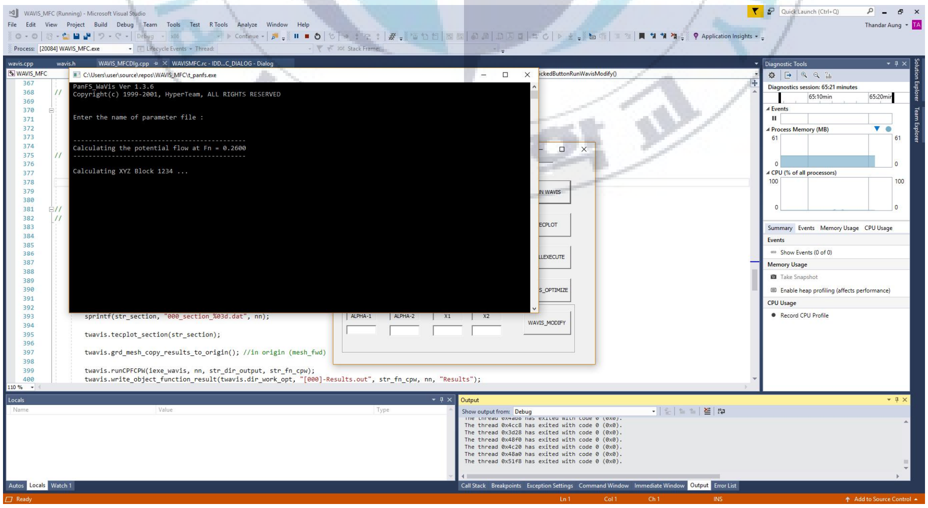


Fig. 5.10 Running the integrated program for WAVIS

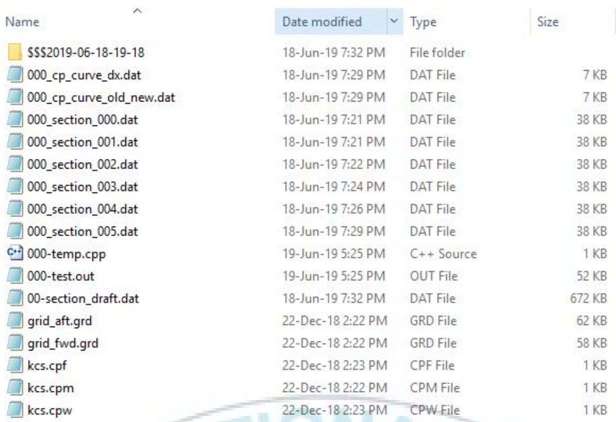


Fig. 5.11 Result files management of the integrated program

5.3 Optimization of the Integrated Program

In the integrated program developed in this study, SQP was applied in the optimization module to minimize C_w with two design variables α_1 and α_2 which are parameters of the Shift method.

In order to verify if the results of the integrated program achieved with SQP are acceptable, the author manually calculated the C_w of KCS ship hull as shown in Table 5.3. In this process, the author ran WAVIS 66 times to predict the range of the two design variables α_1 and α_2 for minimum C_w; the large number of calculation, the closer to the exact optimum value of C_w, requiring great quantities of time. However, since the SQP algorithm was introduced in the integrated program, the optimum result was obtained after 11 iterations with 60 function calls, taking an approximate of 120 minutes to complete the whole calculation. In SQP optimization, the initial prediction values of α_1 and α_2 was 0.0 and 15 respectively. Since the number of iterations and function calls of WAVIS depends on the initial value prediction of design variables, the better the prediction, the less the number of iteration and function calls required.

When calculating manually, the position of optimum wave making resistance value was suggested around the value of α_2, setting it between stations 17 and 18 with an α_1 value of -0.2. In the case of finding the optimum of C_w with SQP, results matched with the manual calculations. The optimum C_w was 0.4261 and the design variables α_1 and α_2 were -0.1924 and 17.8429 respectively, reducing C_w by 10% compared to its original value of 0.4758. The procedure of finding optimum C_w with SQP is shown in Table 5.4.

To check whether or not SQP was able to find the optimum value of C_w with any initial prediction of design variables, the author ran the integrated program with different initial predictors, being able to find the optimum results with different number of function call and iterations in every case. This shows that the integrated program developed by the author works properly.

As shown in Fig. 5.12, through the SQP, the value of wave making resistance gradually decreases, finding the optimum values of C_w. As a summary, Fig. 5.13 ~ Fig. 5.16 display the comparison of original and optimum hull form found by the integrated program developed in this study.

Table 5.3 Optimum finding via manual calculation

α2 \ α1	-0.5	-0.4	-0.3	-0.2	-0.1	0.0	0.1	0.2	0.3	0.4	0.5
13	0.6569	0.5919	0.5470	0.5093	0.4852	0.4748	0.4778	0.5000	0.5297	0.5738	0.6326
14	0.6721	0.5971	0.5401	0.5032	0.4798	0.4747	0.4887	0.5285	0.5820	0.6567	0.7535
15	0.6784	0.5945	0.5305	0.4911	0.4708	0.4748	0.5049	0.5692	0.6561	0.7734	0.9238
16	0.6453	0.5318	0.5001	0.4661	0.4557	0.4748	0.5266	0.6146	0.7414	0.9101	1.1232
17	0.5787	0.5079	0.4622	0.4376	0.4403	0.4748	0.5459	0.6596	0.8187	1.0387	1.3278
18	0.5297	0.4724	0.4399	0.4279	0.4382	0.4748	0.5446	0.6531	0.8276	1.0978	1.5778

