THESIS APPROVAL

GRADUATE SCHOOL, KASETSART UNIVERSITY

Master of Engineering (Mechanical Engineering)

DEGREE

Mechanical Engineering	Mechanical Engineering
FIELD	**DEPARTMENT**

TITLE: Simulation of Cavitation on Marine Propeller Using CFD

NAME: Mr. Prachakon Kaewkhiaw

THIS THESIS HAS BEEN ACCEPTED BY

THESIS ADVISOR

(Associate Professor Varangrat Juntasaro, Ph.D.)

THESIS CO-ADVISOR

(Mr. Yodchai Tiaple, Ph.D.)

DEPARTMENT HEAD

(Associate Professor Chawalit Kittichaikarn, Ph.D.)

APPROVED BY THE GRADUATE SCHOOL ON _____

DEAN

(Associate Professor Gunjana Theeragool, D.Agr.)

THESIS

SIMULATION OF CAVITATION ON MARINE PROPELLER USING CFD

PRACHAKON KAEWKHIAW

A Thesis Submitted in Partial Fulfillment of
the Requirements for the Degree of
Master of Engineering (Mechanical Engineering)
Graduate School, Kasetsart University
2010

Prachakon Kaewkhiaw 2010: Simulation of Cavitation on Marine Propeller Using CFD. Master of Engineering (Mechanical Engineering), Major Field: Mechanical Engineering, Department of Mechanical Engineering.
Thesis Advisor: Associate Professor Varangrat Juntasaro, Ph.D. 204 pages.

The realistic simulation of cavitation on a marine propeller is important for the efficient design of the propeller. However, the flow characteristic occurred on the marine propeller is complicated and difficult to predict due to the combined effects of turbulence, cavitation, complex geometry and multiphase phenomena. There is still currently no turbulence model that can predict these combined effects satisfactory. The nonlinear turbulence model is therefore modified and applied to predict the cavitation on a marine propeller for the first time in this work. It is found that both nonlinear and modified nonlinear turbulence models can predict the cavitation and hence the thrust and torque coefficients much more accurately than the existing Reynolds-averaged Navier-Stokes (RANS) turbulence models including Reynolds-stress model (RSM).

_____ _____ ___ / ___ / ___

Student's signature Thesis Advisor's signature

ACKNOWLEDGEMENTS

I thank many colleagues, friends, family, and faculty members who have helped me with this research. I am most indebted to Associate Professor Dr. Varangrat Juntasaro, my advisor, for her continuous support, guidance, inspiration, encouragement and for sharing her expertise and wisdom regarding my research. Second, I thank Dr. Yodchai Tiaple who have helped me about knowledge marine propeller. I thank Assistant Professor Dr. Putchong Uthayopas for the computing facilities from the Thai National Grid Center (TNGC).

Many thanks are due to my friends in the Computational Mechanics Lab (CML). They made my graduate school experience such an enjoyable one. I am also grateful for the help from my friends in Microsoft HPC & Grid computing and APSTF (SUN) labs.

Lastly, I would like to give special thanks my parents. This would not have been possible without their constant support.

Prachakon Kaewkhiaw

January 2010

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF TABLES (Continued)

LIST OF TABLES (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF ABBREVIATIONS

a_{ij}	Reynolds stress anisotropy tensor
$C_{\varepsilon 1}$, $C_{\varepsilon 2}$	Constants in source and sink terms of ε equation
C_{μ}	Constant in linear eddy viscosity models; a variable in nonlinear models
d_k	Diameter of the particles (or bubbles) of secondary phase k
D	Propeller diameter
D_{Tij}	Turbulent diffusion
E	Empirical constant
E_1, E_2	Extra terms
f	Vapor mass fraction
f_{drag}	Drag function
f_{μ}, $f_{\varepsilon 1}$, $f_{\varepsilon 2}$	Damping functions
g	Gravity
G_k	Production term of turbulent kinetic energy
J	Advance ratios
k	Turbulent kinetic energy
K_t	Thrust coefficient
K_q	Torque coefficient
n	The number of revolutions
P	Pressure
P_v	Vapor pressure
P_{∞}	Pressure far upstream
Re	Reynolds number
Re_t	Turbulent Reynolds number
$\tilde{S}$	Nondimensional strain rate
S_{ij} , $\overline{S}_{ij}$	Mean strain rate tensor

LIST OF ABBREVIATIONS (Continued)

S_p	Parallel relative speed up
t	Time
t_p	Time required solving the problem using p processors
t_1	Time required solving the problem using a single processor
T_{ij}	Stress tensor
u_i, u_j	Velocity components
$\overline{u_i' u_j'}$	Reynolds stresses
u_m	Velocity of mixture
u_{Mk}	Diffusion velocity
u_{qk}	Slip velocity between air and water
u_τ	Friction velocity; $u_\tau = \sqrt{\tau_w}/\rho$
U	Velocity in x-direction
V_a	The inflow speed
$\vec{v}_v$	Velocity vector of the vapor phase
x_i, x_j	Coordinate components
y	Distance from point to the wall
y_v	Viscous sublayer thickness

Greek symbols

δ_{ij}	Kronecker delta; $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ if $i \neq j$
ε	Dissipation rate of turbulent kinetic energy
ε_{ijk}	Alternating tensor
κ	von Karman constant and equals 0.4187
$\alpha_a, \alpha_k, \alpha_w$	Volume fraction of air, phase k and water
μ	Dynamic viscosity

LIST OF ABBREVIATIONS (Continued)

μ_a, μ_w	Dynamic viscosity of air and water
μ_{eff}	Effective viscosity
μ_t	Eddy viscosity
ρ	Density of fluid
ρ_v	Densities of vapor
ω	Specific dissipation rate
$\rho_a, \rho_k, \rho_m, \rho_w$	Density of air, phase k, mixture and water
σ	Cavitation number
σ_k	Turbulent Prandtl number of turbulent kinetic energy
σ_ε	Turbulent Prandtl number of dissipation rate of turbulent kinetic energy
$\tilde{\Omega}$	Nondimensional vorticity
γ	Effective exchange coefficient
Ω_{ij}, $\overline{\Omega}_{ij}$	Mean vorticity tensor
η_o	The efficiency of propeller

Abbreviations

DNS	Direct numerical simulation
RNG	Renormalization group

SIMULATION OF CAVITATION ON MARINE PROPELLER USING CFD

INTRODUCTION

Marine propeller researchers and designers have made numerous efforts to reduce the effect of cavitation, which degrades propeller performance, erodes blade surfaces, produces noise, and causes vibration on the ship hull. However, with increasing demand for heavily loaded propellers, the occurrence of cavitation is unavoidable nowadays. Therefore, the accurate prediction of cavitation is becoming important in the design of propellers to get the good propeller efficiency.

Computational methods for cavitation have been studied for over two decades. The methods can be largely categorized into two groups: single-phase modeling with cavitation interface tracking, and multi-phase modeling with an embedded cavitation interface. The former approach, i.e., single-phase modeling with cavitation interface tracking, has been widely adopted for inviscid flow solution methods, such as potential flow boundary element methods (1993) and Euler equation solvers (1994). These methods have evolved significantly and many successful application results have been presented, as reviewed in (Kinnas *et al*., 2002). Still in many cases they require cumbersome iterative procedures and a considerable amount of preliminary knowledge, such as cavity closure conditions. The latter approach, i.e., multi-phase modeling with an embedded cavitation interface, can be adopted for more general viscous flow solution methods, such as the Reynolds-averaged Navier-Stokes (RANS) equation solvers. Recently, it has become popular within the cavitation research community. This approach is more general for three-dimensional flows and it can include the effect of turbulence within the mixture. The current work therefore chooses the mixture multiphase model with the cavitation model of (Singhal *et al*., 2002) to simulate the cavitation on a marine propeller.

In 2004 (Rhee *et al*., 2004) used the RANS $k-\omega$ turbulence model to simulate the cavitation on a marine propeller to predict the thrust and torque coefficients. Later, Lifante and Frank (2008) used the $k-\omega$ *SST* turbulence model and the Reynolds-stress model (RSM) to simulate the same case as Rhee et al. They found that their results were more accurate than the results from Rhee et al. Therefore, it can be seen that the turbulence model plays an important role on the accuracy of the flow simulation using computational fluid dynamics (CFD). The results from Lifante and Frank are, however, not satisfactory enough. This is because the turbulence models used in their simulation are based on the linear Reynolds-stress expression. The conventional linear turbulence models that based on the linear Reynolds-stress expression are known to have less accuracy then the nonlinear turbulence models that based on the nonlinear Reynolds-stress expression especially for complex flow problems. There are currently many nonlinear turbulence models in the literature, e.g. (Nisizima and Yoshizawa, 1987; Speziale, 1987; Myong and Kasagi, 1990; Rubinstein and Barton, 1990; Shih *et al*., 1993; Lien *et al*., 1996; Craft *et al*., 1996; Apsley and Leschziner, 1998 and Abe *et al*., 2003). (Juntasaro *et al*., 2005) found that the nonlinear turbulence model of Craft *et al*. was the most suitable for complex flow problems.

The RANS nonlinear turbulence model becomes more popular in the last decade. It has been applied to many complex flow problems successfully; for examples, compressible flow through an S-shaped diffusing ducts (Juntasaro *et al*., 2005), flow through a rotating square duct (Gururatana *et al*., 2006) and multiphase recirculating free-surface flow over stepped spillways (Tongkratoke *et al*., 2009). It can be seen from the literature that the nonlinear turbulence model has never been applied to simulate the cavitation on a marine propeller before. Moreover, the model constants have not been optimized for the cavitation prediction on a marine propeller.

Therefore, the linear turbulence models in the CFD software FLUENT are modified in this work using the user-defined function (UDF) by replacing the linear Reynolds-stress term with the nonlinear Reynolds-stress term. The constants in the Craft et al.'s nonlinear turbulence model are also optimized with the experimental

data of (Boswell, 1971) using the MATLAB program. The performances of the modified nonlinear and the Craft et al.'s nonlinear $k\text{-}\varepsilon$ turbulence models in predicting the cavitation on a marine propeller are evaluated by comparing the predicted thrust and torque coefficients with those predicted by the linear $k\text{-}\varepsilon$ turbulence models (standard (1877), RNG (1993) and realizable (1995)) combined with the wall function equations (standard (1974), non-equilibrium (1995) and enhanced (1993]), the linear $k-\omega$ turbulence models (standard (1998) and SST (1994)) and the Reynolds-stress model (RSM) (1989).

OBJECTIVES

1. To predict thrust coefficient, torque coefficient and efficiency of propeller in this case non-cavitation for various advance ratios.

2. To apply the nonlinear turbulence model to simulation of cavitation on marine propeller.

3. To evaluate the performance in predict thrust and torque coefficients of various linear k-ε turbulence models, the Reynolds-stress model (RSM), the Craft et al.'s nonlinear k-ε turbulence model and the modified nonlinear k-ε turbulence model at different cavitation numbers and advance ratios.

LITERATURE REVIEW

Cavitation of marine propeller is born fluid in area propeller has high speed more than other area then make have the pressure lowers than vapor pressure and other area cause boil at propeller. It boils of bubble very much on propeller. In bubble have gas called nuclei when bubble burst will effect destruction and obstruct flow. Moreover flow around propeller disturbed. Furthermore efficiency of propeller is down and force is destruction of bubble cause pitting on propeller which cause damaged of propeller.

Types of cavitation on marine propeller

Types of cavitation on marine propeller can be classified by the following physical appearance as shown in Figure 1

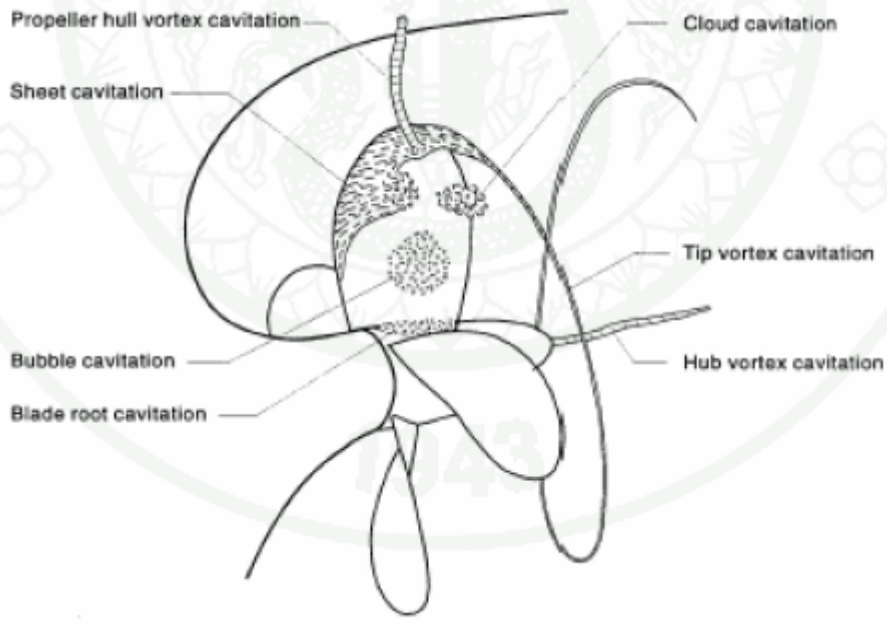


Figure 1 Types of cavitation on marine propeller

Tip and Hub vortex cavitation

The vortex types of cavitation with few exceptions occur at the blade tips and hub of the propeller and they are generated from the core of these vortices where the pressure is very low. When this pressure is lower than the vapor pressure, it occurs vortex cavitation. The tip vortex cavitation is normally first observed some distance behind the tips of the propeller blades which is said to be unattached but as the vortex becomes stronger, either through higher blade loading or decreasing in σ, it moves towards the blade tip and ultimately becomes attached. The hub vortex is formed by the combustion of individual vortices shed from each blade root and although individually these vortices are unlikely to cavity, under the influence of a converging propeller cone the combination of the blade root vortices has a high susceptibility to cavity. When this occurs the resulting cavitation is normally very stable appearing like a rope with strands corresponding to the number of blades of the propeller. This type of cavitation may also harm the rudders behind the propeller causing erosion on them.

Sheet cavitation

Sheet cavitation occurs when the pressure distribution has a strong adverse pressure gradient and the flow separates from the blade surface. Sheet cavitation initially becomes apparent at the leading edges of the propeller blades on the back when the blade sections are working at positive angle of attack. Sheet cavity is generally stable although there are cases where instability may occur. On commercial propeller the sheet cavity gradually merges with the tip vortex.

Bubble cavitation

Bubble cavitation is primarily affected by the pressure distribution which causes high suction pressure in the mid-chord region of the blade section. Thus the combination of camber line and section thickness plays an important role on the susceptibility of a propeller towards bubble cavitation. When the blade sections are

relatively thick and operate at a small angle of attack the bubble cavitation occur. Bubble cavities collapse very violently so that this cavitation is noisy and erosive.

Root Cavitation

This type of cavitation may occur at the blade root and has the shape of a wedge. The top of the wedge can be at the leading edge, but it can also start on the blade itself. Root cavitation is related to the horse shoe vortex developed at root as well as inclined shaft and wake shadow effect created by the shaft brackets, boss, etc. It is commonly observed on controllable pitch propellers (CPP).

Propeller-Hull Vortex (PHV) Cavitation

A special form of cavitation reported in early 1970's is the PHV cavitation. This type of cavitation can be described as the arching of a cavitating vortex between propeller tip and ship's hull and it is pronounced for small tip clearance of the propeller and hull. The PHV is considered to form due to turbulence and other flow disturbances close to the hull, causing a rotation about the stagnation point, which is accentuated away from the hull by the small radius of the control volume forming the vortex. The factors leading to the formation of PHV cavitations are suggested

1. Low advance coefficient
2. Low tip clearance
3. Flat hull surfaces above the propellers

Unsteady Sheet (Cloud) Cavitation

Cloud cavitation is frequently found behind strongly developed steady sheet cavities and generally in moderately separated flow in which small vortices from the origins for small cavities. This type of cavitation appears as a mist or cloud of very small bubbles and its presence should be taken seriously.

Governing Equations

Multiphase Models

Fluent (2006) Multiphase flow can be classified by the following regimes, grouped into four categories as shown in Figure 1:

1. Gas-liquid or liquid-liquid flows (bubbly flow, droplet flow, slug flow and stratified/free-surface flow)
2. Gas-solid flows (particle-laden flow, pneumatic transport and fluidized beds)
3. Liquid-solid flows (slurry flow, hydrotransport and sedimentation)
4. Three-phase flows

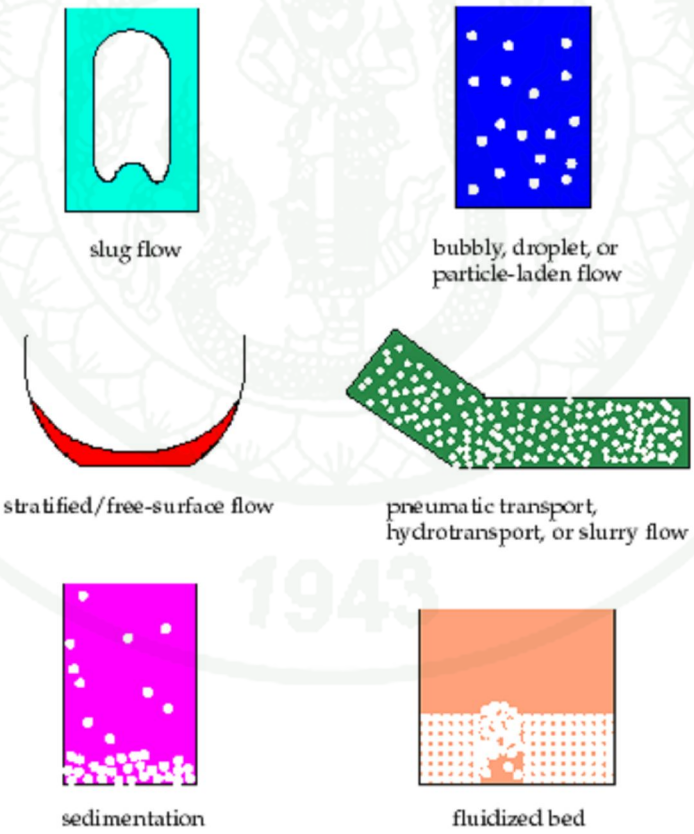


Figure 2 Multiphase flow regimes

Source: Fluent (2006)

Mixture model

The mixture model is designed for two or more phases (fluid or particulate). As in the Eulerian model, the phases are treated as interpenetrating continua. The mixture model can model *n* phases (fluid or particle) by solving the momentum, continuity, the continuity equations for the secondary phases, and algebraic expression for the relative velocities. Applications of the mixture model include particle-laden flows with low loading, bubbly flows, sedimentation, and cyclone separators. The mixture model can also be used without relative velocities for the dispersed phases to model homogeneous multiphase flow.

The mixture model is a simplified multiphase model that can be used to model multiphase flows where the phases move at different velocities, but assume local equilibrium over short spatial length scales. The coupling between the phases should be strong. It can also be used to model homogeneous multiphase flows with very strong coupling and the phases moving at the same velocity.

continuity equation

$$\frac{\partial \rho_m}{\partial t} + \frac{\partial}{\partial x_i}\left(\rho_m u_{m,i}\right) = 0 \tag{1}$$

where the density of mixture (ρ_m) and the velocity of mixture (u_m) are defined as

$$\rho_m = \sum_{k=1}^{n} \alpha_k \rho_k \tag{2}$$

$$u_m = \frac{1}{\rho_m} \sum_{k=1}^{n} \alpha_k \rho_k u_k \tag{3}$$

where α_k and ρ_k are the volume fraction and density of phase k, respectively. The velocity of mixture (u_m) represents the velocity of the mass centre of the mixture flow.

momentum equation

$$\frac{\partial \rho_m u_{m,i}}{\partial t} + \frac{\partial}{\partial x_j}\left(\rho_m u_{m,i} u_{m,j}\right) = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}\left\{\mu_{eff}\left(\frac{\partial u_{m,i}}{\partial x_j} + \frac{\partial u_{m,j}}{\partial x_i}\right)\right\} - \frac{\partial \rho_m \overline{u'_{m,i} u'_{m,j}}}{\partial x_j} + \rho_m g_i \quad (4)$$

continuity equation for phase k

From the continuity equation for secondary phase k, the volume fraction equation for secondary phase k can be obtained

$$\frac{\partial}{\partial t}\left(\alpha_k \rho_k\right) + \frac{\partial}{\partial x_i}\left(\alpha_k \rho_k u_{m,i}\right) = -\frac{\partial}{\partial x_i}\left(\alpha_k \rho_k u_{Mk}\right) \quad (5)$$

relative velocity

Before solving the continuity equations (10) for phase k and the momentum equations (9) for the mixture, the diffusion velocity (u_{Mk}) has to be determined. The diffusion velocity of a phase is usually caused by the density differences, resulting in forces on the bubbles different from those on the fluid. The additional force is balanced by the drag force.

$$u_{Mk} = u_{qk} - \sum_{k=1}^{n} \frac{\alpha_k \rho_k}{\rho_m} u_{qk} \quad (6)$$

where u_{qk} is the slip velocity between air and water, defined as the velocity of air relative to the velocity of water. Following (Manninen *et al.*, 1996), u_{qk} is defined as

$$u_{qk} = \frac{(\rho_m - \rho_m) d_k^2}{18 \mu_{eff} f_{drag}} \left[g_i - (u_{m,i} \cdot \nabla) u_{m,i} - \frac{\partial u_{m,i}}{\partial t} \right] \quad (7)$$

where μ_{eff} is the effective viscosity of mixture and d_k is the diameter of the particles (or bubbles) of secondary phase k. The bubble diameter used in the simulation is 5 mm. Reasonable agreement with the experimental data of Cummings and Chanson (1997) is obtained by using this bubble diameter value. The drag function (f_{drag}) is taken from Clift *et al*. (1978).

$$f_{drag} = \begin{cases} 1 + 0.15 Re^{0.687} & Re \le 1000 \\ 0.0182 Re & Re > 1000 \end{cases} \quad (8)$$

Modeling Turbulence Flows

Fluent (2006) The most fundamental approach for the turbulence study is the direct numerical simulation (DNS). The DNS solves the 3D Navier-Stokes equations directly without using any turbulence model. The results from the DNS are the most accurate when compared with the experimental data. However, the disadvantage of the DNS is the need of the extremely high computing capacity and CPU time. The DNS hence can be used only with a simple geometry and flows with low Reynolds numbers. The considered in this work is Reynolds-averaged Navier-Stokes (RANS) turbulence modeling. The RANS turbulence model is suitable for the engineering problems. It has more advantage than the DNS in the sense that it requires less computational time and computing power. The most popular RANS model is the linear turbulence model.

In this work, various linear $k-\varepsilon$ turbulence models (standard (1972), RNG (1993) and realizable (1995)) combined with the wall function equations (standard [1974], non-equilibrium [1995] and enhanced [1993]), the linear $k-\omega$ turbulence models (standard [1998] and SST [1994]) and the Reynolds-stress model (RSM) [1989] are employed.

Reynolds-Averaged Navier-Stokes (RANS) turbulence models

RANS turbulence modeling can be classified into two groups: Eddy-Viscosity Models (EVMs) and Reynolds-Stress Models (RSMs). EVMs are based on the Boussinesq (1877) hyposthesis where the Reynolds stresses are proportional to the rates of strain as follows

$$-\rho\overline{u_i'u_j'} = \mu_t\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)-\frac{2}{3}\rho k\delta_{ij} \tag{9}$$

where μ_t is the eddy viscosity, $k=\frac{1}{2}\overline{u_iu_i}$ is the kinetic energy of turbulence and δ_{ij} is the Kronecker delta ($\delta_{ij}=1$ if $i=j$ and $\delta_{ij}=0$ if $i\neq j$).

standard $k-\varepsilon$ turbulence model

Launder and Spalding (1972) proposed the standard $k-\varepsilon$ model that was a semi-empirical model based on the model transport equations of the turbulent kinetic energy (k) and its dissipation rate (ε). The k equation was derived from the exact equation and the ε equation was obtained from the physical reasoning. In the derivation of the $k-\varepsilon$ model, it was assumed that the flow was fully developed turbulent, and the effects of molecular viscosity were negligible. Therefore, this model is valid for only fully developed turbulent flows. The turbulent kinetic energy (k) and the dissipation rate (ε) equations are written as follows.

$$\frac{\partial}{\partial t}(\rho k)+\frac{\partial}{\partial x_i}(\rho k u_i)=\frac{\partial}{\partial x_j}\left[\left(\mu+\frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_i}\right]+G_k-\rho\varepsilon \tag{10}$$

$$\frac{\partial}{\partial t}(\rho\varepsilon)+\frac{\partial}{\partial x_i}(\rho\varepsilon u_i)=\frac{\partial}{\partial x_j}\left[\left(\mu+\frac{\mu_t}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x_j}\right]+C_{\varepsilon 1}\frac{\varepsilon}{k}G_k-C_{\varepsilon 2}\rho\frac{\varepsilon^2}{k} \tag{11}$$

modeling of eddy viscosity

The eddy viscosity μ_t is written as follows.

$$\mu_t = \rho C_\mu \frac{k^2}{\varepsilon} \tag{12}$$

RNG k - ε turbulence model

The RNG k - ε model was derived from the instantaneous Navier-Stokes equations, using a mathematical technique called "renormalization group" (RNG) method. The model constants, additional terms and functions of the RNG k - ε model are different from the standard k - ε model. The more description of RNG theory can be found in Choudhury (1993).

$$\frac{\partial}{\partial t}(\rho k) + \frac{\partial}{\partial x_i}(\rho k u_i) = \frac{\partial}{\partial x_j}\left[(\alpha_k \mu_{eff})\frac{\partial k}{\partial x_i}\right] + G_k - \rho\varepsilon \tag{13}$$

$$\frac{\partial}{\partial t}(\rho \varepsilon) + \frac{\partial}{\partial x_i}(\rho \varepsilon u_i) = \frac{\partial}{\partial x_j}\left[(\alpha_\varepsilon \mu_{eff})\frac{\partial \varepsilon}{\partial x_j}\right] + C_{\varepsilon 1}\frac{\varepsilon}{k}G_k - C^*_{\varepsilon 2}\rho\frac{\varepsilon^2}{k} \tag{14}$$

$$C^*_{\varepsilon 2} = C_{\varepsilon 2} + \frac{C_\mu \rho \eta^3 (1 - \eta/\eta_0)}{1 + \beta\eta^3} \tag{15}$$

$$\eta = Sk/\varepsilon \tag{16}$$

$$S = \sqrt{2S_{ij}S_{ij}} \tag{17}$$

$$S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{18}$$

modeling of effective viscosity

The scale elimination procedure in RNG theory results in a differential equation for turbulent viscosity. In high Reynolds number $\mu_{eff} \approx 1$.

realizable k - ε turbulence model

$$\frac{\partial}{\partial t}(\rho k)+\frac{\partial}{\partial x_i}(\rho k u_i)=\frac{\partial}{\partial x_j}\left[\left(\mu+\frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_i}\right]+G_k - \rho\varepsilon \tag{19}$$

$$\frac{\partial}{\partial t}(\rho \varepsilon)+\frac{\partial}{\partial x_i}(\rho \varepsilon u_i)=\frac{\partial}{\partial x_j}\left[\left(\mu+\frac{\mu_t}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x_j}\right]+\rho C_1 S\varepsilon - \rho C_2 \frac{\varepsilon^2}{k+\sqrt{v\varepsilon}} \tag{20}$$

$$\eta = Sk/\varepsilon \tag{21}$$

$$C_1 = max\left[0.43, \frac{\eta}{\eta+5}\right] \tag{22}$$

$$S=\sqrt{2S_{ij}S_{ij}} \tag{23}$$

$$S_{ij}=\frac{1}{2}\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right) \tag{24}$$

modeling of eddy viscosity

$$\mu_t = \rho C_\mu \frac{k^2}{\varepsilon} \tag{25}$$

$$C_\mu = \frac{1}{A_0 + A_s \frac{kU^*}{\varepsilon}} \tag{26}$$

$$\tilde{\Omega}_{ij} = \Omega_{ij} - 2\varepsilon_{ijk}\omega_k \tag{27}$$

$$\Omega_{ij} = \bar{\Omega}_{ij} - \varepsilon_{ijk}\omega_k \tag{28}$$

$$A_s = \sqrt{6}\cos\phi \tag{29}$$

$$\phi = \frac{1}{3}\cos^{-1}\left(\sqrt{6}W\right) \tag{30}$$

$$W = \frac{S_{ij}S_{jk}S_{ki}}{\tilde{S}} \tag{31}$$

$$S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{32}$$

modeling turbulent production in the $k-\varepsilon$ models

The term G_k, representing the production of turbulent kinetic energy, is modeled identically for the standard, RNG, and realizable $k-\varepsilon$ models. From the exact equation for the transport of k, this term may be defined as

$$G_k = -\rho\overline{u_i' u_j'}\frac{\partial u_i}{\partial x_j} \tag{33}$$

$$G_k = \mu_t S^2 \tag{34}$$

$$S = \sqrt{2S_{ij}S_{ij}} \tag{35}$$

near wall treatments

standard wall function

The standard wall function is based on the proposed of Launder and Spalding (1972). This function used the principle of the law of the wall for the mean velocity in the range of y^+ between 30 and 60. In the case of y^+ less than 30, the relation between the mean velocity and the position is considered as linear.

The law-of-the-wall for mean velocity yields

$$u^* = \frac{1}{\kappa} ln\left(Ey^*\right) \tag{36}$$

$$y^* = \frac{\rho C_\mu^{1/4} k^{1/2} y}{\mu} \tag{37}$$

$$y^+ = \frac{\rho u_\tau y}{\mu} \tag{38}$$

where κ is von Karman constant and equals 0.4187, E is empirical constant and equals 9.793, k is the turbulent kinetic energy, y is distance from point to the wall, μ is dynamic viscosity and u_τ is friction velocity.

non-equilibrium wall function

The non-equilibrium wall function is based on the law of the wall of Launder and Spalding (1972) for mean velocity. Kim and Choudhury (1995) modified the wall function into non-equilibrium by adding pressure-gradient effects. Moreover, the two-layer-based concept was adopted to compute the budget of turbulent kinetic energy in the wall neighboring cells.

The log-law for mean velocity sensitized to pressure gradients is

$$\frac{\tilde{u}C_{\mu}^{1/4}k^{1/2}}{\tau_w/\rho}=\frac{1}{\kappa}ln\left(E\frac{\rho C_{\mu}^{1/4}k^{1/2}y}{\mu}\right) \tag{39}$$

$$\tilde{u}=u-\frac{1}{2}\frac{dp}{dx}\left[\frac{y_v}{\rho\kappa\sqrt{k}}ln\left(\frac{y}{y_v}\right)+\frac{y-y_v}{\rho\kappa\sqrt{k}}+\frac{y_v^2}{\mu}\right] \tag{40}$$

where the viscous sublayer thickness (y_v) is computed from

$$y_v=\frac{\mu y_v^*}{\rho C_{\mu}^{1/4}k^{1/2}} \tag{41}$$

where y_v^* = 11.225, κ is von Karman constant and equals 0.4187, E is empirical constant and equals 9.793, k is the turbulent kinetic energy, y is distance from point to the wall and μ is dynamic viscosity.

enhanced wall function

Enhanced wall treatment is a near-wall modeling method that combines a two-layer model with enhanced wall function. The two-layer model considers the mean velocity into laminar and turbulent. In the case of laminar, functions are solved as linear relation. For the case of turbulent, functions are solve by using logarithmic. Then combine both results and add blending function. These modifications become enhanced wall function.

We achieve this by blending the linear (laminar) and the logarithmic (turbulent) laws-of-the-wall using a function suggested by Kader (1993)

$$u^+=e^{\Gamma}u_{lam}^+ + e^{1/\Gamma}u_{turb}^+ \tag{42}$$

where the blending function is given by

$$\Gamma = -\frac{a\left(y^{+}\right)^{4}}{1+by^{+}} \tag{43}$$

where $a = 0.01$ and $b = 5$.

standard $k-\omega$ turbulence model

The standard $k-\omega$ model is an empirical model based on model transport equations for the turbulence kinetic energy (k) and the specific dissipation rate (ω), which can also be thought of as the ratio of ε to k. The production terms have been added to both k and ω equations, which have improved the accuracy of the model for predicting free shear flows. The turbulence kinetic energy, k and the specific dissipation rate, ω equations are written as follows.

$$\frac{\partial}{\partial t}(\rho k)+\frac{\partial}{\partial x_i}(\rho k u_i)=\frac{\partial}{\partial x_j}\left(\Gamma_k \frac{\partial k}{\partial x_j}\right)+G_k-Y_k \tag{44}$$

$$\frac{\partial}{\partial t}(\rho \omega)+\frac{\partial}{\partial x_i}(\rho \omega u_i)=\frac{\partial}{\partial x_j}\left(\Gamma_\omega \frac{\partial \omega}{\partial x_j}\right)+G_\omega-Y_w \tag{45}$$

Modeling the Effective Diffusivity

The effective diffusivities for the $k-\omega$ model are written as follows.

$$\Gamma_k = \mu+\frac{\mu_t}{\sigma_k} \tag{46}$$

$$\Gamma_\omega = \mu+\frac{\mu_t}{\sigma_\omega} \tag{47}$$

$$\mu_t = \alpha^{*}\frac{\rho k}{\omega} \tag{48}$$

SST standard $k-\omega$ turbulence model

The shear-stress transport (SST) $k-\omega$ model was developed to effectively blend the robust and accurate formulation of the $k-\omega$ model in the near-wall region with the free-stream independence of the $k-\varepsilon$ model in the far field. To achieve this, the $k-\varepsilon$ model is converted into a $k-\omega$ formulation. The SST $k-\omega$ model is similar to the standard $k-\omega$ model, but includes the following refinements. The SST $k-\omega$ models are written as follows.

$$\frac{\partial}{\partial t}(\rho k)+\frac{\partial}{\partial x_i}(\rho k u_i)=\frac{\partial}{\partial x_j}\left(\Gamma_k \frac{\partial k}{\partial x_j}\right)+\bar{G}_k-Y_k \tag{49}$$

$$\frac{\partial}{\partial t}(\rho \omega)+\frac{\partial}{\partial x_i}(\rho \omega u_i)=\frac{\partial}{\partial x_j}\left(\Gamma_\omega \frac{\partial \omega}{\partial x_j}\right)+G_\omega-Y_w+D_w \tag{50}$$

Modeling the Effective Diffusivity

The effective diffusivities for the SST $k-\omega$ model are written as follows.

$$\Gamma_k=\mu+\frac{\mu_t}{\sigma_k} \tag{51}$$

$$\Gamma_\omega=\mu+\frac{\mu_t}{\sigma_\omega} \tag{52}$$

$$\mu_t=\frac{\rho k}{\omega}\frac{1}{max\left[\frac{1}{\alpha^*},\frac{SF_2}{\alpha_1 \omega}\right]} \tag{53}$$

Reynolds Stress Model (RSM)

The Reynolds stress model (RSM) is the most elaborate turbulence model that FLUENT provides. Abandoning the isotropic eddy-viscosity hypothesis, the RSM closes the Reynolds-averaged Navier-Stokes equations by solving transport equations for the Reynolds stresses, together with an equation for the dissipation rate. This

means that five additional transport equations are required in 2D flows and seven additional transport equations must be solved in 3D.

The exact transport equations for the transport of the Reynolds stresses, $\rho\overline{u_i'u_j'}$ may be written as follows:

$$\underbrace{\frac{\partial}{\partial_t}\rho\overline{u_i'u_j'}}_{\text{Local Time Derivative}} + \underbrace{\frac{\partial}{\partial x_k}\left(\rho u_k\overline{u_i'u_j'}\right)}_{C_{ij}\equiv\text{Convevtion}} = \underbrace{-\frac{\partial}{\partial x_k}\left[\rho\overline{u_i'u_j'u_k'} + \overline{p\left(\delta_{kj}u_i' + \delta_{ij}u_j'\right)}\right]}_{D_{Tij}\equiv\text{Turbulent Diffusion}} + \underbrace{\frac{\partial}{\partial x_k}\left[\mu\frac{\partial}{\partial x_k}\left(\overline{u_i'u_j'}\right)\right]}_{D_{Lij}\equiv\text{Molecular Diffusion}}$$

$$\underbrace{-\rho\left(\overline{u_i'u_k'}\frac{\partial u_j}{\partial x_k} + \overline{u_j'u_k'}\frac{\partial u_i}{\partial x_k}\right)}_{P_{ij}\equiv\text{Stress Production}} \underbrace{-2\mu\overline{\frac{\partial u_i'}{\partial x_k}}\overline{\frac{\partial u_j'}{\partial x_k}}}_{\varepsilon_{ij}\equiv\text{Dissipation}} \underbrace{-2\rho\Omega_k\left(\overline{u_j'u_m'}\varepsilon_{ikm} + \overline{u_i'u_m'}\varepsilon_{jkm}\right)}_{F_{ij}\equiv\text{Production by System Rotation}} \quad (54)$$

Modeling Turbulent Diffusive Transport

D_{Tij} use a scalar turbulent diffusivity as follows

$$D_{Tij} = \frac{\partial}{\partial x_k}\left(\frac{\mu_t}{\sigma_k}\frac{\partial\overline{u_i'u_j'}}{\partial x_k}\right) \quad (55)$$

The turbulent viscosity, μ_t written as follows.

$$\mu_t = \rho C_\mu\frac{k^2}{\varepsilon} \quad (56)$$

Where C_μ = 0.09

nonlinear and modified nonlinear turbulence models

The Reynolds-stress expression in the nonlinear turbulence model of (Craft *et al*., 1996) is written as

$$-\rho\overline{u_i'u_j'} = -ka_{ij} - \frac{2}{3}\rho k\delta_{ij} \tag{57}$$

where δ_{ij} is the Kronecker delta ($\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ if $i \neq j$) and a_{ij} (Reynolds stress anisotropy tensor) is written as

$$\begin{aligned} a_{ij} = & -\frac{\mu_t}{k}S_{ij} \\ & + C_1\frac{\mu_t}{\varepsilon}\left(S_{ik}S_{jk} - \frac{1}{3}S_{kl}S_{kl}\delta_{ij}\right) \\ & + C_2\frac{\mu_t}{\varepsilon}\left(\Omega_{ik}S_{jk} + \Omega_{jk}S_{ik}\right) \\ & + C_3\frac{\mu_t}{\varepsilon}\left(\Omega_{ik}\Omega_{jk} - \frac{1}{3}\Omega_{kl}\Omega_{kl}\delta_{ij}\right) \\ & + C_4\frac{\mu_t k}{\varepsilon^2}\left(S_{ki}\Omega_{lj} + S_{kj}\Omega_{li}\right)S_{kl} \\ & + C_5\frac{\mu_t k}{\varepsilon^2}\left(\Omega_{il}\Omega_{lm}S_{mj} + S_{il}\Omega_{lm}\Omega_{mj} - \frac{2}{3}S_{lm}\Omega_{mn}\Omega_{nl}\delta_{ij}\right) \\ & + C_6\frac{\mu_t k}{\varepsilon^2}S_{ij}S_{kl}S_{kl} + C_7\frac{\mu_t k}{\varepsilon^2}S_{ij}\Omega_{kl}\Omega_{kl} \end{aligned} \tag{58}$$

where the eddy viscosity (μ_t), the turbulent kinetic energy (k) and its rate of dissipation (ε) are written as

$$\mu_t = \rho C_\mu f_\mu \frac{k^2}{\varepsilon} \tag{59}$$

$$\frac{\partial}{\partial t}(\rho k) + \frac{\partial}{\partial x_i}(\rho k u_i) = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_i}\right] + G_k - \rho\varepsilon + E_1 + E_2 \tag{60}$$

$$\frac{\partial}{\partial t}(\rho\varepsilon) + \frac{\partial}{\partial x_i}(\rho\varepsilon u_i) = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_\varepsilon}\right)\frac{\partial\varepsilon}{\partial x_j}\right] + f_{\varepsilon 1}C_{\varepsilon 1}\frac{\varepsilon}{k}G_k - f_{\varepsilon 2}C_{\varepsilon 2}\rho\frac{\varepsilon^2}{k} + E_1 + E_2 \tag{61}$$

$$G_k = -\rho\overline{u_i' u_j'}\frac{\partial u_i}{\partial x_j} \tag{62}$$

where G_k is production term of turbulent kinetic energy, ($C_{\varepsilon 1}$, $C_{\varepsilon 2}$, σ_k, σ_ε) are model constants, ($f_{\varepsilon 1}$, $f_{\varepsilon 2}$) are damping functions, and (E_1, E_2) are extra terms (Launder and Sharma, (1974)). The C_μ and the damping function (f_μ) of Craft *et al*. (1996) are written as

$$C_\mu = \frac{0.3}{1+0.35\left(max\left(\tilde{S},\tilde{\Omega}\right)\right)^{1.5}} \times\left(1-exp\left(\frac{-0.36}{exp\left(-0.75max\left(\tilde{S},\tilde{\Omega}\right)\right)}\right)\right) \tag{63}$$

$$f_\mu = 1-exp\left[-\left(Re_t/90\right)^{1/2}-\left(Re_t/400\right)^2\right] \tag{64}$$

where Re_t is turbulent Reynolds number

$$Re_t = \frac{\rho k^2}{\mu\varepsilon} \tag{65}$$

$$\tilde{S} = \frac{k}{\varepsilon}\sqrt{1/2S_{ij}S_{ij}} \tag{66}$$

$$\tilde{\Omega} = \frac{k}{\varepsilon}\sqrt{1/2\Omega_{ij}\Omega_{ij}} \tag{67}$$

$$S_{ij} = \left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right) \tag{68}$$

$$\Omega_{ij} = \left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} \right) - \varepsilon_{ijk}\Omega_k \tag{69}$$

The original model constants of Craft et al. and the modified model constants in the present work are listed in Table 1. The modified constants are found using the MATLAB program by optimizing the model constants with the experimental data of (Boswell, 1971).

Table 1 Model constants of the modified nonlinear turbulence model for the cavitation prediction on a marine propeller

Type	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_μ
Craft et al.'s nonlinear	-0.1	0.1	0.26	$-10C_\mu^2$	0	$-5C_\mu^2$	$5C_\mu^2$	$\frac{0.3}{1+0.35\left(max\left(\tilde{S},\tilde{\Omega}\right)\right)^{1.5}}\times\left(1-exp\left(\frac{-0.36}{exp\left(-0.75max\left(\tilde{S},\tilde{\Omega}\right)\right)}\right)\right)$
Modified nonlinear	-0.1	**0.19**	**0.21**	$-10C_\mu^2$	0	$-5C_\mu^2$	$5C_\mu^2$	$\frac{0.3}{1+0.35\left(max\left(\tilde{S},\tilde{\Omega}\right)\right)^{0.5}}\times\left(1-exp\left(\frac{-0.36}{exp\left(-0.75max\left(\tilde{S},\tilde{\Omega}\right)\right)}\right)\right)$

Cavitation Models

Fluent (2006) It has the capability to account for multiphase (N-phase) flows or flows with multiphase species transport, the effects of slip velocities between the liquid and gaseous phases, and the thermal effects and compressibility of both liquid and gas phases.

Vapor Mass Fraction and Vapor Transport

The working fluid is assumed to be a mixture of liquid, vapor and non-condensable gases. Standard governing equations in the mixture model and the mixture turbulence model describe the flow and account for the effects of turbulence. A vapor transport governing equations the vapor mass fraction, f are written as follows

$$\frac{\partial}{\partial t}(\rho f)+\nabla(\rho\vec{v}_v f)=\nabla(\gamma\nabla f)+R_e+R_c \tag{70}$$

Where ρ is the mixture density, $\vec{v}_v$ is the velocity vector of the vapor phase, γ is the effective exchange coefficient, R_e and R_c are the vapor generation and condensation rate terms (or phase change rates). The rate expressions are derived from the Rayleigh- Plesset equations, and limiting bubble size considerations (interface surface area per unit volume of vapor). These rates are functions of the instantaneous, local static pressure are written as follows

When $P < P_{sat}$

$$R_e = C_e\frac{V_{ch}}{\sigma}\rho_l\rho_v\sqrt{\frac{2(P_{sat}-P)}{3\rho l}}(1-f) \tag{71}$$

When $P > P_{sat}$

$$R_e = C_c \frac{V_{ch}}{\sigma} \rho_l \rho_v \sqrt{\frac{2(P - Psat)}{3\rho l}} f \tag{72}$$

where the suffixes l and v are the liquid and vapor phases, V_{ch} is a characteristic velocity, which is approximated by the local turbulence intensity, σ is the surface tension coefficient of the liquid, P_{sat} is the liquid saturation vapor pressure at the given temperature, and C_e and C_c are empirical constants. The default values are $C_e = 0.02$ and $C_c = 0.01$.

Turbulence-Induced Pressure Fluctuations

The cavitation model accounts for the turbulence-induced pressure fluctuations by simply raising the phase-change threshold pressure from P_{sat} to

$$P_v = \frac{1}{2}\left(P_{sat} + P_{turb}\right) \tag{73}$$

Where $P_{turb} = 0.39\rho k$

Effects of Non-condensable Gases

The operating liquid usually contains small finite amounts of non-condensable gases (e.g., dissolved gases, aeration). Even a very small of non-condensable gases can have significant effects on the cavitating flow field due to expansion at low pressures. In the present approach, the working fluid is assumed to be a mixture of the liquid phase and the gaseous phase, with the gaseous phase comprising of the liquid vapor and the non-condensable gases. The ρ density of the mixture, ρ is calculated as

$$\rho = \alpha_v \rho_v + \alpha_g \rho_g + \left(1 - \alpha_v - \alpha_g\right)\rho_l \tag{74}$$

Where ρ_l, ρ_v and ρ_g are the densities of the liquid, the vapor, and the non-condensable gases, respectively, and α_l, α_v, and α_g are the respective volume fractions. The volume fraction α_i written as follows

$$\alpha_i = f_i \frac{\rho}{\rho_i} \tag{75}$$

Phase Change Rates

After accounting for the effects of turbulence-induced pressure fluctuations and non-condensable gases, the final phase rate expressions are written as follows

When $P < P_v$

$$R_e = C_e \frac{\sqrt{k}}{\sigma} \rho_l \rho_v \sqrt{\frac{2(P_v - P)}{3\rho l}} \left(f_v - f_g\right) \tag{76}$$

When $P > P_v$

$$R_e = C_c \frac{\sqrt{k}}{\sigma} \rho_l \rho_v \sqrt{\frac{2(P - P_v)}{3\rho l}} f_v \tag{77}$$

Numerical Method

The computational fluid dynamics software FLUENT is employed in this work. The calculation domain is divided into discrete control volumes by the unstructured grid, which has a high flexibility to fit the complex geometry of the propeller. The governing equations are discretized by using the finite volume method. The pressure–velocity coupling is achieved through the SIMPLE algorithm. The discretized equations are solved using pointwise Gauss-Seidel iterations. The technique of moving reference frame is also employed. The grid-independent study is made for all cavitation cases as presented in Table 2. Figure 3 shows the grid divisions for the propeller at advance ratio, $J = 0.5$ and cavitation number, $\sigma = 3$. The modified nonlinear and the Craft et al.'s nonlinear $k\text{-}\varepsilon$ turbulence models are implemented separately in the FLUENT using the user-defined function.

The parallel performance is indicated by the parallel relative speed up, S_p given by $S_p = t_1/t_p$ where t_p is the time required to solve the problem using p processors and t_1 is the time required to solve the problem using a single processor. The parallel relative speed up is approximately 21 using 64 numbers of processors.

Table 2 Grid sensitivity study for cavitation on marine propeller

Cases	Cavitation numbers [σ]	Advance ratios [J]	Grid Independence (cells)			
			SST $k-\omega$	RSM	Craft et al.' nonlinear	Modified nonlinear
1	2	0.5	2,624,709	2,624,709	2,624,709	2,624,709
2	2	0.6	2,620,135	2,620,135	2,620,135	2,620,135
3	2	0.7	2,659,724	2,659,724	2,659,724	2,659,724
4	3	0.5	2,639,515	2,639,515	2,639,515	2,639,515
5	3	0.6	2,598,372	2,598,372	2,598,372	2,598,372
6	3.5	0.7	2,683,613	2,683,613	2,683,613	2,683,613
7	5	0.5	2,589,341	2,589,341	2,589,341	2,589,341
8	5	0.6	2,714,925	2,714,925	2,714,925	2,714,925
9	5	0.7	2,673,278	2,673,278	2,673,278	2,673,278

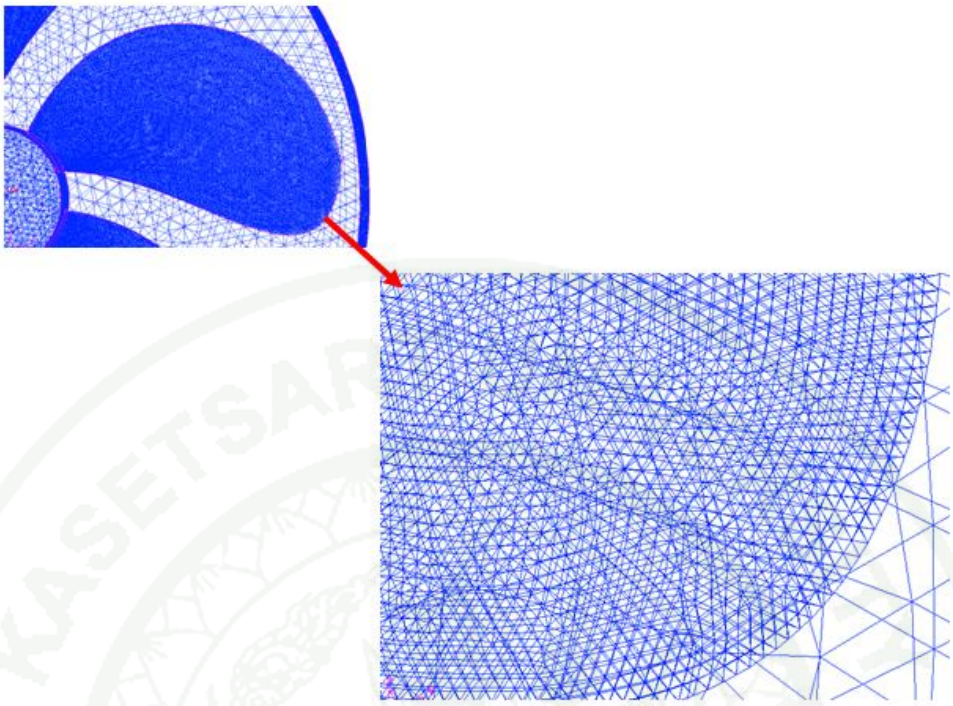


Figure 3 Grid divisions for the propeller at advance ratio $J = 0.5$ and cavitation number $\sigma = 3$

1. Settings for simulation of cavitation on marine propeller

1.1 The linear turbulence model approach

The numerical simulation settings for simulation of cavitation on marine propeller for the linear turbulence model approach are shown in Tables 3 to 13

Table 3 Model settings for the linear $k-\omega$ turbulence model approach

Model	Settings
DTMB 4382	3D
Time	Steady
Multiphase	Mixture
Viscous	$k-\omega$ *SST* turbulence model
Materials	Water-liquid , Water-vapor
Phase Interaction	Cavitation

Table 4 Types of the boundary conditions in each zone for the linear $k-\omega$ turbulence model approach

Name	Types
Blade	Wall
Hub	Wall
Inlet	Pressure-Inlet
Outlet	Presure-Outlet
Volume_in	Fluid
Volume_out	Fluid

Table 5 Setup conditions for the linear $k-\omega$ turbulence model approach

Name	Types	Phase	Wall Motion	Motion	Rotation-Axis Direction
Blade	Wall	Mixture	Moving Wall	Rotational	X=1 , Y=0 ,Z=0
Hub	Wall	Mixture	Moving Wall	Rotational	X=1 , Y=0 ,Z=0

Table 6 Setup motion type for the linear $k-\omega$ turbulence model approach

Name	Types	Phase	Rotation-Axis Direction	Motion Type	Rotational Velocity Speed (rpm)
Volume_in	Fluid	Mixture	X=1, Y=0, Z=0	Moving Reference Frame	1,200
Volume_out	Fluid	Mixture	X=1, Y=0, Z=0	-	-

Table 7 Setup interface zones for the linear $k-\omega$ turbulence model approach

Grid Interface	Interface Zone 1	Interface Zone 2
Interface	Interface_in	Interface_out

Table 8 Setup solution controls for the linear $k-\omega$ turbulence model approach

Equation	Solved
Flow	Yes
Vapor	Yes
Turbulence	Yes

Table 9 Setup relaxation factors for the linear $k-\omega$ turbulence model approach

Variable	Relaxation Factors
Pressure	0.3
Density	1
Body Forces	1
Momentum	0.4
Vaporization Mass	1
Vapor	0.2
Turbulent Kinetic Energy	0.4
Turbulent Dissipation Rate	0.4
Turbulent Viscosity	1

Table 10 Setup discretization for the linear $k-\omega$ turbulence model approach

Variable	Scheme
Pressure-velocity Coupling	SIMPLE
Pressure	Standard
Density	Second Order Upwind
Momentum	Second Order Upwind
Vapor	Second Order Upwind
Turbulent Kinetic Energy	Second Order Upwind
Turbulent Dissipation Rate	Second Order Upwind

Table 11 Setup residual for the linear $k-\omega$ turbulence model approach

Residual	Absolute Criteria
continuity	1.00E-06
x-velocity	1.00E-06
y-velocity	1.00E-06
z-velocity	1.00E-06
k	1.00E-06
omega	1.00E-06
vf-vapor	1.00E-06

1.2 The RSM approach

The numerical simulation settings for simulation of cavitation on marine propeller for the RSM approach are shown in Tables 14 to 24

Table 12 Model settings for the RSM approach

Model	Settings
DTMB 4382	3D
Time	Steady
Multiphase	Mixture
Viscous	Reynolds Stress
Materials	Water-liquid, Water-vapor
Phase Interaction	Cavitation

Table 13 Types of the boundary conditions in each zone for the RSM approach

Name	Types
Blade	Wall
Hub	Wall
Inlet	Pressure-Inlet
Outlet	Presure-Outlet
Volume_in	Fluid
Volume_out	Fluid

Table 14 Setup conditions for the RSM approach

Name	Types	Phase	Wall Motion	Motion	Rotation-Axis Direction
Blade	Wall	Mixture	Moving Wall	Rotational	X=1, Y=0, Z=0
Hub	Wall	Mixture	Moving Wall	Rotational	X=1, Y=0, Z=0

Table 15 Setup motion type for the RSM approach

Name	Types	Phase	Rotation-Axis Direction	Motion Type	Rotational Velocity Speed (rpm)
Volume_in	Fluid	Mixture	X=1, Y=0, Z=0	Moving Reference Frame	1,200
Volume_out	Fluid	Mixture	X=1, Y=0, Z=0	-	-

Table 16 Setup interface zones for the RSM approach

Grid Interface	Interface Zone 1	Interface Zone 2
Interface	Interface_in	Interface_out

Table 17 Setup solution controls for the RSM approach

Equation	Solved
Flow	Yes
Vapor	Yes
Turbulence	Yes
Reynolds Stresses	Yes

Table 18 Setup relaxation factors for the RSM approach

Variable	Relaxation Factors
Pressure	0.3
Density	1
Body Forces	1
Momentum	0.4
Vaporization Mass	1
Vapor	0.2
Turbulent Kinetic Energy	0.4
Turbulent Dissipation Rate	0.4
Turbulent Viscosity	1
Renolds Stresses	0.5

Table 19 Setup discretization for the RSM approach

Variable	Scheme
Pressure-velocity Coupling	SIMPLE
Pressure	Standard
Density	Second Order Upwind
Momentum	Second Order Upwind
Vapor	Second Order Upwind
Turbulent Kinetic Energy	Second Order Upwind
Turbulent Dissipation Rate	Second Order Upwind
Renolds Stresses	Second Order Upwind

Table 20 Setup residual for the RSM approach

Residual	Absolute Criteria
continuity	1.00E-06
x-velocity	1.00E-06
y-velocity	1.00E-06
z-velocity	1.00E-06
k	1.00E-06
epsilon	1.00E-06
uu-stress	1.00E-06
vv-stress	1.00E-06
ww-stress	1.00E-06
uv-stress	1.00E-06
vw-stress	1.00E-06
uw-stress	1.00E-06
vf-vapor	1.00E-06

1.3 The nonlinear turbulence model approach

The numerical simulation settings for simulation of cavitation on marine propeller for the nonlinear turbulence model approach are shown in Tables 25 to 35

Table 21 Model settings for the nonlinear turbulence model approach

Model	Settings
DTMB 4382	3D
Time	Steady
Multiphase	Mixture
Viscous	Standard $k-\omega$ turbulence model
Wall Treatment	Non-equilibrium Wall Treatment
Turbulent Vicosity (User-defined)	Craft et al.
Materials	Water-liquid, Water-vapor
Phase Interaction	Cavitation

Table 22 Types of the boundary conditions in each zone for the nonlinear turbulence model approach

Name	Types
Blade	Wall
Hub	Wall
Inlet	Pressure-Inlet
Outlet	Presure-Outlet
Volume_in	Fluid
Volume_out	Fluid

Table 23 Setup conditions for the nonlinear turbulence model approach

Name	Types	Phase	Wall Motion	Motion	Rotation-Axis Direction
Blade	Wall	Mixture	Moving Wall	Rotational	X=1, Y=0, Z=0
Hub	Wall	Mixture	Moving Wall	Rotational	X=1, Y=0, Z=0

Table 24 Setup motion type for the nonlinear turbulence model approach

Name	Types	Phase	Rotation-Axis Direction	Motion Type	Rotational Velocity Speed (rpm)
Volume_in	Fluid	Mixture	X=1, Y=0, Z=0	Moving Reference Frame	1,200
Volume_out	Fluid	Mixture	X=1, Y=0, Z=0	-	-

Table 25 Setup interface zones for the nonlinear turbulence model approach

Grid Interface	Interface Zone 1	Interface Zone 2
Interface	Interface_in	Interface_out

Table 26 Setup solution controls for the nonlinear turbulence model approach

Equation	Solved
Flow	Yes
Vapor	Yes
Turbulence	Yes

Table 27 Setup relaxation factors for the nonlinear turbulence model approach

Variable	Relaxation Factors
Pressure	0.3
Density	1
Body Forces	1
Momentum	0.4
Vaporization Mass	1
Vapor	0.2
Turbulent Kinetic Energy	0.4
Turbulent Dissipation Rate	0.4
Turbulent Viscosity	1

Table 28 Setup discretization for the nonlinear turbulence model approach

Variable	Scheme
Pressure-velocity Coupling	SIMPLE
Pressure	Standard
Density	Second Order Upwind
Momentum	Second Order Upwind
Vapor	Second Order Upwind
Turbulent Kinetic Energy	Second Order Upwind
Turbulent Dissipation Rate	Second Order Upwind

Table 29 Setup residual for the nonlinear turbulence model approach

Residual	Absolute Criteria
continuity	1.00E-06
x-velocity	1.00E-06
y-velocity	1.00E-06
z-velocity	1.00E-06
k	1.00E-06
epsilon	1.00E-06
vf-vapor	1.00E-06

1.4 The modified nonlinear turbulence model approach

The numerical simulation settings for simulation of cavitation on marine propeller for the modified nonlinear turbulence model approach are shown in Tables 36 to 46

Table 30 Model settings for the nonlinear turbulence model approach

Model	Settings
DTMB 4382	3D
Time	Steady
Multiphase	Mixture
Viscous	Standard $k-\omega$ turbulence model
Wall Treatment	Non-equilibrium Wall Treatment
Turbulent Vicosity (User-defined)	Craft et al.
Materials	Water-liquid, Water-vapor
Phase Interaction	Cavitation

Table 31 Types of the boundary conditions in each zone for the modified nonlinear turbulence model approach

Name	Types
Blade	Wall
Hub	Wall
Inlet	Pressure-Inlet
Outlet	Presure-Outlet
Volume_in	Fluid
Volume_out	Fluid

Table 32 Setup conditions for the modified nonlinear turbulence model approach

Name	Types	Phase	Wall Motion	Motion	Rotation-Axis Direction
Blade	Wall	Mixture	Moving Wall	Rotational	X=1, Y=0, Z=0
Hub	Wall	Mixture	Moving Wall	Rotational	X=1, Y=0, Z=0

Table 33 Setup motion type for the modified nonlinear turbulence model approach

Name	Types	Phase	Rotation-Axis Direction	Motion Type	Rotational Velocity Speed (rpm)
Volume_in	Fluid	Mixture	X=1, Y=0, Z=0	Moving Reference Frame	1,200
Volume_out	Fluid	Mixture	X=1, Y=0, Z=0	-	-

Table 34 Setup interface zones for the modified nonlinear turbulence model approach

Grid Interface	Interface Zone 1	Interface Zone 2
Interface	Interface_in	Interface_out

Table 35 Setup solution controls for the modified nonlinear turbulence model approach

Equation	Solved
Flow	Yes
Vapor	Yes
Turbulence	Yes

Table 36 Setup relaxation factors for the modified nonlinear turbulence model approach

Variable	Relaxation Factors
Pressure	0.3
Density	1
Body Forces	1
Momentum	0.4
Vaporization Mass	1
Vapor	0.2
Turbulent Kinetic Energy	0.4
Turbulent Dissipation Rate	0.4
Turbulent Viscosity	1

Table 37 Setup discretization for the modified nonlinear turbulence model approach

Variable	Scheme
Pressure-velocity Coupling	SIMPLE
Pressure	Standard
Density	Second Order Upwind
Momentum	Second Order Upwind
Vapor	Second Order Upwind
Turbulent Kinetic Energy	Second Order Upwind
Turbulent Dissipation Rate	Second Order Upwind

Table 38 Setup residual for the modified nonlinear turbulence model approach

Residual	Absolute Criteria
continuity	1.00E-06
x-velocity	1.00E-06
y-velocity	1.00E-06
z-velocity	1.00E-06
k	1.00E-06
epsilon	1.00E-06
vf-vapor	1.00E-06

MATERIALS AND METHODS

Materials

1. The Tera Cluster is a part of Thai National Grid Center (TNGC), which contains 2 frontend nodes (1 master, 1 backup) composing of HP DL360 G5, Intel Xeon series 5000, 3.2 GHz, 8 GB Memory and 5 TB Lustre file system on Storage nodes. The 192-compute nodes are connected with Gigabit Ethernet. The 32 of 192 nodes are also connected with Infinband composing of HP DL 140 G3, Intel Xeon series 5000, 3.0 GHz and 4 GB Memory (8 GB in Infinband node). The 4 storage nodes are connected with both Gigabit Ethernet and Infiniband composing of HP DL360 G5, Intel Xeon series 5000, 3.2 GHz, 2 GB Memory and 2 HP MSA500 thus creating 5 TB virtual storage using Lustre file system.

