Designing a Passive Cooling System for the PRISM Steam Generator

Christopher Bowman
Evan Gonzalez
Jesse Norris
Sam Olivier

NUEN 410
Texas A&M University
May 17, 2017

Designing a Passive Cooling System for the PRISM Steam Generator

Christopher Bowman
Evan Gonzalez
Jesse Norris
Sam Olivier

NUEN 410
Texas A&M University
May 17, 2017

Christopher Bowman	Date
Evan Gonzalez	Date
Jesse Norris	Date
Sam Olivier	Date
Dr. Mark Kimber	Date

Contents

Acronyms

ACS Auxiliary Cooling System. ii, 1, 2

BWR Boiling Water Reactor. 23

CFD Computational Fluid Dynamics. iii, 24, 28, 29, 31, 32, 55, 59, 62–64

DOE Department of Energy. ii, 1

EOS Equation of State. 34, 39

ESBWR Economic Simplified Boiling Water Reactor. 6

GCI Grid Convergence Index. 32, 45

GEH General Electric Hitachi Nuclear Energy. ii, 1, 6, 78

HPRC High Performance Research Computing. 32, 43

LEU Low–Enriched Uranium. 48, 49

LMFBR Liquid Metal Fast Breeder Reactor. 51

LOCA Loss of Coolant Accident. 23

LWR Light Water Reactor. 2, 48, 54

NRC Nuclear Regulatory Commission. ii, 1, 53

PRISM Power Reactor Innovative Small Module. ii, iii, 1, 3, 4, 24, 30, 31, 47–49, 51–54, 56, 58, 61–64

RANS Reynolds Averaged Navier Stokes. 33, 37

RCIC Reactor Core Isolation Cooling. 23

RVACS Reactor Vessel Auxiliary Cooling System. ii, iv, 1, 2, 4, 55, 57, 59, 62

SIMPLE Semi Implicit Method for Pressure Linked Equations. 34

TIM Temperature Increment Method. 18, 21

U–TRU–Zr Uranium–Transuranic–Zirconium. 48, 54, 62

UNF Used Nuclear Fuel. iii, 48, 49, 52, 53

Executive Summary

The Power Reactor Innovative Small Module (PRISM) is a pool–type, metal–fueled, small modular, sodium–cooled fast reactor designed by General Electric Hitachi Nuclear Energy (GEH). The design was first conceptualized in the 1980s as a part of the US Advanced Liquid Metal Reactor program administered by the Department of Energy (DOE) [1]. As with all Generation IV nuclear power plants, passive safety and reliability are of the utmost concern. The PRISM design, nearing the end of its Nuclear Regulatory Commission (NRC) review process, has implemented several systems to achieve these standards.

The Auxiliary Cooling System (ACS) transfers heat from the intermediate sodium loop to the tertiary steam loop in order to generate mechanical work, which can then be converted to electricity. In the event of station blackout, this system supplements the primary core cooling system known as the Reactor Vessel Auxiliary Cooling System (RVACS). The ACS is currently designed to allow natural air convection to remove excess decay heat. However, GEH concluded that this natural convection system is insufficient in removing excess decay heat during station blackout conditions.

GEH has considered adding a blower to cool the steam generator with forced convection, however this is no longer a passive system. Should a loss of power occur, the blower would not be able to operate. This would induce a drastic temperature transient in the steam generator causing damage.

This project seeks to design a fully passive ACS, focusing on the steam generator. As mentioned previously, the ACS was designed to use only the natural convection of air. This was done by introducing a metal shroud around the steam generator with openings for air to flow in through the bottom and out through the top.

The proposed solution is to passively cool the PRISM steam generator by diverting steam from the tertiary loop to a safety-related, bypass loop. The diverted steam runs through a

Terry turbine that mechanically connects to a blower located above the steam generator. The Terry turbine was chosen due to its ability to safely handle two–phase flow. This ameliorates the safety risk of damaging the Terry turbine in suboptimal conditions.

Two bypass loop designs were considered. In both designs, a fraction of the steam produced in the still hot steam generator is extracted to a Terry turbine and then vented. In the first design, a bypass loop is formed where steam that is not diverted to the Terry turbine is sent through a passive condenser and returned to the bottom of the steam generator through natural circulation. The second design is an open, once through system where the steam not diverted to the Terry turbine is vented.

To facilitate forced convection heat transfer, the affect of the steam generator's surrounding shroud was investigated. These designs focused on the distance between the steam generator's outer wall and the surrounding shroud. The main designs were: a constant gap and a linearly increasing gap. These designs were tested in a Computational Fluid Dynamics (CFD) model of the steam generator and shroud.

The most optimal design was chosen through a four part analysis of the system. This analysis consisted of Thermodynamics Analysis, Optimization of Turbine-Driven Blower, Determination of Heat Removal from the Steam Generator, and determination of PRISM's decay heat curve. The Thermodynamics Analysis section determined the available, time dependent steam mass flow exiting the steam generator. This function was then used to determine the optimal amount of steam extracted to power the Terry turbine, the resulting Terry turbine power output, and the fan curve of the mechanically powered blower. This fan curve was then used in a CFD analysis of the outside of the steam generator to determine the heat removal rate of the design. In addition, a PRISM Used Nuclear Fuel (UNF) recycling fuel assembly was modeled and burned in MCNP with the intent to characterize the decay heat power. Ultimately, this became unnecessary when more complete PRISM decay heat data was found. This shifted the focus to evaluating the overall feasibility of the system to

remove the necessary decay heat power.

In this analysis it was concluded that the open, once through design was more optimal than the closed design as it is less complex and less expensive. The most optimal steam generator geometry was chosen to be the linearly increasing steam generator shroud gap.

The feasibility of the design was proved by comparing the total energy outputted by the core to the total energy removed by RVACS and the Terry turbine driven blower as function of time. Through this it was shown that, while RVACS alone is sufficient to remove the excess decay heat, it results in the steam generator increasing 260 °C past its operating temperature. With the addition of the Terry turbine driven blower, this temperature increase is reduced to 100 °C. In addition, the total excess decay heat is fully removed in 6.9 hours, opposed to 16.3 hours with RVACS alone. This shows that our design goal of passively protecting the steam generator during station blackout conditions has been met and that the Terry turbine driven blower and once through bypass loop design is feasible.

1 Introduction

The Power Reactor Innovative Small Module (PRISM) is a pool–type, metal–fueled, small modular, sodium–cooled fast reactor designed by General Electric Hitachi Nuclear Energy (GEH). The design was first conceptualized in the 1980s as a part of the US Advanced Liquid Metal Reactor program administered by the Department of Energy (DOE) [1]. As with all Generation IV nuclear power plants, passive safety and reliability are of the utmost priority. The events at the Fukushima Daiichi station raised new concerns as to system performance in beyond design basis accident scenarios. This includes extended station blackout, the primary driver of the accident. The PRISM design, nearing the end of its Nuclear Regulatory Commission (NRC) review process, has implemented several systems to achieve these standards.

Each PRISM module has two sodium loops and one steam loop. The use of sodium allows for higher temperatures and heat transfer within the steam generator resulting in a higher overall plant thermal efficiency [2]. The low vapor pressure of sodium reduces the pressure in the sodium loops and reactor vessel, reducing the risk of pipe rupture and release of radioactive material. An intermediate sodium loop works to isolate the radioactivity induced in the primary loop, while maintaining its advantageous qualities [3].

Figure 1 details the heat removal systems for PRISM. The Reactor Vessel Auxiliary Cooling System (RVACS) acts on the primary sodium loop to remove heat generated by fission from the core to the intermediate sodium loop. Supplementary cooling of heat expelled from the heat exchanger is provided by natural convection of air.

The Auxiliary Cooling System (ACS) transfers heat from the intermediate sodium loop to the tertiary steam loop in order to generate mechanical work, which can then be converted to electricity. The ACS is currently designed to allow natural air convection to remove excess decay heat. However, GEH concluded that this natural convection system is insufficient in

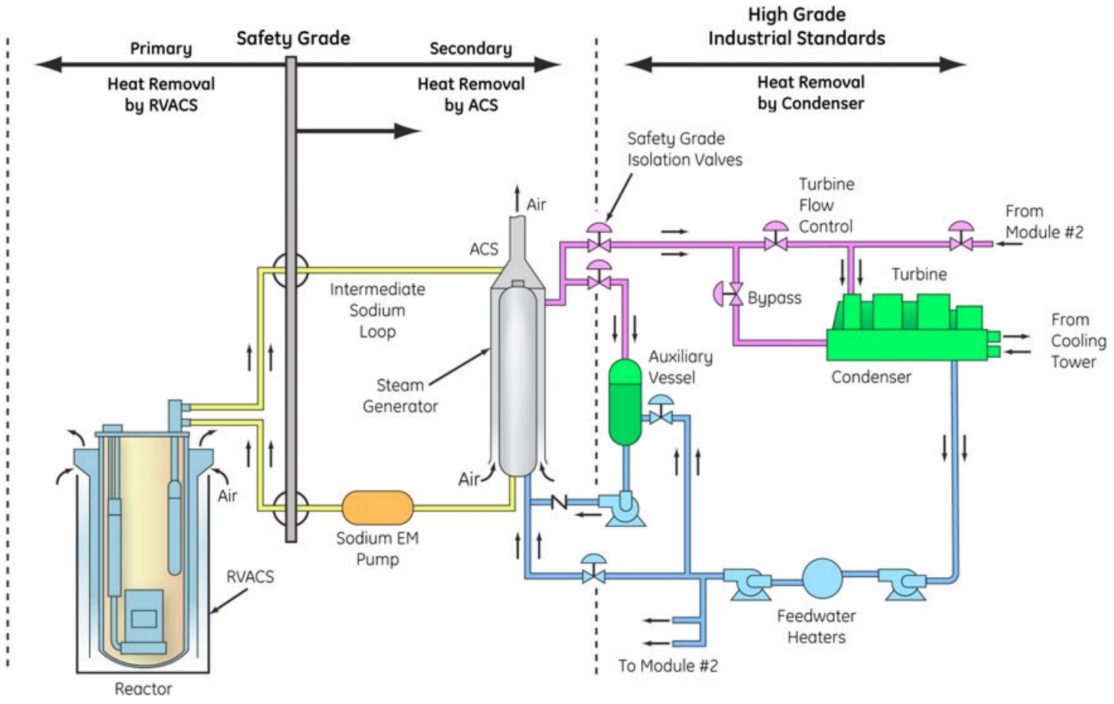

Fig. 1. A diagram of the Nuclear Supply Steam System for the PRISM reactor [1].

removing excess decay heat during shutdown, maintenance, or refueling.

Figure 2 shows the decay heat as a function of time for Light Water Reactors (LWRs). RVACS is capable of removing up to 1% of the core's thermal power rating. Based on Fig. 2, the system will be safe beyond approximately 2.75 hours.

This project seeks to design a fully passive ACS, focusing on the steam generator, capable of protecting the steam generator by removing excess decay heat for at least the first 2.75 hours after station blackout. The proposed design is to add a Terry turbine driven blower to the top of the steam generator. This design will include modifications to the tertiary loop to extract steam produced by the still hot steam generator. The extracted steam then powers a Terry turbine that is mechanically connected to a blower located above the steam generator. The blower cools the outside of the steam generator by pulling air along its outer shell.

Two tertiary loop modifications are proposed: an isolated, closed bypass loop and an open, once–through design. In the closed design, the steam extracted to the Terry turbine will be

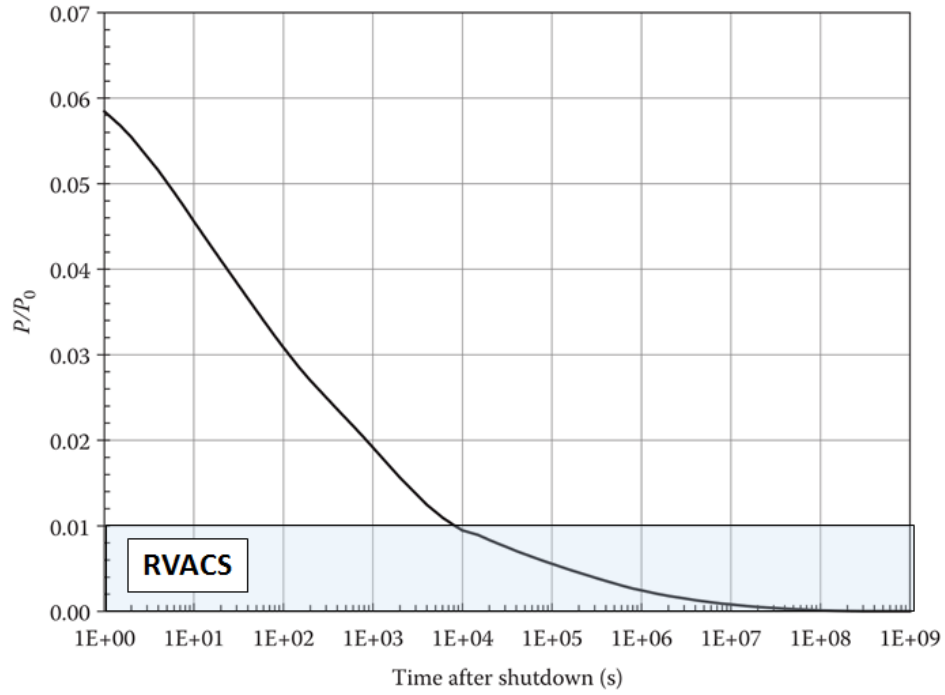

Fig. 2. A plot of the ANS standard decay curve [4].

condensed in a passive condenser before being returned to the bottom of the steam generator. The open, once–through design vents the un-extracted steam. In both cases, the extracted steam is vented after running through the Terry turbine.

The most optimal design will be chosen and evaluated through a four part analysis of the system. This analysis consists of Thermodynamics Analysis, Optimization of Turbine-Driven Blower, Determination of Heat Removal from the Steam Generator, and the determination of the PRISM decay heat power. The following sections present the objectives of the design, the selection and analysis of the most optimal designs, a discussion on the feasibility of the design, and our conclusions and future work.

2 Objectives

The purpose of this project is to increase the safety and reliability of the PRISM system during station blackout by designing a fully passive system to remove excess decay heat from

the steam generator. To be successful, the design must, in concert with RVACS, remove the excess decay heat generated in the core, prevent damaging temperature transients in the steam generator, and operate without on-site power.

3 Approach

The analysis of the Terry turbine driven blower has been separated into four primary areas: thermodynamic analysis of the tertiary loop, optimization of the turbine driven blower, determination and maximization of heat removal on the outside of the steam generator, and determination of the PRISM decay heat power. This section presents the analysis for the selection of the optimal tertiary loop modifications, the resulting mass flow rate of steam available for extraction to the Terry turbine, the process used for optimizing the amount of steam extracted to the Terry turbine, the resulting power delivered to the blower, and the maximization and determination of the blower induced heat removal rate on the outside of the steam generator. A fuel assembly model in MCNP and the resulting fuel composition from a core burn is presented. Finally, the PRISM decay heat power is evaluated.

3.1 Thermodynamics Analysis

3.1.1 Rankine Cycle Theory

An ideal Rankine cycle with superheat is what is being used to model the tertiary loop of the original system. Normally, sub-saturated water enters the steam generator where it is heated to a saturated vapor before being additionally heated to make it a superheated fluid [5]. The steam then expands back to a saturated mixture through a large turbine to the point where it is under the dome on the T–s diagram. The fluid then goes through a condenser which returns it to saturated water. The saturated water is then pumped back

into the steam generator and the cycle continues. The ideal Rankine cycle is an important assumption to make. It allows for entropy to be treated as constant through the turbine and the pump which is necessary to be able to solve for the enthalpies at all the different points in the loop.

In case of a station blackout emergency, the bypass loop being proposed would kick into place. All the water from the tertiary loop would be diverted through the bypass loop. As mentioned earlier, two different system designs are being looked at to be able to remove the necessary amount of heat until RVACS can remove all the decay heat still being produced by the core on its own. These two designs are the Once–Through System Design and the Continuous Loop Design.

An assumption to either of these designs is that the original condenser that is actively powered is elevated so that water will continue to flow to the steam generator from the condenser after the station blackout occurs. For this project, without more information on the original system design, there was too much difficulty to try to figure out how the system will behave after the station blackout occurs.

3.1.2 Design of a Bypass Loop

Continuous Loop Design The first design looked at was the Continuous Loop Design shown below in Figure 3. This loop would also use ideal Rankine cycle with superheat. However, now some of the steam leaving the steam generator would travel through a Terry turbine such as the one shown in Fig. 7 that is connected to a blower on top of the steam generator. The rest of the steam would be vented to the environment. The steam that passes through the Terry turbine would then go through a passive condenser to maintain the passivity of the system before using gravity as a pump to return to the steam generator and continue the new bypass loop.

One difficulty of this idea is that it requires a passive way to cool the steam leaving the Terry

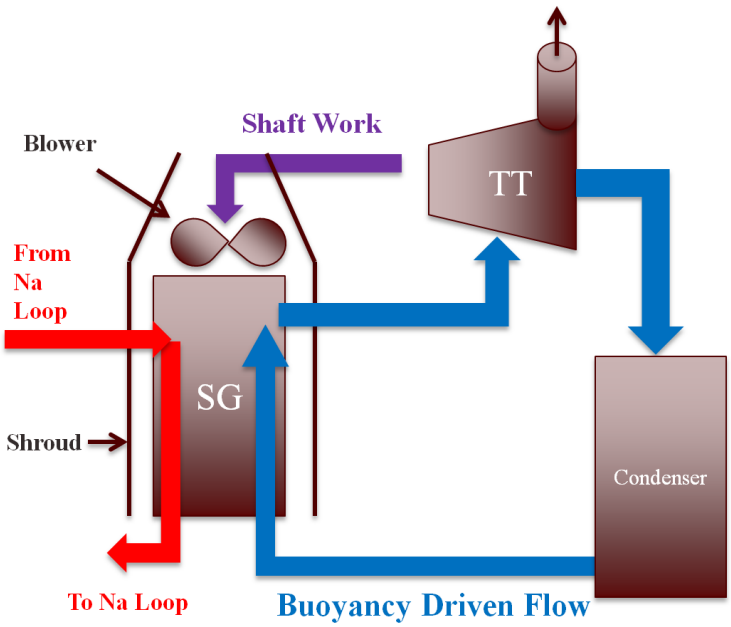

Fig. 3. A diagram of the Continuous Loop Design for the steam generator. The Once–Through Design is the same, but all the steam from the Terry turbine is vented.

turbine in the bypass loop. An AREVA passive condenser or a condensing pool such as in the Economic Simplified Boiling Water Reactor (ESBWR) design are two options. GEH has made it known that it would be preferable to keep the passive safety system as simple as possible. However if the Continuous Loop Design is to be utilized, the steam must be able to be condensed back to saturated water before re-entering the steam generator to be able to solve the system.

The AREVA Emergency Condenser shown below in Fig. 4 is one option for the passive condenser for the bypass loop system. The Emergency Condenser was successfully tested in the full scale within the full pressure range between 0.5 and 8.5 MPa and showed high heat transfer capacities of up to 77 MW [6]. It meets the requirements of being capable of long-term passive heat removal with no active coolant injection system or power supply necessary. The Emergency Condenser works by allowing steam from the vessel to condense inside the U-tube passive condenser, transferring the heat to a water pool and the leading the condensate back to the vessel by gravity [6]. A disadvantage to this condenser is that

Fig. 4. This is an Areva Passive Condenser under construction. [6]

it was designed for a reactor pressure vessel or the primary loop in a PWR. It would be necessary to look into whether it would also work in this bypass loop. Also, a water pool would have to be added for the condenser to sit in, further complicating the system.

Another issue with this design is that it would be difficult to model how the thermodynamic properties change with time. In the original cycle, a constant amount of heat is added to the cycle through the sodium entering the steam generator. Under station blackout, the power being produced from the reactor is solely from decay heat which decreases exponentially with time. As time increases, less and less power will be delivered to the steam generator. This means the steam leaving the steam generator will have a lower quality as well. This is one of the reasons a Terry turbine was selected to power the blower on top of the steam generator in the bypass loop. Terry turbines have been shown to be able to withstand lower qualities of steam to a much larger extent than a normal turbine due to having thicker blades or operating at a lower pressure. Since the power delivered to the steam generator is changing, it would be necessary to recalculate the new temperatures and pressures of the water leaving the steam generator for each cycle through the loop. Additionally, as the steam travels through the Terry turbine and powers the blower, the blower will begin to cool the steam generator and reduce the temperature of the water leaving the steam generator, lowering

its quality, leading to further complications of recalculating the thermodynamic properties throughout the loop as time progresses.

A way of attempting to find the mass flow rate of the Continuous Loop Design was to use a type of fluid mechanics analysis. Normally, a company would come up with its own system curve. They would then obtain the different pump curves from the vendors and find the intersection of the chosen pump curve and the system curve. That intersection would define the operating point for the pump. This operating point dictates the volumetric flow rate and the pressure difference through the pump. This is also important for the thermodynamic analysis of the cycle because the pump work is needed to know to properties of the water entering the steam generator and to continue solving the system for each cycle. The volumetric flow rate would also allow for the determination of the velocity of the water flowing through the pipes and the Reynolds number. By knowing the velocity the fluid is traveling at throughout the loop using the buoyancy–driven flow instead of pumps, it is possible to calculate the number of cycles needed to remove the necessary amount of heat for the required amount of time.

A complication to the fluid mechanics analysis of this design is that the bypass loop utilizes buoyancy–driven flow. This is essential because it eliminates the need for a pump which is an active component and this system is designed to be fully passive.

In Fig. 6, an example on a system curve intersecting with a pump curve is shown. The operating point is labeled at around 45 gpm of flow and 75 ft of total head. A system curve was then attempted to be built for the bypass loop system which is shown in Fig. 5. Unfortunately, since the bypass loop utilizes buoyancy–driven flow, there was no data on pump curves for buoyancy–driven flow to plot against the system curve. Therefore, there was no way to find the operating volumetric flow rate and head loss other than to pick some point on the system curve and justify why that is considered the operating point for this particular buoyancy–driven flow system.