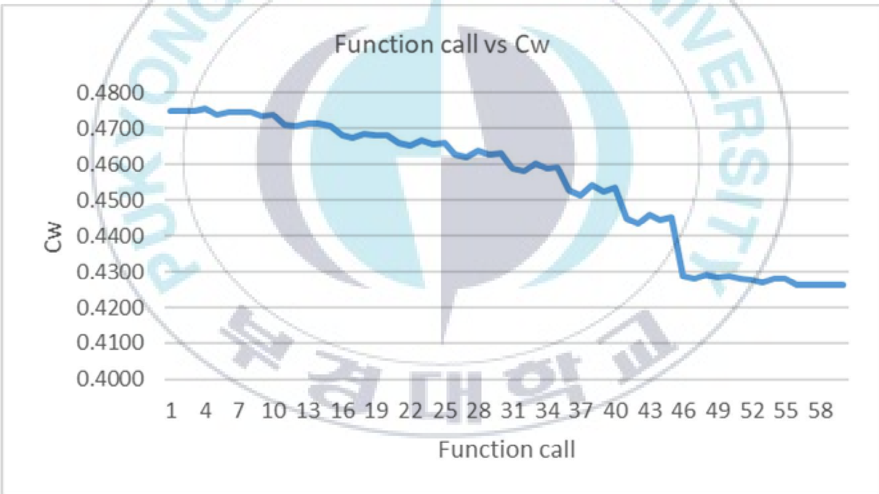


Fig. 5.12 Results of C_w via SQP

Table 5.4 SQP optimization of integrated program

No.Function Call	α2	α1	Cw
1	15.0000	0.0000	0.4748
2	15.0500	0.0000	0.4748
3	14.9500	0.0000	0.4748
4	15.0000	0.0050	0.4757
5	15.0000	-0.0050	0.4740
6	15.0000	-0.0017	0.4745
7	15.0500	-0.0017	0.4745
8	14.9500	-0.0170	0.4745
9	15.0000	0.0033	0.4735
10	15.0000	-0.0067	0.4737
11	15.0000	-0.0289	0.4710
12	15.0500	-0.0289	0.4708
13	14.9500	-0.0289	0.4712
14	15.0000	-0.0239	0.4715
15	15.0000	-0.0339	0.4706
16	15.1414	-0.0720	0.4680
17	15.1914	-0.0720	0.4675
18	15.0914	-0.0720	0.4685
19	15.1414	-0.0670	0.4680
20	15.1414	-0.0770	0.4681
21	15.3574	-0.0948	0.4659
22	15.4074	-0.0948	0.4652
23	15.3074	-0.0948	0.4666
24	15.3574	-0.0898	0.4658
25	15.3574	-0.0998	0.4661
26	15.6137	-0.1170	0.4629
27	15.6637	-0.1170	0.4621
28	15.5636	-0.1170	0.4638
29	15.6137	-0.1120	0.4627
30	15.6137	-0.1220	0.4633
31	15.8569	-0.1298	0.4590
32	15.9069	-0.1298	0.4580
33	15.8069	-0.1298	0.4601
34	15.8569	-0.1248	0.4588
35	15.8568	-0.1348	0.4593
36	16.2269	-0.1602	0.4527

37	16.2769	-0.1602	0.4514
38	16.1769	-0.1602	0.4541
39	16.2269	-0.1552	0.4523
40	16.2269	-0.1652	0.4533
41	16.5996	-0.1777	0.4448
42	16.6496	-0.1777	0.4435
43	16.5496	-0.1777	0.4461
44	16.5996	-0.1727	0.4444
45	16.5996	-0.1827	0.4454
46	17.5806	-0.1971	0.4286
47	17.6306	-0.1971	0.4279
48	17.5306	-0.1971	0.4293
49	17.5806	-0.1921	0.4284
50	17.5806	-0.2021	0.4288
51	18.0000	-0.1989	0.4279
52	18.0500	-0.1989	0.4276
53	17.9500	-0.1989	0.4269
54	18.0000	-0.1939	0.4279
55	18.0000	-0.2039	0.4280
56	17.8429	-0.1874	0.4261
57	17.8929	-0.1874	0.4263
58	17.7929	-0.1874	0.4262
59	17.8429	-0.1824	0.4262
60	17.8429	-0.1924	0.4261

