**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA

INGENIERÍA CIVIL – HIDRÁULICA

# MODELACIÓN TERMODINÁMICA DE FLUIDOS MULTIFÁSICOS: APLICACIÓN EN DISPOSITIVOS DE CONVERSIÓN DE ENERGÍA TÉRMICA OCEÁNICA

TESIS

QUE PARA OPTAR POR EL GRADO DE:

DOCTORA EN INGENIERÍA


PRESENTA:

PAOLA ELIZABETH RODRÍGUEZ OCAMPO

TUTOR PRINCIPAL

RODOLFO SILVA CASARÍN, INSTITUTO DE INGENIERÍA

COMITÉ TUTOR

CARLOS AGUSTÍN ESCALANTE SANDOVAL, FACULTAD DE INGENIERÍA

EDGAR GERARDO MENDOZA BALDWIN, INSTITUTO DE INGENIERÍA

LILIA CHÁVEZ REYES, FACULTAD DE INGENIERÍA

JUAN CARLOS ALCÉRRECA HUERTA, CONACYT – ECOSUR


CIUDAD DE MÉXICO, FEBRERO 2020

**JURADO ASIGNADO:**

| | |
|---|---|
| Presidente: | DR. CARLOS AGUSTÍN ESCALANTE SANDOVAL |
| Secretario: | DRA. LILIA CHÁVEZ REYES |
| 1 er. Vocal: | DR. RODOLFO SILVA CASARÍN |
| 2 do. Vocal: | DR. EDGAR GERARDO MENDOZA BALDWIN |
| 3 er. Vocal: | DR. JUAN CARLOS ALCÉRRECA HUERTA |

Lugar donde se realizó la tesis: CIUDAD DE MÉXICO

**TUTOR DE TESIS:**

RODOLFO SILVA CASARÍN

---

**FIRMA**

*(Seg u n d a ho j a )*

## Agradecimientos

*A mis papás, Delia y Víctor, porque con su ejemplo y apoyo me han enseñado a valorar las oportunidades y a superarme en todos los aspectos de mi vida.*

*A mi esposo, Michael, por toda su ayuda, paciencia y apoyo en esta etapa.*

*A mis hermanos, Juan Carlos e Isaac, y a Judith, con cariño. A mi familia y amigos.*

*A la UNAM por su generosa educación y a CONACYT por contribuir a solventar los gastos de mis estudios.*

*A todos los maestros que me inspiraron y me apoyaron en esta etapa, en especial al Dr. Rodolfo, al Dr. Edgar, al Dr. Juan Carlos y al Dr. Jassiel.*

# Content

# LIST OF FIGURES

# LIST OF TABLES

# ABSTRACT

Ocean Thermal Energy Conversion (OTEC) is a technique for extracting renewable energy by utilizing the temperature difference that exists between the ocean water layers to operate an turbine. The low efficiency of OTEC systems results in large facilities, which utilize and discharge great amounts of seawater with different temperature and nutrient concentration. Numerous authors agree that small ($\leq 1$ MW) land-based plants' water discharge is environmentally acceptable at a depth of 60 m. Nevertheless, due to the large effluent discharged in a commercial-size ($\geq 100$ MW) OTEC plant, an extensive environmental impact study is necessary. Computational Fluid Dynamics (CFD) tools, coupled, with a trophic ecosystem model can provide valuable information by simulating the near-field plume dilution and trajectory inside the complex conditions of coastal waters.

This work consists in the development, validation, and implementation of an OpenFOAM®-based solver suitable for simulating OTEC water discharge. The developed solver is capable of handling multiphase fluid flow with the following characteristics: three fluid phases with different densities and temperatures, i.e., two of the phases are liquids and miscible, and the third one represents air; thus, there is a free-surface condition. The ability to simulate wave generation/absorption was also introduced into the solver. The implemented governing equations can be synthetized as momentum conservation equation, mass conservation equation, scalar quantities conservation equation, volume fraction equation, and energy conservation equation in terms of the temperature field.

A benchmarking experimental model based on a dam-break case was built and numerically simulated to perform validation of the developed solver. The test case included two liquid fluids with different temperature, density, and hydraulic head, in a free-surface condition. The experimental model was instrumented with ten thermistors to measure the temperature variations, which were compared against the temperature series obtained in the numerical simulation results by ten numerical probes distributed analogously.

The validation process was used to estimate the accuracy of the temperature field distribution, and therefore, the energy equation implementation in OpenFOAM®. Three turbulence models were tested for this purpose: zero-equation RANS (Reynolds-Averaged Navier-Stokes) turbulence model, two-equation (k-omega) RANS model, and Large Eddy Simulation (LES) model. Independently of the differences between the turbulence models, none of them exhibits a mean error higher than 17 %. Thus, the validation tests proved an adequate performance of the numerical model and the correct implementation of the energy equation in terms of the temperature field. Finally, a demonstration case is presented to show the model applicability to OCEC water discharge in coastal waters.

## *Resumen*

*La Energía por Gradiente Térmico Oceánico (OTEC, por sus siglas en inglés) es una técnica para extraer energía renovable utilizando la diferencia de temperatura que existe entre las capas del océano para operar una turbina. La baja eficiencia de los sistemas OTEC resulta en la necesidad de grandes instalaciones, las cuales utilizan grandes cantidades de agua del mar con diferentes temperaturas y concentraciones de nutrientes. Numerosos autores coinciden*

*en que la descarga de agua por parte de plantas pequeñas ($\leq 1\ MW$) en tierra es ambientalmente aceptable a una profundidad de 60 m. Sin embargo, dado a los grandes gastos descargados por una planta comercial ($\geq 100\ MW$), un estudio extensivo de impacto ambiental es necesario. Herramientas de la Mecánica de Fluidos Computacional (CFD, por sus siglas en inglés), acopladas con un modelo de ecosistema trófico, puede proveer información valiosa al simular las vecindades de la pluma de descarga, su dilución y trayectoria dentro de las condiciones complejas de las aguas costeras.*

*Este trabajo consiste en el desarrollo, validación e implementación de una herramienta numérica de CFD dentro de* OpenFOAM*, adecuada para simular la descarga de agua de las plantas OTEC. El* solver *generado es capaz de manejar fluido multifásico con las siguientes características: tres fases fluidas con diferentes densidades y temperaturas, i.e., dos de las fases son líquidas y mezclables, y la tercera representa el aire; por lo tanto, existe una condición de superficie libre. La habilidad de simular la generación y absorción del oleaje también se introdujo en el* solver*. Las ecuaciones de gobierno implementadas se pueden resumir como: ecuación de conservación de la cantidad de movimiento, ecuación de conservación de la masa, ecuación de conservación de las cantidades escalares, ecuación de la fracción de volumen y la ecuación de conservación de la energía en términos del campo de temperatura.*

*Un caso experimental representativo que consiste en un caso de rotura de dique (dam-break) fue construido y simulado numéricamente para realizar la validación del* solver *desarrollado. El caso de prueba incluyó dos fluidos con diferente temperatura, densidad y carga hidráulica en una condición de superficie libre. El modelo experimental se instrumentó con diez termistores para medir las variaciones de la temperatura, las cuales fueron comparadas con las series de temperatura obtenidas en los resultados de la simulación numérica con sensores distribuidos análogamente.*

*El proceso de validación se usó para estimar la precisión de la distribución del campo de temperatura y, por lo tanto, la implementación de la ecuación de la energía en* OpenFoam®*. Tres modelos de turbulencia fueron evaluados para este propósito: el modelo RANS (*Reynolds-Averaged Navier-Stokes*, por sus siglas en inglés) de turbulencia de cero ecuaciones, el modelo RANS de dos ecuaciones (k-omega), y el modelo de Simulación de Grandes Remolinos (LES, pos sus siglas en inglés). Independientemente de los resultados de los diferentes modelos de turbulencia, ninguno de ellos exhibe un error promedio mayor al 17 %. Por lo tanto, los análisis de validación prueban un desempeño adecuado del modelo numérico, así como la correcta implementación de la ecuación de la energía, en términos del campo de temperatura. Finalmente, un caso demostrativo es presentado para mostrar la aplicabilidad del modelo a la descarga de agua de las plantas OTEC.*

# 1

# INTRODUCTION

# 1. INTRODUCTION

Energy harvesting is a concept by which energy is captured, stored, and utilized using various sources and employing different devices. Renewable energy harvesting like solar, wind, ocean, hydro, electromagnetic, electrostatic, thermal, vibration, and the human body, differ from the conventional power systems because it does not require fossil fuels (e.g., oil, coal, and gas) for its operation. Factors such as economic, geopolitical, and environmental reasons have motivated the nations to accelerate energy harvesting from renewable sources. Science and engineering communities support the improvement of this technology by boosting the research on renewable energies and seeking to increase the efficiency of the devices for this purpose.

The present work focuses on one type of renewable energy harvesting called Ocean Thermal Energy Conversion (OTEC) (Figure 1), and the implementation of a numerical model to recreate the operation of the OTEC devices, particularly OTEC water discharge in the coasts. The present research seeks to contribute to the feasibility evaluation of the future implementation of this technology in Mexico.

*Figure 1. OTEC Test Facility at the Natural Energy Laboratory of Hawaii Authority, Kailua-Kona. Taken from (Makai Ocean Engineering, 2008).*

## 1.1. JUSTIFICATION

Although different authors (see Nihous and Vega (1993), Vega (2002), Vega (2012), Grandelli et al. (2012)) agree that a discharge depth of approximately $60 - 70\ \mathrm{m}$ is deep enough to avoid substantial environmental impact, most of the studies have been performed with specific OTEC

plants designs and ocean conditions (i.e., waves, winds, tides, river discharges, currents, etc.), without considering the intricate and dynamic nature of the ocean.

It is not possible to determine the real OTEC thermal discharge impact through basic research or the development of ecological theory. Therefore, every conclusion needs to be confirmed with actual field measurements of at least 5 MW capacity pilot plants operating during 1 to 2 continuous years (Asian Development Bank, 2014). However, numerical modeling can be an excellent method to predict, as a first approach, the plume behavior under different discharge configurations and ocean conditions.

Multon (2012) states that numerical modeling analyses are essential to study the behavior of the ejected plume of water. Different configurations of depth, flow rate, currents, and temperature can be tested in advance to choose the most adequate option and minimize the environmental impact. Other advantages of performing numerical modeling include the measurement of the dilution coefficient, and to avoid reinjection into the plant by ensuring the vertical separation from the warm water intake. These values are a function of ocean current conditions.

Numerical modeling tools coupled with a trophic ecosystem model can suitable to perform numerical simulations of OTEC thermal discharge in a detailed fashion, as demonstrated in Grandelli et al. (2012), Lee et al. (2016), and Kim and Kim (2014). However, factors such as expensive licenses and closed source (i.e., proprietary software) can discourage the use of some software packages. Open source and free Computational Fluid Dynamics (CFD) tools can provide valuable information by simulating the near-field plume dilution and trajectory. A CFD numerical tool intended for the simulation of the hydrodynamic behavior of the plume discharged by an OTEC plant represents a great challenge due to the complex nature of thermal discharges in coastal waters.

This work consists in the development of a suitable CFD tool to simulate OTEC discharge water. This tool may include as many parameters as possible to reproduce the behavior of the thermal water discharge in coastal waters.

## 1.2. OVERALL OBJECTIVE

To develop a numerical model capable of solving thermodynamic interaction in multiphase-fluid flow and its implementation for the evaluation of the functioning of the OTEC systems.

## 1.3. SPECIFIC OBJECTIVES

- To develop a Computational Fluid Dynamic (CFD) model capable of simulating, in a detailed fashion, water discharge of OTEC plants in coastal waters, which includes thermal equilibrium, multiphase-fluid flow, currents and waves action in a turbulent flow system.
- To calibrate and validate the developed model through benchmarking experimental tests, i.e., determination of the numerical error and the accuracy of the solution of the implemented equations.

- To implement the numerical model in a representative numerical domain of thermal water discharge in coastal waters.
- To elaborate a conceptual design of a hypothetical OTEC plant that operates in a specific location in Mexico, which meets the suitable thermal resource within an acceptable distance from the shore for OTEC systems. The plant pre-dimensioning will provide general parameters that will be later introduced in the numerical model.
- To characterize the OTEC discharge water and external fluid mechanics through a CFD model under different conditions to describe the possible impact in the discharge area.
- To identify additional benefits and areas of opportunity for the implementation of OTEC technology in Mexico.

# 2

# OCEAN THERMAL ENERGY CONVERSION

# 2. OCEAN THERMAL ENERGY CONVERSION

First, this section describes Ocean Thermal Energy Conversion (OTEC) technology, its importance, and current knowledge. Then, an overview of the numerical modeling applied to OTEC systems is presented. Finally, the relevant implications for this work are summarized.

## 2.1. INTRODUCTION TO OTEC TECHNOLOGY

OTEC is a technique for extracting renewable energy by taking advantage of the thermal resource that exists between different ocean water layers. The sunlight falls on the ocean and water captures it within a shallow surface layer of approximately 100 m thick. The upper layer of the ocean has a temperature value near 28 °C in the regions of the tropical oceans (between approximately 15° north and 15° south). The temperature varies from $\sim 27$ °C to $\sim 29$ °C during the year and remains nearly constant day and night (Avery and Wu, 1994). Warm water is also present far away from the tropics, e.g., in places where power plants discharge. In contrast, a cold water layer that comes from ice-cold water melted in the Polar Regions flows along the ocean bottom towards the equator, displacing the lower-density warm water above. This creates a large reservoir of warm and cold water with a temperature difference $\Delta T$ of 22 – 25 K. OTEC processes use this temperature difference to operate an turbine and harvest electrical energy. According to Dessne et al., (2015) OTEC works efficiently where the temperature difference $\Delta T$ is at least 20 K.

One single 100 MW floating OTEC plant could provide electricity for more than 100 000 people by producing 800 million kWh per year; $120\ 000\ \mathrm{m}^3$of fresh water per day; and move $4\ \mathrm{km}^3$ of high nutrient deep seawater each day, enough to grow 70 ton of shellfish meat each day (Dessne et al., 2015). Nevertheless, current technology and current research in OTEC systems have been developed for much smaller scales (< 1 MW).

The net (electrical) power output of a OTEC plant is approximately 35 % of the gross power generated (Dessne et al., 2015). The rest 65 % supply the pumps of the plant itself. The larger the capacity of the plant, the larger is the margin of power available. The capacity factor of an OTEC plant is nearby 80 %.

In spite of the great extension of the tropical regions where the surface water temperature is suitable for OTEC energy harvesting, there are several limitations when it comes to deciding where to place an energy plant. Conventional OTEC does not seem feasible where the ocean depth is too small or where the deep ocean water not cold enough Figure 2 shows the global distribution of the ocean thermal resource, considering both limiting factors.

Some industrialized countries, such as Japan, India, USA and South Korea, have explored the OTEC technology with great success in the design and operation of small OTEC facilities, not only in the efficient production of electricity but also in the obtainment of useful byproducts. OTEC can be even more valuable in developing countries which are facing energetic and fresh water limitations. Particularly Mexico could take advantage of its geographical position and its vast marine resources to implement this kind of energy production for the benefit of society and the environment.

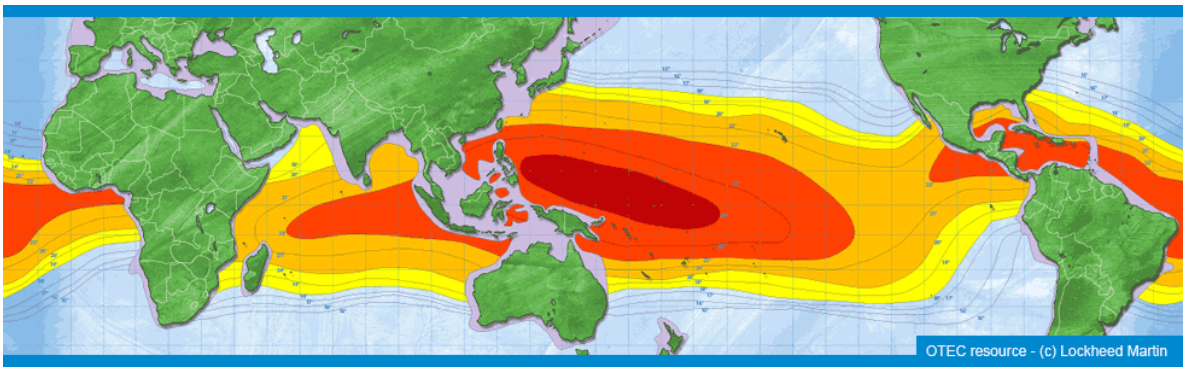


*Figure 2. Global distribution of the ocean thermal resource. Taken from (Lockheed Martin, 2014).*

OTEC plants can be installed i) on-shore, ii) off-shore on floating platforms (similar to oil platforms), or iii) placed on ships. On the one hand, on-shore platforms require shorter power cables, but large distances from deep cold water can be a limitation because of the cold water pipe costs. On the other hand, platforms installed offshore require long power cables to bring electricity to shore. OTEC plants placed on ships can follow spots of warmer surface water, temporarily storing the generated power and useful byproducts. (Dessne et al., 2015).

There are two main categories of the OTEC power systems: i) open-cycle (OC-OTEC) and ii) closed-cycle (CC-OTEC). The main difference between them is that in CC-OTEC, the working fluid never mixes with the seawater, whereas in OC-OTEC, the working fluid is the seawater and is vented after use.

Some of the major components of an OTEC facility are:

i) Coldwater pipe: used to draw cold water from below the thermocline.
ii) Heat exchangers: evaporators and condensers used to transfer heat between cold and warm waters and the working fluid.
iii) Platform/pipe interface: couples the cold and warm water pipes and platform.
iv) Platform.
v) Power cable: transfers electricity back to a shore-based electrical grid.
vi) Position keeping system: ensures that the OTEC facility remains stable and in the same location.
vii) Pumps: draw water (and the working fluid in closed-cycle) through the cold and warm water pipes.
viii) Turbines and generators: used to convert thermal energy into electricity.
ix) Warm water pipe: used to draw water from near the surface.
x) Warm and cold water discharge pipes: used to return the water after the heat has been extracted.
xi) Working fluid (for CC-OTEC only).

*Open-cycle OTEC*

It refers to the use of seawater as the working fluid. In OC-OTEC, warm surface water is deaerated and converted into low-pressure steam; it passes through a turbine and then exits into a condenser that works with deep cold seawater. To evaporate the warm water, a pump lowers the pressure until the water starts to boil. About 0.5 % of the water goes into the turbine chamber, where it drives the turbine, with the remainder falling back into the evaporator discharge pool (Nihous and Vega, 1996). The condensate can be used as desalinated water if a surface condenser is used. Figure 3 shows a diagram of the OC-OTEC.

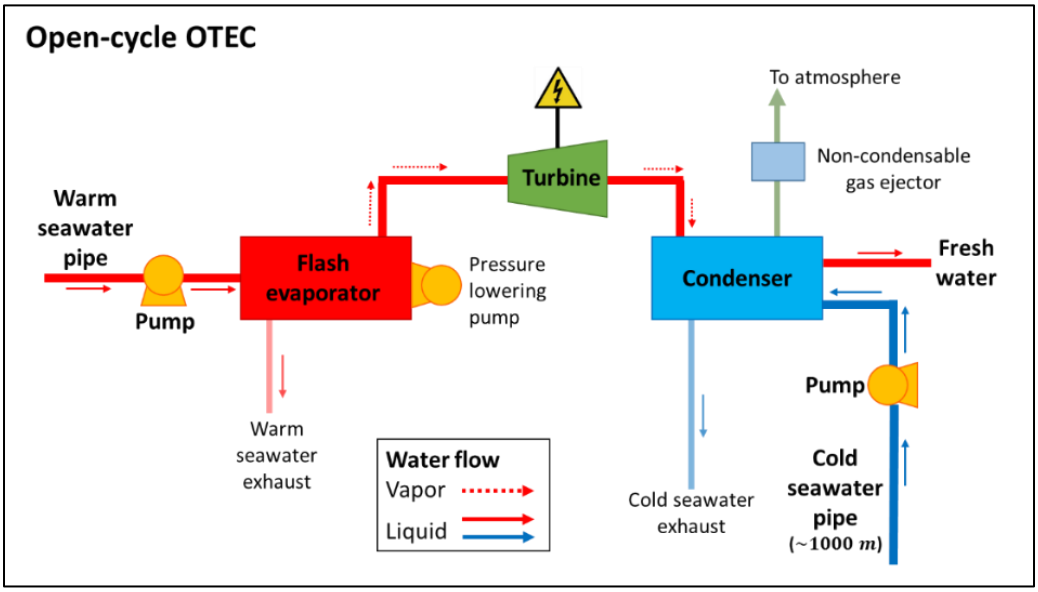


*Figure 3. Diagram of typical open-cycle OTEC engine power cycle.*

Among the by-products that the OTEC open-cycle systems can provide are:

I. Fresh water: Desalinated fresh water is important to supply the communities near the facility.

II. High-nutrient deep seawater: The cold water pumped from the bottom of the sea to drive the condenser is nutrient-rich and suitable to grow several kinds of shellfishes. This can encourage the development of a mariculture or aquaculture industry in close proximity of the OTEC plant.

III. Hydrogen: Desalinated water is a source of hydrogen. Research has been done about how to produce hydrogen with OTEC Technology, based on electrolysis method (leaving oxygen as a byproduct), as well as new ways to store and transport hydrogen for its future utilization in energy production (Ikegami et al., 2002). Hydrogen would be transported in liquid form from the OTEC plant ship to be primarily used as fuel. A 100 MW net plant can be configured to yield 1300 kg per hour of liquid hydrogen. Nowadays, using OTEC only for hydrogen gas production is not economically justifiable (Dessne et al., 2015). Oxygen produced in this process could be pumped back into the ocean to restore the ocean life in areas where needed.

IV. Trace elements: Gold, magnesium, and molybdenum are trace elements contained in great masses in the world's oceans. Today, it is extremely uneconomical to harvest these elements from the ocean, but it might be valuable in the future if the economic and technological situation requires it (Dessne et al., 2015).

*Closed-cycle OTEC*

The conventional Rankine cycle employed in steam engines is essentially the same used by OC-OTEC. CC-OTEC differs by using a different working fluid than water. The chosen working fluid must provide a high vapor pressure at relatively low temperatures. A common working fluid is ammonia with a vapor pressure of approximately 10 atm at 25 °C and 5 atm at 5 °C. Another fluid commonly used is propane, which has a vapor pressure of around 9.5 atm at 25 °C and 5.5 atm at 5 °C.

Warm water is pumped from the ocean surface and heats the working fluid in a heat exchanger. The high-pressure vapor passes through a turbine as it expands, which in turn drives an electric generator. The low-pressure vapor then passes through a condenser that works with cold water pumped from the depths ($\sim 1000$ m) at 5 °C. Unlike the OC-OTEC, the working fluid is not discharged but pumped back to the evaporator to continue the cycle. Figure 4 presents a diagram of the CC-OTEC.

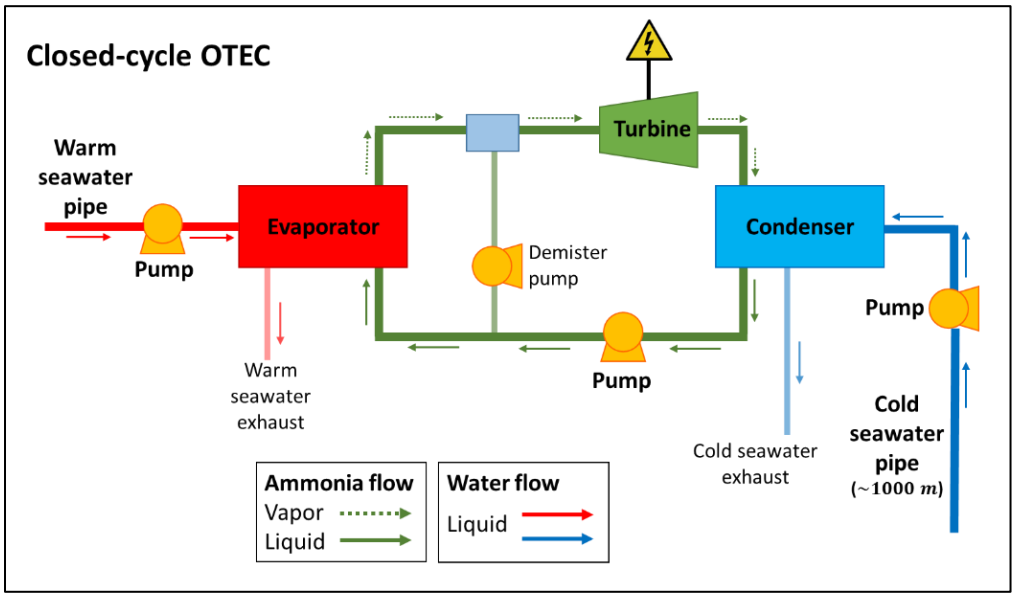


*Figure 4. Diagram of typical closed-cycle OTEC engine power cycle.*

Among the by-products that the CC-OTEC systems can provide are:

I. High-nutrient deep seawater: Alike the OC-OTEC, one of the CC-OTEC byproducts is the deep seawater for aquaculture purposes.
II. Seawater air conditioning (SWAC): SWAC relies on deep seawater being pumped into a cooling system. Places where OTEC is feasible, are usually the places with a high demand for

cool facilities for food storage, hospitals, and buildings in general. Thus, SWAC can be a cost-effective and environmentally-friendly way of cooling (Dessne et al., 2015).

CC-OTEC generates more electricity than OC-OTEC, but it does not generate fresh water. Vega (1999) proposed a two-stage OTEC hybrid cycle to maximize the use of the thermal resource. It consists of the production of electricity in the first stage (closed-cycle) followed by water production in the second stage. In the second stage, the temperature difference available from the first stage is used to produce desalinated water through a flash evaporator and a surface condenser. It is possible to add this second stage process in an OC-OTEC, resulting in doubling water production.

### 2.1.1. Efficiency

The maximum efficiency $\eta_{max}$ of an OTEC system is given by the Carnot cycle, which provides a theoretical limit of the efficiency of converting heat stored in the warm surface water into mechanical work (Equation 1).

$$\eta_{max} = \frac{T_w - T_c}{T_w} = 1 - \frac{T_c}{T_w} \qquad \text{Equation 1}$$

where $T_w$ is the absolute temperature of the warm water and $T_c$ is the absolute temperature of the cold water.

If the annual surface temperature is, for instance, $26\ °C$ $(299.15\ K)$ and $4\ °C$ $(277.15\ K)$ at the bottom, the maximum efficiency would be $\eta_{max} = \frac{22\ K}{299.15\ K} \approx 7.4\ \%$. However, this ideal efficiency does not consider friction, heat losses, or any other factor.

Avery and Wu (1994) consider that only the $\Delta T$ associated with the pressure difference across the turbine is associated with the production of mechanical work giving a gross power efficiency of about half the Carnot limit ($\eta_{max} \approx 3.7\ \%$ in the example). Typically, less than 3 % of the energy extracted from the surface water in OTEC is converted into electricity.

### 2.1.2. Cost implication

The low thermal efficiency means the need for large amounts of ocean water through the power system per kilowatt generated. Thus, the small temperature differences in OTEC technology is not only limiting the efficiency, but it implies large investments in large-scale components. The cost per kWh becomes smaller the bigger the plant; thus, building small plants makes little sense except as demonstrators (Dessne et al., 2015).

On the current market, the cost of electrical power from OTEC plants is higher than the price of oil-produced energy. However, the combination of energy production and byproducts could be feasible for some regions. Experience shows that OTEC technology is sustainable only if the water is used a second time as a by-product, e.g., fresh water and cold water for refrigeration systems.

According to economic evaluations, OTEC's commercial future lies in floating plants of $\sim 100$ MW capacity for industrialized nations and smaller plants for small-island-developing-states. Nowadays it is possible to design a $1.5 - 2$ MW OTEC plant, which is an appropriate size for small markets. For larger markets, it can be considered a modular system of four $25$ MW module capacity (Vega, 1999).

## 2.2. ENVIRONMENTAL IMPACT

When compared to conventional electricity-producing plants that work with fossil fuels, OTEC may offer a benign power production technology, since no noxious by-products are generated, and the handling of hazardous substances is limited to the working fluid (in CC-OTEC heat exchangers). The Asian Development Bank (2014) reports that the amount of carbon dioxide released from a OC-OTEC plant is about 1 % of the amount released by fuel-oil plants, and the value is even lower in CC-OTEC plants. A comprehensive study about the carbon dioxide release from OTEC cycles can be found at Green and Guenther (1990).

Potential environmental impact of the deployment and operation of OTEC plants have been widely studied from different perspectives (e.g., Avery and Wu, 1994, Asian Development Bank, 2014, and Dessne et al., 2015). Each effect can be characterized according to different criteria, e.g., probability, intensity, area of impact, the environment's sensitivity to that impact and its duration (temporary or permanent) (Multon, 2012). Temporary effects relate to the construction phase, while permanent effects relate to the operation phase. Table 1 summarizes the temporary and permanent effects related to OTEC technology.

The construction phase related impacts are similar to those associated with the construction of any other power plant, shipbuilding, or offshore platforms (Asian Development Bank, 2014). These temporal effects include noise and vibration, which may disturb the behavior of marine fauna (Multon, 2012).

On the other hand, the operation of an OTEC plant can be linked to the discharge of oily effluents, biocides, contaminated water, the risk of leaking of the working fluid (in CC-OTEC), the entrainment of fish and sea-living mammals into the pipes, and the thermal discharge of deep sea water.

*Table 1. Summary of the most relevant OTEC potential environmental impacts.*

| **OTEC environmental impact** | | |
|---|---|---|
| **Type** | **Source** | **Notes** |
| **Temporary (Construction phase)** | • noise, acoustic, and vibration | • Temporary impacts are similar to those associated with the construction of any power plant, shipbuilding, and offshore platforms |
| **Permanent (Operation phase)** | • Discharge of oily effluents, biocides, and contaminated water<br>• Risk of leaking of the working fluid<br>• Entrainment of fish and sea-living mammals into the pipes<br>• Thermal discharge of deep sea water | • Fish population reduction is possible due to impingement, entrainment, and the discharge of biocides.<br>• CC-OTEC operates with toxic/pollutant refrigerants; thus, leaking must be avoided.<br>• The disposing of water with different temperature and nutrient concentration in the ocean is the greatest concern related to OTEC plants operation. |

In CC-OTEC the use of biocides to protect the evaporators and ammonia is safe, and its emission should be too low to detect if the safety regulations are followed (Asian Development Bank, 2014). For its part, the discharge of oily effluents and contaminated water are subject to national or regional regulations as applicable, according to the Bureau Veritas (2018). The entrainment of fish and sea-living mammals into the wide pipes used in OTEC (up to 10 m diameter) is another cause of concern (Dessne et al., 2015). Fish can be attracted to the facility's vicinity; thus, fish population reduction is possible due to impingement, entrainment, and the discharge of biocides (Asian Development Bank, 2014). However, the greatest concern related to the operation of commercial-scale OTEC plants is the disposing of water with different temperatures and nutrient concentrations in the ocean.

A positive effect of OTEC is related to the reduction of dependency on traditional fossil fuel-powered plants. One 10 MW OTEC plant could provide reliable and clean energy for approximately 10 000 people, which means the replacement of 50 000 oil barrels and the elimination of releasing 80 000 ton of carbon dioxide into the atmosphere per year (Lockheed Martin, 2014).

### 2.2.1. Water quality and bio-stimulation

The sustained flow of cold, nutrient-rich, and bacteria-free deep ocean water could cause sea surface temperature anomalies and bio-stimulation because of the artificial fertilization. Deep sea water is also slightly more acidic and charged with $CO_2$, with varying levels of trace metals. One potential impact is the decalcification of corals due to the acidification of the surface waters.

The discharged plume of cold deep sea water into upper layers is subject to different processes due to its higher density (e.g., $\sim 1032\ {}^{\mathrm{kg}}/_{\mathrm{m}^3}$ at 1000 m depth versus $\sim 1028\ {}^{\mathrm{kg}}/_{\mathrm{m}^3}$ near the surface), and its difference in temperature. Thus, it descends to reach its equilibrium depth while experiencing mechanical dispersion and thermohaline diffusion, depending on the currents and the differences in temperature and salinity with the surrounding water (Multon, 2012). This process is known as a negative-buoyant discharge flow.

Regardless of the magnitude of the OTEC plant, analyses of the water are essential to study the plume behavior and impact. According to Multon (2012), the analyses include the following parameters: i) physical parameters (temperature, salinity, suspended materials, turbidity); ii) acidity/carbonates/mineralization; iii) oxidizable organic matter and nutrients; iv) trace metallic elements, and v) biology (chlorophyll a and pheophytin).

Measurements of water quality and bio-stimulation (e.g., nutrient concentration and phytoplankton growth) can be also estimated through ecosystem models coupled with hydrodynamic numerical models, e.g., Wang et al. (2016).

### 2.2.2. Influence of discharge depth

In the surface layer of warm seawater, nutrient levels are low, while the deep cold seawater layer presents higher levels of nutrients. Thus, introducing deep-water nutrients into the surface layer could potentially increase the growth of plankton or algae bloom (Bureau Veritas, 2018). Bio-stimulation caused by deep cold ocean water depends on the discharge depth and the resident time in the mixed layer and in the euphotic zone. The Asian Development Bank (2014) states that most biological activity requires radiation levels of at least 10 % of the sea surface value.

Numerous authors (see Nihous and Vega (1993), Vega (2002), Vega (2012), Grandelli et al. (2012)) agree that small ($\leq 1$ MW) land-based plants water discharge is environmentally acceptable at a depth of 60 m.

Studies carried out in Hawaii suggest that, considering a 100 MW OTEC offshore plant, the ecological effects of nutrient displacements would be negligible if the discharge is at 70 m depth (Dessne et al., 2015). Makai Ocean Engineering Company studied the Hawaiian OTEC plant's seawater discharges. In all numerically modeled cases, no increase in plankton levels occurred in the upper 40 m of the ocean. From 40 to 120 m, OTEC-induced plankton growth was low and within the naturally occurring variability. Thus, they concluded that suitably designed large OTEC plants caused no significant increase in biological growth (Makai Ocean Engineering, 2017).

Studies carried out in Puerto Angel, Mexico, show that nutrient concentration is higher when an OTEC plant is continuously operating; however, there are no significant negative environmental effects due to the oceanographic dynamic in the area and the dilution rate (García Huante, 2015)., However, further detailed studies of the oceanographic dynamic with different CFD software, as well as monitoring and security measures are nedded to obtain a better understanding of these aspects.

Nihous and Vega (1993) recognize that, due to the large effluent discharged in a commercial-size 100 MW OTEC plant, an extensive impact study is necessary. Possible ecological effects depend on whether the currents at the suggested depths would dilute the nutrient concentration before phytoplankton growth takes place. Thus, each site must be individually evaluated with a variety of tools.

## 2.3. CURRENT STATUS OF OTEC TECHNOLOGY

### 2.3.1. Antecedents and future perspective

Nowadays, three international consortiums are already competing for the OTEC market: i) the Naval Group (formerly DCNS) in collaboration with several institutions and other industrial partners; ii) American company Lockheed Martin; and iii) Asia, Japanese and Korean partners (OTEC news, 2013, and Multon, 2012). Table 2 summarizes the documented OTEC demonstration plants around the world.

*Table 2. Summary of the most relevant OTEC demonstration plants around the world. Modified from Ravindran (2000) and* Multon (2012).

| S. No. | Agency | Year, Location | Capacity (kW) | | Cycle | Type of plant |
|---|---|---|---|---|---|---|
| | | | Gross | Net | | |
| **1.** | Claude (France) | 1930,Cuba | 22 | - | Open | Shore based |
| **2.** | Mini OTEC (US) | 1979,Hawaii | 50 | 18 | Closed (Rankine) | Floating |
| **3.** | OTEC-1 (US) | 1981,Hawaii | 1000 | - | Closed (Rankine) | Floating |
| **4.** | Toshiba & TEPC (Japan) | 1981,Nauru | 120 | 30 | Closed (Rankine) | Shore based |
| **5.** | NELHA (US) | 1992,Hawaii | 210 | 103 | Open | Shore based |
| **6.** | Saga University (Japan) | 1984,Saga | 75 | - | Closed (Rankine) | Lab model |
| **7.** | Saga University (Japan) | 1995,Saga | 9 | - | Closed (Uehara) | Lab model |
| **8.** | NIOT, India | 2000,Tuticorin | 1000 | - | Closed (Rankine) | Floating |
| **9.** | Saga University (Japan) | 2003, Imari | 30 | - | Closed (Uehara) | Lab model |
| **10.** | SUPRC (Korea) | 2013, Goseong | 20 | | Closed (Rankine) | Shore based |
| **11.** | NELHA (US) | 2015, Hawaii | 100 | - | Closed (Rankine) | Shore based |

Even though Japan has no suitable coastal sites for OTEC implementation, it has contributed to the development of this technology. In 1981 a CC-OTEC plant built by the Tokyo Electric Power Company started operation. It produced 120 kW of electricity, from which 90 kW supplied the plant itself and the remaining 20 kW supplied a school and other places of the Nauru Island (Khaligh and Onar, 2010). Nauru plant operated for a few weeks, but a typhoon destroyed it.