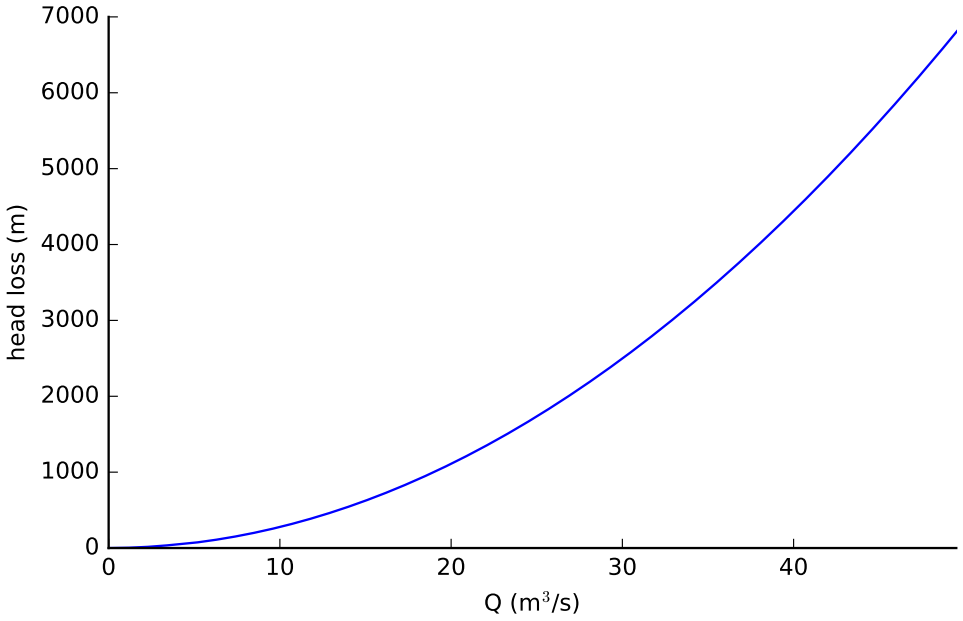

Fig. 5. A plot of the system curve of Bypass Loop.

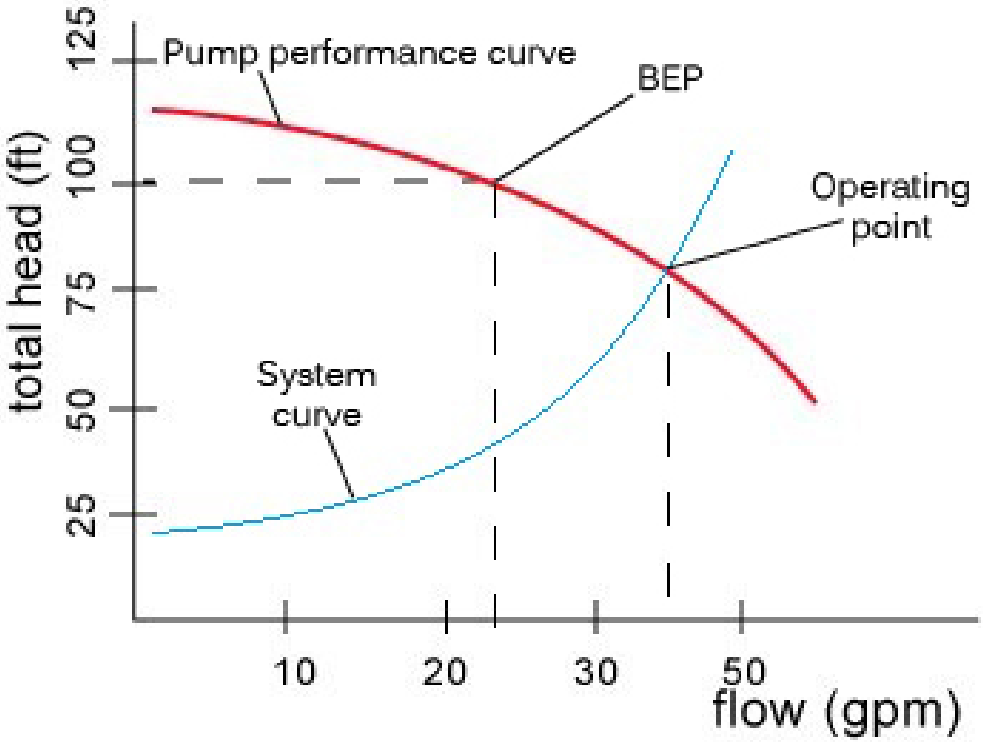

Fig. 6. An example of a pump curve intersecting a system curve. [7]

Fig. 7. An example of a Terry turbine. [8]

Once–Through System Design The other design is the Once–Through System Design. In this design, the steam leaving the steam generator will be still be run through the Terry turbine. However, after it passes through the Terry turbine, instead of being condensed and returning to the steam generator, all the steam is released to the environment. The main concern is if the system will be able to run long enough for the decay heat being produced from the core decreases to the point where RVACS can remove all of it by itself.

This design is advantageous because, unlike the Continuous Loop Design, the only additional complexity is the addition of the Terry turbine to power the blower on top of the steam generator. There is no requirement for an additional passive condenser and path of return to the steam generator because the steam does not need to be condensed after going through the Terry turbine.

Unfortunately, since the steam is not being condensed and returned to the steam generator, the system is continuously losing steam. This means the system will eventually run out of

water to send through the steam generator to remove heat from the core. The main proof of concept calculation that must be done is to prove this system will last for 2.75 hours when the decay heat power is down to below 1% of the original power. The reason all the steam is vented rather than put back through the original condenser is because the original condenser is actively powered and a station blackout eliminates all on–site power and adding a tank of water would increase the complexity of the system which is not desirable.

Scoping Analysis The original estimate on how long this design would last was 23.3 hours. First, the mass of the water was estimated based on an estimated length of pipe and volume of water in the condenser. Then, the change in enthalpy through the steam generator was calculated by tracking the steam as it left the steam generator right before the station blackout occurred. Based on the change in enthalpy and estimating the decay heat right after the station blackout to be 7% of the original power and subtracting 10 MW that is removed by RVACS, the mass flow rate was estimated to be 22 kg/s. By dividing the estimated mass of water in the system by the mass flow rate after the station blackout occurred, the original estimate for how long the Once–Through Design was obtained.

However, there are a multitude of problems with this estimate due to the assumptions made. Some problems with this original estimate are if the pumps continue to run for a little while after the station blackout occurs, if the steam have momentum/how long will it take to slow down to 22 kg/s from 759 kg/s, is the amount of water in the system accurate, and how will the decay heat decreasing affect the change in enthalpy, density, Q and therefore the mass flow rate,

If the pumps continue to run for a period of time even as short as ten minutes after the station blackout occurs when the water is supposed to be slowing down to around 20 kg/s, then the system will lose approximately 7500 kg of water in those ten minutes rather than the 200 kg that this original estimate was counting on. Losing that much more water in the first ten minutes than expected leads to the conclusion that this would significantly reduce

the time the Once–Through System Design would work for.

In order to estimate how much water is in this system, the estimated amounts of water in each component was summed along with the estimated amount of water in the piping between the components.

The outer diameter of the steam generator we are modeling is 4.2672 and the inner diameter of it is 4.191 m [9]. Based on Fig. 8 below, the diameter of the piping the water is flowing through in the middle is then about 1.2192 m. Treating the water in the steam generator as a cylinder, the volume of water in the steam generator can be found using the equation below

$$V_{SG} = \frac{\pi}{4} D^2 L = \frac{\pi}{4}(1.2192)^2(22.9362) = 26.776 \text{ m}^3 \quad , \tag{1}$$

where L is the length of the steam generator which is 22.9362 m [9].

In order to try to account for the other problems associated with this original estimate, a Python script was written to estimate the thermodynamic properties in the loop and try to predict how they will cycle from cycle to cycle.

Original Python Loop Analysis To solve for the thermodynamic properties at the different positions in the cycle, a Python script was developed that utilized the IAPWS95 package. This package allows the user to input the temperature and pressure for superheated steam or temperature or pressure and quality for saturated steam and the package will provide the thermodynamic properties such as enthalpy, entropy and density. The script starts by assuming that right after the station blackout occurs, the temperature and pressure leaving the steam generator is the same as before. Then, using the ideal Rankine cycle assumption of constant entropy,

$$s_3 = s_4 \tag{2}$$

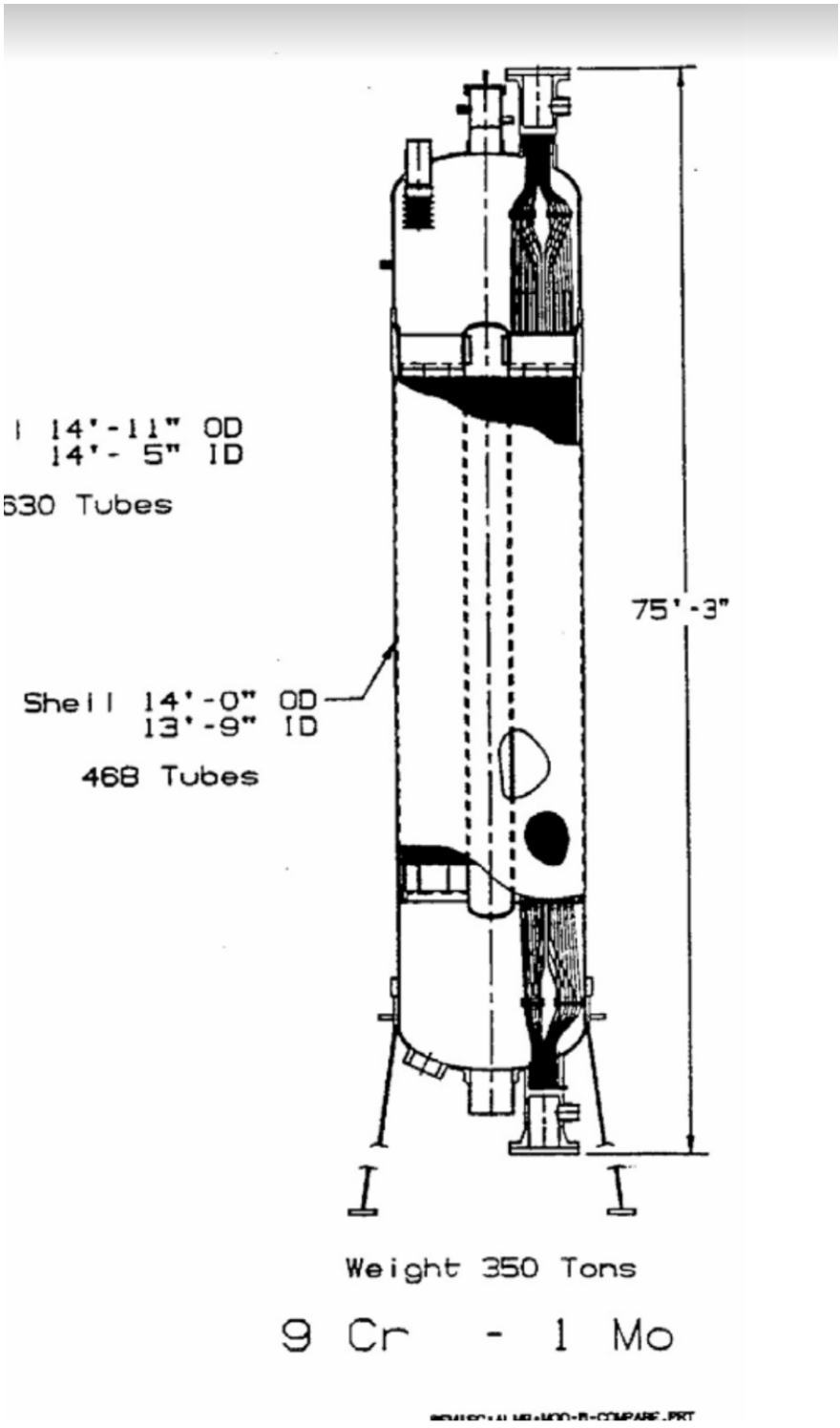

Fig. 8. An 850 MWt MOD-B SG using 9CR-1MO steel. [9]

the quality of the steam leaving the turbine is calculated using the equation

$$x_4 = \frac{s_4 - s_f}{s_{fg}} \quad , \tag{3}$$

Using the quality, the enthalpy of the steam leaving the steam generator can then be calculated by

$$h_4 = h_f + x_4 h_{fg} \quad . \tag{4}$$

The thermodynamic properties of the steam leaving the condenser were found by assuming that the steam leaving the condenser has been fully condensed back to liquid water with a quality of zero and the temperature was the original temperature of the water entering the steam generator.

The original temperature of the water entering the steam generator is 489 K. For this problem, the assumption was made that the water entering the steam generator after station blackout was saturated liquid, meaning the pressure is 2.141 MPa. A document that was provided by GE Hitachi stated the original pressure drop through a 479 MWth Helical Coil steam generator for ALMR plant consideration was 0.5654 MPa. Because of this, we assumed a pressure drop of 0.2757 MPa for after the station blackout to find an estimated mass flow rate through the steam generator after the station blackout occurs. The mass flow rate was calculated by starting with the conservation of energy equation

$$\frac{dE}{dt} = \dot{E}_{in} - \dot{E}_{out} + Q_{ext} \quad , \tag{5}$$

which is then simplified to

$$\dot{m} = \frac{Q}{\Delta h} \quad . \tag{6}$$

To find the mass flow rate right after the accident, an estimate of Q being 50 MWth was used. This number was obtained by taking 7% of the decay heat and subtracting 10 MWth that is removed by RVACS. Using the IAPWS95 package in Python, the enthalpy of the saturated

water entering the steam generator at 489 K is 924.43 kJ/kg. Taking the saturation pressure of the water entering the steam generator of 2.141 MPa and adding 40 psi (0.27579 MPa) to that and combining that with the normal outlet temperature of 725 K gives an outlet enthalpy of 3356.78 kJ/kg. Therefore the change in enthalpy is

$$\Delta h = h_3 - h_2 = 3356.78 - 924.43 = 2432.26 \text{ kJ/kg} \quad .$$

Using this change in enthalpy through the steam generator and a value of 50 MWth being supplied to the steam generator, the mass flow rate is estimated to be

$$\dot{m} = \frac{Q}{\Delta h} = \frac{50 \text{ MWth}}{2432.26 \text{ kJ/kg}} = 20.56 \text{ kg/s} \quad .$$

The difficulty in this problem is then in determining how the thermodynamic properties change throughout the steam generator due to there being two-phase flow. Since the fluid is transitioning from a liquid to a gas, its values of density, quality, temperature and mass flow rate are constantly changing. Since the ultimate goal of the thermodynamic analysis of the system is to determine the mass flow rate and therefore the time the system will last after a station blackout accident, it is imperative to find a way to solve for the mass flow rate through the steam generator.

Two–Phase Mass Flow Rate Method The other way that was attempted to calculate the mass flow rate as a function of decay heat power from the reactor was to use Eq. 7 which was found in the article A Comparative Study of Single-Phase, Two–Phase, and Supercritical Natural Circulation in a Rectangular Loop [10]

$$w = \left\{ \frac{2D^{1+b}\rho_L^2 \beta g Q_h H A^{2-b}}{p \phi_{LO}^2 L_{eff} \mu_L^b} \right\}^{1/3-b} \quad , \tag{7}$$

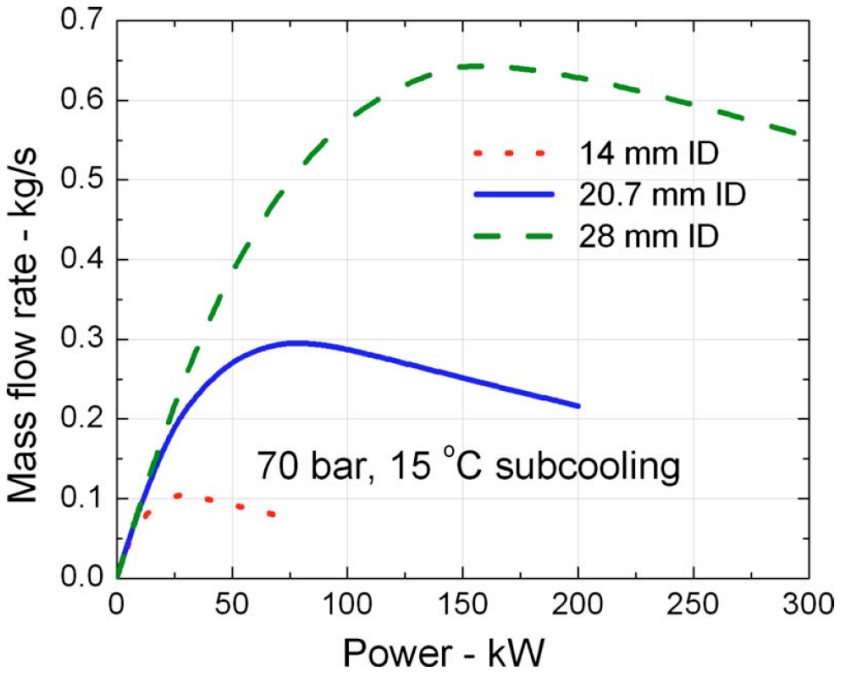

Fig. 9. Effect of loop diameter on steady state performance of two-phase loops. [10]

where w was the mass flow rate in kg/s, D was the diameter of the pipe, ρ was the density, g was gravity, Q was the decay heat power from the reactor, H was the thickness of the distance between the pipes in the article, but here it was assumed to be the diameter, A was the cross–sectional area of the steam generator, p was the pressure of the steam generator, ϕ_{LO} was the two–phase factor, L_{eff} was the effective length of the steam generator, and μ_L was the kinematic viscosity of the fluid flowing through the steam generator. The exponent b is 0.25 for turbulent flow in this case [10]. The results in the article from using this equation are shown below in Fig. 9. The equation used to determine β is

$$\beta = \frac{1}{\bar{v}} \frac{v_{fg}}{h_{fg}} \quad , \tag{8}$$

and the equation

$$\phi_{LO} = \sqrt{(1 + \frac{v_{fg}}{v_f} x)^2} \quad , \tag{9}$$

was used to calculate the two–phase factor throughout the steam generator [10].

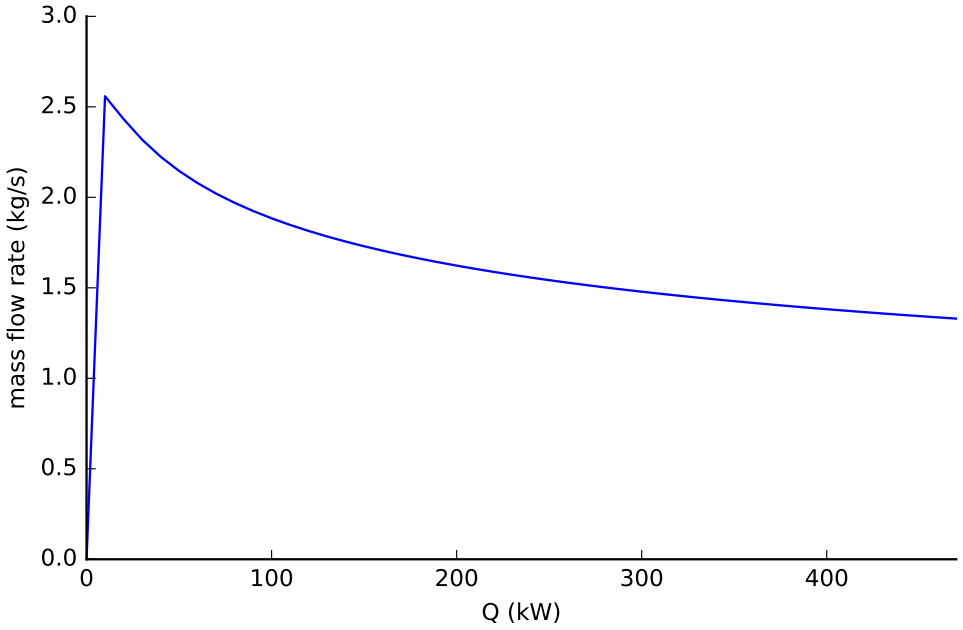

Fig. 10. Decay heat power vs mass flow rate using Eq. 7.

For this method, it was attempted to recreate the plot from was found in the article A Comparative Study of Single-Phase, Two–Phase, and Supercritical Natural Circulation in a Rectangular Loop, but with numbers for the PRISM steam generator. Unfortunately, this was not as simple as hoped. The terms of Q, μ_L, ϕ_{LO}, β, and pressure all change throughout the steam generator so it is more difficult than just plugging a few numbers in.

In order to be able to solve this equation and plot the mass flow rate for various decay heat power values, it was assumed that the heat addition process is isobaric and that the quality increased linearly from zero at the bottom to one at the top. This way, the pressure and quality were known everywhere in the steam generator, which allowed the two–phase factor and density to be determined throughout the steam generator.

Using the assumptions listed above, Fig. 10 was created by inputting multiple decay heat power values into Eq. 7 and solving for the mass flow rate using the numbers for this system design rather then the experiment performed in the article [10].

Temperature Increment Method (TIM) The ultimate goal of the thermodynamics analysis for this project is to find the mass flow rate through the steam generator as a function of time to pass off to the Terry turbine analysis. As discussed earlier, two–phase flow is a very complicated thing to model due to factors such as density, quality, viscosity, velocity, and the specific heat of a fluid that change as a fluid transitions from a liquid to a solid in ways that can sometimes lead to large amounts of error being introduced to a computer model. For instance, as a liquid, water has a density of around 1000 kg/m^3, but a vapor, its density drops to around 1 kg/m^3 which is a difference of 3 orders of magnitude.

In this method, it was attempted to find the mass flow rate by starting with the temperature at the inlet to the steam generator and assuming the quality is zero and then finding the temperature a certain distance along the steam generator away from that using the finite difference method. First, an array of Q values, the decay heat from the reactor is defined and used to find the heat flux by using the equation

$$q'' = \frac{Q}{\pi D L} \quad . \tag{10}$$

For each value of Q, a temperature array is defined to collect the temperatures at each step along the steam generator. For this method to work, the IAPWS95 package was needed to be able to find the density and Prandtl number at each temperature step. This was done by giving the IAPWS95 package the temperature and quality calculated at the current step. The density is then used to determine the mass of water in the steam generator by multiplying the volume of the steam generator by the current value of density. Rayleigh's number (Ra) is then determined using the equation

$$Ra = \frac{g\beta(T_s - T_\infty)L^3}{\nu\alpha} \quad , \tag{11}$$

where g is gravity, β is the expansion coefficient, ν is the dynamic viscosity, and α is the

thermal diffusivity [11]. The Nusselt number is then found for the current temperature step using the formula

$$\mathrm{Nu} = \left\{ 0.825 + \frac{0.387 Ra_L^{1/4}}{[1 + (0.492/Pr)^{9/16}]^{4/9}} \right\}^2 \quad . \tag{12}$$

The heat transfer coefficient is then calculated through Eq. 13

$$h = \frac{Nu\ k}{L} \quad . \tag{13}$$

Next, the temperature at the next step up the steam generator can be calculated the finite difference method with Eq. 14

$$T_{i+1} = T_i + dl(\frac{q''}{h}) \quad , \tag{14}$$

where the $\frac{q''}{h}$ term was found from Newton's Law of Cooling (Eq. 15).

$$q'' = h\ (T_s - T_\infty) \quad . \tag{15}$$

After finding the temperature at the top of the steam generator at each defined power, the computer model uses the IAPWS95 package again by giving it the temperature at the top of the steam generator and assuming an exit quality of one to find the pressure at the exit. The IAPWS95 package then finds the enthalpy at the outlet of the steam generator. Since the inlet conditions of T = 489 K and x = 0 are assumed to be constant, the inlet enthalpy would also stay the same for all Q values at 924.43 kJ/kg. Finally, The mass flow rate can then be determined using Eq. 6.