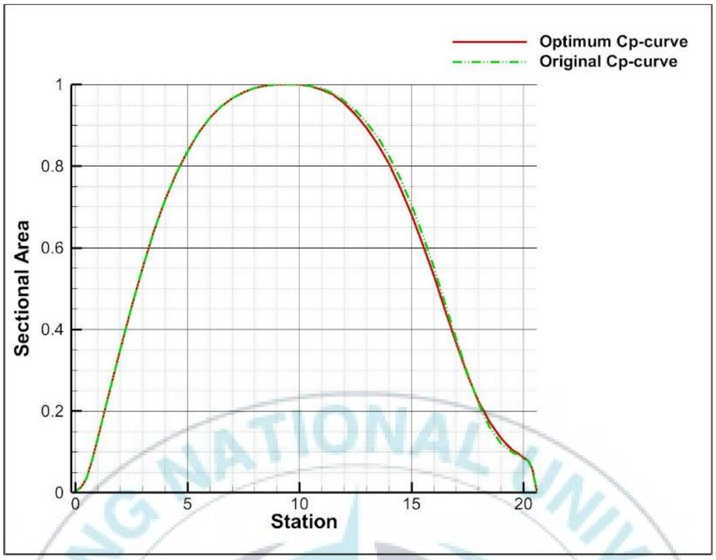


Fig. 5.13 Original and optimum Cp-curve

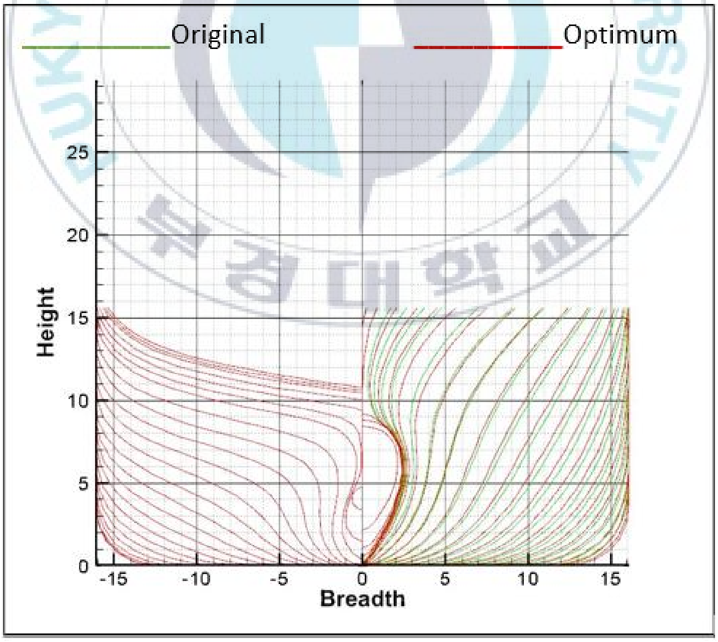


Fig. 5.14 Original (green) and optimum (red) section

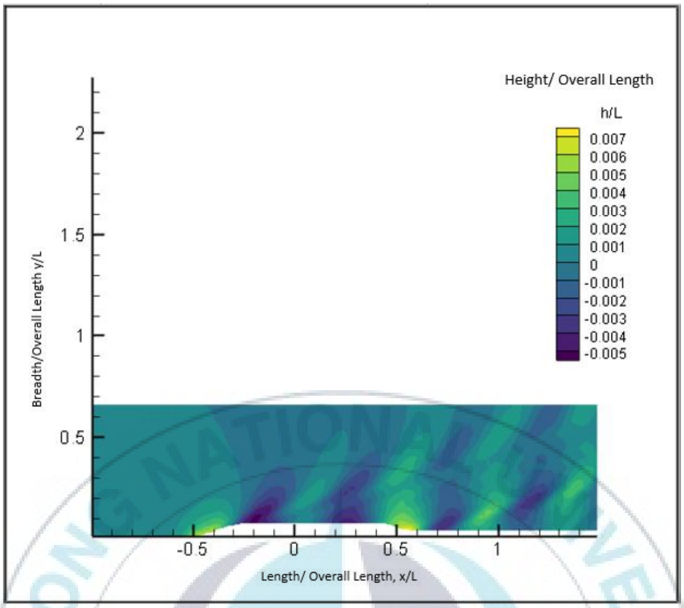


Fig. 5.15 Wave-pattern of original hull form

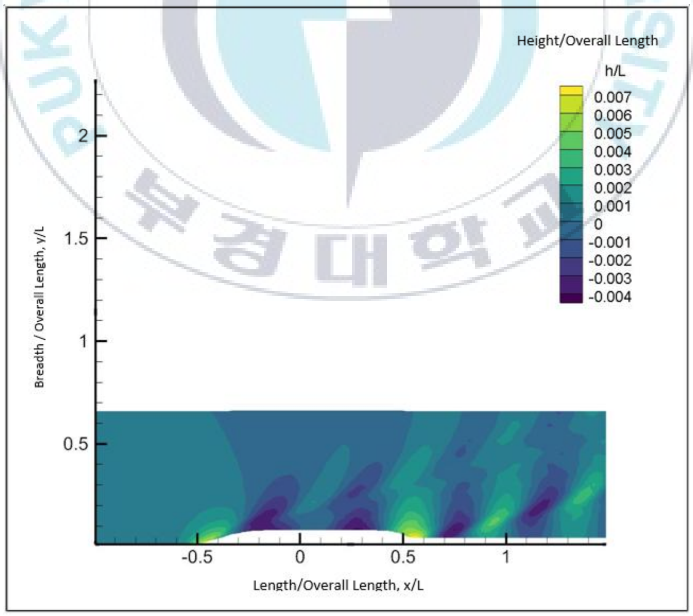


Fig. 5.16 Wave-pattern of optimum hull form

6. Conclusion and Further Study

6.1 Conclusion

The intention of this paper was to develop an integrated program of CAD, CFD and optimization modules to improve computational efficiency without requiring additional systems for the ship building industry. In this regard, already available systems such as WAVIS are used for calculating potential flow, NUBS and Shift method for modification of the geometry of ship hull. After developing and running the program, we can conclude the following:

1) This integrated program combined all three modules needed for optimization process; saving time in connecting modules and resulting more effective for the optimization process.

2) Since this program is constructed using visual C++, other modification and optimization techniques can be implemented to this integrated program to increase its function. Additional methods such as Free Form Deformation (FFD) and Genetic algorithms (GA) can also be used depending on the needs of each designer.

3) The Shift method used for this study for modification of hull form is simple and practical for ship designers. However, shape function represents a disadvantage of the Shift method. In this study, sine function is used as the shape function having a sudden change

characteristic from zero to one, making the Cp-curve knuckles in this area. Even though there are numerous functions available for shape function to avoid the weakness of sine function, the Shift proved to be the ideal method for this paper because it is simplicity and practicality.

4) The number of design variables in this study was two, which is a relatively small number for the modification of ship hull form. Since the design variables only control the range of station to modify, the slope of the Cp-curve and fixed station, the area of the shape function was not under control. Because the area of Cp-curve was not considered as a constraint, the displacement of the ship will be different in all modification.

5) Since SQP is applied for optimization, the optimum value of $\boldsymbol{C_w}$ was found after total of 10 iterations, 60 times function call and within 120 minutes. Since the number of iterations depends on the initial prediction value of design variables, the initial prediction is important. The author used two different initial predictions as extreme values away from the optimum value to verify whether or not SQP was able to find the ideal value of $\boldsymbol{C_w}$ under these conditions obtaining the desired results. In this regard, it can be concluded that the better initial prediction, the number of required iterations and time can be reduced.

6) The author verified the SQP results with the prediction of manual calculations. Since the expected optimum results and the optimum results through the SQP resulted to be the same, the author can

conclude that the integrated program developed in this study was able to properly find the optimum value of $\boldsymbol{C_w}$.

6.2 Further Study

This study leaves several areas of opportunity that can be considered for future research. First, the author believes that the integrated program developed in this study can include additional modification and optimization methods in order to increase the scope and functionality of the program. Because of this, future programmers can add supplementary modification methods in order to obtain better results. Similarly, for the optimization module of the program, future programmers can focus on additional options such as Genetic Algorithm (GA) and Ant Colony Algorithm for better results in the optimization process. Since the ship hull form is key for a successful ship hull design process, the more effective the modification methods, the more successful the ship design system. If the program presented in this dissertation continues its development by adding modification and optimization methods, the challenging task that represents finding the optimum hull form with hydrodynamic performances can become easier, more effective and less time consuming, reducing monetary costs and giving the opportunity to naval architects to focus on other aspects of the ship design process.