Nowadays, Saga University in Japan is one of the leaders of the OTEC investigation and its secondary benefits. In 1998, the National Institute of Ocean Technology in India and the Saga University started a project of a 1 MW (gross) OTEC floating plant, in the southeast of Tuticorin, South India, where the ocean depth of 1200 m is available from 40 km off the mainland. It used ammonia as the working fluid and one HDPE 1000 m long pipe of 0.9 m diameter for the cold water intake (Abraham and Abraham, 2002). Unfortunately, the pipeline broke during installation.

The French companies Akuo Energy and Naval Group will construct and install the biggest OTEC platform ever built in Martinique Island. The project is known as *New Energy for Martinique and Overseas* (NEMO) and consists of a 10 MW CC-OTEC offshore floating plant. The nominal installed capacity is 14 MW and net power 10 MW with ammonia as the working fluid, four turbo-generators, and a 6 m diameter and one 1.1 km depth inlet-pipe (OTEC news, 2013). It was meant to be operational in 2020 (Naval Group, 2014). The NEMO project, as well as other OTEC projects conducted by Naval Group, are based on a prototype operating in La Réunion, a French island located in the Indian Ocean (Naval Energies, 2017).

The United States established the Natural Energy Laboratory of Hawaii Authority (NELHA) in 1974, which is one of the world's leading test facilities for OTEC technology. Moreover, the geophysical characteristics of Hawaii and the electricity costs in Hawaii (the highest ones of the United States), makes this region attractive for renewable energy generation (Khaligh and Onar, 2010). In 1979 a small OTEC plant mounted on a barge in Hawaii produced 50 kW of gross power, with a net output of 18 kW and operated for a few months to demonstrate the OTEC concept. This plant was too small to be scaled to commercial size systems (Vega, 1999). NELHA deployed and operated an offshore 1 MW (gross) OTEC plant in 1981. The facility included a 670 m HDPE cold water pipe of 1.1 m diameter, and cold- and warm-water ducting equipment. Only tube exchangers, without turbine, were tested.

To obtain experimental data under prototypical seawater conditions NELHA installed a 210 kW (gross) OC-OTEC pilot plant. This land-based facility in Hawaii successfully operated from 1993 to 1998 (Cunningham et al., 2010). It produced desalinated water using $\sim 10$ % of the steam produced. It had a $1916$ m long cold water pipe of 1 m diameter, which supplied cold water (6 °C) from a depth of 674 m. Warm surface water had an average temperature of 26 °C. The highest production rates achieved were 255 kW gross power, 103 kW net, and 0.4 l/s of desalinated water, which are world records for OTEC (Vega, 1999).

Also in the NELHA, it was recently (2015) inaugurated the world's largest operational CC-OTEC power plant connected to the U.S. grid. It is a 100 kW power land-based plant that supplies approximately 120 Hawaii homes (Figure 5). Makai Ocean Engineering firm operates the plant (Makai Ocean Engineering, 2017).

Makai Company plans to put up an offshore 100 MW facility (Figure 5). Nevertheless, the growth of the world's OTEC sector has limitations due to high capital costs and financing problems. Thus, the cost reduction of plants is necessary to attract investments. For this purpose, the technology must be first validated with a 5 – 10 MW offshore plant that works for several years (Patel, 2015).

In 2013, South Korea started operation of an on-shore CC-OTEC onshore plant in the Goseong region, with Difluoromethane (R32) as the working fluid. It is a 20 kW (gross) pilot plant which is a down-scaled model of 1 MW OTEC plant. A public demonstration was held in November 2013 after multiple successful runs with a safe operation routine. Based on this success, the team is planning to build and operate a 200 kW OTEC plant and to achieve the final goal of designing a 1 MW OTEC plant (OTEC news, 2014). Nevertheless, the plant only operates in summer because of the temperature difference required: in winter, the surface seawater temperature is about 12 °C, thus additional heat sources are used to perform the tests smoothly (Ho-saeng Lee et al., 2016).

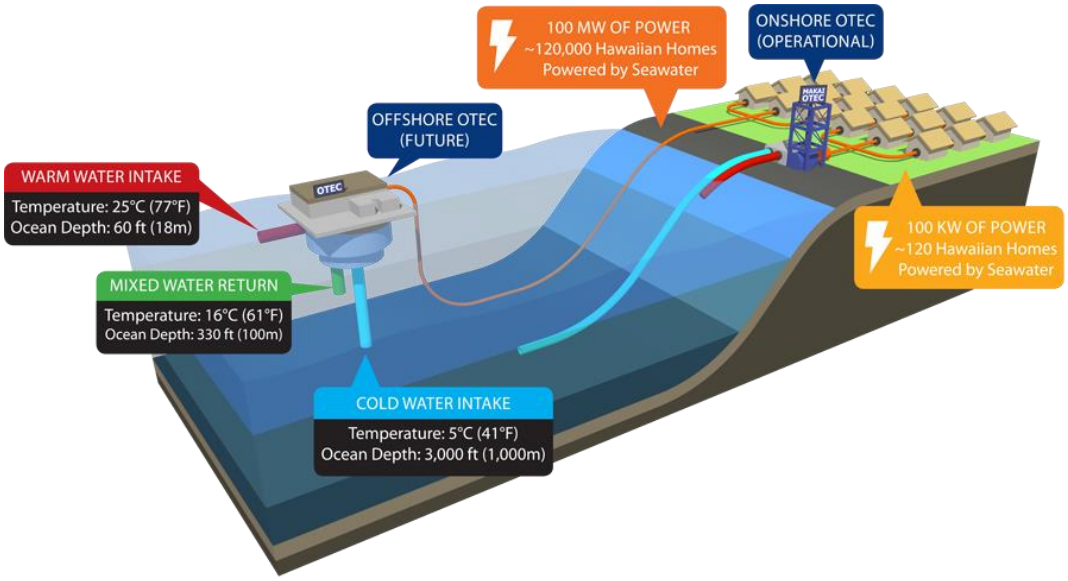


*Figure 5. Makai OTEC projects in Hawaii, U.S. Taken from (Makai Ocean Engineering, 2017b).*

The development of the OTEC plant in South Korea has encouraged the research in different items such as agriculture, aquaculture, fresh water production, salt production and fertilizers, deep water utilization for rice production, etc. This has led to the creation of the industrial complex known as the Seawater Utilization Plant Research Center of Korea (SUPRC), which has boosted the social development of the community (García Huante, 2015). Table 2 summarizes the documented OTEC demonstration plants around the world.

### 2.3.2. Technical readiness

The state of technical readiness of key components of OTEC technology was gathered and summarized during the workshop held by the Coastal Response Research Center in collaboration with the Oceanic and Atmospheric Administration, Office of Response and Restoration and the University of New Hampshire in 2009 (Cunningham et al., 2010). Pilot-scale plants of less than 1 MW have successfully generated energy, even though economic and technical feasibility limitations discourage investment in this field. The qualitative analysis suggested that a < 10 MW floating, CC-OTEC facility is technically feasible using current development. Nevertheless, the technical readiness and scalability of a > 100 MW facility is less clear.

There are seven components which are potentially limiting factors for the success of OTEC: 1) platforms, 2) platform mooring system, 3) platform/pipe interface, 4) heat exchangers, 5) pumps and turbines, 6) power cable, and 7) cold water pipe (CWP). Furthermore, a necessary step in the commercialization and development of OTEC is to understand the challenges associated with a $\geq 100\ \mathrm{MW}$ facility, by the construction and operation of a $\leq 100\ \mathrm{MW}$ facility.

For large OTEC systems, the modular design of some components is possible. For example, for a 10 MW facility, two radial flow turbines each rated at 7 – 8 MW gross power could be used. Increasing the number of turbines improves reliability and net power production. However, the participants of the workshop concluded that unlike the other major components, the power cable, cold water pipe, and the platform/pipe interface present fabrication are not scalable using modular designs (several smaller units) for $\geq 100\ \mathrm{MW}$ facilities, and therefore, further research, modeling, and testing is required.

The CWP is considered to be the single biggest challenge of OTEC because of the enormous dimensions for a commercial size plant. A discussion about the CWP of OTEC systems is given below.

*Cold Water Pipe (CWP)*

CWP and all the water ducting subsystems are major parts of OTEC systems because they handle large water volumes (e.g., $200\ \mathrm{m^3/s}$ for the CWP, and $400\ \mathrm{m^3/s}$ of warm water in a 100 MW plant). CWP represents an important engineering challenge, especially because of the length required to bring cold water to the surface from the ocean depths (depths up to 1000 m), and its large diameter (up to 10 m for a 100 MW plant).

The selection of the CWP diameter depends on the hydraulic head to overcome pipe drag and the drag coefficient given by the CWP material. If this hydraulic head is a small fraction of the total head for the baseline design, the diameter is larger than necessary. The design may also include the ability to detach the pipe from the platform before a large storm event.

According to Lockheed Martin Corporation, for a 100 MW offshore OTEC facility, the CWP is $\sim 1\,000$ m long and 10 m diameter. They considered an advanced carbon-fiber composite as the most suitable material. They selected this material over fiberglass, steel, and HDPE by comparing its behavior under the considered requirements: external pressure, wave-induced cyclic strain motion, wave-induced axial buckling motion, streaming and clump axial weight, platform rotation, and low-cost manufactural configuration.

To satisfy the structural requirements in harsh offshore conditions, the Lockheed Martin team came up with a hollow-core one-piece CWP, which is built directly down from the floating-OTEC platform (OTEC news, 2012). This approach maximizes durability and reliability during operation. It also eliminates major deployment risks and enables affordable fabrication of large diameter pipes. Miller et al. (2012) validated the key elements of the process for a 2 m diameter CWP.

The fabrication methods for $\sim 7$ m diameter CWP are currently available and can be scaled to a $\sim 10$ m diameter CWP. The pipe can be deployed in situ as one whole piece (fabricated onshore) or with a stepwise fabrication. Both methods have been successfully demonstrated on a 2 m diameter scale. The construction and deployment of a $\geq 10$ MWe CWP have not been attempted. Furthermore, the transportation, deployment, and decoupling of a single piece pipe is difficult and would require rowing it from shore. However, segmented pipes are easier to deploy but they present failure risk at the many joints required (Cunningham et al., 2010).

The deployment and operation of CWP in large OTEC facilities is the number one challenge for OTEC development. Research on alternative designs for CWP, its production, delivery, the cost-effectiveness of its materials, equipment, and installation should be conducted.

## 2.4. NUMERICAL MODELING IN OTEC SYSTEMS

Several numerical tools have been developed for the simulation of OTEC systems, which are different from each other depending on the way they address and solve the problem, their complexity, their application's limits, and the variables that they consider. In the following, an overview is presented, regarding some examples of software that has been developed to simulate OTEC components.

Some of the most important OTEC research and development institutions in the world have published their contributions related to the numerical simulation of OTEC thermal discharge in the ocean, its impact on water quality, and bio-stimulation. The American Makai Ocean Engineering, Inc., in Hawaiian, the Korean Research Institute of Ships and Ocean Engineering (KRISO), and the University of Tokyo, Japan, are the pioneers on this topic.

*United States*

In a technical report published by U. S. Department of Energy, Grandelli et al. (2012) described the numerical modeling work by Makai Ocean Engineering, Inc. in Hawaiian waters. They simulated the dilution, circulation, and the biochemical effects of the nutrient-enhanced seawater plumes that are discharged by one or several 100 MW OTEC plants. Environmental Fluid Dynamics Code (EFDC) was implemented and calibrated with the boundary conditions from a surrounding Hawaii Regional Ocean Model (ROM), which provided tides, basin-scale circulation, mesoscale variability, and atmospheric forcing. EFDC is approved by the USA Environmental Protection Agency (EPA)

They modeled different discharge configurations at 70 m depth or more and found that no perturbation occurred in the upper 40 m of the ocean's surface. However, the picoplankton's response in the 110-70 m depth layer had a $\sim 10 - 25\ \%$ increase, which is within the naturally occurring variability. The nanoplankton's response was negligible and the microplankton's enhanced productivity was small. At the time, it was the most sophisticated and realistic plume model developed for OTEC. The model does not calculate the higher-order trophic levers but the results can be extended for this purpose. Finally, Grandelli et al. (2012) suggested that the developed tool could be useful for OTEC regulators and designers.

*Korea*

To illustrate the challenging nature of OTEC environmental studies, Kim and Kim (2014), and Mun et al. (2014) performed numerical modeling of the mixing and dispersion characteristics of thermal discharge in coastal waters in Kosrae, Micronesia. They designed the numerical experiment to understand the impact of the density-driven current followed by the thermal effluent; employing the momentum, continuity, temperature, salinity, and density governing equations.

Kim and Kim (2014) utilized the Finite-Volume Community Ocean Model, FVCOM, with a 3D unstructured-grid, free-surface, and primitive equation. Afterwards, they adopted the CFD software *Flow-3D* with various turbulence models to provide a more robust assessment of the near-field plume dynamics in the vicinity of the discharge. Likewise, Mun et al. (2014) considered field observation and three-dimensional numerical modeling with FVCOM and HYCOM (Hybrid Coordinate Ocean Model) to successfully reproduce the plume behavior.

Both studies concluded that the FVM method is a useful tool for the understanding of the plume behavior of thermal effluent from an OTEC outfall and determining the location that best minimizes its impact upon the coastal waters of the study area, i.e., Kosrae, Micronesia. However, they recommend to conduct further three-dimensional studies and to develop better numerical models.

On their part, Lee et al. (2016) studied the thermal dispersion characteristics of coastal waters of Tarawa, Kiribati. They used field observations with CTD data and Computational Fluid Dynamics (CFD) modeling. The basic model of the numerical simulations was a three-dimensional, free-surface, primitive equation EFDC (Environmental Fluid Dynamic Code) ocean model. HYCOM was also used to specify temperature and salinity initial conditions. Finally, more detailed assessments were obtained with CFD software Flow-3D, which provided results in good agreement with the

observations. By comparing the spreading pattern of thermal effluent according to the location of the OTEC outfall, they found that the main direction of the spreading was along the deep area because of the density of the thermal effluent, i.e., higher density than the background water.

*Japan*

Wang et al. (2016) and Wang and Tabeta (2017) performed numerical simulations of the behavior of discharged water from the OTEC plant for 60 days. The hydrodynamic processes were calculated with MEC-NEST software and a nested-grid system to simulate a 100 MW offshore plant, and a 1 MW onshore plant in Kume Island, Japan. They combined the hydrodynamic model with a low-trophic ecosystem model to calculate the behavior of nutrients concentration, i.e., consumption and production through chemical and biological processes. They observed that the discharged water of the 100 MW offshore plant descended as being mixed with surrounding seawater and horizontally spread at the depth where the density of mixed water became equal to that of the surrounding seawater. For the 1 MW onshore plant, the impacted area was limited in the vicinity of the discharge point due to the smaller flow rate.

Additionally, the changes in nutrient ($NO_3$) and large phytoplankton (PL) were measured. They observed that phytoplankton concentration decreased near the facility since its concentration is low in the discharged water, and the growth rate is small in the low temperature. Whereas in the northeast area of the plant, phytoplankton and nutrient concentration increased. However, they concluded that those changes in temperature and water quality were not significant due to the rapid diffusion in the ocean. Ocean currents were not taken into account and ecosystem parameters need to be adjusted as the used ones correspond to a different latitude (Wang and Tabeta, 2017). Finally, they predicted that the changes in water quality become smaller as the discharge point is set at a deeper location.

## 2.5. SUMMARY AND IMPLICATIONS FOR THE PH.D STUDY

OTEC uses the temperature gradient between warm surface seawater and cold deep seawater. OTEC technology dates from the $19^{th}$ century, and numerous attempts have been carried out to demonstrate its capability to produce clean energy. Some attempts have been successful, while others have failed due to economic, technological, and environmental factors.

The economic analysis has determined that to make OTEC economically viable for commercial purposes, the capacity of one OTEC facility must approach 100 MW. A facility of such size represents great technological challenges related to the design, construction, and installation of major components such as CWP, platforms, CWP/platform interphase, heat exchangers, pumps, turbines, and power cable. The CWP is one of the biggest OTEC challenges because of the enormous dimensions needed for a commercial-size plant. Nevertheless, current technology and current research in OTEC systems have been developed for much smaller scales ($< 1$ MW). Therefore, further research, modeling, and testing are required. Many researchers have concluded that OTEC

technology is only viable if it comes together with the harnessing of its byproducts, such as fresh water and nutrient-rich cold water.

Another limitation of OTEC implementation in the world is the location of suitable places, with the appropriate conditions of the thermal resource at reasonable distances from the shore and depths. Mexico's geographical position makes it a potential site for the exploitation of the thermal gradient source.

One of the major concerns of deploying large plants is the environmental impact, caused mainly because of the continuous discharge of large amounts of water in the ocean at different temperatures and with different nutrient concentration than the medium. Even though different authors suggest a specific discharge depth, each project must be individually studied with its particular conditions, and testing different configurations before taking any design decision. Studies have shown that little environmental damage is caused by the operation of an OTEC facility when it is carefully designed. However, the environmental impact is less clear when it comes to large OTEC facilities that would mobilize great water quantities and would use large pipes (up to 10 m diameter).

Research centers such as the American Makai Ocean Engineering, Inc., in Hawaiian, the Korean Research Institute of Ships and Ocean Engineering (KRISO), and the University of Tokyo, Japan, have explored OTEC thermal discharge with different numerical tools. Nevertheless, they all coincide that better tools need to be developed and tested for more accurate results. Furthermore, expensive licenses and closed-source of some numerical tools can discourage their use.

One of the medullary objectives of this thesis is to develop a numerical model capable of dealing with the multiphase-fluid flow, thermal interactions and wave/current motions in the sea. The model is able to simulate thermal water discharge of an OTEC facility after used in the energy conversion process, providing another approach to the previous studies with different numerical tools. The development, validation, and implementation of this tool will also be valuable for further applications such as the discharge of warm water to the ocean from nuclear (or other power) plants. Furthermore, it will be capable of being improved and extended in the future because of its open-source code, representing a free alternative to the preexisting software.

# 3

# CONCEPTUAL DESIGN OF AN OTEC PLANT IN MEXICO

# 3. CONCEPTUAL DESIGN OF AN OTEC PLANT IN MEXICO

This section presents the conceptual design of a hypothetical OTEC plant. Hypothetically, this plant will operate in a specific location in Mexico, which meets the necessary thermal resource within an acceptable distance from shore. This pre-dimensioning do not represent a detailed design of an OTEC plant, but only provides the general parameters that will be later used by/for the numerical model.

## 3.1. SITE SELECTION AND GENERAL PARAMETERS

### 3.1.1. Site selection

The Thermal Gradient group of the Ocean Energy Innovation Center of Mexico (CEMIE Océano) gathered the following aspects to be taken into account to select the possible sites of interest in Mexico (García Huante et al., 2018):

i. The existence of temperature difference of at least 20 °C between the ocean surface water and the water at depths of 500 and 1000 m. The temperature gradient variations must be studied for each season of the year.
ii. Cold water accessibility. It is convenient that horizontal distance between the shoreline and the bathymetric lines of 500 and 1000 m does not exceed 10 km. This distance obey to economic and technical reasons: in the case of a floating plant, it is important in terms of the length of the power cables and the desalinated water hose; for onshore plants, the cold water pipe length is determinant for the capital cost (see Vega, 1992, and Vega, 2010).
iii. The existence of natural protected areas. Some sites of interest are within natural protected areas, but they have specific allowed activities that could allow the installation and operation of an OTEC facility.
iv. Population, number of houses. Some sites of interest are practically uninhabited, and the installation of an OTEC plant would not be justifiable.

In the characterization of each site of interest, additional factors are to be considered, such as its altitude, oceanographic processes, currents, sediments, distance to the nearest connection to the electrical grid, and economic indicators. However, these factors are not considered to be decisive for a preliminary site selection. The most relevant locations in Mexico that gather the minimum characteristics to consider the possibility of placing an OTEC facility are:

a) In the Pacific Ocean coast: San Agustin Huatulco, Puerto Angel and El Coyote in Oaxaca; Nuxco in Guerrero; Colola in Michoacan; Yelapa in Jalisco; Cabo San Lucas and Diamante Cabo San Lucas and Cabo Pulmo in Baja California Sur.
b) In the Mexican Gulf coast: no site was found that simultaneously meets the minimum temperature gradient and the maximum distance between the shoreline and the bathymetric lines of 500 and 1000 m.

After comparing different locations, the locality of Puerto Angel, Oaxaca, was chosen as the most suitable place to deploy an OTEC plant and demonstrate its operation in Mexico (Table 3, Figure 6).

*Table 3. Characteristics of Puerto Angel, Oaxaca.*

| | |
|---|---|
| Coordinates | 15° 40' 4.129'' N, 96° 29' 29.132'' O |
| Altitude | 20 m |
| Horizontal distance to the bathymetric line of 500 m | 2.59 km |
| Horizontal distance to the bathymetric line of 1000 m | 7.74 km |
| Maximum temperature $T$ at the surface | 31.14 °C |
| Minimum temperature $T$ at the surface | 27.63 °C |
| Average temperature $T$ at 500 m depth | 7.75 °C |
| Average temperature $T$ at 1000 m depth | 4.6 °C |
| Average temperature difference $\Delta T$ at 500 m depth | 20.5 °C |
| Average temperature difference $\Delta T$ at 1000 m depth | 24.9 °C |
| Natural protected areas | it is not within any natural protected area |
| Distance to the connection to the electrical grid | 8.13 km |
| Houses | 906 |
| Population | 2645 |


*Figure 6. Location of Puerto Ángel, Oaxaca, Mexico.*

### 3.1.2. Plant capacity

The plant capacity, in MW, is estimated considering the available data of annual energy consumption in Mexico (Table 4). To obtain the electricity consumption per home, the residential consumption is divided among the total homes with electrical energy in México, as shown in Table 5.

*Table 4. Annual electricity consumption per sector in Mexico* (Energética, 2010).

| Sector | Annual electricity consumption (PJ) |
|---|---|
| Residential | 203.747 |
| Commerce | 79.05 |
| Public | 32.367 |
| Transport | 4.081 |
| Farming | 36.243 |
| Industrial | 539.972 |
| **Total consumption (PJ)** | **895.46** |

*Table 5. Estimation of the electricity consumption per home in Mexico.*

| | |
|---|---|
| Homes with electricity | 31 509 840 |
| Annual electricity consumption per home (PJ/home) | 6.46614E-06 |
| Annual electricity consumption per home (kWh/home) | 1 796.150 |
| Monthly electricity consumption per home (kWh/home) | 149.7 |
| Power consumption per home (W) | 207.9 |

Next, the net power (net capacity) required for the OTEC plant is obtained considering 906 houses in Puerto Angel.

$$P_{net} = \frac{(906) * \left(1796.15 \text{ kWh}/\text{year}\right)}{(365 \text{ d})(24 \text{ h})}$$

$$P_{net} = 185.8 \text{ kW}$$

The OTEC plant net capacity is rounded up to 190 kW. Gross capacity $P_{gross}$ is obtained considering in-house or parasitic electrical loads $P_{loss}$ which represent about 30 to 40 % of $P_{gross}$, so that the exportable power $P_{net}$ is about 60 to 70 % of $P_{gross}$ (Vega, 2012).

$$P_{gross} = 190 \text{ kW}/0.6$$

$$P_{gross} = 316.7 \text{ kW}$$

Thus, the gross capacity of the plant is rounded up to 320 kW. As stated in the theoretical framework, the economic analysis has determined that, to make OTEC economically viable for commercial purposes, the capacity of one OTEC facility must approach 100 MW. Therefore, a 320 kW OTEC plant is considered experimental or demonstrative. This technology has not been demonstrated in Mexico yet; thus, the design, construction, and operation of such a plant, set the foundations for a commercial-size OTEC plant.

Vega (2012) reports that a change of 1 °C in $\Delta T$ produce relative fluctuations of about 10 % in $P_{gross}$ for typical values of 20 °C of thermal resource. In terms of the net power $P_{net}$, this variation is translated into 15 % fluctuation when considering 30 % of $P_{gross}$ for in-plant power consumption. Limitations in the manufacturing and installation of major plant components, as well as capital

availability, determine the plant size. However, state-of-art shows that fabrication methods for a $\leq 10\ \mathrm{MW}$ plant and $\sim 7\ \mathrm{m}$ diameter pipeline are currently available. Thus, the pipeline for a 320 kW plant does not imply an important technological challenge.

### 3.1.3. Cold and warm water flow rate

According to Vega (2012), OTEC general design parameters establish that a cold water rate ($Q_{cw}$) of $2.6\ \mathrm{m^3/s}$ is required per $\mathrm{MW_{net}}$. Therefore, for $190\ \mathrm{kW_{net}}$, $Q_{cw} = 0.5\ \mathrm{m^3/s}$. On the other hand, warm water flow rate ($Q_{ww}$) is about twice the cold water rate, in this case, $Q_{ww} = 1\ \mathrm{m^3/s}$.

## 3.2. DIMENSIONING AND MATERIALS SELECTION

Table 6 summarizes the elementary parameters for the preliminary design of an OTEC plant in Puerto Angel.

*Table 6. General design parameter for an OTEC plant in Puerto Angel, Mexico.*

| **Design parameter** | **Value** |
|---|---|
| Gross power ($\mathrm{kW}$) | 320 |
| Net power ($\mathrm{kW}$) | 190 |
| Average temperature difference ($°C$) (500 m) | 20.5 |
| Average temperature difference ($°C$) (1000 m) | 24.9 |
| Cold water flow ($\mathrm{m^3/s}$) | 0.5 |
| Warm water flow ($\mathrm{m^3/s}$) | 1 |

The dimensioning and the materials are chosen comparing various projects, some of which have been developed and tested, and some which are theoretical conceptions of larger plants (see Nihous and Vega, 1993; Nihous and Vega, 1994; Nihous and Vega, 1996; Vega, 2000 and Vega, 2012).

### 3.2.1. Piping

CWP and all the water ducting subsystems are major parts of OTEC systems because of the large water volume that they handle (e.g., $200\ \mathrm{m^3/s}$ for the CWP, and $400\ \mathrm{m^3/s}$ of warm water feed in a 100 MW plant). To keep the water pumping losses at about 20 to 30 % of the gross power, an average flow speed of fewer than 2m/s is needed (Vega, 2002).

Piping systems in OTEC plants often use Fiberglass Reinforced Plastic pipes (FRP) and High-Density Polyethylene pipes (HDPE). Small plants (up to 2 MW), usually use HDPE pipes or a combination of FRP and HDPE, e.g., FRP in the nearshore portion and HDPE in the rest of the pipe. Larger plants ($\sim 100$ MW) are projected with a 10 m diameter CWP of a FRP sandwich construction (two FRP layers separated by a layer of syntactic foam).

FRP is chosen because of its competitive cost and superior life expectancy under corrosive tropical environment, while HDPE is preferred because of its multiple advantages such as high toughness, non-toxic, corrosion resistance, rust resistance, long service life, good environmental suitability, various pressure ratings (up to $8\ \mathrm{kN/m^2}$), etc. HDPE construction process is also safe, cost-effective and simple. HDPE pipes are available in several diameters ranging from 16 mm to 3500 mm. Nevertheless, FRP has superior material properties, but the unit pricing is higher than HDPE.

### *Cold water pipe (CWP)*

CWP represents an important engineering challenge because of the required length to bring cold water from the depths of the ocean (depths up to 1000 m) to the surface, and its large diameter (up to 10 m for a 100 MW plant). Table 7 summarizes the characteristics of the CWP of different OTEC projects.

*Table 7. Cold water pipe (CWP) characteristics for 10, 10.8 and 0.21 MW OTEC plants.*

| CWP | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Plant Capacity (gross) (MW)** | **Type** | **Materials** | **Length (m)** | **Depth of entrance (m)** | **Diam (m)** | **Flow rate ($m^3/s$)** | **Velocity (m/s)** | **Temp (°C)** |
| 10 | Offshore plantship CC-OTEC | FRP | 1000 | 1000 | 3.9 | 27.7 | 2.32 | 4.5 |
| 1.8 | Land-Based OC-OTEC | Nearshore portion (120 m long) of FRP. The rest of HDPE | 2590 | 1000 | 1.6 | 3.20 | 1.59 | 4 |
| 0.21 | Land-Based OC-OTEC | Nearshore portion of FRP. The rest of HDPE | 2040 | 670 | 1 | 0.42 | 0.53 | 6.1 |

### *Warm water pipe (WWP)*

WWP length is usually small compared against CWP; however, its diameter is larger because the flow rate is about twice the cold water flow rate. Table 8 shows the WWP characteristics for 10, 10.8 and 0.21 MW OTEC plants.

*Table 8. Warm water pipe (WWP) characteristics for 10, 10.8 and 0.21 MW OTEC plants.*

| WWP | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Plant Capacity (gross) (MW)** | **Type** | **Materials** | **Length (m)** | **Depth (m)** | **Diam (m)** | **Flow rate ($m^3/s$)** | **Velocity (m/s)** | **Temp (°C)** |
| 10 | Offshore plantship CC-OTEC | FRP | - | 20 | 2x4.6 | 52.8 | 1.60 | 26 |
| 1.8 | Land-Based OC-OTEC | FRP | 120 | 25 | 2.5 | 6.20 | 1.26 | 26 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0.21 | Land-Based OC-OTEC | Nearshore portion of FRP. The rest of HDPE | - | - | 0.7 | 0.62 | 1.61 | 26 |

*Discharge pipes*

There are three discharge pipes in OC-OTEC systems: warm water discharge pipe, cold water discharge pipe and mixed water discharge pipe (MWDP). When a second stage process for water desalination exists, a single MWDP is preferred over two separated warm and cold seawater discharge pipes. For a two-stages plant, the residual temperature difference between the two discharge water streams drops from $10\ ℃$ to $5\ ℃$ (Vega, 2000). Table 9 shows MWDP for 10 and 10.8 MW OTEC plants.

*Table 9. Mixed water discharge pipe (MWDP) characteristics for 10 and 10.8 MW OTEC plants*

| MWDP | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Plant Capacity (gross) (MW)** | **Type** | **Materials** | **Length (m)** | **Depth (m)** | **Diam (m)** | **Flow rate ($m^3/s$)** | **Velocity (m/s)** |
| 10 | Offshore plantship CC-OTEC | FRP | 60 | 60 | 2x5.5 | 80 | 1.68 |
| 1.8 | Land-Based OC-OTEC | FRP | 190 | 60 | 3 | 9.2 | 1.30 |

*Puerto Angel OTEC plant*

In the conceptual design of the 320 kW OTEC plant there are two alternatives: i) A land-based OC-OTEC plant that uses the thermal gradient between the surface water and the water at 500 m depth, and ii) an offshore plantship CC-OTEC that uses the thermal gradient between the surface water and the water at 1000 m depth.

A land-Based OC-OTEC plant will be considered for the pre-dimensioning of the rest of the components. The plant will include a 1 m diameter CWP, a 1.2 m diameter WWP, and a 1.2 m diameter MWDP as shown in Table 10.

The plant configuration and design specifications are be based on the $1.8\ MW$ OTEC plant design presented by Vega (2000), which is a land-based two-stage OC-OTEC plant. The cycle of this plant is described by Vega (2000) as follows:

> "Cold seawater and warm seawater is supplied to the OTEC plant by a single pipe. Warm water enters the low-pressure evaporation chamber through an array of spouts where flashing occurs. In the first stage, some water is vaporized whereas the remaining flow is drained to the discharge pool. The steam from the evaporator drives a turbine before entering a surface condenser. The condenser is cooled by deep seawater and produces fresh water condensate. The second stage fresh water production unit consists of additional evaporator and surface condenser downstream of the first stage power system, which significantly increases the amount of fresh water produced. Mixed

effluent water is disposed by gravity back to the sea via one single pipe. A disposal depth is of $60$ m to minimize environmental impact."

*Table 10. CWP, WWP and MWDP characteristics for the* $320\ kW$ *OTEC plant in Puerto Angel, Mexico*

| **CWP** | | | | | | |
|---|---|---|---|---|---|---|
| **Materials** | **Length (m)** | **Depth (m)** | **Diam (m)** | **Flow rate ($m^3/s$)** | **Velocity (m/s)** | **Temp (°C)** |
| Nearshore portion (200 m long) of FRP. The rest of HDPE | 2700 | 500 | 1 | 0.5 | 0.64 | 9.2 |
| **WWP** | | | | | | |
| **Materials** | **Length (m)** | **Depth (m)** | **Diam (m)** | **Flow rate ($m^3/s$)** | **Velocity (m/s)** | **Temp (°C)** |
| FRP | 200 | 10 | 1.2 | 1 | 0.88 | 29.7 |
| **MWDP** | | | | | | |
| **Materials** | **Length (m)** | **Depth (m)** | **Diam (m)** | **Flow rate ($m^3/s$)** | **Velocity (m/s)** | **Temp (°C)** |
| FRP | 300 | 60 | 1.2 | 1.5 | 1.33 | 12 |

### 3.2.2. Heat exchangers

The evaporator and the condenser constitute the heat exchangers. The vacuum structure is the building that houses the heat exchangers, the vertical axis turbine, and diffuser. The evaporator consists of a carbon-steel cylindrical vessel that includes a warm water inlet, a warm water outlet, and a steam outlet. The cylindrical vessel is 2.5 m diameter and 3 m high, it houses a manifold with 22 spouts and produces 0.005 kg/s of steam (0.5 % of WW flow). It works at a pressure of 2600 Pa. A baffle plate divides the vessel into pre-deaeration and evaporation chambers. This plate houses vertical spouts to allow water flashing in the evaporator.

Approximately 92 % of the steam condenses into fresh water in the main unit (0.005 kg/s for this case), and residual steam liquefies in the vent condenser; the baseline condenser system uses tube & shell surface condensers design, e.g., a Toshiba condenser. Non-condensable gases can be removed by a compressor train, which draws them from various locations in the power block.

The vacuum pump system evacuates the vacuum chamber and maintains the operating vacuum by removing the small amount of residual steam from the condenser and the non-condensable gases (nitrogen and oxygen).

### 3.2.3. Turbine-generator

The steam turbine is a single rotor radial flow machine with a 3 m diameter rotor and 2.1 m diameter axial outlet, such as the one used in the 210 kW OTEC plant in Hawaii (Nihous and Vega, 1996) or similar. The volumetric flow capacity dictates the dimensions of a turbine rotor, which impose a bound on the maximum power that may be generated due to the size limitations of existing turbo-machinery. According to (Vega, 2000), the specific volume, $\upsilon\ [\mathrm{m}^3/\mathrm{kg}]$, of steam exiting the turbine will be between 2.5 to 5 times larger than the specific volume in conventional combustion or nuclear

power stations; thus, the turbine area required to pass a unit mass flow of steam must be increased accordingly.

## 3.3. SUMMARY OF THE PROPOSED OTEC PLANT

Table 11 summarizes the design specifications for the 320 kW hypothetical OC-OTEC plant in Puerto Angel, Mexico.

*Table 11. Design specifications for the 320 kW OC-OTEC plant in Puerto Angel, Mexico*

| **Vacuum structure** | |
|---|---|
| Material | Reinforced and prestressed (post-tensioned) concrete |
| Main structure | 11 m diameter<br>15 m height<br>20 m overall height including turbine generator |
| Ground Level | 8 m above sea level |
| **OC-OTEC process** | |
| Warm sea water | $1\ m^3/s$<br>29.7 °C<br>1.2 m diameter pipeline<br>inlet depth 10 m<br>200 m length |
| Cold sea water | $0.5\ m^3/s$; 9.2 °C<br>1 m diameter pipeline<br>inlet depth 500 m<br>2700 m length |
| Mixed water | $1.5\ m^3/s$; 12 °C<br>1.2 m diameter pipeline<br>outlet depth 60 m<br>300 m length |
| Evaporator | 2.5 m diameter<br>3 m high<br>manifold with 22 spouts<br>5 l/s of steam (0.5 % of WW flow)<br>Pressure 2600 Pa;<br>WW discharge temperature 25.7 °C |
| Condenser | tube & shell surface condensers |
| Turbine | 3 m diameter rotor<br>radial inlet<br>2.1 m diameter axial outlet. |
| **Power Budget** | |
| **Generator terminals** | **320 kW** |
| Total parasitics | 130 kW |
| **Net power** | **190 kW** |

# 4

# COMPUTATIONAL FLUID DYNAMICS

# 4. COMPUTATIONAL FLUID DYNAMICS

In this section, a general background of thermodynamic numerical modeling of fluids, the conservation principles of the governing equations and their solutions are presented.