The first assumption that was made was that since the IAPWS95 package was unable to calculate the specific heat, thermal conductivity, thermal diffusivity or dynamic viscosity for qualities that were not 0 or 1, the value at the output is averaged with the value input. Those constants are then used in Eq. 11 and Eq. 12 to find the Rayleigh number and Nusselt number at each temperature step along the steam generator.

The second assumption that was made involved the expansion coefficient β. Normally β is found using the equation

$$\beta = -\frac{1}{\rho}(\frac{\partial p}{\partial \rho}) \quad .$$

For an ideal gas, $\rho = p/RT$ which causes β to become 1/T where T is the average absolute temperature in Kelvin through the steam generator [11]. However, steam is not an ideal gas and therefore using this simplification introduces more error into the calculations. This assumption had to be made in order to solve for Rayleigh's number in Eq. 11 because there was no information listed for these temperatures in Appendix A of Fundamentals of Heat and Mass Transfer [11].

The next assumption involved with this model is that the quality of the the fluid increases linearly through the steam generator from zero at the bottom to one at the top. This assumption was necessary because the IAPWS95 package needed the quality and temperature at the current time step to be able to provide the density and Prandtl number in order to be able to solve for the heat transfer coefficient and eventually the next temperature step.

Another assumption with this model is that the diameter of the middle section of the steam generator is 2 meters. As shown in Fig. 8, the outer diameter of the steam generator is 4.2672 m and the inner diameter is 4.191 m, but the diameter of the section in the middle that the water flows through, turning into steam, is not given. Therefore, for the purposes of this model, a diameter of 2 meters was chosen based on an estimation from Fig. 8.

In addition, Eq. 12 uses the Nusselt number correlation for a uniformly heated vertical plate. Since the steam generator is actually shaped closed to a cylinder as shown in Fig. 8, this assumption will introduce even more error to this model. The assumption was made in order to make the model easier to solve.

An important difference to note between this model and the original analysis of the system is that this model reevaluates the pressure at the top of the steam generator with the calculated

temperature and a quality of one rather than assuming this is an isobaric process and keeping the pressure the same as it was at the inlet to the steam generator. This could have a large effect of the accuracy of the model, especially for high values of power because the pressure at the inlet to the steam generator with a temperature of 489 K and a quality of 0 is 2.141 MPa while the pressure at the outlet for a decay heat power of 50 MW and a quality of 1 is 3.476 MPa. Using Eq. 16 below,

$$\% \text{ error} = \frac{\text{actual value - estimated value}}{\text{actual value}} * 100\% \quad , \tag{16}$$

this leads to a percent error of 38.4% for when the decay heat power is at 50 MW. Once the decay heat power decreases to about 1 MW, the pressure is only 2.164 MPa, making that isobaric heat addition process much more reasonable of an assumption, but by that time RVACS will be able to handle the decay heat by itself.

Using the Temperature Incremental method, the mass flow rate was found for decay heat power levels of 50 MW, 40 MW, 30 MW, 20 MW, 10 MW, 5 MW, 3 MW, and 1 MW. The resulting mass flow rates are plotted below in Fig. 11.

In Fig. 11, the mass flow rate decreases as the decay heat power decreases. This is what was expected because adding heat is adding energy to the system so when less energy is being added to the system, it is not willing to move as quickly. Fig. 11 also shows the estimated mass flow using the TIM was similar to the scoping calculations done when the station blackout first occurs, but as the decay power decreases with time, the mass flow rate continues to become farther and farther away from the scoping analysis mass flow rate calculation.

The error in this model could be estimated by extrapolating this line out to a power of 840 MW. This would give a mass flow rate of around 450 kg/s which is pretty far off when compared to the known value of the mass flow rate during normal operation of 759 kg/s. Other than the error introduced by the assumptions already stated, the difference in

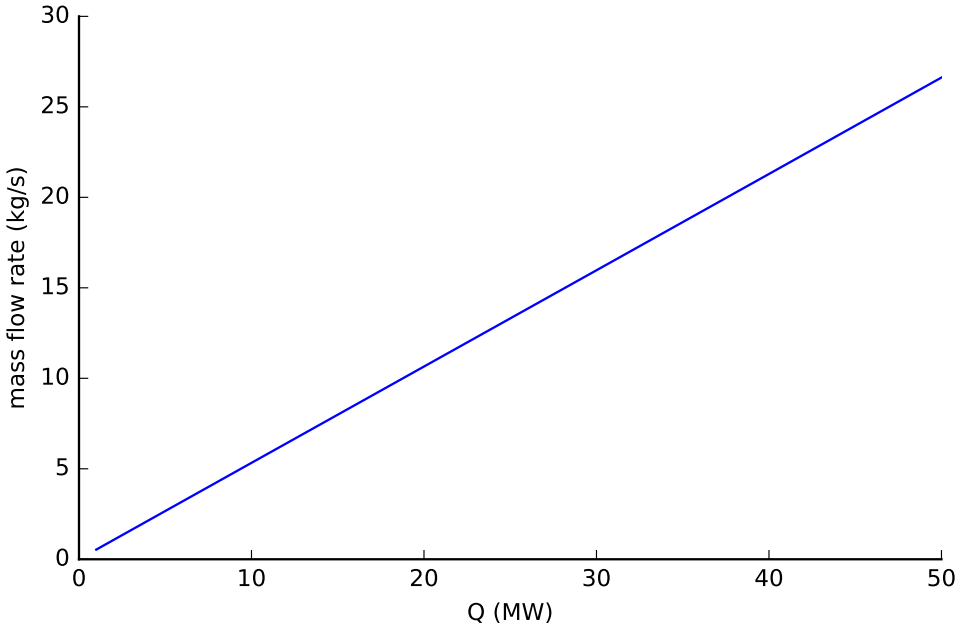

Fig. 11. Mass Flow Rate vs Decay Heat Power using the Temperature Incremental Method.

the estimated mass flow rate can be attributed to the fact that this model only works for qualities of zero to one while in reality, during normal operation, there is sub-saturated water entering the steam generator and superheated steam leaving the steam generator which will affect the mass flow rate of the steam. Also, as mentioned before, two–phase flow is an extremely complicated thing to model and this one is unable to account for most two–phase phenomena such as the vapor and the liquid to be traveling at two different velocities.

3.2 Optimization of Turbine-Driven Blower

3.2.1 Turbine-driven Fan Systems

Terry Turbine In developing a way to passively remove excess heat from the PRISM steam generator, several options were considered. In an attempt to leave the existing system as unaltered as possible, the concept of introducing a Terry turbine to the tertiary water loop was proposed. There were several reasons that this was considered. For one, Terry

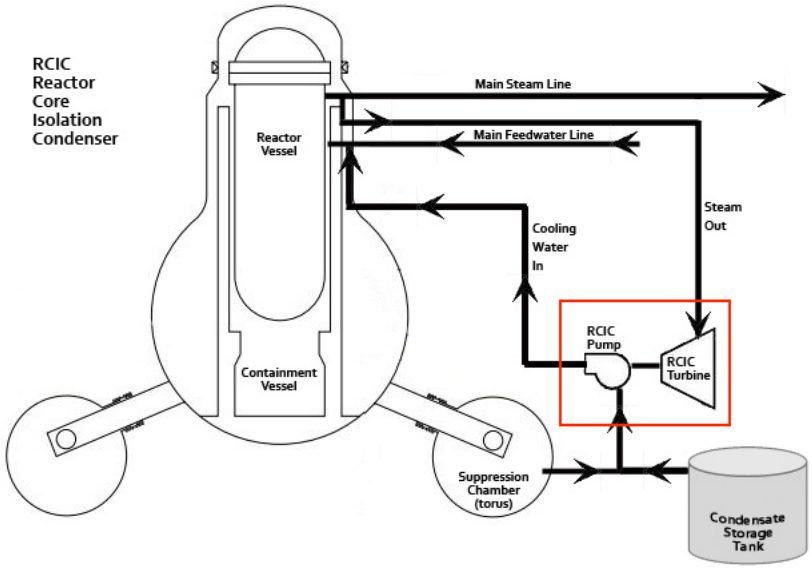

Fig. 12. Diagram of a typical RCIC system. The red box highlights the turbine-driven pump section of the system.

turbines, formerly designed and produced by the Terry Corporation (eventually acquired by Babcock and Wilcox), are a well-established technology in the nuclear safety field. For example, the Reactor Core Isolation Cooling (RCIC) system implemented in many Boiling Water Reactors (BWRs) is known for its utilization of a Terry turbine to power a pump that supplies cooling water to the reactor core from a condensate storage tank in the event of a Loss of Coolant Accident (LOCA) accident (see Fig. 12). The turbines are designed to typically reach capacity within two minutes of startup and are controlled by battery-powered direct current alone [12]. This battery power primarily serves the governor valve, which regulates turbine speeds to prevent overspinning. It should be noted though, that these governor valves have a distinct history of operational failure [13].

Terry turbines are attractive in accident scenarios due to their robust and dynamic nature. Due to their fairly simple mechanical design, Terry turbines have the ability to handle two-phase flow for a short period of time without failing, whereas most steam turbines experience severe damage upon impact with high-speed water droplets [14].

Based on these qualities, it was hypothesized that a Terry turbine could perform a similar

operation in the PRISM steam generator loop, with the key difference being that the Terry turbine would be supplying mechanical shaft work to a blower to induce forced air circulation around the steam generator as opposed to powering a pump to deliver cooling water to the reactor core. In an effort to avoid confusion, it should also be emphasized that there is no assumed breach of the water loop as a result of the accident which caused the loss of on-site power. There are also various assumptions associated with the use of a Terry turbine. These include use of the previously mentioned battery-operated governor valve (the team recognizes concerns as to the degree of passivity associated with reliance on battery-operated systems). Another key assumption relies on a steady flow of steam through the turbine (meaning that they system must rely on buoyancy-driven flow).

Fan Curves In order to couple the output data from the Terry turbine to the induced flow conditions required for the Computational Fluid Dynamics (CFD) simulation in the steam generator, a fan curve for the turbine-driven air flow needed to be generated. Fan curves describe the relationship between the volumetric flow rate of air that a fan is able to supply and the static pressure drop across the fan. Fan curves are specific to the individual fan being used and the shape of the curve varies with the specific type of fan being used [15]. Fan performance data is typically provided with the purchase of a new fan. Unfortunately, for the scope of this project, obtaining real industrial fan performance data proved unfeasible. However, simple fans are governed by affinity laws. Affinity laws describe geometric relationships between fan shaft velocity, pressure drop, power, and air flow rate.

3.2.2 Fan Curve and Steam Extraction Methods

Affinity Laws Referencing fan performance ratings from previous turbine-driven blower experiments [16], fan curve parameters can be tailored to generate theoretical performance curves. The fan specifications used in the following analysis were taken from a naval turbine-driven blower study to provide airflow to boilers (see Fig. 13). This turbine-fan combination

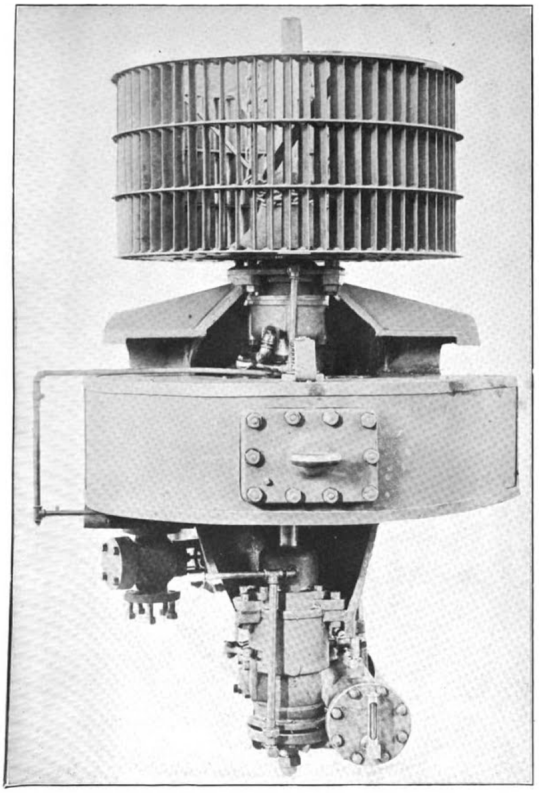

Fig. 13. Turbine-driven fan used to provide forced convection to boilers. [16]

was primarily selected based on the similarity of the two applications and system components. Fan specifications included a rated capacity of 23000 cubic feet of free air per minute against 5 inches of static pressure water gage and a speed of 14000 r.p.m. The diameter of the fan was 33 inches.

The affinity laws are described by, following the equations:

$$Q_2 = Q_1 \left(\frac{N_2}{N_1} \right) \ , \tag{17}$$

where the respective states of Q and N correspond to the volumetric flow rate of air induced by the fan and rotational velocity of the fan.

$$P_2 = P_1 \left(\frac{N_2}{N_1} \right)^2 \ , \tag{18}$$

where the states of P represent the static pressure. And finally,

$$\mathscr{P}_2 = \mathscr{P}_1 \left(\frac{N_2}{N_1} \right)^3 \quad , \tag{19}$$

where the states of $\mathscr{P}$ represent the fan power.

The power rating was determined by the following correlation for turbine-driven blowers [17],

$$\mathscr{P}_{\max} = \frac{Q_{\max} \Delta P_{\max}}{33000 \ \eta} \quad , \tag{20}$$

where the efficiency (η) is taken to be 0.6 (a conservative choice for this type of system), the maximum static pressure rating ($\Delta P_{\max}$) is 5 in. of water, and $Q_{\max}$ from Eq. 17 is 23000 cfm. Given these conditions, $\mathscr{P}_{\max}$ was determined to be 30.20 HP = 22.52 kW. A Python module was then developed to vary the blower rotational velocity and generate the data for the fan curves.

Terry Turbine Steam Extraction With the fan curve being generated, it was necessary to ensure that there would be enough steam of acceptable quality to power the fan. In order to provide a conservative assurance, the fan was assumed to be operating at maximum power for the full duration of operation. With maximum power being previously defined as 22.52 kW, the mass flow rate required to power the fan can be defined as

$$\dot{m}_{\mathrm{t}} = \frac{\mathscr{P}_{\max}}{(h_{\mathrm{t,in}} - h_{\mathrm{t,out}})\eta_{\mathrm{t}}} \quad . \tag{21}$$

The turbine inlet enthalpy ($h_{\mathrm{t,in}}$) can be taken to be equivalent to the steam generator outlet enthalpy as a function of time, which was calculated in the previous thermal loop analysis with Python. The turbine outlet enthalpy is assumed to be that of saturated steam at atmospheric pressure (since the steam from the open system will be vented to the

atmosphere). As one of the trade-offs associated with having such a mechanically robust system, the efficiencies of Terry turbines (η_t) are typically assumed to be about 5%. This efficiency is low as far as turbines go, where typical well-designed steam turbine efficiencies are $> 90\%$ [15].

The previous steam generator analysis written in Python allowed the outlet steam properties to be calculated as a function of power being supplied to the reactor. With the power decay curve supplied as a function of time, the mass flow rate coming out of the steam generator can also be calculated as a function of time. Then, with the mass flow rate of steam required by the turbine as the function of time calculated, the proportion of steam to be extracted from the main steam line as a function of time can then be determined to ensure that the system will run for the required length of time.

3.2.3 Blower and Turbine Performance

Fan Curve By setting the previously mentioned fan specifications as the initial states of the affinity law equations, and dividing the volumetric flow rate by the fan area ($A = \frac{(33\ \text{in})\pi}{4}$), the Fig. 14 was generated.

Figure 14 shows the trade-off relationship between the air velocity and the static pressure drop induced by the blower with the blue curve. As shown previously, both parameters are dependent on the rotational speed of the blower blades. It can be seen then that the fan curve conforms to a smooth decreasing quadratic relationship. This shape is typical of centrifugal blower (the type of blower assumed in this analysis), whereas other types of blowers (axial flow) exhibit more irregularities [18].

The power that the fan requires to induce a given air velocity is also shown on Fig. 14 as the red curve. As predicted, the power requirements respond with a cubic relational increase in response to the rotational speed of the rotor and corresponding air speed. It can also be seen that the maximum power rating 22.52 kW corresponds to an induced air velocity just

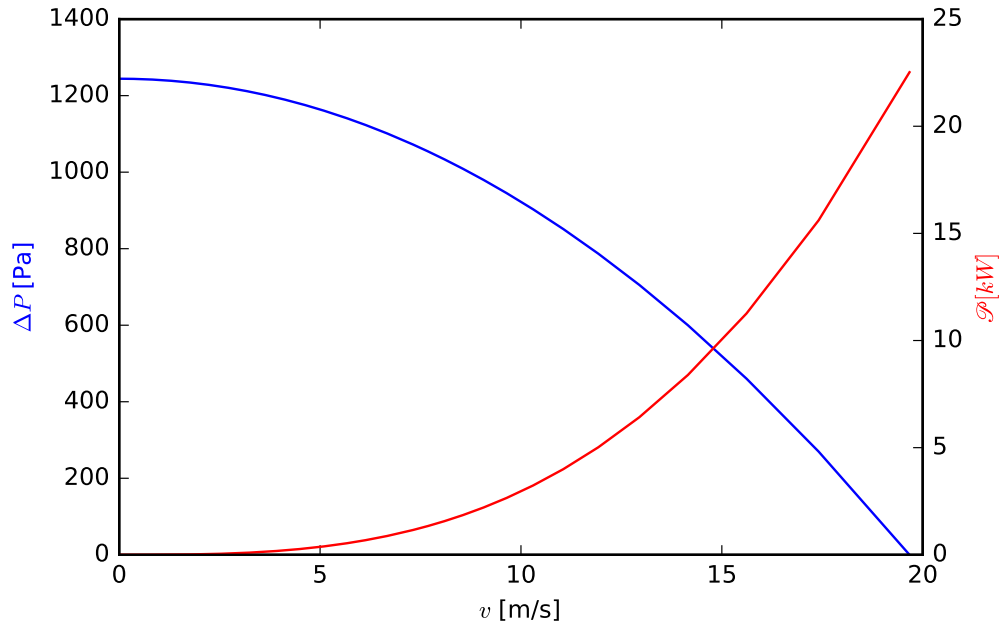

Fig. 14. Fan curve generated with the affinity laws with velocity (v) on the x-axis, static pressure drop (ΔP) on the left-hand y-axis, and power ($\mathscr{P}$) on the right-hand y-axis.

shy of $20\,\mathrm{m\,s^{-1}}$. This fan curve data was printed to an output file with Python so that it could be used as an input for the OpenFOAM CFD simulation.

Mainline Steam Extraction Figure 15 shows the transient nature of the mass flow rate out of the steam turbine with the blue line. It can be seen that the mass flow rate drops dramatically within the a few seconds of reactor shutdown and continues to decrease with increasing time. This was expected, as the loss of power to the pumps would force the water in the system to move by momentum and natural circulation. With the mass flow rate and inlet enthalpy to the turbine now provided as a function of time, Eq. 21 can be used to determine the required mass flow rate to have the turbine provide the appropriate shaft work to power the blower. The ratio of the required mass flow rate of the turbine to the mass flow rate exiting the steam generator can be calculated. This ratio is plotted as function of time in Fig. 15 as the red line.

One important parameter to be aware of from the % of the steam that is extracted from

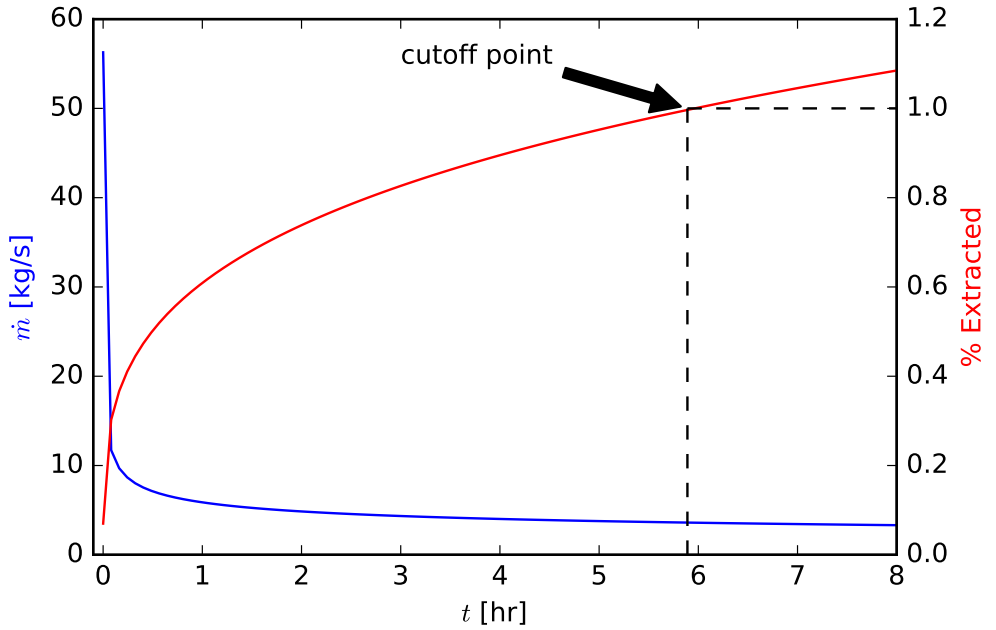

Fig. 15. Plot showing the transient nature of the mass flow rate in the Terry turbine system using the ANS 2005 decay heat curve as an input. Left-hand axis shows the mass flow rate as a function of time after reactor shutdown. Right-hand axis shows the percentage of steam that will need to be extracted from the main steam line so that the Terry turbine can supply sufficient power to the fan.

the main steam line is when the turbine requires 100% of the steam to be extracted. At this point, it is no longer possible to supply the turbine with a high enough quality steam source to power the blower at its power rating. The blower performance fan curve can no longer be used at this point, meaning that the CFD calculation can no longer be trusted. This cutoff point is seen in Fig. 15 at approximately 5.89 hours. This is acceptable, as it was estimated in the scoping analysis that the system would need to run for just under 3 hours in order to be feasible. It can then be said with some degree of confidence that the steam provided by the steam generator after the pumps have been turned off is sufficient to run power the turbine-driven blower. It should be noted that the turbine would technically not be "cutoff" at this point, but that it would no longer be able to supply the necessary work to power the fan at maximum power. This would effectively change the generated fan curve and introduce an extra layer of transient nature to the problem. For this reason,

nothing beyond the "cutoff" point will be considered. This also makes our calculation more conservative.

