

Santa Clara University

Scholar Commons

Mechanical Engineering Master's Theses

Engineering Master's Theses

6-10-2021

Predictive energy optimization model for grid-interactive residential buildings

Patrick J. McCurdy

Follow this and additional works at: https://scholarcommons.scu.edu/mech_mstr



Part of the [Mechanical Engineering Commons](#)

SANTA CLARA UNIVERSITY

Department of Mechanical Engineering

I HEREBY RECOMMEND THAT THE THESIS PREPARED
UNDER MY SUPERVISION BY

Patrick J. McCurdy

ENTITLED

PREDICTIVE ENERGY OPTIMIZATION MODEL FOR
GRID-INTERACTIVE RESIDENTIAL BUILDINGS

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF

MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING



6/9/2021

Thesis Advisor: Dr. Hohyun Lee

date



6/9/2021

Thesis Reader: Dr. On Shun Pak

date



6/11/2021

Department Chair: Dr. Drazen Fabris

date

PREDICTIVE ENERGY OPTIMIZATION MODEL FOR GRID-INTERACTIVE RESIDENTIAL BUILDINGS

By

PJ McCurdy

MASTER'S THESIS

Submitted to the Department of Mechanical Engineering

of

SANTA CLARA UNIVERSITY

in Partial Fulfillment of the Requirements for the degree of
Master of Science in Mechanical Engineering

Santa Clara, California

June 10, 2021

Predictive energy optimization model for grid-interactive residential buildings

PJ McCurdy

Department of Mechanical Engineering
Santa Clara University
2021

ABSTRACT

The consumption and production of energy are more dynamic as distributed energy resources (DER) such as solar photovoltaic (PV) are deployed within the electric distribution system. The traditional energy price at a predetermined rate cannot accommodate these dynamics and can lead to wasted energy and higher costs for both utility companies and consumers. Commercial building and residential energy management systems are usually on a fixed schedule and are not able to respond to changes in energy price instantaneously. There is a need for a real-time pricing structure that can accommodate the fluctuating cost of energy based on supply and demand, and a need for an energy management system that is able to respond to the dynamic utility rate. As such, there is a need for a robust energy management control strategy and methodology to validate new approaches. To address this gap, a strategy to control HVAC systems in a residential house was developed along with a validation methodology. A model of predictive control was implemented to optimize the thermostat setpoints and minimize energy cost for an individual residential house while maintaining thermal comfort of users. Using the dynamic pricing, current indoor temperature, and predicted outdoor temperature and solar radiation, the control algorithm optimizes the energy consumption by adjusting the temperature setpoint on an hourly basis. This model was integrated with EnergyPlus simulation via an open source co-

simulation platform previously developed at the U.S. National Institute of Standards and Technology (NIST). Total energy consumption and cost for consumers were compared between four energy control cases: fixed setpoint, fixed comfort zone with optimization, adaptive comfort zone control, and adaptive comfort zone control with optimization. Control strategies with optimization were found to reduce the total cost compared to those without optimization. Adaptive comfort zone control with optimization resulted in the most significant cost reduction. The simple dynamic pricing model used in simulations was proportional to the demand of energy at that time of day. This work will contribute to the development of utility dynamic pricing models and residential control strategies for grid-interactive buildings and homes. The simulation strategy enables the utility pricing models and control strategies to be tested independently so that a wide range of options can be considered. The outcome of this research can be expanded to different building models or locations in future work.

Keywords: optimization; transactive energy; residential buildings; cost savings; co-simulation;

Acknowledgements

There are many people to thank for the success of this research. The first of which is the project's advisor, Dr. Hohyun Lee of the Santa Clara University Mechanical Engineering Department, who was instrumental in providing guidance throughout the entirety of this research. My labmates, Dr. Chenli Wang and Kaleb Pattawi, provided critical support, guidance, and solidarity throughout the research. National Institute of Standards and Technology (NIST), specifically Tom Roth and Cuong Nguyen, provided support for the research and guidance regarding NIST's co-simulation platform. Dr. Yuhong Liu of the Santa Clara University Computer Science and Engineering Department also provided support. Lastly, thank you to my families and friends for providing support throughout the length of this research.

Table of Contents

1. Introduction.....	1
1.1. Existing Residential Energy Control Strategies.....	10
1.2. Existing Pricing Models.....	13
1.3. Existing Validation Techniques.....	15
1.4. Statement of Goals.....	16
1.5. Impact	17
2. Optimizer	18
2.1. Indoor Temperature Prediction.....	18
2.2. Optimization	24
2.3. Setpoint Temperature Schedule	29
2.4. Pricing Model.....	31
3. Validation Tool.....	35
4. Simulation Setup.....	39
5. Results.....	42
6. Conclusion	53
6.1. Future Work.....	54

References.....	56
Appendices.....	62
Appendix A. MATLAB Linear Regression Code for Indoor Temperature Prediction	62
Appendix B. Python Optimization Code	64
Appendix C. Java Code for Socket Federate in UCEF.....	69
Appendix D. Java Code for Controller Federate in UCEF.....	77
Appendix E. EnergyPlus Input (IDF) File	92
Appendix H. Pricing	174
Appendix I. Excel Solver Data	175

List of Figures

Figure 1. Plot of the annual average global surface temperature relative to 1951-1980 average temperatures (data from NASA [4]).	2
Figure 2. Increasing trend of the annual energy consumption in the United States normalized to the amount in 1949 (data from EIA [1]) and the portion of building energy consumption to total annual energy consumption is also increasing over the past seventy years.....	3
Figure 3. Plot of the total capacity of renewable energy sources in the world from 2000 to 2021 (data from the International Energy Agency (IEA) [8]).	4
Figure 4. Plot of the typical energy demand curve from 2012-2020 during spring in California, (figure from California Independent System Operator (CAISO) [9], used without permission) ...	5
Figure 5. Example of a demand curve and load shifting before and after the peak demand.	6
Figure 6. An example of possible data and electricity flow under transactive energy (figure taken from Zia [14], used without permission).	8
Figure 7. Plot of a Time of Use (TOU) pricing plan from the utility PG&E with the energy demand curve (data from CAISO [15]).	9
Figure 8. Adaptive comfort zone based on outdoor temperature compared to a traditional setpoint control. The comfortable zone (green grid area) and acceptable zone (yellow pattern area) depict the temperature ranges accepted by 90 % and 80 % of people accordingly.	12
Figure 9. Climate zones and moisture regimes in the United States. (Figure from IECC [26], taken without permission).....	13

Figure 10. RMSE of the indoor temperature prediction in °C by size of training set.....	20
Figure 11. Comparison of energy usage predicted by the prediction method and the EnergyPlus data.....	21
Figure 12. Comparison of indoor temperature predicted by the prediction method and the EnergyPlus data.	22
Figure 13. Average percent difference between actual and predicted indoor temperature for each hour of the day.	23
Figure 14. Average percent difference between actual and predicted indoor temperature for each month.	24
Figure 15. Optimal energy consumption result from Excel Solver and CVXOPT test cases.....	28
Figure 16. Indoor temperature result from Excel Solver and CVXOPT test cases.	29
Figure 17. Comparison of modeled indoor temperature with and without fuzzy control.....	31
Figure 18. Wholesale market price of energy according to CAISO for January 1 to 7, 2021 and February 12 to 18, 2021.....	32
Figure 19. Proposed dynamic pricing models for January 1 to 7, 2021 and February 11 to 18, 2021 compared to PG&E TOU plan.	33
Figure 20. Communication exchange between EnergyPlus, controller, market, and weather forecast.....	37
Figure 21. Sequence diagram of simulation with EnergyPlus and UCEF.....	38

Figure 22. Diagram of the exchange of information between EnergyPlus and the co-simulation platform to find the optimal setpoint temperature.	40
Figure 23. Energy consumption and cost for each energy control strategy for January 1 - 7.	43
Figure 24. Energy consumption and cost for each energy control strategy for February 12 - 18.	44
Figure 25. Energy consumption and cost for each energy control strategy for Sept 27 - Oct 3. .	45
Figure 26. a) Plot of the energy consumption for a fixed setpoint and optimization with a fixed comfort zone over a three day time period with the price of energy at each timestep b) Plot of the energy consumption for a fixed setpoint and optimization with a fixed comfort zone over a 12 hour period (Jan 2 12-24) with the price of energy at each timestep.	46
Figure 27. Plot of the indoor temperature for the two cases over Jan 1 - 3 with the price of energy at each timestep.	47
Figure 28. Plot of indoor temperature with the adaptive comfort zone marked where the indoor temperature naturally exceeds the comfort zone due to weather conditions.	48
Figure 29. Plot of the energy consumption during the time when the indoor temperature naturally exceeds the comfort zone due to weather conditions.	49
Figure 30. Indoor temperature in February during a time of high price increase.	50
Figure 31. Energy consumption in February during a time of high price increase.	50
Figure 32. Plot of energy consumption during a change in price.	51

List of Tables

Table 1. List of five US cities and their climate zones for the pricing model review.....	14
Table 2. The values of the unknown constants in the indoor temperature prediction equation, solved for by linear regression.....	20
Table 3. Excel Solver setup to validate early results from CVXOPT.....	28
Table 4. Summary of cost savings between simulations with optimization and the corresponding simulations without optimization.....	42

Nomenclature

T_{indoor}	Indoor Temperature [$^{\circ}\text{C}$]
dt	Timestep
U	Overall Heat Transfer Coefficient [$\text{W}/\text{m}^2\text{K}$]
A	Surface Area of the Building Envelope [m^2]
m	Mass of the Building [kg]
c	Specific Heat of the Building [J/kgK]
$T_{outdoor}$	Outdoor Temperature [$^{\circ}\text{C}$]
Q	Energy [J]
Q_{solar}	Solar Radiation [W/m^2]
E_{used}	Energy Consumption [kWh]
C_1	Constant for Energy Prediction Model [1/s]
C_2	Constant for Energy Prediction Model [$^{\circ}\text{C}/(\text{kWh s})$]
C_3	Constant for Energy Prediction Model [$^{\circ}\text{C}/(\text{Js})$]
CF	Cost Function
P	Price of energy [$\$/\text{kWh}$]
E_{max}	Maximum Energy Consumption [kWh]

$T_{comfort, lower}$	Lower Bound of Temp. Comfort Zone [°C]
$T_{comfort, upper}$	Upper Bound of Temp. Comfort Zone [°C]
α	Proportional Constant of Utility Rate
β	Delivery Charge Per kWh [\$/kWh]

Abbreviations

EIA	US Energy Information Administration
DER	Distributed Energy Resources
PV	Photovoltaics
IEA	International Energy Agency
CAISO	California Independent System Operator
TOU	Time-of-Use
PG&E	Pacific Gas & Electric
HEMS	Home Energy Management System
MPC	Model Predictive Control
PID	Proportional-Integral-Derivative
ASHRAE	American Society of Heating, Refrigeration, and Air- Conditioning Engineers
PMV	Predicted Mean Vote
HVAC	Heating, Ventilation, and Air-Conditioning
IECC	International Energy conservation Code
UCEF	Universal Cyber-Physical Systems Environment for Federation

NIST	U.S. National Institute of Standards and Technology
RMSE	Root Mean Square Error
FMU	Functional Mockup Unit
TMY	Typical Meteorological Year

1. Introduction

Over the last 70 years, energy consumption in the US has tripled [1] and has contributed to negative impacts on the environment. The Intergovernmental Panel on Climate Change, a group of 1,300 independent scientific experts from countries all over the world, reported with more than 95 percent probability that human activities over the past 50 years have warmed our planet [2]. Their report states that human activities have raised atmospheric carbon dioxide levels from 280 parts per million to 414 parts per million in the last 150 years. The panel found that human-produced greenhouse gases such as carbon dioxide, methane, and nitrous oxide have caused much of the observed increase in Earth's temperatures over the past 50 years. This change in temperature is visible in Figure 1 which shows the change in the average global temperature over the last 140 years relative to the 1951-1980 average temperature with data from NASA's Goddard Institute for Space Studies (GISS) [3]. Contrary to the trend of staying relatively consistent over time, the temperature has risen rapidly over the past 50 years, experiencing the warmest year on record in 2016 and 2020 at 1.02 °C over the 1951-1980 average temperature.

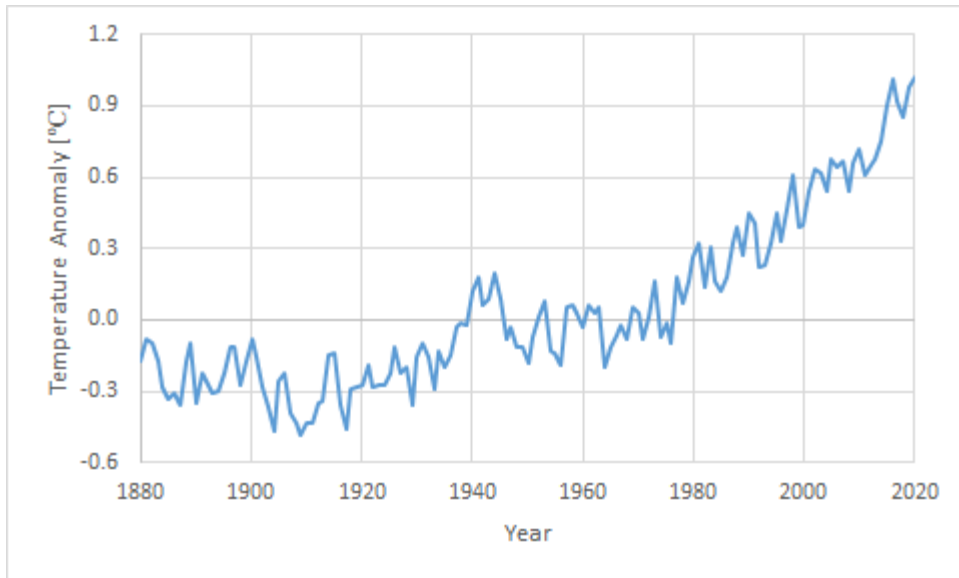


Figure 1. Plot of the annual average global surface temperature relative to 1951-1980 average temperatures (data from NASA [4]).

The average sea level has been rising 3.3 millimeters per year during the past 30 years, threatening the areas with low elevation [5]. Bangladesh, one of the most densely populated countries in the world, is estimated to lose 18 % of its land if sea level rises about 1 meter, affecting millions of people. On the global scale, a 1 meter rise in sea level will take over 220,703 km² country area and impact 57 million people.

In 2019, residential and commercial building energy usage accounted for 40 % of energy consumption in the United States [1]. Among the end-use sectors, the energy consumption from buildings has the highest growth rate and has grown by over four times since 1949 [1]. Figure 2 shows the building energy consumption ratio with respect to that of the year 1949 and the proportion of building energy consumption to total annual energy usage. The total annual buildings energy consumption in the United States has grown from 9268 trillion Btu to 39387 Btu. The contribution of building energy usage to total raised from 29 % to 40 %.

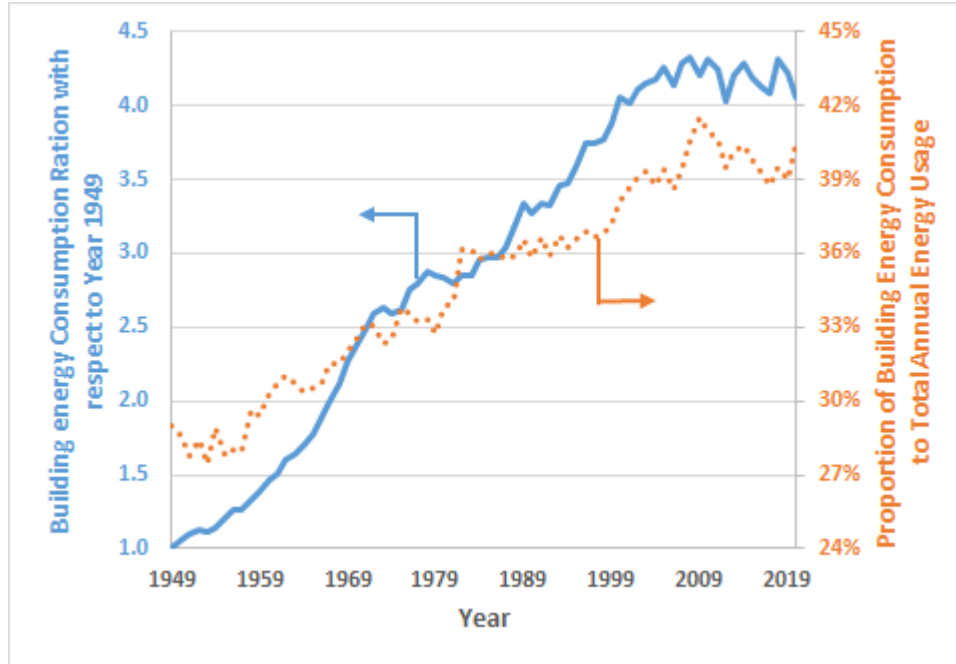


Figure 2. Increasing trend of the annual energy consumption in the United States normalized to the amount in 1949 (data from EIA [1]) and the portion of building energy consumption to total annual energy consumption is also increasing over the past seventy years.

Residential energy consumption makes up 55% of building energy consumption [1]. The average U.S. residential buildings use about 25 % more electricity than necessary [6]. In 2019, the average household used energy that released an estimated 17,320 lbs. of carbon dioxide, which is 70 % more than the average passenger vehicle [7].

The power generation from renewable energy sources can slow down the negative impacts on the environment since their use emits less carbon dioxide than fossil fuels [8]. The use of distributed energy resources (DER) such as solar photovoltaics (PV) has significantly grown in the past two decades. Figure 3 shows the total capacity of renewable energy sources since 2020. The total capacity of renewable energy sources has grown by over 3 times within that time frame.

Although these renewable energy sources can reduce greenhouse gas emissions, there are

challenges in integrating DER due to factors such as uncertainty in output, unknown reliability, and limited capacity.

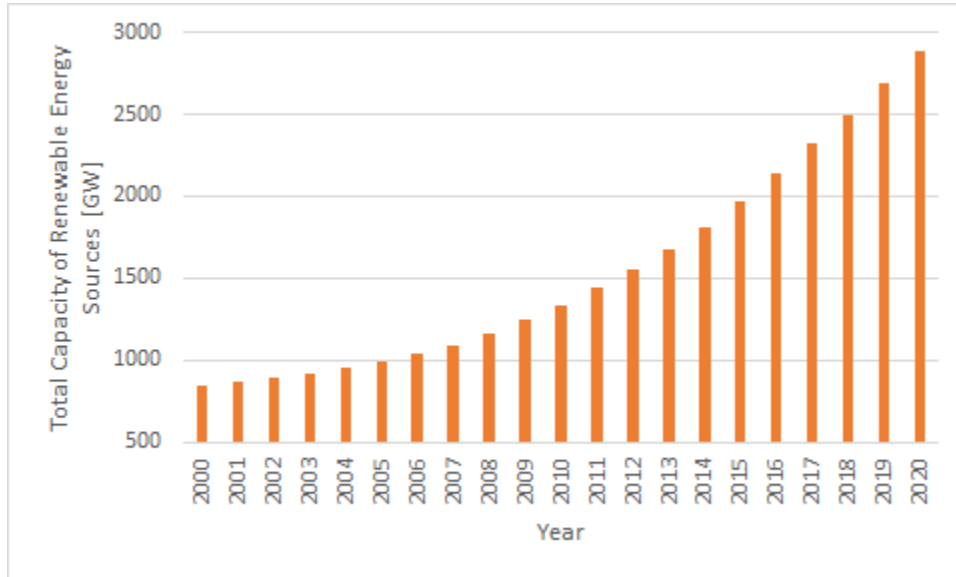


Figure 3. Plot of the total capacity of renewable energy sources in the world from 2000 to 2021 (data from the International Energy Agency (IEA) [8]).

As DER, especially PV, have become more common, they have changed the energy demand curve. The typical trend of energy demand is shown in Figure 4 with varying levels of DER penetration. During the day when PV cells in particular are generating a significant amount of energy, the net load decreases significantly, potentially to the point when more energy is generated than needed. Additionally, utility companies face the new challenge of increasing the rate at which they generate energy to meet demand in the evening as solar generation decreases.

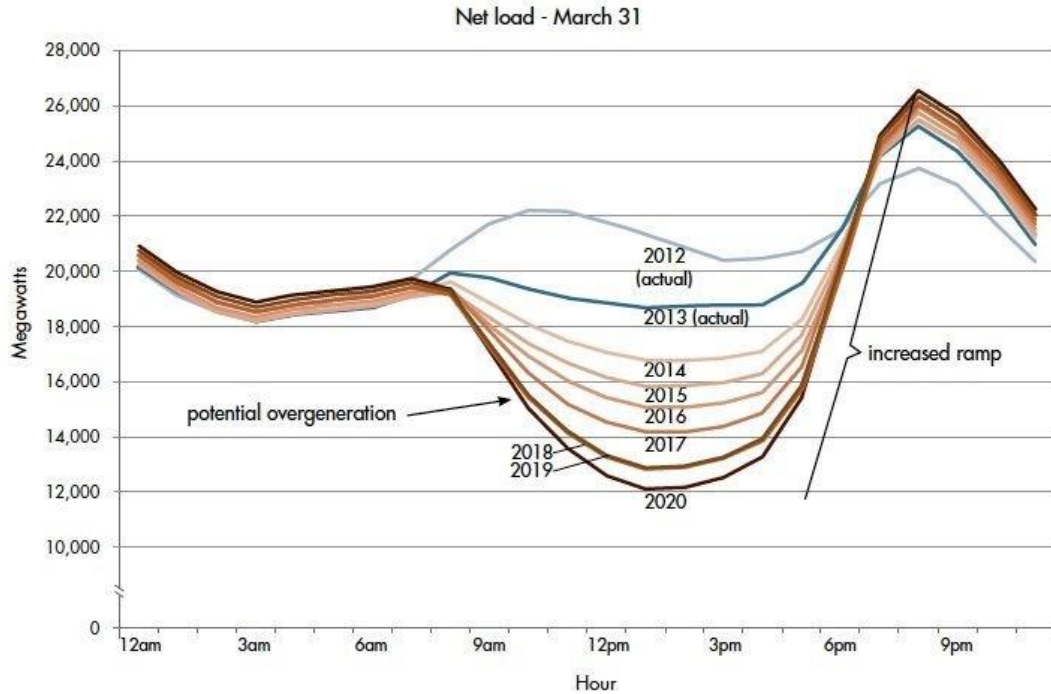


Figure 4. Plot of the typical energy demand curve from 2012-2020 during spring in California, (figure from California Independent System Operator (CAISO) [9], used without permission)

Possible solutions to the potential over-generation during the day and high peak in the evening are to diversify out of PV to be less dependent on the sun, to develop a better prediction of demand and generation, or to shift demand and generation to better align. Shifting the energy load from high-demand period to high-supply period can alleviate the wasted energy and unnecessary costs. Figure 5 illustrates an example of a demand curve and how the demand can be shifted before and after the peak to reduce the peak demand. By reducing the peak demand, the supply capacity required by utility companies can be lowered leading to lower costs and less wanted energy.

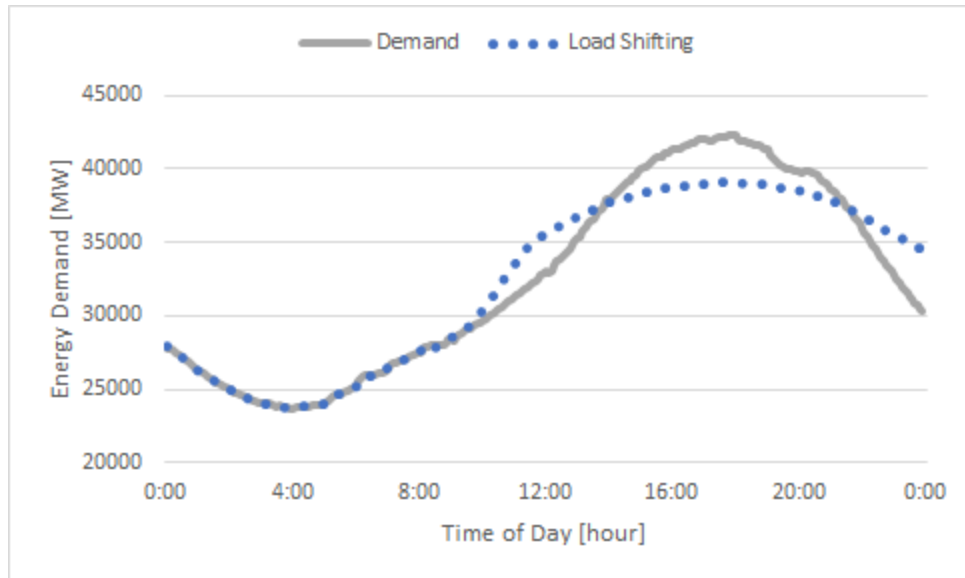


Figure 5. Example of a demand curve and load shifting before and after the peak demand.

A study conducted by the Pecan Street Research Institute found that orienting solar panels to face the west rather than the south resulted in 49 % more electricity during peak demand compared to south-facing panels, resulting in 9 % greater peak reduction [10]. While this provides some alleviation to the misalignment, the change in orientation does not aid in the uncertainty of PV generation. Another way to encourage load shifting is by storing energy either in thermal storage or in a battery. Due to the high cost and maintenance of a battery, it is not economically preferable to integrate a battery into a residential energy system [11]. Additionally the use of a battery still requires incentive for the consumer to utilize it. Thermal energy storage is not common in the residential setting since it usually requires a large installation area and high initial costs. Preheating or precooling a house can be considered short term thermal energy storage with the house acting as the storage unit itself. A study has been conducted using the house as thermal storage to show how doing so can help with load shifting without significantly

affecting thermal comfort [12]. In this way, the demand can be shifted if users preheat or precool their homes. Again, users need incentives to preheat or precool.

These solutions are part of transactive energy: transactive energy is a broad term to cover economic and control techniques used to manage the flow or exchange of energy within an existing power system [13]. Under transactive energy, anyone can trade electricity on the grid; traditional consumers with DER can sell any excess production of energy back into the grid to be used elsewhere. Transactive energy moves away from a centralized grid towards a decentralized network of automated nodes, redesigning the grid to be more efficient. An example of possible data and electricity flow under transactive energy is shown in Figure 6. Key players include conventional generation, renewable energy resources, the utility grid, storage, residential consumers, residential "prosumers" who both consume energy and produce energy through DER, and commercial consumers. Through gathering data from all key players using the transactive platform, the network operator is able to manage the flow of energy more efficiently.

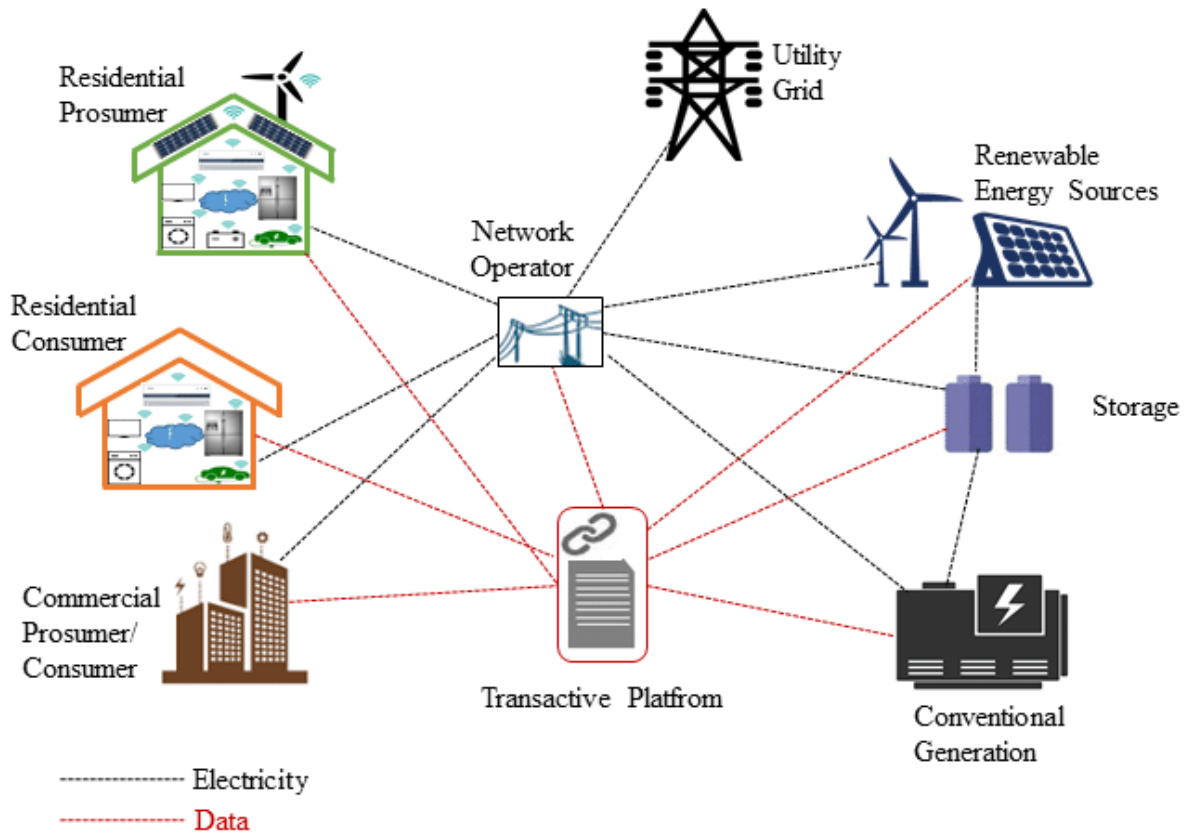


Figure 6. An example of possible data and electricity flow under transactive energy (figure taken from Zia [14], used without permission).

The current grid does not support the level of communication between the wholesale market and consumers required to bring the potential energy savings from transactive energy to fruition. The wholesale market must react to real-time demand and price fluctuations whereas traditionally, residential consumers purchase energy at a predetermined rate. Predetermined rates cannot accommodate these dynamics and lead to wasted energy and higher costs for both utility companies and consumers.

Charging a time-varying price of energy encourages load shifting. As energy infrastructure has evolved, utility companies have been able to change their pricing structure to motivate customers to alter their energy usage. Specifically, utility companies have implemented Time-Of-Use (TOU) pricing methods to have the energy cost to the consumer reflect the actual cost of energy at that time of day. One TOU pricing plan from Pacific Gas & Electric (PG&E), California's biggest utility company, is plotted along with the actual demand in California on a summer day according to California Independent System Operator (CAISO) in Figure 7. As shown in the figure, the price of energy on the TOU plan is increased during 16:00 - 21:00 which aligns with the period of high energy demand.

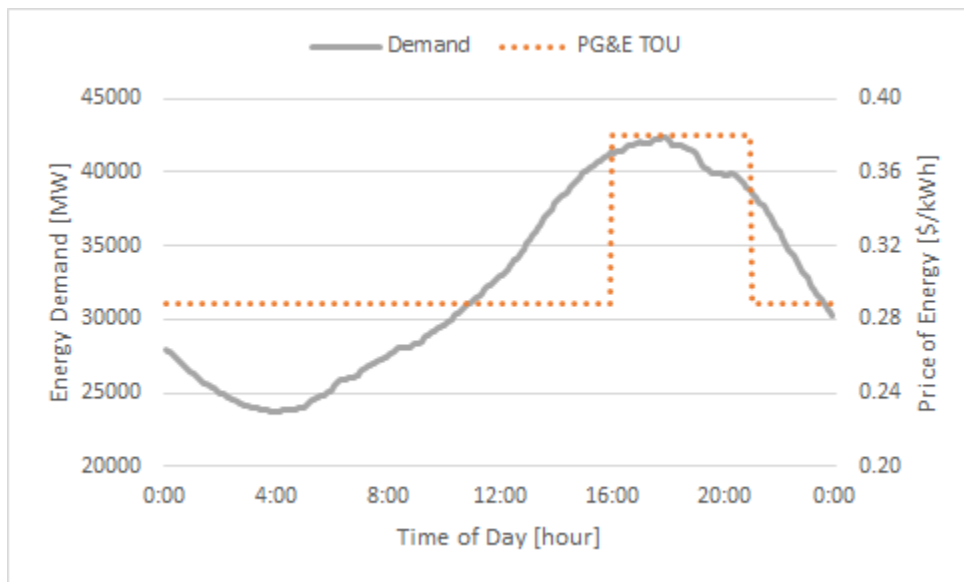


Figure 7. Plot of a Time of Use (TOU) pricing plan from the utility PG&E with the energy demand curve (data from CAISO [15]).

If residential customers can change their energy consumption in response to changes in price, TOU pricing will result in a shift in energy consumption. For now, TOU pricing is a step closer to representing the actual cost of energy than flat rate pricing, but is still an estimate of the appropriate price for that time period. As home energy management systems (HEMS) become

more complex with the ability to respond to the varying price of energy, the utility company can continue to alter its pricing strategy to better reflect the true cost of energy. Eventually, this could drastically reduce peak demand and wasted energy. However, TOU pricing alone is often not enough motivation for a customer to effectively alter their energy usage, and a recent study found no evidence that TOU can contribute to the peak load shaving or reduced energy consumption [16]. Therefore, there is a need for an energy management system that is able to respond to the dynamic utility rate. This thesis is investigating residential control systems to optimize usage based on a dynamic utility rate.

1.1. Existing Residential Energy Control Strategies

One strategy to enhance the sustainable practice of using energy in buildings is Model Predictive Control (MPC). MPC is a control strategy that uses a model to predict the future of a complex system and alters its control accordingly. Often, a cost function is used with MPC to optimize the control under a set of constraints. MPC has been implemented with building simulators to work with various cost functions including minimizing total energy consumption, deviation from comfort zone, peak load, and cost. MPC has been shown to significantly reduce peak load when given grid load levels as a factor to minimize [17]. When compared to ON/OFF and proportional-integral-derivative (PID) control, MPC has been shown to be the most effective control strategy in minimizing cost and thermal discomfort [18]. Studies have used MPC with TOU pricing and demand charges in order to minimize cost [19, 20]. Different price scenarios, namely a day-night tariff, a day-ahead dynamic tariff, and real-time dynamic tariff were implemented with MPC and showed how MPC is able to respond to changes in price [21].

One energy control strategy focused on reducing consumption is using an adaptive comfort zone. Adaptive comfort zone setpoint temperature control, referred to as adaptive control hereafter, is the practice of setting the setpoint temperature to be within a thermal comfort bound that changes based on the outdoor temperature. ASHRAE Standard 55 defines the thermal comfort zone for occupants in buildings based on environmental and personal factors. For the standard, a study was completed using predicted mean vote (PMV) as an index to determine the comfort zone a certain percentage of people would be given the conditions. The comfort zone based on the main environmental factor of outdoor temperature resulting from the study is shown in Figure 8. The study found that 90% of people would be comfortable in the area shaded in blue on the figure, and 80% of people would be comfortable in the area shaded yellow. An example of the traditional setpoint range is also plotted on the figure. Using adaptive control, the energy system saves energy by using a lower setpoint in cold weather and a higher setpoint in warm weather compared to a fixed setpoint. Adaptive control has been found to save up to 54% of energy consumption [22].

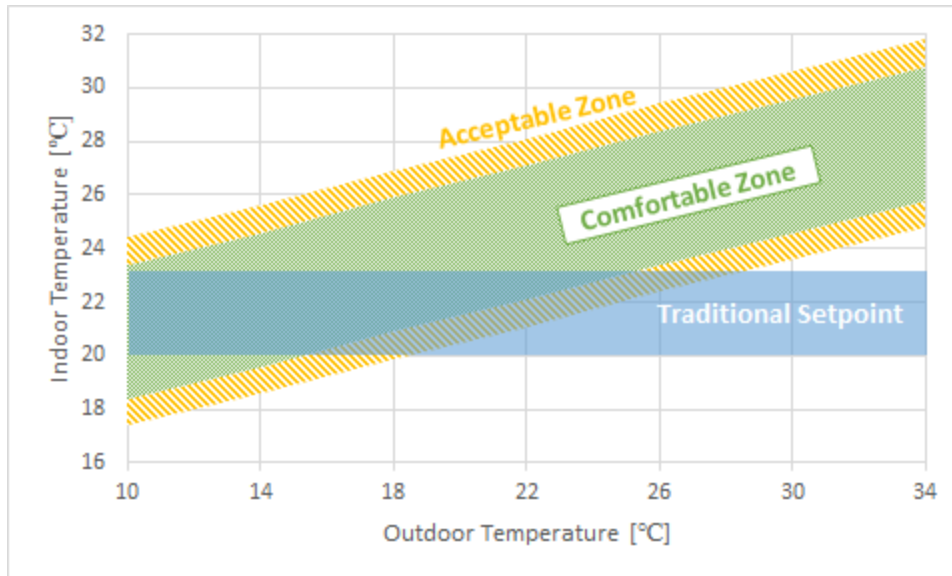


Figure 8. Adaptive comfort zone based on outdoor temperature compared to a traditional setpoint control. The comfortable zone (green grid area) and acceptable zone (yellow pattern area) depict the temperature ranges accepted by 90 % and 80 % of people accordingly.

Another energy control strategy to reduce energy consumption is using occupancy sensing to guide whether the HVAC system is on or off. When added to adaptive control, the occupancy information can add additional energy saving impact by 20% [23]. Another study showed that MPC was effective in prioritizing thermal comfort when implemented with occupancy sensing and weather forecasting [24]. Drawbacks to occupancy sensing include added cost of installation.

Further work can be done to create a cohesive model integrating various points of interest. An adaptable MPC model with the ability to add on additional control methods as more are improved, including adaptive comfort zone, occupancy sensing, and battery storage, can compare the effectiveness and efficiency of the control methods. Implemented with a dynamic pricing model, this MPC method could show the potential cost savings and load shifting.

1.2. Existing Pricing Models

As mentioned, dynamic pricing models have the potential to encourage load shifting, but are not common. A review of existing pricing models in the United States was completed. The United States contains eight climate zones with three moisture regimes as defined by the International Energy Conservation Code (IECC) [25]. Figure 9 shows the climate zones and moisture regimes of the United States.

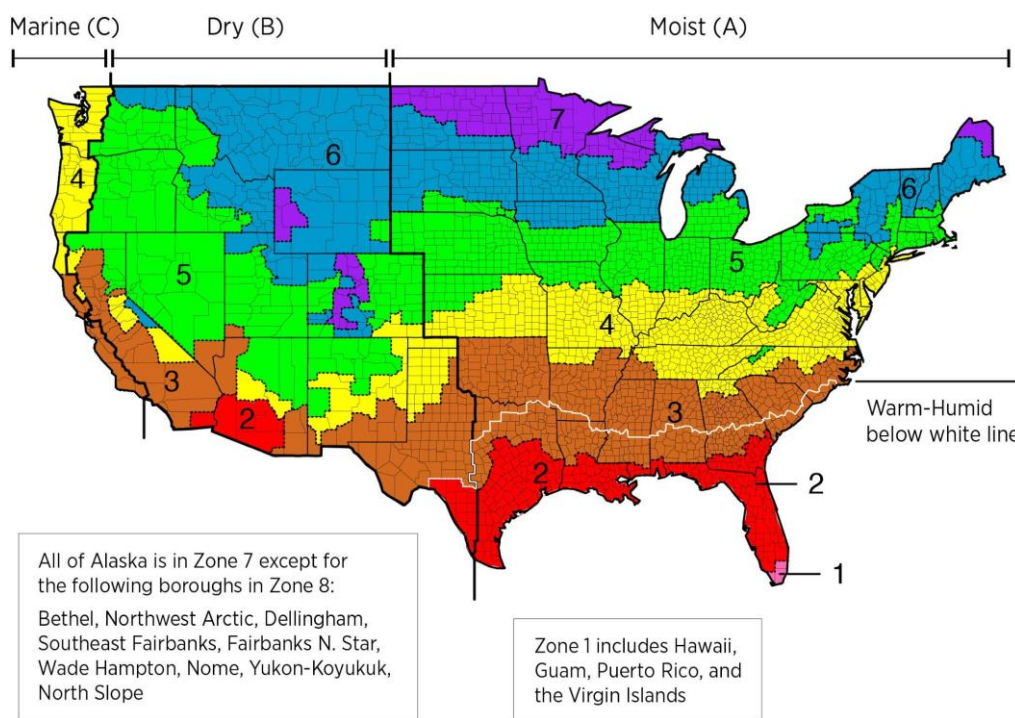


Figure 9. Climate zones and moisture regimes in the United States. (Figure from IECC [26], taken without permission)

For the review, five major cities as listed in Table 1 were selected to represent the range of climate zones. Major cities were more likely to have multiple pricing models while smaller cities tended to have fewer pricing models.

Table 1. List of five US cities and their climate zones for the pricing model review.

City	Climate Zone
Miami, FL	1A
Phoenix, AZ	2B
San Francisco, CA	3C
New York, NY	4A
Fairbanks, AK	8

A major utility provider in each city was chosen and each utility provider offered at least two different types of residential energy pricing plans except for the utility in Fairbanks, AK, which only offered one fixed rate plan. See Table H1 in Appendix H for a brief description of the residential energy pricing models offered by each utility company reviewed. The most common pricing plans are a fixed rate, tiered fixed rate, or time of use (TOU) plan. A fixed rate plan sells energy at one price regardless of total consumption of time of use. A tiered fixed rate plan offers energy at one price until a consumption limit is met, when future consumption is priced at a different rate. This type of plan often encourages less total consumption to prevent paying a higher price. A TOU plan prices energy at different rates based on the time the energy is consumed. TOU encourages less use during high price periods. The details of the plans such as consumption tiers and time periods of varying price are specific to the utility provider. None of the reviewed utility providers offered dynamic pricing or TOU with more than four different time periods. Additionally, all the pricing models posted the pricing for at least 6 month periods, and therefore require predictions and averaging over those periods. These pricing models are not set up to respond to demand in a way that takes advantage of the enhanced technology in the

grid. A dynamic pricing model as suggested creates the potential to shift the energy load when working with residential control methods that work to reduce cost.

1.3. Existing Validation Techniques

Existing studies have used various energy modeling methods to test their hypotheses. The studies on MPC, previously mentioned, have used existing software [14], developed theoretical models [13], or tested on an existing building [18]. Each method comes with its own limitations.

Theoretical models are based on theory and cannot take into account all of the factors in the real world. Testing on existing buildings provides the most realistic results, but requires a lot of time to implement any control system and to record data. This modeling is also limited in terms of repeatability and breadth of applicability since it is building specific. Using building modeling software can provide more accurate results than theoretical models and provide results more quickly than testing on existing buildings. Building modeling software can be altered for different locations and to take into account different factors, making it the best modeling method to validate the effectiveness of energy management techniques and dynamic price strategies.

EnergyPlus [27], the most widely adopted building simulator, has been used with an open-source co-simulation platform called the Universal Cyber-Physical Systems Environment for Federation (UCEF) [28], previously developed at the U.S. National Institute of Standards and Technology (NIST) to model residential buildings with various control strategies including adaptive comfort model and occupancy sensing [22, 23]. This methodology can be built upon to model MPC with dynamic pricing.

1.4. Statement of Goals

The purpose of this thesis is to develop a residential building energy management control system that optimizes energy consumption to minimize cost to the consumer. In order to complete this goal, a dynamic pricing model to be used with the control system and a validation tool to evaluate the effectiveness of paired control strategy and pricing model must be developed. This paper proposes a validation tool based on EnergyPlus, the most widely adopted building simulator, and an open-source co-simulation platform called the Universal Cyber-Physical Systems Environment for Federation (UCEF) [27, 28]. This validation method is scalable to model single homes or multiple homes to model a microgrid. Extensive documentation and building and weather files based on locations are available for EnergyPlus making this validation method widely applicable. Various control strategies can be used individually or in combination using UCEF. Additionally, the pricing model can be altered to test transactive energy theories. MPC can be used to optimize the thermostat setpoints and minimize energy cost for an individual residential house while maintaining thermal comfort of users. Ensuring thermal comfort makes it so this control strategy does not upset users. Minimizing cost rather than energy consumption takes into consideration the surplus of energy during off peak hours and the increased cost of energy during times of high demand. As the utility company's pricing method better correlates to the actual cost of energy, as the proposed dynamic pricing model does, minimizing residential energy costs will in turn minimize overall energy consumption. To test the residential energy control strategy and quantify results, simulations are run for two week long periods in winter in San Francisco.

1.5. Impact

On an individual level, the consumer using the presented residential energy control strategy will experience a small reduction in their energy bill while not sacrificing thermal comfort. At scale, a significant amount of cost will be saved to consumers. Additionally, utility companies will potentially experience cost and energy savings. If the pricing models of utility companies are structured to reflect the cost of energy generation and distribution, reducing the cost to the consumer will also reduce the operating cost of the utility company and may reduce peak consumption by load shifting. Utility companies will be able to influence energy consumption by altering their pricing model.

This simulation strategy is adaptable to incorporate additional considerations and will contribute to the development of utility dynamic pricing models and residential control strategies for grid-interactive buildings and homes. Utility pricing models and control strategies can be tested independently so that a wide range of options can be considered. The outcome of this research can be expanded to different building models or locations in future work.

The rest of the paper is organized as follows: Chapter 2 presents and explains the theory and application of an optimization model used as the residential energy control system. Chapter 3 presents a validation tool and explains its flexibility to continue to be used in the future. Chapter 4 explains the detailed simulation setup to test the optimization and validation tool. Chapter 5 presents analysis of the results of the simulations. Finally, Chapter 6 covers conclusions and future works.

2. Optimizer

The main component of the residential building energy management control system is the optimizer. The optimizer is the control strategy that optimizes the future residential energy consumption to minimize the total cost of energy. Other residential control strategies including occupancy sensing and adaptive control attempt to minimize total energy consumption given certain requirements, whereas the optimizer creates a control strategy in response to the future price of energy. The optimizer is participating in transactive energy by responding to the market, whereas the previously mentioned control strategies only respond to the individual system. To act in this way, the optimizer must predict the future indoor temperature given current conditions, the future environment, and future pricing.

The requirements for the optimizer are that:

1. Total cost is minimized.
2. Indoor temperature stays within a defined comfort zone.
3. The optimizer responds to a dynamic pricing model.
4. The optimizer sets the future setpoint temperature.
5. The optimizer must be integratable with building energy simulation software.

2.1. Indoor Temperature Prediction

Since the optimizer must be able to predict the future energy consumption, there was a need to predict the relationship between indoor temperature and energy consumption. Indoor temperature is a function of energy used, outdoor temperature, solar radiation, and building characteristics.

An approximate representation of the change in indoor temperature is given by:

$$\frac{dT_{indoor}}{dt} = \frac{UA}{mc}(T_{outdoor} - T_{indoor}) + \frac{1}{mc}Q + C_3Q_{solar} \quad (2.1)$$

where T_{indoor} is the indoor temperature, dt is the number of seconds in a timestep, U is the overall heat transfer coefficient, A is the surface area of the building envelope, m is the mass of the building, c is the specific heat of the building, Q is the amount of energy added to the space, $T_{outdoor}$ is the outdoor temperature, Q_{solar} is the solar radiation, and C_3 is a constant, which defines a relation between solar radiation and the energy gain in a house. Equation 2.1 can be rewritten to solve for the indoor temperature at the next timestep as:

$$T_{indoor}^{n+1} = dt \times (C_1 (T_{outdoor}^{n+1} - T_{indoor}^n) + C_2 E_{used}^{n+1} + C_3 Q_{solar}^{n+1}) + T_{indoor}^n \quad (2.2)$$

where superscript n represents the timestep number, dt is the number of seconds in a timestep, and C_1 , C_2 , and C_3 are unknown constants. The constants cannot be precisely found theoretically due to the intricacies of a building thermal model, inaccuracies between design and build, and inconsistencies in factors such as insulation. Rather, the constants can be found experimentally by performing linear regression [29] on data collected by a particular building model.

In order to estimate the constants, an annual EnergyPlus simulation was run and the indoor temperature, outdoor temperature, energy consumed, and solar radiation were recorded. Linear regression was performed using Matlab. Equation 2.2 was made into a matrix form that could then solve for the constants as shown in:

$$\left[\frac{T_{indoor}^{n+1} - T_{indoor}^n}{dt} \right]_{1 \times n} = [C_1 \quad C_2 \quad C_3]_{1 \times 3} \times \begin{bmatrix} T_{outdoor}^{n+1} - T_{indoor}^n \\ E_{used}^{n+1} \\ Q_{solar}^{n+1} \end{bmatrix}_{3 \times n} \quad (2.3)$$

The Matlab code can be found in Appendix A. The training data size for linear regression ranged from 1,000 to 100,000 and consisted of randomly selected points from the EnergyPlus dataset. To check the accuracy of the indoor temperature prediction model, the Root Mean Square Error (RMSE) [30] for each training set was calculated and is plotted in Figure 10. As shown in the

figure, for training sets larger than 20,000, the RMSE was around 0.088 °C which represents when the training set is large enough that increasing the training set size is not likely to improve results.

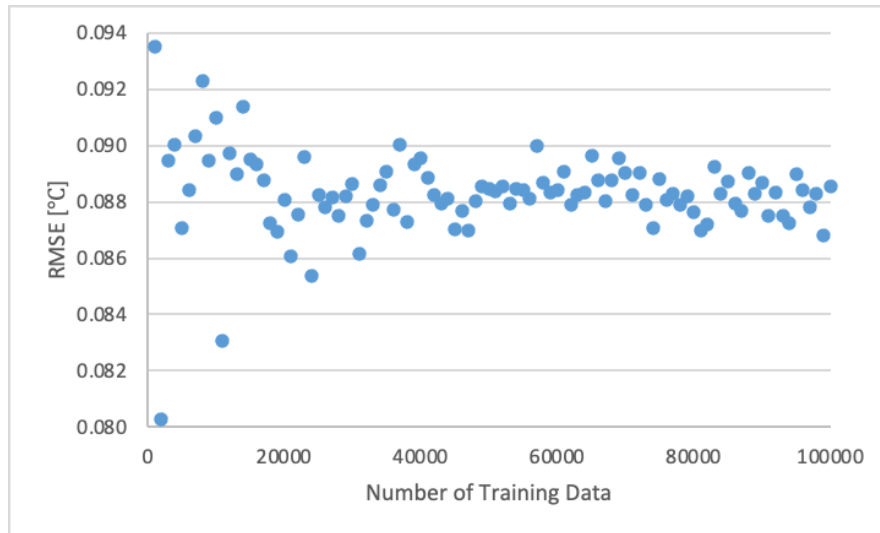


Figure 10. RMSE of the indoor temperature prediction in °C by size of training set.

From these results, the constants for the particular building model were found as shown in Table 2.

Table 2. The values of the unknown constants in the indoor temperature prediction equation, solved for by linear regression.

Constant	Value
C_1 [1/s]	2.66×10^{-5}
C_2 [°C/(kWh s)]	7.20×10^{-3}
C_3 [°C/(J s)]	3.10×10^{-7}

This relation was intended to be used to predict the relationship between indoor temperature and energy consumption for a set time period into the future given the current indoor temperature and the future outdoor temperature and solar radiation. To test the accuracy of using the prediction equation several timesteps into the future, the predicted indoor temperature and energy consumption was calculated for four hours (48 timesteps) into the future given the current indoor temperature. The results were compared to EnergyPlus data. The predicted and actual energy consumption for one four hour period is shown in Figure 11. The error between prediction and actual data propagates, causing the prediction to become less accurate as it predicts farther from the initial timestep. Overall, the average RMSE between the actual and predicted energy consumption per hour was 0.00615 kWh, which is one-twenty-fifth of the average energy consumption per hour, 0.155 kWh.