2. The GAMBIT and FLUENT computational fluid dynamics (CFD) software (version 6.3) operating on Windows and Linux systems.

3. The Microsoft Visual C++ language software (version 6.1) operating on Window system.

4. The MATLAB software (version 6.5).

Methods

1. Study the previous research works on the turbulence modeling for cavitation on marine propeller.

2. Find the best linear k - ε turbulence model for cavitation on marine propeller.

3. Apply the nonlinear turbulence model to the simulation of cavitation on marine propeller.

4. Evaluate the performance in predict thrust and torque coefficients of various linear k-ε turbulence models, the Reynolds-stress model (RSM), the Craft et al.'s nonlinear k-ε turbulence model and the modified nonlinear k-ε turbulence model at different cavitation numbers and advance ratios.

5. Find the parallel performance in terms of the relative speed up for the simulation of cavitation on marine propeller.

RESULTS AND DISCUSSIONS

Non-Cavitation on Marine Propeller

The propeller used as the evaluation case in this work is the DTMB4382 propeller. The propeller specifications are presented in Table 39. Give fluid flows entrance the propeller make steady times and rotating around propeller in open water by it motionless with using rotating reference frame. Consider a wide range of advance ratios, $J = V_a / nD$ between $0.384 \le J \le 1.025$. The values of J were varied by increasing and decreasing V_a, while n was kept constant at 7.8 rps where V_a is the inflow speed, n is the number of revolutions per second and D is diameter of propeller. A grid dependence for this case use about 500,000 cells. The computational domain are inlet is 1.5D upstream, where D is the propeller diameter. The exit is 3.5D downstream. Domain_inner have diameter is 1.2D and length is 2.5D. Domain_outer have diameter is 2.5D and length is 5D.

In Figure 4 show the thrust coefficient$\left[K_t\right]$, the torque coefficient$\left[K_q\right]$ and the efficiency$\left[\eta_o\right]$ at different advance ratios$\left[J\right]$. The results are predicted by the linear $k-\omega$ *SST* turbulence model found it agrees is good with the experimental data from the open water. It can see the advance ratio $\left[J\right]$ increases, the thrust coefficient$\left[K_t\right]$ and the torque coefficients$\left[K_q\right]$ decrease. Where thrust coefficient $K_t = \dfrac{Thrust}{\rho n^2 D^4}$, *Thrust* is propeller thrust, ρ is fluid density, n is the number of revolutions per second, D is diameter of propeller and torque coefficient $K_q = \dfrac{Torque}{\rho n^2 D^5}$, *Torque* is propeller torque. It found maximum error less than *10%* for thrust coefficient$\left[K_t\right]$, the torque coefficient$\left[K_q\right]$ and the efficiency$\left[\eta_o\right]$.

Cavitation on Marine Propeller

The propeller used as the evaluation case in this work is the DTMB4382 propeller. The propeller specifications are presented in Table 39 and the flow specifications are presented in Table 40. The computational domain of the propeller set same non-cavitation case and shown in Fig. 5.

Figures 6 to 11 show the thrust coefficient $[K_t]$ and the torque coefficient $[K_q]$ at different cavitation numbers $[\sigma]$ when advance ratio, $J = 0.5, 0.6$ and 0.7 respectively. The results are predicted by various linear turbulence models; the linear $k\text{-}\varepsilon$ turbulence models (standard [1877], RNG [1993] and realizable [1995]) combined with the wall function equations (standard [1974], non-equilibrium [1995] and enhanced [1993]) and the linear $k-\omega$ turbulence models (standard [1998] and SST [1994]). It is found that the results predicted by the $k-\omega$ *SST* turbulence model are the closest to the experimental data from the cavitation tunnel, when compared to the other linear turbulence models. However, it also can be seen that the predicted thrust coefficient $[K_t]$ and the torque coefficient $[K_q]$ are still far from the experiment. Where cavitation numbers $\left[\sigma = \frac{P_\infty - P_v}{0.5\rho V_a^2}\right]$, P_∞ is pressure far upstream, P_v is vapor pressure, ρ is fluid density, V_a is the inflow speed.

The efficiency of the nonlinear turbulence models in predicting the cavitation on a marine propeller is clearly shown in Figs. 12 to 19. The thrust coefficient $[K_t]$ and torque coefficient $[K_q]$ predicted by both Craft et al.'s and modified nonlinear models show the much higher accurate prediction than those predicted by the $k-\omega$ *SST* turbulence model and also by those predicted by the Reynolds-stress model (RSM). However, there is still discrepancy between the simulation and the experiment for the cavitation numbers $[\sigma]$ more than 3.5. The authors believe that the main factor contributing to this discrepancy is that the conditions employed in the

simulation are not exactly the same as the conditions used in the experiment. First of all, the full cavitation is assumed in the simulation that is contrary to the experiment. Moreover, the real flow characteristic on the propeller is not fully turbulent. The simulation, however, uses the turbulence model without taking effect of the transition region on the propeller.

Tables 41 to 44 further show the error percentages of the predicted thrust coefficient $[K_t]$ and the predicted torque coefficient $[K_q]$ by various turbulence models at different cavitation numbers $[\sigma]$ and advance ratios $[J]$. It is clearly seen that the nonlinear turbulence models give the least error percentages for all cases. The contours velocity magnitude (mixture) of the propeller at advance ratio, $J = 0.5$ and cavitation number, $\sigma = 3$ is show in Figs. 20. The contours and the iso-surface of the vapor volume fraction on the tip of the propeller at advance ratio, $J = 0.5$ and cavitation number, $\sigma = 3$ are also shown in Figs. 21 and 22, respectively.

Table 39 Specifications of DTMB4382 propeller

Specifications	Detail
Model name	4382
Numbers of blade	5
Diameter [mm]	304.8
Hub ratio	0.2
Expanded area ratio	0.725
Skew angles [deg]	36
Section mean line	NACA a = 0.8
Section thickness distribution	NACA 66 (modified)
Design advance coefficient [J]	0.889

Table 40 Flow specifications at different cavitation numbers $[\sigma]$ and advance ratios $[J]$

Cases	Cavitation numbers $[\sigma]$	Advance ratios $[J]$	Speed of flow (m/s)	Number of revolutions per minute (rpm)
1	2	0.5	3.048	1200
2	2	0.6	3.657	1200
3	2	0.7	4.267	1200
4	3	0.5	3.048	1200
5	3	0.6	3.657	1200
6	3.5	0.7	4.267	1200
7	5	0.5	3.048	1200
8	5	0.6	3.657	1200
9	5	0.7	4.267	1200

Table 41 Error percentage of the thrust coefficient $[K_t]$ at different cavitation numbers $[\sigma]$ and advance ratios $[J]$ using linear turbulence models (fix curve)

Cases	Cavitation numbers $[\sigma]$	Advance ratios $[J]$	Error percentage $= \left(\frac{\lvert \text{experimental data - computational data} \rvert}{\text{experimental data}} \times 100\% \right)$										
			A	B	C	D	E	F	G	H	I	J	K
1	2	0.5	26.48	26.93	26.75	27.11	27.02	27.29	23.91	27.21	23.73	7.97	6.91
2	2	0.6	27.31	30.19	27.36	26.91	26.91	27.14	27.47	27.37	27.65	25.26	24.06
3	2	0.7	36.09	37.34	36.12	38.21	36.25	35.51	36.01	39.12	36.03	38.47	30.29
4	3	0.5	7.34	16.87	6.87	17.67	17.66	16.97	16.76	17.87	17.02	18.01	15.26
5	3	0.6	36.02	36.06	36.14	35.35	35.16	36.14	35.61	39.23	35.91	33.35	30.81
6	3.5	0.7	35.67	29.73	29.71	29.41	35.56	29.95	35.56	37.59	36.11	40.38	34.74
7	5	0.5	33.02	29.61	32.11	31.93	29.72	29.24	29.98	29.78	29.31	32.47	27.79
8	5	0.6	37.02	38.16	37.41	37.23	37.53	37.95	37.02	37.62	37.27	42.35	34.84
9	5	0.7	36.56	40.12	40.08	39.61	40.01	40.05	39.79	39.91	37.15	40.62	30.93

Table 41 (Continued)

where

A = Standard k-ε : standard wall function

B = Standard k-ε : non-equilibrium wall function

C = Standard k-ε : enhanced wall treatment

D = RNG k-ε : standard wall function

E = RNG k-ε : non-equilibrium wall function

F = RNG k-ε : enhanced wall treatment

G = Realizable k-ε : standard wall function

H = Realizable k-ε : non-equilibrium wall function

I = Realizable k-ε : enhanced wall treatment

J = Standard $k-\omega$

K = SST $k-\omega$

Table 42 Error percentage of the torque coefficient $[K_q]$ at different cavitation numbers $[\sigma]$ and advance ratios $[J]$ using linear turbulence models (fix curve)

Cases	Cavitation numbers $[\sigma]$	Advance ratios $[J]$	Error percentage = $\left(\frac{\lvert\text{experimental data - computational data}\rvert}{\text{experimental data}} \times 100\%\right)$										
			A	B	C	D	E	F	G	H	I	J	K
1	2	0.5	15.34	15.88	16.15	14.86	15.25	15.84	12.75	12.95	13.41	8.91	8.58
2	2	0.6	28.81	27.57	29.34	26.83	25.81	25.75	23.31	29.93	29.51	25.04	23.96
3	2	0.7	37.31	37.19	36.79	38.73	37.51	38.42	38.18	39.14	38.66	36.69	35.34
4	3	0.5	8.37	20.74	3.01	21.09	20.84	20.63	21.05	20.79	20.85	21.73	21.37
5	3	0.6	34.76	34.71	34.52	35.11	34.81	34.91	37.85	32.43	36.57	34.16	32.84
6	3.5	0.7	36.61	37.79	35.15	36.12	38.42	39.96	37.57	35.21	36.19	35.85	33.26
7	5	0.5	31.75	32.66	35.54	30.85	30.69	38.38	29.81	29.67	30.58	31.05	29.91
8	5	0.6	32.78	37.11	32.78	33.52	39.37	33.73	33.09	39.35	3.88	32.53	39.23
9	5	0.7	37.76	38.99	35.33	36.26	35.23	36.91	38.08	37.89	36.74	34.91	33.22

Table 42 (Continued)

where

A = Standard k-ε : standard wall function

B = Standard k-ε : non-equilibrium wall function

C = Standard k-ε : enhanced wall treatment

D = RNG k-ε : standard wall function

E = RNG k-ε : non-equilibrium wall function

F = RNG k-ε : enhanced wall treatment

G = Realizable k-ε : standard wall function

H = Realizable k-ε : non-equilibrium wall function

I = Realizable k-ε : enhanced wall treatment

J = Standard $k-\omega$

K = SST $k-\omega$

Table 43 Error percentage of the thrust coefficient $[K_t]$ at different cavitation numbers $[\sigma]$ and advance ratios $[J]$ using the modified nonlinear turbulence model (fix curve)

Cases	Cavitation numbers $[\sigma]$	Advance ratios $[J]$	Error percentage $= \left(\frac{\lvert \text{experimental data - computational data} \rvert}{\text{experimental data}} \times 100\% \right)$			
			SST $k-\omega$	RSM	Craft et al' nonlinear	Modified nonlinear
1	2	0.5	6.91	8.04	5.68	2.53
2	2	0.6	24.06	23.43	7.02	1.47
3	2	0.7	30.29	28.17	20.8	20.17
4	3	0.5	15.26	16.27	1.44	0.16
5	3	0.6	30.81	32.56	15.64	15.15
6	3.5	0.7	34.74	34.25	29.48	28.47
7	5	0.5	27.79	28.01	26.58	25.81
8	5	0.6	34.84	37.61	27.02	26.18
9	5	0.7	30.93	33.27	29.31	28.31

Table 44 Error percentage of the torque coefficient $[K_q]$ at different cavitation numbers $[\sigma]$ and advance ratios $[J]$ using the modified nonlinear turbulence model (fix curve)

Cases	Cavitation numbers $[\sigma]$	Advance ratios $[J]$	Error percentage $= \left(\frac{\lvert \text{experimental data - computational data} \rvert}{\text{experimental data}} \times 100\% \right)$			
			SST $k-\omega$	RSM	Craft et al' nonlinear	Modified nonlinear
1	2	0.5	8.58	8.34	8.22	0.66
2	2	0.6	23.96	25.57	15.04	9.81
3	2	0.7	35.34	34.61	26.77	22.41
4	3	0.5	21.37	23.43	10.35	8.56
5	3	0.6	32.84	33.47	26.05	20.43
6	3.5	0.7	33.26	33.55	27.82	25.76
7	5	0.5	29.91	30.34	26.03	22.57
8	5	0.6	39.23	39.46	31.61	25.82
9	5	0.7	33.22	33.39	27.91	25.66

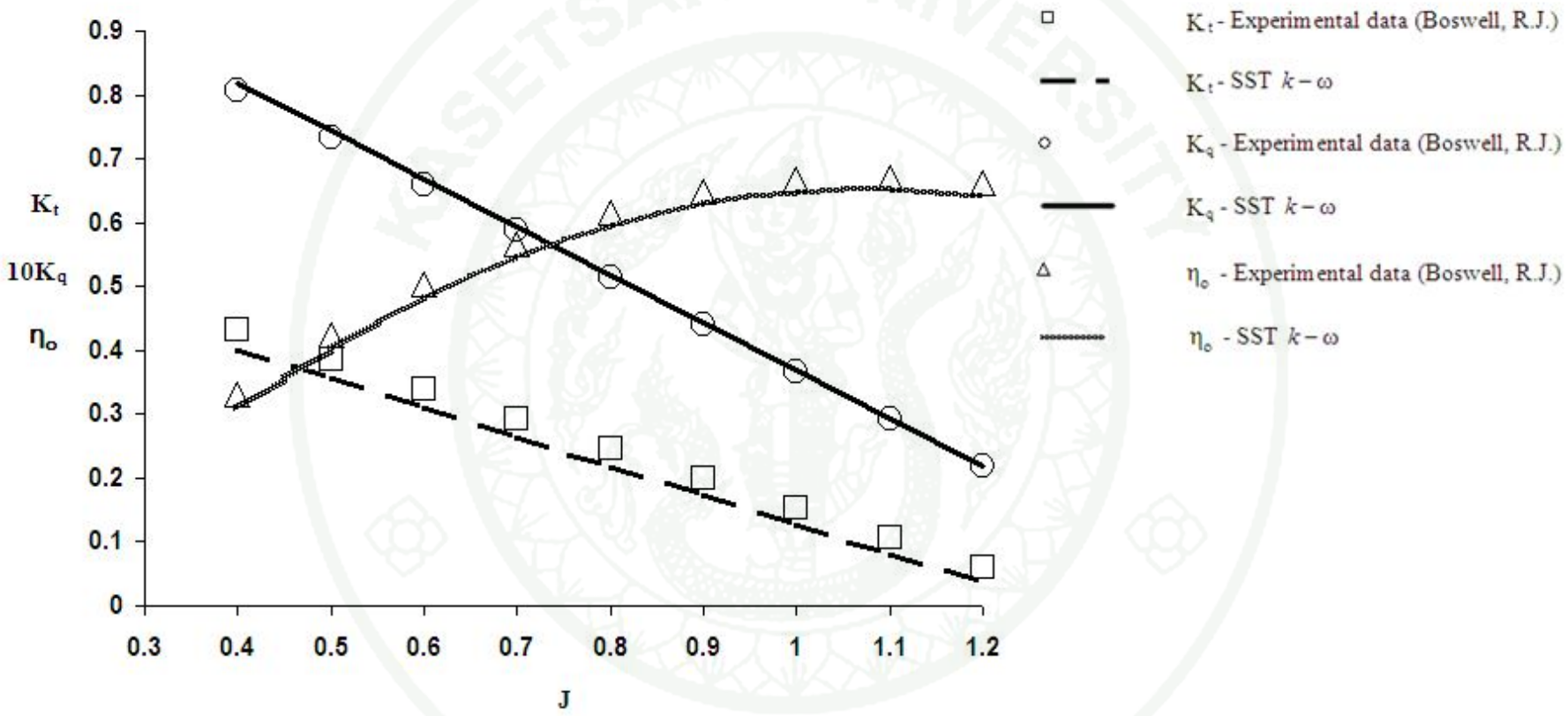


Figure 4 Thrust coefficient $[K_t]$, torque coefficient $[K_q]$ and efficiency $[\eta_o]$ versus advance ratio $[J]$ for the non-cavitation cases. (fix curve)

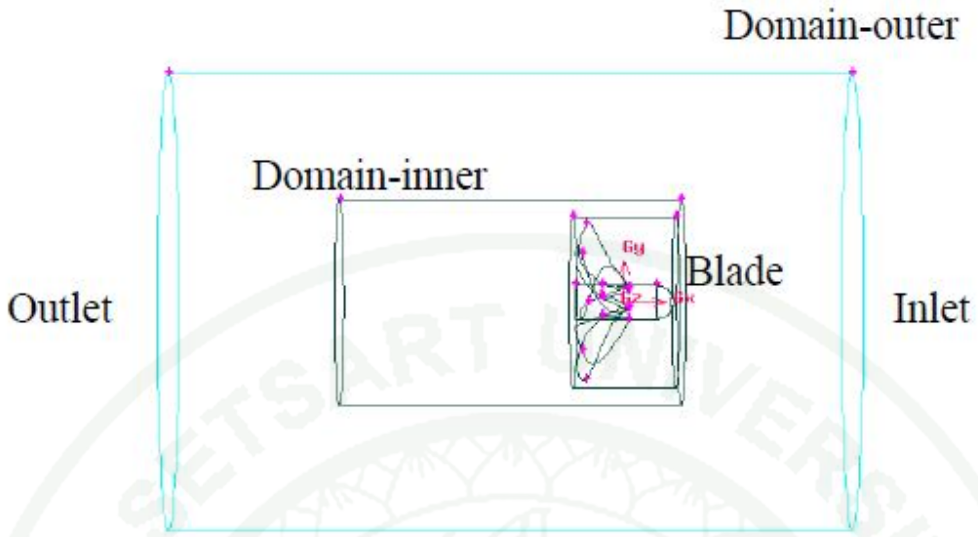


Figure 5 Computational domain of the propeller

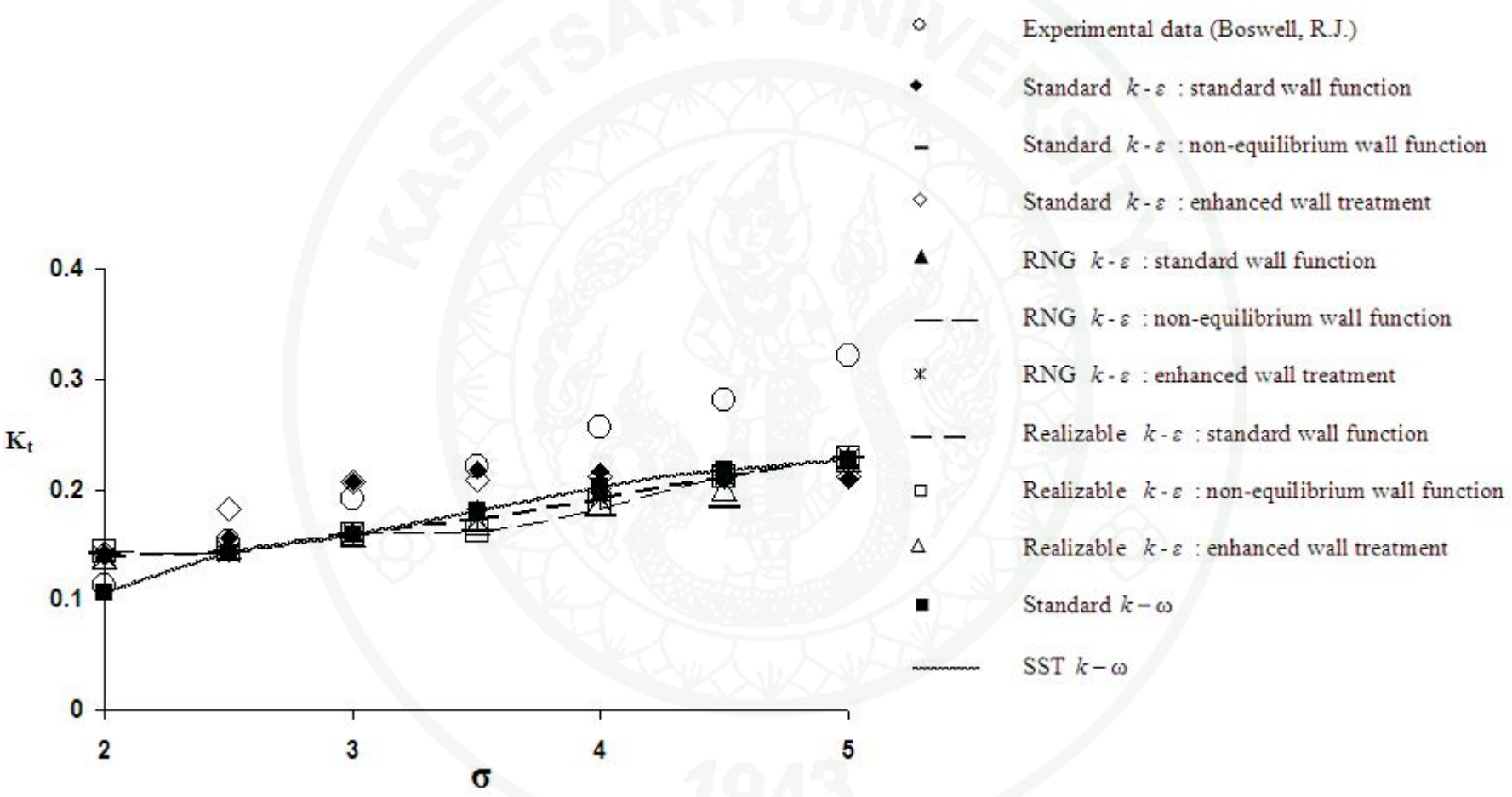


Figure 6 Thrust coefficient $[K_t]$ versus cavitation number $[\sigma]$ at advance ratio $J = 0.5$ using linear turbulence models

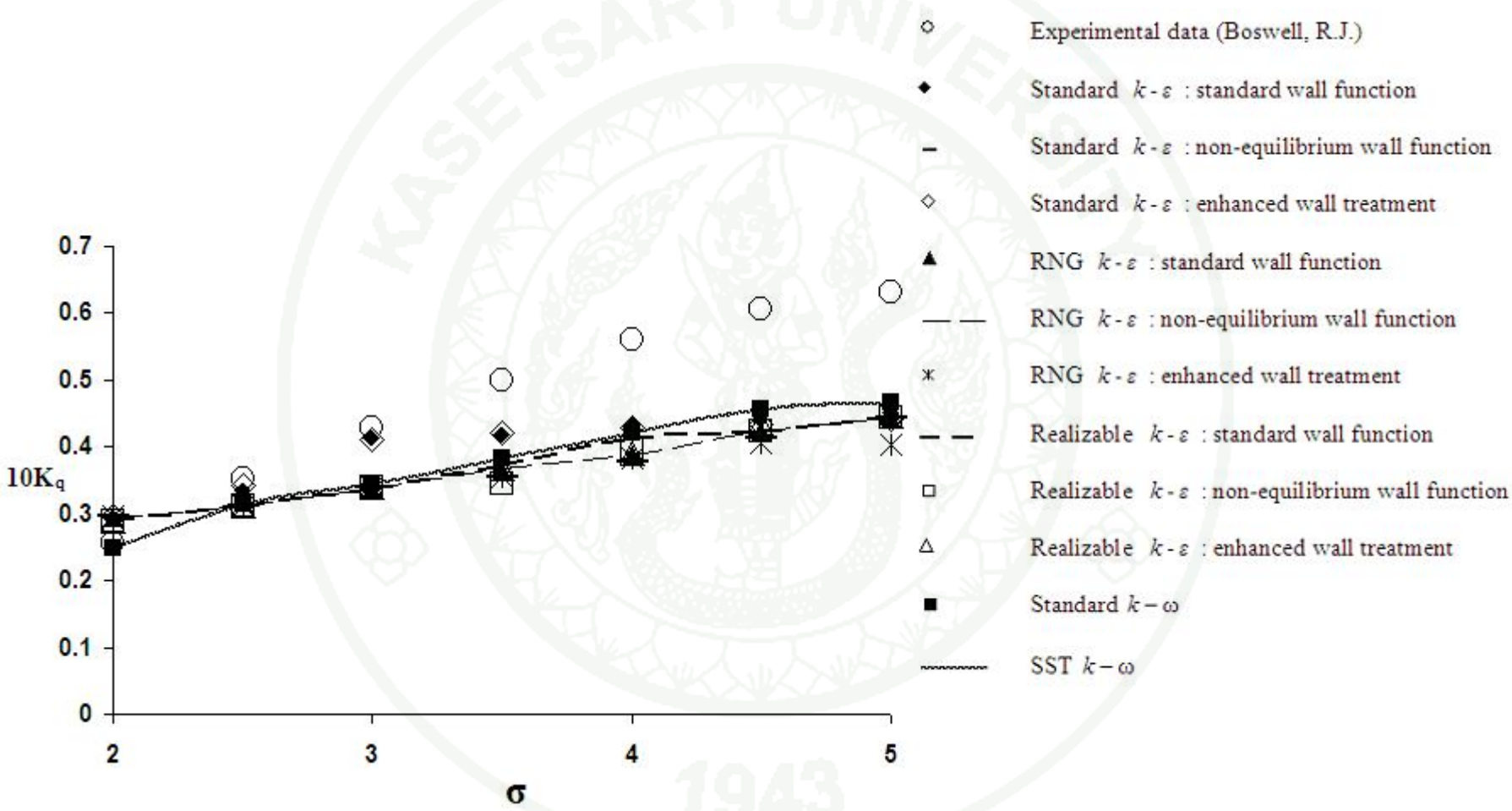


Figure 7 Torque coefficient $\left[K_q\right]$ versus cavitation number $\left[\sigma\right]$ at advance ratio $J = 0.5$ using linear turbulence models

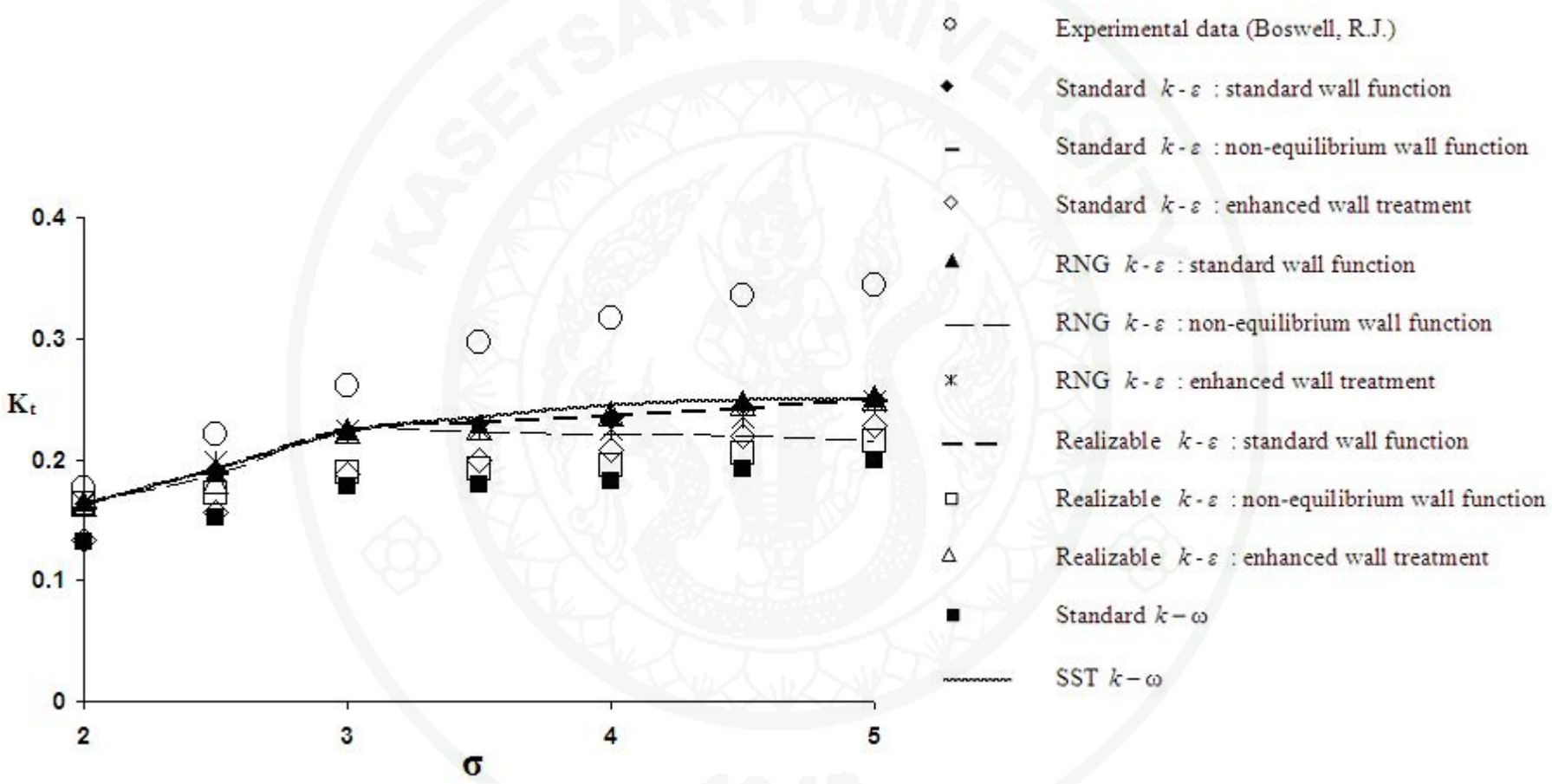


Figure 8 Thrust coefficient $[K_t]$ versus cavitation number $[\sigma]$ at advance ratio $J = 0.6$ using linear turbulence models

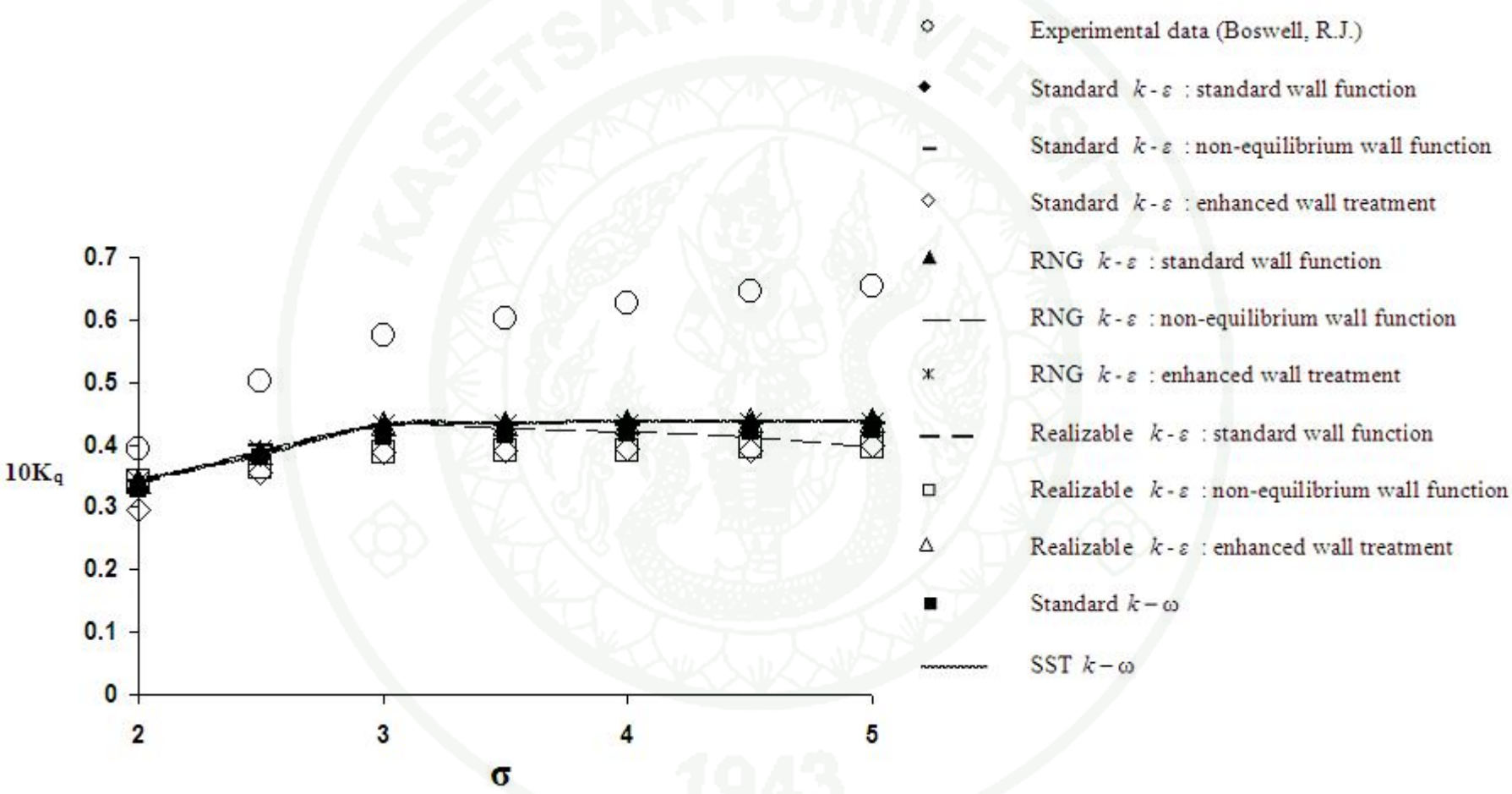


Figure 9 Torque coefficient $\left[K_q\right]$ versus cavitation number $\left[\sigma\right]$ at advance ratio $J = 0.6$ using linear turbulence models

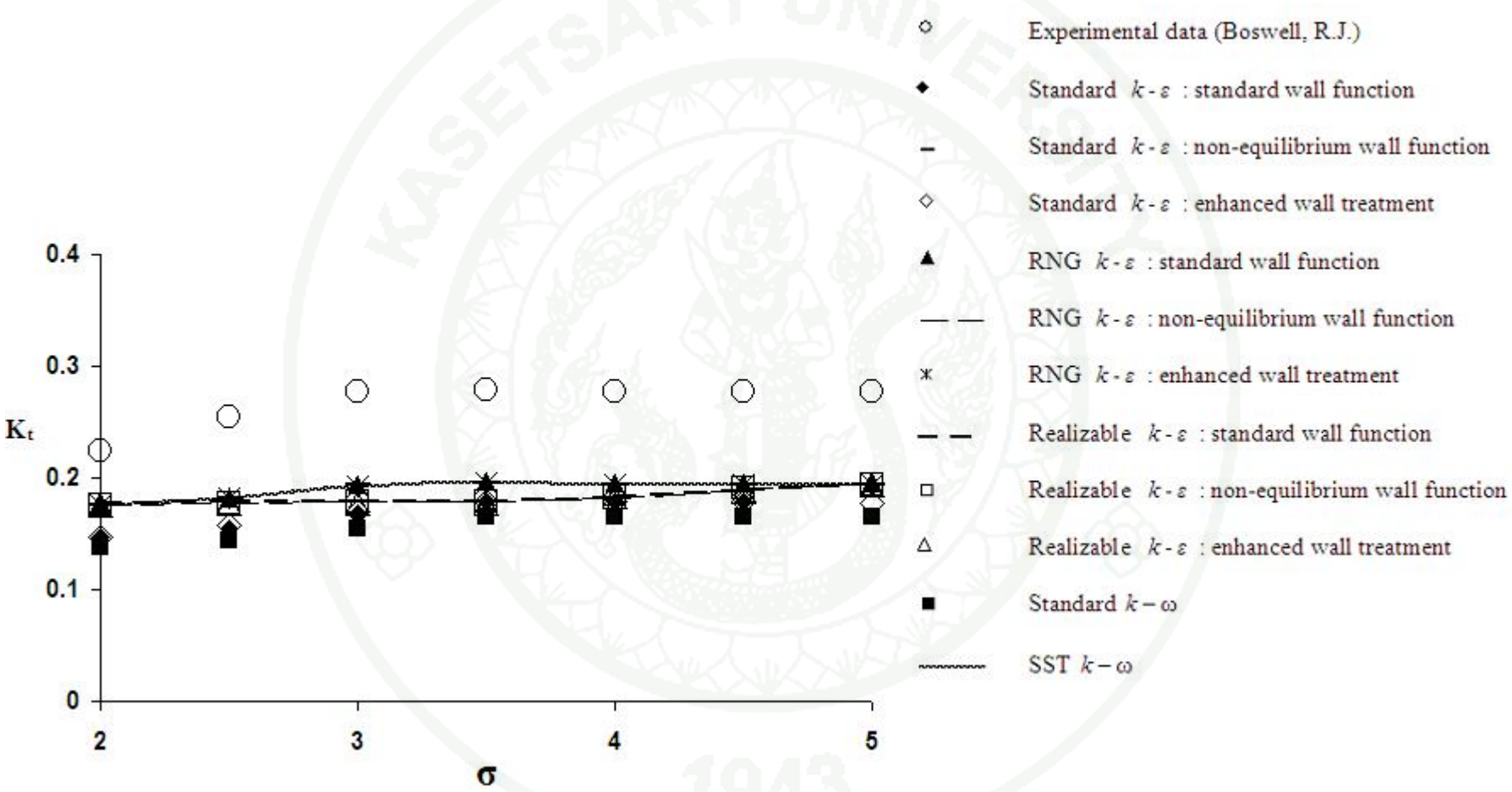


Figure 10 Thrust coefficient $[K_t]$ versus cavitation number $[\sigma]$ at advance ratio $J = 0.7$ using linear turbulence models

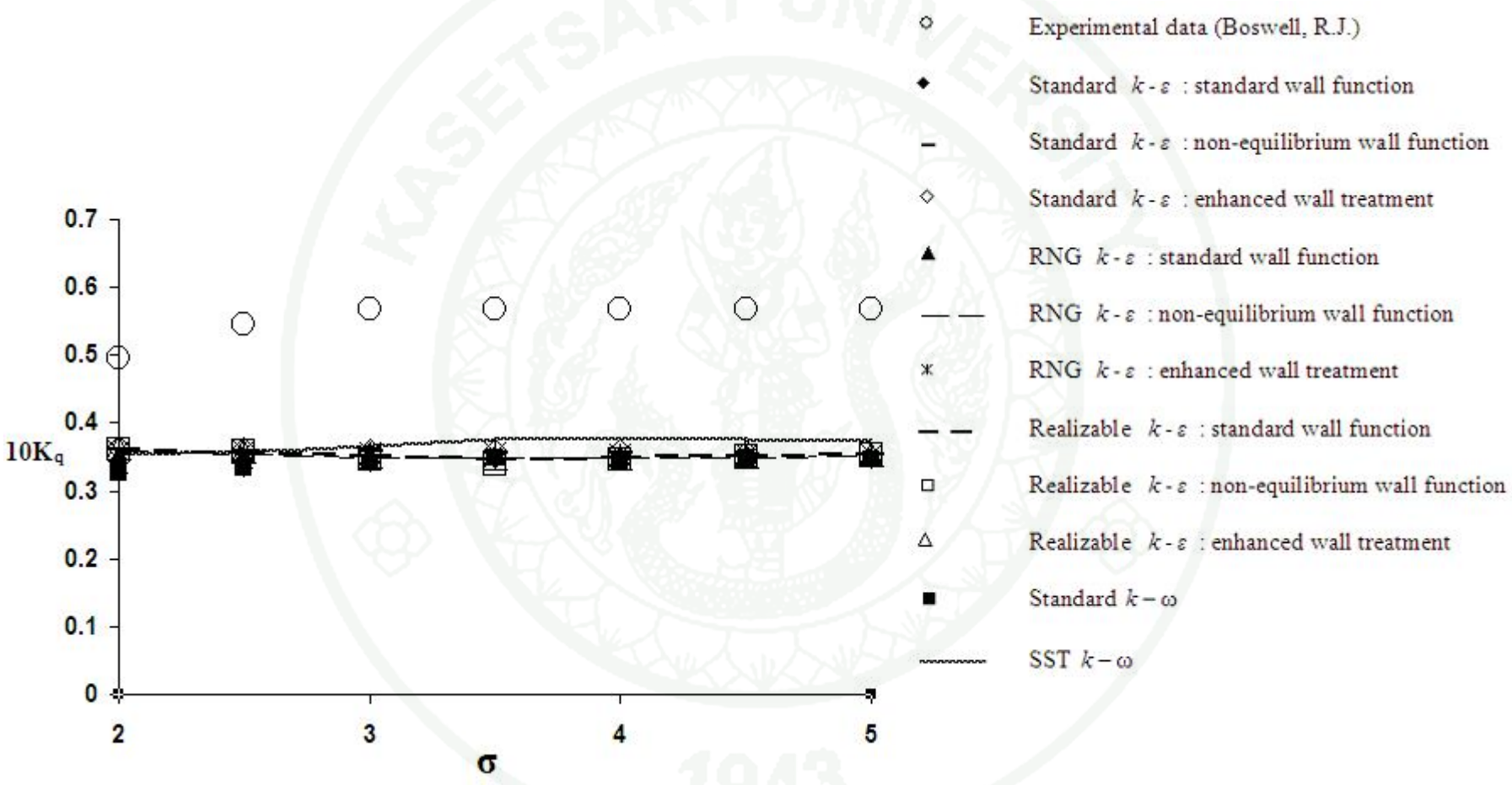


Figure 11 Torque coefficient $\left[K_q\right]$ versus cavitation number $\left[\sigma\right]$ at advance ratio $J = 0.7$ using linear turbulence models

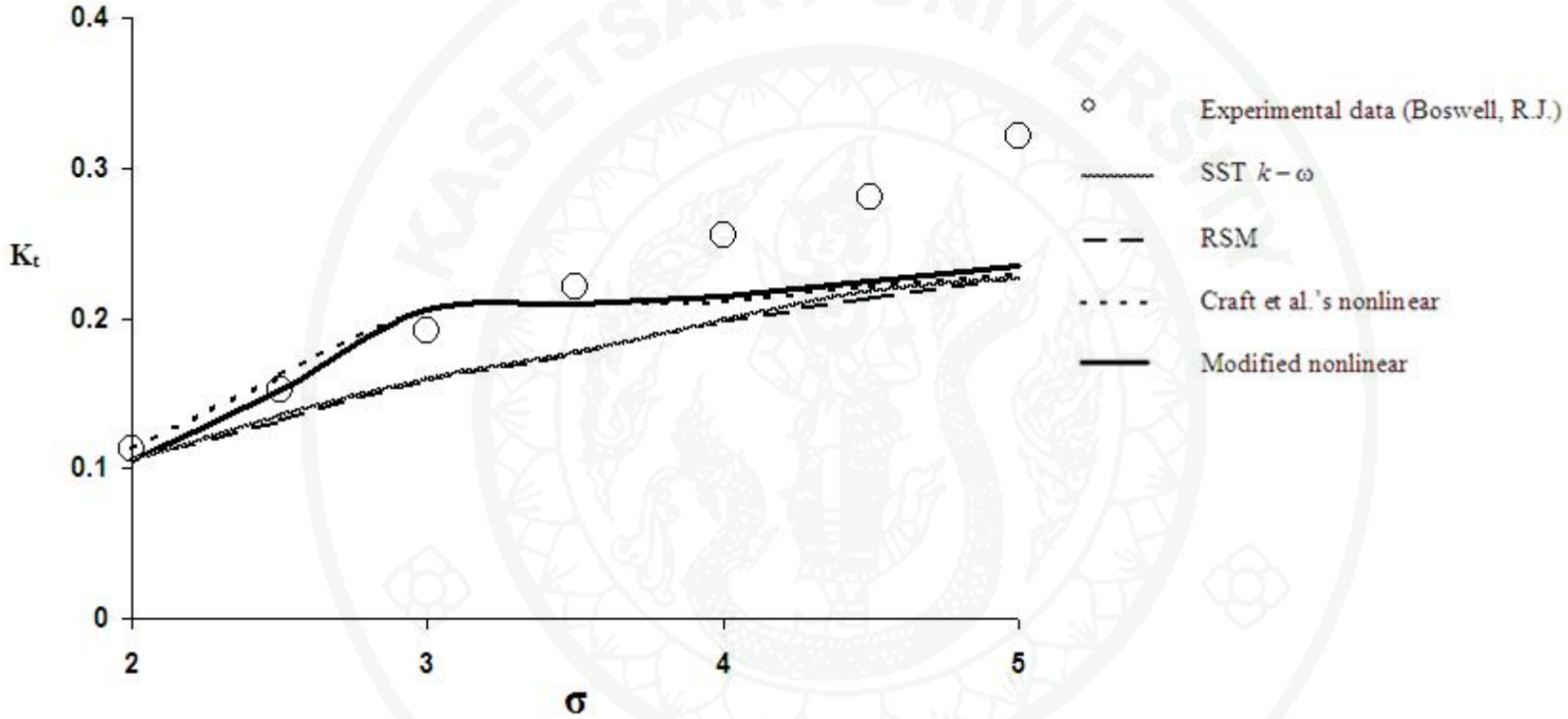


Figure 12 Thrust coefficient $[K_t]$ versus cavitation number $[\sigma]$ at advance ratio $J = 0.5$ using the modified nonlinear turbulence model

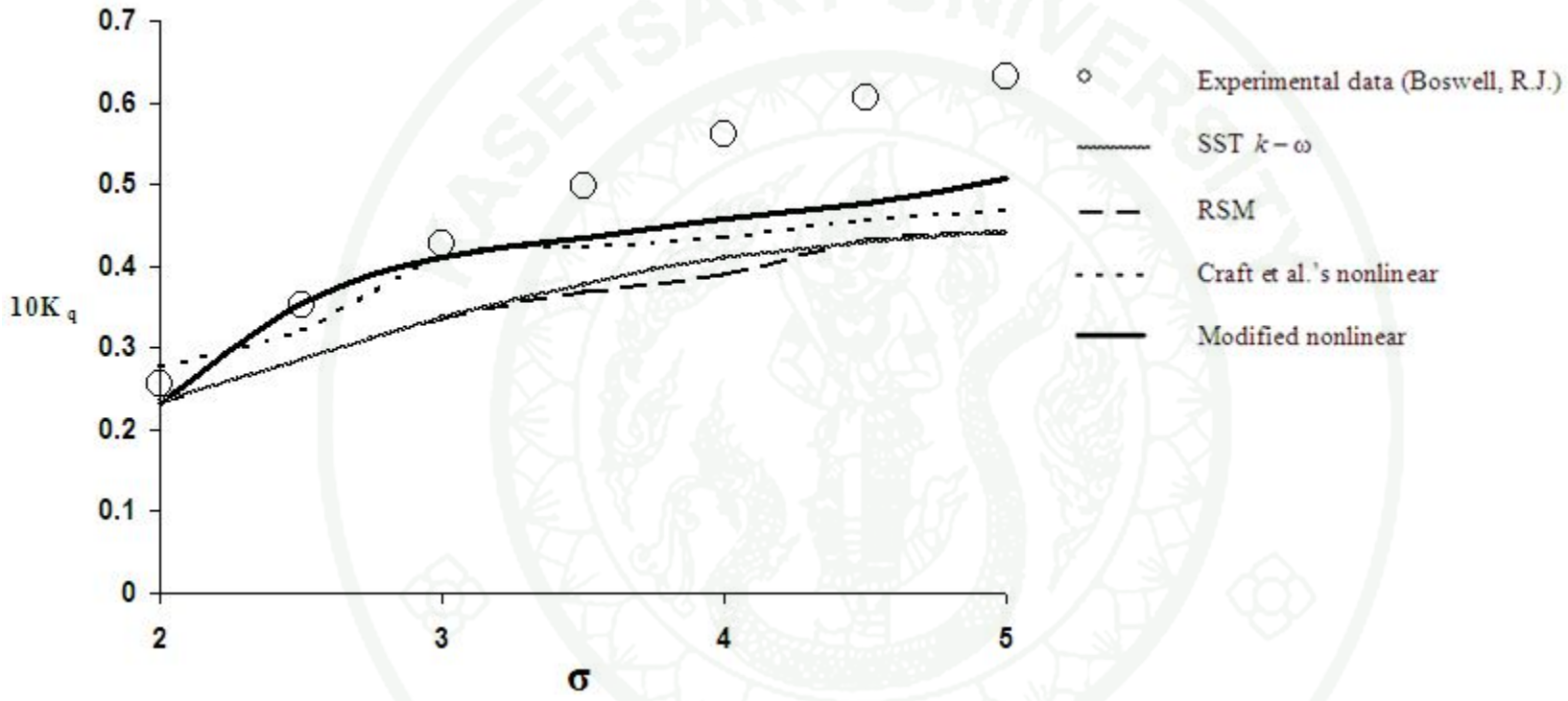


Figure 13 Torque coefficient $\left[K_q\right]$ versus cavitation number $\left[\sigma\right]$ at advance ratio $J = 0.5$ using the modified nonlinear turbulence model

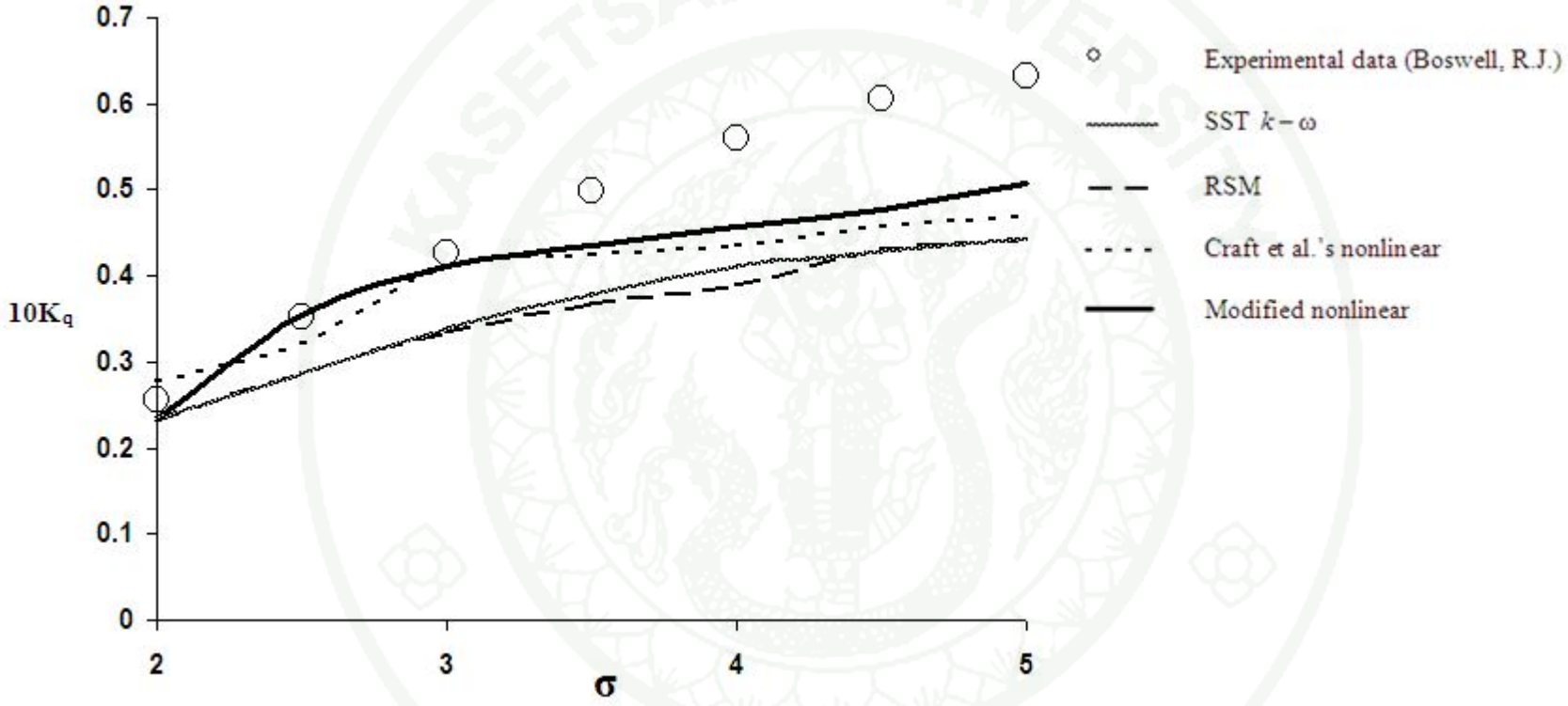


Figure 14 Thrust coefficient $[K_t]$ versus cavitation number $[\sigma]$ at advance ratio $J = 0.6$ using the modified nonlinear turbulence model