Should it be decided that a larger fan be used, the mass flow rate flowing through the Terry turbine would need to be fixed at 100% extraction at 3 hours of operation. By interpolating the mass flow rate and turbine inlet enthalpy values with respect to time, a function can be generated to specify the parameters of interest. After 3 hours of operation at 100% mainline steam extraction, the the power able to be supplied to the Terry turbine is approximately 27 kW (using the decay heat data from the ANS 2005 decay heat curve).

Using the 2014 PRISM decay curve proved to increase the total amount of heat needed to be removed from the system and in turn change the steam properties to allow for a longer Terry turbine optimization time. Figure 16 shows a dramatic increase in the length of time that the system is able to run. Whereas the the previous decay heat curve showed the system operating for roughly 6 hours, the updated decay curve estimates an operating time of approximately 17.5 hours before 100% of the steam needs to be extracted from the main steam line.

There are essentially two contributing factors for determining the amount of time that the Terry turbine system is able to run. The first factor corresponds to the percentage of mainline steam extracted to the Terry turbine. The second factor dictating how long the system runs relates to how much water (mass) is in the system and how long that water will last based on the mass system's mass consumption rate. By numerically integrating the exit mass flow rate of the steam generator over the range of expected operating time in Python, the total mass of water exiting the steam generator was calculated to be 303 049 kg of water.

Based on the average mass flow rate (22 kg) leaving the condenser in the previous steam generator analysis and the amount of water estimated to be residing in the condenser at the loss of power (2.2×10^6 kg for a 1000 MW reactor), the amount of time the system was expected to last based only on the amount of water in the system was estimated to be 27.7

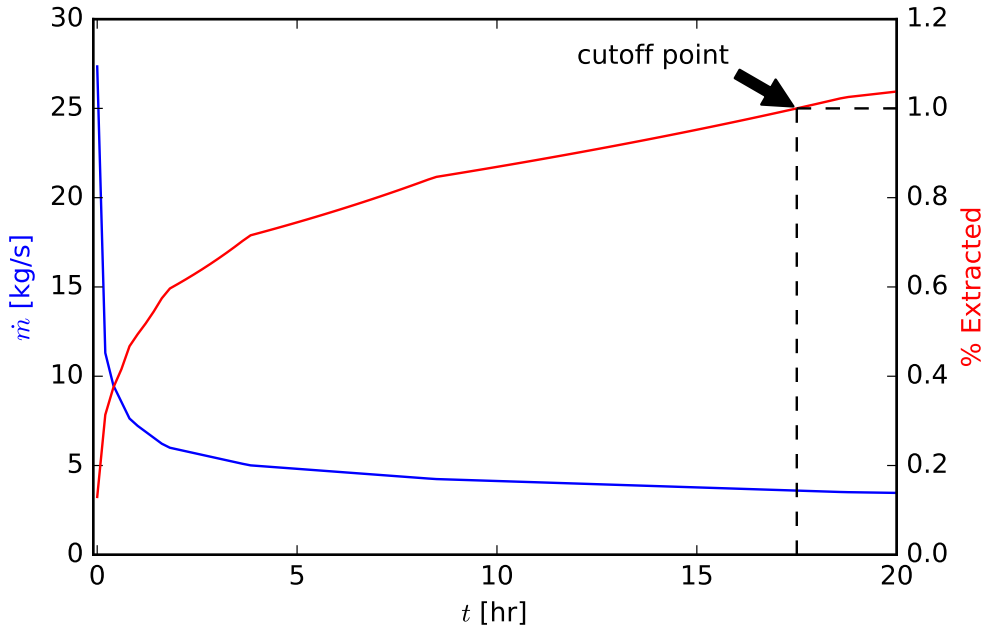

Fig. 16. Plot showing the transient nature of the mass flow rate in the Terry turbine system using the 2014 PRISM decay curve as an input. Left-hand axis shows the mass flow rate as a function of time after reactor shutdown. Right-hand axis shows the percentage of steam that will need to be extracted from the main steam line so that the Terry turbine can supply sufficient power to the fan.

hours. When linearly scaling the condenser from 1000 MW down to 840 MW, the time that the system will last decreases to 23.3 hours. However, this estimated time is still larger than the 17.5 hour limiting factor calculated above. Thus, it is expected that the Terry turbine system will be limited to running for an estimated time of 17.5 hours.

3.3 Determination of Heat Removal from the Steam Generator

To determine the heat removal capacity of the Terry Turbine powered blower, the flow path between the outer surface and the surrounding shroud were modeled in OpenFOAM, an open source CFD code [19]. A computational approach was taken to provide a higher fidelity analysis of the blower's heat removal ability than hand calculations. In addition, a CFD analysis of the steam generator was of interest to GEH.

OpenFOAM was chosen due to its availability on the Texas A&M Nuclear Engineering cluster and High Performance Research Computing (HPRC) supercomputers and the team's prior experience with the software. In addition, OpenFOAM is free and the source code is available for viewing and editing. This allows for the addition of custom solvers and bug fixes that would not be possible in the commercial CFD packages. OpenFOAM's open source license also allowed unconditional use on our personal computers. While the commercial packages are more developed and have more features, OpenFOAM was sufficient for the scope of this analysis.

Due to the growing interest in verification, validation, and uncertainty in the CFD community, this analysis includes a brief foray into verifying the model against the Dittus Boelter heat transfer correlation and the use of the Grid Convergence Index (GCI) to assess the numerical uncertainty of the simulation. An in depth validation study could not be conducted without experimental results. This also precludes discussion of the modeling error associated with the simulation.

The approach taken in this section is: establish the validity of the OpenFOAM computational setup (turbulence model, thermophysical properties, boundary conditions, etc.) by comparing to the Dittus Boelter correlation in a geometrically simplified case, extend the simplified model to the steam generator geometry with blower boundary condition, optimize the distance between the steam generator and shroud, use the optimized shroud gap to determine the heat removal rate from the steam generator surface, and determine the numerical uncertainty of that result using GCI.

3.3.1 Simplified Internal Flow Model

To simplify the geometry and allow for direct comparison to the Dittus Boelter correlation, the initial model considered internal flow of air in a smooth pipe at a constant surface temperature. To reduce the computational expense of the simulation and take advantage of

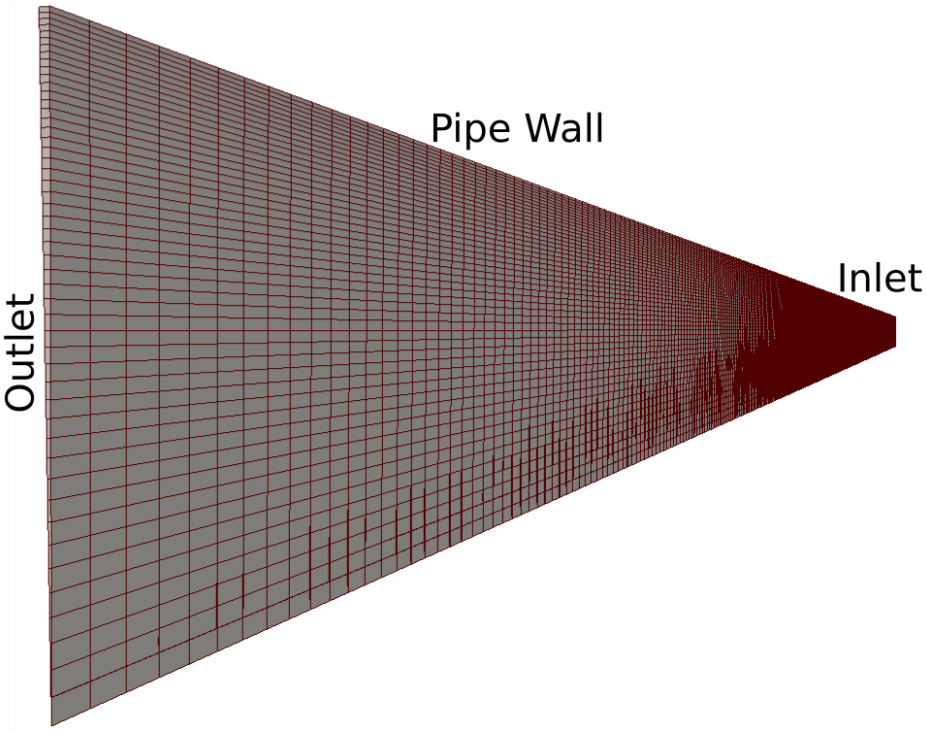

Fig. 17. A rendering of the mesh used for the 2D, internal flow model.

symmetry, the pipe was treated as an axisymmetric wedge. Figure 17 shows the mesh and geometry used in this model. The radius of the pipe was 0.4 m with a total length of 20 m to allow for the flow to develop hydrodynamically and thermally. The mesh was generated with OpenFOAM's *blockMesh* utility. The *blockMeshDict* was auto-populated with GNU m4 [20].

The k-ϵ Reynolds Averaged Navier Stokes (RANS) turbulence model was used. The OpenFOAM k-ϵ model, *kEpsilon*, was chosen for its robustness and reduced meshing requirements due to the use of wall functions. *kEpsilon* is valid for $y^+ \in [30, 100]$ where y^+ is the dimensionless measure of how well the boundary layer velocity gradient is captured. This meant the mesh near the wall had to be carefully chosen and simulations where $y^+ \notin [30, 100]$ had to be thrown out. This made automation of the simulations difficult as any changes in model geometry or parameters could skew the resulting y^+ out of the acceptable range. For example, investigating the change in heat transfer over a variety of inlet velocities could not be automated as each simulation needed to be run multiple times with carefully chosen

meshes to ensure that the boundary layer was correctly resolved.

The model simulated air entering at a uniform velocity and exiting with a zero gradient condition. The pressure was set to atmospheric pressure at the outlet of the pipe with a zero gradient condition at the inlet. The wall of the cylinder was set to a constant temperature of 500 K and the air entered at 300 K. A zero gradient condition was applied for the outlet temperature. The turbulent kinetic energy, k, was given an inlet value of

$$k = \frac{3}{2}(UI)^2\,, \tag{22}$$

where U is the inlet velocity and I the turbulent intensity. I was set to 5% for all simulations. ϵ was set according to

$$\epsilon = 0.09\frac{k^{3/2}}{.07D_h}\,, \tag{23}$$

where D_h is the hydraulic diameter of the pipe. OpenFOAM's turbulent wall functions were used for both k and ϵ on the pipes' wall. Lastly, the *wedge* boundary condition was used for all variables to capture the symmetry of the axisymmetric wedge.

The *buoyantSimpleFoam* solver was used. *buoyantSimpleFoam* is a steady state solver for buoyant flow of compressible fluids using the Semi Implicit Method for Pressure Linked Equations (SIMPLE) algorithm. *buoyantSimpleFoam* iteratively solves for the pressure, velocity, enthalpy, and density of the air. This solver was chosen for its ability to model heat transfer and fluid flow simultaneously. This means that *buoyantSimpleFoam* solves conservation of mass, momentum, and energy preventing the need to use a scalar transport solver to model heat transfer. This approach is also more accurate than using separate fluid and scalar transport solvers and allows the density of the air to change as it's temperature increases.

The buoyant and compressible treatment necessitated that use of an Equation of State (EOS). To enable direct comparison to the Dittus Boelter correlation, the density, ρ, was

held constant at

$$\rho = \frac{p_{\text{ref}}}{RT_{\text{inlet}}}\,, \tag{24}$$

where R is the gas constant for air, T_{inlet} the inlet temperature of the air, and p_{ref} a reference pressure taken to be atmospheric pressure. The viscosity, Prandtl number, and specific heat of air were treated as constant functions of temperature evaluated at the film temperature.

In addition to *buoyantSimpleFoam*, *potentialFoam* was used to initialize the internal velocity field. *potentialFoam* solves the continuity equation and simplified pressure equations to approximate the velocity field. The use of this preconditioning solver helped *buoyantSimpleFoam* converge.

The total heat transfer and heat flux on the walls of the pipe was determined using the OpenFOAM utility *wallHeatFlux*. This was coupled to a custom Python post processing tool which used the output from the *wallHeatFlux* utility to compute the heat transfer coefficient, h, with

$$h = \frac{q''}{T_s - T_\infty}\,, \tag{25}$$

where q'' is the heat flux computed with the *wallHeatFlux* utility, T_s the surface temperature of the pipe (set to 500 K), and T_∞ the free stream temperature. The free stream temperature was assumed to be the centerline temperature. The data extraction process was automated with the *singleGraph* utility which was used to produce a separate file of centerline temperatures and axial distances.

The simulated heat transfer coefficient was compared to the Dittus Boelter correlation:

$$Nu = .023Re^{4/5}Pr^{0.4}\,, \tag{26}$$

where Re and Pr are the Reynolds and Prandtl numbers [11]. The Nusselt Number, Nu, is

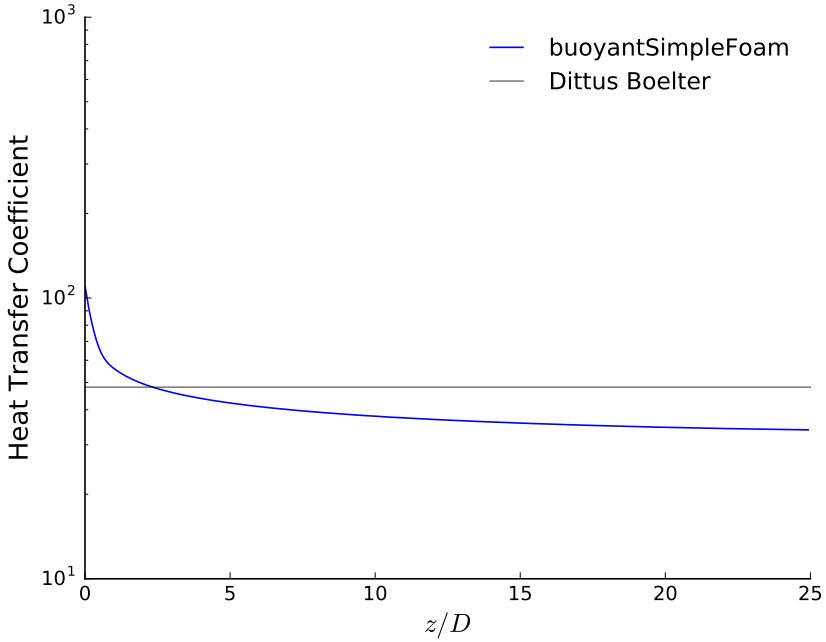

Fig. 18. A plot of the simulated heat transfer coefficient as a function of dimensionless axial distance. The Dittus Boelter value is provided in black.

then used to to find the heat transfer coefficient through

$$h = \frac{kNu}{D_h}\,, \tag{27}$$

where k is the thermal conductivity of air.

The air inlet velocity was set to 20 m/s. This produced a turbulent Reynolds number of 7.12×10^5 that was within the Dittus Boelter correlation's acceptable range. The simulated heat transfer coefficient as a function of number of diameters downstream is provided in Fig. 18. *bouyantSimpleFoam* and the Dittus Boelter correlation differed by 29%. Dittus Boelter is accurate to approximately 25% [11]. This suggests the simulation is just out of range. Qualitatively, the simulation performed well however, the 29% difference prompted investigation.

To verify the 2D, axisymmetric model was a valid assumption, a quarter model was con-

structed. In this case, a fourth of the pipe was modeled in 3D. This model performed similarly to the axisymmetric model. This result supported the use of the axisymmetric model.

The k-ω RANS turbulence model, *kOmegaSST*, was also tested. Use of this model did not greatly improve the accuracy of the model. However, *kOmegaSST* did increase computational expense because the model requires meshing to the wall ($y^+ < 1$).

Modeling in 3D and using *kOmegaSST* both increased the meshing requirements of the simulation without affecting the accuracy of the results. To save computational time, this computational setup was used in later simulations.

3.3.2 Steam Generator Model

The flow path between the steam generator and surrounding shroud was treated as the space between two concentric cylinders (see Fig. 19). As in the simplified model, the rotational symmetry of the problem was used to reduce the problem to two dimensions. The resulting mesh is shown in Fig. 20. The bottom surface in Fig. 20 is the steam generator's outer shell. This will be assumed to be at a constant temperature of 773 K. The top surface of Fig. 20 is the insulated shroud surrounding the steam generator. The left and right sides are reflected using the *wedge* boundary condition. The surface on the front is the outlet and the surface toward the back is the inlet. Changes in boundary conditions from the simplified model are described below.

The *fanPressure* boundary condition was applied at the outlet. This condition applies a pressure jump to simulate the change in pressure induced by a fan described by a user supplied fan curve. This prevented the need to explicitly model the blower in the simulations.

In this application, the *fanPressure* boundary condition applies a pressure drop at the outlet of the shroud gap. Pressure was specified as atmospheric at the inlet of the gap. The blower-lowered pressure at the outlet caused air to be pulled through the gap from inlet to

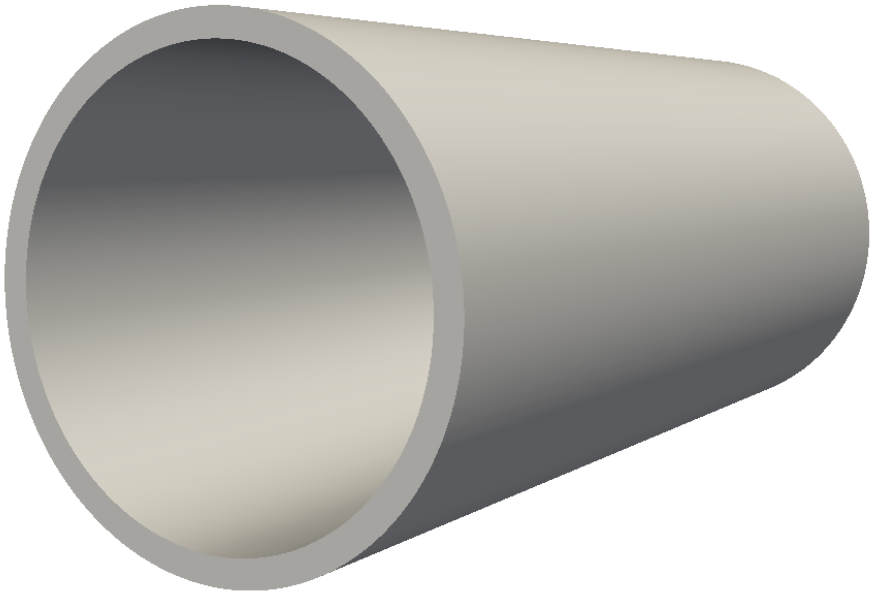

Fig. 19. A reflected rendering of the steam generator geometry. The flow path is shown in gray.

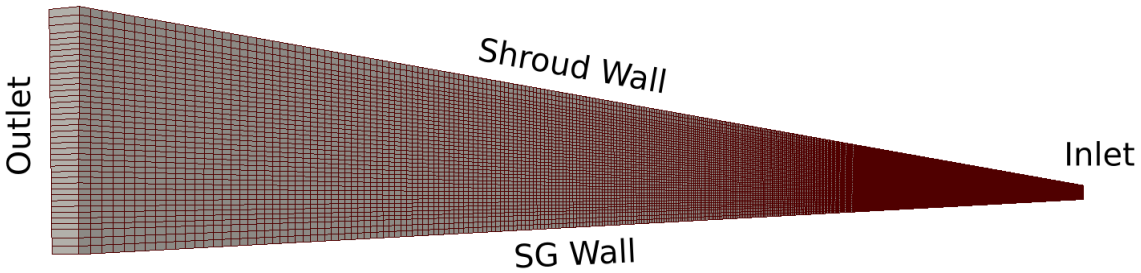

Fig. 20. A rendering of the steam generator mesh.

outlet.

The boundary conditions for velocity were zero velocity on the steam generator and shroud walls and *inletOutlet* and *pressureInletOutletVelocity* at the outlet and inlet. The *pressureInletOutletVelocity* condition sets the velocity according to the pressure gradient at the inlet. Combined with the *fanPressure* pressure condition, the *pressureInletOutletVelocity* condition allows the model to simulate air being pulled into the bottom of the steam generator by the blower. The *inletOutlet* boundary at the outlet helps the solver conserve mass. For outflow, *inletOutlet* defaults to a zero gradient condition. For inflow, it allows the specification of a uniform inlet velocity. The inlet velocity was set to zero.

The steam generator wall temperature was set to a constant value of 773 K and the shroud was given a zero gradient condition to simulate a completely insulated wall. The air at the inlet was assumed to be at a containment temperature of 300 K.

For the steam generator model, compressibility was enabled. Air was treated as an incompressible perfect gas with the following EOS:

$$\rho = \frac{p_{\text{ref}}}{RT}\,, \tag{28}$$

where T is now the local temperature of the air. The use of this EOS allows the density to change as the air is heated.

The mesh was coarsened at the walls to keep y^+ within the acceptable range of the *kEpsilon* turbulence model while maintaining resolution in the center of the flow path.

The major assumptions in this model are the constant temperature steam generator wall, the constant inlet air temperature, and the simplified steam generator geometry. The constant temperature assumptions were made to remove the transiency of the problem. The steam generator and containment temperatures are expected to change over the course of the Terry turbine driven blower's operational time frame. However, this transiency is out of the scope

of this analysis.

3.3.3 Optimization of Shroud Gap

The size and shape of the gap between the steam generator's outer shell and the shroud was optimized to facilitate convective heat transfer. Two shapes were tested with a variety of sizes. The first was a constant shroud gap along the length of the steam generator. The second follows the thermal boundary layer of the air.

Constant Gap The steam generator simulation was run five times with varying gap sizes. The total heat removal outputted using the *wallHeatFlux* utility at each of the gap sizes is provided in Fig. 21. The heat removal rate asymptotically approaches a maximum value. This occurs because the shroud is far enough away to not effect the flow. This suggests that if the shroud gap is too small, heat transfer can be limited. To conserve space in the steam generator room, the most optimal value is then the smallest shroud gap that produces the maximum heat removal rate. The most optimal value is then 40 cm for the constant gap design.

The number of volumes used was varied to accommodate the changing domain size. The smallest shroud gap was run with 45,000 volumes and the largest with 90,000. This kept y^+ within the acceptable range.