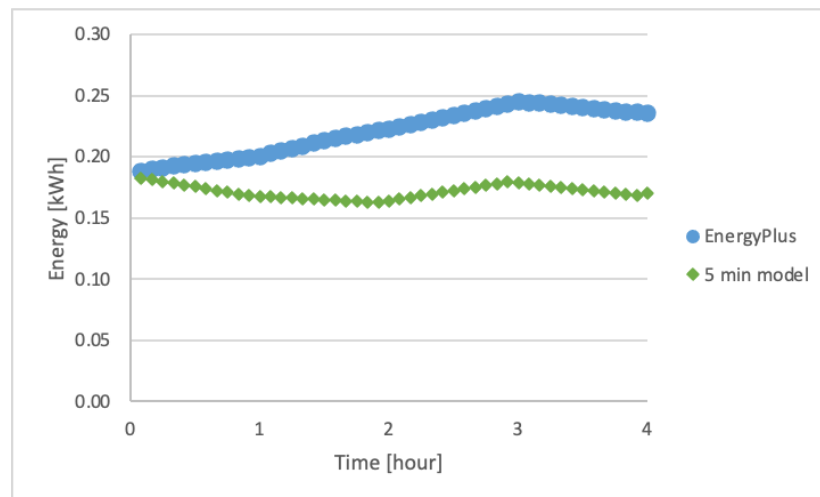


Figure 11. Comparison of energy usage predicted by the prediction method and the EnergyPlus data.

The actual indoor temperature and prediction for 4 hours is shown in Figure 12. Again, the error propagates as the prediction is farther from the ground truth and results in a 0.75 °C difference

after 4 hours. If the prediction is repeated every hour as shown in Figure 12, the prediction remains within 0.3 °C from the actual temperature.

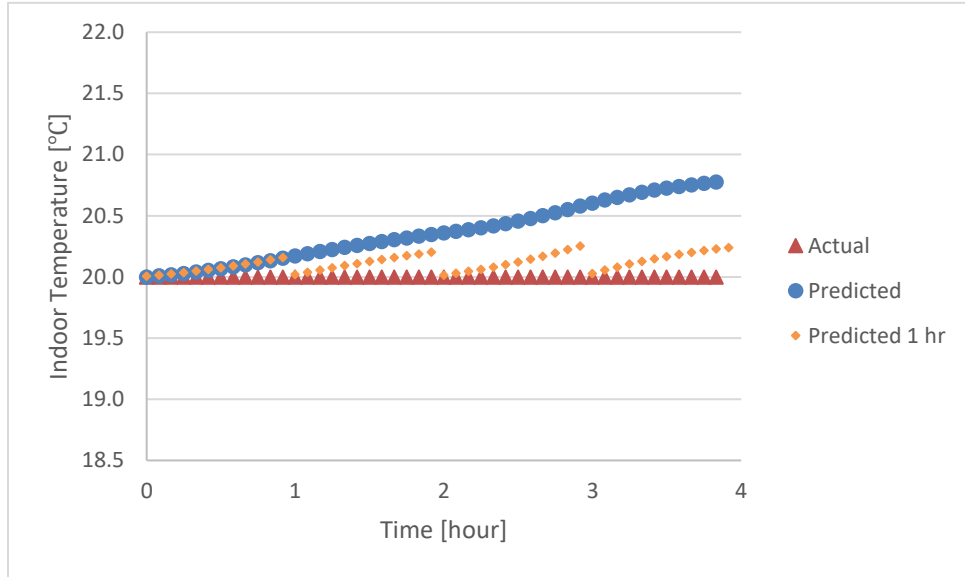


Figure 12. Comparison of indoor temperature predicted by the prediction method and the EnergyPlus data.

The trends in the error were further examined by averaging the percent difference between the actual and predicted indoor temperature for each hour of the day and for each month. The average percent difference between the actual and predicted indoor temperature for each hour is shown in Figure 13. The average percent difference remained below 0.70% for all hours of the day, signifying no significant difference.

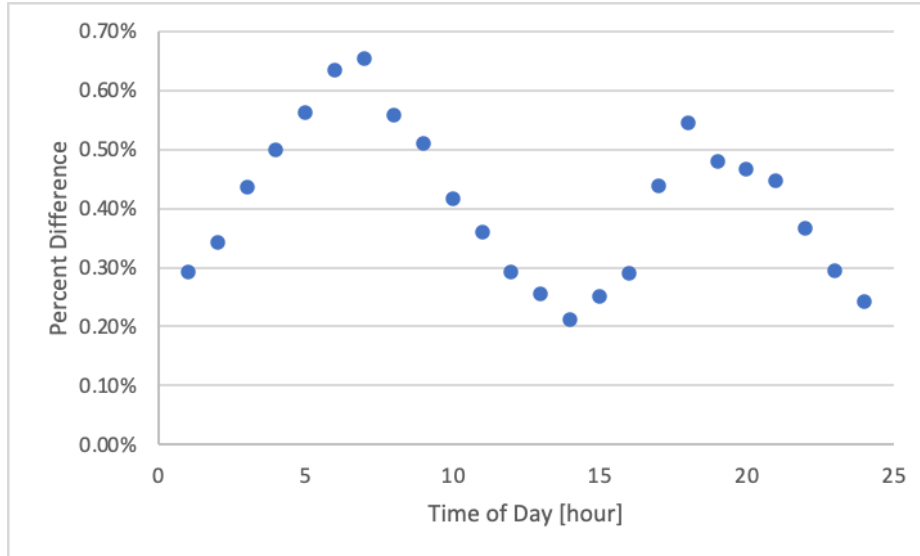


Figure 13. Average percent difference between actual and predicted indoor temperature for each hour of the day.

The average percent difference between the actual and predicted indoor temperature for each month is shown in Figure 14. The average percent difference remained below 0.70% and did not show any significant trends in the error. With no major trends in error found and despite the error propagation, the prediction is accurate enough for the next hour to be effective for optimization. A two hour time period was chosen for the prediction window to be used in the future.

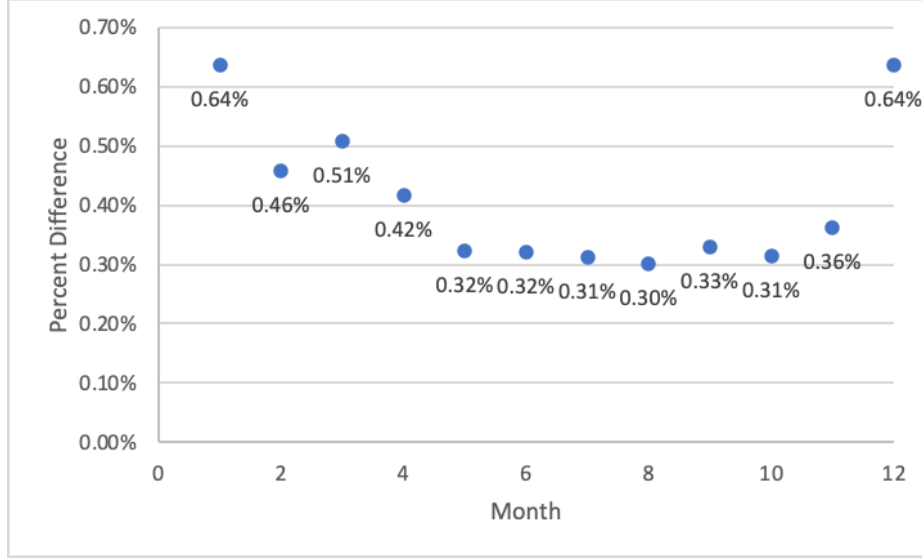


Figure 14. Average percent difference between actual and predicted indoor temperature for each month.

2.2. Optimization

With a relationship between energy consumption and indoor temperature found, optimization could be done. For optimization, an open source software package for convex optimization based on the Python programming language, CVXOPT [31], was used. In this case, the goal is to find the values of energy usage that will minimize the following cost function:

$$CF = \sum_{n=1}^N P^n \times E_{used}^n \quad (2.4)$$

where P^n is the cost of electricity at each timestep and E_{used}^n is the energy used at each timestep.

A set of constraints were used with the objective equation to limit the energy consumption to be positive or negative for heating and cooling respectively and for the indoor temperature at each timestep to be within the defined comfort zone as shown in:

$$E_{used}^n \geq 0 \quad (\text{for heating}) \quad (2.5)$$

$$E_{used}^n \leq 0 \quad (\text{for cooling}) \quad (2.6)$$

$$|E_{used}^n| \leq E_{max} \quad (2.7)$$

$$T_{indoor}^n \geq T_{comfort,lower}^n \quad (2.8)$$

$$T_{indoor}^n \leq T_{comfort,upper}^n \quad (2.9)$$

where $T_{comfort,upper}^n$ and $T_{comfort,lower}^n$ are the upper and lower bound of the indoor temperature comfort zone at each timestep and E_{max} is the maximum energy able to be used in a single timestep. The maximum energy consumption constraint, Equation 2.7, ensures the optimization prediction is realistic given the size of the HVAC equipment and can vary for cooling and heating. The upper and lower bound of the indoor temperature comfort zone were determined from the adaptive comfort model shown previously in Figure 8. This set of constraints is adaptable and can incorporate different energy control strategies including adaptive setpoint control as mentioned and occupancy control.

For implementation with CVXOPT, all constraint equations must be in the form $aE_{used}^n \leq b$, where a and b are constants. Equations 2.8 and 2.9 require the indoor temperature prediction relation, Equation 2.2, to be put in terms of E_{used}^n . Plugging in Equation 2.2, the indoor temperature constraints become:

$$dt \times (C_1(T_{outdoor}^n - T_{indoor}^{n-1}) + C_2 E_{used}^n + C_3 Q_{solar}^n) + T_{indoor}^{n-1} \geq T_{comfort,lower}^n \quad (2.10)$$

$$dt \times (C_1(T_{outdoor}^n - T_{indoor}^{n-1}) + C_2 E_{used}^n + C_3 Q_{solar}^n) + T_{indoor}^{n-1} \leq T_{comfort,upper}^n \quad (2.11)$$

These constraints could be used to optimize the energy consumption one timestep ahead since all variables besides energy consumption are known. However, more than one timestep into the

future, the variables T_{indoor}^{n-1} which represent the indoor temperature at the previous timestep are unknown since they are functions of the energy used at previous timesteps. Therefore, indoor temperature prediction, Equation 2.2, must be used again. Several timesteps into the future, a pattern appears and the constraints take the form

$$-D_m^n \times E_{used}^n \geq -T_{comfort,lower}^n + S^n \quad (2.12)$$

$$D_m^n \times E_{used}^n \leq T_{comfort,upper}^n - S^n \quad (2.13)$$

where

$$D_m^n = dt C_2 \quad (\text{for } n=m) \quad (2.14)$$

$$D_m^n = 0 \quad (\text{for } n < m) \quad (2.15)$$

$$D_m^n = D_m^{n-1}(-dt C_1) + D_m^{n-1} \quad (\text{for } n > m) \quad (2.16)$$

$$S^n = dt \times (C_1(T_{outdoor}^n - S^{n-1}) + C_3 Q_{solar}^n) + S^{n-1} \quad (\text{for } n > 1) \quad (2.17)$$

$$S^1 = dt \times (C_1(T_{outdoor}^1 - T_{indoor}^0) + C_3 Q_{solar}^1) + T_{indoor}^0 \quad (\text{for } n=1) \quad (2.18)$$

As an example of how these equations look in a matrix form, the D and E_{used} matrices on the left hand side of the equation are:

$$D = \begin{bmatrix} dt C_2 & 0 & 0 & 0 \\ dt C_2 - dt^2 C_2 C_1 & dt C_2 & 0 & 0 \\ dt C_2 - 2dt^2 C_2 C_1 + dt^3 C_2 C_1^2 & dt C_2 - dt^2 C_2 C_1 & dt C_2 & 0 \\ dt C_2 - 3 dt^2 C_2 C_1 + 3dt^3 C_2 C_1^2 - dt^4 C_2 C_1^3 & dt C_2 - 2dt^2 C_2 C_1 + dt^3 C_2 C_1^2 & dt C_2 - dt^2 C_2 C_1 & dt C_2 \end{bmatrix}$$

$$E_{used} = \begin{bmatrix} E_{used}^1 \\ E_{used}^2 \\ E_{used}^3 \\ E_{used}^4 \end{bmatrix}$$

And the right hand side of the equation is:

$$\begin{bmatrix} T_{comfort,upper}^1 - (dt \times (C_1(T_{outdoor}^1 - T_{indoor}^0) + C_3 Q_{solar}^1) + T_{indoor}^0) \\ T_{comfort,upper}^2 - (dt \times (C_1(T_{outdoor}^n - S^1) + C_3 Q_{solar}^n) + S^1) \\ T_{comfort,upper}^3 - S^3 \\ T_{comfort,upper}^4 - S^4 \end{bmatrix}$$

With the inputs of the current indoor temperature, the future outdoor temperature and solar radiation, and the price of energy, the energy consumption can be optimized to minimize the total cost of energy while keeping the indoor temperature within the comfort zone.

Excel Solver is a Microsoft Excel add-on program that can validate CVXOPT results. Solver finds the optimal (maximum or minimum) value of a cell subject to limits on the values of other formula cells on a worksheet [32]. Specifically, Solver uses Generalized Reduced Gradient (GRG) Nonlinear Solving method to iteratively solve for the optimal solution. Solver can be used to solve for the optimal energy usage to minimize the total cost of energy as shown in Table 3. In this table, the outdoor temperature, solar radiation, and price of energy at all timesteps are known. The indoor temperature at the first timestep is known. The cost column is the product of the price of energy and amount of energy consumption at each timestep. The energy usage column is left blank for now, since it is the independent variable. The indoor temperature after the first timestep use Equation 2.2 to calculate the value. The sum of the cost at each timestep is calculated in cell F(N+2). This total cost is the objective cell which Solver will minimize by changing the contents of column D. In the Solver setup panel, the constraints that the indoor temperature column must be between the upper and lower comfort bounds are entered.

Table 3. Excel Solver setup to validate early results from CVXOPT.

	A	B	C	D	E	F
1	Outdoor Temperature [°C]	Solar Radiation [J]	Price of Energy [\$/kWh]	Energy Used [kWh]	Indoor Temperature [°C]	Cost [\$]
2	9.40	0	0.20		20.0	= C2 × D2
3	9.35	0	0.20		= Equation 2.2	= C3 × D3
...
N+2						= sum(F2:F(N + 1))

Solver is able to minimize the cost and validate the results of using CVXOPT with the same dataset. The full results of the Solver and CVXOPT comparison can be found in Appendix I. Figure 15 shows the optimal energy consumption according to the Excel Solver and CVXOPT tests and the price of energy at each timestep. The Solver and CVXOPT results match exactly. A high amount of energy is consumed before the price increases and the energy consumption remains low during the price increase.

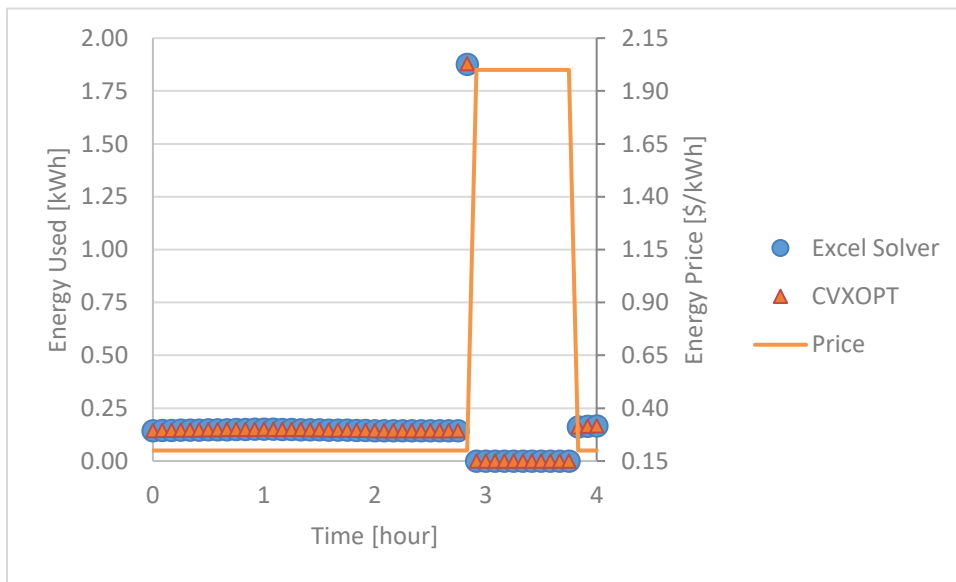


Figure 15. Optimal energy consumption result from Excel Solver and CVXOPT test cases.

The indoor temperature according to Solver and CVXOPT is shown in Figure 16 along with the price of energy at each timestep. Again, the Solver and CVXOPT results match exactly. The indoor temperature remains at the lower comfort bound of 20.0 °C used in this case and then rises just before the price increases as a result of the high energy consumption. The indoor temperature naturally decreases back to the lower comfort bound during the time of high price.

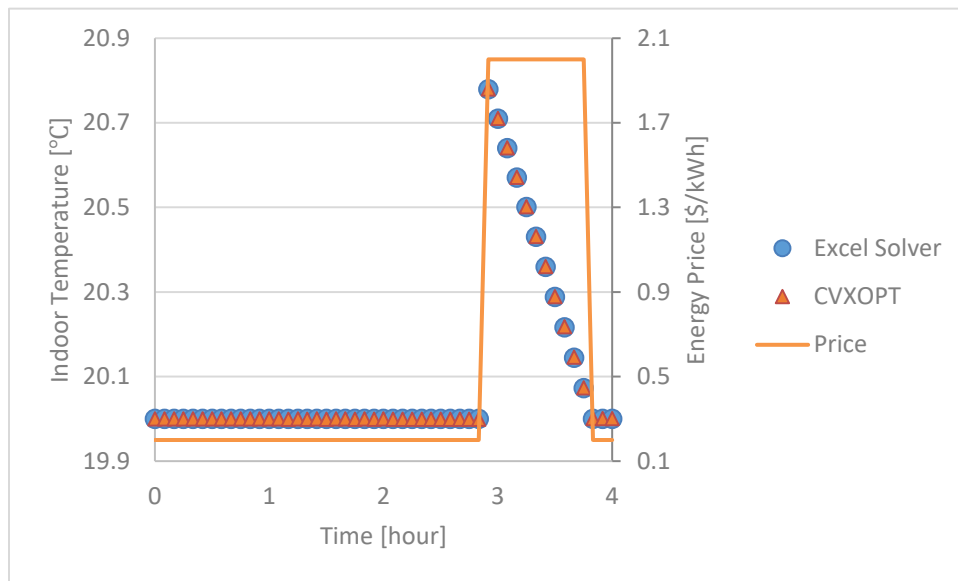


Figure 16. Indoor temperature result from Excel Solver and CVXOPT test cases.

Excel Solver validated the results of CVXOPT. The runtime for the optimization test case for Excel Solver was 3.10 seconds and for CVXOPT was 0.0108. CVXOPT is easier to use repeatedly and is more applicable since it works with Python.

2.3. Setpoint Temperature Schedule

Since the setting that is able to be changed routinely to operate the HVAC system is the setpoint temperature in a building energy simulator and in reality, the optimizer must be able to set the

setpoint temperature schedule for the future. The predicted indoor temperature can be determined based on the CVXOPT optimization output of the optimized energy consumption using Equation 2.2. The predicted indoor temperature can be used as the setpoint temperature for future timesteps.

In order to more accurately model energy consumption in a residential building, "fuzzy control" was implemented to the setpoint temperature schedule. Fuzzy controls models the periodic on/off schedule of an HVAC system. An HVAC system allows the indoor temperature to fluctuate between 1 °C and the setpoint temperature. An example of fuzzy control implementation is depicted in Figure 17. The setpoint temperature is constant at 20.0 °C. A theoretical model without fuzzy control would result in a constant indoor temperature at 20.0 °C. However, in reality, the HVAC system would turn on and off, allowing the temperature to oscillate between 20 and 21 °C. Adding fuzzy control more accurately models energy consumption in a residential building.

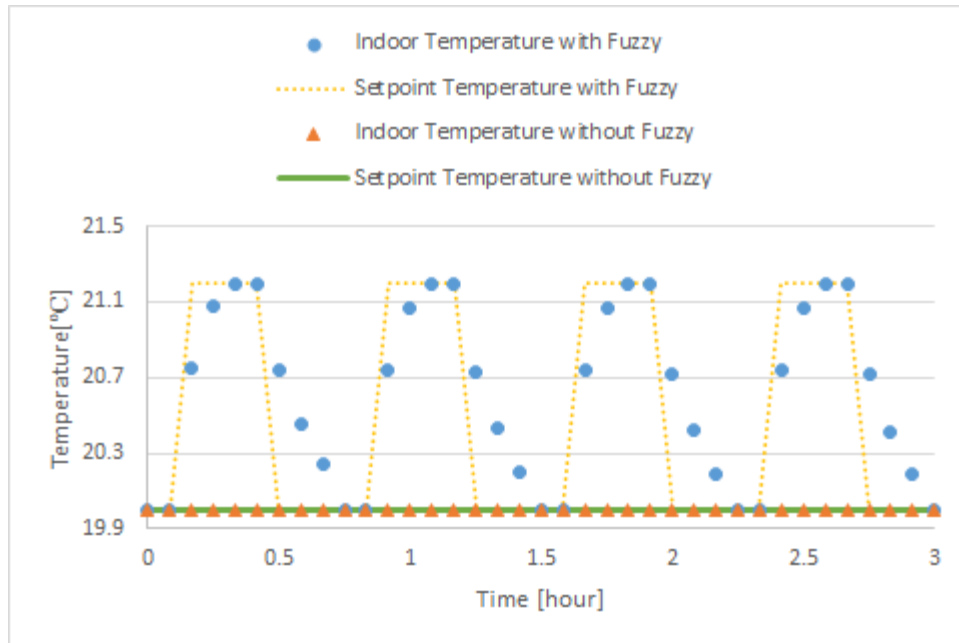


Figure 17. Comparison of modeled indoor temperature with and without fuzzy control.

2.4. Pricing Model

The pricing model is the factor that drives the optimization process and creates the potential to shift the energy load by saving cost. As previously mentioned, the pricing model is defined by utility companies and the two most common models are flat-rate and time-of-use (TOU). With the enhanced technology within the energy grid, a dynamic pricing model is able to take advantage of transactive energy principles. Pricing models under transactive energy can be adjusted in various ways. To show the potential benefits of changing the pricing model, two requirements were set:

1. The pricing model is based on the real-time demand of the market.
2. The difference in cost between peak and non-peak timesteps must be substantial enough for there to be economic benefit to load shifting

To create a pricing model based on the real-time demand of the market, the day-ahead market price of energy for Pacific Gas and Electric Company (PG&E), California's biggest utility company, was taken from California Independent System Operator (CAISO) [33], which oversees the operation of California's bulk electric power system, transmission lines, and electricity market generated and transmitted by its member utilities. The day-ahead market price is recorded hourly by CAISO.

The day-ahead hourly market price was recorded for one week period in January and February: January 1 - 7, 2021 and February 12 - 18, 2021 as shown in Figure 18. As seen in the figure, the wholesale price during the week in January acted periodically with a range of \$50/MWh between its peak and trough. However, the wholesale price for the week in February grew dramatically to a peak of \$953/MWh. The week in January may represent an average week whereas the week in February represents a period of significant demand changes, causing the wholesale price to rise.

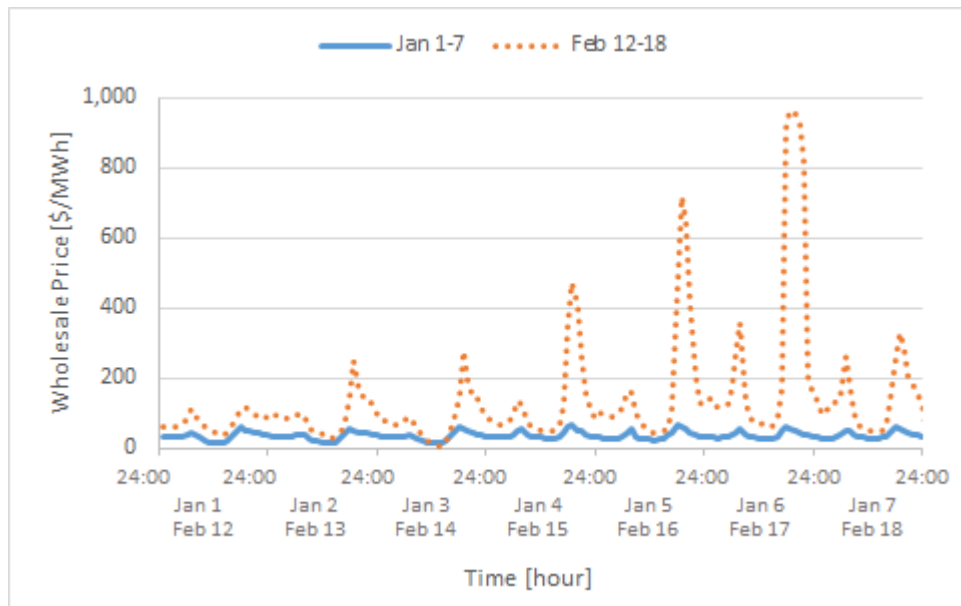


Figure 18. Wholesale market price of energy according to CAISO for January 1 to 7, 2021 and February 12 to 18, 2021.

A linear relationship between the market price and retail price of energy was created by comparing PG&E's range of electricity cost [34] with the range of market price. For dynamic pricing, the difference in pricing based on demand must be significant enough to feel like a penalty to use energy during high demand so that the energy management system shifts usage appropriately. The linear relationship follows the equation:

$$\text{Retail Price} = \alpha \times \text{Wholesale Price} + \beta \quad (2.19)$$

where α is the proportional constant of utility rate and β represents a delivery charge per kWh. For these simulations, the constants $\alpha = 4$ and $\beta = \$0.10$ were used which results in a retail price range of \$0.15 to \$0.38 for the one week period in January and a range of \$0.11 to \$3.94 for the one week period in February as shown in Figure 19. The pricing model for the week in February was also used for the week in September for testing.

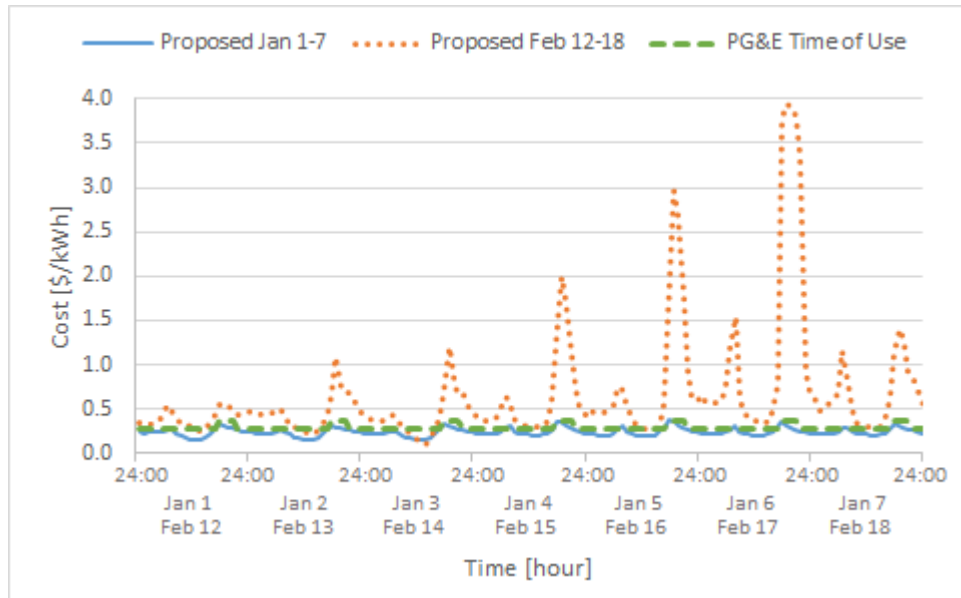


Figure 19. Proposed dynamic pricing models for January 1 to 7, 2021 and February 11 to 18, 2021 compared to PG&E TOU plan.

The pricing structure can be changed and tested in the future. Since the pricing model is the main driver of load shifting, extensive work can be done in the future to evaluate the effectiveness of various pricing strategies.

3. Validation Tool

A validation technique is needed to evaluate residential control strategies and pricing models under transactive energy. Many aspects of the grid under transactive energy, including pricing model and energy control strategy can be rethought. In order to effectively evaluate the results of such changes, a benchmark for analysis is needed. The validation tool must be widely accepted by experts in the field for the results to be upheld. The tool must be able to model different housing types in different locations with varying weather conditions. Consistency and accuracy is vital to ensure the reliability of the results. In summary, the requirements of the validation tool are that:

1. Residential control strategies and pricing models are able to be implemented with an energy building simulator.
2. The validation tool allows for factors including the residential control strategy and pricing model to be changed independently.
3. The validation tool is scalable and adaptable to include multiple housing models or different housing models and locations.

As aforementioned, NIST developed the co-simulation platform, UCEF, which is able to work with EnergyPlus to interact with a simulation. Since EnergyPlus is the most widely adopted building simulator with an extensive library of building models and weather files that can continue to be updated, it fits the requirements for use as a part of the validation tool. EnergyPlus can model the energy consumption which incorporates the physical building parameters and environment condition such as floorplan, material, HVAC system setup, and building location. The U.S. Department of Energy [35] offers EnergyPlus building models for the major climate

zones in the U.S. with different heating and foundation types. EnergyPlus uses typical meteorological year (TMY3) weather files, which are widely available, as an input, allowing the weather data to be altered for different locations.

UCEF is highly adaptable and is a strategic partner for use with EnergyPlus. Developed by the National Institute of Standards and Technology (NIST), Universal CPS Environment for Federation (UCEF) is an open-source Cyber-Physical System (CPS) experiment and testing environment. UCEF integrates multiple software through federates coded in different operating systems and development environments, which makes co-simulation with different software possible and straightforward [36]. UCEF uses IEEE's High Level Architecture (HLA) standard to synchronize time and data transfer among a group of federates [37]. The federates within UCEF use Java as the programming language. CVXOPT can be used by calling a python script from Java.

EnergyPlus has a preset co-simulation interface through the Functional Mock-up Interface (FMI) standard created by Modelisar [38]. The standard connects EnergyPlus simulation platforms to an external model by using a Functional Mock-up Unit (FMU). Previous work has been done to connect the FMU with UCEF via TCP/IP [39]. For each timestep, EnergyPlus sends simulation data to UCEF, UCEF performs some work, and UCEF sends an update to EnergyPlus for how to proceed with the simulation.

The basic method to model transactive energy with EnergyPlus and UCEF is shown in Figure 20. Information is passed between EnergyPlus and the UCEF "controller" through the FMU. The UCEF controller is where the optimization occurs. The controller receives information, specifically the indoor temperature, from EnergyPlus, reads the weather forecast, specifically

outdoor temperature and solar radiation, and reads the pricing model. The market and controller can exchange information such as forecasted pricing and energy predicted energy consumption in the future. The controller then optimizes the future energy usage to minimize the consumer cost, determines the future setpoint temperature, and sends the setpoint temperature schedule to EnergyPlus to continue the simulation.

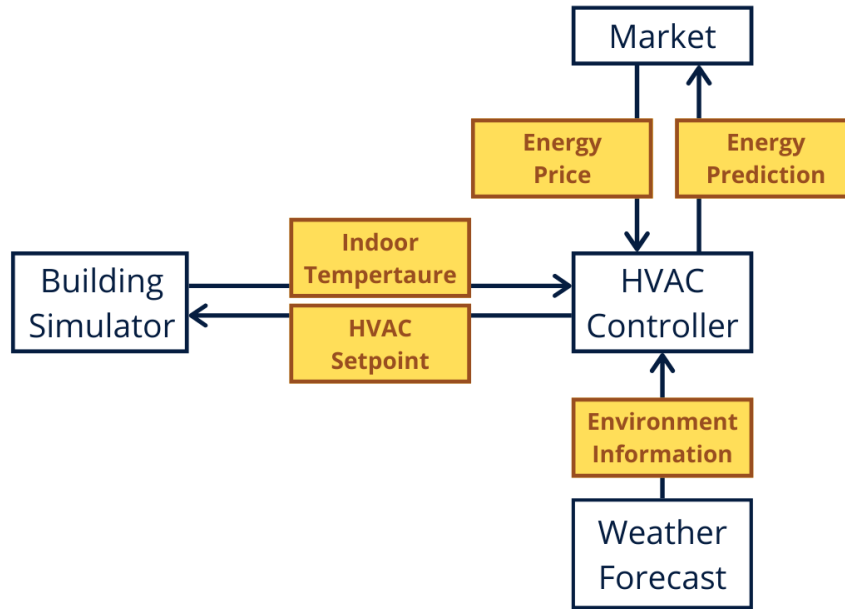


Figure 20. Communication exchange between EnergyPlus, controller, market, and weather forecast.

The sequence of information exchange is shown in Figure 21. This sequence can be expanded to include other transactive energy concepts. The dashed arrows in the figure represent the optional back-and-forth exchange of market price and predicted energy consumption that may lead to finding an equilibrium point between supply and demand at that point in time. This method is scalable to include multiple EnergyPlus simulations simultaneously and allows for the residential energy control system and pricing model to be altered independently.

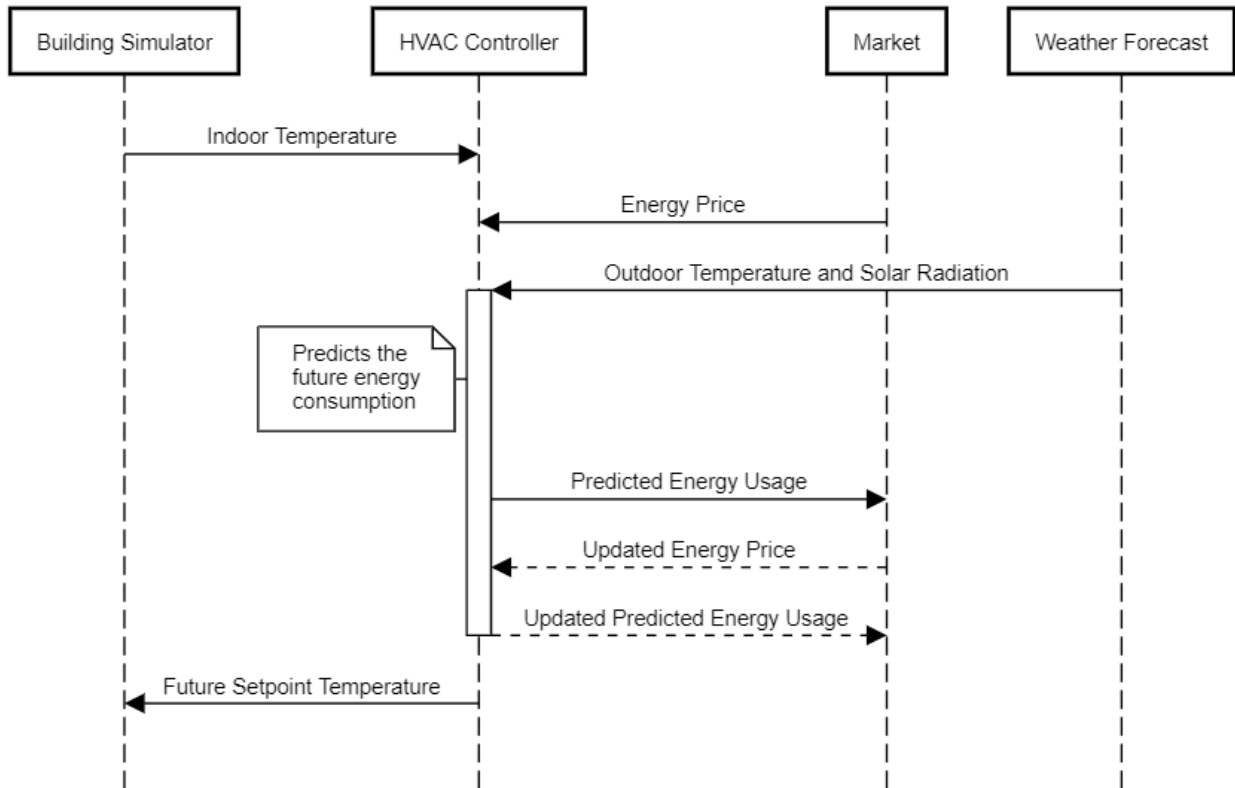


Figure 21. Sequence diagram of simulation with EnergyPlus and UCEF.

4. Simulation Setup

The input to EnergyPlus was a building model and weather file taken from the U.S. Department of Energy [35]. The residential prototype building model was for an approximately 2400 square foot, two-story, single family home in San Francisco with a heat pump and a crawlspace foundation type. The typical meteorological year (TMY3) weather file for Climate Zone 3C [40], where San Francisco is located, was used. The residential building model was modified to send and receive information with a Functional Mockup Unit (FMU) external interface [41]. The modifications can be viewed in the building model text file in Appendix E. The transfer of information is shown in Figure 22. EnergyPlus sends the current indoor temperature for the current timestep to UCEF, which then sends the current indoor temperature to the controller. The controller uses the indoor temperature and the weather information, i.e. the outdoor temperature and solar radiation which it reads from a weather file, to perform optimization for the next two hours and set the setpoint temperatures for the next hour at 5 minute timesteps. The 2 hour ahead prediction for optimization is longer than the timeframe for which the controller sets the setpoint temperature so that the optimization is taking into account future trends. Each hour, the process is repeated. The setpoint information is sent back to EnergyPlus. The python optimization code can be found in Appendix B. The code for the java federates used in UCEF can be found in Appendix C & D.

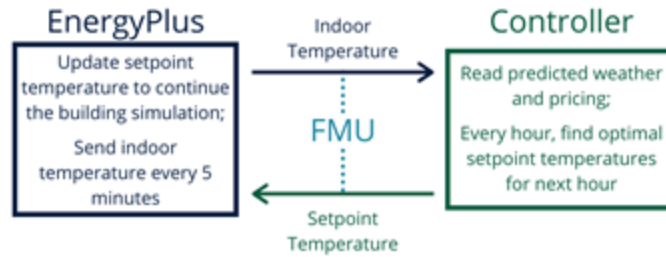


Figure 22. Diagram of the exchange of information between EnergyPlus and the co-simulation platform to find the optimal setpoint temperature.

Three sets of simulations were run for one week periods: January 1-7, February 12-18, and September 27-October 3. The week in January was chosen to represent a random week in winter months. The week in February tests optimization when there is a significant change in cost throughout the day. The week in September was chosen to test the optimization in summer months. September 27-October 3 required the highest amount of cooling energy out of any week in summer months with the building model. The following four energy control strategies were implemented for each time period:

1. Baseline: fixed heating setpoint without optimization
2. Optimization with a Fixed Comfort Zone
3. Adaptive Control without Optimization
4. Adaptive Control with Optimization

The baseline control strategy represents the most common user behavior of setting a fixed setpoint. Optimization with a fixed comfort zone represents a fixed setpoint with preheating or precooling. Adaptive control has been shown to provide significant energy savings while maintaining thermal comfort. The adaptive control without optimization strategy sets the setpoint

temperature as the lower bound of the adaptive comfort model (Figure 8) for heating and the upper bound of the adaptive comfort model for cooling. The last control strategy examines the effects of adding optimization to adaptive control.

5. Results

The total cost for the consumer was compared between the four simulations for the three time periods as shown in Table 4. January 1-7 resulted in the most savings between simulations with and without optimization whereas September 27-October 3 resulted in the least savings. All three time periods found savings of over \$2 when comparing optimization with an adaptive comfort zone to a fixed setpoint.

Table 4. Summary of cost savings between simulations with optimization and the corresponding simulations without optimization.

Simulation Period	Savings from Optimization with Comfort Zone Compared to a Fixed Setpoint	Savings from Optimization with Adaptive Comfort Zone Compared to Adaptive Comfort Alone	Savings from Optimization with Adaptive Comfort Zone Compared to a Fixed Setpoint
January 1-7	\$5.65	\$3.89	\$7.16
February 12-18	\$0.92	\$0.05	\$2.87
September 27-October 3	\$0.03	\$0.19	\$5.20

The total cost and energy usage between the four simulations are shown in Figure 23 for January 1-7. Adding optimization resulted in less energy consumption and less cost for both sets of simulations.

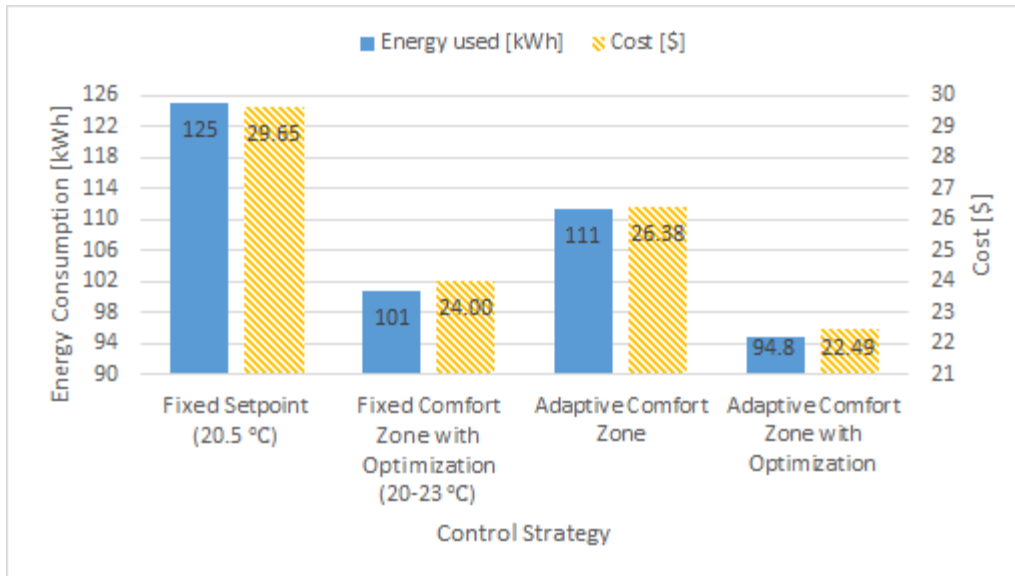


Figure 23. Energy consumption and cost for each energy control strategy for January 1 - 7.

The total cost and energy usage between the four simulations are shown in Figure 24 for February 12-18. When compared to a fixed setpoint temperature, optimization provided a reduction in both cost and energy consumption. However, when compared to an adaptive comfort zone, optimization reduced the cost by \$0.05. The week in January saw more significant savings than the week in February since the weather was colder, resulting in 3-4 times higher energy consumption. Adaptive control with optimization saved around \$3 compared to the fixed setpoint behavior.

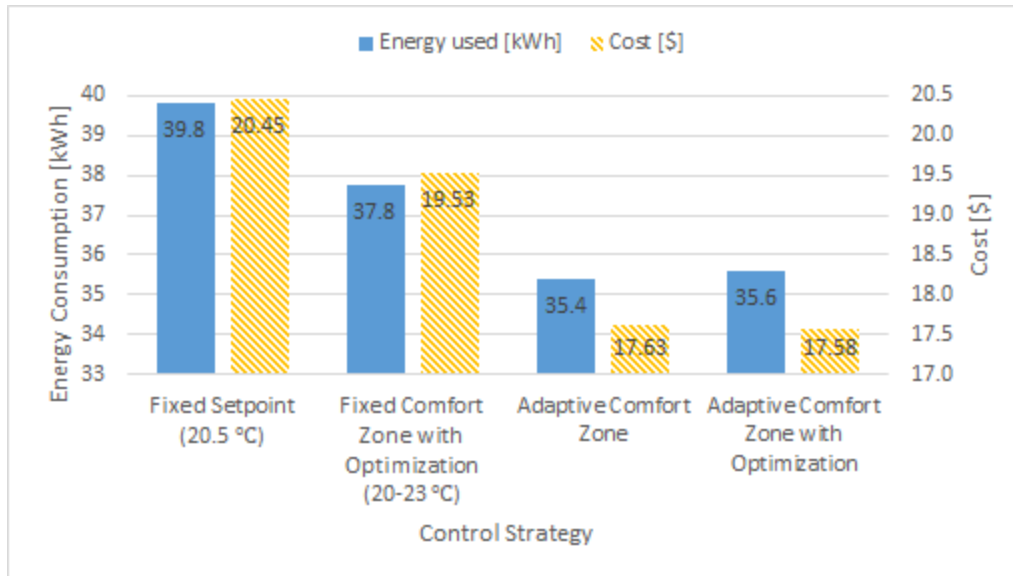


Figure 24. Energy consumption and cost for each energy control strategy for February 12 - 18.

The total cost and energy usage between the four simulations are shown in Figure 25 for September 27-October 3. This simulation resulted in the least amount of cost reduction. This time period was chosen to test optimization during summer months with cooling energy. Although this week was chosen since it resulted in the highest consumption of cooling energy during a simulation with a fixed setpoint, the building model does not consume enough energy for the optimization to make an impact comparable to that in January or February. The location chosen for simulations, San Francisco, does not experience long-term high temperatures and does not require high amounts of air conditioning in the summer. The simulations still resulted in a decrease in cost of \$0.03-0.19 and a decrease in energy consumption of 0.13-0.18 kWh. Optimization with an adaptive comfort zone resulted in \$5.20 savings when compared to a fixed setpoint.

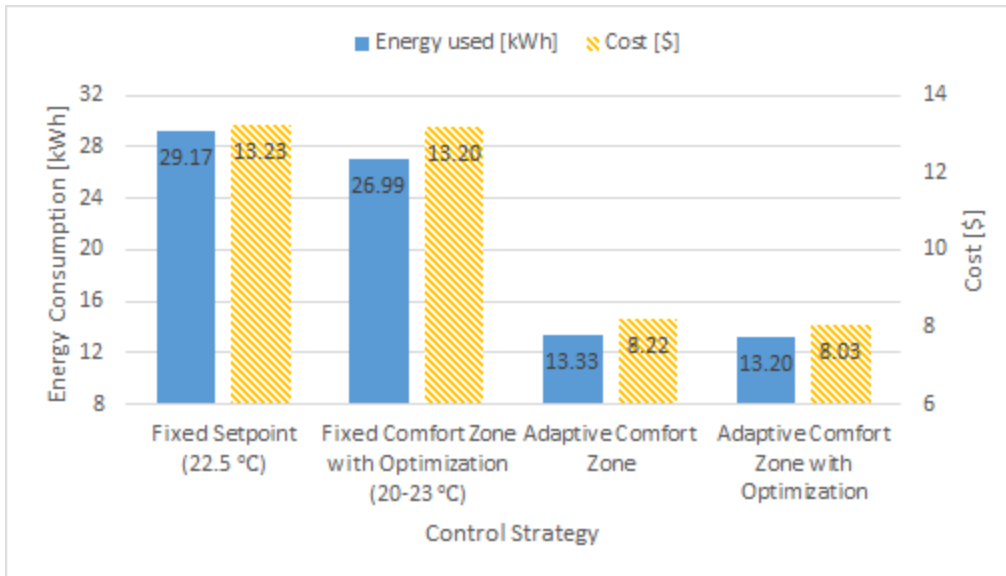


Figure 25. Energy consumption and cost for each energy control strategy for Sept 27 - Oct 3.

There are several areas of interest within the results including preheating, deviation from the comfort zone, and possible limitations of cost savings. To take a closer look at how the optimization is affecting the HVAC system, the energy consumption of the simulation under a fixed setpoint and under optimization for January 1 - 3 are plotted in Figure 26a. The price of energy at each hour is also plotted in the figure. The optimized energy consumption can be seen to increase in comparison to the fixed setpoint energy consumption directly before an increase in price. This increase represents the controller preheating the house to take advantage of the lower cost of energy. The results are shown more closely for Jan 2 12:00 - 24:00 in Figure 26b. The energy usage in the hours leading up to a higher price is significantly greater for the simulation with optimization than the one with a fixed setpoint. After the price increase, the simulation with optimization is able to continue to use less energy than the one with the fixed setpoint.

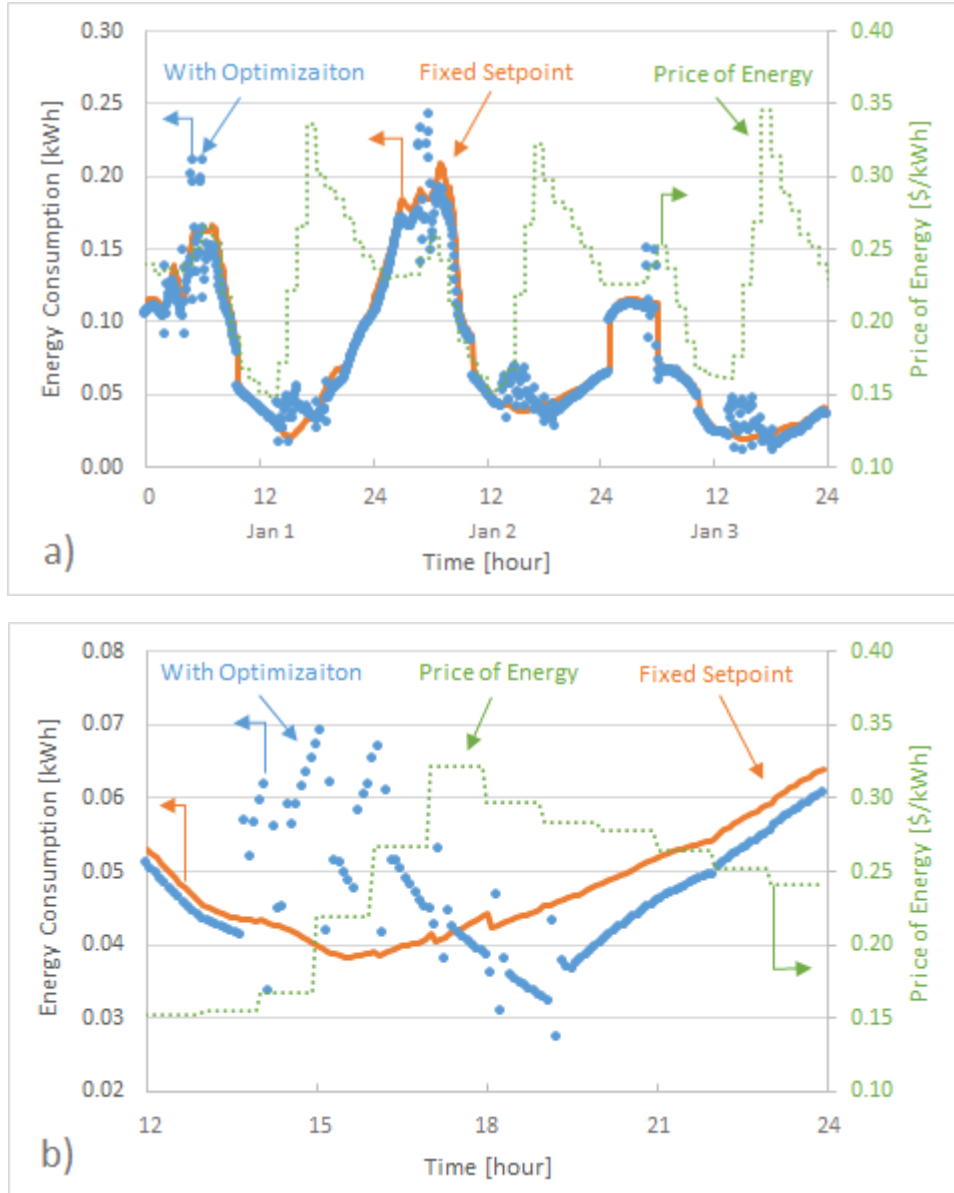


Figure 26. a) Plot of the energy consumption for a fixed setpoint and optimization with a fixed comfort zone over a three day time period with the price of energy at each timestep
 b) Plot of the energy consumption for a fixed setpoint and optimization with a fixed comfort zone over a 12 hour period (Jan 2 12-24) with the price of energy at each timestep.

