Development of a Tiny House Design Tool

To Increase Safety, Efficiency, and Cost-Effectiveness

Michael Joseph Stratton II

A Thesis Submitted to the Graduate Faculty of

GRAND VALLEY STATE UNIVERSITY

In

Partial Fulfillment of the Requirements

For the Degree of

Master of Science in Mechanical Engineering

School of Engineering

December 2020

Acknowledgements

I would like to first acknowledge and thank the thesis committee for their support, insight, and guidance. I would like to especially thank the committee chair, Dr. Lindsay Corneal, for her constant enthusiasm through this longer than usual master's thesis. Being a part-time graduate student while working full time in industry lengthened the overall study, but Dr. Lindsay Corneal was always excited and eager to help.

I would also like to acknowledge and thank Jim Stratton for his aid in testing the Scilab code, Anna Hagstrom for reviewing the Scilab instructions, as well as Tiny House Basics, LLC for providing trailer dimensions, weights, and costs.

Abstract

A growing number of people are choosing to live more sustainably, and this is commonly done through living in an extremely small house, also called a "tiny house". There is no set definition for a tiny house, but it typically refers to a structure less than 500 ft^2, sometimes built by the homeowner, and frequently built on a travel trailer.

The popularity of tiny houses has increased over the years due to different television shows, internet videos, and photo sharing websites, but the technical details are rarely covered, which is why this study examines the structural, stability, weight, and thermal requirements of two gooseneck style tiny houses. A configurable SolidWorks model was created, and the location and quantity of structural supports were determined by performing structural finite element analyses. The center of mass, tip angles, overall weight, and tongue weight were computed to ensure stability while stationary or during transit. The heating and cooling needs for the tiny houses were calculated by performing thermal finite element analyses for twelve host cities of varying geographic locations and regional climates across the contiguous United States.

The result of this study is a Scilab design tool that takes as input items such as location, desired size, utility related information, etc. It calculates the materials needed for the shell of the structure in addition to a projected cost that the user can reference. This design tool, along with the separately provided construction drawings, allows the user to build a safer, more efficient, and more cost-effective tiny house.

Table of Contents

List of Figures

List of Tables

Abbreviations, Acronyms, and Terminology

AC	Acronym for alternating current
approx.	Abbreviation for approximately
assembly	Term referring to a model in SolidWorks that is built from one or more parts
avg.	Abbreviation for average
BOM(s)	Acronym for bill(s) of material
BTU/hour	Abbreviation for British thermal unit per hour (unit of power)
CAD	Acronym for computer aided design
COM	Acronym for center of mass
DC	Acronym for direct current
driver side	Term referring to the side of the tiny house that aligns with the driver's door of the towing vehicle, when attached
dry lumber	Term referring to lumber with a moisture content below 19%
Eq.	Abbreviation for equation
est.	Abbreviation for estimated
FEA	Acronym for finite element analysis (structural or thermal)
ft	Abbreviation for foot (unit of length)
ft^2	Abbreviation for square feet (unit of area)
ft^3	Abbreviation for cubic feet (unit of volume)
goose	Term in this study that refers to the portion of the tiny house that extends over the towing vehicle when attached
gooseneck trailer	Term referring to a trailer that partially extends over the towing vehicle and requires a gooseneck hitch
green lumber	Term referring to lumber with a moisture content above 19%

Heat flux	Term meaning power per unit area
HVAC	Acronym for heating, ventilation, and air conditioning
in	Abbreviation for inch (unit of length)
in^2	Abbreviation for square inches (unit of area)
in^3	Abbreviation for cubic inches (unit of volume)
insul.	Abbreviation for insulation
int.	Abbreviation for interior
K	Abbreviation for Kelvin (unit of temperature)
k	Abbreviation for kilo, meaning 1,000 (e.g. kW is kilowatt)
kg	Abbreviation for kilogram (unit of mass)
kWh	Abbreviation for kilowatt-hour (unit of energy)
lb	Abbreviation for pound (unit of weight/force)
live load	Term in this study referring to heat generated by an electrical device or living entity
lumens	Term referring to the visible light produced by a source over a certain period of time (unit of luminous flux)
m	Abbreviation for meters (unit of length)
m^2	Abbreviation for square meters (unit of area)
m^3	Abbreviation for cubic meters (unit of volume)
mini-split	Term referring to a ductless HVAC system
mph	Abbreviation for miles per hour (unit of velocity)
N	Abbreviation for Newton (unit of weight/force)
no.	Abbreviation for number
part	Term referring to a model in SolidWorks that is a single component

passenger side	Term referring to the side of the tiny house that aligns with the passenger's door of the towing vehicle, when attached
pc	Abbreviation for piece
psi	Abbreviation for pounds per square inch (unit of pressure)
R-Value	Term referring to the level of heat flux "resistance" an insulation has (higher values result in lower heat flux)
s	Abbreviation for second (unit of time)
safety factor	Term in this study referring to yield strength divided by expected stresses
Scilab	Term referring to Scilab 5.5.2 (an open source programming software)
SolidWorks	Term referring to a specific CAD modeling software
superelevation	Term referring to the grade at which a road is angled
temp.	Abbreviation for temperature
tiny house	Term referring to a very small living structure (Term specifically referring to a gooseneck style tiny house beginning in section 1.3)
tongue weight	Term referring to the downward force applied from a trailer to a hitch
traditional house	Term referring to the typical house, which is built on a foundation
traditional trailer	Term referring to the typical travel trailer with standard hitch
U.S.	Abbreviation for United States
W	Abbreviation of Watt (unit of power)
W/m^2	Abbreviation for Watts per square meter (unit of heat flux)
1% cooling dry bulb temperature	Term referring to the temperature at which only 1% of the year has had historically hotter temperatures
1% extreme Wind speed	Term referring to the wind speed at which only 1% of the year has had historically faster wind speeds
1-month precipitation	Term referring to the total precipitation recorded in a 1-month period for a specific location

2 x 4	Term referring to lumber that is 1.5 in thick and 3.5 in wide (length varies)
2 x 6	Term referring to lumber that is 1.5 in thick and 5.5 in wide (length varies)
2 x 8	Term referring to lumber that is 1.5 in thick and 7.5 in wide (length varies)
2 x 10	Term referring to lumber that is 1.5 in thick and 9.5 in wide (length varies)
2 x 12	Term referring to lumber that is 1.5 in thick and 11.5 in wide (length varies)
2D	Abbreviation for two-dimensional
3D	Abbreviation for three-dimensional
99% heating dry bulb temperature	Term referring to the temperature at which 99% of the year has had historically hotter temperatures
°	Abbreviation for degrees (measure of an angle)
°F	Abbreviation for degrees Fahrenheit (unit of temperature)

1. Introduction

A growing number of people are choosing to live more sustainably, and this is commonly done through living in an extremely small house, also called a “tiny house” (Figure 1). There is no set definition for a tiny house, but it typically refers to a structure less than 500 square feet (ft^2), sometimes built by the homeowner, and frequently built on a travel trailer.

The tiny house topic is being investigated because their construction is almost identical to traditional large houses, except with less livable area and foundation differences due to the introduction of a trailer. As current design practices are quite similar, there is a high probability that altering the design would result in improvements. Conducting research and providing drawings and a design tool that properly accounts for various loads and conditions would help grow this sustainable movement by aiding future tiny homeowners with their planning and construction.

1.1. Different Types of Tiny Houses

There are many different styles of tiny houses, and each type has its own unique features. The tiny house on wheels (most popular style) comes in two varieties, and the first is shown in Figure 1. It is built on a travel (traditional) trailer and can be towed with a pickup truck. The second wheeled variety is shown in Figure 2. Instead of being built on a traditional trailer, a gooseneck trailer is used. This style includes a different type of hitch, which allows the structure to be built out further for additional living space. These styles are not as popular as a larger pickup truck is typically required (tiny houses built on gooseneck trailers are typically larger and heavier than tiny houses built on traditional trailers), and the pickup truck must have a gooseneck hitch installed in the bed. For both styles of trailer-built tiny houses, traditional construction skills are required.

Figure 1: Example of a Tiny House on Wheels (Traditional Trailer) [1]

Figure 2: Example of a Tiny House on Wheels (Gooseneck Trailer) [2]

Some tiny houses can be built from a single shipping container, as shown in Figure 3. These models are very sturdy as shipping containers are rated for tens of thousands of pounds (lb), while being transported across the ocean. The exterior shell is nearly complete, but specialty metal working skills are typically needed to complete this type of tiny house. Others can be built on a foundation, following typical construction practices (Figure 4). These are the closest to traditional houses in terms of design and skills needed to build but are smaller in overall footprint.

Figure 3: Example of a Tiny House Built from a Shipping Container [3]

Figure 4: Example of a Tiny House Built on a Foundation [4]

1.2. Reasons for Implementing a Design Tool

The result of this study is a Scilab design tool that asks a user multiple questions. It takes as input the answers and outputs information about the tiny house. This information, along with the separately provided construction drawings, can aid the user in building a tiny house. The reasons for implementing this kind of design tool are to increase safety, efficiency, and cost-effectiveness. By carefully examining the structural, stability, weight, and thermal requirements of a tiny house, the design can be made safer by ensuring the structure will not be prone to tipping when turning on an average road and will have the proper tongue weight percentage on the towing vehicle's hitch. It would also increase safety by taking into account the loads the tiny house would need to withstand during its operational lifetime, so the occupants would be protected while inside the structure.

A design tool would result in a more efficient tiny house because structural supports would only be added where needed, and the overall heat flux would be taken into account based on the climate where tiny house would be located. This way, the occupants can be comfortable without oversizing the heating, ventilation, and air conditioning (HVAC) system.

The cost-effectiveness of the design would increase because the tiny homeowner would know what items and what quantities would need to be purchased. The associated construction drawings would also help reduce waste as lengths of individual components would be known ahead of time. The tiny homeowner can then expect less rework and expect to purchase less materials, again reducing the overall cost of the project.

Finally, the popularity of tiny houses has increased over the years due to different television shows, internet videos, and photo sharing websites, but the technical details are rarely covered. A design tool would expose the general public to items that are not usually shown, which would give them a better understanding of what it takes to build a tiny house.

1.3. Areas of Interest

There were several areas of interest that were at the core of this study, and the focus was put on gooseneck trailer tiny houses. The remainder of this study will refer to them with the general term "tiny houses" if no other signifiers are specified. The first area of interest involved structural items. Instead of spacing structural supports at fixed distances, as traditional house construction practices would require, an analysis would be conducted so the supports could withstand loads that the tiny house would experience. The main loads are supporting their own weight and the weight of the rest of the tiny house, supporting a snow load on the roof in the winter, and withstanding a wind load while being transported.

The second and third areas of interest are stability and weight. Tiny houses need to be stable when being transported on the road, which means they need to be properly balanced. This needs to be true when the tiny house is fully loaded and when it is empty. The empty condition needs to be considered because the tiny homeowner may be renting build space, and once the structure is water tight, the tiny house can be transported to a new location (likely outdoors) to finish working on interior items. The tiny house also needs to be under a certain weight so the trailer is not overloaded.

Fourth were thermal items as the tiny house could be located in a number of different climates. Knowing how much heat is transferred into the living space in the summer and transferred out of the space in the winter is important to the design. This becomes a critical parameter as more windows are added because windows allow heat to transfer in and out of the living space at a greater rate.

The final area of interest is the cost. Being able to select different options and quickly see how those options impact the cost of the project assists in proper budgeting and planning. Costs that are individually shown can help the future tiny homeowner plan what components should be purchased in what order in the event all materials cannot be purchased at once.

2. Literature Review

There are many works that discuss the tiny house topic, and a few of them were the main motivation for this study. Calluari and Alonso-Marroquín [5] discussed the methodology and process for examining a fixed structure tiny house. The model was designed with a shed style roof, a loft space for the bed, and it was supported with four posts secured in the ground. It took into account the material properties of the wooden supports, as well as the grain directions. The study also examined different loading conditions and brace configurations, and it summarized how the resulting deflection of specific points were impacted [5].

Kostoff [6] provided a unique perspective as it discussed how foundation-built tiny houses could be viable living solution for the growing population of “young-old” adults. Interviews were conducted, and while many agreed that foundation-built tiny houses could fit individual lifestyles, spatial restraints were a common concern [6]. As this study focuses on gooseneck style tiny houses, this concern become more prevalent, and considerations need to be made to account for the entire lifespan of the occupants.

Eberle [7] went into great detail about the overall impact a tiny house on wheels (traditional trailer) has on climate change, acidification and ozone depletion. It also contains a list of building materials (specifically focusing on electrical and plumbing) the future tiny homeowner can reference to complete the build as this study focuses on the shell of the tiny house.

Although tiny houses do not have the same exact same construction as traditional residential houses, common structural construction practices and general home building processes were reviewed as well [8, 9].

Lunsford and Lunsford hosted a 4-part webinar series [10, 11, 12, 13] which focused on engineering a tiny house on wheels (traditional trailer). They are HVAC professionals, and they built their own touring tiny home that was presented to interested parties across the country in 2016. While it was done more to promote general building science, the principles still apply to tiny houses.

Lunsford and Lunsford hired a civil engineer to ensure their design was safe when traveling on the highway, and they made it structurally sound enough to withstand tornadoes and hurricanes simultaneously [10]. The considerations of center of mass (COM), wind stress, dynamic stress, water stress, and general shape and framing were all emphasized, and the civil engineer performed the safety calculations prior to construction [10]. They focused on how heat, pressure, moisture, and indoor air quality need to be controlled (maintained at a certain set point), and they discussed how to control them with the minimum HVAC components (heater, cooler/dehumidifier, exhaust fans, fresh air ventilator, water heater, and plumbing) [11, 12]. The basic functions of each HVAC component were also described [12]. There was also a focus on the “bells and whistles” such as formaldehyde-free plywood, built-in furniture, and other items that were going to make the structure feel more like a home [13]. Overall, the work of Lunsford and Lunsford provided a sound background on building science and explained what items needed to be considered when designing a tiny house.

While there are numerous other studies that examined tiny houses, many of them focused on the societal impacts, environmental benefits, and / or general discussions of tiny house living [14, 15, 16, 17, 18]. These topics are just as important, but the focus of this study was put on the technical aspects.

3. Theoretical Background

This study utilized a variety of information from multiple disciplines, and they can be grouped into seven specific categories: utilities, structural, stability, thermal, programming, modeling, and finite element analysis. The user should be aware of some disclaimers, which are listed in Appendix I: List of Disclaimers, and Appendix II: List of Assumptions outlines a number of assumptions that were made in order to complete the study.

3.1. Utilities

There are a few useful topics regarding the various utilities for a tiny house. The first concerns the difference between alternating current (AC) electricity and direct current (DC) electricity. AC electricity, as the name implies, alternates direction at specific intervals over time, while DC electricity remains constant over time. Qualitative examples of these are shown in Figure 5 and Figure 6. As power generation stations are generally located in more remote regions, long transmission lines are needed to reach populated areas. In order to reduce losses as much as possible, AC electricity is utilized. AC electricity is more efficient than DC electricity when transmitted over long ranges because the voltage can be increased, and the current can be decreased after it is generated, which reduces losses in the line. It is then adjusted to nominal values for the general public to use once the transmission is complete. This is done with transformers as AC electricity induces a magnetic field, but the same cannot be done with DC electricity because the constant current does not induce a magnetic field. As transmission lines from the power generation stations carry AC electricity, residential houses are typically wired for it as well. However, personal electronic devices (e.g. laptops and cell phones) utilize DC electricity because they contain batteries, which deliver the needed constant electricity to other components. Therefore, they require AC to DC converters in their chargers.

AC Electricity Varies with Time

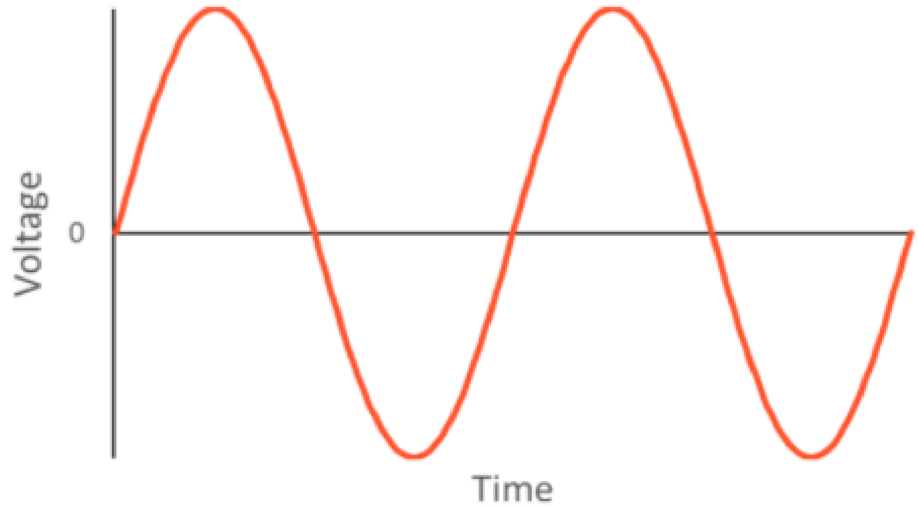


Figure 5: Qualitative Example of AC Electricity, adapted from [19]

DC Electricity Remains Constant with Time

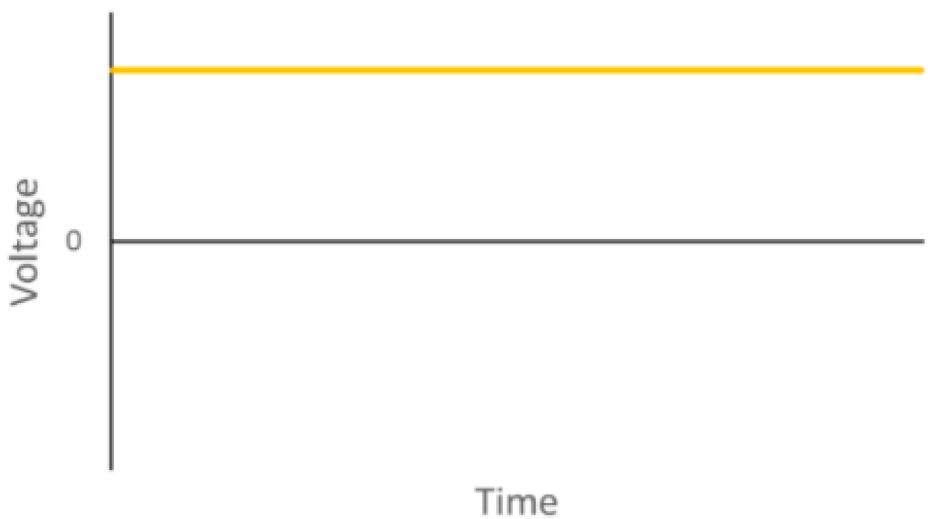


Figure 6: Qualitative Example of DC Electricity, adapted from [19]

The second topic in the utility consideration for a tiny house is the units of a kilowatt-hour (kWh), which is used to measure energy (electricity) consumption. The equation to calculate a Watt (W) from AC electricity is shown in Eq. 1, and the equation to calculate a Watt from DC electricity is shown in Eq. 2. A Watt is energy used per unit time, and a kilowatt (kW) is one thousand Watts (Eq. 3). As a kilowatt is still energy used per unit time, multiplying it by a duration of time (i.e. an hour), the total energy used for a certain amount of time can be determined. Utility companies typically measure total energy used for a month in kWh (Eq. 4). In a more practical sense, one kWh can be thought of as the energy used while a 100 Watt light bulb is powered for 10 hours (Eq. 5), and the estimated average energy consumption for a United States household is 914.3 kWh per month [20].

$$Watts_{AC} = Volts_{RMS} \times Amps_{RMS} \quad \text{Eq. 1}$$

Where RMS is the root mean square value

$$Watts_{DC} = Volts \times Amps \quad \text{Eq. 2}$$

$$1\ kW = 1{,}000\ Watts \quad \text{Eq. 3}$$

$$1\ kWh = 1\ kW \times 1\ hour \quad \text{Eq. 4}$$

$$1\ kWh = (100\ watts \times 10\ hours) \times \frac{1}{1000} \frac{watts}{kW} \quad \text{Eq. 5}$$

The final utility topic is water consumption. To calculate the monthly water consumption, specified values have to be converted. For example, if a washing machine uses 15 gallons per load, the number of loads per week would need to be known. That value could then be adjusted to reflect monthly usage, as shown in Eq. 6 (accounting for two loads of laundry per week). All the different water sources would need to be converted to like units and summed together to determine the total monthly water consumption. The estimated average water consumption in the United States is 3,000 gallons per month per person from one source [21].

$$Example\ monthly\ water\ consumption\ for\ washing\ machine = \quad \text{Eq. 6}$$

$$15 \frac{gallons}{load} \times 2 \frac{loads}{week} \times \frac{52}{12} \frac{weeks}{month} = 130 \frac{gallons}{month}$$

3.2. Structural

In order to provide an adequate background on the structural items associated with this study, the following topics will be reviewed: stress, strain, yield strength, elastic deformation, plastic deformation, and Von Mises stress. Stress, in the engineering sense (denoted by the lowercase Greek letter sigma, σ), is defined as a force applied per unit area. This results in units of pressure (e.g. pounds per square inch or psi, a standard unit in SolidWorks), and an example with one-dimensional components is shown in Figure 7. The calculation is shown in Eq. 7.

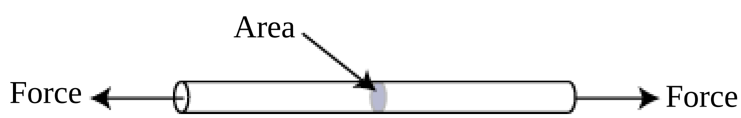


Figure 7: Visual Representation of Stress Components, adapted from [22]

$$\sigma = \frac{Force}{Area} \qquad \textit{Eq. 7}$$

Strain, in the engineering sense (denoted by the lowercase Greek letter epsilon, ϵ), is defined as the change in length of a material divided by the initial length of a material (in the same direction as the force which causes stress). Strain is unitless, and an example with one-dimensional components is shown in Figure 8. The calculation for strain is shown in Eq. 8.

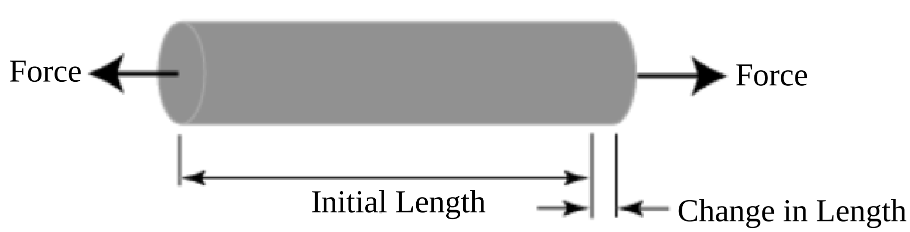


Figure 8: Visual Representation of Strain Components, adapted from [22]

$$\epsilon = \frac{Change\ in\ Length}{Initial\ Length} \qquad \textit{Eq. 8}$$

When stress and strain are plotted against each other, a curve is generated, and each material has its own unique curve. A qualitative example is shown in Figure 9, and the stress-strain curve has a few important items that need to be described.

There is a portion of the curve that is linear, and in that region, the materials acts in an elastic fashion. This means that if the material is under stress, as long as it remains in the elastic region, the material will return to its original shape when the stress is removed. If the material is under stress and the stress breaches the elastic region and enters the plastic region, the material will be permanently deformed when the stress is removed. These zones are shown in Figure 10.

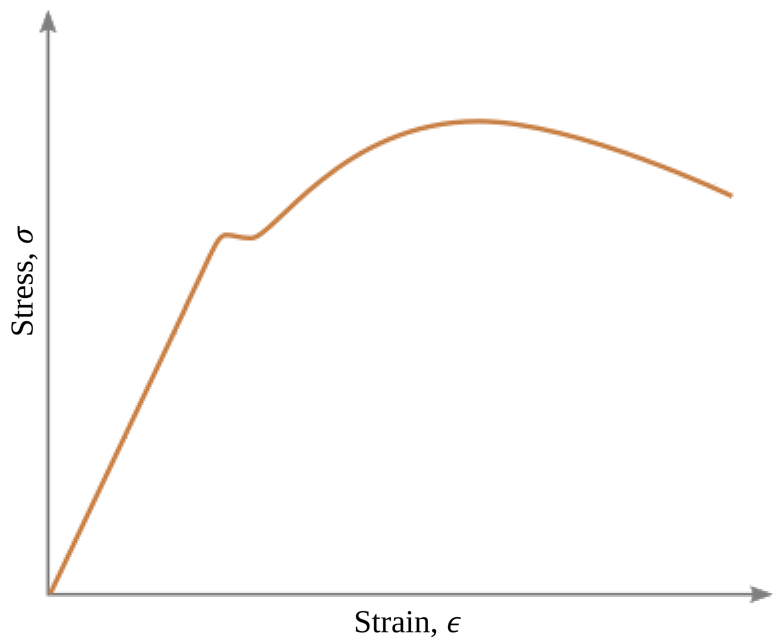


Figure 9: Qualitative Example of a Stress–Strain Curve, adapted from [23]

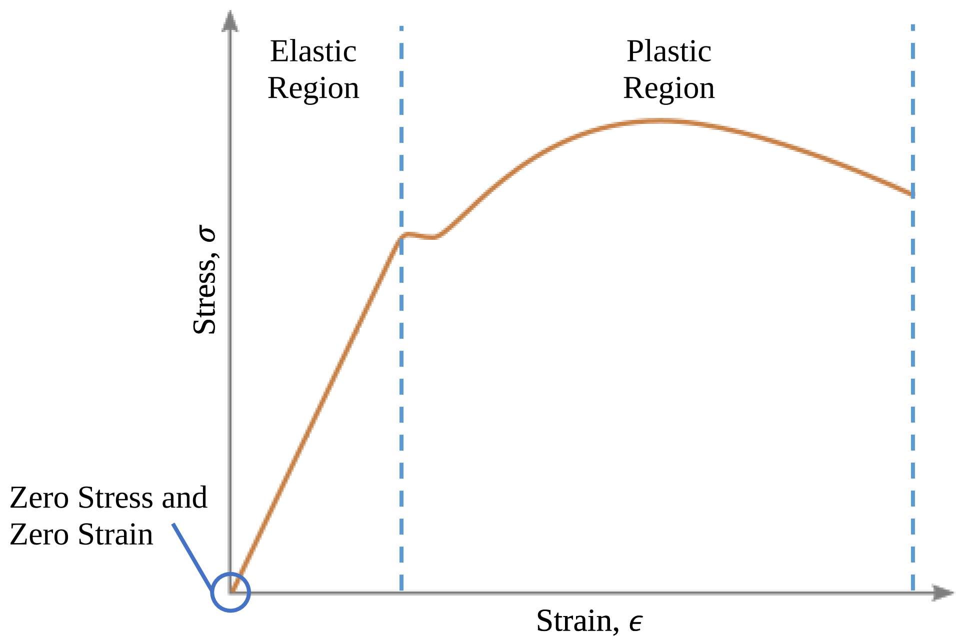


Figure 10: Elastic and Plastic Regions in a Stress–Strain Curve, adapted from [23]

The slope of the line in the elastic region is defined as the modulus of elasticity, E, and it has the same units as stress. The point on the curve that separates the elastic and plastic regions is called the yield strength, and it also has the same units as stress. As long as a stress is below the yield strength, the material will return to its original shape after a stress is removed. These items are visually shown in Figure 11.

It is critical that stresses do not approach the yield strength of the material, because permanent deformation in the plastic zone greatly weakens the material. To protect against this, a safety factor is included in many designs. A safety factor of two, for example, would indicate that stresses would need to be twice as high as expected before the yield strength would be reached. This ensures the material's stresses remain in the elastic zone (Figure 12).

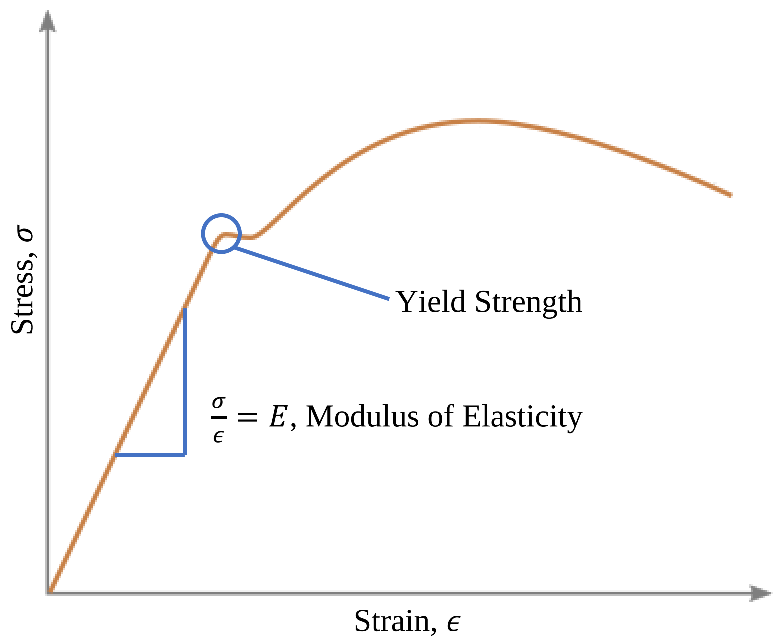


Figure 11: Yield Strength, Modulus of Elasticity on a Stress–Strain Curve, adapted from [23]

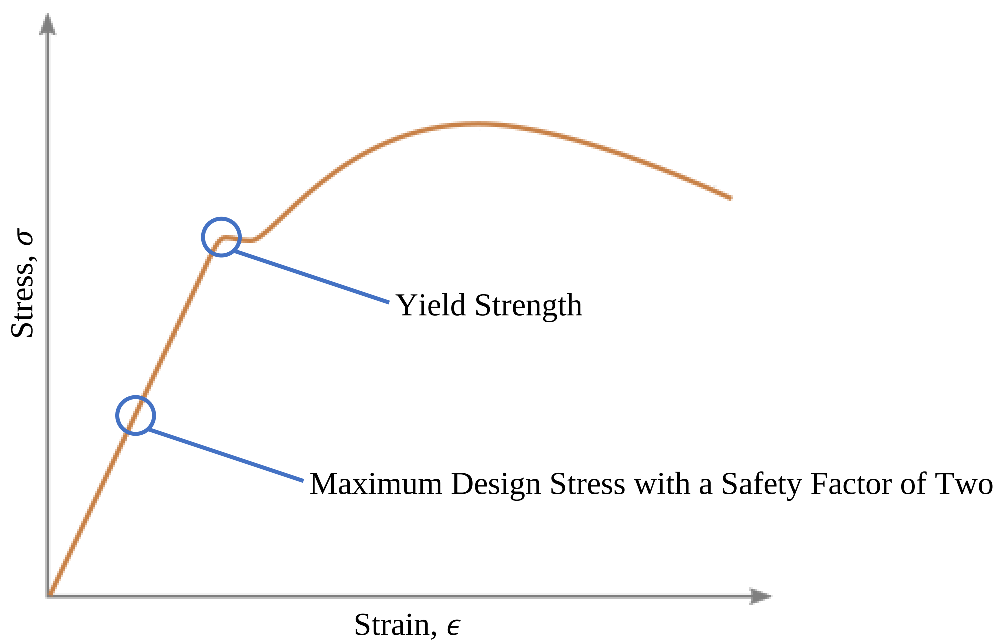


Figure 12: Safety Factor Considerations on a Stress–Strain Curve, adapted from [23]

Finally, real-world stresses actually come in nine components, one axial stress for each of the three dimensions (represented by σ_{xx}, σ_{yy}, and σ_{zz}), and one shear stress for each of the six faces on an element unit (denoted by the lowercase Greek letter tau with subscripts: τ_{xy}, τ_{xz}, τ_{yx}, τ_{yz}, τ_{zx}, and τ_{zy}), as shown in Figure 13 and Eq. 9. To simplify this, the von Mises stress is commonly used during analyses of ductile materials. It takes the nine stress components and reduces them into one value (Eq. 10). The single von Mises stress value can then be compared against the yield strength of the material to determine if the applied stresses pushes the material into the plastic region of its stress-strain curve.

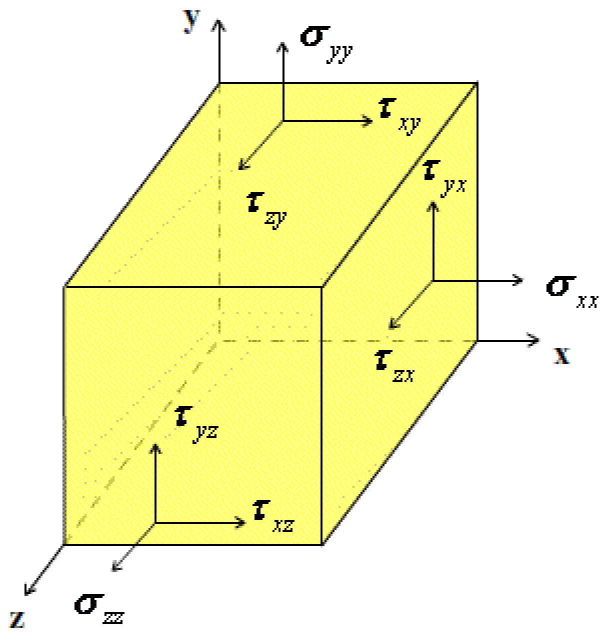


Figure 13: Example of Stress Components on an Element Unit [24]

$$\sigma = \begin{bmatrix} \sigma_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_{zz} \end{bmatrix} \qquad \textit{Eq. 9}$$

$$\sigma_{von\ Mises} = \sqrt{\frac{(\sigma_{xx}-\sigma_{yy})^2+(\sigma_{yy}-\sigma_{zz})^2+(\sigma_{zz}-\sigma_{xx})^2+6\times(\tau_{xy}^2+\tau_{yz}^2+\tau_{zx}^2)}{2}} \qquad \textit{Eq. 10}$$

3.3. Stability

The items the general public may need to know in order to understand stability are the COM, tongue weight, and superelevation. The COM is a point in an object that represents weighted average location of all individual mass elements, relative to the origin. In a more general sense, it can be thought of as the point that the object could be theoretically balanced on. The COM is an important location because if all external forces acting on the object were summed and applied to the COM, the object would react in the same manner. The equations to determine the COM are shown in Eq. 11 to Eq. 15.

$$Origin = [0, 0, 0] \qquad \textit{Eq. 11}$$

$$COM = [x_{COM}, y_{COM}, z_{COM}] \qquad \textit{Eq. 12}$$

$$x_{COM} = \frac{\sum_{i=1}^{N}(m_i \times x_i)}{m} \qquad \textit{Eq. 13}$$

$$y_{COM} = \frac{\sum_{i=1}^{N}(m_i \times y_i)}{m} \qquad \textit{Eq. 14}$$

$$z_{COM} = \frac{\sum_{i=1}^{N}(m_i \times z_i)}{m} \qquad \textit{Eq. 15}$$

Where:

- Σ is the mathematical notation to represent a sum
- i is the index of summation which begins at 1
- N is the total number of elements
- m is the total mass of the object
- m_i is the index of each mass element
- x_i is the index of each x distance from the mass element to the origin
- y_i is the index of each y distance from the mass element to the origin
- z_i is the index of each z distance from the mass element to the origin

Tongue weight is the downward force that the trailer applies on the towing vehicle's hitch. If this value is too high, the towing vehicle's rear axle may not be able to support the load. If this value is too low, there may not be enough weight on the towing vehicle's rear axle to provide adequate traction. It is only when the trailer has the correct weight distribution that the proper tongue weight is applied to the towing vehicle's hitch. Exaggerated examples of these situations are shown in Figure 14. Once the tongue weight is known, the appropriate towing vehicle can be selected.

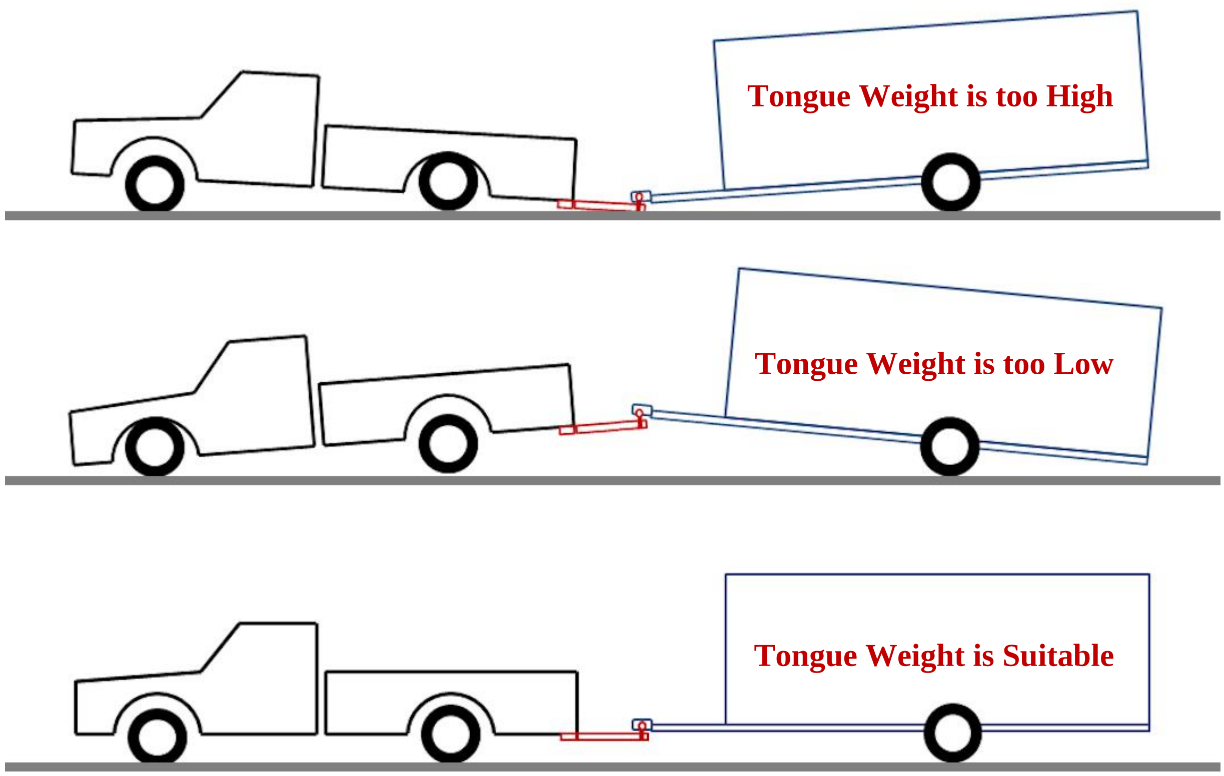


Figure 14: Exaggerated Examples of Different Tongue Weights, adapted from [25]

Superelevation is the grade at which a road is angled. This is done to help egress water off roads and to increase safety while driving on curves. A visual representation of superelevation is shown in Figure 15. Superelevation is an important consideration in the design of a tiny house because the maximum superelevation of the roads where the structure will be transported on may determine if it will tip over or not.

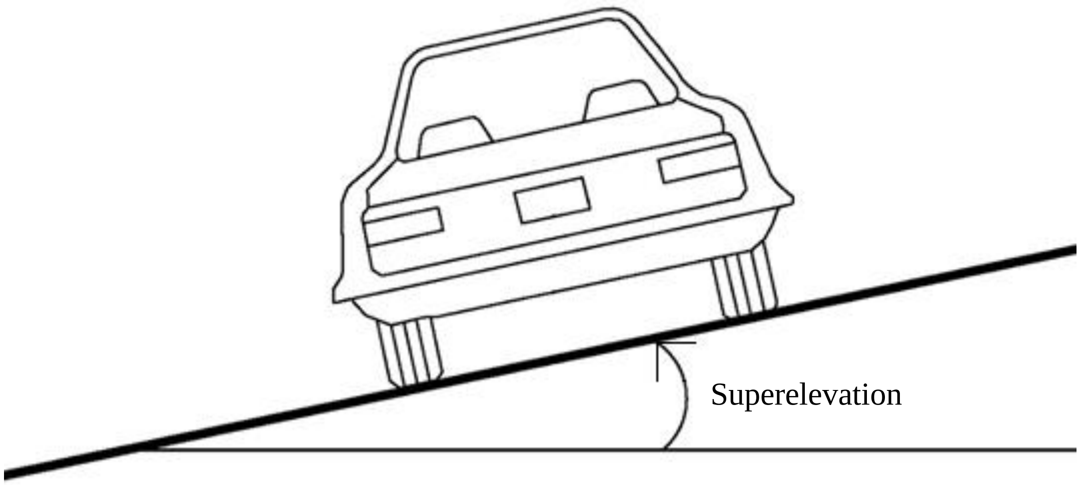


Figure 15: Visual Representation of Superelevation, adapted from [26]

3.4. Thermal

In order to provide an adequate background on the thermal items associated with this study, the following topics will be reviewed: the First Law of Thermodynamics, the difference between heating and cooling, the difference between conduction, convection, and radiation, and the difference between sensible heat and latent heat, and live load heat. The First Law of Thermodynamics, also known as the Law of Conservation of Energy, states that energy can neither be created nor destroyed. It can only be converted from one form to another, and it can be described with Eq. 16. To transfer heat into or out of a system, it must be heated or cooled.

$$\Delta U = W + Q \qquad \textit{Eq. 16}$$

Where:

ΔU is the change in internal energy of the system
W is the work done by the system or on the system
(different than the W used as the abbreviation for Watts)
Q is the heat transferred into or out of the system

Many use the terms heating and cooling as if both can be added to a system, but this is a misconception. Heating means that energy is transferred into a system, which can increase its temperature. Cooling means that energy transferred out of a system, which can decrease its temperature. Heat can be spontaneously transferred out of a system due to a difference in temperature between two substances, and it can be transferred out of a system in a more controlled manner using the standard refrigeration cycle, which can be reviewed in more detail in [27]. Heat can be transferred into a system using conduction, convection, or radiation.

Conduction transfers heat because particles within the material collide with each other, which causes heat to propagate down the material. It can be described with Eq. 17, and an example is heating a metal pot over a fire and the handle getting hotter as well (Figure 16). Convection transfers heat with different fluids (e.g. water or air), and it can be described with Eq. 18, also refered to as Newton's Law of Cooling. An example is the same pot over a fire, but convection would be the hot water heating the surrounding air (Figure 16). Radiation transfer heat through the generation of electromagnetic waves which are either absorbed or reflected. It can be described with Eq. 19, and an example is the heat generated from a fire (Figure 16).

$$\vec{q} = -k \times \nabla T \qquad \textit{Eq. 17}$$

Where:

$\vec{q}$ is the heat flux
k is the material's thermal conductivity
(different than the k used as abbreviation for kilo, as in kilowatt)
∇T is the temperature gradient

$$\dot{Q} = h \times A \times \Delta T \qquad \textit{Eq. 18}$$

Where:

$\dot{Q}$ is the heat transfer per unit time
h is the convective heat transfer coefficient
A is the area over which the heat transfer is occurring
ΔT is the temperature difference between the surface and adjacent fluid

$$\dot{Q} = \sigma \times e \times A \times T^4 \qquad \textit{Eq. 19}$$

Where:

$\dot{Q}$ is the heat transfer per unit time
σ is the Stefan-Boltzmann constant
(different than the σ used to represent stress)
e is the emissivity of the object
A is the area over which the heat transfer is occurring
T is the temperature

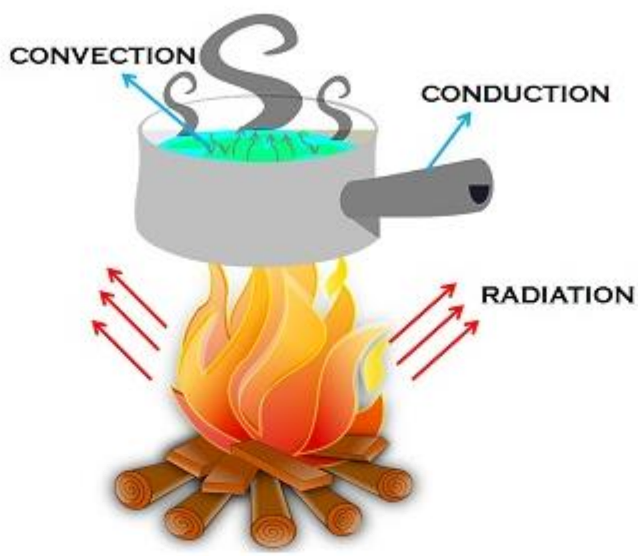


Figure 16: Examples of Conduction, Convection, and Radiation [28]

Heat has different forms, sensible and latent, and it can be generated from different sources. Sensible heat is the heat associated with the change in temperature of an object, without changing state. For example, it is the heat associated with making colder water warmer, and it can be thought of as the heat associated in making a thermometer change. Latent heat is the heat associated with a change in state, without changing temperature. For example, it is the heat associated with making 212°F (100°C) water change state from a liquid to a vapor. Qualitative examples of sensible and latent heat are shown in Figure 17. Sensible heat can be generated from living entities and electrical devices, and this study refers to that as live load heat. Human beings, animals, etc. all radiate heat, and electrical devices have flowing electrons which generate friction within the circuitry, which also generates heat. These heat sources need to be taken into account when calculating the HVAC needs for a space.

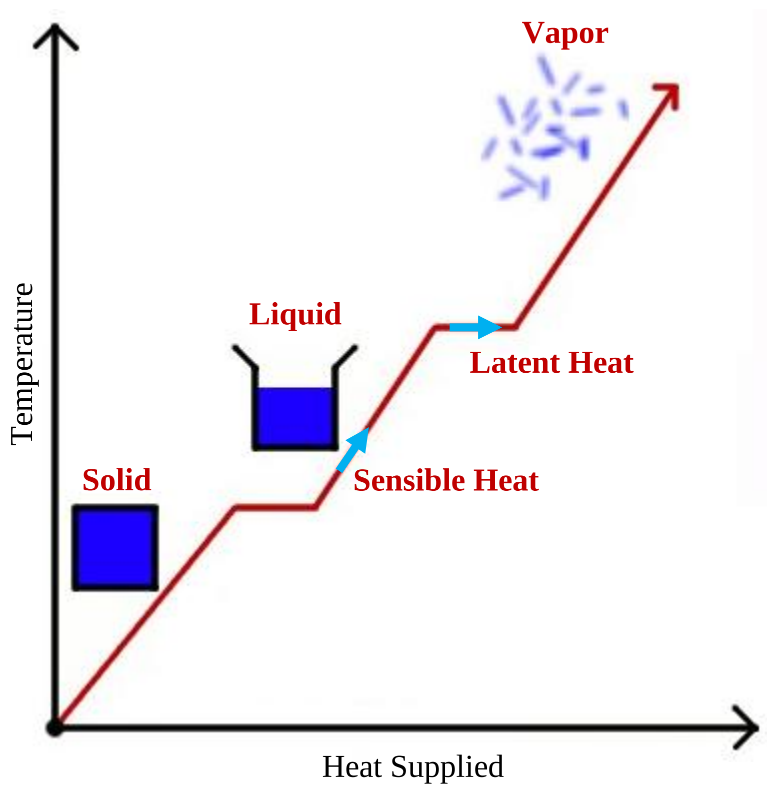


Figure 17: Qualitative Examples of Sensible and Latent Heat, adapted from [29]

3.5. Programming

While individuals do not have to know or understand how the code in the Scilab design tool works or is structured, Table 1 is provided to give insights to the frequently used objects within the design tool code. Each can be looked up in more detail with Scilab’s help website [30].

Table 1: Frequently Used Scilab Code Objects

<table>
<tr><th>Object</th><th>Description</th></tr>
<tr><td>‘ ‘ or “ “</td><td>String designators</td></tr>
<tr><td>%f</td><td>Boolean false</td></tr>
<tr><td>%t</td><td>Boolean true</td></tr>
<tr><td>*</td><td>Multiplication symbol</td></tr>
<tr><td>,</td><td>Comma</td></tr>
<tr><td>/</td><td>Division symbol</td></tr>
<tr><td>//</td><td>Comment symbol</td></tr>
<tr><td>;</td><td>Semicolon</td></tr>
<tr><td>|</td><td>Logical or</td></tr>
<tr><td>+</td><td>Addition symbol</td></tr>
<tr><td>=</td><td>Assignment symbol</td></tr>
<tr><td>==</td><td>Equality symbol</td></tr>
<tr><td>></td><td>Greater than symbol</td></tr>
<tr><td>clc;</td><td>Expression that clears the command window</td></tr>
<tr><td>disp( )</td><td>Displays item within parentheses</td></tr>
<tr><td>else</td><td>Conditional expression that evaluates when all other statements are false</td></tr>
<tr><td>elseif</td><td>Conditional expression that evaluates when alternate statement is true</td></tr>
<tr><td>end</td><td>Terminates expression</td></tr>
<tr><td>endfunction</td><td>Terminates function</td></tr>
<tr><td>exec( )</td><td>Executes item within parenthesis</td></tr>
<tr><td>function</td><td>Object that accepts an input (option), performs a task, and provides an output (option)</td></tr>
<tr><td>if</td><td>Conditional expression that evaluates when true</td></tr>
<tr><td>input( )</td><td>Object that prompts the user to enter a value or string</td></tr>
<tr><td>while</td><td>Looping object which continues until a specified condition is no longer true</td></tr>
</table>

3.6. Modeling

SolidWorks is one of many computer aided design (CAD) software packages that could be used to model a tiny house, and while the Scilab design tool does not require the user to understand or be familiar with the CAD model, the following items outline different functions and features that were used to create it (all linear dimensions are inches). A part can be created by selecting a new "Part" from the start menu (Figure 18). With a blank file, a two-dimensional (2D) sketch can be drawn with different shapes and lines (circles, rectangles, curves, straight lines, etc.), and measurements can be set and named (Figure 19 and Figure 20).

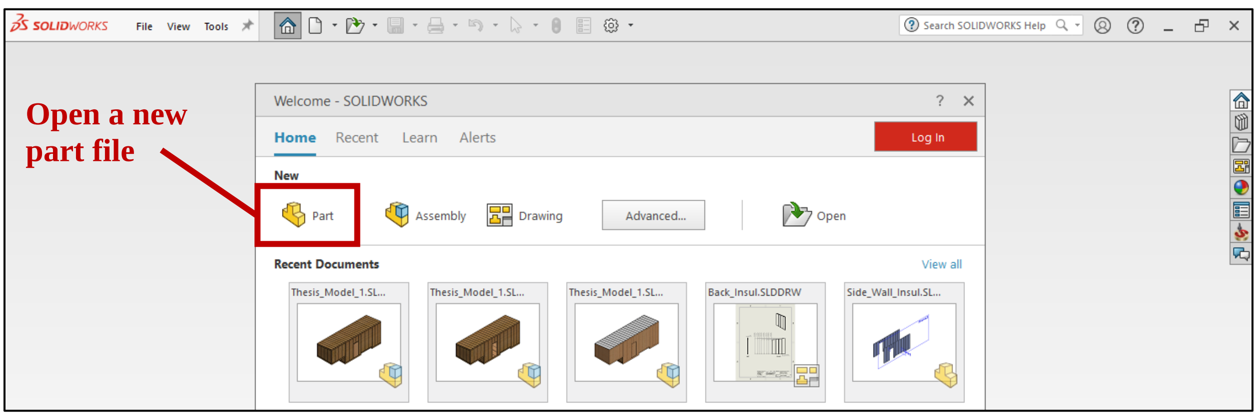


Figure 18: SolidWorks Start Menu

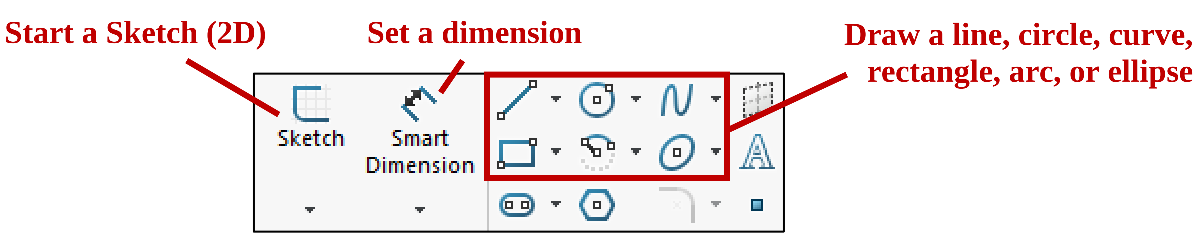


Figure 19: Part of the SolidWorks Sketch Menu

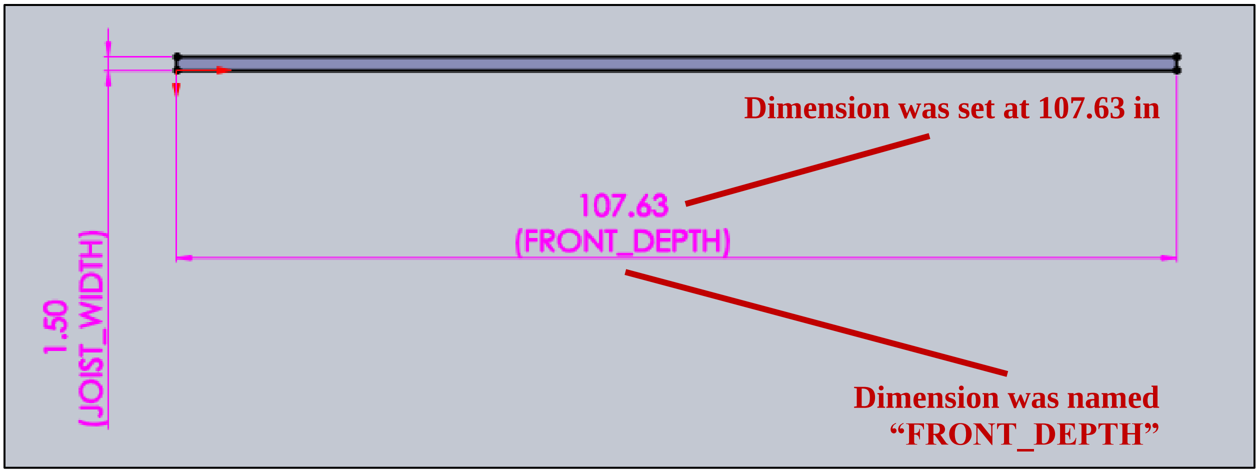


Figure 20: Example of Rectangular Sketch with Named Dimensions

(Top View)

The extrude function can be applied to the 2D sketch to turn it into a three-dimensional (3D) body (Figure 21). The properties can then be set so the body behaves like a certain material (Figure 22). This process can be repeated to create additional bodies, and when applicable, those bodies can be repeated in a linear pattern to save modeling time (Figure 23).