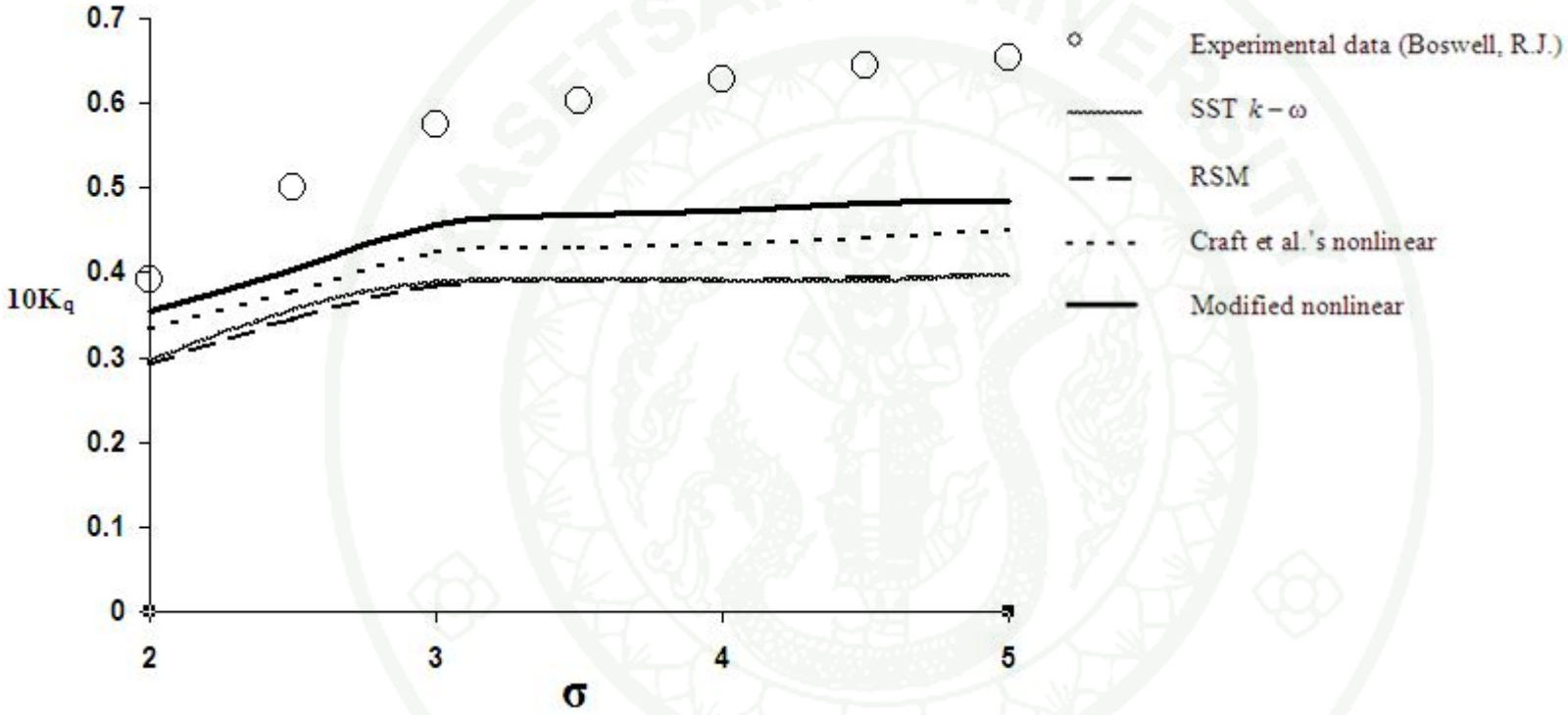


Figure 15 Torque coefficient $\left[K_q\right]$ versus cavitation number $\left[\sigma\right]$ at advance ratio $J = 0.6$ using the modified nonlinear turbulence model

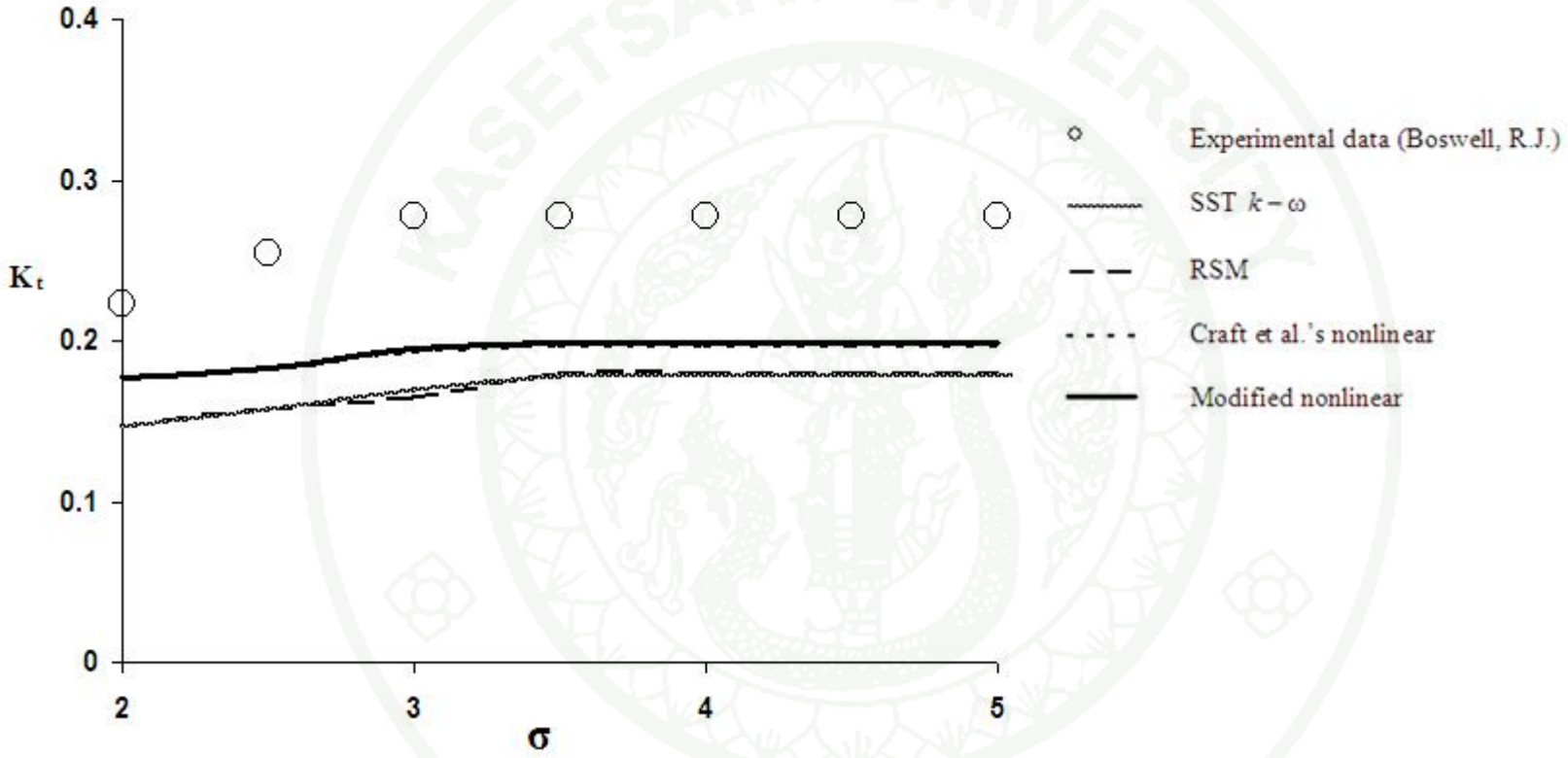


Figure 16 Thrust coefficient $[K_t]$ versus cavitation number $[\sigma]$ at advance ratio $J = 0.7$ using the modified nonlinear turbulence model

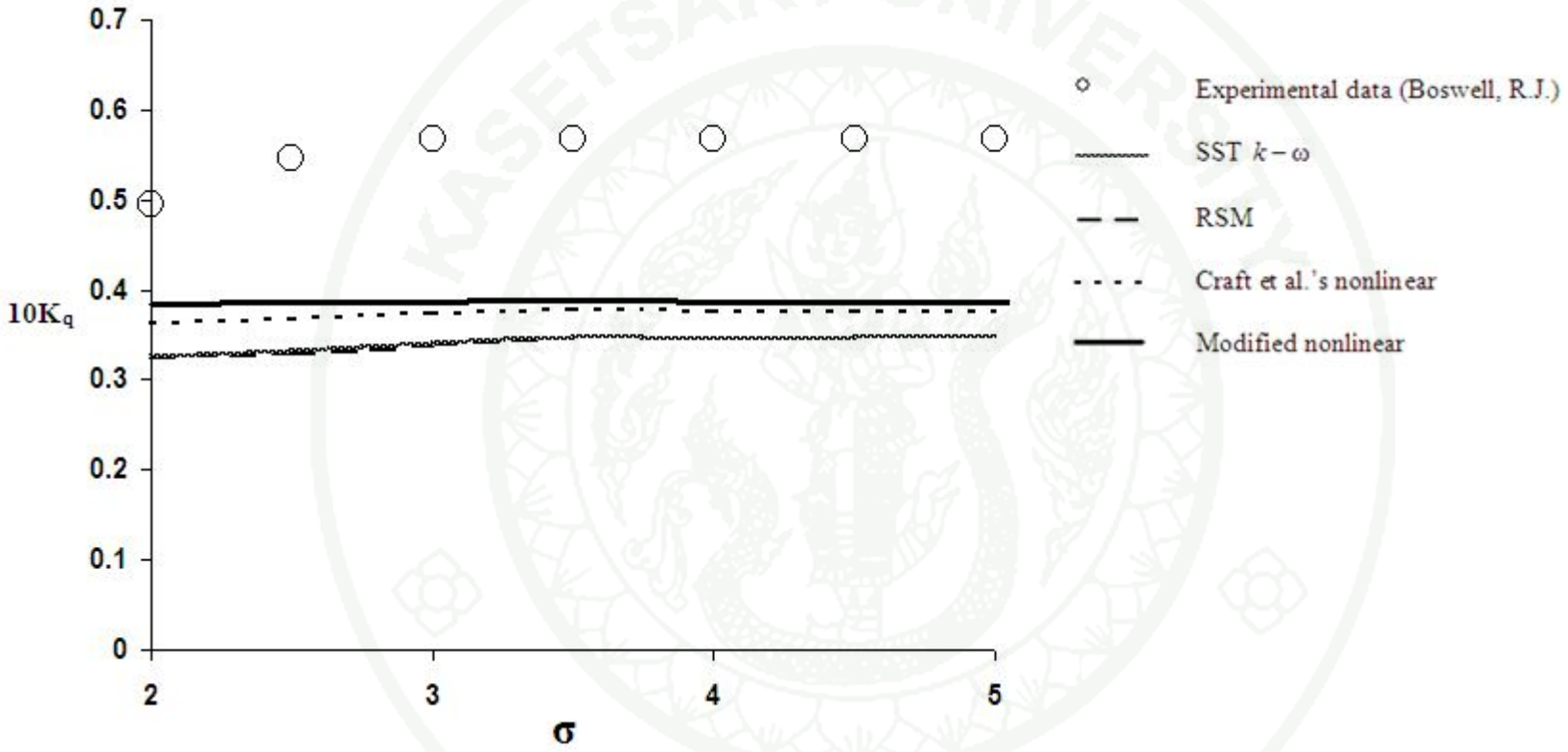


Figure 17 Torque coefficient $\left[K_q\right]$ versus cavitation number $\left[\sigma\right]$ at advance ratio $J = 0.7$ using the modified nonlinear turbulence model

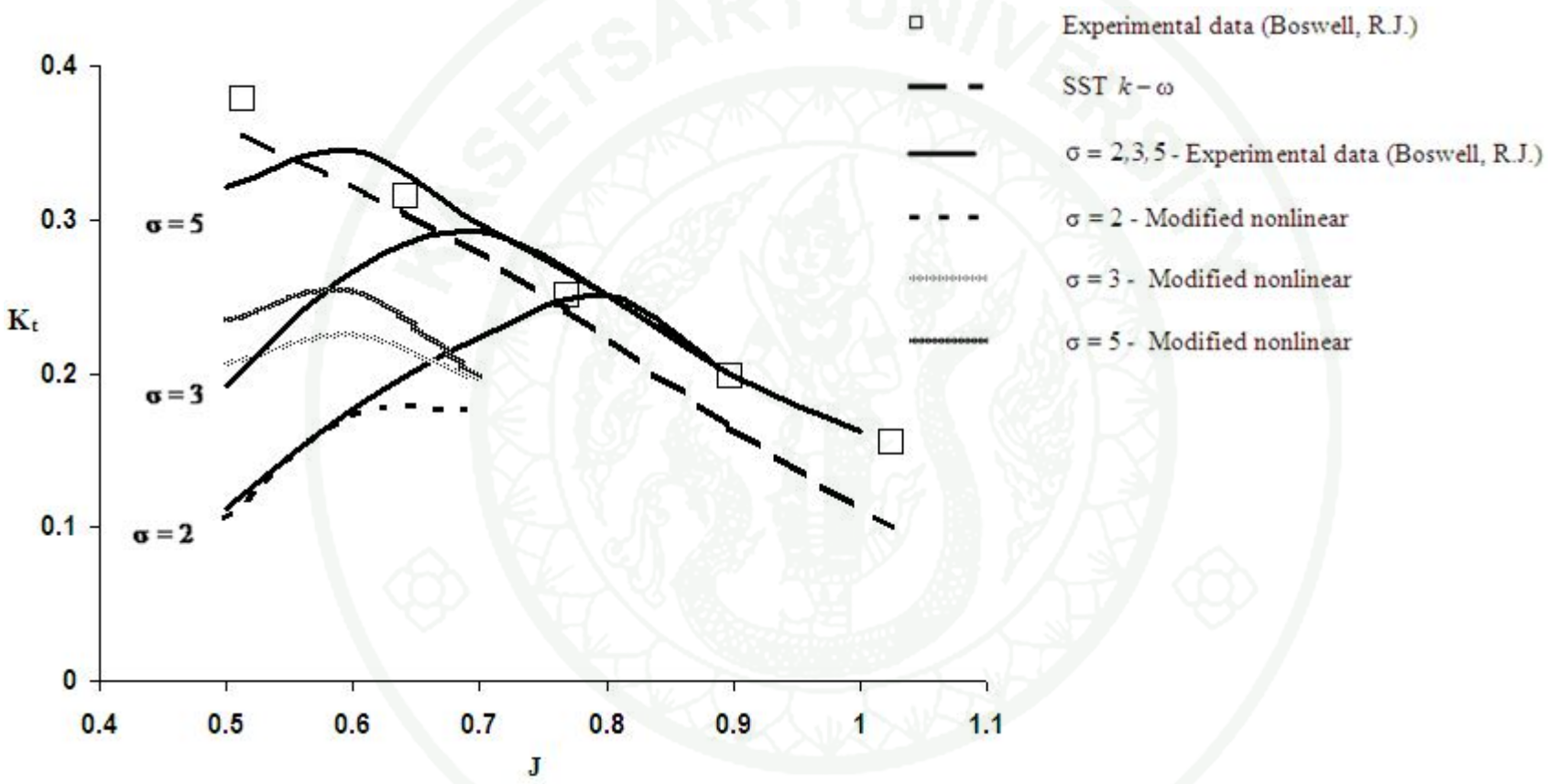


Figure 18 Comparison: Thrust coefficient $[K_t]$ versus advance ratio $[J]$ at different cavitation numbers $[\sigma]$ using the modified nonlinear turbulence model in cavitation cases and using the $SST\ k-\omega$ turbulence model in non-cavitation

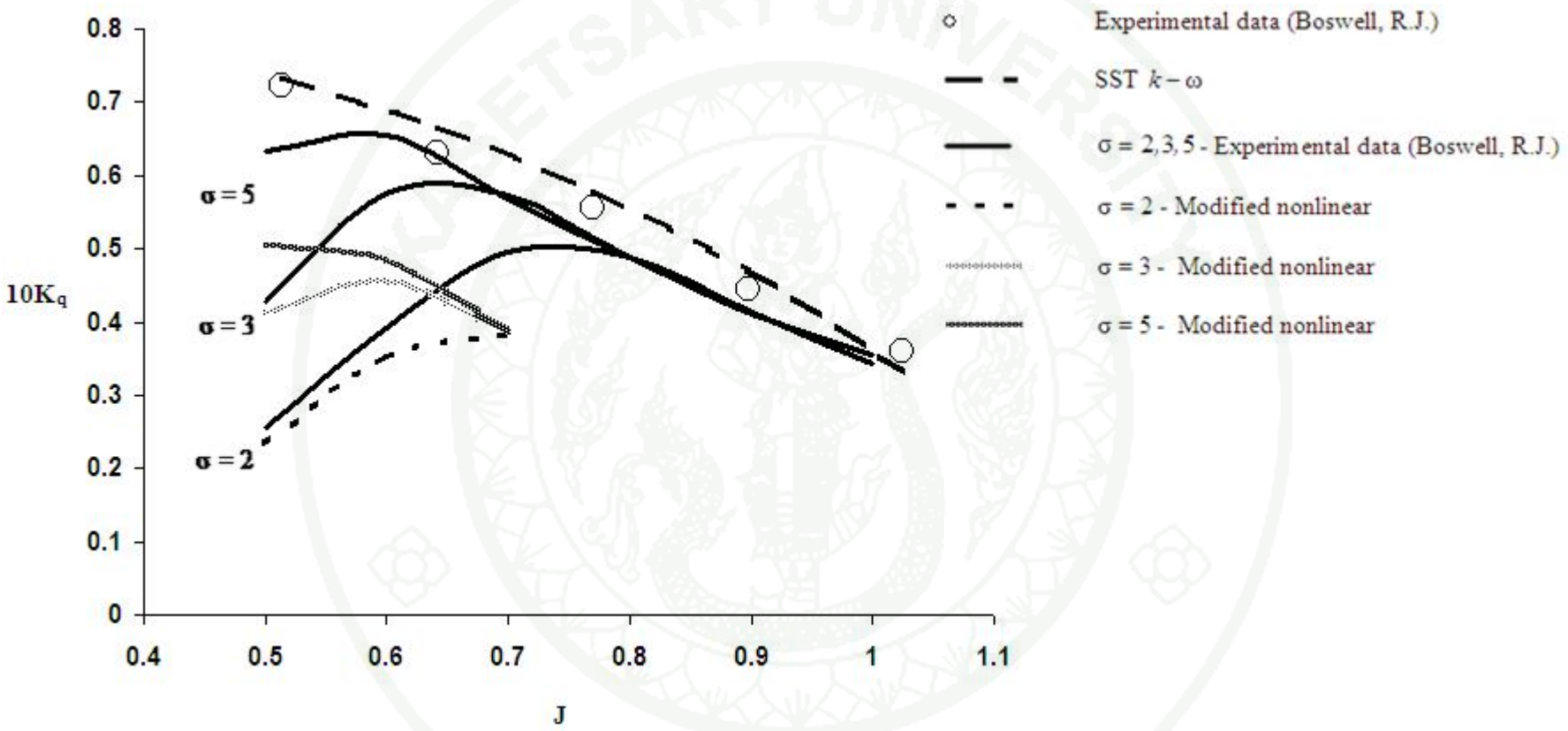


Figure 19 Comparison: Torque coefficient $[K_q]$ versus advance ratio $[J]$ at different cavitation numbers $[\sigma]$ using the modified nonlinear turbulence model in cavitation cases and using the $SST\ k-\omega$ turbulence model in non-cavitation

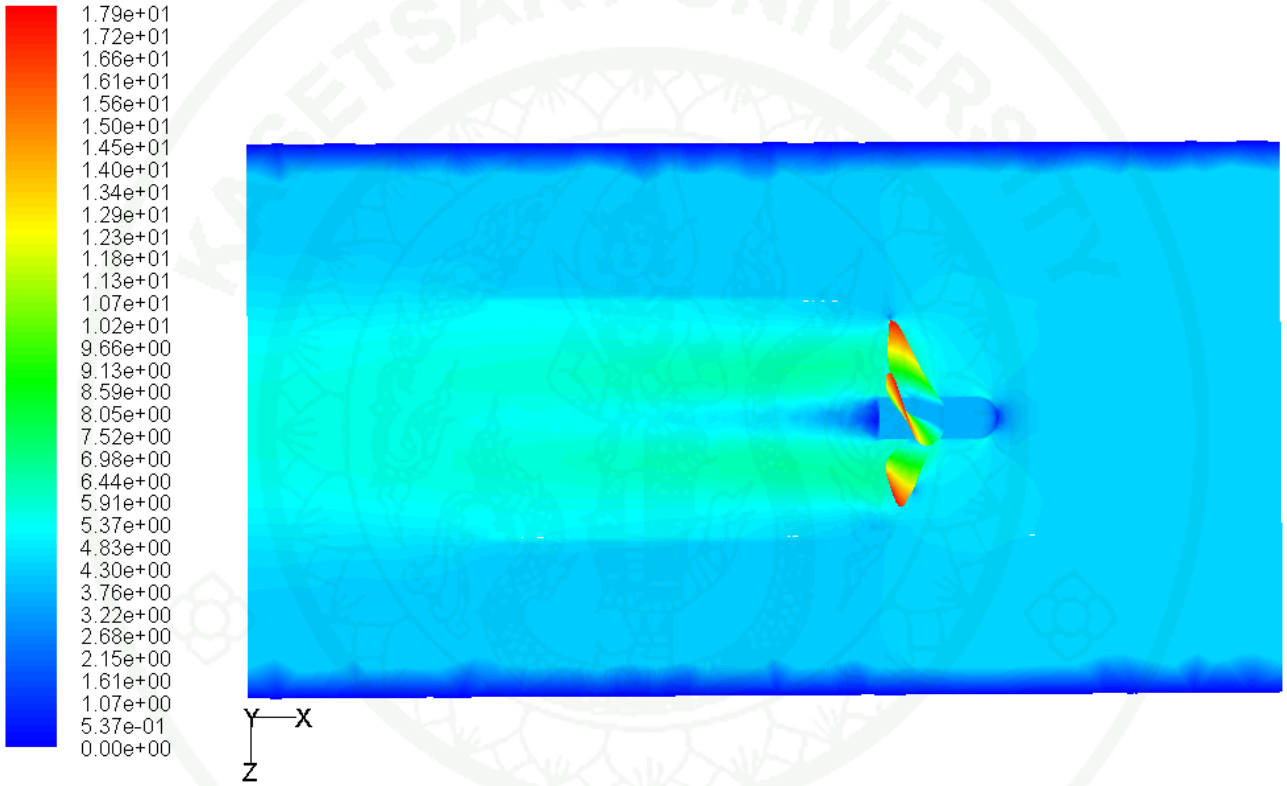


Figure 20 Contours of velocity magnitude (mixture) at cavitation number $\sigma = 3$ and advance ratio $J = 0.5$

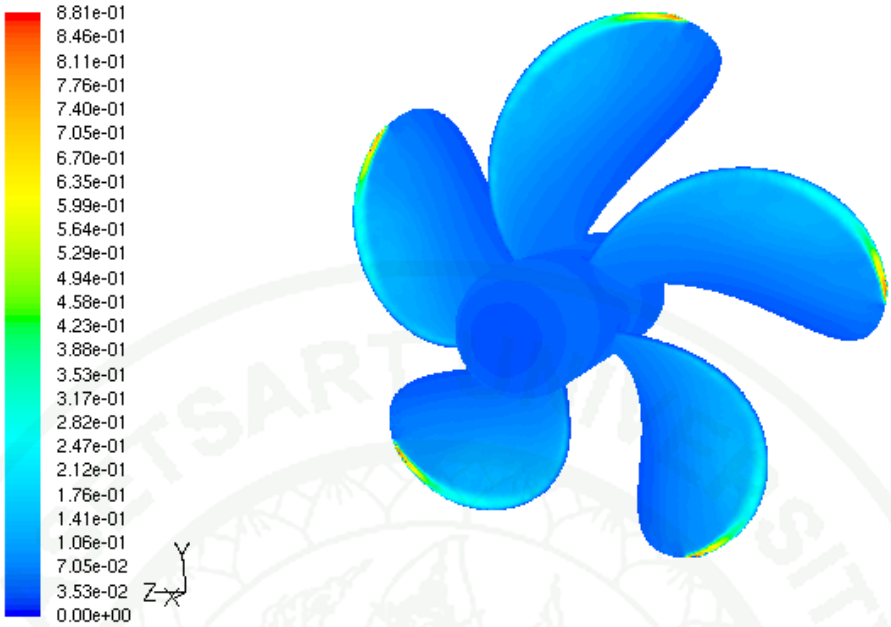


Figure 21 Vapor volume fraction contours at cavitation number $\sigma = 3$ and advance ratio $J = 0.5$

Figure 22 Cavitation on the propeller at cavitation number $\sigma = 3$ and advance ratio $J = 0.5$

CONCLUSIONS

The performance of various linear turbulence models, the Reynolds-stress model (RSM), the nonlinear turbulence model of Craft et al. and the modified nonlinear turbulence model in predicting the combined effects of turbulence, cavitation, complex geometry and multiphase phenomena occurred on the marine propeller at various cavitation numbers and advance ratios is assessed in detail. It is found that the results predicted by the $k-\omega$ SST turbulence model are the closest to the experimental data compared to the other linear models. However, the nonlinear turbulence models show the much higher accurate prediction than the linear models. Furthermore, the nonlinear turbulence models can predict the considered flow closer to the experimental data than the more complicated Reynolds-stress model (RSM). It can be concluded that the nonlinear turbulence models are the most suitable turbulence model for the cavitation prediction of a marine propeller.

The nonlinear turbulence models used in predict for find tools at appropriate for cavitation on marine propeller and it use modify marine propeller in future work. In design marine propeller should consider cavitation everytime because it affects will minus efficiency marine propeller. Should design marine propeller has ogival section at tip blade, use RPM are down and devalue pitch will help abate the occurrence cavitation.

LITERATURE CITED

Abe, K., Y.J. Jang and M.A. Leschziner. 2003. An Investigation of Wall-Anisotropy Expression and Length-Scale Equations for Non-Linear Eddy-Viscosity Models. **International Journal of Heat and Fluid Flow**, 24: 181-198.

Apsley, D.D. and M.A. Leschziner. 1998. A New Low-Reynolds-Number Nonlinear Two-Equation Turbulence Model for Complex Flows. **International Journal of Heat and Fluid Flow** 19: 209-222.

Boswell, R.J. 1971. Design Cavitation Performance and Open Water Performance of a Series of Research Skewed Propeller. **NSRDC Rep. No.3339, March**.

Boussinesq, J. 1877. Theory de L'eco Reynolds stresses element Tourbillant. **Memoires Presentes Par Divers Savants Sciences Mathematique at Physiques** 46-50.

Choudhury, D. 1993. **Introductuon to the Renormalization Group Method and Turbulence Modeling**. Fluent Inc., U.S.A.

Craft, T.J., B.E. Launder and K. Suga. 1996. Development and Application of a Cubic Eddy-Viscosity Model of Turbulence. **International Journal of Heat and Fluid Flow** 17: 108-11.

Deshpande, M., J. Feng and C.L. Merkle. 1994. Cavity Flow Predictions Based on the Euler Equations. **ASME Journal Fluids Engineering** 116(1): 36-44.

Fluent. 2006. **Fluent 6.3 Documentation**. Fluent Inc., U.S.A.

Gururatana, S., V. Juttijudata, E. Juntasaro and V. Juntasaro. 2006. **Prediction of 3D Turbulence Induced Secondary Flows in Rotating Square Duct**. Seoul National University, Seoul, Korea.

Juntasaro, V., J. Buranarote, S. Gururatana and E. Juntasaro. 2005. A New Reynolds-Stress Expression Based on DNS Data in Non-Liner Eddy-Viscosity Turbulence Model For Complex Flows. **The Fourth International Symposium on Turbulence and Shear Flow Phenomena (TSFP 4), June 27-29**. Virginia, U.S.A.

Juntasaro, V., P. Dechaumphai and A.J. Marquis. 2005. Evaluation of the Damping Functions in Low-Reynolds-Number Non-Linear q- Turbulence Model. **International Journal of Computational Fluid Dynamics** 19 (3): 225-234.

Kader, B. 1993. Temperature and Concentration Profiles in Fully Turbulent Boundary Layers. **International Journal of Heat and Mass Transfer** 541-1544.

Kim, S.E. and D. Choudhury. 1995. **A Near-Wall Treatment Using Wall Functions Sensitized to Pressure Gradient**. ASME FED, U.S.A.

Kinnas, S.A. and N.E. Fine. 1993. A Numerical Nonlinear Analysis of the Flow Around Two- and Three-Dimensional Partially Cavitating Hydrofoils **Journal Fluid Mechanics** 254: 151-181.

Kinnas, S.A., J.K. Choi, H. Lee, Y.L. Young, H. Gu, K. Kakar and S. Natarajan. 2002. Prediction of Cavitation Performance of Single or Multi-Component Propulsors and their Interaction with the Hull. **Trans. Soc. Naval Architects and Marine Engineers** 110: 215-244.

Launder, B.E. and N. Sharma. 1974. Application of the Energy-Dissipation Model of Turbulence to the Calculation of Flow near a Spinning Disc. **Letters in Heat and Mass Transfer** 1(2): 131-138.

_____ and D.B. Spalding. 1974. **The Numerical Computation of Turbulent Flows**. Computer Methods in Applied Mechanics and Engineering, U.S.A.

Lien, F.S., W.L. Lien and M.A. Leschziner. 1996. Low-Reynolds-Number Eddy-Viscosity Modeling Based on Non-linear Stress-Strain/Vorticity Relations. **Engineering Turbulence Modeling and Experiments** 3 (1): 91-100.

Lifante, C. and T. Frank. 2008. I**nvestigation of Pressure Fluctuations caused by Turbulent and Cavitating Flow around a P1356 Ship Propeller**. NAFEMS Seminar: Simulation komplexer Strömungsvorgänge (CFD).

Myong, H.K. and N. Kasagi. 1990. Prediction of Anisotropy of the Near-Wall Turbulence with an Anisotropic Low-Reynolds-Number $k-\varepsilon$ Turbulence Model. **Journal of Fluids Engineering** 112: 521-524.

Menter, F.R. 1994. Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications. **AIAA Journal** 32: 1598-1605.

Nisizima, S. and A. Yoshizawa. 1987. Turbulent Channel and Couette Flows using an Anisotropic $k-\varepsilon$ Model. **AIAA Journal** 25: 414-420.

Rhee, S.H., T. Kawamura and H. Li. 2004. **Propeller Cavitation Study Using an Unstructured Grid Based Navier-Stokes Solver**. International Symposium on Cavitation CAV2006, Wageningen, The Netherlands.

Rubinstein, R. and J.M. Barton. 1990. Nonlinear Reynolds Stress Models and the Renormalization Group. **Physics of Fluids** 2: 1472-1476.

Singhal, A.K., M.M. Athavale, H.Y. Li and Y. Jiang. 2002. Mathematical Basis and Validation of the Full Cavitation Model. **ASME Journal Fluids Engineering** 124(3): 617-624.

Shih, T.H., J. Zhu and J.L. Lumley. 1993. A Realizable Reynolds Stress Algebraic Equation Model. **NASA Technical Memorandum 105993**.

Shih, T.H., W.W. Liou, A. Shabbir, Z. Yang and J. Zhu. 1995. A New $k-\varepsilon$ Eddy-Viscosity Model for High Reynolds Number Turbulent Flow. **Computers and Fluids** 24 (3): 227-238.

Speziale, C.G. 1987. On Nonlinear $k-l$ and $k-\varepsilon$ Models of Turbulence. **Journal of Fluid Mechanics** 178: 459-475.

Tongkratoke, A., C. Chinnarasri, A. Pornprommin, P. Dechaumphai and V. Juntasaro. 2009. Nonlinear Turbulence Models for Multiphase Recirculating Free-surface Flow over Stepped Spillways. **International Journal of Computational Fluid Dynamics** 23: 401-409.

Wilcox, D.C. 1998. Turbulence Modeling for CFD. DCW Industries, Inc. La Canada, California.

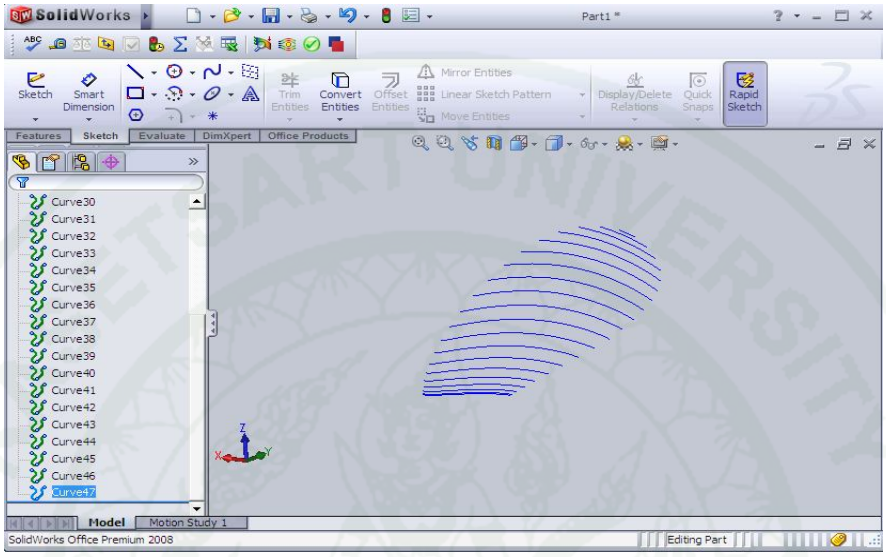
APPENDICES

Appendix A

Using SOLIDWORKS and GAMBIT on Windows System

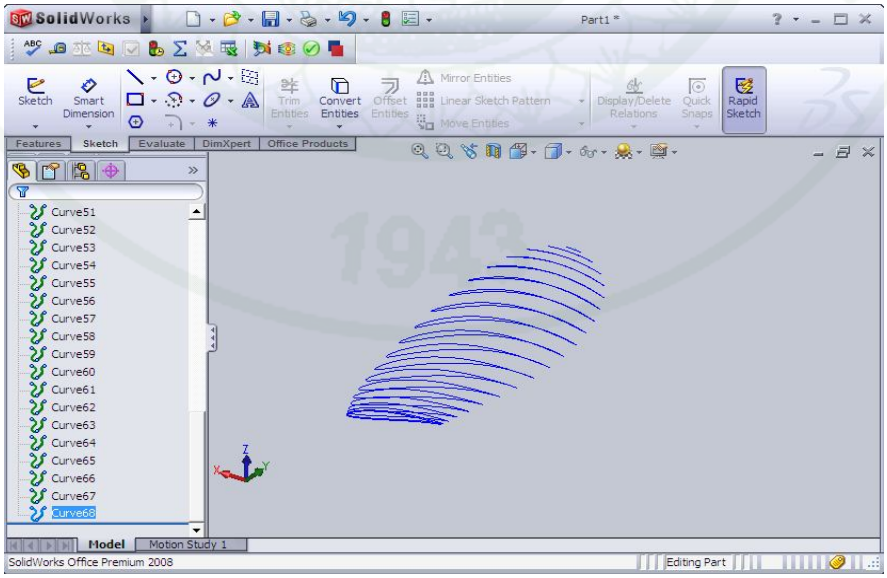
Using SOLIDWORKS on Windows System

1. Insert → Curve → Curve Through XYZ Points → Browse → Pressure.txt → OK



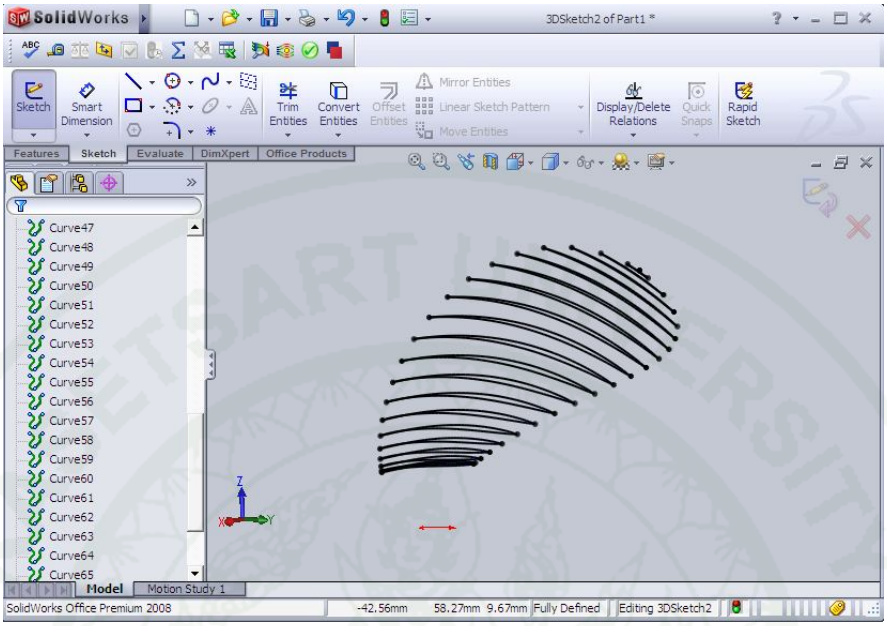
Appendix Figure A1 Procedure of using SOLIDWORKS step 1

2. Insert → Curve → Curve Through XYZ Points → Browse → Suction.txt → OK



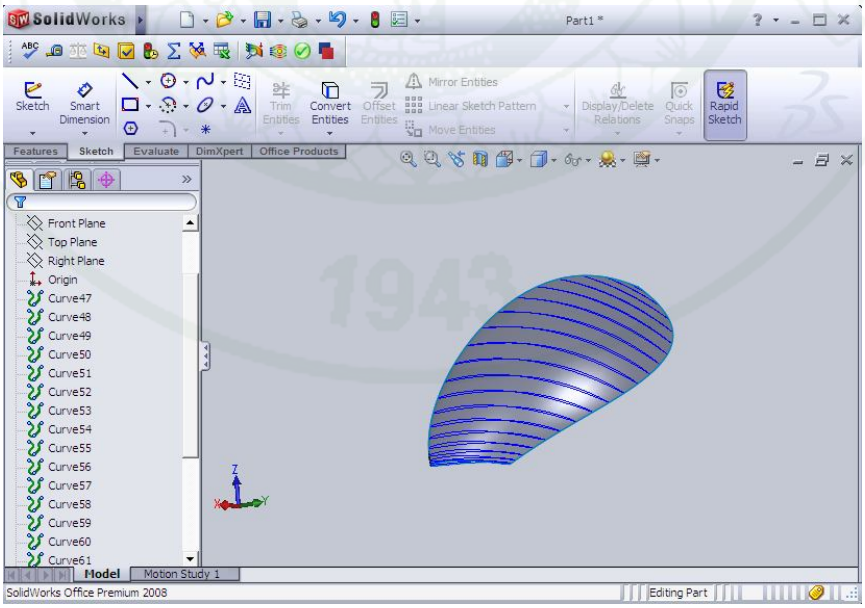
Appendix Figure A2 Procedure of using SOLIDWORKS step 2

3. Sketch → 3D Sketch → Choose Line-Convert Entities



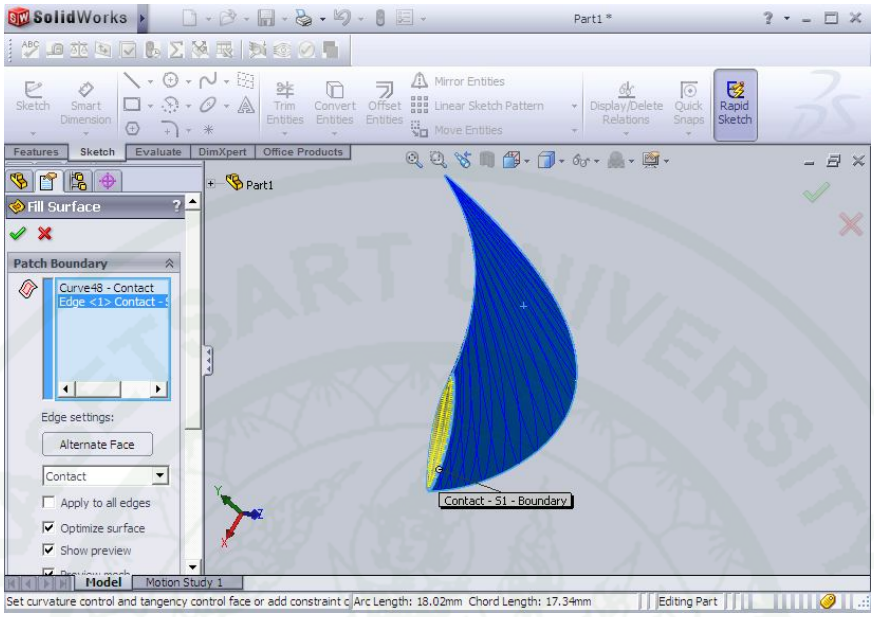
Appendix Figure A3 Procedure of using SOLIDWORKS step 3

4. Insert → Surface → Loft



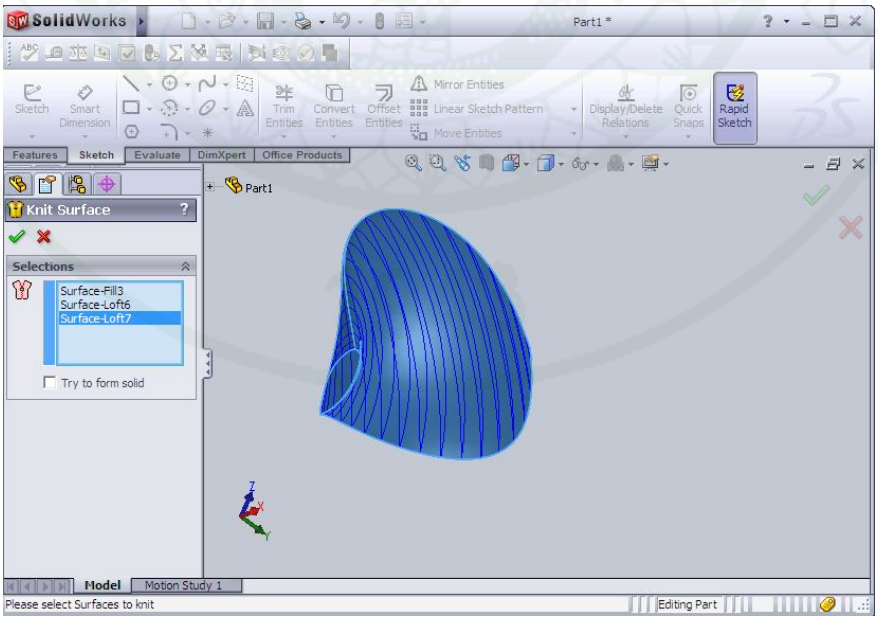
Appendix Figure A4 Procedure of using SOLIDWORKS step 4

5. Insert → Surface → Fill



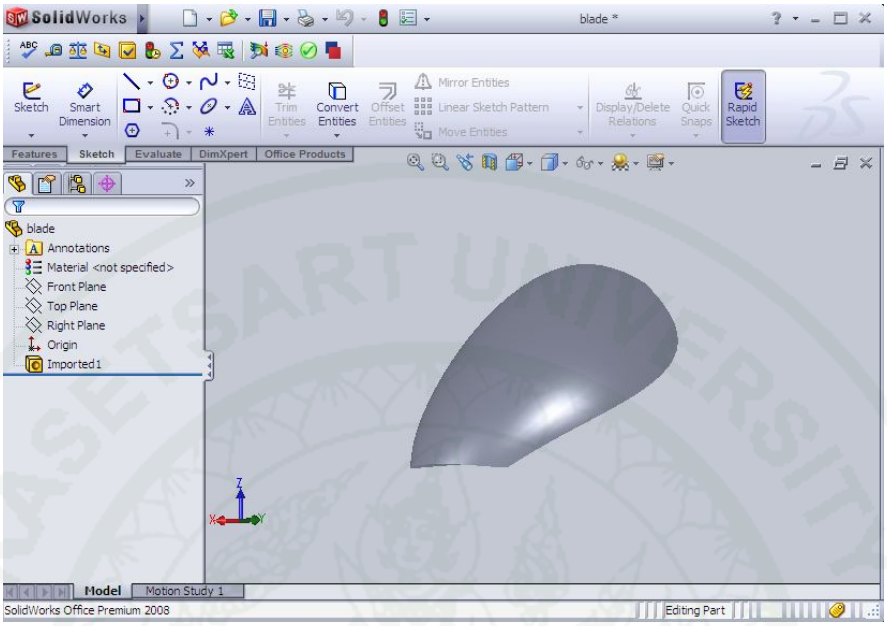
Appendix Figure A5 Procedure of using SOLIDWORKS step 5

6. Insert → Surface → Knit



Appendix Figure A6 Procedure of using SOLIDWORKS step 6

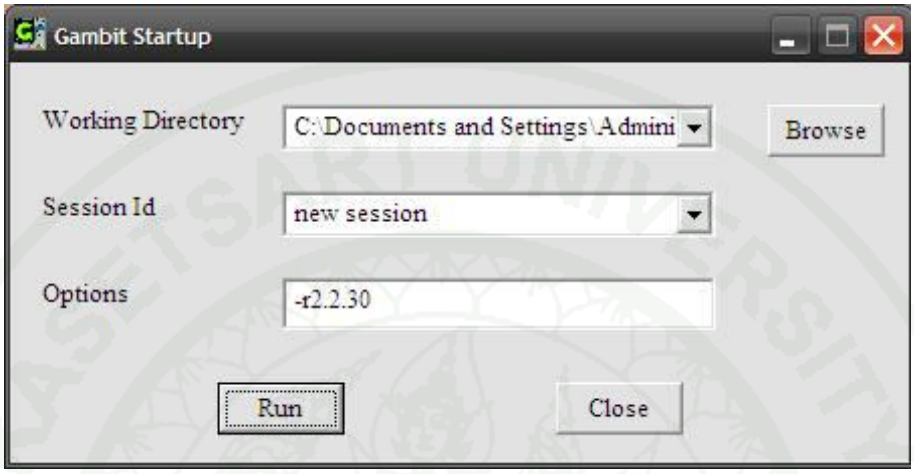
7. File → Save As → File Name → Blade → Type →.step



Appendix Figure A7 Procedure of using SOLIDWORKS step 7

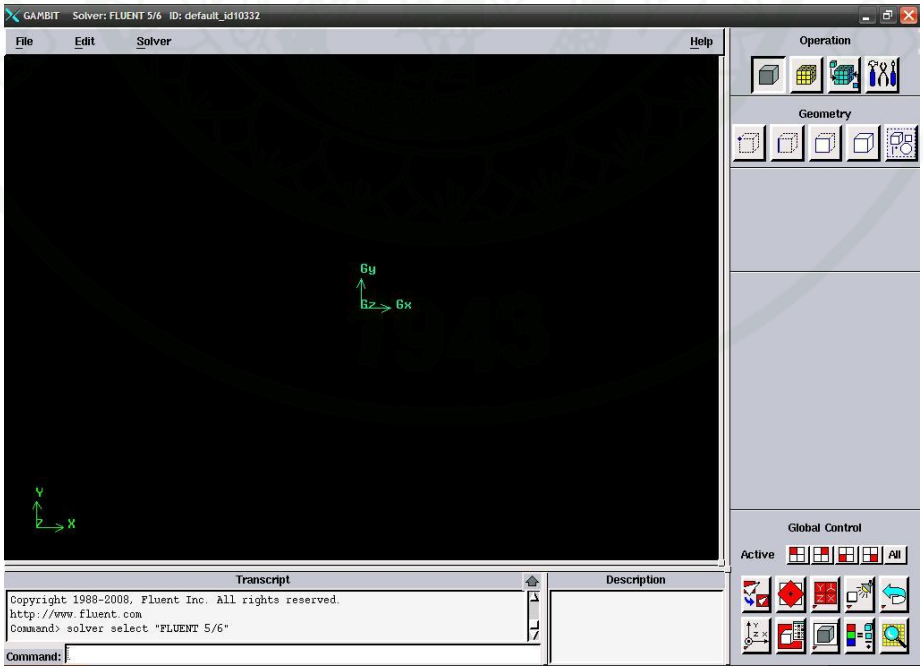
Using GAMBIT on Windows System

8. GAMBIT startup → Run



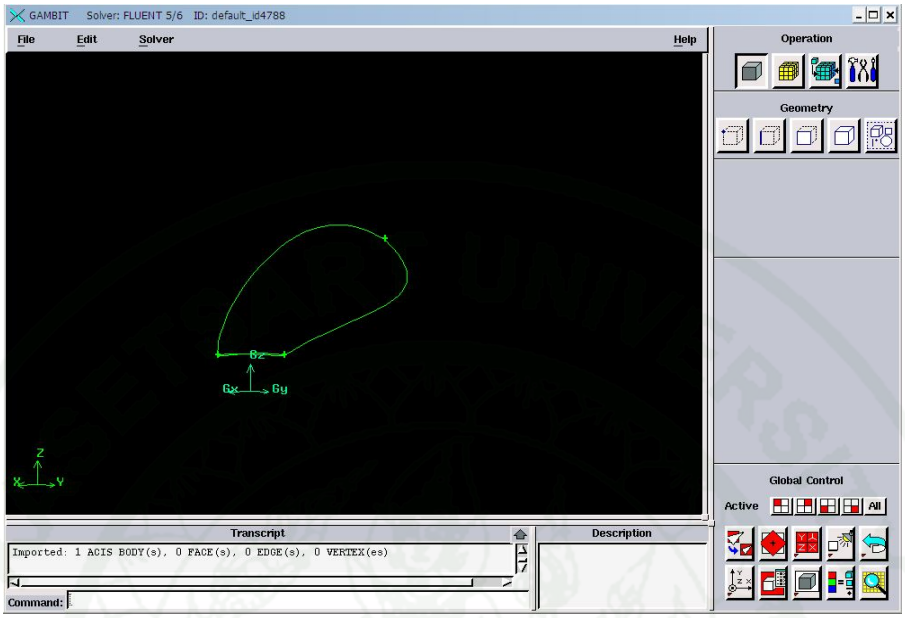
Appendix Figure A8 Procedure of using GAMBIT step 8

9. GAMBIT



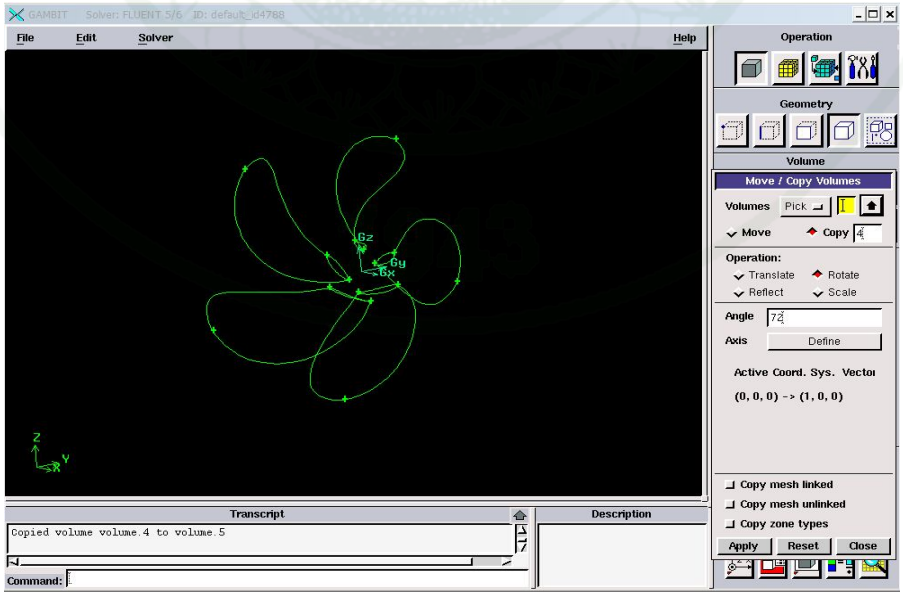
Appendix Figure A9 Procedure of using GAMBIT step 9

10. File → Import → STEP → Blade → Accept



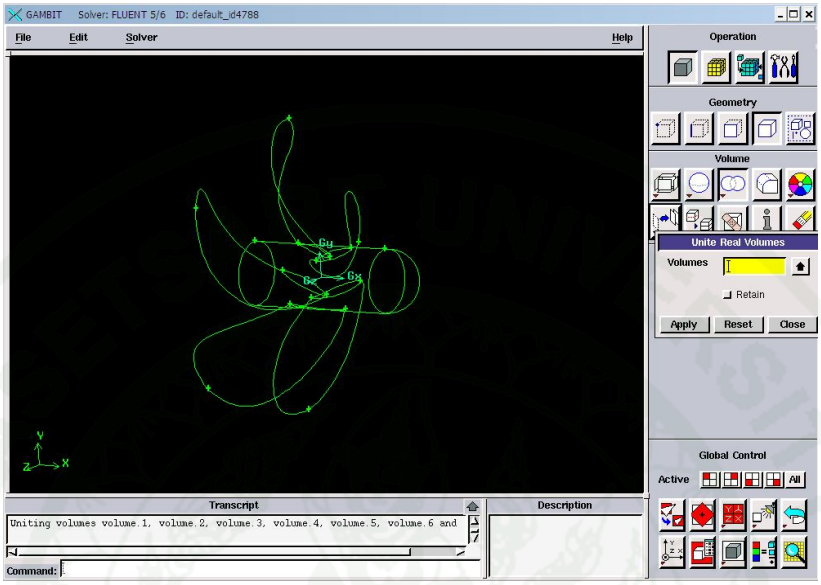
Appendix Figure A10 Procedure of using GAMBIT step 10

11. Operation → Geometry → Volume → Move/Copy Volume → Select Copy = 4 Angle = 72 , Axis → Defind → Select X Position→Apply



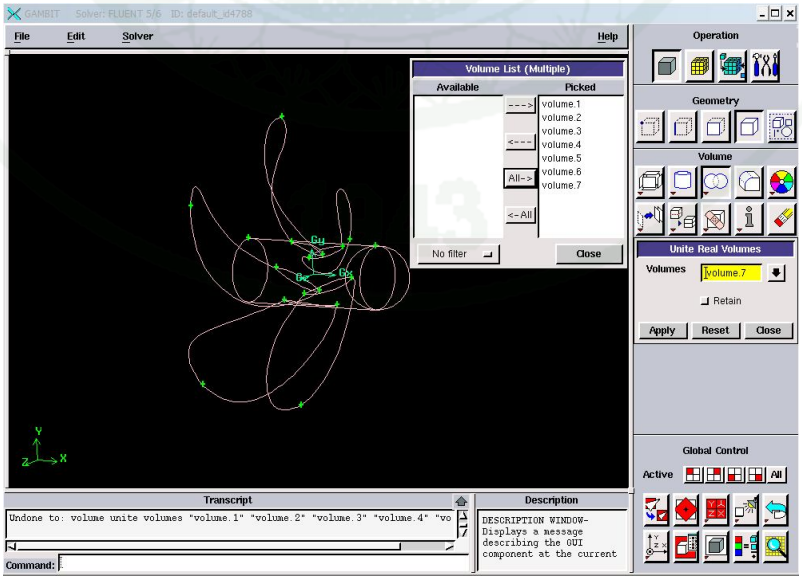
Appendix Figure A11 Procedure of using GAMBIT step 11

12. Operation → Geometry→Volume → Create Real Cylinder → Radius = 30.48 and Height = 300 → Apply



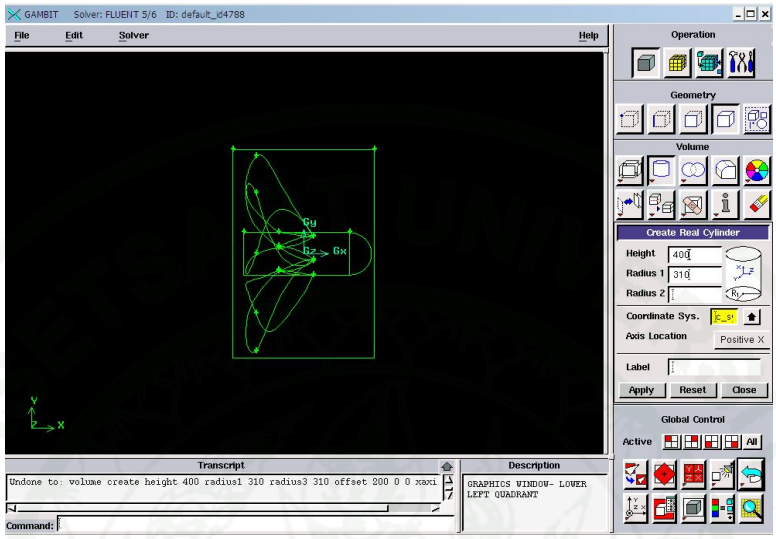
Appendix Figure A12 Procedure of using GAMBIT step 12