References

Abt, C., & Harries, S. (2007). A New Approach to Integration of CAD and CFD for Naval Architects. *6th COMPIT, Contona*, (i), 467–479.

Abt, Claus, & Harries, S. (2007a). Friendship-Framework - Integrating ship-design modelling, simulation, and optimisation. *Naval Architect*, (JAN.), 36–37.

Abt, Claus, & Harries, S. (2007b). Hull Variation and Improvement using the Generalized Lackenby Method. *The Naval Architect*.

Bagheri, H., & Ghassemi, H. (2014). Genetic algorithm applied to optimization of the ship hull form with respect to seakeeping performance. *Transactions of Famena*, *38*(3), 45–58.

Bardis, L., & Vafiadou, M. (1992). Ship-hull geometry representation with B-spline surface patches. *Computer-Aided Design*, *24*(4), 217–222. https://doi.org/10.1016/0010-4485(92)90058-I

Byoung K.choi. (1991). *Surface Modeling for CAD/CAM*.

Campana, E. F., Peri, D., & Rossetti, M. (2001). Design optimization of ship hulls via CFD techniques. *Journal of Ship Research*, (June 2001). Retrieved from http://www.ingentaconnect.com/content/sname/jsr/2001/00000045/00000002/art00006

Campana, E. F., Peri, D., Tahara, Y., & Stern, F. (2006). Shape optimization in ship hydrodynamics using computational fluid dynamics. *Computer Methods in Applied Mechanics and Engineering*, *196*(1–3), 634–651. https://doi.org/10.1016/j.cma.2006.06.003

Cheng, X., Feng, B., Liu, Z., & Chang, H. (2018). Hull surface modification for ship resistance performance optimization based on Delaunay triangulation. *Ocean Engineering*, *153*(August 2017), 333–344. https://doi.org/10.1016/j.oceaneng.2018.01.109

Choi, H. J. (2015). Hull-form optimization of a container ship based on bell-shaped modification function. *International Journal of Naval Architecture and Ocean Engineering*, *7*(3), 478–489. https://doi.org/10.1515/ijnaoe-2015-0034

Contents, B. T. O., Harries, S., Abt, C., & Systems, F. (2008). Back To Contents Tight and Loose Coupling of Cad and Cfd – Methods , Practical Aspects , Advantages. *Framework*, (March), 1–6.

Harries, S., Abt, C., & Hochkirch, K. (2004). Modeling meets Simulation-Process Integration to improve Design. *Honorary Colloquium for Prof. Hagen, Prof. Schlüter and Prof. Thiel*, 1–12.

Harries, Stefan, & Systems, F. (2014). *Practical Shape*. *2014*(November), 1–25.

Horn, B. K. P. (1983). The Curve of Least Energy. *ACM Transactions on Mathematical Software*, *9*(4), 441–460. https://doi.org/10.1145/356056.356061

Huang, F., & Yang, C. (2016). Hull form optimization of a cargo ship for reduced drag. *Journal of Hydrodynamics*, *28*(2), 173–183. https://doi.org/10.1016/S1001-6058(16)60619-4

Hutchison, B. L., & Hochkirch, K. (2007). CFD Hull Form Optimization of a 12 , 000 cu . yd . (9175 m 3) Dredge. *Constraints*, (September).

Interface, G. U. (n.d.). *Calculation of wave resistance and potential flow*. 31–32.

Kim, H.-C. (2013). On the Volumetric Balanced Variation of Ship Forms. *Journal of Ocean Engineering and Technology*, *27*(2), 1–7. https://doi.org/10.5574/ksoe.2013.27.2.001

Kim, H.-J., Choi, H.-J., & Chun, H.-H. (2009). A Practical Hull Form Optimization Method Using the Parametric Modification Function. *Journal of the Society of Naval Architects of Korea*, Vol. 44, pp. 542–550. https://doi.org/10.3744/snak.2007.44.5.542

Kim, H.-J., & Jeong, H. H. (2000). Hull Form Generation of Minimum Wave Resistance by a Nonlinear Optimization Method. 대한조선학회 논문집.

Kim, H. ., & Chun, H. . (2000). Hull form generation of minimum wave resistance by a nonlinear optimization method. *Journal of the Society of Naval Architects of Korea*.

Kim, H., & Yang, C. (2010). A new surface modification approach for CFD-based hull form optimization. *Journal of Hydrodynamics*, *22*(5 SUPPL.

1), 520–525. https://doi.org/10.1016/S1001-6058(09)60246-8

Kostas, K. V., Ginnis, A. I., Politis, C. G., & Kaklis, P. D. (2015). Ship-hull shape optimization with a T-spline based BEM-isogeometric solver. *Computer Methods in Applied Mechanics and Engineering*, *284*, 611–622. https://doi.org/10.1016/j.cma.2014.10.030

Markov, N. E., & Suzuki, K. (2011). Fundamental Studies on Rankine Source Panel Method Fully Based on B-splines. *Journal of the Society of Naval Architects of Japan*, *2000*(187), 13–23. https://doi.org/10.2534/jjasnaoe1968.2000.13

Marques Sanches, F., Ventura, M., & Guedes Soares Supervisor, C. (2016). *Parametric Modelling of Hull Form for Ship Optimization Naval Architecture and Marine Engineering Examination Committee*. (June).