Boundary Layer Gap The laminar, thermal boundary layer, δ_T, for a flat plate in parallel flow is of the form:

$$\delta_T \propto \sqrt{z}\,, \tag{29}$$

where z is the distance from the start of the plate [11]. The boundary layer design attempts to increase heat transfer by accommodating the increasing boundary layer. A possible shroud

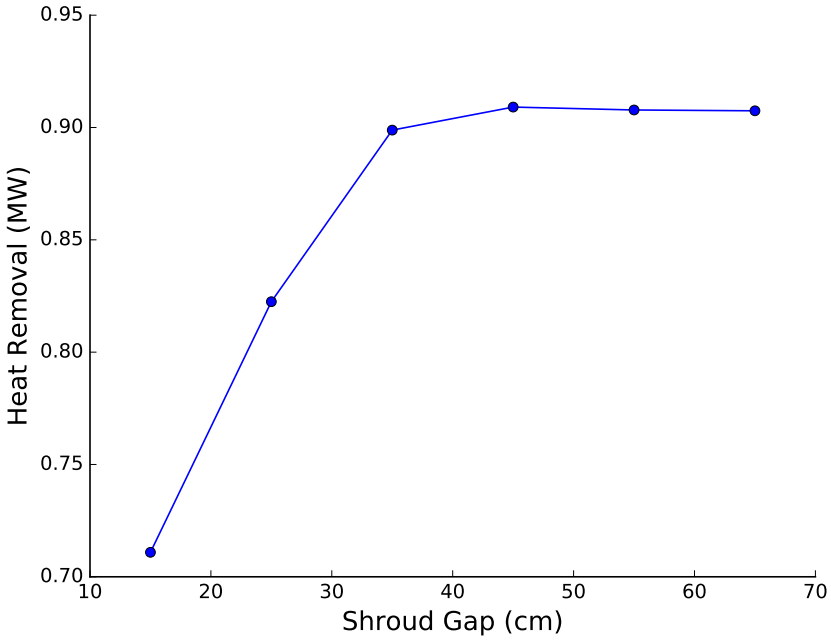

Fig. 21. A plot of the total heat removal versus shroud gap size.

gap, $\delta(z)$, is:

$$\delta(z) = (\delta_2 - \delta_1)\sqrt{\frac{z}{H}} + \delta_1\,, \tag{30}$$

where δ_1 and δ_2 are the gap size at the inlet and outlet and H the height of the steam generator. This was implemented in OpenFOAM but meshing issues caused all simulations of this type to fail. As a work around, a linear gap of the form

$$\delta(z) = \frac{\delta_2 - \delta_1}{H} z + \delta_1 \tag{31}$$

was used instead. A plot of the two gap designs is provided in Fig. 22. A 3D rendering of the linear gap design is provided in Fig. 23.

Initial tests showed that $\delta_2 < \delta_1$ resulted in greatly reduced heat removal. Thus, only $\delta_2 > \delta_1$ was considered in the optimization process to limit the number of required simulations. The difference between the outlet and inlet gap sizes $(\delta_2 - \delta_1)$ was varied for five values of δ_1. The resulting heat removal rates are provided in Fig. 24.

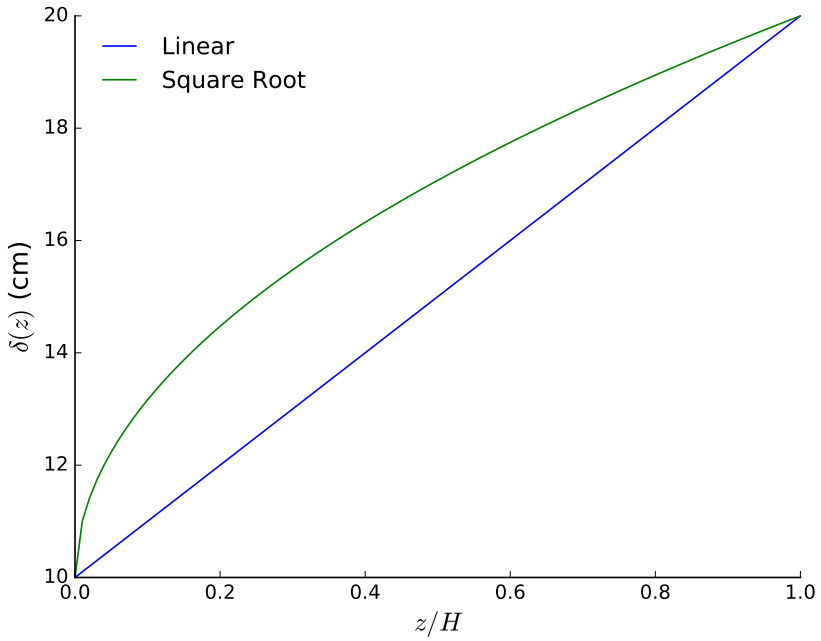

Fig. 22. A plot of the gap width as a function of distance along the steam generator with $\delta_1 = 10\,\text{cm}$ and $\delta_2 = 20\,\text{cm}$ for the linear and square root boundary layer gap designs.

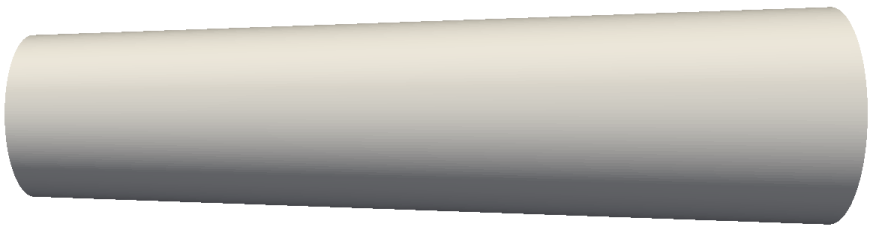

Fig. 23. A 3D rendering of the linear gap design where the shroud gap increases linearly from inlet to outlet.

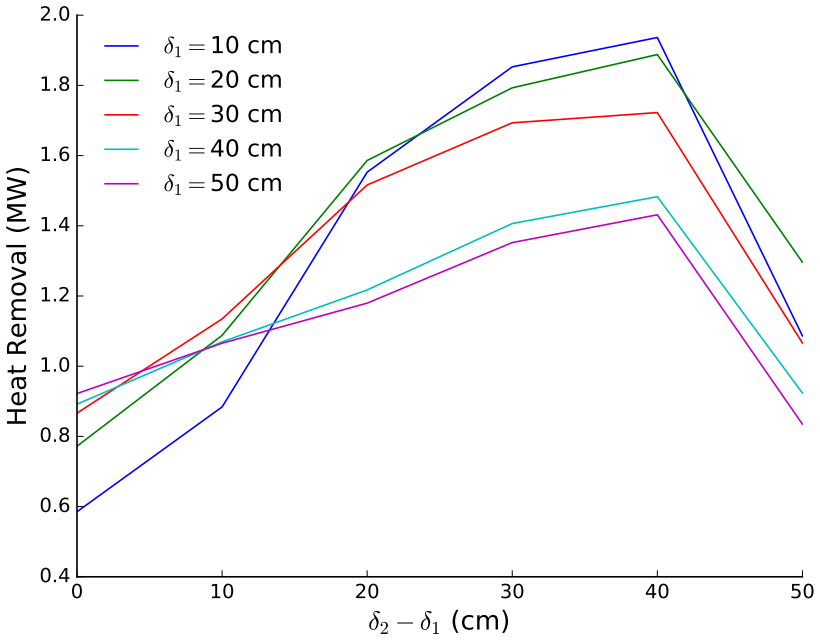

Fig. 24. Plots of the heat removal rate for the linear gap design.

Figure 24 suggests that smaller inlet gaps with a 40 cm larger outlet gap result in the highest heat transfer rates. The optimal design is then the highest heat removal rate which occurs at $\delta_1 = 10\,\text{cm}$ and $\delta_2 - \delta_1 = 40\,\text{cm}$. Compared to the constant gap design, the linearly increasing gap resulted in a 120% increase in heat removal.

Figure 24 required 30 simulations. To increase the time to completion, the simulations were run using eight cores of Texas A&M's HPRC supercomputer, Ada. Due to the changing domain size between simulations, the number of volumes used varied. The smallest domain was run with 45,000 volumes and the largest domain was run with 165,000 volumes. Each simulation took approximately 2.5 CPU hours.

The linearly increasing domain size in each simulation was difficult to mesh with *blockMesh*. The mesh grading was not altered from the constant gap simulations. This meant that the mesh coarsened from inlet to outlet due to the linearly increasing domain size. A better approach would be to linearly increase the number of volumes used along the length of the steam generator or to linearly increase the mesh grading so that the volumes remained

the same distance from the wall. In addition, due to the cylindrical geometry, the steam generator wall was significantly more resolved than the shroud wall as the volumes expanded radially outward from steam generator wall to shroud.

Meshing issues combined with the expense of the simulations resulted in high residual errors in the iterative solver ($\sim 1 \times 10^{-4}$) and suboptimal y^+ values. For small $\delta_2 - \delta_1$, the average y^+ at the steam generator wall was $\sim$ 20. For large $\delta_2 - \delta_1$ it was y^+ was $\sim$ 60. The cylindrical geometry also caused a large discrepancy between the average y^+ at the steam generator wall and at the shroud wall. Thus, these simulations should only be trusted on a qualitative basis as the actual values are subject to numerical and turbulence model error.

3.3.4 Determination of Heat Removal

Using the optimized linear shroud gap design, the heat transfer on the outside of the steam generator was evaluated with the fan curve generated in Section 3.2. The above simulations were all run with OpenFOAM's example fan curve which turned out to be for a much smaller blower than the one chosen in this design. The new fan curve created a much larger pressure drop than the example curve used in the previous simulations. This exacerbated the convergence issues seen in the optimization study.

It was found that convergence issues were caused by the *pressureInletOutletVelocity* boundary condition. As a workaround, the velocity inlet condition was changed to a uniform zero condition. This means that in the simulation, the fan is pulling in stagnant air at the bottom of the steam generator. Simulations with this boundary condition converged quickly even with the new fan curve.

An error within the *bouyantSimpleFoam* source code was also found in the initialization of the pressure variable. The incorrect initialization made the first few iterations prone to crashing. This bug could not be fixed on Ada due to lack of permission to edit source files. However, the fix was added in on our personal computers.

Fig. 25. A rendering of the velocity profile for the first quarter of the flow path.

Fig. 26. A rendering of the temperature profile for the first quarter of the flow path.

The final run used 52 900 volumes. The resulting heat removal rate was 2.98 MW. Figures 25 and 26 show the velocity and temperature profiles in the first quarter of the flow path.

3.3.5 Numerical Uncertainty

The numerical uncertainty was quantified using GCI. The quantity of interest, f, was the total heat removed from the surface of the steam generator. Three simulations using 30 730, 40 000, and 52 900 volumes were run to produce three heat removal rates, f_1, f_2, and f_3. These values are plotted in Fig. 27.

The equation

$$p^{n+1} = \frac{\ln\left|\frac{f_3 - f_2}{f_2 - f_1}\right| + q(p^n)}{\ln r_{12}}\,, \tag{32}$$

where

$$q(p) = \ln\frac{r_{12}^p - s}{r_{23}^p - s}\,, \tag{33a}$$

$$s = \operatorname{sign}\left(\frac{f_3 - f_2}{f_2 - f_1}\right)\,, \tag{33b}$$

$$r_{12} = \sqrt{\frac{N_2}{N_1}},\ r_{23} = \sqrt{\frac{N_3}{N_2}},\ N_1 > N_2 > N_3 \tag{33c}$$

was iterated until

$$\frac{p^{n+1} - p^n}{p^n} < 1 \times 10^{-9}\,. \tag{34}$$

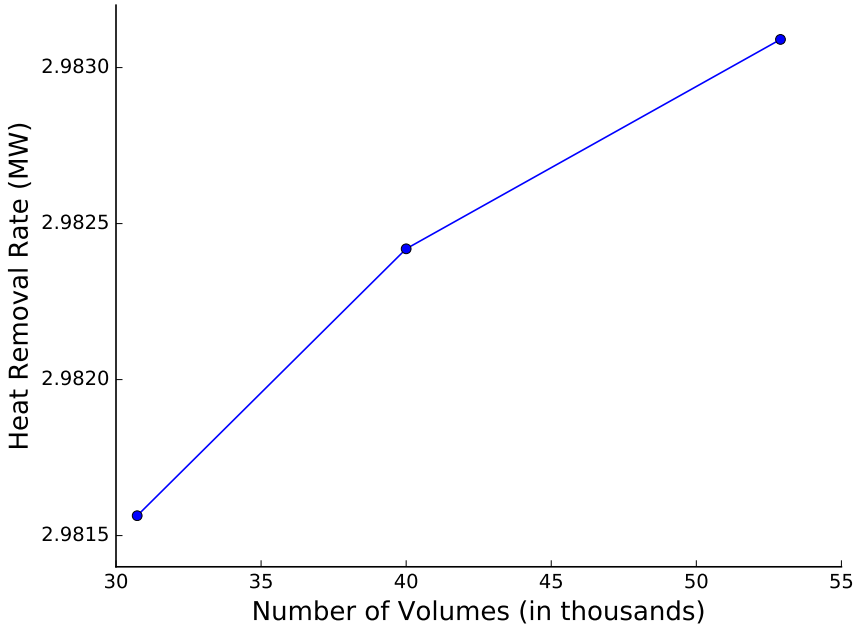

Fig. 27. A plot of the three heat removal rates computed for GCI.

The numerical uncertainty, u_{num}, was then computed as

$$u_{\text{num}} = \frac{F_s}{r_{12}^p - 1} |f_1 - f_2| \,, \tag{35}$$

where

$$F_s = \begin{cases} 1.25 & \frac{p-2}{2} < .1 \\ 3 & \text{otherwise} \end{cases} . \tag{36}$$

This process resulted in an observed order of convergence of 1.177 and a numerical uncertainty of 9.61 kW. This suggests that the mesh is converged. Due to the lack of experimental data to compare to, an assessment of the modeling error can not be performed. Thus, the heat removal rate determined in this analysis is $2.98\,\text{MW} \pm 9.61\,\text{kW}$.

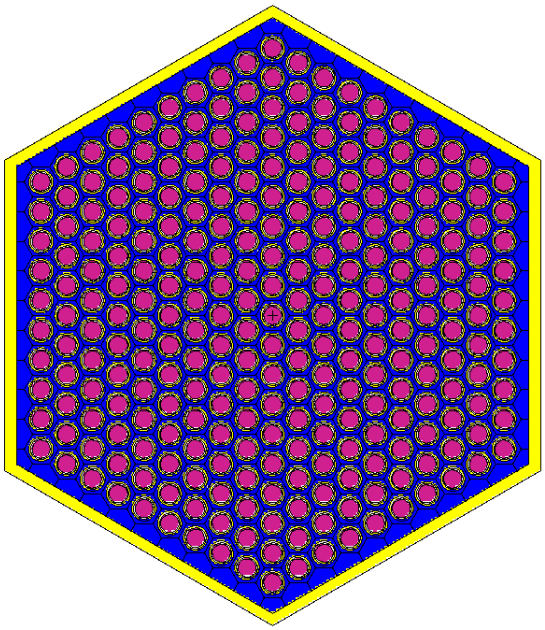

Fig. 28. Cross section of PRISM fuel assembly created in MCNP.
Yellow is the cladding, blue is the coolant, and red is the fuel.

3.4 Modeling PRISM Fuel Assembly to Determine Decay Heat Power

3.4.1 MCNP Model

An MCNP model of a single PRISM fuel assembly was created. The fuel component of this assembly was then burned to determine the resulting fission product generation at various time steps over the course of the 12–month fuel cycle.

Geometry PRISM uses a hexagonal assembly with 271 circular fuel pin [1]. Figure 28 shows a cross section of a single fuel assembly created in MCNP. The necessary dimensions for the assembly and fuel pins are summarized in Table 1 [1, 21].

Using MCNP's hexagonal repeated structure universe, a single fuel pin was reflected to create the 271 fuel pin assembly. To ensure proper reaction rates, fission product yield, and

TABLE 1: Summary of dimensions for PRISM fuel assembly [1, 21].

Dimension	Value (cm)
Active Fuel Height	66.0
Fuel Diameter	0.541 02
Pin Outer Diameter	0.073 66
Clad Thickness	0.055 88
Pin Pitch	0.088 32
Assembly Pitch	15.9563
Assembly Thickness	0.355 60

transuranic build–up in the lattice universe, the total fuel and coolant volumes were specified using MCNP's *MATVOL* card. The volumes were 4111.78 cm^3 and 5753.16 cm^3, respectively.

The six outer planes that make up the sides of the hexagonal assembly were set as reflective boundaries and the top and bottom planes of the fuel assembly were set as vacuum boundaries. This allowed the burn to model a assembly in an infinite medium, giving a conservative estimation of the fission product production and greatly reduced the necessary computation time for the burnup calculation.

Fuel Composition PRISM uses Uranium–Transuranic–Zirconium (U–TRU–Zr) metallic fuel [1]. The primary mission of PRISM is Used Nuclear Fuel (UNF) recycling, which involves burning the plutonium and other long-lived transuranics that are produced during the operation of LWRs. Spent LWR fuel has a variety of transuranics due to neutron capture reactions with both ^{235}U and ^{238}U and their subsequent daughters. These transuranics include any element with a larger atomic number than U, mostly Np, Pu, Am, and Cm. Transuranics are long–lived, contributing to the nuclear waste disposal problem and toxicity of spent fuel. Kang and von Hippel describe the composition of the transuranic mix in Low–Enriched Uranium (LEU) fuel 20–year after discharge with a burnup of 53 MWd/kg_{HM} [22]. The transuranic mix is summarized in Table 2.

The PRISM UNF core configuration involves an inner and outer fuel blanket with different

TABLE 2: Composition of transuranic mix for LEU 20–years after discharge of 53 $\mathrm{MWd/kg_{HM}}$ burnup [22].

Isotope	Half-life (years)	Mass Fraction in TRU
^{237}Np	2.14×10^7	0.066
^{239}Np	2.4 days	1.6×10^{-8}
^{236}Pu	2.9	1.9×10^{-9}
^{238}Pu	87.7	0.024
^{239}Pu	2.4×10^4	0.457
^{240}Pu	6.5×10^3	0.219
^{241}Pu	14.4	0.053
^{242}Pu	3.8×10^5	0.070
^{241}Am	432	0.089
^{242m}Am	141	9.3×10^{-5}
^{243}Am	7400	0.018
^{243}Cm	28.5	5.1×10^{-5}
^{244}Cm	18.1	3.7×10^{-3}
^{245}Cm	8500	3.9×10^{-4}

fuel compositions. The inner fuel zone composition, with higher transuranic and fissile plutonium loads, was chosen for the simulation. The fuel composition is U–22.7%TRU–10%Zr with a fissile plutonium (^{239}Pu and ^{241}Pu) content of 13.5%. This fuel will undergo the most fission, inducing more radioactivity and subsequent decay heat. The density of U-20TRU-10Zr is 15.98 $\mathrm{g/cm^3}$ [2].

The assumption is made that the transuranics available for fuel manufacturing are available in the ratios described in Table 2. Based on the above transuranic mix, the total plutonium profile makes up 82.3% and the non-Pu transuranics make up the other 17.7%. Of the transuranics present in the modeled PRISM fuel, 96% is plutonium and only 4% are the other transuranics. To accommodate this, the transuranic mix was split up into its respective plutonium and non-plutonium nuclides. This composition in renormalized in Table 3.

Table 4 summarizes the fuel composition used for the MCNP model. This fuel maintains the correct fractions for transuranics (22.7%) and fissile plutonium (13.5%) define for the inner fuel zone of the UNF recycling cycle [1].

TABLE 3: Composition of transuranic mix for LEU 20–years after discharge of 53 $\mathrm{MWd/kg_{HM}}$ burnup [22].

Plutonium		Non–Plutonium	
Nuclide	Fraction	Nuclide	Fraction
^{236}Pu	2.31E-09	^{237}Np	0.37239
^{238}Pu	0.02916	^{239}Np	9.03E-08
^{239}Pu	0.55529	^{241}Am	0.50216
^{240}Pu	0.26610	^{242m}Am	0.00052
^{241}Pu	0.06440	^{243}Am	0.10156
^{242}Pu	0.08505	^{243}Cm	0.00029
		^{244}Cm	0.02088
		^{245}Cm	0.00220
Total	1.00	**Total**	1.00

TABLE 4: Fuel composition used within MCNP for U–22.7TRU–10Zr with 13.5% fissile plutonium content.

Isotope	Weight Fraction
^{90}Zr	0.051 45
^{91}Zr	0.011 22
^{92}Zr	0.017 15
^{94}Zr	0.017 38
^{96}Zr	0.002 80
^{235}U[1]	0.004 85
^{238}U[2]	0.668 15
^{237}Np[2]	0.003 41
^{239}Np[2]	8.26×10^{-10}
^{236}Pu[1]	5.03×10^{-10}
^{238}Pu[2]	0.006 35
^{239}Pu[1]	0.120 97
^{240}Pu[2]	0.057 97
^{241}Pu[1]	0.014 03
^{242}Pu[2]	0.018 53
^{241}Am[2]	0.004 59
^{242m}Am[1]	4.80×10^{-6}
^{243}Am[2]	0.000 93
^{243}Cm[1]	2.63×10^{-6}
^{244}Cm[2]	0.000 19
^{245}Cm[1]	2.01×10^{-5}

[1] fissile; [2] fissionable

TABLE 5: HT–9 steel composition nuclides with $\geq 0.1\%$ by weight.

Isotope	Weight Fraction
^{12}C	0.002 00
^{50}Cr	0.005 27
^{52}Cr	0.101 67
^{53}Cr	0.011 53
^{54}Cr	0.002 87
^{54}Fe	0.050 61
^{56}Fe	0.793 73
^{57}Fe	0.018 34
^{58}Fe	0.002 42
^{92}Mo	0.001 48
^{95}Mo	0.001 60
^{96}Mo	0.001 69
^{98}Mo	0.002 46
^{182}W	0.001 34
^{184}W	0.001 55
^{186}W	0.001 44

Cladding and Coolant Composition The fuel cladding and assembly structure within the PRISM system are made of HT-9 steel with a density of 8.1 g/cm^3 [1, 2]. This Fe–12Cr–1Mo–0.5W–0.2C alloy has been extensively tested for Liquid Metal Fast Breeder Reactor (LMFBR) applications and has shown resistance to radiation damage [23]. Table 5 details the HT–9 steel composition used for the MCNP calculation.

PRISM is a sodium–cooled reactor. The coolant is modeled as pure ^{23}Na with a minor impurity of ^{24}Mg. This impurity is used to capture the (n, γ) capture reaction that will take place in the sodium during the burnup calculation in MCNP. With an average temperature of 700 K in the PRISM core, the density of the liquid sodium is 0.852 g/cm^3 [24].

3.4.2 Burnup Calculation

The depletion/burnup calculation couples CINDER with MCNP to preform steady–state flux calculation in MCNP and nuclide depletion calculations in CINDER [25, 26]. The

calculation first runs a steady–state calculation in MCNP to determine the eigenvalue, 63–group fluxes, energy–integrated reaction rates, fission multiplicity, and recoverable energy per fission. Then CINDER takes these values and performs a depletion calculation to determine new isotopic densities and repeats the process for the next time step [25].