13. Operation → Geometry → Volume →Unite Real → Volumes → Select → Every Volumes → Apply



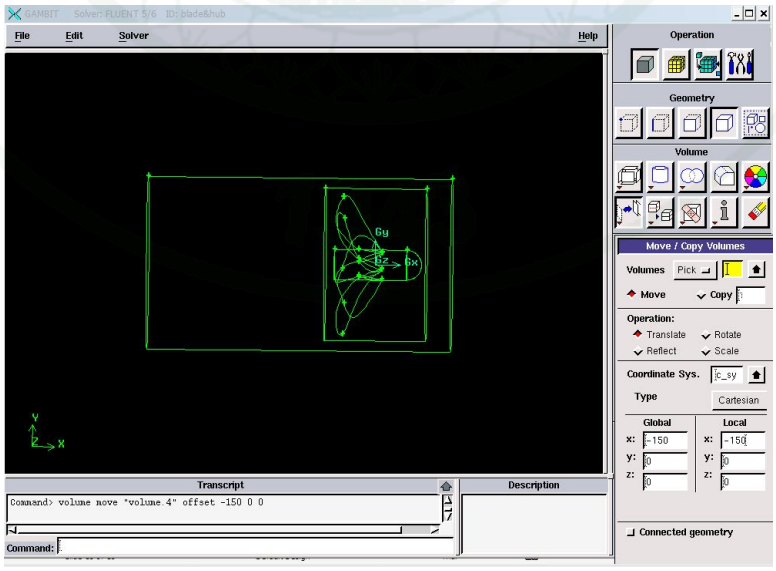
Appendix Figure A13 Procedure of using GAMBIT step 13

14. Operation → Geometry → Volume → Create Real Cylinder → Radius = 155 and Height = 200 → Axis Location → Center of X → Apply



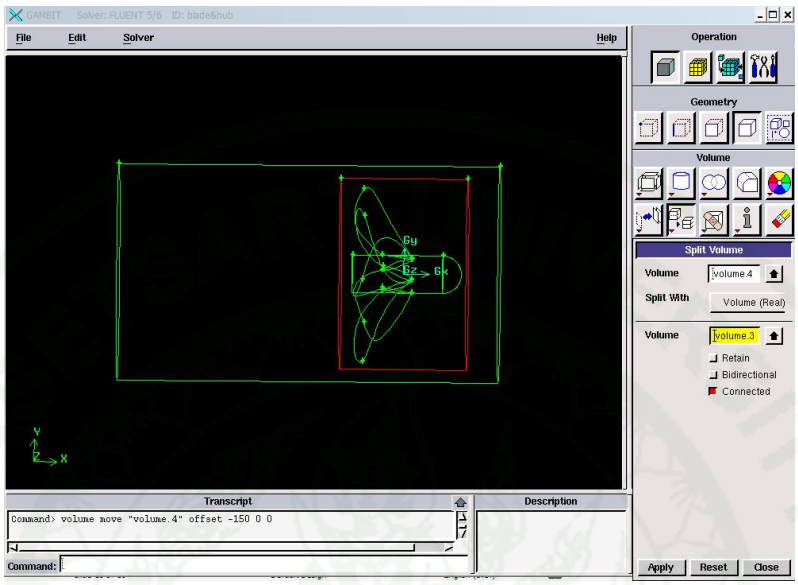
Appendix Figure A14 Procedure of using GAMBIT step 14

15. Operation → Geometry → Volume → Create Real Cylinder → Radius = 170 and Height = 600 → Axis Location → Center of X → Apply → Translate → Local = -150 → Apply



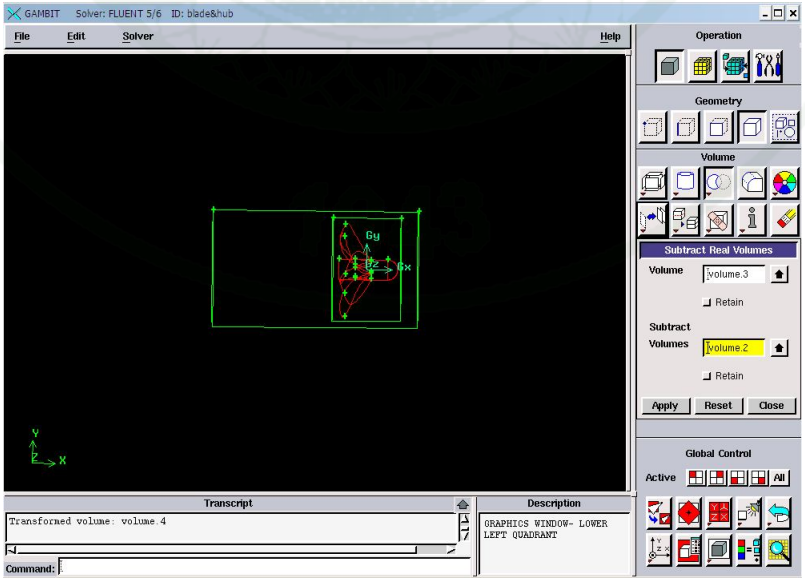
Appendix Figure A15 Procedure of using GAMBIT step 15

16. Operation → Geometry → Volume →Split Volume-Select Volume 4 and Volume 3→Apply



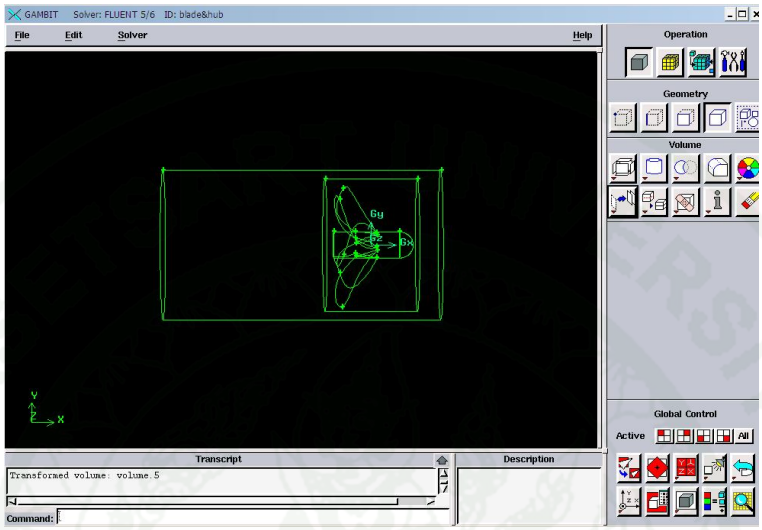
Appendix Figure A16 Procedure of using GAMBIT step 16

17. Operation → Geometry→Volume→Subtract Real Volumes → Select Volume 3 and Subtract Volumes → Select 2 →Apply



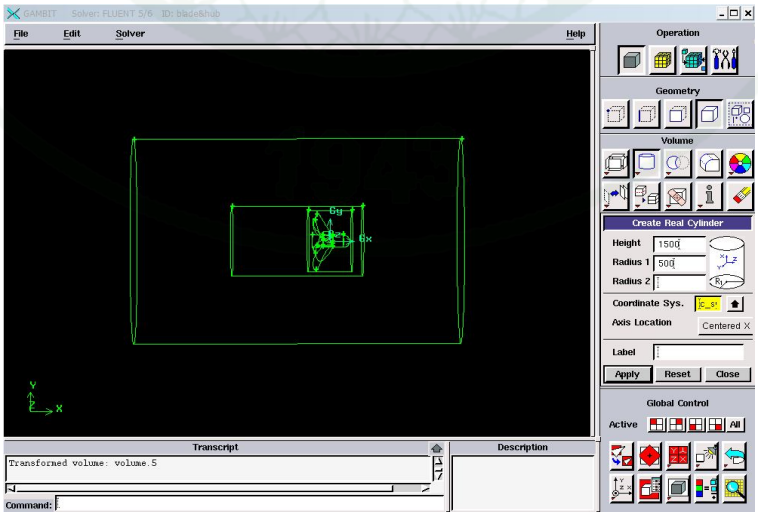
Appendix Figure A17 Procedure of using GAMBIT step 17

18. Operation → Geometry → Volume → Create Real Cylinder → Radius = 170 and Height = 600 → Axis Location → Center of X → Apply → Translate → Local = -150 → Apply



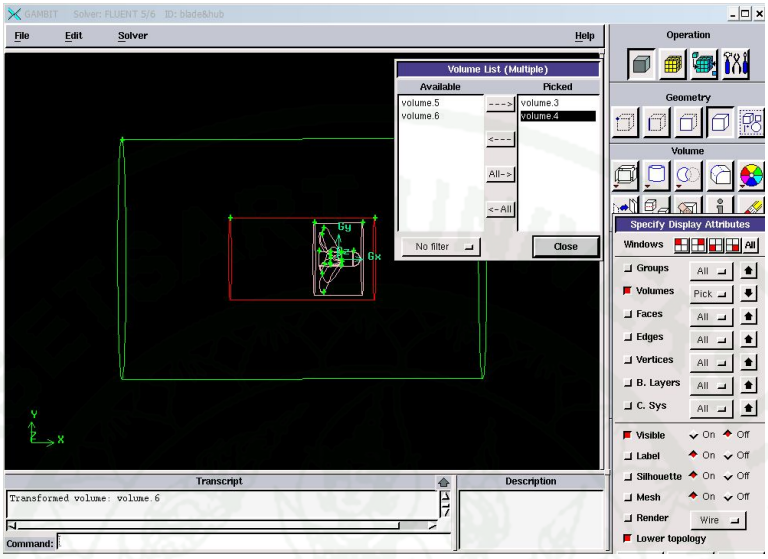
Appendix Figure A18 Procedure of using GAMBIT step 18

19. Operation → Geometry → Volume → Create Real Cylinder → Radius = 500 and Height = 1500 → Axis Location → Center of X → Apply → Translate → Local = -150 → Apply



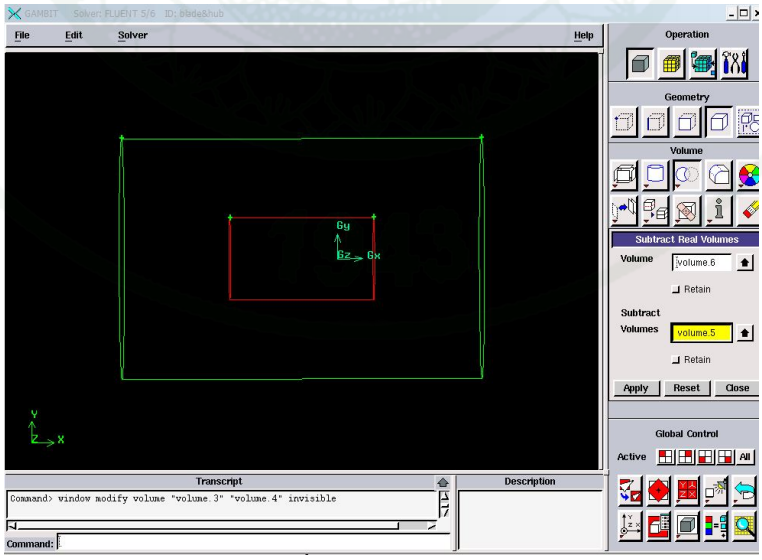
Appendix Figure A19 Procedure of using GAMBIT step 19

20. Operation → Geometry → Volume →Specify Display Attributes → Visible Select → Off → Volumes → Select → Vomumes 3 and 4 →Apply



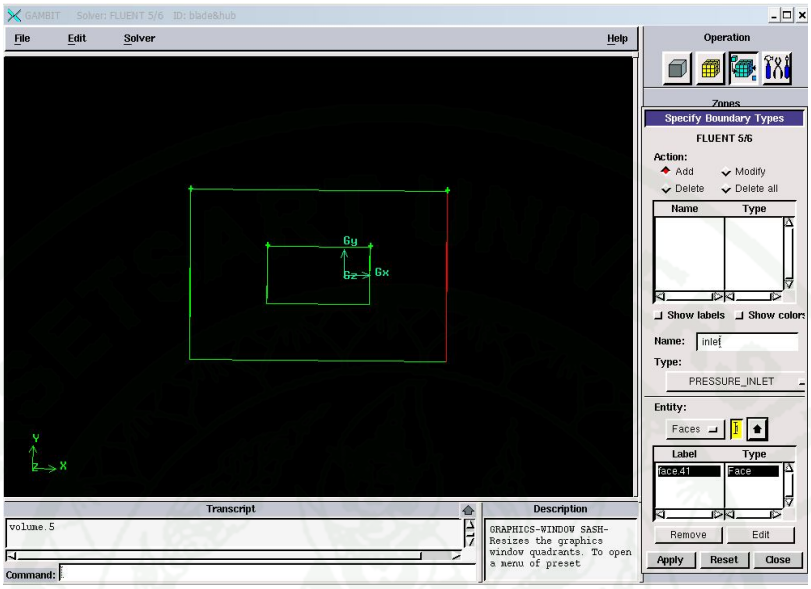
Appendix Figure A20 Procedure of using GAMBIT step 20

21. Operation → Geometry → Volume→Subtract Real Volumes → Select Volume 6 and Subtract Volumes → Select 6 →Apply



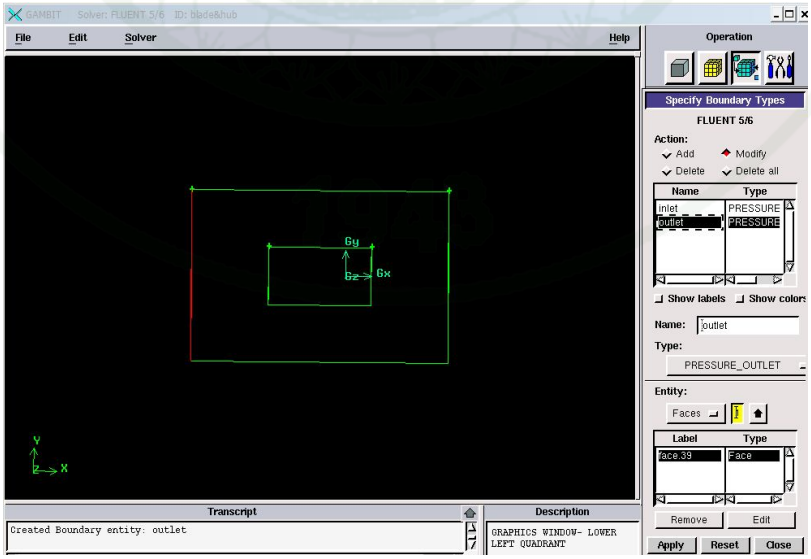
Appendix Figure A21 Procedure of using GAMBIT step 21

22. Operation → Zone → Specify Boundary Types → Name → Inlet → Types → PRESSURE_INLET →Faces → Select → Face 41 → Apply



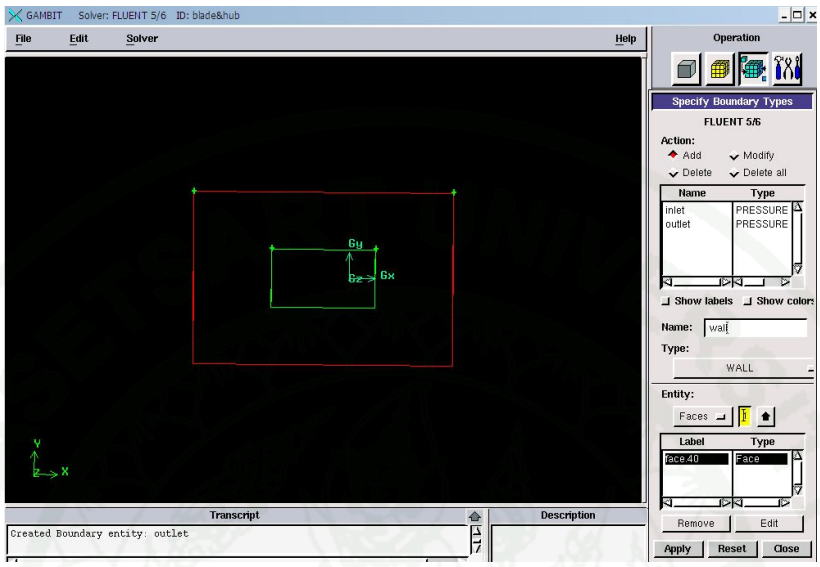
Appendix Figure A22 Procedure of using GAMBIT step 22

23. Operation → Zone → Specify Boundary Types → Name → Outlet → Types → PRESSURE_OULET →Faces → Select → Face 39 → Apply



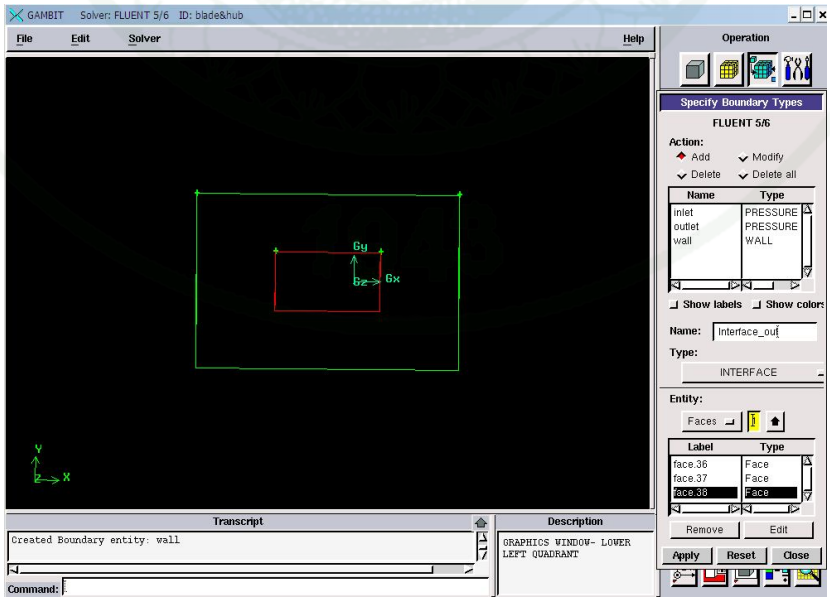
Appendix Figure A23 Procedure of using GAMBIT step 23

24. Operation → Zone → Specify Boundary Types → Name → Wall→ Types → WALL →Faces → Select → Face 40 → Apply



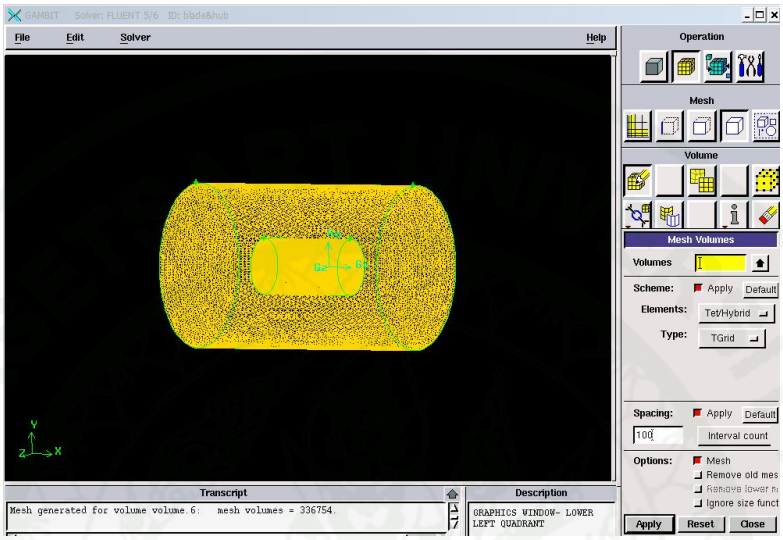
Appendix Figure A24 Procedure of using GAMBIT step 24

25. Operation → Zone → Specify Boundary Types → Name → Interface_out→ Types → INTERFACE →Faces → Select → Face 36,37,38 → Apply



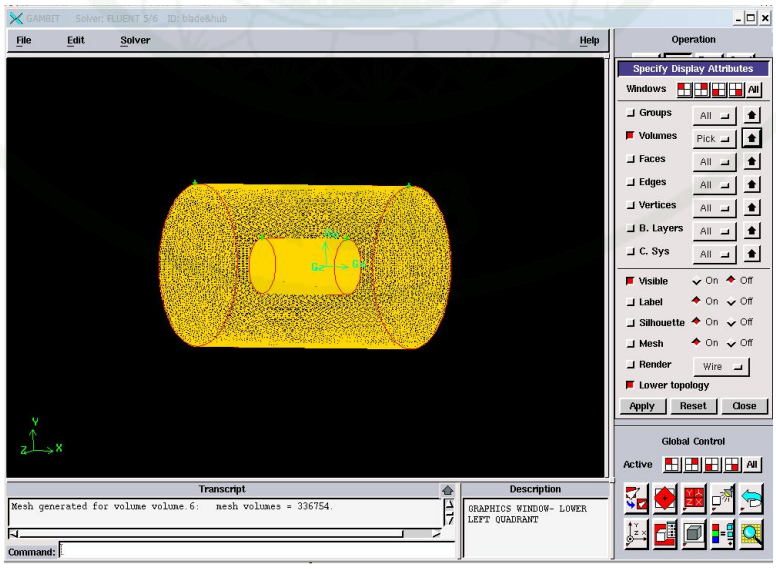
Appendix Figure A25 Procedure of using GAMBIT step 25

26. Operation→Mesh → Volume → Mesh Volume → Elements Selet → Tet/Hybrid → Type → Select → TGrid → Spacing → Select → 100 Interval count → Apply



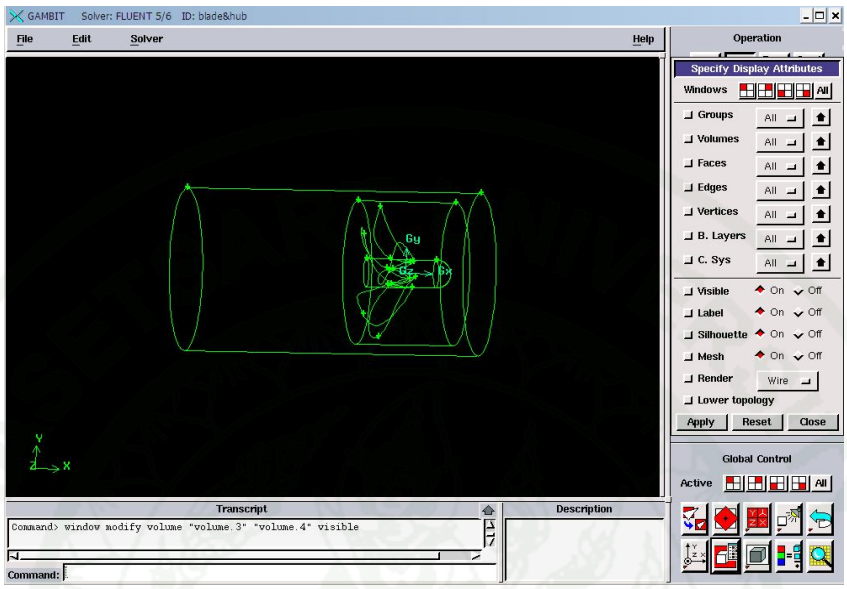
Appendix Figure A26 Procedure of using GAMBIT step 26

27. Operation→Geometry→Volume →Specify Display Attributes → Visible Select → Off → Volumes → Select → Vomumes 6 → Apply



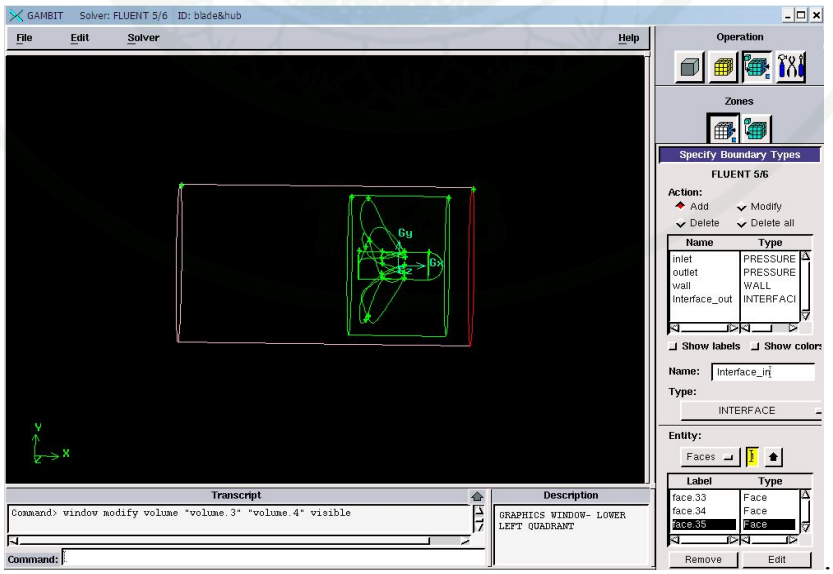
Appendix Figure A27 Procedure of using GAMBIT step 27

28. Operation → Geometry → Volume →Specify Display Attributes → Visible Select → On → Volumes → Select → Vomumes 3 and 4 → Apply



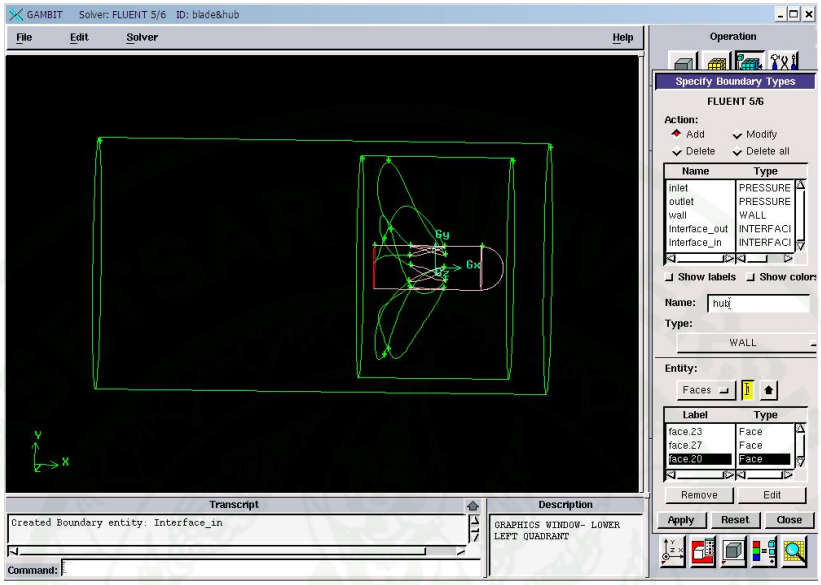
Appendix Figure A28 Procedure of using GAMBIT step 28

29. Operation → Zone → Specify Boundary Types → Name → Interface_in→ Types → INTERFACE →Faces → Select → Face 33,34,35→ Apply



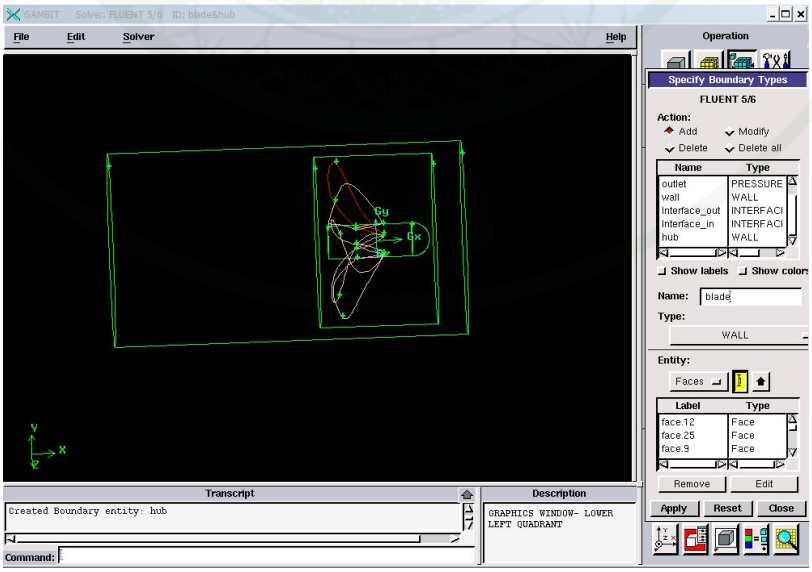
Appendix Figure A29 Procedure of using GAMBIT step 29

30. Operation → Zone → Specify Boundary Types → Name → hub→ Types → WALL →Faces → Select → Face 20,23,27→ Apply



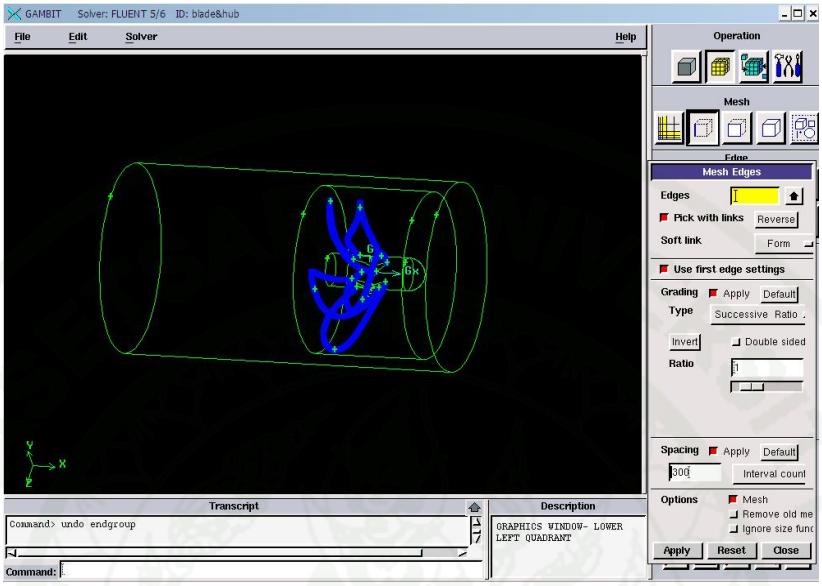
Appendix Figure A30 Procedure of using GAMBIT step 30

31. Operation → Zone → Specify Boundary Types → Name → blade→ Types → WALL →Faces → Select → Face 7,9,12,15,18,24.25,26,28,29→Apply



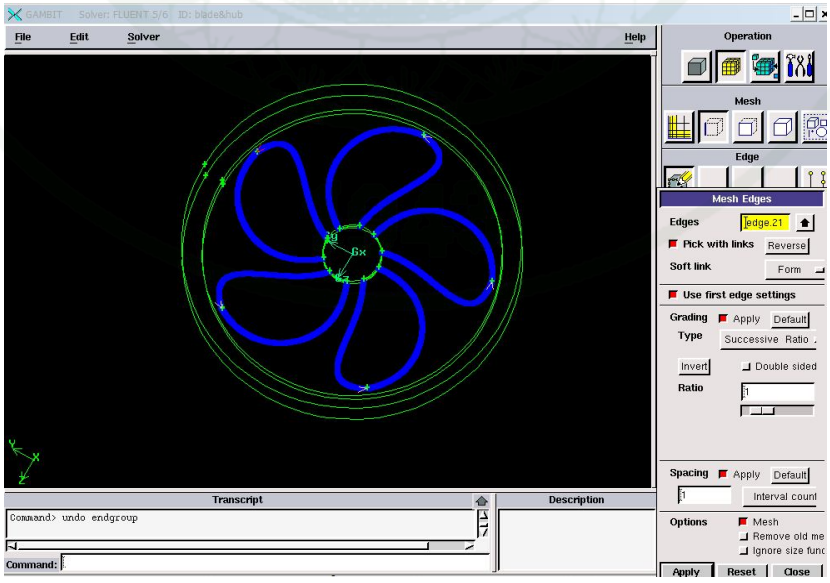
Appendix Figure A31 Procedure of using GAMBIT step 31

32. Operation →Mesh → Edge → Mesh Edges→ Ratio → Select 1 → Spacing → Select→300 Interval count →→ Apply



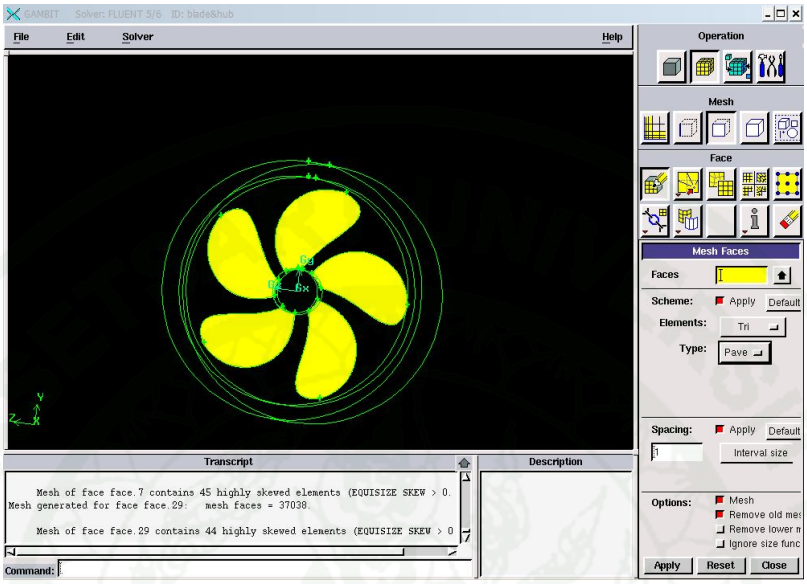
Appendix Figure A32 Procedure of using GAMBIT step 32

33. Operation →Mesh → Edge → Mesh Edges→ Ratio → Select 1 → Spacing → Select → 1 Interval count → Apply



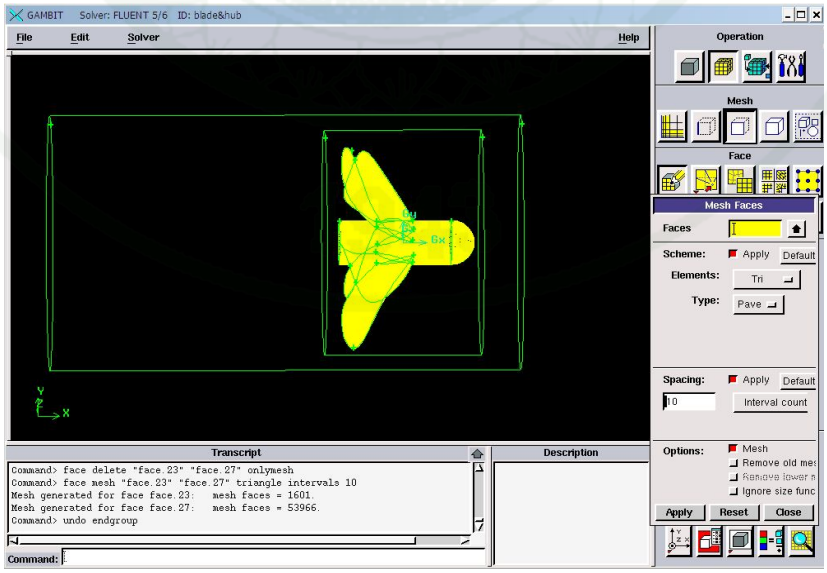
Appendix Figure A33 Procedure of using GAMBIT step 33

34. Operation →Mesh → Face → Mesh Faces→ Elements Selet → Tri → Type → Select → Pave→ Spacing → Select → 10 Interval count →Apply



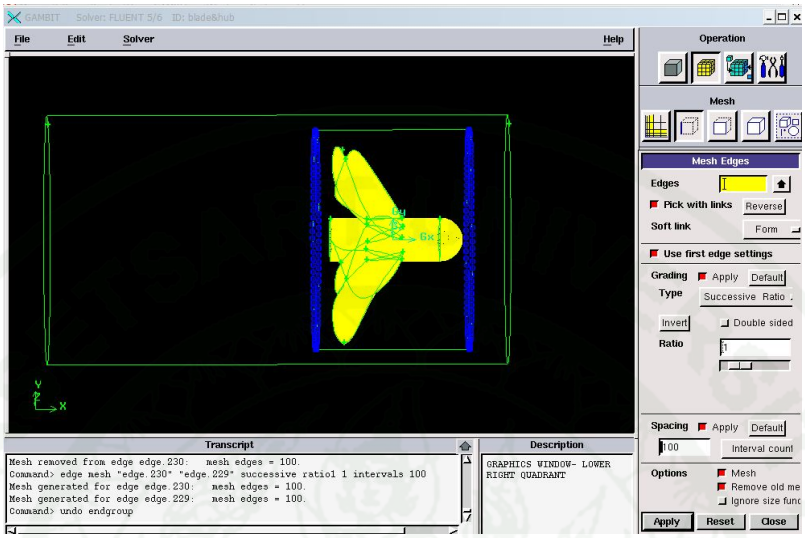
Appendix Figure A34 Procedure of using GAMBIT step 34

35. Operation →Mesh → Face → Mesh Faces→ Elements Selet →Faces → Hub → Tri → Type → Select →Pave→ Spacing → Select → 10 Interval count → Apply



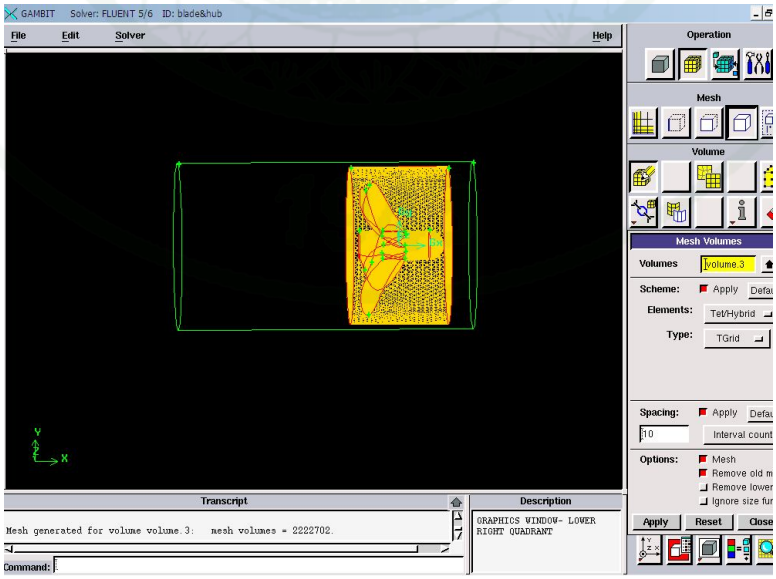
Appendix Figure A35 Procedure of using GAMBIT step 35

36. Operation →Mesh → Edge → Mesh Edges→ Ratio →Select 1 → Edges → Select → 229 and 230 → Spacing → Select → 100 Interval count →Apply



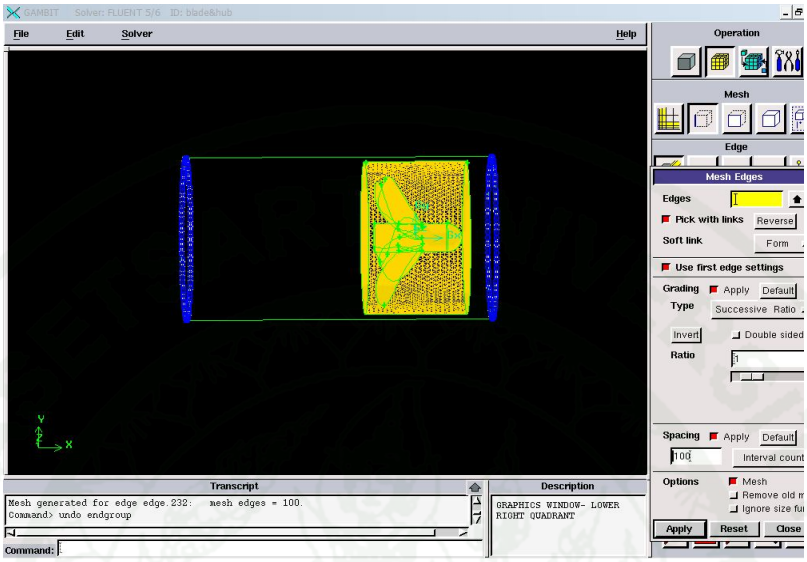
Appendix Figure A36 Procedure of using GAMBIT step 36

37. Operation →Mesh → Volume → Mesh Volumes→ Volumes →Select → Volume 3 → Elements Selet → Tet/Hybrid → Type → Select → TGrid → Spacing → Select →10 Interval count → Apply



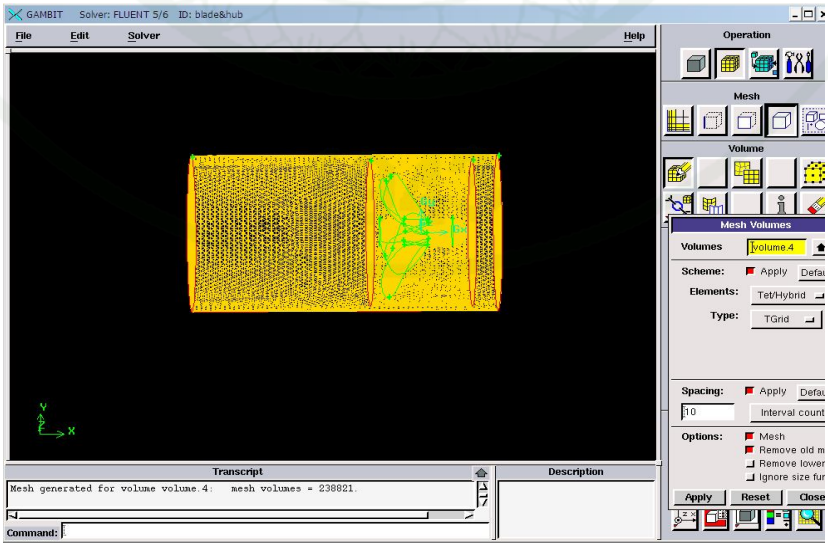
Appendix Figure A37 Procedure of using GAMBIT step 37

38. Operation →Mesh → Edge → Mesh Edges→ Ratio→Select 1 → Edges → Select → 231and 232 → Spacing → Select → 100 Interval count → Apply



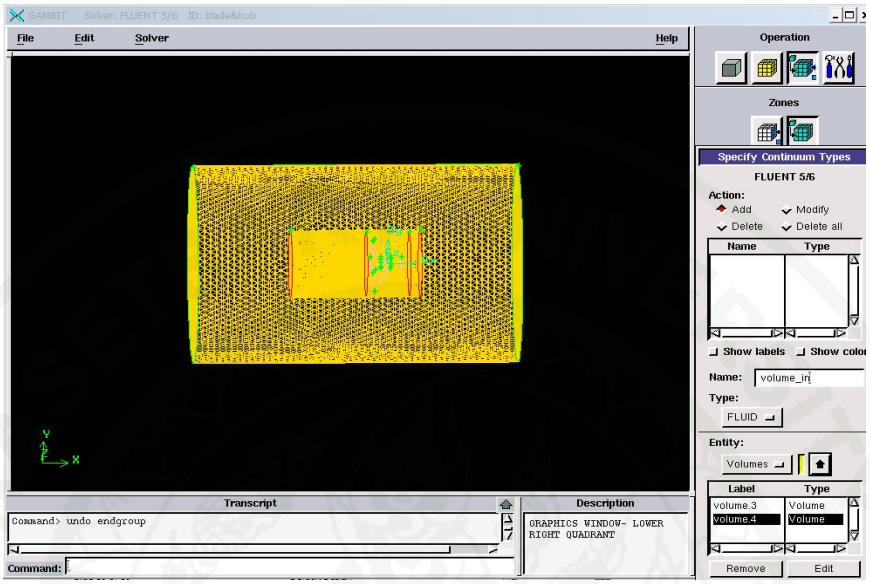
Appendix Figure A38 Procedure of using GAMBIT step 38

39. Operation →Mesh → Volume → Mesh Volumes→ Volumes → Select → Volume 4 → Elements Selet → Tet/Hybrid → Type → Select → TGrid → Spacing →Select →10 Interval count → Apply



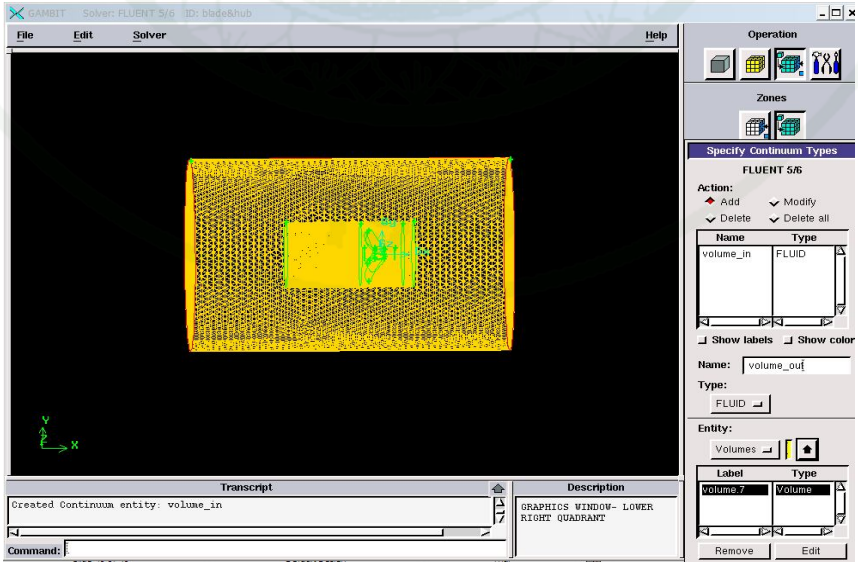
Appendix Figure A39 Procedure of using GAMBIT step 39

40. Operation → Zones → Specify Continuum Types→Name→Select→Volume_in → Type◊Fluid→Entity Volumes→Select→Volume 3 and 4 → Apply



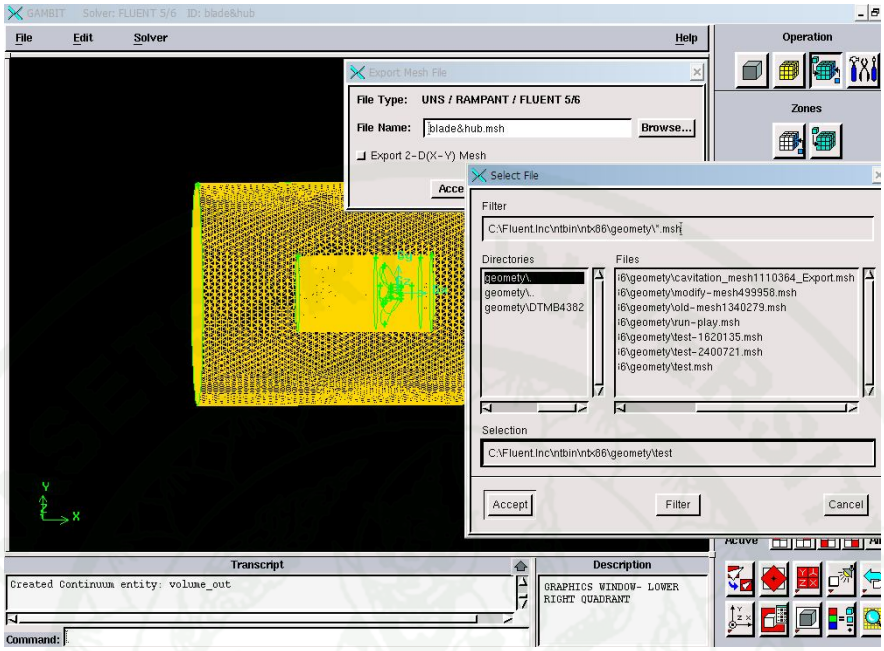
Appendix Figure A40 Procedure of using GAMBIT step 40

41. Operation → Zones → Specify Continuum Types→Name→Select→Volume_out → Type◊Fluid→Entity Volumes→Select→Volume 7 → Apply

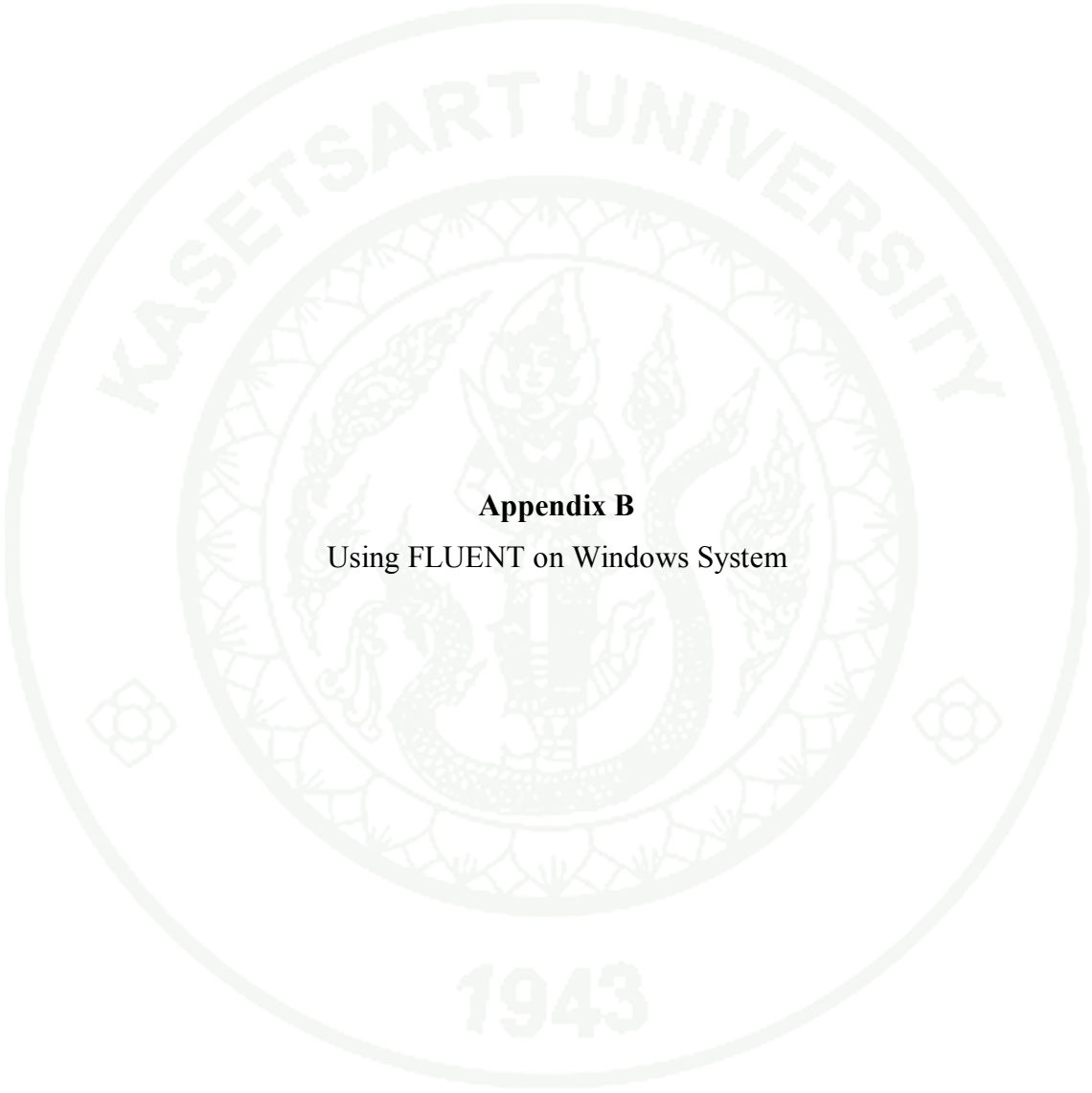


Appendix Figure A41 Procedure of using GAMBIT step 41

42. File → Export → Mesh → Browse → test.mesh → Accept



Appendix Figure A42 Procedure of using GAMBIT step 42



Appendix B

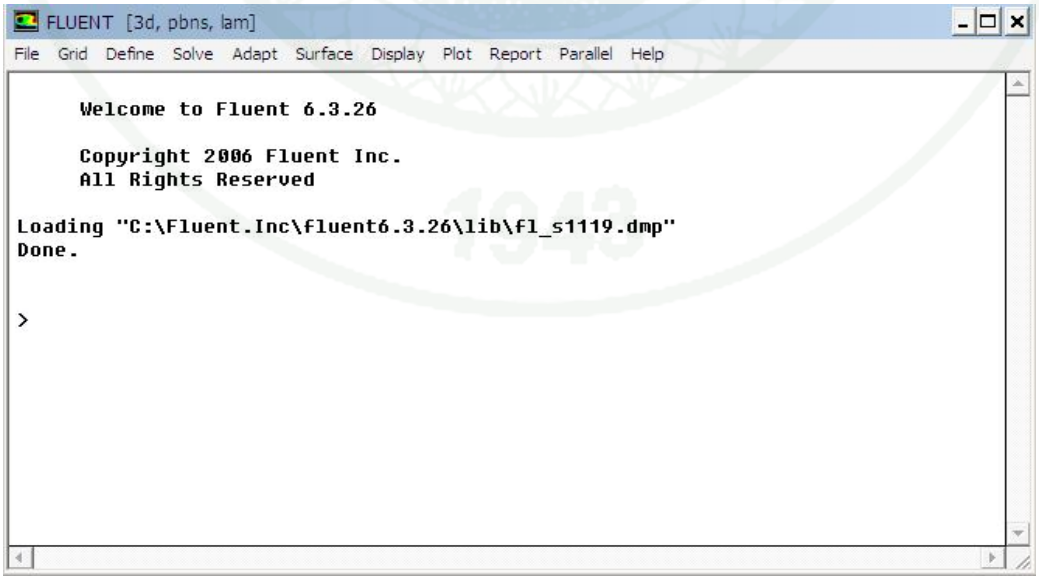
Using FLUENT on Windows System

Using FLUENT on Windows System

1. Select FLUENT version → 3d → Run

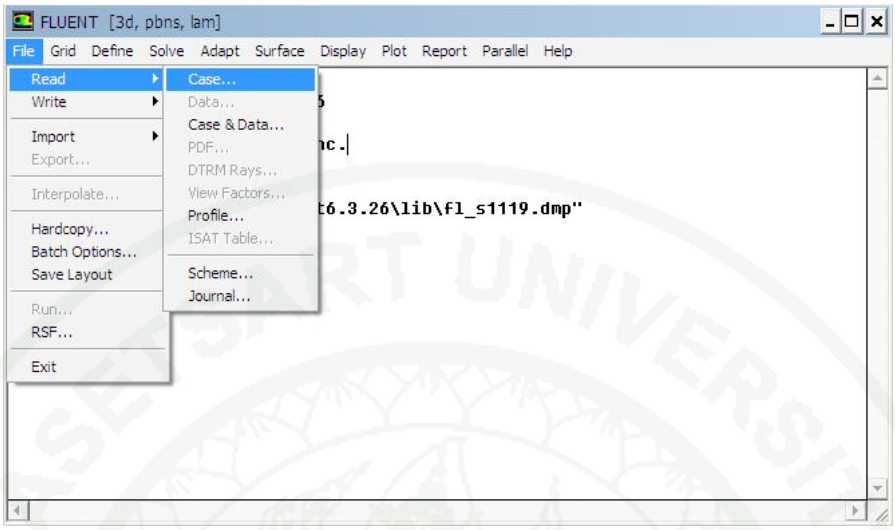
Appendix Figure B1 Procedure of using FLUENT step 1

2. FLUENT command Windows



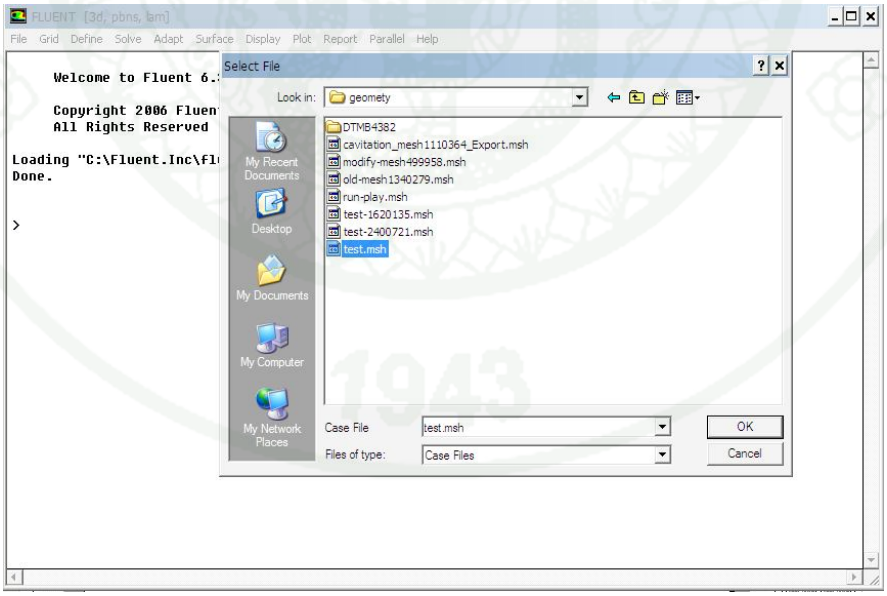
Appendix Figure B2 Procedure of using FLUENT step 2