Nam, J. H., & Bang, N. S. (2017). A curve based hull form variation with geometric constraints of area and centroid. *Ocean Engineering*, *133*(April 2016), 1–8. https://doi.org/10.1016/j.oceaneng.2017.01.031

Park, J. H., Choi, J. E., & Chun, H. H. (2015). Hull-Form optimization of KSUEZMAX to enhance resistance performance. *International Journal of Naval Architecture and Ocean Engineering*, *7*(1), 100–114. https://doi.org/10.1515/ijnaoe-2015-0008

Pérez, F., & Suárez, J. A. (2007). Quasi-developable B-spline surfaces in ship hull design. *CAD Computer Aided Design*, *39*(10), 853–862. https://doi.org/10.1016/j.cad.2007.04.004

Priftis, A., Boulougouris, E., Turan, O., & Papanikolaou, A. (2018). Parametric design and multi-objective optimisation of containerships. *Ocean Engineering*, *156*(October 2017), 347–357. https://doi.org/10.1016/j.oceaneng.2018.02.062

Rodríguez, A., & Fernández-Jambrina, L. (2012). Programmed design of ship forms. *CAD Computer Aided Design*, *44*(7), 687–696. https://doi.org/10.1016/j.cad.2012.03.003

Singiressu S.Rao. (n.d.). *Engineering Optimization Theory and practice(third edition)*.

Sun, J., Lv, X., Liu, W., Ning, H., & Chen, X. (2012). Research on a method of hull form design based on wave-making resistance optimization.

Polish Maritime Research, *19*(3), 16–25. https://doi.org/10.2478/v10012-012-0027-1

Tahara, Y., Kobayashi, H., Kandasamy, M., He, W., Peri, D., Diez, M., … Maritime, N. (2012). *CFD-Based Multiobjective Stochastic Optimization of a Waterjet Propelled High Speed Ship*. (August), 26–31.

Tahara, Y., Tohyama, S., & Katsui, T. (2006). CFD-based multi-objective optimization method for ship design. *International Journal for Numerical Methods in Fluids*, *52*(5), 499–527. https://doi.org/10.1002/fld.1178

Wu, J., Liu, X., Zhao, M., & Wan, D. (2017). Neumann-Michell theory-based multi-objective optimization of hull form for a naval surface combatant. *Applied Ocean Research*, *63*, 129–141. https://doi.org/10.1016/j.apor.2017.01.007

Yu, J. W., Lee, C. M., Lee, I., & Choi, J. E. (2017). Bow hull-form optimization in waves of a 66,000 DWT bulk carrier. *International Journal of Naval Architecture and Ocean Engineering*, *9*(5), 499–508. https://doi.org/10.1016/j.ijnaoe.2017.01.006

ZHANG, P., ZHU, D. xiang, & LENG, W. hao. (2008). Parametric approach to design of hull forms. *Journal of Hydrodynamics*, *20*(6), 804–810. https://doi.org/10.1016/S1001-6058(09)60019-6

Shift 법와 WAVIS 를 이용한 선채 수정

Thandar Aung

부경대학교조선해양시스템공학과

국문요약

최적화 과정을 거친 선박이 더 많은 에너지를 절약하고 경쟁력을 갖기 때문에 전산 유체 역학(CFD)에 기반한 최적의 선박 설계에 대한 요구가 증가하고 있다. 이러한 최적화 과정에서는 선형을 몇 개의 파라미터로 표시할 수 있는 기법이 필요하다. 즉, 최적화 루틴에서 변화된 파라미터로 새로운 선형이 생성되어야 하고 이를 CFD로 계산하여 좀더 나은 목적함수를 찾아가는 과정을 반복해야 한다. 일반적으로 최적의 선박 설계를 찾는 것은 CAD / CAM 및 수정 모듈, CFD 모듈 및 최적화 모듈의 세 가지 모듈로 구성된 프로세스이다. CAD / CAM 모듈은 선체의 형상 정의를 생성한다. 수정 모듈은 최적화 과정의 필요한 조건에 따라 선박 선체를 수정한다. CFD 모듈은 흐름과 선체의 유체 역학 성능을 계산한다. 최적화 과정에서 합리적인 기하학 구조를 얻으려면 효율적이고 효과적인 수정 방법이 중요한 역할을한다. 본 논문은 선형을 수정 기법으로Shift Method를 이용하여 단면적 곡선,Cp 곡선을 변환 시키는 방법을 적용하고 있다. 선박 선체 형태를 표현할 때, Cp 곡선은 선체의 기하학에 영향을 미치는 가장 중요한 측면 중 하나이다. 본 연구의 저자는 이 방법이 간단하고 효과적이기 때문에 선박 선체 형태의 수정 방법으로 선택했다. 그리고 CFD 기법으로는 WAVIS(version 1.3)를 사용하고, 최적화 기법으로는 SQP(Sequential Quadratic Programming)법을 사용하고 있다. 연속적인 최적화 프로세스에 도달하려면 앞서 언급 한 모듈을 연결하여 계산 효율성을 달성해야한다. 본 논문의 의의로서는 이러한 선형 변환 분야(CAD), CFD, 최적화 기법인 SQP를 통합하는 프로그램을 C++ 언어로 작성하였다는 것이다. 따라서 세부분 각각을 다른 프로그램으로 교체할 수 있는 능력이 생기게 되고, 이를 이용하여 계속 발전된 결과를 얻을 수 있다.

키워드: WAVIS, Shift 법, 프로그래밍 언어 C++, 곡선 모델링, 최적화, SQP (Sequential Quadratic Programming)

Acknowledgement

First of all, I would like to express my sincere gratitude to my academic advisor, Dong Joon Kim for his kindness, help, advices and his thoughtful consideration during my two-year master's degree studies in Korea. This would not have been possible without his guidance and help.

I am also grateful with Professors Yong Jig Kim and Jong Yong Park for supervising and providing guidance during my thesis defense. I also would like to recognize all Professors in the Naval Architecture and Marine Systems Engineering department for their great lectures and all the administrative staff for their help and assistance.

I also would like to thank to Ph.D Yoon and Senior Seung-Woo Park for their support during the development of the integrated program for this dissertation. I owe my gratitude to all my lab mates, Seung-Woo Shin, Yeon-Hee Song, Samuel, Zhang Yong Xing who supported me and friendly guided me during the difficult times. I am glad to have such smart, kind and supportive people as my friends and to be with you during my whole master's degree in Korea.