The indoor temperature of the simulations with a fixed setpoint and with optimization with a fixed comfort zone are shown in Figure 27. The indoor temperature for the fixed setpoint temperature case oscillates within 1 °C of the setpoint temperature as expected whereas the

indoor temperature for the optimization case increases before the price of energy increases to take advantage of lower costs. The indoor temperature remains within the set comfort zone of 20 - 23 °C. Preheating is only effective for a limited period until the indoor temperature decreases to the lower bound of the comfort zone. This limitation hinders the ability to shift the load more than approximately an hour.

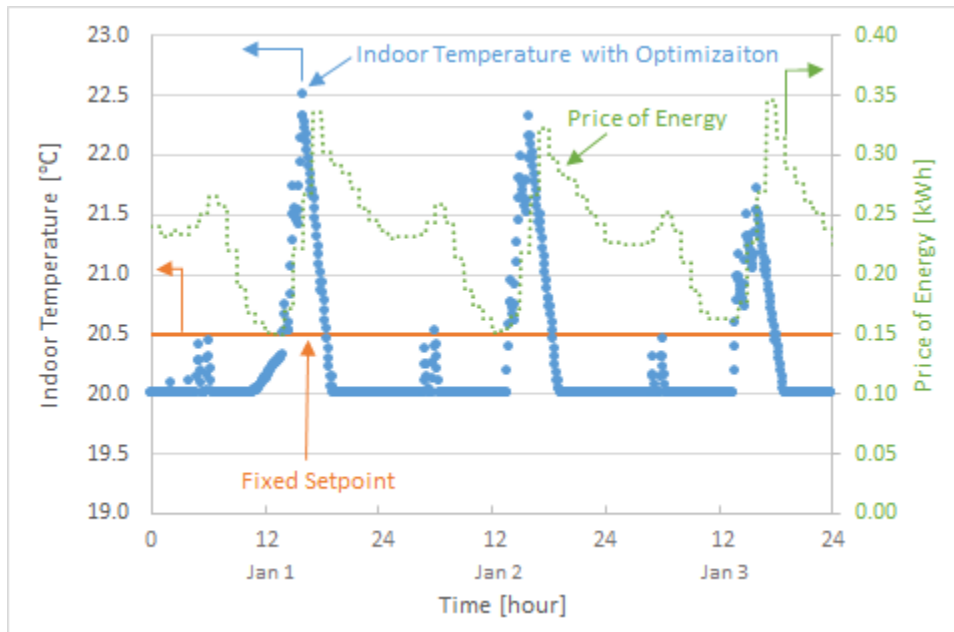


Figure 27. Plot of the indoor temperature for the two cases over Jan 1 - 3 with the price of energy at each timestep.

During the simulation in February, the indoor temperature left the comfort zone on three occasions when the indoor temperature naturally increased due to the weather conditions. A graph of the indoor temperature and adaptive comfort zone are shown in Figure 28 and the corresponding energy consumption for that time period is shown in Figure 29. At first, the energy system is heating the house to remain within the comfort region. Then the heating system stops and the indoor temperature continues to rise on its own due to a high outdoor temperature and solar radiation to the point where it exceeds the thermal comfort zone. The adaptive control

strategy used the air conditioning to keep the temperature within the comfort zone whereas with optimization, the indoor temperature was allowed to exceed the bound since turning on the air conditioning is uncommon in winter months.

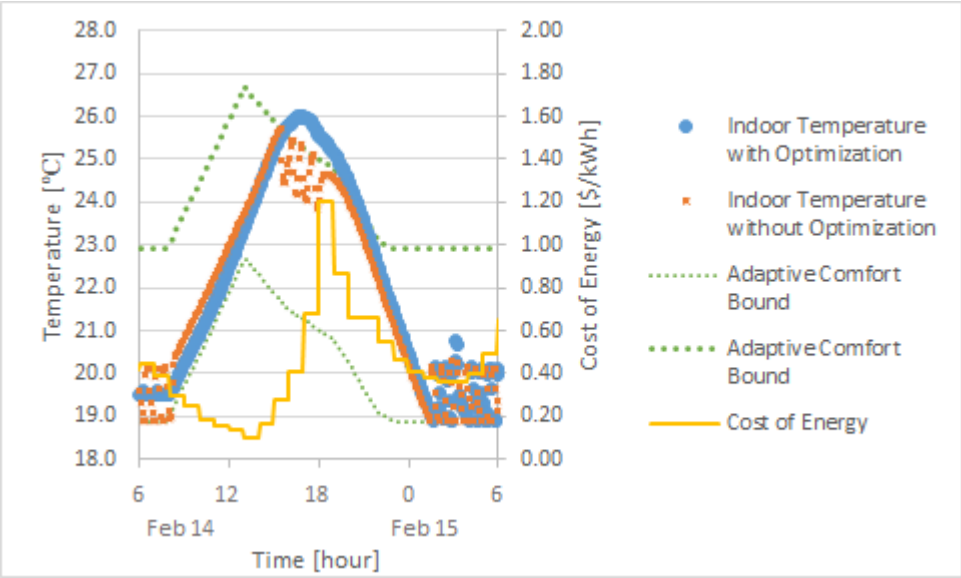


Figure 28. Plot of indoor temperature with the adaptive comfort zone marked where the indoor temperature naturally exceeds the comfort zone due to weather conditions.

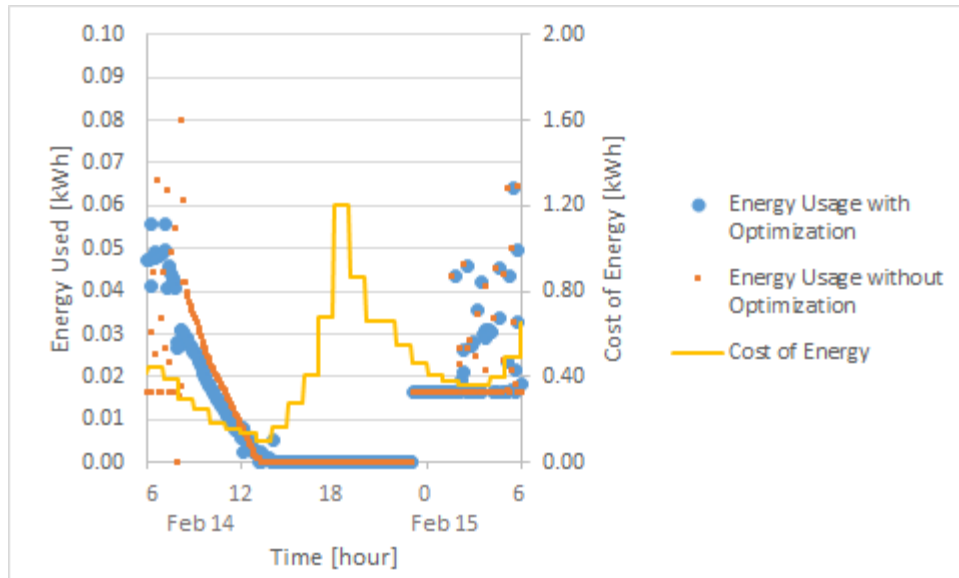


Figure 29. Plot of the energy consumption during the time when the indoor temperature naturally exceeds the comfort zone due to weather conditions.

To reconcile this inconsistency, the both control strategies could both allow or restrict air conditioning in winter months or a cost function could be added to quantify the cost associated with deviation from the thermal comfort zone. This cost function could be added to the optimization objective function to take it into account. In some cases, changing the length of prediction in the optimization may allow the controller to better predict instances where the indoor temperature will deviate from the thermal comfort zone.

Another area of interest is the cause of the week in February savings to be less than those of the week in January. One contributing factor is that there was no energy usage during the times of extreme high prices. The indoor temperature and energy consumption during one of the times of extreme price increase is shown in Figure 30 and Figure 31 respectively. As shown in the figures, no energy is consumed with or without optimization during the time of extremely high

price. The indoor temperature naturally remains within the adaptive comfort zone during that time. Therefore, any potential savings from optimization are not brought to fruition.

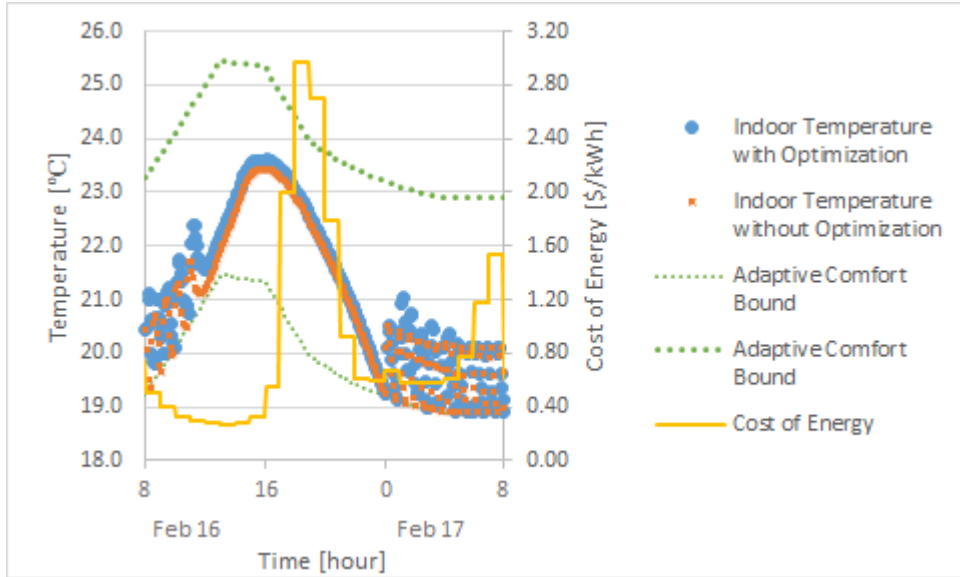


Figure 30. Indoor temperature in February during a time of high price increase.

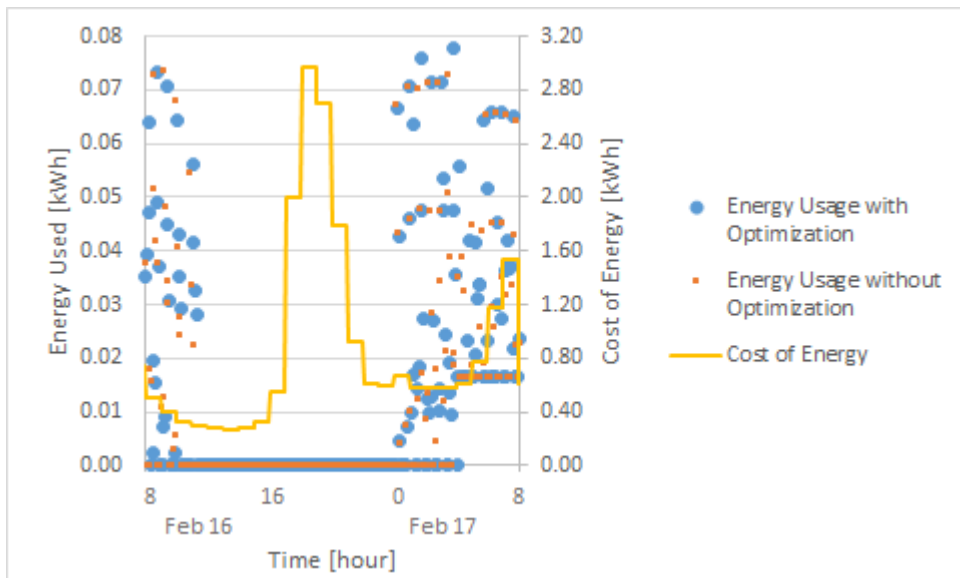


Figure 31. Energy consumption in February during a time of high price increase.

A potential cause of this discrepancy is that the weather file and wholesale cost data are from two different years. Therefore, the wholesale price may not correspond to times of high energy consumption in this set of simulations.

Another limitation on the amount of savings is due to a delay between the controller and EnergyPlus. The resulting energy consumption under optimization during a time of changing price is shown in Figure 32. The control system is attempting to preheat the house by consuming more energy before the price increases, but two timesteps of the resulting energy consumption occur during the time of increased price and not before as intended. This would result in fewer savings than may be possible. One solution would be to adjust the setpoint schedule to send the appropriate setpoint temperatures two timesteps prior than current.

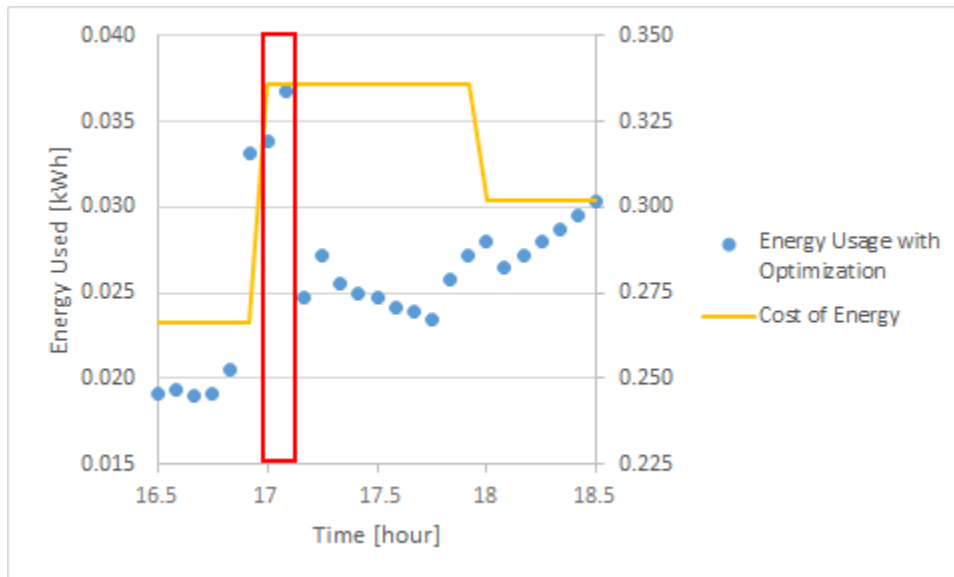


Figure 32. Plot of energy consumption during a change in price.

Overall, results show the optimization working to preheat or precool the house before a price increase. The greater the proportional constant, α , from Equation 2.5, between retail and wholesale price, the greater impact preheating or precooling will have on cost savings. This work

calls for future work to determine the most effective pricing models for load shifting and decreasing overall costs. The simulation strategy can be expanded to include different building models, locations, timeframes, and setpoint control strategies including occupancy sensing.

6. Conclusion

A residential building energy control strategy was designed and implemented using a robust validation technique. The energy control strategy worked to minimize the total cost, set the future setpoint temperature schedule, and was used with building energy simulation software. The control strategy was not able to maintain thermal comfort in three time periods during the week in February simulation.

The validation method of using UCEF and EnergyPlus allowed for testing the energy control strategy with the building simulator and is highly adaptable for different control strategies, pricing models, locations, and building models. The validation method is scalable to include multiple building models with different characteristics in the same simulation. The residential building model used was representative of a typical house found in the San Francisco, California area. Adding fuzzy control to the setpoint temperature schedule made the results more representative of how a residential heating system works. The pricing model used was based on the real-time demand of the market and provided a large enough difference in pricing to favor load shifting.

The optimization control strategy resulted in significant savings in the January, February, and September weeks compared to the fixed setpoint temperature schedule. The month of September resulted in the least cost savings since little cooling energy was used regardless of optimization due to the temperate climate of San Francisco. The optimization control strategy with a fixed comfort zone resulted in less savings compared to the adaptive setpoint temperature strategy. The adaptive setpoint control strategy already provides significant savings compared to a fixed setpoint temperature schedule, so that optimization can only provide a slight improvement. The

benefits of load shifting are limited since the effects of preheating and precooling are only felt for about one hour before the system returns to a normal energy load. Additionally, the time delay between the controller and implementation in EnergyPlus reduced the expected cost savings.

6.1. Future Work

Future work includes improving the indoor temperature prediction. As mentioned, the indoor temperature prediction becomes less accurate as the prediction timeframe increases. Machine learning methods such as reinforced learning can be used to improve the indoor temperature prediction, which will result in more accurate energy predictions. Additionally, the length of prediction performed by the optimizer can be varied to determine if a longer prediction timeframe would provide additional benefits such as avoiding exceeding the comfort zone due to overheating. A cost function for deviation from the comfort zone can be determined and added to the objective function of the optimizer to allow the indoor temperature to leave the comfort zone in pursuit of increased savings. The pricing model can be altered to determine its effectiveness for load shifting and decreasing overall costs. The pricing model can better correlate to the weather by updating the weather data to be from the same year as the pricing data.

The building model, location, and run period can be altered to determine the savings in different locations and during different times of year. The simulations can be repeated for another location with higher outdoor temperatures to test the optimization in summer months. The simulations can be scaled to include multiple house models with varying materials, efficiencies, and control preferences or methods. Distributed energy resources such as photovoltaic panels and energy

storage can be added to the housing models to determine their effect on modeling transactive energy.

References

- [1] US Energy Information Administration (EIA) Independent Study and Analysis, Primary energy Overview, <https://www.eia.gov/totalenergy/data/browser/?tbl=T01.01#/?f=A>, May 2021.
- [2] IPCC, 2014: Summary for Policymakers. In: *Climate Change 2014: Mitigation of Climate Change. Contribution of Working Group III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA.
- [3] NASA, "Global Climate Change: Vital Signs of the Planet: Global Temperature", <https://climate.nasa.gov/vital-signs/global-temperature/>, May 2021.
- [4] NASA, "Global Climate Change: Land-Ocean Temperature Index (C)", https://data.giss.nasa.gov/gistemp/graphs/graph_data/Global_Mean_Estimates_based_on_Land_and_Ocean_Data/graph.txt, May 2021.
- [5] NASA "Global Climate Change: Vital Signs of the Planet: Global Sea Level", <https://climate.nasa.gov/vital-signs/sea-level/>, May 2021.
- [6] LLNL Flow Charts. <https://flowcharts.llnl.gov/>.
- [7] US Energy Information Administration (EIA) Independent Statistics and Analysis, "Annual Energy Outlook 2021 with Projections to 2050", Table A2, May 2021.
- [8] US Energy Information Administration (EIA) Independent Statistics and Analysis, "Renewable energy explained", <https://www.eia.gov/energyexplained/renewable-sources/>, May 2021
- [9] Office of Energy Efficiency and Renewable Energy, "Confronting the Duck Curve: How to Address Over-Generation of Solar Energy", <https://www.energy.gov/eere/articles/confronting-duck-curve-how-address-over-generation-solar-energy>, May 2021.

- [10] Pecan Street Research Institute, "GreenTech Media: Are Solar Panels Facing the Wrong Direction?", <https://www.pecanstreet.org/2013/11/are-solar-panels-facing-the-wrong-direction/#:~:text=Pecan%20Street%20analyzed%2050%20homes,a%2065%20percent%20peak%20reduction.>, accessed May 2021.
- [11] J. Situ and D. Wright, "Economic analysis of residential solar microgrids," *2017 IEEE International Conference on Smart Energy Grid Engineering (SEGE)*, Oshawa, ON, 2017, pp. 96-102, doi: 10.1109/SEGE.2017.8052783.
- [12] S. Wong, N. Lévesque, V. Delisle, A. Gagné and L. Proulx, "Using the thermal energy storage potential of residential homes for ToU rate savings and demand response," *2017 IEEE International Conference on Smart Energy Grid Engineering (SEGE)*, Oshawa, ON, 2017, pp. 223-228.
- [13] M. F. Zia, M. Benbouzid, E. Elbouchikhi, S. M. Muyeen, K. Techato and J. M. Guerrero, "Microgrid Transactive Energy: Review, Architectures, Distributed Ledger Technologies, and Market Analysis," in *IEEE Access*, vol. 8, pp. 19410-19432, 2020, doi: 10.1109/ACCESS.2020.2968402.
- [14] M. F. Zia, E. Elbouchikhi, M. Benbouzid and J. M. Guerrero, "Microgrid Transactive Energy Systems: A Perspective on Design, Technologies, and Energy Markets," *IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society*, 2019, pp. 5795-5800, doi: 10.1109/IECON.2019.8926947.
- [15] California ISO, Today Outlook, <http://www.caiso.com/TodaysOutlook/Pages/default.aspx>, May 2021.

- [16] L. Zhao, Z. Yang and W. Lee, "The Impact of Time-of-Use (TOU) Rate Structure on Consumption Patterns of the Residential Customers," in *IEEE Transactions on Industry Applications*, vol. 53, no. 6, pp. 5130-5138, Nov.-Dec. 2017, doi: 10.1109/TIA.2017.2734039.
- [17] F. Oldewurtel, A. Ulbig, A. Parisio, G. Andersson and M. Morari, "Reducing peak electricity demand in building climate control using real-time pricing and model predictive control," *49th IEEE Conference on Decision and Control (CDC)*, Atlanta, GA, 2010, pp. 1927-1932, doi: 10.1109/CDC.2010.5717458.
- [18] R. Godina, E. M. G. Rodrigues, M. Shafie-khah, E. Poursmaeil and J. P. S. Catalão, "Energy optimization strategy with Model Predictive Control and demand response," *2017 IEEE International Conference on Environment and Electrical Engineering and 2017 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe)*, Milan, 2017, pp. 1-5, doi: 10.1109/EEEIC.2017.7977767.
- [19] J. Ma, S. J. Qin, B. Li and T. Salsbury, "Economic model predictive control for building energy systems," *ISGT 2011*, Anaheim, CA, 2011, pp. 1-6, doi: 10.1109/ISGT.2011.5759140.
- [20] M. J. Risbeck and J. B. Rawlings, "Economic Model Predictive Control for Time-Varying Cost and Peak Demand Charge Optimization," in *IEEE Transactions on Automatic Control*, vol. 65, no. 7, pp. 2957-2968, July 2020, doi: 10.1109/TAC.2019.2939633.
- [21] E. Vrettos, K. Lai, F. Oldewurtel and G. Andersson, "Predictive Control of buildings for Demand Response with dynamic day-ahead and real-time prices," *2013 European Control Conference (ECC)*, Zurich, 2013, pp. 2527-2534, doi: 10.23919/ECC.2013.6669762.
- [22] C. Wang, K. Pattawi, H. Lee, "Energy saving impact of occupancy-driven thermostat for residential buildings," in *Energy & Buildings* vol. 211, 2020.

- [23] C. Wang, "Cost Effective and Non-Intrusive Occupancy Detection in Residential Building Through Machine Learning Algorithm", *Santa Clara University Department of Mechanical Engineering*, 2020. (used for figures, now replaced)
- [24] D. A. Winkler, A. Yadav, C. Chitu and A. E. Cerpa, "OFFICE: Optimization Framework For Improved Comfort & Efficiency," *2020 19th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, Sydney, Australia, 2020, pp. 265-276, doi: 10.1109/IPSN48710.2020.00030.
- [25] 2018 International Energy Conservation Code (IECC), 2020 International Code Council, Inc., <https://codes.iccsafe.org/content/IECC2018P3>, May 2021.
- [26] 2012 International Energy Conservation Code (IECC), 2020 International Code Council, Inc., IECC Climate Zone Map, accessed May 2021.
- [27] EnergyPlus, US Department of Energy Building Technologies Office, <https://energyplus.net/>, April 2021.
- [28] Universal CPS Environment for Federation (UCEF), National Institute of Standards and Technology (NIST), <https://pages.nist.gov/ucef/>, April 2021.
- [29] O. Bretscher, *Linear Algebra with Applications*, 5th ed., Pearson, 2013.
- [30] S. Holmes, "RMS Error", <http://statweb.stanford.edu/~susan/courses/s60/split/node60.html>, 2000-11-28.
- [31] M. Andersen, J. Dahl, L. Vadenberghe, CVXOPT Python Software for Convex Optimization, <https://cvxopt.org/>
- [32] Microsoft Support, "Define and solve a problem by using Solver", Microsoft, accessed June 2021.

- [33] California Independent System Operator (CAISO) Day-Ahead Market Watch Report January 1 - January 7 <http://www.caiso.com/Documents/Day-AheadDailyMarketWatchJan01-2021.html>, April 2021.
- [34] Pacific Gas & Electric (PG&E) Residential Electricity Rates <https://www.pge.com/tariffs/index.page>, May 2021.
- [35] U.S Department of Energy Building Energy Codes Program, Residential Prototype Building Models, https://www.energycodes.gov/development/residential/ieci_models, April 2021.
- [36] Roth, T. , Song, Y. , Burns, M. , Neema, H. , Emfinger, W. and Sztipanovits, J. “Cyber-Physical System Development Environment for Energy Applications”, in 2017 Proceedings of the ASME 2017 11th International Conference on Energy Sustainability (ES2017), Charlotte, NC. doi:10.1115/ES2017-3589.
- [37] 1516-2010 - IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-- Framework and Rules. <https://standards.ieee.org/standard/1516-2010.html>.
- [38] Functional Mock-up Interface. <https://fmi-standard.org/>.
- [39] Singer, J., Roth, T., Wang, C., Nguyen, C. & Lee, H. EnergyPlus Integration Into Cosimulation Environment to Improve Home Energy Saving Through Cyber-Physical Systems Development. *J. Energy Resour. Technol.* 141, (2019).
- [40] EnergyPlus Weather Data by Location, TMY3 San Francisco Intl AP 724940 https://energyplus.net/weather-location/north_and_central_america_wmo_region_4/USA/CA/USA_CA_San.Francisco.Intl.AP.724940_TMY3, April 2021.
- [41] T. Roth, C. Nguyen, J. Singer, C. Wang, H. Lee, "EnergyPlus Integration into Co-Simulation Environment to Improve Home Energy Saving Through Cyber-Physical Systems

Development," *ASME Energy Sustainability Conference*, Lake Buena Vista, FL, 2018,

doi: 10.1115/ES2018-7295

Appendices

Appendix A. MATLAB Linear Regression Code for Indoor Temperature Prediction

```
% MATLAB code of linear regression to solve for indoor temperature
% prediction constants
x=readmatrix('SF Meter HP.xlsx','Sheet','TrainingHP (2)');
itr = 10; % number of iterations to average
n = 1000:1000:100000; % number of data points in each training set
l = length(n); % number of training sets
D = zeros(3,l); % matrix to hold the constants per training set
E = zeros(3,l*itr); % constants in each iteration before averaging
for a=1:itr % a is iteration number
    for i=1:l % loop for each training set
        N = zeros(n(i),3); % creating matrix of dependent variables
        B = zeros(n(i),1); % creating matrix of eused val
        r = randi([2 103985],1,n(i)); % creating array of random numbers to
select random data
        for j=1:n(i)
            m=r(j);
            for k=1:3
                N(j,k)=x(m,k); % matrix of (Outdoor - Indoor Temperatures),
energy used, and solar radiation
            end
            B(j,:) = x(m,4); % (future indoor temp - current indoor temp)/dt
column
        end
        F = N\B;% solving for the constants
```

```

    E(:,i+1*(a-1)) = F;

    D(:,i) = F ./itr + D(:,i);

end

end

```

```

% MATLAB Code to solve for RMSE of Indoor Temperature Prediction
% setting up matrices to be used

eUsed = x(:,2);
Tindoor = x(:,4);
Qsolar = x(:,3);
OutIn = x(:,1);
Tpredict = zeros(length(x),length(D));
Tdifference = zeros(length(x),length(D));
Tsum=zeros(1,length(D));
RMSE = zeros(1,length(D));
for p=1:length(D)
    for o =1:length(x)
        Tpredict(o,p) = D(1,p)*OutIn(o) +D(2,p)*eUsed(o) + D(3,p)*Qsolar(o); %
Calculating the predicted (future indoor temp - current indoor temp)/dt
        Tdifference(o,p)=(Tindoor(o)-Tpredict(o,p))^2; % squaring the difference
between the prediction and the true value
        Tsum(1,p)=Tsum(1,p)+Tdifference(o,p); % summing the squares
    end
    RMSE(1,p) = sqrt(Tsum(1,p)/length(x)); % calculating the RMSE
end

```

Appendix B. Python Optimization Code

```
# energyOptTset

from cvxopt import matrix, solvers
from cvxopt.modeling import op, dot, variable
import time
import pandas as pd
import sys
#start_time=time.process_time()

# Parameters for one week long simulation with 2hr horizon -----
-----
heatorcool = 'heat'
n=24 # number of timesteps within prediction windows (24 x 5-min timesteps
in 2 hr window)
comfortZone_upper = 23.0
comfortZone_lower = 20.0
timestep = 5*60
days = 7
totaltimesteps = days*12*24+3*12
pricingmultfactor = 4.0
pricingoffset = 0.10

occupancy_mode = False
noOccupancyHeat = 12
noOccupancyCool = 32
# Max and min for heating and cooling in adaptive setpoint control
heatTempMax = 26.2
heatTempMin = 18.9
coolTempMax = 30.2
coolTempMin = 22.9

# constant coefficients for indoor temperature equation -----
-----
c1 = 1.72*10**-5
if heatorcool == 'heat':
    c2 = 0.0031
else:
    c2 = -0.0031
c3 = 3.58*10**-7

# get inputs from UCEF -----
-----
day = int(sys.argv[1])
block = int(sys.argv[2]) +1+(day-1)*24 # block goes 0:23 (represents the
hour within a day)
temp_indoor_initial = float(sys.argv[3])

# Get data from excel/csv files -----
-----
# Get outdoor temps
df = pd.read_excel('OutdoorTemp.xlsx', sheet_name='Feb12thru19',header=0)
# print(df.head())
temp_outdoor_all=matrix(df.to_numpy())
```



```

df.columns = ['column1']
# use outdoor temps to get adaptive setpoints using lambda functions
convertOutTempToCoolTemp = lambda x: x*0.31 + 19.8
convertOutTempToHeatTemp = lambda x: x*0.31 + 15.8
adaptive_cooling_setpoints = df.apply(convertOutTempToCoolTemp)
adaptive_heating_setpoints = df.apply(convertOutTempToHeatTemp)
# print(adaptive_heating_setpoints)
# When temps too low or too high set to min or max (See adaptive
setpoints)
adaptive_cooling_setpoints.loc[(adaptive_cooling_setpoints['column1'] <
coolTempMin)] = coolTempMin
adaptive_cooling_setpoints.loc[(adaptive_cooling_setpoints['column1'] >
coolTempMax)] = coolTempMax
adaptive_heating_setpoints.loc[(adaptive_heating_setpoints['column1'] <
heatTempMin)] = heatTempMin
adaptive_heating_setpoints.loc[(adaptive_heating_setpoints['column1'] >
heatTempMax)] = heatTempMax
# change from pd dataframe to matrix
# print(adaptive_heating_setpoints)
adaptive_cooling_setpoints = matrix(adaptive_cooling_setpoints.to_numpy())
adaptive_heating_setpoints = matrix(adaptive_heating_setpoints.to_numpy())
# print(adaptive_heating_setpoints)

# get occupancy data
occupancy_df = pd.read_csv('occupancy_1hr.csv')
occupancy = matrix(occupancy_df['Occupancy'].to_numpy())

# get solar radiation
df = pd.read_excel('Solar.xlsx', sheet_name='Feb12thru19')
q_solar_all=matrix(df.to_numpy())

# get wholesale prices
df = pd.read_excel('WholesalePrice.xlsx', sheet_name='Feb12thru19')
wholesaleprice_all=matrix(df.to_numpy())

# c matrix is hourly cost per kWh of energy (I think this can be deleted)
Output = matrix(0.00, (totaltimesteps,2))
cost =0
c = matrix(0.20, (totaltimesteps,1))

# setting up optimization to minimize energy times price
x=variable(n) # x is energy usage that we are predicting

# A matrix is coefficients of energy used variables in constraint
equations (see PJs equations)
AA = matrix(0.0, (n*2,n))
k = 0
while k<n:
    j = 2*k
    AA[j,k] = timestep*c2
    AA[j+1,k] = -timestep*c2
    k=k+1
k=0
while k<n:

```

```

j=2*k+2
while j<2*n-1:
    AA[j,k] = AA[j-2,k]*-timestep*c1+ AA[j-2,k]
    AA[j+1,k] = AA[j-1,k]*-timestep*c1+ AA[j-1,k]
    j+=2
k=k+1

## Added 12/11
# making sure energy is positive for heating
heat_positive = matrix(0.0, (n,n))
i = 0
while i<n:
    heat_positive[i,i] = -1.0 # setting boundary condition: Energy used at
each timestep must be greater than 0
    i +=1
# making sure energy is negative for cooling
cool_negative = matrix(0.0, (n,n))
i = 0
while i<n:
    cool_negative[i,i] = 1.0 # setting boundary condition: Energy used at
each timestep must be less than 0
    i +=1

d = matrix(0.0, (n,1))
# inequality constraints
heatineq = (heat_positive*x<=d)
coolineq = (cool_negative*x<=d)
energyLimit = matrix(0.25, (n,1)) # .4 before
heatlimiteq = (cool_negative*x<=energyLimit)

# creating S matrix to make b matrix simpler
temp_outdoor= temp_outdoor_all[(block-1)*12:(block-1)*12+n,0] # getting
next two hours of data
q_solar=q_solar_all[(block-1)*12:(block-1)*12+n,0]

# get price for next two hours
cc=wholesaleprice_all[(block-1)*12:(block-
1)*12+n,0]*pricingmultfactor/1000+pricingoffset
S = matrix(0.0, (n,1))
S[0,0] = timestep*(c1*(temp_outdoor[0]-
temp_indoor_initial)+c3*q_solar[0])+temp_indoor_initial

i=1
while i<n:
    S[i,0] = timestep*(c1*(temp_outdoor[i]-S[i-1,0])+c3*q_solar[i])+S[i-
1,0]
    i+=1

# b matrix is constant term in constant equations
b = matrix(0.0, (n*2,1))

adaptiveHeat = adaptive_heating_setpoints[(block-1)*12:(block-1)*12+n,0]
adaptiveCool = adaptive_cooling_setpoints[(block-1)*12:(block-1)*12+n,0]

```

```

if occupancy_mode == True:
    occupancy_Checker = occupancy[(block-1)*12:(block-1)*12+n,0]
    k = 0
    while k<n:
        if occupancy_Checker[k,0] ==1:
            b[2*k,0]=adaptiveCool[k,0]-S[k,0]
            b[2*k+1,0]=-adaptiveHeat[k,0]+S[k,0]
        else:
            b[2*k,0]=noOccupancyCool-S[k,0]
            b[2*k+1,0]=-noOccupancyHeat+S[k,0]
        k=k+1
# else:
#     k = 0
#     while k<n:
#         b[2*k,0]=adaptiveCool[k,0]-S[k,0]
#         b[2*k+1,0]=-adaptiveHeat[k,0]+S[k,0]
#         k=k+1
else:
    k = 0
    while k<n:
        b[2*k,0]=comfortZone_upper-S[k,0]
        b[2*k+1,0]=-comfortZone_lower+S[k,0]
        k=k+1

# time to solve for energy at each timestep
ineq = (AA*x <= b)
if heatercool == 'heat':
    lp2 = op(dot(cc,x),ineq)
    op.addconstraint(lp2, heatineq)
    op.addconstraint(lp2,heatlimiteq)
if heatercool == 'cool':
    lp2 = op(dot(-cc,x),ineq)
    op.addconstraint(lp2, coolineq)
lp2.solve()

# If primal infeasibility is found (optimization cannot be done within
constraints), set the energy usage to 0 and the predicted indoor
temperature to the comfort region bound.
if x.value == None:
    energy = matrix(0.00, (13,1))
    print('energy consumption')
    j=0
    while j<13:
        print(energy[j])
        j = j+1
    j=0
    temp_indoor = matrix(0.0, (totaltimesteps,1))
    while j<13:
        # temp_indoor[j]=adaptiveHeat[j,0]
        temp_indoor[j]=comfortZone_lower
        j=j+1
    print('indoor temp prediction')
    j = 0
    while j<13:

```

```

    print(temp_indoor[j,0])
    j = j+1

print('pricing per timestep')
j = 0
while j<13:
    print(cc[j,0])
    j = j+1

print('outdoor temp')
j = 0
while j<13:
    print(temp_outdoor[j,0])
    j = j+1
print('solar radiation')
j = 0
while j<13:
    print(q_solar[j,0])
    j = j+1
print('adaptive heating setpoints')
j = 0
while j<12:
    print(adaptiveHeat[j,0])
    j=j+1
print('adaptive cooling setpoints')
j = 0
while j<12:
    print(adaptiveCool[j,0])
    j=j+1
quit()

energy = x.value

# print(lp2.objective.value())
temp_indoor = matrix(0.0, (n,1))
temp_indoor[0,0] = temp_indoor_initial
p = 1
while p<n:
    temp_indoor[p,0] = timestep*(c1*(temp_outdoor[p-1,0]-temp_indoor[p-1,0])+c2*energy[p-1,0]+c3*q_solar[p-1,0])+temp_indoor[p-1,0]
    p = p+1
# Output[(block-1)*12:(block-1)*12+n,0] = energy[0:n,0] #0:12
# Output[(block-1)*12:(block-1)*12+n,1] = temp_indoor[0:n,0] #0:12
cost = cost + lp2.objective.value()
#cost = cost + cc[0:12,0].trans()*energy[0:12,0]
# print(ii)
# print(cost)
# print(Output)
# print(temp_indoor)
# temp_indoor_initial= temp_indoor[12,0]
# print(temp_indoor_initial)

# # solve for thermostat temperature at each timestep
# thermo = matrix(0.0, (n,1))

```

```

# i = 0
# while i<n:
#   thermo[i,0] = (-d2*temp_indoor[i,0]-d3*temp_outdoor[i]-
d4*q_solar[i]+energy[i]*1000/12)/d1
#   i = i+1

print('energy consumption')
j=0
while j<13:
    print(energy[j])
    j = j+1

print('indoor temp prediction')
j = 0
while j<13:
    print(temp_indoor[j,0])
    j = j+1

print('pricing per timestep')
j = 0
while j<13:
    print(cc[j,0])
    j = j+1

print('outdoor temp')
j = 0
while j<13:
    print(temp_outdoor[j,0])
    j = j+1
print('solar radiation')
j = 0
while j<13:
    print(q_solar[j,0])
    j = j+1
print('adaptive heating setpoints')
j = 0
while j<12:
    print(adaptiveHeat[j,0])
    j=j+1
print('adaptive cooling setpoints')
j = 0
while j<12:
    print(adaptiveCool[j,0])
    j=j+1

```

Appendix C. Java Code for Socket Federate in UCEF

```

package org.webgme.guest.socket;

import org.webgme.guest.socket.rti.*;

import org.cpswt.config.FederateConfig;
import org.cpswt.config.FederateConfigParser;
import org.cpswt.hla.InteractionRoot;

```

```

import org.cpswt.hla.base.AdvanceTimeRequest;

import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;

// Import extra packages
import org.cpswt.utils.CpswtUtils;
import java.io.*;
import java.net.*;

// Define the Socket type of federate for the federation.

public class Socket extends SocketBase {
    private final static Logger log = LogManager.getLogger();

    private double currentTime = 0;

    // Kaleb // Config stuff TODO
    // START WITH simID as ZERO because java is zero indexed
    int simID = 0;    // Change simID based on socket number

    // Kaleb // Define global variables
    String[] varNames=new String[15]; // will have to add more empty
strings based on how many strings we send/receive
    String[] doubles= new String[15]; // will have to add more empty
strings based on how many strings we send/receive
    String varNameSeparator = "@";
    String doubleSeparator = ",";
    int numVars = 0;
    String eGSH=null, eGSC=null, ePeople=null; //values sent to EnergyPlus
    boolean empty=true;
    boolean receivedSimTime = false;    // this is for "received"
interaction while loop
    // Kaleb //

    public Socket(FederateConfig params) throws Exception {
        super(params);
    }

    private void checkReceivedSubscriptions() {
        InteractionRoot interaction = null;
        while ((interaction = getNextInteractionNoWait()) != null) {
            if (interaction instanceof Controller_Socket) {
                handleInteractionClass((Controller_Socket) interaction);
            }
            else {
                log.debug("unhandled interaction: {}",
interaction.getClassName());
            }
        }
    }

    private void execute() throws Exception {
        if(super.isLateJoiner()) {

```

```

        log.info("turning off time regulation (late joiner)");
        currentTime = super.getLBTS() - super.getLookAhead();
        super.disableTimeRegulation();
    }

    ////////////////////////////////////////////////////
    // TODO perform basic initialization below //
    ////////////////////////////////////////////////////

    // Read IP address and Port number from config.txt
    log.info("create bufferedReader");
    File file= new
File("/home/vagrant/Desktop/Projects/TwoFedEPOpt/EnergyPlusOpt2Fed/EnergyP
lusOpt2Fed_generated/config.txt");
    BufferedReader br = new BufferedReader(new FileReader(file));
    log.info("bufferedReader successful");
    String st = "";
    String ipAdd = "";
    int portNo = 0;
    while ((st = br.readLine())!=null){
        log.info(st);
        if(st.equals("ip_address:")){
            ipAdd = br.readLine();
        }
        if(st.equals("port_number:")){
            portNo = Integer.valueOf(br.readLine());
        }
    }
    log.info(ipAdd);
    log.info(portNo);

    log.info("Waiting for EnergyPlus simulations to join...");
    // end test config.txt

    // Kaleb // Add socket here:
    InetAddress addr = InetAddress.getByName(ipAdd); // the address
needs to be changed in config.txt
    ServerSocket welcomeSocket = new ServerSocket(portNo, 50, addr);
// Can also be changed in config.txt
    java.net.Socket connectionSocket = welcomeSocket.accept(); //
initial connection will be made at this point
    System.out.println("connection successful");
    log.info("connection successful");

    InputStreamReader inFromClient = new
InputStreamReader(connectionSocket.getInputStream());
    BufferedReader buffDummy = new BufferedReader(inFromClient);
    DataOutputStream outToClient = new
DataOutputStream(connectionSocket.getOutputStream());
    // Kaleb // end socket

    AdvanceTimeRequest atr = new AdvanceTimeRequest(currentTime);
    putAdvanceTimeRequest(atr);

```

```

if(!super.isLateJoiner()) {
    log.info("waiting on readyToPopulate...");
    readyToPopulate();
    log.info("...synchronized on readyToPopulate");
}

////////////////////////////////////
// TODO perform initialization that depends on other federates
below //

////////////////////////////////////

if(!super.isLateJoiner()) {
    log.info("waiting on readyToRun...");
    readyToRun();
    log.info("...synchronized on readyToRun");
}

startAdvanceTimeThread();
log.info("started logical time progression");

// Kaleb // Define variables for getting EP data
String header, time="0", varName="", value="";
double varValue=0;
String dataString = "";
// Kaleb //

while (!exitCondition) {
    atr.requestSyncStart();
    enteredTimeGrantedState();

    //////////////////////////////////////
    // TODO send interactions that must be sent every logical //
    // time step below //
    //////////////////////////////////////

    // Set the interaction's parameters.
    //
    //     Socket_Controller vSocket_Controller =
create_Socket_Controller();
    //     vSocket_Controller.set_actualLogicalGenerationTime( <
YOUR VALUE HERE > );
    //     vSocket_Controller.set_dataString( < YOUR VALUE HERE > );
    //     vSocket_Controller.set_federateFilter( < YOUR VALUE HERE
> );
    //     vSocket_Controller.set_originFed( < YOUR VALUE HERE > );
    //     vSocket_Controller.set_simID( < YOUR VALUE HERE > );
    //     vSocket_Controller.set_sourceFed( < YOUR VALUE HERE > );
    //     vSocket_Controller.sendInteraction(getLRC(), currentTime
+ getLookAhead());

    // Kaleb // Getting values from fmu
    if((header = buffDummy.readLine()).equals("TERMINATE")){

```



```

        exitCondition = true;
    }
    time = buffDummy.readLine();
    System.out.println("in loop header=" + header + " t=" + time);

    while(!(varName = buffDummy.readLine()).isEmpty()) {
        value = buffDummy.readLine();
        System.out.println("Received: " + varName + " as " +
value);
        // Add any variable that you want to get from EnergyPlus
here...

        // Names have to match the modelDescription.xml file
        // before @ is varName and before , is value
        // varName first!!!
        if(varName.equals("epSendOutdoorAirTemp")){
            dataString = dataString +varName+varNameSeparator;
            dataString = dataString +value+doubleSeparator;
        }
        if(varName.equals("epSendZoneMeanAirTemp")){
            dataString = dataString +varName+varNameSeparator;
            dataString = dataString +value+doubleSeparator;
        }
        if(varName.equals("epSendZoneHumidity")){
            dataString = dataString +varName+varNameSeparator;
            dataString = dataString +value+doubleSeparator;
        }
        if(varName.equals("epSendHeatingEnergy")){
            dataString = dataString +varName+varNameSeparator;
            dataString = dataString +value+doubleSeparator;
        }
        if(varName.equals("epSendCoolingEnergy")){
            dataString = dataString +varName+varNameSeparator;
            dataString = dataString +value+doubleSeparator;
        }
        if(varName.equals("epSendNetEnergy")){
            dataString = dataString +varName+varNameSeparator;
            dataString = dataString +value+doubleSeparator;
        }
        if(varName.equals("epSendEnergyPurchased")){
            dataString = dataString +varName+varNameSeparator;
            dataString = dataString +value+doubleSeparator;
        }
        if(varName.equals("epSendEnergySurplus")){
            dataString = dataString +varName+varNameSeparator;
            dataString = dataString +value+doubleSeparator;
        }
        if(varName.equals("epSendDayOfWeek")){
            dataString = dataString +varName+varNameSeparator;
            dataString = dataString +value+doubleSeparator;
        }
        if(varName.equals("epSendSolarRadiation")){
            dataString = dataString +varName+varNameSeparator;
            dataString = dataString +value+doubleSeparator;
        }
    }

```

```

        if(varName.equals("epSendHeatingSetpoint")){
            dataString = dataString +varName+varNameSeparator;
            dataString = dataString +value+doubleSeparator;
        }
        if(varName.equals("epSendCoolingSetpoint")){
            dataString = dataString +varName+varNameSeparator;
            dataString = dataString +value+doubleSeparator;
        }
    }
    // for checking timestep
    dataString =
dataString+"timestep"+varNameSeparator+String.valueOf(currentTime)+doubleS
eparator;

    // Send Socket_Controller interaction containing eplus data
    Socket_Controller sendEPData = create_Socket_Controller();
    sendEPData.set_simID(simID);
    sendEPData.set_dataString(dataString);
    log.info("Sent sendEPData interaction from socket{} with {}",
simID , dataString);
    sendEPData.sendInteraction(getLRC());

    // Wait to receive Controller_Socket information containing
setpoints that will be sent to eplus
    while (!receivedSimTime){
        log.info("waiting to receive SimTime...");
        synchronized(lrc){
            lrc.tick();
        }
        checkReceivedSubscriptions();
        if(!receivedSimTime){
            CpswtUtils.sleep(100);
        }
    }
    receivedSimTime = false;
    //

    // Empty Data String for next time step
    dataString = "";

    // send eGSH and eGSC to eplus, if you want to send something
else to EnergyPlus need to add here
    if (empty==true) {
        outToClient.writeBytes("NOUPDATE\r\n\r\n");
    }
    else {
        outToClient.writeBytes("SET\r\n" + time + "\r\n"+
"epGetStartCooling\r\n" + eGSC + "\r\n" + "epGetStartHeating\r\n" + eGSH +
"\r\n" + "\r\n");
        System.out.println("SET\r\n" + time + "\r\n"+
"epGetStartCooling\r\n" + eGSC + "\r\n" + "epGetStartHeating\r\n" + eGSH +
"\r\n" + "\r\n");
    }
    outToClient.flush();

```

```

// Kaleb //

// ReceiveModel vReceiveModel = create_ReceiveModel();
// vReceiveModel.set_dataString(dataString);
// log.info("Sent receiveModel interaction with {}",
dataString);

// vReceiveModel.sendInteraction(getLRC());

// // removing time delay...
// while (!receivedSimTime){
//     log.info("waiting to receive SimTime...");
//     synchronized(lrc){
//         lrc.tick();
//     }
//     checkReceivedSubscriptions();
//     if(!receivedSimTime){
//         CpswtUtils.sleep(1000);
//     }
// }
// receivedSimTime = false;
// // .....

// System.out.println(currentTime);

////////////////////////////////////
// TODO break here if ready to resign and break out of while
loop //
////////////////////////////////////

if (!exitCondition) {
    currentTime += super.getStepSize();
    AdvanceTimeRequest newATR =
        new AdvanceTimeRequest(currentTime);
    putAdvanceTimeRequest(newATR);
    atr.requestSyncEnd();
    atr = newATR;
}

// call exitGracefully to shut down federate
exitGracefully();

////////////////////////////////////
// TODO Perform whatever cleanups are needed before exiting the app
//
////////////////////////////////////
}

```

```

private void handleInteractionClass(Controller_Socket interaction) {
    ////////////////////////////////////////////////////
    // TODO implement how to handle reception of the interaction //
    ////////////////////////////////////////////////////

    // Kaleb //
    // exit while loop above waiting for Controller_Socket
    receivedSimTime = true;

    // epvalues are not empty
    empty = false;

    // get dataString from Controller and separate into varNames and
doubles
    int receivedID = interaction.get_simID();
    String holder = null;
    if(receivedID == simID){
        holder = interaction.get_dataString();
        System.out.println("holder = "+ holder );

        String vars[] = holder.split(doubleSeparator);
        System.out.println("vars[0] = "+vars[0]);
        int j=0;
        for( String token : vars){
            System.out.println("token = " +token);
            String token1[] = token.split(varNameSeparator);
            System.out.println("token1[0] = "+token1[0]);
            System.out.println("token1[1] = "+token1[1]);
            varNames[j] = token1[0];
            doubles[j] = token1[1];
            System.out.println("varNames[j] = "+ varNames[j] );
            System.out.println("doubles[j] = "+ doubles[j] );
            j = j+1;
        }

        for(int i =0; i<j; i++){
            System.out.println("ReceivedData interaction " +
varNames[i] + " as " + doubles[i]);
            // if you are receiving something else besides variables
listed below, add another if()
            if(varNames[i].equals("epGetStartHeating")){
                eGSH = doubles[i];
                System.out.println("Received Heating setpoint as" +
varNames[i] + eGSH);
                log.info("Received Heating setpoint as {} = {}" ,
varNames[i] , eGSH);
            }
            if(varNames[i].equals("epGetStartCooling")){
                eGSC = doubles[i];
                System.out.println("Received Cooling setpoint as" +
varNames[i] + eGSC);
                log.info("Received Cooling setpoint as {} = {}" ,
varNames[i] , eGSC);
            }
        }
    }
}

```

```

        if(varNames[i].equals("epGetPeople")){
            ePeople = doubles[i];
            System.out.println("Received People as" + varNames[i] +
ePeople);
            log.info("Received People as {} = {}" , varNames[i] ,
ePeople);
        }
    }
}
// Kaleb //

}

public static void main(String[] args) {
    try {
        FederateConfigParser federateConfigParser =
            new FederateConfigParser();
        FederateConfig federateConfig =
            federateConfigParser.parseArgs(args, FederateConfig.class);
        Socket federate =
            new Socket(federateConfig);
        federate.execute();
        log.info("Done.");
        System.exit(0);
    }
    catch (Exception e) {
        log.error(e);
        System.exit(1);
    }
}
}
}

```

Appendix D. Java Code for Controller Federate in UCEF

```

package org.webgme.guest.controller;

import org.webgme.guest.controller.rti.*;

import org.cpswt.config.FederateConfig;
import org.cpswt.config.FederateConfigParser;
import org.cpswt.hla.InteractionRoot;
import org.cpswt.hla.base.AdvanceTimeRequest;

import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;

// Importin gother packages
import java.io.*;
import java.net.*;
import org.cpswt.utils.CpswtUtils;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileWriter;
import java.util.Random;    // random num generator