3. File → Read → Case



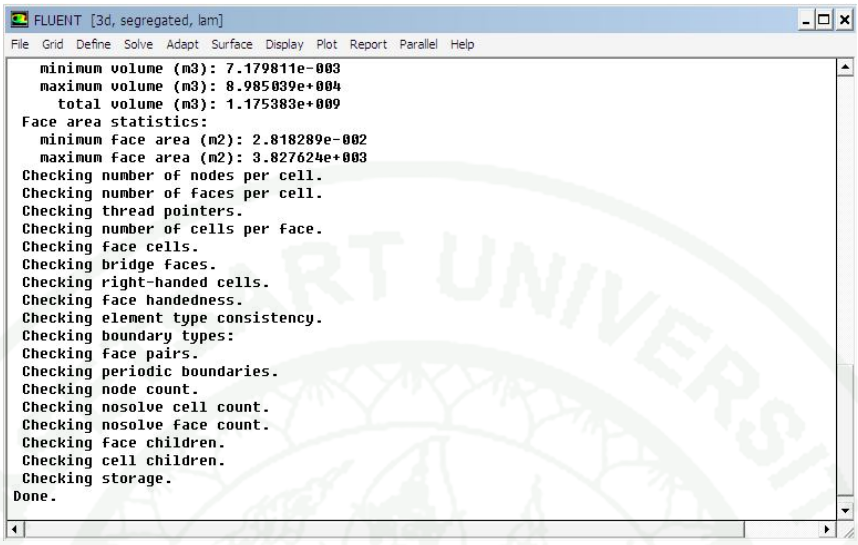
Appendix Figure B3 Procedure of using FLUENT step 3

4. Open file test.msh



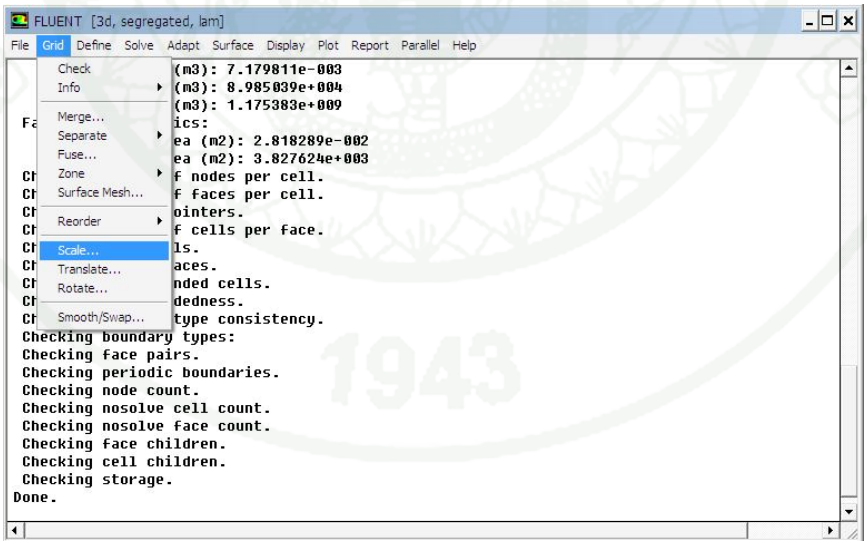
Appendix Figure B4 Procedure of using FLUENT step 4

5. Grid-Chick



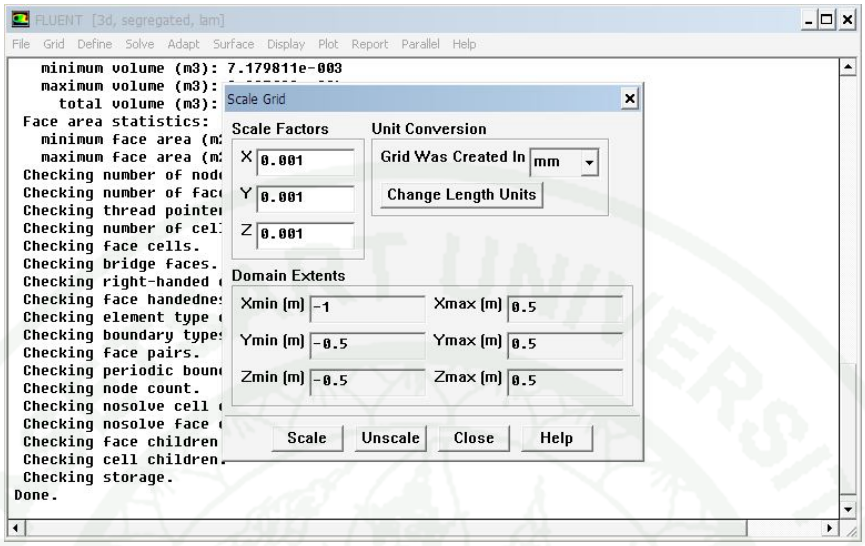
Appendix Figure B5 Procedure of using FLUENT step 5

6. Grid-Scale



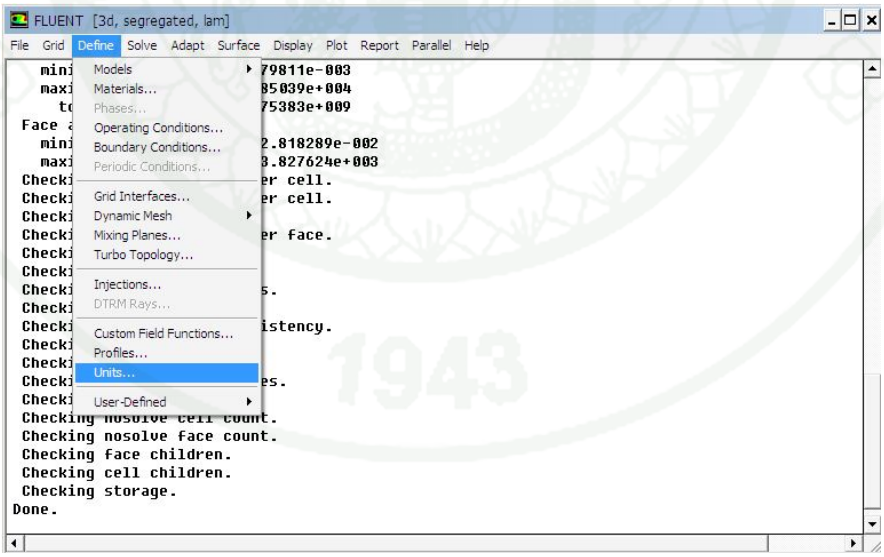
Appendix Figure B6 Procedure of using FLUENT step 6

7. Select Grid was Created in mm and click Scale



Appendix Figure B7 Procedure of using FLUENT step 7

8. Define - Units

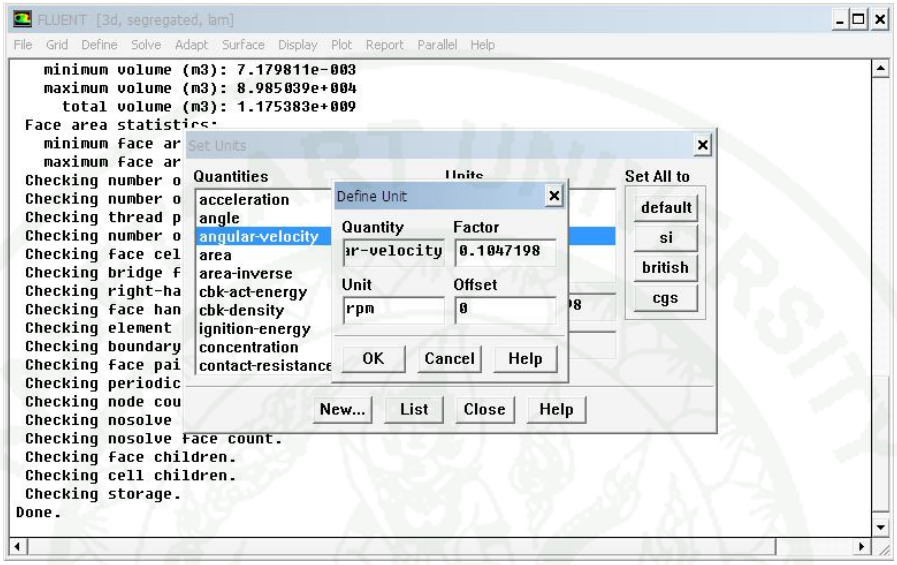


Appendix Figure B8 Procedure of using FLUENT step 8

9. Quantities select angular-velocity

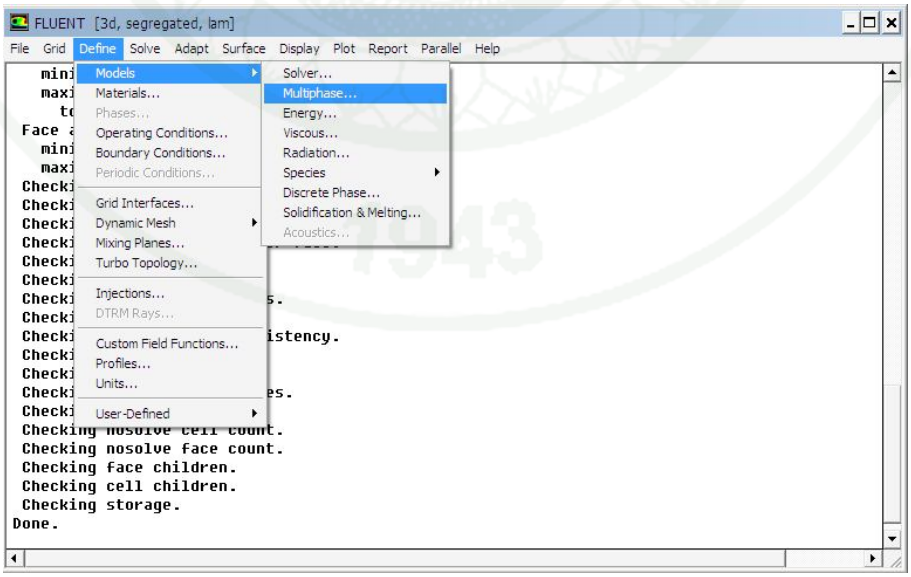
   Unit select rpm

   Click New OK



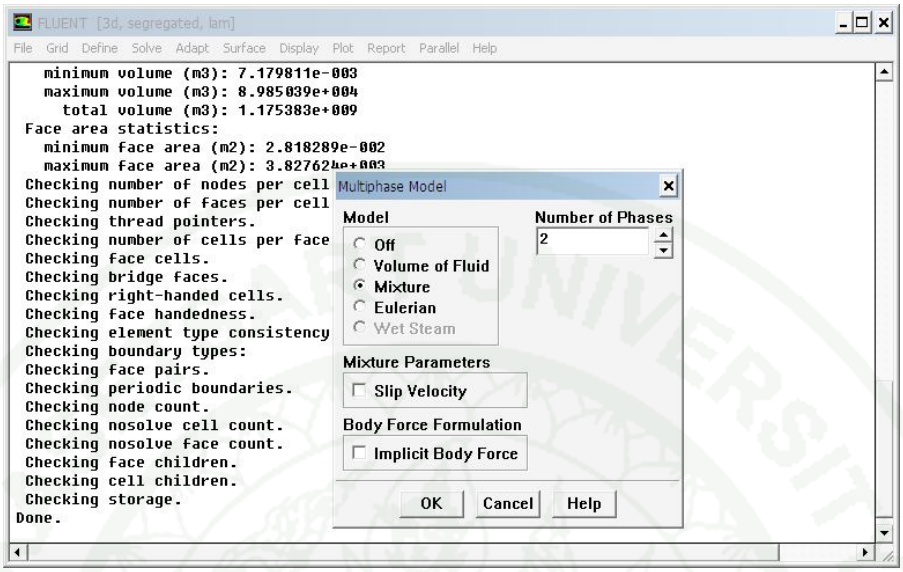
Appendix Figure B9 Procedure of using FLUENT step 9

10. Define-Models-Multiphase



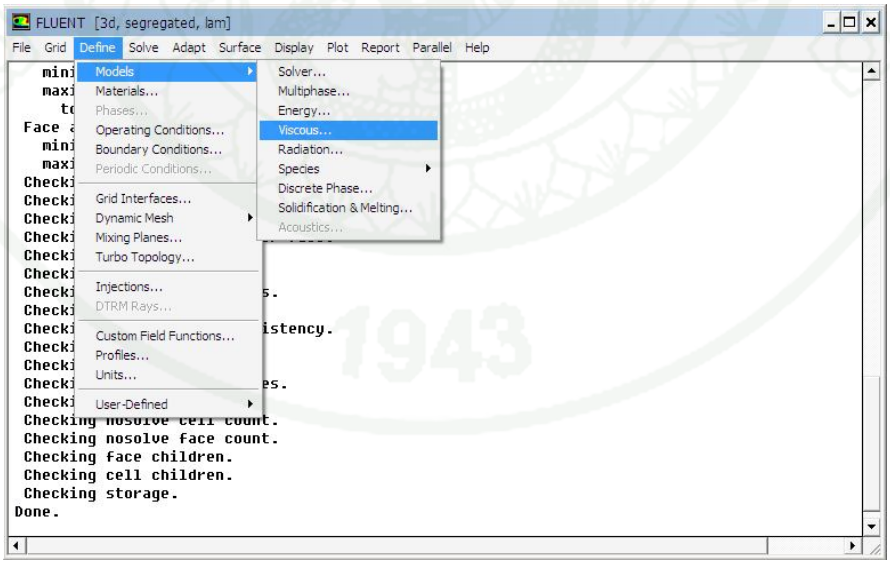
Appendix Figure B10 Procedure of using FLUENT step 10

11. Model select Mixture and click OK



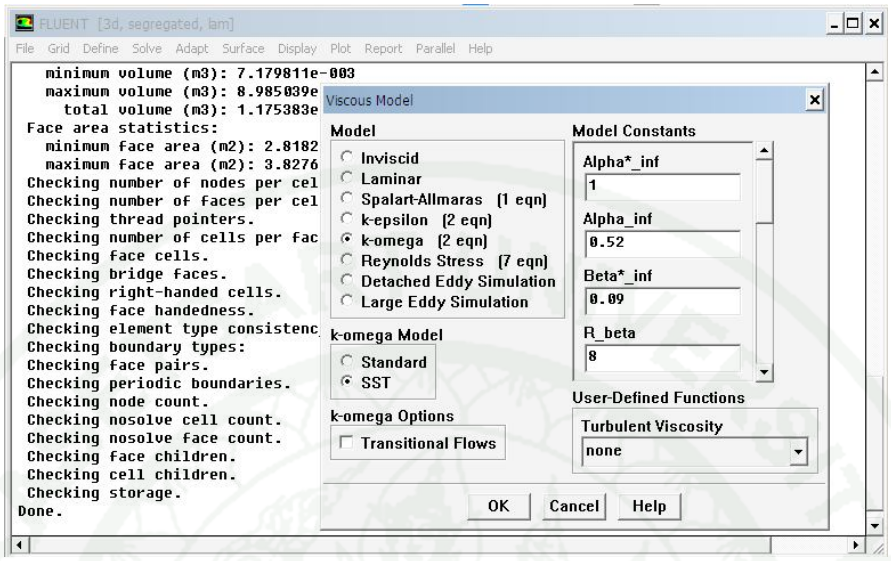
Appendix Figure B11 Procedure of using FLUENT step 11

12. Define-Models-Viscous



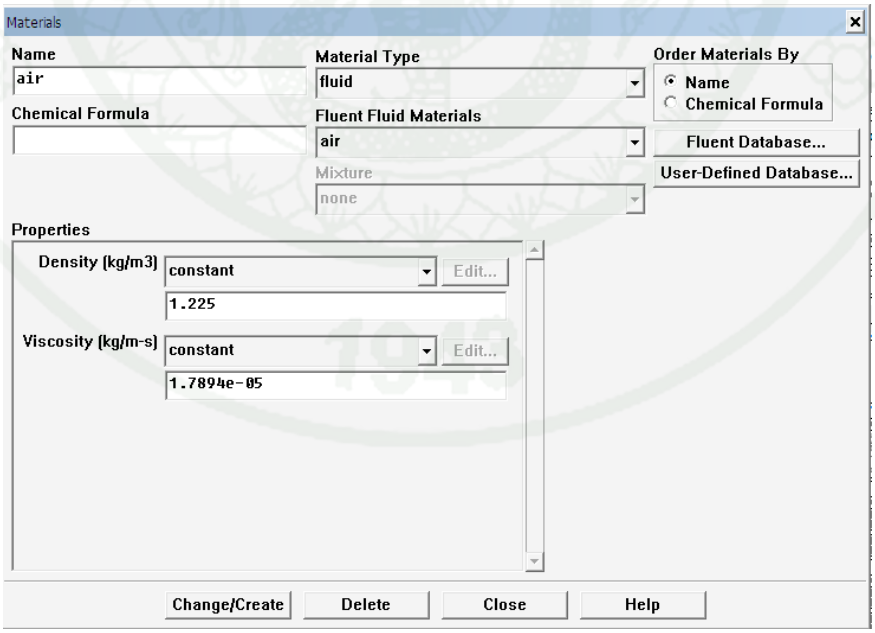
Appendix Figure B12 Procedure of using FLUENT step 12

13. Select k-omega (SST) and click OK



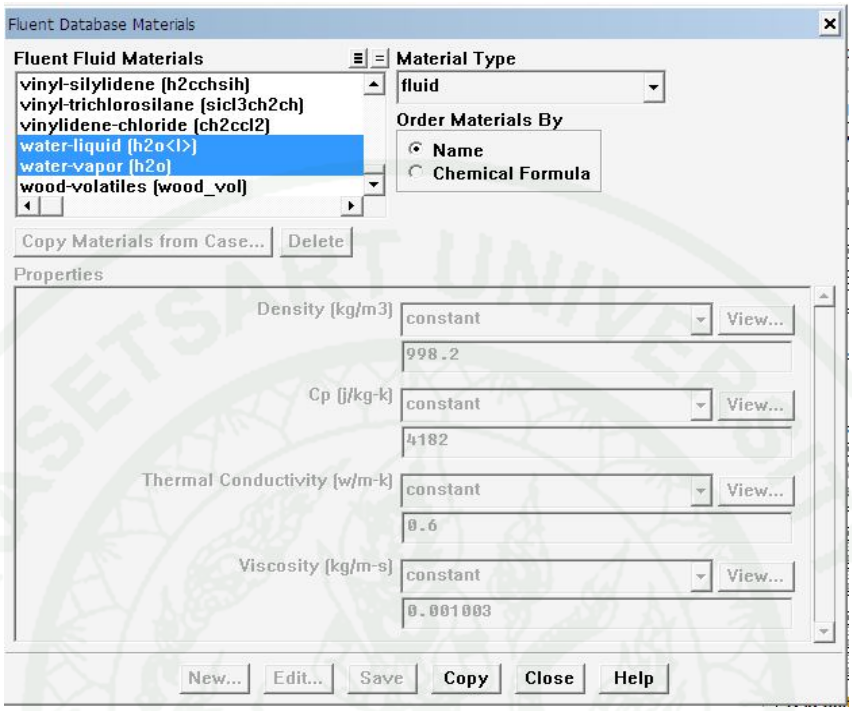
Appendix Figure B13 Procedure of using FLUENT step 13

14. Define-Material-select Fluent Database



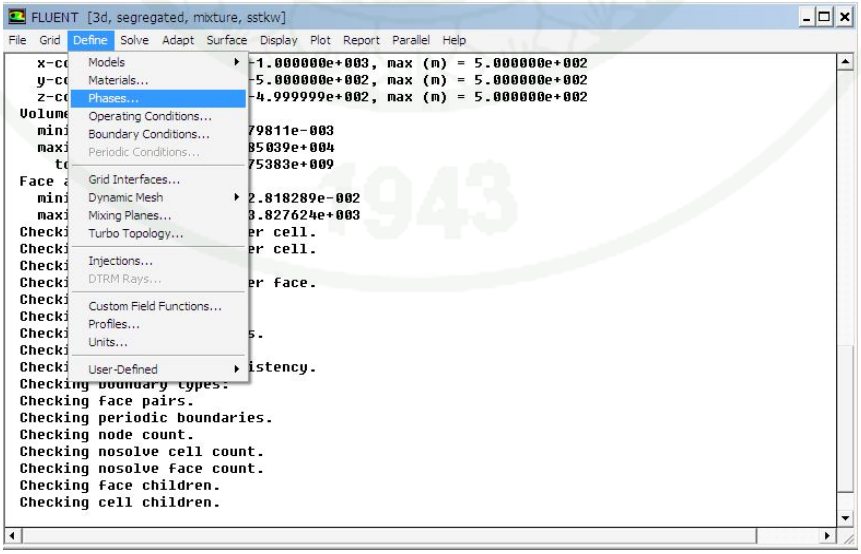
Appendix Figure B14 Procedure of using FLUENT step 14

15. Select water-liquid (h2o<l>) and water-vapor(h2o) and click Copy



Appendix Figure B15 Procedure of using FLUENT step 15

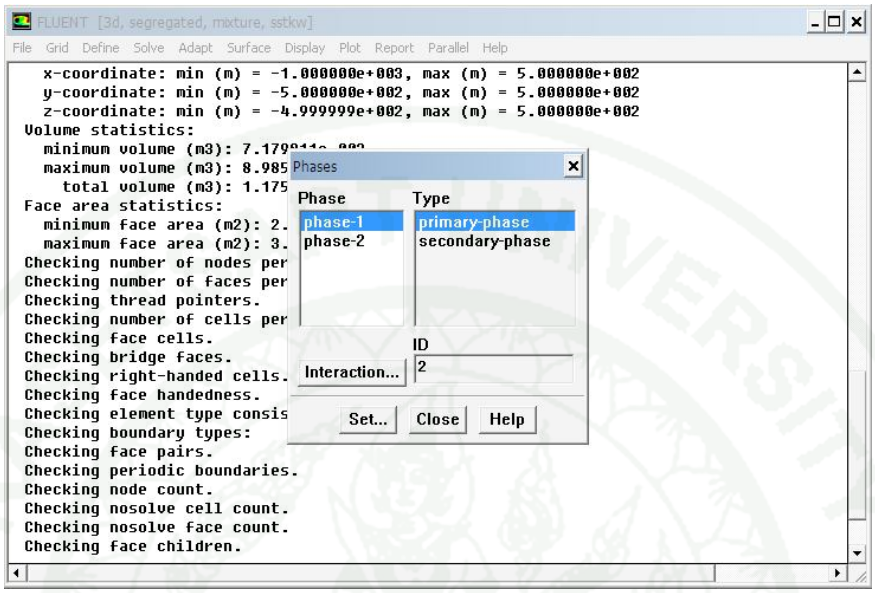
16. Define-phases



Appendix Figure B16 Procedure of using FLUENT step 16

17. Phase select phase 1

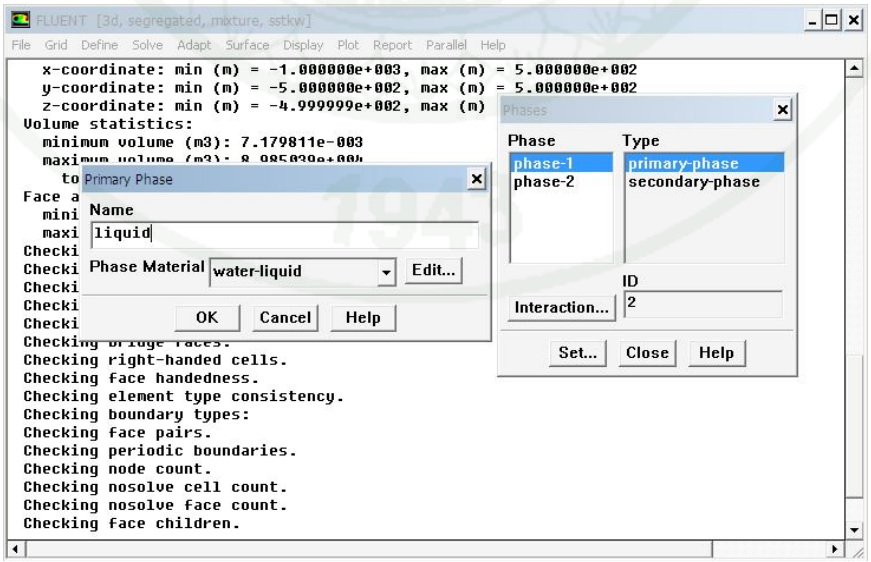
    Type select primary-phase and click set



Appendix Figure B17 Procedure of using FLUENT step 17

18. Primary Phase: Name: liquid

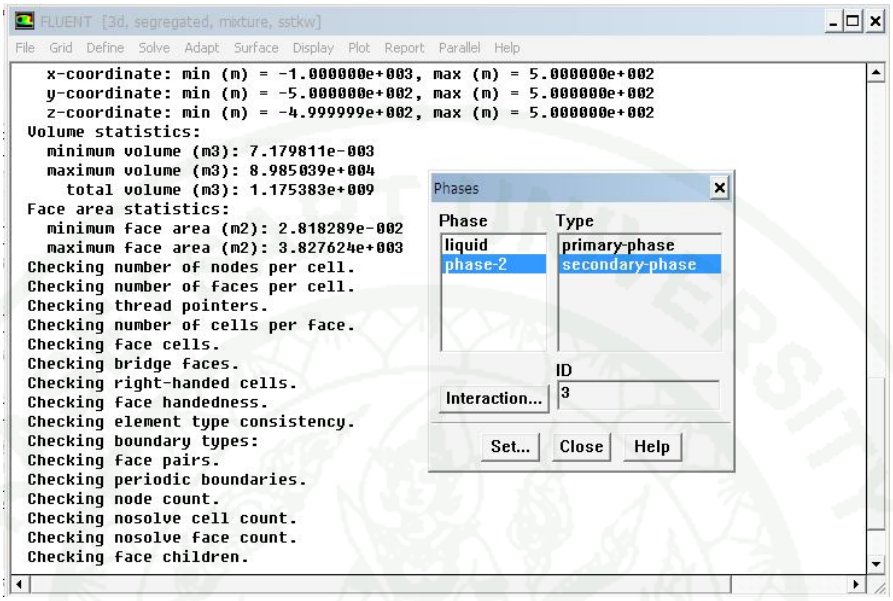
    Phase Material select water-liquid and click OK



Appendix Figure B18 Procedure of using FLUENT step 18

19. Phase select phase-2

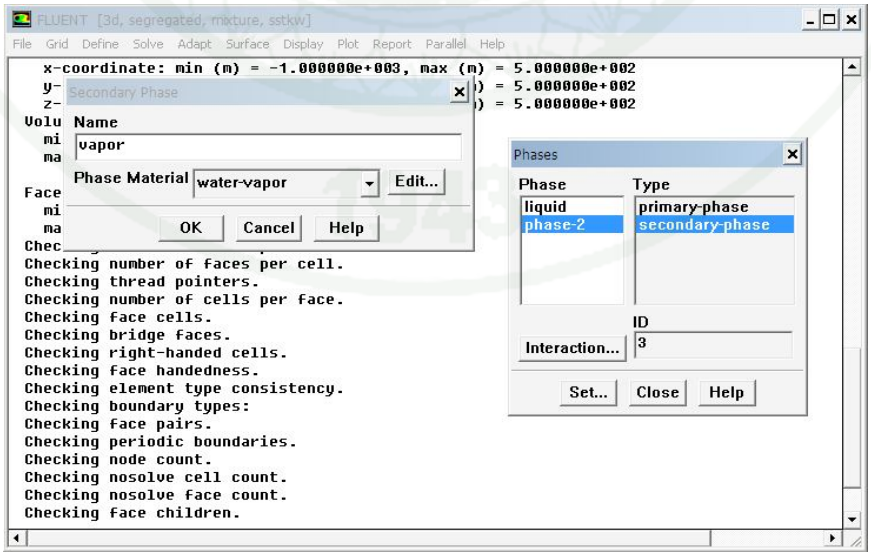
    Phase Type select secondary-phase and click set



Appendix Figure B19 Procedure of using FLUENT step 19

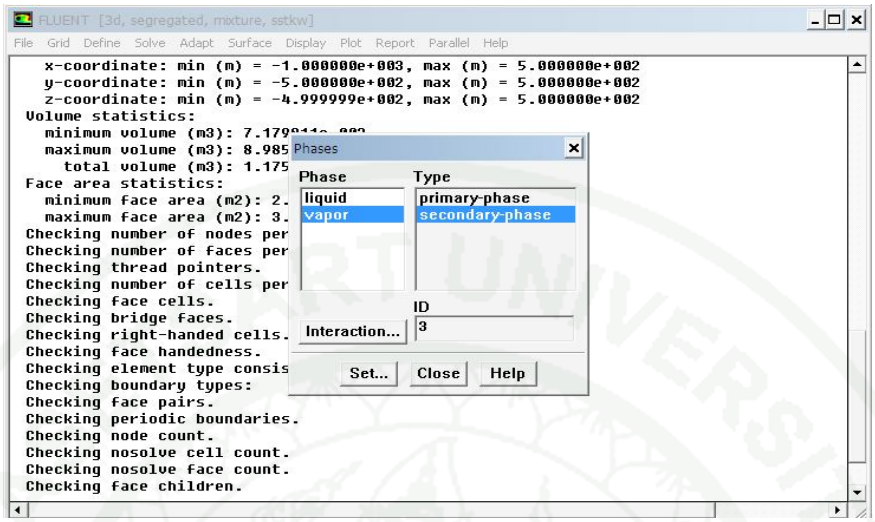
20. Secondary Phase: Name: vapor

    Phase Material select water-vapor and click OK



Appendix Figure B20 Procedure of using FLUENT step 20

21. Select Interaction



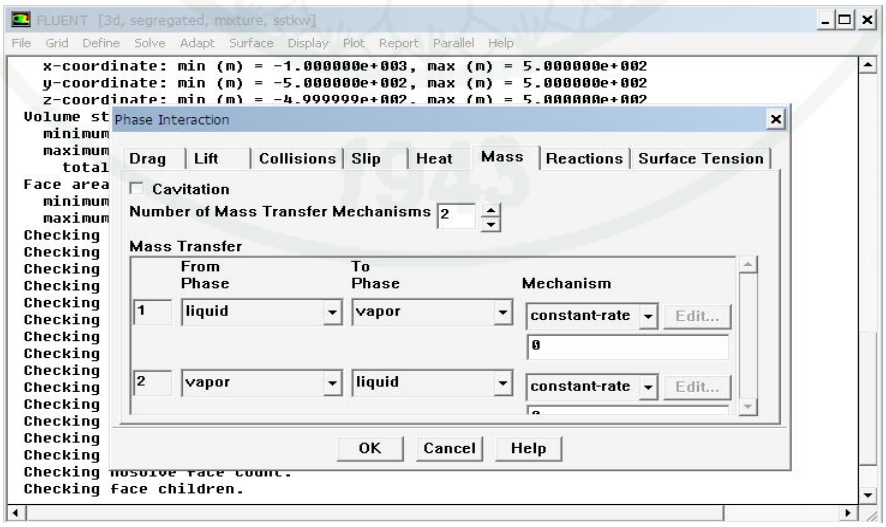
Appendix Figure B21 Procedure of using FLUENT step 21

22. Phase Interaction select Mass

Number of Mass Transfer Mechanisms select enter: 2

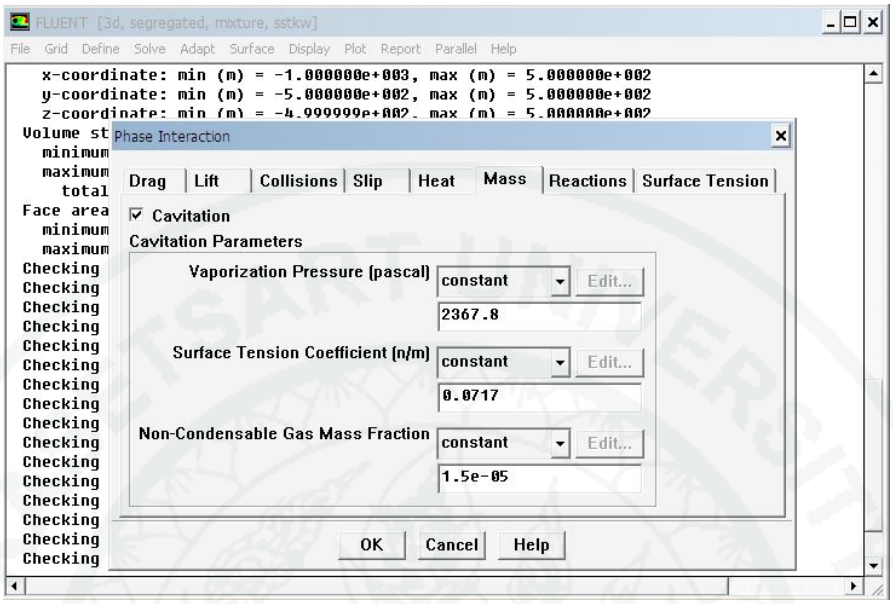
Mass Transfer

1. From Phase select liquid and To Phase select vapor
2. From Phase select vapor and To Phase select liquid



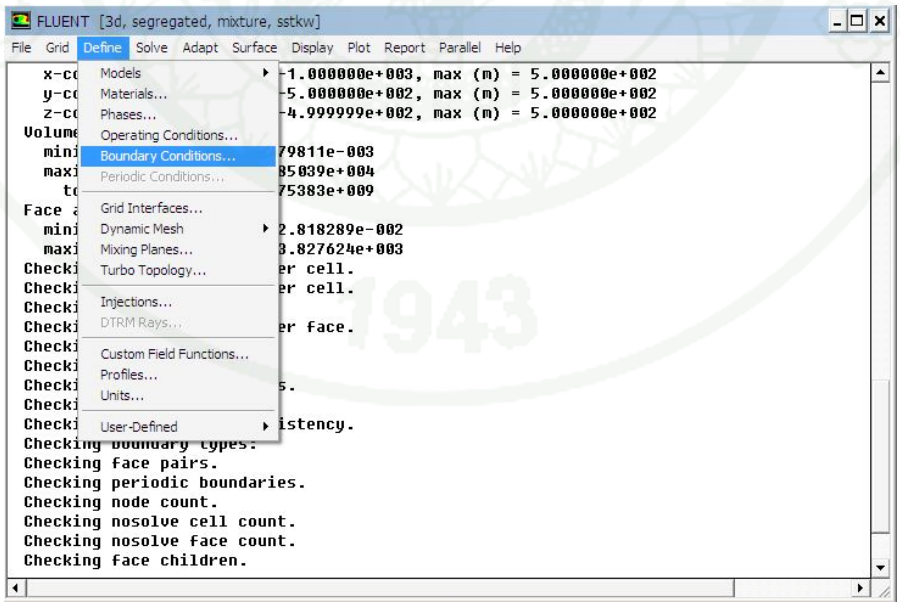
Appendix Figure B22 Procedure of using FLUENT step 22

23. Click Cavitation and click OK



Appendix Figure B23 Procedure of using FLUENT step 23

24. Define-Boundary Conditon

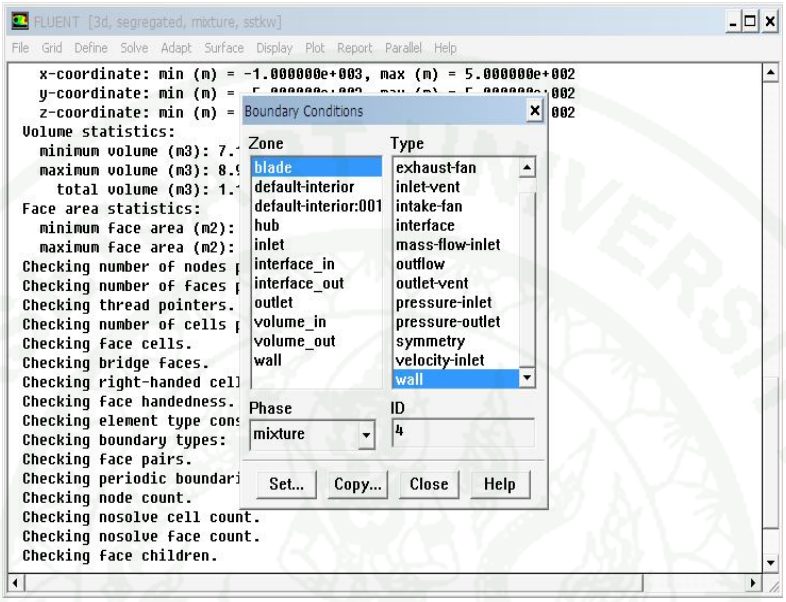


Appendix Figure B24 Procedure of using FLUENT step 24

25. Zone select blade

    Type select wall

    Click set



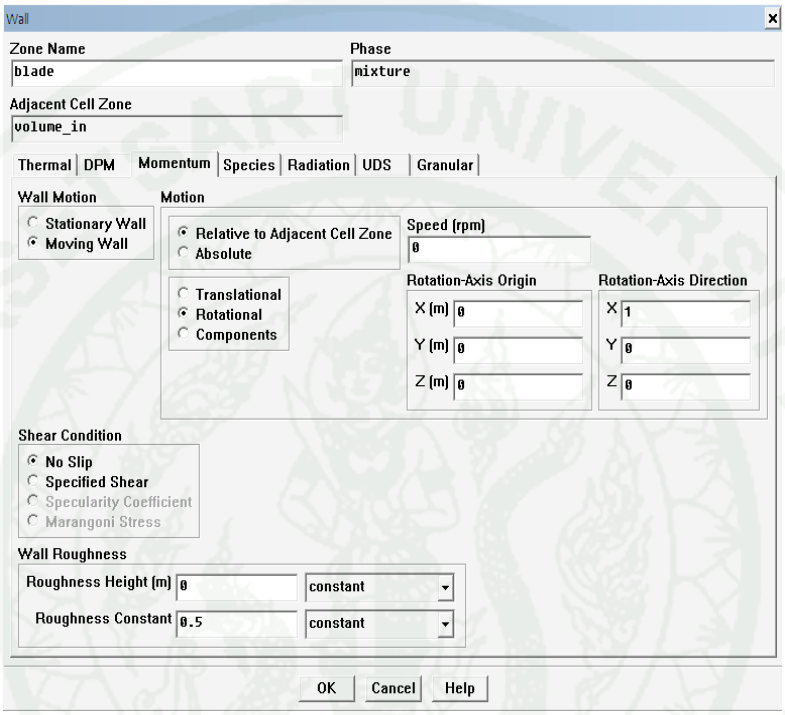
Appendix Figure B25 Procedure of using FLUENT step 25

26. Wall Motion select Moving Wall and click Rotational

Rotation-Axis Direction enter

x = 1 y = 0 z = 0

Click OK

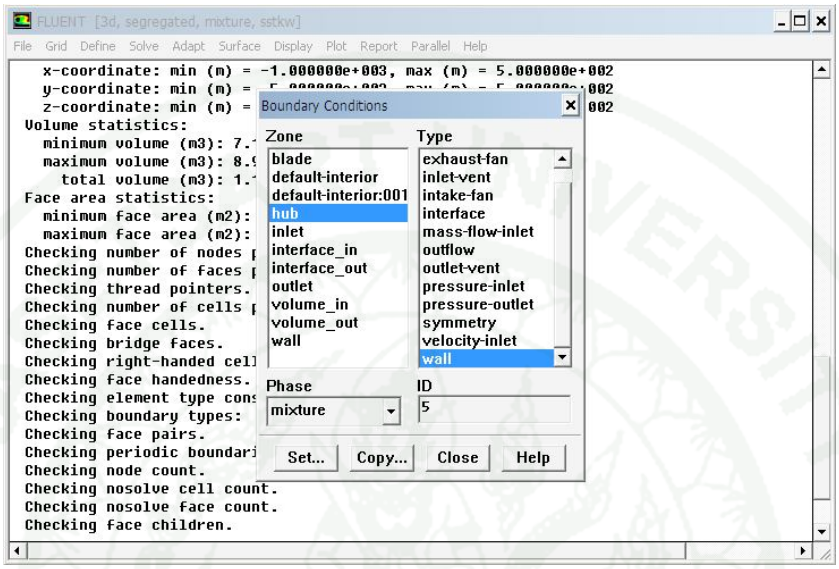


Appendix Figure B26 Procedure of using FLUENT step 26

27. Zone select hub

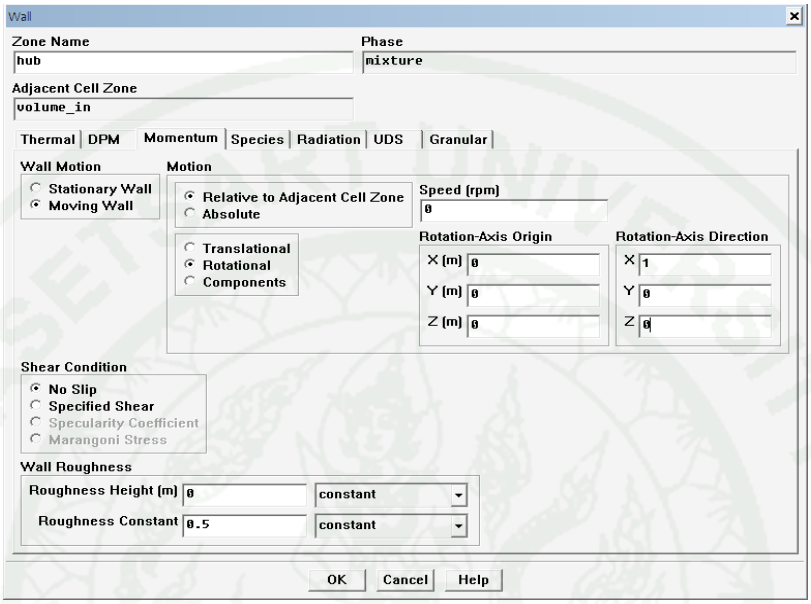
    Type select wall

    Click set



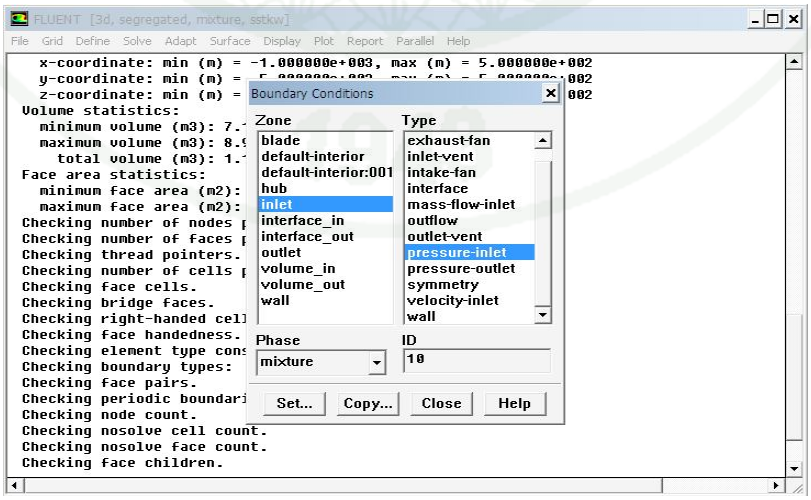
Appendix Figure B27 Procedure of using FLUENT step 27

28. Wall Motion select Moving Wall and click Rotational
    Rotation-Axis Direction enter x =1 y =0 z =0
    Click OK



Appendix Figure B28 Procedure of using FLUENT step 28

29. Zone select Inlet and Type select Pressure-inlet
    Click set



Appendix Figure B29 Procedure of using FLUENT step 29

30. Gauge Total Pressure (pascal) enter-58619

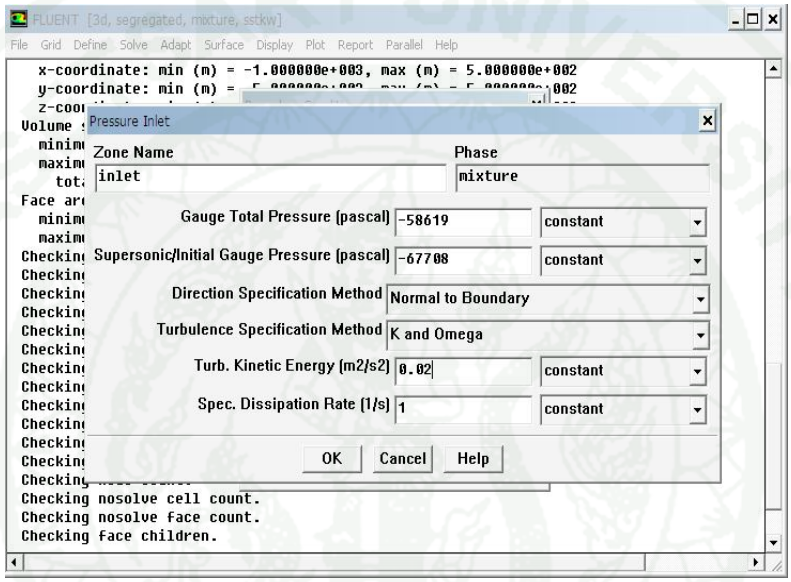
    Supersonic/initial Gauge Pressure (pascal) enter-67708

    Turbulence Specification Method select K and Omega

    Tub. Kinetic Energy (m2/s2) enter 0.02

    Spec. Dissipation (1/s) enter 1

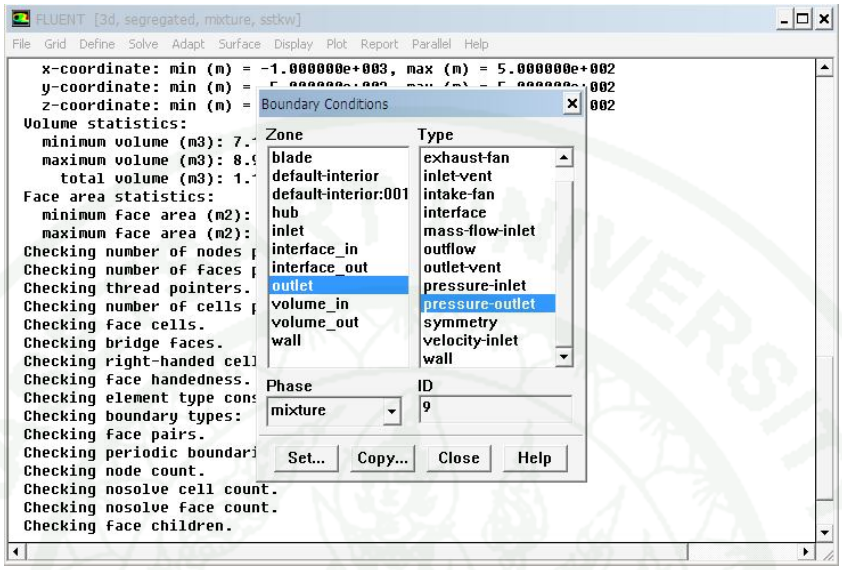
    Click Ok



Appendix Figure B30 Procedure of using FLUENT step 30

31. Zone select Outlet and Type select Pressure-outlet

    Click set



Appendix Figure B31 Procedure of using FLUENT step 31

32. Gauge Pressure (pascal) enter-67708

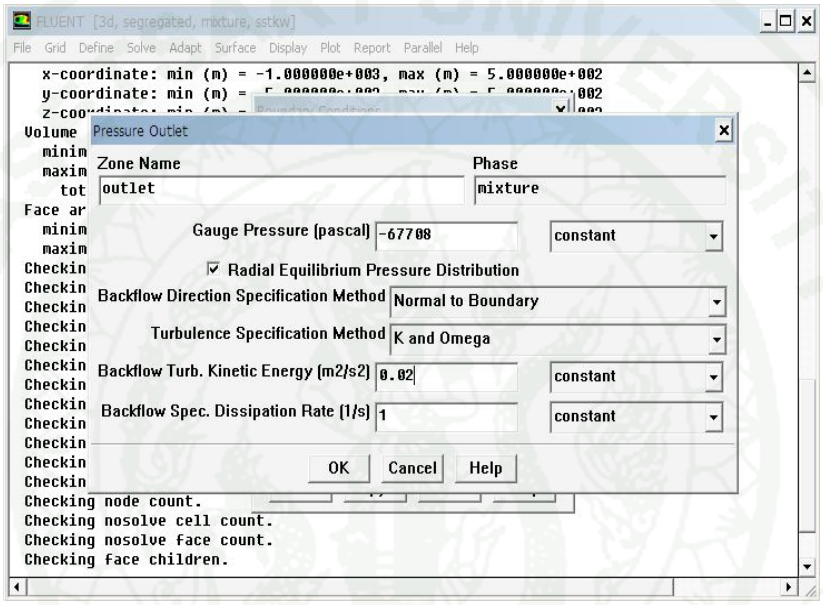
    Select Radial Equilibrium Pressure Distribution

    Turbulence Specification Method select K and Omega

    Tub. Kinetic Energy (m2/s2) enter 0.02

    Spec. Dissipation (1/s) enter 1

    Click Ok

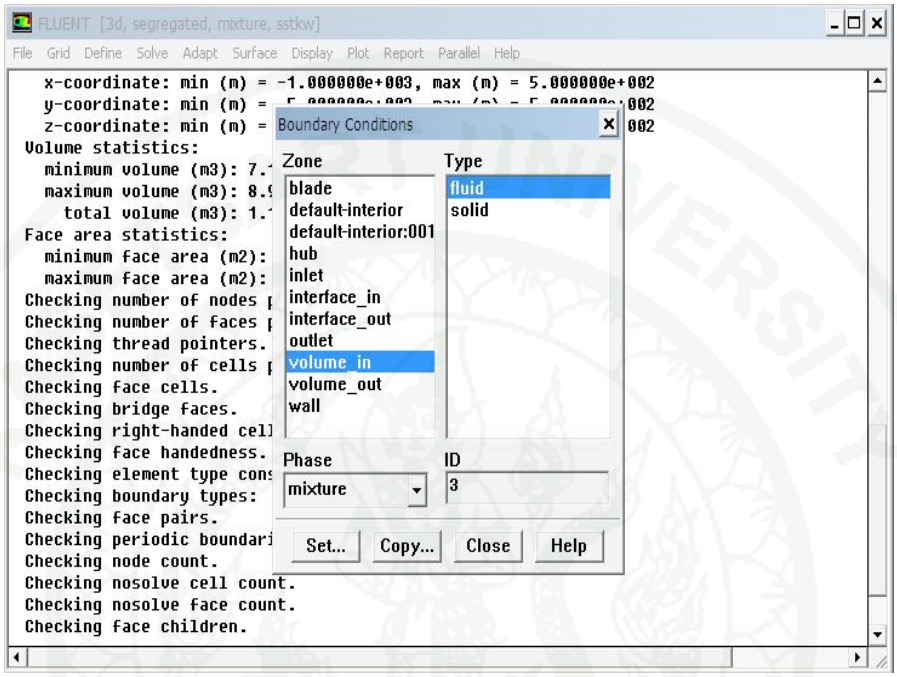


Appendix Figure B32 Procedure of using FLUENT step 32

33. Zone select Volume_in

    Type select Fluid

    Click Set

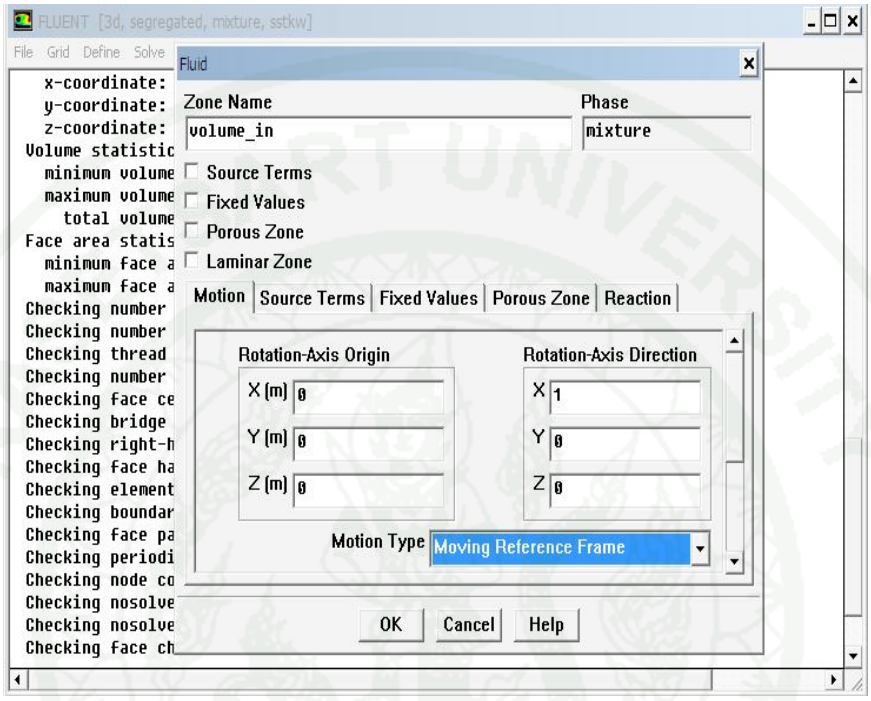


Appendix Figure B33 Procedure of using FLUENT step 33

34. Rotation-Axis Direction

Enter x = 1 y = 0 z = 0

Motion Type select Moving Reference frame



Appendix Figure B34 Procedure of using FLUENT step 34

35. Rotation Velocity Speed (rpm) enter: 1200

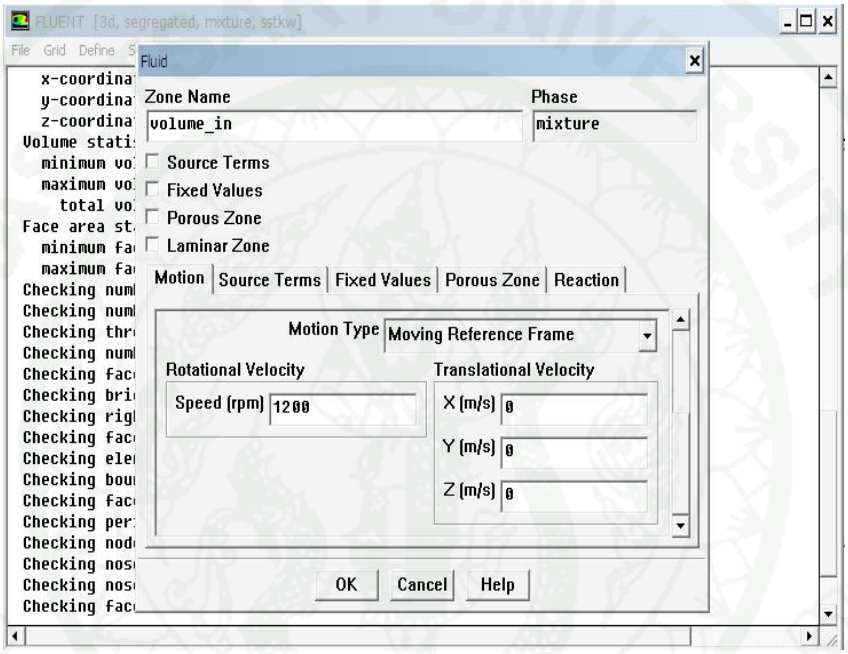
    Translational Velocity

    X(m/s) Enter 0

    Y(m/s) Enter 0

    Z(m/s) Enter 0

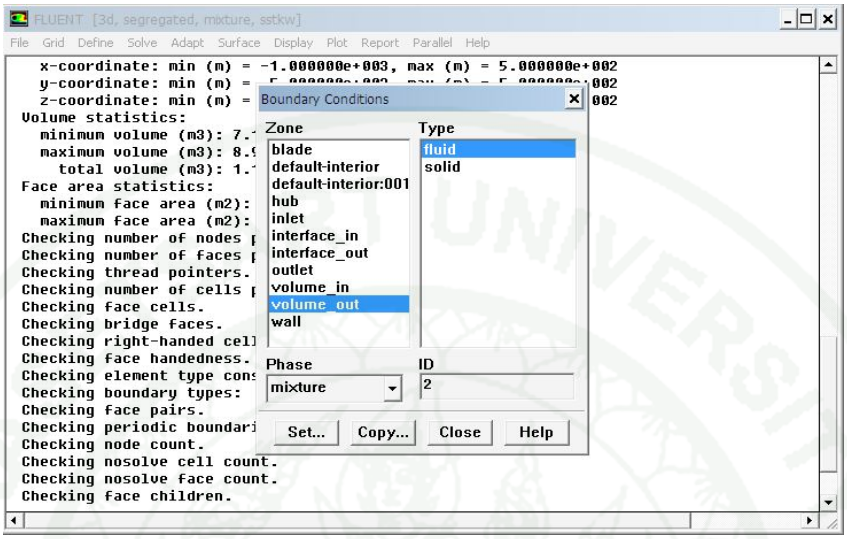
    Click OK



Appendix Figure B35 Procedure of using FLUENT step 35

36. Zone select Volume_out and Type select Fluid

    Click Set

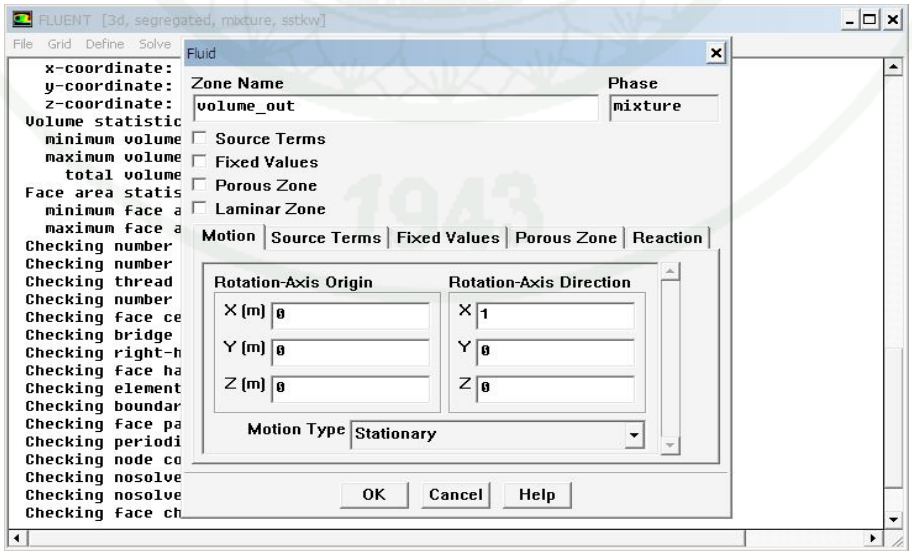


Appendix Figure B36 Procedure of using FLUENT step 36

37. Rotation-Axis Diraction

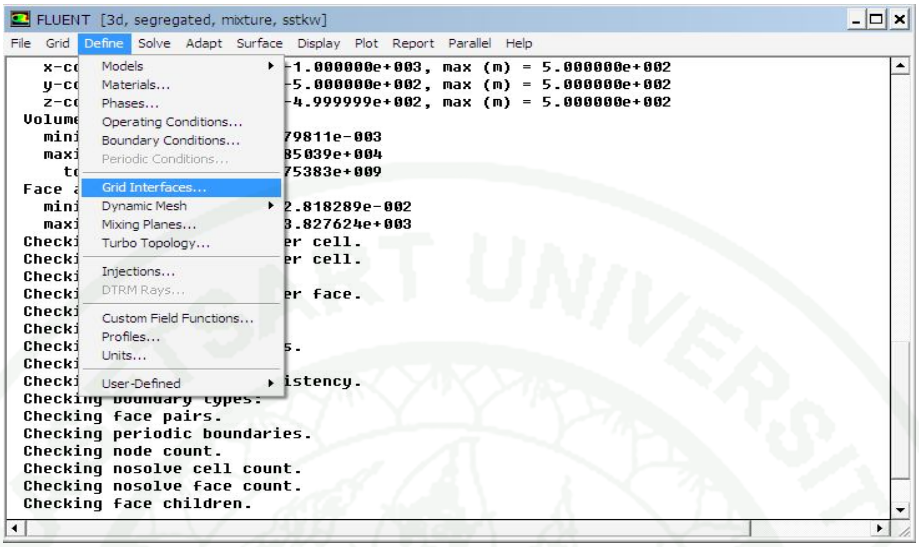
    Enter x = 1 y = 0 z = 0

    Click OK



Appendix Figure B37 Procedure of using FLUENT step 37

38. Define-Grid Interfaces



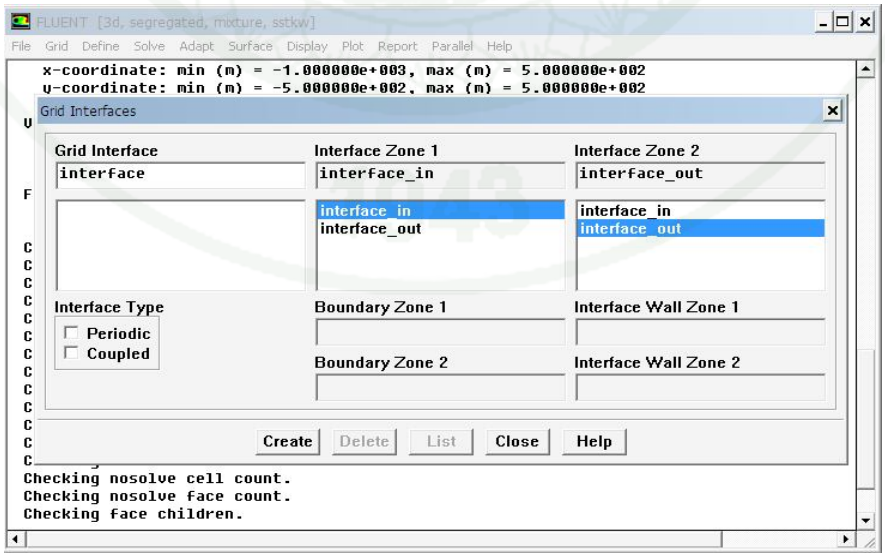
Appendix Figure B38 Procedure of using FLUENT step 38