Additionally, I would like to thank the Korean Government Scholarship Program (KGSP) for providing me with this opportunity to study in this country. Thanks to the KGSP, I was able to achieve this important goal in my life.

Finally, I want to thank my parents and my two sisters who fully support and believed in me. Because of them, I could accomplish my dreams since I was young. I am really glad to have met Mrs. Su Kyung Choi for becoming my guide in this country and always having time for me during the difficult times.

APPENDIXES

APPENDIX A: The surface mesh aft of KCS hull.
APPENDIX B: The surface mesh foreward of KCS hull.
APPENDIX C: Tridiagonal matrix solution.
APPENDIX D: code for modification, WAVIS run and optimization module.

.

APPENDIX A

The surface mesh aft of KCS hull. Since the data is too big for showing all in this dissertation only the sample of surface mesh is shown in here.

grid_aft.grd - Notepad

File Edit Format View Help

```
TB3
  21   11
 0.46724907E+00  0.69906302E-01  0.20869566E-01
 0.46724907E+00  0.69431439E-01  0.18328214E-01
 0.46724907E+00  0.68452843E-01  0.15658423E-01
 0.46724907E+00  0.66867068E-01  0.12962818E-01
 0.46724907E+00  0.64599723E-01  0.10375002E-01
 0.46724907E+00  0.61654411E-01  0.79979477E-02
 0.46724907E+00  0.58058884E-01  0.58935955E-02
 0.46724907E+00  0.53870920E-01  0.40283492E-02
 0.46724907E+00  0.49110927E-01  0.23602527E-02
 0.46724907E+00  0.43771483E-01  0.84947090E-03
 0.46724907E+00  0.37841916E-01 -0.60039602E-03
 0.46724907E+00  0.31870123E-01 -0.18650885E-02
 0.46724907E+00  0.26413038E-01 -0.28733213E-02
 0.46724907E+00  0.21439143E-01 -0.37174888E-02
 0.46724907E+00  0.16914720E-01 -0.44689262E-02
 0.46724907E+00  0.12807503E-01 -0.51864334E-02
 0.46724907E+00  0.90867626E-02 -0.59092646E-02
 0.46724907E+00  0.57325689E-02 -0.66957641E-02
 0.46724907E+00  0.28188720E-02 -0.78275641E-02
 0.46724907E+00  0.74701163E-03 -0.97126467E-02
 0.46724907E+00  0.00000000E+00 -0.12173913E-01
 0.47206739E+00  0.69761492E-01  0.20869566E-01
 0.47206783E+00  0.69199406E-01  0.18453270E-01
 0.47206768E+00  0.68154395E-01  0.15933974E-01
 0.47206751E+00  0.66544145E-01  0.13401451E-01
 0.47206783E+00  0.64330079E-01  0.10951644E-01
 0.47206786E+00  0.61480742E-01  0.86972732E-02
 0.47206801E+00  0.58021765E-01  0.66931229E-02
 0.47206774E+00  0.53998370E-01  0.49133343E-02
 0.47206771E+00  0.49437411E-01  0.32928302E-02
 0.47206777E+00  0.44322643E-01  0.18124746E-02
 0.47206789E+00  0.38633790E-01  0.41667276E-03
 0.47206780E+00  0.32909524E-01 -0.82607270E-03
 0.47206786E+00  0.27677786E-01 -0.18187565E-02
 0.47206771E+00  0.22904633E-01 -0.26267425E-02
 0.47206786E+00  0.18558174E-01 -0.33170949E-02
 0.47206762E+00  0.14605037E-01 -0.39334507E-02
 0.47206771E+00  0.11014709E-01 -0.45148931E-02
 0.47206771E+00  0.77561168E-02 -0.50752223E-02
 0.47206777E+00  0.48058527E-02 -0.56474106E-02
 0.47206786E+00  0.21786741E-02 -0.63828263E-02
 0.47206783E+00  0.00000000E+00 -0.75579477E-02
```

APPENDIX B

The surface mesh foreward of KCS hull. Since the data is too big for showing all in this dissertation only the sample of surface mesh is shown in here.

grid_fwd.grd - Notepad

File Edit Format View Help

```
BB2
   21    9
-0.52738827E+00   0.00000000E+00  -0.13360436E-01
-0.52807534E+00   0.00000000E+00  -0.14047878E-01
-0.52867240E+00   0.00000000E+00  -0.14825011E-01
-0.52917516E+00   0.00000000E+00  -0.15677117E-01
-0.52958727E+00   0.00000000E+00  -0.16587870E-01
-0.52991360E+00   0.00000000E+00  -0.17543782E-01
-0.53015822E+00   0.00000000E+00  -0.18534595E-01
-0.53032494E+00   0.00000000E+00  -0.19552156E-01
-0.53041828E+00   0.00000000E+00  -0.20589679E-01
-0.53044254E+00   0.00000000E+00  -0.21641711E-01
-0.53039706E+00   0.00000000E+00  -0.22703692E-01
-0.53027952E+00   0.00000000E+00  -0.23760101E-01
-0.53009623E+00   0.00000000E+00  -0.24796277E-01
-0.52984893E+00   0.00000000E+00  -0.25808193E-01
-0.52954161E+00   0.00000000E+00  -0.26792595E-01
-0.52917844E+00   0.00000000E+00  -0.27746800E-01
-0.52876323E+00   0.00000000E+00  -0.28668391E-01
-0.52829897E+00   0.00000000E+00  -0.29554959E-01
-0.52778924E+00   0.00000000E+00  -0.30404825E-01
-0.52723944E+00   0.00000000E+00  -0.31217754E-01
-0.52665567E+00   0.00000000E+00  -0.31994782E-01
-0.52442521E+00   0.00000000E+00  -0.11566987E-01
-0.52554923E+00   0.12913040E-02  -0.12319627E-01
-0.52641189E+00   0.23519942E-02  -0.13344389E-01
-0.52720338E+00   0.30129191E-02  -0.14486568E-01
-0.52787948E+00   0.34204461E-02  -0.15632512E-01
-0.52833372E+00   0.36376293E-02  -0.16746473E-01
-0.52861029E+00   0.37519331E-02  -0.17863572E-01
-0.52876335E+00   0.38085631E-02  -0.19007668E-01
-0.52882969E+00   0.38304445E-02  -0.20173853E-01
-0.52884167E+00   0.38368555E-02  -0.21357268E-01
-0.52878195E+00   0.38239849E-02  -0.22546723E-01
-0.52866381E+00   0.38011190E-02  -0.23729360E-01
-0.52847576E+00   0.37610342E-02  -0.24887448E-01
-0.52822030E+00   0.37058853E-02  -0.26020778E-01
-0.52788919E+00   0.36291438E-02  -0.27131164E-01
-0.52747703E+00   0.35210683E-02  -0.28232301E-01
-0.52696323E+00   0.33462443E-02  -0.29346006E-01
-0.52632016E+00   0.30114679E-02  -0.30569298E-01
-0.52562791E+00   0.23474321E-02  -0.31972505E-01
-0.52486634E+00   0.13367181E-02  -0.33521187E-01
-0.52417988E+00   0.00000000E+00  -0.34623194E-01
```