The numerical modeling is used to solve a mathematic model when the exact or analytical solutions are too slow or simply not available/able to provide an answer. This method consists of formulating a mathematical model in such a way that it can be solved by arithmetic operations and the programming of algorithms. Computational fluid dynamics (CFD) uses numerical modeling to simulate the performance of fluids under a stated condition. As a consequence of the development of the digital computer, modern CFD techniques improved its results since its beginnings in the 1950s.

The thermodynamic modeling of fluids allows to predict and to correlate thermodynamic properties and phase equilibria, which is essential for the simulation of chemical and physical processes, and the development and optimization of engineering systems. Several numerical tools have been developed for this purpose. These tools are different from each other depending on the way they address and solve the problem, their complexity, their applications, and the variables that they consider.

## 4.1. NUMERICAL MESH

Among other things, the solution of a numerical model relies on the correct implementation of the numerical mesh for the domain discretization, which consists of the process of transferring a continuous domain into discrete counterparts. The numerical mesh influences the domain discretization, equation discretization, and the formulation of the source code. It also affects the precision, the speed, and the stability of the solution. There are two types of meshes, structured and unstructured (Figure 7). In structured meshes, the cells are labeled with indexes in the directions of the coordinate axis; while unstructured meshes may not have apparent direction.

Numerical modeling and CFD are based on three main discretization methods for the solution of partial differential equations (PDE):

a) Finite difference methods (FDM).

FDM are distinguished because their formulation is relatively easy (compared to other methods). These methods are typically based on the application of the Fourier series, Taylor series expansions, and the Legendre polynomial. Some limitations of FDM methods are found since structured meshes are needed for their implementation. For multidimensional problems, it is necessary to transform curved meshes into structured meshes to work within an orthogonal Cartesian coordinate system.

b) Finite element methods (FEM)

FEM methods are based on the separation of the computational domain into a smaller number of regions (cells); thus, needing a mathematical rigor for their specification and the solution of the formulations. Like the FDM, the solution of the PDE is approximated in the computational domain,

and each cell is a mesh element composed of nodes and vertices. FEM methods have the advantage of not needing a transformation of coordinates of an unstructured mesh into a Cartesian coordinate mesh (Chung T. J., 2010).

c) Finite volume methods (FVM).

For FVM, the formulations could be based either on FDM or FEM. This approach works by integrating the PDE to be solved. The surface integrals of normal fluxes in FVM guarantee the conservation properties through the domain, which allows the usage of complex geometries, and unstructured meshes without a coordinate transformation. In that sense, FVM are more powerful, and they offer the possibility for working with refined meshes if necessary; i.e., manipulating the cells sizes along with the mesh instead of having a very refined/coarse mesh through the whole domain. In this case, the mesh is composed of control volumes or cells with the PDE's solved for each cell.

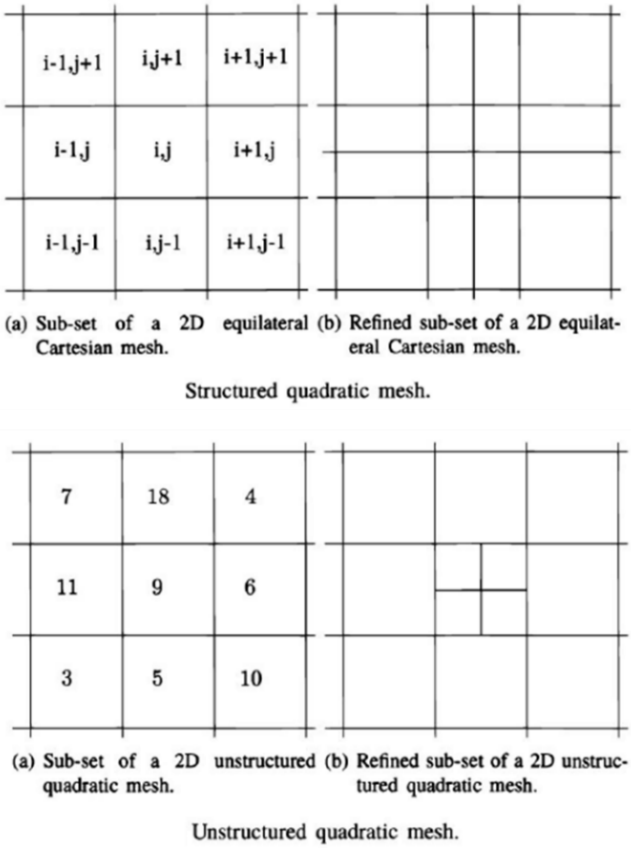


*Figure 7. Structured and unstructured meshes. From Tomislav et al. (2014)*

Since the creation of FDM and FEM for stress analysis, there have been earlier applications in the fields of fluid dynamics and heat transfer (Chung T. J., 2010). However, current CDF mainly uses FEM and FVM because they handle complicated geometries.

## 4.2. CONSERVATION PRINCIPLES

Three fundamental laws yield the governing equations used in a CFD study: i) conservation of mass (the continuity equation), ii) conservation of momentum (Newton's second law), and iii) conservation of energy (first law of thermodynamics). Additionally, the conservation of scalar quantities law supports the calculation when there are multiple fluid-phases involved.

In fluid mechanics, it is convenient to consider the flow within a certain spatial region called *control volume* (CV) instead of a given substance quantity or *control mass* (CM). Mass conservation law relates the changing rate of one extensive property in a given control mass. For mass ($m$) conservation, the equation can be written as in Equation 2, which means that mass can be neither created nor destroyed:

$$\frac{dm}{dt} = 0 \qquad \text{Equation 2}$$

On the other hand, Newton's second law of motion leads to the momentum conservation equation as depicted in Equation 3:

$$\frac{d(mU)}{dt} = \sum f \qquad \text{Equation 3}$$

where $t$ is time, $U$ the velocity, and $f$ the forces acting on the control mass.

Mass and Momentum are both extensive properties. If $\phi$ is any conserved intensive property per unit of mass, the corresponding extensive property $\Phi$ can be expressed as:

$$\Phi = \int_{\Omega_{CM}} \rho\phi \, d\Omega \qquad \text{Equation 4}$$

where $\Omega_{CM}$ is the volume occupied by the CM and $\rho$ the density.

### 4.2.1. Control volume equation

Using the definition of Equation 4, the "control volume equation" (also known as Reynold's transport theorem) is defined. The control volume equation states that the changing rate of the property $\Phi$ in the control mass, is equal to the rate of change of the property within the control volume, plus the net flux of it through the CV boundary, which is caused by the fluid-motion relative to the CV boundary (Ferziger and Peric, 2002). Thus, for a given fixed CV, the control volume equation can be expressed as:

$$\frac{d}{dt}\int_{\Omega_{CM}} \rho\phi \, d\Omega = \frac{\partial}{\partial t}\int_{\Omega_{CV}} \rho\phi \, d\Omega + \int_{S_{CV}} \rho\phi U \cdot n \, dS \qquad \text{Equation 5}$$

where $\Omega_{CV}$ stands for the control volume CV, $S_{CV}$ is the surface enclosing the CV, $n$ is the unit vector orthogonal to $S_{CV}$, and directed outwards (Figure 8). The first and second term on the right side of Equation 5 are the temporal and the convective (diffusion and advection) terms, respectively.

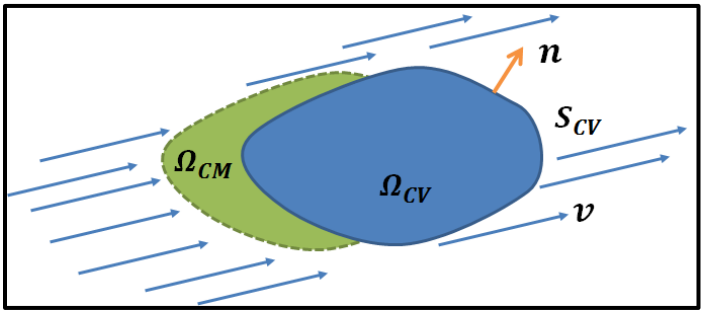


*Figure 8. Control volume approach for the analysis of fluid field flow.*

Four conservation equations follow directly from the control volume equation: mass, momentum, scalar, and energy. Each equation is next described.

### 4.2.2. Mass conservation

Mass conservation equation is obtained by setting $\phi = 1$, and subsequently Equation 5 becomes the left side of Equation 2. If a fixed CV is considered, $\Omega$ represents the CV, and $S$ its surface:

$$\frac{\partial}{\partial t}\int_{\Omega} \rho \, d\Omega + \int_{S} \rho U \cdot n \, dS = 0 \qquad \text{Equation 6}$$

The convection term is transformed from a surface integral into a volume integral by applying the Gauss divergence theorem. If the control volume is infinitesimally small, the differential coordinate-free form of the continuity equation (without reference to any particular coordinate system) is:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho U) = 0 \qquad \text{Equation 7 a)}$$

or

$$\frac{D\rho}{Dt} + \rho \nabla \cdot (U) = 0 \qquad \text{Equation 7 b)}$$

where $\frac{D\rho}{Dt}$ is the material derivative of the density $\rho$. Equation 7 a) and Equation 7 b) are known to be the conservative and non-conservative form of the continuity equation, respectively.

For incompressible flow, the mass conservation equation can be expressed as:

$$\nabla \cdot U = 0 \qquad \text{Equation 8}$$

### 4.2.3. Momentum conservation

The momentum conservation equation is obtained by assuming $\phi = U$. For a fixed fluid-containing volume of space:

$$\frac{\partial}{\partial t}\int_{\Omega} \rho U \, d\Omega + \int_{S} \rho UU \cdot n \, dS = \sum f \qquad \text{Equation 9}$$

Considering surface and body forces per unit mass, momentum equation is written as:

$$\frac{\partial}{\partial t}\int_{\Omega} \rho U \, d\Omega + \int_{S} \rho UU \cdot n \, dS = \int_{S} \boldsymbol{T} \cdot n \, dS + \int_{\Omega} \rho b \, d\Omega \qquad \text{Equation 10}$$

where $\boldsymbol{T}$ is the stress tensor (Equation 11), and $b$ is a source/sink term (body forces per unit mass). For Newtonian fluids:

$$\boldsymbol{T} = -\left(p + \frac{2}{3}\mu \nabla \cdot U\right) I + 2\mu \boldsymbol{S} \qquad \text{Equation 11 a)}$$

or

$$\boldsymbol{T} = -pI + 2\mu \boldsymbol{S} - \frac{2}{3}\mu I \nabla \cdot U \qquad \text{Equation 11 b)}$$

where $\mu$ is the dynamic viscosity, $I$ the unit tensor, $p$ is the static pressure, and $\boldsymbol{S}$ is the rate of strain (deformation) tensor (Equation 12). The term $(2\mu \boldsymbol{S} - \frac{2}{3}\mu I \nabla \cdot U)$ in Equation 11 b) is called the viscous part of the stress tensor.

$$\boldsymbol{S} = \frac{1}{2}(\nabla U + (\nabla U)^T) \qquad \text{Equation 12}$$

A coordinate-free vector form of Equation 10 is obtained by applying the Gauss divergence theorem to the convective and diffusive flux terms (Equation 13).

$$\frac{\partial \rho \mathrm{U}}{\partial t} + \nabla \cdot (\rho UU) = \nabla \cdot \boldsymbol{T} + \rho b \qquad \text{Equation 13}$$

For incompressible flow ($\nabla \cdot U = 0$), the stress tensor $\boldsymbol{T}$ is simplified (Equation 14). The gradient of the velocity vector $U$ is a second rank tensor (matrix of order three). Thus, the summation of the terms $\nabla U + (\nabla U)^T$ is a symmetric matrix.

$$\boldsymbol{T} = -pI + 2\mu\left(\frac{1}{2}(\nabla U + (\nabla U)^T)\right) = -pI + \mu(\nabla U + (\nabla U)^T) \qquad \text{Equation 14}$$

The divergence of the stress tensor $\boldsymbol{T}$ is

$$\nabla \cdot \boldsymbol{T} = \nabla \cdot \left(-pI + \mu(\nabla U + (\nabla U)^T)\right) = -\nabla \cdot (pI) + \nabla \cdot \mu \nabla U = -\nabla p + \mu \nabla^2 U$$

This term is substituted in Equation 13, considering gravity, $g$, as the only body force:

$$\frac{\partial \rho \mathrm{U}}{\partial t} + \nabla \cdot (\rho UU) = \nabla \cdot (\mu \nabla U) - \nabla p + \rho g \qquad \text{Equation 15 a)}$$

or

$$\frac{\partial \mathrm{U}}{\partial t} + U \cdot \nabla U = \upsilon \nabla^2 U - \frac{1}{\rho}\nabla p + g \qquad \text{Equation 15 b)}$$

where $\upsilon$ is de constant kinematic viscosity, $p$ is the pressure. The first term on the left-hand side of the equation is the temporal term, and the second one is the convective term, which makes the equation nonlinear. $\nabla \cdot (\mu \nabla U)$ is known as the viscous term or the diffusion term.

For incompressible flows, it is convenient and more efficient for the numerical solution, to define $p^* = p - \rho g z$ as the dynamic pressure and use it in the place of the pressure. The term $\rho g$ in Equation 15 a) is then considered in the expression $-\nabla(p - \rho g z) = -\nabla(p^*)$. If the actual pressure is needed, one has to add $\rho g z$ to $p^*$.

$$\frac{\partial \rho \mathrm{U}}{\partial t} + \nabla \cdot (\rho U U) = \nabla \cdot (\mu \nabla U) - \nabla(p^*) \qquad \text{Equation 16}$$

This form allows ignoring the effect of the gravitational acceleration when computing fluid motion in a constant density fluid. If there is no fluid velocity, then $p^* = 0$ and the total pressure is only the hydrostatic pressure. In constant density flows, gravity does not play a role in determining the flow field (Ten and Edwards, 2006).

### 4.2.4. Conservation of scalar quantities

The third conservation principle is the conservation of scalar quantities equations. The integral form is also derived from the control volume equation:

$$\frac{\partial}{\partial t} \int_{\Omega} \rho \phi d\Omega + \int_{S} \rho \phi U \cdot n \, dS = \sum f_{\phi} \qquad \text{Equation 17}$$

where $f_{\phi}$ is the transport of the scalar quantity $\phi$ by mechanisms other than convection and any sources or sinks of the scalar. This equation can be written as in Equation 18 for incompressible flow (conservative form).

$$\frac{\partial \rho \phi}{\partial t} + \nabla \cdot (\rho \phi U) = \nabla \cdot (D \nabla \phi) + q_{\phi} \qquad \text{Equation 18}$$

where $D$ is the diffusivity for the quantity $\phi$, and $q_{\phi}$ is a source/sink term of $\phi$. The first term of the right side of Equation 18 is the diffusive term, and the second one any source/sink term. Diffusive transport is usually described by a gradient approximation and is always present, even in stagnant fluids. For heat diffusion, it is described by Fourier's law and, for mass diffusion, by Fick's law.

This equation is used to represent scalar quantities in the model such as the different fluid phases involved. The diffusive term is important to define the mixing between liquid phases. A detailed description on the obtainment of Equation 18 right side is provided by Ferziger & Peric, (2002).

### 4.2.5. Conservation of energy

The energy equation, for most engineering flows, can be written as in Equation 19.

$$\frac{\partial}{\partial t}\int_{\Omega} \rho h \, d\Omega + \int_{S} \rho h U \cdot n \, dS = \int_{S} k\nabla T \cdot n \, dS + \int_{\Omega} (U \cdot \nabla p + \mathbf{S} \cdot \nabla \mathrm{U}) \, d\Omega + \frac{\partial}{\partial t}\int_{\Omega} \rho \, d\Omega \qquad \text{Equation 19}$$

where $h$ is the enthalpy, $T$ is the absolute temperature, $k$ is the thermal conductivity, $k = \mu c_p/Pr$, $\boldsymbol{S_0}$ is the viscous part of the stress tensor $\boldsymbol{T}$, $\boldsymbol{S_0} = \boldsymbol{T} + pI$, $Pr$ is the Prandtl number, $\mu$ is the dynamic viscosity, and $c_p$ is the specific heat capacity at constant pressure. In this equation, radiative heat transfer and internal heat generation due to a possible chemical or nuclear reaction are neglected.

A coordinate-free vector form of the energy equation for incompressible flows ($\nabla \cdot U = 0$) is:

$$\frac{\partial \rho h}{\partial t} + \nabla \cdot (\rho h U) = \nabla \cdot (k \nabla T) - \frac{Dp}{Dt} + q_\phi \qquad \text{Equation 20}$$

where $h$ is the specific enthalpy which is related to specific internal energy, $h = e + p/\rho$, and $q_\phi$ is the dissipation function, representing the work done against viscous forces. The dissipation function is irreversibly converted into internal energy, $q_\phi = (\tau \cdot \nabla)U$.

The relation $dh = c_p dT$ and the energy equation can be written as in Equation 21 for incompressible flows, when considering a fluid with constant specific heat capacity ($c_p$) in former, and neglecting the pressure term for incompressible flows (constant density) in latter equation::

$$\frac{\partial \rho c_p T}{\partial t} + \nabla \cdot \left(\rho c_p T U\right) = \nabla \cdot (k \nabla T) + q_\phi \qquad \text{Equation 21 a)}$$

or

$$\rho c_p \left(\frac{\partial T}{\partial t} + (U \cdot \nabla) T\right) = k \nabla^2 T + q_\phi \qquad \text{Equation 21 b)}$$

Mathematically, compressible and incompressible flows are handled in different ways. CFD codes are usually written for only one of them. Thus, the complete system must be considered either compressible or incompressible for its analysis.

*Heat transfer*

There are three mechanisms of heat transfer: conduction, radiation, and convection. Convection is most closely connected with fluid mechanics. On the other hand, heat conduction is described by Laplace's equation (Equation 22) while unsteady conduction is governed by the heat equation by adding a time derivative to it (Equation 23).

$$\nabla^2 \phi = 0 \qquad \text{Equation 22}$$

$$\frac{\partial \phi}{\partial t} - k \nabla^2 \phi = 0 \qquad \text{Equation 23}$$

where $k$ is the thermal conductivity, and $\nabla^2$ the Laplace operator.

Within flows accompanied by heat transfer, the fluid properties are typically a function of temperature $T$. The properties are usually calculated using first the temperature on the current iteration; then, the temperature is updated, and finally, the process is repeated.

The Boussinesq approximation is useful when the density and temperature variations are small (e.g., temperature differences below $2\ ℃$ for water) and introduces an error of the order of $1\ \%$. This approach consists in treating the density as constant in the unsteady and convection terms, and as variable only in the gravitational term; furthermore, the density is assumed to vary linearly with temperature. In variable density flows, one can split the $\rho g$ term in Equation 15 into two parts: $\rho g = \rho_0 g + (\rho - \rho_0)g$, where $\rho_0$ is a reference density. If the effect of the body force on the mean density is included in the pressure term, the remaining term can be expressed as $(\rho - \rho_0)g = -\rho_0 g\beta(T - T_0)$; where $\beta$ is the coefficient of volumetric expansion (Ferziger and Peric, 2002).

For laminar flow, the dominant processes of convective heat transfer are advection in the stream-wise direction, and conduction in the direction that is normal to the flow. If the flow is turbulent, the role played by conduction in laminar flows is taken by the turbulence, which is represented by a turbulence model.

When temperature variations are significant, the energy (Equation 16) and NS equations (Equation 8 and Equation 16) are coupled and must be solved simultaneously. These three equations need to be solved simultaneously, and they have six unknown scalars: density, pressure, three velocity components, and temperature. They are solved considering six scalar equations: conservation of mass, three components of conservation of momentum, conservation of energy, and equation of state. Nevertheless, when the flow is incompressible, it does not exist any equation of state.

The variation of the transport properties with the temperature also affects the fluid flow. However, the solution can be obtained in a sequential fashion: on each iteration, the momentum equations are first solved using transport properties computed from the previous temperature field; the temperature field is updated after the solution of the momentum equations has been obtained for the new iteration, and the properties are updated (Ferziger and Peric, 2002). This technic is similar to the one used to solve equations with a turbulence model.

In the buoyancy-driven or natural-convection flow, the effect of temperature results in density variations interacting with gravity. Thus, a body force is produced and it may modify the flow considerably, and is the principal driving force in the flow (Ferziger and Peric, 2002). Computation of this kind of flows can be done with the Boussinesq approximation for low-temperature variations (as described above), or through the iteration of the velocity field which precedes the iteration for the temperature and density fields. However, the last procedure may converge more slowly than in isothermal flows, when the coupling is strong. Another option is the solution of the equations as a coupled system, which increases the convergence rate at the cost of the increased complexity of programming. The coupling strength depends on the Rayleigh, Reynolds and Prandtl number (e.g., the coupling is strong for high Prandtl numbers).

## 4.3. SOLUTION OF THE GOVERNING EQUATIONS

The fluid flow is affected by several factors such as:

1. Action of externally applied forces such as pressure differences, shear, rotation and surface tension, which are classified as surface forces (e.g., shear force due to wind blowing above

the ocean, shear forces created by a movement of a rigid wall, pressure, surface tension, etc.), and body forces (e.g., gravity, centrifugal and Coriolis forces, electromagnetic forces, etc.).

2. Density and viscosity of the fluid
3. The speed of the flow. When the flow speed increases, it may lead to instability that causes a type of flow called turbulent. Furthermore, the flow may be considered essentially incompressible at low Mach numbers (e.g., $< 0.3$).
4. Temperature differences, which lead to heat transfer and density differences which give rise to buoyancy. This phenomenon, along with phase changes, leads to important modifications of the flow and give rise to multiphase flow.

In this research the flow is considered to obey Newton's law; thus, the flow is Newtonian. The momentum and the continuity equations are usually known as the Navier-Stokes Equations (NSE) and are particular cases of the *control volume equation* (Equation 5).

NSE are non-linear equations that accurately describe the flow of a Newtonian fluid (Ferziger and Peric, 2002). The momentum and the continuity equations or NSE are difficult to solve, but simplifications are usually justified to reduce computational costs. For incompressible and isothermal flow, the NSE can be expressed as Equation 8 and Equation 15.

To solve the NSE, an approximated solution is given by a time-averaging process, which consists of the decomposition or separation of the flow variable into the mean (time-averaged) component and the fluctuating component. If the flow variable is the velocity $u$, it is separated into the mean component $\overline{U}$ and the fluctuating component $U'$, that is $U(\boldsymbol{x},t) = \overline{U}(\boldsymbol{x}) + U'(\boldsymbol{x},t)$, where $\boldsymbol{x} = (x,y,z)$ is the position of the vector (Figure 9). Analogous for the density $p^*$ and temperature $T$: $p^*(\boldsymbol{x},t) = \bar{p}(\boldsymbol{x}) + p'(\boldsymbol{x},t)$, $T = \overline{T}(\boldsymbol{x}) + T'(\boldsymbol{x},t)$. The decomposed terms are then introduced in the momentum equation for incompressible flows. The average of this equation is taken by considering that the mean of the fluctuating component is equal to zero ($\overline{U'} = 0$).

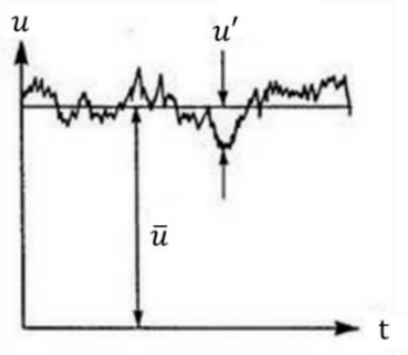


*Figure 9. Decomposition of the flow variable of the velocity U into its mean component and the fluctuating component.*

The averaged equations are known as Reynolds Averaged Navier-Stokes equations (RANS equations). RANS equations can be written as in Equation 8 (mass conservation) and Equation 24 for incompressible flows:

$$\frac{\partial \rho \mathrm{U}}{\partial t} + \nabla \cdot (\rho UU) = \nabla \cdot (\mu \nabla U - \rho\tau) - \nabla(p) + \rho g + \sigma \kappa_\alpha \nabla \alpha \qquad \text{Equation 24}$$

where $\tau$ is the specific Reynolds stress tensor (Equation 25). The expression $(\mu\nabla U - \rho\tau)$ corresponds to the total shear stress. Specific Reynolds stress tensor $\tau = \overline{u_i' u_j'}$ can be defined through the Boussinesq Approximation (Equation 25), which relates Reynolds stresses to mean velocity gradients. This term is symmetric and requires additional modeling to close the RANS equations. Thus, turbulence models must be introduced.

$$\tau = \frac{2}{\rho}\mu_t S - \frac{2}{3}kI \qquad \text{Equation 25}$$

where $\mu_t$ is the dynamic eddy viscosity (assumed as an isotropic scalar quantity $\frac{\mu_t}{\rho} = \upsilon_t$, $\upsilon_t$ is the kinetic eddy viscosity), $S = \frac{1}{2}(\nabla U + (\nabla U)^T)$ the strain rate tensor, $I$ the Kronecker delta, and $k$ the turbulent kinetic energy per unit mass, defined as $k = \frac{1}{2}\overline{u_i' u_j'}$.

### 4.3.1. Volume fraction equation and VOF method

The last term in Equation 24 is the superficial tension effect, where $\sigma$ is the superficial tension coefficient, $\kappa_\alpha$ is the surface curvature, and $\alpha$ represents a scalar field for the identification of the different phases. The value of $\alpha$ is obtained with the Volume of Fluid (VOF) method (Berberović et al., 2009). For a free surface fluid system, $\alpha = 0$ for the air phase and $\alpha = 1$ for the liquid. The intermediate values represent a mixture between the fluids (or interface) (Figure 10).

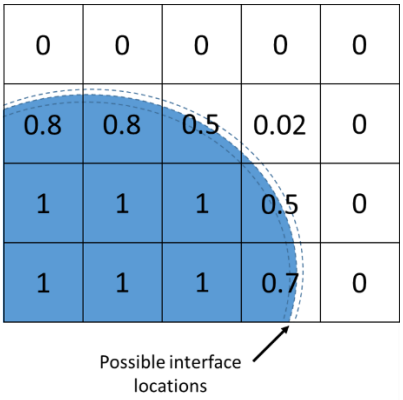


*Figure 10. VOF method for tracking the fluid phases*

The distribution of $\alpha$ is modeled with a convective equation named volume fraction equation (Equation 26), which is a particular case of the conservation of scalar quantities equation.

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot (\alpha U) + \nabla \cdot \left(U_r \alpha (1-\alpha)\right) = 0 \qquad \text{Equation 26}$$

where $U_r$ is a relative velocity. The last term on the left side of Equation 26 is a compression term that limits the interface size.

### 4.3.2. Turbulence models

The Reynolds-Averaged Navier-Stokes equations (RANS) do not form a closed set of equations, so the introduction of approximations is required. Those approximations are known as turbulence models (Ferziger and Peric, 2002)

Turbulence is characterized by several observable properties such as an irregular signal in space and time, rotational flow, high diffusivity, unpredictable character of trajectories, the coexistence of eddies of very different scales and dissipation (Figure 11). Ferziger and Peric (2002) describe the turbulent flow with the following features:

i) it is highly unsteady
ii) it is a three-dimensional phenomenon
iii) the vorticity is one of its principal mechanisms
iv) it includes a process called turbulent diffusion
v) it brings fluids of different momentum content into contact
vi) it reduces the flow kinetic energy through an irreversible dissipative mixing process
vii) the turbulent flow contains coherent structures but also a random component
viii) it fluctuates on a broad range of length and time scales which make the numerical simulation a very difficult task.

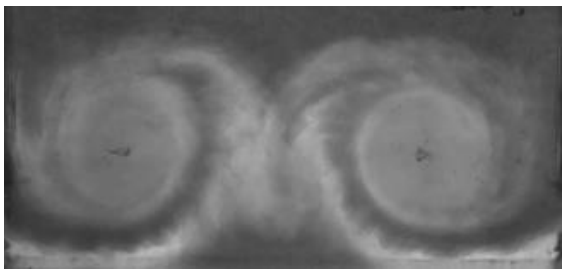

*Figure 11. Example of a turbulent flow. The properties of a turbulent flow such as eddies of different scale, high diffusivity and rotational flow, are present.*

There are different approaches to model turbulence in fluids simulations, which have been developed with different levels of complexity, for instance: a) statistical models with a turbulence-viscosity hypothesis, or b) advanced models that involve transport equations or turbulent large eddy numerical simulations. However, each approach has its performance advantages and limitations. They are not considered to compete but are complementary and are chosen depending on the type of problem to be solved as well as the response to be expected (Schiestel, 2008). Turbulent flows contain variations on a more extended range of length and time scales than laminar flows. The most common approaches are herein briefly described, and further references can be found in Bardina *et al.* (1997), Ferziger and Peric (2002), Mcdonough (2007) and Kobayashi and Tsubokura (2011).

*Reynolds-Averaged Navier-Stokes (RANS) turbulence simulation*

This approach is called one-point closure and is based on the equations obtained by averaging the equations of motion over time. The most common turbulence models are, among others, the zero-

equation and two-equation models. The zero-equation model sets the turbulent viscosity as a function of the velocity field. The two-equation models add two extra transport equations to represent the turbulent properties of the flow. One of the transported variables is the turbulent kinetic energy $k$. The second variable depends on what type of model it is being used (e.g., $k$-$\varepsilon$ where ε is the turbulent dissipation and determines the rate of dissipation of the turbulent kinetic energy, $k$-$\omega$, where $\omega$ is the specific dissipation, and $k$-$kL$, where $kL$ is the turbulent length scale). The introduction of this wide range of RANS approaches began around 1972 and are one of the most common types of turbulence models used in the industry for most engineering problems.

*Large eddy simulation (LES)*

Proposed by Deardorff in 1970, LES was the first computational technique for turbulence modeling. This approach solves the largest scale motions of the flow and approximates the small-scale motions. It is one of the most promising turbulent simulations for engineering purposes (Kobayashi and Tsubokura, 2011). Although LES was not feasible for practical engineering problems in the 70s and 80s (Mcdonough, 2007); the transition from RANS to LES was enhanced thanks to the development of High-Performance Computing technology and the progress of parallel-processor computers (Kobayashi and Tsubokura, 2011).

*Direct numerical simulation (DNS)*

In this approach, the NSE are solved for all the scales of the turbulent fluid flow. The first Direct Numerical Simulation (DNS) was introduced by Orszag and Patterson (1972). The equations are solved without averaging or approximation other than the numerical discretization. Therefore, it is the most accurate approach. However, this method has several limitations, and thus its major role has been focused on a research tool (Ferziger and Peric, 2002). DNS is computationally expensive for practical simulations.

### 4.3.3. Multiphase flow modeling

Any fluid flow consisting of more than one phase or component is called multiphase flow. According to the state of the different phases or components, multiphase flows can be classified as gas/solid, liquid/solid, gas/particle, bubbly flows, etc. (Brennen, 2005). Information about multiphase flows is focused on specific types of fluid flow (e.g., low Reynolds number suspension flows and dusty gas dynamics) or specific applications such as slurry flows and cavitating flows.

The ability to predict fluid flow behavior is critical to assess the efficiency and effectiveness of processes that involve multiphase flows, such as cavitating pumps and turbines. According to Brennen (2005), there are three ways in which the multiphase fluid motion can be explored:

1. Experimentally, through laboratory-sized models equipped with appropriate instrumentation. Although there are many applications in which full-scale laboratory models are possible, in many instances, the laboratory model must have a significantly different scale than the prototype. There are also cases in which laboratory models are not feasible for a variety of reasons.

2. Theoretically, by considering analytical and mathematical equations to model the fluid motion. This method is limited as it requires many simplifications and many complex cases that cannot be solved with enough accuracy or necessary detail.
3. Computationally, using the power and size of modern computers to address the complexity of the flow. The computer power and speed are relevant elements to consider when modeling the flows that are commonly experienced, such as turbulent flows. A reliable theoretical or computational model is thus needed to provide a confident extrapolation to the scale of prototypes.

For a free-surface fluid flow condition, such as waves in the sea, there exists a numerical technique for tracking and locating the free surface or fluid-fluid interface. This technique is used in CFD tools and is called the Volume of Fluid (VOF) method, which is detailed by Berberović *et al.*, (2009). VOF method consists on tracking each fluid through every cell in the computational grid (while all fluids share a single set of momentum equations) using a scalar function that takes the value of zero when the cell is empty, one when the cell is full, and values between zero and one when a fluid interface is found (Figure 12).

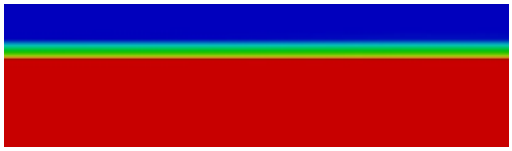

Figure 12. CFD simulation with VOF method. Water in red, air in blue and interface between air and water in green/yellow.

The VOF method introduces only one extra equation; thus, it is computationally friendly. This method is more flexible and efficient than other methods for treating complicated free-boundary configurations and sharp topological changes in non-linear problems (Hirt and Nichols, 1981).

## 4.4. SUMMARY AND IMPLICATIONS FOR THE DISSERTATION

Unstructured numerical meshes are an adequate option to model complex geometries, which is useful for detailed simulations. Even when the domain is relatively large, unstructured meshes allow refinement where necessary. Thus, a CFD tool that handles the finite volume discretization method is preferred.

The conservation principles of mass, momentum, and energy precede the governing equations that describe the fluid flow motion. In this study, the fluid flow involves a free-surface condition, two miscible fluids with different temperature and density, wave propagation, and various fluid flow directions and velocity. Therefore, the Navier Stokes equations (continuity and momentum conservation equations) are the two main governing equations to model the Newtonian fluid. These equations can be solved by an averaging process known as the Reynolds Average Navier Stokes (RANS) Equations. Turbulence models must be introduced to close the RANS equations. The most common turbulence models in CFD are two-equation models and LES models.

On the other hand, the interphase that defines the free-surface condition can be modeled using the VOF method, which is based on a volume fraction conservation equation. The scalar conservation equation, which is a function of the diffusivity coefficient, can be used to model the mixture between the liquid phases. Finally, the temperature distribution can be modeled with the energy conservation equation.

According to the objectives proposed in this study, it is important to identify a CFD platform that can integrate the aspects mentioned above, and that solves the governing equations efficiently in complex domains.

# 5

# DEVELOPMENT OF THE CFD MODEL

# 5. DEVELOPMENT OF THE CFD MODEL

This section describes the process followed to develop a CFD tool to simulate OTEC discharge in coastal waters. The NSE (mass and momentum conservation equations) coupled with the energy, scalar quantities, and volume fraction conservation equations are the governing equations implemented in the developed CFD model to describe OTEC discharge in coastal waters.

OTEC water discharge involves multiphase fluid flow with different temperatures, densities, and fluid flow directions. After evaluating these characteristics and the different CFD software available, the OpenFOAM® CFD platform is selected to implement a new solver capable of simulating the flow described features.

## 5.1. SELECTION OF THE CFD PLATFORM

After comparing different available CFD software, OpenFOAM® is selected as the most appropriate CFD package to simulate thermal multiphase fluid flow associated with OTEC discharge water. This software contains several solvers to simulate a diversity of problems mostly related to fluid mechanics, e.g., porous medium, multiphase flow, waves, free surface, and different turbulence models.

OpenFOAM® is a free, open-source software package developed by OpenCFD (http://www.openfoam.com/). This tool is gaining popularity because of its potential and the cost reduction due to the elimination of license fees (Tomislav et al., 2014), as well as for the possibility of extending/changing/reviewing the originally implemented codes. The key advantage of this CFD toolkit is that the user can freely use and modify the CFD code, which can be a common platform for collaborative projects between developers. However, a considerable amount of effort is needed to learn how to use and extend the platform when compared to commercial CFD software, whose performance, governing equations or their implementation remain as a “black box” to the user.

## 5.2. DESCRIPTION AND IMPLEMENTATION OF `interMixingTemperatureWaveFoam` IN OPENFOAM®

The Annexes of this work include a detailed description of the extension of the OpenFOAM® platform, the code, and a complete case setup with the new solver. In the following a summary of this process is presented, focussing on the conceptual foundations and the implemented equations.