39. Grid Interface = Interface

Interface Zone 1 = interface_in

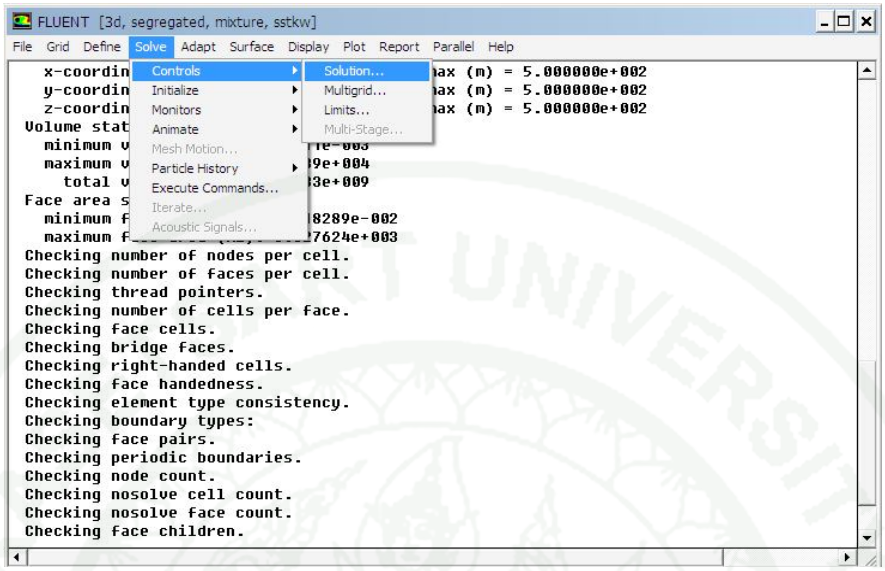
Interface Zone 2 = interface_out

Click Create



Appendix Figure B39 Procedure of using FLUENT step 39

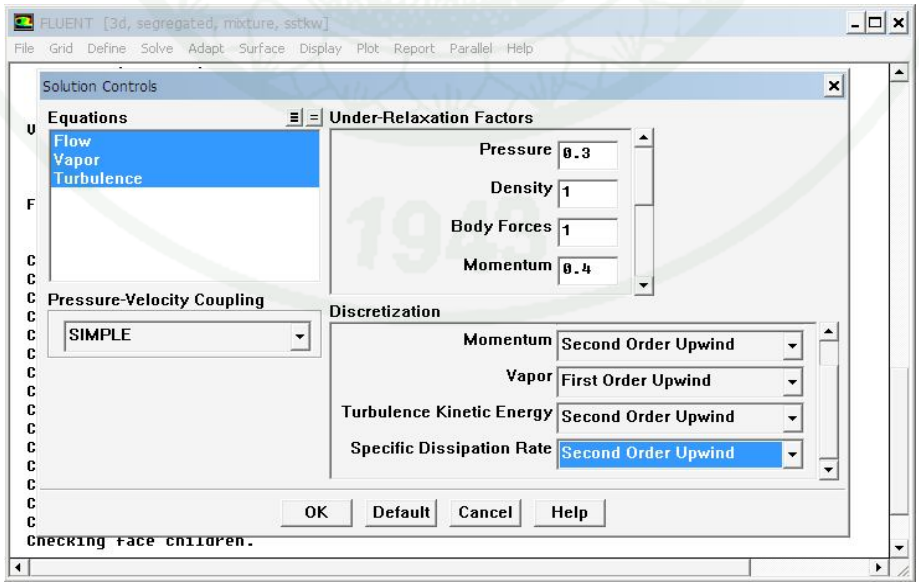
40. Solve-Controls-Solution



Appendix Figure B40 Procedure of using FLUENT step 40

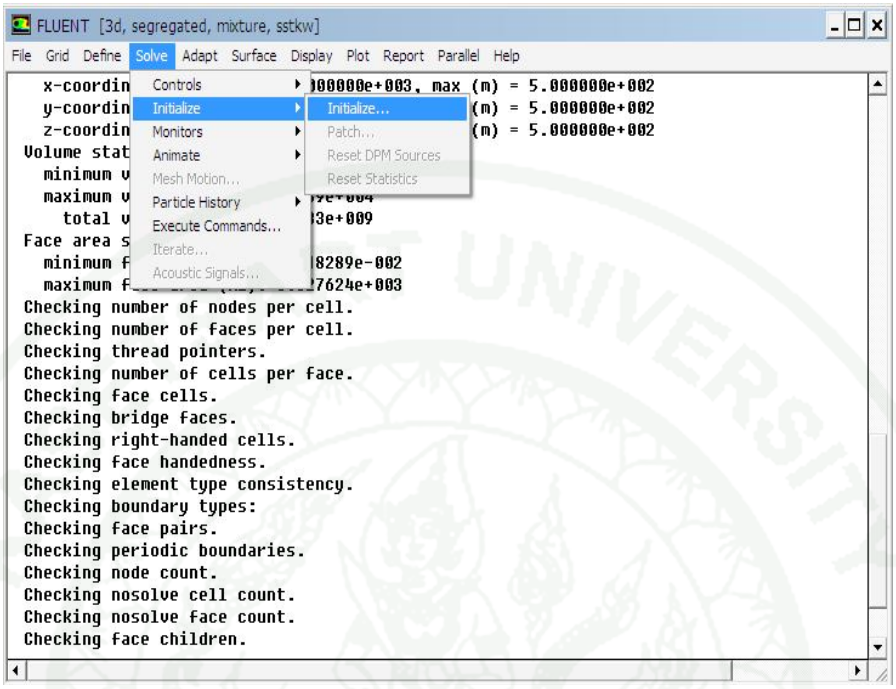
41. Pressure-velocity coupling, set SIMPLE

    Discretization; momentum, turbulence kinetic energy and turbulence dissipation rate set Second Order Upwind



Appendix Figure B41 Procedure of using FLUENT step 41

42. Solve-Initialize-Initialize



Appendix Figure B42 Procedure of using FLUENT step 42

43. Compute Form = inlet

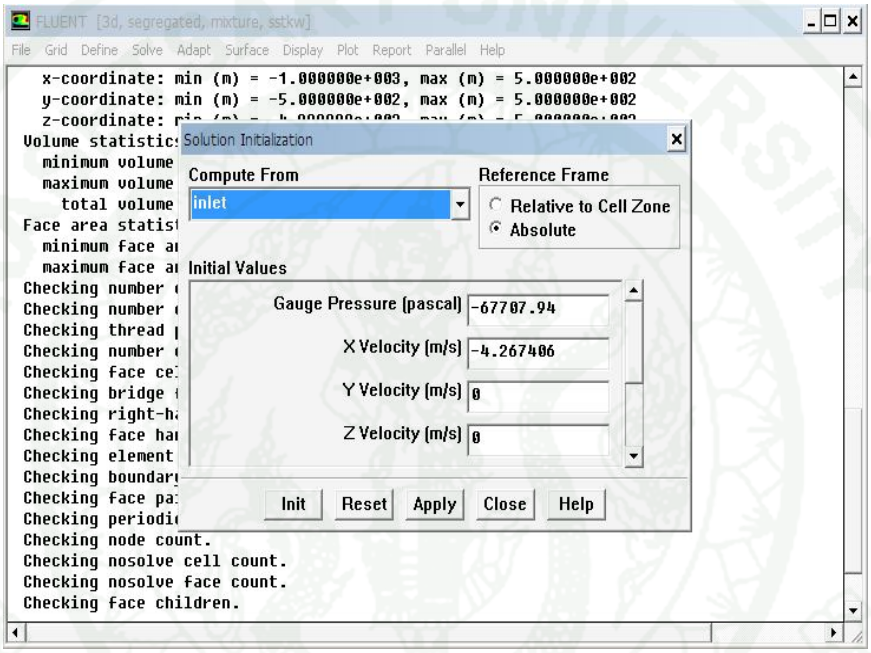
Reference Frame = Absolute

Gauge Pressure (pascal) = -67708

X Velocity (m/s) = -4.2672

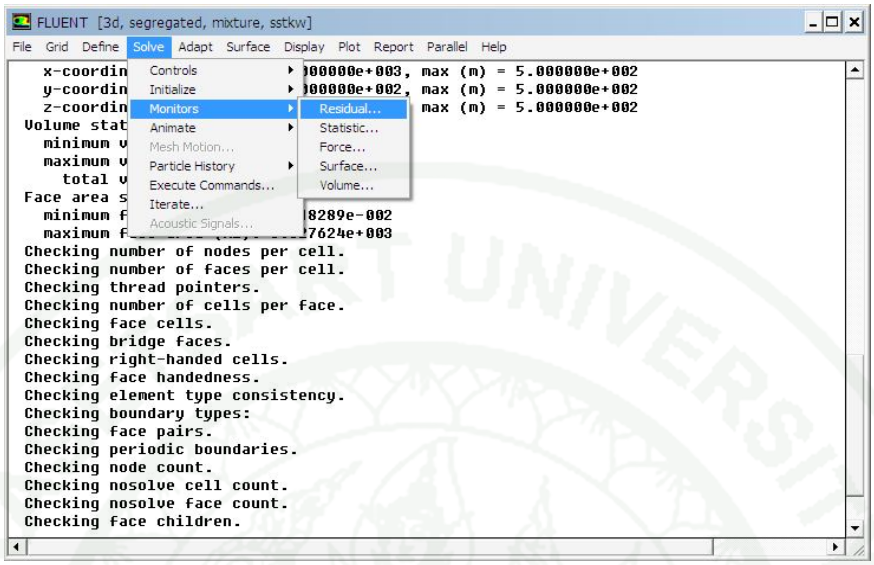
Phase-2 Vapor = 0.001

Click init



Appendix Figure B43 Procedure of using FLUENT step 43

44. Solve-Monitors-Residual

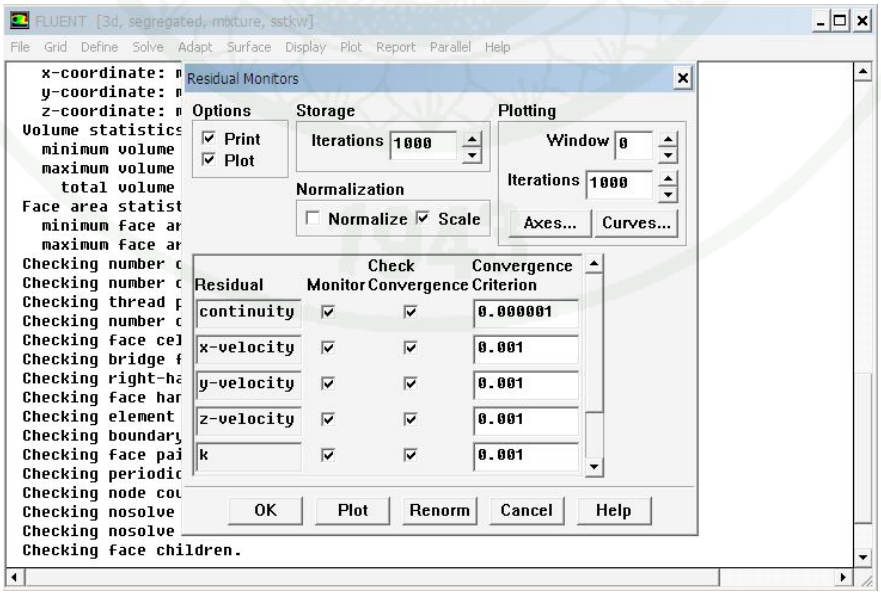


Appendix Figure B44 Procedure of using FLUENT step 44

45. Option select Print and Plot

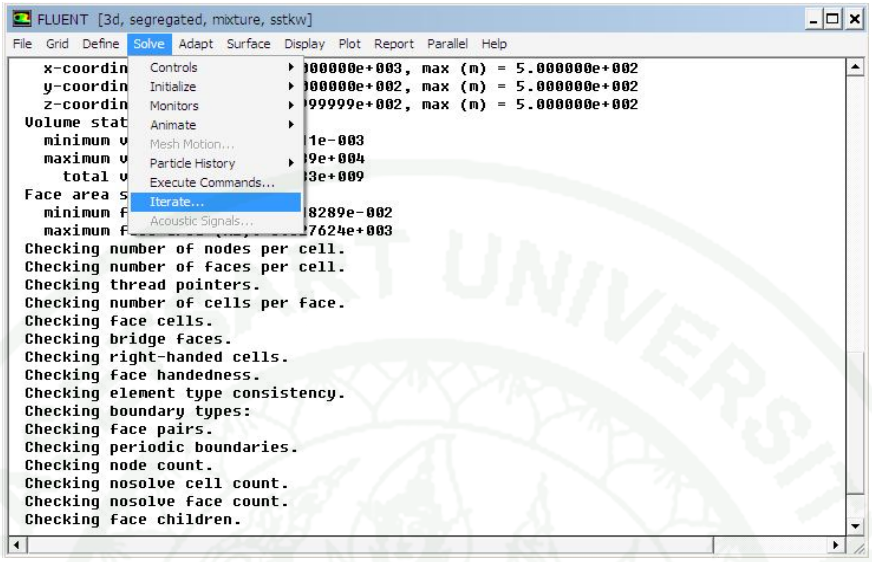
Residual-continuity Convergence Criterion = 0.000001

Click OK



Appendix Figure B45 Procedure of using FLUENT step 45

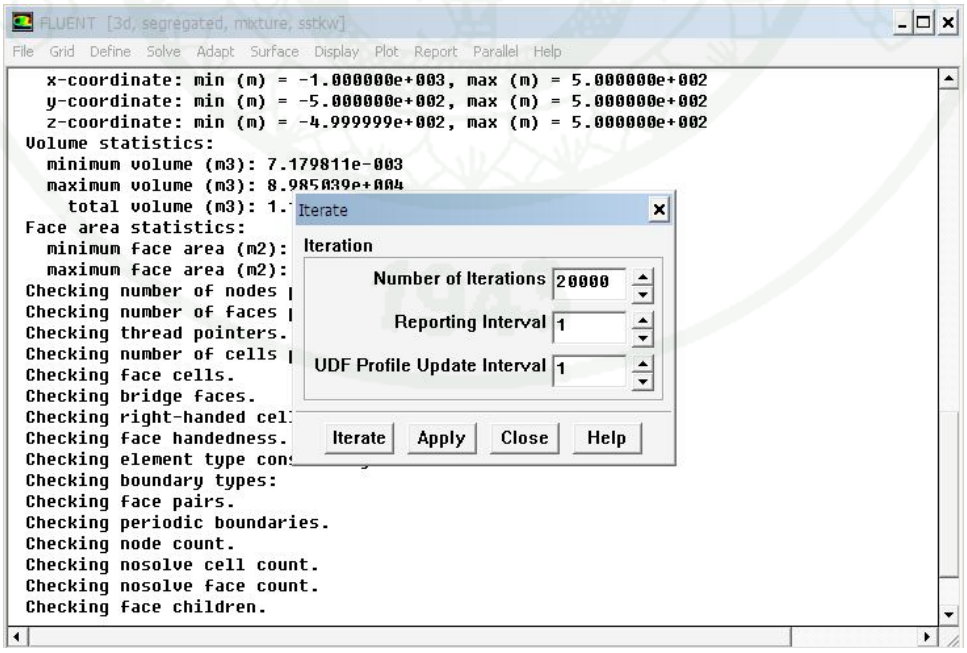
46. Solve-Iterate



Appendix Figure B46 Procedure of using FLUENT step 46

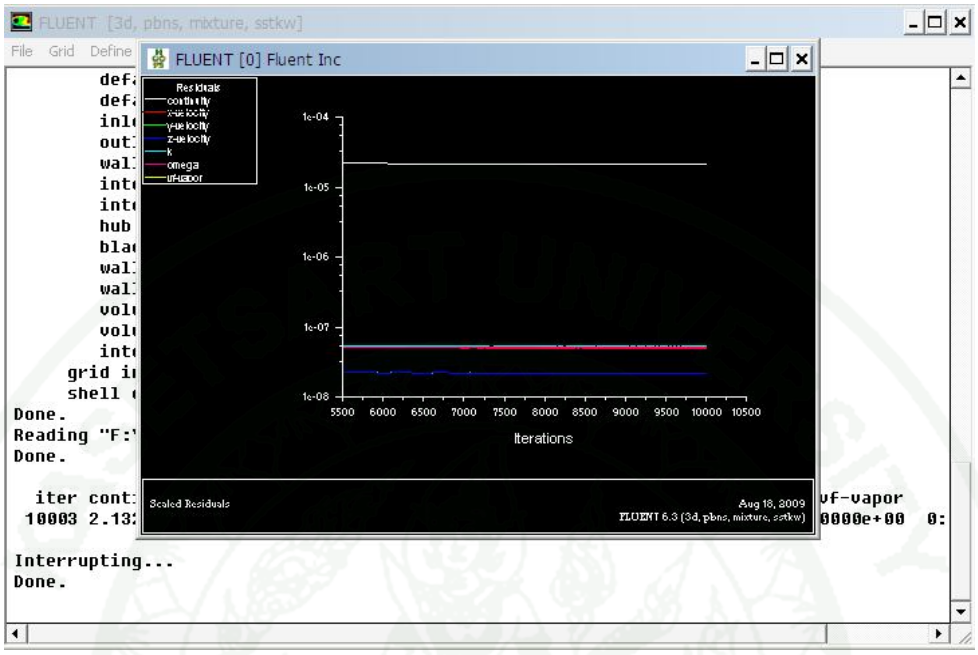
47. Number of Interation = 20000

Click Interate



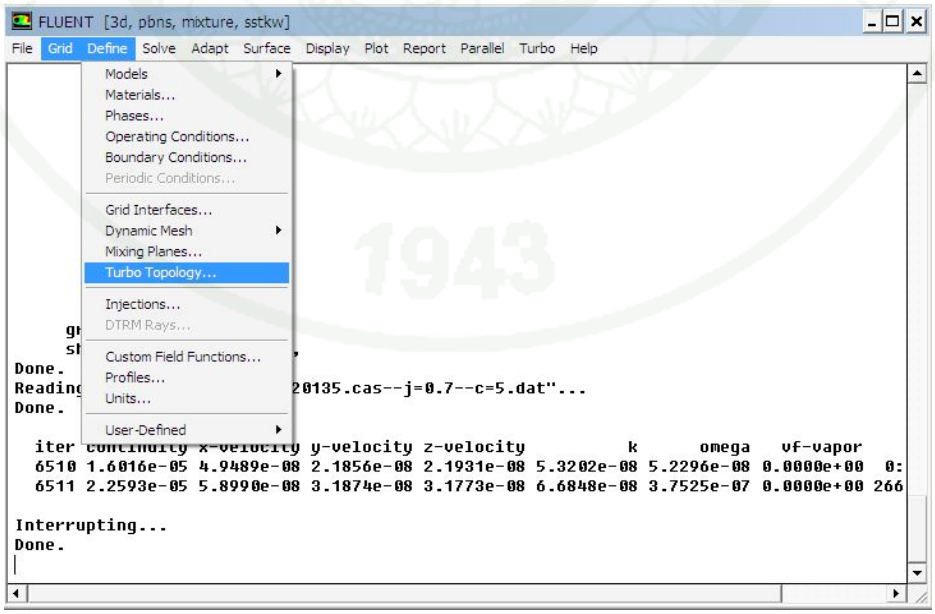
Appendix Figure B47 Procedure of using FLUENT step 47

48. When Solution is Converged will effective follow facture



Appendix Figure B48 Procedure of using FLUENT step 48

49. Check Thrust and Torque by click Define-Turbo Topology

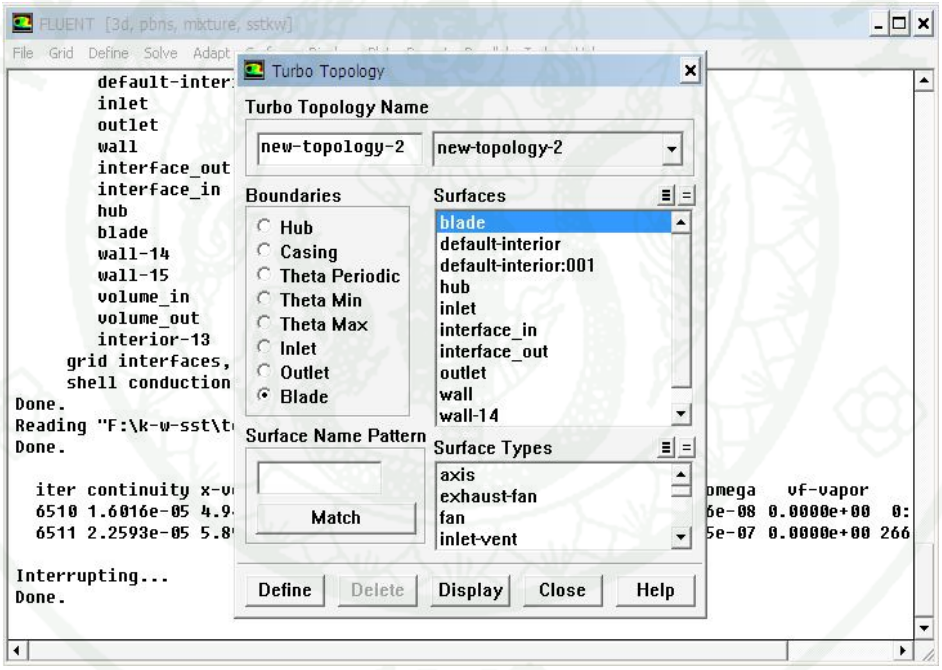


Appendix Figure B49 Procedure of using FLUENT step 49

50. Choose a pair

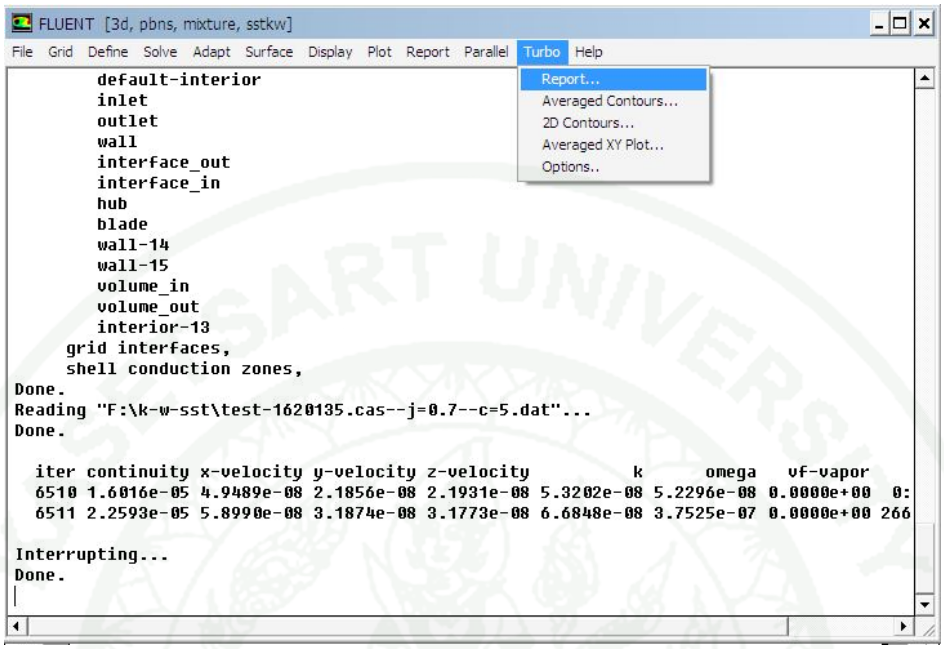
Boundaries	**Surfaces**
Hub	Hub
Casing	Interface_Out
Theda Periodic	default-interior: 001
	(select Volume in)
Inlet	Inlet
Outlet	Outlet
Blade	Blade

Click Define



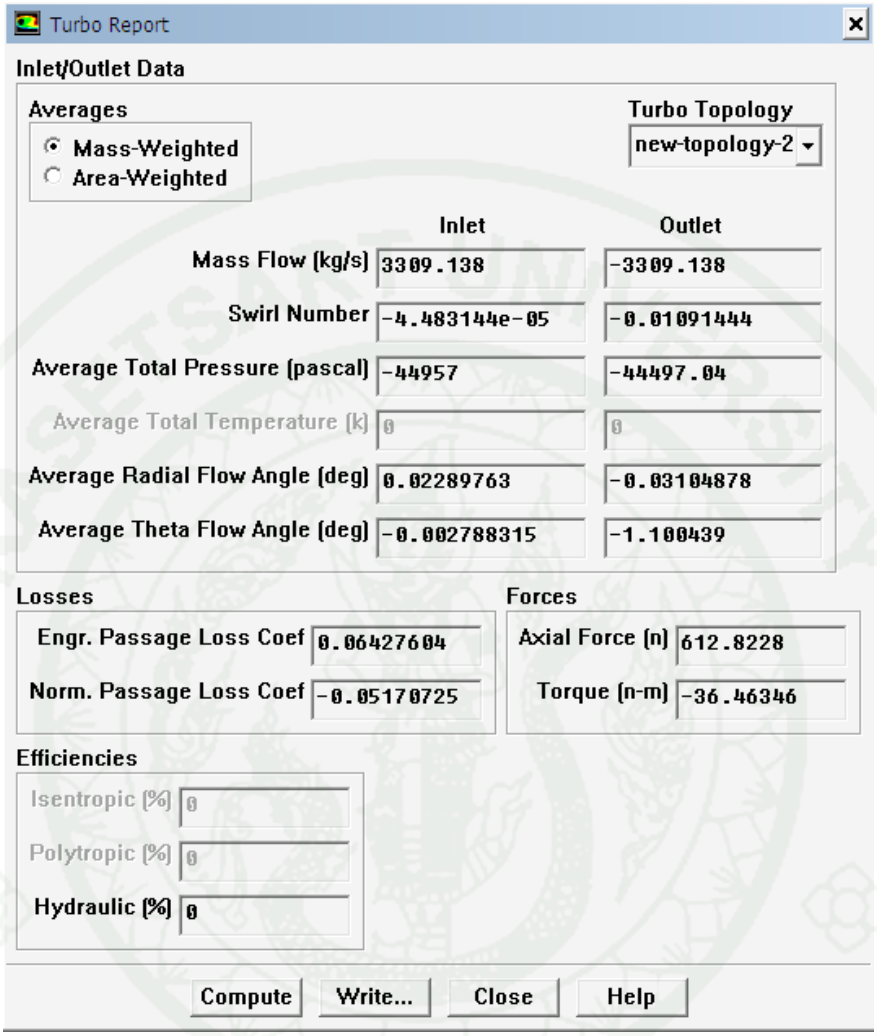
Appendix Figure B50 Procedure of using FLUENT step 50

51. Turbo-Report



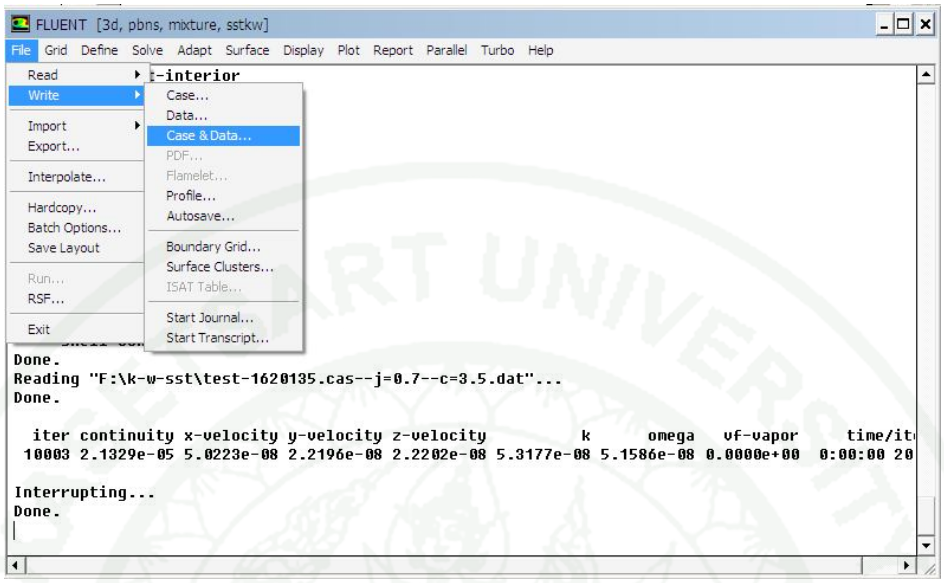
Appendix Figure B51 Procedure of using FLUENT step 51

52. Click Compute



Appendix Figure B52 Procedure of using FLUENT step 52

53. Select write → Case & Data



Appendix Figure B53 Procedure of using FLUENT step 53

Appendix C

Using GAMBIT and FLUENT Parallel on Linux System

Using GAMBIT and Parallel FLUENT on Linux System

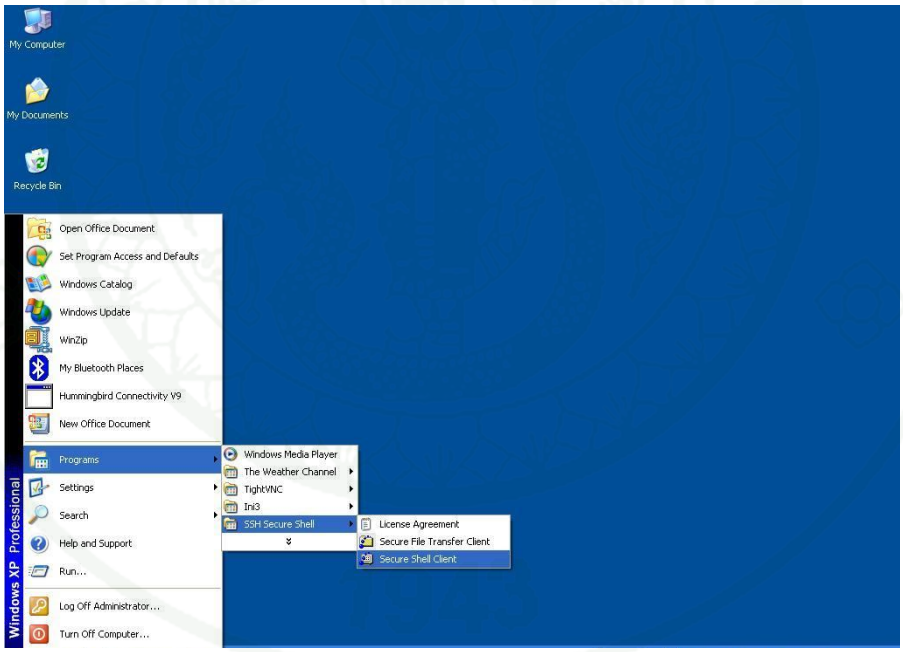
Using GAMBIT and FLUENT software on Tera Cluster (Namely "Thaigrid") at Thai National Grid Centre.

(a) Ask for account Tera Cluster at http://tera.thaigrid.or.th/drupal/th

(b) Installation SSH SecureShellClient and Tightvnc programs (download: http://tera.thaigrid.or.th/drupal/th/user_manual/access)

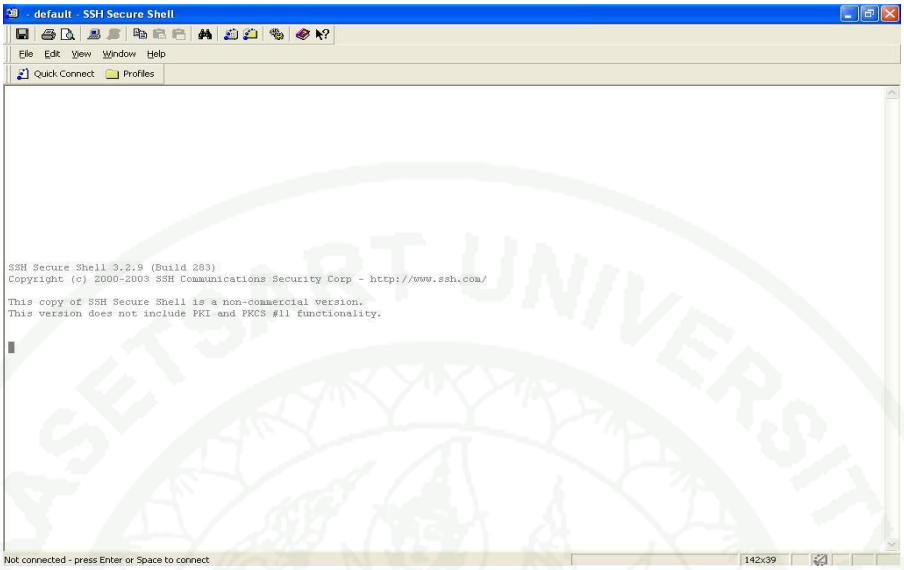
Open to use the window for TightVNC program

1. Start → All program → SSH SecureShellClient → SecureShellClient



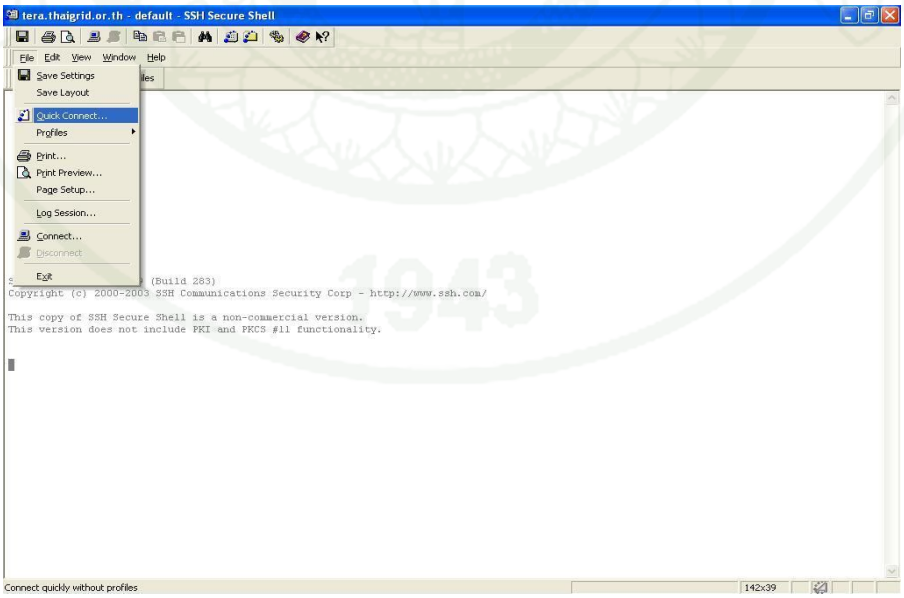
Appendix Figure C1 Procedure of using GAMBIT and parallel FLUENT on Linux system step 1

2. SecureShellClient monitor



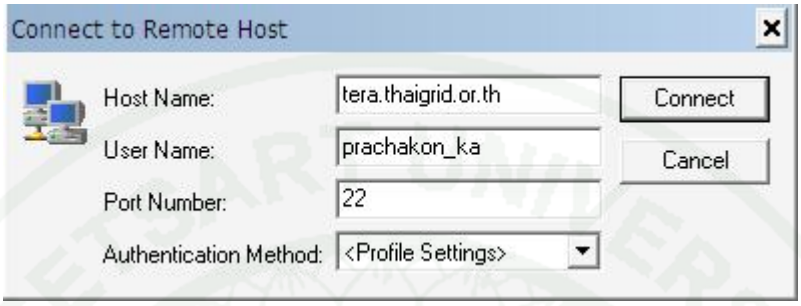
Appendix Figure C2 Procedure of using GAMBIT and parallel FLUENT on Linux system step 2

3. Select file → Quick connect



Appendix Figure C3 Procedure of using GAMBIT and parallel FLUENT on Linux system step 3

4. Connect to remote host → Connect
   (a) Host name, set tera.thaigrid.or.th
   (b) User name, set your account.

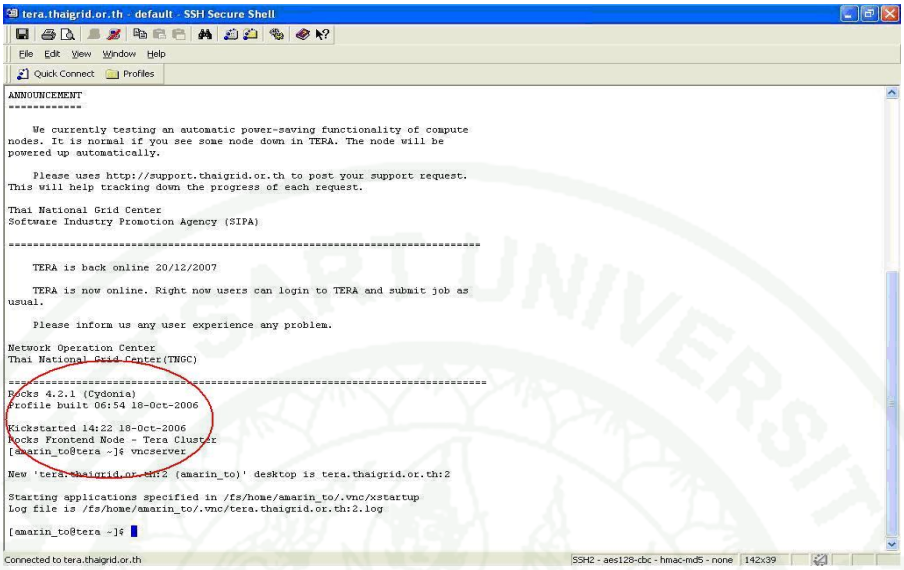


Appendix Figure C4 Procedure of using GAMBIT and parallel FLUENT on Linux system step 4

5. Enter password → OK

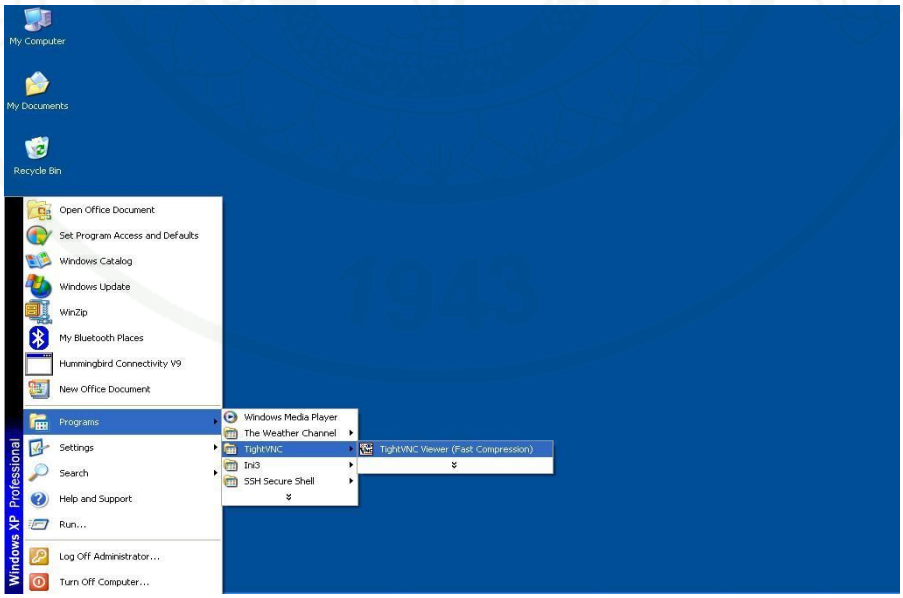
Appendix Figure C5 Procedure of using GAMBIT and parallel FLUENT on Linux system step 5

6. Input “vncserver” (example: tera.thaigrid.or.th:2, using section 2)



Appendix Figure C6 Procedure of using GAMBIT and parallel FLUENT on Linux system step 6

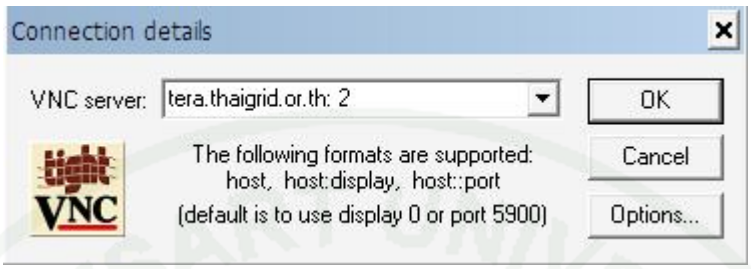
7. Start → All program → TightVNC → TightVNC viewer (fast compression)



Appendix Figure C7 Procedure of using GAMBIT and parallel FLUENT on Linux system step 7

8. Connection details → OK

   (a) VNC server, set tera.thaigrid.or.th:2



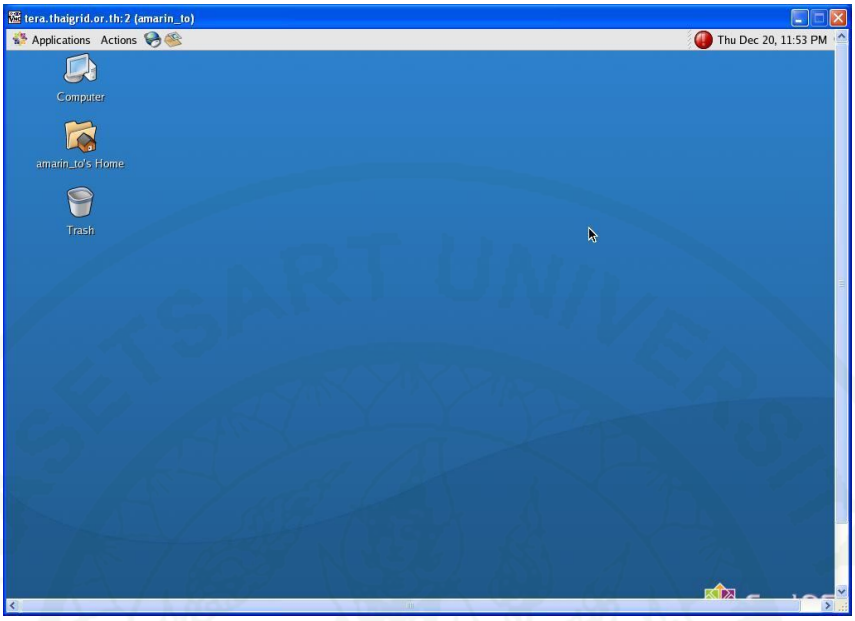
Appendix Figure C8 Procedure of using GAMBIT and parallel FLUENT on Linux system step 8

9. VNC authentication → OK

   (a) Session password, set your password.

Appendix Figure C9 Procedure of using GAMBIT and parallel FLUENT on Linux system step 9

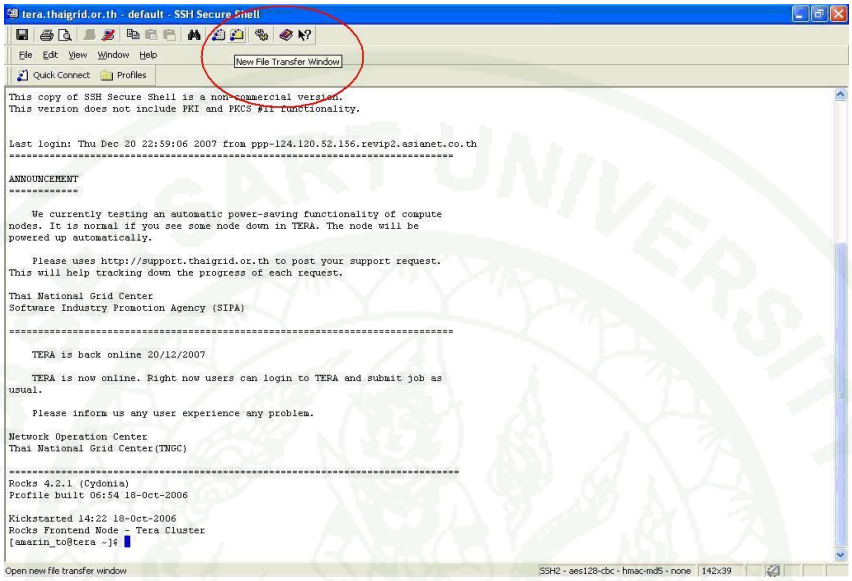
10. Linux monitor



Appendix Figure C10 Procedure of using GAMBIT and parallel FLUENT on Linux system step 10

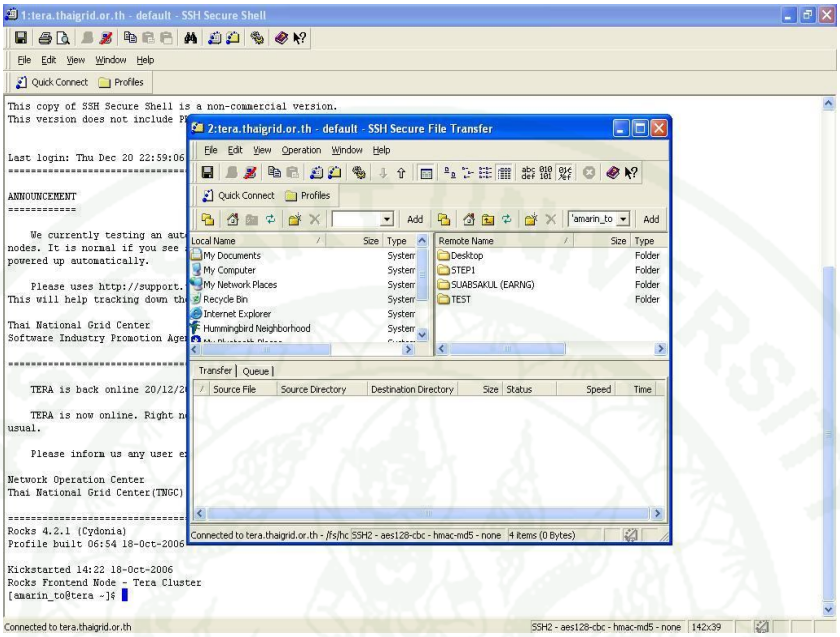
Transferring file for apply on Tera Cluster

1. Select new file transfer window



Appendix Figure C11 Procedure of using GAMBIT and parallel FLUENT on Linux system step 11

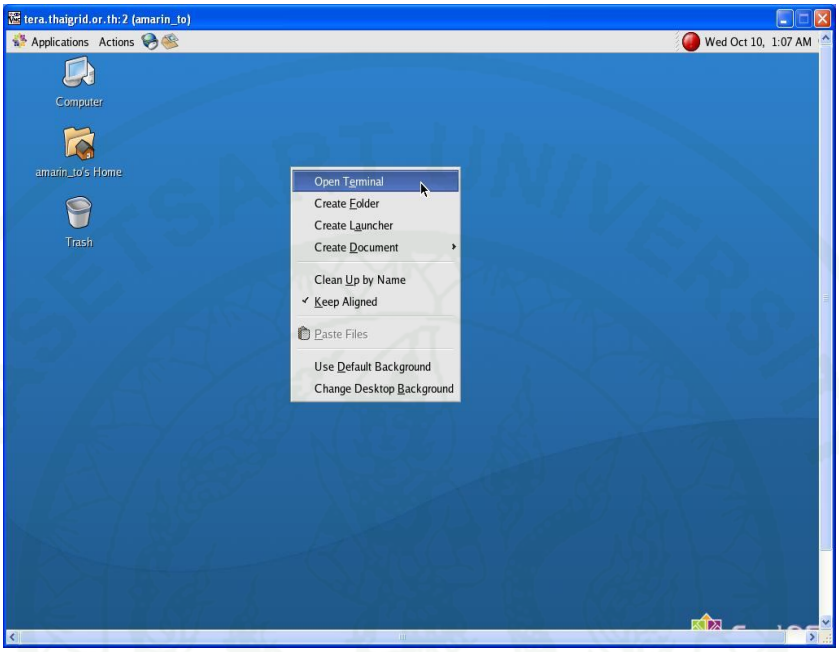
2. Left folder is the part of user at the PC and Right folder is the part of account on Tera.