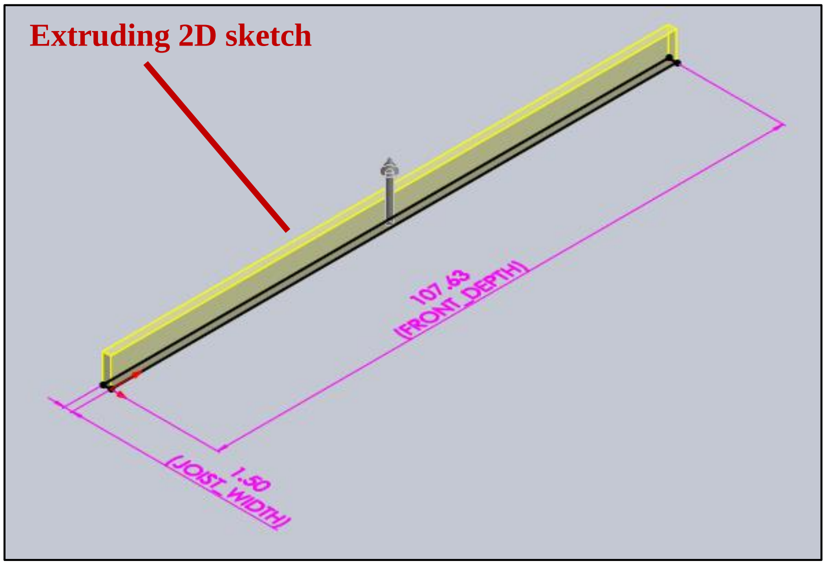


Figure 21: Example of Extrude Function being Applied to Rectangular Sketch

(Isometric View)

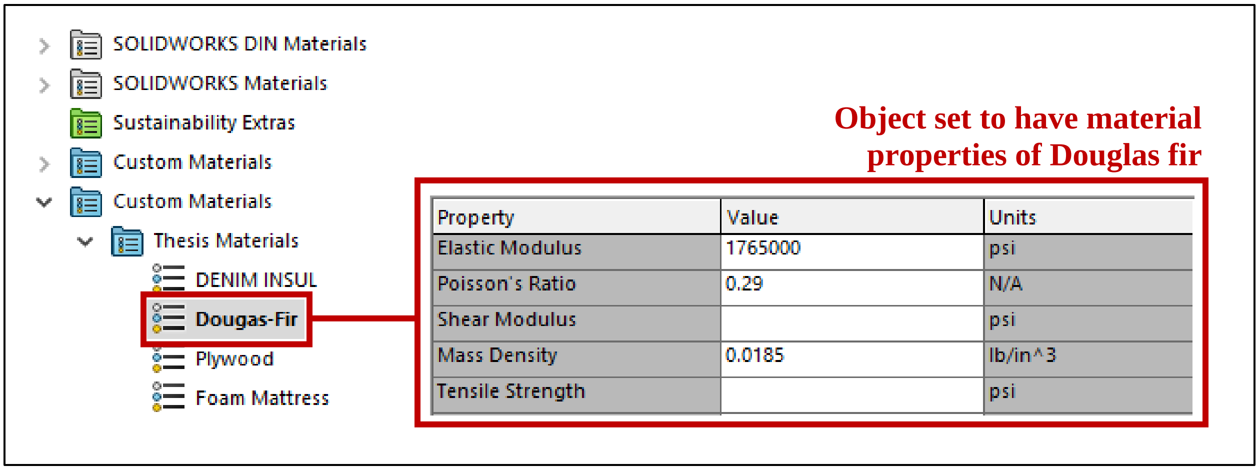


Figure 22: Material Property Window for Douglas fir

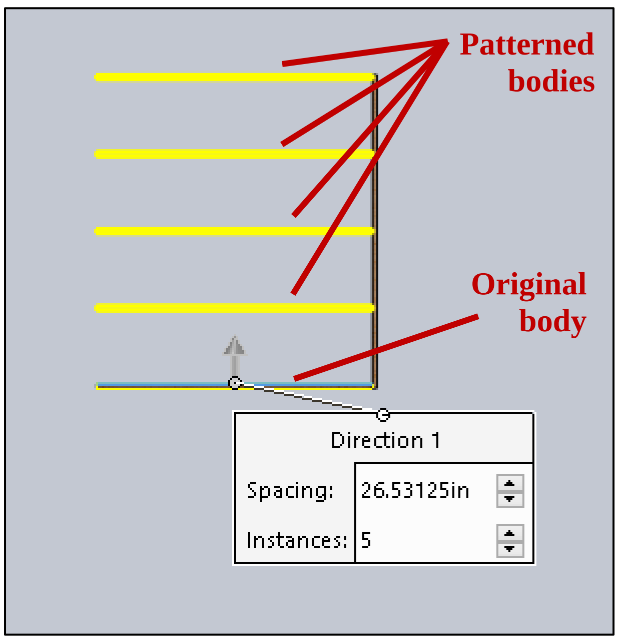


Figure 23: Example of Linear Pattern

(Top View)

The spacing of the linear pattern can be driven by an equation that references various named dimensions to ensure the spacing is always equal, regardless of the pattern number to also save modeling time (Figure 24). Custom planes can also be defined, and objects can be mirrored about those planes as one method to finish a single part (Figure 25 and Figure 26).

Figure 24: Example of Linear Pattern Equation

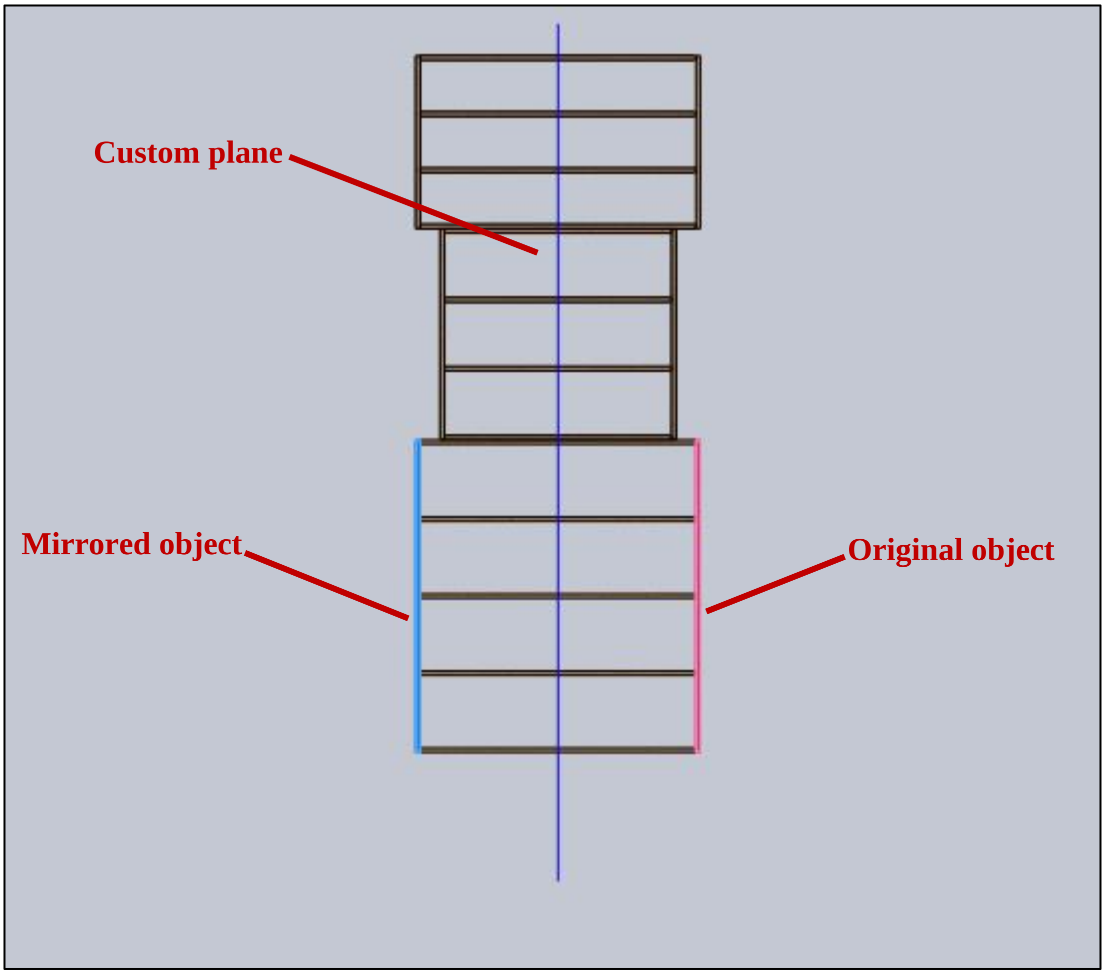


Figure 25: Example of a Custom Plane, Original Object, and a Mirrored Object

(Top View)

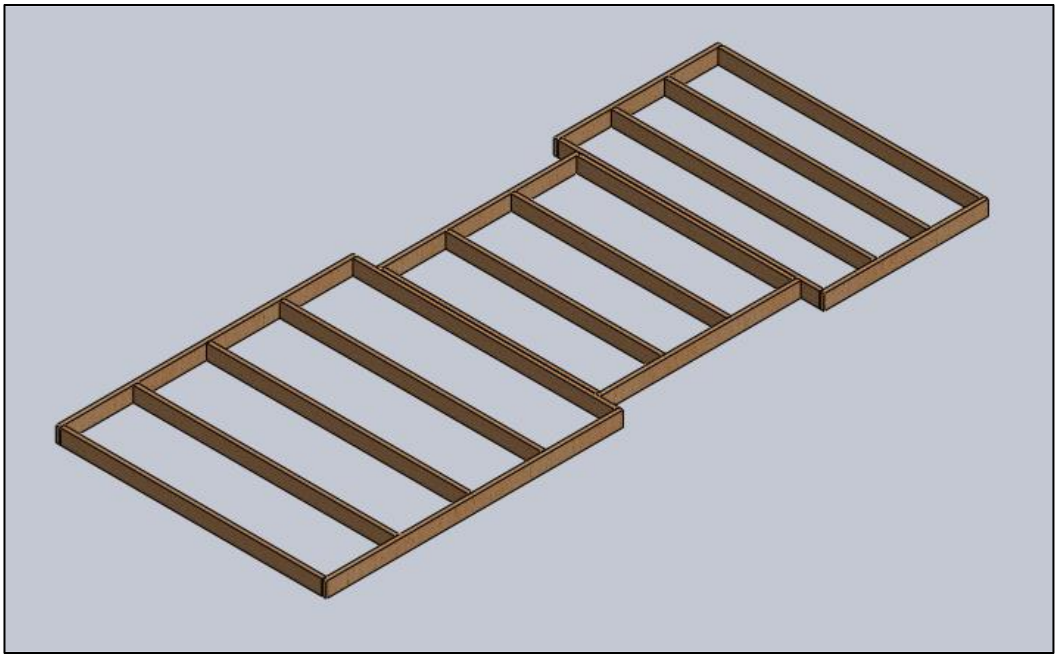


Figure 26: Example of a Completed Part

(Isometric View)

With completed parts, they can be mated together within an assembly (Figure 27 and Figure 28). Mates are useful to make certain new parts are in specific locations, relative to other parts. With more parts, a finished assembly can be created. Finally, whether examining a part or assembly, different values can be recorded from the model. Part of the evaluate menu is shown in Figure 29, an example of the measure tool is shown in Figure 30, an example of the mass properties tool is shown in Figure 31, and an example of the section properties tool is shown in Figure 32.

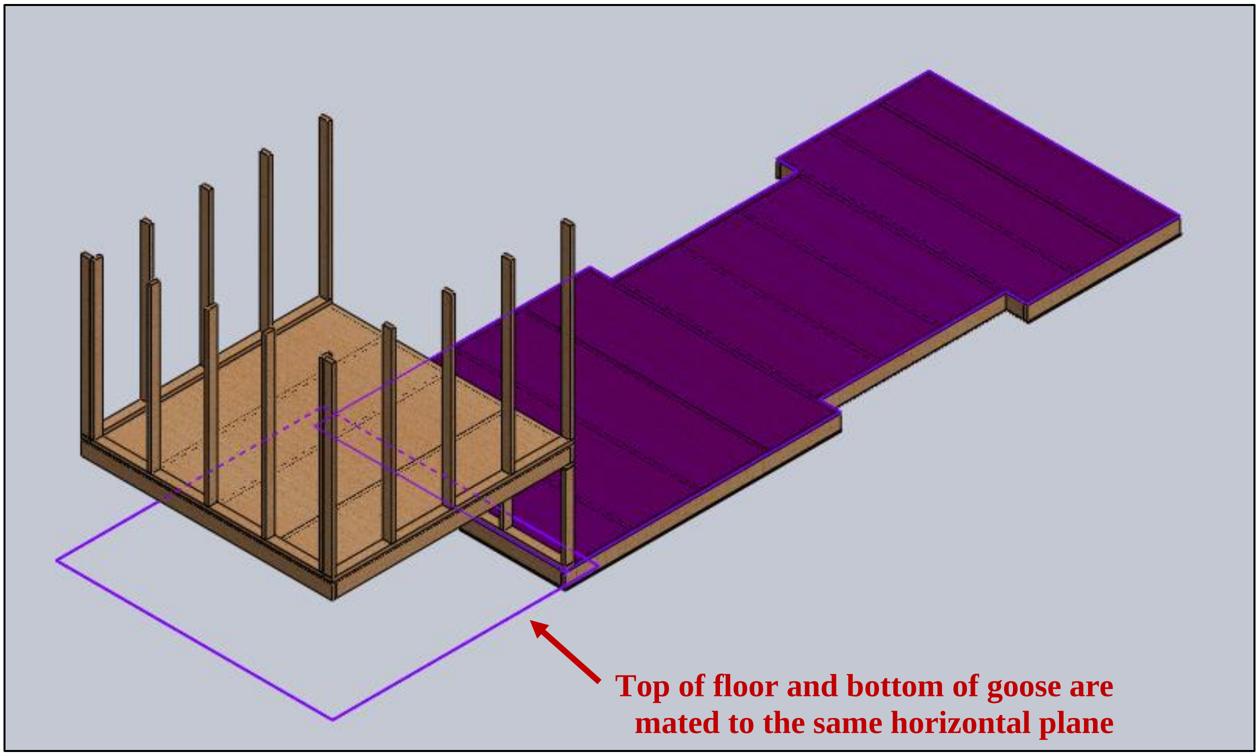


Figure 27: Example of Two Parts Mated on a Plane

(Isometric View)

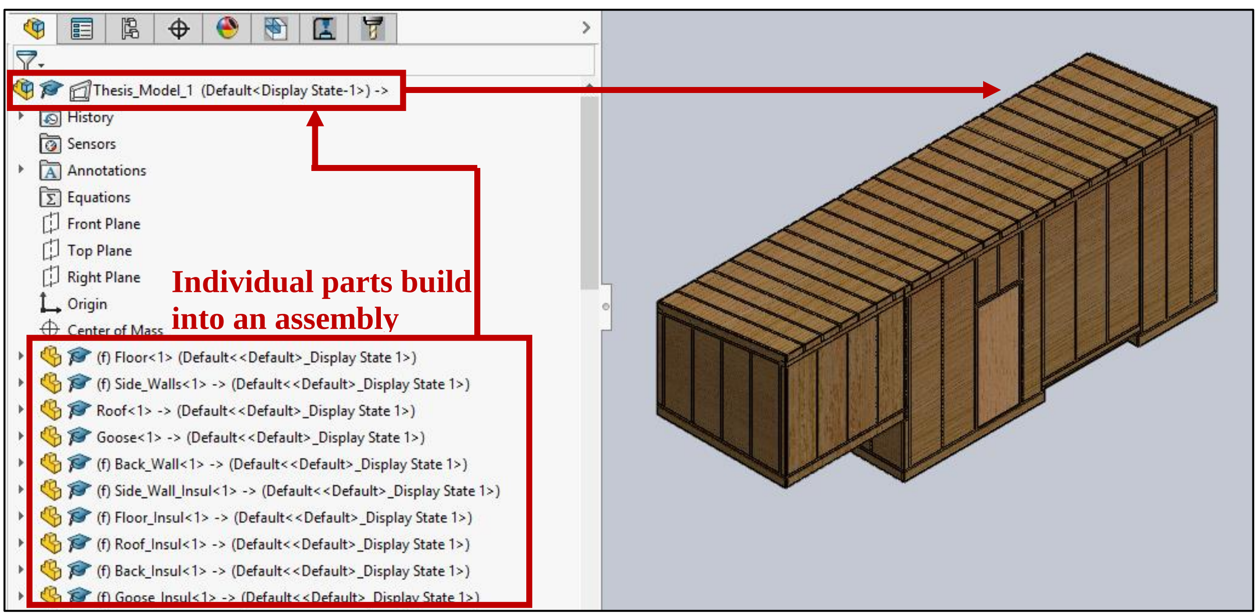


Figure 28: Example of an Assembly

(Isometric View)

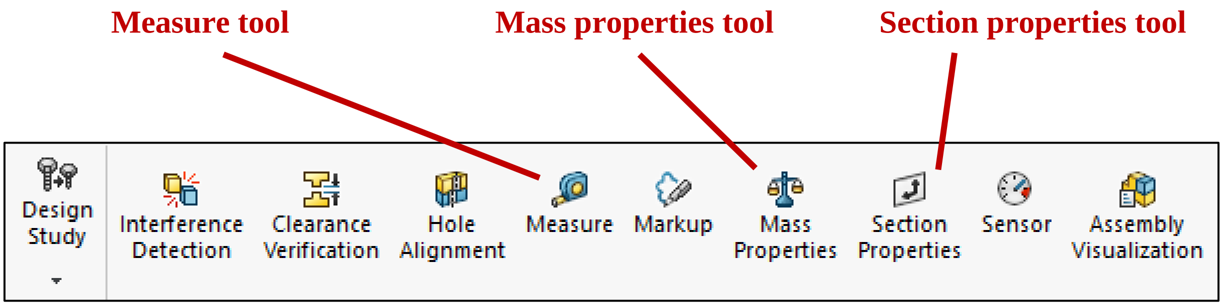


Figure 29: Part of the SolidWorks Evaluate Menu

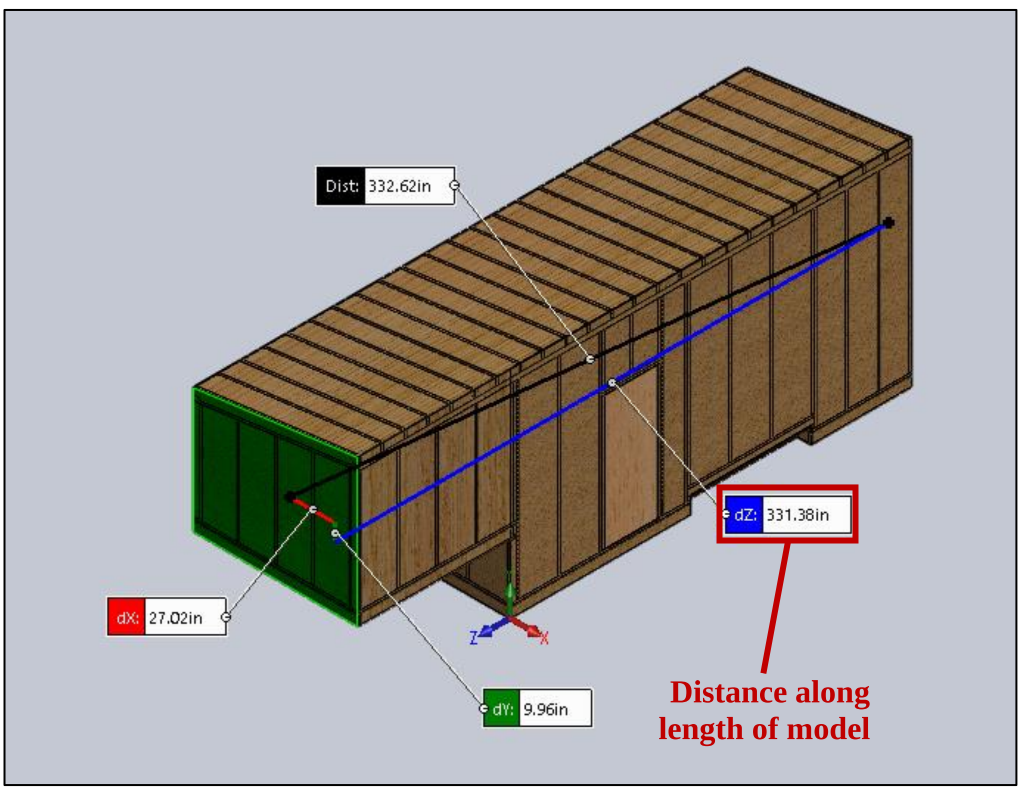


Figure 30: Measure Tool Used to Determine the Distance Between the Front and Rear Walls (Isometric View)

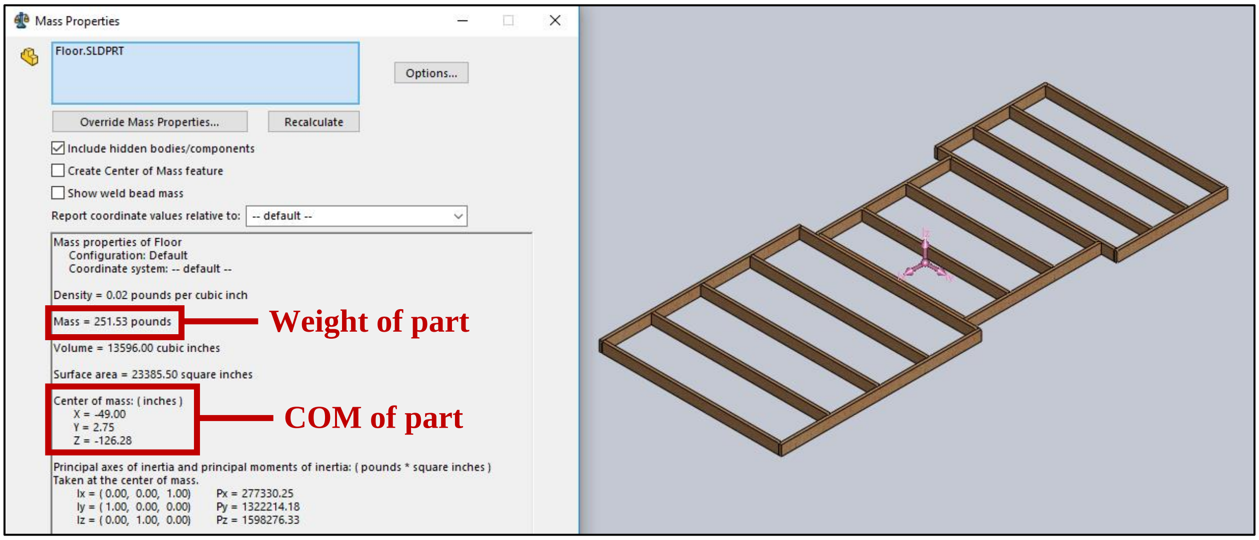


Figure 31: Mass Properties Tool Used to Determine the COM and Weight of the Floor Part

(Isometric View)

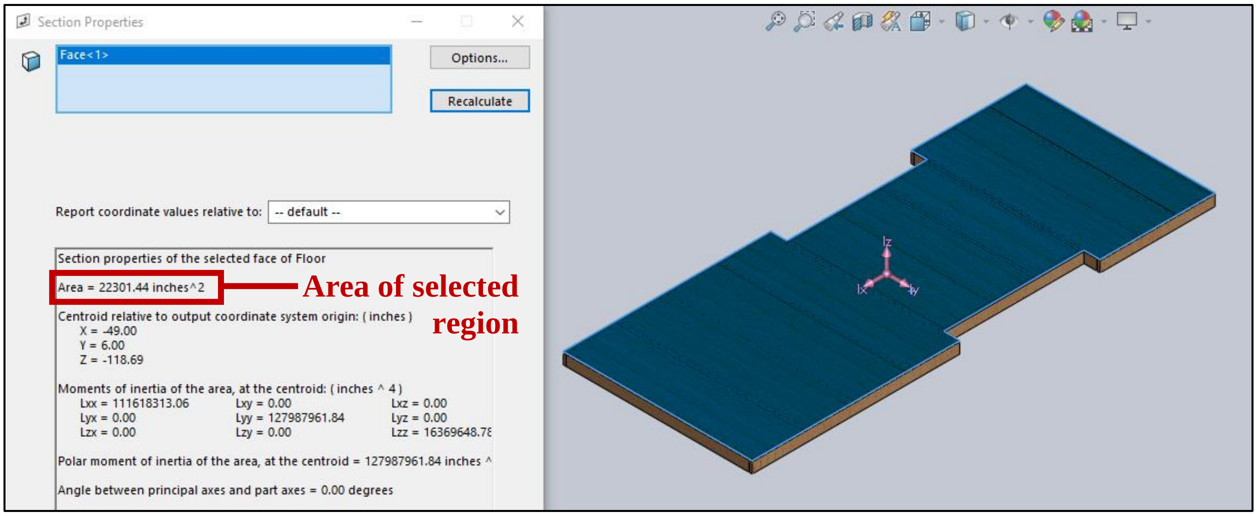


Figure 32: Section Properties Tool Used to Determine Area of Plywood on Floor Part

(Isometric View)

3.7. Finite Element Analysis

SolidWorks Simulation was utilized in order to carry out both the structural Finite Element Analysis (FEA) and the thermal FEA for this study. The goal of FEA is to solve a system of equations to determine the reaction of a material based on a given condition (Eq. 20), and to solve for the reaction, the system must be manipulated (Eq. 21). Table 2 provides descriptions for each variable, based on discipline. When dealing with simple scenarios, the solution can be calculated by hand. Examples of structural and thermal analyses with a cantilevered beam are shown below (Figure 33 and Figure 34).

$$[K][u] = [F] \qquad \textit{Eq. 20}$$

$$[u] = [K]^{-1}[F] \qquad \textit{Eq. 21}$$

Where:

K is the material property
u is the material's reaction
F is the given condition

Table 2: FEA Variable Descriptions Based on Discipline, adapted from [31]

Discipline	**Material Reaction, $[\boldsymbol{u}]$**	**Material Property, $[\boldsymbol{K}]$**	**Condition, $[\boldsymbol{F}]$**
Structural	Displacement	Stiffness	Force
Thermal	Temperature	Conductivity	Heat Source

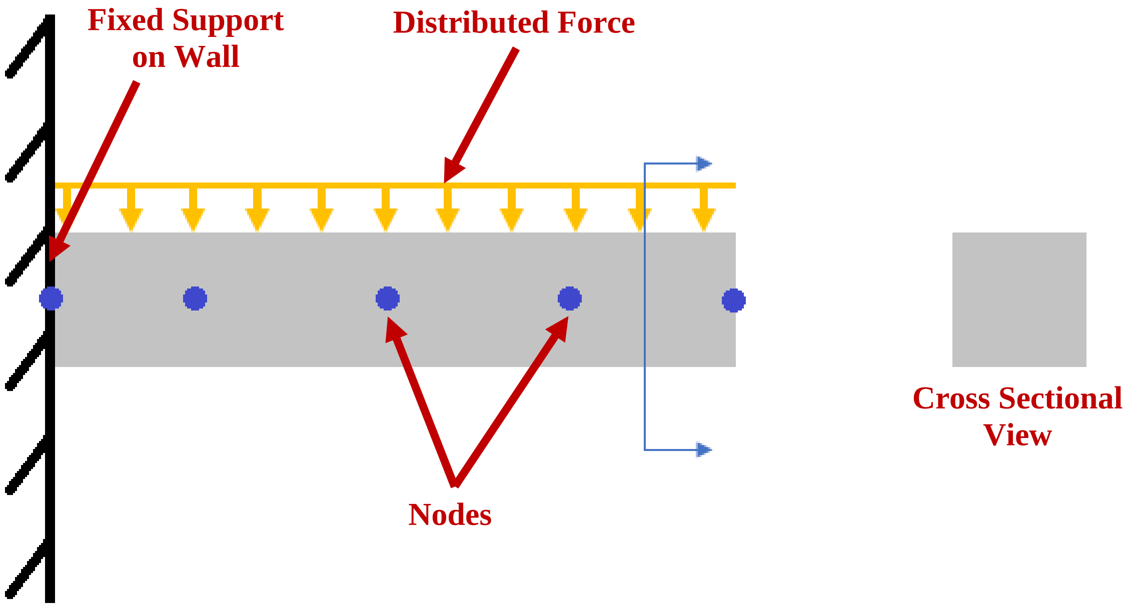


Figure 33: Simple Example of Structural FEA

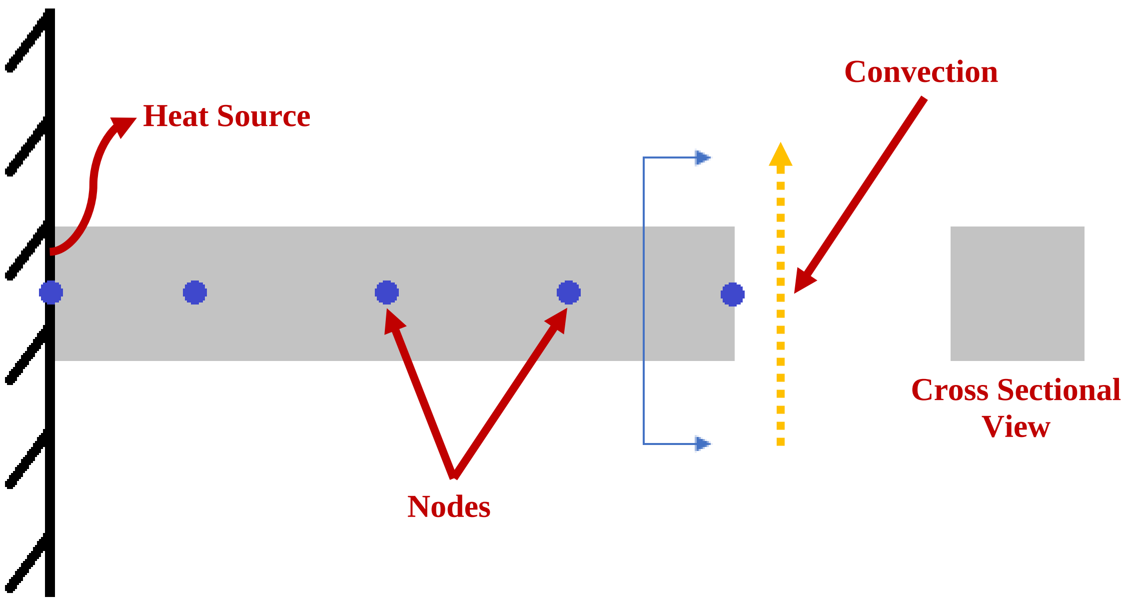


Figure 34: Simple Example of Thermal FEA

In order to solve each problem, nodes needed to be added. These act as locations that the solution focuses on. The more nodes there are, the more accurate the solution will be, but it will also be more difficult to solve as the system of equations becomes larger. In the simple examples, five nodes were used. If the following information for each system is provide, the solution can be determined.

- Items needed to solve the simple structural FEA example:
    - Distance between nodes
    - Width and height of material in cross sectional view
    - Magnitude of distributed force
    - Material in question
- Items needed to solve the simple thermal FEA example:
    - Distance between nodes
    - Width and height of material in cross sectional view
    - Magnitude of heat source and convection
    - Material in question

For each example, two different versions of Eq. 22 would be generated where "N" (the number of nodes) would be five in each case. With matrix multiplication (and possibly a calculator), the displacement of each node in the structural example and temperature of each node in the thermal example can be calculated. With that information, the stresses can be calculated for the structural example, and the heat flux can be calculated for the thermal example.

$$\begin{bmatrix} u_1 \\ \vdots \\ u_N \end{bmatrix} = \begin{bmatrix} K_{11} & \cdots & K_{1N} \\ \vdots & \ddots & \vdots \\ K_{N1} & \cdots & K_{NN} \end{bmatrix}^{-1} \begin{bmatrix} F_1 \\ \vdots \\ F_N \end{bmatrix} \qquad \textit{Eq. 22}$$

In real practice however, the examples shown in Figure 33 and Figure 34 would not be sufficient as they outline ideal situations with simple geometries. SolidWorks Simulation uses the same concepts that were previously described but allows for much greater complexity. Instead of individual nodes, the software generates a mesh for the specified 3D geometry which is comprised of elements, and a full example is shown in Figure 35. The vertices of these elements are nodes, and the nodes serve the same purpose as in the simple examples. The mesh elements can have straight edges, curved edges, or a mix of both to best replicate the imported geometry. After a mesh has been generated, the different boundary conditions and initial conditions can be applied. These could be forces or fixed locations for structural simulations or convection conditions and heat sources for thermal calculations. To solve the simulation, the software creates its own version of Eq. 22, but with "N" being very large (can easily be in the millions for some simulations). With a higher "N" value, the solution is more accurate, but it takes more computational time to solve. To calculate primary results (displacement for structural or temperatures for thermal) the software solves the system of equations iteratively until they converge. From there, secondary results can be calculated as well (stresses for structural or heat flux for thermal).

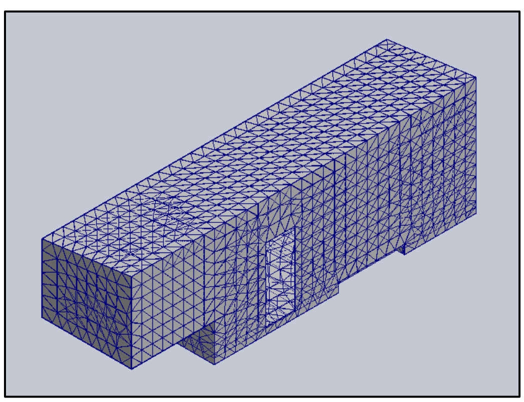


Figure 35: Example of 3D Mesh

(Isometric View)

Figure 36 shows an example of a structural FEA menu, and Figure 37 shows an example of a thermal FEA menu. By going through each menu within SolidWorks, simulations can be set up (Parts, Connections, Fixtures, Loads, Mesh), run (calculates answer), and analyzed (Results). The menu is very similar for thermal FEA simulations, but it accounts for the different boundary conditions.

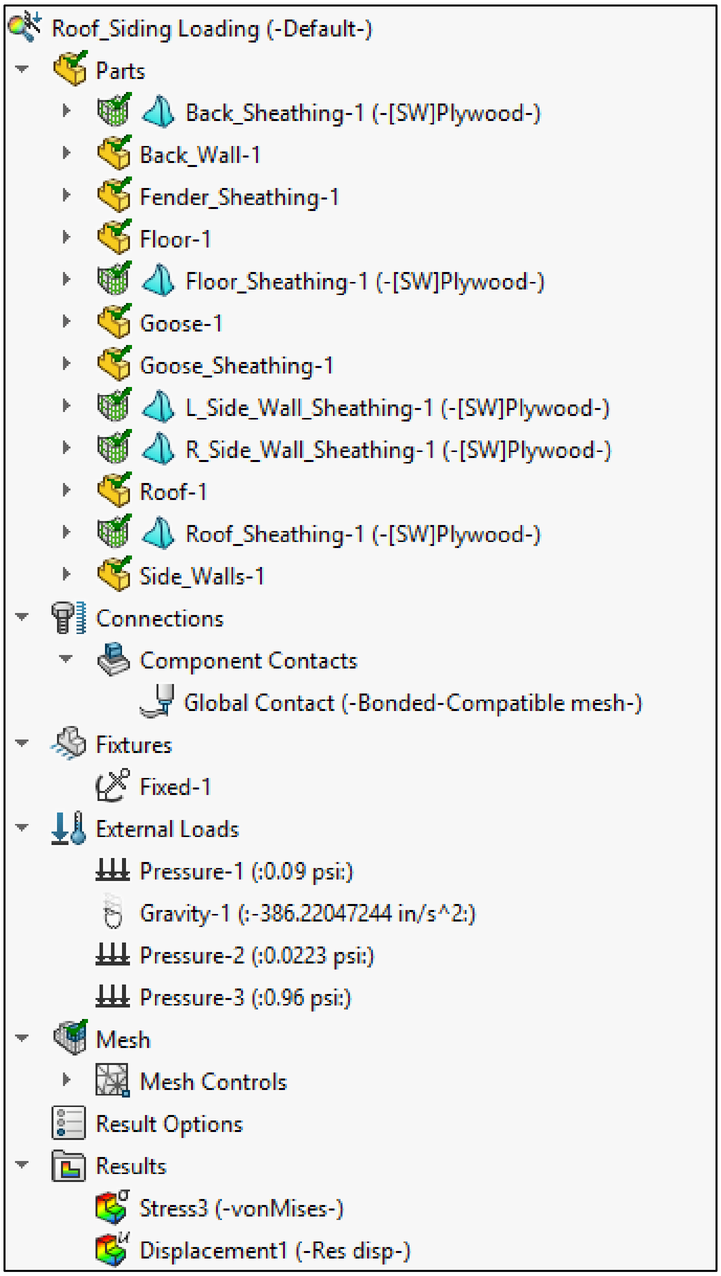


Figure 36: Example of Structural FEA Software Menu

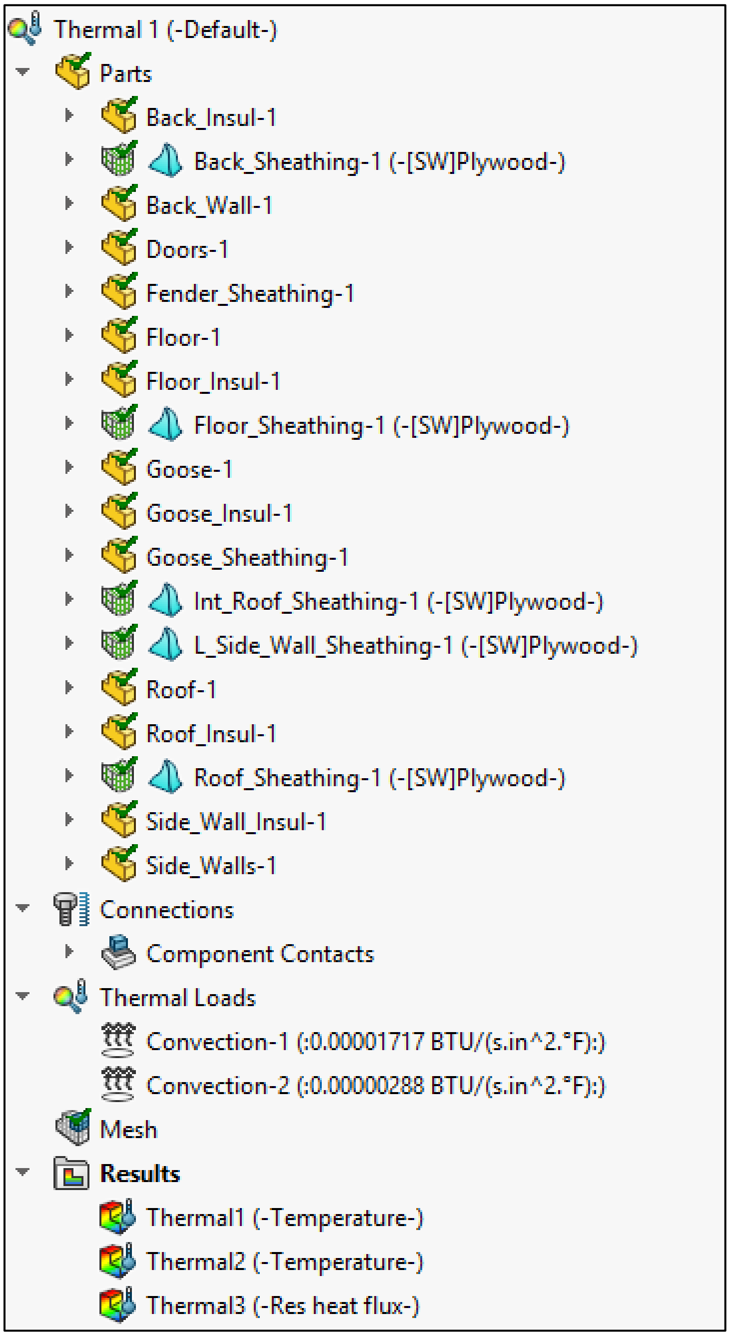


Figure 37: Example of Thermal FEA Software Menu

4. Methodology

Data was first collected for the different regional locations and utility usage values were also researched. These items determined what questions the user should be asked. Next, a SolidWorks model was created, and multiple structural analyses were conducted in addition to verifying a stable COM. Then multiple thermal analyses were conducted. Once finalized, the bills of material (BOMs) and projected cost were determined, and construction drawings were created. This information was then summarized and coded into the resulting Scilab design tool. While the figures and information below may only refer to the 20 foot (ft) model (8 ft goose, 20 ft lower deck) in some instances, the same process was used for the 24 ft model (8 ft goose, 24 ft lower deck).

4.1. Data Collection

To begin the design process, 12 host cities were chosen across the contiguous United States. The location of the cities is shown in Figure 38 and the full listing is provided in Table 3. The selected host cities provided a variation in both physical location and regional weather conditions. For each city, the following values were recorded: 1% extreme wind speed, maximum 1-month precipitation, 99% heating dry bulb temperature, and 1% cooling dry bulb temperature. The summary of this data can be found in Appendix III: Regional Weather Data.

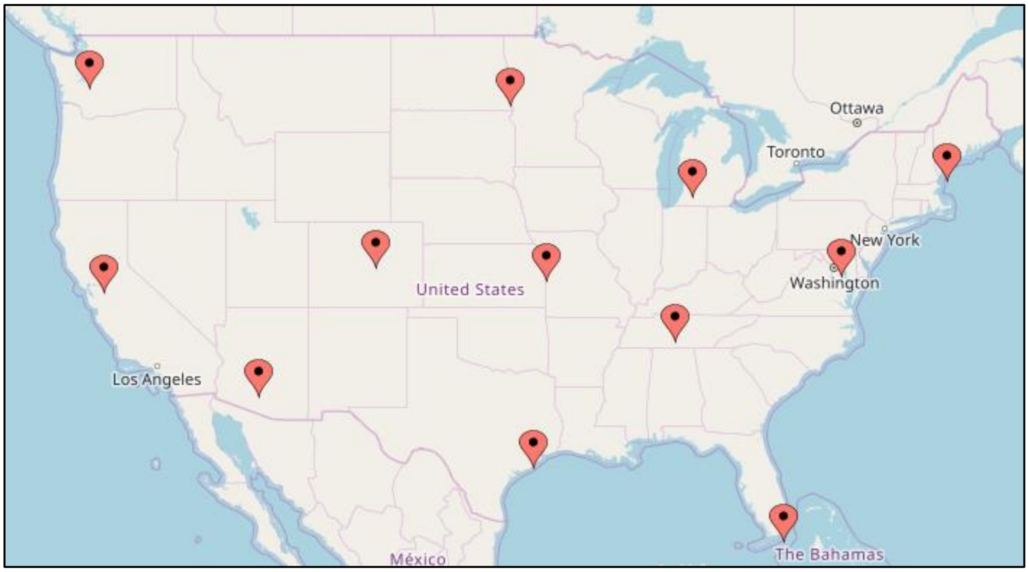


Figure 38: Map of 12 Host Cities, created using [32]

Table 3: List of Host Cities

No.	City	State
1	Baltimore	MD
2	Denver	CO
3	Fargo	ND
4	Grand Rapids	MI
5	Houston	TX
6	Kansas City	MO
7	Miami	FL
8	Nashville	TN
9	Phoenix	AZ
10	Portland	ME
11	Sacramento	CA
12	Seattle	WA

Next, water consumption values, energy consumption values, and live load heat generating items were researched and recorded. Questions were added to the Scilab question list if an item did not have the required units. For example, the reference for the washing machine's water consumption had units of gallons per load, so a question was added asking the user how many loads of laundry should be accounted for per week. It was then extrapolated for the month because the units needed to be gallons per month. A question was also added if an item was decided to be optional (e.g. accounting for a bathtub and shower instead of a shower only). This ensured that all items had the proper units for calculations. The items for water consumption had units of gallons per month and were: bathtub, cleaning, cooking, drinking, hand washing, hygiene (brushing teeth, shaving, etc.), shower, washing dishes, washing machine, water for pets, and watering plants. The items for energy consumption had units of kWh per month and were: cell phone(s), exhaust fans, laptop(s), lights, microwave, mini-split, refrigerator, and washing machine. The items that generated live heat had units of British Thermal Units per hour (BTU/hour) and were: cell phone(s), laptop(s), lights, HVAC system (mini-split), occupant(s), pet(s), and refrigerator. Each section had a small miscellaneous adder included to account for differences in lifestyles. A full list of items, associated values, and corresponding calculations can be found in Appendix IV: Monthly Water Consumption Calculations, Appendix V: Monthly Energy Consumption Calculations, and Appendix VI: Live Load Heat Calculations.

4.2. Configurable SolidWorks Model

Once all the data was collected, a configurable SolidWorks model was created. This model was a critical portion of the study as the remaining analyses relied on it. Each section of the tiny house (side walls, back wall, roof, structural supports, floor, etc.) was modeled as its own part that relied on design tables, and they all fed into a single assembly. A separate spreadsheet hosted constraint dimensions which the model could be updated from. Exaggerated examples are shown in Figure 39 to showcase how different dimensions and number of structural supports were manipulated. A 20 ft model and 24 ft model were created, and these sizes were chosen because they were the longest trailer sizes with two axles available from Tiny House

Basics, LLC, which results in needing a smaller towing vehicle [33]. A set of dimensions was saved for each.

For each section (e.g. above the wheel wells, from the wheel wells to the back wall, etc.), the structural supports were patterned based on equations so they would be equally spaced for the number of supports specified. While this is different compared to the construction of traditional houses (spacing supports a fixed distance, regardless of the number), this could reduce the weight of the structure.

There were only a few portions that were not configurable. First, the goose did not change because its dimensions remained fixed with different trailer lengths. Also, the location of the doors relative to the goose was fixed as the length and height of the steps remained fixed with different trailer lengths. Finally, the dimensions of the wheel wells were fixed as the chosen trailer lengths both utilized two axles, but the distance from the goose to the start of wheel wells changed with trailer length.

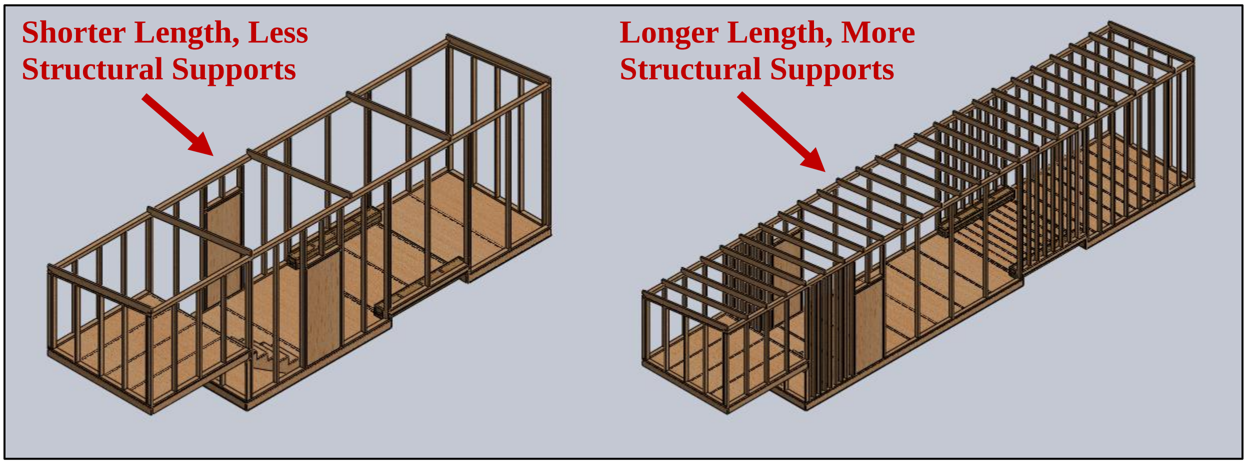


Figure 39: Example of Configurable SolidWorks Model

(Isometric View)

A gooseneck trailer was chosen for a few specific reasons. While it does cost more than a traditional trailer, it adds additional living space which can conveniently be used for a bed. This eliminates the need for a loft with steep stairs or a ladder, which may be difficult for older occupants to navigate. The interviews conducted by Kostoff [6] show that spatial restrictions are a common concern, and the gooseneck trailer helps alleviate this issue to some degree, compared to a traditional trailer. The gooseneck trailer also naturally lowers the COM, making the structure inherently more stable. Additionally, they cannot be transported with a standard vehicle hitch, so the risk of theft is reduced. This can make tiny houses less of novelty and a more realistic living solution for individuals looking for alternatives to traditional houses.

The tiny house was given a shed style roof, with the higher point of the roof on the driver side of the tiny house to help avoid lower hanging tree branches, power lines, etc. during transit. It was also designed with a slope of 7.5 degrees (°) (~1.56:12 pitch) so the average man of 69 inch (in) height (average woman is 63.6 in height) would not hit his head while standing upright next to the bed [34]. An example of the roof sloping from the higher driver side wall to the lower passenger side wall can be seen in Figure 40, and the vertical dimension next to the bed in the goose is shown in Figure 41. As it is a simpler design compared to a gable design with dormers, which is quite popular in the tiny house community, it is also easier to construct which shortens the build time.

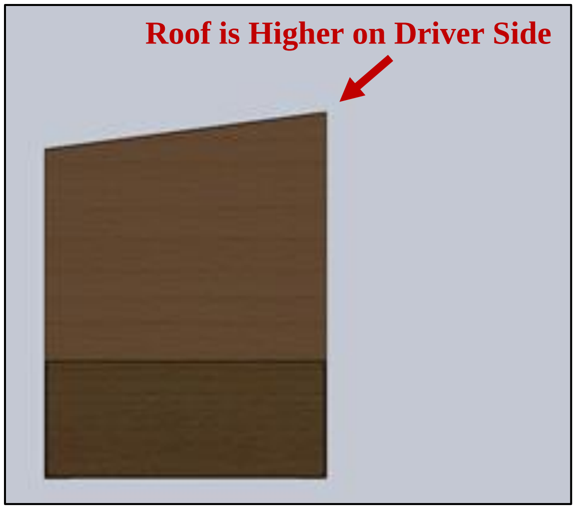


Figure 40: Example of Roof Slope

(Front View)

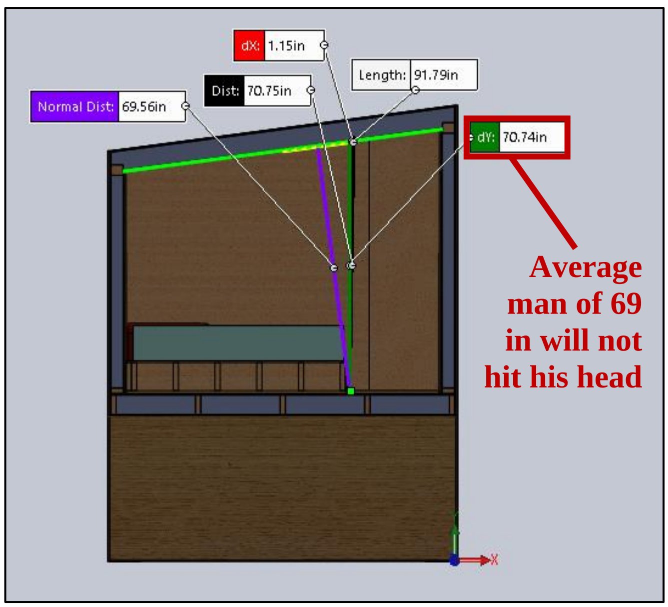


Figure 41: Vertical Dimensions Next to Bed in Goose

(Front View)

The stairs were designed to be ergonomic and comfortable, and each side can be made from a 2 x 12 [35]. The bottom of the stairs determined the location of the door, and a door was included on each side wall to allow the tiny house to adapt during different phases of the tiny homeowner's life.