```

```

import java.lang.*;

// Define the Controller type of federate for the federation.

public class Controller extends ControllerBase {
    private final static Logger log = LogManager.getLogger();

    private double currentTime = 0;

    public Controller(FederateConfig params) throws Exception {
        super(params);
    }

    // Kaleb // defining global variables
    int numSockets = 1; // Change this
    String[] varNames = new String[15]; // add more empty vals if sending
more vars
    String[] doubles = new String[15];
    String[] dataStrings = new String[numSockets];
    String[] holder=new String[numSockets];
    double[] outTemps=new double[numSockets];
    double[] coolTemps= new double[numSockets];
    double[] heatTemps= new double[numSockets];
    double[] heatTempFromOpt= new double[numSockets];
    double[] zoneTemps= new double[numSockets];
    double[] zoneRHs= new double[numSockets];
    double[] heatingEnergy= new double[numSockets];
    double[] coolingEnergy= new double[numSockets];
    double[] netEnergy= new double[numSockets];
    double[] energyPurchased= new double[numSockets];
    double[] energySurplus= new double[numSockets];
    double[] solarRadiation= new double[numSockets];
    double[] receivedHeatTemp= new double[numSockets];
    double[] receivedCoolTemp= new double[numSockets];
    double[] dayOfWeek= new double[numSockets];
    double price = 10; // Set a default price here
    int[] numVars = new int[numSockets];
    String[] futureIndoorTemp = new String[12];

    String varNameSeparator = "@";
    String doubleSeparator = ",";
    String optDataString = "";
    int day = 0;

    int hour=0, nexthour=0, quarter=0, fivemin=0, onemin=0, simulatetime=0;
    double r1 =0.0;
    double Preset_cool=23.0, Preset_heat=20.0; // changed preset cool from
21.0 - PJ
    double event_p=0.0, duration_p=1.0, duration_q=0.0, nextevent_p=0.0;
    int occupancy = 2, check = 0, p=0, r2=0;

    String varname="";
    double value=0.0;

```

```

    double Last_cool=23.0, Last_heat=20.0; // changed Last_heat from 21.0 -
PJ

    boolean receivedSocket = false;
    boolean receivedMarket = false;
    boolean receivedReader = false;

    String timestep_Socket = "";
    String timestep_Reader = "";
    String timestep_Market = "";
    String timestep_Controller = "";

    int fuzzy_heat = 0; // NEEDS TO BE GLOBAL VAR outside of while loop
    int fuzzy_cool = 0; // NEEDS TO BE GLOBAL VAR outside of while loop

    // REMOVE NEXT TWO LINES!! this was for testing preloading weather
    // double[] outTemperature = new
double[] {7.2,6.7,6.1,4.4,4.4,6.1,5,7.8,8.9,9.4,10,10.6,11.1,13.9,13.9,11.1
,11.1,10.6,10.6,8.9,9.4,7.8,6.7,8.9,6.1,5.6,3.9,5.6,7.2,7.8,10,10,10,10,10
,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10};
    // int yeet = 0;
    // Kaleb //

    private void checkReceivedSubscriptions() {
        InteractionRoot interaction = null;
        while ((interaction = getNextInteractionNoWait()) != null) {
            if (interaction instanceof Socket_Controller) {
                handleInteractionClass((Socket_Controller) interaction);
            }
            else {
                log.debug("unhandled interaction: {}",
interaction.getClassName());
            }
        }
    }

    private void execute() throws Exception {
        if(super.isLateJoiner()) {
            log.info("turning off time regulation (late joiner)");
            currentTime = super.getLBTS() - super.getLookAhead();
            super.disableTimeRegulation();
        }

        ////////////////////////////////////////////////////////////////////
        // TODO perform basic initialization below //
        ////////////////////////////////////////////////////////////////////

        AdvanceTimeRequest atr = new AdvanceTimeRequest(currentTime);
        putAdvanceTimeRequest(atr);

        if(!super.isLateJoiner()) {
            log.info("waiting on readyToPopulate...");
            readyToPopulate();
        }
    }

```

```

        log.info("...synchronized on readyToPopulate");
    }

    ////////////////////////////////////////////////////
    // TODO perform initialization that depends on other federates
    below //
    ////////////////////////////////////////////////////

    if(!super.isLateJoiner()) {
        log.info("waiting on readyToRun...");
        readyToRun();
        log.info("...synchronized on readyToRun");
    }

    startAdvanceTimeThread();
    log.info("started logical time progression");

    while (!exitCondition) {
        atr.requestSyncStart();
        enteredTimeGrantedState();

        ////////////////////////////////////////////////////
        // TODO send interactions that must be sent every logical //
        // time step below //
        ////////////////////////////////////////////////////

        // Set the interaction's parameters.
        //
        // Controller_Socket vController_Socket =
        create_Controller_Socket();
        // vController_Socket.set_actualLogicalGenerationTime( <
        YOUR VALUE HERE > );
        // vController_Socket.set_dataString( < YOUR VALUE HERE > );
        // vController_Socket.set_federateFilter( < YOUR VALUE HERE
        > );
        // vController_Socket.set_originFed( < YOUR VALUE HERE > );
        // vController_Socket.set_simID( < YOUR VALUE HERE > );
        // vController_Socket.set_sourceFed( < YOUR VALUE HERE > );
        // vController_Socket.sendInteraction(getLRC(), currentTime
        + getLookAhead());

        System.out.println("timestep before receiving Socket/Reader: "+
        currentTime);
        log.info("timestep before receiving Socket/Reader:
        ",currentTime);
        // waiting to receive Socket_Controller and Reader_Controller
        while (!(receivedSocket)){

            //while (!(receivedSocket) || !(receivedReader)){
            log.info("waiting to receive Socket_Controller
            interaction...");
            synchronized(lrc){

```



```

        lrc.tick();
    }
    checkReceivedSubscriptions();
    if(!receivedSocket){
        CpswtUtils.sleep(100);
    }
    // }else if(!receivedReader){
    //     log.info("waiting on Reader_Controller...");
    //     CpswtUtils.sleep(100);
    // }
}
receivedSocket = false;
receivedReader = false;
System.out.println("timestep after receiving Socket/Reader and
before sending to Market: "+ currentTime);

// // TODO send Controller_Market here! vvvvvvvv
// log.info("sending Controller_Market interaction");
// Controller_Market sendMarket = create_Controller_Market();
// sendMarket.set_dataString("");
// System.out.println("Send controller_market and
Reader_Controller interaction:");
//     sendMarket.sendInteraction(getLRC());

//     log.info("waiting for Market_controller interaction...");
//     // Wait to receive price from market
//     while (!receivedMarket){
//         log.info("waiting to receive Market_Controller
interaction...");
//         synchronized(lrc){
//             lrc.tick();
//         }
//         checkReceivedSubscriptions();
//         if(!receivedMarket){
//             log.info("waiting on Market_Controller...");
//             CpswtUtils.sleep(100);
//         }
//     }
//     receivedMarket = false;
//     log.info("received Market_controller interaction!");
//     System.out.println("timestep after receiving Market: "+
currentTime);

// PJ's optimization
double hour = (double) ((currentTime%288) / 12);
log.info("hour is: ",hour);
System.out.println("hour is:"+hour);
String s = null;
String dataStringOpt = "";
String dataStringOptT = "";
String dataStringOptP = "";
String dataStringOptO = "";

```

```

String dataStringOptS = "";
String sblock = null;
String sday = null;
String separatorOpt = ",";
boolean startSavingE = false;
boolean startSavingT = false;
boolean startSavingP = false;
boolean startSavingO = false;
boolean startSavingS = false;

// comment here to stop optimization
process_
-----

    if (hour == 0){
        day = day+1;
    }

    if (hour%1 == 0){
        try {
            sblock= String.valueOf((int)hour);
            sday = String.valueOf(day);
            dataStringOpt = sblock;
            dataStringOptT = sblock;
            dataStringOptP = sblock;
            dataStringOptO = sblock;
            dataStringOptS = sblock;
            System.out.println("sblock:" +sblock);
            System.out.println("sday:" +sday);
            System.out.println("zonetemp string"
+String.valueOf(zoneTemps[0]));

            // Process p = Runtime.getRuntime().exec("python
./energyOpt.py " +sday + " "+sblock + " "+ String.valueOf(zoneTemps[0])); //
4 hr block method
            Process p = Runtime.getRuntime().exec("python
./energyOptTset2hr.py " +sday + " "+sblock + " "+
String.valueOf(zoneTemps[0])); // 1 timestep method

            BufferedReader stdInput = new BufferedReader(new
InputStreamReader(p.getInputStream()));
            System.out.println("Here is the result");

            while ((s = stdInput.readLine()) != null) {
                System.out.println(s);
                if (startSavingE == true) {
                    dataStringOpt = dataStringOpt + separatorOpt +
s;
                }
                if (startSavingT == true) {
                    dataStringOptT = dataStringOptT + separatorOpt
+ s;
                }
            }

```

```

        if (startSavingP == true) {
            dataStringOptP = dataStringOptP + separatorOpt
+ s;
        }
        if (startSavingO == true) {
            dataStringOptO = dataStringOptO + separatorOpt
+ s;
        }
        if (startSavingS == true) {
            dataStringOptS = dataStringOptS + separatorOpt
+ s;
        }
        if (s .equals("energy consumption")){
            startSavingE = true;
        }
        if (s .equals("indoor temp prediction")){
            startSavingT = true;
        }
        if (s .equals("pricing per timestep")){
            startSavingP = true;
        }
        if (s .equals("outdoor temp")){
            startSavingO = true;
        }
        if (s .equals("solar radiation")){
            startSavingS = true;
        }
        // if (s .equals("Certificate of primal
infeasibility found.")){

            // }
            // System.out.println(dataString);
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
    String vars[] = dataStringOpt.split(separatorOpt);
    String varsT[] = dataStringOptT.split(separatorOpt);
    String varsP[] = dataStringOptP.split(separatorOpt);
    String varsO[] = dataStringOptO.split(separatorOpt);
    String varsS[] = dataStringOptS.split(separatorOpt);

    for (int in =1;in<13;in++) {
        futureIndoorTemp[in-1]=varsT[in];
    }

    // Writing data to file
    try{
        // Create new file

        String
path="/home/vagrant/Desktop/Projects/TwoFedEPOpt/EnergyPlusOpt2Fed/EnergyP
lusOpt2Fed_deployment/DataSummaryfromCVXOPT.txt";

```

```

        File file = new File(path);

        // If file doesn't exists, then create it
        if (!file.exists()) {
            file.createNewFile();
        }

        FileWriter fw = new
FileWriter(file.getAbsolutePath(),true);
        BufferedWriter bw = new BufferedWriter(fw);

        // Write in file
        for (int in =1;in<13;in++) {

bw.write(vars[in]+"\\t"+varsT[in]+"\\t"+varsP[in]+"\\t"+varsO[in]+"\\t"+varsS[
in]+"\\n");
        }

        // Close connection
        bw.close();
    }
    catch(Exception e){
        System.out.println(e);
    }

    // resetting
    startSavingE = false;
    startSavingT = false;
    startSavingP = false;
    startSavingO = false;
    startSavingS = false;
    dataStringOpt = "";
}

// Setting setpoint temp for next hour
System.out.println("determine setpoints loop1");
if (hour%1 == 0){
    p=0;
    System.out.println("p"+String.valueOf(p));
}
heatTemps[0]=Double.parseDouble(futureIndoorTemp[p]);
heatTempFromOpt[0]=Double.parseDouble(futureIndoorTemp[p]);
System.out.println("heatTemp"+String.valueOf(heatTemps[0]));
coolTemps[0]=30.2;
System.out.println("coolTemp: "+String.valueOf(coolTemps[0]));
p=p+1;
System.out.println("p"+String.valueOf(p));

//-----
-----
// Now figure out all stuff that needs to be sent to socket...

// determine heating and cooling setpoints for each simID
// will eventually change this part for transactive energy

```

```

System.out.println("determine setpoints loop2");

// Fuzzy control
for(int i=0;i<numSockets;i++){
    double max_cool_temp = 30.2;
    double min_heat_temp = 18.9;
    double OFFSET = 0.6; // need to change slightly
higher/lower so E+ doesnt have issues

    // I think if we set these as a band
    heatTemps[i]=Double.parseDouble(futureIndoorTemp[p-1])-0.5;
    coolTemps[i]=Double.parseDouble(futureIndoorTemp[p-1])+0.5;

    // Determine minimum and maximum temperatures allowed (we
can probably print this from optimization code too)
    if (outTemps[i]<=10){
        min_heat_temp =18.9;
        max_cool_temp =22.9;
    }else if (outTemps[i]>=33.5){
        min_heat_temp =26.2;
        max_cool_temp =30.2;
    }else {
        min_heat_temp = 0.31*outTemps[i] + 17.8-2;
        max_cool_temp = 0.31*outTemps[i] + 17.8+2;
    }

    // Now set maximum cool and minimum heats:
    if (coolTemps[i]>=max_cool_temp){
        coolTemps[i]=max_cool_temp;
    }
    if (heatTemps[i]<=min_heat_temp){
        heatTemps[i]=min_heat_temp;
    }

    // For Cooling 1 degree under Cooling setpoint:
    if (zoneTemps[i] >= coolTemps[i]-.1){ // first check if
going to exit maximum band
        fuzzy_cool = -1;
    } else if (zoneTemps[i] <= coolTemps[i]-1.1){
        fuzzy_cool = 1;
    }
    coolTemps[i] = coolTemps[i] - 0.6 +fuzzy_cool*OFFSET; //
-0.6 so that oscillates 0.1-1.1 degree under cooling setpoint
    coolTemps[i] = 30.2; // do this for now to avoid turning on
AC

    // For Heating 1 degree under Heating setpoint:
    if (zoneTemps[i] <= heatTemps[i]+.1){ // first check if
going to exit minimum band
        fuzzy_heat = 1;
    } else if (zoneTemps[i] >= heatTemps[i]+1.1){
        fuzzy_heat = -1;
    }
}

```

```

        heatTemps[i] = heatTemps[i] + 0.6 +fuzzy_heat*OFFSET; //
+0.6 so that oscillates 0.1-1.1 degree above heating setpoint
    } //

```

```

// int Fuzzycool=0;
// int Fuzzyheat=0;
// // 0.5 degree fuzzy control (this oscillates indoor temp)
// double offset=0.6;
// if (zoneTemps[0]>=coolTemps[0]-0.5+offset){
//     Fuzzyheat = -1;
//     Fuzzycool = -1;
// }else if (zoneTemps[0]>=coolTemps[0]-0.5-offset){
//     Fuzzyheat = -1;
// }else if (zoneTemps[0]>=heatTemps[0]+0.5+offset){
//     Fuzzyheat = -1;
//     Fuzzycool = 1;
// }else if (zoneTemps[0]>=heatTemps[0]-0.5-offset){
//     Fuzzycool = 1;
// }else{
//     Fuzzyheat = 1;
//     Fuzzycool = 1;
// }
// coolTemps[0] = coolTemps[0] - 0.5 + Fuzzycool*offset;
// heatTemps[0] = heatTemps[0] + 0.5 + Fuzzyheat*offset;

System.out.println("heatTemps[0] = "+heatTemps[0] );
System.out.println("coolTemps[0] = "+coolTemps[0] );

// // use the following loop to solve for heating/cooling setpts
for each EnergyPlus simulation
// // if you only have one EnergyPlus simulation still use the
loop so that it is easy to add more
// // currently, adaptive setpoint control is implemented with
0.5 "fuzzy control"
// for(int i=0;i<numSockets;i++){
//     System.out.println("outTemps[i] = "+ outTemps[i] );
//     zoneTemps[i] = zoneTemps[i];
//     System.out.println("zoneTemps[i] = "+ zoneTemps[i] );
//     // zoneRHs[i] can add this but need to check FMU file and
also edit socket.java

```

```

// // Adaptive Setpoint Control:

```

```

// if (outTemps[0]<=10){
//     heatTemps[0]=18.9;
//     coolTemps[0]=22.9;
// }else if (outTemps[0]>=33.5){
//     heatTemps[0]=26.2;
//     coolTemps[0]=30.2;
// }else {
//     heatTemps[0] = 0.31*outTemps[0] + 17.8-2+0.5;
//     coolTemps[0] = 0.31*outTemps[0] + 17.8+2+0.5;

```

```

// }
// coolTemps[0]=30.2;
// // End Adaptive Setpoint Control

// heatTemps[0] = 20.0;
// coolTemps[0] = 30.2;

// // Fuzzy control
// for(int i=0;i<numSockets;i++){
//     double OFFSET = 0.6; // need to change slightly
higher/lower so E+ doesnt have issues

// // For Cooling 1 degree under Cooling setpoint:
// if (zoneTemps[i] >= coolTemps[i]-.1){
//     fuzzy_cool = -1;
// } else if (zoneTemps[i] <= coolTemps[i]-1.1){
//     fuzzy_cool = +1;
// }
// coolTemps[i] = coolTemps[i] - 0.6
+fuzzy_cool*OFFSET; // -0.6 so that oscillates 0.1-1.1 degree under
cooling setpoint

// // For Heating 1 degree under Heating setpoint:
// if (zoneTemps[i] >= heatTemps[i]+1.1){
//     fuzzy_heat = -1;
// } else if (zoneTemps[i] <= heatTemps[i]+.1){
//     fuzzy_heat = +1;
// }
// heatTemps[i] = heatTemps[i] + 0.6
+fuzzy_heat*OFFSET; // +0.6 so that oscillates 0.1-1.1 degree above
heating setpoint
// } //



---


// // Fuzzy control added for testing 5/16 without optimization
// for(int i=0;i<numSockets;i++){
//     double max_cool_temp = 30.2;
//     double min_heat_temp = 18.9;
//     double OFFSET = 0.6; // need to change slightly
higher/lower so E+ doesnt have issues

// // // I think if we set these as a band
// // heatTemps[i]=Double.parseDouble(futureIndoorTemp[p-1])-
0.5;
// // coolTemps[i]=Double.parseDouble(futureIndoorTemp[p-
1])+0.5;

// // Determine minimum and maximum temperatures allowed (we
can probably print this from optimization code too)
// if (outTemps[i]<=10){
//     min_heat_temp =18.9;
//     max_cool_temp =22.9;
// }else if (outTemps[i]>=33.5){

```

```

//      min_heat_temp =26.2;
//      max_cool_temp =30.2;
//    }else {
//      min_heat_temp = 0.31*outTemps[i] + 17.8-2;
//      max_cool_temp = 0.31*outTemps[i] + 17.8+2;
//    }

//    // Now set maximum cool and minimum heats:
//    if (coolTemps[i]>=max_cool_temp){
//      coolTemps[i]=max_cool_temp;
//    }
//    if (heatTemps[i]<=min_heat_temp){
//      heatTemps[i]=min_heat_temp;
//    }

//    // For Cooling 1 degree under Cooling setpoint:
//    if (zoneTemps[i] >= coolTemps[i]-.1){ // first check if
going to exit maximum band
//      fuzzy_cool = -1;
//    } else if (zoneTemps[i] <= coolTemps[i]-1.1){
//      fuzzy_cool = 1;
//    }
//    coolTemps[i] = coolTemps[i] - 0.6 +fuzzy_cool*OFFSET; //
-0.6 so that oscillates 0.1-1.1 degree under cooling setpoint
//    coolTemps[i] = 30.2; // do this for now to avoid turning
on AC

//    // For Heating 1 degree under Heating setpoint:
//    if (zoneTemps[i] <= heatTemps[i]+.1){ // first check if
going to exit minimum band
//      fuzzy_heat = 1;
//    } else if (zoneTemps[i] >= heatTemps[i]+1.1){
//      fuzzy_heat = -1;
//    }
//    heatTemps[i] = heatTemps[i] + 0.6 +fuzzy_heat*OFFSET; //
+0.6 so that oscillates 0.1-1.1 degree above heating setpoint
// }

// Send values to each socket federate
System.out.println("send to sockets interactions loop");
for(int i=0;i<numSockets;i++){
// simID = i; I am leaving this here to remind myself that i
is simID for each socket

    dataStrings[i] = "epGetStartCooling"+varNameSeparator;
    dataStrings[i] = dataStrings[i] + String.valueOf(coolTemps[i])
+ doubleSeparator;

    dataStrings[i] = dataStrings[i] +
"epGetStartHeating"+varNameSeparator;
    dataStrings[i] = dataStrings[i] + String.valueOf(heatTemps[i])
+ doubleSeparator;

    System.out.println("dataStrings[simID] = "+ dataStrings[i] );

```



```

        // SendModel vSendModel = create_SendModel();
        // vSendModel.set_dataString(dataString);
        // log.info("Sent_sendModel interaction with {}", dataString);
        // vSendModel.sendInteraction(getLRC());

        Controller_Socket sendControls = create_Controller_Socket();
        sendControls.set_dataString(dataStrings[i]);
        sendControls.set_simID(i);
        System.out.println("Send sendControls interaction: " +
coolTemps[i] + " to socket #" + i);
        sendControls.sendInteraction(getLRC());
        // SendModel vSendModel = create_SendModel();
        // vSendModel.set_dataString(dataStrings[i]);
        // System.out.println("Send SendModel interaction: " +
coolTemps[i] + " to socket #" + i);
        // vSendModel.sendInteraction(getLRC());

        dataStrings[i] = "";
    }

    System.out.println("timestep after sending Socket... should
advance after this: "+ currentTime);

    // System.out.println(currentTime);
    // System.out.println(dataString);

    // Writing data to file
    try{
        // Create new file

        String
path="/home/vagrant/Desktop/Projects/TwoFedEPOpt/EnergyPlusOpt2Fed/EnergyP
lusOpt2Fed_deployment/DataSummaryfromEP.txt";
        File file = new File(path);

        // If file doesn't exists, then create it
        if (!file.exists()) {
            file.createNewFile();
        }

        FileWriter fw = new
FileWriter(file.getAbsolutePath(),true);
        BufferedWriter bw = new BufferedWriter(fw);

        // Write in file
        bw.write(currentTime+"\t"+hour+"\t"+ zoneTemps[0)+"\t"+
outTemps[0)+"\t"+ solarRadiation[0)+"\t" + receivedHeatTemp[0)+"\t"+
receivedCoolTemp[0)+"\t"+heatTempFromOpt[0)+"\t"
+heatTemps[0)+"\t"+coolTemps[0)+"\n");

        // Close connection

```

```

        bw.close();
    }
    catch(Exception e){
        System.out.println(e);
    }
}

////////////////////////////////////
// TODO break here if ready to resign and break out of while
loop //
////////////////////////////////////

    if (!exitCondition) {
        currentTime += super.getStepSize();
        AdvanceTimeRequest newATR =
            new AdvanceTimeRequest(currentTime);
        putAdvanceTimeRequest(newATR);
        atr.requestSyncEnd();
        atr = newATR;
    }
}

// call exitGracefully to shut down federate
exitGracefully();

////////////////////////////////////
// TODO Perform whatever cleanups are needed before exiting the app
//
////////////////////////////////////
}

private void handleInteractionClass(Socket_Controller interaction) {
    //////////////////////////////////////
    // TODO implement how to handle reception of the interaction //
    //////////////////////////////////////

    // can now exit while loop waiting for this interaction
    log.info("received RCMModel_Controller interaction");
    receivedSocket = true;

    // Could make global var that holds simIDs but it would just be
    0,1,2,...
    // int simID = 0;
    int simID = interaction.get_simID();
    System.out.println("numVars[simID] = " + numVars[simID]);
    holder[simID] = interaction.get_dataString();
    System.out.println("holder[simID] = "+ holder[simID] );

    System.out.println("handle interaction loop");
}

```

```

// "varName{varNameSplitter}double{doubleSplitter}"!!!
String vars[] = holder[simID].split(doubleSeparator);
System.out.println("vars[0] = "+vars[0]);
System.out.println("length of vars = " + vars.length);
int j=0;
for( String token : vars){
    System.out.println("j = "+j);
    System.out.println("token = " +token);
    String token1[] = token.split(varNameSeparator);
    System.out.println("token1[0] = "+token1[0]);
    System.out.println("token1[1] = "+token1[1]);
    varNames[j] = token1[0];
    doubles[j] = token1[1];
    System.out.println("varNames[j] = "+ varNames[j] );
    System.out.println("doubles[j] = "+ doubles[j] );
    j = j+1;
}

// organize varNames and doubles into vectors of values
for(int i=0; i<j;i++){
    System.out.println("i = "+i);
    if(varNames[i].equals("epSendZoneMeanAirTemp")){
        zoneTemps[simID] = Double.valueOf(doubles[i]);
    }
    else if(varNames[i].equals("epSendOutdoorAirTemp")){
        outTemps[simID] = Double.valueOf(doubles[i]);
    }
    else if(varNames[i].equals("epSendZoneHumidity")){
        zoneRHs[simID] = Double.valueOf(doubles[i]);
    }
    else if(varNames[i].equals("epSendHeatingEnergy")){
        heatingEnergy[simID] = Double.valueOf(doubles[i]);
    }
    else if(varNames[i].equals("epSendCoolingEnergy")){
        coolingEnergy[simID] = Double.valueOf(doubles[i]);
    }
    else if(varNames[i].equals("epSendNetEnergy")){
        netEnergy[simID] = Double.valueOf(doubles[i]);
    }
    else if(varNames[i].equals("epSendEnergyPurchased")){
        energyPurchased[simID] = Double.valueOf(doubles[i]);
    }
    else if(varNames[i].equals("epSendEnergySurplus")){
        energySurplus[simID] = Double.valueOf(doubles[i]);
    }
    else if(varNames[i].equals("epSendDayOfWeek")){
        dayOfWeek[simID] = Double.valueOf(doubles[i]);
    }
    else if(varNames[i].equals("epSendSolarRadiation")){
        solarRadiation[simID] = Double.valueOf(doubles[i]);
    }
    else if(varNames[i].equals("epSendHeatingSetpoint")){
        receivedHeatTemp[simID] = Double.valueOf(doubles[i]);
    }
}

```

```

    }
    else if (varNames[i].equals("epSendCoolingSetpoint")) {
        receivedCoolTemp[simID] = Double.valueOf(doubles[i]);
    }
    else if (varNames[i].equals("price")) {
        price = Double.valueOf(doubles[i]);
    }
    // checking timesteps:
    else if (varNames[i].equals("timestep")) {
        timestep_Socket = doubles[i];
    }
}
}

public static void main(String[] args) {
    try {
        FederateConfigParser federateConfigParser =
            new FederateConfigParser();
        FederateConfig federateConfig =
            federateConfigParser.parseArgs(args, FederateConfig.class);
        Controller federate =
            new Controller(federateConfig);
        federate.execute();
        log.info("Done.");
        System.exit(0);
    }
    catch (Exception e) {
        log.error(e);
        System.exit(1);
    }
}
}
}

```

Appendix E. EnergyPlus Input (IDF) File

!-Generator IDFEditor 1.50
!-Option SortedOrder

!-NOTE: All comments with '!' are ignored by the IDFEditor and are generated automatically.
!- Use '!' comments if they need to be retained when using the IDFEditor.

!- ===== ALL OBJECTS IN CLASS: VERSION =====

!\$Id: eplus.res.template,v 1.15 2019/02/07 17:10:57 d3a061 Exp d3a061 \$

!\$Log: eplus.res.template,v \$

!Revision 1.15 2019/02/07 17:10:57 d3a061

!Added defaults for duct R-values. Added a code default (taken from
!consolidated.materials.template) and a moisture_regime default (from
!the same place).

!

!Revision 1.14 2019/01/28 18:07:01 d3a061

!Added default settings for wwr. Fixed problem that was using the
!'system' parm where it should have used 'hvac_type'; also added
!'missing setting of hvac_type to 'heatpump' where appropriate.
!Fixed bug in how r_crawlwall and r_bsmtwall were being defaulted.
!Beefed up screening of tropical_designation as applied to singe-family

!homes (was accidentally including singlewide manufactured homes in
!the check).
!
!Revision 1.13 2018/12/13 20:26:25 d3a061
!Added the \$TEMPLATE_PATH variable to simplify the process() calls.
!
!Revision 1.12 2018/12/13 20:09:00 d3a061
!Fairly extensive cleanup and refactoring to make make the template work with
!the "canonical" index variables and other such things, including making
!cfm25 a valid input that is coordinated with leakage_ratio.
!
!Revision 1.11 2018/10/15 17:40:35 d3a061
!Changed hard-coded shower_gpm, sink_gpm, and bath_gpm variables to changeable
!parameters with defaults.
!
!Revision 1.10 2018/09/17 17:36:16 d3a061
!Fixed big that was setting \$humidity_designation and/or \$tropical_designation to
!undef instead of ". Added the \$ach25 parameter as a better alternative to the
!\$leakage_ratio that had been used in the past (the latter, if given, still has
!precedence over the former). Added calls to generate a globals_report. And af
!few cleanups of code.
!
!Revision 1.9 2018/09/11 15:38:48 d3a061
!Added kludges to allow more consistent parm column names for climate zones, moisture
!regimes, and humidity/tropical designations. Put in a check to bomb if the user
!specifies any specific LED components to the lighting (needs to be enhanced to properly
!accommodate LEDs, but...later). Fixed default value of \$ua.
!
!Revision 1.8 2018/05/21 17:13:05 d3a061
!Added/clarified comments on the \$max_supply calcs.
!
!Revision 1.7 2018/05/17 21:37:17 d3a061
!Fixed a bunch of typos that were introduced when the setdef() calls were
!inserted in place of ||=. Also added in 2018 IECC default lighting fractions
!that had been inadvertently omitted before.
!
!Revision 1.6 2018/05/08 15:53:18 d3a061
!Replaced all uses of ||= with setdef() or val_or(). Added in re-wired (haha)
!lighting procedure. A few other fix-ups, comments, indents, etc., that were
!left out when the original 8.6 template was created. (Sigh.)
!
!Revision 1.5 2018/02/08 20:23:25 d3a061
!Updated DHW energy factors for new Federal standards that are current as
!of 2018.
!
!Revision 1.4 2018/02/07 16:22:53 d3a061
!Reworked DHW savings for 2015 and 2018 IECCs.
!
!Revision 1.3 2017/04/27 21:41:17 mend509
!Added the additional lighting load for the unconditioned floor area of the semi-conditioned prototype
!
!Revision 1.2 2017/04/24 15:45:58 d3a061
!Dropped all the Windows line ending characters (ugh--it marks every line
!as updated in the diffs), and changed to the "English" version of \$WARNING
!to eliminate the Perl warnings about the deprecated control-W version of
!that variable.
!
!Revision 1.1 2017/04/12 22:43:34 mend509
!Initial revision
!
!Revision 1.6 2015/06/27 00:14:44 mend509
!Revised insulation quality calculation
!
!Revision 1.5 2015/05/21 16:51:15 mend509
!Archive after determination analysis
!
!Revision 1.4 2014/06/27 16:27:02 mend509
!Check in before DHW changes
!
!Revision 1.2 2014/05/19 23:06:43 mend509

```

!Two main revisions:
!1: Re-wired ground temperature sections to read from previously generated gtp files if available
!2: Copied over 2012 IECC into 2015 IECC variables where applicable
!
!Revision 1.1 2014/05/16 23:22:50 mend509
!Initial revision
!
!Revision 1.2 2013/03/29 17:30:21 mend509
!Archived after generating first set of masonry input files
!
!Revision 1.1 2013/03/24 05:30:59 mend509
!Initial revision
!
!-----
! The Residential Prototype Building Models were developed by Pacific Northwest National Laboratory (PNNL),
! under contract with the U.S. Department of Energy (DOE).
! The purpose of developing these building prototype models is to measure energy saving impacts from
! various residential energy codes. Detailed model development descriptions are documented in
! PNNL report titled "Cost-Effectiveness Analysis of the 2009 and 2012 IECC Residential Provisions
! - Technical Support Document".
!-----
! GPARM parameters as run:
! ID = SF+CZ3CWH+USA_CA_San.Francisco.Intl.AP.724940+hp+crawlspace+IECC_2018
! weatherfile = USA_CA_San.Francisco.Intl.AP.724940_TMY3.epw
! climate_zone = Climate Zone 3
! moisture_regime = Marine
! humidity_designation = Not Warm-Humid
! tropical_designation = Not Tropical
! bldg_type = Single-Family
! fndn_type = Crawlspace
! system_tag = Heat Pump
! code = IECC_2018
! permits = 438.804
! cfm25 = 4
! dhw_type =
Version,
  9.2;          !- Version Identifier

!- ===== ALL OBJECTS IN CLASS: SIMULATIONCONTROL =====

!$Id: consolidated.materials.template,v 1.5 2019/02/07 17:04:56 d3a061 Exp d3a061 $
!$Log: consolidated.materials.template,v $
!Revision 1.5 2019/02/07 17:04:56 d3a061
!Eliminated default setting for 'code'. Commented out the semi-cond specs.
!
!Revision 1.4 2018/05/17 21:42:40 d3a061
!Ran dos2unix on it.
!
!Revision 1.3 2017/04/12 22:42:05 mend509
!Added the 2018 IECC hash
!
!Revision 1.2 2017/04/12 21:49:58 d3a061
!Changes to avoid deprecated Perlisms.
!
!Revision 1.1 2017/03/07 22:15:07 d3a061
!Initial revision
!
!Revision 1.8 2015/06/27 00:15:04 mend509
!Revised insulation quality calculation
!
!Revision 1.7 2015/06/04 03:10:16 d31143
!*** empty log message ***
!
!Revision 1.6 2015/06/01 22:09:59 d31143
!added wall insulation degradation factor for FOA.
!
!Revision 1.5 2015/05/21 16:52:14 mend509
!Archive after determination analysis
!

```

```

!Revision 1.4 2015/03/30 22:41:58 mend509
!Wall framing fraction fix didn't work; trying again
!
!Revision 1.3 2015/03/30 22:40:17 mend509
!$framing_fraction for walls defaults to 16 in OC but can take value though the parm file now
!
!Revision 1.2 2014/05/19 23:06:16 mend509
!Copied 2012 IECC envelope requirements into a 2015 hash
!
!Revision 1.1 2014/05/16 23:25:45 mend509
!Initial revision
!
!Revision 1.2 2013/03/29 17:30:59 mend509
!Archived after generating first set of masonry input files
!
!Revision 1.1 2013/03/24 05:34:54 mend509
!Initial revision
!
SimulationControl,
  Yes,          !- Do Zone Sizing Calculation
  Yes,          !- Do System Sizing Calculation
  No,           !- Do Plant Sizing Calculation
  Yes,          !- Run Simulation for Sizing Periods
  Yes;         !- Run Simulation for Weather File Run Periods

!- ===== ALL OBJECTS IN CLASS: BUILDING =====

Building,
  SF+CZ3CWH+USA_CA_San.Francisco.Intl.AP.724940+hp+crawlspace+IECC_2018, !- Name
  0,           !- North Axis {deg}
  Suburbs,     !- Terrain
  0.04,        !- Loads Convergence Tolerance Value
  0.4,         !- Temperature Convergence Tolerance Value {deltaC}
  FullExterior, !- Solar Distribution
  25,          !- Maximum Number of Warmup Days
  6;          !- Minimum Number of Warmup Days

!- ===== ALL OBJECTS IN CLASS: SURFACECONVECTIONALGORITHM:INSIDE =====

SurfaceConvectionAlgorithm:Inside,
  TARP;        !- Algorithm

!- ===== ALL OBJECTS IN CLASS: TIMESTEP =====

Timestep,
  12;          !- Number of Timesteps per Hour

!- ===== ALL OBJECTS IN CLASS: SITE:LOCATION =====

! Location and design-day objects created by:
! Site:Location and design-day objects created by:
! /projects/bigsim/bin/ddy2idf /projects/bigsim/weather/EnergyPlus/tmy3.new/all/USA_CA_San.Francisco.Intl.AP.724940_TMY3.ddy
!
Site:Location,
  San Francisco Intl Ap_CA_USA Design_Conditions, !- Name
  37.62,      !- Latitude {deg}
  -122.40,    !- Longitude {deg}
  -8.00,      !- Time Zone {hr}
  2.00;       !- Elevation {m}

!- ===== ALL OBJECTS IN CLASS: SIZINGPERIOD:DESIGNDAY =====

SizingPeriod:DesignDay,
  San Francisco Intl Ap Ann Htg 99.6% Condns DB, !- Name
  1,          !- Month

```

```

21,           !- Day of Month
WinterDesignDay,   !- Day Type
3.8,           !- Maximum Dry-Bulb Temperature {C}
0.0,           !- Daily Dry-Bulb Temperature Range {deltaC}
DefaultMultipliers, !- Dry-Bulb Temperature Range Modifier Type
,             !- Dry-Bulb Temperature Range Modifier Day Schedule Name
Wetbulb,        !- Humidity Condition Type
3.8,           !- Wetbulb or DewPoint at Maximum Dry-Bulb {C}
,             !- Humidity Condition Day Schedule Name
,             !- Humidity Ratio at Maximum Dry-Bulb {kgWater/kgDryAir}
,             !- Enthalpy at Maximum Dry-Bulb {J/kg}
,             !- Daily Wet-Bulb Temperature Range {deltaC}
101301.,       !- Barometric Pressure {Pa}
2.2,           !- Wind Speed {m/s}
150,           !- Wind Direction {deg}
No,            !- Rain Indicator
No,            !- Snow Indicator
No,            !- Daylight Saving Time Indicator
ASHRAEClearSky, !- Solar Model Indicator
,             !- Beam Solar Day Schedule Name
,             !- Diffuse Solar Day Schedule Name
,             !- ASHRAE Clear Sky Optical Depth for Beam Irradiance (taub) {dimensionless}
,             !- ASHRAE Clear Sky Optical Depth for Diffuse Irradiance (taud) {dimensionless}
0.00;         !- Sky Clearness

```

SizingPeriod:DesignDay,

```

San Francisco Intl Ap Ann Clg .4% Condns DB=>MWB, !- Name
8,           !- Month
21,          !- Day of Month
SummerDesignDay,   !- Day Type
28.3,         !- Maximum Dry-Bulb Temperature {C}
8.5,         !- Daily Dry-Bulb Temperature Range {deltaC}
DefaultMultipliers, !- Dry-Bulb Temperature Range Modifier Type
,             !- Dry-Bulb Temperature Range Modifier Day Schedule Name
Wetbulb,        !- Humidity Condition Type
17.2,         !- Wetbulb or DewPoint at Maximum Dry-Bulb {C}
,             !- Humidity Condition Day Schedule Name
,             !- Humidity Ratio at Maximum Dry-Bulb {kgWater/kgDryAir}
,             !- Enthalpy at Maximum Dry-Bulb {J/kg}
,             !- Daily Wet-Bulb Temperature Range {deltaC}
101301.,       !- Barometric Pressure {Pa}
5.9,         !- Wind Speed {m/s}
310,         !- Wind Direction {deg}
No,          !- Rain Indicator
No,          !- Snow Indicator
No,          !- Daylight Saving Time Indicator
ASHRAETau,    !- Solar Model Indicator
,             !- Beam Solar Day Schedule Name
,             !- Diffuse Solar Day Schedule Name
0.365,        !- ASHRAE Clear Sky Optical Depth for Beam Irradiance (taub) {dimensionless}
2.435;        !- ASHRAE Clear Sky Optical Depth for Diffuse Irradiance (taud) {dimensionless}

```

!- ===== ALL OBJECTS IN CLASS: RUNPERIOD =====

RunPeriod,

```

oneweek,      !- Name
2,           !- Begin Month
12,          !- Begin Day of Month
,            !- Begin Year
2,           !- End Month
18,          !- End Day of Month
,            !- End Year
,            !- Day of Week for Start Day
Yes,         !- Use Weather File Holidays and Special Days
Yes,         !- Use Weather File Daylight Saving Period
No,          !- Apply Weekend Holiday Rule
Yes,         !- Use Weather File Rain Indicators
Yes;         !- Use Weather File Snow Indicators

```


!- ===== ALL OBJECTS IN CLASS: SITE:WATERMAINSTEMPERATURE =====

! Weather_file = USA_CA_San.Francisco.Intl.AP.724940_TMY3.epw;
! Using the stat2idf and the ddy2idf script...
! Water mains temperature correlation object created by:
! Water mains temperature object created by:
! /projects/bigsim/resstd/vm/bin/stat2idf /projects/bigsim/weather/EnergyPlus/tmy3.new/all/USA_CA_San.Francisco.Intl.AP.724940_TMY3.stat
!
Site:WaterMainsTemperature,
 Correlation, !- Calculation Method
 , !- Temperature Schedule Name
 13.78333333333333, !- Annual Average Outdoor Air Temperature {C}
 7.1; !- Maximum Difference In Monthly Average Outdoor Air Temperatures {deltaC}

!- ===== ALL OBJECTS IN CLASS: SCHEDULETYPELIMITS =====

ScheduleTypeLimits,
 any number; !- Name

ScheduleTypeLimits,
 On/Off, !- Name
 0, !- Lower Limit Value
 1, !- Upper Limit Value
 Discrete; !- Numeric Type

ScheduleTypeLimits,
 control_type, !- Name
 0, !- Lower Limit Value
 4, !- Upper Limit Value
 Discrete; !- Numeric Type

ScheduleTypeLimits,
 fraction, !- Name
 0, !- Lower Limit Value
 1, !- Upper Limit Value
 Continuous; !- Numeric Type

ScheduleTypeLimits,
 Temperature, !- Name
 -60, !- Lower Limit Value
 200, !- Upper Limit Value
 Continuous; !- Numeric Type

ScheduleTypeLimits,
 myTemps, !- Name
 , !- Lower Limit Value
 , !- Upper Limit Value
 Continuous, !- Numeric Type
 Temperature; !- Unit Type

!- ===== ALL OBJECTS IN CLASS: SCHEDULE:DAY:HOURLY =====

!
! *** Occupancy and Lighting factors taken from NREL benchmark model ***
!

Schedule:Day:Hourly,
 OccupancyDay, !- Name
 Fraction, !- Schedule Type Limits Name
 1.00000, !- Hour 1
 1.00000, !- Hour 2
 1.00000, !- Hour 3
 1.00000, !- Hour 4
 1.00000, !- Hour 5
 1.00000, !- Hour 6
 1.00000, !- Hour 7
 0.88310, !- Hour 8
 0.40861, !- Hour 9

0.24189,	!- Hour 10
0.24189,	!- Hour 11
0.24189,	!- Hour 12
0.24189,	!- Hour 13
0.24189,	!- Hour 14
0.24189,	!- Hour 15
0.24189,	!- Hour 16
0.29498,	!- Hour 17
0.55310,	!- Hour 18
0.89693,	!- Hour 19
0.89693,	!- Hour 20
0.89693,	!- Hour 21
1.00000,	!- Hour 22
1.00000,	!- Hour 23
1.00000;	!- Hour 24

Schedule:Day:Hourly,

LightingDay,	!- Name
Fraction,	!- Schedule Type Limits Name
0.0625,	!- Hour 1
0.0625,	!- Hour 2
0.0625,	!- Hour 3
0.0625,	!- Hour 4
0.1875,	!- Hour 5
0.390625,	!- Hour 6
0.4375,	!- Hour 7
0.390625,	!- Hour 8
0.171875,	!- Hour 9
0.1171875,	!- Hour 10
0.1171875,	!- Hour 11
0.1171875,	!- Hour 12
0.1171875,	!- Hour 13
0.1171875,	!- Hour 14
0.1171875,	!- Hour 15
0.203125,	!- Hour 16
0.4375,	!- Hour 17
0.609375,	!- Hour 18
0.8203125,	!- Hour 19
0.984375,	!- Hour 20
1,	!- Hour 21
0.6875,	!- Hour 22
0.3828125,	!- Hour 23
0.15625;	!- Hour 24

Schedule:Day:Hourly,

LightingDay_EELighting_OccSensors,	!- Name
Fraction,	!- Schedule Type Limits Name
0.065170403,	!- Hour 1
0.065170403,	!- Hour 2
0.065170403,	!- Hour 3
0.065170403,	!- Hour 4
0.195511208,	!- Hour 5
0.407315016,	!- Hour 6
0.456192818,	!- Hour 7
0.407315016,	!- Hour 8
0.179218607,	!- Hour 9
0.122194505,	!- Hour 10
0.122194505,	!- Hour 11
0.122194505,	!- Hour 12
0.122194505,	!- Hour 13
0.122194505,	!- Hour 14
0.122194505,	!- Hour 15
0.211803808,	!- Hour 16
0.456192818,	!- Hour 17
0.635411425,	!- Hour 18
0.855361533,	!- Hour 19
0.947933128,	!- Hour 20
0.947933128,	!- Hour 21
0.716874428,	!- Hour 22
0.399168715,	!- Hour 23

0.162926006; !- Hour 24

Schedule:Day:Hourly,

LightingDay_EELighting_Garage_OccSensors, !- Name
Fraction, !- Schedule Type Limits Name
0.048125, !- Hour 1
0.048125, !- Hour 2
0.048125, !- Hour 3
0.048125, !- Hour 4
0.144375, !- Hour 5
0.30078125, !- Hour 6
0.336875, !- Hour 7
0.30078125, !- Hour 8
0.13234375, !- Hour 9
0.090234375, !- Hour 10
0.090234375, !- Hour 11
0.090234375, !- Hour 12
0.090234375, !- Hour 13
0.090234375, !- Hour 14
0.090234375, !- Hour 15
0.15640625, !- Hour 16
0.336875, !- Hour 17
0.46921875, !- Hour 18
0.631640625, !- Hour 19
0.7, !- Hour 20
0.7, !- Hour 21
0.529375, !- Hour 22
0.294765625, !- Hour 23
0.1203125; !- Hour 24

Schedule:Day:Hourly,

ExteriorLightingDay, !- Name
Fraction, !- Schedule Type Limits Name
1, !- Hour 1
1, !- Hour 2
1, !- Hour 3
1, !- Hour 4
1, !- Hour 5
1, !- Hour 6
0, !- Hour 7
0, !- Hour 8
0, !- Hour 9
0, !- Hour 10
0, !- Hour 11
0, !- Hour 12
0, !- Hour 13
0, !- Hour 14
0, !- Hour 15
0, !- Hour 16
0, !- Hour 17
0, !- Hour 18
1, !- Hour 19
1, !- Hour 20
1, !- Hour 21
1, !- Hour 22
1, !- Hour 23
1; !- Hour 24

Schedule:Day:Hourly,

LightingDay_EELighting, !- Name
Fraction, !- Schedule Type Limits Name
0.06875, !- Hour 1
0.06875, !- Hour 2
0.06875, !- Hour 3
0.06875, !- Hour 4
0.20625, !- Hour 5
0.4296875, !- Hour 6
0.48125, !- Hour 7
0.4296875, !- Hour 8
0.1890625, !- Hour 9

0.12890625, !- Hour 10
 0.12890625, !- Hour 11
 0.12890625, !- Hour 12
 0.12890625, !- Hour 13
 0.12890625, !- Hour 14
 0.12890625, !- Hour 15
 0.2234375, !- Hour 16
 0.48125, !- Hour 17
 0.6703125, !- Hour 18
 0.90234375, !- Hour 19
 1, !- Hour 20
 1, !- Hour 21
 0.75625, !- Hour 22
 0.42109375, !- Hour 23
 0.171875; !- Hour 24

Schedule:Day:Hourly,

RefrigeratorDay, !- Name
 Fraction, !- Schedule Type Limits Name
 0.8, !- Hour 1
 0.782696177062374, !- Hour 2
 0.765593561368209, !- Hour 3
 0.742857142857143, !- Hour 4
 0.731388329979879, !- Hour 5
 0.731388329979879, !- Hour 6
 0.759959758551308, !- Hour 7
 0.8, !- Hour 8
 0.817102615694165, !- Hour 9
 0.828571428571429, !- Hour 10
 0.8, !- Hour 11
 0.8, !- Hour 12
 0.839839034205231, !- Hour 13
 0.839839034205231, !- Hour 14
 0.828571428571429, !- Hour 15
 0.839839034205231, !- Hour 16
 0.885714285714286, !- Hour 17
 0.971428571428572, !- Hour 18
 1, !- Hour 19
 0.971428571428572, !- Hour 20
 0.942857142857143, !- Hour 21
 0.925553319919517, !- Hour 22
 0.885714285714286, !- Hour 23
 0.828571428571429; !- Hour 24

Schedule:Day:Hourly,

MiscPlugLoadDay, !- Name
 Fraction, !- Schedule Type Limits Name
 0.607490272373541, !- Hour 1
 0.559338521400778, !- Hour 2
 0.552853437094682, !- Hour 3
 0.545071335927367, !- Hour 4
 0.524481193255512, !- Hour 5
 0.585278858625162, !- Hour 6
 0.676232166018158, !- Hour 7
 0.718547341115435, !- Hour 8
 0.607490272373541, !- Hour 9
 0.517023346303502, !- Hour 10
 0.529182879377432, !- Hour 11
 0.529345006485084, !- Hour 12
 0.520428015564202, !- Hour 13
 0.538424124513619, !- Hour 14
 0.568741893644617, !- Hour 15
 0.600356679636835, !- Hour 16
 0.71011673151751, !- Hour 17
 0.862678339818418, !- Hour 18
 0.936608300907912, !- Hour 19
 0.966763942931258, !- Hour 20
 1, !- Hour 21
 0.976653696498055, !- Hour 22
 0.845168612191959, !- Hour 23

0.73443579766537; !- Hour 24

Schedule:Day:Hourly,
CookingRangeDay, !- Name
Fraction, !- Schedule Type Limits Name
0.04715848452508, !- Hour 1
0.04715848452508, !- Hour 2
0.0235458911419424, !- Hour 3
0.0235458911419424, !- Hour 4
0.04715848452508, !- Hour 5
0.0707043756670224, !- Hour 6
0.165088046958378, !- Hour 7
0.283017609391676, !- Hour 8
0.306563500533618, !- Hour 9
0.320771077908218, !- Hour 10
0.283017609391676, !- Hour 11
0.330176093916756, !- Hour 12
0.377334578441836, !- Hour 13
0.306563500533618, !- Hour 14
0.292422625400213, !- Hour 15
0.377334578441836, !- Hour 16
0.613193703308431, !- Hour 17
1, !- Hour 18
0.778348452508004, !- Hour 19
0.400947171824973, !- Hour 20
0.235859124866596, !- Hour 21
0.165088046958378, !- Hour 22
0.103721985058698, !- Hour 23
0.0707043756670224; !- Hour 24

Schedule:Day:Hourly,
DishwasherWeekday, !- Name
Fraction, !- Schedule Type Limits Name
0.115858456545258, !- Hour 1
0.0514758745041469, !- Hour 2
0.0385691669671833, !- Hour 3
0.0257379372520738, !- Hour 4
0.0257379372520738, !- Hour 5
0.0772138117562207, !- Hour 6
0.154503101334295, !- Hour 7
0.23179239091237, !- Hour 8
0.437922322394519, !- Hour 9
0.489398196898665, !- Hour 10
0.425015614857555, !- Hour 11
0.360633032816445, !- Hour 12
0.309081680490443, !- Hour 13
0.347726325279481, !- Hour 14
0.28334374323837, !- Hour 15
0.270437035701407, !- Hour 16
0.28334374323837, !- Hour 17
0.373464262531554, !- Hour 18
0.656883483591778, !- Hour 19
0.837199999999997, !- Hour 20
0.682621420843852, !- Hour 21
0.502304904435629, !- Hour 22
0.334819617742518, !- Hour 23
0.23179239091237; !- Hour 24

Schedule:Day:Hourly,
DishwasherWeekend, !- Name
Fraction, !- Schedule Type Limits Name
0.124134060584205, !- Hour 1
0.0551527226830148, !- Hour 2
0.0413241074648395, !- Hour 3
0.0275763613415074, !- Hour 4
0.0275763613415074, !- Hour 5
0.0827290840245222, !- Hour 6
0.165539037143888, !- Hour 7
0.248348990263252, !- Hour 8
0.469202488279841, !- Hour 9