Following the solver’s name convention in OpenFOAM®, the implemented solver is named `interMixingTemperatureWaveFoam`, where “interMixing” stands for the ability to solve 3 incompressible fluids, two of which are miscible, “Temperature” for the implementation of the energy equation, and “Wave” for the ability of wave generation and absorption. Turbulence models, mesh generation schemes and other generic OpenFOAM® utilities are available for the case setting and running.

The developed `interMixingTemperatureWaveFoam` solver handles the multiphase fluid flow and wave generation/absorption. This multiphase fluid flow consists of three phases with different densities and temperatures, i.e., two of the phases are liquid and miscible, and the third one represents air; thus, there is a free-surface condition.

The governing equations used in `interMixingTemperatureWaveFoam` solver can be summarized as: momentum conservation equation (Equation 27), mass conservation equation (Equation 28), scalar quantities conservation equation (Equation 29), volume fraction equation (Equation 30), and energy conservation equation (Equation 31). These equations are extensively described in Chapter 4.

$$\frac{\partial \rho \mathrm{U}}{\partial t} + \nabla \cdot (\rho U U) = \nabla \cdot (\mu \nabla U) - \nabla p + \rho g \qquad \text{Equation 27}$$

$$\nabla \cdot U = 0 \qquad \text{Equation 28}$$

$$\frac{\partial \rho \phi}{\partial t} + \nabla \cdot (\rho \phi U) = \nabla \cdot (D \nabla \phi) + q_\phi \qquad \text{Equation 29}$$

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot (\alpha U) + \nabla \cdot \big(U_r \alpha (1 - \alpha)\big) = 0 \qquad \text{Equation 30}$$

$$\frac{\partial \rho c_p T}{\partial t} + \nabla \cdot \big(\rho c_p T U\big) = \nabla \cdot (k \nabla T) + q_\phi \qquad \text{Equation 31}$$

Where $t$ is the time, $U$ the velocity, $\rho$ the fluid density, $\mu$ the dynamic viscosity, $p$ the pressure, $g$ the gravitational acceleration, $\phi$ an scalar quantity, $D$ the diffusion coefficient, $\alpha$ a scalar field for the identification of the phases or phase volume fraction, $U_r$ the relative velocity, $c_p$ the specific heat capacity, $T$ the temperature, $k$ the thermal conductivity, and $q_\phi$ is a source/sink term of $\phi$.

Equation 27 and Equation 28 are known as NSE, which are solved using the Reynolds average method, i.e., the Reynolds Averaged Navier-Stokes equations. Therefore, the momentum equation is written as in Equation 32, with $\tau$ as the Reynolds stress tensor (Equation 33), $\mu_t$ the dynamic eddy viscosity or turbulent viscosity, $\kappa_\alpha$ the surface curvature, $\sigma$ the surface tension, $S = \frac{1}{2}(\nabla U + (\nabla U)^T)$ the rate of deformation tensor, $k$ the turbulent kinetic energy per unit mass, and $I$ the Kronecker delta.

$$\frac{\partial \rho \mathrm{U}}{\partial t} + \nabla \cdot (\rho U U) = \nabla \cdot (\mu \nabla U + \rho \tau) - \nabla p + \rho g + \sigma \kappa_\alpha \nabla \alpha \qquad \text{Equation 32}$$

$$\tau = \frac{2}{\rho} \mu_t S - \frac{2}{3} k I \qquad \text{Equation 33}$$

The interphase between the miscible liquid phases and the gaseous phase is obtained with the Volume of fluid method (VOF), which tracks the fluid phase through the scalar field $\alpha$ in an advection equation (Equation 30). VOF method assigns to $\alpha$ the value of 0 for the gaseous phase, 1 for the liquid phase and values between 0 and 1 for their interphase.

Equation 29 involves the molecular diffusivity $D$ to model the mixture between the liquid phases. This coefficient is the proportionality constant of Fick's law and represents the ease with which each solute moves in the solvent. In the International System of Units, $D$ is expressed as $\mathrm{m}^2/\mathrm{s}$ and varies

with temperature; however, for simplicity of the model, it is considered that $D$ remains constant in the time interval studied.

A numerical tool named `waves2Foam` (implemented and validated by Jacobsen et al., 2012) is coupled with the governing equations to include wave generation and absorption through the "wave relaxation zones" method. Rodríguez-Ocampo (2016) integrated Equation 27 to Equation 30, as well as wave generation/absorption models into a solver named `interMixingWaveFoam`. However, the implementation of the energy conservation equation (Equation 31) is required to model the temperature field. Because no phase changes are contemplated in the new solver, and for simplicity of the model, this first approach considers the source term $q_\phi$ as zero.

The values of the Prandtl number $Pr$ and the specific heat capacity $c_p$ are introduced as constant data for each of the three phases, and the thermal conductivity $k$ for the mixture is calculated to model the temperature field with the energy conservation equation. Prandtl number is a dimensionless number that is proportional to the ratio of momentum diffusivity to thermal diffusivity; in other words, it is the quotient between the viscous diffusion rate and the thermal diffusion rate. Thus, it indicates the effectiveness of the conduction compared to the convection when transferring heat; e.g., for large Prandtl numbers convection is more efficient to transfer heat than conduction and vice versa for low ones. Water Prandtl number, for example, is about 4-6 times larger than air. On the other hand, specific heat capacity, in $\frac{\mathrm{J}}{\mathrm{kg \cdot K}}$, is an intensive property that describes the amount of energy (heat) to increase the temperature of one-kilogram mass in one Kelvin. Finally, thermal conductivity, in $\frac{\mathrm{J}}{\mathrm{m \cdot s \cdot K}}$, is also an intensive property that measures the heat conduction capacity of a substance and can be obtained as:

$$k = \rho \upsilon \frac{c_p}{Pr} \qquad \text{Equation 34}$$

Since the model works with three different fluid phases, $k$ is obtained through one weighted arithmetic mean. Therefore, $k$ is affected by the fraction phase in each cell, as shown in **¡Error! No se encuentra el origen de la referencia.**.

$$k = \alpha_1 \rho_1 \upsilon_1 \frac{c_{p1}}{Pr_1} + \alpha_2 \rho_2 \upsilon_2 \frac{c_{p2}}{Pr_2} + \alpha_3 \rho_3 \upsilon_3 \frac{c_{p3}}{Pr_3} \qquad \text{Equation 35}$$

Moreover, specific heat capacity for the mixture $\rho * c_p$, as well as the heat flux $\rho * \varphi * c_p$, are obtained respectively as:

$$\rho * c_p = \alpha_1 \rho_1 c_{p1} + \alpha_2 \rho_2 c_{p2} + \alpha_3 \rho_3 c_{p3} \qquad \text{Equation 36}$$

$$\rho * \varphi * c_p = \alpha_1 \varphi \rho_1 c_{p1} + \alpha_2 \rho_2 \varphi c_{p2} + \alpha_3 \rho_3 \varphi c_{p3} \qquad \text{Equation 37}$$

Equation 35, Equation 36 and Equation 37 are the terms associated to the Laplacian, the partial derivative, and the divergence respectively of the energy conservation equation (Equation 31).

### 5.2.1. Solver's application boundaries

As any other CFD solver in OpenFOAM® platform, `interMixingTemperatureWaveFoam` solver has certain limitations and application boundaries. The main aspects to be considered can be summarized as follows:

1. Source terms in the governing equations are considered zero as a first approach.
2. Phase changes are not considered in the simulation, e.g., boiling and condensation.
3. Gaseous phase is considered to be incompressible.
4. Diffusion coefficient and the viscosity are assumed constants.

The user of the solver must be aware of these application boundaries and each case must not exceed the limitations.

## 5.3. SUMMARY AND DISCUSSION

The implementation of the governing equations in the new solver can be summarized in the following steps:

i) Creation of the solver libraries and folders departing from preexisting native solvers in OpenFOAM®: Native solver `interMixingFoam` is used as a code base for the new `interMixingTemperatureWaveFoam` solver; thus, its folders and files are copied to the project directory and modified with the new solver's name.

ii) Modification of the transport model: the three transport model folders of the native `interMixingFoam` are modified to include the parameters of specific heat capacity and Prandtl number for each phase, as well as the estimation of the thermal conductivity for the mixture.

iii) Incorporation of the energy conservation equation in the code: Equation 31 is incorporated into the new solver's code.

iv) Addition of the wave generation/absorption libraries to the code: The corresponding code lines and libraries of *waves2Foam* toolbox are added to the solver.

v) Compilation of the new solver: The new solver is compiled in the project directory.

vi) Setup of a simple test case: A simple test case is set up to demonstrate the functioning of the new solver and how to declare the new variables.

After testing a simple case with `interMixingTemperatureWaveFoam` solver, it is observed that the distribution of the temperature field is consistent with the distribution of the density field, and the temperature range is stable during the simulation time. Nevertheless, the solver must be validated including different turbulence approaches.

Throughout testing different versions of OpenFOAM® in conjunction with its native solver `interMixingFoam`, artificial disturbances of the velocity field in the gaseous phase were observed. These velocity irregularities do not affect the phase definition (e.g., density fields); however, when implementing `interMixingTemperatureWaveFoam` solver, the disturbances cause a noticeable increase/decrease of the temperature field in the gaseous phase. The degree of

alteration of the temperature field depends on the OpenFOAM® version used. In some versions it causes the case running to diverge and crashes the simulation. The increase/decrease of the temperature field in the gaseous phase is also more noticeable in simulations with air introduction into the liquid phases. It is found that these numerical instabilities can be avoided by artificially incrementing the air's specific heat capacity. Nevertheless, since this solver is developed for OTEC thermal discharge or similar cases, there should not exist great turbulence in the near-surface limit, so these numerical instabilities should not be present due to the air introduction into the liquid phases.

A comprehensive description of the implementation of the new solver can be found in Appendix A. Moreover, a complete case setup can be found in the Appendix B (Setup of a simple test case).

# 6

# VALIDATION OF THE CFD MODEL

# 6. VALIDATION OF THE CFD MODEL

In this chapter, the `interMixingTemperatureWaveFoam` solver validation is presented. The code validation is performed through a benchmark experimental case, based on the methodology proposed in Rodríguez-Ocampo (2016), which is adapted and extended to include the temperature field.

The experimental case is simulated with `interMixingTemperatureWaveFoam` by setting a three-phase model of three fluids with different temperatures in a free-surface condition. For this purpose, an experimental model that reproduces a dam-break case was built. The experimental model consists of an acrylic container with a thin vertical gate that can be rapidly opened. The experimental model is designed to reduce the three-dimensional problem into a two-dimensional. The same arrangement is simulated with the developed solver, and both results are compared to estimate the numerical error.

Since wave generation/propagation is not considered in the validation case, this process only determines the accuracy of the temperature field distribution; and therefore, the energy equation implementation in OpenFOAM®. Validation of the density field distribution is performed by Rodríguez-Ocampo (2016), and validation of the wave generation-absorption module is examined by their authors, Jacobsen et al. (2012).

## 6.1. EXPERIMENTAL SETUP

Laboratory benchmark tests of a wet-dam-break type in a rectangular installation are conducted to evaluate the validity of `interMixingTemperatureWaveFoam` solver. For this purpose, an acrylic container of $0.50$ m length, $0.30$ m height and $0.10$ m width was built, as shown in Figure 13 and Figure 14. The container is divided into two compartments (compartment C1, and compartment C2, Figure 13) that are open at the top (i.e., free-surface condition) and are divided by one vertical slide or gate. The acrylic container is also instrumented with ten thermistors, eight of which are distributed equidistantly in C1 and two in C2 as shown in Figure 14.

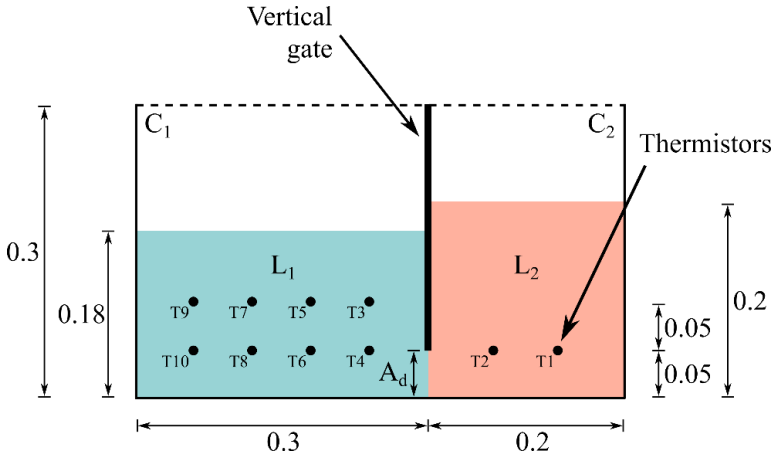


*Figure 13. Schematic representation of the experimental case.*

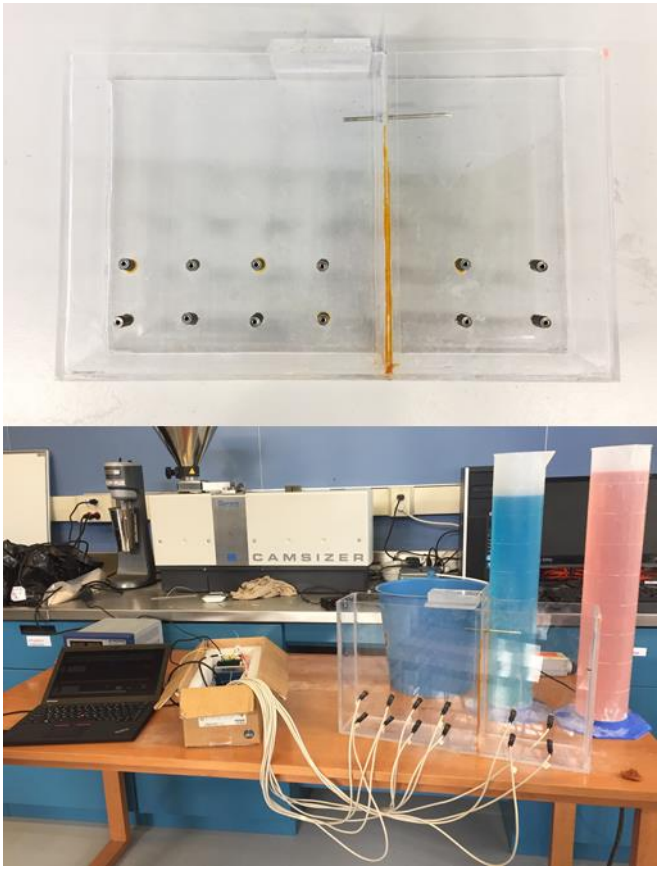


*Figure 14. Acrylic model and its instrumentation with thermistors.*

Before the beginning of the tests, each compartment is filled with water of a certain temperature and density, as described below. C1 is partially filled with liquid L1, i.e., water at room temperature, $\rho_1 = 1002\ \mathrm{kg/m^3}$, $T_1 = 18.8\ ^\circ C$, and $0.18\ \mathrm{m}$ depth. On the other hand, C2 is filled with liquid L2, i.e., heated, $\rho_2 = 989\ \mathrm{kg/m^3}$, $T_2 = 52.2\ ^\circ C$, and $0.20\ \mathrm{m}$ depth, as illustrated in Figure 14. The air is assumed to have density $\rho_a \approx 1\ \mathrm{kg/m^3}$ and the room temperature at the time of the experiments is $T_a = 19\ ^\circ C$.

The water of L2 is heated with an electric resistance until it reaches the desired temperature uniformly; therefore, the water is mixed while being heated. To contrast the two liquid phases and monitor possible leakage before starting the experiment through the vertical gate, the heated is colored with organic vegetable dye, without modifying its density. Furthermore, a thin layer of serum is applied in the borders of the gate to avoid leaking. The water density at the given temperature is measured with a hydrometer 151H ($0.001\ \mathrm{kg/m^3}$ resolution).

The tests considered the sudden release of the gate at the beginning of the experiment ($t = 0.0\ \mathrm{s}$) until it attained a maximum height of 0.05 m (see aperture distance, $A_d$, Figure 14). The release of the gate, allowed multiphase flow mixing due to density, temperature and pressure gradients caused by the different water levels of each compartment.

## 6.2. NUMERICAL SETUP

In this section, the numerical setup to reproduce the experimental results is described. First, the mesh selection through a simple convergence study is presented. Then the solver settings are configured so that the simulation is as similar as possible to the experimental case.

### 6.2.1. Mesh generation and convergence study

The mesh used in the numerical setup is generated with OpenFOAM® utility blockMesh, i.e., the domain is discretized by blocks of hexahedral cells in a Cartesian coordinate system. By default, OpenFOAM® solves the case in three dimensions; however, it can be instructed to solve in two dimensions by specifying a boundary condition called "empty". The mesh is also configured so that there is only one cell in the neglected direction. The other two directions are uniformly discretized so that the cell aspect ratio approaches to one (Figure 18).

Since discretization errors are one of the main sources of computational errors, a grid-space and time-step convergence study is performed. The objective to test different mesh resolutions is to optimize the efficiency and accuracy of the numerical domain.

Three mesh configurations are tested in the convergence study: Mesh A, Mesh B, and Mesh C; where Mesh A is the finest mesh and Mesh C the coarsest one. The refinement factor is set to $\sqrt{2}$ for both spatial and temporal discretization, since both discretization schemes are of second-order (Stern et al., 2001). Table 12 summarizes the characteristics of the three meshes.

*Table 12. Characteristics of Mesh A, Mesh B and Mesh C.*

| | | Mesh A | | Mesh B | | Mesh C | |
|---|---|---|---|---|---|---|---|
| Direction | Dimension (m) | Cell number | Cell size (cm) | Cell number | Cell size (cm) | Cell number | Cell size (cm) |
| X | 0.50 | 186 | 0.2688 | 131 | 0.3817 | 93 | 0.5376 |
| Y | 0.30 | 110 | 0.2727 | 78 | 0.3846 | 55 | 0.5454 |
| Z | 0.10 | 1 | 0.10 | 1 | 0.10 | 1 | 0.10 |

For this mesh convergence study, the case configuration is as follows; however, it is expected that the numerical uncertainties are of the same order for other similar configurations:

i) C1 is partially filled with liquid L1, $\rho_1 \approx 998.2\ \mathrm{kg/m^3}$, $T_1 \approx 20\ ^\circ C$, and 0.18 m depth.
ii) C2 is filled with liquid L2, $\rho_2 \approx 988\ \mathrm{kg/m^3}$, $T_2 \approx 50\ ^\circ C$, and 0.20 m depth.
iii) Air given a density $\rho_a \approx 1\ \mathrm{kg/m^3}$ and $T_a \approx 25\ ^\circ\mathrm{C}$.
iv) Boundary conditions are so that the simulation is two-dimensional and there exists a free-surface condition.

The test case was run for $t = 20$ s and the momentaneous (current) states are stored to disk each 0.05 s. Figure 15 shows the initial distribution of the temperature field in time zero of the convergence-study case.

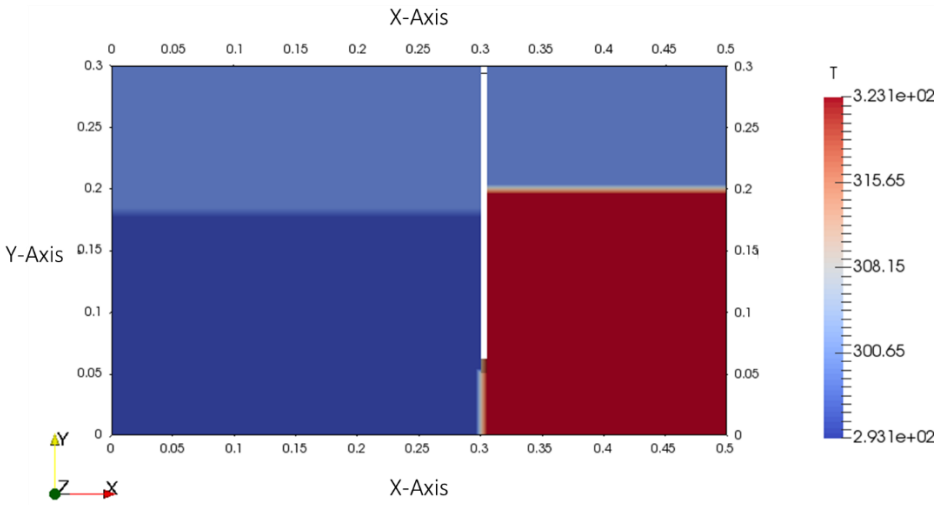


*Figure 15. Distribution of the fluid temperature in time zero. Axes units in meter and temperature in Kelvin.*

The parameter of temperature is used to evaluate the mesh convergence. Instead of comparing the results in only one point (probe) within the domain, three series of probes points are located within the numerical domain, S1, S2 and S3 (Figure 16). Each series consists of 21 probes distributed vertically and with a separation of 0.0025 m in the vertical direction. S1 is aligned with the gate aperture $A_d$ while S2 and S3 are $1$ cm apart from S2, as shown in Figure 16.

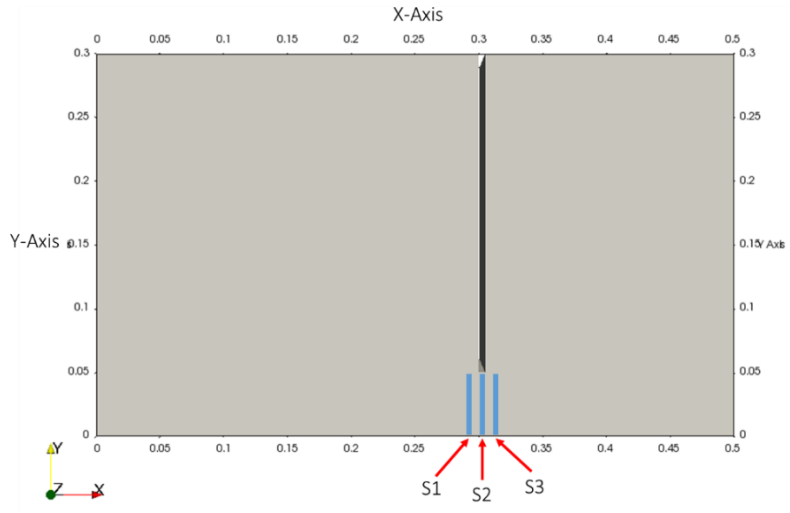


*Figure 16. Distribution of the three series of 21 numerical probes*

The temperature value that is compared between the three meshes is calculated as the weighted average temperature in S1, S2, and S3 (Equation 38):

$$T = 0.50(\overline{S2}) + 0.25(\overline{S1}) + 0.25(\overline{S3}) \qquad \text{Equation 38}$$

where $\overline{S1}$, $\overline{S2}$, and $\overline{S3}$ are the average temperatures of the 21 probes points on each set. The reason to use this method is that it was observed that, qualitatively, the three meshes have similar results but with slight differences on eddies size or slight offsets in time due to the fast nature of the fluid motion in a dam break case. Thus, the mesh accuracy cannot be evaluated with only one probe point.

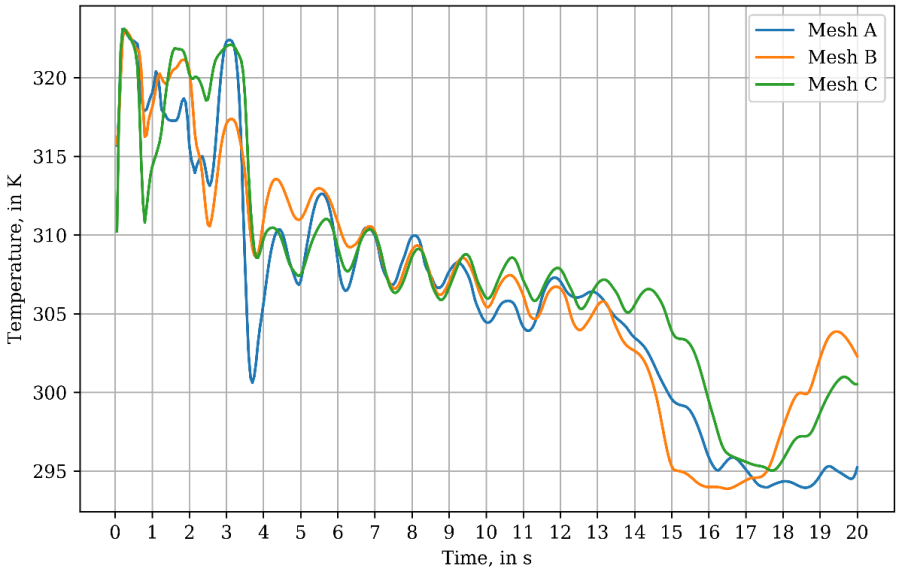


*Figure 17. Time series of temperature in probe point P for the three mesh configurations.*

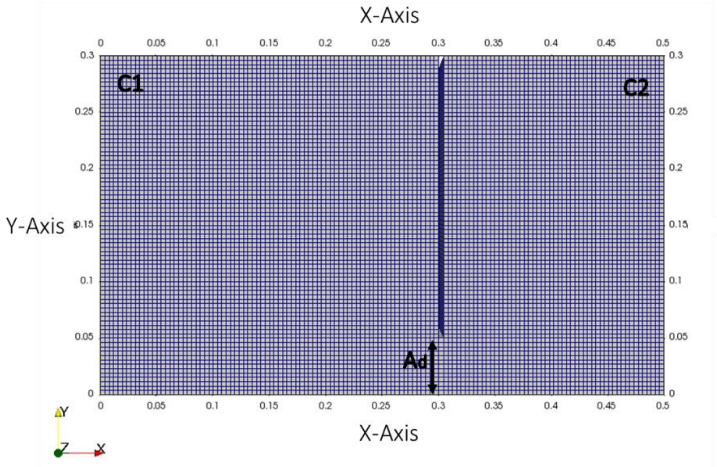


*Figure 18. Numerical domain and mesh of the validation case.*

Temperature variations for Mesh A, Mesh B and Mesh C obtained with Equation 38 are shown in Figure 17. The results show a tendency, which is clearer between $t = 5$ s and $t = 13$ s. Since the results of the finest mesh (Mesh A) vary between Mesh B and C, it can be assumed the convergence of the meshes. Thus, the results are assumed to be hypothetically independent of the mesh resolution.

Taking into account the previous results and considering the optimization on the running time, which is significantly higher for the finest mesh, Mesh B is selected as the most convenient mesh to simulate the experimental test case (Figure 18).

If Mesh A is considered to give the correct results, Mesh B has a relative average error of 7.72 % and Mesh C of 7.45 %. This error is calculated as Equation 39.

$$E_R = \frac{|T_{MA} - T_{MB}|}{\Delta T} * 100 \qquad \text{Equation 39}$$

where $T_{MA}$ is the temperature of Mesh A, in kelvin, $T_{MB}$ the temperature of Mesh B or Mesh C, in kelvin, and $\Delta T$ the temperature difference between the two phases, for this case $\Delta T = 20$ K.

### 6.2.2. Solver settings

After selecting the mesh to be used, the parameters assessed during the experiments are introduced in the model to reproduce the test-case and validate the developed solver. Next, a detailed description of the case setup is presented.

*Constant properties*

As three fluid phases are involved in the simulation (i.e., air, liquid L1, and liquid L2), transport properties, such as kinematic viscosity $\upsilon$, density $\rho$, temperature $T$, and surface tension $\sigma$ are defined as constant for each one. The molecular diffusivity $D$ is a temperature-dependent value; however, it is considered constant for a first approach. $D$ is taken from Holz et al., (2000) for a temperature of $20\,^{\circ}$ C. Table 13 summarizes the input data of the properties of the constant phases.

The gravitational acceleration is also a constant parameter that is set to $9.81\ \mathrm{m/s^2}$ and acts in the negative direction of the vertical coordinate, i.e., Y coordinate in this case.

Finally, the turbulence model is implemented. Three turbulence models are tested to select the most accurate one after the comparison with experimental results. The used turbulence models are:

i) "Laminar" turbulence model. This OpenFOAM® turbulence setting is simply introducing a zero-equation turbulence model in the simulation.
ii) Two-equation (k-$\omega$) RANS model.
iii) LES model.

The case is run three times, one for each turbulence model.

*Table 13. Transport properties defined for the fluid phases and molecular diffusion coefficient for the numerical model setup.*

| Phase property | L1 | L2 | Air |
|---|---|---|---|
| Density (kg/m³) | 1002 | 989 | 1 |
| Kinematic viscosity (m²/s) | 1E-6 | 1E-6 | 1.48E-5 |
| Temperature (K) | 291.95 | 325.35 | 292.15 |
| Prandtl number (1) | 7 | 7 | 0.7 |
| Specific heat capacity (J/kg*K) | 4182 | 4182 | 1433 |
| Surface tension (N/m) | 0.07 | 0.07 | - |
| Molecular diffusion coefficient (m²/s) | 2.023E-9 | | - |

### *Boundary conditions*

Boundary conditions are set for the scalar and vector fields in the numerical simulation:

i) Air field: boundary condition in the top of the domain is set as inletOutlet so that it simulates that the container is opened to the atmosphere, i.e., free-surface condition.
ii) Liquid 1 field: the front and back faces that simulate the lateral walls of the container are given an empty boundary condition, i.e., two dimensional simulation.
iii) Liquid 2 field: the same boundary conditions as L1
iv) Pressure field: boundary condition in the top of the domain is set as totalPressure so that it simulates that the container is opened to the atmosphere.
v) Density field: simple zeroGradient and empty boundary conditions.
vi) Temperature field: simple zeroGradient and empty boundary conditions.
vii) Velocity field: boundary conditions set all fluids with velocity equal to 0 m/s, i.e., all fluids at rest.

### *Fields' settings*

The setting of the fields consist in placing in the numerical domain "fluid blocks" with the required size, density, velocity and temperature. The velocity field for all phases is initially set to $0\ \mathrm{m/s}$, thus the fluids are static in time zero. The air phase fills the domain by default and a block of Liquid L1 and L2 are created so that they simulate the conditions of depth, density and temperature used in the experiments.

*Control settings*

The simulation time is set to 20 s to obtain enough information for the comparison with the experimental data. The intermittent states are saved every 0.1 s to visualize a smooth transition from one time step to another. The Courant number is also defined considering the expression:

$$Co = \frac{\delta t |U|}{\delta x}$$

Where $\delta t$ is the time step, which is 0.001 s for this case; and $\delta x$ is the cell size in the direction of the velocity $U$. The Courant number is a dimensionless parameter that indicates the maximal time step allowed to achieve temporal accuracy and convergence while solving the partial differential equations system numerically. For this case, a Courant number lower than 1.0 is required, thus the maximal Courant number is set to 0.25, and the maximal time step is set to 0.01 s, so convergence within the modeling of the phenomenon can be achieved with a suitable time stepping.

Data of velocity vectors, pressure and temperature are extracted from numerical probes set across the numerical domain, which are located in the same points of the experimental model, i.e., ten numerical probes points.

Since thermistors occupy a certain surface instead of a specific point in space, five numerical probes represent one thermistor. The values given by the five probes are averaged to represent the value given by the thermistor in the experiments. These probes are distributed on each of the ten points as follows: one in the center, one on each superior corner and two in the inferior corners, as schematically represented in Figure 19. Thus, the distribution of the probes forms a $1\ \mathrm{cm}^2$ square.

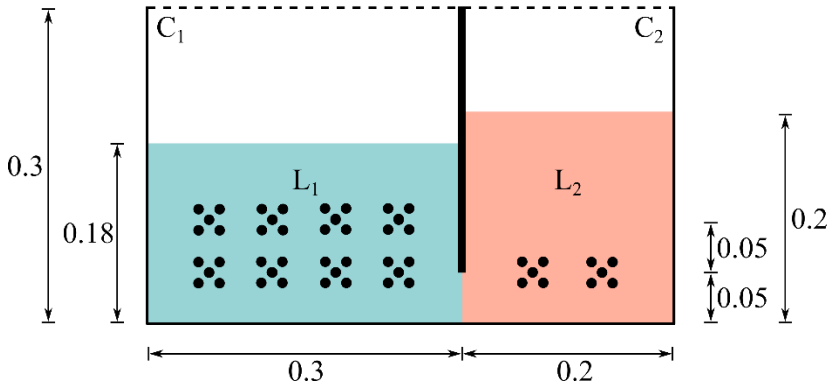


*Figure 19. Distribution of the five numerical probes for each of the ten thermistors.*

*Solution settings*

The solution settings and solution schemes for the Finite Volume OpenFOAM® approach are the same used in any `interMixingTemperatureWaveFoam` case setting. A detailed description of these settings can be found in Appendix B.

## 6.3. COMPARISON OF THE EXPERIMENTAL RESULTS AGAINST THE CFD MODEL RESULTS

In this section, the experimental data are compared against the numerical results with `interMixingTemperatureWaveFoam`. The objective is to obtain the numerical error caused by, among other factors, the simplification of the equations, spatial and temporal discretization,

### 6.3.1. Experimental results

Ten thermistors recorded the temperature of the liquid fluids during 44 s. The experiment was repeated three times so that the averaged fluid flow behavior can be compared to the numerical results.

A table of time steps (in s) and temperature value (in °C) is obtained from each thermistor. Since the recording time starts before the opening of the gate, each file is modified by eliminating the unnecessary data before the study time.

The response of the thermistors (i.e., time resolution) is not fast enough to capture the fast temperature variations during the transient phase of the experiment, which occurs in the first twenty seconds. Therefore, the first 20 seconds are not considered in the validation test and only the last 24 seconds, where the fluid flow and the temperature tend to stabilize, are taken into account.

Figure 20 shows the raw data obtained by the thermistors during the experiments. The thermistor signal has noise; thus, a low-pass filter is used to smooth these disturbances (Figure 21).

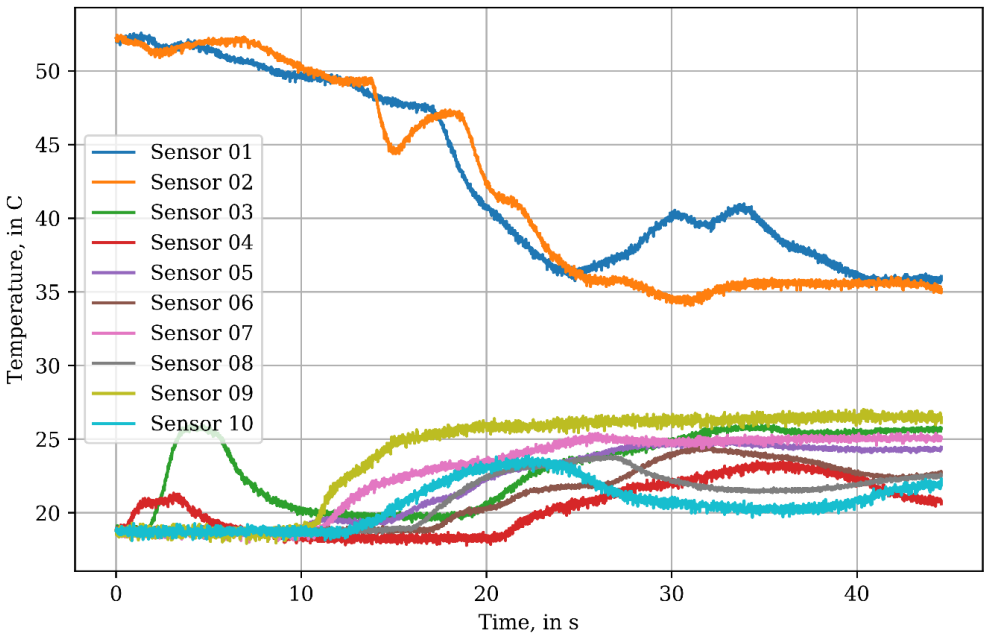


*Figure 20. Raw data obtained by the thermistors.*

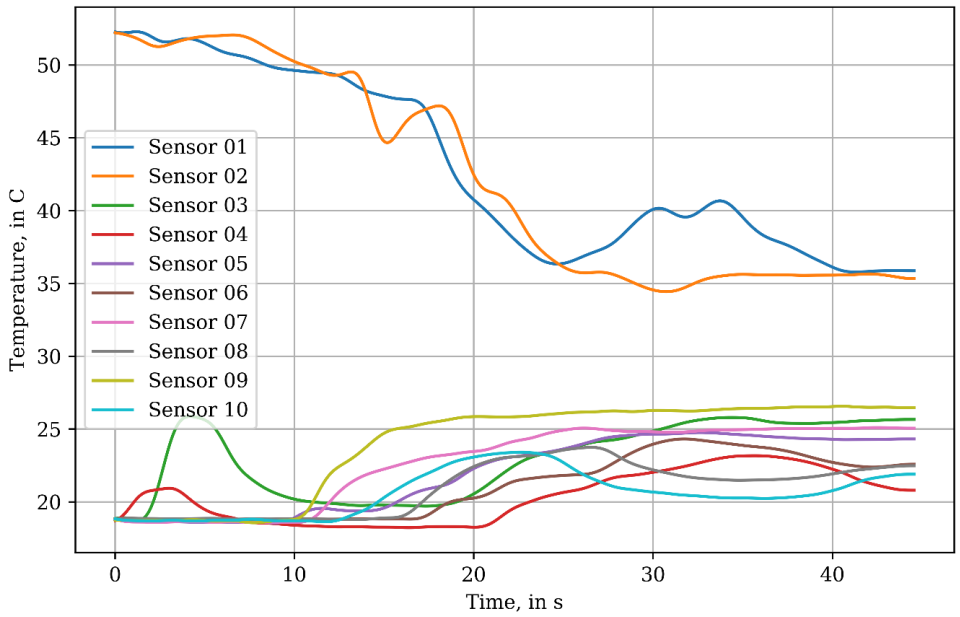


*Figure 21. Temperature signal after the low-pass filter is applied*

### 6.3.2. Numerical results

After the experiments are carried out, the measured data of the initial temperature and density of the liquid phases is introduced in the case numerical setup. The case is run for three turbulence models: zero-equation Reynolds-Averages Navier-Stokes (RANS) turbulence model, two-equation (k-$\omega$) RANS model, and Large Eddy Simulation (LES) model.