It is common for a couple to live together by themselves, have children, and then live by themselves again once the children move out. If parents were to build a tiny house for each child, the double door design allows the first door to be the main entrance, and the second door would allow the additional tiny houses to be connected. When each child moves out, he or she can take the tiny house, and the parents do not have to worry about moving, selling and buying another house.

With information provided by Tiny House Basics, LLC the location of the wheel wells, the dimensions of the floor, and the height of the trailer (which determined the overall internal height) were set [36]. These dimensions allowed for the rest of the model to be designed, shown in Figure 42, and the final assembly consisted of the following parts:

- Floor Joists
- Side Wall Joists
- Roof Joists
- Goose
- Back Wall
- Side Wall Insulation
- Floor Insulation
- Roof Insulation
- Back Wall Insulation
- Goose Insulation
- Doors
- Goose Sheathing
- Back Wall Sheathing
- Roof Sheathing
- Left Side Wall Sheathing
- Right Side Wall Sheathing
- Floor Sheathing
- Wheel Well Sheathing
- Interior Ceiling
- Exterior Siding
- Roof
- Interior Side Wall Shiplap
- Floor Top
- Stairs
- Bedframe
- Bed Mattress
- Couch Frame
- Couch Cushion
- Bathtub
- Bathroom Floor Tile
- Toilet
- Oven
- Washing/Dryer Unit
- Bathroom Sink
- Kitchen Counter
- Kitchen Sink
- Bathroom Wall
- Kitchen Table
- Closet
- Refrigerator

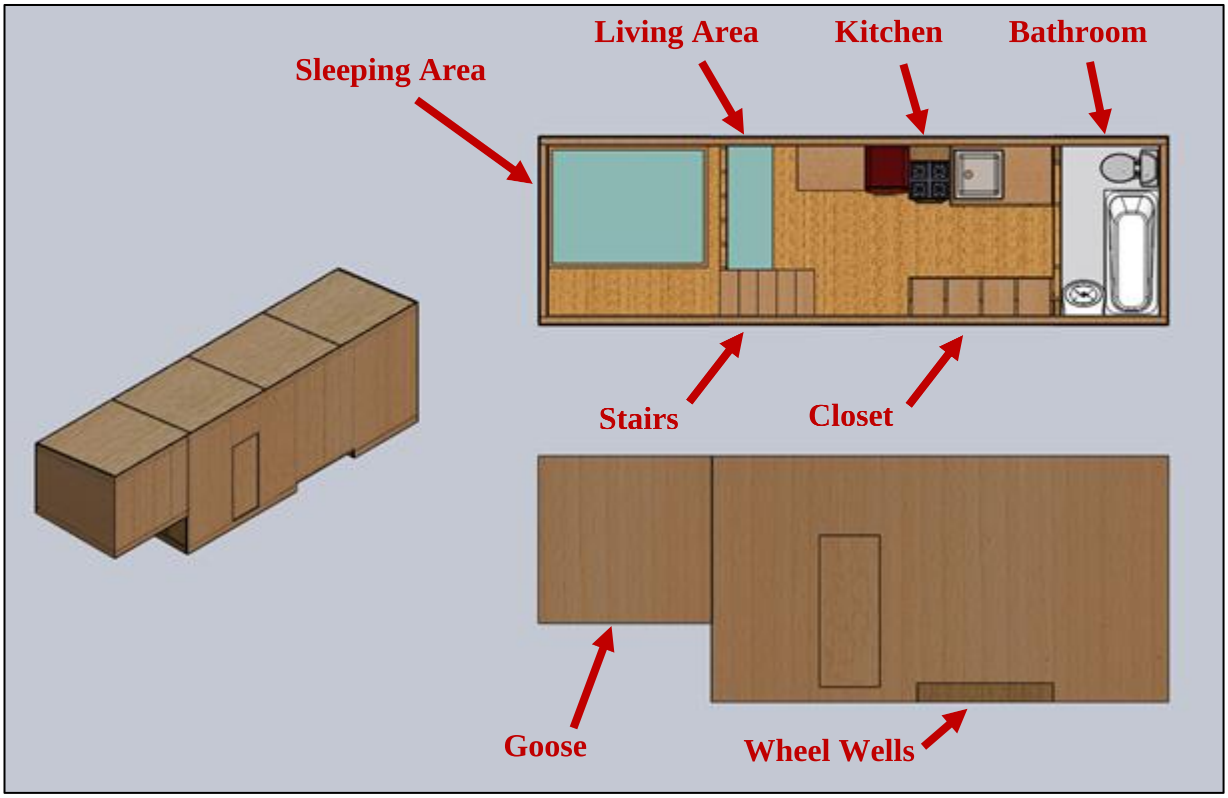


Figure 42: Example of Finished Tiny House Model

(Left: Exterior Isometric View, Upper Right: Interior Top View,

Lower Right: Exterior Driver Side View)

The internal dimension on the higher side wall was set so that the overall height of the tiny house was less than 130 in (Figure 43). Including the trailer deck height of 26 in [37], this means the overall height of the tiny house has approximately six in of buffer on the common U.S. height requirement of 162 in (13 ft 6 in) for vehicles and trailers [38]. The internal width was set so that the overall width of the structure would be less than 100 in (8 ft 4 in) when 2 x 4s, sheathing, and siding are accounted for (Figure 44), which is two in less than the maximum width requirement of 102 in (8 ft 6 in) for vehicles and trailers [38]. Floor joists and roof joists were designed using 2 x 6s for extra strength, rigidity and room for insulation space, and 2 x 4s were used for vertical supports. While 2 x 4s do not provide as much depth for insulation in the walls, they do increase the usable living space.

The U.S. Depart of Energy outlines recommends insulation values for different zones throughout the country, shown in Figure 45, and Zone 1 reflects regions that can use the lowest R-Value insulation, and Zone 7 reflects regions that should use the highest R-Value insulation (excluding Alaska) [39]. Only one type of denim insulation was used for the tiny house models, regardless of city, so the difference in R-value is accounted for later in the study with thermal simulations to know how much additional load the HVAC system will need to account for.

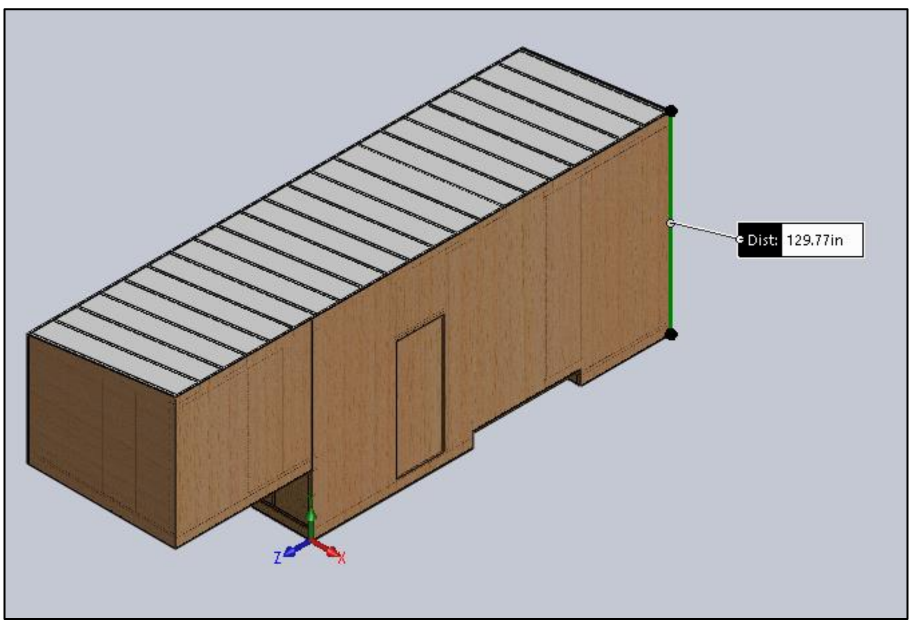


Figure 43: Overall Tiny House Height (Excluding Trailer)

(Isometric View)

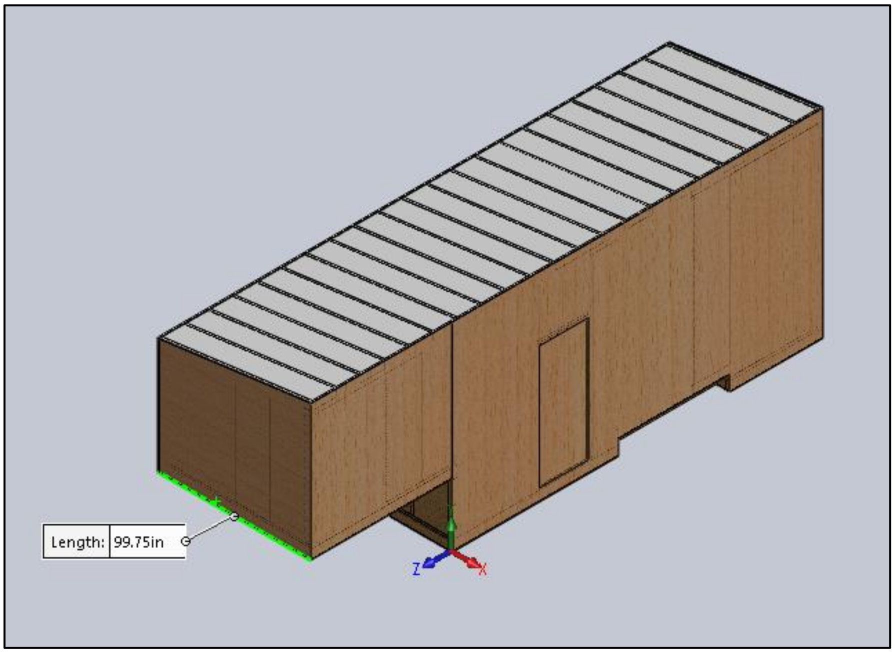


Figure 44: Overall Tiny House Width

(Isometric View)

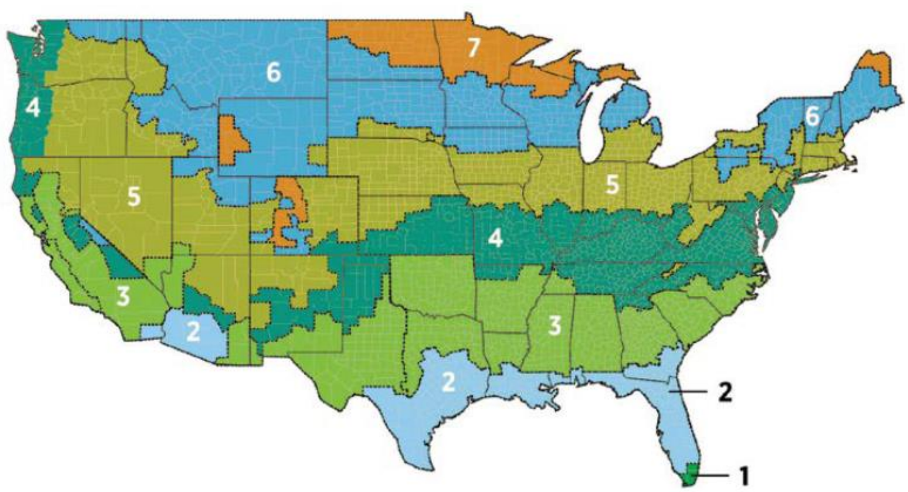


Figure 45: U.S. Insulation Zones [39]

While traditional houses generally utilize fiberglass insulation, denim insulation was chosen for the tiny house models. Denim insulation can be created from a recycled source, which aligns with the sustainability philosophy of many tiny house enthusiasts, despite it potentially being slightly more expensive [40]. It is safe to touch and does not contain chemicals, which makes it safer to install (i.e. does not require gloves to install), and it provides improved sound deadening properties [40]. While it can weigh more than traditional fiberglass insulation with the same thermal properties, the weight is accounted for later in the study with a stability and weight analysis.

Many different television shows, internet videos, and photo sharing websites showcase tiny houses being insulated with spray foam. This was considered, but spray foam insulation typically requires a professional to install. Open cell spray foam is not recommended for exterior walls, which is almost the entirety of the tiny houses in this study, and closed cell spray foam can develop cracks. As tiny houses are designed to be transported on the road, this increases the likelihood cracks would develop.

While a window is not shown in the generic model, it is included as an option in the Scilab design tool. The 20 ft model was reviewed and the smallest measured distance between vertical studs was recorded. This was used as a “catch-all” for window size. A 16 in width, 24 in height window would fit, so this window was selected as it could fit between any two vertical supports in the tiny house that were not located next to the doors [41]. While the window is small, it is somewhat proportional to the overall size of the tiny house, and the impact the single pane of glass has on the HVAC system is accounted for later in the thermal simulations. A more traditional double pane window was reviewed, but as it was over four times as large, it would be more difficult to freely locate within the tiny house structure, and the heat flux could potentially be larger. The user is able to specify the number of windows (one to six) that are desired, and they are accounted for in the design tool's calculations.

Regarding the few final design items, a door was omitted from the bathroom wall as some users may want a small swing door, and other may want a sliding barn style door. While up to the user, the gap between vertical studs was left at 31.5 in so either could be pursued. The bathroom stopped at the wheel well because it was a natural boundary. Shiplap type siding was accounted for on the interior instead of drywall (fire codes may need to be reviewed by the occupant) as the drywall may crack during transit (similar to the spray foam insulation). Last, the kitchen and closet were put over the wheel wells to use up the awkward space. The over-engineered closet helped balance out the appliances on the kitchen side, and the interior depth was set at 18 in as a coat hanger was measured at 16.5 in. While some may think the kitchen is too small, it is approximately the same size as the author's current kitchen, which has been successfully used for three years (Figure 46). The design of the bed frame allows for long storage drawers to be pulled out over the couch, which can be used for seasonal storage, and there is extra space behind the washer/dryer and stove/oven for utility connections.

Figure 46: Author's Current Kitchen

4.3. Structural Analysis

A structural analysis was needed because it would ensure the tiny house would be able to support its own weight, the weight of the items and belongings inside, as well as be able to withstand wind loads during transit and snow loads for colder regions. Only some of the parts were included in the analysis as others provided no structural support (e.g. insulation was omitted for this analysis but was included in overall weight and stability analysis). Each of the items of interest (items 1 through 12) listed below are identified in Figure 47 showing the number of floor joists, Figure 48 showing the vertical supports on the front and back walls, Figure 49 showing the vertical supports on the side walls, and Figure 50 showing the number of roof joists.

1. Number of floor joists between the goose and the wheel wells
2. Number of floor joists in the wheel wells
3. Number of floor joists between the wheel wells and trailer rear
4. Number of supports on the front wall of the goose
5. Number of supports on the side walls of the goose
6. Number of supports on the vertical wall at the end of the goose
7. Number of supports before the door
8. Number of supports after the door
9. Number of supports over the wheel wells
10. Number of supports between the wheel wells and back wall
11. Number of supports on the back wall
12. Number of roof rafters

Figure 47: Number of Floor Joists

(Bottom View)

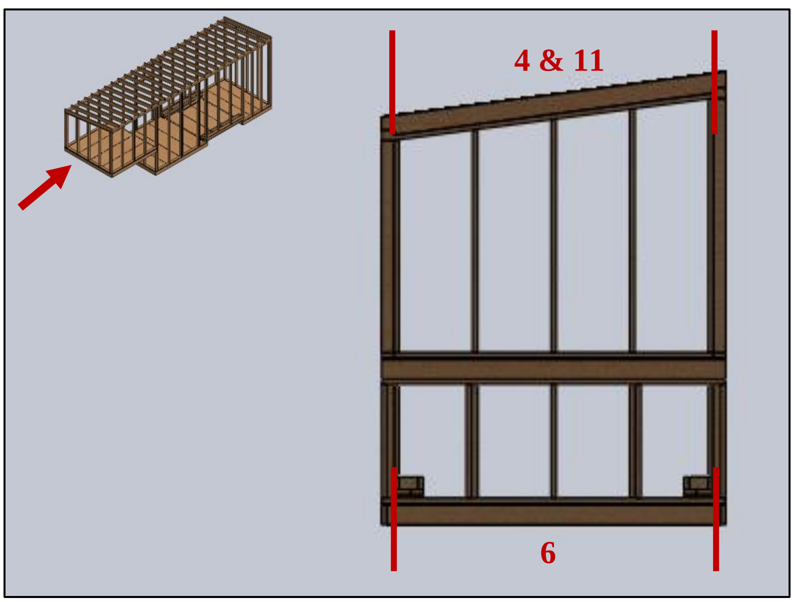


Figure 48: Number of Vertical Supports on Front Goose Wall and Back Wall

(Front View)

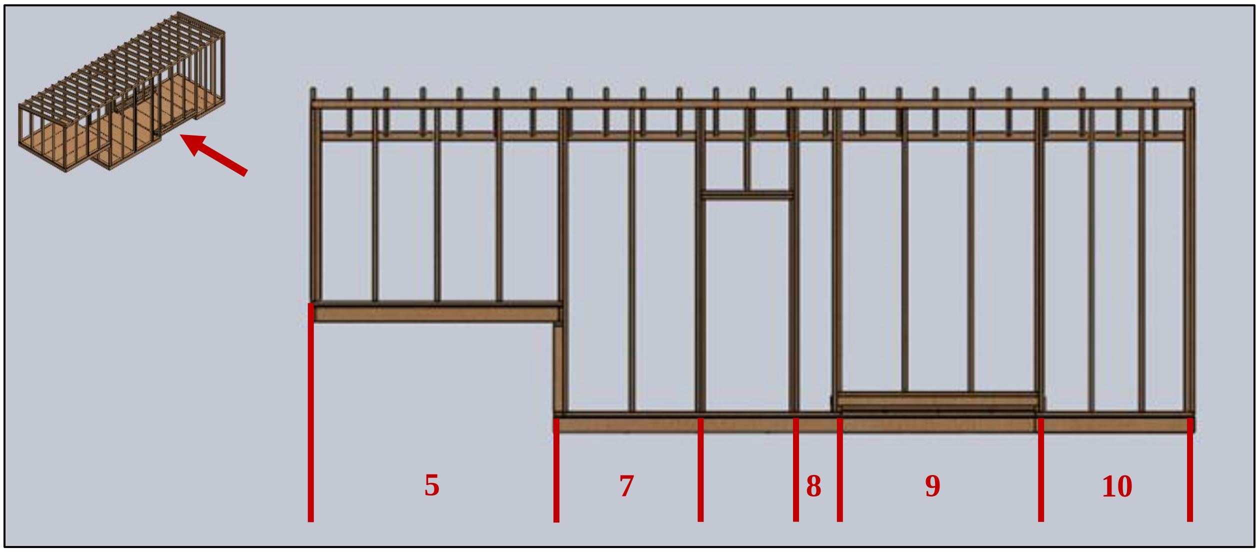


Figure 49: Number of Vertical Supports on Side Walls

(Driver Side View)

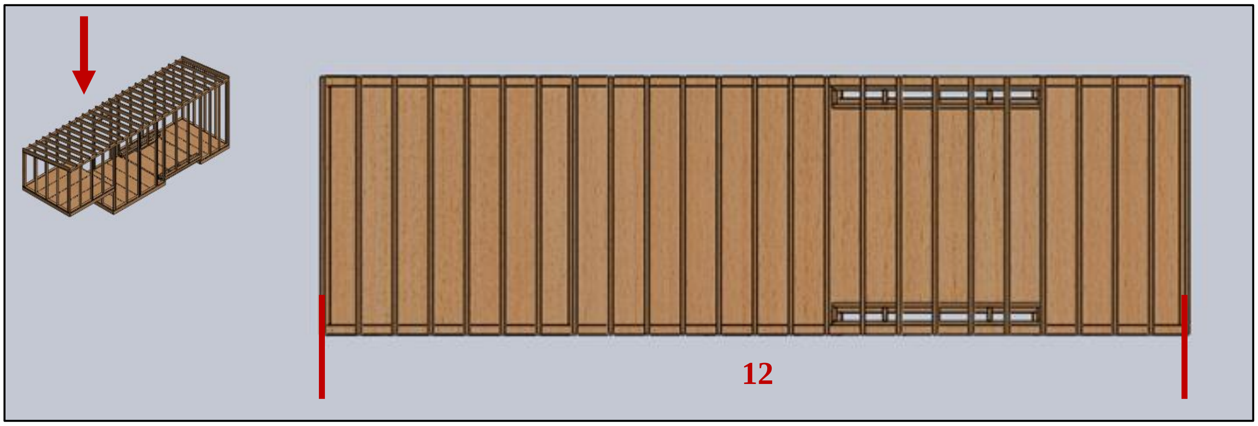


Figure 50: Number of Roof Joists

(Top View)

The floor and roof joints were 2 x 6s for additional rigidity, strength, and insulation potential, and the rest of the structural supports were 2 x 4s to maximize internal floor space. The parts were then assigned the material properties of either Douglas fir or plywood. The full list of materials used and their properties (some from the built-in SolidWorks database and others were custom) can be found in Appendix VII: Material Properties.

The overall approach was inspired by Calluari and Alonso-Marroquín [5], and some items were adapted to fit this analysis. Once all the materials and corresponding properties were assigned, SolidWorks Simulation was used for two structural simulations for each trailer size. The first simulated the tiny house being stationary and experiencing a worst-case snow load. The second simulated wind loads the tiny house would experience when pulled down the highway at 55 miles per hour (mph). All components were given a bonded global contact to prevent intersections and to simulate being nailed and screwed together. Adding individual nails and screws would have been too computationally rigorous to run, and an accurate simulation could not have been run with the provided educational SolidWorks license.

Continuing the setup of the structural FEA, different boundary conditions were applied to the model, which are visually represented in Figure 51 and Figure 52. As the tiny house would be built on a steel trailer, it was assumed that the bottom was rigid. This was simulated by fixing the exterior floors in space, which is represented with green markers. Next, gravity was turned on within the software, which is the large light blue arrow. The floor loading was calculated based on a 14,000 lb maximum capacity and the corresponding area of each trailer size (238 ft^2 for the 20 ft trailer and 272 ft^2 for the 24 ft trailer), which is represented with dark blue arrows. The exterior siding and interior shiplap were omitted in order to be able to reduce the number of mesh elements, and their weight was accounted for with a downward facing pressure boundary condition on the interior and exterior of the tiny house, represented with yellow arrows.

The final boundary condition for the stationary simulation, represented with red arrows in Figure 51, was the worst anticipated snow loading based on the maximum 1-month precipitation from Fargo, ND. While not all the tiny houses would be placed in an area that snows, this was done to account for the fact that the tiny house may be relocated to a colder region later in its life.

The final boundary conditions for the highway simulation were wind loads. A pressure boundary condition equivalent to 55 mph, shown with orange arrows in Figure 52, was applied to the front, side, and top faces of the tiny house. To simulate the worst-case scenario, a pressure boundary condition equivalent to a 28.1 mph cross wind (largest extreme 1% wind speed from the chosen host cities), shown with purple arrows, was applied to the larger side wall. The calculations for the individual structural boundary conditions can be found in Appendix VIII: Structural Analysis Boundary Conditions.

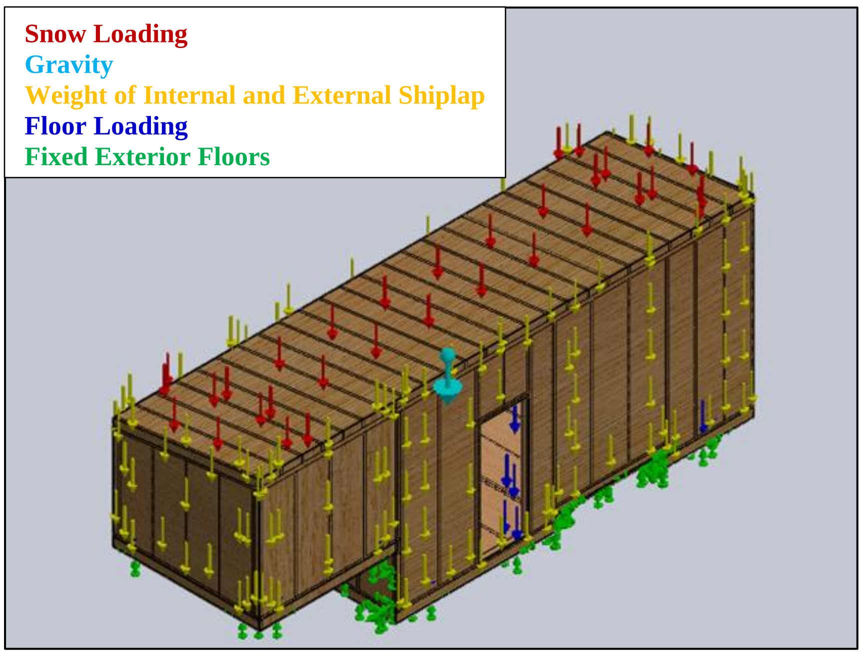


Figure 51: Visual Example of Stationary Boundary Conditions on the Tiny House (Isometric View)

Figure 52: Visual Example of Mobile Boundary Conditions on the Tiny House

(Isometric View)

Finally, the mesh was generated with the sliding scale built into SolidWorks (Figure 53). A few preliminary meshes were generated, but there were reoccurring errors even when the finest mesh setting was used. The mesh size ultimately had to be manually overridden as the smallest mesh element needed to be smaller than the thickness of the 3/8 in plywood that was in the model (Figure 54). There were only two bodies that had difficulties meshing after that, and individual mesh controls were applied so the entire model could mesh properly, as shown in Figure 55.

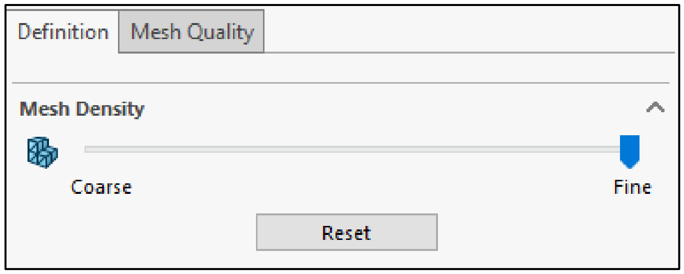


Figure 53: SolidWorks Meshing Scale

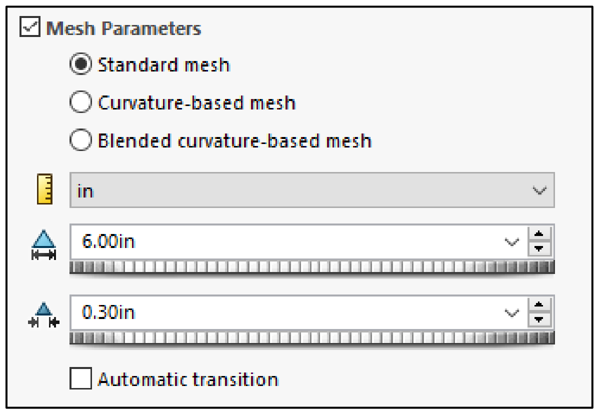


Figure 54: Manual Override of Structural Mesh Size

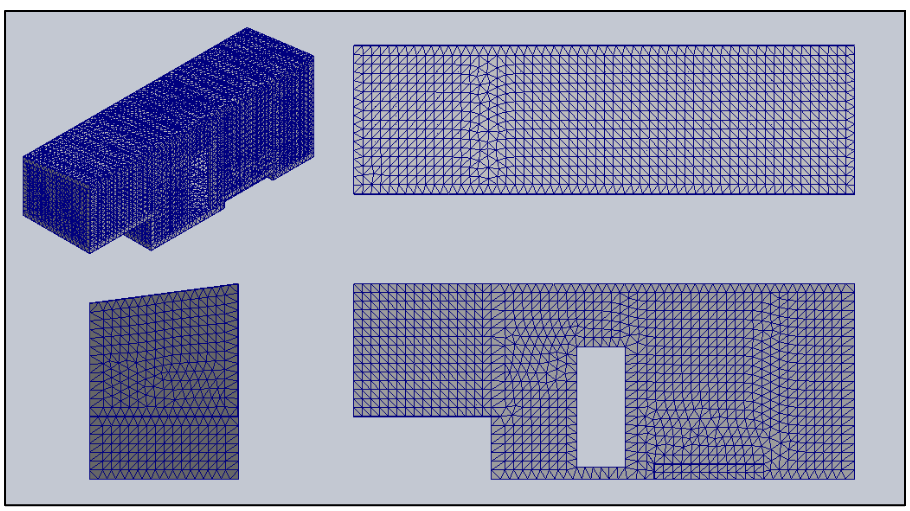


Figure 55: Fully Meshed Structural Model

(Upper Left: Isometric View, Upper Right: Top View,

Lower Left: Front View, Lower Right: Driver Side View)

The mesh model had a fairly uniform pattern of elements when the structure was consistent (i.e. areas of the geometry that were not transitioning around the wheel wells or goose), and the elements had an aspect ratio of approximately one, which is an indication of a suitable mesh. With a working mesh, the structural analyses were able to be run.

4.4. Stability Analysis

Even though the structural analyses confirmed the tiny house could support the simulated loads, a stability analysis needed to be conducted to make sure the tiny house would maneuver safely while being towed, as well as not tip when turning on a banked road. To investigate this, the COM needed to be located.

When the model was first created, the origin was set on the lower driver side wall where the goose starts (shown in Figure 56), and the COM is referenced against this point. As some parts were omitted from the structural analysis, the remaining items were added back in to give a more accurate weight distribution, as well as account for many household items that occupants may add (i.e. bed, refrigerator, sink, etc.), as shown in Figure 57. The full list of added items, their weights, and their dimensions can be found in Appendix IX: Weights and Dimensions of Household Items. To locate the COM, a built-in SolidWorks feature was enabled, and this information was used to complete the stability analysis and calculate the tongue weight percentage. The location of the COM is shown in Figure 58.

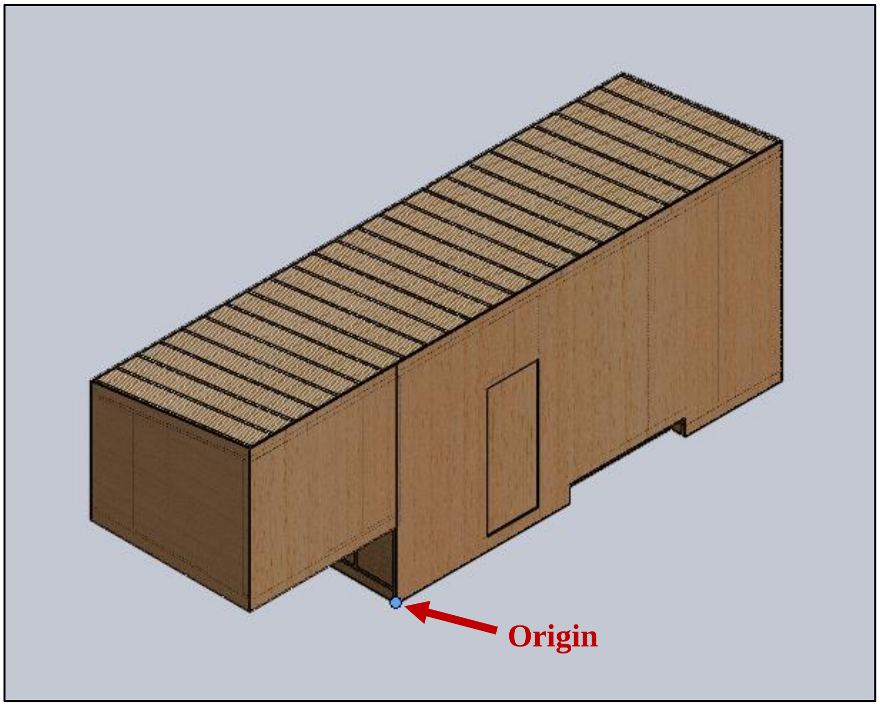


Figure 56: Location of Origin

(Isometric View)

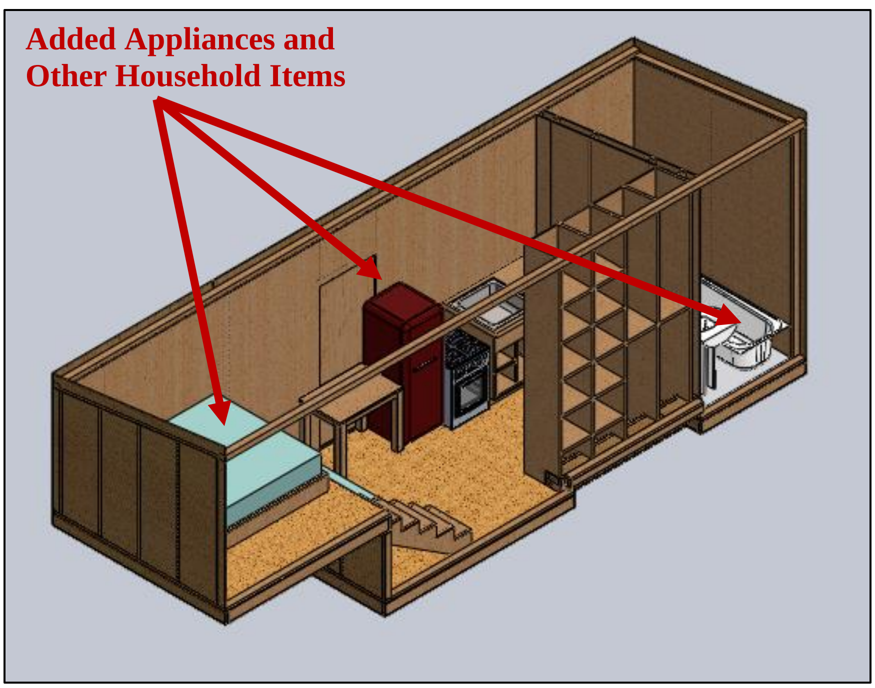


Figure 57: Completed Tiny House Interior

(Isometric View)

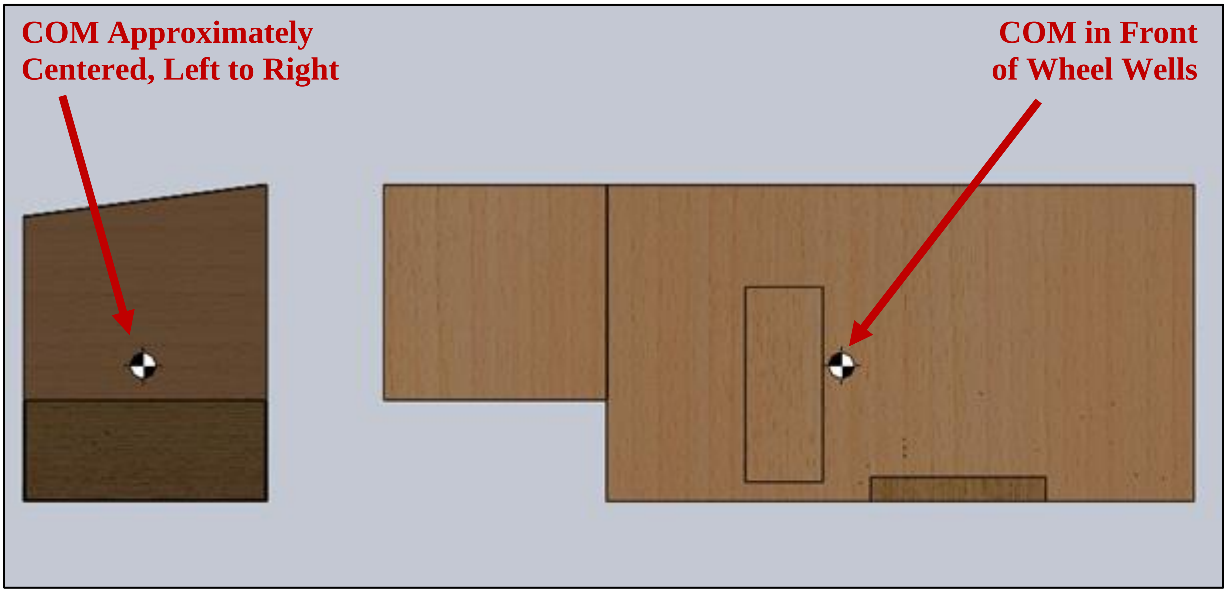


Figure 58: COM on Tiny House Model

(Left: Front View, Right: Driver Side View)

4.5. Thermal Analysis – Tiny House Structure

Once the structure was finalized and the stability was deemed suitable, SolidWorks Simulation was utilized again to perform thermal analyses to determine the heat flux through the structure at different external temperatures. This information would be used to help size the HVAC system to ensure occupants would stay warm in the winter and cool in the summer, regardless of selected host city.

The model resulting from the structural analyses and stability analysis was used, and denim insulation parts were placed in the wall cavities to reduce the heat flux through the structure. Again, a full list of materials and their properties can be found in Appendix VII: Material Properties. The same bonded global contact setting that was used in the structural analyses was used here.

For each of the 12 cities and for each of the two trailer sizes, a simulation was run with the 99% heating dry bulb temperature as the boundary condition for the external air temperature, and another simulation was run with the 1% cooling dry bulb temperature as the boundary condition for the external air temperature. This boundary condition is shown with green markers in Figure 59 and Figure 60. Again, the full list of cities and their corresponding temperatures can be found in Appendix III: Regional Weather Data. In each of the 48 simulations, a 73°F boundary condition was set for the internal air temperature, which is shown with orange markers in Figure 60, which is showing a cross section so the markers would be more visible. The calculations for the individual thermal boundary conditions can be found in Appendix X: Thermal Analysis Boundary Conditions.

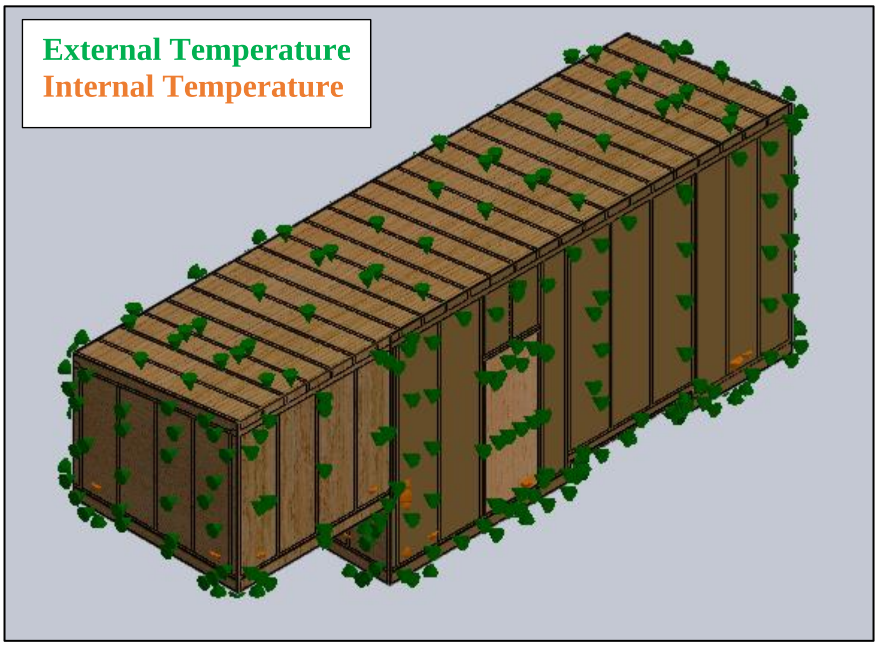


Figure 59: Visual Example of External Boundary Condition on the Tiny House (Isometric View)

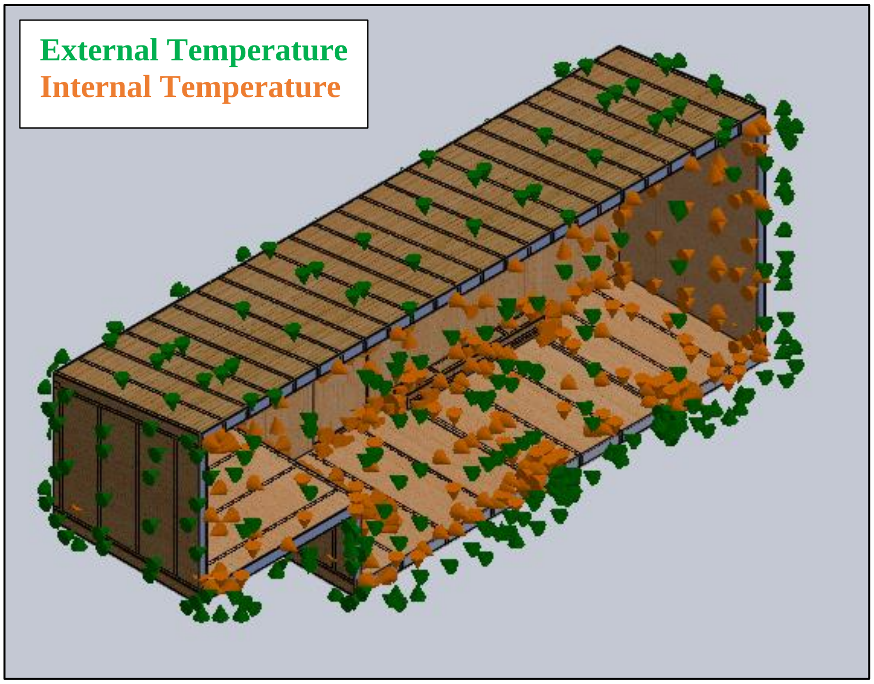


Figure 60: Visual Example of External and Internal Boundary Conditions on the Tiny House (Isometric View)

Finally, the mesh was generated with the same sliding scale as the structural analyses. The mesh size had to be manually overridden again (Figure 61) as the smallest mesh element needed to be smaller than the thickness of the smallest piece of insulation, which was located in the wheel wells. This time there were three bodies that had difficulties meshing, and individual mesh controls were applied so the entire model could mesh properly (Figure 62).

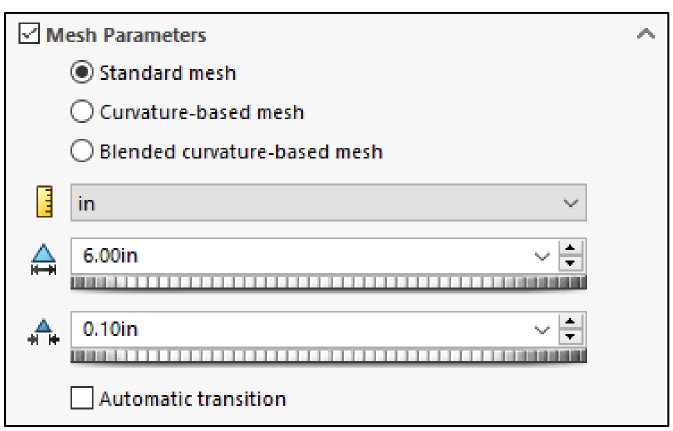


Figure 61: Manual Override of Thermal Mesh Size

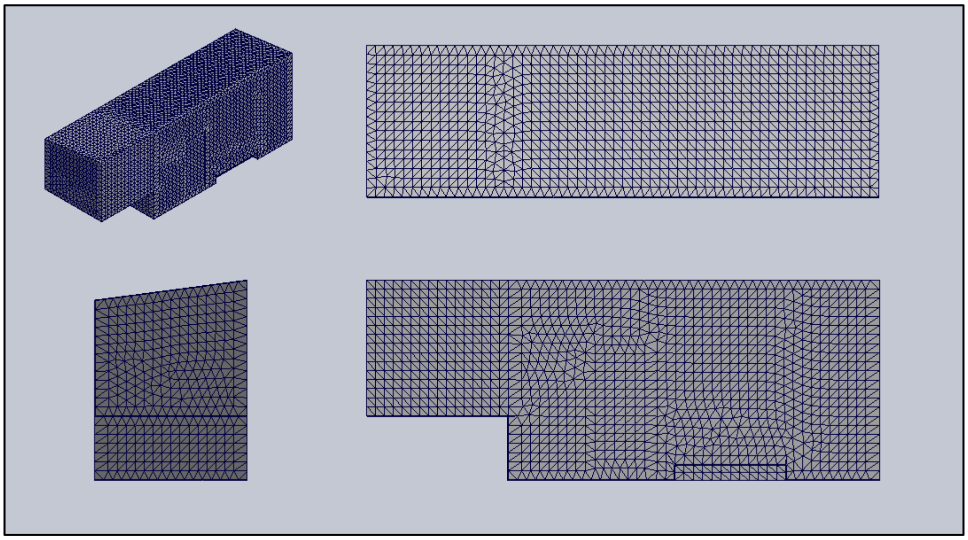


Figure 62: Fully Meshed Thermal Model

(Upper Left: Isometric View, Upper Right: Top View,

Lower Left: Front View, Lower Right: Driver Side View)

Resembling the structural mesh, the thermal mesh model had a fairly uniform pattern of elements when the structure was consistent (i.e. areas of the geometry that were not transitioning around the wheel wells or goose), and the elements had an aspect ratio of approximately one, which is an indication of a suitable mesh. With a working mesh, the thermal analyses were able to be run to determine the heat flux through the structure.

4.6. Thermal Analysis – Window

The previous section specifically omitted windows because it was unknown how many windows each user would want. To account for windows in the HVAC calculations, a window simulation was run for each city at each 99% heating dry bulb temperature and 1% cooling dry bulb temperature. After the user specifies how many windows are desired, the resulting heat flux would be added (or subtracted) with the other heat generating items, shown in Appendix VI: Live Load Heat Calculations.

The window was modeled as a 3/32 in thick rectangle with the material property set to glass (Figure 63). The same internal and external temperature boundary conditions and color markers were used as in the previous section, as shown in Figure 64. As the geometry of this part was significantly simpler and it did have multiple parts next to one another (i.e. insulation directly next to wood supports), the mesh was kept at the default setting which was halfway between coarse and fine. There were no meshing errors, the elements were uniform, and they had an aspect ratio of approximately one (Figure 65). All the window simulations were run afterwards.

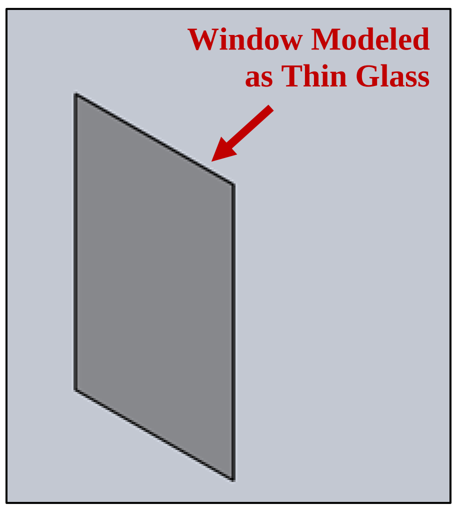


Figure 63: Example of Window Model

(Isometric View)

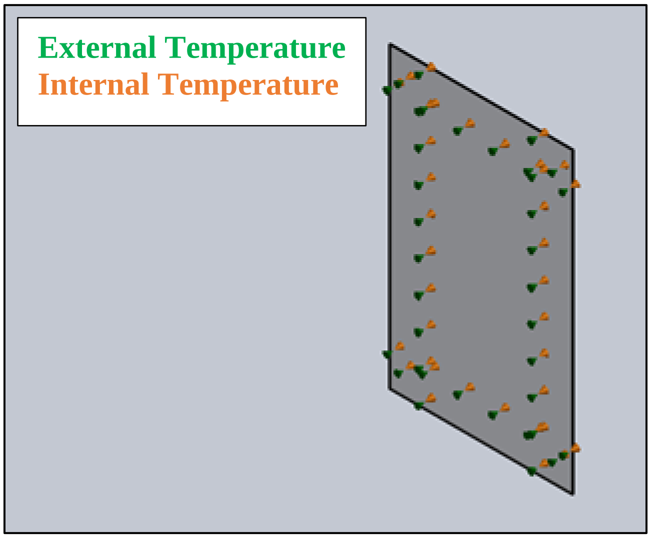


Figure 64: Visual Representation of External and Internal Boundary Conditions on Window

(Isometric View)

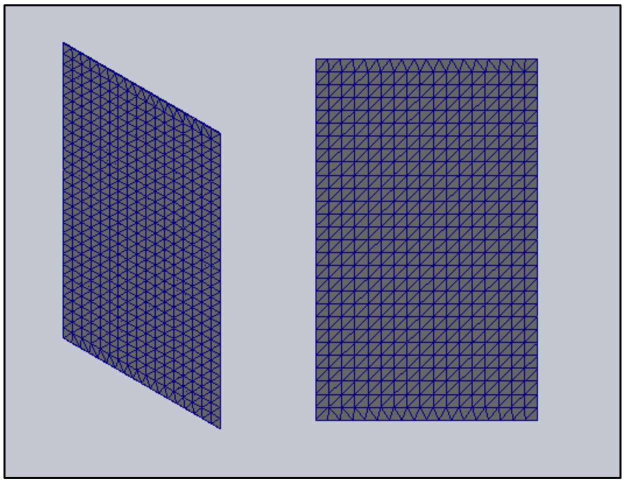


Figure 65: Fully Meshed Thermal Window Model

(Left: Isometric View, Right: Front View)

4.7. BOMs, Projected Cost, Drawings, and Scilab Design Tool

Once all the data was collected and recorded, the final items could be completed. The BOMs were created by summing the matching parts together (e.g. added together the lengths of all 2 x 4s) for the entire assembly. All unique parts were kept separate, and once the full list was generated, components were costed online. Drawings were then generated for each part in the 20 ft and 24 ft assemblies. All the information was summarized into tables and coded into the Scilab design tool that the user can run. Scilab was chosen as the software for the design tool because it is an open source program. This means the user can download and run the software for free and use it without a subscription or license, which allows more people to have access to the design tool.

5. Results and Discussion

The results from the different analyses determined many design parameters for the tiny houses. The structural analysis results stipulated the number of supports and joists for each of the two trailer length models. The stability analysis results showed the towing vehicle would have the proper tongue weight and that the tiny house would not tip when turning on a banked road. The thermal analysis results provided the heat transfer quantities for each city and design temperature, and all this information was summarized into BOMs, a projected cost, construction drawings, and the final Scilab design tool. Note that the figures below only refer to the 20 ft model, but all tasks were duplicated for the 24 ft model in order to obtain full results.

5.1. Structural Analysis Results

Once a structural simulation was finished running the von Mises stress (referred to as stress for the remainder of section 5) and resulting deflection were reviewed. These parameters were essential to a safe design because if the stress or deflection was too high, the structure could fail. The structure was deemed safe if the maximum stress at any location was less than 190 psi and the maximum deflection of any component was less than 0.09375 in (3/32 in).

The maximum stress of 190 psi represented a safety factor of two, given the yield strength of green Douglas fir (perpendicular to the grain direction) is 380 psi [42]. The 190 psi value was used as a universal limit, even though not all loads were perpendicular to the grain direction, which simplified the analysis. While dry Douglas fir has a higher yield strength, some drier areas of the United States use green lumber, and the design needed to be applicable for any city in Table 3 [43]. Green lumber has a higher moisture content than dry lumber, and some areas build with it as it is less expensive, easier to cut, and is less likely to split [44]. Green lumber shrinks as it dries, and this must be taken into account if building with it.

The maximum allowable deflection for floor supports, according to the Residential Structural Design Guide [45], is the length (units of in) of the support divided by 360. The roof joists were not considered to determine deflection because the floor joists are shorter and would results in a lower acceptable deflection value. The shortest non-perimeter floor support was 72.75 in, which gives an acceptable deflection of 0.202 in. Even though a boundary condition was that the floors were fixed in space, a non-perimeter component was selected because it is unknown where supports will be located on the physical trailer. This 0.202 in value can be rounded down to 0.1875 in for a cleaner construction fraction (3/16 in), and the maximum deflection value of 0.09375 in (3/32 in) was obtained by incorporating a safety factor of two. The 0.09375 in value was used as a universal limit, which simplified the analysis.