0.524355210962856, !- Hour 10
 0.455373873061666, !- Hour 11
 0.386392535160476, !- Hour 12
 0.331158943382618, !- Hour 13
 0.3725639199423, !- Hour 14
 0.303582582041111, !- Hour 15
 0.289753966822935, !- Hour 16
 0.303582582041111, !- Hour 17
 0.400140281283808, !- Hour 18
 0.703803732419762, !- Hour 19
 0.897, !- Hour 20
 0.731380093761269, !- Hour 21
 0.538183826181031, !- Hour 22
 0.358735304724125, !- Hour 23
 0.248348990263252; !- Hour 24

Schedule:Day:Hourly,

DishwasherVacation, !- Name
 Fraction, !- Schedule Type Limits Name
 0, !- Hour 1
 0, !- Hour 2
 0, !- Hour 3
 0, !- Hour 4
 0, !- Hour 5
 0, !- Hour 6
 0, !- Hour 7
 0, !- Hour 8
 0, !- Hour 9
 0, !- Hour 10
 0, !- Hour 11
 0, !- Hour 12
 0, !- Hour 13
 0, !- Hour 14
 0, !- Hour 15
 0, !- Hour 16
 0, !- Hour 17
 0, !- Hour 18
 0, !- Hour 19
 0, !- Hour 20
 0, !- Hour 21
 0, !- Hour 22
 0, !- Hour 23
 0; !- Hour 24

Schedule:Day:Hourly,

ClothesWasherWeekday, !- Name
 Fraction, !- Schedule Type Limits Name
 0.0796474528960224, !- Hour 1
 0.0637009071877185, !- Hour 2
 0.0318078157711097, !- Hour 3
 0.0318078157711097, !- Hour 4
 0.0637009071877185, !- Hour 5
 0.0955939986043261, !- Hour 6
 0.191187997208654, !- Hour 7
 0.414354361479414, !- Hour 8
 0.62157418004187, !- Hour 9
 0.7332, !- Hour 10
 0.717253454291697, !- Hour 11
 0.637520725750175, !- Hour 12
 0.573734542916957, !- Hour 13
 0.510033635729239, !- Hour 14
 0.446247452896022, !- Hour 15
 0.414354361479414, !- Hour 16
 0.430300907187718, !- Hour 17
 0.414354361479414, !- Hour 18
 0.414354361479414, !- Hour 19
 0.414354361479414, !- Hour 20
 0.414354361479414, !- Hour 21
 0.398407815771109, !- Hour 22
 0.270920725750174, !- Hour 23

0.143433635729239; !- Hour 24

Schedule:Day:Hourly,

ClothesWasherWeekend, !- Name
Fraction, !- Schedule Type Limits Name
0.0974410327983253, !- Hour 1
0.0779319609211445, !- Hour 2
0.038913817166783, !- Hour 3
0.038913817166783, !- Hour 4
0.0779319609211445, !- Hour 5
0.116950104675506, !- Hour 6
0.233900209351012, !- Hour 7
0.506922889043964, !- Hour 8
0.760436496859734, !- Hour 9
0.897, !- Hour 10
0.877490928122819, !- Hour 11
0.779945568736915, !- Hour 12
0.701909281228193, !- Hour 13
0.623977320307048, !- Hour 14
0.545941032798325, !- Hour 15
0.506922889043964, !- Hour 16
0.526431960921144, !- Hour 17
0.506922889043964, !- Hour 18
0.506922889043964, !- Hour 19
0.506922889043964, !- Hour 20
0.506922889043964, !- Hour 21
0.487413817166783, !- Hour 22
0.331445568736915, !- Hour 23
0.175477320307048; !- Hour 24

Schedule:Day:Hourly,

ClothesWasherVacation, !- Name
Fraction, !- Schedule Type Limits Name
0, !- Hour 1
0, !- Hour 2
0, !- Hour 3
0, !- Hour 4
0, !- Hour 5
0, !- Hour 6
0, !- Hour 7
0, !- Hour 8
0, !- Hour 9
0, !- Hour 10
0, !- Hour 11
0, !- Hour 12
0, !- Hour 13
0, !- Hour 14
0, !- Hour 15
0, !- Hour 16
0, !- Hour 17
0, !- Hour 18
0, !- Hour 19
0, !- Hour 20
0, !- Hour 21
0, !- Hour 22
0, !- Hour 23
0; !- Hour 24

Schedule:Day:Hourly,

dhw_profile_day, !- Name
fraction, !- Schedule Type Limits Name
0.006, !- Hour 1
0.003, !- Hour 2
0.001, !- Hour 3
0.001, !- Hour 4
0.003, !- Hour 5
0.021, !- Hour 6
0.075, !- Hour 7
0.079, !- Hour 8
0.076, !- Hour 9

0.067, !- Hour 10
 0.061, !- Hour 11
 0.05, !- Hour 12
 0.042, !- Hour 13
 0.038, !- Hour 14
 0.033, !- Hour 15
 0.038, !- Hour 16
 0.043, !- Hour 17
 0.058, !- Hour 18
 0.068, !- Hour 19
 0.065, !- Hour 20
 0.06, !- Hour 21
 0.047, !- Hour 22
 0.041, !- Hour 23
 0.024; !- Hour 24

Schedule:Day:Hourly,

ClothesDryerWeekday, !- Name
 Fraction, !- Schedule Type Limits Name
 0.0996818663838815, !- Hour 1
 0.0598091198303289, !- Hour 2
 0.0398727465535526, !- Hour 3
 0.0199363732767763, !- Hour 4
 0.0398727465535526, !- Hour 5
 0.0598091198303289, !- Hour 6
 0.15949098621421, !- Hour 7
 0.31898197242842, !- Hour 8
 0.486427370202556, !- Hour 9
 0.685791102970318, !- Hour 10
 0.785472969354199, !- Hour 11
 0.817391304347826, !- Hour 12
 0.745600222800647, !- Hour 13
 0.677836691410393, !- Hour 14
 0.610073160020138, !- Hour 15
 0.578154825026511, !- Hour 16
 0.558218451749735, !- Hour 17
 0.546236490032885, !- Hour 18
 0.518345705196183, !- Hour 19
 0.510391293636256, !- Hour 20
 0.526300116756109, !- Hour 21
 0.546236490032885, !- Hour 22
 0.438600212089077, !- Hour 23
 0.239236479321316; !- Hour 24

Schedule:Day:Hourly,

ClothesDryerWeekend, !- Name
 Fraction, !- Schedule Type Limits Name
 0.121951219512195, !- Hour 1
 0.0731707317073171, !- Hour 2
 0.0487804878048781, !- Hour 3
 0.024390243902439, !- Hour 4
 0.0487804878048781, !- Hour 5
 0.0731707317073171, !- Hour 6
 0.195121951219512, !- Hour 7
 0.390243902439024, !- Hour 8
 0.59509731460951, !- Hour 9
 0.8389997536339, !- Hour 10
 0.960950973146095, !- Hour 11
 1, !- Hour 12
 0.912170485341217, !- Hour 13
 0.829268292682927, !- Hour 14
 0.746366100024637, !- Hour 15
 0.707317073170732, !- Hour 16
 0.682926829268293, !- Hour 17
 0.668268046316827, !- Hour 18
 0.634146341463415, !- Hour 19
 0.624414880512441, !- Hour 20
 0.643877802414388, !- Hour 21
 0.668268046316827, !- Hour 22
 0.536585365853659, !- Hour 23

0.292682926829268; !- Hour 24

Schedule:Day:Hourly,

ClothesDryerVacation, !- Name
Fraction, !- Schedule Type Limits Name
0, !- Hour 1
0, !- Hour 2
0, !- Hour 3
0, !- Hour 4
0, !- Hour 5
0, !- Hour 6
0, !- Hour 7
0, !- Hour 8
0, !- Hour 9
0, !- Hour 10
0, !- Hour 11
0, !- Hour 12
0, !- Hour 13
0, !- Hour 14
0, !- Hour 15
0, !- Hour 16
0, !- Hour 17
0, !- Hour 18
0, !- Hour 19
0, !- Hour 20
0, !- Hour 21
0, !- Hour 22
0, !- Hour 23
0; !- Hour 24

Schedule:Day:Hourly,

SinksWeekday, !- Name
Fraction, !- Schedule Type Limits Name
0.160052687884461, !- Hour 1
0.0767484287242575, !- Hour 2
0.0575330636533838, !- Hour 3
0.0575330636533838, !- Hour 4
0.0767484287242575, !- Hour 5
0.204926216902916, !- Hour 6
0.480271095212623, !- Hour 7
0.704412677186414, !- Hour 8
0.742843407328163, !- Hour 9
0.697969878309708, !- Hour 10
0.614778650708746, !- Hour 11
0.563462322813587, !- Hour 12
0.550689756619418, !- Hour 13
0.512259026477668, !- Hour 14
0.467385497459214, !- Hour 15
0.486600862530089, !- Hour 16
0.544246957742712, !- Hour 17
0.736400608451458, !- Hour 18
0.84525, !- Hour 19
0.781274137469912, !- Hour 20
0.640323783097085, !- Hour 21
0.544246957742712, !- Hour 22
0.44828316394758, !- Hour 23
0.307332809574753; !- Hour 24

Schedule:Day:Hourly,

SinksWeekend, !- Name
Fraction, !- Schedule Type Limits Name
0.169851832040653, !- Hour 1
0.0814473121155389, !- Hour 2
0.0610554961219577, !- Hour 3
0.0610554961219577, !- Hour 4
0.0814473121155389, !- Hour 5
0.217472719978604, !- Hour 6
0.509675447980744, !- Hour 7
0.747539983952929, !- Hour 8
0.788323615940091, !- Hour 9

0.74070272800214, !- Hour 10
 0.652418159935812, !- Hour 11
 0.597960016047071, !- Hour 12
 0.58440545600428, !- Hour 13
 0.543621824017117, !- Hour 14
 0.496000936079166, !- Hour 15
 0.516392752072746, !- Hour 16
 0.57756820005349, !- Hour 17
 0.781486359989302, !- Hour 18
 0.897, !- Hour 19
 0.829107247927254, !- Hour 20
 0.679527280021396, !- Hour 21
 0.57756820005349, !- Hour 22
 0.47572907194437, !- Hour 23
 0.326149104038513; !- Hour 24

Schedule:Day:Hourly,

SinksVacation, Fraction,	!- Name !- Schedule Type Limits Name
0,	!- Hour 1
0,	!- Hour 2
0,	!- Hour 3
0,	!- Hour 4
0,	!- Hour 5
0,	!- Hour 6
0,	!- Hour 7
0,	!- Hour 8
0,	!- Hour 9
0,	!- Hour 10
0,	!- Hour 11
0,	!- Hour 12
0,	!- Hour 13
0,	!- Hour 14
0,	!- Hour 15
0,	!- Hour 16
0,	!- Hour 17
0,	!- Hour 18
0,	!- Hour 19
0,	!- Hour 20
0,	!- Hour 21
0,	!- Hour 22
0,	!- Hour 23
0;	!- Hour 24

Schedule:Day:Hourly,

ShowersWeekday, Fraction,	!- Name !- Schedule Type Limits Name
0.076044361285933,	!- Hour 1
0.0345332539547544,	!- Hour 2
0.0242088790610646,	!- Hour 3
0.0345332539547544,	!- Hour 4
0.0968355162442594,	!- Hour 5
0.366693315189659,	!- Hour 6
0.837199999999997,	!- Hour 7
0.830293349209049,	!- Hour 8
0.674644497363497,	!- Hour 9
0.529320020411635,	!- Hour 10
0.425506650790951,	!- Hour 11
0.335577785337642,	!- Hour 12
0.242159993196122,	!- Hour 13
0.207555536655894,	!- Hour 14
0.179857730906617,	!- Hour 15
0.186764381697567,	!- Hour 16
0.214462187446844,	!- Hour 17
0.27676444973635,	!- Hour 18
0.300973328797414,	!- Hour 19
0.300973328797414,	!- Hour 20
0.297484402109202,	!- Hour 21
0.290577751318252,	!- Hour 22
0.207555536655894,	!- Hour 23

0.148742201054601; !- Hour 24

Schedule:Day:Hourly,
ShowersWeekend, !- Name
Fraction, !- Schedule Type Limits Name
0.0814761013777853, !- Hour 1
0.0369999149515223, !- Hour 2
0.0259380847082837, !- Hour 3
0.0369999149515223, !- Hour 4
0.103752338833135, !- Hour 5
0.392885694846062, !- Hour 6
0.897, !- Hour 7
0.889600017009696, !- Hour 8
0.722833390032318, !- Hour 9
0.56712859329818, !- Hour 10
0.455899982990304, !- Hour 11
0.359547627147474, !- Hour 12
0.259457135567274, !- Hour 13
0.222380932131315, !- Hour 14
0.19270471168566, !- Hour 15
0.200104694675966, !- Hour 16
0.229780915121619, !- Hour 17
0.296533339003232, !- Hour 18
0.322471423711515, !- Hour 19
0.322471423711515, !- Hour 20
0.318733287974145, !- Hour 21
0.311333304983841, !- Hour 22
0.222380932131315, !- Hour 23
0.159366643987073; !- Hour 24

Schedule:Day:Hourly,
ShowersVacation, !- Name
Fraction, !- Schedule Type Limits Name
0, !- Hour 1
0, !- Hour 2
0, !- Hour 3
0, !- Hour 4
0, !- Hour 5
0, !- Hour 6
0, !- Hour 7
0, !- Hour 8
0, !- Hour 9
0, !- Hour 10
0, !- Hour 11
0, !- Hour 12
0, !- Hour 13
0, !- Hour 14
0, !- Hour 15
0, !- Hour 16
0, !- Hour 17
0, !- Hour 18
0, !- Hour 19
0, !- Hour 20
0, !- Hour 21
0, !- Hour 22
0, !- Hour 23
0; !- Hour 24

Schedule:Day:Hourly,
BathsWeekday, !- Name
Fraction, !- Schedule Type Limits Name
0.0492758944582015, !- Hour 1
0.0246379472291013, !- Hour 2
0.0246379472291013, !- Hour 3
0.0246379472291013, !- Hour 4
0.0492758944582015, !- Hour 5
0.123189736145504, !- Hour 6
0.295719195628041, !- Hour 7
0.369633037315344, !- Hour 8
0.418908931773546, !- Hour 9

0.369633037315344, !- Hour 10
 0.295719195628041, !- Hour 11
 0.221741525061908, !- Hour 12
 0.197103577832807, !- Hour 13
 0.147827683374605, !- Hour 14
 0.147827683374605, !- Hour 15
 0.147827683374605, !- Hour 16
 0.24644330116984, !- Hour 17
 0.295719195628041, !- Hour 18
 0.492886602339681, !- Hour 19
 0.640714285714285, !- Hour 20
 0.640714285714285, !- Hour 21
 0.492886602339681, !- Hour 22
 0.418908931773546, !- Hour 23
 0.24644330116984; !- Hour 24

Schedule:Day:Hourly,

BathsWeekend, !- Name
 Fraction, !- Schedule Type Limits Name
 0.0689862522414824, !- Hour 1
 0.0344931261207412, !- Hour 2
 0.0344931261207412, !- Hour 3
 0.0344931261207412, !- Hour 4
 0.0689862522414824, !- Hour 5
 0.172465630603706, !- Hour 6
 0.414006873879259, !- Hour 7
 0.517486252241482, !- Hour 8
 0.586472504482965, !- Hour 9
 0.517486252241482, !- Hour 10
 0.414006873879259, !- Hour 11
 0.310438135086671, !- Hour 12
 0.27594500896593, !- Hour 13
 0.206958756724447, !- Hour 14
 0.206958756724447, !- Hour 15
 0.206958756724447, !- Hour 16
 0.345020621637776, !- Hour 17
 0.414006873879259, !- Hour 18
 0.690041243275553, !- Hour 19
 0.897, !- Hour 20
 0.897, !- Hour 21
 0.690041243275553, !- Hour 22
 0.586472504482965, !- Hour 23
 0.345020621637776; !- Hour 24

Schedule:Day:Hourly,

BathsVacation, !- Name
 Fraction, !- Schedule Type Limits Name
 0, !- Hour 1
 0, !- Hour 2
 0, !- Hour 3
 0, !- Hour 4
 0, !- Hour 5
 0, !- Hour 6
 0, !- Hour 7
 0, !- Hour 8
 0, !- Hour 9
 0, !- Hour 10
 0, !- Hour 11
 0, !- Hour 12
 0, !- Hour 13
 0, !- Hour 14
 0, !- Hour 15
 0, !- Hour 16
 0, !- Hour 17
 0, !- Hour 18
 0, !- Hour 19
 0, !- Hour 20
 0, !- Hour 21
 0, !- Hour 22
 0, !- Hour 23

0; !- Hour 24

Schedule:Day:Hourly,
DHWDistDay, !- Name
Fraction, !- Schedule Type Limits Name
0.142553149370226, !- Hour 1
0.0764866759550322, !- Hour 2
0.0516611840277747, !- Hour 3
0.0584039294664903, !- Hour 4
0.121469841058603, !- Hour 5
0.366180268451559, !- Hour 6
0.833258955739606, !- Hour 7
0.999999987228391, !- Hour 8
0.995483090581232, !- Hour 9
0.916474762598139, !- Hour 10
0.800898393293271, !- Hour 11
0.682564390940485, !- Hour 12
0.590066600109555, !- Hour 13
0.522371887032053, !- Hour 14
0.466005974303267, !- Hour 15
0.46573704253214, !- Hour 16
0.527514234916123, !- Hour 17
0.637905701335668, !- Hour 18
0.727588642310272, !- Hour 19
0.732551127624518, !- Hour 20
0.681468471942116, !- Hour 21
0.610804704600985, !- Hour 22
0.464292318119835, !- Hour 23
0.299867724445383; !- Hour 24

!- ===== ALL OBJECTS IN CLASS: SCHEDULE:WEEK:COMPACT =====

Schedule:Week:Compact,
RefrigeratorWeek, !- Name
For: AllDays, !- DayType List 1
RefrigeratorDay; !- Schedule:Day Name 1

Schedule:Week:Compact,
MiscPlugLoadWeek, !- Name
For: AllDays, !- DayType List 1
MiscPlugLoadDay; !- Schedule:Day Name 1

Schedule:Week:Compact,
CookingRangeWeek, !- Name
For: AllDays, !- DayType List 1
CookingRangeDay; !- Schedule:Day Name 1

Schedule:Week:Compact,
DishwasherWeek, !- Name
For: Weekdays, !- DayType List 1
DishwasherWeekday, !- Schedule:Day Name 1
For: CustomDay1, !- DayType List 2
DishwasherVacation, !- Schedule:Day Name 2
For: AllOtherDays, !- DayType List 3
DishwasherWeekend; !- Schedule:Day Name 3

Schedule:Week:Compact,
ClothesWasherWeek, !- Name
For: Weekdays, !- DayType List 1
ClothesWasherWeekday, !- Schedule:Day Name 1
For: CustomDay1, !- DayType List 2
ClothesWasherVacation, !- Schedule:Day Name 2
For: AllOtherDays, !- DayType List 3
ClothesWasherWeekend; !- Schedule:Day Name 3

Schedule:Week:Compact,
dhw_profile_week, !- Name
AllDays, !- DayType List 1
dhw_profile_day; !- Schedule:Day Name 1

Schedule:Week:Compact,
ClothesDryerWeek, !- Name
For: Weekdays, !- DayType List 1
ClothesDryerWeekday, !- Schedule:Day Name 1
For: CustomDay1, !- DayType List 2
ClothesDryerVacation, !- Schedule:Day Name 2
For: AllOtherDays, !- DayType List 3
ClothesDryerWeekend; !- Schedule:Day Name 3

Schedule:Week:Compact,
SinksWeek, !- Name
For: Weekdays, !- DayType List 1
SinksWeekday, !- Schedule:Day Name 1
For: CustomDay1, !- DayType List 2
SinksVacation, !- Schedule:Day Name 2
For: AllOtherDays, !- DayType List 3
SinksWeekend; !- Schedule:Day Name 3

Schedule:Week:Compact,
ShowersWeek, !- Name
For: Weekdays, !- DayType List 1
ShowersWeekday, !- Schedule:Day Name 1
For: CustomDay1, !- DayType List 2
ShowersVacation, !- Schedule:Day Name 2
For: AllOtherDays, !- DayType List 3
ShowersWeekend; !- Schedule:Day Name 3

Schedule:Week:Compact,
BathsWeek, !- Name
For: Weekdays, !- DayType List 1
BathsWeekday, !- Schedule:Day Name 1
For: CustomDay1, !- DayType List 2
BathsVacation, !- Schedule:Day Name 2
For: AllOtherDays, !- DayType List 3
BathsWeekend; !- Schedule:Day Name 3

Schedule:Week:Compact,
DHWDistWeek, !- Name
For: AllDays, !- DayType List 1
DHWDistDay; !- Schedule:Day Name 1

Schedule:Week:Compact,
OccupancyWeek, !- Name
AllDays, !- DayType List 1
OccupancyDay; !- Schedule:Day Name 1

Schedule:Week:Compact,
LightingProfileWeek, !- Name
For: AllDays, !- DayType List 1
LightingDay; !- Schedule:Day Name 1

Schedule:Week:Compact,
LightingProfileWeek_EELighting, !- Name
For: AllDays, !- DayType List 1
LightingDay_EELighting; !- Schedule:Day Name 1

Schedule:Week:Compact,
LightingProfileWeek_EELighting_interior_controls, !- Name
For: AllDays, !- DayType List 1
LightingDay_EELighting_OccSensors; !- Schedule:Day Name 1

Schedule:Week:Compact,
LightingProfileWeek_EELighting_garage_controls, !- Name
For: AllDays, !- DayType List 1
LightingDay_EELighting_Garage_OccSensors; !- Schedule:Day Name 1

Schedule:Week:Compact,
ExteriorLightingProfileWeek, !- Name
For: AllDays, !- DayType List 1

ExteriorLightingDay; !- Schedule:Day Name 1

!- ===== ALL OBJECTS IN CLASS: SCHEDULE:YEAR =====

Schedule:Year,

Occupancy, !- Name
Fraction, !- Schedule Type Limits Name
OccupancyWeek, !- Schedule:Week Name 1
1, !- Start Month 1
1, !- Start Day 1
12, !- End Month 1
31; !- End Day 1

Schedule:Year,

InteriorLighting, !- Name
Fraction, !- Schedule Type Limits Name
LightingProfileWeek, !- Schedule:Week Name 1
1, !- Start Month 1
1, !- Start Day 1
12, !- End Month 1
31; !- End Day 1

Schedule:Year,

InteriorLightingHE, !- Name
Fraction, !- Schedule Type Limits Name
LightingProfileWeek_EELighting, !- Schedule:Week Name 1
1, !- Start Month 1
1, !- Start Day 1
12, !- End Month 1
31; !- End Day 1

Schedule:Year,

InteriorLightingHE_OS, !- Name
Fraction, !- Schedule Type Limits Name
LightingProfileWeek_EELighting_interior_controls, !- Schedule:Week Name 1
1, !- Start Month 1
1, !- Start Day 1
12, !- End Month 1
31; !- End Day 1

Schedule:Year,

InteriorLightingHE_VS, !- Name
Fraction, !- Schedule Type Limits Name
LightingProfileWeek_EELighting_garage_controls, !- Schedule:Week Name 1
1, !- Start Month 1
1, !- Start Day 1
12, !- End Month 1
31; !- End Day 1

Schedule:Year,

ExteriorLighting, !- Name
Fraction, !- Schedule Type Limits Name
ExteriorLightingProfileWeek, !- Schedule:Week Name 1
1, !- Start Month 1
1, !- Start Day 1
12, !- End Month 1
31; !- End Day 1

Schedule:Year,

Refrigerator, !- Name
Fraction, !- Schedule Type Limits Name
RefrigeratorWeek, !- Schedule:Week Name 1
1, !- Start Month 1
1, !- Start Day 1
12, !- End Month 1
31; !- End Day 1

Schedule:Year,

MiscPlugLoad, !- Name

Fraction, !- Schedule Type Limits Name
MiscPlugLoadWeek, !- Schedule:Week Name 1
1, !- Start Month 1
1, !- Start Day 1
12, !- End Month 1
31; !- End Day 1

Schedule:Year,
CookingRange, !- Name
Fraction, !- Schedule Type Limits Name
CookingRangeWeek, !- Schedule:Week Name 1
1, !- Start Month 1
1, !- Start Day 1
12, !- End Month 1
31; !- End Day 1

Schedule:Year,
dhw_sch, !- Name
fraction, !- Schedule Type Limits Name
dhw_profile_week, !- Schedule:Week Name 1
1, !- Start Month 1
1, !- Start Day 1
12, !- End Month 1
31; !- End Day 1

Schedule:Year,
Dishwasher, !- Name
Fraction, !- Schedule Type Limits Name
DishwasherWeek, !- Schedule:Week Name 1
1, !- Start Month 1
1, !- Start Day 1
12, !- End Month 1
31; !- End Day 1

Schedule:Year,
ClothesWasher, !- Name
Fraction, !- Schedule Type Limits Name
ClothesWasherWeek, !- Schedule:Week Name 1
1, !- Start Month 1
1, !- Start Day 1
12, !- End Month 1
31; !- End Day 1

Schedule:Year,
ClothesDryer, !- Name
Fraction, !- Schedule Type Limits Name
ClothesDryerWeek, !- Schedule:Week Name 1
1, !- Start Month 1
1, !- Start Day 1
12, !- End Month 1
31; !- End Day 1

Schedule:Year,
Sinks, !- Name
Fraction, !- Schedule Type Limits Name
SinksWeek, !- Schedule:Week Name 1
1, !- Start Month 1
1, !- Start Day 1
12, !- End Month 1
31; !- End Day 1

Schedule:Year,
Showers, !- Name
Fraction, !- Schedule Type Limits Name
ShowersWeek, !- Schedule:Week Name 1
1, !- Start Month 1
1, !- Start Day 1
12, !- End Month 1
31; !- End Day 1

Schedule:Year,
 Baths, !- Name
 Fraction, !- Schedule Type Limits Name
 BathsWeek, !- Schedule:Week Name 1
 1, !- Start Month 1
 1, !- Start Day 1
 12, !- End Month 1
 31; !- End Day 1

Schedule:Year,
 DHWDist, !- Name
 Fraction, !- Schedule Type Limits Name
 DHWDistWeek, !- Schedule:Week Name 1
 1, !- Start Month 1
 1, !- Start Day 1
 12, !- End Month 1
 31; !- End Day 1

!- ===== ALL OBJECTS IN CLASS: SCHEDULE:COMPACT =====

Schedule:Compact,
 BA_shower_sch, !- Name
 any number, !- Schedule Type Limits Name
 Through: 12/31, !- Field 1
 For: AllDays, !- Field 2
 Until: 1:00, !- Field 3
 0.0020472197777778, !- Field 4
 Until: 2:00, !- Field 5
 0.0009305544444445, !- Field 6
 Until: 3:00, !- Field 7
 0.00055833266666667, !- Field 8
 Until: 4:00, !- Field 9
 0.0009305544444445, !- Field 10
 Until: 5:00, !- Field 11
 0.00260555244444444, !- Field 12
 Until: 6:00, !- Field 13
 0.00967776622222222, !- Field 14
 Until: 7:00, !- Field 15
 0.0219610848888889, !- Field 16
 Until: 8:00, !- Field 17
 0.021774974, !- Field 18
 Until: 9:00, !- Field 19
 0.0176805344444444, !- Field 20
 Until: 10:00, !- Field 21
 0.0137722057777778, !- Field 22
 Until: 11:00, !- Field 23
 0.0111666533333333, !- Field 24
 Until: 12:00, !- Field 25
 0.0087472117777778, !- Field 26
 Until: 13:00, !- Field 27
 0.00632777022222222, !- Field 28
 Until: 14:00, !- Field 29
 0.0053972157777778, !- Field 30
 Until: 15:00, !- Field 31
 0.00483888311111111, !- Field 32
 Until: 16:00, !- Field 33
 0.00465277222222222, !- Field 34
 Until: 17:00, !- Field 35
 0.00558332666666667, !- Field 36
 Until: 18:00, !- Field 37
 0.00725832466666667, !- Field 38
 Until: 19:00, !- Field 39
 0.00781665733333333, !- Field 40
 Until: 20:00, !- Field 41
 0.00781665733333333, !- Field 42
 Until: 21:00, !- Field 43
 0.00781665733333333, !- Field 44
 Until: 22:00, !- Field 45
 0.00763054644444444, !- Field 46

Until: 23:00, !- Field 47
0.00539721577777778, !- Field 48
Until: 24:00, !- Field 49
0.00390832866666667; !- Field 50

Schedule:Compact,

BA_bath_sch, !- Name
any number, !- Schedule Type Limits Name
Through: 12/31, !- Field 1
For: AllDays, !- Field 2
Until: 1:00, !- Field 3
0.00019054454545454545, !- Field 4
Until: 2:00, !- Field 5
9.52722727272727e-05, !- Field 6
Until: 3:00, !- Field 7
9.52722727272727e-05, !- Field 8
Until: 4:00, !- Field 9
9.52722727272727e-05, !- Field 10
Until: 5:00, !- Field 11
0.00019054454545454545, !- Field 12
Until: 6:00, !- Field 13
0.00045254329545454545, !- Field 14
Until: 7:00, !- Field 15
0.00109563113636364, !- Field 16
Until: 8:00, !- Field 17
0.0013814479545454545, !- Field 18
Until: 9:00, !- Field 19
0.0015719925, !- Field 20
Until: 10:00, !- Field 21
0.0013814479545454545, !- Field 22
Until: 11:00, !- Field 23
0.00109563113636364, !- Field 24
Until: 12:00, !- Field 25
0.000833632386363636, !- Field 26
Until: 13:00, !- Field 27
0.000738360113636364, !- Field 28
Until: 14:00, !- Field 29
0.000547815568181818, !- Field 30
Until: 15:00, !- Field 31
0.000547815568181818, !- Field 32
Until: 16:00, !- Field 33
0.000547815568181818, !- Field 34
Until: 17:00, !- Field 35
0.000928904659090909, !- Field 36
Until: 18:00, !- Field 37
0.00109563113636364, !- Field 38
Until: 19:00, !- Field 39
0.00183399125, !- Field 40
Until: 20:00, !- Field 41
0.00238180681818182, !- Field 42
Until: 21:00, !- Field 43
0.00238180681818182, !- Field 44
Until: 22:00, !- Field 45
0.00183399125, !- Field 46
Until: 23:00, !- Field 47
0.0015719925, !- Field 48
Until: 24:00, !- Field 49
0.000928904659090909; !- Field 50

Schedule:Compact,

BA_sink_sch, !- Name
any number, !- Schedule Type Limits Name
Through: 12/31, !- Field 1
For: AllDays, !- Field 2
Until: 1:00, !- Field 3
0.00458624035087719, !- Field 4
Until: 2:00, !- Field 5
0.0022931201754386, !- Field 6
Until: 3:00, !- Field 7
0.00163794298245614, !- Field 8

Until: 4:00, !- Field 9
 0.00163794298245614, !- Field 10
 Until: 5:00, !- Field 11
 0.0022931201754386, !- Field 12
 Until: 6:00, !- Field 13
 0.0058965947368421, !- Field 14
 Until: 7:00, !- Field 15
 0.0137587210526316, !- Field 16
 Until: 8:00, !- Field 17
 0.0203104929824561, !- Field 18
 Until: 9:00, !- Field 19
 0.0216208473684211, !- Field 20
 Until: 10:00, !- Field 21
 0.0203104929824561, !- Field 22
 Until: 11:00, !- Field 23
 0.0176897842105263, !- Field 24
 Until: 12:00, !- Field 25
 0.0163794298245614, !- Field 26
 Until: 13:00, !- Field 27
 0.0160518412280702, !- Field 28
 Until: 14:00, !- Field 29
 0.0147414868421053, !- Field 30
 Until: 15:00, !- Field 31
 0.0140863096491228, !- Field 32
 Until: 16:00, !- Field 33
 0.0134311324561404, !- Field 34
 Until: 17:00, !- Field 35
 0.0157242526315789, !- Field 36
 Until: 18:00, !- Field 37
 0.0212932587719298, !- Field 38
 Until: 19:00, !- Field 39
 0.0245691447368421, !- Field 40
 Until: 20:00, !- Field 41
 0.0226036131578947, !- Field 42
 Until: 21:00, !- Field 43
 0.01867255, !- Field 44
 Until: 22:00, !- Field 45
 0.0157242526315789, !- Field 46
 Until: 23:00, !- Field 47
 0.0131035438596491, !- Field 48
 Until: 24:00, !- Field 49
 0.00884489210526316; !- Field 50

Schedule:Compact,
 activity_sch, !- Name
 any number, !- Schedule Type Limits Name
 Through: 12/31, !- Field 1
 For: AllDays, !- Field 2
 Until: 24:00, !- Field 3
 117.28; !- Field 4

Schedule:Compact,
 inf_sch, !- Name
 any number, !- Schedule Type Limits Name
 Through: 12/31, !- Field 1
 For: AllDays, !- Field 2
 Until: 24:00, !- Field 3
 1; !- Field 4

Schedule:Compact,
 zone_control_type, !- Name
 control_type, !- Schedule Type Limits Name
 Through: 12/31, !- Field 1
 For: AllDays, !- Field 2
 Until 24:00, !- Field 3
 4; !- Field 4

Schedule:Compact,
 shading_2012iecc, !- Name
 fraction, !- Schedule Type Limits Name

Through: 12/31, !- Field 1
For: AllDays, !- Field 2
Until 24:00, !- Field 3
0; !- Field 4

Schedule:Compact,
shading_2009iecc, !- Name
any number, !- Schedule Type Limits Name
Through: 5/30, !- Field 1
For: AllDays, !- Field 2
Until 24:00, !- Field 3
0.85, !- Field 4
Through: 8/31, !- Field 5
For: AllDays, !- Field 6
Until 24:00, !- Field 7
0.7, !- Field 8
Through: 12/31, !- Field 9
For: AllDays, !- Field 10
Until: 24:00, !- Field 11
0.85; !- Field 12

Schedule:Compact,
dhw_setpt, !- Name
Temperature, !- Schedule Type Limits Name
Through: 12/31, !- Field 1
For: AllDays, !- Field 2
Until 24:00, !- Field 3
48; !- Field 4

Schedule:Compact,
dhw_setpt_hpwh, !- Name
Temperature, !- Schedule Type Limits Name
Through: 12/31, !- Field 1
For: AllDays, !- Field 2
Until 24:00, !- Field 3
44; !- Field 4

Schedule:Compact,
Supply-Air-Temp-Sch, !- Name
Temperature, !- Schedule Type Limits Name
Through: 12/31, !- Field 1
For: AllDays, !- Field 2
Until: 24:00, !- Field 3
12; !- Field 4

Schedule:Compact,
always_avail, !- Name
On/Off, !- Schedule Type Limits Name
Through: 12/31, !- Field 1
For: AllDays, !- Field 2
Until: 24:00, !- Field 3
1; !- Field 4

Schedule:Compact,
always_off, !- Name
On/Off, !- Schedule Type Limits Name
Through: 12/31, !- Field 1
For: AllDays, !- Field 2
Until: 24:00, !- Field 3
0; !- Field 4

Schedule:Compact,
heating_sch_HRef, !- Name
Temperature, !- Schedule Type Limits Name
Through: 12/31, !- Field 1
For: AllDays, !- Field 2
Until: 24:00, !- Field 3
20; !- Field 4

Schedule:Compact,

cooling_sch_HRef, !- Name
Temperature, !- Schedule Type Limits Name
Through: 12/31, !- Field 1
For: AllDays, !- Field 2
Until: 24:00, !- Field 3
25.55; !- Field 4

Schedule:Compact,
heating_sch, !- Name
Temperature, !- Schedule Type Limits Name
Through: 5/1, !- Field 1
For: AllDays, !- Field 2
Until: 24:00, !- Field 3
21.5, !- Field 4
Through:11/1, !- Field 5
For: AllDays, !- Field 6
Until: 24:00, !- Field 7
15, !- Field 8
Through:12/31, !- Field 9
For: AllDays, !- Field 10
Until: 24:00, !- Field 11
21.5; !- Field 12

Schedule:Compact,
cooling_sch, !- Name
Temperature, !- Schedule Type Limits Name
Through: 5/1, !- Field 1
For: AllDays, !- Field 2
Until: 24:00, !- Field 3
28, !- Field 4
Through:11/1, !- Field 5
For: AllDays, !- Field 6
Until: 24:00, !- Field 7
23, !- Field 8
Through:12/31, !- Field 9
For: AllDays, !- Field 10
Until: 24:00, !- Field 11
28; !- Field 12

Schedule:Compact,
fan_cycle, !- Name
any number, !- Schedule Type Limits Name
Through: 12/31, !- Field 1
For: AllDays, !- Field 2
Until: 24:00, !- Field 3
0; !- Field 4

Schedule:Compact,
DaysOfWeek, !- Name
any number, !- Schedule Type Limits Name
Through: 12/31, !- Field 1
For: Weekdays, !- Field 2
Until:24:00, !- Field 3
1, !- Field 4
For: Weekends, !- Field 5
Until:24:00, !- Field 6
2, !- Field 7
For: AllOtherDays, !- Field 8
Until:24:00, !- Field 9
2; !- Field 10

!- ===== ALL OBJECTS IN CLASS: SCHEDULE:CONSTANT =====

Schedule:Constant,
DWWaterTempSchedule, !- Name
Temperature, !- Schedule Type Limits Name
48.8888888888889; !- Hourly Value

Schedule:Constant,

```

CWWaterTempSchedule,  !- Name
Temperature,          !- Schedule Type Limits Name
48.8888888888889;    !- Hourly Value

Schedule:Constant,
SinkSensSchedule,    !- Name
Fraction,            !- Schedule Type Limits Name
0.687777777777778;  !- Hourly Value

Schedule:Constant,
SinkLatSchedule,     !- Name
Fraction,            !- Schedule Type Limits Name
0.312222222222222;  !- Hourly Value

Schedule:Constant,
ShowerSensSchedule, !- Name
Fraction,            !- Schedule Type Limits Name
0.51280276816609;   !- Hourly Value

Schedule:Constant,
ShowerLatSchedule,  !- Name
Fraction,            !- Schedule Type Limits Name
0.48719723183391;   !- Hourly Value

Schedule:Constant,
BathSensSchedule,   !- Name
Fraction,            !- Schedule Type Limits Name
1;                   !- Hourly Value

Schedule:Constant,
BathLatSchedule,    !- Name
Fraction,            !- Schedule Type Limits Name
0;                   !- Hourly Value

Schedule:Constant,
SSBWaterTempSchedule, !- Name
Temperature,          !- Schedule Type Limits Name
40.5555555555556;    !- Hourly Value

Schedule:Constant,
WaterHeaterSP1Schedule, !- Name
Temperature,          !- Schedule Type Limits Name
48.8888888888889;    !- Hourly Value

Schedule:Constant,
WaterHeaterSP2Schedule, !- Name
Temperature,          !- Schedule Type Limits Name
40.5555555555556;    !- Hourly Value

Schedule:Constant,
DHWSupplySetpoint,  !- Name
Temperature,          !- Schedule Type Limits Name
48.8888888888889;    !- Hourly Value

!$Id: hvac.template,v 1.6 2019/01/28 18:26:57 d3a061 Exp $
!$Log: hvac.template,v $
!Revision 1.6 2019/01/28 18:26:57 d3a061
!Removed r_duct and r_returnduct defaults because those were
!moved into eplus.res.template.
!
!Revision 1.5 2018/12/13 20:11:13 d3a061
!Added one minor comment (question related to duct_loc that will need to
!be addressed at some point).
!
!Revision 1.4 2018/09/17 17:42:08 d3a061
!Cleaned up some units conversions. Added a few warnings to comments.
!
!Revision 1.3 2018/05/17 17:20:40 d3a061
!Ran dos2unix on it.
!
```

```

!Revision 1.2 2017/04/12 22:42:28 mend509
!Added 2018 IECC defaults
!
!Revision 1.1 2017/04/12 22:05:59 mend509
!Initial revision
!
!Revision 1.4 2015/05/21 16:51:39 mend509
!Archive after determination analysis
!
!Revision 1.2 2014/05/26 23:34:46 mend509
!Check-in before adding MF boiler model
!
!Revision 1.1 2014/05/16 23:26:26 mend509
!Initial revision
!
!Revision 1.1 2013/03/24 05:34:08 mend509
!Initial revision
!
Schedule:Constant,
  boiler_setpt,      !- Name
  Temperature,      !- Schedule Type Limits Name
  80;                !- Hourly Value

Schedule:Constant,
  Compressor Setpoint, !- Name
  Temperature,        !- Schedule Type Limits Name
  50;                  !- Hourly Value

Schedule:Constant,
  DOAShightemp,      !- Name
  Temperature,        !- Schedule Type Limits Name
  200;                !- Hourly Value

Schedule:Constant,
  DOASlowtemp,       !- Name
  Temperature,        !- Schedule Type Limits Name
  -60;                !- Hourly Value

!- ===== ALL OBJECTS IN CLASS: MATERIAL =====

Material,
  sheathing_consol_layer, !- Name
  Rough,                  !- Roughness
  0.0127,                 !- Thickness {m}
  0.0940184,              !- Conductivity {W/m-K}
  685.008,                !- Density {kg/m3}
  1172.332;               !- Specific Heat {J/kg-K}

Material,
  ceil_consol_layer,     !- Name
  Rough,                  !- Roughness
  0.34039219657243,      !- Thickness {m}
  0.0617176,              !- Conductivity {W/m-K}
  41.9286,                !- Density {kg/m3}
  776.25126;              !- Specific Heat {J/kg-K}

Material,
  floor_consol_layer,    !- Name
  Rough,                  !- Roughness
  0.1397,                 !- Thickness {m}
  0.0491038947368421,    !- Conductivity {W/m-K}
  55.074,                 !- Density {kg/m3}
  916.9311;               !- Specific Heat {J/kg-K}

Material,
  bsmtwall_consol_layer, !- Name
  Rough,                  !- Roughness
  0.0889,                 !- Thickness {m}
  0.0579573076923077,    !- Conductivity {W/m-K}

```

120.801, !- Density {kg/m3}
1036.25775; !- Specific Heat {J/kg-K}

Material,
crawlwall_consol_layer, !- Name
Rough, !- Roughness
0.0889, !- Thickness {m}
0.0579573076923077, !- Conductivity {W/m-K}
120.801, !- Density {kg/m3}
1036.25775; !- Specific Heat {J/kg-K}

Material,
wall_consol_layer, !- Name
Rough, !- Roughness
0.1397, !- Thickness {m}
0.057165, !- Conductivity {W/m-K}
120.801, !- Density {kg/m3}
1036.25775; !- Specific Heat {J/kg-K}

Material,
Very High Reflectivity Surface, !- Name
Smooth, !- Roughness
0.0005, !- Thickness {m}
237, !- Conductivity {W/m-K}
2702, !- Density {kg/m3}
903, !- Specific Heat {J/kg-K}
0.90, !- Thermal Absorptance
0.05, !- Solar Absorptance
0.05; !- Visible Absorptance

Material,
GypsumBoard-5/16in, !- Name
Rough, !- Roughness
7.93953E-03, !- Thickness {m}
0.1586200, !- Conductivity {W/m-K}
640, !- Density {kg/m3}
1129.6, !- Specific Heat {J/kg-K}
0.9, !- Thermal Absorptance
0.4, !- Solar Absorptance
0.1; !- Visible Absorptance

Material,
CopperPipe, !- Name
MediumRough, !- Roughness
1.90500386169072E-02, !- Thickness {m}
401, !- Conductivity {W/m-K}
2243.000, !- Density {kg/m3}
837.0000, !- Specific Heat {J/kg-K}
0.9000000, !- Thermal Absorptance
0.6500000, !- Solar Absorptance
0.6500000; !- Visible Absorptance

Material,
F08 Metal surface, !- Name
Smooth, !- Roughness
0.0008, !- Thickness {m}
45.28, !- Conductivity {W/m-K}
7824, !- Density {kg/m3}
500; !- Specific Heat {J/kg-K}

Material,
Concrete_4in, !- Name
Rough, !- Roughness
0.1014984, !- Thickness {m}
1.312098, !- Conductivity {W/m-K}
2242.8, !- Density {kg/m3}
465.2; !- Specific Heat {J/kg-K}

Material,
Asphalt_shingle, !- Name

MediumRough, !- Roughness
6.33985285170672E-03, !- Thickness {m}
0.08186, !- Conductivity {W/m-K}
1121.2917044623, !- Density {kg/m3}
1255.20000949809, !- Specific Heat {J/kg-K}
, !- Thermal Absorptance
0.75; !- Solar Absorptance

Material,
Wood_shingle, !- Name
MediumSmooth, !- Roughness
1.27000257446048E-02, !- Thickness {m}
0.11388, !- Conductivity {W/m-K}
426.090847695673, !- Density {kg/m3}
1631.76001234752; !- Specific Heat {J/kg-K}

Material,
Slate_shingle, !- Name
MediumSmooth, !- Roughness
1.27000257446048E-02, !- Thickness {m}
1.44219, !- Conductivity {W/m-K}
1601.845292089, !- Density {kg/m3}
1255.20000949809; !- Specific Heat {J/kg-K}

Material,
cement_stucco, !- Name
MediumSmooth, !- Roughness
1.905E-02, !- Thickness {m}
0.721, !- Conductivity {W/m-K}
1865.58, !- Density {kg/m3}
878.640006648665; !- Specific Heat {J/kg-K}

Material,
syn_stucco, !- Name
MediumSmooth, !- Roughness
0.3048E-02, !- Thickness {m}
8.65E-02, !- Conductivity {W/m-K}
400, !- Density {kg/m3}
878.640006648665; !- Specific Heat {J/kg-K}

Material,
Drywall_1/2in, !- Name
MediumSmooth, !- Roughness
1.27000257446048E-02, !- Thickness {m}
0.16009, !- Conductivity {W/m-K}
800.922646044499, !- Density {kg/m3}
1087.84000823168; !- Specific Heat {J/kg-K}

Material,
OSB_5/8in, !- Name
MediumSmooth, !- Roughness
0.015875032180756, !- Thickness {m}
0.1163, !- Conductivity {W/m-K}
544.627399310259, !- Density {kg/m3}
1213.36000918149; !- Specific Heat {J/kg-K}

Material,
OSB_7/16in, !- Name
MediumSmooth, !- Roughness
0.0111125, !- Thickness {m}
0.1163, !- Conductivity {W/m-K}
544.627399310259, !- Density {kg/m3}
1213.36000918149; !- Specific Heat {J/kg-K}

Material,
Blown_R30, !- Name
MediumRough, !- Roughness
0.212598430964684, !- Thickness {m}
0.04119, !- Conductivity {W/m-K}
9.61107175253399, !- Density {kg/m3}

836.800006332062; !- Specific Heat {J/kg-K}

Material,

Blown_R30_top, !- Name
MediumRough, !- Roughness
0.117348237880148, !- Thickness {m}
0.04119, !- Conductivity {W/m-K}
9.61107175253399, !- Density {kg/m3}
836.800006332062, !- Specific Heat {J/kg-K}
0.9, !- Thermal Absorptance
0.7, !- Solar Absorptance
0.7; !- Visible Absorptance

Material,

Plywood_3/4in, !- Name
Rough, !- Roughness
0.01905, !- Thickness {m}
0.1154577, !- Conductivity {W/m-K}
544.68, !- Density {kg/m3}
674.54, !- Specific Heat {J/kg-K}
0.9, !- Thermal Absorptance
0.7, !- Solar Absorptance
0.7; !- Visible Absorptance

Material,

Batt_R19, !- Name
MediumRough, !- Roughness
2.54000514892096E-02, !- Thickness {m}
3.47522010738099E-03, !- Conductivity {W/m-K}
9.61107175253399, !- Density {kg/m3}
836.800006332062, !- Specific Heat {J/kg-K}
0.9, !- Thermal Absorptance
0.7, !- Solar Absorptance
0.7; !- Visible Absorptance

Material,

Lumber_2x4, !- Name
Rough, !- Roughness
0.0890016, !- Thickness {m}
0.1154577, !- Conductivity {W/m-K}
512.64, !- Density {kg/m3}
767.58, !- Specific Heat {J/kg-K}
0.9, !- Thermal Absorptance
0.7, !- Solar Absorptance
0.7; !- Visible Absorptance

Material,

Carpet_n_pad, !- Name
MediumSmooth, !- Roughness
2.54000514892096E-02, !- Thickness {m}
6.01314018580031E-02, !- Conductivity {W/m-K}
32.03690584178, !- Density {kg/m3}
836.800006332062, !- Specific Heat {J/kg-K}
0.9, !- Thermal Absorptance
0.7, !- Solar Absorptance
0.7; !- Visible Absorptance

Material,

Batt_R13, !- Name
MediumRough, !- Roughness
0.0889, !- Thickness {m}
0.03876, !- Conductivity {W/m-K}
9.61107175253399, !- Density {kg/m3}
836.800006332062, !- Specific Heat {J/kg-K}
0.9, !- Thermal Absorptance
0.7, !- Solar Absorptance
0.7; !- Visible Absorptance

Material,

OSB_1/2in, !- Name

MediumSmooth, !- Roughness
 1.27000257446048E-02, !- Thickness {m}
 0.1163, !- Conductivity {W/m-K}
 544.627399310259, !- Density {kg/m3}
 1213.36000918149, !- Specific Heat {J/kg-K}
 0.9, !- Thermal Absorptance
 0.7, !- Solar Absorptance
 0.7; !- Visible Absorptance

Material,
 soil_12in, !- Name
 Rough, !- Roughness
 0.3048, !- Thickness {m}
 1.731, !- Conductivity {W/m-K}
 1842.3, !- Density {kg/m3}
 232.6, !- Specific Heat {J/kg-K}
 0.9, !- Thermal Absorptance
 0.7, !- Solar Absorptance
 0.7; !- Visible Absorptance

Material,
 door_const, !- Name
 Smooth, !- Roughness
 0.0396277251435213, !- Thickness {m}
 0.0720096, !- Conductivity {W/m-K}
 512.64, !- Density {kg/m3}
 767.58; !- Specific Heat {J/kg-K}

Material,
 Gyp_board_1/2in, !- Name
 Rough, !- Roughness
 0.01271016, !- Thickness {m}
 0.1586200, !- Conductivity {W/m-K}
 640, !- Density {kg/m3}
 1129.6, !- Specific Heat {J/kg-K}
 0.9, !- Thermal Absorptance
 0.4, !- Solar Absorptance
 0.1; !- Visible Absorptance

Material,
 Std Wood 6inch, !- Name
 MediumSmooth, !- Roughness
 0.15, !- Thickness {m}
 0.12, !- Conductivity {W/m-K}
 540.0000, !- Density {kg/m3}
 1210, !- Specific Heat {J/kg-K}
 0.9000000, !- Thermal Absorptance
 0.7000000, !- Solar Absorptance
 0.7000000; !- Visible Absorptance

Material,
 Pipe Insulation, !- Name
 VeryRough, !- Roughness
 0.0127032520325203, !- Thickness {m}
 0.03317175, !- Conductivity {W/m-K}
 91.0, !- Density {kg/m3}
 836.0, !- Specific Heat {J/kg-K}
 0.9, !- Thermal Absorptance
 0.5, !- Solar Absorptance
 0.5; !- Visible Absorptance

!- ===== ALL OBJECTS IN CLASS: MATERIAL:NOMASS =====

Material:NoMass,
 Manf_wall_airgap, !- Name
 Smooth, !- Roughness
 0.12; !- Thermal Resistance {m2-K/W}

Material:NoMass,

```

Bldg_paper_felt,    !- Name
Smooth,            !- Roughness
1.05666113069662E-02; !- Thermal Resistance {m2-K/W}

Material:NoMass,
R_high,           !- Name
MediumRough,     !- Roughness
177;             !- Thermal Resistance {m2-K/W}

!- ===== ALL OBJECTS IN CLASS: MATERIAL:AIRGAP =====

Material:AirGap,
Air_4_in_vert,    !- Name
0.158499169604493; !- Thermal Resistance {m2-K/W}

Material:AirGap,
3/4in_air_space, !- Name
0.08513;         !- Thermal Resistance {m2-K/W}

Material:AirGap,
3/4in_Reflective_air_space, !- Name
0.246554;       !- Thermal Resistance {m2-K/W}

!- ===== ALL OBJECTS IN CLASS: WINDOWMATERIAL:SIMPLEGLAZINGSYSTEM =====

WindowMaterial:SimpleGlazingSystem,
Glass,           !- Name
1.817152,       !- U-Factor {W/m2-K}
0.216875,       !- Solar Heat Gain Coefficient
0.88;           !- Visible Transmittance

!- ===== ALL OBJECTS IN CLASS: WINDOWMATERIAL:GLAZING =====

!$Id: Materials.idf,v 1.1 2018/05/07 16:36:33 d3a061 Exp $
!$Log: Materials.idf,v $
!Revision 1.1 2018/05/07 16:36:33 d3a061
!Initial revision
!
!Revision 1.2 2015/05/21 16:55:03 mend509
!Archive after determination analysis
!
!Revision 1.1 2014/05/16 23:26:01 mend509
!Initial revision
!
!Revision 1.2 2013/03/29 17:31:35 mend509
!Archived after generating first set of masonry input files.\
!
!Revision 1.1 2013/03/24 05:35:41 mend509
!Initial revision
!
WindowMaterial:Glazing,
Clear Acrylic Plastic, !- Name
SpectralAverage,       !- Optical Data Type
,                     !- Window Glass Spectral Data Set Name
0.003,               !- Thickness {m}
0.92,                !- Solar Transmittance at Normal Incidence
0.05,                !- Front Side Solar Reflectance at Normal Incidence
0.05,                !- Back Side Solar Reflectance at Normal Incidence
0.92,                !- Visible Transmittance at Normal Incidence
0.05,                !- Front Side Visible Reflectance at Normal Incidence
0.05,                !- Back Side Visible Reflectance at Normal Incidence
0.00,                !- Infrared Transmittance at Normal Incidence
0.90,                !- Front Side Infrared Hemispherical Emissivity
0.90,                !- Back Side Infrared Hemispherical Emissivity
0.90;                !- Conductivity {W/m-K}

WindowMaterial:Glazing,

```

```

Diffusing Acrylic Plastic, !- Name
SpectralAverage, !- Optical Data Type
, !- Window Glass Spectral Data Set Name
0.0022, !- Thickness {m}
0.90, !- Solar Transmittance at Normal Incidence
0.08, !- Front Side Solar Reflectance at Normal Incidence
0.08, !- Back Side Solar Reflectance at Normal Incidence
0.90, !- Visible Transmittance at Normal Incidence
0.08, !- Front Side Visible Reflectance at Normal Incidence
0.08, !- Back Side Visible Reflectance at Normal Incidence
0.00, !- Infrared Transmittance at Normal Incidence
0.90, !- Front Side Infrared Hemispherical Emissivity
0.90, !- Back Side Infrared Hemispherical Emissivity
0.90; !- Conductivity {W/m-K}

```

!- ===== ALL OBJECTS IN CLASS: WINDOWMATERIAL:BLIND =====

!*** Properties for blinds taken from E+ dataset for 'Blinds with Medium Reflectivity Slats'***

!