Figure 22 shows the qualitative comparison between the three turbulence models and the results of the temperature field on certain time instants of the simulations. Since the distribution on the temperature field depends also on the distribution of the density field as the mixing takes place, the turbulence model has a great impact on these results.

It can be observed in Figure 22 that the temperature differences are more “stratified” with the two-equation (k-$\omega$) turbulence model, i.e., the mixing is less evident as in zero-equation and LES models. The temperature in C1 is lowest in the zero-equation model and the mixture is the most uniform.

It is also noticeable that after 20 seconds the fluid mixing tends to stabilize. As a consequence, the temperature gradually becomes stratified and uniform in certain regions above and below the gate aperture.

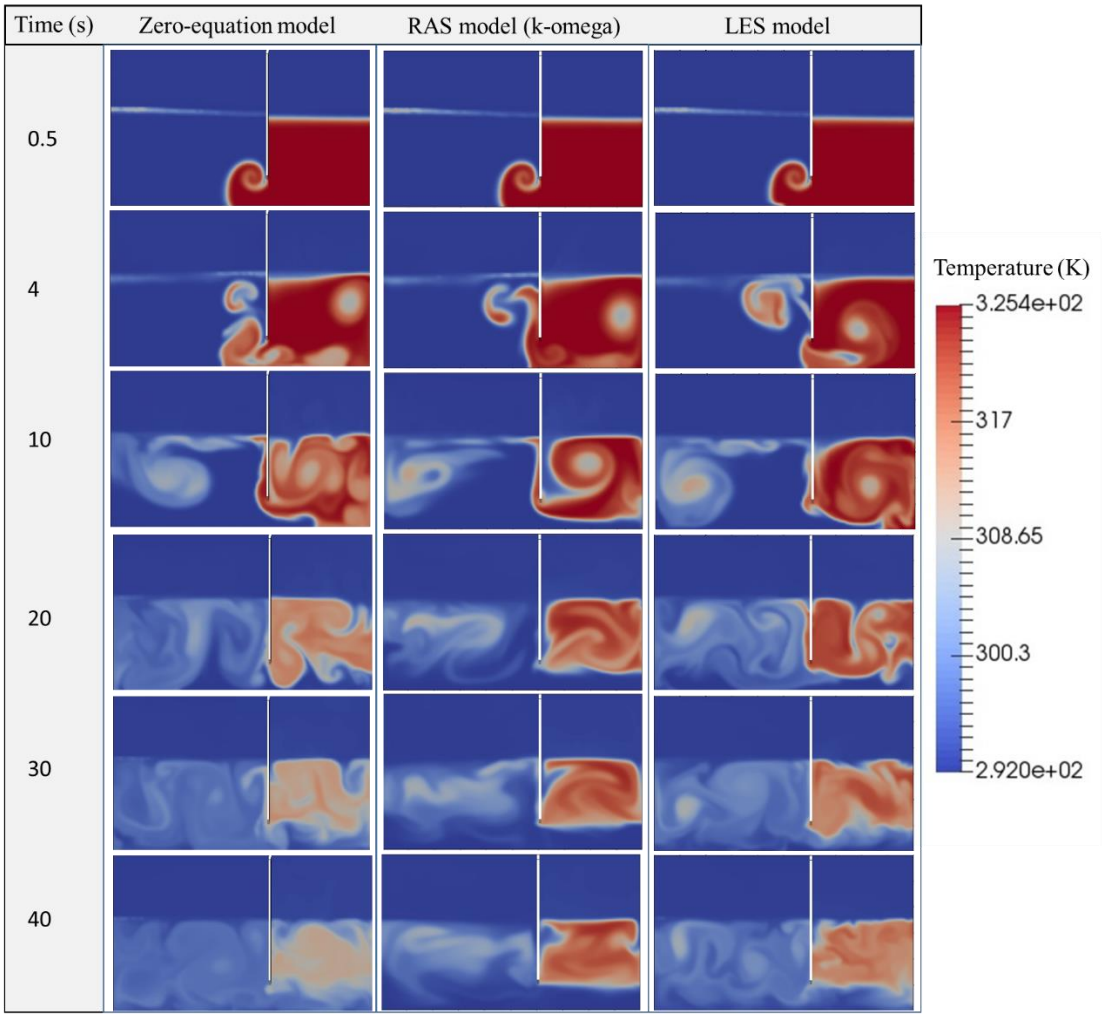


*Figure 22. Temperature field for different turbulence models.*

For each turbulence model, the temperature is captured by 50 numerical probes that represent the 10 thermistors. Figure 23, Figure 24, and Figure 25 show the temperature against time for the three turbulence models simulated.

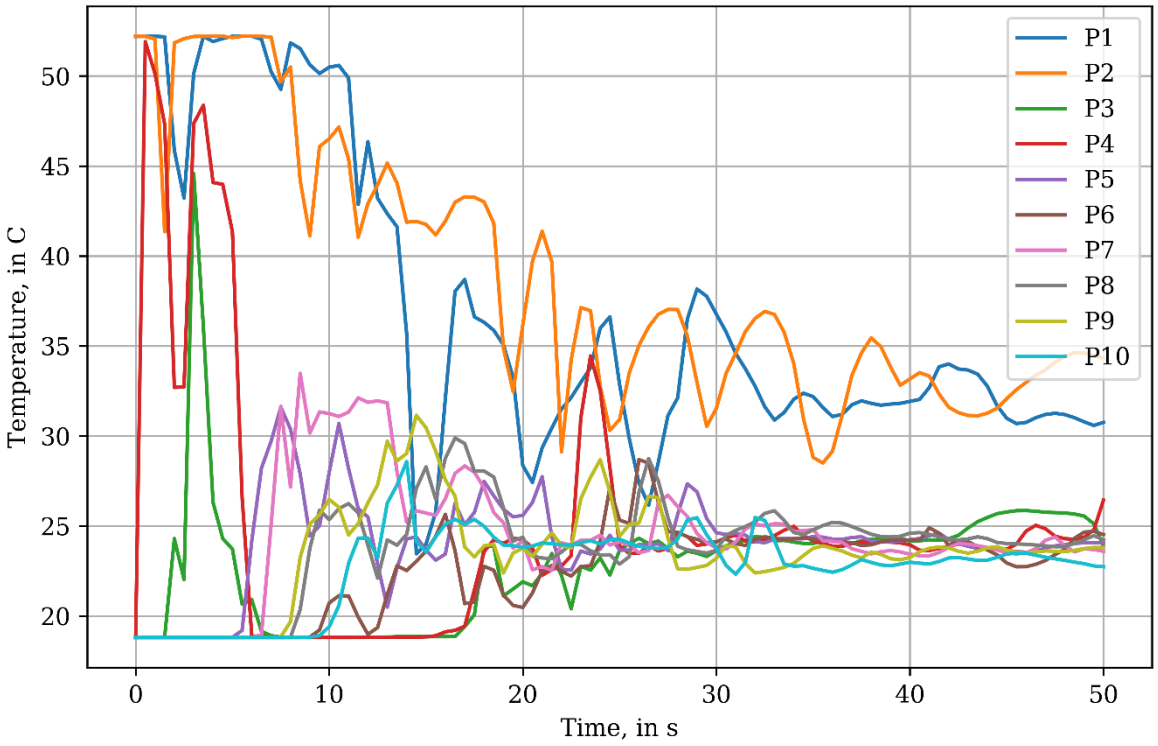


*Figure 23. Zero-equation turbulence model results of temperature for each set of numerical probes.*

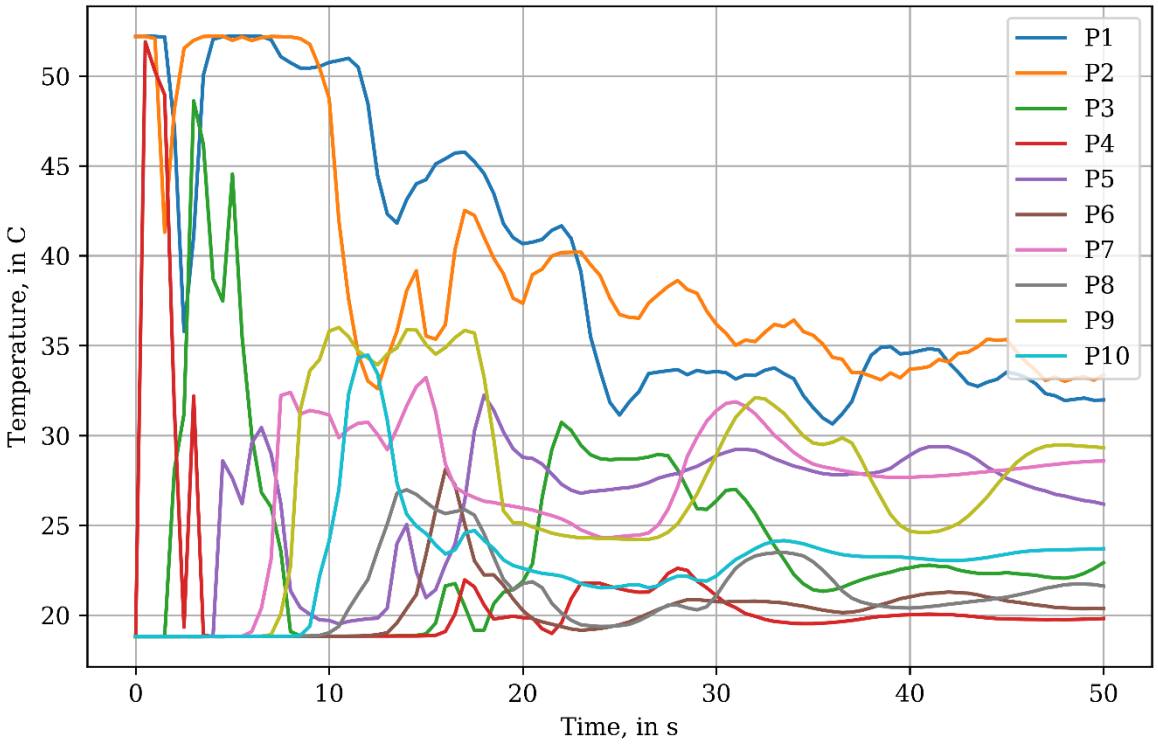


*Figure 24. Two-equation (k-ω) turbulence model results of temperature for each set of numerical probes.*

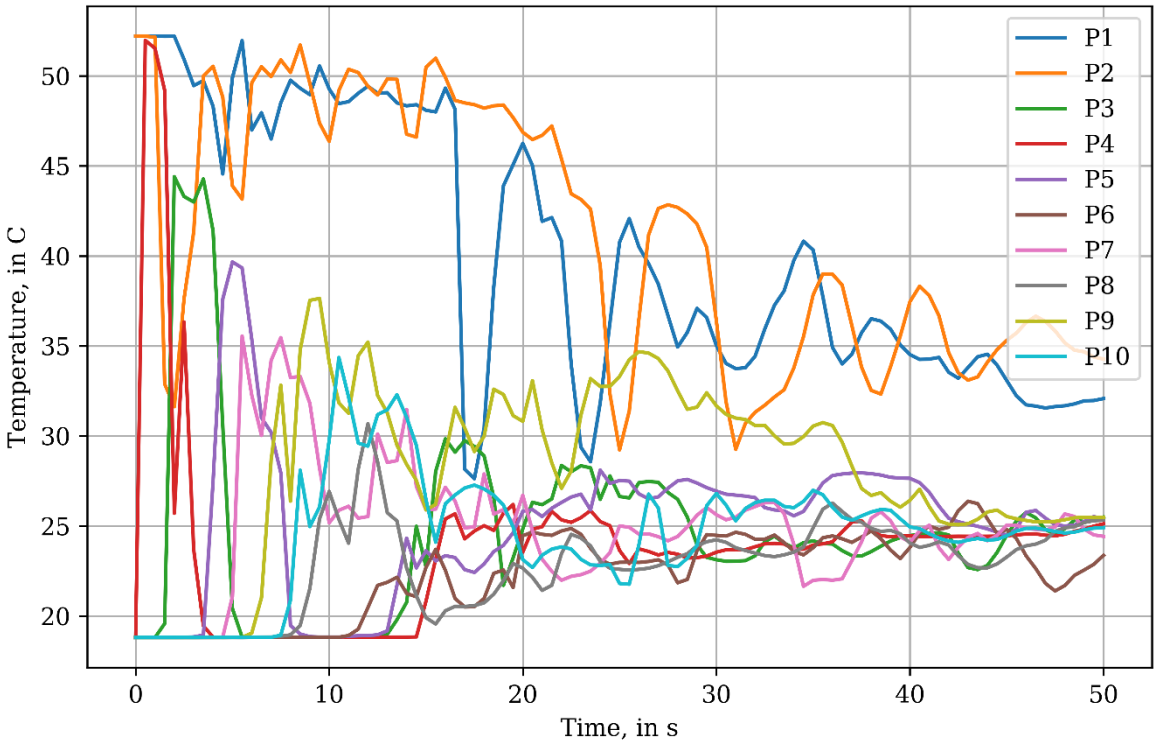


*Figure 25. LES turbulence model results of temperature for each set of numerical probes.*

### 6.3.3. Mean relative error

Experimental data (after the low-pass filter) is used to calculate the relative mean errors of the developed solver. For this purpose, each thermistor signal is compared against the signal of the ten analogous sets of numerical probes. For each turbulence model the relative error $E_i$ is estimated as in Equation 40 for each probe set:

$$E_i = \frac{|T_E - T_M|}{\Delta T} * 100 \qquad \text{Equation 40}$$

where $T_E$ is the temperature in the experiments, in °C, $T_M$ the temperature of the numerical simulation, in °C, and $\Delta T$ the temperature difference between the two phases in time zero, for this case $\Delta T = 33.4\,°C$. This error is estimated for the data between 20 s and 44 s where the temperature tends to stabilize and gradually becomes time-independent.

Figure 26, Figure 27, and Figure 28 show the error for each of the ten probes sets and the mean error for each time step. E1-E10 are the error series for each numerical probe comparing P1-P10 against T1-T10, respectively.

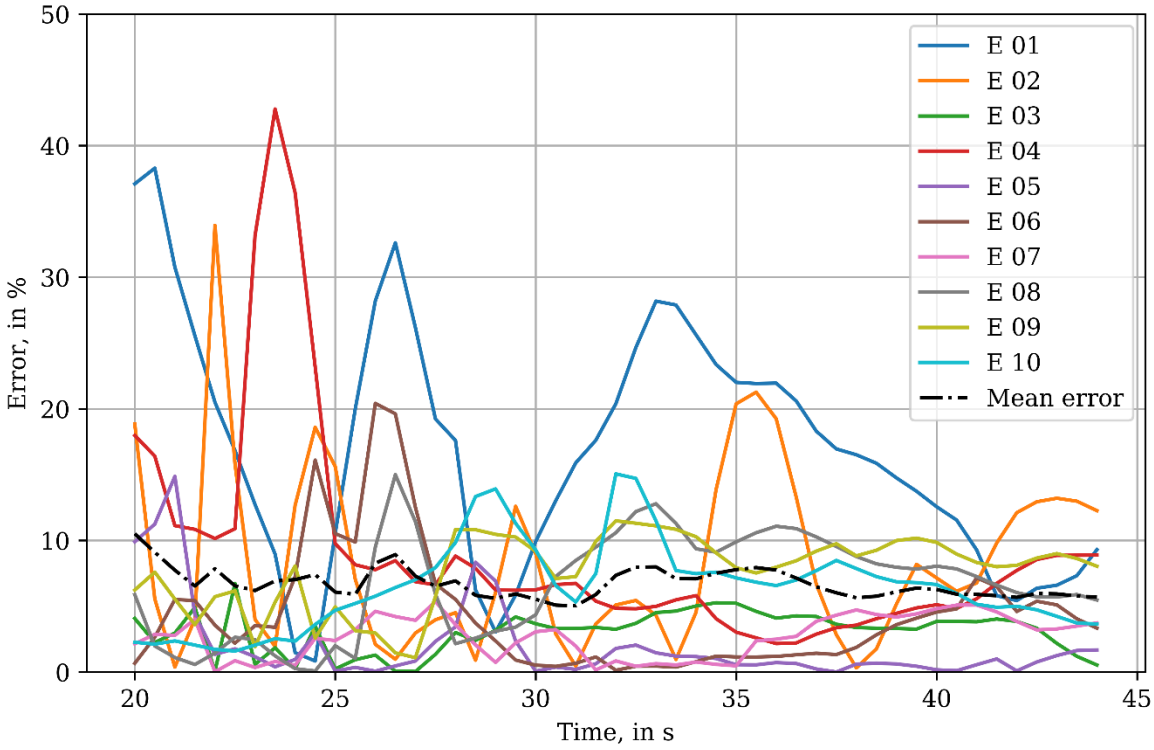


*Figure 26. Zero-equation turbulence model relative error.*

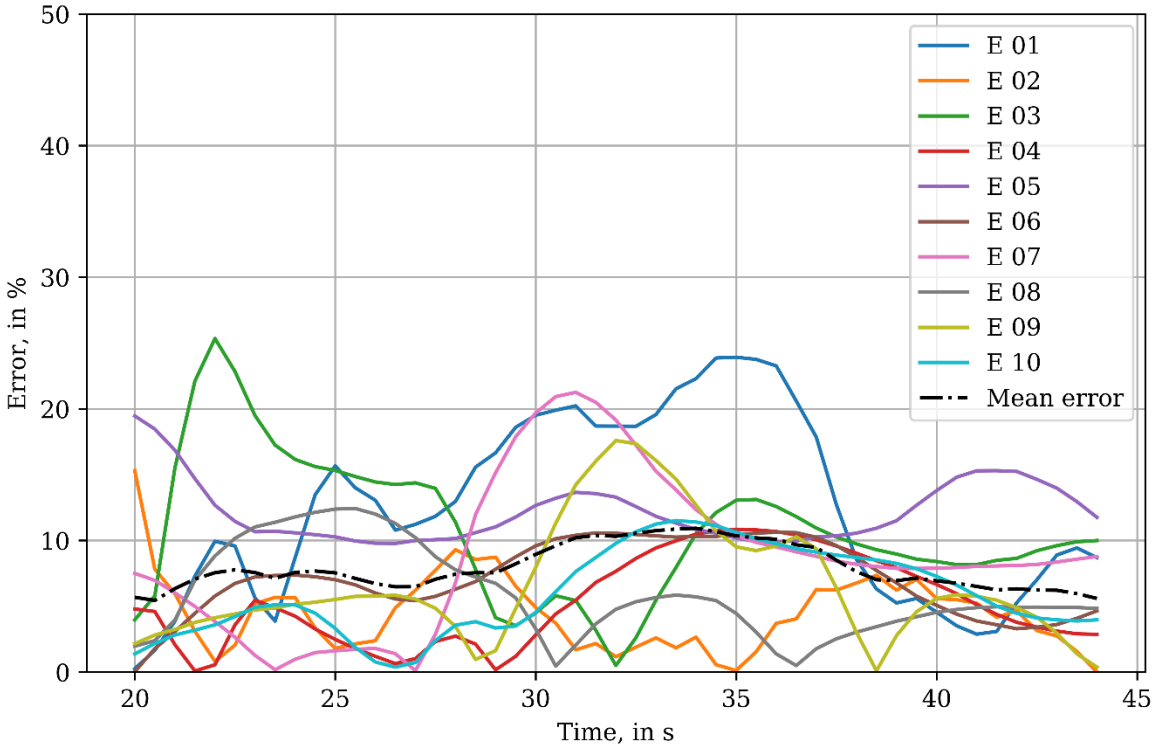


*Figure 27. Two-equation (k-ω) turbulence model relative error.*

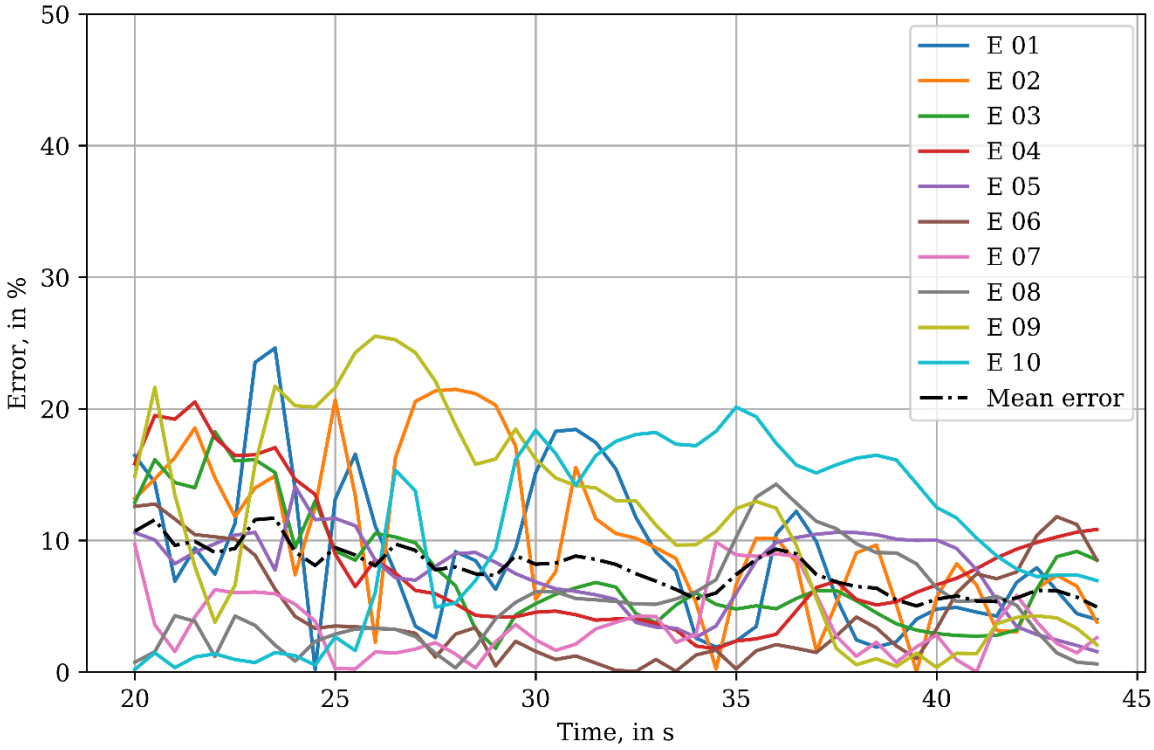


*Figure 28. LES turbulence model relative error.*

Figure 29 compares the mean relative error through time for the three turbulence models tested. Zero-equation model gives the lowest mean error through time, while the LES model gives the highest error before t= 30 s, and two-equation (k-$\omega$) model yields the highest error after t= 30 s.

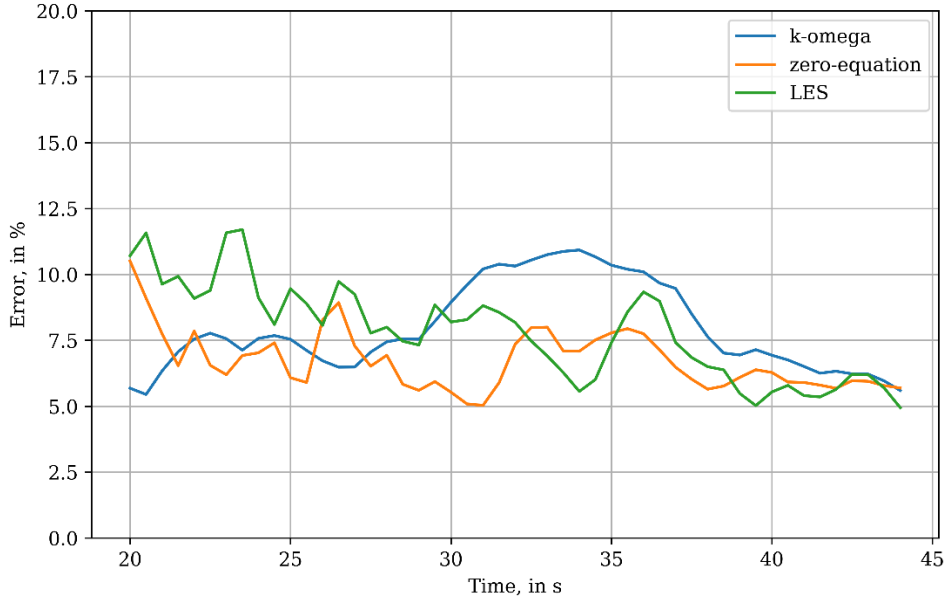


*Figure 29. Comparison of the mean error through time for the three turbulence models tested.*

On the other hand, the mean relative error for each probes set is averaged in Table 14. The lowest averaged relative error per sensor are marked in blue, while the highest in orange.

*Table 14. Averaged relative error, in %, of the ten probes sets and the three turbulence models tested.*

| | Averaged Relative Error | | |
|---|---|---|---|
| Sensor | Zero-Equation model | Two-equation (k-$\omega$) RANS model | LES model |
| 1 | 16.86 | 12.42 | 8.90 |
| 2 | 8.48 | 4.46 | 10.49 |
| 3 | 3.06 | 10.72 | 7.54 |
| 4 | 9.04 | 5.29 | 8.12 |
| 5 | 1.97 | 12.18 | 7.73 |
| 6 | 4.37 | 7.09 | 4.48 |
| 7 | 2.71 | 8.79 | 3.63 |
| 8 | 6.58 | 5.90 | 5.18 |
| 9 | 7.73 | 6.63 | 11.59 |
| 10 | 6.48 | 5.91 | 10.42 |

The Zero-equation turbulence model has the highest mean error in E1, E4 and E8 error series; nevertheless, E3, E5, E6, and E7 give the lowest mean error, while E2, E9, and E10 sow intermediate values compared to the two other turbulence models. Two-equation (k-$\omega$) model shows four of the lowest mean errors but also four of the highest.

Sensor 1 and 2 are located in compartment C1, which was filled with heated water. Zero-equation model produce the highest errors in this C1, while it produces the lowest errors in C2. As observed in Figure 22, this model produces the most uniform mixture, when compared against Two-equation (k-$\omega$) and LES models. During the experiments, it was clear that the flow tends to form defined layers of heated and warm water in C1. Thus, it is expected that zero-equation model has higher errors.

Independently of the differences between the turbulence models, none of them exhibit a mean error higher than 17 %; thus, the behavior of the energy equation in terms of temperature is considered good enough to simulate the temperature field and the temperature equilibrium between liquid phases.

### 6.3.4. Absolute error and maximum difference

The absolute value of the difference between the temperatures measured in the experimental model and the temperatures given by the analogous numerical probes is obtained. Then, the maximum difference, in °C, is obtained for each probes sets (Table 15).

For the zero-equation model, the maximum absolute error corresponds to sensor 4, with 14.29 °C. In the Two-equation (k-$\omega$) model, the maximum absolute error corresponds to sensor 3, with 8.47

°C. In the LES model, the maximum absolute error corresponds to sensor 9, with 8.53 °C. Even though the zero-equation model shows the highest absolute error in one of the sensors, it has the lowest mean absolute errors.

*Table 15. Absolute errors, in °C.*

| | Absolute error | | | | | |
|---|---|---|---|---|---|---|
| | Zero-Equation model | | Two-equation (k-$\omega$) RANS model | | LES model | |
| Sensor | Mean absolute error | Maximum difference | Mean absolute error | Maximum difference | Mean absolute error | Maximum difference |
| 1 | 5.63 | 12.79 | 4.15 | 7.99 | 2.97 | 8.22 |
| 2 | 2.83 | 11.34 | 1.49 | 5.11 | 3.50 | 7.18 |
| 3 | 1.02 | 2.25 | 3.58 | **8.47** | 2.52 | 6.10 |
| 4 | 3.02 | **14.29** | 1.77 | 3.62 | 2.71 | 6.86 |
| 5 | 0.66 | 4.97 | 4.07 | 6.50 | 2.58 | 4.71 |
| 6 | 1.46 | 6.82 | 2.37 | 3.55 | 1.50 | 4.26 |
| 7 | 0.90 | 1.79 | 2.93 | 7.10 | 1.21 | 3.30 |
| 8 | 2.20 | 5.01 | 1.97 | 4.15 | 1.73 | 4.77 |
| 9 | 2.58 | 3.84 | 2.21 | 5.88 | 3.87 | **8.53** |
| 10 | 2.16 | 5.03 | 1.97 | 3.84 | 3.48 | 6.73 |
| Mean Value | 2.25 | 6.81 | 2.65 | 5.62 | 2.61 | 6.06 |

### 6.3.5. Root mean squared errors

To strengthen the validation process, the Root-Mean-Square errors (RMSE), in °C, are calculated for each probes sets, as shown in Table 16. Consistently with the mean absolute errors, the zero-equation model shows the lowest mean errors

*Table 16.Root mean squared errors.*

| | RMSE | | |
|---|---|---|---|
| Sensor | Zero-Equation model | Two-equation (k-$\omega$) RANS model | LES model |
| 1 | **6.39** | 4.74 | 3.55 |
| 2 | 3.65 | 1.77 | **4.02** |
| 3 | 1.15 | **3.99** | 2.89 |
| 4 | **4.10** | 2.09 | 3.21 |
| 5 | 1.21 | **4.14** | 2.77 |
| 6 | 2.12 | **2.54** | 1.98 |
| 7 | 1.05 | **3.51** | 1.50 |
| 8 | **2.54** | 2.27 | 2.08 |

| | | | |
|---|---|---|---|
| 9 | 2.74 | 2.67 | **4.65** |
| 10 | 2.43 | 2.25 | **4.12** |

Consistently with the previous results, zero-equation model shows the highest errors for the two sensors located in C1, while errors in C2 are the lowest compared against the other turbulence models.

## 6.4. SUMMARY

In this chapter, the validation of the `interMixingTemperatureWaveFoam` CFD model was conducted. The experimental and numerical results were contrasted through the calculation of the mean relative error, the absolute error and maximum difference, and the RMSE.

The study case to perform validation consisted in a wet-dam-break case, divided into compartment 1 and 2 (C1 and C2), where C1 contained water at room temperature, and C2 heated water. The experimental facility was instrumented with ten thermistors, whereas the numerical case temperature results were tracked through virtual probes distributed analogously to the thermistors. The numerical simulations were run with three turbulence models: zero-equation RANS turbulence model, two-equation (k-$\omega$) RANS model, and LES model.

On the one hand, mean relative errors for each probe point resulted lower than 17 % (Table 14), while mean relative errors through time were lower than 12.5 % for each turbulence model (Figure 29). On the other hand zero-equation model showed a maximum absolute error of 14.29 °C, two-equation (k-$\omega$) model of 8.47 °C, and LES model of 8.53 °C. Even though the zero-equation model shows the highest absolute error in one of the sensors, it has the lowest mean absolute errors. Finally, RMSE estimations showed that zero-equation model has the highest errors in C1, whereas errors in C2 were the lowest compared against the two other turbulence models. Thus, the behavior of the energy equation in terms of temperature is considered good enough to simulate the temperature field and the temperature equilibrium between liquid phases.

# 7

# IMPLEMENTATION OF THE CFD MODEL TO AN OTEC DISCHARGE CASE

# 7. IMPLEMENTATION OF THE CFD MODEL TO AN OTEC DISCHARGE CASE

This section describes the solver implementation once its error is estimated in the validation process, in terms of the temperature field distribution. Thus, the application of `interMixingTemperatureWaveFoam` solver to OTEC discharge water in coastal waters is demonstrated.

To illustrate the features and the potential of the developed numerical model, a case is set up based on the parameters obtained in the conceptual design of an OTEC plant in Mexico (see Chapter 3). If necessary, characteristics of bathymetry, water temperature, and real wave conditions can be adapted to the specific site to be studied with this CFD tool.

## 7.1. CASE DESCRIPTION

The demonstration case (Figure 30) consists of a three-dimensional domain of 80 x 80 x 50 meters in the X, Y and Z directions respectively. It simulates OTEC thermal water discharge at a certain depth through a buried pipe in the ocean floor.

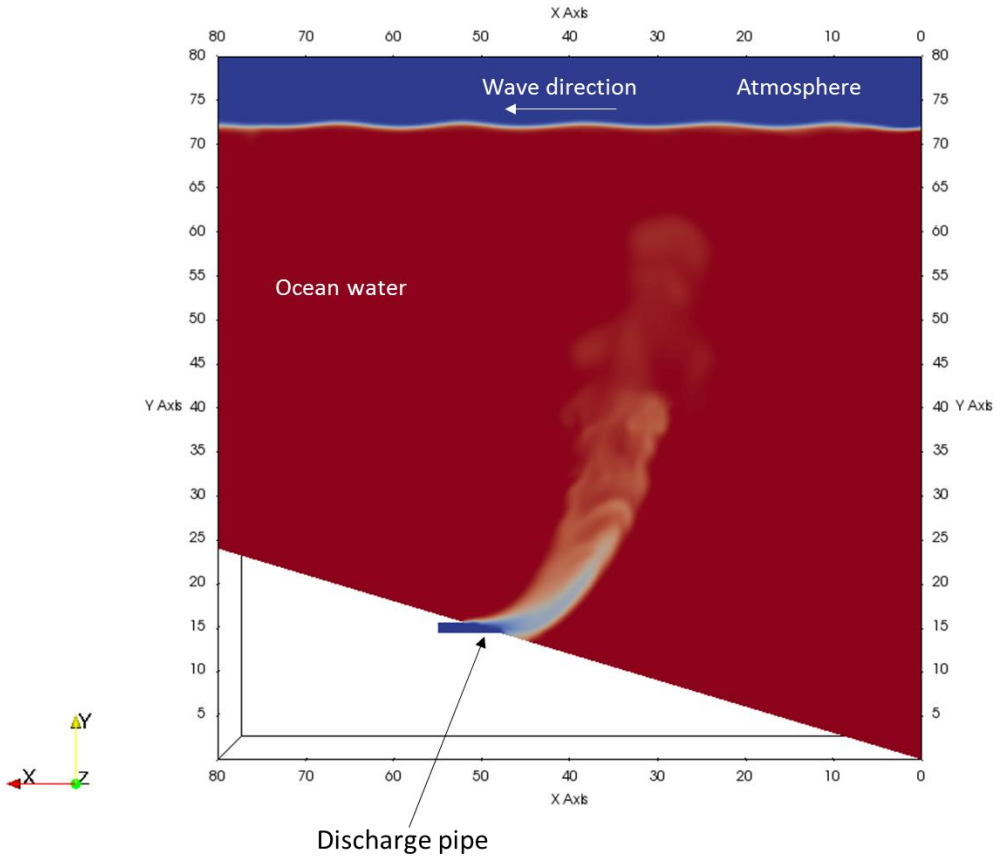


*Figure 30. Schematic representation of the demonstration case of OTEC thermal water discharge.*

The pipe discharges at a depth of 60 m, which is the recommended discharge depth according to different authors (see Chapter 1). Based on the conceptual design of an OTEC plant in Mexico in Chapter 3, the pipe has a diameter of 1.2 m, and it discharges at a flow rate of 1.5 $m^3/s$, which is translated into a horizontal velocity of 1.33 m/s. Furthermore, the water discharges with a temperature of 11.8 °C and each phase is given a density that corresponds to its salinity and temperature.