The first structural simulation was initially run with arbitrary values for the number of different structural elements, and then stress and deflection overlays were generated. Examples of these are shown in Figure 66 and Figure 67. Note that red on the color scale reflects the maximum value for the overlay and dark blue reflects the minimum value for the overlay. The colors do not necessarily mean the design passed or failed as the numeric values that corresponds to the colors also needed to be examined.

In Figure 66, the maximum stress is shown as 3.373e+02 psi (373.3 psi), which is higher than the 190 psi limit, so additional structural supports needed to be added. In Figure 67, the maximum deflection is shown as 4.994e-02 in (0.04994 in), which is less than the 0.09375 in limit, but it still provided insights for where additional supports should be added (higher deflection potentially implies higher stress). From Figure 66, the inside of the model cannot be shown, but the areas of highest stress needed to be found. To do this, the iso clipping tool within SolidWorks was used. This tool has a sliding scale, and only the results for a single color are shown (Figure 68). As the rest of the model was transparent, areas of higher stress could be isolated by increasing the sliding scale value to the 190 psi limit (Figure 69).

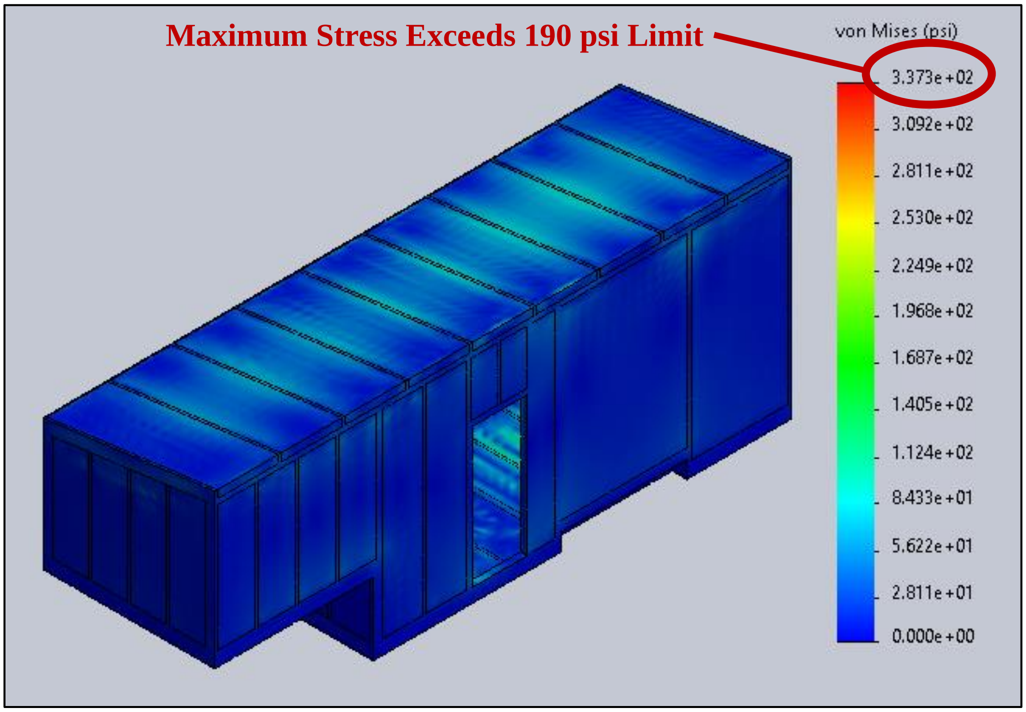


Figure 66: Example of Preliminary Stress Results

(Isometric View)

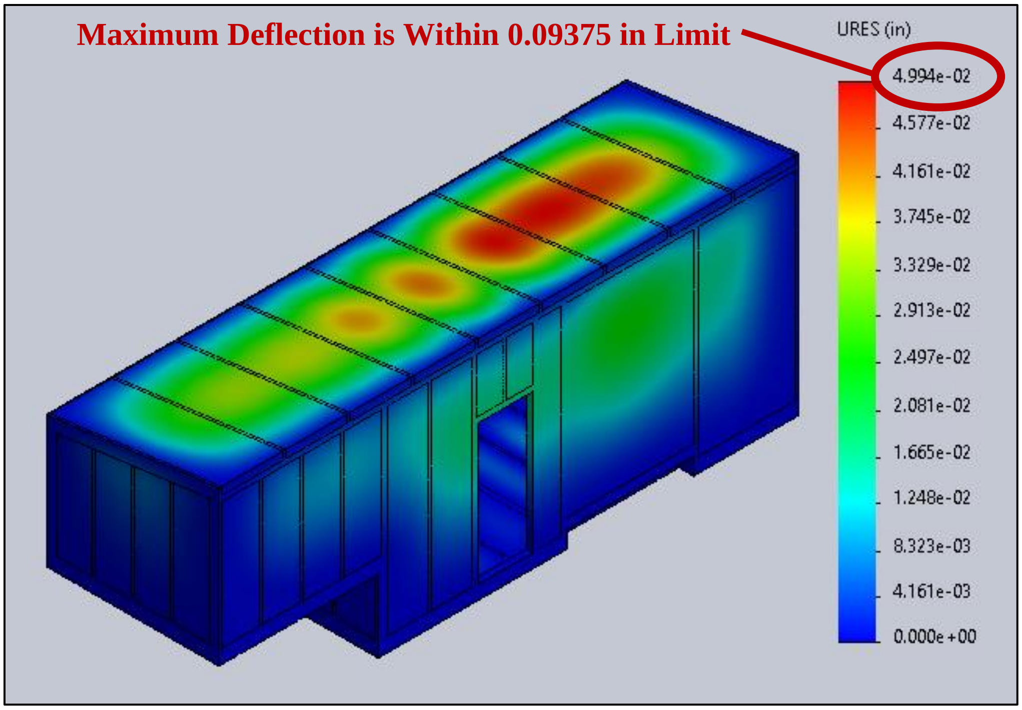


Figure 67: Example of Preliminary Deflection Results

(Isometric View)

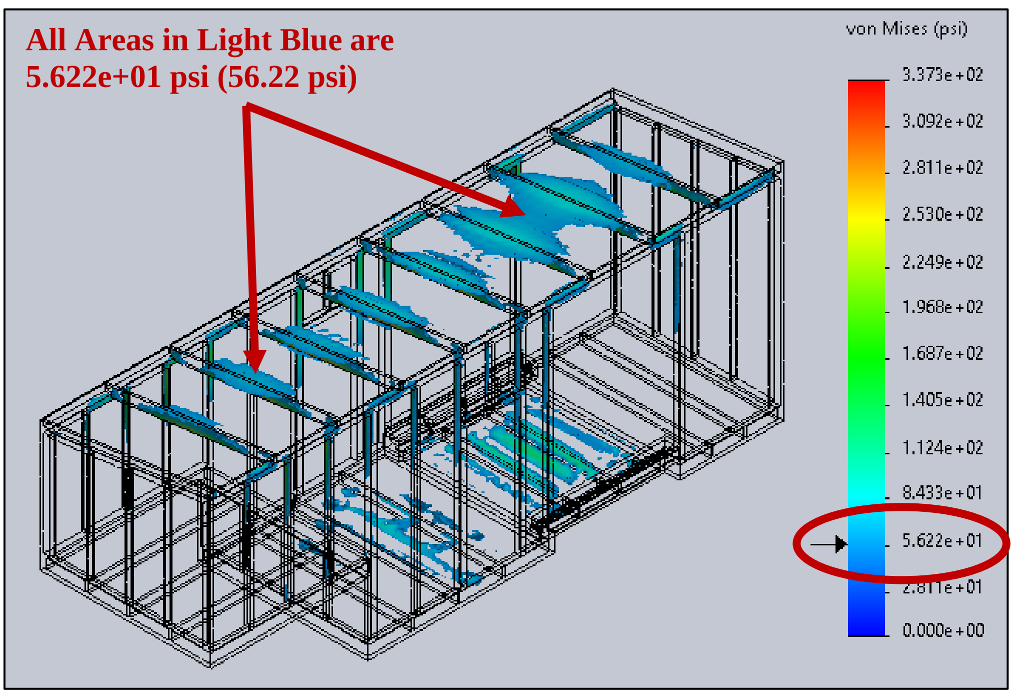


Figure 68: Example of Iso Clipping Tool (below 190 psi limit)

(Isometric View)

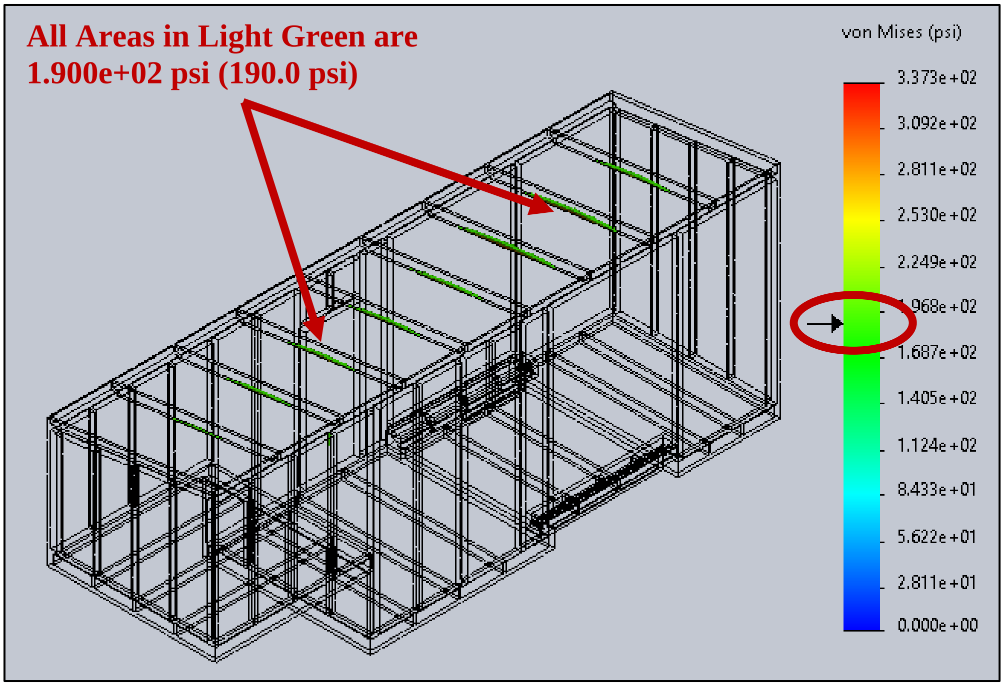


Figure 69: Example of Iso Clipping Tool (at 190 psi limit)

(Isometric View)

Using the iso clipping tool and standard overlays, it was determined that the areas of highest stress were on the undersides of the roof rafters. Additional structural supports were then placed in strategic locations, the model was updated, and another simulation was run. This process was repeated for stationary and mobile boundary conditions on both models (nine rounds of simulations for the 20 ft model and two rounds of simulations for the 24 ft model) until the maximum stress was as close to 190 psi without exceeding it (the maximum deflection was not a limiting factor) as to reduce the weight of structure as much as possible. While the bottom of the roof rafters still had the highest stress values, they were below the 190 psi requirement. Visual results for the 20 ft model are shown in Figure 70, Figure 71, Figure 72, and Figure 73. Numeric stress and deflection results for the 20 ft model and 24 ft model are shown in Table 4, and the final number of structural supports for each model is shown in Table 5.

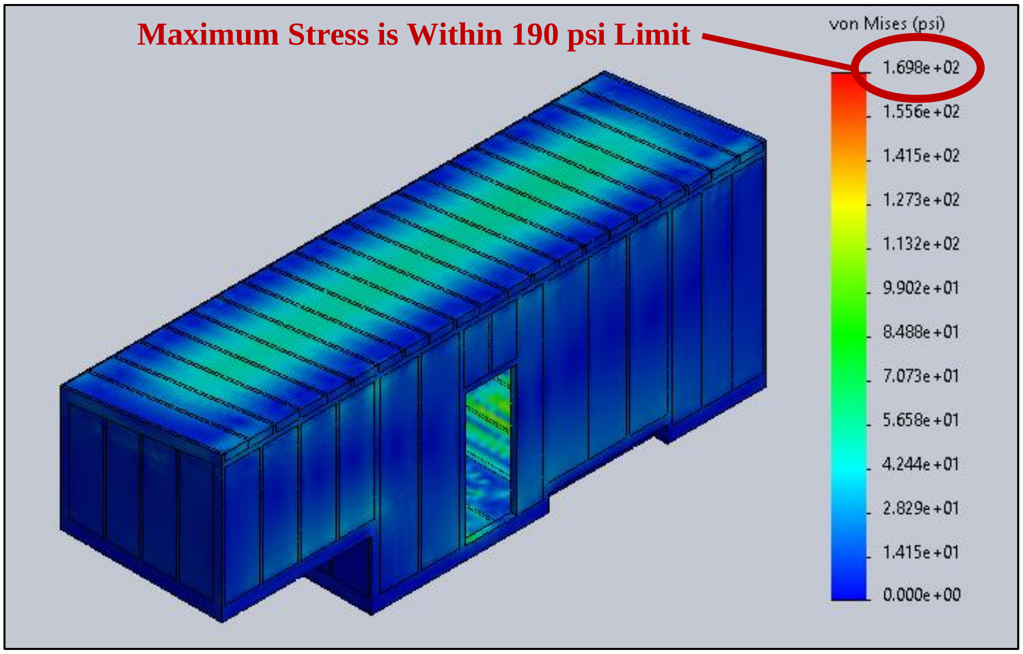


Figure 70: Final Stress Results of 20 ft Model, Stationary Boundary Conditions (Isometric View)

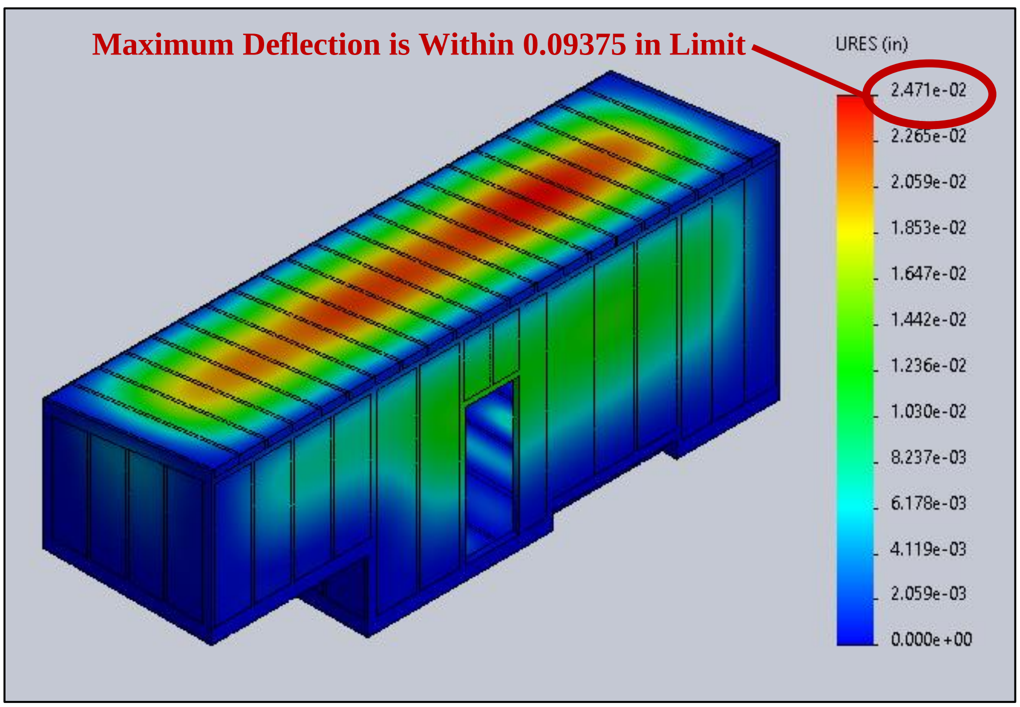


Figure 71: Final Deflection Results of 20 ft Model, Stationary Boundary Conditions (Isometric View)

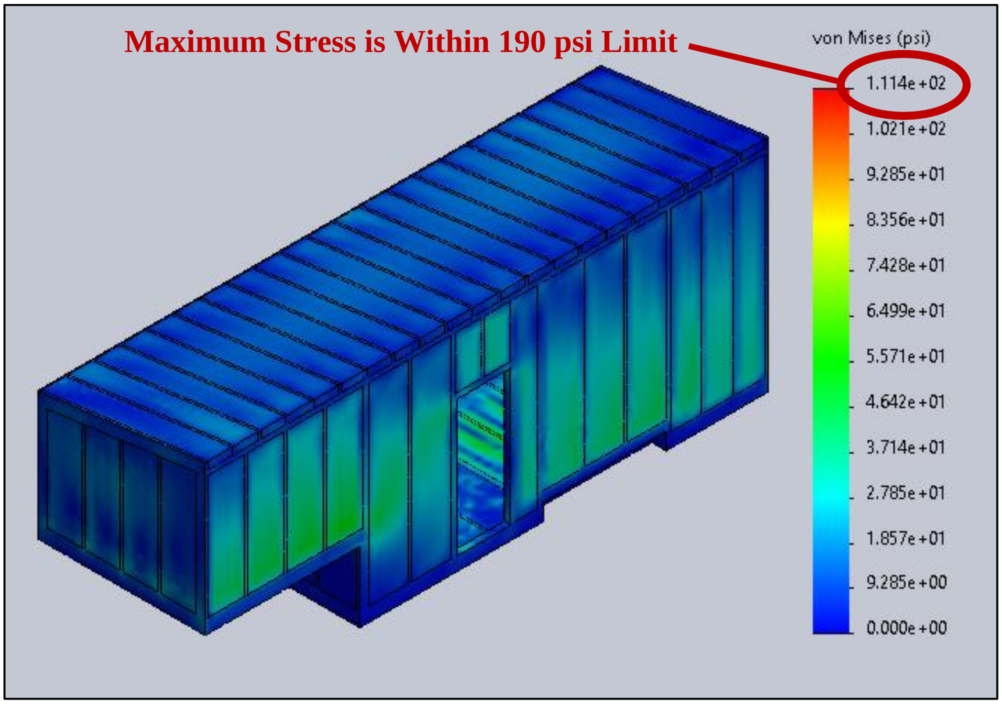


Figure 72: Final Stress Results of 20 ft Model, Mobile Boundary Conditions (Isometric View)

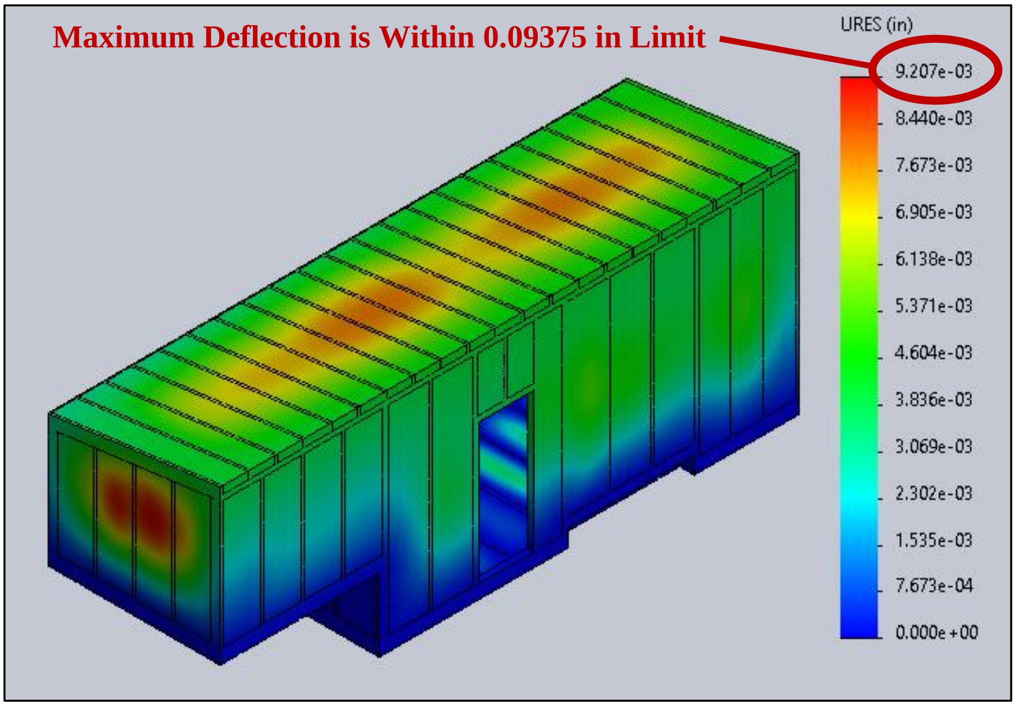


Figure 73: Final Deflection Results of 20 ft Model, Mobile Boundary Conditions (Isometric View)

Table 4: Stress and Displacement Values Structural Analysis

	20 ft Model: Maximum Values	24 ft Model: Maximum Values	Allowable Values	20 ft Model: Safety Factors	24 ft Model: Safety Factors
Stress — Stationary [psi]	169.7	171.4	190	2.24	2.22
Stress — Mobile [psi]	114.1	112.2	190	3.33	3.39
Displacement — Stationary [in]	0.0247	0.0255	0.09375	7.59	7.35
Displacement — Mobile [in]	0.0092	0.0093	0.09375	20.38	20.16

Table 5: Final Number of Structural Supports for Each Tiny House Model

Item Number	Item Description	Amount for 20 ft Model	Amount for 24 ft Model
1	No. of floor joists between the goose and the wheel wells	5	6
2	No. of floor joists in the wheel wells	4	4
3	No. of floor joists between the wheel wells and trailer rear	4	4
4	No. of supports on the front wall of the goose	5	5
5	No. of supports on the side walls of the goose	5	5
6	No. of supports on the vertical wall at the end of the goose	5	5
7	No. of supports before the door	3	3
8	No. of supports after the door	2	3
9	No. of supports over the wheel wells	4	4
10	No. of supports between the wheel wells and back wall	4	4
11	No. of supports on the back wall	5	5
12	No. of roof rafters	22	24

5.2. Stability Analysis and Weight Results

With sound structures, stability needed to be examined next. To determine how stable the model would be, a few additional measurements were added to the SolidWorks model. First, a 26 in dimension, which was specified by Tiny House Basics, LLC, was added below the model to represent where the wheels would contact the road (Figure 74) [37]. Then another line was added which connected the wheel-road contact point to the COM. The angle between these two lines specifies the angle at which the tiny house would tip over. As the tiny house was asymmetric, this was repeated for the other side, also shown in Figure 74. As the two angles are very similar (31.89° for the driver side and 31.19° for the passenger side on the 20 ft model), this indicated that the COM was also nearly centered left-to-right on the model (Figure 75). Note that the tiny house does not include the trailer, so that will bring the COM down further, which will increase the tip angles from what is shown. This process was repeated for the 24 ft model, and considering the "recognized maximum value" for superelevation for roads in the United States is 12% (~6.84°), both tiny house models can be transported safely on the road (either being towed at a safe speed for the prescribed turn, or stationary if there is standstill traffic) [46]. The tip angle results for both models are shown in Table 6.

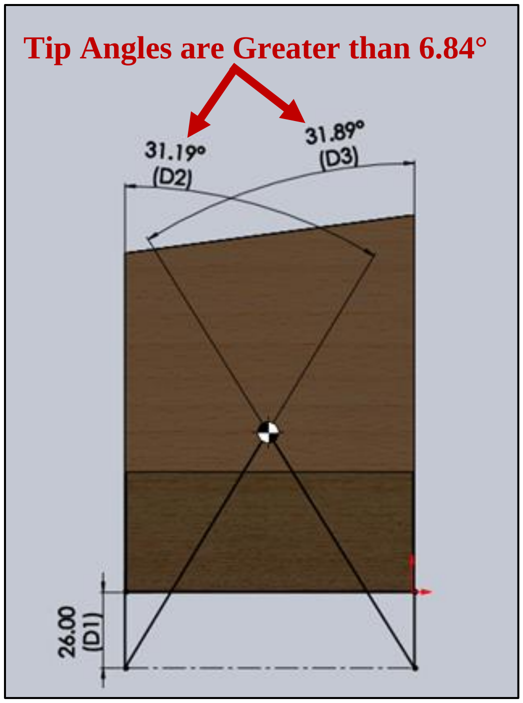


Figure 74: Tip Angles for Loaded 20 ft Model

(Front View)

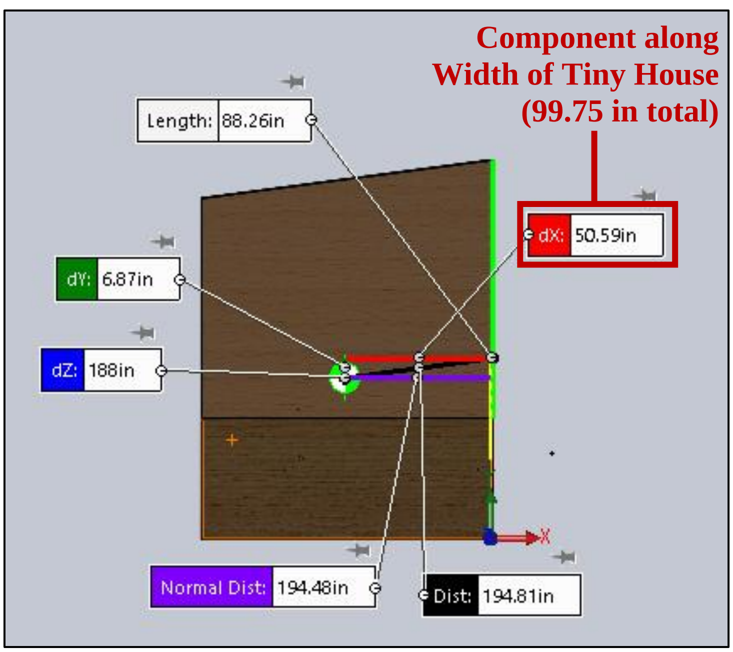


Figure 75: Distance from the Driver Side Wall to the COM

(Front View)

Table 6: Summary of Tip Angles for Loaded Models

	20 ft Model	24 ft Model
Tip Angle — Driver Side [°]	31.89	31.94
Tip Angle — Passenger Side [°]	31.19	31.40
COM distance from center plane [in]	0.719	0.565
COM distance from center plane [%]	1.43	1.13

To determine the tongue weight, the free body diagram, shown in Figure 76, was solved. Note that the arrow lengths are arbitrary and are not indicative of actual values. The forces in the vertical direction were summed and set equal to zero (the tiny house is static), and the moments about the COM were summed and set equal to zero (again, the tiny house is static) which produced two equations. Considering the hitch connection can be aligned with the front of the trailer, the front face was used for that measurement [33]. These equations were then manipulated into a system of equations (Eq. 23 to Eq. 26) and solved to determine the tongue weight.

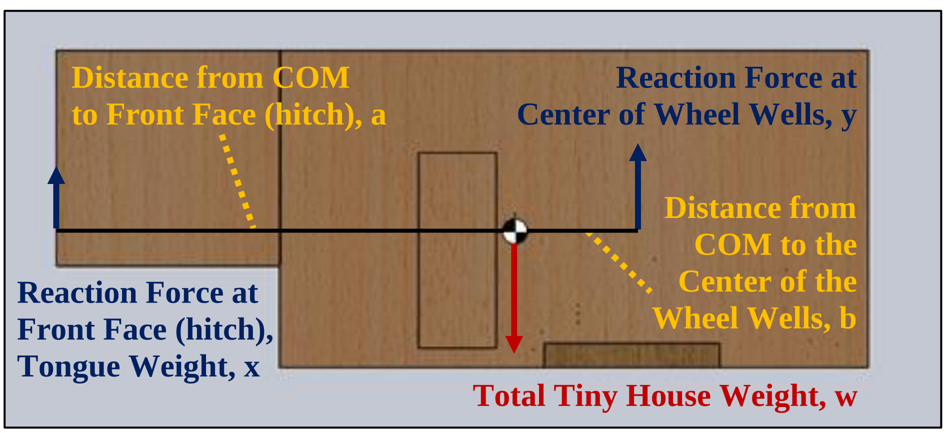


Figure 76: Free Body Diagram for COM and Tongue Weight

(Driver Side View)

$$x + y - w = 0 \quad \textit{Eq. 23}$$

$$x \times a - y \times b = 0 \quad \textit{Eq. 24}$$

$$\begin{bmatrix} 1 & 1 \\ a & -b \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} w \\ 0 \end{bmatrix} \quad \textit{Eq. 25}$$

$$x = \frac{b \times w}{a + b} \quad \textit{Eq. 26}$$

All the needed variables were gathered from the SolidWorks model. The overall weight of the tiny house was displayed using the mass properties function (Figure 77), the distance from the center of the wheel wells to the COM was measured using an evaluation function (Figure 78), and the distance from the front of the tiny house to the COM was also measured using an evaluation function (Figure 79). These values were input into Eq. 26 which yielded a tongue weight of 1,561.85 lb, which is 20.3% of the total tiny house weight. It is recommended to have the tongue weight be 15% to 30% of the total trailer weight for gooseneck style trailers, and the value is within that window [47]. While the model did not fully account for personal possessions or the trailer itself, knowing what the tongue weight percentage is helps the user plan for a safe transit. This process was repeated for the 24 ft model, and the results are shown in Table 7.

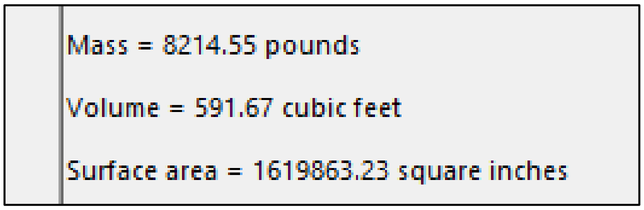


Figure 77: Mass Properties of the Loaded 20 ft Model

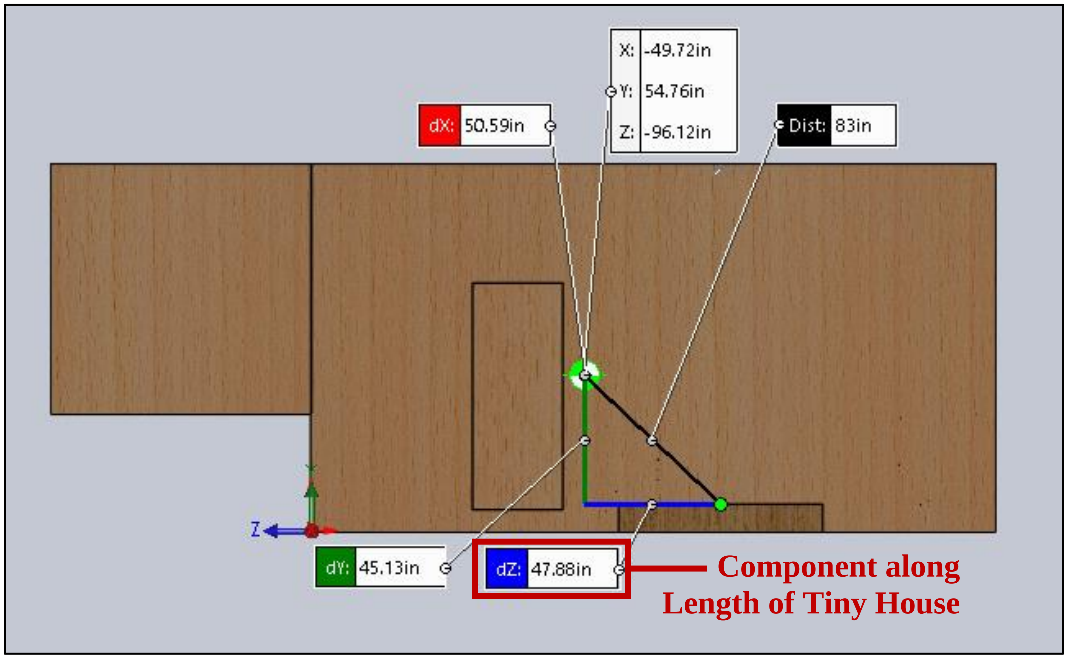


Figure 78: Distance from the Center of the Wheel Well to the COM

(Driver Side View)

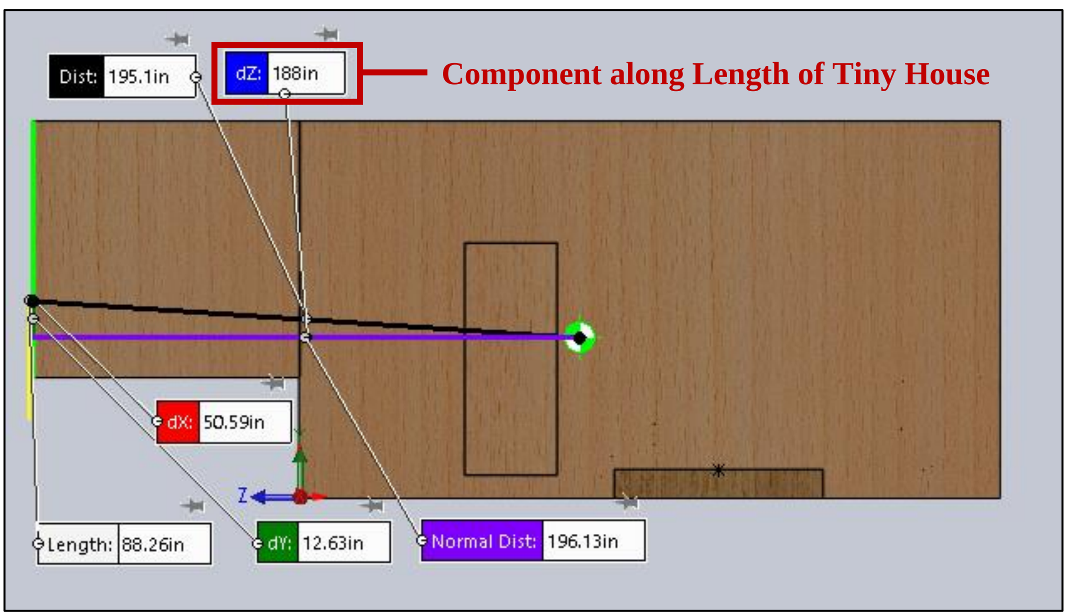


Figure 79: Distance from the Front of the Tiny House to the COM

(Driver Side View)

Table 7: Summary of Loaded Tiny House Tongue Weight Percentages

	20 ft Model	24 ft Model
Distance from the Front Face to COM [in]	188.00	212.25
Distance from COM to Center of the Wheel Wells [in]	47.88	52.42
Total Weight [lb]	8,214.55	8,971.17
Tongue Weight Percentage [%]	20.3%	19.8%

Realizing that some users may want to transport the tiny house after the exterior walls and siding is constructed but before the interior items are completed, the same stability analysis previously described was conducted on an empty model. By removing the internal components, this causes the COM to move upwards and closer to the front face, as shown below in Figure 80. While emptying the interior reduces the weight of the structure, the tip angles slightly decrease but are still well above the ~6.84° superelevation value (Table 8). The tongue weight percentages increase but are still within the 15% to 30% recommended value (Table 9). These changes may impact the specifications of the towing vehicle, so knowing the difference in tongue weight percentage between a fully loaded and empty tiny house can improve transit planning.

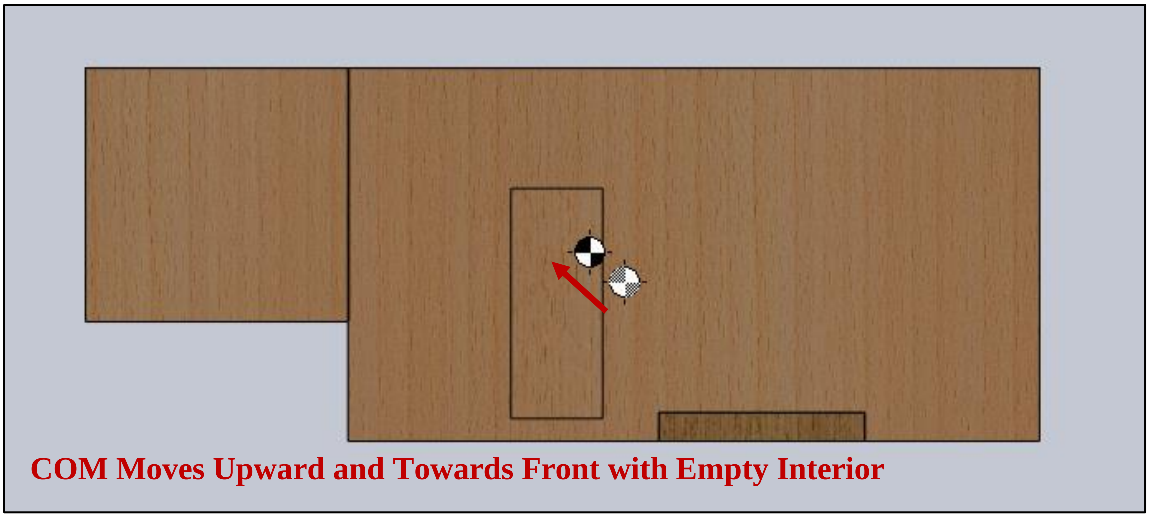


Figure 80: New COM Position with Empty Interior

(Driver Side View)

Table 8: Summary of Tip Angles for Empty Models

	20 ft Model	24 ft Model
Tip Angle — Driver Side [°]	27.91	27.96
Tip Angle — Passenger Side [°]	29.20	29.26
COM distance from center plane [in]	1.33	1.34
COM distance from center plane [%]	2.67	2.77

Table 9: Summary of Empty Tiny House Tongue Weight Percentages

	20 ft Model	24 ft Model
Distance from the Front Face to COM [in]	176.16	198.63
Distance from COM to Center of the Wheel Wells [in]	59.71	66.04
Total Weight [lb]	5,310.27	5,892.73
Tongue Weight Percentage [%]	25.3%	24.9%

Finally, the overall weight of the loaded structure was investigated. The trailer that the tiny house would be built on was not included in the model because Tiny House Basics, LLC's trailers are custom built for each client, so the exact weight distribution would be unknown unless a trailer was purchased. As a placeholder, Tiny House Basics, LLC was able to provide approximate weights for the 20 ft and 24 ft gooseneck trailers. The 20 ft and 24 ft models were originally designed with asphalt shingles for the roof to mimic the look of a traditional house. This was done before the weights of the trailers were obtained, and the result was that the asphalt shingles added too much weight to the structure. There was very little weight remaining for final items, so both models were updated to reflect a metal roof. This greatly reduced the weight of the structure and put each model safely under the 14,000 lb trailer capacity (Table 10). Shiplap siding is popular in the tiny house community, which is why it was used in the models, but vinyl siding could be used to reduce weight further.

Table 10: Summary of Loaded Tiny House Weights

	20 ft Model	24 ft Model
SolidWorks Model Weight (including interior items) [lb]	8,214.55	8,971.17
Estimated Trailer Weights from Tiny House Basics, LLC [lb]	3,400.00	3,800.00
Estimated Total Weight for Tiny House Shell [lb]	11,614.55	12,771.17
Trailer Capacity [lb]	14,000.00	14,000.00
Remaining Weight for Final Items [lb]	2,385.45	1,228.83

5.3. Thermal Analysis Results

Once a thermal simulation was finished running the resulting heat flux, internal temperature, and external temperature were reviewed. These factors summarize how much heat is exiting the structure in the winter and how much heat is entering the structure in the summer, all of which needs to be accounted for by the HVAC system. The figures below reference results for Grand Rapids, MI in the winter for the 20 ft model, but the same process was used for the other cities, temperatures, and the 24 ft model.

Several overlays were generated to view the results. The external temperature of the model was first examined with an overlay (Figure 81). For this simulation, the exterior surface temperature was approximately 7.7°F, verified by nodal probes. To view the inside space, a cross sectional view was enabled (Figure 82) and using nodal probes again, the interior surface was approximately 72°F. The cross section was examined further to view the temperature distribution through the structure (Figure 83). Seeing that the temperature gradually decreased from the interior to the exterior, the results aligned with physical expectations.

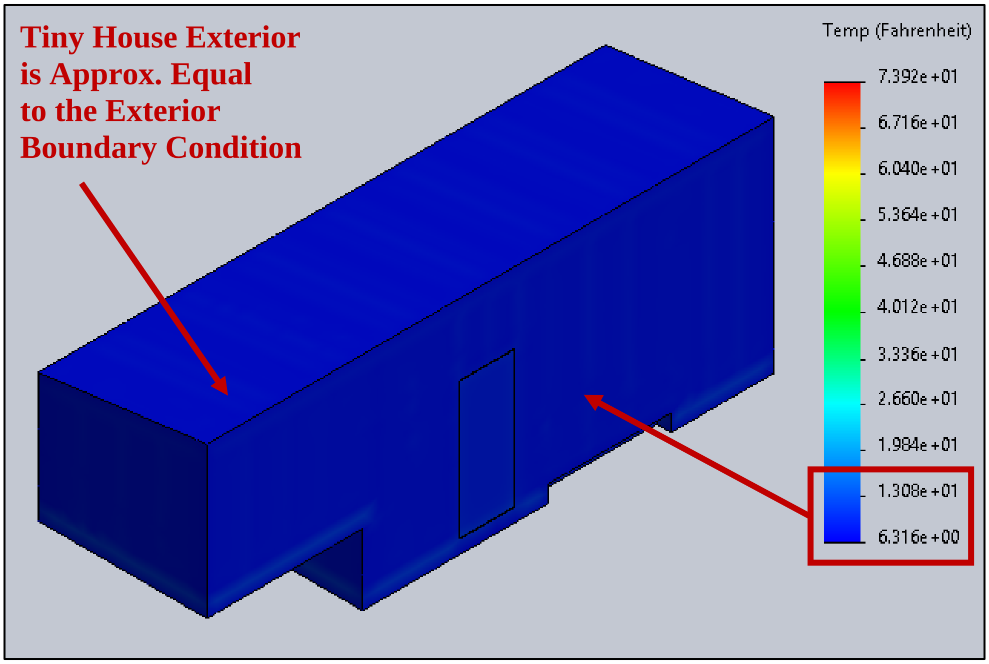


Figure 81: 20 ft Model Exterior Temp., Grand Rapids, MI Winter Boundary Conditions

(Isometric View)

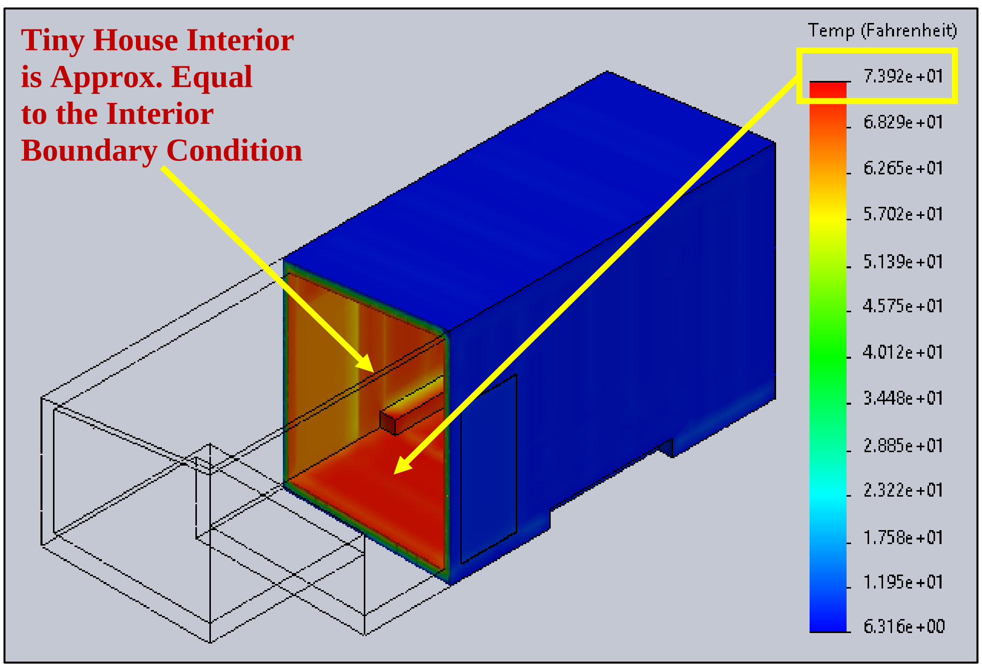


Figure 82: 20 ft Model Interior Temp., Grand Rapids, MI Winter Boundary Conditions

(Isometric View)

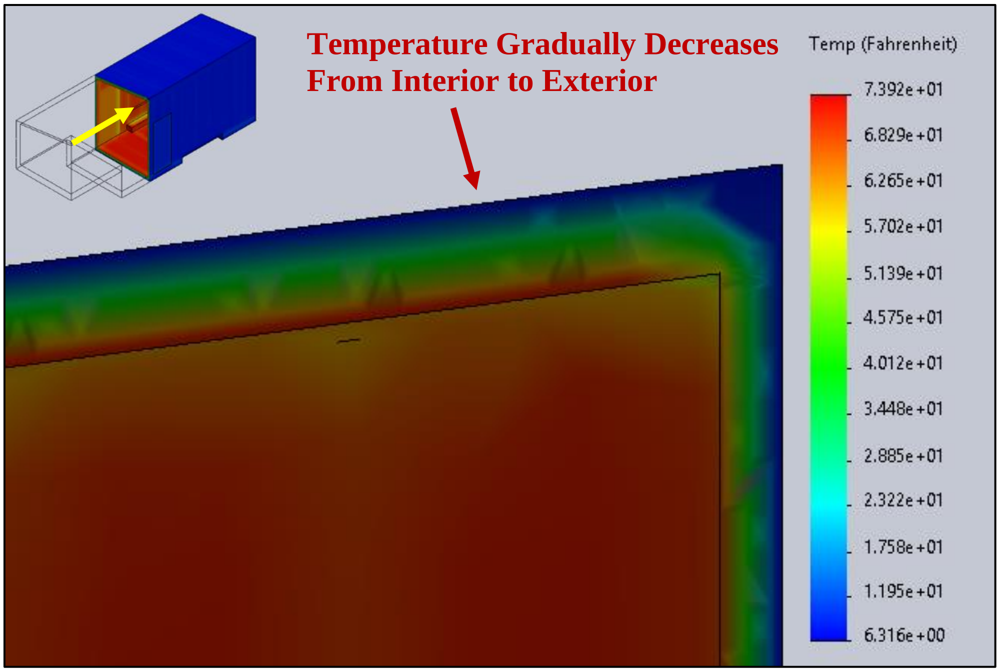


Figure 83: 20 ft Model Temp. Distribution, Grand Rapids, MI Winter Boundary Conditions (Front View)

After reviewing the temperature distribution, a heat flux overlay was generated (Figure 84). Instead of red being the maximum value, it was capped at 75 Watts per square meter (W/m^2) as some nodes had extremely high values which skewed the color scale and visual representation. The pink color was added to represent any heat flux value greater than 75 W/m^2. Again, the visual results align with physical expectations as values are lower in areas where insulation is located, and values are higher where wood supports are located.

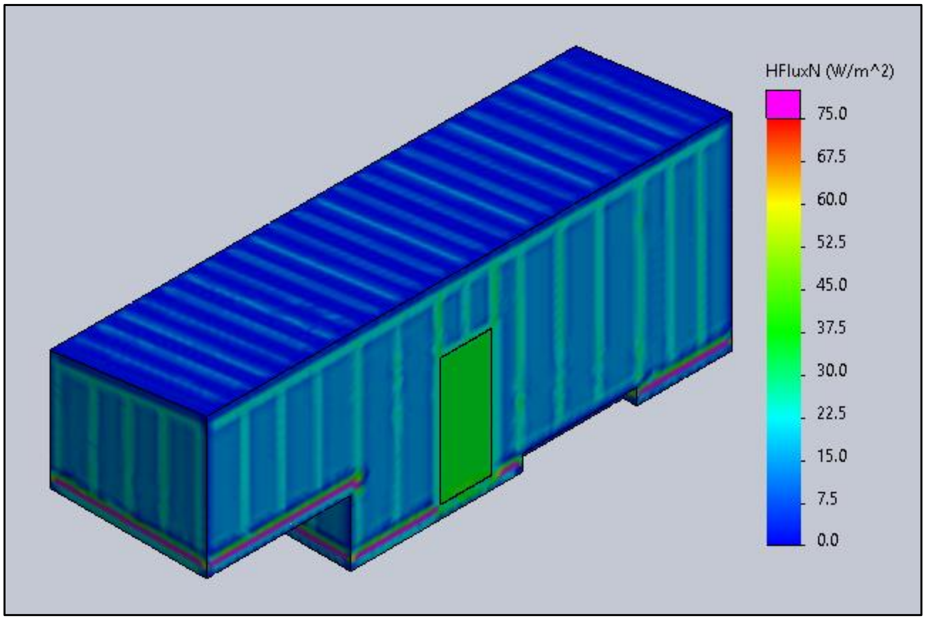


Figure 84: 20 ft Model Heat Flux, Grand Rapids, MI Winter Boundary Conditions

To determine the heat transfer rate through the structure, the nodal heat flux values were exported as a list (Figure 85 and Figure 86) so they could be averaged in a spreadsheet editor. In order to calculate the heat transfer rate, the average heat flux value needed to be multiplied by the surface area of the structure. The section properties tool was used to determine this value for the 20 ft model, 24 ft model, and for a single window. Individual sides were added together to obtain the overall exterior surface area. This value was used to calculate the heat transfer rate. The summary of these calculations, an example using a window simulation, as well as the additional capacity needed by the HVAC system due to air changes, can be found in Appendix XI: Thermal Analysis Calculations. Graphs of the heat transfer rate values are shown in Figure 88 and Figure 89, and numeric summaries are shown in Table 11 and Table 12. If the user selected more than one window, the value for the single window was multiplied by the total number of windows. The results aligned with physical expectations because if all other variables are held constant, the difference in temperature between the interior and exterior is doubled, the heat transfer rate should double as well. The results were then saved and used in the design tool.