CINDER tracks the time–dependent reactions of 3400 isotopes and MCNP specifies three burnup tiers for fission product generation. To characterize the decay heat from the fuel assemblies, the most detailed burnup tier was used in the MCNP calculation. This tier accumulates about 120 various fission products. In addition, MCNP will track the production of various nuclear processes for any material specified in the materials within the burnup calculation. Care was taken to include ^{24}Mg in the sodium coolant, a product of the (n, γ) capture reaction by ^{23}Na, and all of the relevant transuranics.

The fuel assembly burn was calculated at 4.375 MW for 18 cycles. The first month was split up into 7 cycles of 0.3, 0.3, 0.4, 1, 2, 5, 21 days. This allowed the initial fission product transients as the fresh fuel is being burned to be resolved. The next 11 cycles were each for 30 days. This brought the duration of the burn to 1–year, the same time as the UNF recycling fuel cycle.

There are three additional components of the MCNP input that should be mentioned:

1. The most up–to–date ENDF/B–VII.1 continuous–energy neutron cross sections were used for all nuclides for $E \in (0, 20\,\text{MeV}]$ in the MCNP calculation.

2. MCNP's *MPHYS* card is utilized, enabling physics models by MCNP. This is important for nuclide cross sections which are not included in the ENDF data libraries. It should be noted that the modeled cross sections are mostly nuclides present in the HT–9 cladding composition and sodium coolant and some delayed–neutron cross sections nuclides in the fuel.

3. No $S(\alpha, \beta)$ treatment is used because PRISM is fast reactor, so the majority of

neutrons have a high enough energy that the lower energy interaction can be ignored and MCNP does not include any $S(\alpha, \beta)$ libraries that are relevant to the materials present in the PRISM reactor.

Burnup Results Table 6 in Appendix A summarizes the resulting fission products and actinides present in the fuel composition following 6– and 12–month burn durations. The activity of the fuel assembly did not change significantly between the 6–month and 12–month burns. This means the decay heat is relatively independent of the fuel burnup, which is expected [3]. The average discharge burnup of the PRISM UNF recycling driver fuel assembly is 87.51 $\mathrm{MWd/kg_{HM}}$ [1].

Two nuclides stick out in Table 6: ^{239}U with a half–life of 23.45 minutes and ^{239}Np with a half–life of 2.356 days. Together these two isotopes make up over 36% of the activity in the burned fuel. These two nuclides come the from the same production chain, beginning with the neutron capture reaction of ^{238}U producing ^{239}U which β^- decays to ^{239}Np. These nuclides are the most prevalent at the onset of the reactor shutdown and beginning of decay power generation. However, all of the nuclides listed in Table 6 will lead to many other decay products that will greatly contribute to the decay heat generated by the burnt fuel assembly.

The goal of this burnup calculation was to to characterize the PRISM decay heat power using a nuclide decay code, such as ORIGEN, and the fuel composition from Table 6. Ultimately, this became unnecessary when more complete PRISM decay heat power results were found in a previously submitted NRC document. This decay curve gave the information that was need to integrate the neutronics aspect of this problem with the thermodynamic analysis of the tertiary loop. Future work could focus on extending the already determined fuel composition to fully characterize the PRISM decay power using the MCNP burnup results.

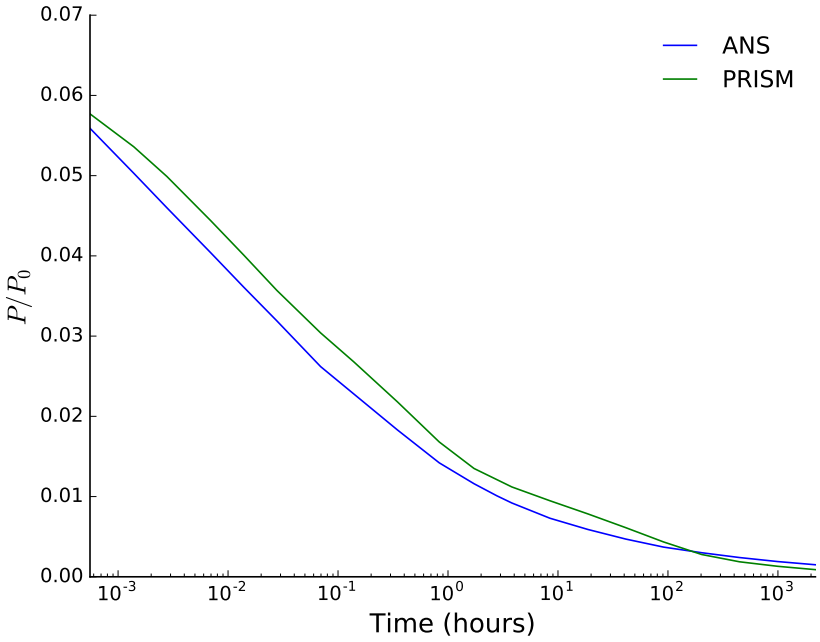

Fig. 29. Comparison of ANSI/ANS-5.1-2005 Decay Heat Power and PRISM Preliminary Safety Information Document [2, 4, 21].

3.4.3 Comparison of ANSI/ANS-5.1-2005 and PRISM Decay Heat Power

The ANSI/ANS-5.1-2005 Decay Heat Power and PRISM Decay Heat Power functions are shown in Fig. 29. The plot highlights the need to characterize the unique decay heat power from the U–TRU–Zr fuel of PRISM as opposed to the standard UO_2 fuel of LWRs. The fuel in PRISM emits small, but significant, additional power from its fission and decay products. This is important to validate the turbine driver blower design. If only the ANS Decay Heat Power was utilized, the needs of the system would have been dangerously undershoot.

4 Discussion

4.1 Description of Optimized Design

The above analysis concluded that the once through bypass loop design was more optimal than the closed loop design. In the once through design, the steam generator is isolated and a fraction of the steam exiting the steam generator is extracted to the Terry turbine and then vented. The rest is vented out of containment. It was concluded that this design can run on the water already present in the tertiary loop for 17.5 hours.

The extracted steam runs through a Terry turbine that mechanically powers a blower located above the steam generator inducing forced convection heat transfer along the outside of the steam generator's outer shell. Heat transfer has been optimized by designing a linearly increasing shroud gap. This creates a flow path for the air that accommodates the increasing thermal boundary layer.

This design does not require on-site power as it is completely powered by stored and decay energy present in the steam generator. It remains to be shown that the chosen configuration of once through bypass loop, Terry turbine power output, blower fan curve, and the steam generator's external geometry are capable of protecting the steam generator.

4.2 Feasibility

The CFD analysis of the steam generator determined that the turbine driven blower can remove up to 3.0 MW from the steam generator's outer surface. This is in addition to the overall heat removal rate of RVACS: ∼1% of full power or 8.4 MW. Based on the constant temperature boundary condition of the steam generator in the CFD analysis, a safety factor was placed on the heat removal capacity of the turbine driven blower. For the feasibility

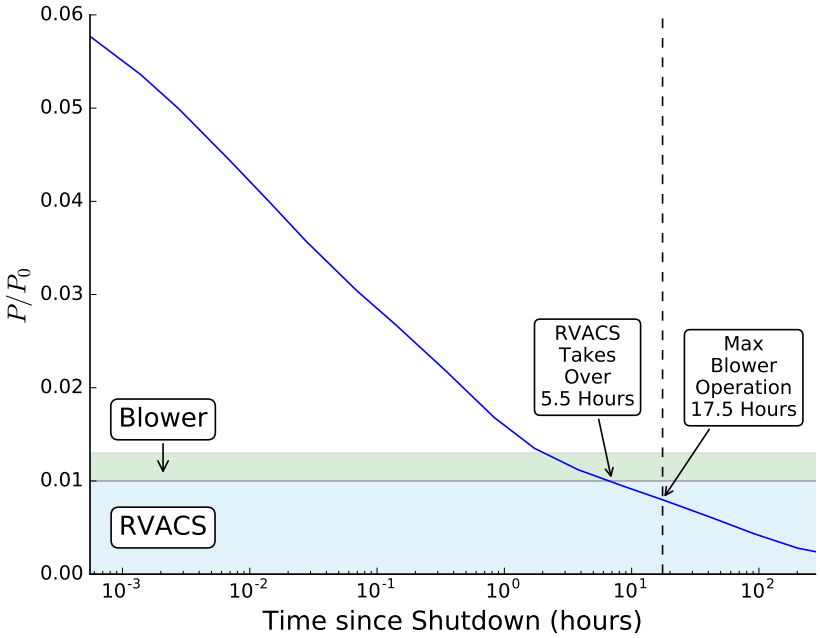

Fig. 30. A plot of the heat generation rate by the core and the heat removal rates by RVACS and the turbine driven blower. The PRISM decay curve was found in [21].

analysis, the blower was assumed to remove 2.5 MW. These heat generation and heat removal rates are shown in Fig. 30. For the design to be successful, it must be shown that the turbine driven blower removes the excess stored energy before it is unable to operate and that the resulting temperature transient does not damage the steam generator.

4.2.1 Removal of Stored Energy

The energy stored by the steam generator from the start of station blackout to time t can be defined as:

$$E(t) = \int_0^t Q_{\text{decay}}(t') - Q_{\text{blower}} - Q_{\text{RVACS}} \, \mathrm{d}t' \,, \tag{37}$$

where $Q_{\text{decay}}(t)$ is the PRISM decay heat curve found in [21]. Q_{blower} and Q_{RVACS} are assumed to be constant at 2.5 MW and 8.4 MW, respectively. Equation 37 was numerically integrated with Simpson's Rule.

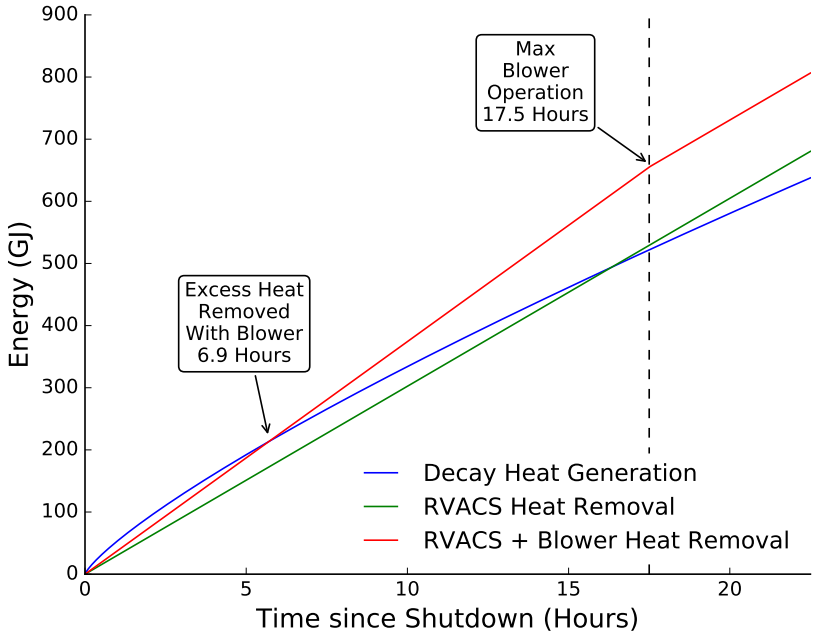

Fig. 31. Total energy from decay heat generation versus the heat removal capacity of RVACS alone and RVACS with the turbine driven blower.

Figure 31 shows the time dependent energy generation and time dependent energy removal for RVACS alone and for RVACS with the blower. The crossing point of energy generation and energy removal is when the system switches from storing energy and heating up to releasing energy and cooling down. This occurs at 16.3 hours for RVACS alone and at 6.9 hours for RVACS and the blower. The turbine driven blower is capable of running for 17.5 hours, 10.6 hours longer than is needed to clear the excess decay energy from the system. This shows that the design is capable of removing the excess stored energy with a wide safety margin of 10.6 hours.

4.2.2 Evaluation of Temperature Transient

The time dependent steam generator temperature was found through

$$T(t) = T_0 + \frac{E(t)}{mc_p(T)}\,, \tag{38}$$

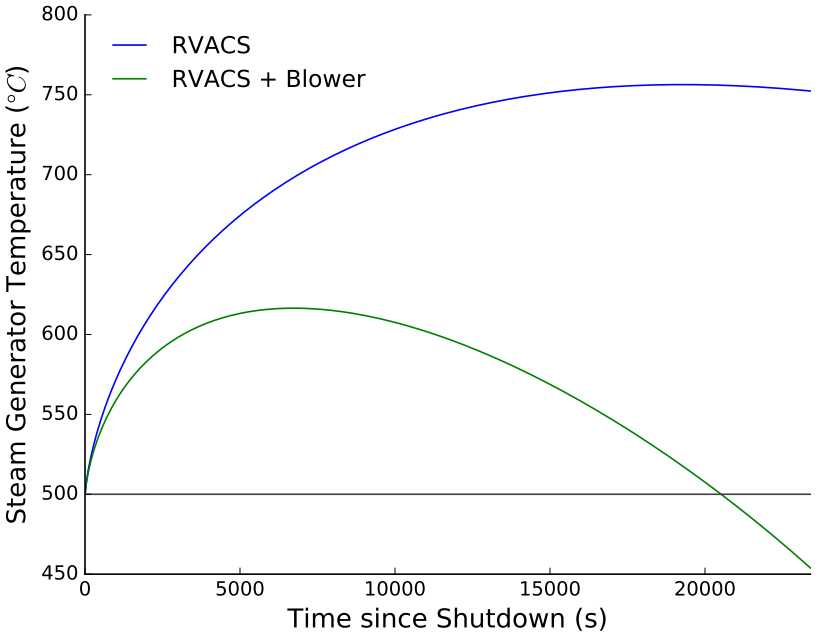

Fig. 32. Temperature of the PRISM steam generator following shutdown with and without the passively–powered blower.

where $T(t)$ is the temperature of the steam generator, $T_0 = 500\,^{\circ}\mathrm{C}$, the initial steam generator temperature, m the total mass of steel in the steam generator, and $c_p(T)$ the temperature dependent specific heat of the steel.

Due to the expected large change in temperature, the specific heat was allowed to be a function of temperature. The empirical function used was:

$$c_p(T) = 450 + 0.28 \cdot T - 2.91 \times 10^{-4} \cdot T^2 + 1.34 \times 10^{-7} \cdot T^3\,, \tag{39}$$

where $c_p(T)$ is in $J/kg/K$ and T is the temperature of the steel in $^{\circ}C$ [27]. This models the specific heat of standard stainless steel, whereas the PRISM steam generator is made of STBA 26 (Fe–9 Cr–1Mo) steel [9]. Allowing the specific heat to be a function of temperature makes Eq. 38 a nonlinear equation. This nonlinearity was handled with Picard Iteration.

Figure 32 shows the increase in the temperature of the steam generator relative to the initial

temperature T_0 following station blackout with and without the addition of the turbine driven blower. The addition of the blower reduces the maximum temperature increase on the steam generator from 260 °C with RVACS alone to 100 °C. This is below the melting point of the steel suggesting that the system will successfully protect the steam generator during station blackout conditions.

4.3 Limitations of the Models Used

A major limitation of the model is that the system is not allowed to feedback on itself. For example, cooling the steam generator with the blower would decrease its surface temperature. This would result in a lowered heat removal rate from the blower. It could also affect the steam production on the inside of the steam generator. This would propagate out to less power production from the Terry turbine and a lowered blower power. With lower blower power, less heat would be removed from the steam generator's surface starting a new cycle of heating the steam generator. The problem is made more interesting by the decaying core power. This means the real physics of the problem could be an oscillating, transient solution.

The model does not allow for interconnection between the steam generator wall temperature and the internal steam production. This was applied through the assumption of a constant surface temperature boundary condition in the CFD model and the use of several assumptions in quantifying the steam production in the steam generator. These include: the temperature water entering the bottom of the steam generator was assumed to be constant and at a quality of zero, the steam quality increased linearly through the steam generator, steam exited the the steam generator at a quality of one, and that a Nusselt correlation for uniformly heated vertical plates is applicable.

In addition, the heat removal ability of RVACS and heat transfer in the primary and secondary loops was not modeled. It was assumed that RVACS and energy losses in the pipes would contribute a flat 1% reduction from the core. In reality, RVACS's heat removal ability

is temperature dependent. The use of the high heat transfer efficiency between core and steam generator makes our analysis conservative.

These modeling inaccuracies are balanced by a large safety margin of 10.6 hours. In addition, the Terry turbine is capable of producing more power than is currently being drawn. Thus, a larger fan could be used. There also remains the possibility that the Terry Turbine could continue operating with lower quality steam but at lower power output. This would further increase the operating time and safety margin. It has been previously noted though that the Terry turbine would require a governing valve to keep the power level constant. It is currently assumed that a battery will be able to provide the power necessary to operate this valve.

4.4 Economic Analysis

The two designs considered involved using a relief valve to provide steam to a Terry turbine that powers a blower, forcing convection around the steam generator. The helical coil steam generator design has been shown to be the most cost-effective option, save the advanced straight tube steam generator, costing 5.73 million 1988 US dollars (approximately 11.57 million dollars today) according to 33 in Appendix B. There are additional advantages associated with using the helical coil steam generator [28], making it the most attractive option. The additional P91 piping associated with introducing the Terry turbine is projected to cost $1000 per ton of pipe [29]. Estimating the density of P91 pipe to be 7693 kg/m^3 and the volume of piping to be

$$V_{pipe} = \frac{\pi}{4}(D_o^2 - D_i^2) = \frac{\pi}{4}((1.524)^2 - (1.3)^2) = 99.4\text{m}^3 \tag{40}$$

yields a piping cost of $99,400 using the equation

$$Cost_{pipe} = V_{pipe} * \rho_{pipe} \tag{41}$$

Then, by multiplying the piping cost by a correction factor of four according to Marks Brothers Inc. in order to account for the qualification of nuclear grade materials, the projection increases to approximately one million dollars. The estimated cost of a blower that will work under the conditions required is projected to be $5000. Our best estimate for the cost of the Terry turbine so far is $500,000. These estimates combine for a total projected cost of 13.1 million dollars to introduce a Terry turbine system to remove decay heat. However, since the steam generator was already part of the initial design, the additional costs would be $902,600, or the cost of the Once–Through System Design. If the Continuous Loop Design were to be selected, an additional 50 m of pipe is estimated to be needed along with a passive condenser. Using Eq. 40, the extra pipe would cost $99,400 and the estimated cost of the passive condenser by multiplying the cost of a normal, small condenser is $20,000 which was obtained from Alibaba. This would adjust the estimated additional cost of the Continuous Loop Design to $1,022,000. These estimates such as the length of piping required are believed to be larger than what would actually be required, but it was determined overestimating the cost would be safer.

5 Conclusions

This analysis investigated the use of a Terry turbine driven blower to passively cool the PRISM steam generator during station blackout conditions. The most optimal bypass loop design, which extracts energy from the still hot steam generator to power the blower, was shown to be the open, once through design where all of the steam produced in the steam generator is either vented or extracted to the Terry turbine and then vented.

It was shown that the resulting mass flow rate of steam was sufficient to mechanically power a blower for the required time to protect the steam generator by comparing the integrated mass flow rate to the mass of water already present in the system.

The turbine-driven blower properties were selected in reference to previous naval turbine-driven blower experiments. Based on these properties, the fan affinity laws were used to generate a relationship between the volumetric flow rate, pressure drop, and power required by the fan. Based on the transient steam properties calculated with Python, the percentage of steam needed to be extracted from the steam generator mainline was plotted to show the time at which the Terry turbine could no longer power the blower (17.5 hours).

In addition, the steam generator's external geometry was optimized to facilitate forced convection heat transfer using a CFD model of the airflow on its outer shell. It was concluded that a linearly increasing shroud gap results in 120% greater heat transfer. The linear design was also optimized to yield maximum heat transfer rates on the outside of the steam generator.

The fission product and transuranic composition of the U–TRU–Zr fuel was determine at shutdown by modeling and burning the PRISM fuel assembly in MCNP. The resulting fuel composition can be used to characterize the decay heat power from the PRISM system.

To show the feasibility of the design, the resulting heat removal from the steam generator was compared to the PRISM decay heat curve. By generating a comparison of the stored energy in the steam generator from decay power to the heat removal rates from RVACS and the turbine driven blower, it was shown that the design is capable of removing all the excess stored energy in 6.9 hours. The system is capable of running for 17.5 hours proving that the removal of stored energy is possible with this design.

The effect of stored energy on the steam generator's temperature was also investigated to show that the resulting temperature transient was not damaging. With the assumption of a 500 °C operating temperature, the blower-cooled steam generator experienced a 100 °C

increase in temperature before the system began to cool.

This study showed that our design goal of passively protecting the PRISM steam generator during station blackout conditions has been met and that the Terry turbine driven blower and once through bypass loop design is feasible.

6 Future Work

Our design has been shown to be feasible but the models used required many assumptions to simplify the process. These assumptions were discussed in Section 4.3. This section aims to provide guidance on the creation of more physically accurate models and lists work that was not finished within the time constraints of this project.

The system's ability to feedback on itself should be investigated. This analysis would benefit from higher fidelity, transient, interconnected models of the steam generator's outer surface and internals. This could be achieved by creating a CFD model of the steam generator. The production of steam can also be more accurately determined using computational two phase flow modeling techniques. The current CFD model could be expanded to include more elements of the steam generator geometry.

An experiment on a Terry turbine could also be designed to provide data on the turbine's ability to convert steam to power and operational characteristics. This would allow for better determination of the turbine's power output. Additionally, this would allow for a better selection of Terry turbine, as it was assumed in this project that the power rating of the Turbine is not fixed.

The square root shroud gap design will also be explored and compared to the linear gap design. This will require a more advanced meshing tool than OpenFOAM's *blockMesh* utility.

Other blower models could also be incorporated into the CFD analysis. The heat transfer

on the outside of the steam generator can be furthered optimized by testing more fan curves in the CFD model to find which blowers induce the most convective heat transfer.

The effect of pump spin down immediately following a loss of power should also be investigated. If the tertiary loop remains at a high mass flow rate while the vents are open, the system would lose water much faster. Modifications to the once through design's operating procedures may need to be considered to account for this.

ORIGEN will be used to compute a decay curve. Due to time constraints, we were forced to use an outdated PRISM decay curve. While we do not expect there to be large discrepancies it is worth exploring.

Also, the secondary loop was not explicitly modeled. A more thorough analysis would include determining the heat loss as the sodium flows through the pipes and intermediate heat exchanger in the secondary loop.

Lastly, the exact details of implementing this system in a PRISM module need to be designed. This includes the system that initiates isolation of the bypass loop and the mechanical connection from the Terry turbine to the blower.

Acknowledgements

We would like to thank Drs. Eric Loewen and Russell Stachowski for providing us with this project and assisting us with information and technical help along the way. We would also like to thank our on campus advisor, Dr. Mark Kimber, for his guidance, supervision, and insight. Dr. Lane Carasik was also very helpful in reviewing our work and providing suggestions for modeling the steam production. Dr. Sunil Chirayath was instrumental in assisting with the development of the MCNP code used for this project. Finally, we would like to thank Dr. Karen Kirkland for bringing this project to our attention and providing guidance with our scoping calculations.