Appendix Figure C12 Procedure of using GAMBIT and parallel FLUENT on Linux system step 12

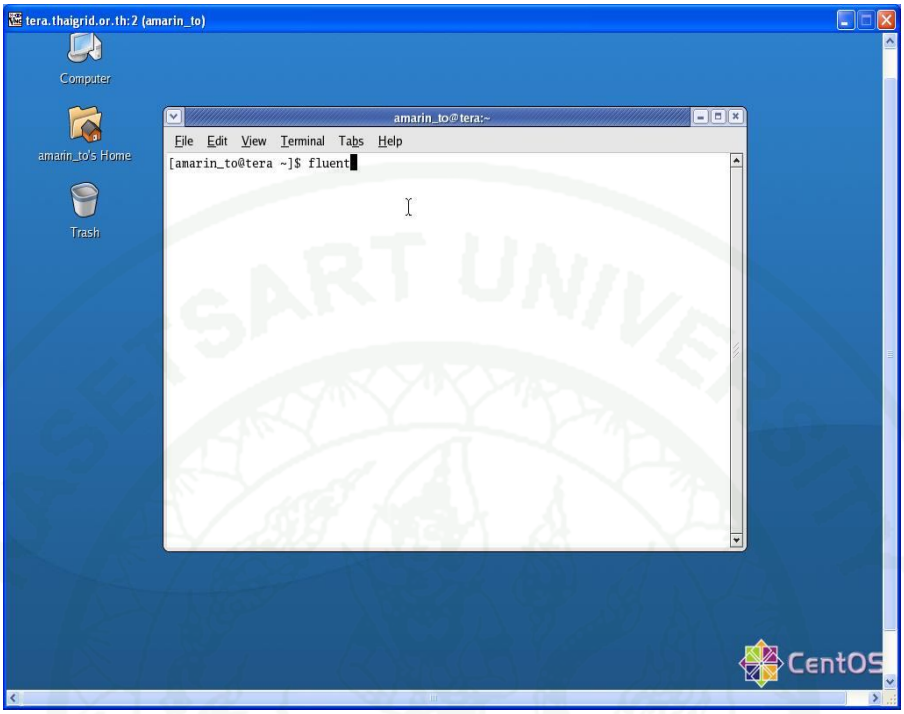
Using GAMBIT and FLUENT software on Linux system

1. Open terminal



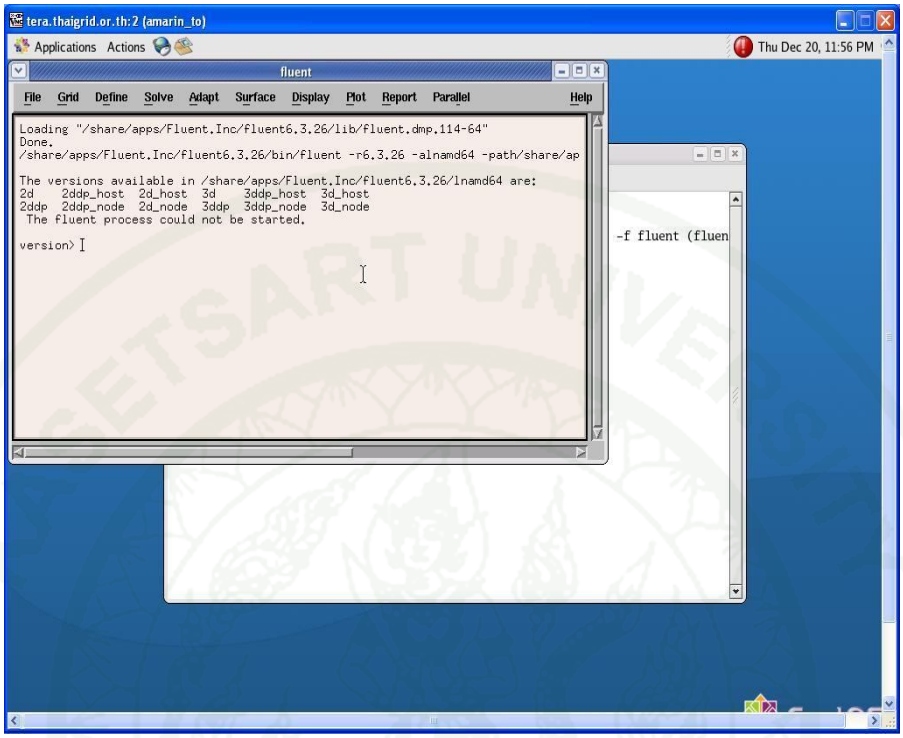
Appendix Figure C13 Procedure of using GAMBIT and parallel FLUENT on Linux system step 13

2. Input “fluent or gambit”



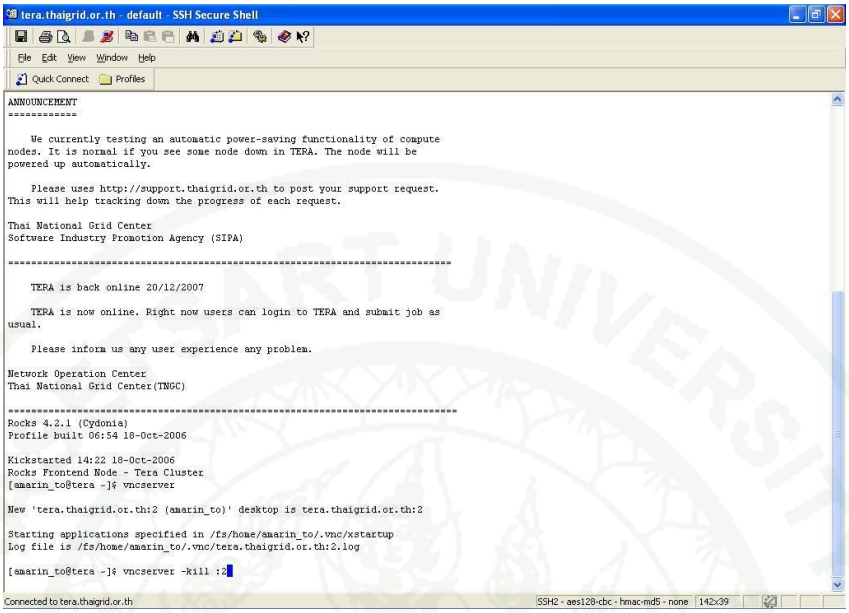
Appendix Figure C14 Procedure of using GAMBIT and parallel FLUENT on Linux system step 14

3. FLUENT monitor



Appendix Figure C15 Procedure of using GAMBIT and parallel FLUENT on Linux system step 15

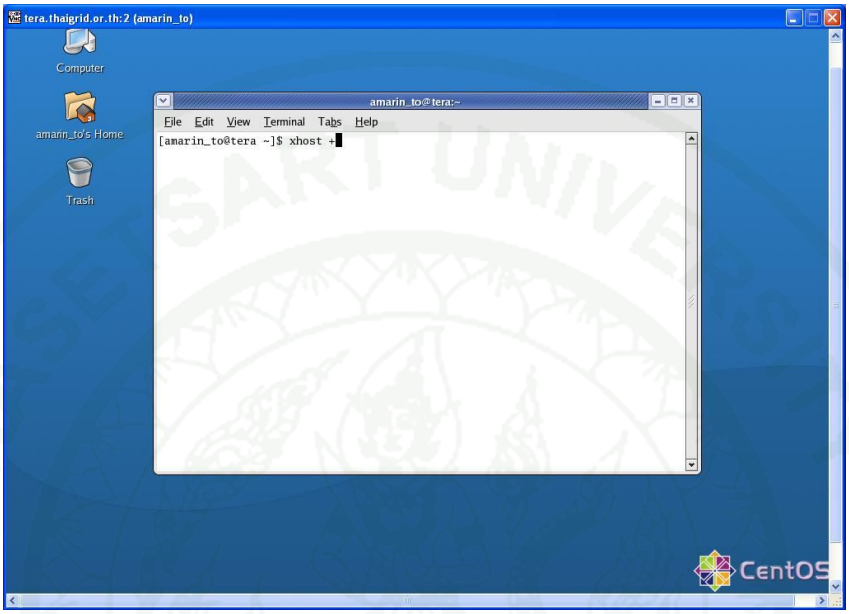
4. Input “vncserver –kill :2” for close TightVNC program



Appendix Figure C16 Procedure of using GAMBIT and parallel FLUENT on Linux system step 16

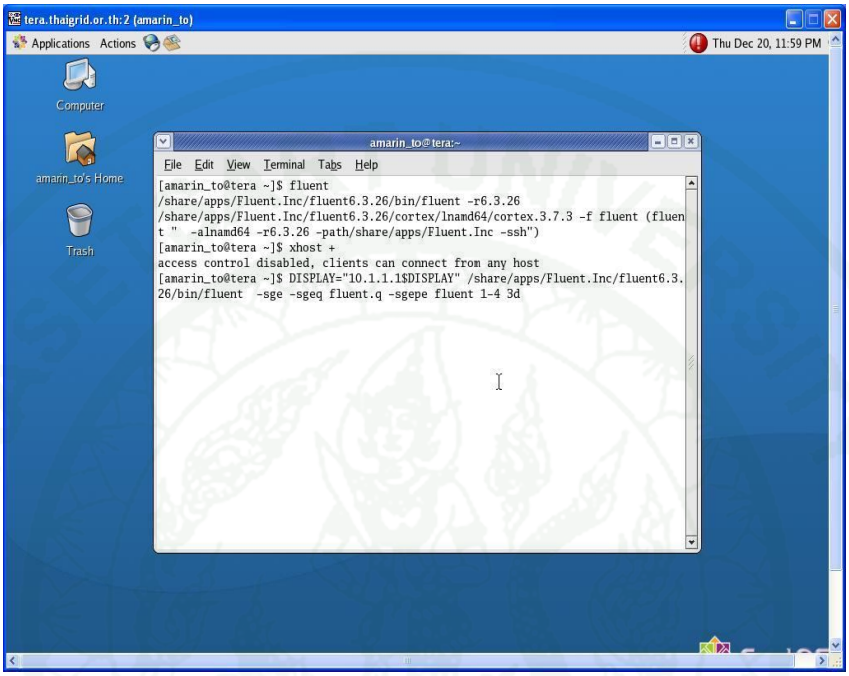
Using parallel FLUENT on Linux system

1. Terminal monitor, set “xhost +”



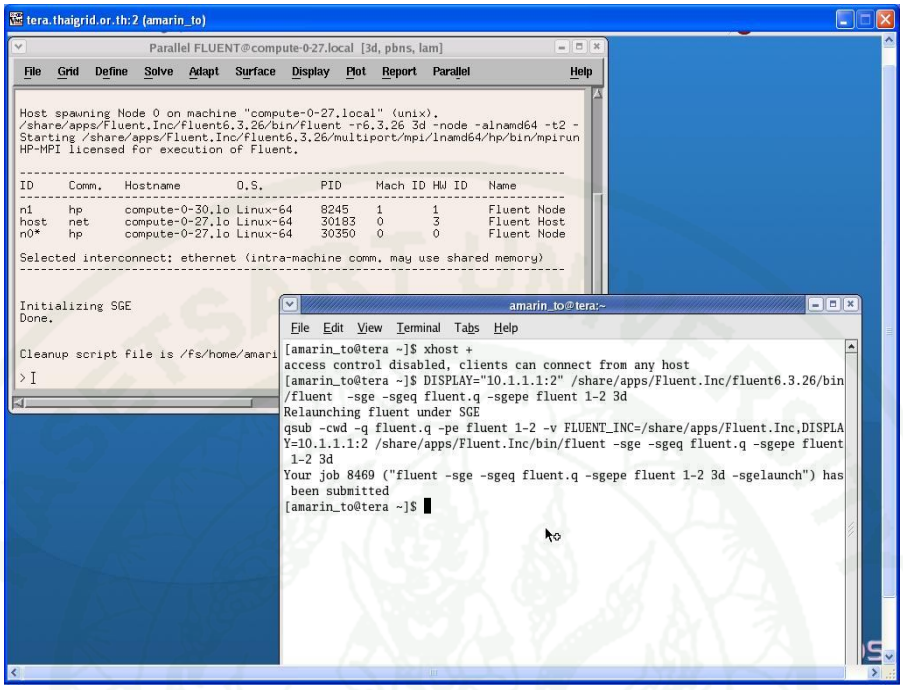
Appendix Figure C17 Procedure of using GAMBIT and parallel FLUENT on Linux system step 17

2. Input “DISPLAY= “10.1.1.1:$DISPLAY” /share/apps/Fluent.Inc/fluent6.3.26/ bin/fluent –sge –sgeq fluent.q –sgepe fluent 1-20 3d” (for 3D and 20 nodes)



Appendix Figure C18 Procedure of using GAMBIT and parallel FLUENT on Linux system step 18

3. Parallel FLUENT monitor



Appendix Figure C19 Procedure of using GAMBIT and parallel FLUENT on Linux system step 19

Appendix D

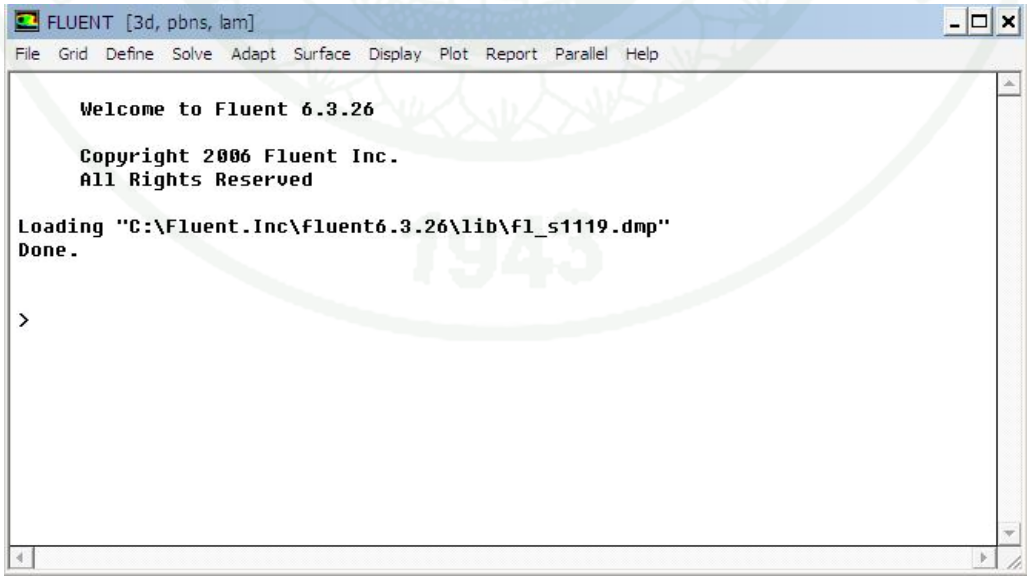
Using User-defined Function (UDF) on Windows and Linux System

Using User-defined Function (UDF) on Windows and Linux System

1. Select FLUENT version → 3d → Run

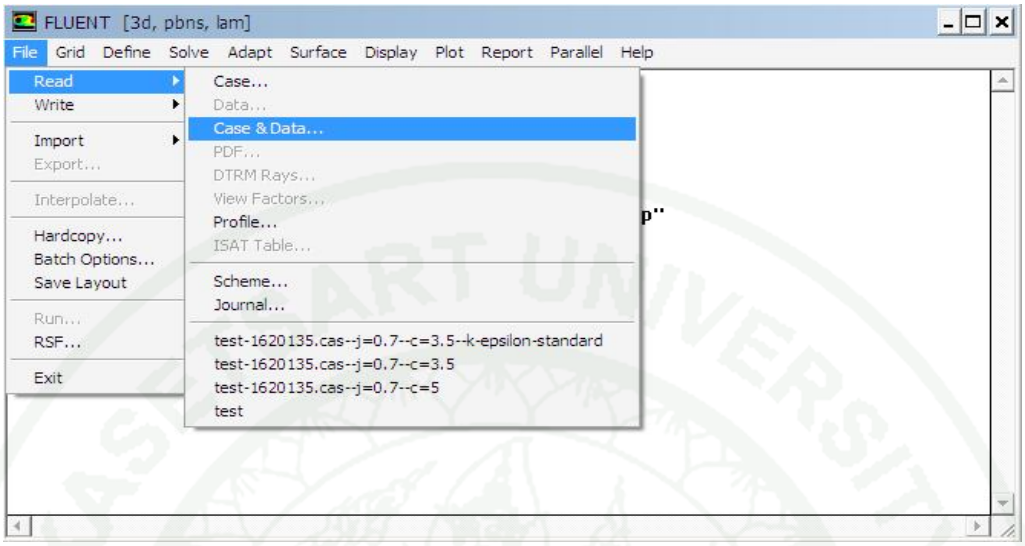
Appendix Figure D1 Procedure of using UDF step 1

2. FLUENT command windows



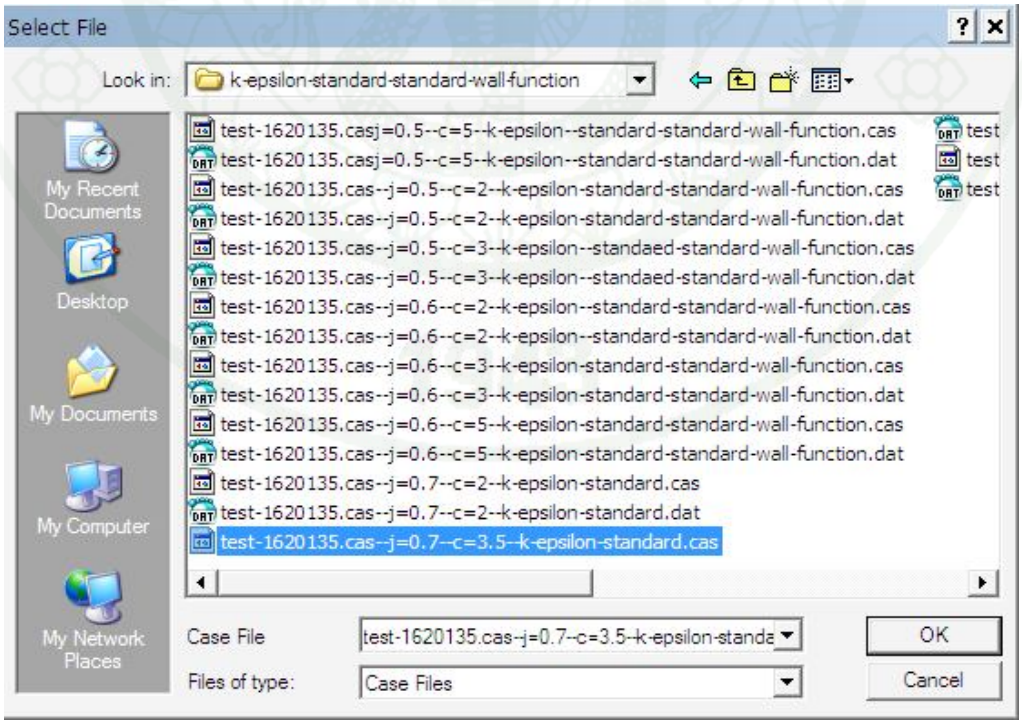
Appendix Figure D2 Procedure of using UDF step 2

3. Select file → Read → Case & Data



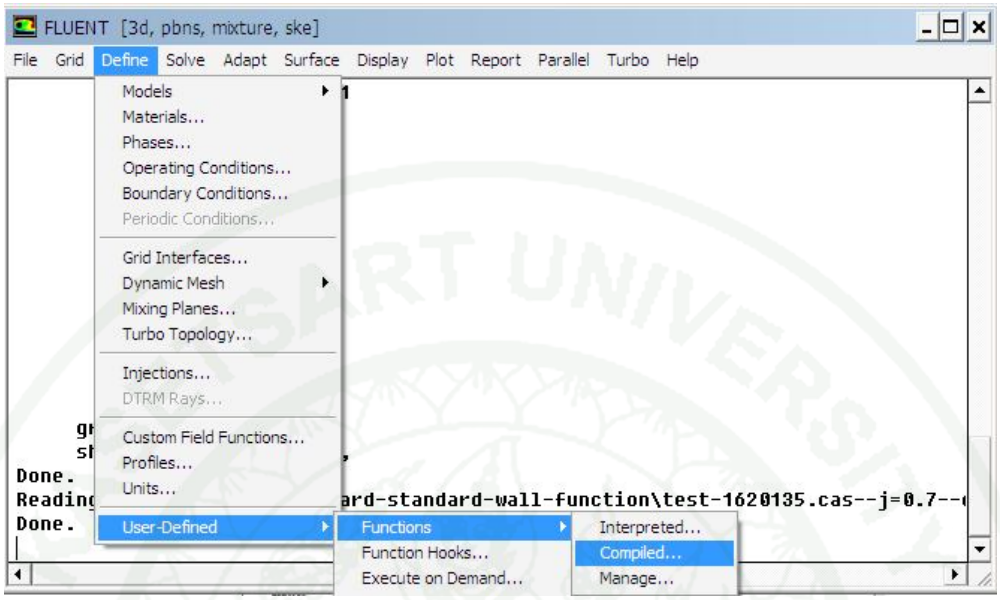
Appendix Figure D3 Procedure of using UDF step 3

4. Open file test-1620135.cas--j=0.7--c=3.5--k-epsilon-statndard.cas



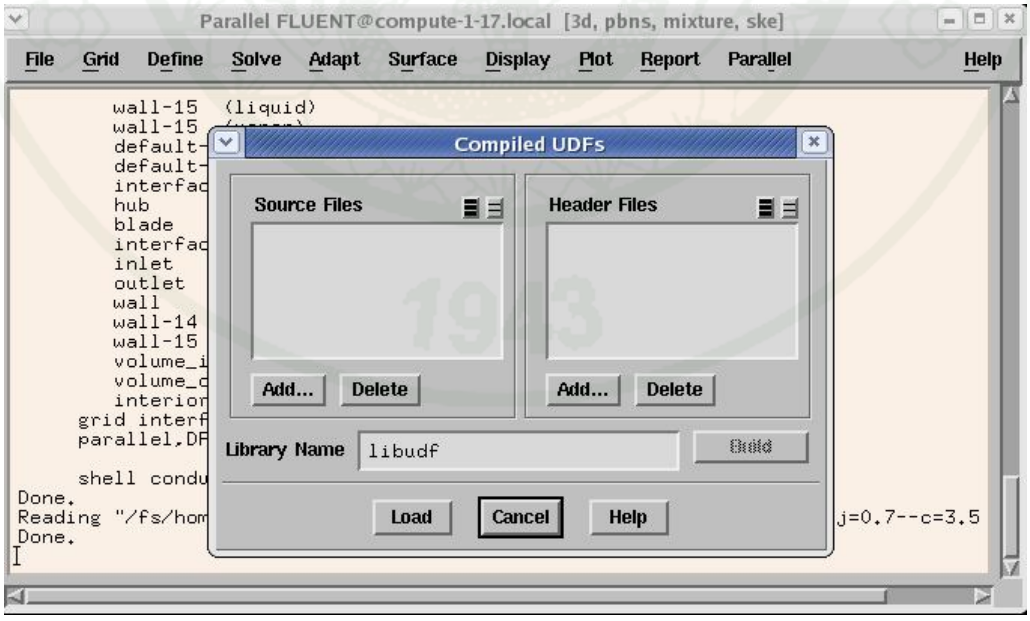
Appendix Figure D4 Procedure of using UDF step 4

5. Select define → User-defined → Functions → Compiled



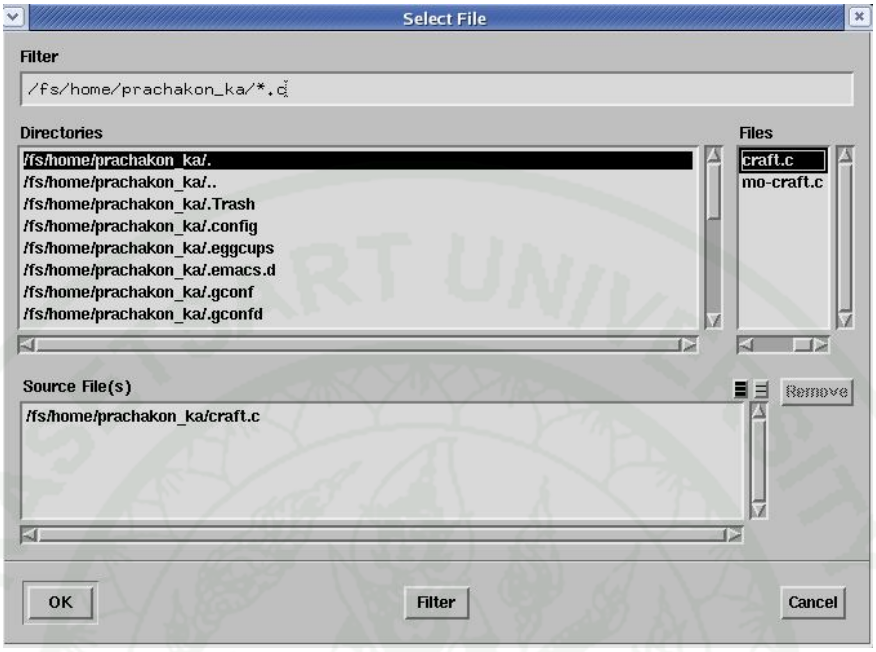
Appendix Figure D5 Procedure of using UDF step 5

6. Compiled UDF → Source files → Add



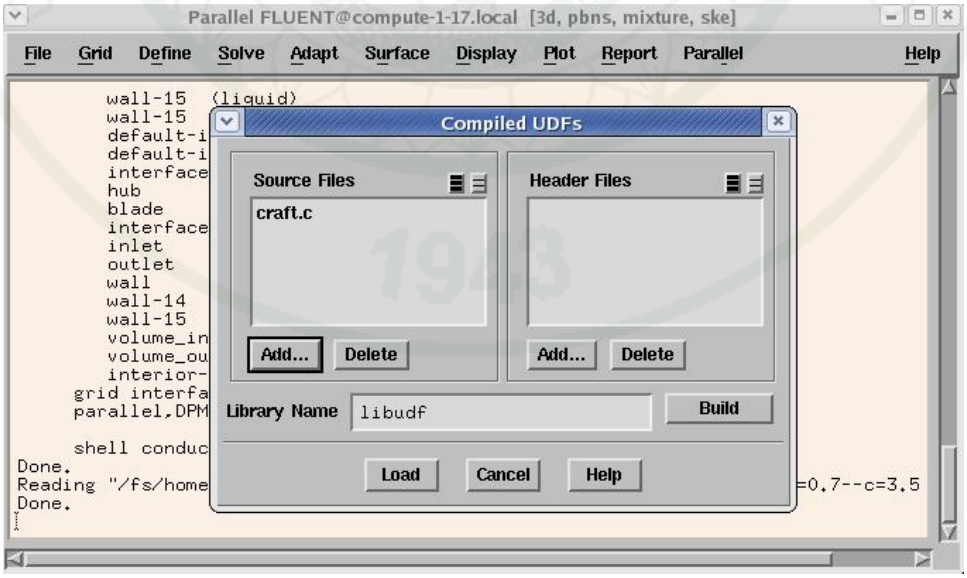
Appendix Figure D6 Procedure of using UDF step 6

7. Select source file → OK



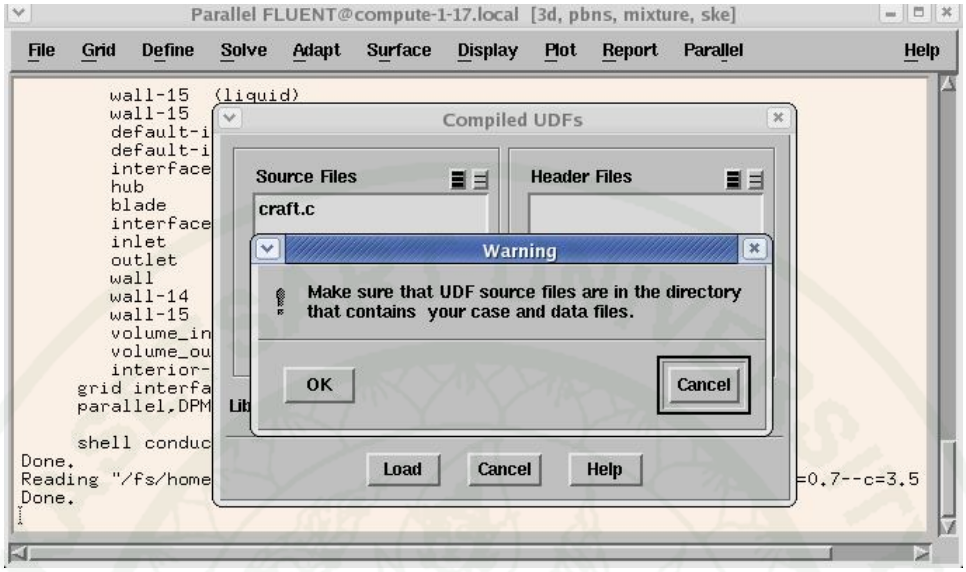
Appendix Figure D7 Procedure of using UDF step 7

8. Compiled UDF → Enter non_linear_craft_3d for the library name →Build



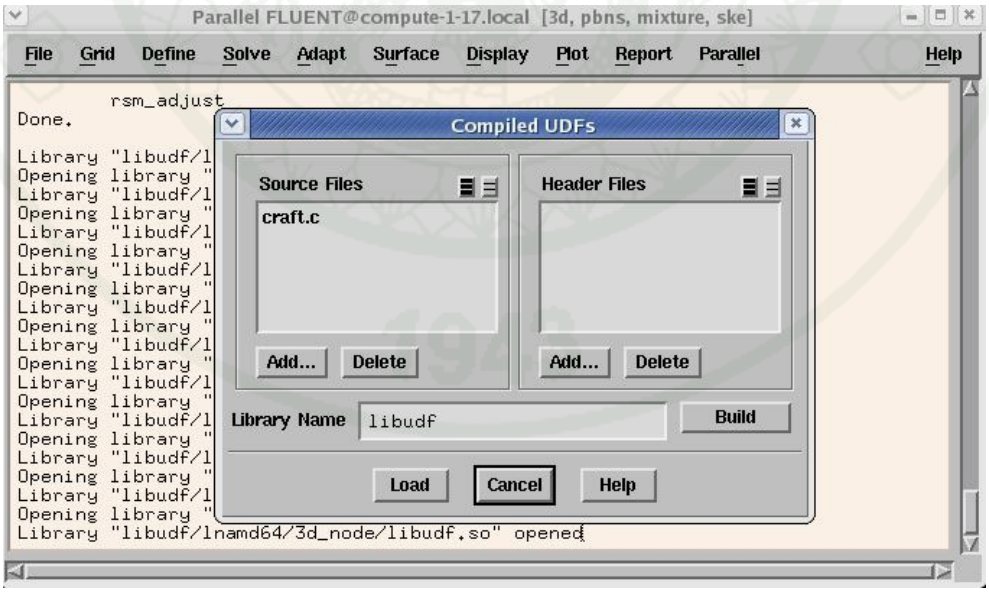
Appendix Figure D8 Procedure of using UDF step 8

9. Information → O.K



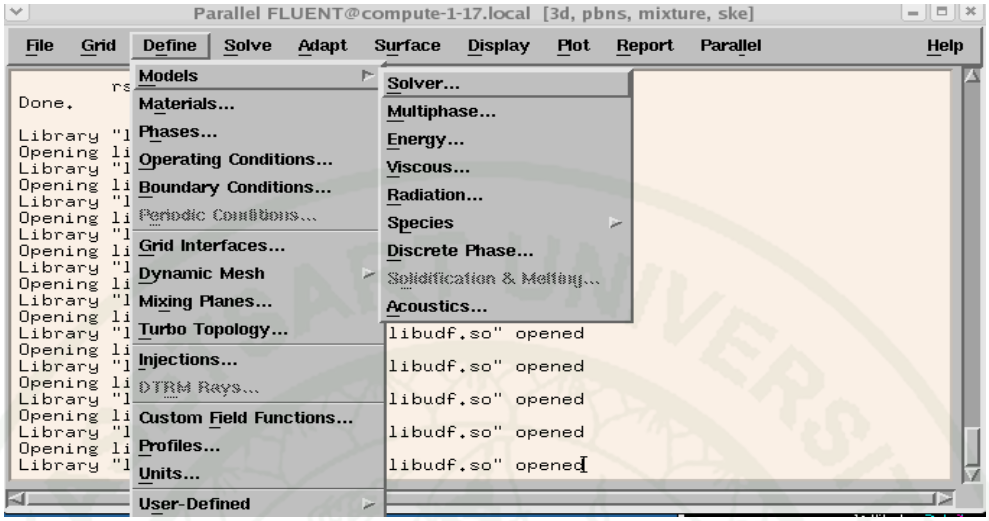
Appendix Figure D9 Procedure of using UDF step 9

10. Compiled UDF → Load



Appendix Figure D10 Procedure of using UDF step 10

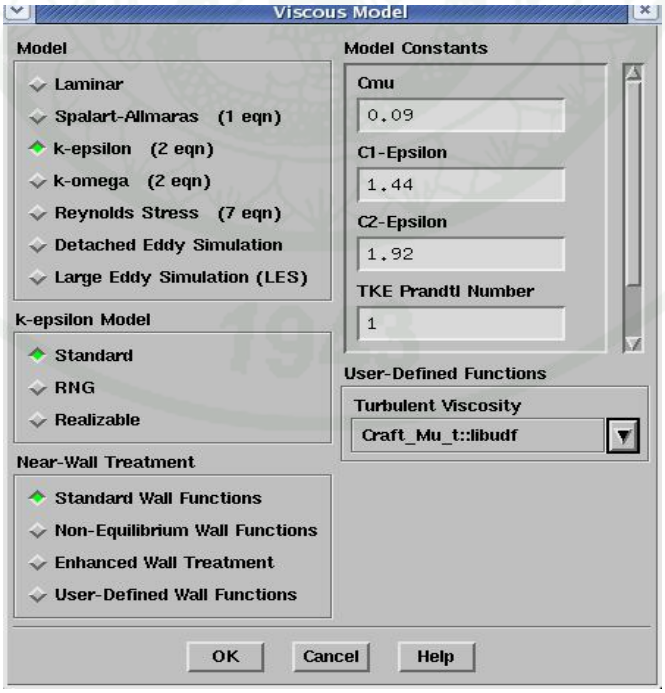
11. Select define → Models → Viscous



Appendix Figure D11 Procedure of using UDF step 11

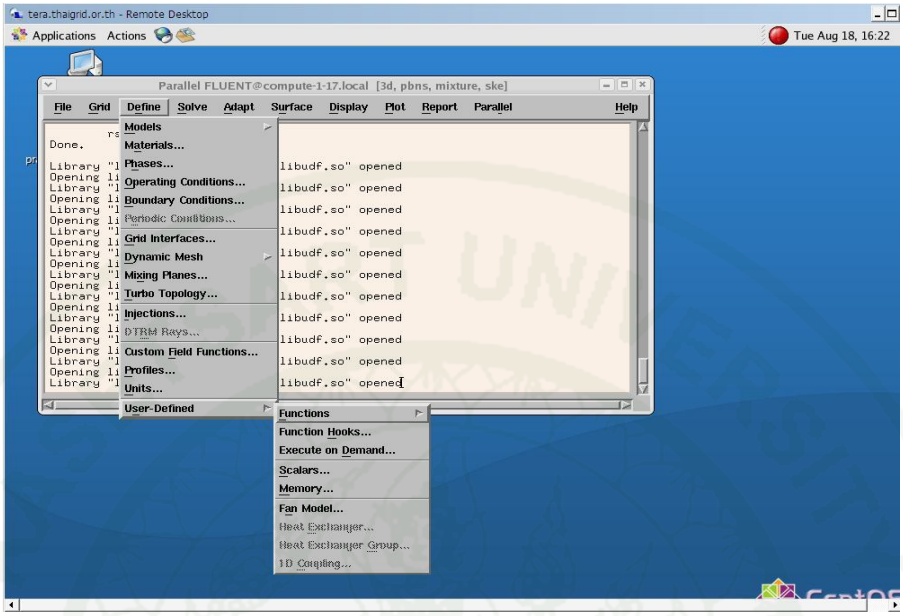
12. Viscous model → OK

(a) User-defined functions, turbulent viscosity, select Craft_Mu_t



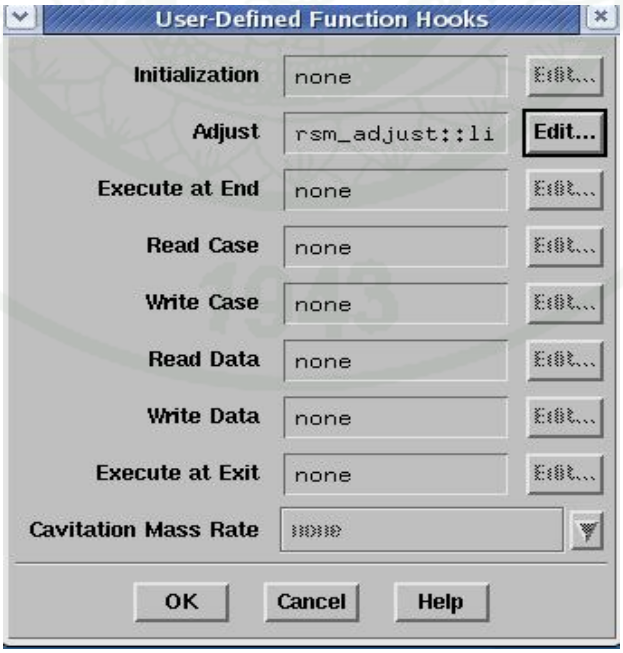
Appendix Figure D12 Procedure of using UDF step 12

13. Select define → User-defined → Function hooks



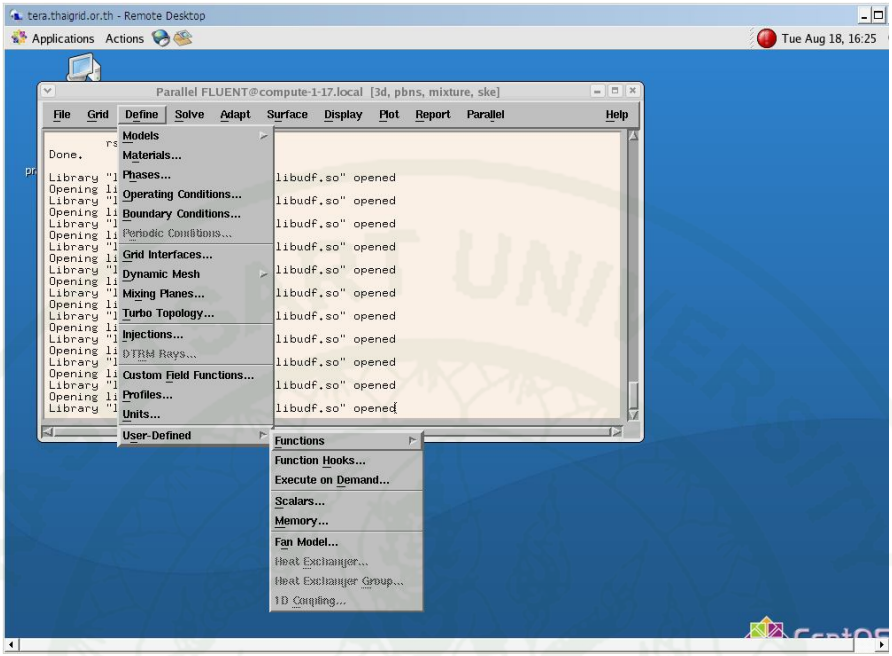
Appendix Figure D13 Procedure of using UDF step 13

14. User-defined function hooks → Adjust function, select rsm_adjust → OK



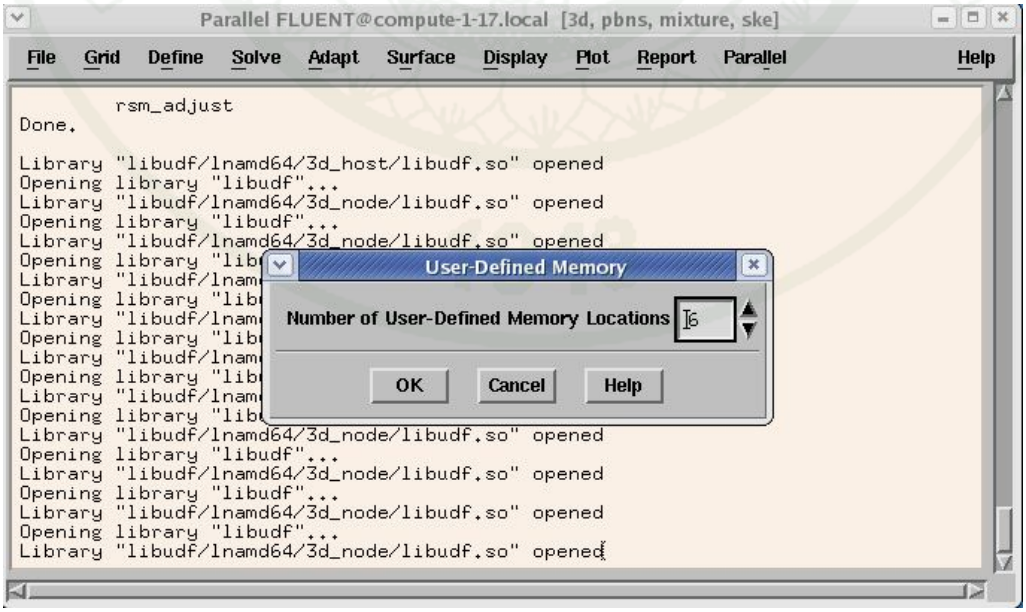
Appendix Figure D14 Procedure of using UDF step 14

15. Select define → User-defined → Memory



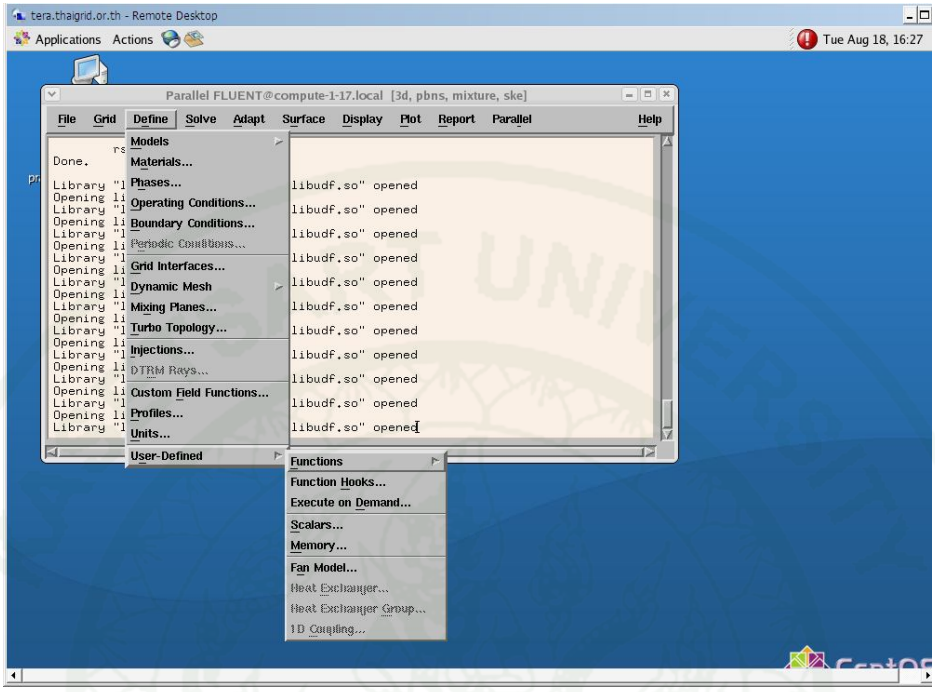
Appendix Figure D15 Procedure of using UDF step 15

16. User-defined memory → Number of user-defined memory locations, set 6 → OK



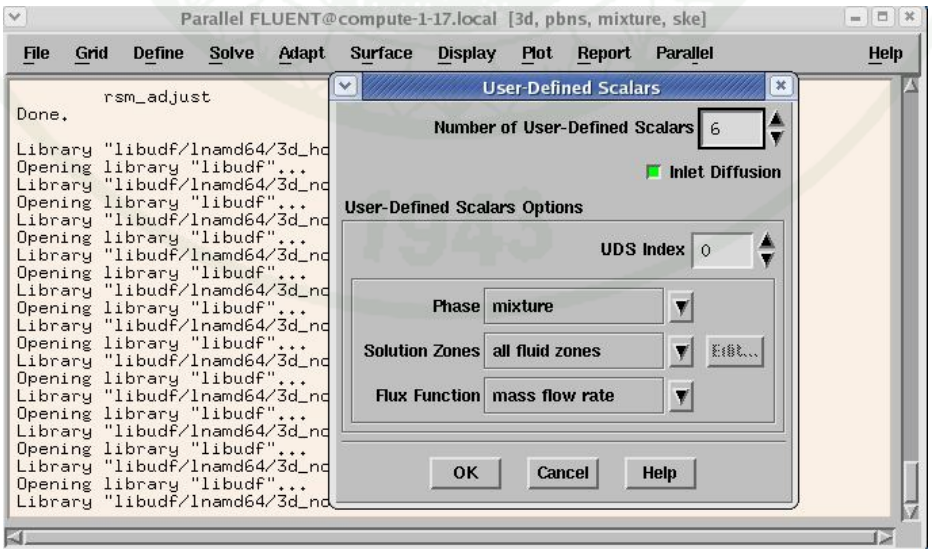
Appendix Figure D16 Procedure of using UDF step 16

17. Select define → User-defined → Scalars



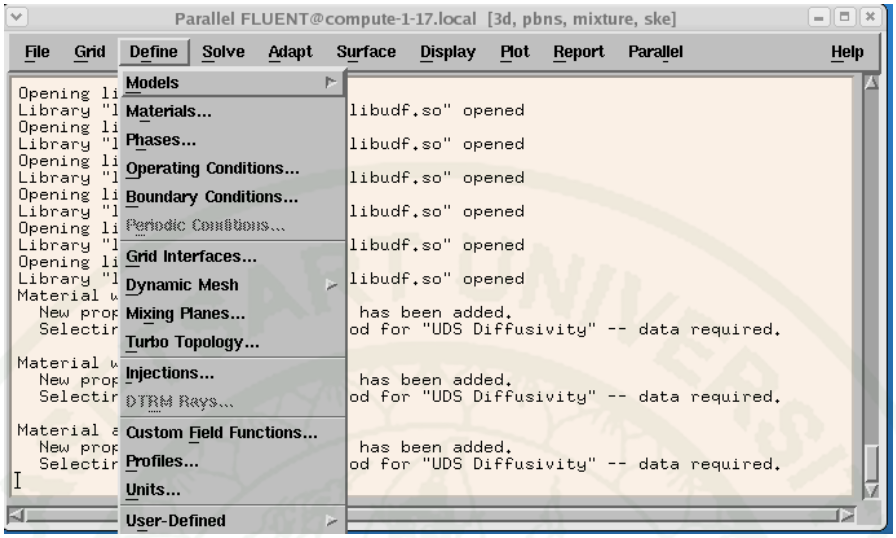
Appendix Figure D17 Procedure of using UDF step 17

18. User-defined scalars → Number of user-defined scalars, set 6 → OK



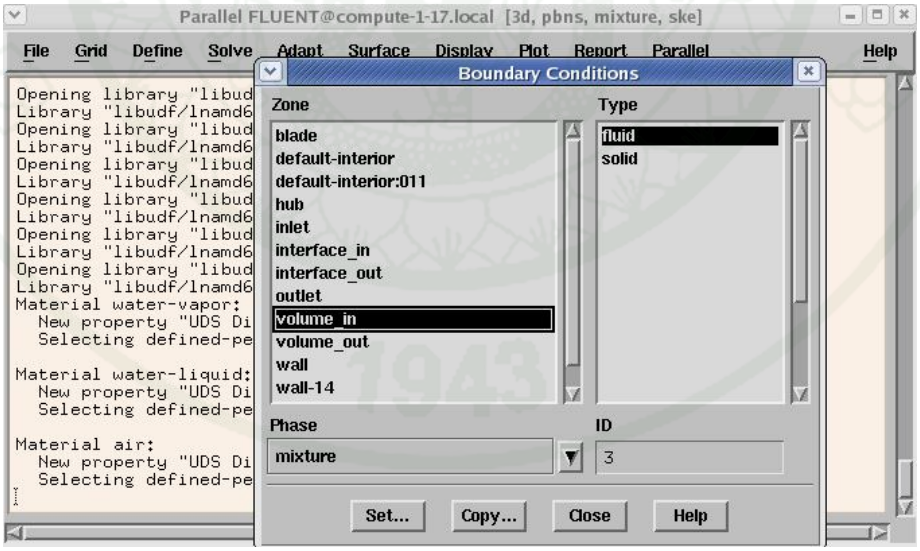
Appendix Figure D18 Procedure of using UDF step 18

19. Select define → Boundary conditions



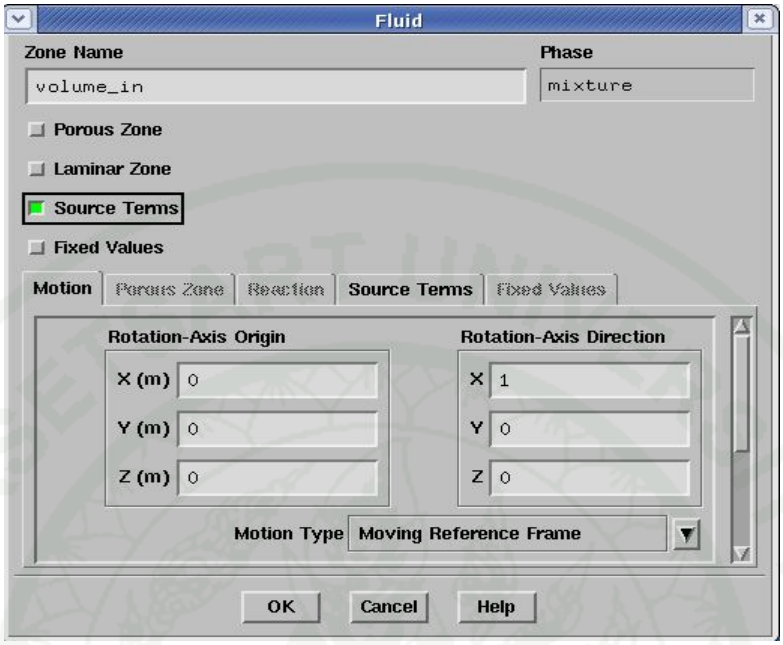
Appendix Figure D19 Procedure of using UDF step 19

20. Boundary conditions → Zone, select volume_in → Set



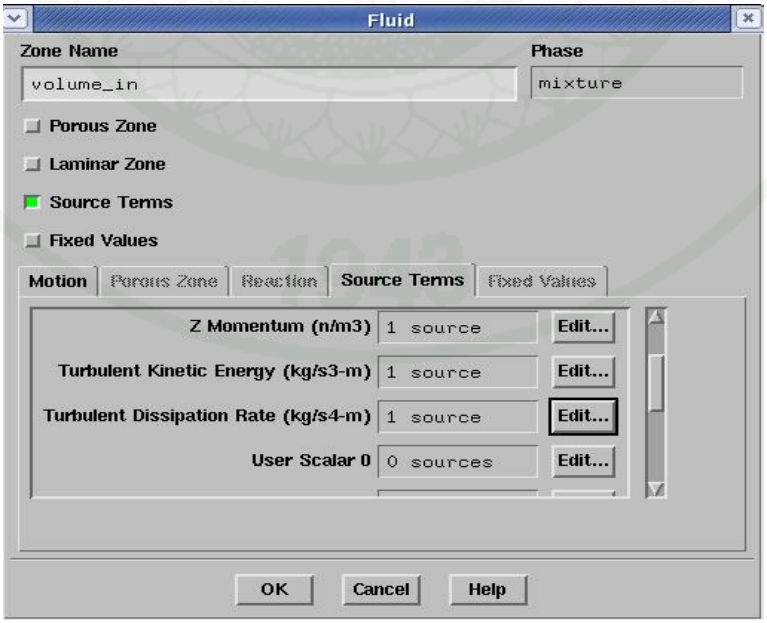
Appendix Figure D20 Procedure of using UDF step 20

21. Fluid → Select source terms



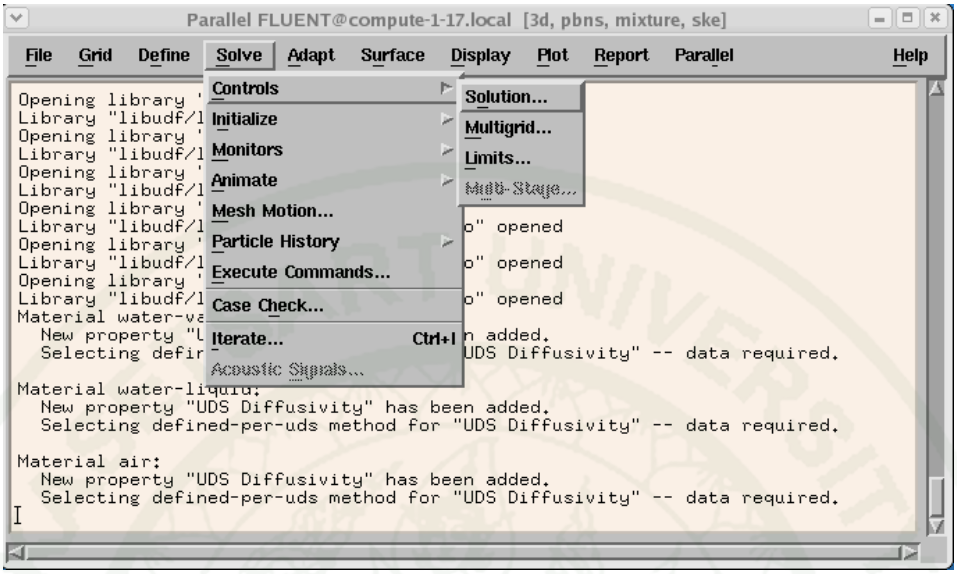
Appendix Figure D21 Procedure of using UDF step 21

22. Fluid → Source terms, input source term in transport equation → OK



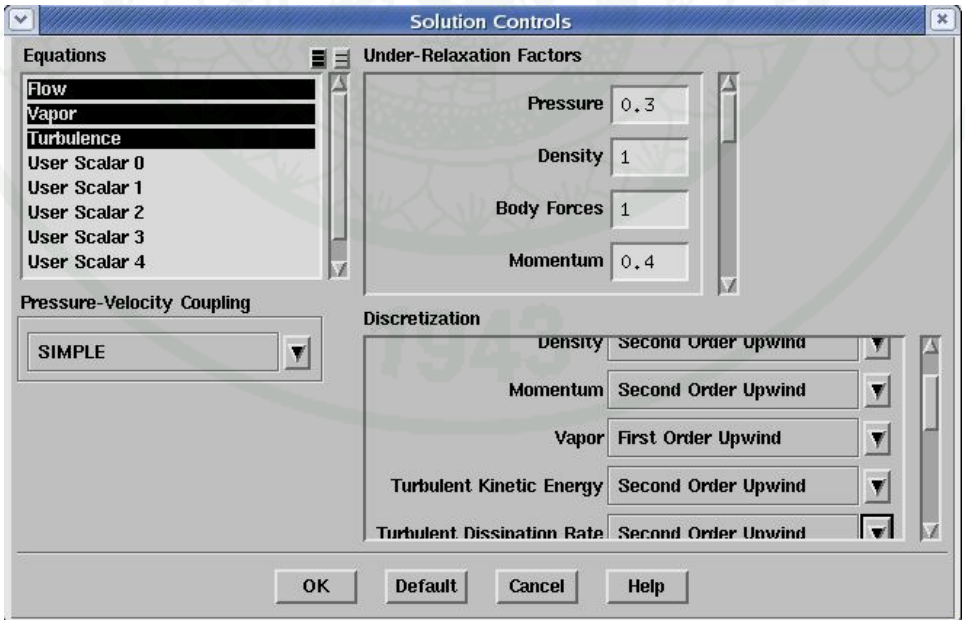
Appendix Figure D22 Procedure of using UDF step 22

23. Select solve → Controls → Solution



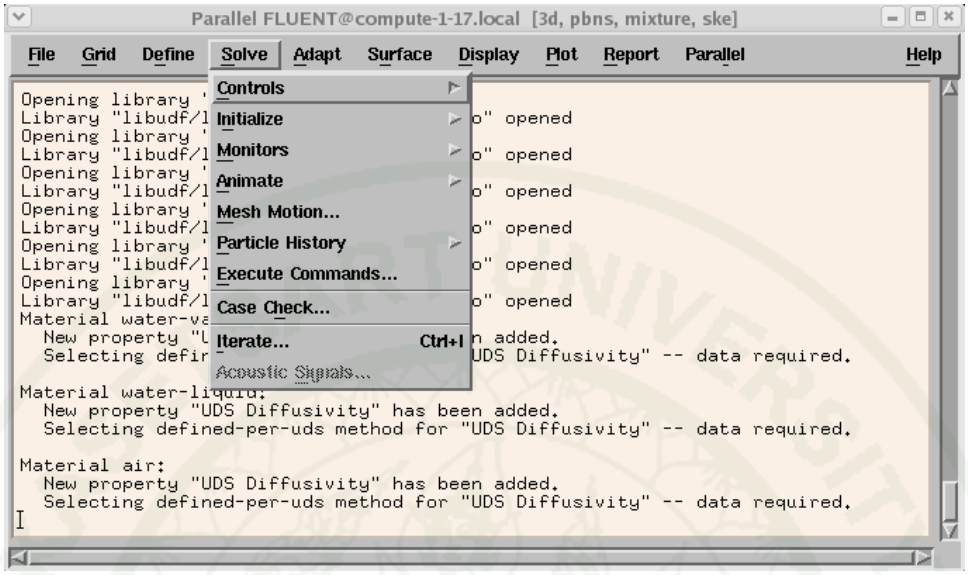
Appendix Figure D23 Procedure of using UDF step 23

24. Solution controls → Equations → Output user-defined scalar → OK



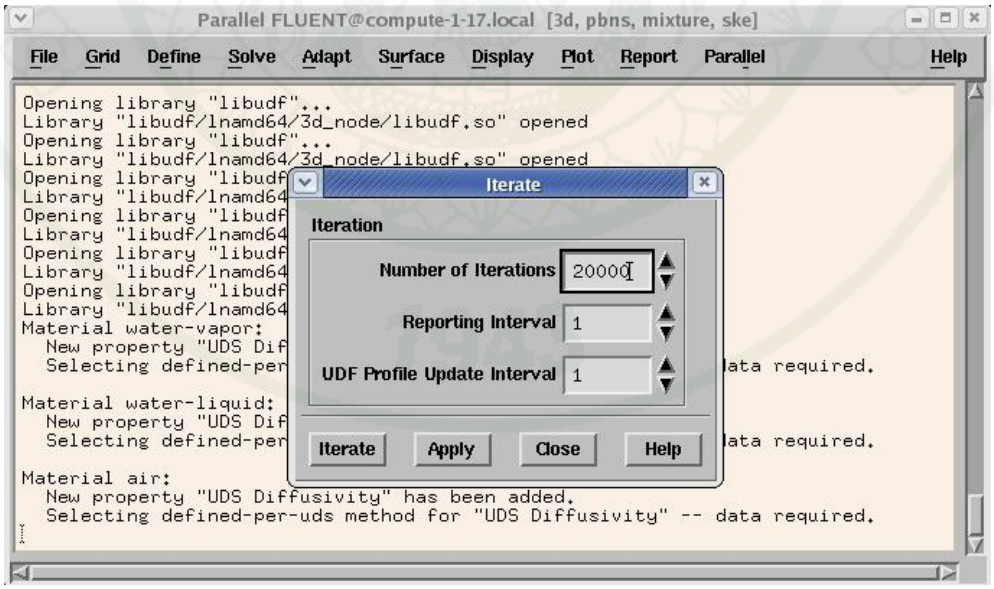
Appendix Figure D24 Procedure of using UDF step 24

25. Select solve → Iterate



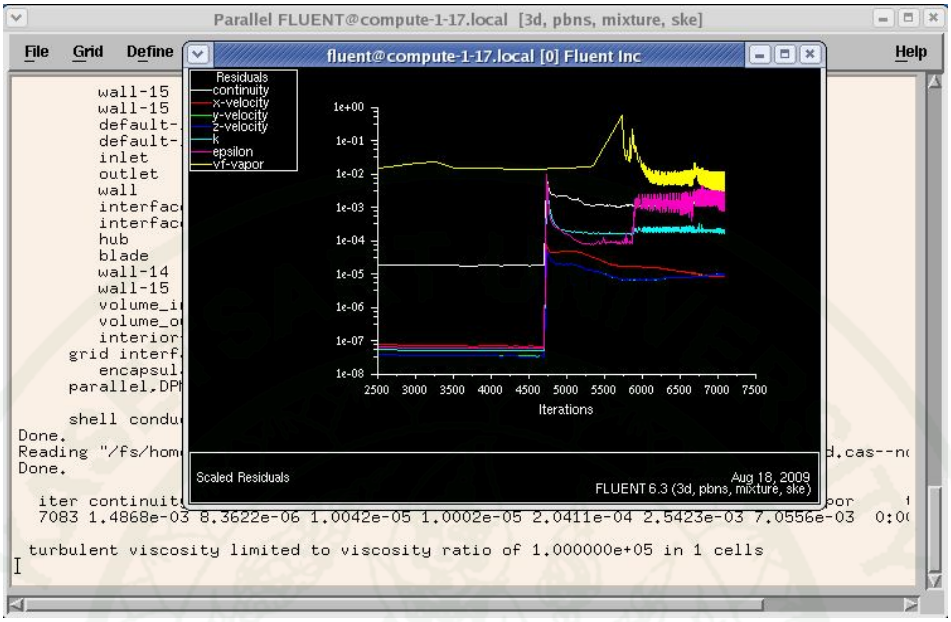
Appendix Figure D25 Procedure of using UDF step 25

26. Iteration → Iterate → Number of iterations, set 20,000.



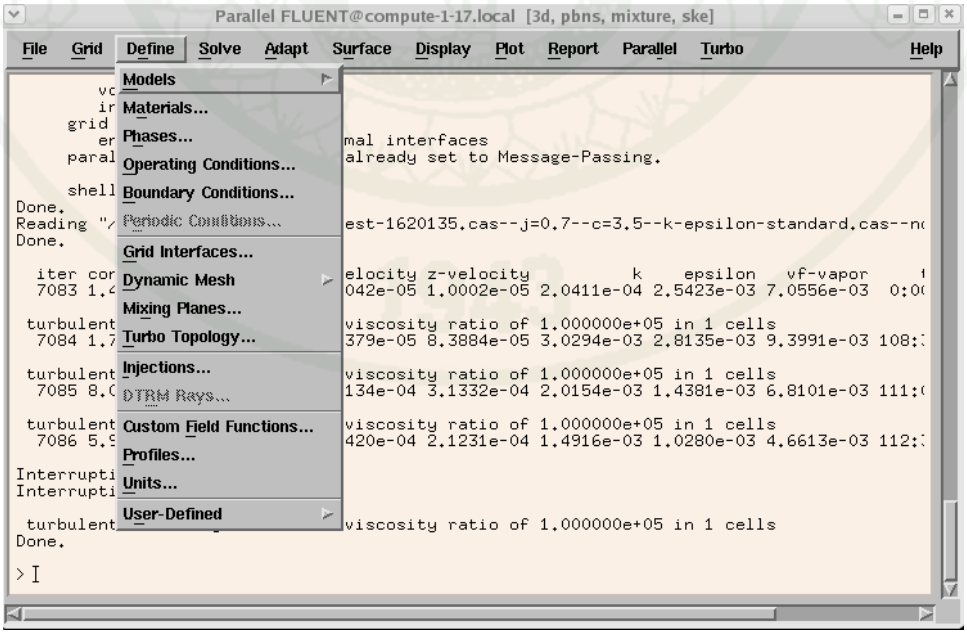
Appendix Figure D26 Procedure of using UDF step 26

27. When Solution is Converged will effective follow ficture



Appendix Figure D27 Procedure of using UDF step 27

28. Check Thrust and Torque by click Define-Turbo Topology



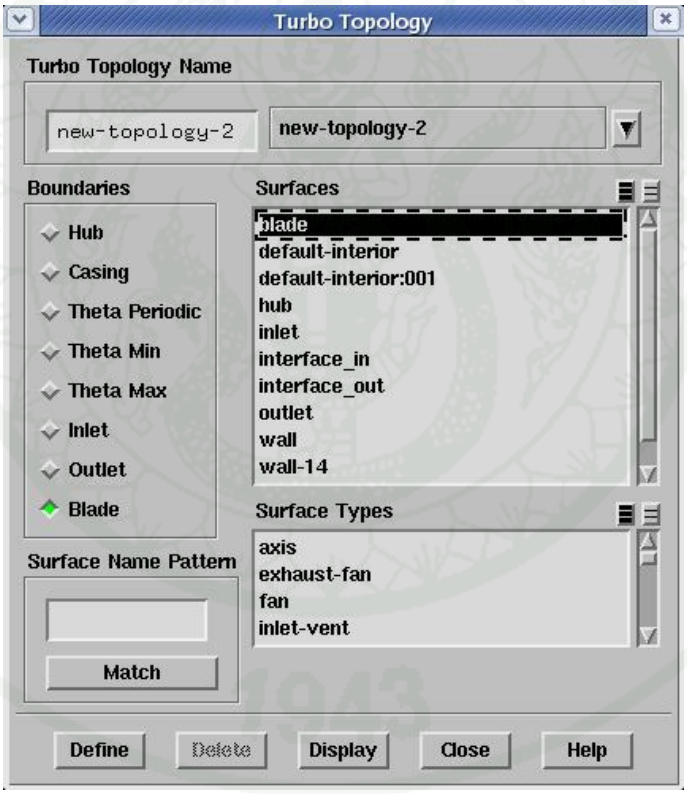
Appendix Figure D28 Procedure of using UDF step 28

29. Choose a pair

Boundaries	**Surfaces**
Hub	Hub
Casing	Interface_Out
Theda Periodic	default-interior:001
	(select Volume in)
Inlet	Inlet
Outlet	Outlet
Blade	Blade

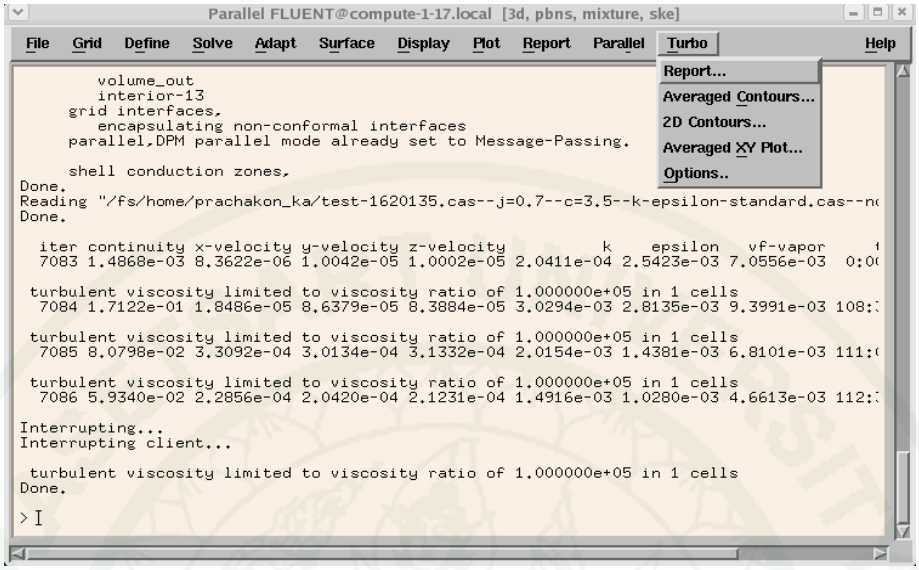
Click Define

.