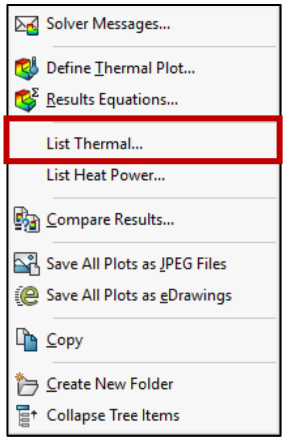


Figure 85: SolidWorks Tool Used to Export Thermal Results

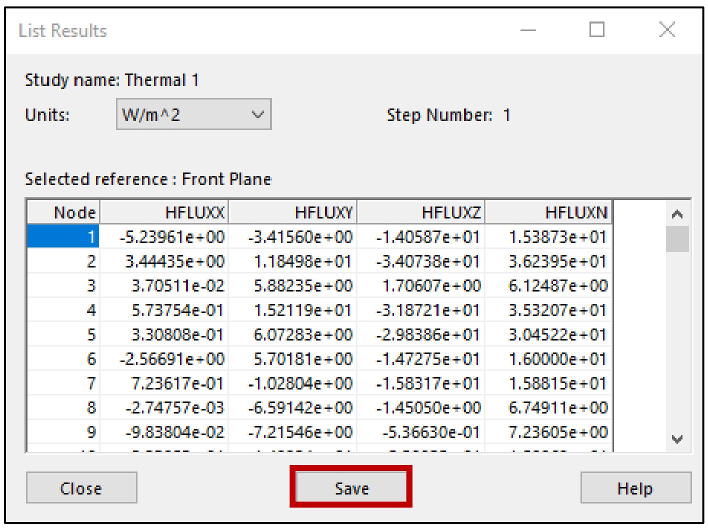


Figure 86: Heat Flux List to be Exported

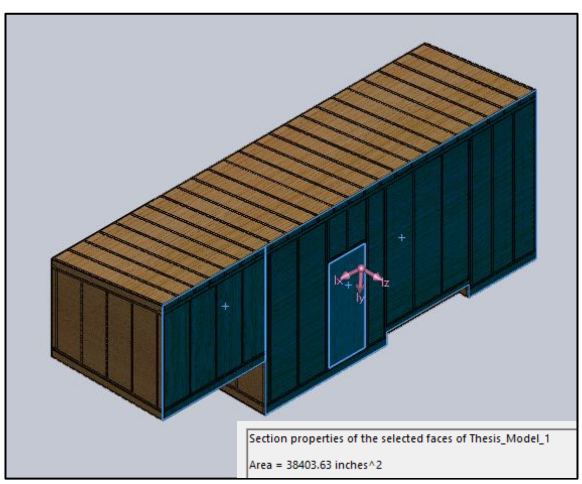


Figure 87: Example of Section Properties Tool Used to Determine Surface Area

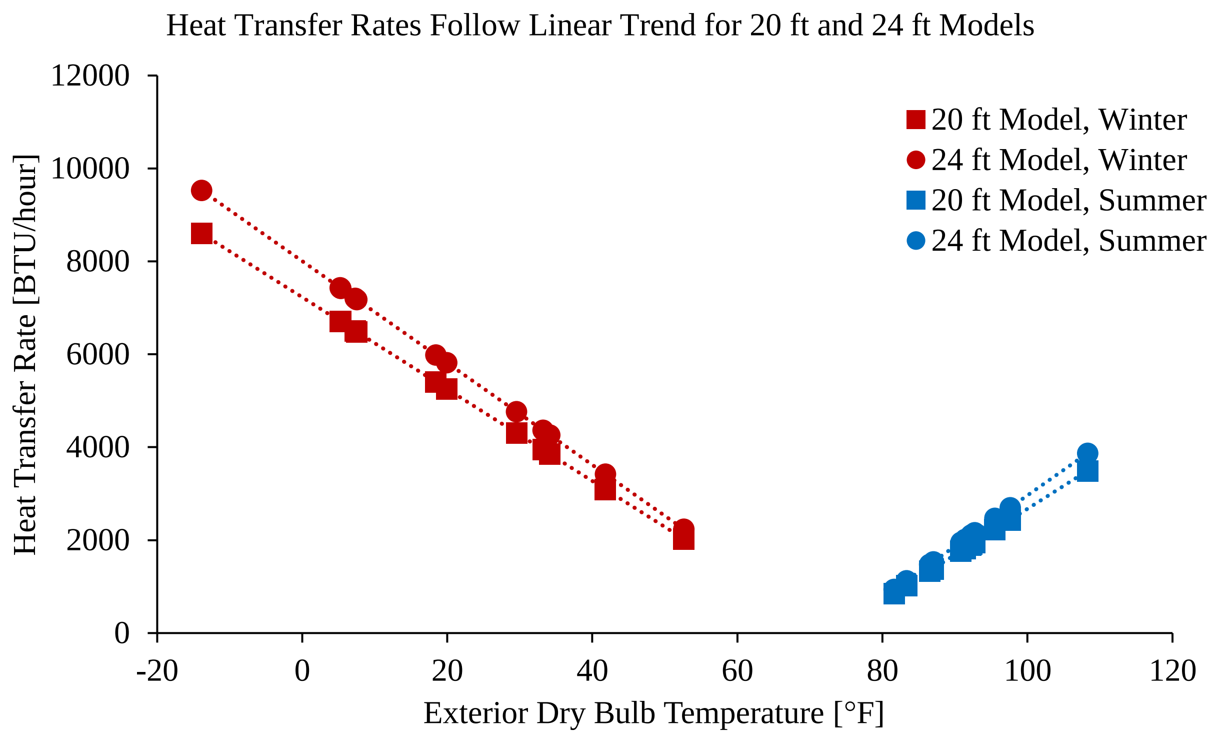


Figure 88: Heat Transfer Rate Results for 20 ft and 24 ft Models

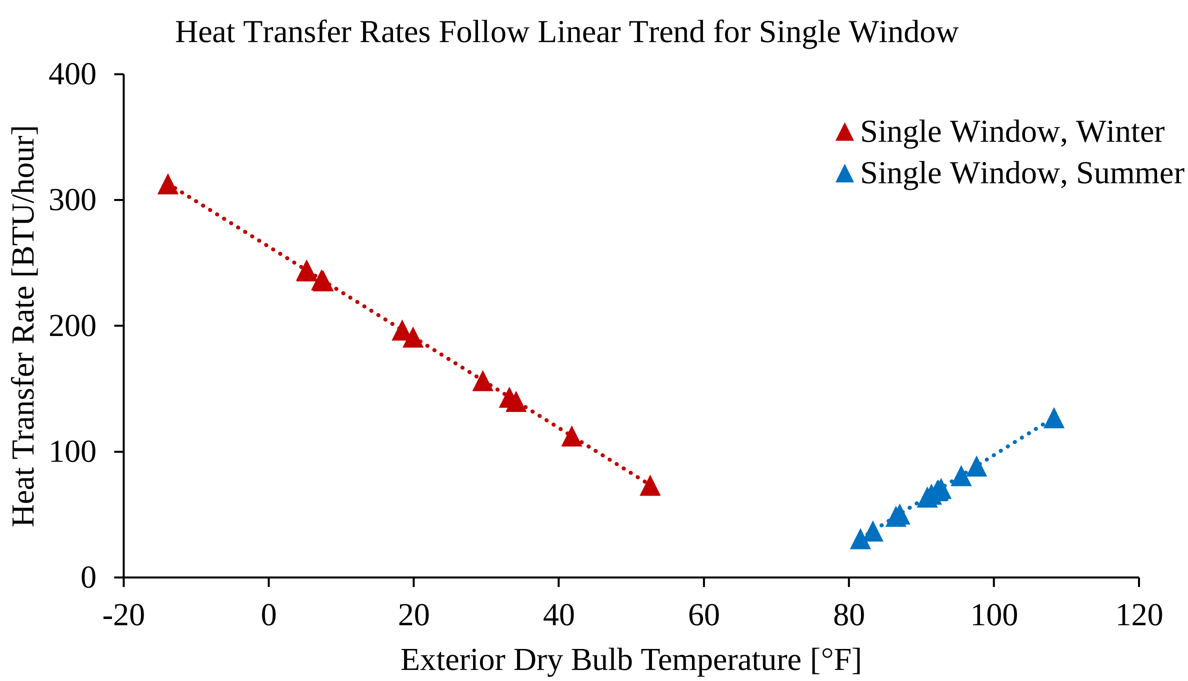


Figure 89: Heat Transfer Rate Results for Single Window

Table 11: Summary of Winter Heat Transfer Rates

No.	City	State	99% Dry Bulb Heating Temp. [°F]	Single Window Heat Transfer Rate, Winter [BTU/hour]	20 ft Model Heat Transfer Rate, Winter [BTU/hour]	24 ft Model Heat Transfer Rate, Winter [BTU/hour]
1	Baltimore	MD	18.4	196.7	5408.7	5988.2
2	Denver	CO	5.3	243.9	6703.9	7424.4
3	Fargo	ND	-13.9	313.0	8602.2	9529.4
4	Grand Rapids	MI	7.3	236.7	6506.2	7205.1
5	Houston	TX	34.1	140.1	3856.5	4267.0
6	Kansas City	MO	7.5	235.9	6486.4	7183.2
7	Miami	FL	52.6	73.5	2027.4	2238.7
8	Nashville	TN	19.9	191.3	5260.4	5823.8
9	Phoenix	AZ	41.8	112.5	3095.2	3422.8
10	Portland	ME	5.2	244.2	6713.8	7435.4
11	Sacramento	CA	33.2	143.4	3945.5	4365.6
12	Seattle	WA	29.5	156.7	4311.3	4771.3

Table 12: Summary of Summer Heat Transfer Rates

No.	City	State	1% Dry Bulb Cooling Temp. [°F]	Single Window Heat Transfer Rate, Summer [BTU/hour]	20 ft Model Heat Transfer Rate, Summer [BTU/hour]	24 ft Model Heat Transfer Rate, Summer [BTU/hour]
1	Baltimore	MD	91.4	66.3	1822.3	2020.4
2	Denver	CO	92.2	69.2	1901.6	2108.2
3	Fargo	ND	87.0	50.4	1386.5	1537.8
4	Grand Rapids	MI	86.5	48.6	1337.0	1483.0
5	Houston	TX	95.5	81.0	2228.4	2470.1
6	Kansas City	MO	92.7	71.0	1951.1	2163.0
7	Miami	FL	90.8	64.1	1762.9	1954.6
8	Nashville	TN	92.3	69.5	1911.5	2119.1
9	Phoenix	AZ	108.3	127.2	3496.3	3873.9
10	Portland	ME	83.3	37.1	1020.0	1132.1
11	Sacramento	CA	97.6	88.6	2436.4	2700.4
12	Seattle	WA	81.6	31.0	851.6	945.6

5.4. Final Information for the User

With all the results gathered from the various analyses, a list of the items that the user would need to build the tiny house could be created. All parts were measured, and values of like components were summed and recorded to create the BOMs. The components were then costed to the identical or similar product online. The associated BOMs and costs for the 20 ft model and 24 ft model are shown in Table 13 and Table 14. The rows in light blue reflect items that are adjusted in the Scilab design tool based on the user's answers. Finally, the Scilab design tool was coded. When run, it asks the user a series of questions that relate to the design and outputs information that will aid the user in constructing a tiny house. Note that a 36,000 BTU/hour mini-split was used as a placeholder in the BOMs for costing purposes as it is more than sufficient for any host city based on maximum values from Table 11 and Table 12 when including the additional capacity needed for air changes, but the Scilab design tool outputs more accurate heating and cooling needs. A flowchart is shown in Figure 90. The list of questions that the user will need to answer, as well as the list of available responses and the rationale behind the responses can be found in Appendix XII: Tiny House Design Tool Input Questions and Response Rationale. Appendix XIII: Tiny House Design Tool Instructions and Example Printout provides step-by-step instructions, for installing and opening the software and running the Scilab design tool. An example printout is also included so the user will know what to expect and how the results will be formatted. Printouts of the Scilab Code can be found in:

- Appendix XIV: Printout of Scilab Code: Master File
- Appendix XV: Printout of Scilab Code: Input Questions
- Appendix XVI: Printout of Scilab Code: Calculate Water Consumption
- Appendix XVII: Printout of Scilab Code: Calculate Live Load
- Appendix XVIII: Printout of Scilab Code: Calculate HVAC Needs
- Appendix XIX: Printout of Scilab Code: Calculate Energy Consumption
- Appendix XX: Printout of Scilab Code: Calculate BOM and Approx. Cost
- Appendix XXI: Printout of Scilab Code: Summary and Closing

Table 13: BOM and Costs for 20 ft Model

(Highlighted blue items change in the Scilab design tool with user's answers.)

Component	**Unit**	**$ / Unit**	**Amount: 20 ft Model**	**Subtotal: 20 ft Model**
2 x 4 (cost calculated from [48])	in	$ 0.04	8,902.81	$ 340.35
2 x 6 (cost calculated from [49])	in	$ 0.05	4,501.52	$ 210.07
2 x 8 (cost calculated from [50])	in	$ 0.08	1,736.00	$ 141.41
2 x 12 (cost calculated from [51])	in	$ 0.14	960.00	$ 131.20
Bathroom Sink (cost obtained from [52])	pc	$ 313.04	1.00	$ 313.04
Bathroom Tile (cost calculated from [53])	ft^2	$ 4.99	31.84	$ 158.89
Insulation (cost calculated from [54])	ft^2	$ 0.89	1,221.64	$ 1,082.12
Kitchen Counter (cost estimated)	pc	$ 250.00	1.00	$ 250.00
Kitchen Sink (cost obtained from [55])	pc	$ 289.00	1.00	$ 289.00
Metal Roof (cost calculated from [56])	ft^2	$ 1.00	228.42	$ 229.37
Mini-Split (cost obtained from [57])	pc	$ 1798.99	1.00	$ 1798.99
Plywood (cost calculated from [58]	ft^2	$ 0.37	2,309.51	$ 862.46
Queen Mattress (cost obtained from [59]	pc	$ 263.13	1.00	$ 263.13
Refrigerator (cost obtained from [60])	pc	$ 389.00	1.00	$ 389.00
Shiplap (cost calculated from [61])	ft^2	$ 1.38	537.19	$ 742.66
Siding (cost calculated from [61])	ft^2	$ 1.38	612.32	$ 846.53
Stove/Oven (cost obtained from [62]	pc	$ 555.17	1.00	$ 555.17
Toilet (cost obtained from [63])	pc	$ 99.00	1.00	$ 99.00
20 ft Trailer (cost obtained from manufacturer's email)	pc	$ 6,699.00	1.00	$ 6,699.00
Bathtub (cost obtained from [64])	pc	$ 323.48	1.00	$ 323.48
Microwave (cost obtained from [65]	pc	$ 59.21	1.00	$ 59.21
Washing Machine (cost obtained from [66])	pc	$ 999.00	1.00	$ 999.00
Window (cost calculated from [67])	pc	$ 81.23	6.00	$ 487.38
2 x 4s for Windows (cost calculated from [48])	in	$ 0.04	216.00	$ 8.26

Table 14: BOM and Costs for 24 ft Model

(Highlighted blue items change in the Scilab design tool with user's answers.)

Component	Unit	$ / Unit	Amount: 24 ft Model	Subtotal: 24 ft Model
2 x 4 (cost calculated from [48])	in	$ 0.04	9,362.34	$ 357.91
2 x 6 (cost calculated from [49])	in	$ 0.05	4,891.52	$ 228.27
2 x 8 (cost calculated from [50])	in	$ 0.08	1,736.00	$ 141.41
2 x 12 (cost calculated from [51])	in	$ 0.14	960.00	$ 131.20
Bathroom Sink (cost obtained from [52])	pc	$ 313.04	1.00	$ 313.04
Bathroom Tile (cost calculated from [53])	ft^2	$ 4.99	43.75	$ 218.31
Insulation (cost calculated from [54])	ft^2	$ 0.89	1,414.52	$ 1,252.98
Kitchen Counter (cost estimated)	pc	$ 250.00	1.00	$ 250.00
Kitchen Sink (cost obtained from [55])	pc	$ 289.00	1.00	$ 289.00
Metal Roof (cost calculated from [56])	ft^2	$ 1.00	261.36	$ 262.46
Mini-Split (cost obtained from [57])	pc	$ 1798.99	1.00	$ 1798.99
Plywood (cost calculated from [58]	ft^2	$ 0.37	2,594.36	$ 968.83
Queen Mattress (cost obtained from [59]	pc	$ 263.13	1.00	$ 263.13
Refrigerator (cost obtained from [60])	pc	$ 389.00	1.00	$ 389.00
Shiplap (cost calculated from [61])	ft^2	$ 1.38	610.95	$ 844.64
Siding (cost calculated from [61])	ft^2	$ 1.38	693.76	$ 959.12
Stove/Oven (cost obtained from [62]	pc	$ 555.17	1.00	$ 555.17
Toilet (cost obtained from [63])	pc	$ 99.00	1.00	$ 99.00
24 ft Trailer (cost obtained from manufacturer's email)	pc	$ 6,999.00	1.00	$ 6,999.00
Bathtub (cost obtained from [64])	pc	$ 323.48	1.00	$ 323.48
Microwave (cost obtained from [65]	pc	$ 59.21	1.00	$ 59.21
Washing Machine (cost obtained from [66])	pc	$ 999.00	1.00	$ 999.00
Window (cost calculated from [67])	pc	$ 81.23	6.00	$ 487.38
2 x 4s for Windows (cost calculated from [48])	in	$ 0.04	216.00	$ 8.26

After the Scilab design tool is run, the user will need to know how to build the individual parts as well as understand how each part is related to others. Appendix XXII: Tiny House Part Relations includes diagrams of how the 20 ft model is built up, and the same process can be used for the 24 ft model. Finally, construction drawings can be found in Appendix XXIII: Tiny House Construction Drawings so the user can build a real life version of either tiny house that was presented in this study that is safe, efficient, and cost-effective.

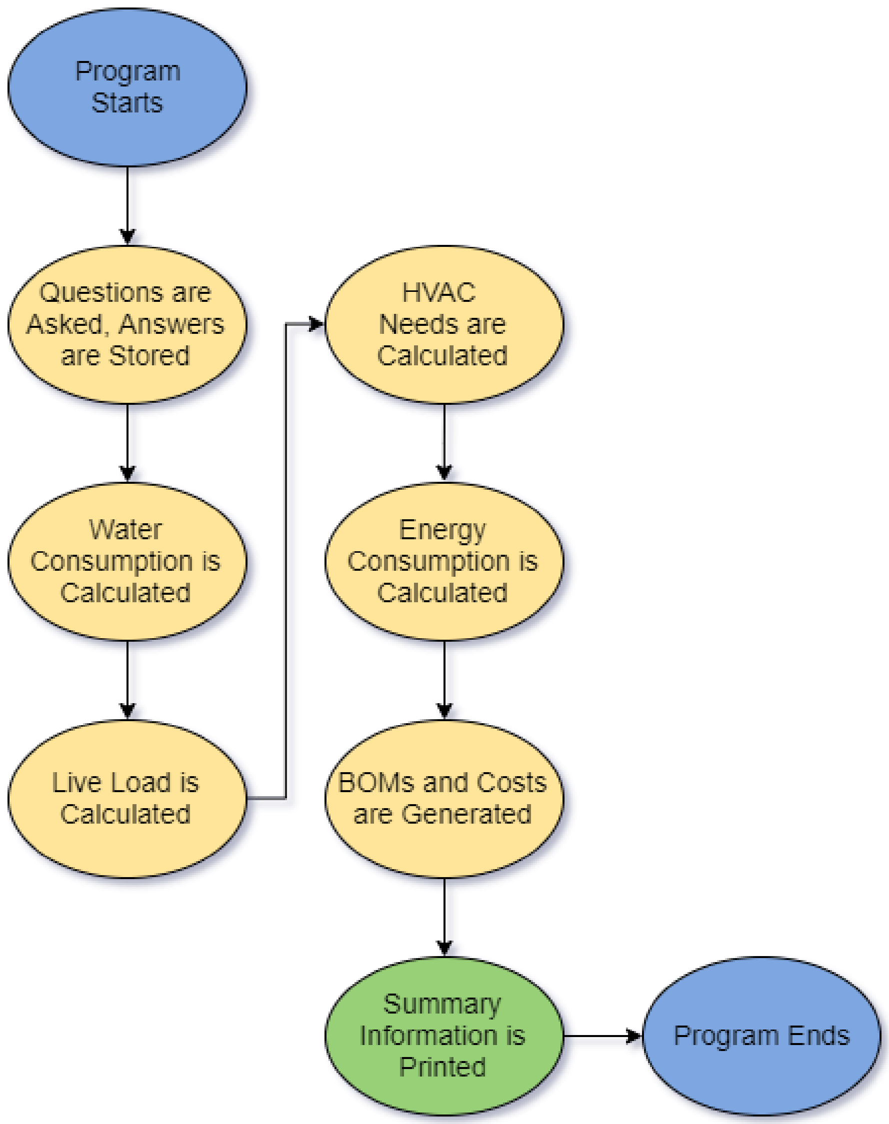


Figure 90: Tiny House Design Tool Code Flowchart

6. Limitations

During this process, a few limitations arose that were worth noting. While simulations were being ran and data was being collected, a 26 ft gooseneck trailer was going to be a third trailer size option for the user to choose from. This size was ultimately omitted because the weight of the tiny house shell and trailer exceeded the 14,000 lb capacity of the 2-axle trailer for the stated boundary conditions. Another limitation was that the FEA mesh for the thermal analyses was not as fine as it could have been. Exporting the thermal data for all the nodes proved to be more computationally intensive than expected, and it frequently caused SolidWorks to crash. The mesh was ultimately made as fine as it could be without crashing the software. Finally, all the analysis (except for the tip angles), had to be performed without the trailer being modeled. Tiny House Basics, LLC’s trailers are custom to each client, so the exact weight distribution would be unknown unless a trailer was purchased. Results were presented without the trailer as this was the most accurate alternative.

7. Lessons Learned

While working on this study, there were a few lessons learned that are being shared so individuals that work on a similar topic in the future can progress with less difficulty. The first is with respect to the SolidWorks model. The models used for this study were generated by starting with the interior and then adding components outward. About halfway through the structural analyses, it was noticed that the width of the exterior siding was not accounted for, so the overall exterior width exceeded the 102 in (8 ft 6 in) requirement [38]. Even though it was only about three in wider, it needed to be corrected. This ultimately resulted in repeating the structural analyses with the reduced width. This could have been avoided if the models were first created based on the maximum exterior dimensions and components were added in an inward fashion.

A second lesson learned concerns how parts and assembly were constructed. Each individual item (e.g. interior walls, floor) was designed as a part, and they were all combined into an assembly using top-down modeling. This worked well for the analyses, but it made the BOM generation quite tedious. As different pieces of dimensional lumber (e.g. 2 x 4s) were used in the same model, SolidWorks could not distinguish between the different pieces. This required the BOMs to be generated manually, and this could have been avoided if individual items were created as subassemblies with each piece of lumber being a separate part. The subassemblies could be built up the same way into the main assembly, but the difference is that the built-in feature for generating BOMs could have been used and time could have been saved.

Finally, the custom units within SolidWorks were not used as effectively as they could have been. For example, heat flux values were exported as W/m^2 as that was one of the default units, but if some time had been spent creating custom BTU/hour/ft^2 units, the thermal calculations could have been completed more quickly.

8. Recommendations for Future Work

Although this study does not cover all the topics required to construct a fully livable tiny house, there are a few recommendations for future work should others decide to pursue the topic in more detail. The first is with regards to overall weight. The original hypothesis was that weight would be a concern, but not a driving condition. This was incorrect, and items had to be adapted to meet the 14,000 lb requirement. Instead of focusing on materials that are typically used in the construction of a typical house (e.g. asphalt shingles), it is recommended to use lightweight materials wherever possible. This would not only make the structure lighter and reduce the towing capacity requirement, it would also potentially allow for less axles, reducing the overall cost of the build.

Another recommendation would be to adapt each tiny house to a specific city. The structural models used for this study were the same, regardless of city, to account for the fact that the user may relocate his or her tiny house. If the tiny house is relocated to an area with heavy snowfall, the structure would need to be able to support the accompanying loads. If the tiny house was going to stay in a warmer climate (e.g. Miami, FL), the same process used in this study could be used again but with different boundary conditions. This would reduce cost and weight as less materials would be needed.

Finally, other areas of the design could be investigated. For example, the bathroom wall was not designed to be load bearing, as the user has different door choices, but it could be. This may reduce the number of roof rafters, lower the overall weight, and possibly reduce costs. Another example could be to use the material properties of dry lumber, which may result in the reduction of other structural supports.

9. Conclusion

By carefully examining the structural, stability, weight, and thermal requirements of a 20 ft and 24 ft tiny house, the general public can learn about the technical details of a tiny house that are rarely covered by different television shows, internet videos, and photo sharing websites. This was done using a variety of information from multiple disciplines including: utilities, stress, COM, tongue weight, superelevation, heat transfer, modeling, and finite element analysis.

Data was collected, and a configurable SolidWorks model was created, which allowed for a structural analysis to be run. This was needed to ensure the tiny house would be able to support its own weight, the weight of the items and belongings inside, as well as be able to withstand wind loads during transit and snow loads for colder regions.

After that, the stability of the model was analyzed by first locating the COM. Tip angles, tongue weight percentages, and remaining trailer capacity were then calculated, and all values were suitable. This was done for both a fully loaded model with appliances and interior items as well as an empty shell. The empty condition needed to be considered because the tiny homeowner may be renting build space, and once the structure is water tight, the tiny house can be transported to a new location (likely outdoors) to finish working on interior items. In both situations, the stability and tongue weight percentages need to be within specified values.

Next, a thermal analysis was performed on each model and a single window. This determined how much heat was exiting the structure in the winter and how much heat was entering the structure in the summer, all of which needed to be accounted for by the HVAC system, in addition to the live load heat and air changes. Windows impacted the HVAC needs due to their low insulation properties, but the additional heat flux was accounted for those as well.

With all the results gathered from the various analyses, all parts were measured, and values of like components were summed and recorded to create the BOMs. The BOMs were costed online, and the final Scilab design tool was created. When run, the Scilab design tool asks the user a series of questions that relate to the design and output information that will aid the user in constructing a tiny house. The Scilab design tool, in addition to the separate construction drawings, resulting from this study allows users to build a safer, more efficient, and more cost-effective tiny house.

Appendix I: List of Disclaimers

By using the information in this thesis and associated Scilab design tool, the user acknowledges and accepts the following disclaimers:

- This study, the associated Scilab design tool, and construction drawings help provide information regarding the shell of a tiny house. It is the user's responsibility to acquire permits and hire licensed professionals as needed for various utility work.
- This study, the associated Scilab design tool, and construction drawings are only to be used as a reference to gauge what the result "could be" given the information input by the user.
- This study only considers a predetermined list of locations in the contiguous United States.
- This study only considers tiny houses built on "gooseneck" trailers.
- This study does not account for the tiny house's response to earthquakes.
- This study does not account for the effects of radiation from the sun.
- This study does not account for wheelchair or handicap accessibility and may not meet specific residential building requirements.
- The projected costs do not account for any applicable shipping/delivery fees or taxes.
- There are many additional considerations that must be accounted for, such as properly securing objects during transit, using a large enough transport vehicle, etc., and these items are the responsibility of the user.
- If the user outsources the build to a professional tiny house construction company, this thesis could be used as checklist to assist in the project, but it is not an all-inclusive list.
- It is the user's responsibility to follow all relevant national, state, and local codes and laws.
- While the information in this thesis and the resulting Scilab design tool were designed to be as accurate as possible, the author, Grand Valley State University, and affiliated associates cannot be held responsible for any actions taken by the user.

Appendix II: List of Assumptions

- A global contact in the SolidWorks model is sufficient to take the place of individual nails, screws, etc.
- Cost of ½ in plywood was the same as 3/8 in plywood
- Cost of dry Douglas fir is the same as green Douglas fir
- Couch cushion has the same density as the queen bed mattress
- Doors have the material property of Douglas fir
- Each cell (smart) phone consumes 25% the energy of a laptop and generates 25% the heat of a laptop
- Each laptop is charged once per day
- Each laptop takes three hours to fully charge
- Exhaust fans run for four hours per day
- Exterior siding has the same density as Douglas fir
- Grain direction is along the length of the lumber
- Heat generated from a pet is 25% the heat generated by an occupant
- Humidity control can be maintained by the dehumidification capabilities in the mini-split (moisture generated from shower/bath, cooking, etc. will require dehumidification)
- HVAC runs for twelve hours per day, average
- Kitchen sink is constructed of stainless steel
- Lights are on for 8 hours per day, average
- Lights will be light emitting diodes
- Microwave, if selected, will be 1,000 W and will run for 30 minutes a day, average
- Power input to lights is completely converted to heat (for live load calculations)
- Safety factor of two is sufficient for structural analysis [45]
- Sea level pressure is sufficient for highway wind loading and cross wind loading

- The number of air changes per hour is six
- The steel trailer creates a rigid frame for the floor
- The trailer has support jacks on the corners, which are deployed when the tiny house is not being transported
- Two doors meets fire egress requirements
- Two in of rain will produce one in of snow
- User may need to adjust the amount of bathroom tile if a bathtub is selected as the full bathroom floor was accounted for with tile to allow users to pursue alternatives bathtubs if desired (e.g. a horse trough)
- User will account for shrinkage of green lumber if building with it (construction drawings reference dry lumber)
- User will add vapor barrier, house wrap, and other building materials as needed
- User will be using a Windows PC to run Scilab 5.5.2
- User will utilize provided construction drawings to determine the full amount of materials that will need to be purchased (no buffer for scrap accounted for in BOMs)
- Washing machine, if selected, runs 30 minutes per load
- Water for each pet is the same as 1 bath per week
- 1% cooling dry bulb temperature occurs in the summer
- 280 ft^2 is used for lighting calculations (allowing for more light in the kitchen, for example)
- 30 days = one month (for energy calculations)
- 52 weeks = one year (for water calculations)
- 55 mph is the fastest speed the tiny house will be towed on the highway
- 99% heating dry bulb temperature occurs in the winter

Appendix III: Regional Weather Data

Table 15: Summary of Regional Weather Data

(data in this table was obtained from [68] using 2017 and "IP" settings)

No.	City	State	1% Extreme Wind Speed [mph]	Max. 1-Month Precipitation [in]	99% Heating Dry Bulb Temp. [°F]	1% Cooling Dry Bulb Temp. [°F]
1	Baltimore	MD	22	13.3	18.4	91.4
2	Denver	CO	27.1	7.2	5.3	92.2
3	Fargo	ND	28.1	11.7	-13.9	87
4	Grand Rapids	MI	24.9	9.1	7.3	86.5
5	Houston	TX	20	19.2	34.1	95.5
6	Kansas City	MO	25.4	15.5	7.5	92.7
7	Miami	FL	20	24.4	52.6	90.8
8	Nashville	TN	19	13.6	19.9	92.3
9	Phoenix	AZ	18.8	5.2	41.8	108.3
10	Portland	ME	23.2	15.2	5.2	83.3
11	Sacramento	CA	23.9	12.1	33.2	97.6
12	Seattle	WA	20.4	15.6	29.5	81.6

Appendix IV: Monthly Water Consumption Calculations

The equations and calculations below were used to estimate the monthly water consumption of the tiny house occupant(s). Appendix XVI: Printout of Scilab Code: Calculate Water Consumption shows how the information was coded into the Scilab design tool. Once all the items had the units of gallons per week, the result was multiplied by 52 weeks per year and divided by 12 months per year to obtain units of gallons per month. The estimated average water consumption in the United States is 3,000 gallons per month per person from one source [21]. If the user selected the maximum values the design tool allows for, the monthly water consumption would be at least 27.5% lower (approximately 4,345 gallons compared to 6,000 gallons for two occupants for one month).

$$Water\ for\ pets = 36\,\frac{gallons}{week \times pet} \qquad \textit{Eq. 27}$$

$$Water\ for\ washing\ machine = 15\,\frac{gallons}{load} \qquad \textit{Eq. 28}$$

(value of 15 obtained from [69])

$$Water\ for\ hand\ washing = 2.5\,\frac{gallons}{day \times occupant} \times 7\,\frac{days}{week} \qquad \textit{Eq. 29}$$

(value of 2.5 obtained from [69])

$$Water\ for\ washing\ dishes^{*} = 50\,\frac{gallons}{week} \qquad \textit{Eq. 30}$$

$$Water\ for\ cooking^{*} = 0.5\,\frac{gallons}{meal} \times 3\,\frac{meals}{day} \times 7\,\frac{days}{week} \qquad \textit{Eq. 31}$$

$$Water\ for\ household\ cleaning^{*} = 3\,\frac{gallons}{week} \qquad \textit{Eq. 32}$$

**Indicates self-estimate*

$$Water\ for\ drinking = 1\frac{gallon}{day \times occupant} \times 7\frac{days}{week} \qquad \textit{Eq. 33}$$

(value of 1 obtained from [69])

$$Water\ for\ showers = 20\frac{gallons}{shower} \times 7\frac{days}{week} \qquad \textit{Eq. 34}$$

(value of 20 obtained from [69])

$$Water\ for\ baths = 36\frac{gallons}{bath} \qquad \textit{Eq. 35}$$

(value of 36 obtained from [69])

$$Water\ for\ gen.hygiene = 2.5\frac{gallons}{day \times occupant} \times 7\frac{days}{week} \qquad \textit{Eq. 36}$$

(value of 2.5 obtained from [69])

$$Water\ for\ plants^{*} = 1\frac{cup}{day \times plant} \times \frac{1}{16}\ \frac{gallons}{cup} \times 7\frac{days}{week} \qquad \textit{Eq. 37}$$

$$Miscellaneous\ water = 1\frac{gallon}{day} \times 7\frac{days}{week} \qquad \textit{Eq. 38}$$

**Indicates self-estimate*

Appendix V: Monthly Energy Consumption Calculations

The equations and calculations on the following page were used to estimate the monthly energy consumption of the tiny house occupant(s). Placeholder values were used to complete these calculations, but when specific items are sourced, the energy consumption will be listed as part of the product's technical specifications. Appendix XIX: Printout of Scilab Code: Calculate Energy Consumption shows how the information was coded into the Scilab design tool. Once all the items had the units of watt-hours (Wh) per month, the result was divided by 1,000 to obtain units of kWh per month. The estimated average energy consumption for a United States household is 914.3 kWh per month [20]. If the user selected the maximum values the design tool allows for, the monthly energy consumption would be at least 46.8% lower (approximately 485.5 kWh per month compared to 914.3 kWh per month for one household).

$$Energy\ for\ lights = \frac{1}{80}\frac{Watts}{lumen} \times \frac{6690}{240}\frac{lumens}{ft^2} \times 280\ ft^2 \times 8\frac{hours\ running}{day} \times 30\frac{days}{month} \quad \textit{Eq. 39}$$

(value of 80 obtained from [70], value of 6690 obtained from [71])

$$Energy\ for\ laptops = 70\ Watts \times 3\frac{hours\ charging}{day \times laptop} \times 30\frac{days}{month} \quad \textit{Eq. 40}$$

(value of 70 obtained from [72])

$$Energy\ for\ cell\ phones = 17.5\ Watts \times 3\frac{hours\ charging}{day \times cell\ phone} \times 30\frac{days}{month} \quad \textit{Eq. 41}$$

$$Energy\ for\ microwave^* = 1{,}000\ Watts \times 0.5\frac{hours\ running}{day} \times 30\frac{days}{month} \quad \textit{Eq. 42}$$

$$Energy\ for\ refrigerator = 160\ Watts \times 20.4\frac{hours\ running}{day} \times 30\frac{days}{month} \quad \textit{Eq. 43}$$

(value of 160 obtained from [72], value of 20.4 (85% of 24) obtained from [73])

$$Energy\ for\ washing\ machine = 1{,}200\ Watts \times 0.5\frac{hours\ running}{load} \times \frac{52}{12}\frac{weeks}{months} \quad \textit{Eq. 44}$$

(value of 1,200 obtained from [72])

$$Energy\ for\ mini-split = 800\ Watts \times 12\frac{hours\ running}{day} \times 30\frac{days}{month} \quad \textit{Eq. 45}$$

(value of 800 obtained from [74])

$$Energy\ for\ exhaust\ fans = 12\ Watts \times 12\frac{hours\ running}{day} \times 30\frac{days}{month} \quad \textit{Eq. 46}$$

(value of 12 obtained from [75])

$$Miscellaneous\ energy = 500\ Watt\text{-}hours \quad \textit{Eq. 47}$$

**Indicates self-estimate*

Table 16: Average Monthly Energy Consumption for a United States Household

(data obtained from [20])

State	No. of Customers	Avg. kWh per month	Avg. kWh per month × No. of Customers
Alabama	2,229,472	1,236	2,755,627,392
Alaska	287,526	572	164,464,872
Arizona	2,808,352	1,028	2,886,985,856
Arkansas	1,388,358	1,156	1,604,941,848
California	13,591,152	546	7,420,768,992
Colorado	2,326,976	691	1,607,940,416
Connecticut	1,503,701	724	1,088,679,524
Delaware	432,449	977	422,502,673
District of Columbia	274,613	787	216,120,431
Florida	9,423,022	1,110	10,459,554,420
Georgia	4,354,021	1,142	4,972,291,982
Hawaii	436,266	518	225,985,788
Idaho	743,567	944	701,927,248
Illinois	5,289,573	744	3,935,442,312
Indiana	2,863,358	1,006	2,880,538,148
Iowa	1,385,756	892	1,236,094,352
Kansas	1,266,044	934	1,182,485,096
Kentucky	1,980,209	1,166	2,308,923,694
Louisiana	2,085,055	1,282	2,673,040,510
Maine	709,848	572	406,033,056
Maryland	2,332,517	1,005	2,344,179,585
Massachusetts	2,784,243	607	1,690,035,501
Michigan	4,365,529	671	2,929,269,959
Minnesota	2,420,325	786	1,902,375,450
Mississippi	1,290,281	1,247	1,608,980,407
Missouri	2,792,459	1,118	3,121,969,162
Montana	509,526	850	433,097,100
Nebraska	849,898	1,021	867,745,858
Nevada	1,183,660	947	1,120,926,020
New Hampshire	622,671	621	386,678,691
New Jersey	3,568,044	690	2,461,950,360
New Mexico	889,841	639	568,608,399
New York	7,190,906	604	4,343,307,224
North Carolina	4,550,420	1,129	5,137,424,180
North Dakota	382,596	1,118	427,742,328
Ohio	4,964,855	914	4,537,877,470
Oklahoma	1,764,980	1,139	2,010,312,220
Oregon	1,750,240	901	1,576,966,240
Pennsylvania	5,390,428	864	4,657,329,792
Rhode Island	442,005	589	260,340,945
South Carolina	2,290,200	1,159	2,654,341,800
South Dakota	400,150	1,045	418,156,750
Tennessee	2,882,992	1,283	3,698,878,736
Texas	11,148,784	1,176	13,110,969,984
Utah	1,091,162	742	809,642,204
Vermont	315,138	560	176,477,280
Virginia	3,431,574	1,165	3,997,783,710
Washington	3,076,868	957	2,944,562,676
West Virginia	859,039	1,133	973,291,187
Wisconsin	2,700,651	693	1,871,551,143
Wyoming	272,427	841	229,111,107
		Weighted kWh per month avg.	914.3

Appendix VI: Live Load Heat Calculations

The equations and calculations on the following page were used to estimate the additional live heat load generated by personal items and appliances that the HVAC system would need to account for. Appendix XVII: Printout of Scilab Code: Calculate Live Load shows how the information was coded into the Scilab design tool. Once all the items had the units of BTU/hour, the result was saved in the code and printed to the command window for the user to reference.

$$Heat\ from\ lights = \frac{1}{80}\frac{Watts}{lumen} \times \frac{6690}{240}\frac{lumens}{ft^2} \times 280\ ft^2 \times 3.412\frac{BTU}{Watt \times hour} \quad \text{Eq. 48}$$

(value of 80 obtained from [70], value of 6690 obtained from [71])

$$Heat\ from\ occupants = 356\frac{BTU}{hour \times occupant} \quad \text{Eq. 49}$$

(value of 356 obtained from [76])

$$Heat\ from\ pets = 89\frac{BTU}{hour \times pet} \quad \text{Eq. 50}$$

$$Heat\ from\ laptops = 55.6\frac{Watts}{laptop} \times 3.412\frac{BTU}{watt \times hour} \quad \text{Eq. 51}$$

(value of 55.6 obtained from [77])

$$Heat\ from\ cell\ phones = 13.9\frac{Watts}{cell\ phone} \times 3.412\frac{BTU}{watt \times hour} \quad \text{Eq. 52}$$

$$Heat\ from\ refrigerator = 690\frac{Watts}{m^3} \times \frac{0.02832\ m^3}{ft^3} \times 18 ft^3 \times 3.412\frac{BTU}{Watt \times hour} \quad \text{Eq. 53}$$

(value of 690 obtained from [78])

$$Energy\ for\ HVAC = 500\ watts \times 3.412\frac{BTU}{Watt \times hour} \quad \text{Eq. 54}$$

(value of 500 obtained from [75])

$$Miscellaneous\ Heat = 2{,}000\frac{BTU}{hour} \quad \text{Eq. 55}$$

Appendix VII: Material Properties

Table 17: Material Properties Used for Structural and Thermal Analyses

Property	Douglas Fir	3/8 in Plywood	Denim Insulation
Elastic Modulus	1,7650,000 psi (obtained from [79])	1,009,463 psi (obtained from [42])	N/A
Poisson's Ratio	0.29 (obtained from [42])	0.30 (obtained from [80])	N/A
Density	0.01850 lb / in^3 (obtained from [79])	0.02074 lb / in^3**	7.88×10^{-4} lb / in^3**
Yield Strength	380 psi (obtained from [42])	4500 psi (obtained from [81])	N/A
Thermal Conductivity	1.60×10^{-6} BTU / (in × s × °F)**	2.71×10^{-6} BTU / (in × s × °F)**	5.19×10^{-7} BTU / (in × s × °F)**

(second is abbreviated as s, square inches is abbreviated as in^2, meters is abbreviated as m, square meters is abbreviated as m^2, Kelvin is abbreviated as K)

***Indicates that units were converted from reference, see corresponding equations and calculations below*

$$Douglas\ fir\ thermal\ conductivity = 0.83\,\frac{BTU \times in}{hour \times ft^2 \times {}^\circ F} \times \frac{1}{144}\frac{ft^2}{in^2} \times \frac{1}{3600}\frac{hour}{s} = 1.60 \times 10^{-6}\,\frac{BTU}{in \times s \times {}^\circ F} \qquad Eq.\ 56$$

(value of 0.83 obtained from [42])

$$Plywood\ density = 1.12\,\frac{lb}{ft^2} \times \frac{1}{144}\frac{ft^2}{in^2} \times \frac{3}{8}in = 0.02074\,\frac{lb}{in^3} \qquad Eq.\ 57$$

(value of 1.12 obtained from [82])

$$Plywood\ R-Value = 0.47\ \frac{m^2 \times K}{Watts} = \frac{thickness\ in\ m}{\lambda_{Plywood}} \qquad \textit{Eq. 58}$$

(value of 0.47 obtained from [83])

$$\lambda_{Plywood} = \frac{3}{8} in \times \frac{0.0254}{1} \frac{m}{in} \times \frac{1}{0.47}\ \frac{Watts}{m^2 \times \mathrm{K}} = \ 0.20266 \frac{Watts}{m \times K} \qquad \textit{Eq. 59}$$

$$Plywood\ thermal\ conductivity = 0.20266 \frac{Watts}{m \times K}\ \times \qquad \textit{Eq. 60}$$

$$0.00193 \frac{BTU \times in \times m \times K}{s \times ft^2 \times {}^\circ F \times Watts} \times \frac{1}{144} \frac{ft^2}{in^2} = 2.71 \times 10^{-6}\ \frac{BTU}{in \times s \times {}^\circ F}$$

$$Denim\ insulation\ density = 5.9 \frac{lb}{batt}\ \times \frac{1\ batt}{23\ in \times 3.5\ in \times 93\ in} = \qquad \textit{Eq. 61}$$

$$7.88\ \times 10^{-4} \frac{lb}{in^3}$$

(value of 5.9 obtained from [40])

$$Denim\ insulation\ R-Value = 13\ \frac{{}^\circ F \times ft^2 \times hour}{BTU} \qquad \textit{Eq. 62}$$

(value of 13 obtained from [40])

$$Denim\ insulation\ thermal\ conductivity = \frac{1}{13}\ \frac{BTU}{{}^\circ F \times ft^2 \times hour} \times \qquad \textit{Eq. 63}$$

$$\frac{1}{3600} \frac{hour}{s}\ \times \frac{1}{144} \frac{ft^2}{in^2} \times 3.5\ in = 5.19\ \times 10^{-7}\ \frac{BTU}{s \times in \times {}^\circ F}$$

Appendix VIII: Structural Analysis Boundary Conditions

The boundary conditions below were used to constrain the structural analysis to worst-case scenarios that the tiny house may experience over its operational lifetime. As it was unknown where structural supports would be on the trailer, an additional safety factor of two was added for the interior floor loading.

Table 18: Summary of Stationary Boundary Conditions

Boundary Condition	**20 ft Model**	**24 ft Model**
Snow Loading	0.09 psi***	0.09 psi***
Gravity	-386.22 in / s^2	-386.22 in / s^2
Internal and External Shiplap	0.0223 psi***	0.0223 psi***
Interior Floor	0.96 psi***	0.84 psi***
Exterior Floors	Fixed	Fixed

Table 19: Summary of Mobile Boundary Conditions

Boundary Condition	**20 ft Model**	**24 ft Model**
Highway Wind Loading	0.054 psi***	0.054 psi***
Crosswind Loading	0.014 psi***	0.014 psi***
Gravity	-386.22 in / s^2	-386.22 in / s^2
Internal and External Shiplap	0.0223 psi***	0.0223 psi***
Interior Floor	0.96 psi***	0.84 psi***
Exterior Floors	Fixed	Fixed

(cubic feet is abbreviated as ft^3, cubic meters is abbreviated as m^3, kilogram is abbreviated as kg, and Newton is abbreviated as N)

****Indicates that value resulted from calculations, see following pages*

$$Snow\ Loading = 20\frac{lb}{ft^3} \times \frac{1}{1728}\frac{ft^3}{in^3} 15.6\ in\ rain\ \times \frac{1}{2}\frac{in\ snow}{in\ rain} = 0.09\ psi \quad \text{Eq. 64}$$

(value of 20 obtained from [84])

$$p = \frac{1}{2}\rho v^2 \quad \text{Eq. 65}$$

Where:

p is the air pressure
ρ is the air density
v is the air velocity

$$Highway\ Wind\ Loading = \frac{1}{2}\left(1.225\frac{kg}{m^3}\right) \times \left(55\frac{miles}{hour} \times \frac{1609}{1}\frac{meters}{mile} \times \frac{1}{3600}\frac{hour}{s}\right)^2 \times \frac{1}{4.4488}\frac{lb}{N} \times \frac{1}{1550}\frac{m^2}{in^2} = 0.054\ psi \quad \text{Eq. 66}$$

$$Cross\ Wind\ Loading = \frac{1}{2}\left(1.225\frac{kg}{m^3}\right) \times \left(28.1\frac{miles}{hour} \times \frac{1609}{1}\frac{m}{mile} \times \frac{1}{3600}\frac{hour}{s}\right)^2 \times \frac{1}{4.4488}\frac{lb}{N} \times \frac{1}{1550}\frac{m^2}{in^2} = 0.014\ psi \quad \text{Eq. 67}$$

$$Interior\ Floor\ Loading_{20\ ft\ Model} = 14{,}000\ lb \times \frac{1}{202.5\ ft^2} \times \frac{1}{144}\frac{ft^2}{in^2} \times 2\ safety\ factor = 0.96\ psi \quad \text{Eq. 68}$$

$$Interior\ Floor\ Loading_{24\ ft\ Model} = 14{,}000\ lb \times \frac{1}{232.5\ ft^2} \times \frac{1}{144}\frac{ft^2}{in^2} \times 2\ safety\ factor = 0.84\ psi \quad \text{Eq. 69}$$

$$Internal\ and\ External\ Shiplap\ Loading = 0.0185\frac{lb}{in^3} \times 0.625\ in\ thick \times 2\ sides = 0.0223\ psi \quad \text{Eq. 70}$$

(value of 0.0185 obtained from [79], value of 0.625 obtained from [85])

Appendix IX: Weights and Dimensions of Household Items

Within the SolidWorks model, several household items were added not only to provide more accurate stability and weight calculations, but also to help visualize the space more effectively. A summary is provided below in Table 20.

Table 20: Summary of Weights and Dimensions of Household Items

Item	Width [in]	Depth [in]	Height [in]	Weight [lb]
Bathtub (adapted from [86])	66.14	28.74	14.57	63.58
Bathroom Sink (adapted from [87])	23	18.5	32.5	48.48
Bathroom Tile Floor (value of 0.75 obtained from [88])	89.25	Based on Model	0.75	Based on Model
Bed Frame	64	84	8	222.00
Closet	76	20	115.76	492.72
Couch	65	30	30	80.04
Couch Cushion	65	24	10	22.38
Kitchen Counter (adapted from [89])	55	32.18	37.68	285.37
Kitchen Sink (adapted from [89])	25.15	25.65	8.17	102.41
Oak Floor (value of 0.75 obtained from [90])	89.25	Based on Model	0.75	Based on Model
Queen Bed Mattress (value of 68.6 obtained from [91]	60	80	10	68.6
Refrigerator (adapted from [92])	23.03	25.71	60.63	117
Stairs	24	50	32.58	37.84
Stove/Oven (adapted from [93])	21.65	22.15	31.34	150
Table (adapted from [94])	36	24	31	55
Toilet (adapted from [95])	19.5	30.45	34	86
Washer/Dryer Combo (adapted from [96])	23.62	24.02	33.46	145.5

Appendix X: Thermal Analysis Boundary Conditions

The boundary conditions below were used in the thermal analysis. As it was unknown how many windows each user would desire, separate simulations were performed for those. An internal temperature of 73°F was selected because it was halfway between the recommended range of 68°F to 78°F [97]. The window was 3/32 in thick, based on typical glass thickness, and it was also modeled as 16 in wide and 26 in high to roughly match the cutout necessary to install the window [98][98, 99].

Table 21: Summary of Thermal Boundary Conditions

Boundary Condition	**20 ft Model**	**24 ft Model**
Internal Temperature	73°F	73°F
External Temperature	Defined by Host City	Defined by Host City
Internal Convection Heat Transfer Coefficient	2.87×10^{-6} BTU/(s×in^2×°F)***	2.87×10^{-6} BTU/(s×in^2×°F)***
External Convection Heat Transfer Coefficient	1.71×10^{-5} BTU/(s×in^2×°F)***	1.71×10^{-5} BTU/(s×in^2×°F)***

****Indicates that value resulted from calculations, see below*

$$Internal\ Heat\ Transfer\ Coefficient = 8.44\frac{Watts}{m^2\times K}\times \quad \textit{Eq. 71}$$

$$\frac{3.412}{1}\frac{BTU}{hour\times Watts}\times\frac{1}{3600}\frac{hour}{s}\times\frac{1}{1550}\frac{m^2}{in^2}\times\frac{5}{9}\frac{K}{°F} = 2.87\times10^{-6}\frac{BTU}{s\times in^2\times °F}$$

(value of 8.44 obtained from [100])

$$External\ Heat\ Transfer\ Coefficient = 50.5\frac{Watts}{m^2\times K}\times \quad \textit{Eq. 72}$$

$$\frac{3.412}{1}\frac{BTU}{hour\times Watts}\times\frac{1}{3600}\frac{hour}{s}\times\frac{1}{1550}\frac{m^2}{in^2}\times\frac{5}{9}\frac{K}{°F} = 1.71\times10^{-5}\frac{BTU}{s\times in^2\times °F}$$

(value of 50.5 obtained from [100])

Appendix XI: Thermal Analysis Calculations

Figure 91 below shows the heat flux value for the window simulation using Grand Rapids, MI's winter boundary condition, and Figure 92 shows the surface area of the window. These values were multiplied together and then converted to BTU/hour to determine the heat transfer rate through the window. This calculation process was repeated for all host cities' temperatures for not only the windows, but also for the overall tiny house models. Table 22 to Table 27 shows a summary of the calculations.