References

[1] B. S. TRIPLETT, E. P. LOEWEN, and B. J. DOOIES, "PRISM: A Competitive Small Modular Sodium-Cooled Reactor," **178** (2012).

[2] N. E. TODREAS and M. S. KAZIMI, *Nuclear Systems Thermal Hydraulic Fundamentals*, vol. 1, Taylor & Francis Group, LLC, 2nd ed. (2012), Reading.

[3] J. J. DUDERSTADT and L. J. HAMILTON, *Nuclear Reactor Analysis*, John Wiley & Sons (1976), Reading.

[4] E. SHWAGERAUS and E. FRIDMAN, "Decay Power Calculations for Safety Analysis of Innovative Reactor Systems," Proceedings of International Youth Nuclear Congress, IYNC08.

[5] M. HUANG and K. GRAMOLL, *Thermodynamics - Case Study Solution* (2017).

[6] AREVA, *Emergency Condenser Passive Heat Removal from a Pressure Vessel*, AREVA GmbH (2014), Reading.

[7] ENGINEERING 360, *Pump Flow*, IEEE GlobalSpec (2017), Reading.

[8] BWC TECHNOLOGIES, *Terry Turbine Overhauls* (2017), Reading.

[9] E. P. LOEWEN, *Selection of the Reference Steam Generator for the Advanced Liquid Metal Reactor*, General Electric Hitachi (2007), Reading.

[10] P. K. VIJAYAN, M. SHARMA, D. S. PILKHWAL, D. SAHA, and R. K. SINHA, *A Comparative Study of Single-Phase, Two-Phase, and Supercritical Natural Circulation in a Rectangular Loop*, J. Eng. Gas Turbines Power (2010), Reading.

[11] T. L. BERGMAN and A. S. LAVINE, *Fundamentals of Heat and Mass Transfer*, John Wiley & Sons, Inc. (2011), Reading.

[12] AMERICAN NUCLEAR SOCIETY, "Safety System Descriptions for Station Blackout Mitigation: Isolation Condenser, Reactor Core Isolation Cooling, and High-Pressure Coolant Injection," Tech. rep. (2012).

[13] J. R. BOARDMAN, "Operating experience feedback report-reliability of safety-related steam turbine-drive standby pumps used in US commercial nuclear power plants," Tech. rep., International Atomic Energy Agency (1995).

[14] Q. ZHOU, N. LI, X. CHEN, A. YONEZU, T. XU, S. HUI, and D. ZHANG, "Water Drop Erosion on Turbine Blades: Numerical Framework and Applications," *Materials Transactions*, **49**, *7*, 1606–1615 (2008).

[15] EMB-PABST, "Centrifugal Fans and Blowers," Tech. rep. (2007).

[16] W. J. A. LONDON, "Comparative Test of Three Types of Turbine-Drive Forced-Draft Fans," *Journal of the American Society of Naval Engineers*, **24**, *1* (1912).

[17] T. G. HICKS, *Handbook of Mechanical Engineering Calculations, Second Edition*, McGraw-Hill Education (2006), Reading.

[18] Y. A. SENGAL and M. A. BOLES, *Thermodynamics*, McGraw-Hill Professional, 2 ed. (2008).

[19] H. G. WELLER, G. TABOR, H.JASAK, and C. FUREBY, *A tensorial approach to computational continuum mechanics using object-oriented techniques*, vol. 12, Computers in Physics (1998), Reading.

[20] R. SEINDAL, F. PINARD, G. V. VAUGHAN, and E. BLAKE, *GNU M4*, Free Software Foundation, Inc. (2016), Reading.

[21] "PRISM Preliminary Safety Information Document Volume I," Tech. Rep. GEFR–00793, Nuclear Regulatory Commission (December 1987).

[22] J. KANG and F. V. HIPPEL, "Limited Proliferation-Resistance Benefits from Recycling Unseperated Transuranics and Lanthanides from Light-Water Reactor Spent Fuel," **13**, 169–181 (2005).

[23] D. S. GELLES, "Development of Martensitic Steels for High Neutron Damage Applications," **239**, 99–106 (1996).

[24] J. K. FINK, *Tables of Thermodynamic Properties of Sodium*, Argonne National Laboratory (1982).

[25] "Initial MCNP6 Release Overview MCNP6 Version 1.0," Tech. Rep. LA–UR–13–22934, Los Alamos National Laboratory.

[26] W. B. WILSON, S. T. COWELL, T. R. ENGLAND, A. C. HAYES, and P. MOLLER, "A Manual for CINDER'90 Version 07.4 Codes and Data," Tech. Rep. LA–UR–07–8412, Los Alamos National Laboratory (December 2007).

[27] J.-M. FRANSSEN and P. V. REAL, *Fire Design of Steel Structures*, EECS – European Convention for Constructional Steelwork (2010).

[28] J. J. WHITMAN, *Design of Passive Decay Heat Removal System for the Lead Cooled Flexible Conversion Ratio Fast Reactor* (2007), Reading.

[29] RELIANT PIPES AND TUBES PVT. LTD., *ASTM A335 P91 Alloy steel Seamless Pipe Price* (2013), Reading.

A Fuel Assembly Composition Results

TABLE 6: Fuel composition following 6– and 12–month burns.

	6–Month			12–Month		
Isotope	Mass (g)	Activity (Ci)	Act. Frac.	Mass (g)	Activity (Ci)	Act. Frac.
^{89}Sr	2.0440	5.9380×10^{4}	0.00813	2.2180	6.4430×10^{4}	0.00852
^{90}Sr	6.6310	9.3670×10^{2}	0.00013	1.3180×10^{1}	1.8610×10^{3}	0.00025
^{90}Y	1.8910×10^{-3}	1.0280×10^{3}	0.00014	3.6470×10^{-3}	1.9830×10^{3}	0.00026
^{91}Y	3.3670	8.2670×10^{4}	0.01132	3.7750	9.2680×10^{4}	0.01225
^{92}Y	1.1490×10^{-2}	1.1060×10^{5}	0.01515	1.1490×10^{-2}	1.1070×10^{5}	0.01464
^{93}Y	4.0740×10^{-2}	1.3610×10^{5}	0.01864	4.0750×10^{-2}	1.3610×10^{5}	0.01800
^{95}Zr	6.8950	1.4820×10^{5}	0.02030	7.8930	1.6960×10^{5}	0.02242
^{97}Zr	9.6620×10^{-2}	1.8490×10^{5}	0.02533	9.6700×10^{-2}	1.8500×10^{5}	0.02446
^{95}Nb	3.1590	1.2430×10^{5}	0.01703	4.1980	1.6520×10^{5}	0.02184
^{97}Nb	6.8740×10^{-3}	1.8500×10^{5}	0.02534	6.8800×10^{-3}	1.8520×10^{5}	0.02449
^{98}Nb	4.7640×10^{-6}	1.9190×10^{5}	0.02629	4.7650×10^{-6}	1.9200×10^{5}	0.02539
^{99}Nb	1.6100×10^{-5}	1.2240×10^{5}	0.01677	1.6100×10^{-5}	1.2240×10^{5}	0.01618
^{99}Mo	4.3330×10^{-1}	2.0820×10^{5}	0.02852	4.3440×10^{-1}	2.0870×10^{5}	0.02759
^{103}Ru	6.8830	2.2240×10^{5}	0.03047	7.1970	2.3260×10^{5}	0.03075
^{105}Ru	2.7370×10^{-2}	1.8410×10^{5}	0.02522	2.7490×10^{-2}	1.8490×10^{5}	0.02445
^{106}Ru	1.3030×10^{1}	4.3240×10^{4}	0.00592	2.2310×10^{1}	7.4020×10^{4}	0.00979
^{105}Rh	2.1770×10^{-1}	1.8390×10^{5}	0.02519	2.1870×10^{-1}	1.8470×10^{5}	0.02442
^{111}Ag	8.2380×10^{-2}	1.3020×10^{4}	0.00178	8.2950×10^{-2}	1.3110×10^{4}	0.00173
^{125}Sn	2.3730×10^{-2}	2.5730×10^{3}	0.00035	2.3710×10^{-2}	2.5710×10^{3}	0.00034
^{125}Sb	–	–	–	1.0740	1.1260×10^{3}	0.00015
^{132}Te	5.8580×10^{-1}	1.7800×10^{5}	0.02438	5.8620×10^{-1}	1.7810×10^{5}	0.02355
^{131}I	1.0500	1.3020×10^{5}	0.01784	1.0500	1.3030×10^{5}	0.01723
^{135}I	6.2730×10^{-2}	2.2180×10^{5}	0.03038	6.2760×10^{-2}	2.2190×10^{5}	0.02934
^{133}Xe	1.2860	2.4100×10^{5}	0.03301	1.2880	2.4140×10^{5}	0.03192
^{135}Xe	1.0170×10^{-1}	2.5850×10^{5}	0.03541	1.0180×10^{-1}	2.5880×10^{5}	0.03422
^{136}Cs	5.4510×10^{-2}	3.9800×10^{3}	0.00055	6.8430×10^{-2}	4.9960×10^{3}	0.00066
^{137}Cs	2.9780×10^{1}	2.5920×10^{3}	0.00036	5.9200×10^{1}	5.1530×10^{3}	0.00068
^{140}Ba	2.5970	1.9010×10^{5}	0.02604	2.5990	1.9020×10^{5}	0.02515
^{140}La	3.4270×10^{-1}	1.9060×10^{5}	0.02611	3.4310×10^{-1}	1.9080×10^{5}	0.02523

	6–Month			12–Month		
Isotope	Mass (g)	Activity (Ci)	Act. Frac.	Mass (g)	Activity (Ci)	Act. Frac.
^{141}Ce	6.2110	1.7710×10^{5}	0.02426	6.3490	1.8100×10^{5}	0.02393
^{143}Ce	2.3430×10^{-1}	1.5570×10^{5}	0.02133	2.3440×10^{-1}	1.5580×10^{5}	0.02060
^{144}Ce	1.5300×10^{1}	4.8740×10^{4}	0.00668	2.5160×10^{1}	8.0140×10^{4}	0.01060
^{143}Pr	2.3220	1.5640×10^{5}	0.02142	2.3110	1.5560×10^{5}	0.02057
^{147}Nd	9.1930×10^{-1}	7.4410×10^{4}	0.01019	9.2260×10^{-1}	7.4680×10^{4}	0.00987
^{147}Pm	8.9370	8.2900×10^{3}	0.00114	1.7440×10^{1}	1.6180×10^{4}	0.00214
^{148}Pm	–	–	–	8.4090×10^{-3}	1.3820×10^{3}	0.00018
^{149}Pm	1.1740×10^{-1}	4.6530×10^{4}	0.00637	1.1800×10^{-1}	4.6790×10^{4}	0.00619
^{151}Pm	3.8580×10^{-2}	2.8210×10^{4}	0.00386	3.8770×10^{-2}	2.8340×10^{4}	0.00375
^{153}Sm	3.5440×10^{-2}	1.5700×10^{4}	0.00215	3.5950×10^{-2}	1.5920×10^{4}	0.00210
^{155}Eu	–	–	–	2.0640	1.0180×10^{3}	0.00013
^{156}Eu	9.9980×10^{-2}	5.5120×10^{3}	0.00076	1.0230×10^{-1}	5.6400×10^{3}	0.00075
^{157}Eu	2.7840×10^{-3}	3.6630×10^{3}	0.00050	2.7840×10^{-3}	3.6620×10^{3}	0.00048
^{237}U	2.2160×10^{-1}	1.8090×10^{4}	0.00248	2.3450×10^{-1}	1.9140×10^{4}	0.00253
^{239}U	4.0560×10^{-2}	1.3590×10^{6}	0.18616	4.1730×10^{-2}	1.3990×10^{6}	0.18498
^{238}Np	1.3720×10^{-1}	3.5540×10^{4}	0.00487	1.3760×10^{-1}	3.5660×10^{4}	0.00472
^{239}Np	5.8580	1.3590×10^{6}	0.18616	6.0280	1.3980×10^{6}	0.18485
^{238}Pu	4.0180×10^{2}	6.8810×10^{3}	0.00094	3.9050×10^{2}	6.6880×10^{3}	0.00088
^{240}Pu	3.7670×10^{3}	8.5470×10^{2}	0.00012	3.7230×10^{3}	8.4470×10^{2}	0.00011
^{241}Pu	8.6570×10^{2}	8.9470×10^{4}	0.01226	8.1410×10^{2}	8.4130×10^{4}	0.01112
^{243}Pu	1.8540×10^{-2}	4.8230×10^{4}	0.00661	1.8990×10^{-2}	4.9410×10^{4}	0.00653
^{241}Am	3.0420×10^{2}	1.0430×10^{3}	0.00014	3.0480×10^{2}	1.0450×10^{3}	0.00014
^{244}Am	7.9270×10^{-3}	1.0080×10^{4}	0.00138	9.1390×10^{-3}	1.1620×10^{4}	0.00154
^{242}Cm	7.2740	2.4080×10^{4}	0.00330	1.0900×10^{1}	3.6080×10^{4}	0.00477
^{244}Cm	1.4290×10^{1}	1.1560×10^{3}	0.00016	1.6400×10^{1}	1.3270×10^{3}	0.00018
Totals	6.5695×10^{4}	7.3000×10^{6}	1.00000	6.5680×10^{4}	7.5630×10^{6}	1.00000

B Thermal Hydraulics Codes

Code 1: System Curve Method

```
#Calculating a system curve and finding the mass flow rate
import numpy as np
from scipy.optimize import root, fixed_point
import matplotlib.pyplot as plt
%matplotlib inline

#Define a root finding function for the friction factor equation
def fric_fac(x):
    d = 0.22
    k = 0.2e-3
    return (-2*np.log10((2.51/(Re*np.sqrt(x))) + (k/(3.71*d))) - 1.0/np.sqrt(x))

#Assume a Q (volumetric flow rate)
#Q = 0.1 #m^3/s
Q = np.arange(0.1,50,0.1)
h_f = np.zeros([Q.size])

#Calculate the head loss for all different values of Q to plot a system curve
for i in range(0,len(Q)-1):
    g = 9.81 #m/s^2
    e = 0.2e-3 #m
    #e = 2.667 #mm
    L = 8.1 #m
    #D = 31.75 #mm
    D = 0.360 #m
    A = (np.pi/4)*D**2
    v = Q[i]/A #m/s
    viscosity = sat_steam.nu

    #Caclculate Reynolds number
    Re = v*D/viscosity
    #print(Re)

    #Use Reynolds number to determine friction factor
    if Re<4000:
        f = 64/Re
    elif Re<=1e8 and (e/D)<0.05:
        system = root(fric_fac,0.02)
        f = system.x
        f = f[0]
    else:
```

```
    f = (1.14 - 0.869*np.log(e/D))**-2

#print(e/D)
#f = 0.0184 #Using the Darby-Weisbach Friction Factor Formula

#Calculate the different head losses

#Find the head loss in the pipe
h_f_pipe = f*(L/D)*(v**2/2/g)

#Create an array for all minor loss coefficients such as a bell shaped entrance
    or 90 degree elbow
#Valves
num_globe_open = 0
num_angle_open = 0
num_gate_open = 0
num_gate_quarter_closed = 0
num_gate_half_closed = 0
num_gate_3quarter_closed = 0
k_valves =
    (num_globe_open*10)+(num_angle_open*2)+(num_gate_open*0.15)+(num_gate_quarter_closed*0.26)+(
#Elbows
num_reg_90_flanged = 0
num_reg_90_thread = 0
num_longr_90_flanged = 0
num_longr_90_thread = 0
num_longr_45_thread = 0
num_reg_45_thread = 0
k_elbows =
    (num_reg_90_flanged*0.3)+(num_reg_90_thread*1.5)+(num_longr_90_flanged*0.2)+(num_longr_90_th
#Tees
num_line_flow_flanged = 0
num_line_flow_thread = 0
num_branch_flow_flanged = 0
num_branch_flow_thread = 0
k_tees =
    (num_line_flow_flanged*0.2)+(num_line_flow_thread*0.9)+(num_branch_flow_flanged*1)+(num_bran

#k = k_valves + k_elbows + k_tees
k = np.array([0.04, 0.14])
#Calculate the local losses
h_f_local = np.sum(k)*(v**2/2/g)

#Add the pipe head loss to the local head loss to find the total head loss
h_f[i] = h_f_pipe + h_f_local
#print('The total head loss is ',h_f,'m^2/s')
```

```
ax = plt.subplot(111)

ax.plot(Q, h_f, color='blue')
#ax.set_title('Assumed System Curve')
ax.set_xlabel('Q (m$^3$/s)')
ax.set_ylabel('head loss (m)')

ax.spines['right'].set_visible(False)
ax.spines['top'].set_visible(False)
ax.yaxis.set_ticks_position('left')
ax.xaxis.set_ticks_position('bottom')

ax.set_xlim(0,49.5)

plt.tight_layout()
plt.show()
```

Code 2: Original Python Loop Analysis Method

```
#Original Python Loop Analysis
import numpy as np
import scipy
from iapws import IAPWS95

#Steam leaving SG
m_dot = 759 #kg/s
T3 = 725 #K
P3 = 14.7 #MPa
steam = IAPWS95(P=P3,T=T3)
s3 = steam.s
h3 = steam.h
#print('h3 = ',h3)
#print('The entropy of the steam is ',steam.s)

#Properties of water entering the SG
T1 = 489 #K
sat_steam = IAPWS95(T=T1,x=0)
rho1 = sat_steam.rho
P1 = sat_steam.P
print('P1 = ',P1)
s_f = sat_steam.s
h_f = sat_steam.h
v_f = sat_steam.v
print('hf = ',h_f)
#print('nu = ',sat_steam.nu)
#print('s_f = ',s_f)
#print ('v_f = ',v_f,'kg/m^3')
```

```
#Properties of steam exiting the Terry Turbine
sat_vap = IAPWS95(T=T1,x=1)
s_fg = sat_vap.s - sat_steam.s
h_fg = sat_vap.h - sat_steam.h
#print('s_fg = ',s_fg)


#Assume ideal Rankine cycle where entropy doesn't increase
s4 = s3
#print('s_4 = ',s4)

#Use the entropy values to find the quality of the steam
quality = (s4 - s_f)/s_fg
#print('The quality is ',quality)

#Use the quality to find the enthalpy of the steam exiting the Terry Turbine
h4 = h_f + quality*h_fg
#print('h4 = ',h4,'kJ/kg')

#Calculate the specific work of the turbine
w_turbine = h3 - h4
#Calculate the work of the turbine
W_turbine = 0.95*m_dot*w_turbine
print('The shaft work of the orginal turbine is ',W_turbine/1000,'MW')

#To calculate the pump work by the gravity/buoyancy driven flow, need to have the
    mass flow rate and the pressure drop


#However, we have to take into account we are releasing some of the steam each time
    to make sure the rest
#is fully condensed before it re-enters the SG
#m_dot *= F
#where F is some factor relating the old and new mass flow rates after some steam is
    released

#Assume a Pressure increase based on GEH document
P2 = P1 + 0.27579 #MPa
print('P2 = ',P2)
h2 = IAPWS95(P=P2,T=489).h
print('h3 - h2 = ',h3-h2,'k/kg')
Q = 50e3 #kW
print('Q/h = ',Q/(h3-h2),'kg/s')
#print("P2 = ",P2)
w_pump = v_f*(P2-P1)*1000
#print(w_pump,'kJ/kg')
```

```
W_pump = w_pump*m_dot
#print(W_pump,'kW')

#In order to run multiple cycles, need to find how much steam is released to the
    environment
#and how that affects the thermodynamic properties and mass flow rate

P3_new = P2 #because isobaric process
T3_new = T3 #assume steam continues to exit at the same temperature
#print('The new pressure is ',P3_new)

new_steam = IAPWS95(P=P3_new,T=T3_new)
s3_new = new_steam.s
h3_new = new_steam.h
print('h3_new = ',h3_new)

#Assume ideal Rankine cycle where entropy doesn't increase
s4_new = s3_new
#print('s4_new = ',s4_new)

#Use the entropy values to find the quality of the steam
new_quality = (s4_new - s_f)/s_fg
#print('The new quality is ',new_quality)

#Use the quality to find the enthalpy of the steam exiting the Terry Turbine
h4_new = h_f + new_quality*h_fg
#print('h4_new = ',h4_new,'kJ/kg')
TT_efficiency = 0.05
w_turbine = h3_new - h4_new
W_turbine = m_dot*w_turbine*TT_efficiency
#print('The shaft work of the Terry Turbine for its first cycle is
    ',W_turbine/1000,'MW')
print('h3_new - h2 = ',h3_new-h2,'k/kg')
Q = 50e3 #kW
print('Q/h = ',Q/(h3_new-h2),'kg/s')
```

Code 3: Two–Phase Analysis Method

```
from iapws import IAPWS95
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

P = 2.13942 #MPa
T = IAPWS95(P=P,x=0).T
vf = IAPWS95(P=P,x=0).v
vg = IAPWS95(P=P,x=1).v
```

```
vfg = vg - vf
vavg = (vf+vg)/2
x = IAPWS95(T=T,P=P).x
phi2 = (1 + (vfg/vf)*x)**2
h0 = IAPWS95(P=P,x=0).h
h = IAPWS95(P=P,T=T).h
hfg = IAPWS95(P=P,x=1).h - h0
B = (1/vavg)*(vfg/hfg)
rho0 = IAPWS95(P=P,T=T).rho
mu = IAPWS95(P=P,x=0).mu
D = 2 #m
A = (np.pi/4)*D**2
b = 0.25
p = 0.316
#rho = rho0*(1-B*(h-h0))
rho = 1000 #kg/m^3
L = 20 #m
H = 4.32
g = 9.81 #m/s^2
Q = np.arange(0,500,10)
mdot = np.zeros(Q.size)
x = 0
for i in range(0,len(Q)-1):
    mdot[i] = ((2*D**(1+b)*rho**2*B*g*Q[i]*H*A**(2-b))/(p*phi2*L*mu**b))**((1/3)-b)
    x += 4/500
    phi2 = (1 + (vfg/vf)*x)**2
    rho = IAPWS95(P=P,x=x).rho

#print(rho)
#print(x)
ax = plt.subplot(111)

ax.plot(Q, mdot, color='blue')
#ax.set_title('Assumed System Curve')
ax.set_xlabel('Q (kW)')
ax.set_ylabel('mass flow rate (kg/s)')

ax.spines['right'].set_visible(False)
ax.spines['top'].set_visible(False)
ax.yaxis.set_ticks_position('left')
ax.xaxis.set_ticks_position('bottom')

ax.set_xlim(0,480)

plt.tight_layout()
plt.show()
```