APPENDIX C

Tridiagonal matrix solution

Presented in this appendix is an approach to solve a linear equation system whose coefficient matrix is "tridiagonal".

Let's consider a system of "n+1" linear equations of the form

$$Ax = d, \qquad \text{(A.1)}$$

where,

$$A = \begin{bmatrix} b_0 & c_0 & 0 & . & . & . & . & . \\ a_1 & b_1 & c_1 & 0 & . & . & . & . \\ 0 & a_2 & b_2 & c_2 & 0 & . & . & . \\ . & 0 & . & . & . & . & . & . \\ . & . & . & . & . & . & . & 0 \\ . & . & . & . & 0 & a_{n-1} & b_{n-1} & c_{n-1} \\ . & . & . & . & . & 0 & a_n & b_n \end{bmatrix} \qquad \text{(A.2)}$$

$$x = [x_0\, x_1\, x_2 \dots\dots x_{n-1} x_n]^T,$$

$$d = [d_0\, d_1\, d_2 \dots\dots d_{n-1} d_n]^T$$

In order to meet the "diagonal dominance" condition, we need to have

$$|b_i| \geq |c_{i-1}| + |a_{i+1}| \qquad \text{(A.3)}$$

with the inequality holding for at least one "i".

The coefficient matrix ***A*** can be decomposed into a product of two *"bidiagonal"* matrices ***L*** and ***U*** as follows:

$$A = L\,U \qquad \text{(A.4)}$$

Then, the solution to (A.1) can be obtained by solving the two linear equation systems (A.5) and (A.6). That is, first solve the following for ***y*** and ***x*** respectively.

$$Ly = d \qquad \text{(A.5)}$$

$$U\,x = y \tag{A.6}$$

Now, let the bidiagonal matrices ***L***, ***U*** have the following forms:

$$L = \begin{bmatrix} \beta_0 & . & . & . & . & . & . & . \\ \alpha_1 & \beta_1 & . & . & . & . & . & . \\ . & \alpha_2 & \beta_2 & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & \alpha_{n-1} & \beta_{n-1} & . \\ . & . & . & . & . & . & \alpha_n & \beta_n \end{bmatrix} \tag{A.7}$$

$$U = \begin{bmatrix} 1 & \gamma_1 & . & . & . & . & . \\ . & 1 & \gamma_2 & . & . & . & . \\ . & . & 1 & \gamma_3 & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & 1 & \gamma_n \\ . & . & . & . & . & . & 1 \end{bmatrix} \tag{A.8}$$

On multiplying ***L*** and ***U*** and then equating the result with ***A***, the elements of the bidiagonal matrices are obtained as follows:

$$\alpha_i = c_i \qquad \text{for i = 1,…, n} \tag{A.9}$$

$$\gamma_{i+1} = {}^{c_i}/_{\beta_i} \; ; \qquad \beta_{i+1} = b_{i+1} - \alpha_{i+1}\gamma_{i+1} \quad \text{for i = 0,…, n-1} \tag{A.10}$$

with $\beta_0 = b_0$

Now we are ready to solve (A.5) and (A.6). The solution to (A.5) is obtained after a "forward substitution" pass (with $y_0 = {}^{d_0}/_{\beta_0}$)

$$y_i = (d_i - \alpha_i y_{i-1})/\beta_i \qquad \text{for i = 1,…, n} \tag{A.11}$$

By combining *(A.11)* and *(A.10)* to eliminate α_i, βi,

$\gamma_{i+1} = c_i / (b_i - \alpha_i \gamma_i)$; $y_i = (d_i - \alpha_i y_{i-1})/(b_i - \alpha_i \gamma_i)$ for i = 1,…., n (A.12)

with $\gamma_1 = {}^{c_0}/_{b_0}$ and $y_0 = {}^{d_0}/_{b_0}$

(A.6) is similarly solved by a "backward substitution" pass (with $x_n = y_n$):

$x_i = (y_i - \gamma_{i+1} x_{i+1})$ for i = n-1,….0. (A.13)

Thus, the solution to (A.1) is obtained from (A.12) and (A.13).

APPENDIX D

This appendix shows the code for modification, WAVIS run and optimization module.