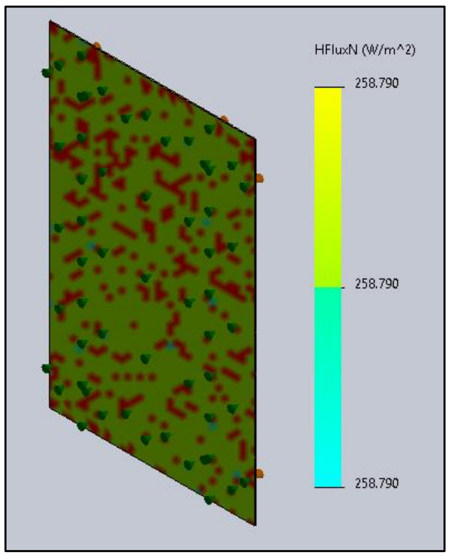


Figure 91: Single Window Heat Flux Results

(Isometric View)

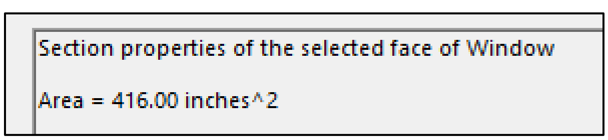


Figure 92: Surface Area of Window

$$Window\ Heat\ Transfer_{Grand\ Rapids, MI:\ Winter} = 258.8\frac{Watts}{m^2} \times 416.0\ in^2 \times \frac{1}{1550}\frac{m^2}{in^2} \times 3.412\frac{BTU}{Watts \times hour} = 237.0\frac{BTU}{hour} \quad Eq.\ 73$$

Table 22: Summary of Single Window Thermal Calculations, Winter Boundary Conditions

No.	City	State	99% Heating Dry Bulb Temp. [°F]	Avg. Heat Flux [W/m^2]	Surface Area [in^2]	Surface Area [m^2]	Heat Transfer Rate [W]	Heat Transfer Rate [BTU/hour]
1	Baltimore	MD	18.4	215.1	416.0	0.268	57.6	196.7
2	Denver	CO	5.3	266.7	416.0	0.268	71.5	243.9
3	Fargo	ND	-13.9	342.3	416.0	0.268	91.7	313.0
4	Grand Rapids	MI	7.3	258.8	416.0	0.268	69.4	236.7
5	Houston	TX	34.1	153.3	416.0	0.268	41.1	140.1
6	Kansas City	MO	7.5	258.0	416.0	0.268	69.1	235.9
7	Miami	FL	52.6	80.4	416.0	0.268	21.5	73.5
8	Nashville	TN	19.9	209.2	416.0	0.268	56.1	191.3
9	Phoenix	AZ	41.8	123.0	416.0	0.268	33.0	112.5
10	Portland	ME	5.2	267.1	416.0	0.268	71.6	244.2
11	Sacramento	CA	33.2	156.8	416.0	0.268	42.0	143.4
12	Seattle	WA	29.5	171.3	416.0	0.268	45.9	156.7

Table 23: Summary of 20 ft Model Thermal Calculations, Winter Boundary Conditions

No.	City	State	99% Heating Dry Bulb Temp. [°F]	Avg. Heat Flux [W/m^2]	Surface Area [in^2]	Surface Area [m^2]	Heat Transfer Rate [W]	Heat Transfer Rate [BTU/hour]
1	Baltimore	MD	18.4	15.0	163913.4	105.8	1585.1	5408.7
2	Denver	CO	5.3	18.6	163913.4	105.8	1964.7	6703.9
3	Fargo	ND	-13.9	23.8	163913.4	105.8	2521.1	8602.2
4	Grand Rapids	MI	7.3	18.0	163913.4	105.8	1906.8	6506.2
5	Houston	TX	34.1	10.7	163913.4	105.8	1130.2	3856.5
6	Kansas City	MO	7.5	18.0	163913.4	105.8	1901.0	6486.4
7	Miami	FL	52.6	5.6	163913.4	105.8	594.2	2027.4
8	Nashville	TN	19.9	14.6	163913.4	105.8	1541.7	5260.4
9	Phoenix	AZ	41.8	8.6	163913.4	105.8	907.1	3095.2
10	Portland	ME	5.2	18.6	163913.4	105.8	1967.6	6713.8
11	Sacramento	CA	33.2	10.9	163913.4	105.8	1156.3	3945.5
12	Seattle	WA	29.5	11.9	163913.4	105.8	1263.5	4311.3

Table 24: Summary of 24 ft Model Thermal Calculations, Winter Boundary Conditions

No.	City	State	99% Heating Dry Bulb Temp. [°F]	Avg. Heat Flux [W/m^2]	Surface Area [in^2]	Surface Area [m^2]	Heat Transfer Rate [W]	Heat Transfer Rate [BTU/hour]
1	Baltimore	MD	18.4	14.7	185001.1	119.4	1755.0	5988.2
2	Denver	CO	5.3	18.2	185001.1	119.4	2175.9	7424.4
3	Fargo	ND	-13.9	23.4	185001.1	119.4	2792.8	9529.4
4	Grand Rapids	MI	7.3	17.7	185001.1	119.4	2111.6	7205.1
5	Houston	TX	34.1	10.5	185001.1	119.4	1250.5	4267.0
6	Kansas City	MO	7.5	17.6	185001.1	119.4	2105.2	7183.2
7	Miami	FL	52.6	5.5	185001.1	119.4	656.1	2238.7
8	Nashville	TN	19.9	14.3	185001.1	119.4	1706.8	5823.8
9	Phoenix	AZ	41.8	8.4	185001.1	119.4	1003.1	3422.8
10	Portland	ME	5.2	18.3	185001.1	119.4	2179.1	7435.4
11	Sacramento	CA	33.2	10.7	185001.1	119.4	1279.4	4365.6
12	Seattle	WA	29.5	11.7	185001.1	119.4	1398.3	4771.3

Table 25: Summary of Single Window Thermal Calculations, Summer Boundary Conditions

No.	City	State	1% Cooling Dry Bulb Temp. [°F]	Avg. Heat Flux [W/m^2]	Surface Area [in^2]	Surface Area [m^2]	Heat Transfer Rate [W]	Heat Transfer Rate [BTU/hour]
1	Baltimore	MD	91.4	72.5	416.0	0.268	19.4	66.3
2	Denver	CO	92.2	75.6	416.0	0.268	20.3	69.2
3	Fargo	ND	87	55.1	416.0	0.268	14.8	50.4
4	Grand Rapids	MI	86.5	53.2	416.0	0.268	14.3	48.6
5	Houston	TX	95.5	88.6	416.0	0.268	23.8	81.0
6	Kansas City	MO	92.7	77.6	416.0	0.268	20.8	71.0
7	Miami	FL	90.8	70.1	416.0	0.268	18.8	64.1
8	Nashville	TN	92.3	76.0	416.0	0.268	20.4	69.5
9	Phoenix	AZ	108.3	139.0	416.0	0.268	37.3	127.2
10	Portland	ME	83.3	40.6	416.0	0.268	10.9	37.1
11	Sacramento	CA	97.6	96.9	416.0	0.268	26.0	88.6
12	Seattle	WA	81.6	33.9	416.0	0.268	9.1	31.0

Table 26: Summary of 20 ft Model Thermal Calculations, Summer Boundary Conditions

No.	City	State	1% Cooling Dry Bulb Temp. [°F]	Avg. Heat Flux [W/m^2]	Surface Area [in^2]	Surface Area [m^2]	Heat Transfer Rate [W]	Heat Transfer Rate [BTU/hour]
1	Baltimore	MD	91.4	5.1	163913.4	105.8	534.1	1822.3
2	Denver	CO	92.2	5.3	163913.4	105.8	557.3	1901.6
3	Fargo	ND	87	3.8	163913.4	105.8	406.3	1386.5
4	Grand Rapids	MI	86.5	3.7	163913.4	105.8	391.8	1337.0
5	Houston	TX	95.5	6.2	163913.4	105.8	653.1	2228.4
6	Kansas City	MO	92.7	5.4	163913.4	105.8	571.8	1951.1
7	Miami	FL	90.8	4.9	163913.4	105.8	516.7	1762.9
8	Nashville	TN	92.3	5.3	163913.4	105.8	560.2	1911.5
9	Phoenix	AZ	108.3	9.7	163913.4	105.8	1024.7	3496.3
10	Portland	ME	83.3	2.8	163913.4	105.8	298.9	1020.0
11	Sacramento	CA	97.6	6.8	163913.4	105.8	714.0	2436.4
12	Seattle	WA	81.6	2.4	163913.4	105.8	249.6	851.6

Table 27: Summary of 24 ft Model Thermal Calculations, Summer Boundary Conditions

No.	City	State	1% Cooling Dry Bulb Temp. [°F]	Avg. Heat Flux [W/m^2]	Surface Area [in^2]	Surface Area [m^2]	Heat Transfer Rate [W]	Heat Transfer Rate [BTU/hour]
1	Baltimore	MD	91.4	5.0	185001.1	119.4	592.1	2020.4
2	Denver	CO	92.2	5.2	185001.1	119.4	617.8	2108.2
3	Fargo	ND	87	3.8	185001.1	119.4	450.7	1537.8
4	Grand Rapids	MI	86.5	3.6	185001.1	119.4	434.6	1483.0
5	Houston	TX	95.5	6.1	185001.1	119.4	723.9	2470.1
6	Kansas City	MO	92.7	5.3	185001.1	119.4	633.9	2163.0
7	Miami	FL	90.8	4.8	185001.1	119.4	572.8	1954.6
8	Nashville	TN	92.3	5.2	185001.1	119.4	621.1	2119.1
9	Phoenix	AZ	108.3	9.5	185001.1	119.4	1135.3	3873.9
10	Portland	ME	83.3	2.8	185001.1	119.4	331.8	1132.1
11	Sacramento	CA	97.6	6.6	185001.1	119.4	791.4	2700.4
12	Seattle	WA	81.6	2.3	185001.1	119.4	277.1	945.6

In addition to the heat flux through the structure due to the difference in temperature between the interior and exterior surfaces, the HVAC system also needs to be able to condition fresh air brought in from the exterior to the interior set point of 73°F. Using the 99% heating dry bulb and 1% cooling dry bulb temperatures from the host cities, the air density at that temperature was found using a psychrometric calculator. Then the difference to the internal set point was calculated. The air density and difference to the internal set point were multiplied together and then multiplied by the heat capacity of air, the number of air changes per hour, and the internal volume of the tiny house. The final value is the additional capacity (in BTU/hour) required by the HVAC system to condition six air changes in one hour. The summary of these calculations are shown in Table 28 to Table 31. The design tool calculates the capacity needed by the HVAC system for both summer and winter based on the user's answers.

Table 28: Summary of 20 ft Model Air Change Calculations, Winter Boundary Conditions

No.	City	State	99% Heating Dry Bulb Temp. [°F]	Air Density [lb/ft^3]	Difference to Int. Temp. [°F]	Heat Capacity [BTU/(lb×°F)]	Air Changes Per Hour	20 ft Model Int. Volume [ft^3]	Additional Capacity for Air Changes [BTU/hour]
1	Baltimore	MD	18.4	0.083	54.6	0.240	6	1675.3	10932.5
2	Denver	CO	5.3	0.085	67.7	0.240	6	1675.3	13931.5
3	Fargo	ND	-13.9	0.089	86.9	0.240	6	1675.3	18659.2
4	Grand Rapids	MI	7.3	0.085	65.7	0.240	6	1675.3	13472.4
5	Houston	TX	34.1	0.080	38.9	0.240	6	1675.3	7535.2
6	Kansas City	MO	7.5	0.085	65.5	0.240	6	1675.3	13415.5
7	Miami	FL	52.6	0.077	20.4	0.240	6	1675.3	3803.8
8	Nashville	TN	19.9	0.083	53.1	0.240	6	1675.3	10593.6
9	Phoenix	AZ	41.8	0.079	31.2	0.240	6	1675.3	5945.7
10	Portland	ME	5.2	0.085	67.8	0.240	6	1675.3	13952.1
11	Sacramento	CA	33.2	0.080	39.8	0.240	6	1675.3	7719.2
12	Seattle	WA	29.5	0.081	43.5	0.240	6	1675.3	8510.3

Table 29: Summary of 24 ft Model Air Change Calculations, Winter Boundary Conditions

No.	City	State	99% Heating Dry Bulb Temp. [°F]	Air Density [lb/ft^3]	Difference to Int. Temp. [°F]	Heat Capacity [BTU/(lb×°F)]	Air Changes Per Hour	24 ft Model Int. Volume [ft^3]	Additional Capacity for Air Changes [BTU/hour]
1	Baltimore	MD	18.4	0.083	54.6	0.240	6	1954.8	12756.1
2	Denver	CO	5.3	0.085	67.7	0.240	6	1954.8	16255.5
3	Fargo	ND	-13.9	0.089	86.9	0.240	6	1954.8	21771.8
4	Grand Rapids	MI	7.3	0.085	65.7	0.240	6	1954.8	15719.7
5	Houston	TX	34.1	0.080	38.9	0.240	6	1954.8	8792.2
6	Kansas City	MO	7.5	0.085	65.5	0.240	6	1954.8	15653.4
7	Miami	FL	52.6	0.077	20.4	0.240	6	1954.8	4438.3
8	Nashville	TN	19.9	0.083	53.1	0.240	6	1954.8	12360.8
9	Phoenix	AZ	41.8	0.079	31.2	0.240	6	1954.8	6937.5
10	Portland	ME	5.2	0.085	67.8	0.240	6	1954.8	16279.5
11	Sacramento	CA	33.2	0.080	39.8	0.240	6	1954.8	9006.8
12	Seattle	WA	29.5	0.081	43.5	0.240	6	1954.8	9929.9

Table 30: Summary of 20 ft Model Air Change Calculations, Summer Boundary Conditions

No.	City	State	1% Cooling Dry Bulb Temp. [°F]	Air Density [lb/ft^3]	Difference to Int. Temp. [°F]	Heat Capacity [BTU/(lb×°F)]	Air Changes Per Hour	20 ft Model Int. Volume [ft^3]	Additional Capacity for Air Changes [BTU/hour]
1	Baltimore	MD	91.4	0.071	18.4	0.240	6	1675.3	3168.7
2	Denver	CO	92.2	0.071	19.2	0.240	6	1675.3	3301.8
3	Fargo	ND	87	0.072	14	0.240	6	1675.3	2431.2
4	Grand Rapids	MI	86.5	0.072	13.5	0.240	6	1675.3	2347.7
5	Houston	TX	95.5	0.071	22.5	0.240	6	1675.3	3842.1
6	Kansas City	MO	92.7	0.071	19.7	0.240	6	1675.3	3383.0
7	Miami	FL	90.8	0.071	17.8	0.240	6	1675.3	3069.7
8	Nashville	TN	92.3	0.071	19.3	0.240	6	1675.3	3314.3
9	Phoenix	AZ	108.3	0.069	35.3	0.240	6	1675.3	5865.9
10	Portland	ME	83.3	0.073	10.3	0.240	6	1675.3	1803.6
11	Sacramento	CA	97.6	0.070	24.6	0.240	6	1675.3	4182.9
12	Seattle	WA	81.6	0.073	8.6	0.240	6	1675.3	1510.1

Table 31: Summary of 24 ft Model Air Change Calculations, Summer Boundary Conditions

No.	City	State	1% Cooling Dry Bulb Temp. [°F]	Air Density [lb/ft^3]	Difference to Int. Temp. [°F]	Heat Capacity [BTU/(lb×°F)]	Air Changes Per Hour	24 ft Model Int. Volume [ft^3]	Additional Capacity for Air Changes [BTU/hour]
1	Baltimore	MD	91.4	0.071	18.4	0.240	6	1954.8	3697.3
2	Denver	CO	92.2	0.071	19.2	0.240	6	1954.8	3852.6
3	Fargo	ND	87	0.072	14	0.240	6	1954.8	2836.8
4	Grand Rapids	MI	86.5	0.072	13.5	0.240	6	1954.8	2739.3
5	Houston	TX	95.5	0.071	22.5	0.240	6	1954.8	4483.1
6	Kansas City	MO	92.7	0.071	19.7	0.240	6	1954.8	3947.4
7	Miami	FL	90.8	0.071	17.8	0.240	6	1954.8	3581.7
8	Nashville	TN	92.3	0.071	19.3	0.240	6	1954.8	3867.2
9	Phoenix	AZ	108.3	0.069	35.3	0.240	6	1954.8	6844.4
10	Portland	ME	83.3	0.073	10.3	0.240	6	1954.8	2104.5
11	Sacramento	CA	97.6	0.070	24.6	0.240	6	1954.8	4880.7
12	Seattle	WA	81.6	0.073	8.6	0.240	6	1954.8	1762.0

Appendix XII: Tiny House Design Tool Input Questions and Response Rationale

When the design tool is started, the user will be asked a series of questions. They are shown below in Table 32. The trailer lengths were limited to 20 ft and 24 feet because those were the largest trailer lengths from Tiny House Basics, LLC with 2 axles that were able to stay under the 14,000 lb capacity limit with the given boundary conditions. The 12 location numbers correspond to the 12 host cities, which are also outlined in the design tool. The number of occupants was limited to two to reflect a couple. If two people were living in the tiny house and each person had a personal and business issued cell phone (smart phone) this would total four devices, which is why up to four cell phones (smart phones) were allotted. The same rationale was used for laptops. The number of pets was limited at two as each of the maximum two occupants may have a pet. Microwaves and washing machines are common household appliances, but not every user may want them, so they were made optional. It is only when the user answers "Yes" to the washing machine question that the number of loads of laundry question is asked. Up to four loads of laundry per week accounted for two loads of whites and two loads of colors per week. The number of showers per day took into account that each of the possible two occupants may take up to two showers per day because of exercising and their general morning routine. An option for zero showers per day was allowed to account for gym memberships and alternative shower solutions. As not all users may want a bathtub, that item was made optional. It is only when the user answers "Yes" to the bathtub question that the number of baths per week question is asked. The number of baths per week was limited to seven, which accounts for one bath per day. Finally, the number of windows was limited to six based on the impact they have the overall heat transfer rate.

Table 32: Summary of Tiny House Design Tool Input Questions and Available Responses

No.	Question	Available Responses											
1	What should the length of the living space (trailer) be?	20	24										
2	Where will the tiny house be located?	1	2	3	4	5	6	7	8	9	10	11	12
3	How many occupants will live in the tiny house?	1	2										
4	How many cell phones (smart phones) should be accounted for?	1	2	3	4								
5	How many laptops should be accounted for?	1	2	3	4								
6	How many pets should be accounted for?	0	1	2									
7	How many plants should be accounted for?	0	1	2	3	4							
8	Should a microwave be accounted for?	Yes	No										
9	Should a washing machine be accounted for?	Yes	No										
9a	How many loads of laundry per week should be accounted for?	1	2	3	4								
10	How many showers per day should be accounted for?	0	1	2	3	4							
11	Should a bathtub be accounted for?	Yes	No										
11a	How many baths per week should be accounted for?	1	2	3	4	5	6	7					
12	How many windows should be accounted for?	0	1	2	3	4	5	6					

Appendix XIII: Tiny House Design Tool Instructions and Example Printout

The instructions below (Figure 93 to Figure 99) can be used to install Scilab 5.5.2, and they can also be used to navigate the software until the design tool is running. The design tool outlines what answers are acceptable, and while it was coded to be as robust as possible, inputting special characters (e.g. $, %, *, etc.) may cause the design tool to abort or crash. Instructions for restarting the design tool is also shown below (Figure 100).

Figure 93: Tiny House Design Tool Instructions: Steps 1 & 2

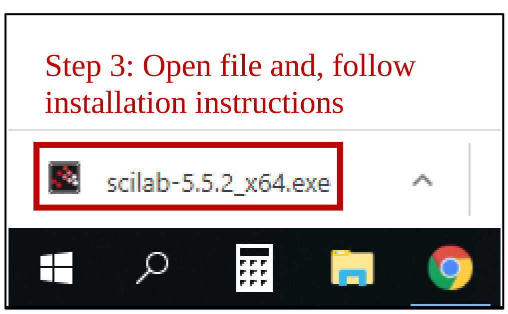
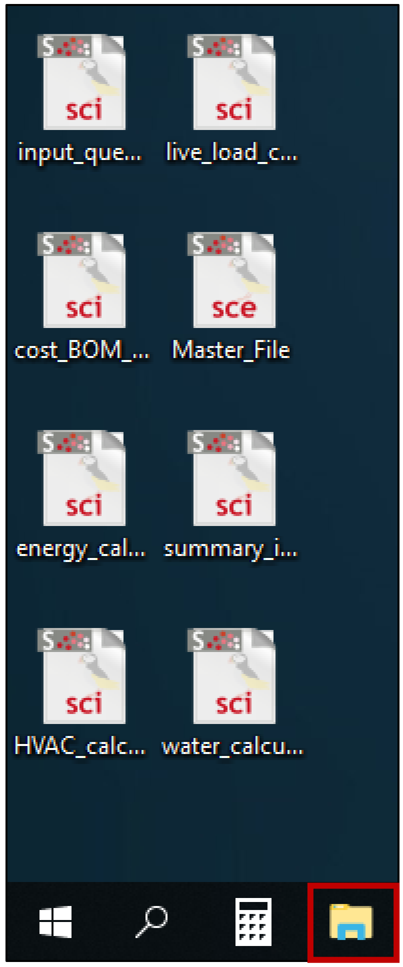
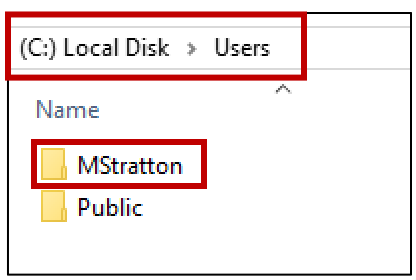


Figure 94: Tiny House Design Tool Instructions: Step 3



Step 4: Copy and paste the 8 Tiny House Design Tool files to the desktop and open windows explorer

Figure 95: Tiny House Design Tool Instructions: Step 4



Step 5: Navigate to the user folder and write down the username (this will be needed later).

Figure 96: Tiny House Design Tool Instructions: Step 5

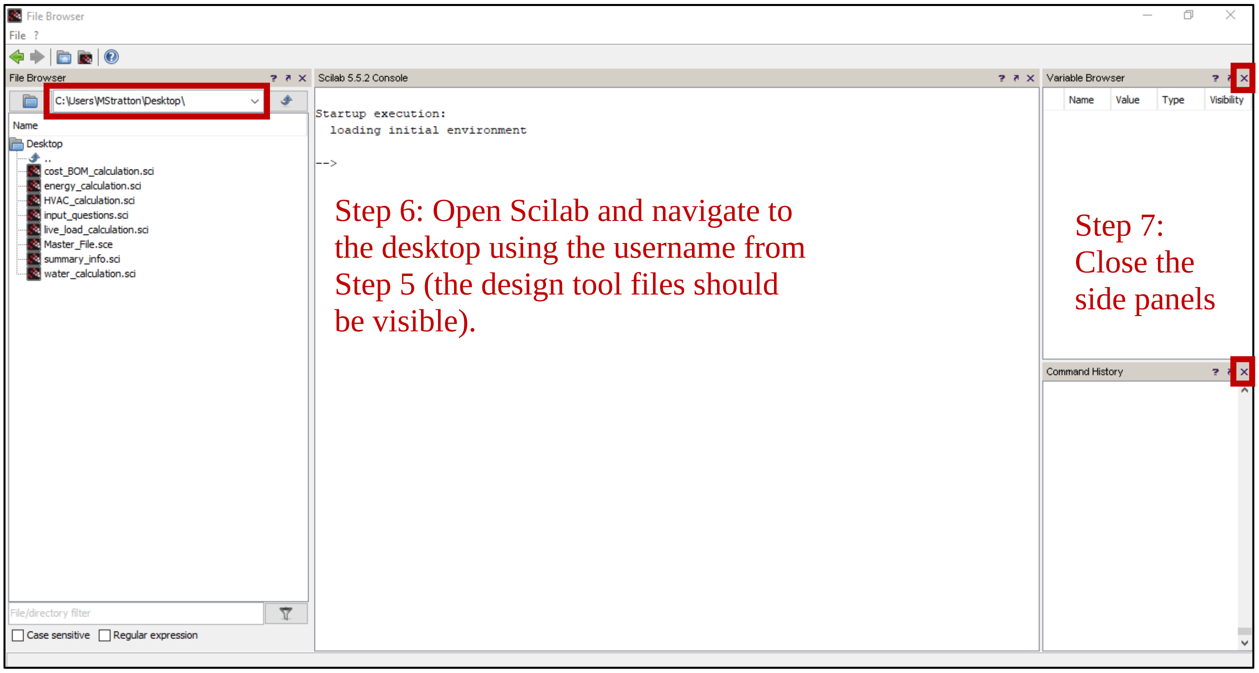


Figure 97: Tiny House Design Tool Instructions: Steps 6 & 7

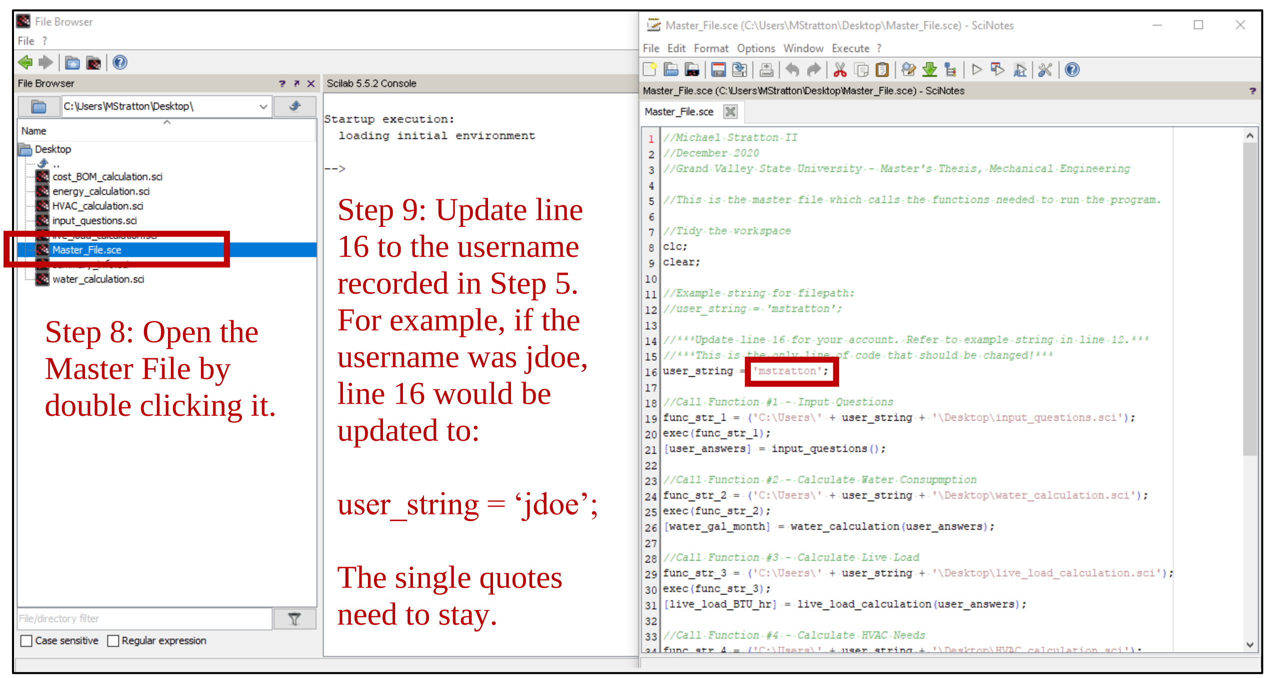


Figure 98: Tiny House Design Tool Instructions: Steps 8 & 9

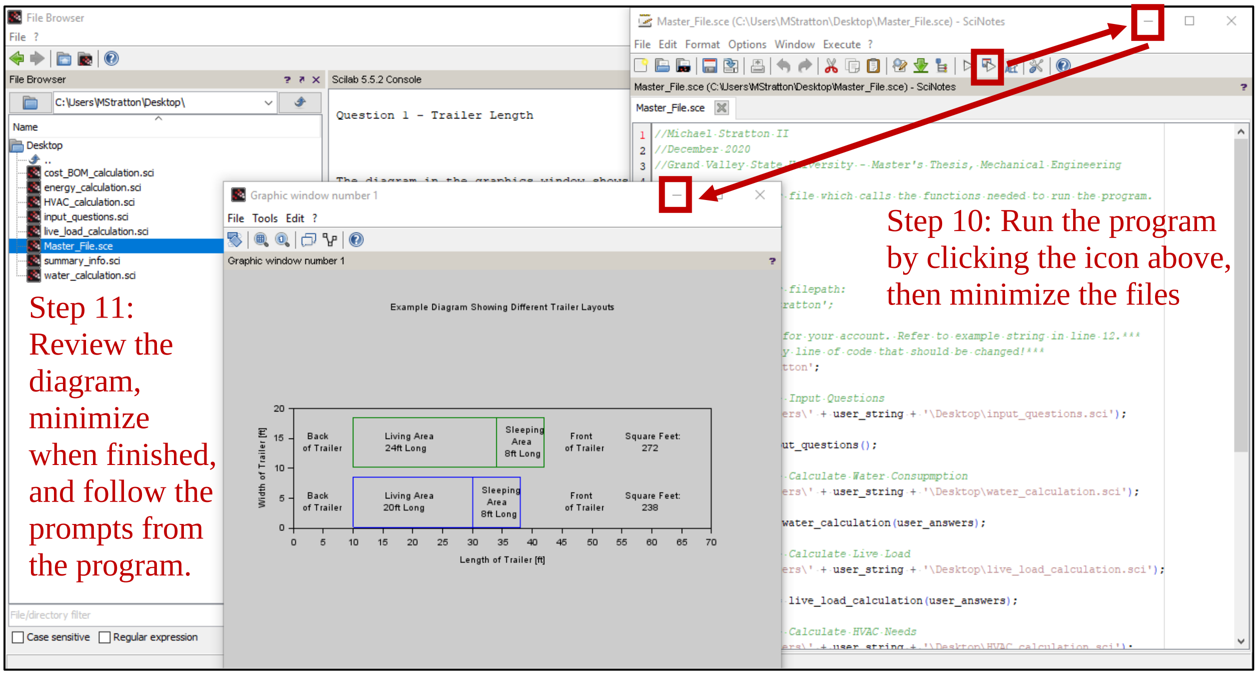


Figure 99: Tiny House Design Tool Instructions: Steps 10 & 11

```
 !--error 144
Undefined operation for the given operands.
check or define function %fptr_o_s for overloading.
at line     333 of function input_questions called by :
[user_answers] = input_questions();
at line      21 of exec file called by :
exec('C:\Users\MStratton\Desktop\Master_File.sce', -1)

-->
```

If a similar error appears, the program can be restarted by checking all steps were followed correctly, and if they were, Step 10 can be repeated.

Figure 100: Tiny House Design Tool Instructions: Restart the Program

The text on the following page shows an example printout from the design tool when Grand Rapids, MI and the smallest available values are used. The text can be selected and copied into a text editor for future reference, or it can be printed into a PDF document using the print function, shown below in Figure 101.

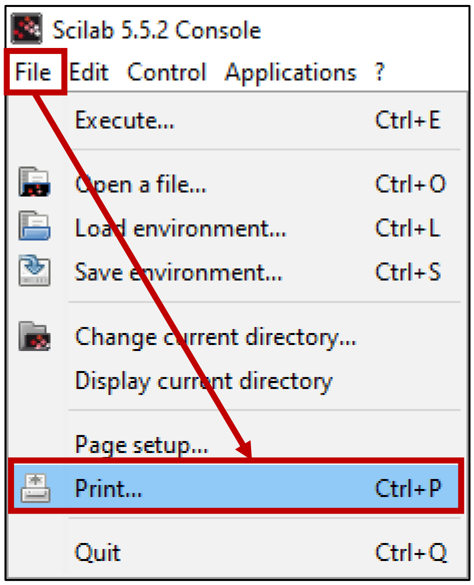


Figure 101: Tiny House Design Tool Instructions: Print Results

All calculations are finished, and the summary for your tiny house

is shown on the following lines.

The length of trailer chosen (excluding the gooseneck) in feet is:

20.

The city chosen for the tiny house to reside in is:

Grand Rapids, MI

The number of occupants to account for is:

1.

The number of cell phones to account for is:

1.

The number of laptops to account for is:

1.

The number of pets to account for is:

0.

The number of plants to account for is:

0.

Should a microwave be accounted for?

No.

Should a washing machine be accounted for?

No.

The number of showers per day to account for is:

0.

Should a bathtub be accounted for?

No.

The number of windows to account for is:

0.

The approximate gallons per month of water usage is:

487.5

Compared to the national average of 3,000 gallons of water used

per month, per person, your usage is:

83.75

percent lower.

The 99% dry bulb heating temperature in degrees Fahrenheit

for your chosen location is:

7.3

The needed heating capacity in BTU/hr for your location/options is:

19978.6

The 1% dry bulb cooling temperature in degrees Fahrenheit

for your chosen location is:

86.5

The needed cooling capacity in BTU/hr for your location/options is:

3684.6

The approximate kWh per month energy usage is:

436.53

Compared to the national average of 914.3 kWh of energy used

per month, your usage is:

52.255277

percent lower.

The amount, in inches, of 2 x 12s needed is:

960.

The amount, in inches, of 2 x 4s needed is:

8902.81

The amount, in inches, of 2 x 6s needed is:

4501.52

The amount, in inches, of 2 x 8s needed is:

1736.

The amount, in pieces, of bathroom sinks needed is:

1.

The amount, in square feet, of bathroom tile needed is:

31.84

The amount, in square feet, of cotton/denim insulation needed is:

1221.64

The amount, in pieces, of kitchen counters needed is:

1.

The amount, in pieces, of kitchen sinks needed is:

1.

The amount, in sqft, of metal roof needed is:

228.42

The amount, in pieces, of HVAC mini-splits needed is:

1.

The amount, in sqft, of plywood needed is:

2309.51

The amount, in pieces, of queen mattresses needed is:

1.

The amount, in pieces, of refrigerators needed is:

1.

The amount, in square feet, of shiplap needed is:

537.19

The amount, in square feet, of siding needed is:

612.32

The amount, in pieces, of stove/ovens needed is:

1.

The amount, in pieces, of toilets needed is:

1.

The amount, in pieces, of trailers needed is:

1.

The amount, in pieces, of bathtubs needed is:

0.

The amount, in pieces, of microwaves needed is:

0.

The amount, in pieces, of washer/dryer combination needed is:

0.

The amount, in pieces, of windows needed is:

0.

The approximate cost in USD for the shell of the tiny house is:

15401.39

Thank you for using this tool. Please scroll to the top of the

terminal window and enjoy building your tiny house!

All code was written by Michael Stratton II

Appendix XIV: Printout of Scilab Code: Master File

```
//Michael Stratton II
//December 2020
//Grand Valley State University - Master's Thesis, Mechanical Engineering

//This is the master file which calls the functions needed to run the program.

//Tidy the workspace
clc;
clear;

//Example string for filepath:
//user_string = 'mstratton';

//***Update line 16 for your account. Refer to example string in line 12.***
//***This is the only line of code that should be changed!***
user_string = ‘mstratton’;

//Call Function #1 - Input Questions
func_str_1 = ('C:\Users\' + user_string + '\Desktop\input_questions.sci');
exec(func_str_1);
[user_answers] = input_questions();

//Call Function #2 - Calculate Water Consupmption
func_str_2 = ('C:\Users\' + user_string + '\Desktop\water_calculation.sci');
exec(func_str_2);
[water_gal_month] = water_calculation(user_answers);

//Call Function #3 - Calculate Live Load
func_str_3 = ('C:\Users\' + user_string + '\Desktop\live_load_calculation.sci');
exec(func_str_3);
[live_load_BTU_hr] = live_load_calculation(user_answers);

//Call Function #4 - Calculate HVAC Needs
func_str_4 = ('C:\Users\' + user_string + '\Desktop\HVAC_calculation.sci');
exec(func_str_4);
[HVAC_data] = HVAC_calculation(user_answers,live_load_BTU_hr)

//Call Function #5 - Calculate Energy Consumption
func_str_5 = ('C:\Users\' + user_string + '\Desktop\energy_calculation.sci');
exec(func_str_5);
[energy_kWh_month] = energy_calculation(user_answers);

//Call Function #6 - Calculate BOM and Approximate Cost
func_str_6 = ('C:\Users\' + user_string + '\Desktop\cost_BOM_calculation.sci');
exec(func_str_6);
[cost_and_BOM] = cost_BOM_calculation(user_answers);

//Call Function #7 - Summary and Closing
func_str_7 = ('C:\Users\' + user_string + '\Desktop\summary_info.sci');
exec(func_str_7);
[tiny_house_summary] = summary_info(user_answers, water_gal_month, HVAC_data, energy_kWh_month,
cost_and_BOM);
```