WindowMaterial:Blind,

```

int_blind, !- Name
Horizontal, !- Slat Orientation
0.025, !- Slat Width {m}
0.01875, !- Slat Separation {m}
0.001, !- Slat Thickness {m}
45.0, !- Slat Angle {deg}
221, !- Slat Conductivity {W/m-K}
0.0, !- Slat Beam Solar Transmittance
0.5, !- Front Side Slat Beam Solar Reflectance
0.5, !- Back Side Slat Beam Solar Reflectance
0.0, !- Slat Diffuse Solar Transmittance
0.5, !- Front Side Slat Diffuse Solar Reflectance
0.5, !- Back Side Slat Diffuse Solar Reflectance
0.0, !- Slat Beam Visible Transmittance
0.5, !- Front Side Slat Beam Visible Reflectance
0.5, !- Back Side Slat Beam Visible Reflectance
0.0, !- Slat Diffuse Visible Transmittance
0.5, !- Front Side Slat Diffuse Visible Reflectance
0.5, !- Back Side Slat Diffuse Visible Reflectance
0.0, !- Slat Infrared Hemispherical Transmittance
0.9, !- Front Side Slat Infrared Hemispherical Emissivity
0.9, !- Back Side Slat Infrared Hemispherical Emissivity
0.050, !- Blind to Glass Distance {m}
0.5, !- Blind Top Opening Multiplier
0.5, !- Blind Bottom Opening Multiplier
0.5, !- Blind Left Side Opening Multiplier
0.5, !- Blind Right Side Opening Multiplier
, !- Minimum Slat Angle {deg}
; !- Maximum Slat Angle {deg}

```

!- ===== ALL OBJECTS IN CLASS: CONSTRUCTION =====

Construction,

```

Exterior Wall, !- Name
syn_stucco, !- Outside Layer
sheathing_consol_layer, !- Layer 2
OSB_7/16in, !- Layer 3
wall_consol_layer, !- Layer 4
Drywall_1/2in; !- Layer 5

```

Construction,

```

InteriorFurnishings, !- Name
Std Wood 6inch; !- Outside Layer

```

Construction,

```

ceiling-floor-layer, !- Name
Lumber_2x4; !- Outside Layer

```

Construction,
Exterior Floor, !- Name
floor_consol_layer, !- Outside Layer
Plywood_3/4in, !- Layer 2
Carpet_n_pad; !- Layer 3

Construction,
Interior Floor, !- Name
Plywood_3/4in, !- Outside Layer
Carpet_n_pad; !- Layer 2

Construction,
Interior Ceiling, !- Name
ceiling_consol_layer, !- Outside Layer
Drywall_1/2in; !- Layer 2

Construction,
attic floor, !- Name
Drywall_1/2in, !- Outside Layer
ceiling_consol_layer; !- Layer 2

Construction,
fndn_roof, !- Name
Carpet_n_pad, !- Outside Layer
Plywood_3/4in, !- Layer 2
ceiling_consol_layer; !- Layer 3

Construction,
interiorwall, !- Name
Drywall_1/2in, !- Outside Layer
OSB_5/8in, !- Layer 2
Drywall_1/2in; !- Layer 3

Construction,
Interior Wall, !- Name
Drywall_1/2in, !- Outside Layer
Air_4_in_vert, !- Layer 2
Drywall_1/2in; !- Layer 3

Construction,
Exterior Roof, !- Name
Asphalt_shingle, !- Outside Layer
OSB_1/2in; !- Layer 2

Construction,
Exterior Window, !- Name
Glass; !- Outside Layer

Construction,
Interior Window, !- Name
Glass; !- Outside Layer

Construction,
Exterior Door, !- Name
door_const; !- Outside Layer

Construction,
Interior Door, !- Name
door_const; !- Outside Layer

Construction,
Gable_end, !- Name
cement_stucco, !- Outside Layer
Bldg_paper_felt, !- Layer 2
OSB_5/8in, !- Layer 3
Air_4_in_vert, !- Layer 4
Drywall_1/2in; !- Layer 5

Construction,
crawl_floor, !- Name

```

R_high,      !- Outside Layer
soil_12in;   !- Layer 2

Construction,
window_w_blinds,    !- Name
Glass,             !- Outside Layer
int_blind;         !- Layer 2

Construction,
Insulated Pipe,    !- Name
Pipe Insulation,   !- Outside Layer
CopperPipe;       !- Layer 2

Construction,
Plain Pipe,        !- Name
CopperPipe;       !- Outside Layer

Construction,
TDD Pipe,          !- Name
Very High Reflectivity Surface; !- Outside Layer

Construction,
TDD Dome,          !- Name
Clear Acrylic Plastic; !- Outside Layer

Construction,
TDD Diffuser,      !- Name
Diffusing Acrylic Plastic; !- Outside Layer

Construction,
crawl wall,        !- Name
Concrete_4in;     !- Outside Layer

!- ===== ALL OBJECTS IN CLASS: GLOBALGEOMETRYRULES =====

GlobalGeometryRules,
LowerLeftCorner,    !- Starting Vertex Position
Counterclockwise,  !- Vertex Entry Direction
Relative;          !- Coordinate System

!- ===== ALL OBJECTS IN CLASS: ZONE =====

Zone,
living_unit1,      !- Name
0.0,              !- Direction of Relative North {deg}
0.0,              !- X Origin {m}
0.0,              !- Y Origin {m}
0.0,              !- Z Origin {m}
,                 !- Type
1;                !- Multiplier

Zone,
attic_unit1,       !- Name
0.0,              !- Direction of Relative North {deg}
0.0,              !- X Origin {m}
0.0,              !- Y Origin {m}
0.0,              !- Z Origin {m}
,                 !- Type
1;                !- Multiplier

Zone,
crawlspcse_unit1,  !- Name
0.0,              !- Direction of Relative North {deg}
0.0,              !- X Origin {m}
0.0,              !- Y Origin {m}
0.0,              !- Z Origin {m}
,                 !- Type
1;                !- Multiplier

```

!- ===== ALL OBJECTS IN CLASS: BUILDINGSURFACE:DETAILED =====

!\$Id: scalable.geometry.template,v 1.3 2018/11/20 17:57:10 d3a061 Exp d3a061 \$
!\$Log: scalable.geometry.template,v \$
!Revision 1.3 2018/11/20 17:57:10 d3a061
!Ran dos2unix on it.
!
!Revision 1.2 2018/10/16 21:09:50 d31143
!VM made changes to commented-out code line, we don't know why.
!
!Revision 1.4 2015/05/21 16:52:28 mend509
!Archive after determination analysis
!
!Revision 1.3 2014/09/08 17:51:39 mend509
!Version saved after correct foundation gtp references
!
!Revision 1.2 2014/05/19 23:08:33 mend509
!Re-worked below-grade sections to match the "read-from-gtp-file" option added in the main tmpl
!
!Revision 1.1 2014/05/16 23:26:12 mend509
!Initial revision
!
!Revision 1.2 2012/10/24 22:37:53 mend509
!*** empty log message ***
!
!Revision 1.1 2012/08/21 16:51:40 mend509
!Initial revision

BuildingSurface:Detailed,

Inter zone floor 1, !- Name
Floor, !- Surface Type
Interior Floor, !- Construction Name
living_unit1, !- Zone Name
Adiabatic, !- Outside Boundary Condition
, !- Outside Boundary Condition Object
NoSun, !- Sun Exposure
NoWind, !- Wind Exposure
0.0, !- View Factor to Ground
4, !- Number of Vertices
0, !- Vertex 1 X-coordinate {m}
0, !- Vertex 1 Y-coordinate {m}
2.74390243902439, !- Vertex 1 Z-coordinate {m}
0, !- Vertex 2 X-coordinate {m}
9.09981820971244, !- Vertex 2 Y-coordinate {m}
2.74390243902439, !- Vertex 2 Z-coordinate {m}
12.1330909462833, !- Vertex 3 X-coordinate {m}
9.09981820971244, !- Vertex 3 Y-coordinate {m}
2.74390243902439, !- Vertex 3 Z-coordinate {m}
12.1330909462833, !- Vertex 4 X-coordinate {m}
0, !- Vertex 4 Y-coordinate {m}
2.74390243902439; !- Vertex 4 Z-coordinate {m}

BuildingSurface:Detailed,

ceiling_unit1, !- Name
Ceiling, !- Surface Type
Interior Ceiling, !- Construction Name
living_unit1, !- Zone Name
Zone, !- Outside Boundary Condition
attic_unit1, !- Outside Boundary Condition Object
NoSun, !- Sun Exposure
NoWind, !- Wind Exposure
0.0, !- View Factor to Ground
4, !- Number of Vertices
0, !- Vertex 1 X-coordinate {m}
0, !- Vertex 1 Y-coordinate {m}
5.33536585365854, !- Vertex 1 Z-coordinate {m}
12.1330909462833, !- Vertex 2 X-coordinate {m}
0, !- Vertex 2 Y-coordinate {m}
5.33536585365854, !- Vertex 2 Z-coordinate {m}

12.1330909462833, !- Vertex 3 X-coordinate {m}
9.09981820971244, !- Vertex 3 Y-coordinate {m}
5.33536585365854, !- Vertex 3 Z-coordinate {m}
0, !- Vertex 4 X-coordinate {m}
9.09981820971244, !- Vertex 4 Y-coordinate {m}
5.33536585365854; !- Vertex 4 Z-coordinate {m}

BuildingSurface:Detailed,

Roof_front_unit1, !- Name
Roof, !- Surface Type
Exterior Roof, !- Construction Name
attic_unit1, !- Zone Name
Outdoors, !- Outside Boundary Condition
, !- Outside Boundary Condition Object
SunExposed, !- Sun Exposure
WindExposed, !- Wind Exposure
0.0, !- View Factor to Ground
4, !- Number of Vertices
0, !- Vertex 1 X-coordinate {m}
0, !- Vertex 1 Y-coordinate {m}
5.33536585365854, !- Vertex 1 Z-coordinate {m}
12.1330909462833, !- Vertex 2 X-coordinate {m}
0, !- Vertex 2 Y-coordinate {m}
5.33536585365854, !- Vertex 2 Z-coordinate {m}
12.1330909462833, !- Vertex 3 X-coordinate {m}
4.54990910485622, !- Vertex 3 Y-coordinate {m}
6.6995631975537, !- Vertex 3 Z-coordinate {m}
0, !- Vertex 4 X-coordinate {m}
4.54990910485622, !- Vertex 4 Y-coordinate {m}
6.6995631975537; !- Vertex 4 Z-coordinate {m}

BuildingSurface:Detailed,

Roof_back_unit1, !- Name
Roof, !- Surface Type
Exterior Roof, !- Construction Name
attic_unit1, !- Zone Name
Outdoors, !- Outside Boundary Condition
, !- Outside Boundary Condition Object
SunExposed, !- Sun Exposure
WindExposed, !- Wind Exposure
0.0, !- View Factor to Ground
4, !- Number of Vertices
12.1330909462833, !- Vertex 1 X-coordinate {m}
9.09981820971244, !- Vertex 1 Y-coordinate {m}
5.33536585365854, !- Vertex 1 Z-coordinate {m}
0, !- Vertex 2 X-coordinate {m}
9.09981820971244, !- Vertex 2 Y-coordinate {m}
5.33536585365854, !- Vertex 2 Z-coordinate {m}
0, !- Vertex 3 X-coordinate {m}
4.54990910485622, !- Vertex 3 Y-coordinate {m}
6.6995631975537, !- Vertex 3 Z-coordinate {m}
12.1330909462833, !- Vertex 4 X-coordinate {m}
4.54990910485622, !- Vertex 4 Y-coordinate {m}
6.6995631975537; !- Vertex 4 Z-coordinate {m}

BuildingSurface:Detailed,

Roof_right_unit1, !- Name
Wall, !- Surface Type
Gable_end, !- Construction Name
attic_unit1, !- Zone Name
Outdoors, !- Outside Boundary Condition
, !- Outside Boundary Condition Object
SunExposed, !- Sun Exposure
WindExposed, !- Wind Exposure
, !- View Factor to Ground
3, !- Number of Vertices
12.1330909462833, !- Vertex 1 X-coordinate {m}
0, !- Vertex 1 Y-coordinate {m}
5.33536585365854, !- Vertex 1 Z-coordinate {m}
12.1330909462833, !- Vertex 2 X-coordinate {m}

9.09981820971244, !- Vertex 2 Y-coordinate {m}
5.33536585365854, !- Vertex 2 Z-coordinate {m}
12.1330909462833, !- Vertex 3 X-coordinate {m}
4.54990910485622, !- Vertex 3 Y-coordinate {m}
6.6995631975537; !- Vertex 3 Z-coordinate {m}

BuildingSurface:Detailed,

Roof_left_unit1, !- Name
Wall, !- Surface Type
Gable_end, !- Construction Name
attic_unit1, !- Zone Name
Outdoors, !- Outside Boundary Condition
, !- Outside Boundary Condition Object
SunExposed, !- Sun Exposure
WindExposed, !- Wind Exposure
, !- View Factor to Ground
3, !- Number of Vertices
0, !- Vertex 1 X-coordinate {m}
9.09981820971244, !- Vertex 1 Y-coordinate {m}
5.33536585365854, !- Vertex 1 Z-coordinate {m}
0, !- Vertex 2 X-coordinate {m}
0, !- Vertex 2 Y-coordinate {m}
5.33536585365854, !- Vertex 2 Z-coordinate {m}
0, !- Vertex 3 X-coordinate {m}
4.54990910485622, !- Vertex 3 Y-coordinate {m}
6.6995631975537; !- Vertex 3 Z-coordinate {m}

BuildingSurface:Detailed,

Wall_ldf_1.unit1, !- Name
Wall, !- Surface Type
Exterior Wall, !- Construction Name
living_unit1, !- Zone Name
Outdoors, !- Outside Boundary Condition
, !- Outside Boundary Condition Object
SunExposed, !- Sun Exposure
WindExposed, !- Wind Exposure
, !- View Factor to Ground
4, !- Number of Vertices
0, !- Vertex 1 X-coordinate {m}
0, !- Vertex 1 Y-coordinate {m}
0.152439024390244, !- Vertex 1 Z-coordinate {m}
12.1330909462833, !- Vertex 2 X-coordinate {m}
0, !- Vertex 2 Y-coordinate {m}
0.152439024390244, !- Vertex 2 Z-coordinate {m}
12.1330909462833, !- Vertex 3 X-coordinate {m}
0, !- Vertex 3 Y-coordinate {m}
2.74390243902439, !- Vertex 3 Z-coordinate {m}
0, !- Vertex 4 X-coordinate {m}
0, !- Vertex 4 Y-coordinate {m}
2.74390243902439; !- Vertex 4 Z-coordinate {m}

BuildingSurface:Detailed,

Wall_sdr_1.unit1, !- Name
Wall, !- Surface Type
Exterior Wall, !- Construction Name
living_unit1, !- Zone Name
Outdoors, !- Outside Boundary Condition
, !- Outside Boundary Condition Object
SunExposed, !- Sun Exposure
WindExposed, !- Wind Exposure
, !- View Factor to Ground
4, !- Number of Vertices
12.1330909462833, !- Vertex 1 X-coordinate {m}
0, !- Vertex 1 Y-coordinate {m}
0.152439024390244, !- Vertex 1 Z-coordinate {m}
12.1330909462833, !- Vertex 2 X-coordinate {m}
9.09981820971244, !- Vertex 2 Y-coordinate {m}
0.152439024390244, !- Vertex 2 Z-coordinate {m}
12.1330909462833, !- Vertex 3 X-coordinate {m}
9.09981820971244, !- Vertex 3 Y-coordinate {m}

2.74390243902439, !- Vertex 3 Z-coordinate {m}
12.1330909462833, !- Vertex 4 X-coordinate {m}
0, !- Vertex 4 Y-coordinate {m}
2.74390243902439; !- Vertex 4 Z-coordinate {m}

BuildingSurface:Detailed,

Wall_ldb_1.unit1, !- Name
Wall, !- Surface Type
Exterior Wall, !- Construction Name
living_unit1, !- Zone Name
Outdoors, !- Outside Boundary Condition
, !- Outside Boundary Condition Object
SunExposed, !- Sun Exposure
WindExposed, !- Wind Exposure
, !- View Factor to Ground
4, !- Number of Vertices
12.1330909462833, !- Vertex 1 X-coordinate {m}
9.09981820971244, !- Vertex 1 Y-coordinate {m}
0.152439024390244, !- Vertex 1 Z-coordinate {m}
0, !- Vertex 2 X-coordinate {m}
9.09981820971244, !- Vertex 2 Y-coordinate {m}
0.152439024390244, !- Vertex 2 Z-coordinate {m}
0, !- Vertex 3 X-coordinate {m}
9.09981820971244, !- Vertex 3 Y-coordinate {m}
2.74390243902439, !- Vertex 3 Z-coordinate {m}
12.1330909462833, !- Vertex 4 X-coordinate {m}
9.09981820971244, !- Vertex 4 Y-coordinate {m}
2.74390243902439; !- Vertex 4 Z-coordinate {m}

BuildingSurface:Detailed,

Wall_sdl_1.unit1, !- Name
Wall, !- Surface Type
Exterior Wall, !- Construction Name
living_unit1, !- Zone Name
Outdoors, !- Outside Boundary Condition
, !- Outside Boundary Condition Object
SunExposed, !- Sun Exposure
WindExposed, !- Wind Exposure
, !- View Factor to Ground
4, !- Number of Vertices
0, !- Vertex 1 X-coordinate {m}
9.09981820971244, !- Vertex 1 Y-coordinate {m}
0.152439024390244, !- Vertex 1 Z-coordinate {m}
0, !- Vertex 2 X-coordinate {m}
0, !- Vertex 2 Y-coordinate {m}
0.152439024390244, !- Vertex 2 Z-coordinate {m}
0, !- Vertex 3 X-coordinate {m}
0, !- Vertex 3 Y-coordinate {m}
2.74390243902439, !- Vertex 3 Z-coordinate {m}
0, !- Vertex 4 X-coordinate {m}
9.09981820971244, !- Vertex 4 Y-coordinate {m}
2.74390243902439; !- Vertex 4 Z-coordinate {m}

BuildingSurface:Detailed,

Wall_ldf_2.unit1, !- Name
Wall, !- Surface Type
Exterior Wall, !- Construction Name
living_unit1, !- Zone Name
Outdoors, !- Outside Boundary Condition
, !- Outside Boundary Condition Object
SunExposed, !- Sun Exposure
WindExposed, !- Wind Exposure
, !- View Factor to Ground
4, !- Number of Vertices
0, !- Vertex 1 X-coordinate {m}
0, !- Vertex 1 Y-coordinate {m}
2.74390243902439, !- Vertex 1 Z-coordinate {m}
12.1330909462833, !- Vertex 2 X-coordinate {m}
0, !- Vertex 2 Y-coordinate {m}
2.74390243902439, !- Vertex 2 Z-coordinate {m}

12.1330909462833, !- Vertex 3 X-coordinate {m}
0, !- Vertex 3 Y-coordinate {m}
5.33536585365854, !- Vertex 3 Z-coordinate {m}
0, !- Vertex 4 X-coordinate {m}
0, !- Vertex 4 Y-coordinate {m}
5.33536585365854; !- Vertex 4 Z-coordinate {m}

BuildingSurface:Detailed,

Wall_sdr_2.unit1, !- Name
Wall, !- Surface Type
Exterior Wall, !- Construction Name
living_unit1, !- Zone Name
Outdoors, !- Outside Boundary Condition
, !- Outside Boundary Condition Object
SunExposed, !- Sun Exposure
WindExposed, !- Wind Exposure
, !- View Factor to Ground
4, !- Number of Vertices
12.1330909462833, !- Vertex 1 X-coordinate {m}
0, !- Vertex 1 Y-coordinate {m}
2.74390243902439, !- Vertex 1 Z-coordinate {m}
12.1330909462833, !- Vertex 2 X-coordinate {m}
9.09981820971244, !- Vertex 2 Y-coordinate {m}
2.74390243902439, !- Vertex 2 Z-coordinate {m}
12.1330909462833, !- Vertex 3 X-coordinate {m}
9.09981820971244, !- Vertex 3 Y-coordinate {m}
5.33536585365854, !- Vertex 3 Z-coordinate {m}
12.1330909462833, !- Vertex 4 X-coordinate {m}
0, !- Vertex 4 Y-coordinate {m}
5.33536585365854; !- Vertex 4 Z-coordinate {m}

BuildingSurface:Detailed,

Wall_ldb_2.unit1, !- Name
Wall, !- Surface Type
Exterior Wall, !- Construction Name
living_unit1, !- Zone Name
Outdoors, !- Outside Boundary Condition
, !- Outside Boundary Condition Object
SunExposed, !- Sun Exposure
WindExposed, !- Wind Exposure
, !- View Factor to Ground
4, !- Number of Vertices
12.1330909462833, !- Vertex 1 X-coordinate {m}
9.09981820971244, !- Vertex 1 Y-coordinate {m}
2.74390243902439, !- Vertex 1 Z-coordinate {m}
0, !- Vertex 2 X-coordinate {m}
9.09981820971244, !- Vertex 2 Y-coordinate {m}
2.74390243902439, !- Vertex 2 Z-coordinate {m}
0, !- Vertex 3 X-coordinate {m}
9.09981820971244, !- Vertex 3 Y-coordinate {m}
5.33536585365854, !- Vertex 3 Z-coordinate {m}
12.1330909462833, !- Vertex 4 X-coordinate {m}
9.09981820971244, !- Vertex 4 Y-coordinate {m}
5.33536585365854; !- Vertex 4 Z-coordinate {m}

BuildingSurface:Detailed,

Wall_sdl_2.unit1, !- Name
Wall, !- Surface Type
Exterior Wall, !- Construction Name
living_unit1, !- Zone Name
Outdoors, !- Outside Boundary Condition
, !- Outside Boundary Condition Object
SunExposed, !- Sun Exposure
WindExposed, !- Wind Exposure
, !- View Factor to Ground
4, !- Number of Vertices
0, !- Vertex 1 X-coordinate {m}
9.09981820971244, !- Vertex 1 Y-coordinate {m}
2.74390243902439, !- Vertex 1 Z-coordinate {m}
0, !- Vertex 2 X-coordinate {m}

0, !- Vertex 2 Y-coordinate {m}
 2.74390243902439, !- Vertex 2 Z-coordinate {m}
 0, !- Vertex 3 X-coordinate {m}
 0, !- Vertex 3 Y-coordinate {m}
 5.33536585365854, !- Vertex 3 Z-coordinate {m}
 0, !- Vertex 4 X-coordinate {m}
 9.09981820971244, !- Vertex 4 Y-coordinate {m}
 5.33536585365854; !- Vertex 4 Z-coordinate {m}

BuildingSurface:Detailed,

Floor_unit1, !- Name
 Floor, !- Surface Type
 Exterior Floor, !- Construction Name
 living_unit1, !- Zone Name
 Zone, !- Outside Boundary Condition
 crawlspace_unit1, !- Outside Boundary Condition Object
 NoSun, !- Sun Exposure
 NoWind, !- Wind Exposure
 0.0, !- View Factor to Ground
 4, !- Number of Vertices
 0, !- Vertex 1 X-coordinate {m}
 0, !- Vertex 1 Y-coordinate {m}
 0.152439024390244, !- Vertex 1 Z-coordinate {m}
 0, !- Vertex 2 X-coordinate {m}
 9.09981820971244, !- Vertex 2 Y-coordinate {m}
 0.152439024390244, !- Vertex 2 Z-coordinate {m}
 12.1330909462833, !- Vertex 3 X-coordinate {m}
 9.09981820971244, !- Vertex 3 Y-coordinate {m}
 0.152439024390244, !- Vertex 3 Z-coordinate {m}
 12.1330909462833, !- Vertex 4 X-coordinate {m}
 0, !- Vertex 4 Y-coordinate {m}
 0.152439024390244; !- Vertex 4 Z-coordinate {m}

BuildingSurface:Detailed,

BGWall_upper_ldf, !- Name
 Wall, !- Surface Type
 Crawl Wall, !- Construction Name
 crawlspace_unit1, !- Zone Name
 Outdoors, !- Outside Boundary Condition
 , !- Outside Boundary Condition Object
 SunExposed, !- Sun Exposure
 WindExposed, !- Wind Exposure
 , !- View Factor to Ground
 4, !- Number of Vertices
 0, !- Vertex 1 X-coordinate {m}
 0, !- Vertex 1 Y-coordinate {m}
 0, !- Vertex 1 Z-coordinate {m}
 12.1330909462833, !- Vertex 2 X-coordinate {m}
 0, !- Vertex 2 Y-coordinate {m}
 0, !- Vertex 2 Z-coordinate {m}
 12.1330909462833, !- Vertex 3 X-coordinate {m}
 0, !- Vertex 3 Y-coordinate {m}
 0.152439024390244, !- Vertex 3 Z-coordinate {m}
 0, !- Vertex 4 X-coordinate {m}
 0, !- Vertex 4 Y-coordinate {m}
 0.152439024390244; !- Vertex 4 Z-coordinate {m}

BuildingSurface:Detailed,

BGWall_lower_ldf, !- Name
 Wall, !- Surface Type
 Crawl Wall, !- Construction Name
 crawlspace_unit1, !- Zone Name
 Ground, !- Outside Boundary Condition
 , !- Outside Boundary Condition Object
 NoSun, !- Sun Exposure
 NoWind, !- Wind Exposure
 , !- View Factor to Ground
 4, !- Number of Vertices
 0, !- Vertex 1 X-coordinate {m}
 0, !- Vertex 1 Y-coordinate {m}

```

-0.457317073170732,    !- Vertex 1 Z-coordinate {m}
12.1330909462833,    !- Vertex 2 X-coordinate {m}
0,                      !- Vertex 2 Y-coordinate {m}
-0.457317073170732,    !- Vertex 2 Z-coordinate {m}
12.1330909462833,    !- Vertex 3 X-coordinate {m}
0,                      !- Vertex 3 Y-coordinate {m}
0,                      !- Vertex 3 Z-coordinate {m}
0,                      !- Vertex 4 X-coordinate {m}
0,                      !- Vertex 4 Y-coordinate {m}
0;                      !- Vertex 4 Z-coordinate {m}

```

BuildingSurface:Detailed,

```

BGWall_upper_sdr,    !- Name
Wall,                !- Surface Type
Crawl Wall,          !- Construction Name
crawlspace_unit1,    !- Zone Name
Outdoors,            !- Outside Boundary Condition
,                    !- Outside Boundary Condition Object
SunExposed,          !- Sun Exposure
WindExposed,         !- Wind Exposure
,                    !- View Factor to Ground
4,                  !- Number of Vertices
12.1330909462833,    !- Vertex 1 X-coordinate {m}
0,                  !- Vertex 1 Y-coordinate {m}
0,                  !- Vertex 1 Z-coordinate {m}
12.1330909462833,    !- Vertex 2 X-coordinate {m}
9.09981820971244,    !- Vertex 2 Y-coordinate {m}
0,                  !- Vertex 2 Z-coordinate {m}
12.1330909462833,    !- Vertex 3 X-coordinate {m}
9.09981820971244,    !- Vertex 3 Y-coordinate {m}
0.152439024390244,    !- Vertex 3 Z-coordinate {m}
12.1330909462833,    !- Vertex 4 X-coordinate {m}
0,                  !- Vertex 4 Y-coordinate {m}
0.152439024390244;    !- Vertex 4 Z-coordinate {m}

```

BuildingSurface:Detailed,

```

BGWall_lower_sdr,    !- Name
Wall,                !- Surface Type
Crawl Wall,          !- Construction Name
crawlspace_unit1,    !- Zone Name
Ground,              !- Outside Boundary Condition
,                    !- Outside Boundary Condition Object
NoSun,               !- Sun Exposure
NoWind,              !- Wind Exposure
,                    !- View Factor to Ground
4,                  !- Number of Vertices
12.1330909462833,    !- Vertex 1 X-coordinate {m}
0,                  !- Vertex 1 Y-coordinate {m}
-0.457317073170732,    !- Vertex 1 Z-coordinate {m}
12.1330909462833,    !- Vertex 2 X-coordinate {m}
9.09981820971244,    !- Vertex 2 Y-coordinate {m}
-0.457317073170732,    !- Vertex 2 Z-coordinate {m}
12.1330909462833,    !- Vertex 3 X-coordinate {m}
9.09981820971244,    !- Vertex 3 Y-coordinate {m}
0,                  !- Vertex 3 Z-coordinate {m}
12.1330909462833,    !- Vertex 4 X-coordinate {m}
0,                  !- Vertex 4 Y-coordinate {m}
0;                  !- Vertex 4 Z-coordinate {m}

```

BuildingSurface:Detailed,

```

BGWall_upper_ldb,    !- Name
Wall,                !- Surface Type
Crawl Wall,          !- Construction Name
crawlspace_unit1,    !- Zone Name
Outdoors,            !- Outside Boundary Condition
,                    !- Outside Boundary Condition Object
SunExposed,          !- Sun Exposure
WindExposed,         !- Wind Exposure
,                    !- View Factor to Ground
4,                  !- Number of Vertices

```

```

12.1330909462833,    !- Vertex 1 X-coordinate {m}
9.09981820971244,    !- Vertex 1 Y-coordinate {m}
0,                   !- Vertex 1 Z-coordinate {m}
0,                   !- Vertex 2 X-coordinate {m}
9.09981820971244,    !- Vertex 2 Y-coordinate {m}
0,                   !- Vertex 2 Z-coordinate {m}
0,                   !- Vertex 3 X-coordinate {m}
9.09981820971244,    !- Vertex 3 Y-coordinate {m}
0.152439024390244,   !- Vertex 3 Z-coordinate {m}
12.1330909462833,    !- Vertex 4 X-coordinate {m}
9.09981820971244,    !- Vertex 4 Y-coordinate {m}
0.152439024390244;   !- Vertex 4 Z-coordinate {m}

```

BuildingSurface:Detailed,

```

BGWall_lower_ldb,    !- Name
Wall,                !- Surface Type
Crawl Wall,          !- Construction Name
crawlspcse_unit1,    !- Zone Name
Ground,              !- Outside Boundary Condition
,                    !- Outside Boundary Condition Object
NoSun,               !- Sun Exposure
NoWind,              !- Wind Exposure
,                    !- View Factor to Ground
4,                   !- Number of Vertices
12.1330909462833,    !- Vertex 1 X-coordinate {m}
9.09981820971244,    !- Vertex 1 Y-coordinate {m}
-0.457317073170732, !- Vertex 1 Z-coordinate {m}
0,                   !- Vertex 2 X-coordinate {m}
9.09981820971244,    !- Vertex 2 Y-coordinate {m}
-0.457317073170732, !- Vertex 2 Z-coordinate {m}
0,                   !- Vertex 3 X-coordinate {m}
9.09981820971244,    !- Vertex 3 Y-coordinate {m}
0,                   !- Vertex 3 Z-coordinate {m}
12.1330909462833,    !- Vertex 4 X-coordinate {m}
9.09981820971244,    !- Vertex 4 Y-coordinate {m}
0;                   !- Vertex 4 Z-coordinate {m}

```

BuildingSurface:Detailed,

```

BGWall_upper_sdl,    !- Name
Wall,                !- Surface Type
Crawl Wall,          !- Construction Name
crawlspcse_unit1,    !- Zone Name
Outdoors,            !- Outside Boundary Condition
,                    !- Outside Boundary Condition Object
SunExposed,          !- Sun Exposure
WindExposed,         !- Wind Exposure
,                    !- View Factor to Ground
4,                   !- Number of Vertices
0,                   !- Vertex 1 X-coordinate {m}
9.09981820971244,    !- Vertex 1 Y-coordinate {m}
0,                   !- Vertex 1 Z-coordinate {m}
0,                   !- Vertex 2 X-coordinate {m}
0,                   !- Vertex 2 Y-coordinate {m}
0,                   !- Vertex 2 Z-coordinate {m}
0,                   !- Vertex 3 X-coordinate {m}
0,                   !- Vertex 3 Y-coordinate {m}
0.152439024390244,   !- Vertex 3 Z-coordinate {m}
0,                   !- Vertex 4 X-coordinate {m}
9.09981820971244,    !- Vertex 4 Y-coordinate {m}
0.152439024390244;   !- Vertex 4 Z-coordinate {m}

```

BuildingSurface:Detailed,

```

BGWall_lower_sdl,    !- Name
Wall,                !- Surface Type
Crawl Wall,          !- Construction Name
crawlspcse_unit1,    !- Zone Name
Ground,              !- Outside Boundary Condition
,                    !- Outside Boundary Condition Object
NoSun,               !- Sun Exposure
NoWind,              !- Wind Exposure

```

```

,           !- View Factor to Ground
4,         !- Number of Vertices
0,         !- Vertex 1 X-coordinate {m}
9.09981820971244, !- Vertex 1 Y-coordinate {m}
-0.457317073170732, !- Vertex 1 Z-coordinate {m}
0,         !- Vertex 2 X-coordinate {m}
0,         !- Vertex 2 Y-coordinate {m}
-0.457317073170732, !- Vertex 2 Z-coordinate {m}
0,         !- Vertex 3 X-coordinate {m}
0,         !- Vertex 3 Y-coordinate {m}
0,         !- Vertex 3 Z-coordinate {m}
0,         !- Vertex 4 X-coordinate {m}
9.09981820971244, !- Vertex 4 Y-coordinate {m}
0;        !- Vertex 4 Z-coordinate {m}

```

BuildingSurface:Detailed,

```

Extfloor_unit1, !- Name
Floor,         !- Surface Type
Interior Floor, !- Construction Name
crawlspace_unit1, !- Zone Name
Ground,       !- Outside Boundary Condition
,            !- Outside Boundary Condition Object
NoSun,       !- Sun Exposure
NoWind,     !- Wind Exposure
0.0,        !- View Factor to Ground
4,         !- Number of Vertices
0,         !- Vertex 1 X-coordinate {m}
0,         !- Vertex 1 Y-coordinate {m}
-0.457317073170732, !- Vertex 1 Z-coordinate {m}
0,         !- Vertex 2 X-coordinate {m}
9.09981820971244, !- Vertex 2 Y-coordinate {m}
-0.457317073170732, !- Vertex 2 Z-coordinate {m}
12.1330909462833, !- Vertex 3 X-coordinate {m}
9.09981820971244, !- Vertex 3 Y-coordinate {m}
-0.457317073170732, !- Vertex 3 Z-coordinate {m}
12.1330909462833, !- Vertex 4 X-coordinate {m}
0,         !- Vertex 4 Y-coordinate {m}
-0.457317073170732; !- Vertex 4 Z-coordinate {m}

```

!- ===== ALL OBJECTS IN CLASS: WINDOW =====

Window,

```

Window_ldf_1.unit1, !- Name
Exterior Window, !- Construction Name
Wall_ldf_1.unit1, !- Building Surface Name
,            !- Frame and Divider Name
1,         !- Multiplier
1,         !- Starting X Coordinate {m}
0.914634146341463, !- Starting Z Coordinate {m}
2.70719591738945, !- Length {m}
1.52439024390244; !- Height {m}

```

Window,

```

Window_ldb_1.unit1, !- Name
Exterior Window, !- Construction Name
Wall_ldb_1.unit1, !- Building Surface Name
,            !- Frame and Divider Name
1,         !- Multiplier
2.74190122145512, !- Starting X Coordinate {m}
0.914634146341463, !- Starting Z Coordinate {m}
2.70719591738945, !- Length {m}
1.52439024390244; !- Height {m}

```

Window,

```

Window_sdr_1.unit1, !- Name
Exterior Window, !- Construction Name
Wall_sdr_1.unit1, !- Building Surface Name
,            !- Frame and Divider Name
1,         !- Multiplier

```


1, !- Starting X Coordinate {m}
0.914634146341463, !- Starting Z Coordinate {m}
2.70719591738945, !- Length {m}
1.52439024390244; !- Height {m}

Window,

Window_sdl_1.unit1, !- Name
Exterior Window, !- Construction Name
Wall_sdl_1.unit1, !- Building Surface Name
, !- Frame and Divider Name
1, !- Multiplier
1, !- Starting X Coordinate {m}
0.914634146341463, !- Starting Z Coordinate {m}
2.70719591738945, !- Length {m}
1.52439024390244; !- Height {m}

Window,

Window_ldf_2.unit1, !- Name
Exterior Window, !- Construction Name
Wall_ldf_2.unit1, !- Building Surface Name
, !- Frame and Divider Name
1, !- Multiplier
1, !- Starting X Coordinate {m}
0.914634146341463, !- Starting Z Coordinate {m}
2.70719591738945, !- Length {m}
1.52439024390244; !- Height {m}

Window,

Window_ldb_2.unit1, !- Name
Exterior Window, !- Construction Name
Wall_ldb_2.unit1, !- Building Surface Name
, !- Frame and Divider Name
1, !- Multiplier
2.74190122145512, !- Starting X Coordinate {m}
0.914634146341463, !- Starting Z Coordinate {m}
2.70719591738945, !- Length {m}
1.52439024390244; !- Height {m}

Window,

Window_sdr_2.unit1, !- Name
Exterior Window, !- Construction Name
Wall_sdr_2.unit1, !- Building Surface Name
, !- Frame and Divider Name
1, !- Multiplier
1, !- Starting X Coordinate {m}
0.914634146341463, !- Starting Z Coordinate {m}
2.70719591738945, !- Length {m}
1.52439024390244; !- Height {m}

Window,

Window_sdl_2.unit1, !- Name
Exterior Window, !- Construction Name
Wall_sdl_2.unit1, !- Building Surface Name
, !- Frame and Divider Name
1, !- Multiplier
1, !- Starting X Coordinate {m}
0.914634146341463, !- Starting Z Coordinate {m}
2.70719591738945, !- Length {m}
1.52439024390244; !- Height {m}

!- ===== ALL OBJECTS IN CLASS: DOOR =====

Door,

Door_ldb_unit1, !- Name
Exterior Door, !- Construction Name
Wall_ldb_1.unit1, !- Building Surface Name
1, !- Multiplier
0.5, !- Starting X Coordinate {m}
0, !- Starting Z Coordinate {m}

1.74190122145513, !- Length {m}
2.13414634146341; !- Height {m}

!- ===== ALL OBJECTS IN CLASS: WINDOWSHADINGCONTROL =====

WindowShadingControl,
Shades-living_unit1, !- Name
living_unit1, !- Zone Name
1, !- Shading Control Sequence Number
InteriorBlind, !- Shading Type
window_w_blinds, !- Construction with Shading Name
OnIfScheduleAllows, !- Shading Control Type
shading_2012iecc, !- Schedule Name
, !- Setpoint {W/m2, W or deg C}
Yes, !- Shading Control Is Scheduled
No, !- Glare Control Is Active
, !- Shading Device Material Name
, !- Type of Slat Angle Control for Blinds
, !- Slat Angle Schedule Name
, !- Setpoint 2 {W/m2 or deg C}
, !- Daylighting Control Object Name
Sequential, !- Multiple Surface Control Type
Window_ldf_1.unit1, !- Fenestration Surface 1 Name
Window_ldb_1.unit1, !- Fenestration Surface 2 Name
Window_sdr_1.unit1, !- Fenestration Surface 3 Name
Window_sdl_1.unit1, !- Fenestration Surface 4 Name
Window_ldf_2.unit1, !- Fenestration Surface 5 Name
Window_ldb_2.unit1, !- Fenestration Surface 6 Name
Window_sdr_2.unit1, !- Fenestration Surface 7 Name
Window_sdl_2.unit1; !- Fenestration Surface 8 Name

!- ===== ALL OBJECTS IN CLASS: INTERNALMASS =====

InternalMass,
Internalmass_unit1, !- Name
InteriorFurnishings, !- Construction Name
living_unit1, !- Zone or ZoneList Name
9.89591078066915; !- Surface Area {m2}

!- ===== ALL OBJECTS IN CLASS: SHADING:BUILDING:DETAILED =====

Shading:Building:Detailed,
PV 01, !- Name
always_avail, !- Transmittance Schedule Name
4, !- Number of Vertices
10.2402114, !- Vertex 1 X-coordinate {m}
0.089178218455, !- Vertex 1 Y-coordinate {m}
6.112603860979, !- Vertex 1 Z-coordinate {m}
10.2402114, !- Vertex 2 X-coordinate {m}
4.458910922759, !- Vertex 2 Y-coordinate {m}
7.382125909208, !- Vertex 2 Z-coordinate {m}
0.51633122807, !- Vertex 3 X-coordinate {m}
4.458910922759, !- Vertex 3 Y-coordinate {m}
7.382125909208, !- Vertex 3 Z-coordinate {m}
0.51633122807, !- Vertex 4 X-coordinate {m}
0.089178218455, !- Vertex 4 Y-coordinate {m}
6.112603860979; !- Vertex 4 Z-coordinate {m}

!- ===== ALL OBJECTS IN CLASS: SHADING:OVERHANG =====

Shading:Overhang,
Overhang_sdr_1.unit1, !- Name
Window_sdr_1.unit1, !- Window or Door Name
0, !- Height above Window or Door {m}
90, !- Tilt Angle from Window/Door {deg}
0, !- Left extension from Window/Door Width {m}

```

0,          !- Right extension from Window/Door Width {m}
0.0152439024390244;  !- Depth {m}

Shading:Overhang,
  Overhang_sdr_2.unit1,  !- Name
  Window_sdr_2.unit1,  !- Window or Door Name
0,          !- Height above Window or Door {m}
90,        !- Tilt Angle from Window/Door {deg}
0,          !- Left extension from Window/Door Width {m}
0,          !- Right extension from Window/Door Width {m}
0.0152439024390244;  !- Depth {m}

!- ===== ALL OBJECTS IN CLASS: GROUNDHEATTRANSFER:CONTROL =====

GroundHeatTransfer:Control,
  gtp_control,          !- Name
  no,                  !- Run Basement Preprocessor
  yes;                 !- Run Slab Preprocessor

!- ===== ALL OBJECTS IN CLASS: GROUNDHEATTRANSFER:SLAB:MATERIALS =====

GroundHeatTransfer:Slab:Materials,
  2,          !- NMAT: Number of materials
  0.16,       !- ALBEDO: Surface Albedo: No Snow
  0.4,        !- ALBEDO: Surface Albedo: Snow
  0.9,        !- EPSLW: Surface Emissivity: No Snow
  0.9,        !- EPSLW: Surface Emissivity: Snow
  0.75,       !- Z0: Surface Roughness: No Snow {cm}
  0.05,       !- Z0: Surface Roughness: Snow {cm}
  6,          !- HIN: Indoor HConv: Downward Flow {W/m2-K}
  9;          !- HIN: Indoor HConv: Upward {W/m2-K}

!- ===== ALL OBJECTS IN CLASS: GROUNDHEATTRANSFER:SLAB:MATLPROPS =====

GroundHeatTransfer:Slab:MatlProps,
  2300,       !- RHO: Slab Material density {kg/m3}
  1200,       !- RHO: Soil Density {kg/m3}
  650,        !- CP: Slab CP {J/kg-K}
  1200,       !- CP: Soil CP {J/kg-K}
  0.9,        !- TCON: Slab k {W/m-K}
  1;          !- TCON: Soil k {W/m-K}

!- ===== ALL OBJECTS IN CLASS: GROUNDHEATTRANSFER:SLAB:BOUNDCONDS =====

GroundHeatTransfer:Slab:BoundConds,
  FALSE,      !- EVTR: Is surface evapotranspiration modeled
  TRUE,       !- FIXBC: is the lower boundary at a fixed temperature
  10,         !- TDEEPin {C}
  FALSE;      !- USRHflag: Is the ground surface h specified by the user?

!- ===== ALL OBJECTS IN CLASS: GROUNDHEATTRANSFER:SLAB:BLDGPROPS =====

GroundHeatTransfer:Slab:BldgProps,
  10,         !- IYRS: Number of years to iterate
  0,          !- Shape: Slab shape
  4,          !- HBLDG: Building height {m}
  22.22,     !- TIN1: January Indoor Average Temperature Setpoint {C}
  22.22,     !- TIN2: February Indoor Average Temperature Setpoint {C}
  22.22,     !- TIN3: March Indoor Average Temperature Setpoint {C}
  22.22,     !- TIN4: April Indoor Average Temperature Setpoint {C}
  22.22,     !- TIN5: May Indoor Average Temperature Setpoint {C}
  23.88,     !- TIN6: June Indoor Average Temperature Setpoint {C}
  23.88,     !- TIN7: July Indoor Average Temperature Setpoint {C}
  23.88,     !- TIN8: August Indoor Average Temperature Setpoint {C}
  23.88,     !- TIN9: September Indoor Average Temperature Setpoint {C}

```

```

22.22,      !- TIN10: October Indoor Average Temperature Setpoint {C}
22.22,      !- TIN11: November Indoor Average Temperature Setpoint {C}
22.22,      !- TIN12: December Indoor Average Temperature Setpoint {C}
0,          !- TINamp: Daily Indoor sine wave variation amplitude {deltaC}
0.1;       !- ConvTol: Convergence Tolerance

!- ===== ALL OBJECTS IN CLASS: GROUNDHEATTRANSFER:SLAB:INSULATION =====

GroundHeatTransfer:Slab:Insulation,
0,          !- RINS: R value of under slab insulation {m2-K/W}
0,          !- DINS: Width of strip of under slab insulation {m}
0,          !- RVINS: R value of vertical insulation {m2-K/W}
1.5,       !- ZVINS: Depth of vertical insulation {m}
0;         !- IVINS: Flag: Is there vertical insulation

!- ===== ALL OBJECTS IN CLASS: GROUNDHEATTRANSFER:SLAB:EQUIVALENTSLAB =====

GroundHeatTransfer:Slab:EquivalentSlab,
2.59947574997227, !- APRatio: The area to perimeter ratio for this slab {m}
0.1,       !- SLABDEPTH: Thickness of slab on grade {m}
15,        !- CLEARANCE: Distance from edge of slab to domain edge {m}
15;        !- ZCLEARANCE: Distance from bottom of slab to domain bottom {m}

!- ===== ALL OBJECTS IN CLASS: GROUNDHEATTRANSFER:BASEMENT:SIMPARAMETERS =====

GroundHeatTransfer:Basement:SimParameters,
0.1,       !- F: Multiplier for the ADI solution
15;        !- IYRS: Maximum number of yearly iterations:

!- ===== ALL OBJECTS IN CLASS: GROUNDHEATTRANSFER:BASEMENT:MATLPROPS =====

GroundHeatTransfer:Basement:MatlProps,
6,          !- NMAT: Number of materials in this domain
2243,      !- Density for Foundation Wall {kg/m3}
2243,      !- density for Floor Slab {kg/m3}
311,       !- density for Ceiling {kg/m3}
1500,      !- density for Soil {kg/m3}
2000,      !- density for Gravel {kg/m3}
449,       !- density for Wood {kg/m3}
880,       !- Specific heat for foundation wall {J/kg-K}
880,       !- Specific heat for floor slab {J/kg-K}
1530,      !- Specific heat for ceiling {J/kg-K}
840,       !- Specific heat for soil {J/kg-K}
720,       !- Specific heat for gravel {J/kg-K}
1530,      !- Specific heat for wood {J/kg-K}
1.4,       !- Thermal conductivity for foundation wall {W/m-K}
1.4,       !- Thermal conductivity for floor slab {W/m-K}
0.09,      !- Thermal conductivity for ceiling {W/m-K}
1.1,       !- thermal conductivity for soil {W/m-K}
1.9,       !- thermal conductivity for gravel {W/m-K}
0.12;      !- thermal conductivity for wood {W/m-K}

!- ===== ALL OBJECTS IN CLASS: GROUNDHEATTRANSFER:BASEMENT:INSULATION =====

GroundHeatTransfer:Basement:Insulation,
2.28929665762688, !- REXT: R Value of any exterior insulation {m2-K/W}
True;        !- INSFULL: Flag: Is the wall fully insulated?

!- ===== ALL OBJECTS IN CLASS: GROUNDHEATTRANSFER:BASEMENT:SURFACEPROPS =====

GroundHeatTransfer:Basement:SurfaceProps,
0.16,      !- ALBEDO: Surface albedo for No snow conditions
0.4,       !- ALBEDO: Surface albedo for snow conditions
0.94,      !- EPSLN: Surface emissivity No Snow