Code for modification

```
void CWAVISMFCDlg::OnBnClickedButtonRunWavisModify()
{
	int ncp;
	int ncp2;
	float cp[maxcp][2], cp2[maxcp][2],g[maxcp][2];
	float dx[maxcp][2];
	char str_dir_output[maxstr];
	char str_fn_cpw[maxstr];
	int nn;
	int iexe_wavis;

	tprint("OnBnClickedButtonRunWavisModify");

	iexe_wavis = 1; //executed

	twavis.getNewOptFolderName(str_dir_output);

	twavis.grd_mesh_backup_origin(); //in backup

//------------------------------------------------------------------------------------/
// 0: calculate origin
//------------------------------------------------------------------------------------/
	nn = 0;
	twavis.runCPFCPW(iexe_wavis, nn, str_dir_output, str_fn_cpw);
	twavis.write_object_function_result(twavis.dir_work_opt,            "[000]-
Results.out", str_fn_cpw, nn, "Results");

//------------------------------------------------------------------------------------/
//1: calculate
//------------------------------------------------------------------------------------/
	twavis.grd_mesh_restore_origin();
	twavis.get_grd_cp_curve(&ncp, cp);
	nn = 1;
	twavis.grd_mesh_copy_origin_to_results();
	float x;
	float x1, x2, a1, a2;

	x1 = 10.0f;
```

```
            x2 = 20.0f;
            a1 = 0.5f;
            a2 = 15.0f;

            write_tecplot_2d("000_cp_curve_old_new.dat", "CP-Curve", "x y", ncp,
cp);

            twavis.get_shape_curveG(x1, x2, a1, a2, ncp, g);

            write_tecplot_2d("000_cp_curve_dx.dat", "CP-Curve-dx", "x y", ncp, g);

            twavis.tecplot_section("000_section_000.dat");

            char str_section[200];
            char str[maxstr];
            x = 15.0f;

            for (x = 13.0f; x <= 17.0 + 0.01; x += 1.0f){
                        x1 = 10.0f;
                        x2 = 20.0f;
                        a1 = 0.5f;
                        a2 = x;

                        twavis.get_shape_curveG(x1, x2, a1, a2, ncp, g);
                        twavis.modify_cp_with_Shape(ncp, cp, g, cp2);

                        add_tecplot_2d("000_cp_curve_old_new.dat", ncp, cp2);
                        add_tecplot_2d("000_cp_curve_dx.dat", ncp, g);

                        twavis.modify_GRD_cpcurve_fwd(ncp, cp2);
                        sprintf(str_section, "000_section_%03d.dat", nn);
                        twavis.tecplot_section(str_section);
                        twavis.grd_mesh_copy_results_to_origin();

                        twavis.runCPFCPW(iexe_wavis, nn, str_dir_output, str_fn_cpw);
                        twavis.write_object_function_result(twavis.dir_work_opt,
"[000]-Results.out", str_fn_cpw, nn, "Results");
                        nn++;
            }
}
```

Code for Running WAVIS

```
void CWAVISMFCDlg::OnBnClickedButtonRunWavis()
{
	char direxe[maxstr], strcpf[maxstr], strcpw[maxstr];

	GetDlgItem(IDC_STATIC_CPF)->GetWindowText(strcpf, maxstr);
	if (strlen(strcpf) != 0)
	{
		char strdir[maxstr];
		GetCurrentDirectory(maxstr, strdir);

		GetDlgItem(IDC_EDIT_DIR_EXE)->GetWindowTextA(direxe,
maxstr);
		strcat_s(direxe, "\\");
		strcat_s(direxe, "t_fsgen.exe");

		_spawnl(_P_NOWAIT, direxe, direxe, strcpf, NULL);

		Sleep(2 * 1000);

		keybd_event(0, 'Y', 0, 0);
		keybd_event('Y', 0, 0, 0);
		Sleep(1 * 1000);
		keybd_event('Y', 0, KEYEVENTF_KEYUP, 0);
		keybd_event(0, 'Y', KEYEVENTF_KEYUP, 0);
	}

	GetDlgItem(IDC_STATIC_CPW)->GetWindowText(strcpw, maxstr);
	if (strlen(strcpw) != 0)
	{
		GetDlgItem(IDC_EDIT_DIR_EXE)->GetWindowText(direxe,
maxstr);
		strcat_s(direxe, "\\");
		strcat_s(direxe, "t_panfs.exe");
		_spawnl(_P_WAIT, direxe, direxe, strcpw, NULL);
	}
}
```

Code for optimization

```
void CWAVISMFCDlg::OnBnClickedButtonRunWavisOptimize()
{
	int nf, nc;
	double x[10];
	double cl[10],cu[10];
	int mit;
	double xmax;
	double tol;
	double f;
	double gmax;

	float x1 = 10.0;
	float x2 = 20.0;

	int nn;
	int nsp;
	float cp[maxcp][2], g[maxcp][2], cp_new[maxcp][2];
	int iexe_wavis;
	char str_dir_output[maxstr], str_fn_cpw[maxstr];

	nf = 2;
	x[0] = 18.0;
	x[1] = 0.45;

	nc = 2;
	cl[0] = 13.0;
	cu[0] = 18.0;

// Origin
//	cl[1] = -0.5;
//	cu[1] = 0.5;

// Modified for scaling
	cl[1] = -5.0;
	cu[1] =  5.0;

	mit = 200;
	xmax = 1.0;
	tol = 1.0e-03;

	iexe_wavis = 1; //executed
	nc_obj = 0;

	tprint("OnBnClickedButtonRunWavisOptimize");
```

```
            tprint(twavis.dir_work_ori);

            twavis.getNewOptFolderName(str_dir_output);
            strcpy(twavis.dir_out, str_dir_output);
            tprint("twavis.dir_out", twavis.dir_out);
            twavis.grd_mesh_backup_origin();

            twavis.grd_mesh_backup_origin();
            nn = 0;
            twavis.get_grd_cp_curve(&nsp, cp);
            write_tecplot_2d("000_cp_curve_old_new_opti.dat", "CP-curve", "x y", nsp,
cp);

            twavis.get_shape_curveG(x1,x2, x[1], x[0], nsp, g);
            write_tecplot_2d("000_cp_curve_dx_opti.dat", "CP-curve-dx", "x y", nsp, g);

            twavis.modify_cp_with_Shape(nsp, cp, g, cp_new);
            twavis.tecplot_section("000-section-000.dat");

            twavis.runCPFCPW(iexe_wavis, nn, twavis.dir_out, str_fn_cpw);
            twavis.write_object_function_result(twavis.dir_work_opt, "[000]-
Results.out", str_fn_cpw, nn, "Results");

            twavis.grd_mesh_restore_origin();

            tprint("OnBnClickedButtonRunWavisOptimize");

            psqp_simple(nf, nc, x, cl, cu, mit, xmax, tol, &f, &gmax);

            return;
}
```