Appendix XV: Printout of Scilab Code: Input Questions

```
function [user_answers]=input_questions()

//This function asks the user numerous questions and returns the answers
//to those questions which will be used to properly size the tiny house
//and the various associated equipment.

//tidy the workspace and clear previous variables
clc;

testA = %f; //Booleans used in the subsequent loops to obtain correct conditions
testB = %f;

//********Question 1**********************************************************
//********Question 1**********************************************************
//********Question 1**********************************************************

  clc;
  disp('Question 1 - Trailer Length');
  disp(' ');

  xcords = [10,30,38,38,30,30,30,10,10];
  ycords = [0.2,0.2,0.2,8.5,8.5,0.2,8.5,8.5,0.2];
  xcords2 = [10,34,42,42,34,34,34,10,10];
  ycords2 = [10.2,10.2,10.2,18.5,18.5,10.2,18.5,18.5,10.2];
  figure(1);
  plot(xcords,ycords,xcords2,ycords2);
  title('Example Diagram Showing Different Trailer Layouts');
  isoview(0,70,0,20); //forces square axis on figure
  xlabel('Length of Trailer [ft]');
  ylabel('Width of Trailer [ft]');
  xstring(2,4,'Back')
  xstring(1,2,'of Trailer');
  xstring(46,4,'Front');
  xstring(45,2,'of Trailer');
  xstring(55,4,'Square Feet:');
  xstring(58,2,'238');
  xstring(15,4,'Living Area');
  xstring(14.7,2,'20ft Long');
  xstring(31,5,'Sleeping');
  xstring(32,3,'Area');
  xstring(31,1,'8ft Long');

  xstring(2,14,'Back')
  xstring(1,12,'of Trailer');
  xstring(46,14,'Front');
  xstring(45,12,'of Trailer');
  xstring(55,14,'Square Feet:');
  xstring(58,12,'272');
  xstring(15,14,'Living Area');
  xstring(15,12,'24ft Long');
  xstring(35,15,'Sleeping');
  xstring(36,13,'Area')
  xstring(35,11,'8ft Long');

  h = gca();
  h.data_bounds = [0,0; 70,20]; //set limits for x and y axes

  disp('The diagram in the graphics window shows how the trailer will be');
```

```
disp('divided. The upper 8ft gooseneck section will be devoted to a');
disp('sleeping area, and the lower section will be devoted to the living');
disp('space. The number entered will reflect the living space. (Graphics');
disp('window shown separately, and it will close automatically.)');
disp('');
disp('Press `Enter` after every entry for all questions.');
disp('');

while testA == %f //will keep looping until information is correct
  disp('What should the length of the living space (trailer) be?');
  disp('Input must be 20 or 24 only (feet).');
  disp('');
  trailer_length_ft = input('');
  disp(' ');
  if trailer_length_ft == 20 | ...
    trailer_length_ft == 24
    testA = %t;
    testB = %f;
    while testB == %f
      clc;
      disp('You have input: ');
      disp('');
      disp(trailer_length_ft);
      disp('');
      disp('as the living space length.');
      disp('Is this correct? (Please type `Yes` or `No`)');
      disp('');
      y_or_n = string(input(' ','s'));
      if y_or_n == 'Yes' | ...
          y_or_n == 'YES' | ...
          y_or_n == 'yes' | ... //accepts multiple
          y_or_n == 'Y' | ...  //yes answers
          y_or_n == 'y'
        testB = %t; //returns true if input was correct
      elseif y_or_n == 'No' | ...
          y_or_n == 'NO' | ...
          y_or_n == 'no' | ... //accepts multiple
          y_or_n == 'N' | ... //no answers
          y_or_n == 'n'
        disp('Code will return to prompt for an updated input.');

        tic;
        sleep(1500); //1.5 seconds
        toc;

        clc;
        testB = %t; //break out of inner while loop and
                    //return to outer while loop
        testA = %f;
        //will reprompt if user wants to change answer
      else
        disp('Error, please type `Yes` or `No`.');
        disp(' ');
        //reprompts for an input if anything else is entered
      end
    end
  else
    clc;
    disp('Error - Incorrect Input');
    disp(' ');
  end
```

```
    end
    testA = %f; //Reset booleans as same ones are used in following questions
    testB = %f;

    close(figure(1)); //closes the trailer figure to clear up space
                //for the next question

//********Question 2**********************************************************
//********Question 2**********************************************************
//********Question 2**********************************************************

    clc;
    disp('Question 2 - Location of the Tiny House');
    disp(' ');

    disp('This question is being asked to determine where the tiny house');
    disp('will be located so the associated regional conditions can be');
    disp('accounted for. The numbers 1 - 12 correspond to the following');
    disp('cities. If the tiny house will be placed in a different city,');
    disp('enter the number corresponding to a city with a similar climate.');
    disp('');

    while testA == %f //will keep looping until information is correct
      disp('Where will the tiny house be located?');
      disp('Input must be either:');
      disp(' 1 (Baltimore, MD),      7 (Miami, FL),');
      disp(' 2 (Denver, CO),         8 (Nashville, TN),');
      disp(' 3 (Fargo, ND),          9 (Phoenix, AZ),');
      disp(' 4 (Grand Rapids, MI),   10 (Portland, ME),');
      disp(' 5 (Houston, TX),        11 (Sacramento, CA), or');
      disp(' 6 (Kansas City, MO),    12 (Seattle, WA)');
      disp('');
      city_num = input('');
      disp(' ');
      if city_num == 1 | ...
        city_num == 2 | ...
        city_num == 3 | ...
        city_num == 4 | ...
        city_num == 5 | ...
        city_num == 6 | ...
        city_num == 7 | ...
        city_num == 8 | ...
        city_num == 9 | ...
        city_num == 10 | ...
        city_num == 11 | ...
        city_num == 12
        testA = %t;
        testB = %f;
        while testB == %f
          clc;
          disp('You have input: ');
          disp('');
          if city_num == 1
            disp("(Baltimore, MD)  ", city_num);
          elseif city_num == 2
            disp("(Denver, CO)  ", city_num);
          elseif city_num == 3
            disp("(Fargo, ND)  ", city_num);
          elseif city_num == 4
            disp("(Grand Rapids, MI)  ", city_num);
          elseif city_num == 5
```

```
            disp("(Houston, TX)  ", city_num);
          elseif city_num == 6
            disp("(Kansas City, MO)  ", city_num);
          elseif city_num == 7
            disp("(Miami, FL)  ", city_num);
          elseif city_num == 8
            disp("(Nashville, TN)  ", city_num);
          elseif city_num == 9
            disp("(Phoenix, AZ)  ", city_num);
          elseif city_num == 10
            disp("(Portland, ME)  ", city_num);
          elseif city_num == 11
            disp("(Sacramento, CA)  ", city_num);
          else
            disp("(Seattle, WA)  ", city_num);
          end
          disp('');
          disp('as the city number.');
          disp('Is this correct? (Please type `Yes` or `No`)');
          disp('');
          y_or_n = string(input(' ','s'));
          if y_or_n == 'Yes' | ...
                y_or_n == 'YES' | ...
                y_or_n == 'yes' | ... //accepts multiple
                y_or_n == 'Y' | ...   //yes answers
                y_or_n == 'y'
            testB = %t; //returns true if input was correct
          elseif y_or_n == 'No' | ...
                y_or_n == 'NO' | ...
                y_or_n == 'no' | ... //accepts multiple
                y_or_n == 'N' | ...  //no answers
                y_or_n == 'n'
            disp('Code will return to prompt for an updated input.');

            tic;
            sleep(1500); //1.5 seconds
            toc;

            clc;
            testB = %t; //break out of inner while loop and
                        //return to outer while loop
            testA = %f;
            //will reprompt if user wants to change answer
          else
            disp('Error, please type `Yes` or `No`.');
            disp(' ');
            //reprompts for an input if anything else is entered
          end
        end
      else
        clc;
        disp('Error - Incorrect Input');
        disp(' ');
      end
    end
    testA = %f; //Reset booleans as same ones are used in following questions
    testB = %f;

//********Question 3***********************************************************
//********Question 3***********************************************************
//********Question 3***********************************************************
```

```
clc;
disp('Question 3 - Number of Occupants');
disp(' ');

disp('This question is being asked to determine how many people will be');
disp('occupying the space. People give off heat and consume water, so');
disp('knowing the number of occupants will help make the HVAC and water');
disp('calculations more accurate.');
disp('');

while testA == %f //will keep looping until information is correct
  disp('How many occupants will live in the tiny house?');
  disp('Input must be either 1 or 2.');
  disp('');
  occupant_num = input('');
  disp(' ');
  if occupant_num == 1 | ...
    occupant_num == 2
    testA = %t;
    testB = %f;
    while testB == %f
      clc;
      disp('You have input: ');
      disp('');
      disp(occupant_num);
      disp('');
      disp('as the number of occupants.');
      disp('Is this correct? (Please type `Yes` or `No`)');
      disp('');
      y_or_n = string(input(' ','s'));
      if y_or_n == 'Yes' | ...
          y_or_n == 'YES' | ...
          y_or_n == 'yes' | ... //accepts multiple
          y_or_n == 'Y' | ...  //yes answers
          y_or_n == 'y'
        testB = %t; //returns true if input was correct
      elseif y_or_n == 'No' | ...
          y_or_n == 'NO' | ...
          y_or_n == 'no' | ... //accepts multiple
          y_or_n == 'N' | ...  //no answers
          y_or_n == 'n'
        disp('Code will return to prompt for an updated input.');

        tic;
        sleep(1500); //1.5 seconds
        toc;

        clc;
        testB = %t; //break out of inner while loop and
                    //return to outer while loop
        testA = %f;
        //will reprompt if user wants to change answer
      else
        disp('Error, please type `Yes` or `No`.');
        disp(' ');
        //reprompts for an input if anything else is entered
      end
    end
  else
    clc;
```

```
        disp('Error - Incorrect Input');
        disp(' ');
      end
    end
    testA = %f; //Reset booleans as same ones are used in following questions
    testB = %f;

//********Question 4**********************************************************
//********Question 4**********************************************************
//********Question 4**********************************************************

    clc;
    disp('Question 4 - Number of Cell Phones (Smart Phones)');
    disp(' ');

    disp('This question is being asked to determine how many devices will be');
    disp('in the space. Each device gives off heat and consumes energy, so');
    disp('knowing the number of overall devices will help make the HVAC and');
    disp('energy calculations more accurate.');
    disp('');

    while testA == %f //will keep looping until information is correct
      disp('How many cell phones (smart phones) should be accounted for?');
      disp('(for all occupants)');
      disp('Input must be either 1, 2, 3, or 4.');
      disp('');
      cell_phone_num = input('');
      disp(' ');
      if cell_phone_num == 1 | ...
        cell_phone_num == 2 | ...
        cell_phone_num == 3 | ...
        cell_phone_num == 4
        testA = %t;
        testB = %f;
        while testB == %f
          clc;
          disp('You have input: ');
          disp('');
          disp(cell_phone_num);
          disp('');
          disp('as the number of cell phones (smart phones).');
          disp('Is this correct? (Please type `Yes` or `No`)');
          disp('');
          y_or_n = string(input(' ','s'));
          if y_or_n == 'Yes' | ...
              y_or_n == 'YES' | ...
              y_or_n == 'yes' | ... //accepts multiple
              y_or_n == 'Y' | ...  //yes answers
              y_or_n == 'y'
            testB = %t; //returns true if input was correct
          elseif y_or_n == 'No' | ...
              y_or_n == 'NO' | ...
              y_or_n == 'no' | ... //accepts multiple
              y_or_n == 'N' | ...  //no answers
              y_or_n == 'n'
            disp('Code will return to prompt for an updated input.');

            tic;
            sleep(1500); //1.5 seconds
            toc;
```

```
            clc;
            testB = %t; //break out of inner while loop and
                        //return to outer while loop
            testA = %f;
            //will reprompt if user wants to change answer
          else
            disp('Error, please type `Yes` or `No`.');
            disp(' ');
            //reprompts for an input if anything else is entered
          end
        end
      else
        clc;
        disp('Error - Incorrect Input');
        disp(' ');
      end
    end
    testA = %f; //Reset booleans as same ones are used in following questions
    testB = %f;

//********Question 5************************************************************
//********Question 5************************************************************
//********Question 5************************************************************

    clc;
    disp('Question 5 - Number of Laptops');
    disp(' ');

    disp('This question is being asked to determine how many laptops will be');
    disp('in the space. Laptops gives off heat and consumes energy, so');
    disp('knowing the number of overall laptops will help make the HVAC and');
    disp('energy calculations more accurate.');
    disp('');

    while testA == %f //will keep looping until information is correct
      disp('How many laptops should be accounted for?');
      disp('(for all occupants)');
      disp('Input must be either 1, 2, 3, or 4.');
      disp('');
      laptop_num = input('');
      disp(' ');
      if laptop_num == 1 | ...
        laptop_num == 2 | ...
        laptop_num == 3 | ...
        laptop_num == 4
        testA = %t;
        testB = %f;
        while testB == %f
          clc;
          disp('You have input: ');
          disp('');
          disp(laptop_num);
          disp('');
          disp('as the number of laptops.');
          disp('Is this correct? (Please type `Yes` or `No`)');
          disp('');
          y_or_n = string(input(' ','s'));
          if y_or_n == 'Yes' | ...
              y_or_n == 'YES' | ...
              y_or_n == 'yes' | ... //accepts multiple
              y_or_n == 'Y' | ...   //yes answers
```

```
            y_or_n == 'y'
         testB = %t; //returns true if input was correct
      elseif y_or_n == 'No' | ...
            y_or_n == 'NO' | ...
            y_or_n == 'no' | ... //accepts multiple
            y_or_n == 'N' | ...  //no answers
            y_or_n == 'n'
         disp('Code will return to prompt for an updated input.');

         tic;
         sleep(1500); //1.5 seconds
         toc;

         clc;
         testB = %t; //break out of inner while loop and
                     //return to outer while loop
         testA = %f;
         //will reprompt if user wants to change answer
      else
         disp('Error, please type `Yes` or `No`.');
         disp(' ');
         //reprompts for an input if anything else is entered
      end
    end
  else
    clc;
    disp('Error - Incorrect Input');
    disp(' ');
  end
end
testA = %f; //Reset booleans as same ones are used in following questions
testB = %f;

//********Question 6************************************************************
//********Question 6************************************************************
//********Question 6************************************************************

clc;
disp('Question 6 - Number of Pets');
disp(' ');

disp('This question is being asked to determine how many pets will be');
disp('occupying the space. Pets, like people, give off heat and consume');
disp('water, so knowing the number of pets will help make the HVAC and');
disp('water calculations more accurate.');
disp('');

while testA == %f //will keep looping until information is correct
  disp('How many pets should be accounted for?');
  disp('Input must be either 0, 1, or 2');
  disp('');
  pet_num = input('');
  disp(' ');
  if pet_num == 0 | ...
    pet_num == 1 | ...
    pet_num == 2
    testA = %t;
    testB = %f;
    while testB == %f
      clc;
      disp('You have input: ');
```

```
            disp('');
            disp(pet_num);
            disp('');
            disp('as the number of pets.');
            disp('Is this correct? (Please type `Yes` or `No`)');
            disp('');
            y_or_n = string(input(' ','s'));
            if y_or_n == 'Yes' | ...
                y_or_n == 'YES' | ...
                y_or_n == 'yes' | ... //accepts multiple
                y_or_n == 'Y' | ...  //yes answers
                y_or_n == 'y'
              testB = %t; //returns true if input was correct
            elseif y_or_n == 'No' | ...
                y_or_n == 'NO' | ...
                y_or_n == 'no' | ... //accepts multiple
                y_or_n == 'N' | ...  //no answers
                y_or_n == 'n'
              disp('Code will return to prompt for an updated input.');

              tic;
              sleep(1500); //1.5 seconds
              toc;

              clc;
              testB = %t; //break out of inner while loop and
                          //return to outer while loop
              testA = %f;
              //will reprompt if user wants to change answer
            else
              disp('Error, please type `Yes` or `No`.');
              disp(' ');
              //reprompts for an input if anything else is entered
            end
          end
        else
          clc;
          disp('Error - Incorrect Input');
          disp(' ');
        end
      end
      testA = %f; //Reset booleans as same ones are used in following questions
      testB = %f;

//********Question 7**********************************************************
//********Question 7**********************************************************
//********Question 7**********************************************************

      clc;
      disp('Question 7 - Number of Plants');
      disp(' ');

      disp('This question is being asked to determine how many plants will be');
      disp('occupying the space. Plants consume water, so knowing the number');
      disp('of plants will help make the water calculations more accurate.');
      disp('');

      while testA == %f //will keep looping until information is correct
        disp('How many plants should be accounted for?');
        disp('Input must be either 0, 1, 2, 3, or 4');
        disp('');
```

```
      plant_num = input('');
      disp(' ');
      if plant_num == 0 | ...
         plant_num == 1 | ...
         plant_num == 2 | ...
         plant_num == 3 | ...
         plant_num == 4
         testA = %t;
         testB = %f;
         while testB == %f
            clc;
            disp('You have input: ');
            disp('');
            disp(plant_num);
            disp('');
            disp('as the number of plants.');
            disp('Is this correct? (Please type `Yes` or `No`)');
            disp('');
            y_or_n = string(input(' ','s'));
            if y_or_n == 'Yes' | ...
                  y_or_n == 'YES' | ...
                  y_or_n == 'yes' | ... //accepts multiple
                  y_or_n == 'Y' | ...  //yes answers
                  y_or_n == 'y'
               testB = %t; //returns true if input was correct
            elseif y_or_n == 'No' | ...
                  y_or_n == 'NO' | ...
                  y_or_n == 'no' | ... //accepts multiple
                  y_or_n == 'N' | ...  //no answers
                  y_or_n == 'n'
               disp('Code will return to prompt for an updated input.');

               tic;
               sleep(1500); //1.5 seconds
               toc;

               clc;
               testB = %t; //break out of inner while loop and
                           //return to outer while loop
               testA = %f;
               //will reprompt if user wants to change answer
            else
               disp('Error, please type `Yes` or `No`.');
               disp(' ');
               //reprompts for an input if anything else is entered
            end
         end
      else
         clc;
         disp('Error - Incorrect Input');
         disp(' ');
      end
   end
   testA = %f; //Reset booleans as same ones are used in following questions
   testB = %f;


//********Question 8**********************************************************
//********Question 8**********************************************************
//********Question 8**********************************************************
```

```
clc;
disp('Question 8 - Is a Microwave Desired?');
disp(' ');

disp('This question is being asked to determine if the occupant(s)');
disp('want a microwave in the tiny house. Microwaves give off heat and');
disp('consume energy, so knowing if one is wanted or now will help make');
disp('the HVAC and energy calculations more accurate.');
disp('');

while testA == %f //will keep looping until information is correct
  disp('Should a microwave be accounted for?');
  disp('Input must be either `Yes` or `No`');
  disp('');
  microwave_y_or_n = string(input(' ','s'));
  disp(' ');
  if microwave_y_or_n == 'Yes' | ...
        microwave_y_or_n == 'YES' | ...
        microwave_y_or_n == 'yes' | ... //accepts multiple
        microwave_y_or_n == 'Y' | ...   //yes answers
        microwave_y_or_n == 'y'
     microwave_y_or_n = 'Yes';
     microwave_num = 1;
     testA = %t;
     testB = %f;
     while testB == %f
        clc;
        disp('You have input: ');
        disp('');
        disp(microwave_y_or_n);
        disp('');
        disp('as wanting a microwave.');
        disp('Is this correct? (Please type `Yes` or `No`)');
        disp('');
        y_or_n = string(input(' ','s'));
        if y_or_n == 'Yes' | ...
              y_or_n == 'YES' | ...
              y_or_n == 'yes' | ... //accepts multiple
              y_or_n == 'Y' | ...   //yes answers
              y_or_n == 'y'
           testB = %t; //returns true if input was correct
        elseif y_or_n == 'No' | ...
              y_or_n == 'NO' | ...
              y_or_n == 'no' | ... //accepts multiple
              y_or_n == 'N' | ... //no answers
              y_or_n == 'n'
           disp('Code will return to prompt for an updated input.');

           tic;
           sleep(1500); //1.5 seconds
           toc;

           clc;
           testB = %t; //break out of inner while loop and
                       //return to outer while loop
           testA = %f;
           //will reprompt if user wants to change answer
        else
           disp('Error, please type `Yes` or `No`.');
           disp(' ');
           //reprompts for an input if anything else is entered
```

```
        end
      end
    elseif microwave_y_or_n == 'No' | ...
        microwave_y_or_n == 'NO' | ...
        microwave_y_or_n == 'no' | ... //accepts multiple
        microwave_y_or_n == 'N' | ...  //no answers
        microwave_y_or_n == 'n'
      microwave_y_or_n = 'No';
      microwave_num = 0;
      testA = %t;
      testB = %f;
      while testB == %f
        clc;
        disp('You have input: ');
        disp('');
        disp(microwave_y_or_n);
        disp('');
        disp('as wanting a microwave.');
        disp('Is this correct? (Please type `Yes` or `No`)');
        disp('');
        y_or_n = string(input(' ','s'));
        if y_or_n == 'Yes' | ...
            y_or_n == 'YES' | ...
            y_or_n == 'yes' | ... //accepts multiple
            y_or_n == 'Y' | ...   //yes answers
            y_or_n == 'y'
          testB = %t; //returns true if input was correct
        elseif y_or_n == 'No' | ...
            y_or_n == 'NO' | ...
            y_or_n == 'no' | ... //accepts multiple
            y_or_n == 'N' | ...  //no answers
            y_or_n == 'n'
          disp('Code will return to prompt for an updated input.');

          tic;
          sleep(1500); //1.5 seconds
          toc;

          clc;
          testB = %t; //break out of inner while loop and
                      //return to outer while loop
          testA = %f;
          //will reprompt if user wants to change answer
        else
          disp('Error, please type `Yes` or `No`.');
          disp(' ');
          //reprompts for an input if anything else is entered
        end
      end
    else
      clc;
      disp('Error - Incorrect Input');
      disp(' ');
    end
  end
  testA = %f; //Reset booleans as same ones are used in following questions
  testB = %f;

//********Question 9**********************************************************
//********Question 9**********************************************************
//********Question 9**********************************************************
```

```
clc;
disp('Question 9 - Is a Washing Machine Desired?');
disp(' ');

disp('This question is being asked to determine if the occupant(s)');
disp('want a washing machine in the tiny house. Washing machines consume');
disp('water and energy, so knowing if one is wanted or now will help make');
disp('the water and energy calculations more accurate.');
disp('');

while testA == %f //will keep looping until information is correct
  disp('Should a washing machine be accounted for?');
  disp('Input must be either `Yes` or `No`');
  disp('');
  washing_mach_y_or_n = string(input(' ','s'));
  disp(' ');
  if washing_mach_y_or_n == 'Yes' | ...
      washing_mach_y_or_n == 'YES' | ...
      washing_mach_y_or_n == 'yes' | ... //accepts multiple
      washing_mach_y_or_n == 'Y' | ...  //yes answers
      washing_mach_y_or_n == 'y'
    washing_mach_y_or_n = 'Yes';
    washing_mach_num = 1;
    testA = %t;
    testB = %f;
    while testB == %f
      clc;
      disp('You have input: ');
      disp('');
      disp(washing_mach_y_or_n);
      disp('');
      disp('as wanting a washing machine.');
      disp('Is this correct? (Please type `Yes` or `No`)');
      disp('');
      y_or_n = string(input(' ','s'));
      if y_or_n == 'Yes' | ...
          y_or_n == 'YES' | ...
          y_or_n == 'yes' | ... //accepts multiple
          y_or_n == 'Y' | ...  //yes answers
          y_or_n == 'y'
        testB = %t; //returns true if input was correct
      elseif y_or_n == 'No' | ...
          y_or_n == 'NO' | ...
          y_or_n == 'no' | ... //accepts multiple
          y_or_n == 'N' | ...  //no answers
          y_or_n == 'n'
        disp('Code will return to prompt for an updated input.');

        tic;
        sleep(1500); //1.5 seconds
        toc;

        clc;
        testB = %t; //break out of inner while loop and
                    //return to outer while loop
        testA = %f;
        //will reprompt if user wants to change answer
      else
        disp('Error, please type `Yes` or `No`.');
        disp(' ');
```

```
            //reprompts for an input if anything else is entered
          end
        end
      elseif washing_mach_y_or_n == 'No' | ...
          washing_mach_y_or_n == 'NO' | ...
          washing_mach_y_or_n == 'no' | ... //accepts multiple
          washing_mach_y_or_n == 'N' | ...  //no answers
          washing_mach_y_or_n == 'n'
        washing_mach_y_or_n = 'No';
        washing_mach_num = 0;
        testA = %t;
        testB = %f;
        while testB == %f
          clc;
          disp('You have input: ');
          disp('');
          disp(washing_mach_y_or_n);
          disp('');
          disp('as wanting a washing machine.');
          disp('Is this correct? (Please type `Yes` or `No`)');
          disp('');
          y_or_n = string(input(' ','s'));
          if y_or_n == 'Yes' | ...
              y_or_n == 'YES' | ...
              y_or_n == 'yes' | ... //accepts multiple
              y_or_n == 'Y' | ...   //yes answers
              y_or_n == 'y'
            testB = %t; //returns true if input was correct
          elseif y_or_n == 'No' | ...
              y_or_n == 'NO' | ...
              y_or_n == 'no' | ... //accepts multiple
              y_or_n == 'N' | ...  //no answers
              y_or_n == 'n'
            disp('Code will return to prompt for an updated input.');

            tic;
            sleep(1500); //1.5 seconds
            toc;

            clc;
            testB = %t; //break out of inner while loop and
                        //return to outer while loop
            testA = %f;
            //will reprompt if user wants to change answer
          else
            disp('Error, please type `Yes` or `No`.');
            disp(' ');
            //reprompts for an input if anything else is entered
          end
        end
      else
        clc;
        disp('Error - Incorrect Input');
        disp(' ');
      end
    end
    testA = %f; //Reset booleans as same ones are used in following questions
    testB = %f;

//********Question 9a*********************************************************
//********Question 9a*********************************************************
```

```
//********Question 9a**********************************************************

if washing_mach_num == 1;

   clc;
   disp('Question 9a - Number of Laundry Loads Per Week');
   disp(' ');

   disp('This question is being asked to determine how many loads of laundry');
   disp('should be accounted for per week. Laundry requires water so');
   disp('knowing the number of loads per week will help make the water');
   disp('calculations more accurate.');
   disp('');

   while testA == %f //will keep looping until information is correct
      disp('How many loads of laundry per week should be accounted for?');
      disp('(for all occupants)');
      disp('Input must be either 1, 2, 3, or 4.');
      disp('');
      laundry_num = input('');
      disp(' ');
      if laundry_num == 1 | ...
         laundry_num == 2 | ...
         laundry_num == 3 | ...
         laundry_num == 4
         testA = %t;
         testB = %f;
         while testB == %f
            clc;
            disp('You have input: ');
            disp('');
            disp(laundry_num);
            disp('');
            disp('as the number of loads of laundry per week.');
            disp('Is this correct? (Please type `Yes` or `No`)');
            disp('');
            y_or_n = string(input(' ','s'));
            if y_or_n == 'Yes' | ...
                  y_or_n == 'YES' | ...
                  y_or_n == 'yes' | ... //accepts multiple
                  y_or_n == 'Y' | ...   //yes answers
                  y_or_n == 'y'
               testB = %t; //returns true if input was correct
            elseif y_or_n == 'No' | ...
                  y_or_n == 'NO' | ...
                  y_or_n == 'no' | ... //accepts multiple
                  y_or_n == 'N' | ...  //no answers
                  y_or_n == 'n'
               disp('Code will return to prompt for an updated input.');

               tic;
               sleep(1500); //1.5 seconds
               toc;

               clc;
               testB = %t; //break out of inner while loop and
                           //return to outer while loop
               testA = %f;
               //will reprompt if user wants to change answer
            else
               disp('Error, please type `Yes` or `No`.');
```

```
            disp(' ');
            //reprompts for an input if anything else is entered
          end
        end
      else
        clc;
        disp('Error - Incorrect Input');
        disp(' ');
      end
    end
    testA = %f; //Reset booleans as same ones are used in following questions
    testB = %f;

else
    laundry_num = 0;
    testA = %f; //Reset booleans as same ones are used in following questions
    testB = %f;
end
//////////////////////
///////////////////////
//////////////////

//********Question 10***********************************************************
//********Question 10***********************************************************
//********Question 10***********************************************************

    clc;
    disp('Question 10 - Number of Showers Per Day');
    disp(' ');

    disp('This question is being asked to determine how many showers should');
    disp('planned for per day, accounting for all occupants. Showers consume');
    disp('water, so knowing the number of showers per day will help make the');
    disp('water calculations more accurate.');
    disp('');

    while testA == %f //will keep looping until information is correct
      disp('How many showers per day should be accounted for?');
      disp('(for all occupants)');
      disp('Input must be either 0, 1, 2, 3, or 4');
      disp('');
      shower_num = input('');
      disp(' ');
      if shower_num == 0 | ...
        shower_num == 1 | ...
        shower_num == 2 | ...
        shower_num == 3 | ...
        shower_num == 4
        testA = %t;
        testB = %f;
        while testB == %f
          clc;
          disp('You have input: ');
          disp('');
          disp(shower_num);
          disp('');
          disp('as the number of showers per day.');
          disp('Is this correct? (Please type `Yes` or `No`)');
          disp('');
          y_or_n = string(input(' ','s'));
          if y_or_n == 'Yes' | ...
```

```
            y_or_n == 'YES' | ...
            y_or_n == 'yes' | ... //accepts multiple
            y_or_n == 'Y' | ...  //yes answers
            y_or_n == 'y'
          testB = %t; //returns true if input was correct
        elseif y_or_n == 'No' | ...
            y_or_n == 'NO' | ...
            y_or_n == 'no' | ... //accepts multiple
            y_or_n == 'N' | ...  //no answers
            y_or_n == 'n'
          disp('Code will return to prompt for an updated input.');

          tic;
          sleep(1500); //1.5 seconds
          toc;

          clc;
          testB = %t; //break out of inner while loop and
                      //return to outer while loop
          testA = %f;
          //will reprompt if user wants to change answer
        else
          disp('Error, please type `Yes` or `No`.');
          disp(' ');
          //reprompts for an input if anything else is entered
        end

      end

    else
      clc;
      disp('Error - Incorrect Input');
      disp(' ');
    end
  end
  testA = %f; //Reset booleans as same ones are used in following questions
  testB = %f;

//********Question 11**********************************************************
//********Question 11**********************************************************
//********Question 11**********************************************************

  clc;
  disp('Question 11 - Is a Bathtub Desired?');
  disp(' ');

  disp('This question is being asked to determine if the occupant(s)');
  disp('want a bathtub in the tiny house. Bathtubs consume water so');
  disp('knowing if one is wanted or now will help make the water');
  disp('calculations more accurate.');
  disp('');

  while testA == %f //will keep looping until information is correct
    disp('Should a bathtub be accounted for?');
    disp('Input must be either `Yes` or `No`');
    disp('');
    bathtub_y_or_n = string(input(' ','s'));
    disp(' ');
    if bathtub_y_or_n == 'Yes' | ...
        bathtub_y_or_n == 'YES' | ...
        bathtub_y_or_n == 'yes' | ... //accepts multiple
```

```
        bathtub_y_or_n == 'Y' | ...   //yes answers
        bathtub_y_or_n == 'y'
    bathtub_y_or_n = 'Yes';
    bathtub_num = 1;
    testA = %t;
    testB = %f;
    while testB == %f
        clc;
        disp('You have input: ');
        disp('');
        disp(bathtub_y_or_n);
        disp('');
        disp('as wanting a bathtub.');
        disp('Is this correct? (Please type `Yes` or `No`)');
        disp('');
        y_or_n = string(input(' ','s'));
        if y_or_n == 'Yes' | ...
                y_or_n == 'YES' | ...
                y_or_n == 'yes' | ... //accepts multiple
                y_or_n == 'Y' | ...   //yes answers
                y_or_n == 'y'
            testB = %t; //returns true if input was correct
        elseif y_or_n == 'No' | ...
                y_or_n == 'NO' | ...
                y_or_n == 'no' | ... //accepts multiple
                y_or_n == 'N' | ...  //no answers
                y_or_n == 'n'
            disp('Code will return to prompt for an updated input.');

            tic;
            sleep(1500); //1.5 seconds
            toc;

            clc;
            testB = %t; //break out of inner while loop and
                        //return to outer while loop
            testA = %f;
            //will reprompt if user wants to change answer
        else
            disp('Error, please type `Yes` or `No`.');
            disp(' ');
            //reprompts for an input if anything else is entered
        end
    end
elseif bathtub_y_or_n == 'No' | ...
        bathtub_y_or_n == 'NO' | ...
        bathtub_y_or_n == 'no' | ... //accepts multiple
        bathtub_y_or_n == 'N' | ...  //no answers
        bathtub_y_or_n == 'n'
    bathtub_y_or_n = 'No';
    bathtub_num = 0;
    testA = %t;
    testB = %f;
    while testB == %f
        clc;
        disp('You have input: ');
        disp('');
        disp(bathtub_y_or_n);
        disp('');
        disp('as wanting a bathtub.');
        disp('Is this correct? (Please type `Yes` or `No`)');
```

```
            disp('');
            y_or_n = string(input(' ','s'));
            if y_or_n == 'Yes' | ...
                y_or_n == 'YES' | ...
                y_or_n == 'yes' | ... //accepts multiple
                y_or_n == 'Y' | ...   //yes answers
                y_or_n == 'y'
              testB = %t; //returns true if input was correct
            elseif y_or_n == 'No' | ...
                y_or_n == 'NO' | ...
                y_or_n == 'no' | ... //accepts multiple
                y_or_n == 'N' | ...  //no answers
                y_or_n == 'n'
              disp('Code will return to prompt for an updated input.');

              tic;
              sleep(1500); //1.5 seconds
              toc;

              clc;
              testB = %t; //break out of inner while loop and
                          //return to outer while loop
              testA = %f;
              //will reprompt if user wants to change answer
            else
              disp('Error, please type `Yes` or `No`.');
              disp(' ');
              //reprompts for an input if anything else is entered
            end
          end
        else
          clc;
          disp('Error - Incorrect Input');
          disp(' ');
        end
      end
      testA = %f; //Reset booleans as same ones are used in following questions
      testB = %f;

//********Question 11a*********************************************************
//********Question 11a*********************************************************
//********Question 11a*********************************************************

if bathtub_num == 1 // only ask this question if the user wants a bathtub.

  clc;
  disp('Question 11a - Number of Baths Per Week');
  disp(' ');

  disp('This question is being asked to determine how many baths should');
  disp('planned for per week. Baths consume water, so knowing the number');
  disp('of baths per week will help make the water calculations more');
  disp('accurate.');
  disp('');

  while testA == %f //will keep looping until information is correct
    disp('How many baths per week should be accounted for?');
    disp('(for all occupants)');
    disp('Input must be either 1, 2, 3, or 4');
    disp('');
    bath_num = input('');
```

```
      disp(' ');
      if bath_num == 1 | ...
         bath_num == 2 | ...
         bath_num == 3 | ...
         bath_num == 4
         testA = %t;
         testB = %f;
         while testB == %f
            clc;
            disp('You have input: ');
            disp('');
            disp(bath_num);
            disp('');
            disp('as the number of baths per week.');
            disp('Is this correct? (Please type `Yes` or `No`)');
            disp('');
            y_or_n = string(input(' ','s'));
            if y_or_n == 'Yes' | ...
                  y_or_n == 'YES' | ...
                  y_or_n == 'yes' | ... //accepts multiple
                  y_or_n == 'Y' | ...   //yes answers
                  y_or_n == 'y'
               testB = %t; //returns true if input was correct
            elseif y_or_n == 'No' | ...
                  y_or_n == 'NO' | ...
                  y_or_n == 'no' | ... //accepts multiple
                  y_or_n == 'N' | ...  //no answers
                  y_or_n == 'n'
               disp('Code will return to prompt for an updated input.');

               tic;
               sleep(1500); //1.5 seconds
               toc;

               clc;
               testB = %t; //break out of inner while loop and
                           //return to outer while loop
               testA = %f;
               //will reprompt if user wants to change answer
            else
               disp('Error, please type `Yes` or `No`.');
               disp(' ');
               //reprompts for an input if anything else is entered
            end
         end
      else
         clc;
         disp('Error - Incorrect Input');
         disp(' ');
      end
   end
   testA = %f; //Reset booleans as same ones are used in following questions
   testB = %f;
else
   bath_num = 0;
   testA = %f; //Reset booleans as same ones are used in following questions
   testB = %f;
end

//********Question 12**********************************************************
```
//********Question 12**

```
//********Question 12*********************************************************

   clc;
   disp('Question 12 - Number of Windows');
   disp(' ');

   disp('This question is being asked to determine how many windows should');
   disp('be accounted for. Windows reduce the energy efficieny of the tiny');
   disp('house, so knowing the number of windows will help make the HVAC');
   disp('calculations more accurate.');
   disp('');

   while testA == %f //will keep looping until information is correct
      disp('How many windows should be accounted for?');
      disp('Input must be either 0, 1, 2, 3, 4, 5, or 6');
      disp('');
      window_num = input('');
      disp(' ');
      if window_num == 0 | ...
         window_num == 1 | ...
         window_num == 2 | ...
         window_num == 3 | ...
         window_num == 4 | ...
         window_num == 5 | ...
         window_num == 6
         testA = %t;
         testB = %f;
         while testB == %f
            clc;
            disp('You have input: ');
            disp('');
            disp(window_num);
            disp('');
            disp('as the number of windows.');
            disp('Is this correct? (Please type `Yes` or `No`)');
            disp('');
            y_or_n = string(input(' ','s'));
            if y_or_n == 'Yes' | ...
                  y_or_n == 'YES' | ...
                  y_or_n == 'yes' | ... //accepts multiple
                  y_or_n == 'Y' | ...   //yes answers
                  y_or_n == 'y'
               testB = %t; //returns true if input was correct
            elseif y_or_n == 'No' | ...
                  y_or_n == 'NO' | ...
                  y_or_n == 'no' | ... //accepts multiple
                  y_or_n == 'N' | ...  //no answers
                  y_or_n == 'n'
               disp('Code will return to prompt for an updated input.');

               tic;
               sleep(1500); //1.5 seconds
               toc;

               clc;
               testB = %t; //break out of inner while loop and
                           //return to outer while loop
               testA = %f;
               //will reprompt if user wants to change answer
            else
               disp('Error, please type `Yes` or `No`.');
```

```
                disp(' ');
                //reprompts for an input if anything else is entered
            end
        end
    else
        clc;
        disp('Error - Incorrect Input');
        disp(' ');
    end
  end
  testA = %f; //Reset booleans
  testB = %f;

user_answers = [trailer_length_ft;  //1
        city_num;         //2
        occupant_num;      //3
        cell_phone_num;    //4
        laptop_num;       //5
        pet_num;          //6
        plant_num;        //7
        microwave_num;     //8
        washing_mach_num;  //9
        laundry_num;       //9a
        shower_num;        //10
        bathtub_num;       //11
        bath_num;         //11a
        window_num;];      //12

endfunction
```

Appendix XVI: Printout of Scilab Code: Calculate Water Consumption

```
function [water_gal_month]=water_calculation(user_answers)

//This function takes the user's answers and calculates the monthly
//water consumption in gallons per month.

//Water - Pets
pet_water_gpw = 36*user_answers(6); //36 gal per week * # of pets

//Water - Washing machine
laundry_water_gpw = 15*user_answers(10); //15 gal per load * loads per week

//Water - Washing hands
hand_washing_water_gpw = 2.5*7*user_answers(3);
  //2.5 gal per day * 7 days per week * # occupants

//Water - Washing dishes
dish_washing_water_gpw = 50; //50 gal per week

//Water - Cooking
cooking_water_gpw = 0.5*3*7*user_answers(3);
  //0.5 gal per meal * 3 meals per day * 7 days per week * # occupants

//Water - Cleaning
gen_cleaning_water_gpw = 3;
  //3 gal per week

//Water - Drinking
drinking_water_gpw = 1*7*user_answers(3);
  //1 gal 7 days per week * # occupants

//Water - Showers
shower_water_gpw = 20*user_answers(11)*7;
  //20 gal per shower * showers per day * 7 days per week

//Water - Baths
bath_water_gpw = 36*user_answers(13);
  //36 gal per bath * baths per week

//Water - Brushing Teeth / General Hygiene
gen_hygiene_water_gpw = 2.5*7*user_answers(3);
  //2.5 gal per day * 7 days per week * # occupants

//Water - Plants
plant_water_gpw = 1*7*1/16*user_answers(7);
  //1 cup per plant per day * 7 days per week * 1 gal per 16 cups * # plants

//Water - Miscellaneous
misc_water_gpw = 7;
  //1 gal per day misc. * 7 days per week

water_gal_month = (pet_water_gpw + laundry_water_gpw + ...
  hand_washing_water_gpw + dish_washing_water_gpw + cooking_water_gpw + ...
  gen_cleaning_water_gpw + drinking_water_gpw + shower_water_gpw + ...
  bath_water_gpw + gen_hygiene_water_gpw + plant_water_gpw + ...
  misc_water_gpw)*52/12;
  //gal per week * 52 weeks per year / 12 months per year

endfunction
```

Appendix XVII: Printout of Scilab Code: Calculate Live Load

```
function [live_load_BTU_hr]=live_load_calculation(user_answers)

//This function takes the user's answers and calculates the live load
//in BTU per hr.

//Live Load - Lights
lights_BTU_hr = 1/80*6690/240*280*3.412
  //1 watt per 80 lumens * 6690 lumens per 240 sqft * 280 sqft *
  //3.412 BTU per hour per watt

//Live Load - Occupant(s)
occupant_BTU_hr = 356*user_answers(3)
  //356 BTU per hour * num occupant

//Live Load - Pet(s)
pet_BTU_hr = 89*user_answers(6)
  //89 BTU per hour * num pet

//Live Load - Laptop(s)
laptop_BTU_hr = 55.6*3.412*user_answers(5)
  //55.6 watts * 3.412 BTU per hour per watt * num laptops

//Live Load - Cell Phones (Smart Phones)
cell_phone_BTU_hr = 13.9*3.412*user_answers(4)
  //13.9 watts * 3.412 BTU per hour per watt * num cell phones

//Live Load - Refrigerator
refrig_BTU_hr = 690*0.028317*18*3.412
  //690 watts per m^3 * 0.028317 m^3 per ft^3 * 18 ft^3 *
  //3.412 BTU per hour per watt

//Live Load - HVAC Mini-Split
minisplit_BTU_hr = 500*3.412
  //500 watts * 3.412 BTU per hour per watt

//Live Load - Miscellaneous
misc_BTU_hr = 2000
  ///2000 BTU per hour miscellaneous

live_load_BTU_hr = (lights_BTU_hr + occupant_BTU_hr + pet_BTU_hr + ...
          laptop_BTU_hr + cell_phone_BTU_hr + refrig_BTU_hr + ...
          minisplit_BTU_hr + misc_BTU_hr);

endfunction
```

Appendix XVIII: Printout of Scilab Code: Calculate HVAC Needs

```
function [HVAC_data]=HVAC_calculation(user_answers, live_load_BTU_hr)

//This function takes the user's answers and calculates the live load
//in BTU per hr.

heating_matrix = [1,   18.4,    196.7,   5408.7,    5988.2;
            2,   5.3,     243.9,   6703.9,    7424.4;
            3,   -13.9,   313.0,   8602.2,    9529.4;
            4,   7.3,     236.7,   6506.2,    7205.1;
            5,   34.1,    140.1,   3856.5,    4267.0;
            6,   7.5,     235.9,   6486.4,    7183.2;
            7,   52.6,    73.5,    2027.4,    2238.7;
            8,   19.9,    191.3,   5260.4,    5823.8;
            9,   41.8,    112.5,   3095.2,    3422.8;
            10,  5.2,     244.2,   6713.8,    7435.4;
            11,  33.2,    143.4,   3945.5,    4365.6;
            12,  29.5,    156.7,   4311.3,    4771.3;];
            //city #
                //99% dry bulb heating temp.
                      //single window BTU per hr loss
                            //20ft BTU per hr loss
                                  //24ft BTU per hr loss

cooling_matrix = [1,   91.4,    66.3,    1822.3,    2020.4;
            2,   92.2,    69.2,    1901.6,    2108.2;
            3,   87.0,    50.4,    1386.5,    1537.8;
            4,   86.5,    48.6,    1337.0,    1483.0;
            5,   95.5,    81.0,    2228.4,    2470.1;
            6,   92.7,    71.0,    1951.1,    2163.0;
            7,   90.8,    64.1,    1762.9,    1954.6;
            8,   92.3,    69.5,    1911.5,    2119.1;
            9,   108.3,   127.2,   3496.3,    3873.9;
            10,  83.3,    37.1,    1020.0,    1132.1;
            11,  97.6,    88.6,    2436.4,    2700.4;
            12,  81.6,    31.0,    851.6,     945.6;];
            //city #
                //1% dry bulb cooling temp.
                      //single window BTU per hr gain
                            //20ft BTU per hr gain
                                  //24ft BTU per hr gain

air_changes_matrix = [1,  10932.5,   3168.6,    12756.1,   3697.2;
              2,  13931.5,   3301.8,    16255.5,   3852.6;
              3,  18659.2,   2431.2,    21771.8,   2836.8;
              4,  13472.4,   2347.6,    15719.7,   2739.3;
              5,  7535.2,    3842.1,    8792.2,    4483.1;
              6,  13415.5,   3383.0,    15653.4,   3947.4;
              7,  3803.8,    3069.6,    4438.3,    3581.7;
              8,  10593.6,   3314.3,    12360.8,   3867.2;
              9,  5945.7,    5865.9,    6937.5,    6844.4;
              10, 13952.1,   1803.6,    16279.5,   2104.5;
              11, 7719.2,    4182.9,    9006.8,    4880.7;
              12, 8510.3,    1510.1,    9929.9,    1762.0;];
              //city number
                 //20ft winter
                        //20ft summer
                               //24ft winter
                                      //24ft summer
```

```
if user_answers(1) == 20
  trailer_column = 4; //use 4th column in matrices above for 20ft trailer
  air_ex = [2,3]; //use 2nd and 3rd columns for air changes matrix
else
  trailer_column = 5; //use 5th column in matrices above for 24ft trailer
  air_ex = [4,5]; //use 4th and 5th columns for air changes matrix
end

for loop_num = 1:12 //loop through all cities
  if user_answers(2) == loop_num //the city the user picked matches the loop

    //
    //heating for winter
    one_percent_heat_temp = heating_matrix(loop_num,2) //second column

    window_heat_BTU_hr = heating_matrix(loop_num,3)*user_answers(14);
      //BTU per hr heat loss from all windows

    trailer_heat_BTU_hr = heating_matrix(loop_num,trailer_column) + ...
      air_changes_matrix(loop_num,air_ex(1));
      //BTU per hr heat loss from trailer shell

    HVAC_heat_BTU_hr = window_heat_BTU_hr + trailer_heat_BTU_hr;
      //add subtotals

    //
    //cooling for summer
    one_percent_cool_temp = cooling_matrix(loop_num,2) //second column

    window_cool_BTU_hr = cooling_matrix(loop_num,3)*user_answers(14);
      //BTU per hr heat loss from all windows

    trailer_cool_BTU_hr = cooling_matrix(loop_num,trailer_column) + ...
      air_changes_matrix(loop_num,air_ex(2));
      //BTU per hr heat loss from trailer shell

    HVAC_cool_BTU_hr = window_cool_BTU_hr + trailer_cool_BTU_hr;
      //add subtotals
  end
end

HVAC_data = [one_percent_heat_temp;
       HVAC_heat_BTU_hr;
       one_percent_cool_temp;
       HVAC_cool_BTU_hr;];

endfunction
```

Appendix XIX: Printout of Scilab Code: Calculate Energy Consumption

```
function [energy_kWh_month]=energy_calculation(user_answers)

//This function takes the user's answers and calculates the monthly
//energy consumption in kWh per month.

//Energy - Lights
light_Wh_month = 1/80*6690/240*280*8*30;
  //1 watt per 80 lumens * 6690 lumens per 240 ft^2 * 280 ft^2 *
  //8 hours per day * 30 days per month

//Energy - Laptop(s)
laptop_Wh_month = 70*3*1*30*user_answers(5)
  //70 watts * 3 hours to charge * 1 charge per day * 30 days per month *
  //num laptops

//Energy - Cell Phones (Smart Phones)
phone_Wh_month = 17.5*3*1*30*user_answers(4);
  //17.5 watts * 3 hours to charge * 1 charge per day * 30 days per month *
  //num cell phones

//Energy - Microwave
microwave_Wh_month = 1000*0.5*30*user_answers(8);
  //1000 watts * 0.5 hours per day * 30 days per month * num microwaves

//Energy - Refrigerator
refrig_Wh_month = 160*24*0.85*30;
  //160 watts * 24 hours per day * 85% of day running * 30 days per month

//Energy - Washing Machine
wash_mach_Wh_month = 1200*0.5*user_answers(10)*52/12;
  //1200 watts * 0.5 hours per load * loads per week * weeks per month

//Energy for HVAC Mini-Split
minisplit_Wh_month = 800*12*30;
  //800 watts * 12 hours per day * 30 days per month

//Energy - Exhaust Fans
exhaust_fan_Wh_month = 12*12*30;
  //12 watts * 12 hours per day * 30 days per month

//Energy - Miscellaneous
misc_Wh_month = 500*30;
  //500 Wh per day * 30 days per month for misc small appliances

energy_kWh_month = (light_Wh_month + ...
            laptop_Wh_month + ...
            phone_Wh_month + ...
            microwave_Wh_month + ...
            refrig_Wh_month + ...
            wash_mach_Wh_month + ...
            minisplit_Wh_month + ...
            exhaust_fan_Wh_month + ...
            misc_Wh_month)/1000;
            // watt hours * 1 kWh / 1000 Wh

endfunction
```

Appendix XX: Printout of Scilab Code: Calculate BOM and Approx. Cost

```
function [cost_and_BOM]=cost_BOM_calculation(user_answers);

//This function takes the user's answers and calculates the tiny house
//shell cost along with the associated preliminary BOM.

BOM_fix =  []; //delcare variables this time because of future appends;
cost_fix = 0;
BOM_var =  [];
cost_var = 0;

BOM_matrix_fix = [...
    0.1367      960.00     131.20     960.00      131.20; //2x12,      in
    0.0382      8902.81     340.35     9362.34     357.91; //2x4,       in
    0.0467      4501.52     210.07     4891.52     228.27; //2x6,       in
    0.0815      1736.00     141.41     1736.00     141.41; //2x8,       in
    313.0400     1.00       313.04     1.00        313.04; //b. sink,   pc
    4.9900      31.84      158.89     43.75      218.31; //b. tile,   sqft
    0.8858      1221.64     1082.12    1414.52     1252.98;//insul,     sqft
    250.0000     1.00       250.00     1.00        250.00; //counter,   pc
    289.0000     1.00       289.00     1.00        289.00; //k. sink,   pc
    1.0042      228.42     229.37     261.36      262.46; //m. roof,   sqft
    1798.99      1.00       1798.99    1.00        1798.99; //minisplit, pc
    0.3734      2309.51     862.46     2594.36     968.83; //plywood,   sqft
    263.1300     1.00       263.13     1.00        263.13; //q. mat,    pc
    389.0000     1.00       389.00     1.00        389.00; //refrig,    pc
    1.3825      537.19     742.66     610.95      844.64; //shiplap,   sqft
    1.3825      612.32     846.53     693.76      959.12; //siding,    sqft
    555.1700     1.00       555.17     1.00        555.17; //range,     pc
    99.0000      1.00      99.00      1.00        99.00;];//toilet,    pc
    //$ per qty
                //20ft qty
                        //20ft subtotal
                              //24ft qty
                                      //24ft subtotal

if user_answers(1) == 20
    BOM_fix = BOM_matrix_fix(:,2); //take qty's from 20ft model
else
    BOM_fix = BOM_matrix_fix(:,4); //take qty's from 24ft model
end

if user_answers(1) == 20
    cost_fix = sum(BOM_matrix_fix(:,3)); //sum $ from 20ft model
else
    cost_fix = sum(BOM_matrix_fix(:,5)); //sum $ from 24ft model
end

BOM_matrix_var = [...
    6699.0000    1.00      6699.00    0.00        0.00;  //20ft,      pc
    6999.0000    0.00      0.00       1.00        6999.00;//24ft,      pc
    323.4800     1.00      323.48     1.00        323.48; //b. tub,    pc
    59.2100      1.00      59.21      1.00        59.21;  //microwave, pc
    999.0000     1.00      999.00     1.00        999.00; //washing,   pc
    81.2300      6.00      487.38     6.00        487.38; //window,    pc
    0.0382       216.00    8.26       216.00      8.26;]; //wind 2x4,  in
                                           //based on #14
    //$ per qty
                //20ft qty
```

```
                    //20ft subtotal
                          //24ft qty
                                 //24ft subtotal

//Account for trailer chosen:
if user_answers(1) == 20
  BOM_var = [BOM_var; BOM_matrix_var(1,2);];
          //append qty 1 of 20ft trailer
  cost_var = cost_var + BOM_matrix_var(1,3);
          //add cost of 20ft trailer

else
  BOM_var = [BOM_var; BOM_matrix_var(2,4);];
          //append qty 1 of 24ft trailer
  cost_var = cost_var + BOM_matrix_var(2,5);
          //add cost of 20ft trailer
end

//Acccount for bathtub:
if user_answers(12) == 1
  BOM_var = [BOM_var; 1;];
          //append qty 1 bathtub
  cost_var = cost_var + BOM_matrix_var(3,1);
          //add cost of bathtub

else
  BOM_var = [BOM_var; 0;];
          //append qty 0 bathtub
  //no changes to cost_variable
end

//Acccount for microwave:
if user_answers(8) == 1
  BOM_var = [BOM_var; 1;];
          //append qty 1 microwave
  cost_var = cost_var + BOM_matrix_var(4,1);
          //add cost of microwave

else
  BOM_var = [BOM_var; 0;];
          //append qty 0 microwave
  //no changes to cost_variable
end

//Acccount for washing machine;
if user_answers(9) == 1
  BOM_var = [BOM_var; 1;];
          //append qty 1 washing machine
  cost_var = cost_var + BOM_matrix_var(5,1);
          //add cost of washing machine

else
  BOM_var = [BOM_var; 0;];
          //append qty 0 washing machine
  //no changes to cost_variable
end

//Acccount for windows
if user_answers(14) > 0
  BOM_var = [BOM_var; user_answers(14);];
          //append input qty of windows
```

```
   cost_var = cost_var + BOM_matrix_var(6,1)*user_answers(14);
           //add cost of all windows

else
   BOM_var = [BOM_var; 0;];
           //append qty 0 washing machine
   //no changes to cost_variable
end

//Account for extra 2x4s if windows are chosen (ignore reduction in insul)
if user_answers(14) > 0
   BOM_fix(2) = BOM_fix(2) + user_answers(14)*2*18;
     //add 18in of 2x4 (more than necessary) on top and bottom per window
   cost_fix = cost_fix + BOM_matrix_fix(2,1)*2*18*user_answers(14);
     //add cost for 18in of 2x4 on top and bottom per window
end

BOM_final = [BOM_fix; BOM_var;];        //add BOMs together
cost_final = cost_fix + cost_var;       //add costs together

cost_and_BOM = [cost_final; BOM_final;];   //combine into one list

endfunction
```

Appendix XXI: Printout of Scilab Code: Summary and Closing

```
function [tiny_house_summary]=summary_info(user_answers, water_gal_month, ...
                    HVAC_data, energy_kWh_month, cost_and_BOM);
//This function takes in all previous information and summarizes everything
//for the user

//tidy the command window
clc;

//Print closing statement to command window
  disp('All calculations are finished, and the summary for your tiny house');
  disp('is shown in the following lines.');
  disp('-----------------------------------------------------------------');
  disp(' ');

  tiny_house_summary = [user_answers;
            water_gal_month;
            HVAC_data;
            energy_kWh_month;
            cost_and_BOM;];

  disp('The length of trailer chosen (excluding the gooseneck) in feet is:');
  disp(tiny_house_summary(1));
  disp('-----------------------------------------------------------------');
  disp('');


  disp('The city chosen for the tiny house to reside in is:');
  if tiny_house_summary(2) == 1
    disp('Baltimore, MD');
  elseif tiny_house_summary(2) == 2
    disp('Denver, CO');
  elseif tiny_house_summary(2) == 3
    disp('Fargo, ND');
  elseif tiny_house_summary(2) == 4
    disp('Grand Rapids, MI');
  elseif tiny_house_summary(2) == 5
    disp('Houston, TX');
  elseif tiny_house_summary(2) == 6
    disp('Kansas City, MO');
  elseif tiny_house_summary(2) == 7
    disp('Miami, FL');
  elseif tiny_house_summary(2) == 8
    disp('Nashville, TN');
  elseif tiny_house_summary(2) == 9
    disp('Phoenix, AZ');
  elseif tiny_house_summary(2) == 10
    disp('Portland, ME');
  elseif tiny_house_summary(2) == 11
    disp('Sacramento, CA');
  else
    disp('Seattle, WA');
  end
  disp('-----------------------------------------------------------------');
  disp('');


  disp('The number of occupants to account for is:');
  disp(tiny_house_summary(3));
```

```
disp('-----------------------------------------------------------------');
disp('');


disp('The number of cell phones to account for is:');
disp(tiny_house_summary(4));
disp('-----------------------------------------------------------------');
disp('');


disp('The number of laptops to account for is:');
disp(tiny_house_summary(5));
disp('-----------------------------------------------------------------');
disp('');


disp('The number of pets to account for is:');
disp(tiny_house_summary(6));
disp('-----------------------------------------------------------------');
disp('');


disp('The number of plants to account for is:');
disp(tiny_house_summary(7));
disp('-----------------------------------------------------------------');
disp('');


disp('Should a microwave be accounted for?');
if tiny_house_summary(8) == 1
   disp('Yes.');
else
   disp('No.');
end
disp('-----------------------------------------------------------------');
disp('');


disp('Should a washing machine be accounted for?');
if tiny_house_summary(9) == 1
   disp('Yes.');
else
   disp('No.');
end
disp('-----------------------------------------------------------------');
disp('');


if tiny_house_summary(9) == 1
   disp('The number of loads of laundry per week to account for is:');
   disp(tiny_house_summary(10));
   disp('-----------------------------------------------------------------');
   disp('');
end


disp('The number of showers per day to account for is:');
disp(tiny_house_summary(11));
disp('-----------------------------------------------------------------');
disp('');
```

```
disp('Should a bathtub be accounted for?');
if tiny_house_summary(12) == 1
   disp('Yes.');
else
   disp('No.');
end
disp('----------------------------------------------------------------');
disp('');


if tiny_house_summary(12) == 1
   disp('The number of baths per week to account for is:');
   disp(tiny_house_summary(13));
   disp('----------------------------------------------------------------');
   disp('');
end


disp('The number of windows to account for is:');
disp(tiny_house_summary(14));
disp('----------------------------------------------------------------');
disp('');


disp('The approximate gallons per month of water usage is:');
disp(tiny_house_summary(15));
disp('');
disp('Compared to the national average of 3,000 gallons of water used');
disp('per month, per person, your usage is:');
if tiny_house_summary(3) == 1 //number of occupants
   disp(abs(((tiny_house_summary(15)/3000)-1)*100)); // compare 1:1
   if tiny_house_summary(15)/3000 > 1
      disp('percent higher.');
   else
      disp('percent lower.');
   end
else
   disp(abs(((tiny_house_summary(15)/6000)-1)*100)); // compare 2:2
   if tiny_house_summary(15)/6000 > 1
      disp('percent higher.');
   else
      disp('percent lower.');
   end
end
disp('----------------------------------------------------------------');
disp('');


disp('The 99% dry bulb heating temperature in degrees Fahrenheit');
disp('for your chosen location is:');
disp(tiny_house_summary(16));
disp('----------------------------------------------------------------');
disp('');


disp('The needed heating capacity in BTU/hr for your location/options is:');
disp(tiny_house_summary(17));
disp('----------------------------------------------------------------');
disp('');
```

```
disp('The 1% dry bulb cooling temperature in degrees Fahrenheit');
disp('for your chosen location is:');
disp(tiny_house_summary(18));
disp('------------------------------------------------------------------');
disp('');


disp('The needed cooling capacity in BTU/hr for your location/options is:');
disp(tiny_house_summary(19));
disp('------------------------------------------------------------------');
disp('');


disp('The approximate kWh per month energy usage is:');
disp(tiny_house_summary(20));
disp('');
disp('Compared to the national average of 914.3 kWh of energy used');
disp('per month, your usage is:');
disp(abs(((tiny_house_summary(20)/914.3)-1)*100));
if tiny_house_summary(20)/914.3 > 1
   disp('percent higher.');
else
   disp('percent lower.');
end
disp('------------------------------------------------------------------');
disp('');


disp('The amount, in inches, of 2 x 12s needed is:');
disp(tiny_house_summary(22));
disp('------------------------------------------------------------------');
disp('');


disp('The amount, in inches, of 2 x 4s needed is:');
disp(tiny_house_summary(23));
disp('------------------------------------------------------------------');
disp('');


disp('The amount, in inches, of 2 x 6s needed is:');
disp(tiny_house_summary(24));
disp('------------------------------------------------------------------');
disp('');


disp('The amount, in inches, of 2 x 8s needed is:');
disp(tiny_house_summary(25));
disp('------------------------------------------------------------------');
disp('');


disp('The amount, in pieces, of bathroom sinks needed is:');
disp(tiny_house_summary(26));
disp('------------------------------------------------------------------');
disp('');


disp('The amount, in square feet, of bathroom tile needed is:');
disp(tiny_house_summary(27));
```

```
disp('-----------------------------------------------------------------');
disp('');


disp('The amount, in square feet, of cotton/denim insulation needed is:');
disp(tiny_house_summary(28));
disp('-----------------------------------------------------------------');
disp('');


disp('The amount, in pieces, of kitchen counters needed is:');
disp(tiny_house_summary(29));
disp('-----------------------------------------------------------------');
disp('');


disp('The amount, in pieces, of kitchen sinks needed is:');
disp(tiny_house_summary(30));
disp('-----------------------------------------------------------------');
disp('');


disp('The amount, in sqft, of metal roof needed is:');
disp(tiny_house_summary(31));
disp('-----------------------------------------------------------------');
disp('');


disp('The amount, in pieces, of HVAC mini-splits needed is:');
disp(tiny_house_summary(32));
disp('-----------------------------------------------------------------');
disp('');


disp('The amount, in sqft, of plywood needed is:');
disp(tiny_house_summary(33));
disp('-----------------------------------------------------------------');
disp('');


disp('The amount, in pieces, of queen mattresses needed is:');
disp(tiny_house_summary(34));
disp('-----------------------------------------------------------------');
disp('');


disp('The amount, in pieces, of refrigerators needed is:');
disp(tiny_house_summary(35));
disp('-----------------------------------------------------------------');
disp('');


disp('The amount, in square feet, of shiplap needed is:');
disp(tiny_house_summary(36));
disp('-----------------------------------------------------------------');
disp('');


disp('The amount, in square feet, of siding needed is:');
disp(tiny_house_summary(37));
disp('-----------------------------------------------------------------');
```

```
disp('');


disp('The amount, in pieces, of stove/ovens needed is:');
disp(tiny_house_summary(38));
disp('------------------------------------------------------------------');
disp('');


disp('The amount, in pieces, of toilets needed is:');
disp(tiny_house_summary(39));
disp('------------------------------------------------------------------');
disp('');


disp('The amount, in pieces, of trailers needed is:');
disp(tiny_house_summary(40));
disp('------------------------------------------------------------------');
disp('');


disp('The amount, in pieces, of bathtubs needed is:');
disp(tiny_house_summary(41));
disp('------------------------------------------------------------------');
disp('');


disp('The amount, in pieces, of microwaves needed is:');
disp(tiny_house_summary(42));
disp('------------------------------------------------------------------');
disp('');


disp('The amount, in pieces, of washer/dryer combination needed is:');
disp(tiny_house_summary(43));
disp('------------------------------------------------------------------');
disp('');


disp('The amount, in pieces, of windows needed is:');
disp(tiny_house_summary(44));
disp('------------------------------------------------------------------');
disp('');


disp('The approximate cost in USD for the shell of the tiny house is:');
disp(tiny_house_summary(21));
disp('------------------------------------------------------------------');
disp('');


disp('Thank you for using this tool. Please scroll to the top of the');
disp('terminal window and enjoy building your tiny house!');
disp('');
disp('All code was written by Michael Stratton II');
//trailer_length_ft;          1
//city_num;                   2
//occupant_num;               3
//cell_phone_num;             4
//laptop_num;                 5
//pet_num;                    6
```

```
//plant_num;              7
//microwave_num;          8
//washing_mach_num;       9
//laundry_num;            10
//shower_num;             11
//bathtub_num;            12
//bath_num;               13
//window_num;             14
//water_gal_month;        15
//one_percent_heat_temp;  16
//HVAC_heat_BTU_hr;       17
//one_percent_cool_temp;  18
//HVAC_cool_BTU_hr;       19
//energy_kWh_month;       20
//cost                    21
//2x12,     in            22
//2x4,      in            23
//2x6,      in            24
//2x8,      in            25
//b. sink,  pc            26
//b. tile,  sqft          27
//insul,    sqft          28
//counter,  pc            29
//k. sink,  pc            30
//m. roof,  sqft          31
//minisplit, pc           32
//plywood,  sqft          33
//q. mat,   pc            34
//refrig,   pc            35
//shiplap,  sqft          36
//siding,   sqft          37
//range,    pc            38
//toilet,   pc            39
//trailer,  pc            40
//b. tub,   pc            41
//microwave, pc           42
//washing,  pc            43
//window,   pc            44

endfunction
```

Appendix XXII: Tiny House Part Relations

This section can be used in conjunction with Appendix XXIII: Tiny House Construction Drawings to build the tiny house. The drawings in the following appendix are individualized for each part, and this section can help the user visualize how the different parts fit together. Only the 20 ft model is shown, but the same outline can be used for the 24 ft model. The numbered list below shows the order of added components (or if a view was rotated). In the list below, left signifies the driver side and right signifies the passenger side.