```

```

0.86,      !- EPSLN: Surface emissivity with Snow
6,         !- VEGHT: Surface roughness No snow conditions {cm}
0.25,     !- VEGHT: Surface roughness Snow conditions {cm}
False;    !- PET: Flag, Potential evapotranspiration on?

!- ===== ALL OBJECTS IN CLASS: GROUNDHEATTRANSFER:BASEMENT:BLDGDATA =====

GroundHeatTransfer:Basement:BldgData,
0.20000006162114,    !- DWALL: Wall thickness {m}
0.243828108701145,  !- DSLAB: Floor slab thickness {m}
0.3,                !- DGRAVXY: Width of gravel pit beside basement wall {m}
0.2,                !- DGRAVZN: Gravel depth extending above the floor slab {m}
0.1;               !- DGRAVZP: Gravel depth below the floor slab {m}

!- ===== ALL OBJECTS IN CLASS: GROUNDHEATTRANSFER:BASEMENT:INTERIOR =====

GroundHeatTransfer:Basement:Interior,
True,             !- COND: Flag: Is the basement conditioned?
0.92,            !- HIN: Downward convection only heat transfer coefficient {W/m2-K}
4.04,            !- HIN: Upward convection only heat transfer coefficient {W/m2-K}
3.08,            !- HIN: Horizontal convection only heat transfer coefficient {W/m2-K}
6.13,            !- HIN: Downward combined (convection and radiation) heat transfer coefficient {W/m2-K}
9.26,            !- HIN: Upward combined (convection and radiation) heat transfer coefficient {W/m2-K}
8.29;           !- HIN: Horizontal combined (convection and radiation) heat transfer coefficient {W/m2-K}

!- ===== ALL OBJECTS IN CLASS: GROUNDHEATTRANSFER:BASEMENT:COMBLDG =====

GroundHeatTransfer:Basement:ComBldg,
21,             !- January average temperature {C}
21,             !- February average temperature {C}
21,             !- March average temperature {C}
21,             !- April average temperature {C}
24,             !- May average temperature {C}
24,             !- June average temperature {C}
24,             !- July average temperature {C}
24,             !- August average temperature {C}
24,             !- September average temperature {C}
24,             !- October average temperature {C}
21,             !- November average temperature {C}
21,             !- December average temperature {C}
21;            !- Daily variation sine wave amplitude {deltaC}

!- ===== ALL OBJECTS IN CLASS: GROUNDHEATTRANSFER:BASEMENT:EQUIVSLAB =====

GroundHeatTransfer:Basement:EquipSlab,
2.59947574997227, !- APRatio: The area to perimeter ratio for this slab {m}
True;             !- EquipSizing: Flag

!- ===== ALL OBJECTS IN CLASS: GROUNDHEATTRANSFER:BASEMENT:EQUIVAUTOGRID =====

GroundHeatTransfer:Basement:EquipAutoGrid,
15,              !- CLEARANCE: Distance from outside of wall to edge of 3-D ground domain {m}
0.1,            !- SlabDepth: Thickness of the floor slab {m}
1.21914054350572; !- BaseDepth: Depth of the basement wall below grade {m}

!- ===== ALL OBJECTS IN CLASS: PEOPLE =====

People,
people_unit1,   !- Name
living_unit1,   !- Zone or ZoneList Name
Occupancy,      !- Number of People Schedule Name
People,         !- Number of People Calculation Method
3,              !- Number of People
,               !- People per Zone Floor Area {person/m2}

```

```

,          !- Zone Floor Area per Person {m2/person}
0,         !- Fraction Radiant
autocalculate,    !- Sensible Heat Fraction
activity_sch,     !- Activity Level Schedule Name
,          !- Carbon Dioxide Generation Rate {m3/s-W}
No,          !- Enable ASHRAE 55 Comfort Warnings
ZoneAveraged;    !- Mean Radiant Temperature Calculation Type

```

!- ===== ALL OBJECTS IN CLASS: LIGHTS =====

```

Lights,
Living Hardwired Lighting1, !- Name
living_unit1,      !- Zone or ZoneList Name
InteriorLightingHE, !- Schedule Name
Watts/Area,       !- Design Level Calculation Method
,                !- Lighting Level {W}
1.15188831694383, !- Watts per Zone Floor Area {W/m2}
,                !- Watts per Person {W/person}
0,               !- Return Air Fraction
0.6,            !- Fraction Radiant
0.2,            !- Fraction Visible
0;              !- Fraction Replaceable

```

```

Lights,
Living Plug-in Lighting1, !- Name
living_unit1,      !- Zone or ZoneList Name
InteriorLightingHE, !- Schedule Name
Watts/Area,       !- Design Level Calculation Method
,                !- Lighting Level {W}
0.478467682251396, !- Watts per Zone Floor Area {W/m2}
,                !- Watts per Person {W/person}
0,               !- Return Air Fraction
0.6,            !- Fraction Radiant
0.2,            !- Fraction Visible
0;              !- Fraction Replaceable

```

!- ===== ALL OBJECTS IN CLASS: ELECTRIC EQUIPMENT =====

```

!
! $Id: appliances.template,v 1.7 2018/12/13 20:12:17 d3a061 Exp $
!
```

```

ElectricEquipment,
dishwasher1,      !- Name
living_unit1,    !- Zone or ZoneList Name
always_off,      !- Schedule Name
EquipmentLevel,  !- Design Level Calculation Method
65.698787492023, !- Design Level {W}
,                !- Watts per Zone Floor Area {W/m2}
,                !- Watts per Person {W/person}
0.15,           !- Fraction Latent
0.6,            !- Fraction Radiant
0.25,           !- Fraction Lost
dishwasher;     !- End-Use Subcategory

```

```

ElectricEquipment,
refrigerator1,   !- Name
living_unit1,    !- Zone or ZoneList Name
always_off,      !- Schedule Name
EquipmentLevel,  !- Design Level Calculation Method
91.0575745202123, !- Design Level {W}
,                !- Watts per Zone Floor Area {W/m2}
,                !- Watts per Person {W/person}
0,               !- Fraction Latent
1,               !- Fraction Radiant
0,               !- Fraction Lost
refrigerator;    !- End-Use Subcategory

```

```

ElectricEquipment,

```

```

clotheswasher1,    !- Name
living_unit1,     !- Zone or ZoneList Name
always_off,       !- Schedule Name
EquipmentLevel,   !- Design Level Calculation Method
28.4784377542718, !- Design Level {W}
,                !- Watts per Zone Floor Area {W/m2}
,                !- Watts per Person {W/person}
0,               !- Fraction Latent
0.8,            !- Fraction Radiant
0.2,            !- Fraction Lost
clotheswasher;    !- End-Use Subcategory

ElectricEquipment,
electric_dryer1,  !- Name
living_unit1,     !- Zone or ZoneList Name
always_off,       !- Schedule Name
EquipmentLevel,   !- Design Level Calculation Method
213.064557285022, !- Design Level {W}
,                !- Watts per Zone Floor Area {W/m2}
,                !- Watts per Person {W/person}
0.05,           !- Fraction Latent
0.15,           !- Fraction Radiant
0.8,            !- Fraction Lost
electric_dryer;  !- End-Use Subcategory

ElectricEquipment,
electric_range1, !- Name
living_unit1,     !- Zone or ZoneList Name
always_off,       !- Schedule Name
EquipmentLevel,   !- Design Level Calculation Method
248.154224774405, !- Design Level {W}
,                !- Watts per Zone Floor Area {W/m2}
,                !- Watts per Person {W/person}
0.3,            !- Fraction Latent
0.4,            !- Fraction Radiant
0.3,            !- Fraction Lost
electric_range;  !- End-Use Subcategory

ElectricEquipment,
television1,     !- Name
living_unit1,     !- Zone or ZoneList Name
always_off,       !- Schedule Name
EquipmentLevel,   !- Design Level Calculation Method
0,              !- Design Level {W}
,                !- Watts per Zone Floor Area {W/m2}
,                !- Watts per Person {W/person}
0,              !- Fraction Latent
1,              !- Fraction Radiant
0,              !- Fraction Lost
television;      !- End-Use Subcategory

ElectricEquipment,
electric_mels1,  !- Name
living_unit1,     !- Zone or ZoneList Name
always_off,       !- Schedule Name
EquipmentLevel,   !- Design Level Calculation Method
567.464237516869, !- Design Level {W}
,                !- Watts per Zone Floor Area {W/m2}
,                !- Watts per Person {W/person}
0.06,           !- Fraction Latent
0.69,           !- Fraction Radiant
0.25,           !- Fraction Lost
electric_mels;   !- End-Use Subcategory

ElectricEquipment,
IECC_Adj1,       !- Name
living_unit1,     !- Zone or ZoneList Name
always_off,       !- Schedule Name
Watts/Area,      !- Design Level Calculation Method
,                !- Design Level {W}

```

1.54356736989469, !- Watts per Zone Floor Area {W/m2}
 , !- Watts per Person {W/person}
 0.0624390461422629, !- Fraction Latent
 0.41190936353998, !- Fraction Radiant
 0.251045347957769, !- Fraction Lost
 IECC_adj; !- End-Use Subcategory

!- ===== ALL OBJECTS IN CLASS: ZONEINFILTRATION:EFFECTIVELEAKAGEAREA =====

ZoneInfiltration:EffectiveLeakageArea,
 Living_ShermanGrimsrud_unit1, !- Name
 living_unit1, !- Zone Name
 always_avail, !- Schedule Name
 357.402730122675, !- Effective Air Leakage Area {cm2}
 0.00029, !- Stack Coefficient
 0.000231; !- Wind Coefficient

ZoneInfiltration:EffectiveLeakageArea,
 AtticVent_unit1, !- Name
 attic_unit1, !- Zone Name
 always_avail, !- Schedule Name
 370, !- Effective Air Leakage Area {cm2}
 0.00029, !- Stack Coefficient
 0.000231; !- Wind Coefficient

ZoneInfiltration:EffectiveLeakageArea,
 CrawlVent_unit1, !- Name
 crawlspace_unit1, !- Zone Name
 always_avail, !- Schedule Name
 370, !- Effective Air Leakage Area {cm2}
 0.00029, !- Stack Coefficient
 0.000231; !- Wind Coefficient

!- ===== ALL OBJECTS IN CLASS: ZONEVENTILATION:DESIGNFLOWRATE =====

ZoneVentilation:DesignFlowRate,
 Ventilation_unit1, !- Name
 living_unit1, !- Zone or ZoneList Name
 always_avail, !- Schedule Name
 Flow/Zone, !- Design Flow Rate Calculation Method
 0.0283168464628752, !- Design Flow Rate {m3/s}
 , !- Flow Rate per Zone Floor Area {m3/s-m2}
 , !- Flow Rate per Person {m3/s-person}
 , !- Air Changes per Hour {1/hr}
 Exhaust, !- Ventilation Type
 0, !- Fan Pressure Rise {Pa}
 1, !- Fan Total Efficiency
 1, !- Constant Term Coefficient
 0, !- Temperature Term Coefficient
 0, !- Velocity Term Coefficient
 0, !- Velocity Squared Term Coefficient
 -100, !- Minimum Indoor Temperature {C}
 , !- Minimum Indoor Temperature Schedule Name
 100, !- Maximum Indoor Temperature {C}
 , !- Maximum Indoor Temperature Schedule Name
 -100, !- Delta Temperature {deltaC}
 , !- Delta Temperature Schedule Name
 -100, !- Minimum Outdoor Temperature {C}
 , !- Minimum Outdoor Temperature Schedule Name
 100, !- Maximum Outdoor Temperature {C}
 , !- Maximum Outdoor Temperature Schedule Name
 40; !- Maximum Wind Speed {m/s}

!- ===== ALL OBJECTS IN CLASS: EXTERIOR:LIGHTS =====

Exterior:Lights,
 Exterior-Lights_unit1, !- Name

ExteriorLighting, !- Schedule Name
47.3410734408986, !- Design Level {W}
, !- Control Option
Exterior-Lights; !- End-Use Subcategory

Exterior:Lights,
Garage-Lights_unit1, !- Name
InteriorLightingHE, !- Schedule Name
7.84186549576547, !- Design Level {W}
, !- Control Option
Garage-Lights; !- End-Use Subcategory

!- ===== ALL OBJECTS IN CLASS: DESIGNSPECIFICATION:OUTDOORAIR =====

DesignSpecification:OutdoorAir,
SZ_DSOA_living_unit1, !- Name
Flow/Zone, !- Outdoor Air Method
0, !- Outdoor Air Flow per Person {m3/s-person}
, !- Outdoor Air Flow per Zone Floor Area {m3/s-m2}
0.0283168464628752; !- Outdoor Air Flow per Zone {m3/s}

!- ===== ALL OBJECTS IN CLASS: SIZING:ZONE =====

Sizing:Zone,
living_unit1, !- Zone or ZoneList Name
SupplyAirTemperature, !- Zone Cooling Design Supply Air Temperature Input Method
12, !- Zone Cooling Design Supply Air Temperature {C}
, !- Zone Cooling Design Supply Air Temperature Difference {deltaC}
SupplyAirTemperature, !- Zone Heating Design Supply Air Temperature Input Method
50, !- Zone Heating Design Supply Air Temperature {C}
, !- Zone Heating Design Supply Air Temperature Difference {deltaC}
0.008, !- Zone Cooling Design Supply Air Humidity Ratio {kgWater/kgDryAir}
0.008, !- Zone Heating Design Supply Air Humidity Ratio {kgWater/kgDryAir}
SZ_DSOA_living_unit1, !- Design Specification Outdoor Air Object Name
, !- Zone Heating Sizing Factor
, !- Zone Cooling Sizing Factor
DesignDay, !- Cooling Design Air Flow Method
, !- Cooling Design Air Flow Rate {m3/s}
0.000762, !- Cooling Minimum Air Flow per Zone Floor Area {m3/s-m2}
, !- Cooling Minimum Air Flow {m3/s}
, !- Cooling Minimum Air Flow Fraction
DesignDay, !- Heating Design Air Flow Method
, !- Heating Design Air Flow Rate {m3/s}
0.002032, !- Heating Maximum Air Flow per Zone Floor Area {m3/s-m2}
0.1415762, !- Heating Maximum Air Flow {m3/s}
0.3; !- Heating Maximum Air Flow Fraction

!- ===== ALL OBJECTS IN CLASS: SIZING:SYSTEM =====

Sizing:System,
Central_System_unit1, !- AirLoop Name
Sensible, !- Type of Load to Size On
autosize, !- Design Outdoor Air Flow Rate {m3/s}
1, !- Central Heating Maximum System Air Flow Ratio
7, !- Preheat Design Temperature {C}
0.008, !- Preheat Design Humidity Ratio {kgWater/kgDryAir}
11, !- Precool Design Temperature {C}
0.008, !- Precool Design Humidity Ratio {kgWater/kgDryAir}
12, !- Central Cooling Design Supply Air Temperature {C}
50, !- Central Heating Design Supply Air Temperature {C}
NonCoincident, !- Type of Zone Sum to Use
No, !- 100% Outdoor Air in Cooling
No, !- 100% Outdoor Air in Heating
0.008, !- Central Cooling Design Supply Air Humidity Ratio {kgWater/kgDryAir}
0.008, !- Central Heating Design Supply Air Humidity Ratio {kgWater/kgDryAir}
DesignDay, !- Cooling Supply Air Flow Rate Method
, !- Cooling Supply Air Flow Rate {m3/s}

```

,          !- Cooling Supply Air Flow Rate Per Floor Area {m3/s-m2}
,          !- Cooling Fraction of Autosized Cooling Supply Air Flow Rate
,          !- Cooling Supply Air Flow Rate Per Unit Cooling Capacity {m3/s-W}
DesignDay, !- Heating Supply Air Flow Rate Method
,          !- Heating Supply Air Flow Rate {m3/s}
,          !- Heating Supply Air Flow Rate Per Floor Area {m3/s-m2}
,          !- Heating Fraction of Autosized Heating Supply Air Flow Rate
,          !- Heating Fraction of Autosized Cooling Supply Air Flow Rate
,          !- Heating Supply Air Flow Rate Per Unit Heating Capacity {m3/s-W}
,          !- System Outdoor Air Method
1.0,       !- Zone Maximum Outdoor Air Fraction {dimensionless}
CoolingDesignCapacity, !- Cooling Design Capacity Method
autosize,  !- Cooling Design Capacity {W}
,          !- Cooling Design Capacity Per Floor Area {W/m2}
,          !- Fraction of Autosized Cooling Design Capacity
HeatingDesignCapacity, !- Heating Design Capacity Method
autosize,  !- Heating Design Capacity {W}
,          !- Heating Design Capacity Per Floor Area {W/m2}
,          !- Fraction of Autosized Heating Design Capacity
OnOff;     !- Central Cooling Capacity Control Method

```

!- ===== ALL OBJECTS IN CLASS: SIZING:PLANT =====

```

Sizing:Plant,
  DHW Loop_unit1,    !- Plant or Condenser Loop Name
  Heating,           !- Loop Type
  48.8888888888889,  !- Design Loop Exit Temperature {C}
  5.55555555555556; !- Loop Design Temperature Difference {deltaC}

```

!- ===== ALL OBJECTS IN CLASS: ZONECONTROL:THERMOSTAT =====

```

ZoneControl:Thermostat,
  Zone Thermostat_unit1, !- Name
  living_unit1,         !- Zone or ZoneList Name
  zone_control_type,    !- Control Type Schedule Name
  ThermostatSetpoint:DualSetpoint, !- Control 1 Object Type
  thermostat_living Dual SP Control; !- Control 1 Name

```

!- ===== ALL OBJECTS IN CLASS: THERMOSTATSETPOINT:DUALSETPOINT =====

```

ThermostatSetpoint:DualSetpoint,
  thermostat_living Dual SP Control, !- Name
  myStartHeating,    !- Heating Setpoint Temperature Schedule Name
  myStartCooling;    !- Cooling Setpoint Temperature Schedule Name

```

!- ===== ALL OBJECTS IN CLASS: AIRTERMINAL:SINGLEDUCT:UNCONTROLLED =====

```

AirTerminal:SingleDuct:Uncontrolled,
  ZoneDirectAir_unit1, !- Name
  always_avail,        !- Availability Schedule Name
  Zone Inlet Node_unit1, !- Zone Supply Air Node Name
  autosize;           !- Maximum Air Flow Rate {m3/s}

```

!- ===== ALL OBJECTS IN CLASS: ZONEHVAC:EQUIPMENTLIST =====

```

ZoneHVAC:EquipmentList,
  ZONEEQUIPMENT_unit1, !- Name
  SequentialLoad,       !- Load Distribution Scheme
  AirTerminal:SingleDuct:Uncontrolled, !- Zone Equipment 1 Object Type
  ZoneDirectAir_unit1, !- Zone Equipment 1 Name
  1,                    !- Zone Equipment 1 Cooling Sequence
  1,                    !- Zone Equipment 1 Heating or No-Load Sequence
  ,                    !- Zone Equipment 1 Sequential Cooling Fraction Schedule Name
  ;                    !- Zone Equipment 1 Sequential Heating Fraction Schedule Name

```

!- ===== ALL OBJECTS IN CLASS: ZONEHVAC:EQUIPMENTCONNECTIONS =====

ZoneHVAC:EquipmentConnections,
living_unit1, !- Zone Name
ZoneEquipment_unit1, !- Zone Conditioning Equipment List Name
zone inlet nodes_unit1, !- Zone Air Inlet Node or NodeList Name
, !- Zone Air Exhaust Node or NodeList Name
Zone Node_unit1, !- Zone Air Node Name
Zone Outlet Node_unit1; !- Zone Return Air Node or NodeList Name

!- ===== ALL OBJECTS IN CLASS: FAN:ONOFF =====

Fan:OnOff,
Supply Fan_unit1, !- Name
always_avail, !- Availability Schedule Name
0.50054, !- Fan Total Efficiency
400, !- Pressure Rise {Pa}
autosize, !- Maximum Flow Rate {m3/s}
0.863, !- Motor Efficiency
1, !- Motor In Airstream Fraction
air loop inlet node_unit1, !- Air Inlet Node Name
cooling coil air inlet node_unit1, !- Air Outlet Node Name
, !- Fan Power Ratio Function of Speed Ratio Curve Name
, !- Fan Efficiency Ratio Function of Speed Ratio Curve Name
General; !- End-Use Subcategory

!- ===== ALL OBJECTS IN CLASS: COIL:COOLING:DX:SINGLESPEED =====

Coil:Cooling:DX:SingleSpeed,
DX Cooling Coil_unit1, !- Name
always_avail, !- Availability Schedule Name
autosize, !- Gross Rated Total Cooling Capacity {W}
autosize, !- Gross Rated Sensible Heat Ratio
4.06853019625951, !- Gross Rated Cooling COP {W/W}
autosize, !- Rated Air Flow Rate {m3/s}
, !- Rated Evaporator Fan Power Per Volume Flow Rate {W/(m3/s)}
Cooling Coil Air Inlet Node_unit1, !- Air Inlet Node Name
Heating Coil Air Inlet Node_unit1, !- Air Outlet Node Name
HPACCoolCapFT, !- Total Cooling Capacity Function of Temperature Curve Name
HPACCoolCapFFF, !- Total Cooling Capacity Function of Flow Fraction Curve Name
HPACCOOLEIRFT, !- Energy Input Ratio Function of Temperature Curve Name
HPACCOOLEIRFFF, !- Energy Input Ratio Function of Flow Fraction Curve Name
HPACCOOLPLFFPLR; !- Part Load Fraction Correlation Curve Name

!- ===== ALL OBJECTS IN CLASS: COIL:HEATING:ELECTRIC =====

Coil:Heating:Electric,
Supp Heating Coil_unit1, !- Name
always_avail, !- Availability Schedule Name
1, !- Efficiency
autosize, !- Nominal Capacity {W}
Supp Heating Coil Air Inlet Node_unit1, !- Air Inlet Node Name
Air Loop Outlet Node_unit1; !- Air Outlet Node Name

!- ===== ALL OBJECTS IN CLASS: COIL:HEATING:DX:SINGLESPEED =====

Coil:Heating:DX:SingleSpeed,
Main DX Heating Coil_unit1, !- Name
always_avail, !- Availability Schedule Name
autosize, !- Gross Rated Heating Capacity {W}
3.69308080013886, !- Gross Rated Heating COP {W/W}
autosize, !- Rated Air Flow Rate {m3/s}
, !- Rated Supply Fan Power Per Volume Flow Rate {W/(m3/s)}
Heating Coil Air Inlet Node_unit1, !- Air Inlet Node Name
Supp Heating Coil Air Inlet Node_unit1, !- Air Outlet Node Name

HPACHeatCapFT, !- Heating Capacity Function of Temperature Curve Name
 HPACHeatCapFFF, !- Heating Capacity Function of Flow Fraction Curve Name
 HPACHeatEIRFT, !- Energy Input Ratio Function of Temperature Curve Name
 HPACHeatEIRFFF, !- Energy Input Ratio Function of Flow Fraction Curve Name
 HPACCOOLPLFFPLR, !- Part Load Fraction Correlation Curve Name
 Defrost_EIR_FT, !- Defrost Energy Input Ratio Function of Temperature Curve Name
 -17.78, !- Minimum Outdoor Dry-Bulb Temperature for Compressor Operation {C}
 , !- Outdoor Dry-Bulb Temperature to Turn On Compressor {C}
 5.0, !- Maximum Outdoor Dry-Bulb Temperature for Defrost Operation {C}
 200.0, !- Crankcase Heater Capacity {W}
 10.0, !- Maximum Outdoor Dry-Bulb Temperature for Crankcase Heater Operation {C}
 ReverseCycle, !- Defrost Strategy
 OnDemand; !- Defrost Control

!- ===== ALL OBJECTS IN CLASS: AIRLOOPHVAC:UNITARYHEATPUMP:AIRTOAIR =====

AirLoopHVAC:UnitaryHeatPump:AirToAir,
 Heat_Pump_unit1, !- Name
 always_avail, !- Availability Schedule Name
 Air Loop Inlet node_unit1, !- Air Inlet Node Name
 Air Loop Outlet Node_unit1, !- Air Outlet Node Name
 autosize, !- Cooling Supply Air Flow Rate {m3/s}
 autosize, !- Heating Supply Air Flow Rate {m3/s}
 0.0, !- No Load Supply Air Flow Rate {m3/s}
 living_unit1, !- Controlling Zone or Thermostat Location
 Fan:OnOff, !- Supply Air Fan Object Type
 Supply Fan_unit1, !- Supply Air Fan Name
 Coil:Heating:DX:SingleSpeed, !- Heating Coil Object Type
 Main DX Heating Coil_unit1, !- Heating Coil Name
 Coil:Cooling:DX:SingleSpeed, !- Cooling Coil Object Type
 DX Cooling Coil_unit1, !- Cooling Coil Name
 Coil:Heating:Electric, !- Supplemental Heating Coil Object Type
 Supp Heating Coil_unit1, !- Supplemental Heating Coil Name
 50, !- Maximum Supply Air Temperature from Supplemental Heater {C}
 10, !- Maximum Outdoor Dry-Bulb Temperature for Supplemental Heater Operation {C}
 BlowThrough, !- Fan Placement
 fan_cycle; !- Supply Air Fan Operating Mode Schedule Name

!- ===== ALL OBJECTS IN CLASS: AIRLOOPHVAC =====

AirLoopHVAC,
 Central System_unit1, !- Name
 , !- Controller List Name
 availability list, !- Availability Manager List Name
 autosize, !- Design Supply Air Flow Rate {m3/s}
 Air Loop Branches_unit1, !- Branch List Name
 , !- Connector List Name
 Air Loop Inlet Node_unit1, !- Supply Side Inlet Node Name
 Return Air Mixer Outlet_unit1, !- Demand Side Outlet Node Name
 Zone Equipment Inlet Node_unit1, !- Demand Side Inlet Node Names
 Air Loop Outlet Node_unit1; !- Supply Side Outlet Node Names

!- ===== ALL OBJECTS IN CLASS: AIRLOOPHVAC:ZONESPLITTER =====

AirLoopHVAC:ZoneSplitter,
 Zone Supply Air Splitter_unit1, !- Name
 Zone Equipment Inlet Node_unit1, !- Inlet Node Name
 Zone Inlet Node_unit1; !- Outlet 1 Node Name

!- ===== ALL OBJECTS IN CLASS: AIRLOOPHVAC:SUPPLYPATH =====

AirLoopHVAC:SupplyPath,
 SupplyPath_unit1, !- Name
 Zone Equipment Inlet Node_unit1, !- Supply Air Path Inlet Node Name
 AirLoopHVAC:ZoneSplitter,!- Component 1 Object Type
 Zone Supply Air Splitter_unit1; !- Component 1 Name

!- ===== ALL OBJECTS IN CLASS: AIRLOOPHVAC:ZONEMIXER =====

AirLoopHVAC:ZoneMixer,
Zone Return Air Mixer_unit1, !- Name
Return Air Mixer Outlet_unit1, !- Outlet Node Name
Zone Outlet Node_unit1; !- Inlet 1 Node Name

!- ===== ALL OBJECTS IN CLASS: AIRLOOPHVAC:RETURNPATH =====

AirLoopHVAC:ReturnPath,
ReturnPath_unit1, !- Name
Return Air Mixer Outlet_unit1, !- Return Air Path Outlet Node Name
AirLoopHVAC:ZoneMixer, !- Component 1 Object Type
Zone Return Air Mixer_unit1; !- Component 1 Name

!- ===== ALL OBJECTS IN CLASS: BRANCH =====

Branch,
Air Loop Main Branch_unit1, !- Name
, !- Pressure Drop Curve Name
AirLoopHVAC:UnitaryHeatPump:AirToAir, !- Component 1 Object Type
Heat Pump_unit1, !- Component 1 Name
Air Loop Inlet Node_unit1, !- Component 1 Inlet Node Name
Air Loop Outlet Node_unit1; !- Component 1 Outlet Node Name

Branch,
Mains Inlet Branch_unit1,!- Name
, !- Pressure Drop Curve Name
Pump:VariableSpeed, !- Component 1 Object Type
Mains Pressure_unit1, !- Component 1 Name
Mains Inlet Node_unit1, !- Component 1 Inlet Node Name
Mains Pressure Outlet Node_unit1; !- Component 1 Outlet Node Name

Branch,
DHW Supply Outlet Branch_unit1, !- Name
, !- Pressure Drop Curve Name
Pipe:Adiabatic, !- Component 1 Object Type
DHW Supply Outlet Pipe_unit1, !- Component 1 Name
DHW Supply Outlet Pipe Inlet Node_unit1, !- Component 1 Inlet Node Name
DHW Supply Outlet Node_unit1; !- Component 1 Outlet Node Name

Branch,
DHW Demand Inlet Branch_unit1, !- Name
, !- Pressure Drop Curve Name
Pipe:Adiabatic, !- Component 1 Object Type
DHW Demand Inlet Pipe_unit1, !- Component 1 Name
DHW Demand Inlet Node_unit1, !- Component 1 Inlet Node Name
DHW Demand Inlet Pipe Outlet Node_unit1; !- Component 1 Outlet Node Name

Branch,
Water Sink Branch_unit1, !- Name
, !- Pressure Drop Curve Name
WaterUse:Connections, !- Component 1 Object Type
DHW Sinks_unit1, !- Component 1 Name
Water Sink Inlet Node_unit1, !- Component 1 Inlet Node Name
Water Sink outlet Node_unit1; !- Component 1 Outlet Node Name

Branch,
Water Shower Branch_unit1, !- Name
, !- Pressure Drop Curve Name
WaterUse:Connections, !- Component 1 Object Type
DHW Showers_unit1, !- Component 1 Name
Water Shower Inlet Node_unit1, !- Component 1 Inlet Node Name
Water Shower Outlet Node_unit1; !- Component 1 Outlet Node Name

Branch,

Water ClothesWasher Branch_unit1, !- Name
, !- Pressure Drop Curve Name
WaterUse:Connections, !- Component 1 Object Type
DHW ClothesWasher_unit1, !- Component 1 Name
Water ClothesWasher Inlet Node_unit1, !- Component 1 Inlet Node Name
Water ClothesWasher Outlet Node_unit1; !- Component 1 Outlet Node Name

Branch,

Water Dishwasher Branch_unit1, !- Name
, !- Pressure Drop Curve Name
WaterUse:Connections, !- Component 1 Object Type
DHW DishWasher_unit1, !- Component 1 Name
Water DishWasher Inlet Node_unit1, !- Component 1 Inlet Node Name
Water DishWasher outlet Node_unit1; !- Component 1 Outlet Node Name

Branch,

Water Bath Branch_unit1, !- Name
, !- Pressure Drop Curve Name
WaterUse:Connections, !- Component 1 Object Type
DHW Baths_unit1, !- Component 1 Name
Water Bath Inlet Node_unit1, !- Component 1 Inlet Node Name
Water bath Outlet Node_unit1; !- Component 1 Outlet Node Name

Branch,

Mains Makeup Branch_unit1, !- Name
, !- Pressure Drop Curve Name
Pipe:Adiabatic, !- Component 1 Object Type
Mains Makeup Pipe_unit1, !- Component 1 Name
Mains Makeup Pipe Inlet Node_unit1, !- Component 1 Inlet Node Name
Mains Makeup Node_unit1; !- Component 1 Outlet Node Name

Branch,

Water Heater Branch_unit1, !- Name
, !- Pressure Drop Curve Name
WaterHeater:Mixed, !- Component 1 Object Type
Water Heater_unit1, !- Component 1 Name
Water Heater Use Inlet Node_unit1, !- Component 1 Inlet Node Name
Water Heater Use Outlet Node_unit1; !- Component 1 Outlet Node Name

!- ===== ALL OBJECTS IN CLASS: BRANCHLIST =====

BranchList,

Air Loop Branches_unit1, !- Name
Air Loop Main Branch_unit1; !- Branch 1 Name

BranchList,

DHW Supply Branches_unit1, !- Name
Mains Inlet Branch_unit1, !- Branch 1 Name
Water Heater Branch_unit1, !- Branch 2 Name
DHW Supply Outlet Branch_unit1; !- Branch 3 Name

BranchList,

DHW Demand Branches_unit1, !- Name
DHW Demand Inlet Branch_unit1, !- Branch 1 Name
Water Sink Branch_unit1, !- Branch 2 Name
Water Shower Branch_unit1, !- Branch 3 Name
Water ClothesWasher Branch_unit1, !- Branch 4 Name
Water Dishwasher Branch_unit1, !- Branch 5 Name
Water Bath Branch_unit1, !- Branch 6 Name
Mains Makeup Branch_unit1; !- Branch 7 Name

!- ===== ALL OBJECTS IN CLASS: CONNECTOR:SPLITTER =====

Connector:Splitter,

DHW Demand Splitter_unit1, !- Name
DHW Demand Inlet Branch_unit1, !- Inlet Branch Name
Water Sink Branch_unit1, !- Outlet Branch 1 Name
Water Shower Branch_unit1, !- Outlet Branch 2 Name

```

Water ClothesWasher Branch_unit1, !- Outlet Branch 3 Name
Water Dishwasher Branch_unit1, !- Outlet Branch 4 Name
Water Bath Branch_unit1; !- Outlet Branch 5 Name

Connector:Splitter,
  DHW Supply Splitter_unit1, !- Name
  Mains Inlet Branch_unit1,!- Inlet Branch Name
  Water Heater Branch_unit1; !- Outlet Branch 1 Name

!- ===== ALL OBJECTS IN CLASS: CONNECTOR:MIXER =====

Connector:Mixer,
  DHW Demand Mixer_unit1, !- Name
  Mains Makeup Branch_unit1, !- Outlet Branch Name
  Water Sink Branch_unit1, !- Inlet Branch 1 Name
  Water Shower Branch_unit1, !- Inlet Branch 2 Name
  Water ClothesWasher Branch_unit1, !- Inlet Branch 3 Name
  Water Dishwasher Branch_unit1, !- Inlet Branch 4 Name
  Water Bath Branch_unit1; !- Inlet Branch 5 Name

Connector:Mixer,
  DHW Supply Mixer_unit1, !- Name
  DHW Supply Outlet Branch_unit1, !- Outlet Branch Name
  Water Heater Branch_unit1; !- Inlet Branch 1 Name

!- ===== ALL OBJECTS IN CLASS: CONNECTORLIST =====

ConnectorList,
  DHW Demand Connectors_unit1, !- Name
  Connector:Splitter,    !- Connector 1 Object Type
  DHW Demand Splitter_unit1, !- Connector 1 Name
  Connector:Mixer,      !- Connector 2 Object Type
  DHW Demand Mixer_unit1; !- Connector 2 Name

ConnectorList,
  DHW Supply Connectors_unit1, !- Name
  Connector:Splitter,    !- Connector 1 Object Type
  DHW Supply Splitter_unit1, !- Connector 1 Name
  Connector:Mixer,      !- Connector 2 Object Type
  DHW Supply Mixer_unit1; !- Connector 2 Name

!- ===== ALL OBJECTS IN CLASS: NODELIST =====

NodeList,
  Zone Inlet Nodes_unit1, !- Name
  Zone Inlet Node_unit1; !- Node 1 Name

!- ===== ALL OBJECTS IN CLASS: OUTDOORAIR:NODE =====

OutdoorAir:Node,
  outside air inlet node_unit1, !- Name
  0.914355407629293;    !- Height Above Ground {m}

!- ===== ALL OBJECTS IN CLASS: OUTDOORAIR:NODELIST =====

OutdoorAir:NodeList,
  outdoor air node_unit1; !- Node or NodeList Name 1

!- ===== ALL OBJECTS IN CLASS: PIPE:ADIABATIC =====

Pipe:Adiabatic,
  DHW Supply Outlet Pipe_unit1, !- Name
  DHW Supply Outlet Pipe Inlet Node_unit1, !- Inlet Node Name
  DHW Supply Outlet Node_unit1; !- Outlet Node Name

```

Pipe:Adiabatic,
 Mains Makeup Pipe_unit1, !- Name
 Mains Makeup Pipe Inlet Node_unit1, !- Inlet Node Name
 Mains Makeup Node_unit1; !- Outlet Node Name

Pipe:Adiabatic,
 DHW Demand Inlet Pipe_unit1, !- Name
 DHW Demand Inlet Node_unit1, !- Inlet Node Name
 DHW Demand Inlet Pipe Outlet Node_unit1; !- Outlet Node Name

!- ===== ALL OBJECTS IN CLASS: PUMP:VARIABLESPEED =====

Pump:VariableSpeed,
 Mains Pressure_unit1, !- Name
 Mains Inlet Node_unit1, !- Inlet Node Name
 Mains Pressure Outlet Node_unit1, !- Outlet Node Name
 autosize, !- Design Maximum Flow Rate {m3/s}
 1, !- Design Pump Head {Pa}
 0, !- Design Power Consumption {W}
 1, !- Motor Efficiency
 0, !- Fraction of Motor Inefficiencies to Fluid Stream
 0, !- Coefficient 1 of the Part Load Performance Curve
 1, !- Coefficient 2 of the Part Load Performance Curve
 0, !- Coefficient 3 of the Part Load Performance Curve
 0, !- Coefficient 4 of the Part Load Performance Curve
 0, !- Design Minimum Flow Rate {m3/s}
 Intermittent; !- Pump Control Type

!- ===== ALL OBJECTS IN CLASS: WATERHEATER:MIXED =====

WaterHeater:Mixed,
 Water Heater_unit1, !- Name
 0.196841372, !- Tank Volume {m3}
 dhw_setpt, !- Setpoint Temperature Schedule Name
 2, !- Deadband Temperature Difference {deltaC}
 50, !- Maximum Temperature Limit {C}
 Cycle, !- Heater Control Type
 autosize, !- Heater Maximum Capacity {W}
 0, !- Heater Minimum Capacity {W}
 0, !- Heater Ignition Minimum Flow Rate {m3/s}
 , !- Heater Ignition Delay {s}
 electricity, !- Heater Fuel Type
 1, !- Heater Thermal Efficiency
 , !- Part Load Factor Curve Name
 , !- Off Cycle Parasitic Fuel Consumption Rate {W}
 , !- Off Cycle Parasitic Fuel Type
 , !- Off Cycle Parasitic Heat Fraction to Tank
 , !- On Cycle Parasitic Fuel Consumption Rate {W}
 , !- On Cycle Parasitic Fuel Type
 , !- On Cycle Parasitic Heat Fraction to Tank
 Zone, !- Ambient Temperature Indicator
 , !- Ambient Temperature Schedule Name
 living_unit1, !- Ambient Temperature Zone Name
 , !- Ambient Temperature Outdoor Air Node Name
 1.23162315386149, !- Off Cycle Loss Coefficient to Ambient Temperature {W/K}
 1, !- Off Cycle Loss Fraction to Zone
 1.23162315386149, !- On Cycle Loss Coefficient to Ambient Temperature {W/K}
 1, !- On Cycle Loss Fraction to Zone
 0, !- Peak Use Flow Rate {m3/s}
 , !- Use Flow Rate Fraction Schedule Name
 , !- Cold Water Supply Temperature Schedule Name
 Water Heater use inlet node_unit1, !- Use Side Inlet Node Name
 Water Heater use outlet node_unit1, !- Use Side Outlet Node Name
 1, !- Use Side Effectiveness
 , !- Source Side Inlet Node Name
 , !- Source Side Outlet Node Name
 1, !- Source Side Effectiveness


```

autosize,      !- Use Side Design Flow Rate {m3/s}
0,            !- Source Side Design Flow Rate {m3/s}
1.5;         !- Indirect Water Heating Recovery Time {hr}

!- ===== ALL OBJECTS IN CLASS: WATERHEATER:SIZING =====

WaterHeater:Sizing,
  Water Heater_unit1,  !- WaterHeater Name
  ResidentialHUD-FHAMinimum, !- Design Mode
  ,                    !- Time Storage Can Meet Peak Draw {hr}
  ,                    !- Time for Tank Recovery {hr}
  ,                    !- Nominal Tank Volume for Autosizing Plant Connections {m3}
  3,                   !- Number of Bedrooms
  3;                   !- Number of Bathrooms

!- ===== ALL OBJECTS IN CLASS: PLANTLOOP =====

PlantLoop,
  DHW Loop_unit1,      !- Name
  Water,               !- Fluid Type
  ,                   !- User Defined Fluid Type
  DHW Loop Operation_unit1,!- Plant Equipment Operation Scheme Name
  DHW Supply Outlet Node_unit1, !- Loop Temperature Setpoint Node Name
  100,                !- Maximum Loop Temperature {C}
  0,                  !- Minimum Loop Temperature {C}
  autosize,           !- Maximum Loop Flow Rate {m3/s}
  0,                  !- Minimum Loop Flow Rate {m3/s}
  autocalculate,      !- Plant Loop Volume {m3}
  Mains Inlet Node_unit1, !- Plant Side Inlet Node Name
  DHW Supply Outlet Node_unit1, !- Plant Side Outlet Node Name
  DHW Supply Branches_unit1, !- Plant Side Branch List Name
  DHW Supply Connectors_unit1, !- Plant Side Connector List Name
  DHW Demand Inlet Node_unit1, !- Demand Side Inlet Node Name
  Mains Makeup Node_unit1, !- Demand Side Outlet Node Name
  DHW Demand Branches_unit1, !- Demand Side Branch List Name
  DHW Demand Connectors_unit1, !- Demand Side Connector List Name
  Optimal;           !- Load Distribution Scheme

!- ===== ALL OBJECTS IN CLASS: PLANTEQUIPMENTLIST =====

PlantEquipmentList,
  DHW Plant Equipment_unit1, !- Name
  WaterHeater:Mixed,        !- Equipment 1 Object Type
  Water Heater_unit1;       !- Equipment 1 Name

!- ===== ALL OBJECTS IN CLASS: PLANTEQUIPMENTOPERATION:HEATINGLOAD =====

PlantEquipmentOperation:HeatingLoad,
  DHW Control Scheme_unit1,!- Name
  0.0,                      !- Load Range 1 Lower Limit {W}
  10000000000000000,        !- Load Range 1 Upper Limit {W}
  DHW Plant Equipment_unit1; !- Range 1 Equipment List Name

!- ===== ALL OBJECTS IN CLASS: PLANTEQUIPMENTOPERATIONSCHMES =====

PlantEquipmentOperationSchemes,
  DHW Loop Operation_unit1,!- Name
  PlantEquipmentOperation:HeatingLoad, !- Control Scheme 1 Object Type
  DHW Control Scheme_unit1,!- Control Scheme 1 Name
  always_avail;            !- Control Scheme 1 Schedule Name

!- ===== ALL OBJECTS IN CLASS: EXTERNALINTERFACE =====

ExternalInterface,

```

```

FunctionalMockupUnitImport; !- Name of External Interface

!- ===== ALL OBJECTS IN CLASS: EXTERNALINTERFACE:FUNCTIONALMOCKUPUNITIMPORT =====

ExternalInterface:FunctionalMockupUnitImport,
  Joe_ep_fmu.fmu,      !- FMU File Name
  0,                  !- FMU Timeout {ms}
  1;                  !- FMU LoggingOn

!- ===== ALL OBJECTS IN CLASS: EXTERNALINTERFACE:FUNCTIONALMOCKUPUNITIMPORT:FROM:VARIABLE
=====

ExternalInterface:FunctionalMockupUnitImport:From:Variable,
  Whole Building,      !- Output:Variable Index Key Name
  Facility Net Purchased Electric Energy , !- Output:Variable Name
  Joe_ep_fmu.fmu,      !- FMU File Name
  Joe_ep_fmu,          !- FMU Instance Name
  epSendNetEnergy;     !- FMU Variable Name

ExternalInterface:FunctionalMockupUnitImport:From:Variable,
  living_unit1,        !- Output:Variable Index Key Name
  Zone Air Temperature, !- Output:Variable Name
  Joe_ep_fmu.fmu,      !- FMU File Name
  Joe_ep_fmu,          !- FMU Instance Name
  epSendZoneMeanAirTemp; !- FMU Variable Name

ExternalInterface:FunctionalMockupUnitImport:From:Variable,
  DaysOfWeek,          !- Output:Variable Index Key Name
  Schedule Value ,     !- Output:Variable Name
  Joe_ep_fmu.fmu,      !- FMU File Name
  Joe_ep_fmu,          !- FMU Instance Name
  epSendDayOfWeek;     !- FMU Variable Name

ExternalInterface:FunctionalMockupUnitImport:From:Variable,
  Environment,         !- Output:Variable Index Key Name
  Site Outdoor Air Drybulb Temperature, !- Output:Variable Name
  Joe_ep_fmu.fmu,      !- FMU File Name
  Joe_ep_fmu,          !- FMU Instance Name
  epSendOutdoorAirTemp; !- FMU Variable Name

ExternalInterface:FunctionalMockupUnitImport:From:Variable,
  living_unit1,        !- Output:Variable Index Key Name
  Zone Air Relative Humidity, !- Output:Variable Name
  Joe_ep_fmu.fmu,      !- FMU File Name
  Joe_ep_fmu,          !- FMU Instance Name
  epSendZoneHumidity;  !- FMU Variable Name

ExternalInterface:FunctionalMockupUnitImport:From:Variable,
  Whole Building,      !- Output:Variable Index Key Name
  Facility Total Purchased Electric Energy , !- Output:Variable Name
  Joe_ep_fmu.fmu,      !- FMU File Name
  Joe_ep_fmu,          !- FMU Instance Name
  epSendEnergyPurchased; !- FMU Variable Name

ExternalInterface:FunctionalMockupUnitImport:From:Variable,
  Whole Building,      !- Output:Variable Index Key Name
  Facility Total Surplus Electric Energy , !- Output:Variable Name
  Joe_ep_fmu.fmu,      !- FMU File Name
  Joe_ep_fmu,          !- FMU Instance Name
  epSendEnergySurplus; !- FMU Variable Name

ExternalInterface:FunctionalMockupUnitImport:From:Variable,
  Environment,         !- Output:Variable Index Key Name
  Site Direct Solar Radiation Rate per Area , !- Output:Variable Name
  Joe_ep_fmu.fmu,      !- FMU File Name
  Joe_ep_fmu,          !- FMU Instance Name
  epSendSolarRadiation; !- FMU Variable Name

```

```

ExternalInterface:FunctionalMockupUnitImport:From:Variable,
  living_unit1,      !- Output:Variable Index Key Name
  Zone Thermostat Heating Setpoint Temperature , !- Output:Variable Name
  Joe_ep_fmu.fmu,   !- FMU File Name
  Joe_ep_fmu,       !- FMU Instance Name
  epSendHeatingSetpoint; !- FMU Variable Name

```

```

ExternalInterface:FunctionalMockupUnitImport:From:Variable,
  living_unit1,      !- Output:Variable Index Key Name
  Zone Thermostat Cooling Setpoint Temperature , !- Output:Variable Name
  Joe_ep_fmu.fmu,   !- FMU File Name
  Joe_ep_fmu,       !- FMU Instance Name
  epSendCoolingSetpoint; !- FMU Variable Name

```

```

!- ===== ALL OBJECTS IN CLASS: EXTERNALINTERFACE:FUNCTIONALMOCKUPUNITIMPORT:TO:SCHEDULE
=====

```

```

ExternalInterface:FunctionalMockupUnitImport:To:Schedule,
  myStartHeating,   !- Name
  myTemps,          !- Schedule Type Limits Names
  Joe_ep_fmu.fmu,   !- FMU File Name
  Joe_ep_fmu,       !- FMU Instance Name
  epGetStartHeating, !- FMU Variable Name
  20;               !- Initial Value

```

```

ExternalInterface:FunctionalMockupUnitImport:To:Schedule,
  myStartCooling,   !- Name
  myTemps,          !- Schedule Type Limits Names
  Joe_ep_fmu.fmu,   !- FMU File Name
  Joe_ep_fmu,       !- FMU Instance Name
  epGetStartCooling, !- FMU Variable Name
  23;               !- Initial Value

```

```

!- ===== ALL OBJECTS IN CLASS: AVAILABILITYMANAGER:SCHEDULED =====

```

```

AvailabilityManager:Scheduled,
  System availability, !- Name
  always_avail;       !- Schedule Name

```

```

!- ===== ALL OBJECTS IN CLASS: AVAILABILITYMANAGERASSIGNMENTLIST =====

```

```

AvailabilityManagerAssignmentList,
  availability list, !- Name
  AvailabilityManager:Scheduled, !- Availability Manager 1 Object Type
  System availability; !- Availability Manager 1 Name

```

```

!- ===== ALL OBJECTS IN CLASS: SETPOINTMANAGER:SCHEDULED =====

```

```

SetpointManager:Scheduled,
  DHW Loop Setpoint Manager_unit1, !- Name
  Temperature, !- Control Variable
  DHWSupplySetpoint, !- Schedule Name
  DHW Supply Outlet Node_unit1; !- Setpoint Node or NodeList Name

```

```

!- ===== ALL OBJECTS IN CLASS: GENERATOR:PHOTOVOLTAIC =====

```

```

Generator:Photovoltaic,
  PV1, !- Name
  PV 01, !- Surface Name
  PhotovoltaicPerformance:Simple, !- Photovoltaic Performance Object Type
  PV, !- Module Performance Name
  Decoupled, !- Heat Transfer Integration Mode
  1, !- Number of Series Strings in Parallel {dimensionless}
  1; !- Number of Modules in Series {dimensionless}

```

!- ===== ALL OBJECTS IN CLASS: PHOTOVOLTAICPERFORMANCE:SIMPLE =====

PhotovoltaicPerformance:Simple,
PV, !- Name
1, !- Fraction of Surface Area with Active Solar Cells {dimensionless}
Fixed, !- Conversion Efficiency Input Mode
0.12, !- Value for Cell Efficiency if Fixed
always_avail; !- Efficiency Schedule Name

!- ===== ALL OBJECTS IN CLASS: ELECTRICLOADCENTER:GENERATORS =====

ElectricLoadCenter:Generators,
PV, !- Name
PV1, !- Generator 1 Name
Generator:Photovoltaic, !- Generator 1 Object Type
6800, !- Generator 1 Rated Electric Power Output {W}
always_avail, !- Generator 1 Availability Schedule Name
; !- Generator 1 Rated Thermal to Electrical Power Ratio

!- ===== ALL OBJECTS IN CLASS: ELECTRICLOADCENTER:INVERTER:SIMPLE =====

ElectricLoadCenter:Inverter:Simple,
Simple Ideal Inverter, !- Name
always_avail, !- Availability Schedule Name
, !- Zone Name
0, !- Radiative Fraction
0.96; !- Inverter Efficiency

!- ===== ALL OBJECTS IN CLASS: ELECTRICLOADCENTER:DISTRIBUTION =====

ElectricLoadCenter:Distribution,
Simple Electric Load Distribution, !- Name
PV, !- Generator List Name
Baseload, !- Generator Operation Scheme Type
0, !- Generator Demand Limit Scheme Purchased Electric Demand Limit {W}
, !- Generator Track Schedule Name Scheme Schedule Name
, !- Generator Track Meter Scheme Meter Name
DirectCurrentWithInverter, !- Electrical Buss Type
Simple Ideal Inverter, !- Inverter Name
, !- Electrical Storage Object Name
, !- Transformer Object Name
TrackFacilityElectricDemandStoreExcessOnSite, !- Storage Operation Scheme
, !- Storage Control Track Meter Name
, !- Storage Converter Object Name
1; !- Maximum Storage State of Charge Fraction

!- ===== ALL OBJECTS IN CLASS: WATERUSE:EQUIPMENT =====

WaterUse:Equipment,
Clothes Washer_unit1, !- Name
Domestic Hot Water, !- End-Use Subcategory
1.6219189818e-06, !- Peak Flow Rate {m3/s}
ClothesWasher, !- Flow Rate Fraction Schedule Name
CWWaterTempSchedule; !- Target Temperature Schedule Name