As a preliminary demonstration and, due to the lack of detailed bathymetry data available, the ocean floor is simply represented with a flat ramp. The turbulence model is set as zero-equation since this model is proven to yield adequate results of dilution and eddy-formation in liquid-phases (Rodríguez-Ocampo, 2016). The wave parameters are also preliminary set for a regular wave train.

## 7.2. FEATURES AND CAPABILITES OF THE MODEL

`interMixingTemperatureWaveFoam` solver allows to introduce the temperature field as temperature blocks or layers to simulate the real temperature stratification in the ocean water, as shown in Figure 31.

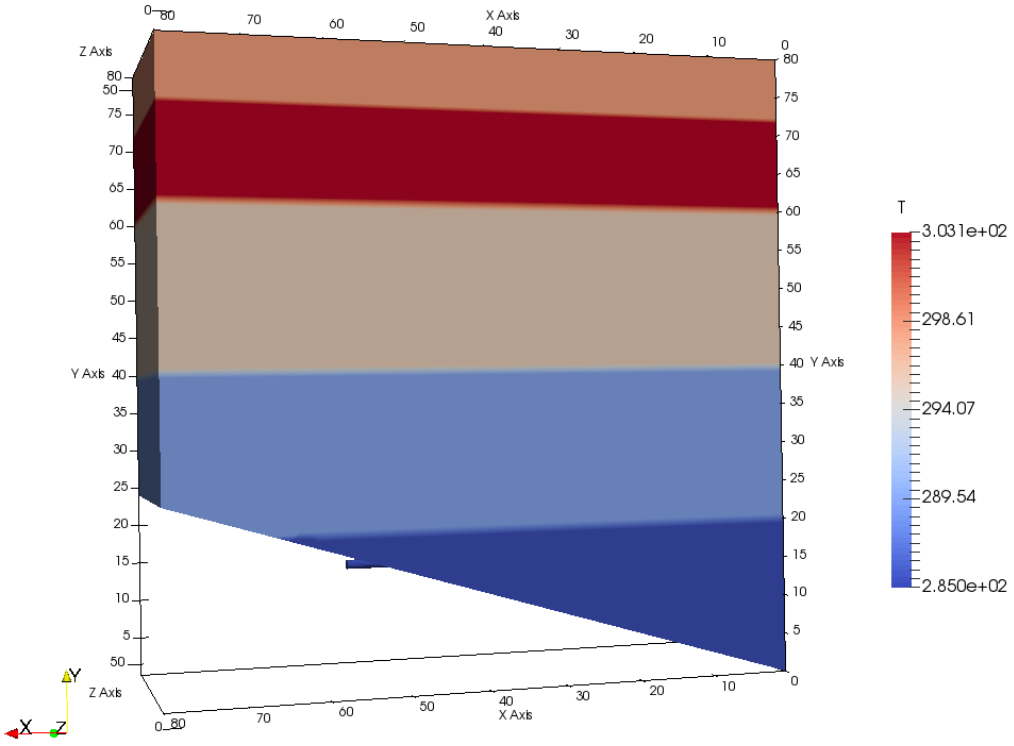


*Figure 31. Temperature layers that simulate the temperature gradient in the ocean profile.*

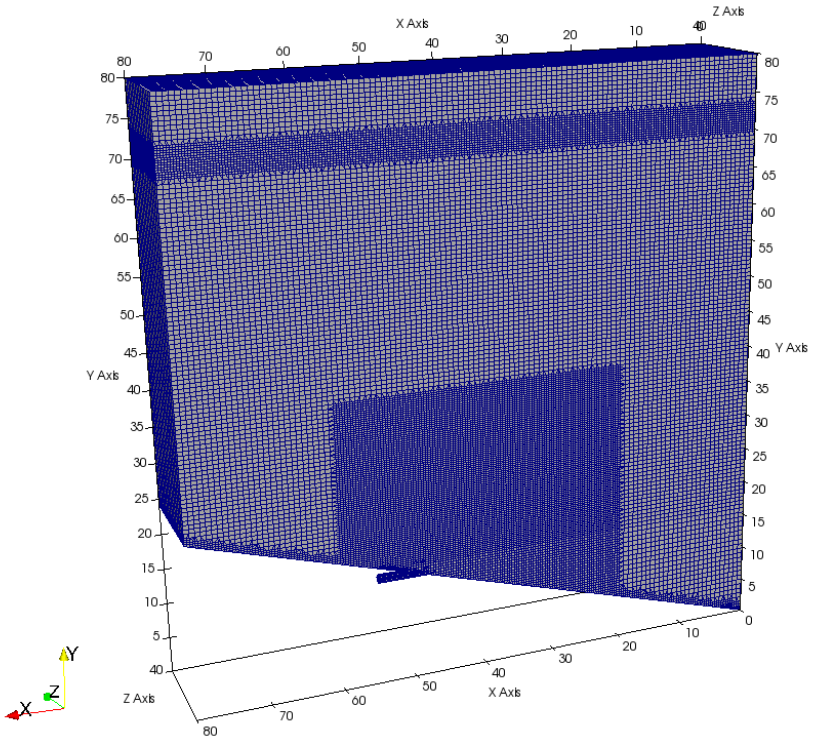


*Figure 32. Refinement of the mesh in the vicinity of the discharge pipe and the free-surface of the ocean.*

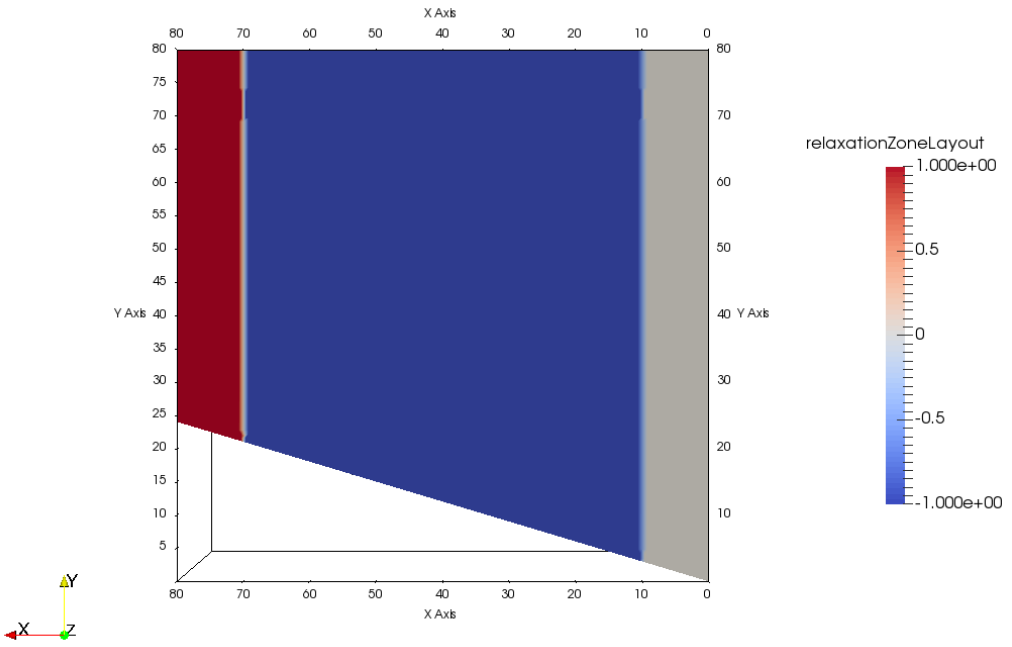


*Figure 33. Relaxation zones layout. Gray represents the wave-generation zone and red the wave-absorption zone.*

The mesh can be also refined in the zones where more precision is required, e.g., the zone near to the discharge plume, and the zone that surrounds the free-surface (Figure 32). The refinement in the non-structured mesh allows more efficient running times, without losing detail in the zones of interest. However, a full 3-dimensional case may require significant running time and appropriate computational resources, preferably a cluster.

Since `waves2foam` is contained in the presented `interMixingTemperatureWaveFoam` solver, waves and currents are simulated through relaxation zones; i.e., one relaxation zone for the wave generation and one for the wave absorption (Figure 33).

Different conditions and parameters can be straightforwardly introduced to simulate different scenarios in a preliminary design phase of an OTEC project; thus, contributing to the feasibility evaluation of the implementation of this technology in Mexico.

## 7.1. INFLUENCE ZONE AND AVERAGE TEMPERATURE VARIATIONS

A demonstrative case is run with uniform sea temperature (Figure 34) and 100 s of simulation time (Figure 35). This case demonstrates the capability of the developed `interMixingTemperatureWaveFoam` model to simulate in a detailed fashion OTEC discharge water in the ocean.

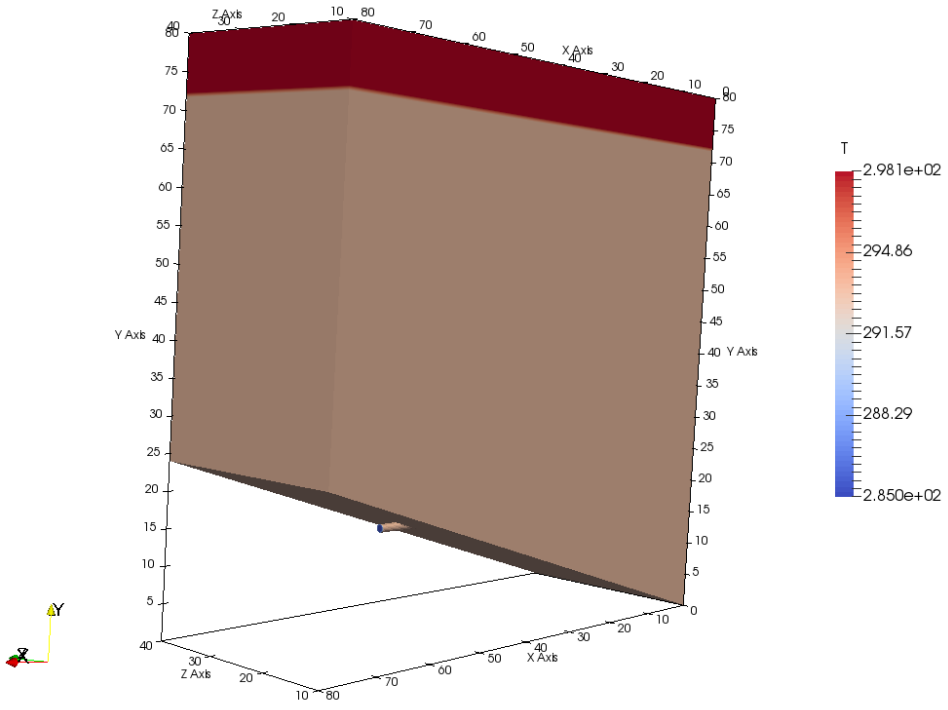


*Figure 34. Temperature field in time zero.*

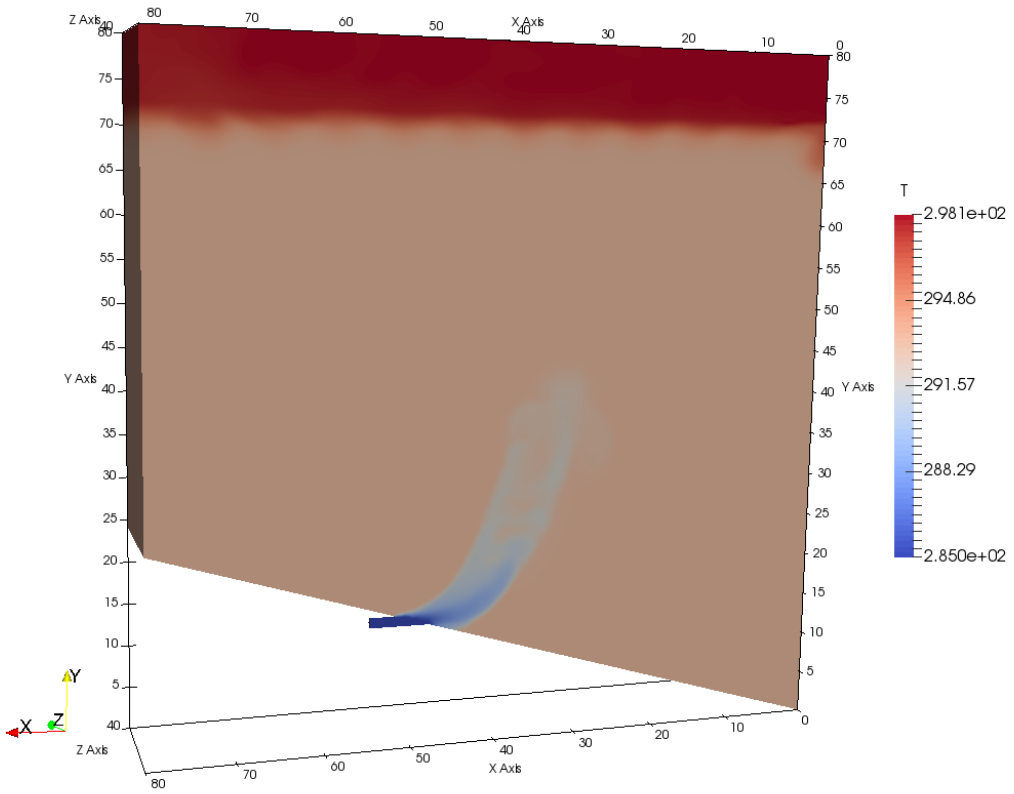


*Figure 35. Temperature field after 100 s.*

Five cases were tested, each one with a different discharge temperature, as described in Table 17.

*Table 17. Discharge temperature for the tested cases.*

| Case | Discharge temperature (°C) |
|---|---|
| 1 | 9 |
| 2 | 10 |
| 3 | 11 |
| 4 | 12 |
| 5 | 13 |

Table 18 summarizes the most important parameters introduced in the cases. These parameters remain constant for the five tested cases. Therefore, each case only differs in the discharge temperature.

*Table 18. General input parameters for the tested cases.*

| | Gravitational acceleration | Submarine discharge (1.2 m diameter) |
|---|---|---|
| Direction | a (m/s²) | Velocity v (m/s) |
| x | 0 | -1.33 |
| y | -9.81 | 0 |
| z | 0 | 0 |

| Phase properties | | | |
|---|---|---|---|
| Phase property | Phase 1 (air) | Phase 2 (water) | Phase 3 (water) |
| Density (kg/m³) | 1 | 1030 | 1000 |
| Kinematic viscosity (m²/s) | 1.48x(10)^(-5) | 1x(10)^(-6) | 1x(10)^(-6) |
| Surface tension (N/m) | 0.05 | | |
| Molecular diffusion coefficient (m²/s) | | 1.26x(10)^(-9) | |
| Temperature (K) | 298 | 308 | *Varying* |
| Specific heat capacity (J/kg*K) | 1045.2 | 1433.4 | 1433.4 |
| Prandtl number (1) | 0.87 | 3.37 | 3.37 |

| Wave properties | |
|---|---|
| Type | Regular wave train |
| Wave theory | Linear |
| Period T (s) | 4 |
| Wave height H (m) | 0.1 |
| Depth h (m) | 2 |
| Simulation time (s) | 30 |

The influence zone is defined by the affected volume after 100 s of simulation. At this time the discharge plume has an approximated extension of 12 m in a vertical plane and an influence area between the coordinates (30, 12, 25) and (42, 42, 25). A total of 874 probes points are located within this area to measure the temperature. Table 19 shows the mean temperature in the influence zone, the standard deviation, and percentile values, for each of the 5 considered cases.

*Table 19. Mean temperature in the influence area for different discharge temperatures.*

| Case | Discharge temperature (°C) | Mean Temperature (°C) | Standard deviation | Percentile 5 | Percentile 20 | Percentile 50 |
|---|---|---|---|---|---|---|
| 1 | 9 | 28.43 | 2.21 | 23.64 | 27.61 | 29.14 |
| 2 | 10 | 28.51 | 2.11 | 23.94 | 27.72 | 29.18 |
| 3 | 11 | 28.58 | 2 | 24.24 | 27.84 | 29.22 |
| 4 | 12 | 28.66 | 1.89 | 24.54 | 27.95 | 29.26 |
| 5 | 13 | 28.73 | 1.79 | 24.85 | 28.06 | 29.3 |

Figure 36 shows a linear tendency on the temperature increments of the influence zone. A linear regression yields a value of $r^2$=0.9995, which indicates a good correlation between the predictions and the numerical results.

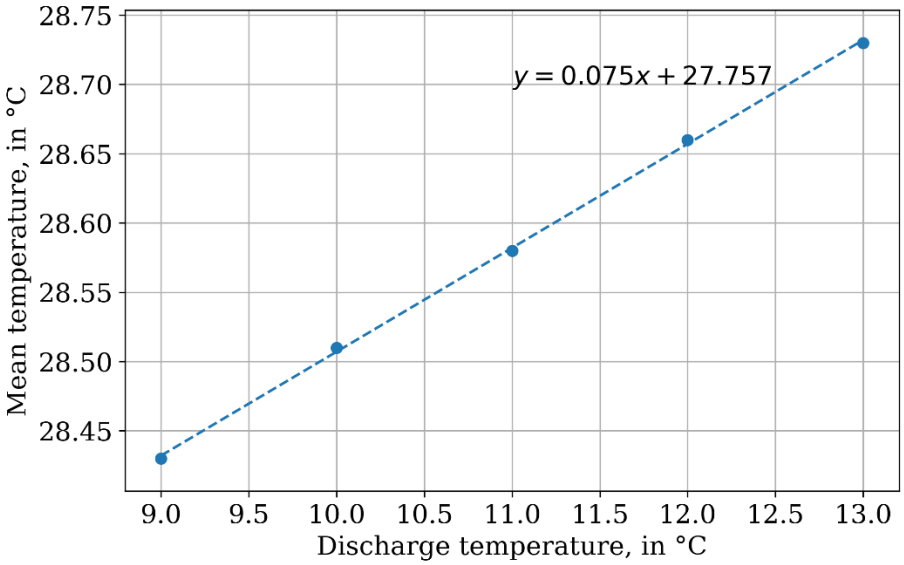


*Figure 36. Mean temperature variations in the influence area.*

In this Chapter it is demonstrated the applicability of the developed `interMixingTemperatureWaveFoam` solver to simulate the discharge plume of OTEC plants under different conditions of temperature, density, and wave/currents field. Naturally, this solver is not limited to OTEC facilities but it can be worthwhile in cases where there are two liquid fluid-phases in a free-surface condition.

# 8

# CONCLUSIONS

# 8. CONCLUSIONS

## 8.1. OTEC TECHNOLOGY

Ocean Thermal Energy Conversion (OTEC) is a technique for extracting renewable energy by utilizing the temperature difference that exists between the ocean water layers to operate an engine. The low efficiency of OTEC systems results in large facilities, which represent great technological challenges and potential environmental impact.

One of the major limitations of OTEC implementation in the world is the location of suitable places, with the appropriate conditions of the thermal resource at reasonable distances from the shore or where the energy is going to be harnessed. Mexico's geographical position makes it a potential site for the exploitation of the thermal gradient resource.

Nowadays, OTEC technology implementation is being evaluated, particularly in the South Pacific Ocean. Small prototypes have been proposed and are under study to determine the feasibility of deploying one OTEC plant on the Mexican coasts.

The economic analysis has determined that, to make OTEC economically viable for commercial purposes, the capacity of one OTEC facility must approach 100 MW. A facility of such size represents great technological challenges related to the design, construction, and installation of major components such as CWP, platforms, CWP/platform interphase, heat exchangers, pumps, turbines, and power cable. Nevertheless, current technology and current research in OTEC systems have been developed for much smaller scales (< 1 MW). Therefore, further research, modeling, and testing are required.

## 8.2. IMPORTANCE OF CFD NUMERICAL MODELING IN OTEC PROJECTS

Even though the potential environmental impacts of OTEC technology are relatively small if compared with conventional energy generation plants, the continuous discharge of large amounts of cold, nutrient-rich, and bacteria-free deep ocean water is the biggest concern related to the environmental effect of commercial-scale plants operation (> 10 MW).

The experience with the operation, monitoring and numerical simulation of small OTEC plants ($\leq$ 1 MW) shows that thermal water discharge is environmentally acceptable at a depth of 60 m. However, the environmental impact is less clear when it comes to large OTEC facilities that would mobilize great water quantities and would use large pipes (up to 10 m diameter).

The lack of experience in the continuous operation of commercial-scale plants makes it hard to predict the real impact on the ocean ecosystem and coastal physical processes. Additionally, since the mixing and dispersion characteristics depend on the specific site of interest, as well as the dimensions of the plant, it is not possible to establish a generic discharge depth. Thus, each site must be individually evaluated with actual field measurements and different tools such as numerical modeling of fluid dynamics and the ecosystem’s response.

This work provides a free and open-source CFD numerical tool that can help to evaluate, in a detailed fashion, the behavior of the plume of an OTEC plant's discharge water. By simulating different configurations of water discharge, it is possible to make better design decisions and minimize the negative environmental impact of large OTEC plants.

## 8.3. DEVELOPED CFD MODEL

After comparing different available CFD software, OpenFOAM® was selected as the most appropriate CFD package to simulate thermal multiphase fluid flow associated with OTEC discharge water. Following the solver's name convention in OpenFOAM®, the developed solver was named `interMixingTemperatureWaveFoam`. The governing equations implemented in `interMixingTemperatureWaveFoam` solver can be synthetized as: momentum conservation equation, mass conservation equation, scalar quantities conservation equation, volume fraction equation, and energy conservation equation in terms of the temperature field.

The model is capable of simulating thermal water discharge of an OTEC facility after its use in the energy conversion process, providing another approach to previous studies with different numerical and analytical tools. The development, validation, and implementation of this tool is also valuable for further applications.

### 8.3.1. VALIDATION OF THE CFD MODEL

Before implementing the solver in a demonstrative case for its main application (i.e., OTEC water discharge), it was validated with a benchmarking test. Different authors previously performed validation of the density field distribution, and validation of the wave generation-absorption module. Since wave generation/propagation is not considered in the validation case, this process only determined the accuracy of the temperature field distribution; and therefore, the energy equation implementation in OpenFOAM®.

Laboratory benchmark tests of a wet-dam-break type in a rectangular installation were conducted to evaluate the validity of `interMixingTemperatureWaveFoam` solver. For this purpose, an translucent container was built. The container was divided into two compartments that were open at the top (i.e., free-surface condition) and divided by one vertical slide or gate. The container was instrumented with ten thermistors to capture the temperature of the contained fluids.

The experimental model was simulated with the developed solver, and both results were compared to estimate the numerical error with three turbulence models: zero-equation turbulence model, two-equation (k-$\omega$) RANS model, and Large Eddy Simulation (LES) model. Independently of the differences between the turbulence models, none of them exhibits a mean error higher than 17 %.

The most important causes of discrepancy between the experimental model and the numerical simulation are:

i) The presence of a gate that must be rapidly opened against the ideal dam-break case, where there exists no gate but a membrane which “disappears” instantly.

ii) To install the gate, two rails were placed to guide the displacement upwards of the gate. These rails were designed to interfere as little as possible in the fluid motion.

iii) The installation of the thermistors also interferes in the fluid motions. To minimize this effect, the thermistors were adapted through perforations in the front wall of the acrylic container, so that the wall in contact with the fluid was as smooth as possible.

iv) The precision when filling the two compartments of the container and the measurement of the temperature and density. It was observed that little variations in the water level caused great differences in fluid flow behavior.

Regardless of the above-mentioned factors, the validation tests showed an adequate performance of the numerical model.

## 8.4. RECOMMENDATIONS FOR FUTURE REASERCH

The thermistors’ time resolution during the validation experiments was not precise enough to capture the fast temperature variations during the transient phase of the experiment. Further research with more precise instrumentation could be valuable to validate the transient phenomena and to apply the developed solver in cases where this feature is relevant.

Due to the versatility of CFD models, the developed solver can be also implemented in a variety of different cases that involve multiphase flows, miscibility of two liquid phases with different temperature/density, wave generation/absorption, and the need to evaluate different turbulence models. Thus, the potential applications of the solver besides OTEC thermal water discharge include submarine groundwater discharge, sewage discharge in coastal waters, and water discharge in coastal waters of nuclear plants, among others. Since OpenFOAM® is a free open-source code, `interMixingTemperatureWaveFoam` can be improved and extended in the future by users and developers.

# 9

# APPENDICES

# 9. APPENDICES

## 9.1. APPENDIX A. SOLVER IMPLEMENTATION IN OPENFOAM®

OpenFOAM® has an Open-Source General Public License, which allows the users to freely use and modify the CFD code. Thus, the user's community has developed several extensions of the software to simulate specific cases that are not contemplated in the official OpenFOAM® version. Extensions are available in the contributions web page (https://openfoamwiki.net/index.php/Contrib).

For the current work, OpenFOAM® version 2.3.1 was used and extended by adding a new solver capable of handle the multiphase fluid flow and wave generation/absorption. The multiphase fluid flow consists of three phases with different densities and temperatures, i.e., two of the phases are liquid and miscible, and the third one represents air; thus, there is a free-surface condition.

Following the solver's name convention in OpenFOAM®, the implemented solver is named `interMixingTemperatureWaveFoam`, where "interMixing" stands for the ability to solve three incompressible fluids, two of which are miscible, "Temperature" for the implementation of the energy equation in terms of the temperature field, and "Wave" for the ability of wave generation and absorption. Turbulence models, mesh generation schemes and other generic OpenFOAM® utilities are available for the case setting and running.

The implementation process of `interMixingTemperatureWaveFoam` can be summarized in the following steps:

i) Creation of the solver libraries and folders departing from preexisting native solvers in OpenFOAM®.
ii) Modification of the transport model to include the parameters of thermal conductivity, specific heat capacity and Prandtl number for each phase.
iii) Incorporation of the energy conservation equation in the code.
iv) Addition of the wave generation/absorption libraries to the code.
v) Compilation of the new solver.
vi) Setup of a simple test case.

A detailed description of this process is presented hereunder. This section is written as a manual so that anyone can reproduce these steps in a computer with OpenFOAM® 2.3.1. Code written in red is the one that is modified or added to the preexisting solver's code.

### 9.1.1. Creation of the solver's files

The new solver is not coded from scratch, but from pre-existing libraries/solvers by adding or changing code lines, i.e., through available codes included within the OpenFOAM® framework.

The start point is the OpenFOAM® native solver called `interMixingFoam`, which in turn evolved from `interFoam`. Both of them solve incompressible fluids using VOF method to capture the interface. On one hand, `interFoam` solves the two-fluid flow system; on the other hand, `interMixingFoam` includes a third miscible liquid phase. The mixture between the liquid phases

is solved separately and then incorporated into the VOF method as a single liquid phase. Validation of `interMixingFoam` with tree turbulence models is found in Rodríguez-Ocampo (2016).

One contribution that is particularly useful for this work is the library `waves2Foam`, which was developed and validated by Jacobsen *et al.* (2011). This toolbox is also based on the `interFoam` solver, and it is used to generate/absorb free surface water waves considering two incompressible fluid phases. Thus, the `waves2Foam` solver must be correctly downloaded and compiled before starting the new solver implementation.

It is recommended to compile the new solver outside OpenFOAM®'s directory; therefore, `waves2Foam` directory will be used to implement `interMixingTemperatureWaveFoam`. The first step is to copy the folder of `interMixingFoam` (located in *$FOAM_APP/solvers/multiphase/interFoam/interMixingFoam/)* into the `waves2Foam` solver's directory (*$WAVES_DIR/…/waves2Foam/applications/solvers/solvers2*31). This folder is renamed as `interMixingTemperatureWaveFoam` (Figure 37).


*Figure 37. Location of the new solver interMixingTemperatureWaveFoam implemented with OpenFOAM®*

To implement the new solver's name, `interMixingFoam.C` file is renamed as `interMixingTemperatureWaveFoam.C`, and the *make/files* directory is modified as shown in Listing 1.

*Listing 1. Make/files content within interMixingTemperatureWaveFoam folder.*

```
incompressibleThreePhaseMixtureTemp/incompressibleThreePhaseMixtureTemp.C
threePhaseInterfacePropertiesTemp/threePhaseInterfacePropertiesTemp.C
immiscibleIncompressibleThreePhaseMixtureTemp/immiscibleIncompressibleThreePhaseMixtureTemp.
C
interMixingTemperatureWaveFoam.C

EXE = $(FOAM_APPBIN)/interMixingTemperatureWaveFoam
```

In *make/options*, the new transport-model files and `waves2Foam` utilities are called. Thus, it is modified as in Listing 2.

*Listing 2. make/options content after adding and modifying the executable libraries and their location*

```
EXE_INC = \
    -I.. \
    -I$(LIB_SRC)/transportModels/twoPhaseMixture/lnInclude \
    -IincompressibleThreePhaseMixtureTemp \
    -IimmiscibleIncompressibleThreePhaseMixtureTemp \
    -IthreePhaseInterfacePropertiesTemp \
    -I$(LIB_SRC)/transportModels/interfaceProperties/lnInclude \
    -I$(LIB_SRC)/transportModels/twoPhaseProperties/alphaContactAngle/alphaContactAngle \
    -I$(LIB_SRC)/turbulenceModels/incompressible/turbulenceModel \
```

```
    -I$(LIB_SRC)/finiteVolume/lnInclude \
    -I$(LIB_SRC)/transportModels \
    -I$(LIB_SRC)/meshTools/lnInclude \
    -I$(LIB_SRC)/fvOptions/lnInclude \
    -I$(LIB_SRC)/sampling/lnInclude  \
    -DOFVERSION=230 \
    -DEXTBRANCH=1 \
    -DXVERSION=$(WAVES_XVERSION) \
    -I./../../../../src/waves2Foam/lnInclude \
    -I./../../../../src/waves2FoamSampling/lnInclude \
    -I. \
    -I$(WAVES_SRC)/waves2Foam/lnInclude \
    -I$(WAVES_SRC)/waves2FoamSampling/lnInclude \
    -I$(WAVES_GSL_INCLUDE)

EXE_LIBS = \
    -ltwoPhaseMixture \
    -linterfaceProperties \
    -ltwoPhaseProperties \
    -lincompressibleTransportModels \
    -lincompressibleTurbulenceModel \
    -lincompressibleRASModels \
    -lincompressibleLESModels \
    -lfiniteVolume \
    -lmeshTools \
    -lfvOptions \
    -lsampling \
    -L$(WAVES_LIBBIN) \
    -L$(WAVES_LIBBIN) \
    -L$(FOAM_USER_LIBBIN) \
    -lwaves2Foam \
    -lwaves2FoamSampling \
    -L$(WAVES_GSL_LIB) \
    -lgsl \
    -lgslcblas
```

Because `interMixingFoam` comes from `interFoam`, some necessary files have to be copied from `interFoam` directory to the new solver location, i.e., `alphaCourantNo.H`, `correctPhi.H`, `pEqn.H`, `setDeltaT.H` and `UEqn.H` (Figure 38).

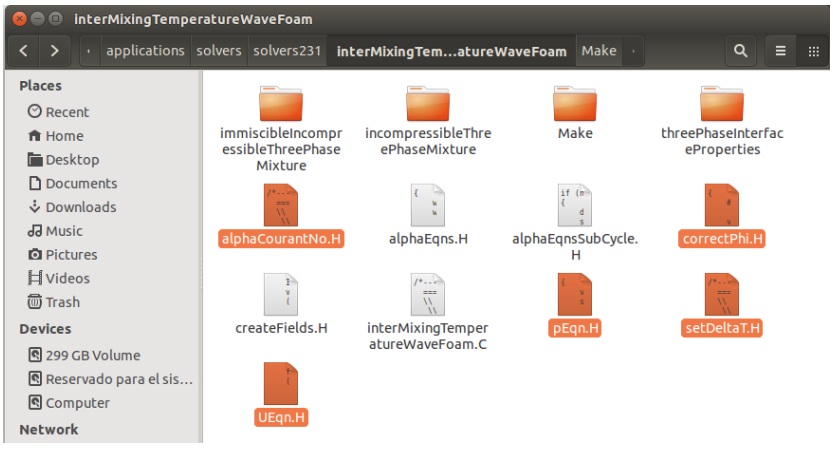


*Figure 38. `interMixingTemperatureWaveFoam` content after adding the missing files of `interFoam` and renaming the .C file.*

### 9.1.2. Modification of the transport model files

The structure of both `interMixingFoam` and derived `interMixingTemperatureWaveFoam` is presented in Figure 39.

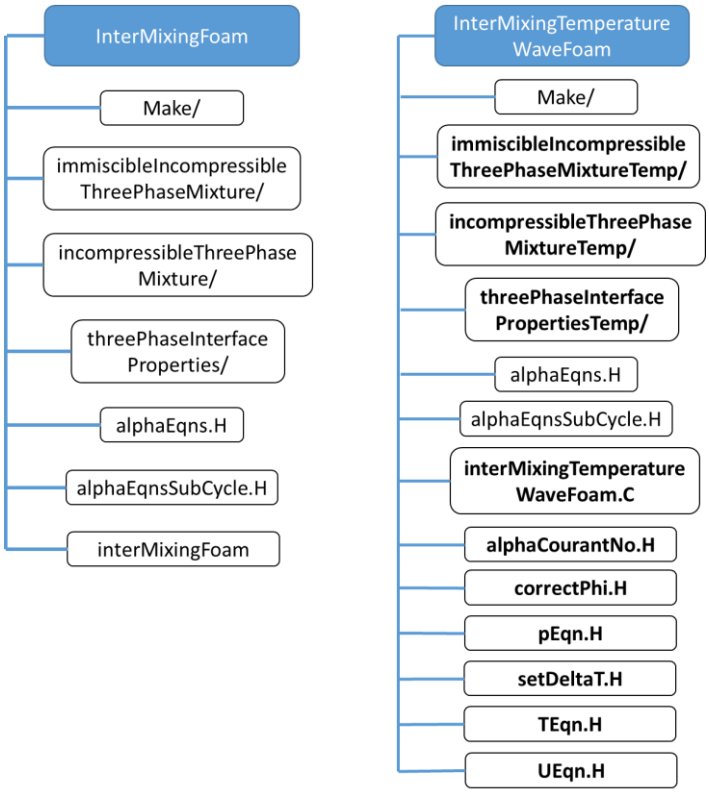


*Figure 39. The tree of solver interMixingFoam and interMixingTemperatureWaveFoam.*

The folders *immiscibleIncompressibleThreePhaseMixture/*, *incompressibleThreePhaseMixture/* and *threePhaseInterfaceProperties/* directories are renamed as *immiscibleIncompressibleThreePhaseMixtureTemp/*, *incompressibleThreePhaseMixtureTemp/*, and *threePhaseInterfacePropertiesTemp/*, correspondingly, including their respective .C and .H files (Figure 40). These files contain the transport model that will be used by the solver. These new names must also be considered in the *make/files* and *make/options* directories as shown in Listing 1 and Listing 2.

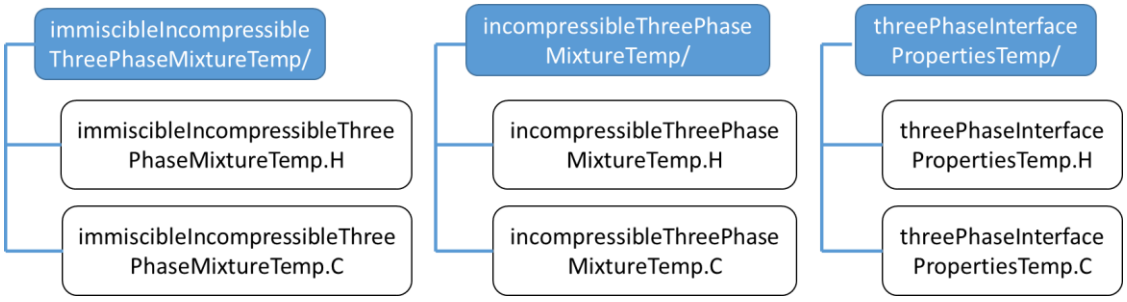


*Figure 40. Content of each folder of the transport model.*

The code for each .C and .H files within the three folders previously mentioned is modified to consider their new names. It is also added the necessary code for the computation of the thermal conductivity and the declaration of specific heat capacity $Cp$ and Prandtl number $Pr$ for each phase. Thermal conductivity $k$ is calculated according to the following expression:

$$k = \alpha_1 \rho_1 \upsilon_1 \frac{c_{p1}}{Pr_1} + \alpha_2 \rho_2 \upsilon_2 \frac{c_{p2}}{Pr_2} + \alpha_3 \rho_3 \upsilon_3 \frac{c_{p3}}{Pr_3} \qquad \text{Equation 41}$$

Thus, the code is modified as shown in Listing 3, Listing 4, Listing 5, Listing 6, Listing 7 and Listing 8. Some code listings are abbreviated as //…// to optimize space.