Code 4: Temperature Increment Method

```
# Increment the temperature along the steam generator and use the
#outlet thermodynamic properties to find the mass flow rate
from iapws import IAPWS95 as IAP
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
D = 2 #m
L = 20 #m
T_in = 489 #K
T_out0 = 725 #K
T_inf = 773 #K
Tavg = (T_in + T_out0)/2 #K
B = 1/Tavg #K^-1
h_in = IAP(T=T_in,x=0).h
dl = 0.1

SG_vol = (np.pi/4)*D**2*L

Q = np.array([50000,40000,30000,20000,10000,5000,3000,1000]) #kW
mfr = np.zeros(Q.size)

count = 0
for j in Q:

    q = j/(np.pi*D*L)

    l = np.arange(0,20,0.1)
    T = np.zeros(l.size)
    T[0] = T_in
    x = 0


    nu =  (1.41821808601e-07 + 1.25720772599e-06)/2
    alpha = (1.64939779919e-07 + 9.66927877973e-07)/2
    k = (0.639437919031 + 0.0441971880127)/2
    cp = (4.66345906522 + 3.46308063893)/2


    for i in range(0,len(l)-1):
        #print("\nIteration {} (T={}, x={})".format(i, '%.1f'%T[i], '%.3f'%x))
        vals = IAP(T=T[i],x=x)

        rho = vals.rho #kg/m^3
        Pr = vals.Pr
        #print(" rho:", rho)
```

```
        m = rho*SG_vol #kg H20
        #print(" m:", m)

        g = 9.81 #m/s^2
        #Fund of Heat and Mass Transfer (FHMT) Eq. 9.25
        Ra = (g*B*(Tavg-T[i])*L**3)/(nu*alpha)
        #Fund of Heat and Mass Transfer (FHMT) Eq. 9.26
        Nu = (0.825 + ((0.387*Ra**(1/6))/((1 + (0.492/Pr)**(9/16))**(4/9))))**2
        #Nu = (1/24)*Ra*(D/L)*(1-np.exp(-35/(Ra*(D/L))))**(3/4)
        h = Nu*k/L #kW/m^2/K
        x += 1/199


        T[i+1] = T[i] + dl*(q/(h))

    plt.plot(l,T)
    plt.ylabel('Length along steam generator (m) ')
    plt.xlabel('Temperature (K)')
    plt.show()



    P = IAP(T=T[-1],x=1).P
    #print('The pressure for Q = ',j/1000,'MW is ',P,'MPa')
    h_out = IAP(T=T[-1],P=P).h

    del_h = h_out - h_in

    mdot = j/del_h
    mfr[count] = mdot

    print('The mass flow rate for Q = ',j/1000,'MW is ',mdot,'kg/s')
    #print('The outlet enthalpy for Q = ',j/1000,'MW is ',h_out,'kJ/kg')
    count += 1

ax = plt.subplot(111)

ax.plot(Q/1000, mfr, color='blue')
#ax.set_title('Assumed System Curve')
ax.set_xlabel('Q (MW)')
ax.set_ylabel('mass flow rate (kg/s)')

ax.spines['right'].set_visible(False)
ax.spines['top'].set_visible(False)
ax.yaxis.set_ticks_position('left')
ax.xaxis.set_ticks_position('bottom')
```

```
plt.tight_layout()
plt.show()
```

C Steam Generator Comparison

	Double Wall Straight Tube	Single Wall Construction			
		Straight Tube	Hockey Stick	Helical Coil	Advanced Straight Tube
Shell Diameter: **Mid-Section, ft** **Sodium Inlet/Outlet**	9.8 12.0	7.2 9.0	7.2 9.0	11.0 ---	6.7 9.0
Shell Height, ft	73.0	70.5	80.0	42.0	92.0
Max. Shell Width, ft **w/o support skirt**	13.8	10.8	24.0	12.8	10.8
Shell Thickness, inch ➢ **Top Head** ➢ **Bottom Head** ➢ **Mid-section** ➢ **Na Inlet/Outlet** ➢ **Bellows**	2.0 2.0 2.0 2.75 0.75	2.0 1.75 1.5 2.0 0.75	2.0 1.75 1.5 2.0 ---	2.25 2.25 2.25 --- ---	2.0 1.75 1.5 2.0 0.75 (if required)
Thermal Expansion	Shell Bellows	Shell Bellows	90 Degree Bend	Coiled Tubes	Shell Flexibility or Bellows
Tubesheet Dia. Ft	9.83	8.0	8.0	4.6 Hemisphere	7.0
Tubesheet Thickness, inch	19	15	15	6.5	13.6
Weight, Tons ➢ **Dry** ➢ **Wet**	388 533	180 256	217 316	161 251	153 229
Estimated Capital Cost of SG, millions of 1988 dollars	12.5	5.88	7.85	5.73	5.7

GE-Hitachi
Nuclear Energy

19
GEH
9/5/2007

Fig. 33. Comparison of steam generators GEH looked into [9]

D MCNP Input using GNU M4

The typical MCNP input file was generalized using GNU M4, a macroprocessor in the GNU Toolchain [20]. The usable MCNP input file (input.inp) can be generated from an M4 file (input.m4) using M4 in a Linux terminal, such as *m4 input.m4 > input.inp*.
M4 allows macros to be defined allowing the cells and surfaces within MCNP input files be easily generalized, greatly reducing the amount of work needed to make minor geometrical changes to the geometry.

Code 5: MCNP Input using GNU M4

```
dnl> --- M4 Definitions -----------------------------------------------------
dnl> M4 File of PRISM UNF Fuel Assembly
dnl> Function Definitions
changecom(//)changequote([,])dnl>
define(calc, [esyscmd(perl -e 'printf ($1)')])dnl>
define(Sqrt, [esyscmd(perl -e 'print sqrt($1)')])dnl>
define(Sin, [esyscmd(perl -e 'use Math::Trig; print sin(deg2rad($1))')])dnl>
define(Cos, [esyscmd(perl -e 'use Math::Trig; print cos(deg2rad($1))')])dnl>
dnl> Geometry Definitions
dnl> "PRISM Preliminary Safety Information Document: Volume 1"
dnl> NRC Accession # ML082880369
dnl>   active core length = 66.0 cm
dnl>   fuel diameter = 0.213 in (0.54102 cm)
dnl>   clad thickness = 0.022 in (0.05588 cm)
dnl>   pin outer diameter = 0.290 in (0.7366 cm)
dnl>   pin pitch/diameter = 1.199
dnl>   assembly pitch = 6.282 in (15.95628 cm)
dnl>   assembly thickness = 0.140 in (0.3556 cm)
define(core_length, 66.0)dnl>
define(fuel_diameter, 0.54102)dnl>
define(fuel_clad_thickness, 0.05588)dnl>
define(fuel_clad_outer_diameter, 0.7366)dnl>
define(PtD, 1.199)dnl>
define(assembly_pitch, 15.95628)dnl>
define(assembly_thickness, 0.3556)dnl>
define(fuel_clad_inner_diameter, calc(fuel_clad_outer_diameter -
   calc(2*fuel_clad_thickness)))dnl>
define(fuel_pitch, calc(PtD * fuel_clad_outer_diameter))dnl>
dnl> Hexagonal Geometry Definitions
dnl>   http://www.drking.org.uk/hexagons/misc/dims.html
define(fuel_hex_a, fuel_pitch)dnl>
define(fuel_hex_r, calc(fuel_hex_a/2))dnl>
define(fuel_hex_R, calc(fuel_hex_a/Sqrt(3)))dnl>
define(assembly_hex_a, assembly_pitch)dnl>
define(assembly_hex_r, calc(assembly_hex_a/2))dnl>
```

```
define(assembly_hex_R, calc(calc(assembly_hex_a/Sqrt(3))))dnl>
define(assembly_hexinner_r, calc(assembly_hex_r - assembly_thickness))dnl>
define(assembly_hexinner_R, calc(assembly_hex_R - assembly_thickness))dnl>
dnl> --- End of M4 ---------------------------------------------------------
dnl>
dnl> --- MCNP Input ---------------------------------------------------------
PRISM UNF Fuel Assembly
c --- Cell Cards
2001 0      -1101 +1102 -1103 +1104 -1105 +1106 +100 -101 fill=2 imp:n=1
2002 0      -1301 +1302 -1303 +1304 -1305 +1306 u=2 lat=2 imp:n=1 $ Hex Lattice
     fill=-10:10 -10:10 0:0
     12 20R
     12 9R          11 11 11 11 11 11 11 11 11 11              12
     12 8R         11 11 11 11 11 11 11 11 11 11 11            12
     12 7R        11 11 11 11 11 11 11 11 11 11 11 11          12
     12 6R       11 11 11 11 11 11 11 11 11 11 11 11 11        12
     12 5R      11 11 11 11 11 11 11 11 11 11 11 11 11 11      12
     12 4R     11 11 11 11 11 11 11 11 11 11 11 11 11 11 11    12
     12 3R    11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11  12
     12 2R   11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 12
     12 1R  11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 12
     12    11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 12
     12     11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 12 1R
     12      11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 12 2R
     12       11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11  12 3R
     12        11 11 11 11 11 11 11 11 11 11 11 11 11 11 11    12 4R
     12         11 11 11 11 11 11 11 11 11 11 11 11 11 11      12 5R
     12          11 11 11 11 11 11 11 11 11 11 11 11 11        12 6R
     12           11 11 11 11 11 11 11 11 11 11 11 11          12 7R
     12            11 11 11 11 11 11 11 11 11 11 11            12 8R
     12             11 11 11 11 11 11 11 11 11 11              12 9R
                                                               12 20R
2003 1  -15.98 -1001 +100 -101      vol=15.1726 tmp=8.6170e-8 u=11 imp:n=1 $ Fuel
   slug
2004 0         +1001 -1002 +100 -101             u=11 imp:n=1 $ Gas gap
2005 15 -8.1   +1002 -1003 +100 -101 tmp=7.7553e-8 u=11 imp:n=1 $ Clading
2006 5  -0.852 +1003 -1010 +100 -101 vol=16.4584 tmp=6.0319e-8 u=11 imp:n=1 $ Coolant
2010 5  -0.852 -1010 +100 -101      vol=44.5837 tmp=6.0319e-8 u=12 imp:n=1 $ Coolant
   filled structure
2011 15 -8.1  (+1101:-1102:+1103:-1104:+1105:-1106) &
               -1201 +1202 -1203 +1204 -1205 +1206 +100 -101 &
               tmp=7.7553e-8 imp:n=1 $ Assembly surround
99999 0        (+1201:-1202:+1203:-1204:+1205:-1206:-100:+101) imp:n=0 $ Outside
   world

c --- Surface Cards
c - Height
```

```
100  pz  -calc(core_length/2)
101  pz  calc(core_length/2)
c - Fuel Pin
1001 cz calc(fuel_diameter/2)     $ Fuel slug
1002 cz calc(fuel_clad_inner_diameter/2) $ Gas gap
1003 cz calc(fuel_clad_outer_diameter/2) $ Clading
1010 cz calc(fuel_clad_outer_diameter) $ Coolant
c - Assembly
1101 px  assembly_hexinner_r $ hexagonal side FA
1102 px -assembly_hexinner_r $ hexagonal side FA
1103 p   1  calc(calc(2*assembly_hexinner_r)/assembly_hexinner_R) 0
   calc(2*assembly_hexinner_r) $ Q1 hexagonal sides
1104 p   1  calc(calc(2*assembly_hexinner_r)/assembly_hexinner_R) 0
   -calc(2*assembly_hexinner_r) $ Q2 hexagonal sides
1105 p  -1  calc(calc(2*assembly_hexinner_r)/assembly_hexinner_R) 0
   calc(2*assembly_hexinner_r) $ Q3 hexagonal sides
1106 p  -1  calc(calc(2*assembly_hexinner_r)/assembly_hexinner_R) 0
   -calc(2*assembly_hexinner_r) $ Q4 hexagonal sides
1201* px assembly_hex_r $ hexagonal side FA
1202* px -assembly_hex_r $ hexagonal side FA
1203* p  1  calc(calc(2*assembly_hex_r)/assembly_hex_R) 0 calc(2*assembly_hex_r) $ Q1
   hexagonal sides
1204* p  1  calc(calc(2*assembly_hex_r)/assembly_hex_R) 0 -calc(2*assembly_hex_r) $
   Q2 hexagonal sides
1205* p -1  calc(calc(2*assembly_hex_r)/assembly_hex_R) 0 calc(2*assembly_hex_r) $ Q3
   hexagonal sides
1206* p -1  calc(calc(2*assembly_hex_r)/assembly_hex_R) 0 -calc(2*assembly_hex_r) $
   Q4 hexagonal sides
1301 py  fuel_hex_r $ top hexagonal side
1302 py -fuel_hex_r $ bottom hexagonal side
1303 p   calc(calc(2*fuel_hex_r)/fuel_hex_R) 1 0 calc(2*fuel_hex_r) $ Q1 hexagonal
   sides
1304 p   calc(calc(2*fuel_hex_r)/fuel_hex_R) 1 0 -calc(2*fuel_hex_r) $ Q2 hexagonal
   sides
1305 p   calc(calc(2*fuel_hex_r)/fuel_hex_R) -1 0 calc(2*fuel_hex_r) $ Q3 hexagonal
   sides
1306 p   calc(calc(2*fuel_hex_r)/fuel_hex_R) -1 0 -calc(2*fuel_hex_r) $ Q4 hexagonal
   sides

mode n
mphys
kcode 10000 1 50 500
ksrc  0 0 calc(core_length/4)
      0 0 -calc(core_length/4)
c --- Burn Card
c - 4.375 MWth per fuel assembly (840 MWth w/ 192 assemblies)
c burn time=0.3 0.3 0.4 1 2 5 21 30 30 30 30 30 30 30 30 30 30 30
```

```
c      mat=1 5
c      matvol=4111.78 5753.16
c      power=4.375
c      pfrac=1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
c      bopt=1 24 1
c --- U-TRU-Zr Fuel (U-22.7%TRU-10%Zr with 13.5% fissile Pu)
c - "PRISM: A Competitive Small Modular Sodium-Cooled Reactor"
c - B.S. Triplett, E.P. Loewen, B.J. Dooies (http://dx.doi.org/10.13182/NT178-186)
m1    40090.82c -0.05145
      40091.82c -0.01122
      40092.82c -0.01715
      40094.82c -0.01738
      40096.82c -0.00280
      92235.82c -0.00485
      92238.82c -0.66815
      93237.82c -0.00341
      93239.82c -8.26e-10
      94236.82c -5.03e-10
      94238.82c -0.00635
      94239.82c -0.12097
      94240.82c -0.05797
      94241.82c -0.01403
      94242.82c -0.01853
      95241.82c -0.00459
      95242.82c -4.80e-6
      95243.82c -0.00093
      96243.82c -2.63e-6
      96244.82c -0.00019
      96245.82c -2.01e-5
c --- Sodium Coolant
c - Mg-24 impurity added to track Na-24 n-capture reaction
m5    11023.81c +1.00
      12024.81c +2.0e-10
c --- HT-9 Steel (Fe-12Cr-1Mo-0.5W-0.2C)
c - "Development of Materensitic Steels for High Neutron Damage Applications"
c - D.S. Gelles (http://dx.doi.org/10.1016/S0022-3115(96)00474-6)
m15    6012.82c -0.00200
      24050.82c -0.00527
      24052.82c -0.10167
      24053.82c -0.01153
      24054.82c -0.00287
      26054.82c -0.05061
      26056.82c -0.79373
      26057.82c -0.01834
      26058.82c -0.00242
      42092.82c -0.00148
      42095.82c -0.00160
```

```
      42096.82c -0.00169
      42098.82c -0.00246
      74182.82c -0.00134
      74184.82c -0.00155
      74186.82c -0.00144
dnl> --- End of MCNP ----------------------------------------------------------
```

E CFD Post Processing Codes

Code 1: GCI

```
#!/usr/bin/env python3

import numpy as np
import matplotlib.pyplot as plt

from hidespines import *

def calcGCI(f, N, tol=1e-9):
   ''' computes observed order of convergence and sigma from
      non uniform refinement
   '''

   f1 = f[0]
   f2 = f[1]
   f3 = f[2]

   BAD = ''

   # check for oscillatory convergence
   if (f2/f1 - 1> 0 and f2/f3 - 1> 0) or (f2/f1 - 1< 0 and f2/f3 - 1< 0):
      # print('oscillatory convergence', f2/f1, f2/f3)

      BAD += 'oscillatory '
   # asymptotic convergence, good for GCI
   elif (f1 < f2 < f3 or f1 > f2 > f3):
      BAD += ''

   else:
      # print('unknown behavior')
      BAD += 'unknown '

   # calculate refinement factor
   r12 = (N[0]/N[1])**(1/1)
   r23 = (N[1]/N[2])**(1/1)

   converged = 0 # store if converged, break if 1
   pold = 1 # guess 2
   alpha = np.fabs((f3 - f2)/(f2 - f1)) # ratio of f's, common for iteration
   q = lambda p: np.log((r12**p - alpha/np.fabs(alpha))/(r23**p -
       alpha/np.fabs(alpha)))
   while (not(converged)):
      p = np.fabs(np.log(alpha) + q(pold))/np.log(r12)
      # compare to old p
```

```
      if (np.fabs(p - pold)/pold < tol):
         converged = 1

      # update pold
      pold = p

   # set factor of safety
   Fs = 3
   if (np.fabs(p - 2)/2 < .1): # if close to expected p = 2
      Fs = 1.25

   # deviation
   sigma = Fs/(r12**p - 1) * np.fabs(f1 - f2)

   if (p < 0): # negative order of convergence
      # print('p negative')
      BAD += 'negative p '

   if (sigma < 0): # negative uncertainty
      BAD += 'negative sigma '

   if (p > 10): # p too large
      # print('p > 20')
      BAD += 'p too large '

   return p, sigma, BAD

Ny, Nz, whf = np.loadtxt('gci', unpack=True)

N = Ny*Nz

p, sigma, bad = calcGCI(whf, N)

print(p, sigma, bad)

plt.plot(Ny*Nz/1e6, whf/1e6, '-o', clip_on=False)
plt.xlabel('Number of Volumes (in millions)', fontsize=16)
plt.ylabel('Heat Removal Rate (MW)', fontsize=16)
hidespines(plt.gca())
plt.savefig('gci.pdf')
plt.show()
```

Code 2: Heat Transfer Coefficient

```
#!/usr/bin/env python3

import numpy as np
```

```
import matplotlib.pyplot as plt

from util.geth import *

# mu = 1.831e-5
mu = 2.286e-5
mus = 2.670e-5
# Pr = .705
Pr = .688
# k = .0257
k = 3.365e-2
P = 1e5
Tb = 293
Ts = 500
Tf = .5*(Tb + Ts)

print('Film Temperature =', Tf)

R = 8.314
nmol = 28.96

U = 20
L = 20
r = .4
D = 2*r

# rho = P/(287.058 * Tb)
rho = 1.18586764829

print('rho =', rho)

Re = rho*U*D/mu

print('Re = {:.6e}'.format(Re))

print('lam entrance =', .05*Re)
print('lam T entrance =', .05*Re*Pr)

print('turb entrance =', 1.359*Re**(1/4))
print('turb T entrance =', 10)

qT = k/D*.023*Re**(4/5)*Pr**.4 * (Ts - Tb)
qL = k/D*3.66*(Ts - Tb)

print('qT =', qT)
print('qL =', qL)
```

```
z, h, rho = getH(Ts)

hb = k/D * .023 * Re**(4/5) * Pr**.4
hs = k/D * .027 * Re**(4/5) * Pr**(1/3) * (mu/mus)**.14
hl = 3.66*k/D

print('Percent Error =', np.fabs(h[-1] - hb)/hb*100)

print(np.mean(rho))

plt.figure()
plt.semilogy(z/D, h)
plt.axhline(hb, color='k', alpha=.5)
# plt.axhline(hs, color='b', alpha=.5)
plt.axhline(hl, color='b', alpha=.5)
plt.xlabel(r'$z/D$')
plt.ylabel('Heat Transfer Coefficient')

plt.figure()
plt.plot(z/D, rho)

plt.show()
```

Code 3: Reconstructing OpenFOAM Heat Flux

```
#!/usr/bin/env python3

import numpy as np
import matplotlib.pyplot as plt

import os

def getTdir():

   # find largest time directory
   tdir = '0'
   for f in os.listdir('.'):

      if (f[0].isdigit() and f[-1].isdigit() and os.path.isdir(f)):

         if (float(f) > float(tdir)):

            tdir = f

   tdir += '/'

   return tdir
```

```
def getH(Ts):

   tdir = getTdir()

   print('using time directory ' + tdir)

   initDir = '0/'

   # get z for wall patch
   zfile = initDir + 'ccz'

   z = [] # store z locations

   f = open(zfile, 'r')

   for line in f:

      if line.startswith('boundaryField'):

         next(f)
         line = next(f)

         if ('wall' in line):

            for line in f:

               if line.startswith('('):

                  for line in f:

                     if (line.startswith(')')):
                        break

                     z.append(float(line.strip()))

                  break

   z = np.array(z)
   f.close()

   # get wall heat flux
   f = open(tdir + 'wallHeatFlux', 'r')

   whf = [] # store wall heat flux

   for line in f:
```

```
      if line.startswith('boundaryField'):

         next(f)
         line = next(f)

         if ('wall' in line):

            for line in f:

               if line.startswith('('):

                  for line in f:

                     if (line.startswith(')')):

                        break

                     whf.append(float(line.strip()))

                  break

   whf = np.array(whf)
   f.close()

   sdir = 'postProcessing/singleGraph/' + tdir

   zc, Tc, rho = np.loadtxt(sdir + 'line_T_rho.xy', unpack=True)

   z = z[:-1]
   whf = whf[:-1]

   h = whf/(Ts - Tc)

   return z, h, rho

def getU():

   tdir = getTdir()

   zc, Uz = np.loadtxt('postProcessing/singleGraph/' + tdir + 'line_U.xy',
      unpack=True, usecols=(0,3))

   return zc, Uz
```

F Contributing Authors

Section	Authors
Executive Summary	E. Gonzalez, S. Olivier, J. Norris, C. Bowman
Introduction	E. Gonzalez, S. Olivier, C. Bowman, J. Norris
Objectives	S. Olivier, C. Bowman
Thermodynamic Analysis	C. Bowman
Optimization of Turbine-Driven Blower	E. Gonzalez
Heat Removal from the Steam Generator	S. Olivier
Modeling PRISM Fuel Assembly	J. Norris
Description of Optimized Design	S. Olivier
Feasibility	J. Norris, S. Olivier
Limitations of Model	S. Olivier, C. Bowman, E. Gonzalez
Economic Analysis	C. Bowman
Conclusions	S. Olivier, J. Norris, E. Gonzalez, C. Bowman
Future Work	S. Olivier
Acknowledgments	C. Bowman