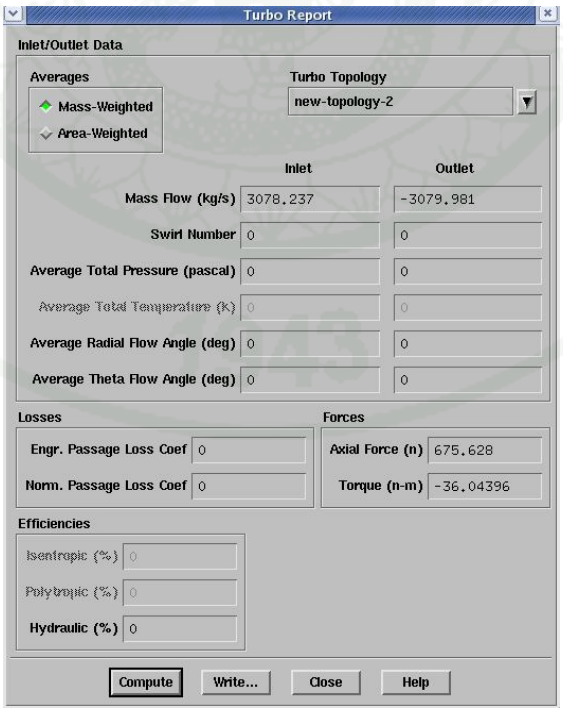
Appendix Figure D29 Procedure of using UDF step 29

30. Turbo-Report



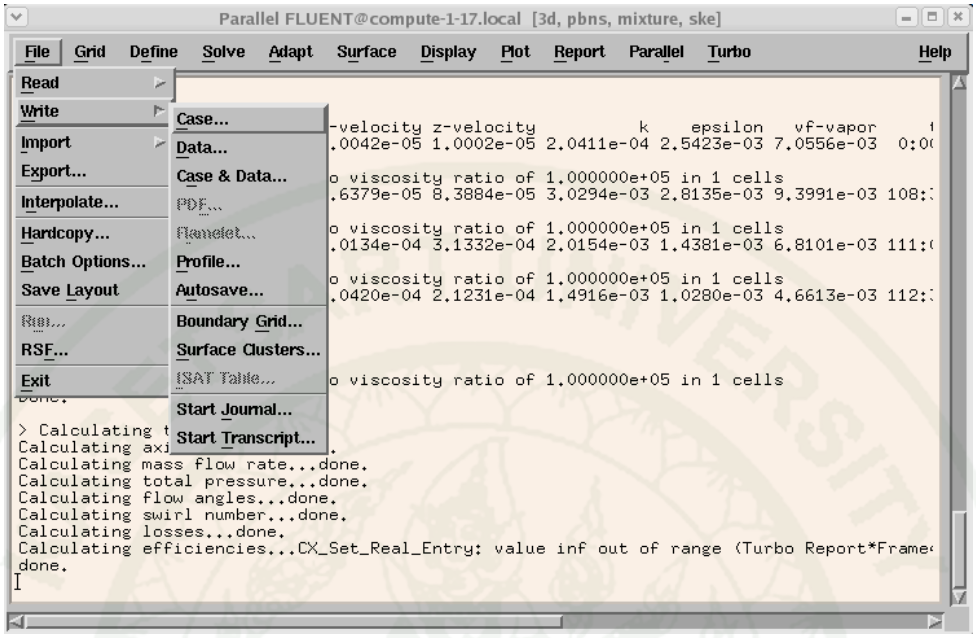
Appendix Figure D30 Procedure of using UDF step 30

31. Click Compute



Appendix Figure D31 Procedure of using UDF step 31

32. Select write → Case & Data



Appendix Figure D32 Procedure of using UDF step 32

Appendix E

UDF of Nonlinear Turbulence Model for Simulation of Cavitation on Marine Propeller

UDF of Nonlinear Turbulence Model
for Simulation of Cavitation on Marine Propeller

```
#include "udf.h"
#include "math.h"
/* Turbulence model constants */
const real C_1=-0.1;
const real C_2=0.1;
const real C_3=0.26;
const real C_5=0.0;
// User-defined scalars : Define Reynolds stress
enum
{
UU,
VV,
WW,
UV,
UW,
VW
};
// Define source in x momentun equation
DEFINE_SOURCE(u_source, c, t, dS, eqn)
{
real source;
dS[eqn]=0.0;
source= - C_R(c,t) * ( C_UDSI_G(c,t,UU)[0] +
C_UDSI_G(c,t,UV)[1]+C_UDSI_G(c,t,UW)[2] );
return source;
}
// Define source in y momentun equation
DEFINE_SOURCE(v_source, c, t, dS, eqn)
{
```

```
real source;
dS[eqn]= 0.0;
source = - C_R(c,t)*(C_UDSI_G(c,t,UV)[0]+ C_UDSI_G(c,t,VV)[1] +
C_UDSI_G(c,t,VW)[2] ) ;
return source;
}
// Define source in z momentun equation
DEFINE_SOURCE(w_source, c, t, dS, eqn)
{
real source;
dS[eqn]= 0.0;
source= -C_R(c,t)*(C_UDSI_G(c,t,UW)[0] + C_UDSI_G(c,t,VW)[1] +
C_UDSI_G(c,t,WW)[2] );
return source;
}
DEFINE_SOURCE(k_source, c, t, dS, eqn)
{
real prod;
dS[eqn]= 0.0;
prod=-C_R(c,t)*(C_UDSI(c,t,UU)*C_DUDX(c,t)+C_UDSI(c,t,UV)*C_DVDX(c,t)+
C_UDSI(c,t,UW)*C_DWDX(c,t)+C_UDSI(c,t,UV)*C_DUDY(c,t)+C_UDSI(c,t,VV)
*C_DVDY(c,t)+C_UDSI(c,t,VW)*C_DWDY(c,t)+C_UDSI(c,t,UW)*C_DUDZ(c,t)+
C_UDSI(c,t,VW)*C_DVDZ(c,t) + C_UDSI(c,t,WW)*C_DWDZ(c,t));
return prod;
}

DEFINE_SOURCE(e_source, c, t, dS, eqn)
{
real prod;
real c_1e =1.44;
real c_2e =1.92;
real k   = C_K(c,t);
```

```
real e   = C_D(c,t);
real u = C_MU_L(c,t);
real p   = C_R(c,t);
real v, Ret,fe2,cc;
        v = u/p;
        Ret=k*k/(e*v);
        dS[eqn]= 0.0;
        prod= - c_1e*C_D(c,t)/C_K(c,t)* C_R(c,t)
                                * ( C_UDSI(c,t,UU)*C_DUDX(c,t) +
C_UDSI(c,t,UV)*C_DVDX(c,t) + C_UDSI(c,t,UW)*C_DWDX(c,t)+
                                        C_UDSI(c,t,UV)*C_DUDY(c,t) +
C_UDSI(c,t,VV)*C_DVDY(c,t) + C_UDSI(c,t,VW)*C_DWDY(c,t)+
                                        C_UDSI(c,t,UW)*C_DUDZ(c,t) +
C_UDSI(c,t,VW)*C_DVDZ(c,t) + C_UDSI(c,t,WW)*C_DWDZ(c,t)
                );
        fe2=1.0- 0.3*(exp(-Ret*Ret));
        return prod- (c_2e*C_R(c,t)*e*e/k)+ (fe2*c_2e*C_R(c,t)*e*e/k );
}
// Define Eddy  viscosity
DEFINE_TURBULENT_VISCOSITY(Craft_Mu_t,c,t)
{
real S,W,c_mu,T,mu_t;
real S11, S12, S13, S21, S22, S23, S31, S32, S33;
real W11, W12, W13, W21, W22, W23, W31, W32, W33;
// Define variable
real u = C_MU_L(c,t); //laminar viscosity
real p   = C_R(c,t);     //density
real v, Ret,fmu;
real k   = C_K(c,t);     //Turbulent kinetic energy(k)
real e   = C_D(c,t);     //Dissipation rate (epsilon)
        T=C_K(c,t)/C_D(c,t); // k/e
// Define stress tensor
```

```
                S11 = 0.5*( C_DUDX(c,t)+C_DUDX(c,t) );
                S12 = 0.5*( C_DUDY(c,t)+C_DVDX(c,t) );
                S13 = 0.5*( C_DUDZ(c,t)+C_DWDX(c,t) );
                S21 = 0.5*( C_DVDX(c,t)+C_DUDY(c,t) );
                S22 = 0.5*( C_DVDY(c,t)+C_DVDY(c,t) );
                S23 = 0.5*( C_DVDZ(c,t)+C_DWDY(c,t) );
                S31 = 0.5*( C_DWDX(c,t)+C_DUDZ(c,t) );
                S32 = 0.5*( C_DWDY(c,t)+C_DVDZ(c,t) );
                S33 = 0.5*( C_DWDZ(c,t)+C_DWDZ(c,t) );
S=T*sqrt(
2*(S11*S11+S12*S12+S13*S13+S21*S21+S22*S22+S23*S23+S31*S31+S32*S32
+S33*S33) );
// Define vorticity tensor
                W11 = 0.5*( C_DUDX(c,t)-C_DUDX(c,t) );
                W12 = 0.5*( C_DUDY(c,t)-C_DVDX(c,t) );
                W13 = 0.5*( C_DUDZ(c,t)-C_DWDX(c,t) );
                W21 = 0.5*( C_DVDX(c,t)-C_DUDY(c,t) );
                W22 = 0.5*( C_DVDY(c,t)-C_DVDY(c,t) );
                W23 = 0.5*( C_DVDZ(c,t)-C_DWDY(c,t) );
                W31 = 0.5*( C_DWDX(c,t)-C_DUDZ(c,t) );
                W32 = 0.5*( C_DWDY(c,t)-C_DVDZ(c,t) );
                W33 = 0.5*( C_DWDZ(c,t)-C_DWDZ(c,t) );
W=T*sqrt(
2*(W11*W11+W12*W12+W13*W13+W21*W21+W22*W22+W23*W23+W31*W
31+W32*W32+W33*W33) );
// Define kinenatic viscosity
v = u/p;
// Define turbulent Reynolds number
Ret=k*k/(e*v);
// Define Constant Cmu
c_mu=(0.3/(1.0+(0.35*  (MAX(S,W))*1.5)))*(1.0-exp(-0.36/(exp(-0.75*  MAX(S,W)
))));
```

```
// Define Damping function of Craft et al.(1996)
fmu=(1.0-exp(-pow((Ret/90.),0.5)-pow((Ret/400.),2) ));
// Define Eddy viscosity
mu_t=c_mu*T*C_K(c,t);
return mu_t;
}
DEFINE_ADJUST(rsm_adjust,domain)
{
Thread  *t;
cell_t  c;
real X;
real mu_t, c_mu;
real S11, S12, S13, S21, S22, S23, S31, S32, S33;
real W11, W12, W13, W21, W22, W23, W31, W32, W33;
real S, W, Sij, Wij;
real C_4, C_6, C_7;
real P11, P22, P33, Pkk;
real tau_w, u_tauw, u_star, y_star, Gk, u_mag;
real u ;
real p ;
real v, Ret,fmu;
real k ;
real e ;
/* Set the turbulent viscosity */
thread_loop_c(t,domain)
if (FLUID_THREAD_P(t))
{
begin_c_loop(c,t)
{
// Define variable
u= C_MU_L(c,t); //laminar viscosity
p= C_R(c,t);     //density
```

```
k= C_K(c,t);     //Turbulent kinetic energy(k)
e= C_D(c,t);     //Dissipation rate (epsilon)
X=C_K(c,t)/C_D(c,t); // k/e
// Define stress tensor
                    S11 = 0.5*( C_DUDX(c,t)+C_DUDX(c,t) );
                    S12 = 0.5*( C_DUDY(c,t)+C_DVDX(c,t) );
                    S13 = 0.5*( C_DUDZ(c,t)+C_DWDX(c,t) );
                    S21 = 0.5*( C_DVDX(c,t)+C_DUDY(c,t) );
                    S22 = 0.5*( C_DVDY(c,t)+C_DVDY(c,t) );
                    S23 = 0.5*( C_DVDZ(c,t)+C_DWDY(c,t) );
                    S31 = 0.5*( C_DWDX(c,t)+C_DUDZ(c,t) );
                    S32 = 0.5*( C_DWDY(c,t)+C_DVDZ(c,t) );
                    S33 = 0.5*( C_DWDZ(c,t)+C_DWDZ(c,t) );
S=X*sqrt(
2*(S11*S11+S12*S12+S13*S13+S21*S21+S22*S22+S23*S23+S31*S31+S32*S32
+S33*S33) );
// Define vorticity tensor
                    W11 = 0.5*( C_DUDX(c,t)-C_DUDX(c,t) );
                    W12 = 0.5*( C_DUDY(c,t)-C_DVDX(c,t) );
                    W13 = 0.5*( C_DUDZ(c,t)-C_DWDX(c,t) );
                    W21 = 0.5*( C_DVDX(c,t)-C_DUDY(c,t) );
                    W22 = 0.5*( C_DVDY(c,t)-C_DVDY(c,t) );
                    W23 = 0.5*( C_DVDZ(c,t)-C_DWDY(c,t) );
                    W31 = 0.5*( C_DWDX(c,t)-C_DUDZ(c,t) );
                    W32 = 0.5*( C_DWDY(c,t)-C_DVDZ(c,t) );
                    W33 = 0.5*( C_DWDZ(c,t)-C_DWDZ(c,t) );
W=X*sqrt(
2*(W11*W11+W12*W12+W13*W13+W21*W21+W22*W22+W23*W23+W31*W
31+W32*W32+W33*W33) );
// Define kinenatic viscosity
v = u/p;
// Define turbulent Reynolds number
```

```
Ret=k*k/(e*v);
// Define Constant Cmu
c_mu=(0.3/(1.0+(0.35*  (MAX(S,W))*1.5)))*(1.0-exp(-0.36/(exp(-0.75*  MAX(S,W)
))));
// Define Damping function of Craft et al.(1996)
fmu=(1.0-exp(-pow((Ret/90.),0.5)-pow((Ret/400.),2) ));
// Define Eddy viscosity
mu_t=c_mu*fmu*C_K(c,t)*C_K(c,t)*C_D(c,t);
// Define constant of Craft et al.(1996)
C_4=-10.0*SQR(c_mu);
C_6=-5.0*SQR(c_mu);
C_7=5.0*SQR(c_mu);
// Define tensor index = SklSkl
Sij=S11*S11+S12*S12+S13*S13+S21*S21+S22*S22+S23*S23+S31*S31+S32*S32
+S33*S33;
// Define tensor index = WklWkl
Wij=W11*W11+W12*W12+W13*W13+W21*W21+W22*W22+W23*W23+W31*
W31+W32*W32+W33*W33;
//**********************************************************************
**//
//************** IMPORTANT define only anisotropy tensor
*******************//
//************** and multiply k in the Reynolds stress
************************//
//**************  anisotropy
tensor*************************************//
//**********************************************************************
*//
//**********************************************************************
*//
// Define anisotropy tensor of normal Reynolds stress
C_UDSI(c,t,UU)=
```

```
                    C_1*mu_t*X*   (S11*S11+S12*S12+S13*S13-1./3.*Sij)
                   + 2.*C_2*mu_t*X*   (W12*S12+W13*S13)
                   + C_3*mu_t*X*   (W12*W12+W13*W13-1./3.*Wij)
                   -2.*C_4*mu_t*X*X*
(W12*(S11*S12+S12*S22+S13*S23)+W13*(S11*S13+S12*S23+S13*S33))
                   +   C_6*mu_t*X*X* (S11*Sij)
                   +   C_7*mu_t*X*X* (S11*Wij);

C_UDSI(c,t,VV)=
                       C_1*mu_t*X*   (S12*S12+S22*S22+S23*S23-1./3.*Sij)
                   + 2.*C_2*mu_t*X*   (W23*S23-W12*S12)
                    +   C_3*mu_t*X*   (W12*W12+W23*W23-1./3.*Wij)
                   + 2.*C_4*mu_t*X*X* (W12*(S11*S12+S12*S22+S13*S23)-
W23*(S12*S13+S22*S23+S23*S33))
                    +   C_6*mu_t*X*X* (S22*Sij)
                    +   C_7*mu_t*X*X* (S22*Wij);

C_UDSI(c,t,WW)=
                      C_1*mu_t*X*   (S13*S13+S23*S23+S33*S33-1./3.*Sij)
                   - 2.*C_2*mu_t*X*   (W13*S13+W23*S23)
                   + C_3*mu_t*X*   (W13*W13+W23*W23-1./3.*Wij)
                   +2.*C_4*mu_t*X*X*
(W13*(S11*S13+S12*S23+S13*S33)+W23*(S12*S13+S22*S23+S23*S33))
                   +   C_6*mu_t*X*X* (S33*Sij)
                   +   C_7*mu_t*X*X* (S33*Wij);
// Define anisotropy tensor of shear Reynolds stress
        C_UDSI(c,t,UV)=
                      C_1*mu_t*X*   (S11*S12+S12*S22+S13*S23)
                   +   C_2*mu_t*X*   (W12*(S22-S11)+W13*S23+W23*S13)
                   +   C_3*mu_t*X*   (W13*W23)
```

```
                    + C_4*mu_t*X*X* (W12*(S11*S11+S13*S13-S22*S22-
S23*S23)-W13*(S12*S13+S22*S23+S23*S33)-
W23*(S11*S13+S12*S23+S13*S33))
                    +   C_6*mu_t*X*X* (S12*Sij)
                    +   C_7*mu_t*X*X* (S12*Wij);

      C_UDSI(c,t,UW)=
                        C_1*mu_t*X*   (S11*S13+S12*S23+S13*S33)
                    +   C_2*mu_t*X*   (W13*(S33-S11)+W12*S23-W23*S12)
                    -   C_3*mu_t*X*   (W12*W23)
                    +   C_4*mu_t*X*X* (W13*(S11*S11+S12*S12-S23*S23-
S33*S33)-
W12*(S11*S13+S22*S23+S23*S33)+W23*(S11*S12+S12*S22+S13*S23))
                    +   C_6*mu_t*X*X* (S13*Sij)
                    +   C_7*mu_t*X*X* (S13*Wij);

      C_UDSI(c,t,VW)=
                        C_1*mu_t*X*   (S12*S13+S22*S23+S23*S33)
                    +   C_2*mu_t*X*   (W23*(S33-S22)-W12*S13-W13*S12)
                    +   C_3*mu_t*X*   (W12*W13)
                    +   C_4*mu_t*X*X* (W23*(S12*S12+S22*S22-S13*S13-
S33*S33)+W12*(S11*S13+S12*S23+S13*S33)+W13*(S11*S12+S12*S22+S13*S2
3))
                    +   C_6*mu_t*X*X* (S23*Sij)
                    +   C_7*mu_t*X*X* (S23*Wij);
// Define memory Reynolds stress by User defined Memory
//*****************************************************************
***//
//**************IMPORTANT define Linear and Nonlinear
Term*****************//
//*****************************************************************
**//
```

```
C_UDMI(c,t,0)=        2./3.*C_K(c,t) -2.*mu_t*S11  + C_UDSI(c,t,UU);
C_UDMI(c,t,1)=        2./3.*C_K(c,t) -2.*mu_t*S22  + C_UDSI(c,t,VV);
C_UDMI(c,t,2)=        2./3.*C_K(c,t) -2.*mu_t*S33  + C_UDSI(c,t,WW);
C_UDMI(c,t,3)=       -2.*mu_t*S12  + C_UDSI(c,t,UV);
C_UDMI(c,t,4)=       -2.*mu_t*S13  + C_UDSI(c,t,UW);
C_UDMI(c,t,5)=       -2.*mu_t*S23  + C_UDSI(c,t,VW);
}
end_c_loop(c,t)
}
}
```

User-defined function of modified nonlinear turbulence model for Simulation of Cavitation on Marine propeller

```
#include "udf.h"
#include "math.h"
/* Turbulence model constants */
const real C_1=-0.1;
const real C_2=0.19;
const real C_3=0.21;
const real C_5=0.0;
// User-defined scalars : Define Reynolds stress
enum
{
UU,
VV,
WW,
UV,
UW,
VW
};
```

```
// Define source in x momentun equation
DEFINE_SOURCE(u_source, c, t, dS, eqn)
{
real source;
dS[eqn]=0.0;
source= - C_R(c,t) * ( C_UDSI_G(c,t,UU)[0] +
C_UDSI_G(c,t,UV)[1]+C_UDSI_G(c,t,UW)[2] );
return source;
}
// Define source in y momentun equation
DEFINE_SOURCE(v_source, c, t, dS, eqn)
{
real source;
dS[eqn]= 0.0;
source = - C_R(c,t)*(C_UDSI_G(c,t,UV)[0]+ C_UDSI_G(c,t,VV)[1] +
C_UDSI_G(c,t,VW)[2] ) ;
return source;
}
// Define source in z momentun equation
DEFINE_SOURCE(w_source, c, t, dS, eqn)
{
real source;
dS[eqn]= 0.0;
source= -C_R(c,t)*(C_UDSI_G(c,t,UW)[0] + C_UDSI_G(c,t,VW)[1] +
C_UDSI_G(c,t,WW)[2] );
return source;
}
DEFINE_SOURCE(k_source, c, t, dS, eqn)
{
real prod;
dS[eqn]= 0.0;
```

```
prod=-C_R(c,t)*(C_UDSI(c,t,UU)*C_DUDX(c,t)+C_UDSI(c,t,UV)*C_DVDX(c,t)+
C_UDSI(c,t,UW)*C_DWDX(c,t)+C_UDSI(c,t,UV)*C_DUDY(c,t)+C_UDSI(c,t,VV)
*C_DVDY(c,t)+C_UDSI(c,t,VW)*C_DWDY(c,t)+C_UDSI(c,t,UW)*C_DUDZ(c,t)+
C_UDSI(c,t,VW)*C_DVDZ(c,t) + C_UDSI(c,t,WW)*C_DWDZ(c,t));
return prod;
}

DEFINE_SOURCE(e_source, c, t, dS, eqn)
{
real prod;
real c_1e =1.44;
real c_2e =1.92;
real k   = C_K(c,t);
real e   = C_D(c,t);
real u = C_MU_L(c,t);
real p   = C_R(c,t);
real v, Ret,fe2,cc;
        v = u/p;
        Ret=k*k/(e*v);
        dS[eqn]= 0.0;
        prod= - c_1e*C_D(c,t)/C_K(c,t)* C_R(c,t)
                                        * ( C_UDSI(c,t,UU)*C_DUDX(c,t) +
C_UDSI(c,t,UV)*C_DVDX(c,t) + C_UDSI(c,t,UW)*C_DWDX(c,t)+
                                                C_UDSI(c,t,UV)*C_DUDY(c,t) +
C_UDSI(c,t,VV)*C_DVDY(c,t) + C_UDSI(c,t,VW)*C_DWDY(c,t)+
                                                C_UDSI(c,t,UW)*C_DUDZ(c,t) +
C_UDSI(c,t,VW)*C_DVDZ(c,t) + C_UDSI(c,t,WW)*C_DWDZ(c,t)
                    );
        fe2=1.0- 0.3*(exp(-Ret*Ret));
        return prod- (c_2e*C_R(c,t)*e*e/k)+ (fe2*c_2e*C_R(c,t)*e*e/k );
}
// Define Eddy  viscosity
```

```
DEFINE_TURBULENT_VISCOSITY(Craft_Mu_t,c,t)
{
real S,W,c_mu,T,mu_t;
real S11, S12, S13, S21, S22, S23, S31, S32, S33;
real W11, W12, W13, W21, W22, W23, W31, W32, W33;
// Define variable
real u = C_MU_L(c,t); //laminar viscosity
real p   = C_R(c,t);     //density
real v, Ret,fmu;
real k   = C_K(c,t);     //Turbulent kinetic energy(k)
real e   = C_D(c,t);     //Dissipation rate (epsilon)
        T=C_K(c,t)/C_D(c,t); // k/e
// Define stress tensor
                        S11 = 0.5*( C_DUDX(c,t)+C_DUDX(c,t) );
                        S12 = 0.5*( C_DUDY(c,t)+C_DVDX(c,t) );
                        S13 = 0.5*( C_DUDZ(c,t)+C_DWDX(c,t) );
                        S21 = 0.5*( C_DVDX(c,t)+C_DUDY(c,t) );
                        S22 = 0.5*( C_DVDY(c,t)+C_DVDY(c,t) );
                        S23 = 0.5*( C_DVDZ(c,t)+C_DWDY(c,t) );
                        S31 = 0.5*( C_DWDX(c,t)+C_DUDZ(c,t) );
                        S32 = 0.5*( C_DWDY(c,t)+C_DVDZ(c,t) );
                        S33 = 0.5*( C_DWDZ(c,t)+C_DWDZ(c,t) );
S=T*sqrt(
2*(S11*S11+S12*S12+S13*S13+S21*S21+S22*S22+S23*S23+S31*S31+S32*S32
+S33*S33) );
// Define vorticity tensor
                        W11 = 0.5*( C_DUDX(c,t)-C_DUDX(c,t) );
                        W12 = 0.5*( C_DUDY(c,t)-C_DVDX(c,t) );
                        W13 = 0.5*( C_DUDZ(c,t)-C_DWDX(c,t) );
                        W21 = 0.5*( C_DVDX(c,t)-C_DUDY(c,t) );
                        W22 = 0.5*( C_DVDY(c,t)-C_DVDY(c,t) );
                        W23 = 0.5*( C_DVDZ(c,t)-C_DWDY(c,t) );
```

```
                    W31 = 0.5*( C_DWDX(c,t)-C_DUDZ(c,t) );
                    W32 = 0.5*( C_DWDY(c,t)-C_DVDZ(c,t) );
                    W33 = 0.5*( C_DWDZ(c,t)-C_DWDZ(c,t) );
W=T*sqrt(
2*(W11*W11+W12*W12+W13*W13+W21*W21+W22*W22+W23*W23+W31*W
31+W32*W32+W33*W33) );
// Define kinenatic viscosity
v = u/p;
// Define turbulent Reynolds number
Ret=k*k/(e*v);
// Define Constant Cmu
c_mu=(0.3/(1.0+(0.35* (MAX(S,W))*0.5)))*(1.0-exp(-0.36/(exp(-0.75* MAX(S,W)
))));
// Define Damping function of Craft et al.(1996)
fmu=(1.0-exp(-pow((Ret/90.),0.5)-pow((Ret/400.),2) ));
// Define Eddy viscosity
mu_t=c_mu*T*C_K(c,t);
return mu_t;
}
DEFINE_ADJUST(rsm_adjust,domain)
{
Thread  *t;
cell_t  c;
real X;
real mu_t, c_mu;
real S11, S12, S13, S21, S22, S23, S31, S32, S33;
real W11, W12, W13, W21, W22, W23, W31, W32, W33;
real S, W, Sij, Wij;
real C_4, C_6, C_7;
real P11, P22, P33, Pkk;
real tau_w, u_tauw, u_star, y_star, Gk, u_mag;
real u ;
```

```
real p ;
real v, Ret,fmu;
real k ;
real e ;
/* Set the turbulent viscosity */
thread_loop_c(t,domain)
if (FLUID_THREAD_P(t))
{
begin_c_loop(c,t)
{
// Define variable
u= C_MU_L(c,t); //laminar viscosity
p= C_R(c,t);     //density
k= C_K(c,t);     //Turbulent kinetic energy(k)
e= C_D(c,t);     //Dissipation rate (epsilon)
X=C_K(c,t)/C_D(c,t); // k/e
// Define stress tensor
                        S11 = 0.5*( C_DUDX(c,t)+C_DUDX(c,t) );
                        S12 = 0.5*( C_DUDY(c,t)+C_DVDX(c,t) );
                        S13 = 0.5*( C_DUDZ(c,t)+C_DWDX(c,t) );
                        S21 = 0.5*( C_DVDX(c,t)+C_DUDY(c,t) );
                        S22 = 0.5*( C_DVDY(c,t)+C_DVDY(c,t) );
                        S23 = 0.5*( C_DVDZ(c,t)+C_DWDY(c,t) );
                        S31 = 0.5*( C_DWDX(c,t)+C_DUDZ(c,t) );
                        S32 = 0.5*( C_DWDY(c,t)+C_DVDZ(c,t) );
                        S33 = 0.5*( C_DWDZ(c,t)+C_DWDZ(c,t) );
S=X*sqrt(
2*(S11*S11+S12*S12+S13*S13+S21*S21+S22*S22+S23*S23+S31*S31+S32*S32
+S33*S33) );
// Define vorticity tensor
                        W11 = 0.5*( C_DUDX(c,t)-C_DUDX(c,t) );
                        W12 = 0.5*( C_DUDY(c,t)-C_DVDX(c,t) );
```

```
W13 = 0.5*( C_DUDZ(c,t)-C_DWDX(c,t) );
W21 = 0.5*( C_DVDX(c,t)-C_DUDY(c,t) );
W22 = 0.5*( C_DVDY(c,t)-C_DVDY(c,t) );
W23 = 0.5*( C_DVDZ(c,t)-C_DWDY(c,t) );
W31 = 0.5*( C_DWDX(c,t)-C_DUDZ(c,t) );
W32 = 0.5*( C_DWDY(c,t)-C_DVDZ(c,t) );
W33 = 0.5*( C_DWDZ(c,t)-C_DWDZ(c,t) );
W=X*sqrt(
2*(W11*W11+W12*W12+W13*W13+W21*W21+W22*W22+W23*W23+W31*W
31+W32*W32+W33*W33) );
// Define kinenatic viscosity
v = u/p;
// Define turbulent Reynolds number
Ret=k*k/(e*v);
// Define Constant Cmu
c_mu=(0.3/(1.0+(0.35* (MAX(S,W))*0.5)))*(1.0-exp(-0.36/(exp(-0.75* MAX(S,W)
))));
// Define Damping function of Craft et al.(1996)
fmu=(1.0-exp(-pow((Ret/90.),0.5)-pow((Ret/400.),2) ));
// Define Eddy viscosity
mu_t=c_mu*fmu*C_K(c,t)*C_K(c,t)*C_D(c,t);
// Define constant of Craft et al.(1996)
C_4=-10.0*SQR(c_mu);
C_6=-5.0*SQR(c_mu);
C_7=5.0*SQR(c_mu);
// Define tensor index = SklSkl
Sij=S11*S11+S12*S12+S13*S13+S21*S21+S22*S22+S23*S23+S31*S31+S32*S32
+S33*S33;
// Define tensor index = WklWkl
Wij=W11*W11+W12*W12+W13*W13+W21*W21+W22*W22+W23*W23+W31*
W31+W32*W32+W33*W33;
```

```
//*****************************************************************
**//
//************** IMPORTANT define only anisotropy tensor
******************//
//************** and multiply k in the Reynolds stress
************************//
//**************  anisotropy
tensor*************************************//
//*****************************************************************
*//
//*****************************************************************
*//
// Define anisotropy tensor of normal Reynolds stress
C_UDSI(c,t,UU)=
                        C_1*mu_t*X*   (S11*S11+S12*S12+S13*S13-1./3.*Sij)
                       + 2.*C_2*mu_t*X*   (W12*S12+W13*S13)
                       + C_3*mu_t*X*   (W12*W12+W13*W13-1./3.*Wij)
                       -2.*C_4*mu_t*X*X*
(W12*(S11*S12+S12*S22+S13*S23)+W13*(S11*S13+S12*S23+S13*S33))
                       +    C_6*mu_t*X*X* (S11*Sij)
                       +    C_7*mu_t*X*X* (S11*Wij);

C_UDSI(c,t,VV)=
                            C_1*mu_t*X*   (S12*S12+S22*S22+S23*S23-1./3.*Sij)
                       + 2.*C_2*mu_t*X*   (W23*S23-W12*S12)
                        +    C_3*mu_t*X*   (W12*W12+W23*W23-1./3.*Wij)
                       + 2.*C_4*mu_t*X*X* (W12*(S11*S12+S12*S22+S13*S23)-
W23*(S12*S13+S22*S23+S23*S33))
                        +    C_6*mu_t*X*X* (S22*Sij)
                        +    C_7*mu_t*X*X* (S22*Wij);
```

```
C_UDSI(c,t,WW)=
                        C_1*mu_t*X*   (S13*S13+S23*S23+S33*S33-1./3.*Sij)
                    - 2.*C_2*mu_t*X*   (W13*S13+W23*S23)
                    + C_3*mu_t*X*   (W13*W13+W23*W23-1./3.*Wij)
                    +2.*C_4*mu_t*X*X*
(W13*(S11*S13+S12*S23+S13*S33)+W23*(S12*S13+S22*S23+S23*S33))
                    +   C_6*mu_t*X*X* (S33*Sij)
                    +   C_7*mu_t*X*X* (S33*Wij);
// Define anisotropy tensor of shear Reynolds stress
        C_UDSI(c,t,UV)=
                        C_1*mu_t*X*   (S11*S12+S12*S22+S13*S23)
                    +   C_2*mu_t*X*   (W12*(S22-S11)+W13*S23+W23*S13)
                    +   C_3*mu_t*X*   (W13*W23)
                    + C_4*mu_t*X*X* (W12*(S11*S11+S13*S13-S22*S22-
S23*S23)-W13*(S12*S13+S22*S23+S23*S33)-
W23*(S11*S13+S12*S23+S13*S33))
                    +   C_6*mu_t*X*X* (S12*Sij)
                    +   C_7*mu_t*X*X* (S12*Wij);

        C_UDSI(c,t,UW)=
                        C_1*mu_t*X*   (S11*S13+S12*S23+S13*S33)
                    +   C_2*mu_t*X*   (W13*(S33-S11)+W12*S23-W23*S12)
                    -   C_3*mu_t*X*   (W12*W23)
                    +   C_4*mu_t*X*X* (W13*(S11*S11+S12*S12-S23*S23-
S33*S33)-
W12*(S11*S13+S22*S23+S23*S33)+W23*(S11*S12+S12*S22+S13*S23))
                    +   C_6*mu_t*X*X* (S13*Sij)
                    +   C_7*mu_t*X*X* (S13*Wij);

        C_UDSI(c,t,VW)=
                        C_1*mu_t*X*   (S12*S13+S22*S23+S23*S33)
                    +   C_2*mu_t*X*   (W23*(S33-S22)-W12*S13-W13*S12)
```

```
                    +   C_3*mu_t*X*   (W12*W13)
                    +   C_4*mu_t*X*X* (W23*(S12*S12+S22*S22-S13*S13-
S33*S33)+W12*(S11*S13+S12*S23+S13*S33)+W13*(S11*S12+S12*S22+S13*S2
3))
                    +   C_6*mu_t*X*X* (S23*Sij)
                    +   C_7*mu_t*X*X* (S23*Wij);
// Define memory Reynolds stress by User defined Memory
//*************************************************************
***//
//**************IMPORTANT define Linear and Nonlinear
Term*****************//
//*************************************************************
**//
C_UDMI(c,t,0)=      2./3.*C_K(c,t) -2.*mu_t*S11  + C_UDSI(c,t,UU);
C_UDMI(c,t,1)=      2./3.*C_K(c,t) -2.*mu_t*S22  + C_UDSI(c,t,VV);
C_UDMI(c,t,2)=      2./3.*C_K(c,t) -2.*mu_t*S33  + C_UDSI(c,t,WW);
C_UDMI(c,t,3)=      -2.*mu_t*S12  + C_UDSI(c,t,UV);
C_UDMI(c,t,4)=      -2.*mu_t*S13  + C_UDSI(c,t,UW);
C_UDMI(c,t,5)=      -2.*mu_t*S23  + C_UDSI(c,t,VW);
}
end_c_loop(c,t)
}
}
```

Appendix F

UDF of Modified Nonlinear Turbulence Model for Simulation of Cavitation on Marine Propeller

UDF of Modified Nonlinear Turbulence Model
for Simulation of Cavitation on Marine Propeller

```
#include "udf.h"
#include "math.h"
/* Turbulence model constants */
const real C_1=-0.1;
const real C_2=0.19;
const real C_3=0.21;
const real C_5=0.0;
// User-defined scalars : Define Reynolds stress
enum
{
UU,
VV,
WW,
UV,
UW,
VW
};
// Define source in x momentun equation
DEFINE_SOURCE(u_source, c, t, dS, eqn)
{
real source;
dS[eqn]=0.0;
source= - C_R(c,t) * ( C_UDSI_G(c,t,UU)[0] +
C_UDSI_G(c,t,UV)[1]+C_UDSI_G(c,t,UW)[2] );
return source;
}
// Define source in y momentun equation
DEFINE_SOURCE(v_source, c, t, dS, eqn)
{
```

```
real source;
dS[eqn]= 0.0;
source = - C_R(c,t)*(C_UDSI_G(c,t,UV)[0]+ C_UDSI_G(c,t,VV)[1] +
C_UDSI_G(c,t,VW)[2] ) ;
return source;
}
// Define source in z momentun equation
DEFINE_SOURCE(w_source, c, t, dS, eqn)
{
real source;
dS[eqn]= 0.0;
source= -C_R(c,t)*(C_UDSI_G(c,t,UW)[0] + C_UDSI_G(c,t,VW)[1] +
C_UDSI_G(c,t,WW)[2] );
return source;
}
DEFINE_SOURCE(k_source, c, t, dS, eqn)
{
real prod;
dS[eqn]= 0.0;
prod=-C_R(c,t)*(C_UDSI(c,t,UU)*C_DUDX(c,t)+C_UDSI(c,t,UV)*C_DVDX(c,t)+
C_UDSI(c,t,UW)*C_DWDX(c,t)+C_UDSI(c,t,UV)*C_DUDY(c,t)+C_UDSI(c,t,VV)
*C_DVDY(c,t)+C_UDSI(c,t,VW)*C_DWDY(c,t)+C_UDSI(c,t,UW)*C_DUDZ(c,t)+
C_UDSI(c,t,VW)*C_DVDZ(c,t) + C_UDSI(c,t,WW)*C_DWDZ(c,t));
return prod;
}

DEFINE_SOURCE(e_source, c, t, dS, eqn)
{
real prod;
real c_1e =1.44;
real c_2e =1.92;
real k   = C_K(c,t);
```

```
real e   = C_D(c,t);
real u = C_MU_L(c,t);
real p   = C_R(c,t);
real v, Ret,fe2,cc;
        v = u/p;
        Ret=k*k/(e*v);
        dS[eqn]= 0.0;
        prod= - c_1e*C_D(c,t)/C_K(c,t)* C_R(c,t)
                                * ( C_UDSI(c,t,UU)*C_DUDX(c,t) +
C_UDSI(c,t,UV)*C_DVDX(c,t) + C_UDSI(c,t,UW)*C_DWDX(c,t)+
                                C_UDSI(c,t,UV)*C_DUDY(c,t) +
C_UDSI(c,t,VV)*C_DVDY(c,t) + C_UDSI(c,t,VW)*C_DWDY(c,t)+
                                C_UDSI(c,t,UW)*C_DUDZ(c,t) +
C_UDSI(c,t,VW)*C_DVDZ(c,t) + C_UDSI(c,t,WW)*C_DWDZ(c,t)
                );
        fe2=1.0- 0.3*(exp(-Ret*Ret));
        return prod- (c_2e*C_R(c,t)*e*e/k)+ (fe2*c_2e*C_R(c,t)*e*e/k );
}
// Define Eddy  viscosity
DEFINE_TURBULENT_VISCOSITY(Craft_Mu_t,c,t)
{
real S,W,c_mu,T,mu_t;
real S11, S12, S13, S21, S22, S23, S31, S32, S33;
real W11, W12, W13, W21, W22, W23, W31, W32, W33;
// Define variable
real u = C_MU_L(c,t); //laminar viscosity
real p   = C_R(c,t);     //density
real v, Ret,fmu;
real k   = C_K(c,t);     //Turbulent kinetic energy(k)
real e   = C_D(c,t);     //Dissipation rate (epsilon)
        T=C_K(c,t)/C_D(c,t); // k/e
// Define stress tensor
```

```
                    S11 = 0.5*( C_DUDX(c,t)+C_DUDX(c,t) );
                    S12 = 0.5*( C_DUDY(c,t)+C_DVDX(c,t) );
                    S13 = 0.5*( C_DUDZ(c,t)+C_DWDX(c,t) );
                    S21 = 0.5*( C_DVDX(c,t)+C_DUDY(c,t) );
                    S22 = 0.5*( C_DVDY(c,t)+C_DVDY(c,t) );
                    S23 = 0.5*( C_DVDZ(c,t)+C_DWDY(c,t) );
                    S31 = 0.5*( C_DWDX(c,t)+C_DUDZ(c,t) );
                    S32 = 0.5*( C_DWDY(c,t)+C_DVDZ(c,t) );
                    S33 = 0.5*( C_DWDZ(c,t)+C_DWDZ(c,t) );
S=T*sqrt(
2*(S11*S11+S12*S12+S13*S13+S21*S21+S22*S22+S23*S23+S31*S31+S32*S32
+S33*S33) );
// Define vorticity tensor
                    W11 = 0.5*( C_DUDX(c,t)-C_DUDX(c,t) );
                    W12 = 0.5*( C_DUDY(c,t)-C_DVDX(c,t) );
                    W13 = 0.5*( C_DUDZ(c,t)-C_DWDX(c,t) );
                    W21 = 0.5*( C_DVDX(c,t)-C_DUDY(c,t) );
                    W22 = 0.5*( C_DVDY(c,t)-C_DVDY(c,t) );
                    W23 = 0.5*( C_DVDZ(c,t)-C_DWDY(c,t) );
                    W31 = 0.5*( C_DWDX(c,t)-C_DUDZ(c,t) );
                    W32 = 0.5*( C_DWDY(c,t)-C_DVDZ(c,t) );
                    W33 = 0.5*( C_DWDZ(c,t)-C_DWDZ(c,t) );
W=T*sqrt(
2*(W11*W11+W12*W12+W13*W13+W21*W21+W22*W22+W23*W23+W31*W
31+W32*W32+W33*W33) );
// Define kinenatic viscosity
v = u/p;
// Define turbulent Reynolds number
Ret=k*k/(e*v);
// Define Constant Cmu
c_mu=(0.3/(1.0+(0.35* (MAX(S,W))*0.5)))*(1.0-exp(-0.36/(exp(-0.75* MAX(S,W)
))));
```

```
// Define Damping function of Craft et al.(1996)
fmu=(1.0-exp(-pow((Ret/90.),0.5)-pow((Ret/400.),2) ));
// Define Eddy viscosity
mu_t=c_mu*T*C_K(c,t);
return mu_t;
}
DEFINE_ADJUST(rsm_adjust,domain)
{
Thread  *t;
cell_t  c;
real X;
real mu_t, c_mu;
real S11, S12, S13, S21, S22, S23, S31, S32, S33;
real W11, W12, W13, W21, W22, W23, W31, W32, W33;
real S, W, Sij, Wij;
real C_4, C_6, C_7;
real P11, P22, P33, Pkk;
real tau_w, u_tauw, u_star, y_star, Gk, u_mag;
real u ;
real p ;
real v, Ret,fmu;
real k ;
real e ;
/* Set the turbulent viscosity */
thread_loop_c(t,domain)
if (FLUID_THREAD_P(t))
{
begin_c_loop(c,t)
{
// Define variable
u= C_MU_L(c,t); //laminar viscosity
p= C_R(c,t);     //density
```

```
k= C_K(c,t);     //Turbulent kinetic energy(k)
e= C_D(c,t);     //Dissipation rate (epsilon)
X=C_K(c,t)/C_D(c,t); // k/e
// Define stress tensor
                    S11 = 0.5*( C_DUDX(c,t)+C_DUDX(c,t) );
                    S12 = 0.5*( C_DUDY(c,t)+C_DVDX(c,t) );
                    S13 = 0.5*( C_DUDZ(c,t)+C_DWDX(c,t) );
                    S21 = 0.5*( C_DVDX(c,t)+C_DUDY(c,t) );
                    S22 = 0.5*( C_DVDY(c,t)+C_DVDY(c,t) );
                    S23 = 0.5*( C_DVDZ(c,t)+C_DWDY(c,t) );
                    S31 = 0.5*( C_DWDX(c,t)+C_DUDZ(c,t) );
                    S32 = 0.5*( C_DWDY(c,t)+C_DVDZ(c,t) );
                    S33 = 0.5*( C_DWDZ(c,t)+C_DWDZ(c,t) );
S=X*sqrt(
2*(S11*S11+S12*S12+S13*S13+S21*S21+S22*S22+S23*S23+S31*S31+S32*S32
+S33*S33) );
// Define vorticity tensor
                    W11 = 0.5*( C_DUDX(c,t)-C_DUDX(c,t) );
                    W12 = 0.5*( C_DUDY(c,t)-C_DVDX(c,t) );
                    W13 = 0.5*( C_DUDZ(c,t)-C_DWDX(c,t) );
                    W21 = 0.5*( C_DVDX(c,t)-C_DUDY(c,t) );
                    W22 = 0.5*( C_DVDY(c,t)-C_DVDY(c,t) );
                    W23 = 0.5*( C_DVDZ(c,t)-C_DWDY(c,t) );
                    W31 = 0.5*( C_DWDX(c,t)-C_DUDZ(c,t) );
                    W32 = 0.5*( C_DWDY(c,t)-C_DVDZ(c,t) );
                    W33 = 0.5*( C_DWDZ(c,t)-C_DWDZ(c,t) );
W=X*sqrt(
2*(W11*W11+W12*W12+W13*W13+W21*W21+W22*W22+W23*W23+W31*W
31+W32*W32+W33*W33) );
// Define kinenatic viscosity
v = u/p;
// Define turbulent Reynolds number
```

```
Ret=k*k/(e*v);
// Define Constant Cmu
c_mu=(0.3/(1.0+(0.35* (MAX(S,W))*0.5)))*(1.0-exp(-0.36/(exp(-0.75* MAX(S,W)
)))));
// Define Damping function of Craft et al.(1996)
fmu=(1.0-exp(-pow((Ret/90.),0.5)-pow((Ret/400.),2) ));
// Define Eddy viscosity
mu_t=c_mu*fmu*C_K(c,t)*C_K(c,t)*C_D(c,t);
// Define constant of Craft et al.(1996)
C_4=-10.0*SQR(c_mu);
C_6=-5.0*SQR(c_mu);
C_7=5.0*SQR(c_mu);
// Define tensor index = SklSkl
Sij=S11*S11+S12*S12+S13*S13+S21*S21+S22*S22+S23*S23+S31*S31+S32*S32
+S33*S33;
// Define tensor index = WklWkl
Wij=W11*W11+W12*W12+W13*W13+W21*W21+W22*W22+W23*W23+W31*
W31+W32*W32+W33*W33;
//*****************************************************************
**//
//************** IMPORTANT define only anisotropy tensor
*******************//
//************** and multiply k in the Reynolds stress
************************//
//**************  anisotropy
tensor*************************************//
//*****************************************************************
*//
//*****************************************************************
*//
// Define anisotropy tensor of normal Reynolds stress
C_UDSI(c,t,UU)=
```

```
                    C_1*mu_t*X*   (S11*S11+S12*S12+S13*S13-1./3.*Sij)
                   + 2.*C_2*mu_t*X*   (W12*S12+W13*S13)
                   + C_3*mu_t*X*   (W12*W12+W13*W13-1./3.*Wij)
                   -2.*C_4*mu_t*X*X*
(W12*(S11*S12+S12*S22+S13*S23)+W13*(S11*S13+S12*S23+S13*S33))
                   +   C_6*mu_t*X*X* (S11*Sij)
                   +   C_7*mu_t*X*X* (S11*Wij);

C_UDSI(c,t,VV)=
                       C_1*mu_t*X*   (S12*S12+S22*S22+S23*S23-1./3.*Sij)
                   + 2.*C_2*mu_t*X*   (W23*S23-W12*S12)
                    +   C_3*mu_t*X*   (W12*W12+W23*W23-1./3.*Wij)
                   + 2.*C_4*mu_t*X*X* (W12*(S11*S12+S12*S22+S13*S23)-
W23*(S12*S13+S22*S23+S23*S33))
                    +   C_6*mu_t*X*X* (S22*Sij)
                    +   C_7*mu_t*X*X* (S22*Wij);

C_UDSI(c,t,WW)=
                      C_1*mu_t*X*   (S13*S13+S23*S23+S33*S33-1./3.*Sij)
                   - 2.*C_2*mu_t*X*   (W13*S13+W23*S23)
                   + C_3*mu_t*X*   (W13*W13+W23*W23-1./3.*Wij)
                   +2.*C_4*mu_t*X*X*
(W13*(S11*S13+S12*S23+S13*S33)+W23*(S12*S13+S22*S23+S23*S33))
                   +   C_6*mu_t*X*X* (S33*Sij)
                   +   C_7*mu_t*X*X* (S33*Wij);
// Define anisotropy tensor of shear Reynolds stress
        C_UDSI(c,t,UV)=
                      C_1*mu_t*X*   (S11*S12+S12*S22+S13*S23)
                   +   C_2*mu_t*X*   (W12*(S22-S11)+W13*S23+W23*S13)
                   +   C_3*mu_t*X*   (W13*W23)
```

```
                    + C_4*mu_t*X*X* (W12*(S11*S11+S13*S13-S22*S22-
S23*S23)-W13*(S12*S13+S22*S23+S23*S33)-
W23*(S11*S13+S12*S23+S13*S33))
                    +   C_6*mu_t*X*X* (S12*Sij)
                    +   C_7*mu_t*X*X* (S12*Wij);

      C_UDSI(c,t,UW)=
                       C_1*mu_t*X*   (S11*S13+S12*S23+S13*S33)
                    +   C_2*mu_t*X*   (W13*(S33-S11)+W12*S23-W23*S12)
                    -   C_3*mu_t*X*   (W12*W23)
                    +   C_4*mu_t*X*X* (W13*(S11*S11+S12*S12-S23*S23-
S33*S33)-
W12*(S11*S13+S22*S23+S23*S33)+W23*(S11*S12+S12*S22+S13*S23))
                    +   C_6*mu_t*X*X* (S13*Sij)
                    +   C_7*mu_t*X*X* (S13*Wij);

      C_UDSI(c,t,VW)=
                       C_1*mu_t*X*   (S12*S13+S22*S23+S23*S33)
                    +   C_2*mu_t*X*   (W23*(S33-S22)-W12*S13-W13*S12)
                    +   C_3*mu_t*X*   (W12*W13)
                    +   C_4*mu_t*X*X* (W23*(S12*S12+S22*S22-S13*S13-
S33*S33)+W12*(S11*S13+S12*S23+S13*S33)+W13*(S11*S12+S12*S22+S13*S2
3))
                    +   C_6*mu_t*X*X* (S23*Sij)
                    +   C_7*mu_t*X*X* (S23*Wij);
// Define memory Reynolds stress by User defined Memory
//*****************************************************************
***//
//**************IMPORTANT define Linear and Nonlinear
Term*****************//
//*****************************************************************
**//
```

```
C_UDMI(c,t,0)=        2./3.*C_K(c,t) -2.*mu_t*S11  + C_UDSI(c,t,UU);
C_UDMI(c,t,1)=        2./3.*C_K(c,t) -2.*mu_t*S22  + C_UDSI(c,t,VV);
C_UDMI(c,t,2)=        2./3.*C_K(c,t) -2.*mu_t*S33  + C_UDSI(c,t,WW);
C_UDMI(c,t,3)=       -2.*mu_t*S12  + C_UDSI(c,t,UV);
C_UDMI(c,t,4)=       -2.*mu_t*S13  + C_UDSI(c,t,UW);
C_UDMI(c,t,5)=       -2.*mu_t*S23  + C_UDSI(c,t,VW);
}
end_c_loop(c,t)
}
}
```

CURRICULUM VITAE

NAME : Mr. Prachakon Kaewkhiaw

BIRTH DATE : February 7, 1985

BIRTH PLACE : Suphanburi, Thailand

EDUCATION :

YEAR	**INSTITUTE**	**DEGREE/DIPLOMA**
2007	Kasetsart Univ.	B.Eng. (Naval Architecture And Marine Engineering)