*Listing 3. Code of immiscibleIncompressibleThreePhaseMixtureTemp.H of interMixingTemperatureWaveFoam solver.*

```
\*---------------------------------------------------------------------------*/
#ifndef immiscibleIncompressibleThreePhaseMixtureTemp_H   //modified
#define immiscibleIncompressibleThreePhaseMixtureTemp_H  //modified

#include "incompressibleThreePhaseMixtureTemp.H"  //modified
#include "threePhaseInterfacePropertiesTemp.H"  //modified

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

namespace Foam
{

/*---------------------------------------------------------------------------*\
          Class immiscibleIncompressibleThreePhaseMixtureTemp Declaration
\*---------------------------------------------------------------------------*/

class immiscibleIncompressibleThreePhaseMixtureTemp  //modified
:
    public incompressibleThreePhaseMixtureTemp,  //modified
    public threePhaseInterfacePropertiesTemp  //modified
{

public:

    // Constructors

        //- Construct from components
        immiscibleIncompressibleThreePhaseMixtureTemp  //modified
        (
            const volVectorField& U,
            const surfaceScalarField& phi
        );

    //- Destructor
    virtual ~immiscibleIncompressibleThreePhaseMixtureTemp()  //modified
    {}


    // Member Functions

        //- Correct the transport and interface properties
        virtual void correct()
        {
            incompressibleThreePhaseMixtureTemp::correct();  //modified
            threePhaseInterfacePropertiesTemp::correct();  //modified
        }
```

```
};

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
} // End namespace Foam
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
#endif

// ************************************************************************* //
```

*Listing 4. Code of immiscibleIncompressibleThreePhaseMixtureTemp.C of interMixingTemperatureWaveFoam solver.*

```
\*---------------------------------------------------------------------------*/

#include "immiscibleIncompressibleThreePhaseMixtureTemp.H"  //modified

// * * * * * * * * * * * * * * * * * * Constructors  * * * * * * * * * * * * * * * * * * //

Foam::immiscibleIncompressibleThreePhaseMixtureTemp::  //modified
immiscibleIncompressibleThreePhaseMixtureTemp  //modified
(
    const volVectorField& U,
    const surfaceScalarField& phi
)
:
    incompressibleThreePhaseMixtureTemp(U, phi),  //modified
    threePhaseInterfacePropertiesTemp  //modified
    (
        static_cast<incompressibleThreePhaseMixtureTemp&>(*this)  //modified
    )
{}
// ************************************************************************* //
```

*Listing 5. Code of incompressibleThreePhaseMixtureTemp.H of interMixingTemperatureWaveFoam solver.*

```
\*---------------------------------------------------------------------------*/

#ifndef incompressibleThreePhaseMixtureTemp_H  //modified
#define incompressibleThreePhaseMixtureTemp_H  //modified

#include "incompressible/transportModel/transportModel.H"
#include "IOdictionary.H"
#include "incompressible/viscosityModels/viscosityModel/viscosityModel.H"
#include "dimensionedScalar.H"
#include "volFields.H"

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

namespace Foam
{
/*---------------------------------------------------------------------------*\
                Class incompressibleThreePhaseMixtureTemp Declaration
\*---------------------------------------------------------------------------*/

class incompressibleThreePhaseMixtureTemp  //modified
:
    public IOdictionary,
```

```
    public transportModel
{
    // Private data

        word phase1Name_;
        word phase2Name_;
        word phase3Name_;

        volScalarField alpha1_;
        volScalarField alpha2_;
        volScalarField alpha3_;

        const volVectorField& U_;
        const surfaceScalarField& phi_;

        volScalarField nu_;

        autoPtr<viscosityModel> nuModel1_;
        autoPtr<viscosityModel> nuModel2_;
        autoPtr<viscosityModel> nuModel3_;

        dimensionedScalar rho1_;
        dimensionedScalar rho2_;
        dimensionedScalar rho3_;

//-------------------------Modified----------------------//
        dimensionedScalar cp1_;
        dimensionedScalar cp2_;
        dimensionedScalar cp3_;
        dimensionedScalar Pr1_;
        dimensionedScalar Pr2_;
        dimensionedScalar Pr3_;
//-------------------------End-----------------------//

    // Private Member Functions

        //- Calculate and return the laminar viscosity
        void calcNu();

public:

    // Constructors

        //- Construct from components
        incompressibleThreePhaseMixtureTemp  //modified

         //... //

        //- Return const-access to phase3 density
        const dimensionedScalar& rho3() const
        {
            return rho3_;
        };

//-------------------------Modified----------------------//
        const dimensionedScalar& Pr1() const
        {
            return Pr1_;
        };
        const dimensionedScalar& Pr2() const
        {
            return Pr2_;
        };
```

```
        const dimensionedScalar& Pr3() const
        {
            return Pr3_;
        };
        const dimensionedScalar& cp1() const
        {
            return cp1_;
        };
        const dimensionedScalar& cp2() const
        {
            return cp2_;
        };
        const dimensionedScalar& cp3() const
        {
            return cp3_;
        };
//--------------------------End-----------------------//

         //... //

        //- Return the face-interpolated dynamic laminar viscosity
        tmp<surfaceScalarField> muf() const;

//-------------------------Modified---------------------//
        tmp<surfaceScalarField> kappaf() const;
//-------------------------Modified---------------------//

        //- Return the kinematic laminar viscosity
        tmp<volScalarField> nu() const
        {
            return nu_;
        }

        //- Return the laminar viscosity for patch
        tmp<scalarField> nu(const label patchi) const
        {
            return nu_.boundaryField()[patchi];
        }

        //- Return the face-interpolated dynamic laminar viscosity
        tmp<surfaceScalarField> nuf() const;

        //- Correct the laminar viscosity
        void correct()
        {
            calcNu();
        }

        //- Read base transportProperties dictionary
        bool read();
};

} // End namespace Foam

#endif

// ************************************************************************* //
```

*Listing 6. Code of incompressibleThreePhaseMixtureTemp.C of interMixingTemperatureWaveFoam solver*

```
\*---------------------------------------------------------------------------*/
#include "incompressibleThreePhaseMixtureTemp.H"  //modified
#include "addToRunTimeSelectionTable.H"
#include "surfaceFields.H"
#include "fvc.H"

// * * * * * * * * * * * * * Private Member Functions  * * * * * * * * * * * * * //
//- Calculate and return the laminar viscosity
void Foam::incompressibleThreePhaseMixtureTemp::calcNu()  //modified
{
    nuModel1_->correct();
    nuModel2_->correct();
    nuModel3_->correct();

    // Average kinematic viscosity calculated from dynamic viscosity
    nu_ = mu()/(alpha1_*rho1_ + alpha2_*rho2_ + alpha3_*rho3_);
}

// * * * * * * * * * * * * * * * * * * Constructors  * * * * * * * * * * * * * * * * * * //
Foam::incompressibleThreePhaseMixtureTemp::incompressibleThreePhaseMixtureTemp
//modified
        //... //
    rho1_(nuModel1_->viscosityProperties().lookup("rho")),
    rho2_(nuModel2_->viscosityProperties().lookup("rho")),
    rho3_(nuModel3_->viscosityProperties().lookup("rho")),

//------------------------Modified---------------------//
    cp1_(nuModel1_->viscosityProperties().lookup("cp")),
    cp2_(nuModel2_->viscosityProperties().lookup("cp")),
    cp3_(nuModel3_->viscosityProperties().lookup("cp")),
    Pr1_(nuModel1_->viscosityProperties().lookup("Pr")),
    Pr2_(nuModel2_->viscosityProperties().lookup("Pr")),
    Pr3_(nuModel3_->viscosityProperties().lookup("Pr"))
//------------------End--------------------------------------//

{
    alpha3_ == 1.0 - alpha1_ - alpha2_;
    calcNu();
}

// * * * * * * * * * * * * * * * * Member Functions  * * * * * * * * * * * * * * * * //
Foam::tmp<Foam::volScalarField>
Foam::incompressibleThreePhaseMixtureTemp::mu() const  //modified
{
    return tmp<volScalarField>
    (
        new volScalarField
        (
            "mu",
            alpha1_*rho1_*nuModel1_->nu()
          + alpha2_*rho2_*nuModel2_->nu()
          + alpha3_*rho3_*nuModel3_->nu()
        )
    );
}

Foam::tmp<Foam::surfaceScalarField>
Foam::incompressibleThreePhaseMixtureTemp::muf() const  //modified
{
        //... //
```

```
}

Foam::tmp<Foam::surfaceScalarField>
Foam::incompressibleThreePhaseMixtureTemp::nuf() const  //modified
{
        //... //
}
//-----------------------Modified-------------------------//
Foam::tmp<Foam::surfaceScalarField>
Foam::incompressibleThreePhaseMixtureTemp::kappaf() const  //modified
{
    surfaceScalarField alpha1f(fvc::interpolate(alpha1_));
    surfaceScalarField alpha2f(fvc::interpolate(alpha2_));
    surfaceScalarField alpha3f(fvc::interpolate(alpha3_));

    return tmp<surfaceScalarField>
    (
        new surfaceScalarField
        (
            "kappaf",
            (
             alpha1f*rho1_*cp1_*(1/Pr1_)*fvc::interpolate(nuModel1_->nu())
             + alpha2f*rho2_*cp2_*(1/Pr2_)*fvc::interpolate(nuModel2_->nu())
                + alpha3f*rho3_*cp3_*(1/Pr3_)*fvc::interpolate(nuModel3_->nu())
          )
        )
    );
}
//-------------------------End-------------------------------//

bool Foam::incompressibleThreePhaseMixtureTemp::read()  //modified
{
    if (transportModel::read())
    {
         //... //
        {
            nuModel1_->viscosityProperties().lookup("rho") >> rho1_;
            nuModel2_->viscosityProperties().lookup("rho") >> rho2_;
            nuModel3_->viscosityProperties().lookup("rho") >> rho3_;

//-------------------------Modified---------------------//
           nuModel1_->viscosityProperties().lookup("cp") >> cp1_;
           nuModel2_->viscosityProperties().lookup("cp") >> cp2_;
           nuModel3_->viscosityProperties().lookup("cp") >> cp3_;
           nuModel1_->viscosityProperties().lookup("Pr") >> Pr1_;
           nuModel2_->viscosityProperties().lookup("Pr") >> Pr2_;
           nuModel3_->viscosityProperties().lookup("Pr") >> Pr3_;
//-------------------------End----------------------//

            return true;
        }
        else
        {
            return false;
        }
    }
    else
    {
        return false;
    }
}

// ************************************************************************* //
```

*Listing 7. Code of threePhaseInterfacePropertiesTemp.H of interMixingTemperatureWaveFoam solver.*

```
*---------------------------------------------------------------------------*/
#ifndef threePhaseInterfacePropertiesTemp_H  //modified
#define threePhaseInterfacePropertiesTemp_H  //modified

#include "incompressibleThreePhaseMixtureTemp.H"  //modified
#include "surfaceFields.H"

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

namespace Foam
{

/*---------------------------------------------------------------------------*\
             Class threePhaseInterfacePropertiesTemp Declaration
\*---------------------------------------------------------------------------*/

class threePhaseInterfacePropertiesTemp  //modified
{
    // Private data

        const incompressibleThreePhaseMixtureTemp& mixture_;  //modified

        //- Compression coefficient
        scalar cAlpha_;

        //- Surface tension 1-2
        dimensionedScalar sigma12_;

        //- Surface tension 1-3
        dimensionedScalar sigma13_;

        //- Stabilisation for normalisation of the interface normal
        const dimensionedScalar deltaN_;

        surfaceScalarField nHatf_;
        volScalarField K_;

    // Private Member Functions

        //- Disallow default bitwise copy construct and assignment
        threePhaseInterfacePropertiesTemp(const
threePhaseInterfacePropertiesTemp&);  //modified
        void operator=(const threePhaseInterfacePropertiesTemp&);  //modified

        //- Correction for the boundary condition on the unit normal nHat on
        //  walls to produce the correct contact dynamic angle
        //  calculated from the component of U parallel to the wall
        void correctContactAngle
        (
            surfaceVectorField::GeometricBoundaryField& nHat
        ) const;

        //- Re-calculate the interface curvature
        void calculateK();

public:
    //- Conversion factor for degrees into radians
    static const scalar convertToRad;

    // Constructors
```

```
        //- Construct from volume fraction field alpha and IOdictionary
        threePhaseInterfacePropertiesTemp
        (
            const incompressibleThreePhaseMixtureTemp& mixture  //modified
        );

    // Member Functions
        scalar cAlpha() const
        {
            return cAlpha_;
        }

        const dimensionedScalar& deltaN() const
        {
            return deltaN_;
        }

        const surfaceScalarField& nHatf() const
        {
            return nHatf_;
        }

        const volScalarField& K() const
        {
            return K_;
        }

        tmp<volScalarField> sigma() const
        {
            volScalarField limitedAlpha2(max(mixture_.alpha2(), scalar(0)));
            volScalarField limitedAlpha3(max(mixture_.alpha3(), scalar(0)));

            return
                (limitedAlpha2*sigma12_ + limitedAlpha3*sigma13_)
               /(limitedAlpha2 + limitedAlpha3 + SMALL);
        }

        tmp<volScalarField> sigmaK() const
        {
            return sigma()*K_;
        }

        tmp<surfaceScalarField> surfaceTensionForce() const;  //modified

        //- Indicator of the proximity of the interface
        //  Field values are 1 near and 0 away for the interface.
        tmp<volScalarField> nearInterface() const;

        void correct()
        {
            calculateK();
        }
};

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
} // End namespace Foam
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

#endif

// ************************************************************************* //
```

*Listing 8. Code of threePhaseInterfacePropertiesTemp.C of interMixingTemperatureWaveFoam solver.*

```
\*---------------------------------------------------------------------------*/
#include "threePhaseInterfacePropertiesTemp.H"  //modified
#include "alphaContactAngleFvPatchScalarField.H"
#include "mathematicalConstants.H"
#include "surfaceInterpolate.H"
#include "fvcDiv.H"
#include "fvcGrad.H"
#include "fvcSnGrad.H"

// * * * * * * * * * * * * * * * * Static Member Data  * * * * * * * * * * * * * //

const Foam::scalar Foam::threePhaseInterfacePropertiesTemp::convertToRad =
    Foam::constant::mathematical::pi/180.0;  //modified


// * * * * * * * * * * * * * * * Private Member Functions  * * * * * * * * * * * //

// Correction for the boundary condition on the unit normal nHat on
// walls to produce the correct contact angle.

// The dynamic contact angle is calculated from the component of the
// velocity on the direction of the interface, parallel to the wall.

void Foam::threePhaseInterfacePropertiesTemp::correctContactAngle
(
    surfaceVectorField::GeometricBoundaryField& nHatb
) const
{
        //... //
}


void Foam::threePhaseInterfacePropertiesTemp::calculateK()  //modified
{
        //... //
}


// * * * * * * * * * * * * * * * * Constructors  * * * * * * * * * * * * * * * //

Foam::threePhaseInterfacePropertiesTemp::threePhaseInterfacePropertiesTemp
//modified
(
    const incompressibleThreePhaseMixtureTemp& mixture  //modified
)
:
        //... // Line 214

// * * * * * * * * * * * * * * * Member Functions  * * * * * * * * * * * * * * * //

Foam::tmp<Foam::surfaceScalarField>
Foam::threePhaseInterfacePropertiesTemp::surfaceTensionForce() const  //modified
{
    return fvc::interpolate(sigmaK())*fvc::snGrad(mixture_.alpha1());
}


Foam::tmp<Foam::volScalarField>
Foam::threePhaseInterfacePropertiesTemp::nearInterface() const  //modified
{
```

```
    return max
    (
        pos(mixture_.alpha1() - 0.01)*pos(0.99 - mixture_.alpha1()),
        pos(mixture_.alpha2() - 0.01)*pos(0.99 - mixture_.alpha2())
    );
}

// ************************************************************************* //
```

### 9.1.3. Addition of the energy equation

The files `alphaEqns.H`, `alphaEqnsSubCycle.H`, `createFields.H`, and `interMixingTemperatureWaveFoam.C` are adapted so that energy equation can be computed in a new file called `TEqn.H`.

First, `createFields.H` is modified by creating a temperature field "T" and adding specific heat capacity $Cp$ and Prandtl number $Pr$ for each phase. Relaxation zones for wave generation/absorption are also created in this step. Thus, the following lines are added in `createFields.H` (Listing 9).

*Listing 9. Content of createFields.H file of interMixingTemperatureWaveFoam solver.*

```
    Info<< "Reading field p_rgh\n" << endl;
    volScalarField p_rgh
    (
        IOobject
        (
            "p_rgh",
            runTime.timeName(),
            mesh,
            IOobject::MUST_READ,
            IOobject::AUTO_WRITE
        ),
        mesh
    );

    Info<< "Reading field U\n" << endl;
    volVectorField U
    (
        IOobject
        (
            "U",
            runTime.timeName(),
            mesh,
            IOobject::MUST_READ,
            IOobject::AUTO_WRITE
        ),
        mesh
    );

    #include "createPhi.H"

    Info<< "Reading transportProperties\n" << endl;
    immiscibleIncompressibleThreePhaseMixtureTemp mixture(U, phi); //modified

    volScalarField& alpha1(mixture.alpha1());
    volScalarField& alpha2(mixture.alpha2());
    volScalarField& alpha3(mixture.alpha3());
```

```
    const dimensionedScalar& rho1 = mixture.rho1();
    const dimensionedScalar& rho2 = mixture.rho2();
    const dimensionedScalar& rho3 = mixture.rho3();

    dimensionedScalar D23(mixture.lookup("D23"));

//------------------Modified---------------------//
    const dimensionedScalar& cp1 = mixture.cp1();
    const dimensionedScalar& cp2 = mixture.cp2();
    const dimensionedScalar& cp3 = mixture.cp3();
//----------------------End----------------------//

    // Need to store rho for ddt(rho, U)
    volScalarField rho
    (
        IOobject
        (
            "rho",
            runTime.timeName(),
            mesh,
            IOobject::READ_IF_PRESENT,
            IOobject::AUTO_WRITE  //added by eli
        ),

        alpha1*rho1 + alpha2*rho2 + alpha3*rho3,
        alpha2.boundaryField().types()
    );
    rho.oldTime();

//------------------Modified---------------------//
Info<< "Reading field T\n" << endl;
      volScalarField T
      (
            IOobject
            (
                  "T",
                  runTime.timeName(),
                  mesh,
                  IOobject::MUST_READ,
                  IOobject::AUTO_WRITE
            ),
            mesh
      );
//----------------------End----------------------//

    // Mass flux
    // Initialisation does not matter because rhoPhi is reset after the
    // alpha solution before it is used in the U equation.

         //... //

    if (p_rgh.needReference())
    {
        p += dimensionedScalar
        (
            "p",
            p.dimensions(),
            pRefValue - getRefCellValue(p, pRefCell)
        );
        p_rgh = p - rho*gh;
    }
```

```
//------------------Modified---------------------//
Info<< "Reading / calculating rho*cp\n" << endl;
volScalarField rhoCp
(
   IOobject
   (
    "rho*Cp",
    runTime.timeName(),
    mesh,
    IOobject::NO_READ,
    IOobject::NO_WRITE
   ),
   alpha1*rho1*cp1 + alpha2*rho2*cp2 + alpha3*rho3*cp3,
   alpha2.boundaryField().types()
);
rhoCp.oldTime();
Info<< "Reading / calculating rho*phi*cp\n" << endl;
   surfaceScalarField rhoPhiCpf
   (
    IOobject
    (
     "rho*phi*cpf",
     runTime.timeName(),
     mesh,
     IOobject::NO_READ,
     IOobject::NO_WRITE
    ),
    rhoPhi*cp2
);
//---------------------End----------------------//


    fv::IOoptionList fvOptions(mesh);


    // MULES Correction
    tmp<surfaceScalarField> tphiAlphaCorr0;

//------------------Modified---------------------//
    relaxationZone relaxing(mesh, U, alpha2);
//------------------End----------------------//
```

alphaEqns.H and alphaEqnsSubCycle.H files are modified as showed in Listing 10 and Listing 11.

*Listing 10. Content of alphaEqns.H file of interMixingTemperatureWaveFoam solver.*

```
         //... //Line 153

        // Construct the complete mass flux
        rhoPhi =
              phiAlpha1*(rho1 - rho3)
            + (phiAlpha2 + alpha2Eqn.flux())*(rho2 - rho3)
            + phi*rho3;

        alpha3 = 1.0 - alpha1 - alpha2;

//------------------Modified---------------------//
rhoPhiCpf = phiAlpha1*(rho1*cp1 - rho3*cp3)
```

```
            + (phiAlpha2 + alpha2Eqn.flux())*(rho2*cp2 - rho3*cp3)
            + phi*rho3*cp3;

//-----------------------End-----------------------//

        //... //
```

*Listing 11. Content of alphaEqnsSubCycle.H file of interMixingTemperatureWaveFoam solver*

```
        //... //Line 31

{
    volScalarField rhoNew(alpha1*rho1 + alpha2*rho2 + alpha3*rho3);
    rho == rhoNew;
}

//-------------------Modified--------------------//
rhoCp == alpha1*rho1*cp1 + alpha2*rho2*cp2 + alpha3*rho3*cp3;
//-----------------------End-----------------------//
```

As previously mentioned, a file called `TEqn.H` has to be created to solve the energy equation in terms of temperature, specific heat capacity and thermal conductivity (Listing 12).

*Listing 12. Content of TEqn.H file of interMixingTemperatureWaveFoam solver.*

```
surfaceScalarField kappaf = mixture.kappaf();
fvScalarMatrix TEqn
(
fvm::ddt(rhoCp, T)
+ fvm::div(rhoPhiCpf, T)
- fvm::laplacian(kappaf, T)
);
TEqn.solve();
```

The file `interMixingTemperatureWaveFoam.C` is adapted considering the above mentioned changes as in Listing 13.

*Listing 13. Content of interMixingTemperatureWaveFoam.C file.*

```
\*---------------------------------------------------------------------------*/

#include "fvCFD.H"
#include "CMULES.H"
#include "subCycle.H"
#include "immiscibleIncompressibleThreePhaseMixtureTemp.H" //modified
#include "turbulenceModel.H"
#include "pimpleControl.H"
#include "fvIOoptionList.H"
#include "fixedFluxPressureFvPatchScalarField.H"

#include "relaxationZone.H" // added
#include "externalWaveForcing.H" // added

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
```

```
int main(int argc, char *argv[])
{
    #include "setRootCase.H"
    #include "createTime.H"
    #include "createMesh.H"
    #include "initContinuityErrs.H"
    #include "readGravitationalAcceleration.H" // added
    #include "readWaveProperties.H"  // added
    #include "createExternalWaveForcing.H"  //added
    #include "createFields.H"
    #include "readTimeControls.H"
    #include "CourantNo.H"
    #include "setInitialDeltaT.H"

    pimpleControl pimple(mesh);

    #include "createPrghCorrTypes.H"
    #include "correctPhi.H"

    // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

    Info<< "\nStarting time loop\n" << endl;

    while (runTime.run())
    {
        #include "readTimeControls.H"
        #include "CourantNo.H"
        #include "alphaCourantNo.H"
        #include "setDeltaT.H"

        runTime++;

        Info<< "Time = " << runTime.timeName() << nl << endl;

        externalWave->step();  //added

        // --- Pressure-velocity PIMPLE corrector loop
        while (pimple.loop())
        {
            #include "alphaControls.H"
            #include "alphaEqnsSubCycle.H"
            relaxing.correct();   //added

            mixture.correct();

            #include "UEqn.H"

            #include "TEqn.H"  //added

            // --- Pressure corrector loop
            while (pimple.correct())
            {
                #include "pEqn.H"
            }

            if (pimple.turbCorr())
            {
                turbulence->correct();
            }
        }

        #include "continuityErrs.H" //Modified
```

```
        runTime.write();

        Info<< "ExecutionTime = " << runTime.elapsedCpuTime() << " s"
            << "  ClockTime = " << runTime.elapsedClockTime() << " s"
            << nl << endl;
    }

    Info<< "End\n" << endl;

    return 0;
}


// ************************************************************************* //
```

Finally the solver is compiled by running *./Allwmake* of the `waves2Foam` directory

## 9.2. APPENDIX B. SETUP OF A SIMPLE TEST CASE

A representative case is set up to demonstrate how to declare the new variables and the performance of `interMixingTemperatureWaveFoam`.

### 9.2.1. CASE DESCRIPTION

This case consists of a rectangular domain partially filled with water (phase2). Waves are generated on the left side of the domain and absorbed on the right side. The rest of the domain is filled with air (phase1), i.e., free-surface condition. In the middle of the domain, it is placed a point-source water discharge (phase3) with different temperature and density than the medium (Figure 41). Thus, in time zero, the water phase is static and the internal velocity field is zero. After that, the waves are generated and the water discharge at the bottom of the domain begins.

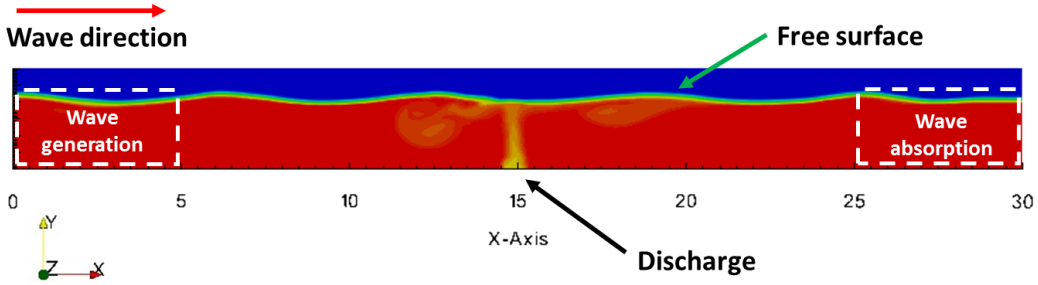


*Figure 41. Schematic representation of the test case. Phase2 (water) is represented in red, phase3 (water) in yellow, and phase1 in blue (air).*

The solver needs the following input data:

i) Environmental properties such as the value and direction of the gravitational acceleration
ii) Transport properties such as kinematic viscosity, density, surface tension, specific heat capacity, Prandtl number, and diffusivity between miscible phases
iii) Turbulence properties
iv) Wave properties
v) Temperature field. The corresponding values are introduced in the International System of Units.

The gravitational acceleration is considered as constant and the simulation type of this case is set as *laminar*, which means that the simulation will not use a specific turbulence model. Other turbulence models available are RANS and LES. On the other hand, the wave train characteristics (period, height, direction, and wave theory), and the relaxation zones are defined in the corresponding case file. The wave theory for the wave generation is set as linear (airy) in this case. Finally, the internal temperature field is set as 298 K, which is constant in time zero for phase 1 and phase 2. The submarine discharge has a constant temperature of 308 K, and the liquid flowing through it acquire this temperature.

*Table 20 and*

Table 21 summarize the basic data of the setup of this case (domain size, fluid properties, wave properties, initial conditions, etc.). A detailed description on how to set the described case is given below.

*Table 20. Characteristics of the domain, mesh, gravity, freshwater flow rate and phases properties for the test case.*

| | **Domain and mesh** | | **Gravitational acceleration** | **Submarine spring (0.8 m long)** |
|---|---|---|---|---|
| Direction | Dimension (m) | Cell number | a (m/s²) | Velocity v (m/s) |
| x | 30 | 357 | 0 | 0 |
| y | 3 | 60 | -9.81 | 0.2 |
| z | 0.1 | 1 | 0 | 0 |
| | | | | |
| **Phase properties** | | | | |
| **Phase property** | | **Phase 1 (air)** | **Phase 2 (water)** | **Phase 3 (water)** |
| Density (kg/m³) | | 1 | 1030 | 1000 |
| Kinematic viscosity (m²/s) | | 1.48x(10)^(-5) | 1x(10)^(-6) | 1x(10)^(-6) |
| Surface tension (N/m) | | 0.05 | | |
| Molecular diffusion coefficient (m²/s) | | - | 1.26x(10)^(-9) | |
| Temperature (K) | | 298 | 308 | 298 |
| Specific heat capacity (J/kg*K) | | 1045.2 | 1433.4 | 1433.4 |
| Prandtl number (1) | | 0.87 | 3.37 | 3.37 |

*Table 21. Wave properties and simulation time for the test case.*

| **Wave properties** | |
|---|---|
| Type | Regular wave train |
| Wave theory | Linear |
| Period T (s) | 4 |
| Wave height H (m) | 0.1 |
| Depth h (m) | 2 |
| Simulation time (s) | 30 |

As in any other case in OpenFOAM®, a folder must be designated for the case setup. When the case is run, the results are automatically saved within this directory, which is named "interMixingTemperatureWaveFoam". There are three folders inside: i) *0*, ii) *constant,* and iii) *system* (Figure 42). A folder named *0.org* is created to save the original files of the boundary conditions.

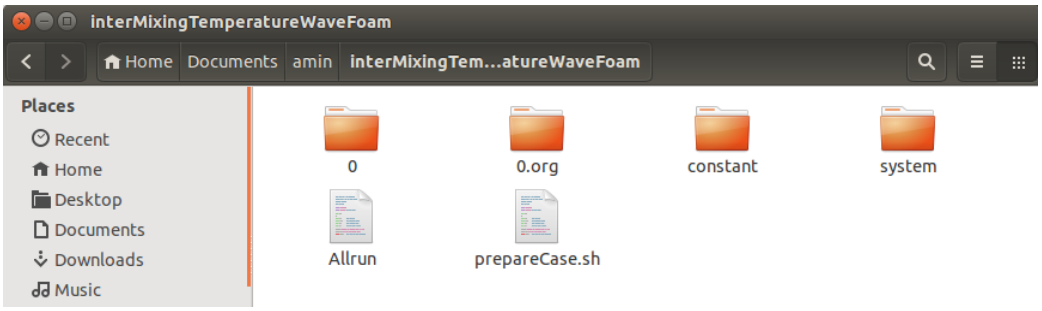


*Figure 42. Content of the test case directory.*

### 9.2.2. Mesh generation

Mesh generation files are located inside *constant/polyMesh* directory.

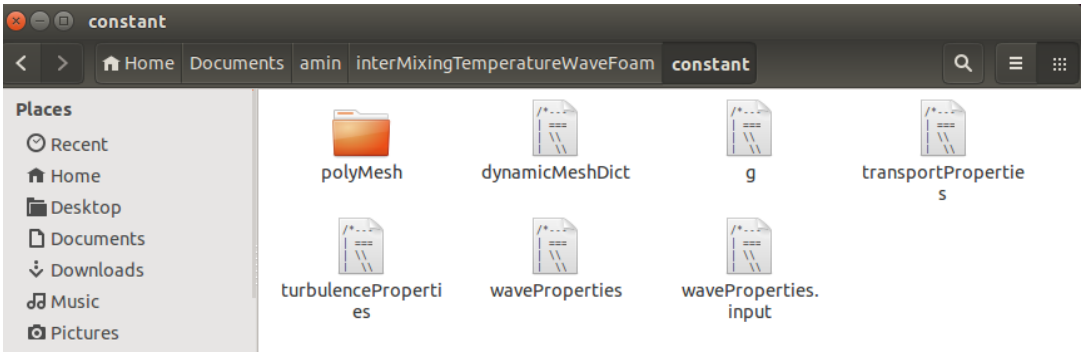


*Figure 43. Content of "constant" folder.*

*polyMesh* folder contains *blockMeshDict*, which is an input dictionary for the mesh generation (Listing 14). This file is designed to be run with *blockMesh*, a pre-processing OpenFOAM® tool which generates simple, unstructured meshes.

The domain is 30 m long, 3 m high, and 0.10 m width. 357 divisions in the X direction, 60 divisions in the Y direction, and one in the Z direction (2D condition) are set. The number for divisions on each side defines the cell's shape and size. The cell shape should be approximately a square to maintain the aspect ratio close to 1 and decrease numerical diffusion problems. Finally, each block patch is defined.

Boundary conditions for this case are defined hereunder. The patch in the bottom of the domain has a `wall` boundary condition, i.e., impermeable. Front/back patches have an `empty` boundary condition, which is used for 2D modeling. The patch that represents the opening to the atmosphere will allow the inflow and outflow of fluid, which maintains the free-surface condition of the water block. The walls that represent the inlet and outlet of the domain are set with a generic `patch` condition, and they will be later configured to generate and absorb waves, respectively.

*Listing 14. blockMeshDict entries for the test case.*

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

convertToMeters 1;
```

```
vertices

(
    (  0 -2 0 )
    ( 30 -2 0 )
    (  0  1 0 )
    ( 30  1 0 )

    (  0 -2 0.1 )
    ( 30 -2 0.1 )
    (  0  1 0.1 )
    ( 30  1 0.1 )
);

blocks

(
    hex (0 1 3 2 4 5 7 6) ( 357 60 1 ) simpleGrading (1 1 1)
);

edges

(
);

patches

(
    patch inlet
    (
        (0 4 6 2)
    )
    wall bottom
    (
        (0 1 5 4)
    )
    patch outlet
    (
        (1 5 7 3)
    )

    patch atmosphere
    (
        (2 3 7 6)
    )

    wall frontBack
    (
        (0 1 3 2)
        (4 5 7 6)
    )
);

mergePatchPairs

(
);

// ************************************************************************* //
```

To define the mesh in a simple manner, the patch for the submerged discharge is later created with other OpenFOAM® applications. This patch is done with two OpenFOAM® mesh tools: `topoSet` and `createPatch`. Both applications require dictionaries located in the *system* directory. *topoSetDict* is an input dictionary where a new cell set called "*ojo*" and its coordinates are defined (Listing 15). On the other hand, *createPatchDict* contains the instructions to create the patch "*ojo*" from the cell set generated by the `topoSet` tool (Listing 16).

*Listing 15. "topoSetDict" content for the generation of submerged discharge patch in the domain.*

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

// List of actions. Each action is a dictionary with e.g.

actions

(
    {
        name    ojo;
        type    faceSet;
        action  new;
        source  boxToFace;
        sourceInfo
        {
            box    (14.5 -2 0) (15.3001 -2 0.10001);
        }
    }

);

// ************************************************************************* //
```

*Listing 16. "createPatchDict" content for the generation of submerged discharge patch in the domain.*

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

pointSync false;

// Patches to create.

patches

(
    {
        // Name of new patch
        name ojo;
        // Dictionary to construct new patch from
        patchInfo
        {
            type patch;
        }
        set ojo;
    }
```

```
);


// ************************************************************************* //
```

The following instructions generate the final mesh. These are run in the terminal, within the case directory.

```
blockMesh
topoSet
createPatch -overwrite
```

Once the mesh is generated, it can be viewed in a compatible visualizer tool, e.g., Paraview®, by running `paraFoam`.

### 9.2.3. Physical and fluid properties

Physical and fluid properties required by `interMixingTemperatureWaveFoam` are set in the *constant* folder directory (Figure 43). The solver needs data of environmental properties such as value and direction of the gravitational acceleration, transport properties, turbulence properties, and wave properties. Each value is given considering International System units.

Kinematic viscosity, density, surface tension, specific heat capacity, Prandtl number, and diffusivity between miscible phases are defined in the *transportProperties* dictionary (Listing *17*).

*Listing 17. Definition of fluid phases transport properties in transportProperties dictionary*

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

phases (air water other);

air
{
    transportModel  Newtonian;
    nu              nu [0 2 -1 0 0 0 0] 1.48e-05;
    rho             rho [1 -3 0 0 0 0 0] 1;
    Pr              Pr [0 0 0 0 0 0 0] 0.87;
    cp                cp [0 2 -2 -1 0 0 0]   1045.2;
}

water
{
    transportModel  Newtonian;
    nu              nu [0 2 -1 0 0 0 0]  1e-6;
    rho             rho [1 -3 0 0 0 0 0] 1030;
    Pr              Pr [0 0 0 0 0 0 0] 3.366;  // Laminar Prandtl number
    cp              cp [0 2 -2 -1 0 0 0]  1433.4;
}

other
{
    transportModel  Newtonian;
```

```
    nu              nu [0 2 -1 0 0 0 0]  1e-6;
    rho             rho [1 -3 0 0 0 0 0] 1000;
    Pr              Pr [0 0 0 0 0 0 0] 3.366;  // Laminar Prandtl number
    cp              cp [0 2 -2 -1 0 0 0]  1433.4;
}

// Surface tension coefficients
sigma12           sigma12 [1 0 -2 0 0 0 0] 0.05;
sigma13           sigma13 [1 0 -2 0 0 0 0] 0.04;

// Diffusivity between miscible phases
D23               D23   [0 2 -1 0 0 0 0]  3e-09;

// ************************************************************************* //
```

The only environmental property required is the gravitational acceleration and is defined in environmentalProperties input dictionary (Listing 18).