WaterUse:Equipment,
Dishwasher_unit1, !- Name
Domestic Hot Water, !- End-Use Subcategory
6.36685353e-07, !- Peak Flow Rate {m3/s}
Dishwasher, !- Flow Rate Fraction Schedule Name
DWWaterTempSchedule; !- Target Temperature Schedule Name

WaterUse:Equipment,
Sinks_unit1, !- Name
Domestic Hot Water, !- End-Use Subcategory

7.1934e-05, !- Peak Flow Rate {m3/s}
BA_sink_sch, !- Flow Rate Fraction Schedule Name
SSBWaterTempSchedule; !- Target Temperature Schedule Name

WaterUse:Equipment,
Showers_unit1, !- Name
Domestic Hot Water, !- End-Use Subcategory
0.000141975, !- Peak Flow Rate {m3/s}
BA_shower_sch, !- Flow Rate Fraction Schedule Name
SSBWaterTempSchedule; !- Target Temperature Schedule Name

WaterUse:Equipment,
Baths_unit1, !- Name
Domestic Hot Water, !- End-Use Subcategory
0.00027764, !- Peak Flow Rate {m3/s}
BA_bath_sch, !- Flow Rate Fraction Schedule Name
SSBWaterTempSchedule; !- Target Temperature Schedule Name

!- ===== ALL OBJECTS IN CLASS: WATERUSE:CONNECTIONS =====

WaterUse:Connections,
DHW Sinks_unit1, !- Name
Water Sink Inlet Node_unit1, !- Inlet Node Name
Water Sink Outlet Node_unit1, !- Outlet Node Name
, !- Supply Water Storage Tank Name
, !- Reclamation Water Storage Tank Name
, !- Hot Water Supply Temperature Schedule Name
, !- Cold Water Supply Temperature Schedule Name
None, !- Drain Water Heat Exchanger Type
, !- Drain Water Heat Exchanger Destination
, !- Drain Water Heat Exchanger U-Factor Times Area {W/K}
Sinks_unit1; !- Water Use Equipment 1 Name

WaterUse:Connections,
DHW Showers_unit1, !- Name
Water Shower Inlet Node_unit1, !- Inlet Node Name
Water Shower Outlet Node_unit1, !- Outlet Node Name
, !- Supply Water Storage Tank Name
, !- Reclamation Water Storage Tank Name
, !- Hot Water Supply Temperature Schedule Name
, !- Cold Water Supply Temperature Schedule Name
None, !- Drain Water Heat Exchanger Type
, !- Drain Water Heat Exchanger Destination
, !- Drain Water Heat Exchanger U-Factor Times Area {W/K}
Showers_unit1; !- Water Use Equipment 1 Name

WaterUse:Connections,
DHW ClothesWasher_unit1, !- Name
Water ClothesWasher Inlet Node_unit1, !- Inlet Node Name
Water ClothesWasher Outlet Node_unit1, !- Outlet Node Name
, !- Supply Water Storage Tank Name
, !- Reclamation Water Storage Tank Name
, !- Hot Water Supply Temperature Schedule Name
, !- Cold Water Supply Temperature Schedule Name
None, !- Drain Water Heat Exchanger Type
, !- Drain Water Heat Exchanger Destination
, !- Drain Water Heat Exchanger U-Factor Times Area {W/K}
Clothes Washer_unit1; !- Water Use Equipment 1 Name

WaterUse:Connections,
DHW DishWasher_unit1, !- Name
Water DishWasher Inlet Node_unit1, !- Inlet Node Name
Water DishWasher Outlet Node_unit1, !- Outlet Node Name
, !- Supply Water Storage Tank Name
, !- Reclamation Water Storage Tank Name
, !- Hot Water Supply Temperature Schedule Name
, !- Cold Water Supply Temperature Schedule Name
None, !- Drain Water Heat Exchanger Type
, !- Drain Water Heat Exchanger Destination

```

,          !- Drain Water Heat Exchanger U-Factor Times Area {W/K}
Dishwasher_unit1;    !- Water Use Equipment 1 Name

WaterUse:Connections,
  DHW Baths_unit1,    !- Name
  Water Bath Inlet Node_unit1, !- Inlet Node Name
  Water Bath Outlet Node_unit1, !- Outlet Node Name
,          !- Supply Water Storage Tank Name
,          !- Reclamation Water Storage Tank Name
,          !- Hot Water Supply Temperature Schedule Name
,          !- Cold Water Supply Temperature Schedule Name
None,        !- Drain Water Heat Exchanger Type
,          !- Drain Water Heat Exchanger Destination
,          !- Drain Water Heat Exchanger U-Factor Times Area {W/K}
Baths_unit1;    !- Water Use Equipment 1 Name

!- ===== ALL OBJECTS IN CLASS: CURVE:QUADRATIC =====

Curve:Quadratic,
  HPACCoolCapFFF,    !- Name
  0.8,        !- Coefficient1 Constant
  0.2,        !- Coefficient2 x
  0.0,        !- Coefficient3 x**2
  0.5,        !- Minimum Value of x
  1.5;       !- Maximum Value of x

Curve:Quadratic,
  HPACCOOLEIRFFF,    !- Name
  1.156,       !- Coefficient1 Constant
  -0.1816,     !- Coefficient2 x
  0.0256,     !- Coefficient3 x**2
  0.5,        !- Minimum Value of x
  1.5;       !- Maximum Value of x

Curve:Quadratic,
  HPACCOOLPLFFPLR,  !- Name
  0.85,       !- Coefficient1 Constant
  0.15,       !- Coefficient2 x
  0.0,        !- Coefficient3 x**2
  0.0,        !- Minimum Value of x
  1.0;       !- Maximum Value of x

Curve:Quadratic,
  HPACHeatEIRFFF,   !- Name
  1.3824,     !- Coefficient1 Constant
  -0.4336,    !- Coefficient2 x
  0.0512,    !- Coefficient3 x**2
  0.0,        !- Minimum Value of x
  1.0;       !- Maximum Value of x

Curve:Quadratic,
  ACCoolingCAPFFF,  !- Name
  0.718605468, !- Coefficient1 Constant
  0.410099989, !- Coefficient2 x
  -0.128705457, !- Coefficient3 x**2
  0,          !- Minimum Value of x
  1.5,       !- Maximum Value of x
  0,          !- Minimum Curve Output
  2,          !- Maximum Curve Output
  Dimensionless, !- Input Unit Type for X
  Dimensionless; !- Output Unit Type

Curve:Quadratic,
  ACCoolingEIRFFF,  !- Name
  1.32299905,    !- Coefficient1 Constant
  -0.477711207,  !- Coefficient2 x
  0.154712157,   !- Coefficient3 x**2
  0,             !- Minimum Value of x
  1.5,          !- Maximum Value of x

```

0, !- Minimum Curve Output
2, !- Maximum Curve Output
Dimensionless, !- Input Unit Type for X
Dimensionless; !- Output Unit Type

! Part Load Fraction curve as a function of Part Load Ratio is default from
! Table 6. BEopt AC Rated Value Inputs of NREL report NREL/TP-5500-56354
Curve:Quadratic,

ACCoolingPLFFPLR, !- Name
0.90, !- Coefficient1 Constant
0.10, !- Coefficient2 x
0.0, !- Coefficient3 x**2
0.0, !- Minimum Value of x
1.0; !- Maximum Value of x

Curve:Quadratic,

HPCoolingCAPFFF, !- Name
0.718664047, !- Coefficient1 Constant
0.41797409, !- Coefficient2 x
-0.136638137, !- Coefficient3 x**2
0, !- Minimum Value of x
1.5, !- Maximum Value of x
0, !- Minimum Curve Output
2, !- Maximum Curve Output
Dimensionless, !- Input Unit Type for X
Dimensionless; !- Output Unit Type

Curve:Quadratic,

HPCoolingEIRFFF, !- Name
1.143487507, !- Coefficient1 Constant
-0.13943972, !- Coefficient2 x
-0.004047787, !- Coefficient3 x**2
0, !- Minimum Value of x
1.5, !- Maximum Value of x
0, !- Minimum Curve Output
2, !- Maximum Curve Output
Dimensionless, !- Input Unit Type for X
Dimensionless; !- Output Unit Type

! Part Load Fraction curve as a function of Part Load Ratio is default curve
! from Table 6. BEopt AC Rated Value Inputs of NREL report NREL/TP-5500-56354
Curve:Quadratic,

HPCOOLPLFFPLR, !- Name
0.90, !- Coefficient1 Constant
0.10, !- Coefficient2 x
0.0, !- Coefficient3 x**2
0.0, !- Minimum Value of x
1.0; !- Maximum Value of x

Curve:Quadratic,

HPHeatingCAPFFF, !- Name
0.694045465, !- Coefficient1 Constant
0.474207981, !- Coefficient2 x
-0.168253446, !- Coefficient3 x**2
0, !- Minimum Value of x
1.5, !- Maximum Value of x
0, !- Minimum Curve Output
2, !- Maximum Curve Output
Dimensionless, !- Input Unit Type for X
Dimensionless; !- Output Unit Type

Curve:Quadratic,

HPHeatingEIRFFF, !- Name
2.185418751, !- Coefficient1 Constant
-1.942827919, !- Coefficient2 x
0.757409168, !- Coefficient3 x**2
0, !- Minimum Value of x
1.5, !- Maximum Value of x
0, !- Minimum Curve Output
2, !- Maximum Curve Output

Dimensionless, !- Input Unit Type for X
Dimensionless; !- Output Unit Type

! Part Load Fraction curve as a function of Part Load Ratio is default curve
! from Table 6. BEopt AC Rated Value Inputs of NREL report NREL/TP-5500-56354

Curve:Quadratic,
 HPHeatPLFFPLR, !- Name
 0.90, !- Coefficient1 Constant
 0.10, !- Coefficient2 x
 0.0, !- Coefficient3 x**2
 0.0, !- Minimum Value of x
 1.0; !- Maximum Value of x

Curve:Quadratic,
 HPLowStageHeatingCAPFFF, !- Name
 0.741466907, !- Coefficient1 Constant
 0.378645444, !- Coefficient2 x
 -0.119754733, !- Coefficient3 x**2
 0, !- Minimum Value of x
 1.5, !- Maximum Value of x
 0, !- Minimum Curve Output
 2, !- Maximum Curve Output
 Dimensionless, !- Input Unit Type for X
 Dimensionless; !- Output Unit Type

Curve:Quadratic,
 HPLowStageHeatingEIRFFF, !- Name
 2.153618211, !- Coefficient1 Constant
 -1.737190609, !- Coefficient2 x
 0.584269478, !- Coefficient3 x**2
 0, !- Minimum Value of x
 1.5, !- Maximum Value of x
 0, !- Minimum Curve Output
 2, !- Maximum Curve Output
 Dimensionless, !- Input Unit Type for X
 Dimensionless; !- Output Unit Type

Curve:Quadratic,
 HPHighStageHeatingCAPFFF, !- Name
 0.76634609, !- Coefficient1 Constant
 0.32840943, !- Coefficient2 x
 -0.094701495, !- Coefficient3 x**2
 0, !- Minimum Value of x
 1.5, !- Maximum Value of x
 0, !- Minimum Curve Output
 2, !- Maximum Curve Output
 Dimensionless, !- Input Unit Type for X
 Dimensionless; !- Output Unit Type

Curve:Quadratic,
 HPHighStageHeatingEIRFFF, !- Name
 2.001041353, !- Coefficient1 Constant
 -1.58869128, !- Coefficient2 x
 0.587593517, !- Coefficient3 x**2
 0, !- Minimum Value of x
 1.5, !- Maximum Value of x
 0, !- Minimum Curve Output
 2, !- Maximum Curve Output
 Dimensionless, !- Input Unit Type for X
 Dimensionless; !- Output Unit Type

! Part Load Fraction curve as a function of Part Load Ratio is default curve
! from Table 6. BEopt AC Rated Value Inputs of NREL report NREL/TP-5500-56354

Curve:Quadratic,
 HP2StageHeatingPLFFPLR, !- Name
 0.93, !- Coefficient1 Constant
 0.07, !- Coefficient2 x
 0.0, !- Coefficient3 x**2
 0.0, !- Minimum Value of x
 1.0; !- Maximum Value of x

Curve:Quadratic,
 HPLowStageCoolingCAPFFF, !- Name
 0.655239515, !- Coefficient1 Constant
 0.511655216, !- Coefficient2 x
 -0.166894731, !- Coefficient3 x**2
 0, !- Minimum Value of x
 1.5, !- Maximum Value of x
 0, !- Minimum Curve Output
 2, !- Maximum Curve Output
 Dimensionless, !- Input Unit Type for X
 Dimensionless; !- Output Unit Type

Curve:Quadratic,
 HPLowStageCoolingEIRFFF, !- Name
 1.639108268, !- Coefficient1 Constant
 -0.998953996, !- Coefficient2 x
 0.359845728, !- Coefficient3 x**2
 0, !- Minimum Value of x
 1.5, !- Maximum Value of x
 0, !- Minimum Curve Output
 2, !- Maximum Curve Output
 Dimensionless, !- Input Unit Type for X
 Dimensionless; !- Output Unit Type

Curve:Quadratic,
 HPHighStageCoolingCAPFFF, !- Name
 0.618281092, !- Coefficient1 Constant
 0.569060264, !- Coefficient2 x
 -0.187341356, !- Coefficient3 x**2
 0, !- Minimum Value of x
 1.5, !- Maximum Value of x
 0, !- Minimum Curve Output
 2, !- Maximum Curve Output
 Dimensionless, !- Input Unit Type for X
 Dimensionless; !- Output Unit Type

Curve:Quadratic,
 HPHighStageCoolingEIRFFF, !- Name
 1.570774717, !- Coefficient1 Constant
 -0.914152018, !- Coefficient2 x
 0.343377302, !- Coefficient3 x**2
 0, !- Minimum Value of x
 1.5, !- Maximum Value of x
 0, !- Minimum Curve Output
 2, !- Maximum Curve Output
 Dimensionless, !- Input Unit Type for X
 Dimensionless; !- Output Unit Type

! Part Load Fraction curve as a function of Part Load Ratio is default curve
 ! from Table 6. BEopt AC Rated Value Inputs of NREL report NREL/TP-5500-56354

Curve:Quadratic,
 HP2StageCoolingPLFFPLR, !- Name
 0.93, !- Coefficient1 Constant
 0.07, !- Coefficient2 x
 0.0, !- Coefficient3 x**2
 0.0, !- Minimum Value of x
 1.0; !- Maximum Value of x

Curve:Quadratic,
 ACLowStageCoolingCAPFFF, !- Name
 0.65673024, !- Coefficient1 Constant
 0.516470835, !- Coefficient2 x
 -0.172887149, !- Coefficient3 x**2
 0, !- Minimum Value of x
 1.5, !- Maximum Value of x
 0, !- Minimum Curve Output
 2, !- Maximum Curve Output
 Dimensionless, !- Input Unit Type for X
 Dimensionless; !- Output Unit Type

Curve:Quadratic,
 ACLowStageCoolingEIRFFF, !- Name
 1.562945114, !- Coefficient1 Constant
 -0.791859997, !- Coefficient2 x
 0.230030877, !- Coefficient3 x**2
 0, !- Minimum Value of x
 1.5, !- Maximum Value of x
 0, !- Minimum Curve Output
 2, !- Maximum Curve Output
 Dimensionless, !- Input Unit Type for X
 Dimensionless; !- Output Unit Type

Curve:Quadratic,
 ACHighStageCoolingCAPFFF, !- Name
 0.690334551, !- Coefficient1 Constant
 0.464383753, !- Coefficient2 x
 -0.154507638, !- Coefficient3 x**2
 0, !- Minimum Value of x
 1.5, !- Maximum Value of x
 0, !- Minimum Curve Output
 2, !- Maximum Curve Output
 Dimensionless, !- Input Unit Type for X
 Dimensionless; !- Output Unit Type

Curve:Quadratic,
 ACHighStageCoolingEIRFFF, !- Name
 1.31565404, !- Coefficient1 Constant
 -0.482467162, !- Coefficient2 x
 0.166239001, !- Coefficient3 x**2
 0, !- Minimum Value of x
 1.5, !- Maximum Value of x
 0, !- Minimum Curve Output
 2, !- Maximum Curve Output
 Dimensionless, !- Input Unit Type for X
 Dimensionless; !- Output Unit Type

! Part Load Fraction curve as a function of Part Load Ratio is default from
 ! Table 6. BEopt AC Rated Value Inputs of NREL report NREL/TP-5500-56354

Curve:Quadratic,
 AC2StageCoolingPLFFPLR, !- Name
 0.93, !- Coefficient1 Constant
 0.07, !- Coefficient2 x
 0.0, !- Coefficient3 x**2
 0.0, !- Minimum Value of x
 1.0; !- Maximum Value of x

! NREL benchmark Cool-PLF-fPLR

Curve:Quadratic,
 Cool-PLF-fPLR, !- Name
 0.80141423, !- Coefficient1 Constant
 0.23744685, !- Coefficient2 x
 -0.0393773, !- Coefficient3 x**2
 0, !- Minimum Value of x
 1, !- Maximum Value of x
 0.7, !- Minimum Curve Output
 1; !- Maximum Curve Output

!- ===== ALL OBJECTS IN CLASS: CURVE:CUBIC =====

Curve:Cubic,
 HPACHeatCapFT, !- Name
 0.758746, !- Coefficient1 Constant
 0.027626, !- Coefficient2 x
 0.000148716, !- Coefficient3 x**2
 0.0000034992, !- Coefficient4 x**3
 -20.0, !- Minimum Value of x
 20.0; !- Maximum Value of x

```

Curve:Cubic,
  HPACHeatCapFFF,      !- Name
  0.84,                !- Coefficient1 Constant
  0.16,                !- Coefficient2 x
  0.0,                 !- Coefficient3 x**2
  0.0,                 !- Coefficient4 x**3
  0.5,                 !- Minimum Value of x
  1.5;                 !- Maximum Value of x

```

```

Curve:Cubic,
  HPACHeatEIRFT,      !- Name
  1.19248,             !- Coefficient1 Constant
  -0.0300438,          !- Coefficient2 x
  0.00103745,          !- Coefficient3 x**2
  -0.000023328,        !- Coefficient4 x**3
  -20.0,               !- Minimum Value of x
  20.0;                !- Maximum Value of x

```

```

Curve:Cubic,
  Fan-EIR-fPLR,       !- Name
  0.00000000,          !- Coefficient1 Constant
  1.00000000,          !- Coefficient2 x
  0.00000000,          !- Coefficient3 x**2
  0.00000000,          !- Coefficient4 x**3
  0,                   !- Minimum Value of x
  1,                   !- Maximum Value of x
  0,                   !- Minimum Curve Output
  1;                   !- Maximum Curve Output

```

```

Curve:Cubic,
  ConstantCubic,      !- Name
  1,                   !- Coefficient1 Constant
  0,                   !- Coefficient2 x
  0,                   !- Coefficient3 x**2
  0,                   !- Coefficient4 x**3
  -100,                !- Minimum Value of x
  100;                 !- Maximum Value of x

```

!- ===== ALL OBJECTS IN CLASS: CURVE:BIQUADRATIC =====

```

! Heating Capacity as a Function of Temperature Curve (Air Wet-bulb T, Condenser Water Inlet T)
! Colmac HPA 2000 Air Source Heat Pump Water Heater
! Rating point: 29.4 C (85 F) Entering Air Dry-bulb Temperature
!               22.2 C (72 F) Entering Air Wet-bulb Temperature
!               56.7 C (132.2 F) Entering Condenser Water Temperature
! Capacity = 17028.6 W (58117 BTUH) water heating capacity
! COP = 3.37 W/W (12.21 BTUH/W) water heating efficiency
!

```

```

Curve:Biquadratic,
  HPWHHeatingCapFTemp, !- Name
  0.369827,             !- Coefficient1 Constant
  0.043341,             !- Coefficient2 x
  -0.00023,             !- Coefficient3 x**2
  0.000466,             !- Coefficient4 y
  0.000026,             !- Coefficient5 y**2
  -0.00027,             !- Coefficient6 x*y
  0.0,                  !- Minimum Value of x
  40.0,                 !- Maximum Value of x
  20.0,                 !- Minimum Value of y
  90.0,                 !- Maximum Value of y
  ,                      !- Minimum Curve Output
  ,                      !- Maximum Curve Output
  Temperature,          !- Input Unit Type for X
  Temperature,          !- Input Unit Type for Y
  Dimensionless;        !- Output Unit Type

```

```

Curve:Biquadratic,
  HPWHHeatingCOPFTemp, !- Name
  1.19713,              !- Coefficient1 Constant

```

```

0.077849,      !- Coefficient2 x
-0.0000016,    !- Coefficient3 x**2
-0.02675,      !- Coefficient4 y
0.000296,      !- Coefficient5 y**2
-0.00112,      !- Coefficient6 x*y
0.0,           !- Minimum Value of x
40.0,          !- Maximum Value of x
20.0,          !- Minimum Value of y
90.0,          !- Maximum Value of y
,              !- Minimum Curve Output
,              !- Maximum Curve Output
Temperature,    !- Input Unit Type for X
Temperature,    !- Input Unit Type for Y
Dimensionless;  !- Output Unit Type

```

```

Curve:Biquadratic,
HPACCoolCapFT,  !- Name
0.766956,      !- Coefficient1 Constant
0.0107756,     !- Coefficient2 x
-0.0000414703, !- Coefficient3 x**2
0.00134961,    !- Coefficient4 y
-0.000261144, !- Coefficient5 y**2
0.000457488,   !- Coefficient6 x*y
12.77778,     !- Minimum Value of x
23.88889,     !- Maximum Value of x
21.11111,     !- Minimum Value of y
46.11111;     !- Maximum Value of y

```

```

Curve:Biquadratic,
HPACCOOLEIRFT, !- Name
0.297145,      !- Coefficient1 Constant
0.0430933,     !- Coefficient2 x
-0.000748766,  !- Coefficient3 x**2
0.00597727,    !- Coefficient4 y
0.000482112,   !- Coefficient5 y**2
-0.000956448,  !- Coefficient6 x*y
12.77778,     !- Minimum Value of x
23.88889,     !- Maximum Value of x
21.11111,     !- Minimum Value of y
46.11111;     !- Maximum Value of y

```

```

Curve:Biquadratic,
Defrost_EIR_FT, !- Name
1,              !- Coefficient1 Constant
0,              !- Coefficient2 x
0,              !- Coefficient3 x**2
0,              !- Coefficient4 y
0,              !- Coefficient5 y**2
0,              !- Coefficient6 x*y
0,              !- Minimum Value of x
100,            !- Maximum Value of x
0,              !- Minimum Value of y
100;            !- Maximum Value of y

```

! This is a dummy waste heat function of temp curve required for the
! multispeed coils. The curve is set as a constant curve that would yield
! a factor of 1 at all temperatures for simplicity.
! However, this curve is not used if the fuel type is electricity (which is
! what we limit ourselves to in terms of multispeed DX equipment) per
! the E+ version 8.0 I/O reference Page 1301.

```

Curve:Biquadratic,
dummy-waste-heat-curve, !- Name
1,              !- Coefficient1 Constant
0,              !- Coefficient2 x
0,              !- Coefficient3 x**2
0,              !- Coefficient4 y
0,              !- Coefficient5 y**2
0,              !- Coefficient6 x*y
0,              !- Minimum Value of x
50,             !- Maximum Value of x

```

0, !- Minimum Value of y
50, !- Maximum Value of y
1, !- Minimum Curve Output
1, !- Maximum Curve Output
Dimensionless, !- Input Unit Type for X
Dimensionless, !- Input Unit Type for Y
Dimensionless; !- Output Unit Type

! These coefficients were taken directly from the report below. For further details users may need
! to refer this report.

! D. Cutler, J. Winkler, N. Kruis, and C. Christensen
! National Renewable Energy Laboratory
! M. Brandemuehl
! University of Colorado

! Improved Modeling of Residential Air Conditioners and Heat Pumps for Energy Calculations

! Technical Report
! NREL/TP-5500-56354
! January 2013

! Air Conditioner Performance Curves were taken from: Tables 9, 10, 16 and 17:

! ***** Single Speed Cooling *****

Curve:Biquadratic,

ACCoolingCAPFTemp, !- Name
1.5509, !- Coefficient1 Constant
-0.07505, !- Coefficient2 x
0.0031, !- Coefficient3 x**2
0.0024, !- Coefficient4 y
-0.00005, !- Coefficient5 y**2
-0.00043, !- Coefficient6 x*y
0, !- Minimum Value of x
50, !- Maximum Value of x
0, !- Minimum Value of y
50, !- Maximum Value of y
0, !- Minimum Curve Output
5, !- Maximum Curve Output
Temperature, !- Input Unit Type for X
Temperature, !- Input Unit Type for Y
Dimensionless; !- Output Unit Type

Curve:Biquadratic,

ACCoolingEIRFTemp, !- Name
-0.30428, !- Coefficient1 Constant
0.11805, !- Coefficient2 x
-0.00342, !- Coefficient3 x**2
-0.00626, !- Coefficient4 y
0.0007, !- Coefficient5 y**2
-0.00047, !- Coefficient6 x*y
0, !- Minimum Value of x
50, !- Maximum Value of x
0, !- Minimum Value of y
50, !- Maximum Value of y
0, !- Minimum Curve Output
5, !- Maximum Curve Output
Temperature, !- Input Unit Type for X
Temperature, !- Input Unit Type for Y
Dimensionless; !- Output Unit Type

! Heat Pump Performance Curves were taken from: Tables 14, 15, 18 and 19:

! ***** Single Speed Cooling *****

Curve:Biquadratic,

HPCoolingCAPFTemp, !- Name
1.55736, !- Coefficient1 Constant
-0.074448, !- Coefficient2 x
0.003099, !- Coefficient3 x**2
0.00146, !- Coefficient4 y
-0.000041, !- Coefficient5 y**2
-0.000427, !- Coefficient6 x*y

0, !- Minimum Value of x
50, !- Maximum Value of x
0, !- Minimum Value of y
50, !- Maximum Value of y
0, !- Minimum Curve Output
5, !- Maximum Curve Output
Temperature, !- Input Unit Type for X
Temperature, !- Input Unit Type for Y
Dimensionless; !- Output Unit Type

Curve:Biquadratic,

HPCoolingEIRFTemp, !- Name
-0.350448, !- Coefficient1 Constant
0.11681, !- Coefficient2 x
-0.0034, !- Coefficient3 x**2
-0.001226, !- Coefficient4 y
0.000601, !- Coefficient5 y**2
-0.000467, !- Coefficient6 x*y
0, !- Minimum Value of x
50, !- Maximum Value of x
0, !- Minimum Value of y
50, !- Maximum Value of y
0, !- Minimum Curve Output
5, !- Maximum Curve Output
Temperature, !- Input Unit Type for X
Temperature, !- Input Unit Type for Y
Dimensionless; !- Output Unit Type

! ***** Single Speed Heating *****

Curve:Biquadratic,

HPHeatingCAPFTemp, !- Name
0.876825, !- Coefficient1 Constant
-0.002955, !- Coefficient2 x
-0.000058, !- Coefficient3 x**2
0.025335, !- Coefficient4 y
0.000196, !- Coefficient5 y**2
-0.000043, !- Coefficient6 x*y
0, !- Minimum Value of x
50, !- Maximum Value of x
0, !- Minimum Value of y
50, !- Maximum Value of y
0, !- Minimum Curve Output
5, !- Maximum Curve Output
Temperature, !- Input Unit Type for X
Temperature, !- Input Unit Type for Y
Dimensionless; !- Output Unit Type

Curve:Biquadratic,

HPHeatingEIRFTemp, !- Name
0.704658, !- Coefficient1 Constant
0.008767, !- Coefficient2 x
0.000625, !- Coefficient3 x**2
-0.009037, !- Coefficient4 y
0.000738, !- Coefficient5 y**2
-0.001025, !- Coefficient6 x*y
0, !- Minimum Value of x
50, !- Maximum Value of x
0, !- Minimum Value of y
50, !- Maximum Value of y
0, !- Minimum Curve Output
5, !- Maximum Curve Output
Temperature, !- Input Unit Type for X
Temperature, !- Input Unit Type for Y
Dimensionless; !- Output Unit Type

! ***** Low Speed Heating *****

Curve:Biquadratic,

HPLowStageHeatingCAPFTemp, !- Name
0.84613, !- Coefficient1 Constant
-0.002279, !- Coefficient2 x

-0.000047, !- Coefficient3 x**2
0.026703, !- Coefficient4 y
0.000201, !- Coefficient5 y**2
-0.000079, !- Coefficient6 x*y
0, !- Minimum Value of x
50, !- Maximum Value of x
0, !- Minimum Value of y
50, !- Maximum Value of y
0, !- Minimum Curve Output
5, !- Maximum Curve Output
Temperature, !- Input Unit Type for X
Temperature, !- Input Unit Type for Y
Dimensionless; !- Output Unit Type

Curve:Biquadratic,

HPLowStageHeatingEIRFTemp, !- Name
0.551837, !- Coefficient1 Constant
0.02038, !- Coefficient2 x
0.000546, !- Coefficient3 x**2
-0.009638, !- Coefficient4 y
0.000785, !- Coefficient5 y**2
-0.00125, !- Coefficient6 x*y
0, !- Minimum Value of x
50, !- Maximum Value of x
0, !- Minimum Value of y
50, !- Maximum Value of y
0, !- Minimum Curve Output
5, !- Maximum Curve Output
Temperature, !- Input Unit Type for X
Temperature, !- Input Unit Type for Y
Dimensionless; !- Output Unit Type

! ***** High Speed Heating *****

Curve:Biquadratic,

HPHighStageHeatingCAPFTemp, !- Name
0.818223, !- Coefficient1 Constant
0.001981, !- Coefficient2 x
-0.000203, !- Coefficient3 x**2
0.028703, !- Coefficient4 y
0.000207, !- Coefficient5 y**2
-0.000071, !- Coefficient6 x*y
0, !- Minimum Value of x
50, !- Maximum Value of x
0, !- Minimum Value of y
50, !- Maximum Value of y
0, !- Minimum Curve Output
5, !- Maximum Curve Output
Temperature, !- Input Unit Type for X
Temperature, !- Input Unit Type for Y
Dimensionless; !- Output Unit Type

Curve:Biquadratic,

HPHighStageHeatingEIRFTemp, !- Name
0.81584, !- Coefficient1 Constant
-0.00615, !- Coefficient2 x
0.001021, !- Coefficient3 x**2
-0.001301, !- Coefficient4 y
0.001083, !- Coefficient5 y**2
-0.001487, !- Coefficient6 x*y
0, !- Minimum Value of x
50, !- Maximum Value of x
0, !- Minimum Value of y
50, !- Maximum Value of y
0, !- Minimum Curve Output
5, !- Maximum Curve Output
Temperature, !- Input Unit Type for X
Temperature, !- Input Unit Type for Y
Dimensionless; !- Output Unit Type

! ***** Low Speed Cooling *****

Curve:Biquadratic,
 HPLowStageCoolingCAPFTemp, !- Name
 1.658788, !- Coefficient1 Constant
 -0.083453, !- Coefficient2 x
 0.003424, !- Coefficient3 x**2
 0.002433, !- Coefficient4 y
 -0.000045, !- Coefficient5 y**2
 -0.000534, !- Coefficient6 x*y
 0, !- Minimum Value of x
 50, !- Maximum Value of x
 0, !- Minimum Value of y
 50, !- Maximum Value of y
 0, !- Minimum Curve Output
 5, !- Maximum Curve Output
 Temperature, !- Input Unit Type for X
 Temperature, !- Input Unit Type for Y
 Dimensionless; !- Output Unit Type

Curve:Biquadratic,
 HPLowStageCoolingEIRFTemp, !- Name
 -0.582916, !- Coefficient1 Constant
 0.158101, !- Coefficient2 x
 -0.004398, !- Coefficient3 x**2
 -0.020335, !- Coefficient4 y
 0.00108, !- Coefficient5 y**2
 -0.00064, !- Coefficient6 x*y
 0, !- Minimum Value of x
 50, !- Maximum Value of x
 0, !- Minimum Value of y
 50, !- Maximum Value of y
 0, !- Minimum Curve Output
 5, !- Maximum Curve Output
 Temperature, !- Input Unit Type for X
 Temperature, !- Input Unit Type for Y
 Dimensionless; !- Output Unit Type

! ***** High Speed Cooling *****

Curve:Biquadratic,
 HPHighStageCoolingCAPFTemp, !- Name
 1.472738, !- Coefficient1 Constant
 -0.067222, !- Coefficient2 x
 0.00292, !- Coefficient3 x**2
 0.000052, !- Coefficient4 y
 -0.00003, !- Coefficient5 y**2
 -0.000359, !- Coefficient6 x*y
 0, !- Minimum Value of x
 50, !- Maximum Value of x
 0, !- Minimum Value of y
 50, !- Maximum Value of y
 0, !- Minimum Curve Output
 5, !- Maximum Curve Output
 Temperature, !- Input Unit Type for X
 Temperature, !- Input Unit Type for Y
 Dimensionless; !- Output Unit Type

Curve:Biquadratic,
 HPHighStageCoolingEIRFTemp, !- Name
 -0.488196, !- Coefficient1 Constant
 0.099162, !- Coefficient2 x
 -0.00237, !- Coefficient3 x**2
 0.019503, !- Coefficient4 y
 0.00043, !- Coefficient5 y**2
 -0.001097, !- Coefficient6 x*y
 0, !- Minimum Value of x
 50, !- Maximum Value of x
 0, !- Minimum Value of y
 50, !- Maximum Value of y
 0, !- Minimum Curve Output
 5, !- Maximum Curve Output
 Temperature, !- Input Unit Type for X

Temperature, !- Input Unit Type for Y
Dimensionless; !- Output Unit Type

! ***** Low Speed Cooling *****

Curve:Biquadratic,
ACLowStageCoolingCAPFTemp, !- Name
1.66458, !- Coefficient1 Constant
-0.08039, !- Coefficient2 x
0.0033, !- Coefficient3 x**2
0.00124, !- Coefficient4 y
-0.00003, !- Coefficient5 y**2
-0.00052, !- Coefficient6 x*y
0, !- Minimum Value of x
50, !- Maximum Value of x
0, !- Minimum Value of y
50, !- Maximum Value of y
0, !- Minimum Curve Output
5, !- Maximum Curve Output
Temperature, !- Input Unit Type for X
Temperature, !- Input Unit Type for Y
Dimensionless; !- Output Unit Type

Curve:Biquadratic,
ACLowStageCoolingEIRFTemp, !- Name
-0.42738, !- Coefficient1 Constant
0.14191, !- Coefficient2 x
-0.00412, !- Coefficient3 x**2
-0.01406, !- Coefficient4 y
0.00083, !- Coefficient5 y**2
-0.00043, !- Coefficient6 x*y
0, !- Minimum Value of x
50, !- Maximum Value of x
0, !- Minimum Value of y
50, !- Maximum Value of y
0, !- Minimum Curve Output
5, !- Maximum Curve Output
Temperature, !- Input Unit Type for X
Temperature, !- Input Unit Type for Y
Dimensionless; !- Output Unit Type

! ***** High Speed Cooling *****

Curve:Biquadratic,
ACHighStageCoolingCAPFTemp, !- Name
1.36788, !- Coefficient1 Constant
-0.06257, !- Coefficient2 x
0.0028, !- Coefficient3 x**2
0.00504, !- Coefficient4 y
-0.00007, !- Coefficient5 y**2
-0.00045, !- Coefficient6 x*y
0, !- Minimum Value of x
50, !- Maximum Value of x
0, !- Minimum Value of y
50, !- Maximum Value of y
0, !- Minimum Curve Output
5, !- Maximum Curve Output
Temperature, !- Input Unit Type for X
Temperature, !- Input Unit Type for Y
Dimensionless; !- Output Unit Type

Curve:Biquadratic,
ACHighStageCoolingEIRFTemp, !- Name
0.04232, !- Coefficient1 Constant
0.07892, !- Coefficient2 x
-0.00238, !- Coefficient3 x**2
-0.00304, !- Coefficient4 y
0.00053, !- Coefficient5 y**2
-0.00032, !- Coefficient6 x*y
0, !- Minimum Value of x
50, !- Maximum Value of x
0, !- Minimum Value of y

50, !- Maximum Value of y
0, !- Minimum Curve Output
5, !- Maximum Curve Output
Temperature, !- Input Unit Type for X
Temperature, !- Input Unit Type for Y
Dimensionless; !- Output Unit Type

! NREL benchmark Cool-CAP-FT

Curve:Biquadratic,
Cool-Cap-ft, !- Name
1.26489391, !- Coefficient1 Constant
-0.035054982, !- Coefficient2 x
0.00211086, !- Coefficient3 x**2
-0.001526886, !- Coefficient4 y
-0.0000070308, !- Coefficient5 y**2
-0.0004691844, !- Coefficient6 x*y
-100, !- Minimum Value of x
100, !- Maximum Value of x
-100, !- Minimum Value of y
100; !- Maximum Value of y

! NREL benchmark Cool-EIR-FT

Curve:Biquadratic,
Cool-EIR-ft, !- Name
0.38402403, !- Coefficient1 Constant
0.029696724, !- Coefficient2 x
-0.0011329308, !- Coefficient3 x**2
0.006490674, !- Coefficient4 y
0.0002626992, !- Coefficient5 y**2
-0.0001207224, !- Coefficient6 x*y
-100, !- Minimum Value of x
100, !- Maximum Value of x
-100, !- Minimum Value of y
100; !- Maximum Value of y

!- ===== ALL OBJECTS IN CLASS: OUTPUT:VARIABLEDICTIONARY =====

Output:VariableDictionary,
Regular; !- Key Field

!- ===== ALL OBJECTS IN CLASS: OUTPUT:CONSTRUCTIONS =====

Output:Constructions,
Constructions, !- Details Type 1
Materials; !- Details Type 2

!- ===== ALL OBJECTS IN CLASS: OUTPUT:TABLE:SUMMARYREPORTS =====

Output:Table:SummaryReports,
InputVerificationandResultsSummary, !- Report 1 Name
EquipmentSummary, !- Report 2 Name
ClimaticDataSummary, !- Report 3 Name
EnvelopeSummary, !- Report 4 Name
AllSummary; !- Report 5 Name

!- ===== ALL OBJECTS IN CLASS: OUTPUT:TABLE:MONTHLY =====

Output:Table:Monthly,
FanSplit, !- Name
3, !- Digits After Decimal
Air System Cooling Coil Total Cooling Energy, !- Variable or Meter 1 Name
HoursNonZero, !- Aggregation Type for Variable or Meter 1
Air System Fan Electric Energy, !- Variable or Meter 2 Name
SumOrAverageDuringHoursShown, !- Aggregation Type for Variable or Meter 2
Air System Heating Coil Total Heating Energy, !- Variable or Meter 3 Name
HoursNonZero, !- Aggregation Type for Variable or Meter 3

Air System Fan Electric Energy, !- Variable or Meter 4 Name
SumOrAverageDuringHoursShown, !- Aggregation Type for Variable or Meter 4
Air System Fan Electric Energy, !- Variable or Meter 5 Name
SumOrAverage; !- Aggregation Type for Variable or Meter 5

Output:Table:Monthly,
CoilLoads, !- Name
2, !- Digits After Decimal
Heating Coil Heating Rate, !- Variable or Meter 1 Name
SumOrAverage, !- Aggregation Type for Variable or Meter 1
Heating Coil Heating Rate, !- Variable or Meter 2 Name
SumOrAverage, !- Aggregation Type for Variable or Meter 2
Cooling Coil Total Cooling Rate, !- Variable or Meter 3 Name
SumOrAverage; !- Aggregation Type for Variable or Meter 3

Output:Table:Monthly,
Water Heater: Loads, !- Name
2, !- Digits After Decimal
Water Heater Total Demand Rate, !- Variable or Meter 1 Name
SumOrAverage, !- Aggregation Type for Variable or Meter 1
Water Heater Total Demand Energy, !- Variable or Meter 2 Name
SumOrAverage, !- Aggregation Type for Variable or Meter 2
Water Heater Heating Rate, !- Variable or Meter 3 Name
SumOrAverage, !- Aggregation Type for Variable or Meter 3
Water Heater Heating Energy, !- Variable or Meter 4 Name
SumOrAverage; !- Aggregation Type for Variable or Meter 4

Output:Table:Monthly,
Heating and Cooling Loads, !- Name
2, !- Digits After Decimal
Zone/Sys Sensible Cooling Energy, !- Variable or Meter 1 Name
SumOrAverage, !- Aggregation Type for Variable or Meter 1
Zone/Sys Sensible Heating Energy, !- Variable or Meter 2 Name
SumOrAverage; !- Aggregation Type for Variable or Meter 2

Output:Table:Monthly,
Lighting Loads, !- Name
2, !- Digits After Decimal
Zone Lights Electric Consumption, !- Variable or Meter 1 Name
SumOrAverage, !- Aggregation Type for Variable or Meter 1
Exterior Lights Electric Consumption, !- Variable or Meter 2 Name
SumOrAverage; !- Aggregation Type for Variable or Meter 2

Output:Table:Monthly,
InternalGains, !- Name
2, !- Digits After Decimal
Zone People Total Heat Gain, !- Variable or Meter 1 Name
SumOrAverage, !- Aggregation Type for Variable or Meter 1
Zone Lights Total Heat Gain, !- Variable or Meter 2 Name
SumOrAverage, !- Aggregation Type for Variable or Meter 2
Zone Electric Equipment Total Heat Gain, !- Variable or Meter 3 Name
SumOrAverage, !- Aggregation Type for Variable or Meter 3
Zone Gas Equipment Total Heat Gain, !- Variable or Meter 4 Name
SumOrAverage, !- Aggregation Type for Variable or Meter 4
Zone Other Equipment Total Heat Gain, !- Variable or Meter 5 Name
SumOrAverage, !- Aggregation Type for Variable or Meter 5
Zone Total Internal Total Heat Gain, !- Variable or Meter 6 Name
SumOrAverage; !- Aggregation Type for Variable or Meter 6

!- ===== ALL OBJECTS IN CLASS: OUTPUTCONTROL:TABLE:STYLE =====

OutputControl:Table:Style,
CommaAndHTML, !- Column Separator
InchPound; !- Unit Conversion

!- ===== ALL OBJECTS IN CLASS: OUTPUT:VARIABLE =====

Output:Variable,

```

living_unit1,      !- Key Value
Zone Thermostat Cooling Setpoint Temperature , !- Variable Name
Timestep;         !- Reporting Frequency

Output:Variable,
*,                !- Key Value
Site Outdoor Air Relative Humidity , !- Variable Name
Timestep;         !- Reporting Frequency

Output:Variable,
*,                !- Key Value
Zone Air Relative Humidity , !- Variable Name
Timestep;         !- Reporting Frequency

Output:Variable,
Whole Building,   !- Key Value
Facility Total Purchased Electric Energy , !- Variable Name
Timestep;         !- Reporting Frequency

Output:Variable,
Environment,      !- Key Value
Site Direct Solar Radiation Rate per Area , !- Variable Name
Timestep;         !- Reporting Frequency

Output:Variable,
Environment,      !- Key Value
Site Diffuse Solar Radiation Rate per Area , !- Variable Name
Timestep;         !- Reporting Frequency

Output:Variable,
Whole Building,   !- Key Value
Facility Net Purchased Electric Energy , !- Variable Name
Timestep;         !- Reporting Frequency

Output:Variable,
Whole Building,   !- Key Value
Facility Total Surplus Electric Energy , !- Variable Name
Timestep;         !- Reporting Frequency

Output:Variable,
DaysOfWeek,      !- Key Value
Schedule Value , !- Variable Name
Timestep;         !- Reporting Frequency

Output:Variable,
*,                !- Key Value
Site Outdoor Air Drybulb Temperature , !- Variable Name
Timestep;         !- Reporting Frequency

Output:Variable,
living_unit1,    !- Key Value
Zone Air Temperature , !- Variable Name
Timestep;         !- Reporting Frequency

Output:Variable,
living_unit1,    !- Key Value
Zone Thermostat Heating Setpoint Temperature , !- Variable Name
Timestep;         !- Reporting Frequency

!- ===== ALL OBJECTS IN CLASS: OUTPUT:METER =====

Output:Meter,
Electricity:Facility, !- Key Name
Timestep;             !- Reporting Frequency

Output:Meter,
Cooling:Electricity, !- Key Name
Timestep;             !- Reporting Frequency

```

Output:Meter,
Heating:Electricity, !- Key Name
Timestep; !- Reporting Frequency

Output:Meter,
Photovoltaic:ElectricityProduced , !- Key Name
Timestep; !- Reporting Frequency

Output:Meter,
Cogeneration:ElectricityPurchased , !- Key Name
Timestep; !- Reporting Frequency

Output:Meter,
Cogeneration:ElectricitySurplusSold , !- Key Name
Timestep; !- Reporting Frequency

Appendix H. Pricing

Table H1. Summary of the types of pricing models from the major utility companies in the five cities reviewed.

Location	Company	Type of Plan	Details
Fairbanks, AK	Golden Valley Electric Association	Fixed Rate	All energy is charged at one fixed rate.
Miami, FL	Florida Power & Light	Tiered Fixed Rate	Price decreases when monthly consumption totals over 1,000 kWh.
		Time of Use	Peak hours are: Nov. 1 - March 31: Monday through Friday between 6 a.m. to 10 a.m., and 6 p.m. to 10 p.m. April 1 – Oct. 31: Monday through Friday during the hours from noon to 9 p.m.
New York, NY	Con Edison (delivery Rates)	Tiered Fixed Rate	Price increases when consumption totals over 250 kWh.
		Time of Use	Price increased during peak hours: 8am - 12am
Phoenix, AZ	Arizona Public Service Co.	Fixed Rate	All energy is charged at one fixed rate.
		Time of Use	"Saver Choice"; off-peak: 8pm-3pm; super off-peak winter: 10am-3pm weekdays; on-peak: 3pm-8pm weekdays; compatible with renewable energy; Grid access charge is \$.93 per kW-dc of generation
			"Saver Choise Plus"; off-peak: 8pm-3pm; on-peak: 3pm-8pm weekdays; compatible with renewable energy; demand charge: A charge for the single highest hour (averaged) of energy used weekdays between 3 pm - 8 pm during a billing cycle.
San Francisco, CA	Pacific Gas & Electric	Tiered Fixed Rate	3 tiers of increase: baseline, 101-400% baseline, and over 400% baseline. Baseline Allowance is determined by where you live (your Baseline Territory), your primary heating source and the season (summer or winter).
		Tiered Time of Use	Peak: 4-9pm everyday, with same tiers according to baseline as above
		Time of Use	Peak: 5-8pm weekdays; no baseline allowance; winter is Oct-May; summer is June-Sept; if you use higher than baseline,

			this may work better for you than E-TOU-C
--	--	--	---

Appendix I. Excel Solver Data

Time [hour]	Outdoor Temperature [°C]	Solar Radiation [J]	Price of Energy [\$/kWh]	Excel Solver			CVXOPT		
				Indoor Temperature [°C]	Energy Used [kWh]	Cost [\$]	Indoor Temperature [°C]	Energy Used [kWh]	Cost [\$]
0	9.4	0	0.2	20	0.14	0.03	20.00	0.14	0.03
0.083333	9.358334	0	0.2	20	0.14	0.03	20.00	0.15	0.03
0.166667	9.316667	0	0.2	20	0.15	0.03	20.00	0.15	0.03
0.25	9.275	0	0.2	20	0.15	0.03	20.00	0.15	0.03
0.333333	9.233334	0	0.2	20	0.15	0.03	20.00	0.15	0.03
0.416667	9.191667	0	0.2	20	0.15	0.03	20.00	0.15	0.03
0.5	9.15	0	0.2	20	0.15	0.03	20.00	0.15	0.03
0.583333	9.108334	0	0.2	20	0.15	0.03	20.00	0.15	0.03
0.666667	9.066667	0	0.2	20	0.15	0.03	20.00	0.15	0.03
0.75	9.025	0	0.2	20	0.15	0.03	20.00	0.15	0.03
0.833333	8.983334	0	0.2	20	0.15	0.03	20.00	0.15	0.03
0.916667	8.941667	0	0.2	20	0.15	0.03	20.00	0.15	0.03
1	8.9	0	0.2	20	0.15	0.03	20.00	0.15	0.03
1.083333	8.941667	0	0.2	20	0.15	0.03	20.00	0.15	0.03
1.166667	8.983334	0	0.2	20	0.15	0.03	20.00	0.15	0.03
1.25	9.025	0	0.2	20	0.15	0.03	20.00	0.15	0.03
1.333333	9.066667	0	0.2	20	0.15	0.03	20.00	0.15	0.03
1.416667	9.108334	0	0.2	20	0.15	0.03	20.00	0.15	0.03
1.5	9.15	0	0.2	20	0.15	0.03	20.00	0.15	0.03
1.583333	9.191667	0	0.2	20	0.15	0.03	20.00	0.15	0.03
1.666667	9.233334	0	0.2	20	0.15	0.03	20.00	0.15	0.03
1.75	9.275	0	0.2	20	0.15	0.03	20.00	0.15	0.03
1.833333	9.316667	0	0.2	20	0.15	0.03	20.00	0.15	0.03
1.916667	9.358334	0	0.2	20	0.14	0.03	20.00	0.15	0.03
2	9.4	0	0.2	20	0.14	0.03	20.00	0.14	0.03
2.083333	9.4	0	0.2	20	0.14	0.03	20.00	0.14	0.03
2.166667	9.4	0	0.2	20	0.14	0.03	20.00	0.14	0.03
2.25	9.4	0	0.2	20	0.14	0.03	20.00	0.14	0.03
2.333333	9.4	0	0.2	20	0.14	0.03	20.00	0.14	0.03
2.416667	9.4	0	0.2	20	0.14	0.03	20.00	0.14	0.03
2.5	9.4	0	0.2	20	0.14	0.03	20.00	0.14	0.03
2.583333	9.4	0	0.2	20	0.14	0.03	20.00	0.14	0.03
2.666667	9.4	0	0.2	20.0000069	0.14	0.03	20.00	0.14	0.03
2.75	9.4	0	0.2	20	0.14	0.03	20.00	0.14	0.03

2.833333	9.4	0	0.2	20.0000176	1.88	0.38	20.00	1.88	0.38
2.916667	9.4	0	2	20.7791561	0.00	0.00	20.78	0.00	0.00
3	9.4	0	2	20.7095156	0.00	0.00	20.71	0.00	0.00
3.083333	9.266666	0	2	20.6403014	0.00	0.00	20.64	0.00	0.00
3.166667	9.133333	0	2	20.5706948	0.00	0.00	20.57	0.00	0.00
3.25	9	0	2	20.5006981	0.00	0.00	20.50	0.00	0.00
3.333333	8.866667	0	2	20.4303138	0.00	0.00	20.43	0.00	0.00
3.416667	8.733334	0	2	20.3595443	0.00	0.00	20.36	0.00	0.00
3.5	8.6	0	2	20.2883919	0.00	0.00	20.29	0.00	0.00
3.583333	8.466666	0	2	20.2168589	0.00	0.00	20.22	0.00	0.00
3.666667	8.333333	0	2	20.1449478	0.00	0.00	20.15	0.00	0.00
3.75	8.2	0	2	20.0726607	0.00	0.00	20.07	0.00	0.00
3.833333	8.066667	0	0.2	20	0.16	0.03	20.00	0.16	0.03
3.916667	7.933333	0	0.2	20	0.16	0.03	20.00	0.16	0.03
4	7.8	0	0.2	20	0.17	0.03	20.00	0.17	0.03
Totals:					7.35	1.47		7.35	1.47