*Listing 18. Definition of the gravitational acceleration in environmentalProperties dictionary.*

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 1 -2 0 0 0 0];
value           ( 0 -9.81 0 );

// ************************************************************************* //
```

*turbulenceProperties* dictionary defines whether the solution includes or not a turbulence model, and which turbulence model will be used. For this case, the simulation type is set as *laminar*, which means that the simulation will not use a specific turbulence model (Listing 19).

*Listing 19. Definition of the simulation type in* turbulenceProperties *dictionary.*

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

simulationType laminar;

// ************************************************************************* //
```

In the dictionary *waveProperties.input* the wave train characteristics (period, height, direction and wave theory), and the relaxation zones are defined *(*Listing 20*)*. Further information in this scope is given by Jacobsen et al., (2011).

*Listing 20. Entries for waveProperties.input dictionary.*

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

seaLevel     0.00;

// A list of the relaxation zones in the simulation. The parameters are given
// in <name>Coeffs below.
relaxationNames (inlet outlet);

initializationName outlet;
```

```
inletCoeffs
{
    // Wave type to be used at boundary "inlet" and in relaxation zone "inlet"
    waveType    stokesFirst;

    // Ramp time of 2 s
    Tsoft       4;

    // Water depth at the boundary and in the relaxation zone
    depth       2;

    // Wave period
    period      4.0;

    // Phase shift in the wave
    phi         0.000000;

    // Wave number vector, k.
    direction  (1.0 0.0 0.0);

    // Wave height
    height      0.1;

    // Specifications on the relaxation zone shape and relaxation scheme
    relaxationZone
    {
        relaxationScheme Spatial;
        relaxationShape  Rectangular;
        beachType        Empty;

        relaxType   INLET;
        startX      (0 0.0 -1);
        endX        (5 0.0  1);
        orientation      (1.0 0.0 0.0);
    }
};
outletCoeffs
{
    waveType    potentialCurrent;
    U           (0 0 0);
    Tsoft       2;

    relaxationZone
    {
        relaxationScheme Spatial;
        relaxationShape  Rectangular;
        beachType        Empty;

        relaxType   OUTLET;
        startX      (25 0.0 -1);
        endX        (30 0.0  1);
        orientation      (1.0 0.0 0.0);
    }
};
// ************************************************************************* //
```

### 9.2.4. Field boundaries and initial conditions

Boundary and initial conditions are defined within the *0* directory (Figure 44). This directory contains one file per variable field considered in the case, i.e., *alpha.air*, *alpha.water*, *alpha.other*, *p_rgh* (dynamic pressure), *rho* (density), *T* (temperature), and *U* (velocity).

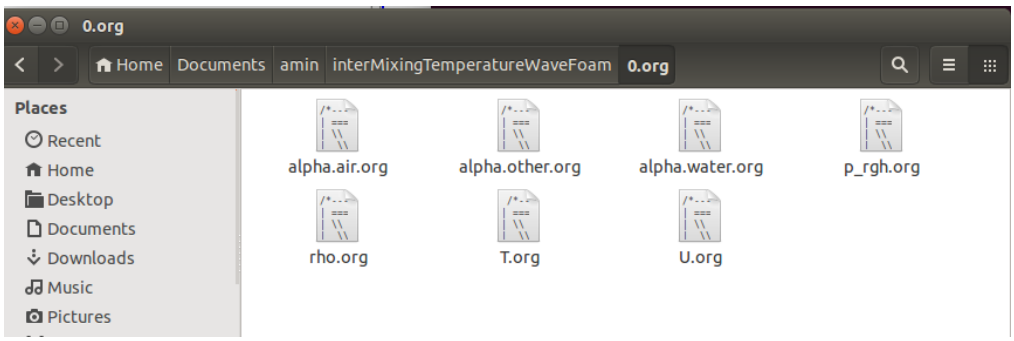


*Figure 44. Files contained in 0.org directory.*

Input dictionary *alpha.air* contains the boundary conditions for the air phase (Listing 21). Most of the boundary conditions are set as *zeroGradient*, which means that the gradient of the field is normal at that boundary. Patches *ojo* and *atmosphere* have an *inletOutlet* boundary condition, which means that flow can go outward or inward the domain. The patch *atmosphere* is in contact with the air phase, so it is given uniform values of one.

*Listing 21 Entries for the alpha.air input dictionary.*

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 0 0 0 0 0 0];
internalField   uniform 0;
boundaryField
{
    inlet
    {
        type            zeroGradient;
    }
    bottom
    {
        type            zeroGradient;
    }
    outlet
    {
        type            zeroGradient;
    }
    atmosphere
    {
        type            inletOutlet;
        inletValue      uniform 1;
        value           uniform 1;
    }
    ojo
    {
        type            inletOutlet;
        inletValue      uniform 0;
        value           uniform 0;
    }
```

```
    frontBack
    {
        type            zeroGradient;
    }
}
// ************************************************************************* //
```

*Alpha.water* dictionary refers to phase2 (Listing 22). The boundary conditions for this phase are similar to the ones in the *alpha.air* dictionary except for the *inlet* and *atmosphere* patches. The boundary condition for the *inlet* patch is *waveAlpha*, and it generates a wave train.

*Listing 22. Entries for the alpha.water input dictionary.*

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 0 0 0 0 0 0];
internalField   uniform 0;
boundaryField
{
    inlet
    {
        type            waveAlpha;
        refValue        uniform 0;
        refGrad         uniform 0;
        valueFraction   uniform 1;
        value           uniform 0;
    }
    bottom
    {
        type            zeroGradient;
    }
    outlet
    {
        type            zeroGradient;
    }
    atmosphere
    {
        type            inletOutlet;
        inletValue      uniform 0;
        value           uniform 0;
    }
    ojo
    {
        type            inletOutlet;
        inletValue      uniform 0;
        value           uniform 0;
    }
    frontBack
    {
        type            zeroGradient;
    }
}
// ************************************************************************* //
```

Finally, a*lpha.other* dictionary refers to phase3 (Listing 23).

*Listing 23. Entries for the alpha.other input dictionary.*

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 0 0 0 0 0 0];

internalField   uniform 0;

boundaryField
{
    inlet
    {
        type            zeroGradient;
    }
    bottom
    {
        type            zeroGradient;
    }
    outlet
    {
        type            zeroGradient;
    }
    atmosphere
    {
        type            inletOutlet;
        inletValue      uniform 0;
        value           uniform 0;
    }
    ojo
    {
        type            inletOutlet;
        inletValue      uniform 1;
        value           uniform 1;
    }
    frontBack
    {
        type            zeroGradient;
    }
}

// ************************************************************************* //
```

For its part, *p_rgh* input dictionary contains information for the dynamic pressure field, which is defined as $p\_rgh = p - (rho * gh)$; where $p$ is the total pressure, and $rho * gh$ represents the hydrostatic pressure. Patch *atmosphere* has a *totalPressure* boundary condition, which fixes the total pressure $p$ when the velocity $U$ changes (Listing 24).

*Listing 24. Entries for the p_rgh (pressure) input dictionary.*

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [1 -1 -2 0 0 0 0];
internalField   uniform 0;
boundaryField
{
    inlet
    {
        type            zeroGradient;
    }
    bottom
    {
        type            zeroGradient;
```

```
    }
    outlet
    {
        type            zeroGradient;
    }
    atmosphere
    {
        type            totalPressure;
        U               U;
        phi             phi;
        rho             none;
        psi             none;
        gamma           1;
        p0              uniform 0;
        value           uniform 0;
    }
    ojo
    {
        type            zeroGradient;
    }
    frontBack
    {
        type            zeroGradient;
    }
}
// ************************************************************************* //
```

*rho* (density) input dictionary is defined hereunder (Listing 25).

*Listing 25. Entries for the rho (density) input dictionary.*

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [1 -3 0 0 0 0 0];
internalField   uniform 0;
boundaryField
{
    inlet
    {
        type            zeroGradient;
    }
    bottom
    {
        type            zeroGradient;
    }
    outlet
    {
        type            zeroGradient;
    }
    atmosphere
    {
        type            zeroGradient;
    }
    frontBack
    {
        type            zeroGradient;
    }
    ojo
    {
        type            zeroGradient;
    }
```

```
}
// ************************************************************************* //
```

*T* (temperature) input dictionary is defined hereunder (Listing 26). The internal field is set with a temperature of $298\ \mathrm{K}$. Patch *ojo* has a constant temperature of $308\ \mathrm{K}$, and the liquid flowing through it will acquire this temperature.

*Listing 26. Entries for the T (temperature) input dictionary.*

```
//*********************************//

dimensions       [0 0 0 1 0 0 0];
internalField    uniform 298;
boundaryField
{
    inlet
    {
        type            zeroGradient;
    }
    bottom
    {
        type            zeroGradient;
    }
    outlet
    {
        type            zeroGradient;
    }
    atmosphere
    {
        type            zeroGradient;
    }
    ojo
    {
        type           fixedValue;
        value          uniform 308;
    }
    frontBack
    {
        type            zeroGradient;
    }
}
//*********************************//
```

Finally, the velocity field $U$ is initialized in the *U* input dictionary (Listing 27). The patch *inlet* has *waveVelocity* boundary condition, which generates the wave train velocity. The *bottom* and *outlet* patches have a *fixedValue* boundary condition, which sets the initial velocity value to zero. Atmosphere patch has a *pressureInletOutletVelocity* boundary condition, which is used for patches adjacent to the atmosphere. The patch that simulates the spring, *ojo*, has a *fixedValue* boundary condition to set the constant flow through this patch.

*Listing 27. Entries for the U (velocity) input dictionary*

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 1 -1 0 0 0 0];
```

```
internalField   uniform (0 0 0);

boundaryField
{
    inlet
    {
        type            waveVelocity;
        refValue        uniform ( 0 0 0 );
        refGradient     uniform ( 0 0 0 );
        valueFraction   uniform 1;
        value           uniform ( 0 0 0 );
    }
    bottom
    {
        type            fixedValue;
        value           uniform ( 0 0 0 );
    }
    outlet
    {
        type            fixedValue;
        value           uniform ( 0 0 0 );
    }
    atmosphere
    {
        type            pressureInletOutletVelocity;
        value           uniform ( 0 0 0 );
    }
    frontBack
    {
        type            zeroGradient;
    }
    ojo
    {
        type            fixedValue;
        value           uniform (0 0.2 0);
    }
}
// ************************************************************************* //
```

### 9.2.5. Initial field

Initial field is set in the input dictionary named *setFieldsDict* (Listing 28). Phase2 partially fills the computational domain. The rest of the domain is filled with air (alpha1). Additionally, the velocity field is initially set to 0 m/s.

*Listing 28. Input dictionary setFieldsDict .*

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
defaultFieldValues
(
    volScalarFieldValue alpha.air 1
    volScalarFieldValue alpha.other 0
    volScalarFieldValue alpha.water 0
);

regions
(
    boxToCell
    {
```

```
        box (0 -2 0) (30 0 0.1);
        fieldValues
        (
            volScalarFieldValue alpha.air 0
            volScalarFieldValue alpha.other 0
            volScalarFieldValue alpha.water 1
            volScalarFieldValue rho 1030
            volScalarFieldValue T 308
        );
    }
);
// ************************************************************************* //
```

Once the mesh is generated, the following commands are run:

```
setWaveParameters
setFields
```

The pre-processing utilities `setWaveParameters` and `setFields` use the information from *waveProperties.input* and *setFieldsDict* dictionaries, respectively. `setWaveParameters` utility reads *waveProperties.input* file and sets the wave parameters for a given wave theory based on a set of input variables. On the other hand, `setFields` utility set the values of the fluid phase that will be present in a certain domain region.

### 9.2.6. Control

The *controlDict* input dictionary contains data related to the time control and writing of the solution. This dictionary is located in the case *system* folder (Figure 45). In this case, the solution is given for 30 s of simulation, and the data is saved each 0.2 s (Listing 29).

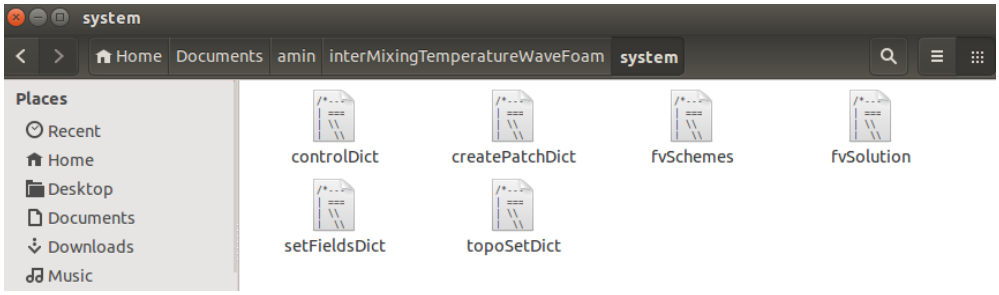


*Figure 45. Files contained in system directory.*

For this case Courant number of less of 1 is required; thus, the maximal Courant number is 0.5, and the maximal time step is 1 s.

*Listing 29. Entries for the controlDict input dictionary.*

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

application                 interMixingTemperatureWaveFoam;
startFrom                   latestTime;
```

```
startTime                 0;
stopAt                    endTime;
endTime                   30;
deltaT                    0.001;
writeControl              adjustableRunTime;
writeInterval             0.2;
purgeWrite                0;
writeFormat               ascii;
writePrecision            6;
writeCompression          uncompressed;
timeFormat                general;
timePrecision             6;
runTimeModifiable         yes;
adjustTimeStep            on;
maxCo                     0.5;
maxAlphaCo                0.5;
maxDeltaT                  1;

functions
{
    #includeIfPresent "../waveGaugesNProbes/surfaceElevationAnyName_controlDict";
}

// ************************************************************************* //
```

### 9.2.7. Solution schemes

The fvSchemes and *fvSolution* files must be adapted to the new variables and equations of the `interMixingTemperatureWaveFoam` solver, as shown in Listing 30 and Listing 31.

*Listing 30. Entries for the fvSchemes input dictionary.*

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

ddtSchemes
{
    default         Euler;
}
gradSchemes
{
    default         Gauss linear;
    grad(U)         Gauss linear;
    grad(alpha1)     Gauss linear;
}
divSchemes
{
    div(rhoPhi,U)  Gauss limitedLinearV 1;
    div(phi,alpha)  Gauss vanLeer;
  div(rho*phi*cpf,T)      Gauss upwind;
    div(phirb,alpha) Gauss interfaceCompression;
    div((muEff*dev(T(grad(U))))) Gauss linear;
}
laplacianSchemes
{
    default         Gauss linear corrected;
}
interpolationSchemes
{
    default         linear;
}

snGradSchemes
```

```
{
    default         corrected;
}
fluxRequired
{
    default         no;
    p_rgh;
    pcorr;
    "alpha.*";
    T;
}

// ************************************************************************* //
```

*Listing 31. Entries for the fvSolution input dictionary.*

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

solvers

{
    "alpha.*"
    {
        nAlphaCorr      2;
        nAlphaSubCycles 1;
        alphaOuterCorrectors yes;
        cAlpha          1;

        MULESCorr       yes;
        nLimiterIter    3;

        solver          smoothSolver;
        smoother        symGaussSeidel;
        tolerance       1e-8;
        relTol          0;
    }

    pcorr GAMG
    {
        tolerance        1e-7;
        relTol           0.0;

        smoother         DIC;//GaussSeidel;
        nPreSweeps       0;
        nPostSweeps      2;
        nFinestSweeps    2;

        cacheAgglomeration true;
        nCellsInCoarsestLevel 10;
        agglomerator     faceAreaPair;
        mergeLevels      1;
    };

    pcorr GAMG
    {
        tolerance        1e-7;
        relTol           0.0;

        smoother         DIC;//GaussSeidel;
        nPreSweeps       0;
        nPostSweeps      2;
```

```
        nFinestSweeps    2;

        cacheAgglomeration true;
        nCellsInCoarsestLevel 10;
        agglomerator     faceAreaPair;
        mergeLevels      1;
    };

    p_rgh GAMG
    {
        tolerance        1e-7;
        relTol           0.0;

        smoother         DIC;//GaussSeidel;
        nPreSweeps       0;
        nPostSweeps      2;
        nFinestSweeps    2;

        cacheAgglomeration true;
        nCellsInCoarsestLevel 10;
        agglomerator     faceAreaPair;
        mergeLevels      1;
    };

    p_rghFinal GAMG
    {
        tolerance        1e-8;
        relTol           0.0;

        smoother         DIC;//GaussSeidel;
        nPreSweeps       0;
        nPostSweeps      2;
        nFinestSweeps    2;

        cacheAgglomeration true;
        nCellsInCoarsestLevel 10;
        agglomerator     faceAreaPair;
        mergeLevels      1;
    };

  T
  {
   solver    BICCG;
   preconditioner   DILU;
   tolerance   1e-7;
   relTol   0;
  }

  TFinal
  {
      $T;
      tolerance       1e-07;
      relTol          0;
  }

    U PBiCG
    {
        preconditioner   DILU;
        tolerance        1e-09;
        relTol           0;
    };
    UFinal PBiCG
    {
```

```
        preconditioner   DILU;
        tolerance        1e-09;
        relTol           0;
    };
    gamma PBiCG
    {
        preconditioner   DILU;
        tolerance        1e-07;
        relTol           0;
    };
}
PIMPLE
{
    momentumPredictor yes;
    nOuterCorrectors 1;
    nCorrectors     3;
    nNonOrthogonalCorrectors 1;
}
relaxationFactors
{
    fields
    {
    }
    equations
    {
        ".*" 1;
    }
}
// ************************************************************************* //
```

### 9.2.8. Case running and post-processing

The case is run with the respective commands for mesh generation, wave parameters, field's settings, and the execution of `interMixingTemperatureWaveFoam`. Instead of running individually each application in the terminal, a bash script file can be created for this purpose. The applications that need to be run before `interMixingTemperatureWaveFoam` are:

1) *blockMesh*: generates a simple, unstructured mesh which represents the case physical domain.
2) *topoSet*: defines the cell set that will be used as a new patch.
3) *createPatch*: creates the patch that represents the spring at the bottom of the domain.
4) *setWaveParameters*: sets the wave parameters, e.g., wave number and wavelength.
5) *setFields*: sets the default values of the fluid phases within the domain.
6) *relaxationZoneLayout* (optional): this application is optional and can be used to visualize the correct specification of the relaxation zones in the domain.

Therefore, a bash script named *prepareCase.sh* is created to prepare the case before running the applications by copying the files of *0.org* folder into *0* folder of the case directory (Listing 32). A second script named *Allrun* is also created to run each application including the new solver developed (Listing 32). To run the complete case, type *./Allrun* in the terminal within the test case directory.

*Listing 32. prepareCase.sh bash script created for the case preparation with `interMixingTemperatureWaveFoam`.*

```
#!/bin/bash

### CLEAR THE 0 DIRECTORY
(cd 0; rm -f *)

### COPY RELEVANT FIELDS

cp 0.org/U.org 0/U
cp 0.org/p_rgh.org 0/p_rgh
cp 0.org/rho.org 0/rho

cp 0.org/T.org 0/T

cp 0.org/alpha.air.org 0/alpha.air
cp 0.org/alpha.other.org 0/alpha.other
cp 0.org/alpha.water.org 0/alpha.water
```

*Listing 33. Allrun bash script created for the case running with `interMixingTemperatureWaveFoam`.*

```
#!/bin/bash

# Source tutorial run functions
. $WM_PROJECT_DIR/bin/tools/RunFunctions

exec="prepareCase.sh"

# Set application name
application="interMixingTemperatureWaveFoam"

# Create the computational mesh
runApplication blockMesh

runApplication topoSet
runApplication createPatch -overwrite

# Compute the wave parameters
runApplication setWaveParameters

runApplication setFields

# Set the wave field
runApplication setWaveField

runApplication relaxationZoneLayout

# Run the application
runApplication $application
```

### 9.2.9. Post-processing

Post-processing is done in the software *Paraview*, which is an open-source, multi-platform visualization and data analysis application. Scalar and vector fields for each time step can be visualized to demonstrate the ability of the new `interMixingTemperatureWaveFoam` solver

to reproduce a wave train in a free surface condition and the inflow of a liquid with different density and temperature than the medium. For example, it can be visualized the case mesh, the distribution of the phases (alpha) in terms of the density $\rho$, the free-surface, the distribution of the temperature field, velocity field vectors and streamlines for each time-step.

Figure 46, Figure 47, and Figure 48 show some screenshots of the test case presented above to demonstrate the ability of the new `interMixingTemperatureWaveFoam` solver to reproduce a wave train in a free surface condition and the inflow of a liquid with different density and temperature than the medium.

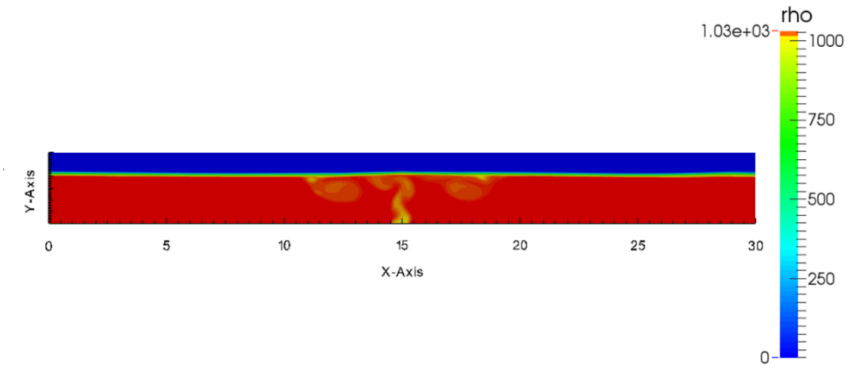


*Figure 46. Density field after 20 s of simulation.*

Figure 46 shows the distribution of the phases (alpha) in terms of the density $\rho$ after 20 s of simulation. Blue represents the gas phase (air), red the liquid phase with density $1030\ \mathrm{kg/m^3}$ (saltwater), yellow the liquid phase with density $1000\ \mathrm{kg/m^3}$ (fresh water), and orange their mixture. The thin layer in green and yellow that separates liquid phases from the gas phase is the interphase obtained by the VOF method, i.e., the free surface affected by the wave train and the submerged discharge.

The buoyancy effect is observed as a consequence of the submerged discharge of water with lower density than the medium. Thus, the formation of a layer or plume near the surface is observed. In this region, mixture between the liquid phases takes place, supported by the velocity field perturbation due to the incident wave train (Figure 47). Eddies near the influence zone of the submerged discharge are also present. The way these eddies are formed depends on the turbulence model used; thus, it is necessary a validation process.

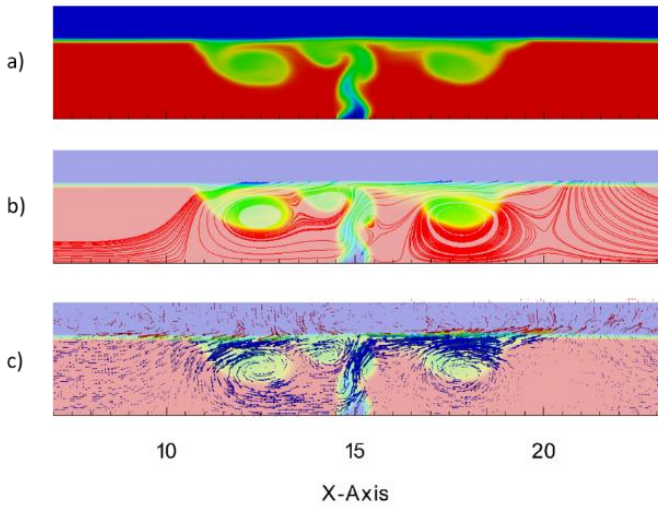


*Figure 47. Eddies and water plume after 20 s of simulation. In a) the three fluid phases and mixture between the liquid ones are shown, in b) the streamlines of the liquid phases, and in c) the velocity field vectors of the three phases.*

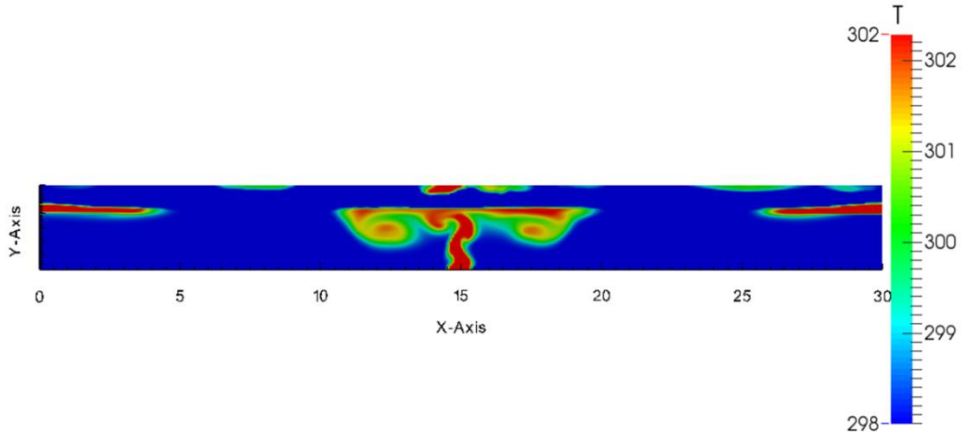


*Figure 48. Temperature field after 20 s of simulation.*

Finally, Figure 48 shows the distribution of the temperature field, which is assumed to be the same for the liquid and gas phase in time zero. This simplification is accepted since the study is focused on the liquid phases and heat exchange between air and water is not significant in this case.

On the other hand, the submerged discharge input has a higher temperature than the medium. This fact also influences the buoyancy effect; however, it is observed that the density gradient reigned this phenomenon instead of the temperature differences for this particular case. Nevertheless, the distribution of the temperature field is consistent with the distribution of the density field, and the temperature range is stable during the simulation time.

# 10. REFERENCES


Abraham, Ravindran, Abraham, Raju, 2002. The Indian 1 MW demonstration OTEC plant and the development activities. OCEANS 2, 1622–1628.

Asian Development Bank, 2014. Wave Energy Conversion and Ocean Thermal Energy Conversion: Potential in Developing Member Countries. Asian Development Bank, Philipines.

Avery, W.H., Wu, C., 1994. Renewable Energy from the Ocean: a Guide to OTEC, 1st ed. Oxford University Press, New York, New York.

Bardina, J.E., Field, M., Huang, P.G., Coakley, T.J., Field, M., Aeronautics, N., 1997. Turbulence Modeling Validation , Testing , and Development.

Berberović, E., Van Hinsberg, N.P., Jakirlić, S., Roisman, I. V., Tropea, C., 2009. Drop impact onto a liquid layer of finite thickness: Dynamics of the cavity evolution. Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys. 79. doi:10.1103/PhysRevE.79.036306

Brennen, C.E., 2005. Fundamentals of Multiphase Flows.

Bureau Veritas, 2018. Classification and Certification of Ocean Thermal Energy Converter ( OTEC ) - Tentative Rules - Guidance Note.

Chung T. J., 2010. Computational Fluid Dynamics, 2nd ed. Cambridge University Press, University of Alabama, Huntsville.

Cunningham, J.J., Magdol, Z.E., E. Kinner, N., 2010. Technical Readiness of Ocean Thermal Energy Conversion (OTEC), Coastal Response Research Center. Durham, NH.

Dessne, P., Golmen, L., Banerjee, S., Duckers, L., Blanchard, R., 2015. Otec Matters. University of Boras.

Energética, S. de I., 2010. Balance Nacional de Energía [WWW Document]. Secr. Energía, México. URL http://sie.energia.gob.mx/bdiController.do?action=cuadro&cvecua=IE7C04 (accessed 11.1.17).

Ferziger, J.H., Peric, M., 2002. Computational Methods for Fluid Dynamics, 3rd ed, Vasa. Springer. doi:10.1016/S0898-1221(03)90046-0

García Huante, A., 2015. POSIBLES EFECTOS OCEANOGRÁFICOS POR LA OPERACIÓN DE UNA PLANTA OTEC (OCEAN THERMAL ENERGY CONVERSION) EN LA ZONA DE PUERTO ÁNGEL, OAXACA, MÉXICO. Universidad Nacional Autónoma de México.

García Huante, A., Rodríguez Cueto, Y., Silva, R., Mendoza, E., Vega, L., 2018. Determination of the Potential Thermal Gradient for the Mexican Pacific Ocean. J. Mar. Sci. Eng. 6, 20. doi:10.3390/jmse6010020

Grandelli, P., Rocheleau, P.E.G., Hamrick, J., Chuch, M., Powell, B., 2012. Modeling the Physical and Biochemical Influence of Ocean Thermal Energy Conversion Plant Discharges into their Adjacent Waters, U. S. Department of Energy. Kailua, Hawaii.

Green, H.J., Guenther, P.R., 1990. Carbon Dioxide Release from OTEC Cycles, in: International Conference on Ocean Energy Recovery. Solar Energy Research Institute, Honolulu, Hawaii.

Hirt, C.W., Nichols, B.D., 1981. Volume of fluid (VOF) method for the dynamics of free boundaries. J. Comput. Phys. 39, 201–225. doi:10.1016/0021-9991(81)90145-5

Holz, M., Heil, S.R., Sacco, A., 2000. Temperature-dependent self-diffusion coefficients of water and six selected molecular liquids for calibration in accurate 1H NMR PFG measurements. Phys. Chem. Chem. Phys. 2, 4740–4742. doi:10.1039/b005319h

Ikegami, Y., Fukumiya, K., Okura, K., Jitsuhara, S., Uehara, H., 2002. Hydrogen Production Using OTEC, in: Proceedings of the International Offshore and Polar Engineering Conference. pp. 626–630.

Jacobsen, N.G., Fuhrman, D.R., Fredsøe, J., 2012. A wave generation toolbox for the open-source CFD library: OpenFoam. Int. J. Numer. Methods Fluids 1073–1088. doi:10.1002/fld

Khaligh, A., Onar, O.C., 2010. Energy harvesting : solar, wind, and ocean energy conversion systems. Taylor & Francis Group, United States of America.

Kim, J., Kim, H.-J., 2014. Numerical Modeling of Thermal Discharges in Coastal Waters, in: International Conference on Hydroinformatics. CUNY Academic Works.

Kobayashi, T., Tsubokura, M., 2011. Current Status on Large-Eddy Simulation for Engineering Applications 10, 149–152.

Lee, Hwang-ki, Kim, B., Kim, J., Kim, H., 2016. OTEC thermal dispersion in coastal waters of, in: Oceans 2016 - Shanghai. Oceans 2016, Shanghai, China.

Lee, Ho-saeng, Kim, H., Lee, S., 2016. Performance Characteristics of 20Kw Ocean Thermal Energy 1–7.

Lockheed Martin, 2014. Tapping into the Deep Blue for Green Electricity Nets Honor [WWW Document]. URL http://news.lockheedmartin.com/news-releases (accessed 4.25.17).

Makai Ocean Engineering, 2017a. Ocean Thermal Energy Conversion [WWW Document]. URL https://www.makai.com/ocean-thermal-energy-conversion/ (accessed 5.5.17).

Makai Ocean Engineering, 2017b. Ocean Thermal Energy Conversion [WWW Document].

Makai Ocean Engineering, 2008. Renewable Energy. Ocean Thermal Energy Convertion [WWW Document]. URL https://www.makai.com/ocean-thermal-energy-conversion/ (accessed 4.26.17).

Mcdonough, J.M., 2007. INTRODUCTORY LECTURES on TURBULENCE Physics , Mathematics and Modeling.

Miller, A., Rosario, T., Ascari, M., 2012. Selection and validation of a minimum-cost Cold Water Pipe material, configuration, and fabrication method for Ocean Thermal Energy Conversion (OTEC) systems. Soc. Adv. Mater. Process Eng. 28.

Multon, B., 2012. Marine Renewable Energy Handbook, First. ed. ISTE Ltd; John Wiley & Sons, Inc., Great Britain.

Mun, J.Y., Kim, H., Kim, J., 2014. Numerical Modeling Approach of an OTEC Thermal Discharges in Coastal Waters of Kosrae , Micronesia 3, 445–448.

Naval Energies, 2017. Ocean Thermal Energy Conversion [WWW Document]. URL

https://www.naval-energies.com/en/produit/ocean-thermal-energy-conversion/ (accessed 10.16.18).

Naval Group, 2014. Akuo Energy and DCNS awarded European NER 300* funding: a crucial step for the marine renewable energy sector [WWW Document]. URL https://www.naval-group.com/en/news/akuo-energy-and-dcns-awarded-european-ner-300-funding-a-crucial-step-for-the-marine-renewable-energy-sector/ (accessed 10.16.18).

Nihous, G.C., Vega, L.A., 1996. Performance test report: Analysis of Representative Time History Records obtained at the USDOE 210 kW OC-OTEC Experimental Facility in the Power Production Mode.

Nihous, G.C., Vega, L.A., 1994. Design of a 5 MW OTEC Pre-Commercial Plant.

Nihous, G.C., Vega, L.A., 1993. Design of a 100 MW OTEC hydrogen plantship. Mar. Struct. 6, 207–221.

Orszag, S.A., Patterson, G.S., 1972. Numerical Simulation of Three-Dimensional Homogeneous Isotropic Turbulence. Phys. Rev. Lett. 28, 76–79. doi:10.1103/PhysRevLett.28.76

OTEC news, 2014. 20kW OTEC pilot plant public demonstration in South Korea [WWW Document]. OTEC Found. URL http://www.otecnews.org/2014/01/20kw-otec-pilot-plant-public-demonstration-south-korea/

OTEC news, 2013. OTEC testing in the Indian Ocean [WWW Document]. OTEC Found. URL http://www.otecnews.org/2013/04/otec-testing-in-the-indian-ocean/ (accessed 4.27.17).

OTEC news, 2012. Advanced Composite Cold Water Pipe [WWW Document]. OTEC Found. URL http://www.otecnews.org/2012/07/advanced-composite-cold-water-pipe/

Patel, S., 2015. Largest OTEC Facility Inaugurated in Hawaii [WWW Document]. Power. URL http://www.powermag.com/largest-otec-facility-inaugurated-in-hawaii/

Ravindran, M., 2000. The Indian 1 MW Floating OTEC Plant [WWW Document]. Natl. Inst. Ocean Technonlgy. URL http://www.clubdesargonautes.org/otec/vol/vol11-2-1.htm

Rodríguez-Ocampo, P.E., 2016. Modelación numérica de flujos multifásicos: interacción del oleaje con descargas submarinas de agua dulce. UNAM, Mexico.

Schiestel, R., 2008. Modeling and Simulation of Turbulent Flows. ISTE. doi:10.1002/9780470610848

Stern, F., Wilson, R. V, Coleman, H.W., Paterson, E.G., 2001. Comprehensive Approach to Verification and Validation of CFD Simulations—Part 1: Methodology and Procedures. J. Fluids Eng. 123, 793–802.

Ten, S., Edwards, M., 2006. An Introduction to the Fundamentals of PMD in Fibers. White Pap. 12.

Tomislav, M., Höpken, J., Mooney, K., 2014. The OpenFOAM Technology Primer, First. ed. Sourceflux, Germany.

Vega, L.A., 2012. Ocean Thermal Energy Conversion. Encycl. Sustain. Sci. Technol. 36, 7296–7328. doi:10.4031/002533202787908626

Vega, L.A., 2010. Economics of Ocean Thermal Energy Conversion ( OTEC ): An Update, in: Offshore Technology Conference. Houston, Texas, pp. 3–6. doi:10.4043/21016-MS

Vega, L.A., 2002. Ocean Thermal Energy Conversion Primer. Mar. Technol. Soc. J. 36, 25–35. doi:10.4031/002533202787908626

Vega, L.A., 2000. LAND BASED OC-OTEC PLANT FOR THE PRODUCTION OF ELECTRICITY AND FRESH WATER : Conceptual Definition and Design Considerations for Pacific Island Nations.

Vega, L.A., 1999. OTEC overview [WWW Document]. URL http://www.otecnews.org/portal/otec-articles/ocean-thermal-energy-conversion-otec-by-l-a-vega-ph-d/

Vega, L.A., 1992. Economics of Ocean Thermal Energy Conversion (OTEC), in: Ocean Energy Recovery: The State of the Art, American Society of Civil Engineers. pp. 152–181. doi:10.1016/0960-1481(93)90047-K

Wang, Z., Tabeta, S., 2017. Numerical simulations of ecosystem change due to discharged water from ocean thermal energy conversion plant 1–5.

Wang, Z., Tabeta, S., Kitakoji, Y., Okamura, S., 2016. Numerical simulations for behavior of discharged water from ocean thermal energy conversion plant, in: Techno-Ocean. IEEE, Kobe, Japan, pp. 225–234.