1. Floor (main and goose)
2. Fender sheathing (bottom)
3. Floor joists (main and goose)
4. Floor insul. (main and goose)
5. Floor plywood (main and goose)
6. Side walls
7. Goose
8. Rotate view
9. Back wall
10. Roof joists
11. Back wall sheathing
12. Left wall sheathing (and goose)
13. Rotate view
14. Right wall sheathing (and goose)
15. Goose sheathing (front)
16. Roof sheathing
17. Insul. (left wall and fenders)
18. Fender sheathing (top)
19. Back wall insulation
20. Goose insulation
21. Goose insulation (rotated view)
22. Rotate view
23. Metal Roof
24. Exterior Siding
25. Doors (right side)
26. Door (left side, rotated view)
27. Rotate view
28. Roof insulation
29. Interior roof sheathing
30. Interior shiplap (rotated view)
31. Top floor
32. Stairs
33. Bathroom tile
34. Bathtub
35. Toilet
36. Bathroom sink
37. Bathroom wall
38. Washer/dryer
39. Kitchen counter
40. Kitchen sink
41. Stove/oven
42. Refrigerator
43. Kitchen table
44. Closet
45. Couch and cushion
46. Bed and mattress
47. Finished model (rotated view)

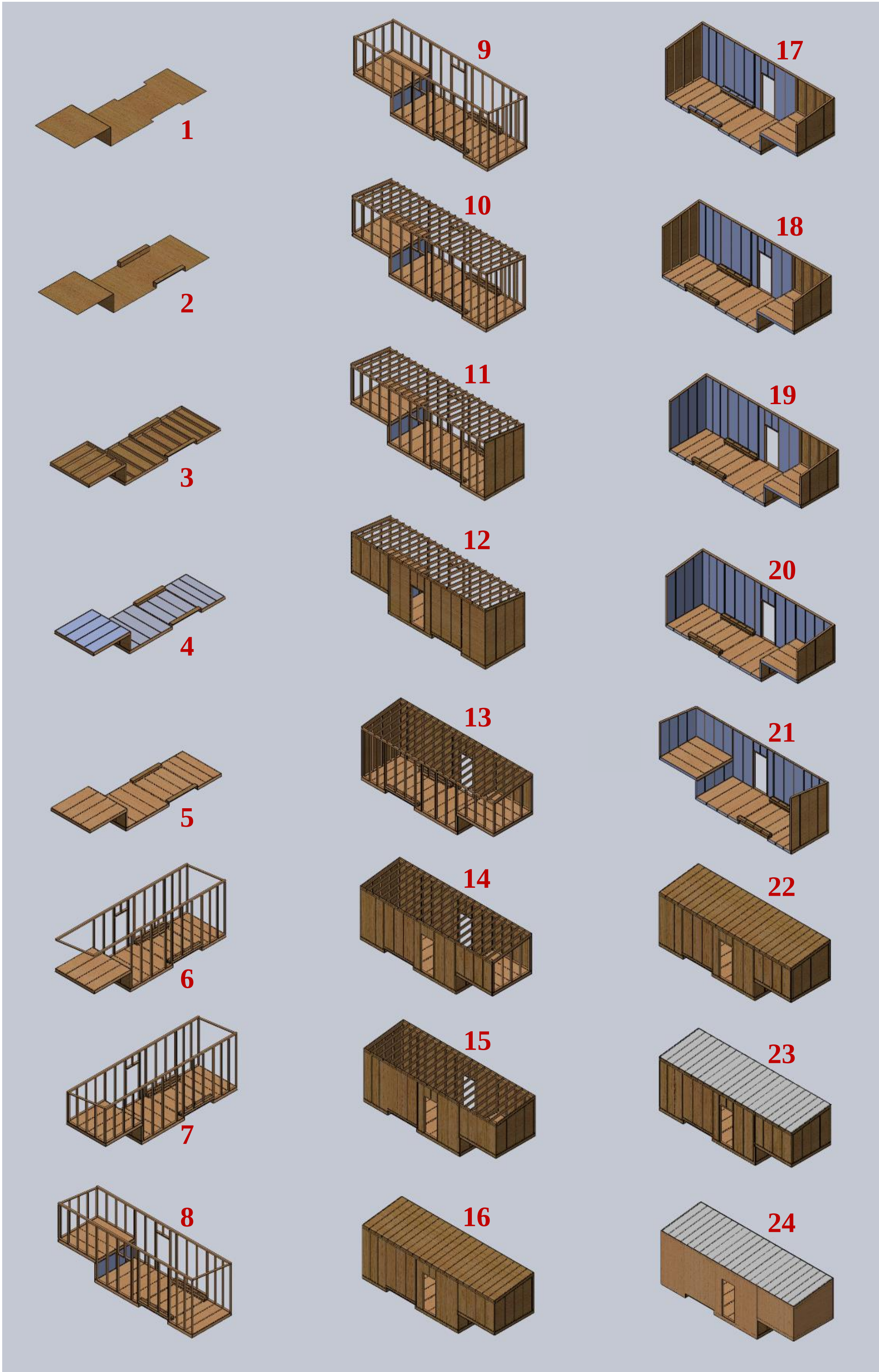


Figure 102: Tiny House Part Relations (Part 1 of 2)

(Multiple Views)

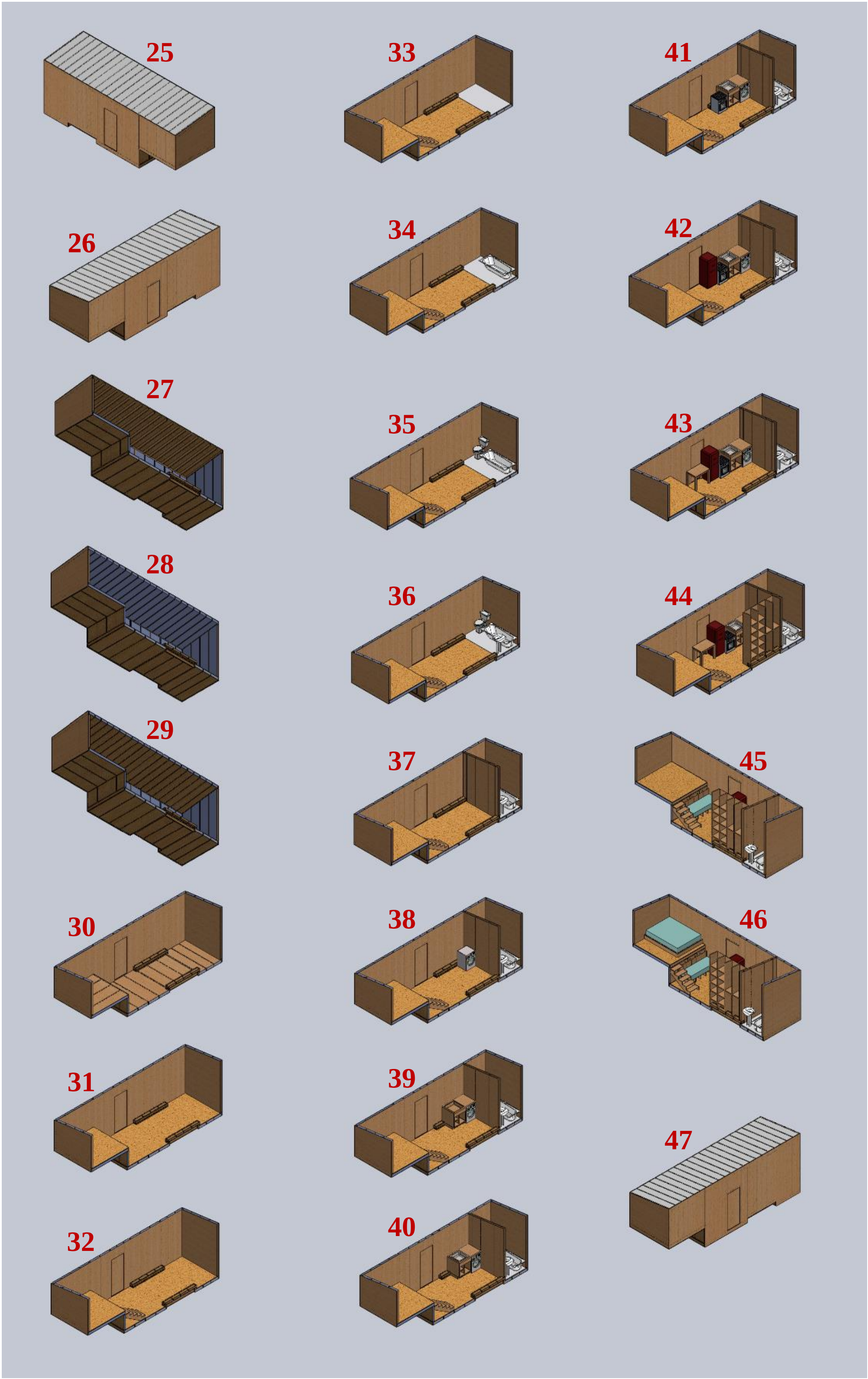


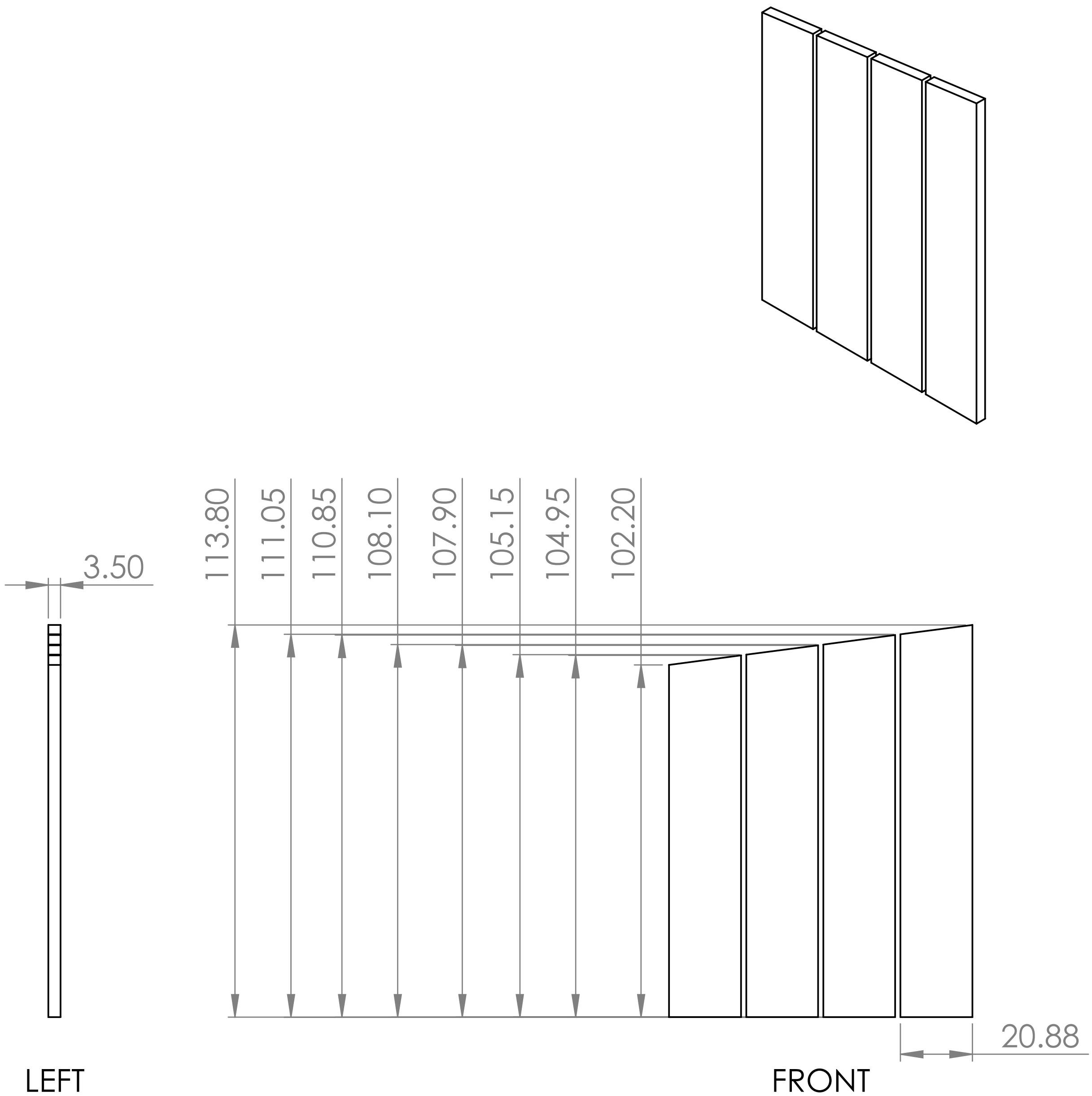
Figure 103: Tiny House Part Relations (Part 2 of 2)

(Multiple Views)

Appendix XXIII: Tiny House Construction Drawings

The construction drawings on the follow pages can assist the user in building a tiny house in conjunction with the results from the Scilab design tool and Appendix XXII: Tiny House Part Relations. The list below shows what drawings are included. The first drawings (without a length signifier in the name) are the same for both tiny house models. The remaining drawings are unique to either the 20 ft or 24 ft design, as indicated in the name. Only shell related items (vertical supports, floor, roof, insulation, etc.) were included in the construction drawings. Interior items such as a refrigerator, sink, etc. will have to be selected by the user. Appendix IX: Weights and Dimensions of Household Items can be referenced to determine what size items were used in the model. In the list below, L (left) signifies the driver side, and R (right) signifies the passenger side.

- Back_Insul
- Back_Sheathing
- Back_Wall
- Bathroom_Wall
- Bed_Frame
- Closet
- Couch_Frame
- Fenders
- Fender_Insul
- Fender_Sheathing
- Goose
- Goose_Insul
- Goose_Sheathing
- Stairs
- Bathroom_Tile_20ft
- Exterior_Siding_20ft
- Floor_20ft
- Floor_Insul_20ft
- Floor_Sheathing_20ft
- Floor_Top_20ft
- Int_Roof_Sheathing_20ft
- Interior_Shiplap_20ft
- L_Side_Wall_Sheathing_20ft
- R_Side_Wall_Sheathing_20ft
- Roof_20ft
- Roof_Insul_20ft
- Roof_Metal_20ft
- Roof_Sheathing_20ft
- Side_Walls_20ft
- Side_Wall_Insul_20ft
- Bathroom_Tile_24ft
- Exterior_Siding_24ft
- Floor_24ft
- Floor_Insul_24ft
- Floor_Sheathing_24ft
- Floor_Top_24ft
- Int_Roof_Sheathing_24ft
- Interior_Shiplap_24ft
- L_Side_Wall_Sheathing_24ft
- R_Side_Wall_Sheathing_24ft
- Roof_24ft
- Roof_Insul_24ft
- Roof_Metal_24ft
- Roof_Sheathing_24ft
- Side_Walls_24ft
- Side_Wall_Insul_24ft

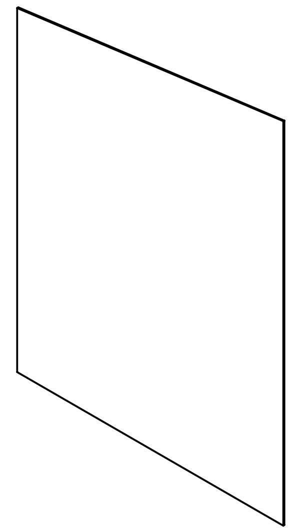


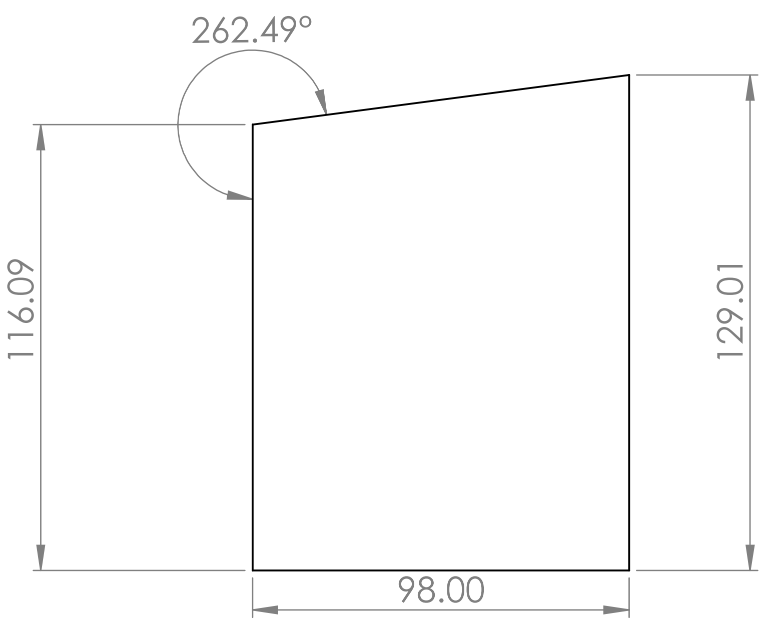
DIMENSIONS ARE IN INCHES AND DEGREES UNLESS OTHERWISE NOTED TOLERANCES: LINEAR ±0.005 ANGULAR: ±0.005		NAME	DATE
	DRAWN	M. STRATTON	07/22/20
	COMMENTS: INSULATION WILL NEED TO BE CUT TO FIT, POSSIBLY IN MULTIPLE PIECES FOR ONE SECTION.		
DO NOT SCALE DRAWING			

PROPRIETARY: THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF M. STRATTON IN CONJUNCTION WITH GVSU.

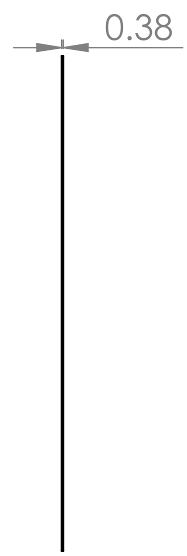
SIZE	DWG. NO.	REV.
A	Back_Insul	

SCALE:1:50	WEIGHT:	SHEET 1 OF 1





FRONT

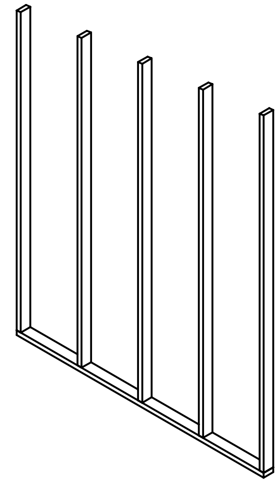


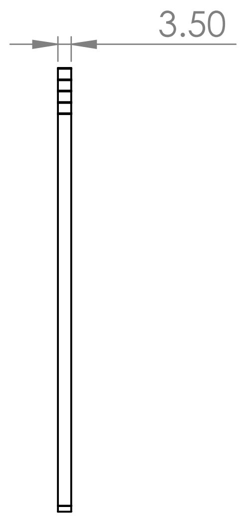
RIGHT

DIMENSIONS ARE IN INCHES AND DEGREES UNLESS OTHERWISE NOTED TOLERANCES: LINEAR ±0.005 ANGULAR: ±0.005		NAME	DATE	**PROPRIETARY:** THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF M. STRATTON IN CONJUNCTION WITH GVSU.
	DRAWN	M. STRATTON	07/22/20	
	COMMENTS: PLYWOOD WILL NEED TO BE CUT TO FIT, POSSIBLY IN MULTIPLE PIECES FOR ONE SECTION. PLYWOOD IS 3/8IN THICK.			SIZE **A** DWG. NO. Back_Sheathing REV.
DO NOT SCALE DRAWING				SCALE:1:50 WEIGHT: SHEET 1 OF 1

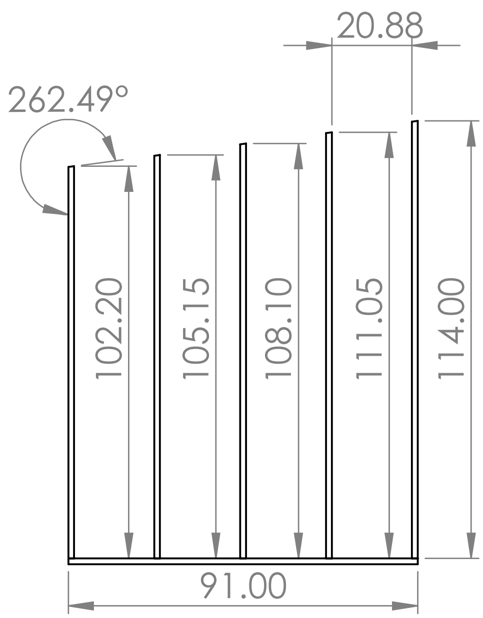
2 1

B B





LEFT

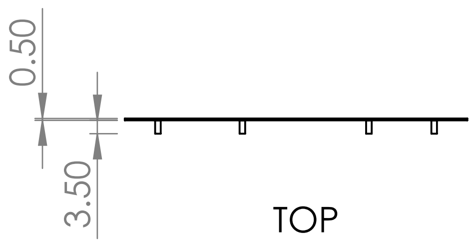


FRONT

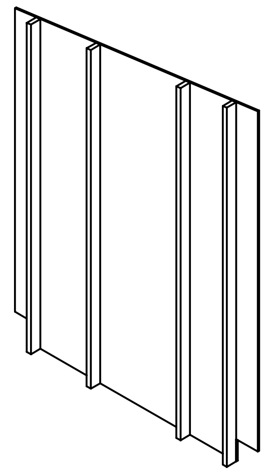
A A

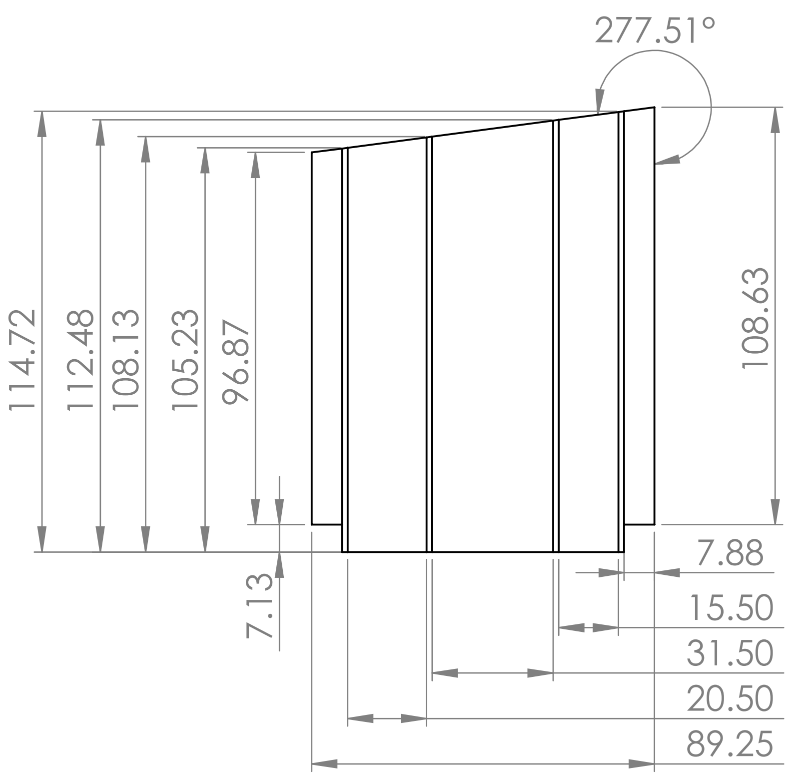
DIMENSIONS ARE IN INCHES AND DEGREES UNLESS OTHERWISE NOTED TOLERANCES: LINEAR ±0.005 ANGULAR: ±0.005		NAME	DATE	**PROPRIETARY:** THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF M. STRATTON IN CONJUNCTION WITH GVSU.	
	DRAWN	M. STRATTON	07/22/20		
	COMMENTS: ALL STUDS ARE 2X4s			SIZE **A** DWG. NO. Back_Wall	REV.
DO NOT SCALE DRAWING				SCALE:1:50 WEIGHT:	SHEET 1 OF 1

2 1



TOP

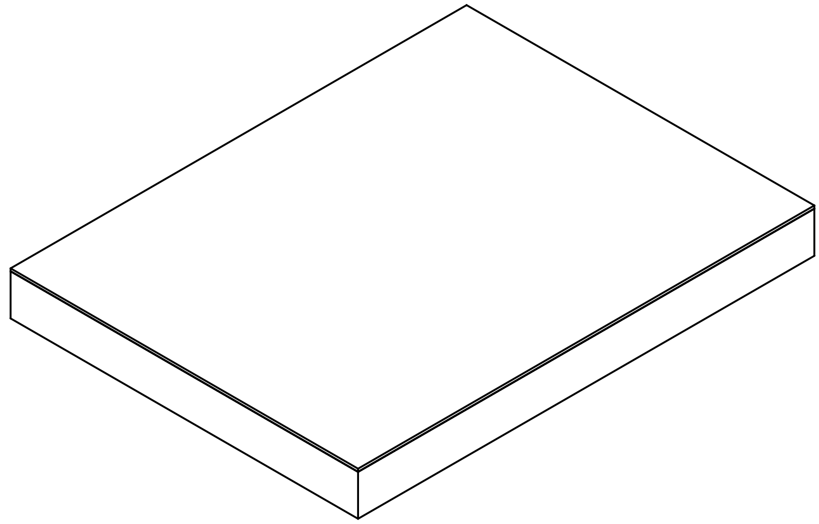


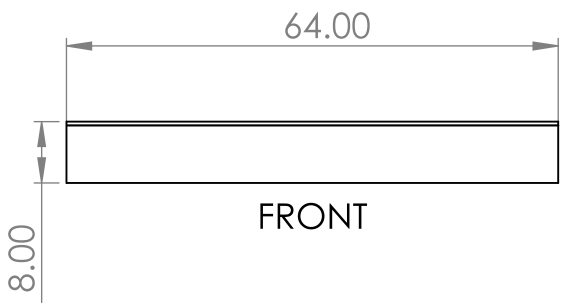


FRONT

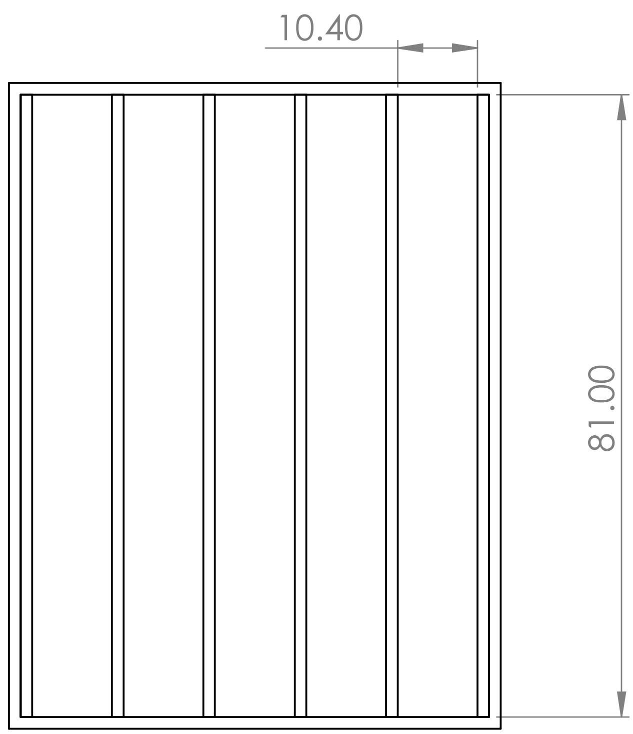
DIMENSIONS ARE IN INCHES AND DEGREES UNLESS OTHERWISE NOTED TOLERANCES: LINEAR ±0.005 ANGULAR: ±0.005		NAME	DATE
	DRAWN	M. STRATTON	07/27/20
DO NOT SCALE DRAWING	COMMENTS: PLYWOOD WILL NEED TO BE CUT TO FIT, POSSIBLY IN MULTIPLE PIECES FOR ONE SECTION. ALL STUDS ARE 2X4s. A 2ND PLYWOOD SECTION WILL NEED TO BE MADE FOR THE OPEN SIDE OF THE STUDS. A CUTOUT CAN BE MADE FOR A SLIDING BARNDOOR. PLWOOD IS 1/2IN THICK.		

SIZE	DWG. NO.	REV.
A	Bathroom_Wall	
SCALE:1:50	WEIGHT:	SHEET 1 OF 1





FRONT



BOTTOM

DIMENSIONS ARE IN INCHES
AND DEGREES UNLESS
OTHERWISE NOTED
TOLERANCES:
LINEAR ±0.005
ANGULAR: ±0.005

DO NOT SCALE DRAWING

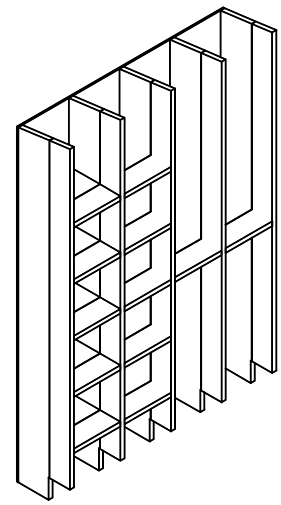
	NAME	DATE
DRAWN	M. STRATTON	07/22/20

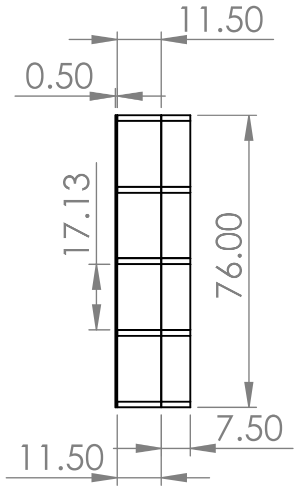
COMMENTS: ALL STUDS ARE 2X8s.
PLYWOOD IS 1/2IN THICK.

PROPRIETARY: THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF M. STRATTON IN CONJUNCTION WITH GVSU.

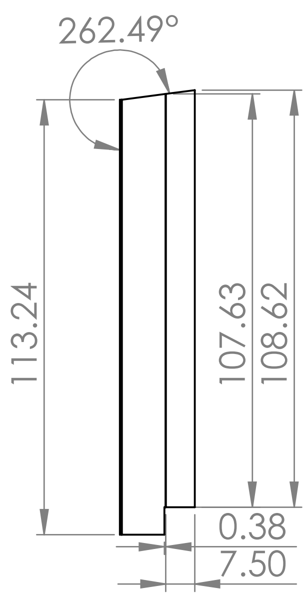
SIZE	DWG. NO.	REV.
A	Bed_Frame	

SCALE:1:25 | WEIGHT: | SHEET 1 OF 1

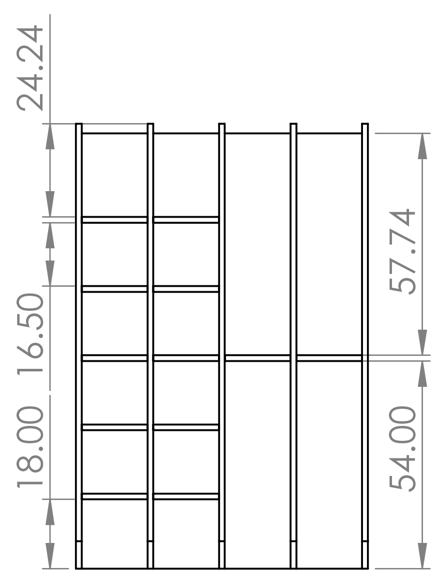




TOP



FRONT



RIGHT

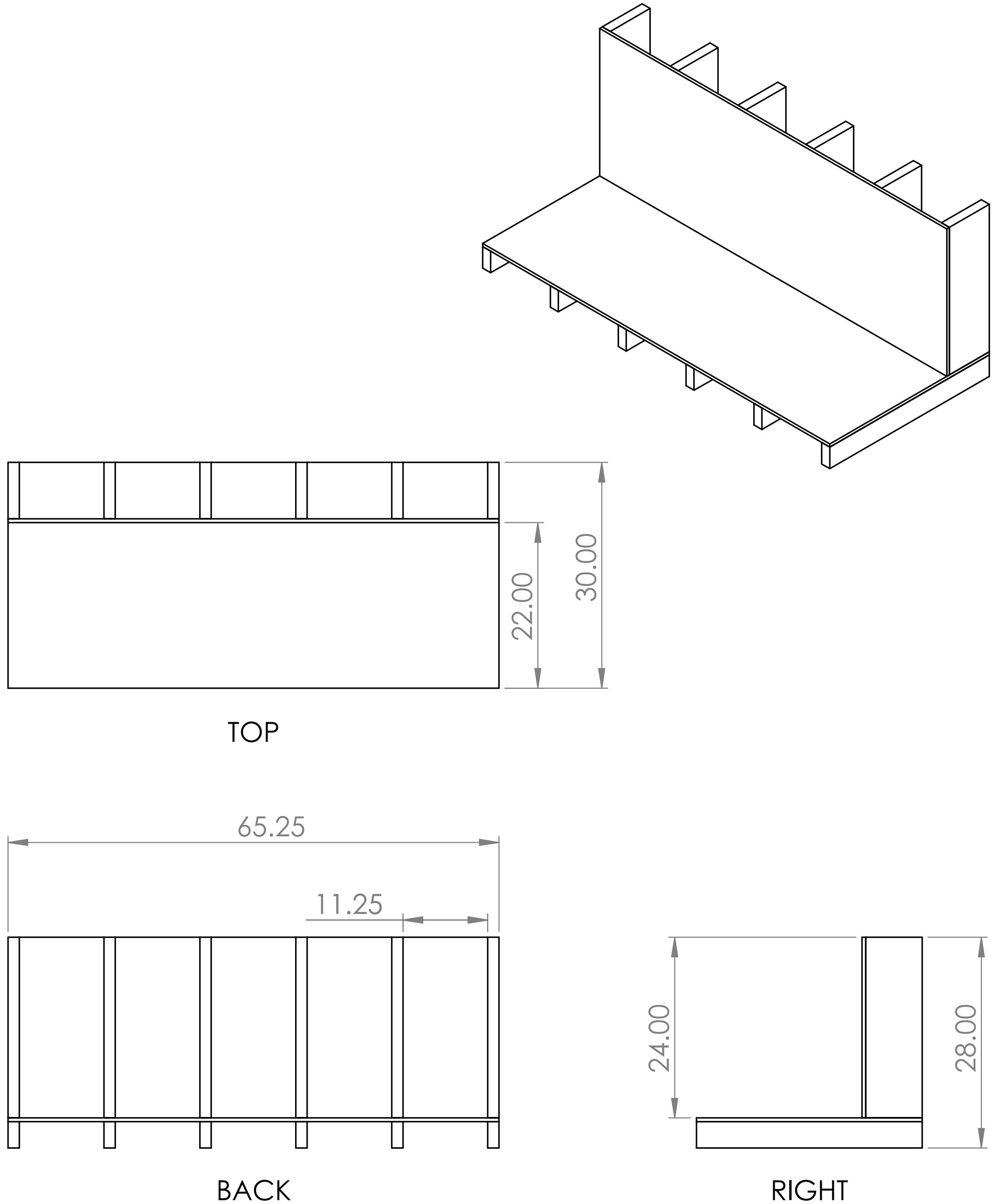
DIMENSIONS ARE IN INCHES AND DEGREES UNLESS OTHERWISE NOTED TOLERANCES: LINEAR ±0.005 ANGULAR: ±0.005		
	NAME	DATE
DRAWN	M. STRATTON	07/27/20
DO NOT SCALE DRAWING		

COMMENTS: PLYWOOD WAS USED A DOOR PLACEHOLDER AND WILL NEED TO BE CUT TO FIT, POSSIBLY IN MULTIPLE PIECES FOR ONE SECTION. STUDS AND SHELVES ARE 2X12s AND 2X8s.

PLWOOD IS 1/2IN THICK.

SIZE A | DWG. NO. Closet | REV.

SCALE:1:50 | WEIGHT: | SHEET 1 OF 1



DIMENSIONS ARE IN INCHES AND DEGREES UNLESS OTHERWISE NOTED
TOLERANCES:
LINEAR ±0.005
ANGULAR: ±0.005

DO NOT SCALE DRAWING

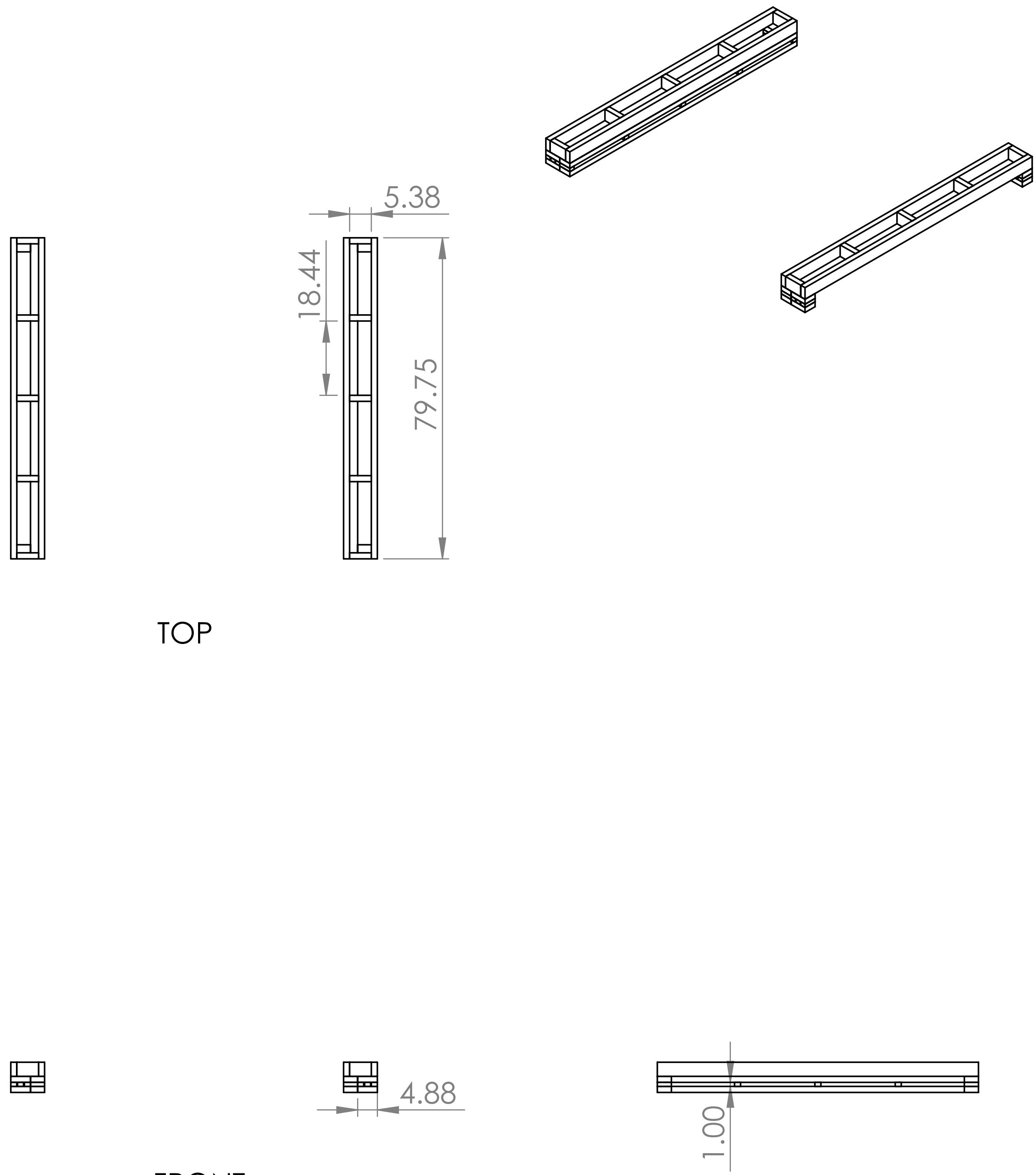
	NAME	DATE
DRAWN	M. STRATTON	07/22/20

COMMENTS: VERTICAL MEMBERS ARE 2X8s, HORIZONTAL MEMBERS ARE 2X4s, PLYWOOD IS 1/2IN THICK

PROPRIETARY: THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF M. STRATTON IN CONJUNCTION WITH GVSU.

SIZE **A** DWG. NO. Couch_Frame REV.

SCALE:1:20 WEIGHT: SHEET 1 OF 1



DIMENSIONS ARE IN INCHES AND DEGREES UNLESS OTHERWISE NOTED
TOLERANCES:
LINEAR ±0.005
ANGULAR: ±0.005

DO NOT SCALE DRAWING

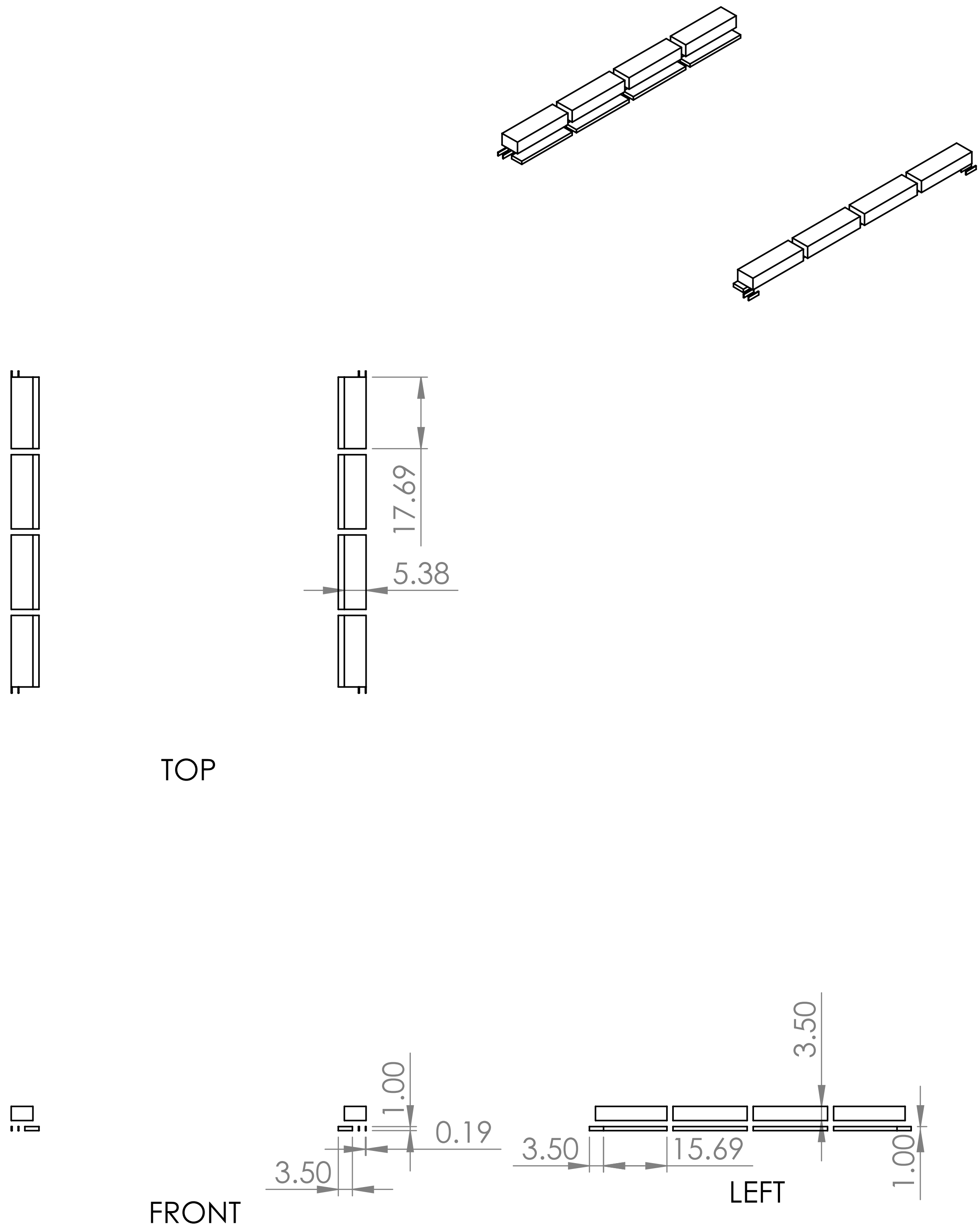
	NAME	DATE
DRAWN	M. STRATTON	07/27/20

COMMENTS: ALL STUDS ARE 2X4s.

FENDER SECTIONS ARE A MIRROR PAIR.

SIZE A | DWG. NO. Fenders | REV.

SCALE:1:40 | WEIGHT: | SHEET 1 OF 1



DIMENSIONS ARE IN INCHES AND DEGREES UNLESS OTHERWISE NOTED
TOLERANCES:
LINEAR ±0.005
ANGULAR: ±0.005

DO NOT SCALE DRAWING

	NAME	DATE
DRAWN	M. STRATTON	07/27/20

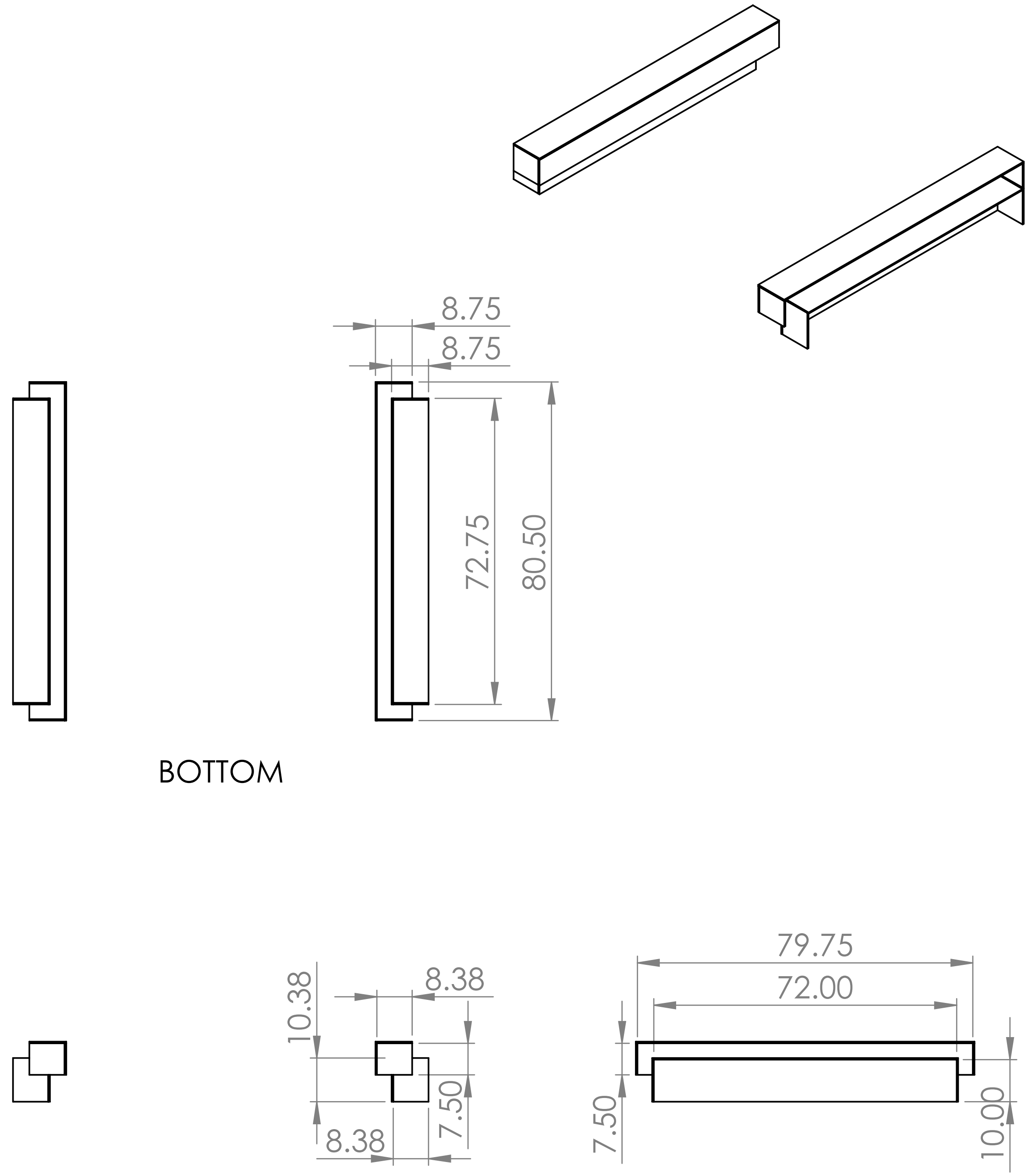
COMMENTS: INSULATION WILL NEED TO BE CUT TO FIT, POSSIBLY IN MULTIPLE PIECES FOR ONE SECTIONS.

FENDER INSULATION PIECES ARE A MIRROR PAIR.

PROPRIETARY: THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF M. STRATTON IN CONJUNCTION WITH GVSU.

SIZE **A** DWG. NO. Fender_Insul REV.

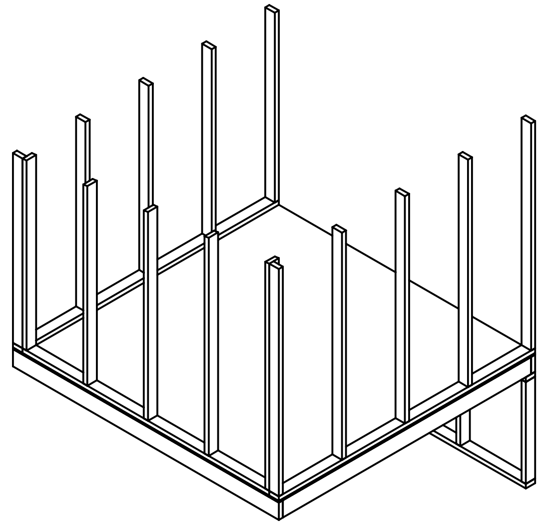
SCALE:1:40 WEIGHT: SHEET 1 OF 1

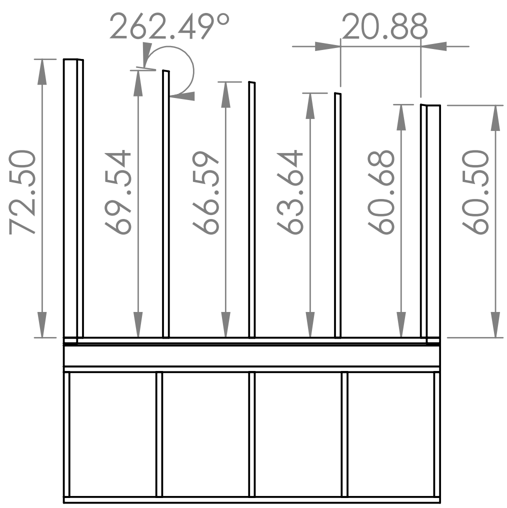


DIMENSIONS ARE IN INCHES AND DEGREES UNLESS OTHERWISE NOTED TOLERANCES: LINEAR ±0.005 ANGULAR: ±0.005		NAME / DATE	**PROPRIETARY:** THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF M. STRATTON IN CONJUNCTION WITH GVSU.
	DRAWN	M. STRATTON / 07/27/20	
DO NOT SCALE DRAWING	COMMENTS: PLYWOOD WILL NEED TO BE CUT TO FIT, POSSIBLY IN MULTIPLE PIECES FOR ONE SECTION. ALL PLYWOOD IS 3/8IN THICK. FENDER SECTIONS ARE A MIRROR PAIR.		SIZE A / DWG. NO. Fender_Sheathing / REV. SCALE:1:40 / WEIGHT: / SHEET 1 OF 1

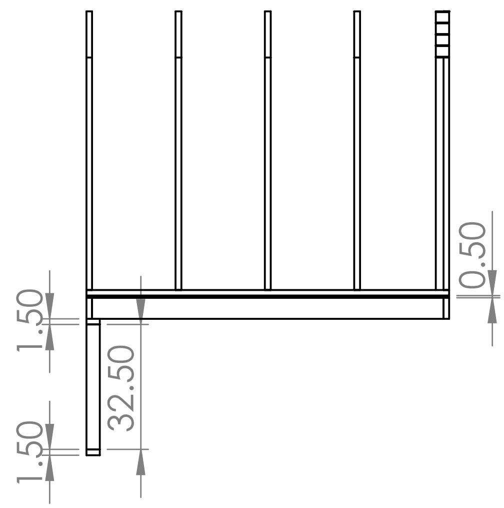
2 1

B B



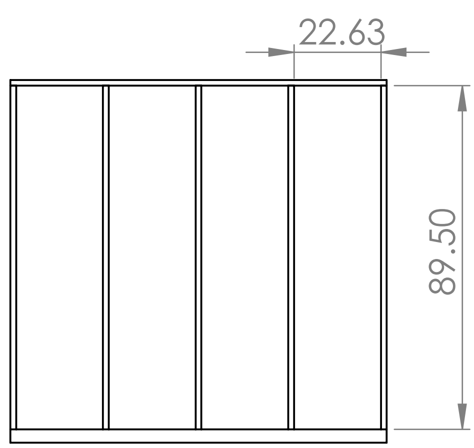


BACK

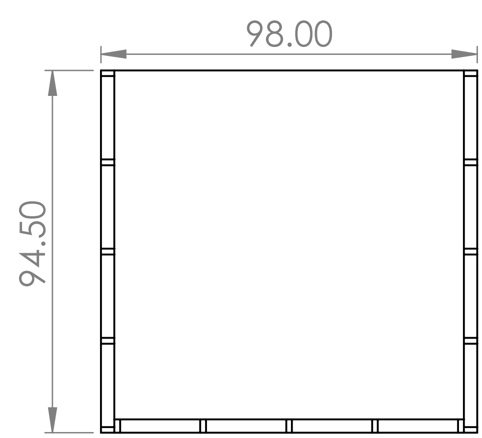


LEFT

A A



BOTTOM



TOP

DIMENSIONS ARE IN INCHES AND DEGREES UNLESS OTHERWISE NOTED
TOLERANCES:
LINEAR ±0.005
ANGULAR: ±0.005

DO NOT SCALE DRAWING

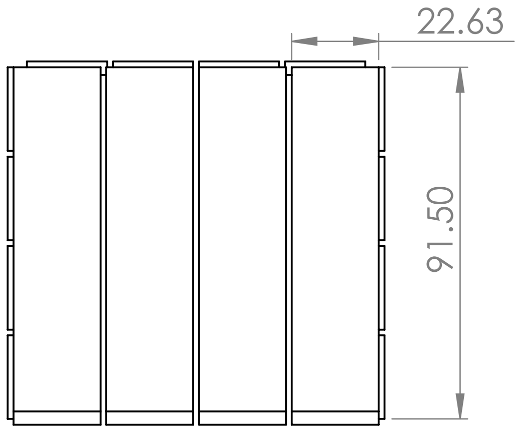
	NAME	DATE
DRAWN	M. STRATTON	07/22/20

COMMENTS: ALL STUDS ARE 2X4s EXCEPT FOR GOOSE FLOOR JOISTS, WHICH ARE 2X6s. PLYWOOD SHEET IS 1/2IN.

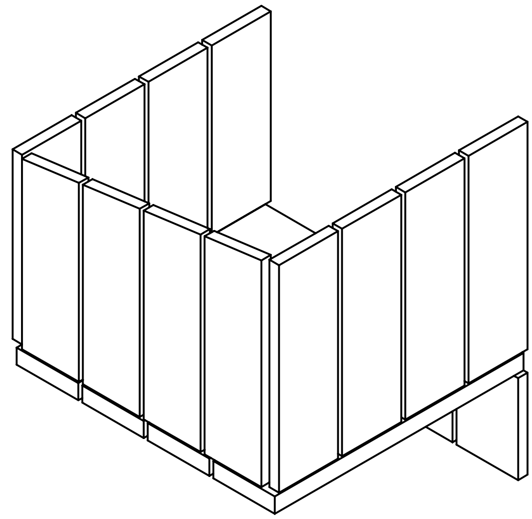
SIZE	DWG. NO.	REV.
A	Goose	

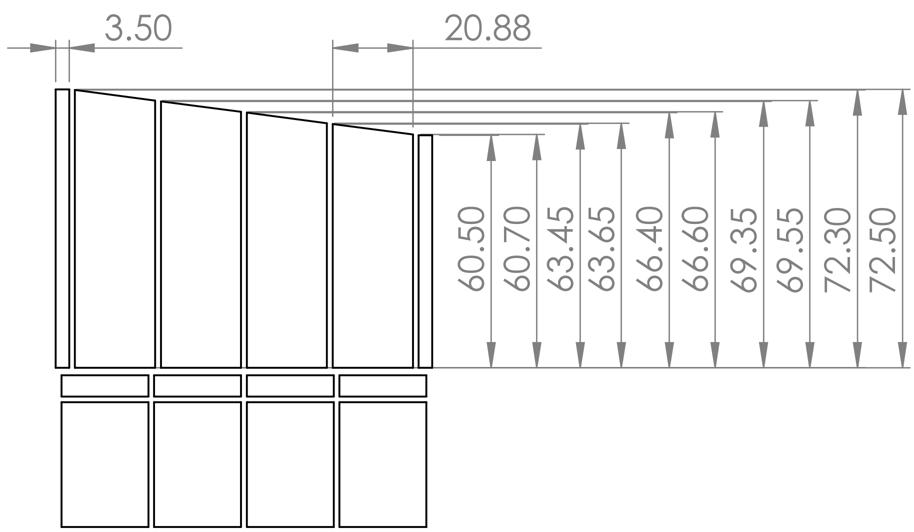
SCALE:1:50 WEIGHT: SHEET 1 OF 1

2 1

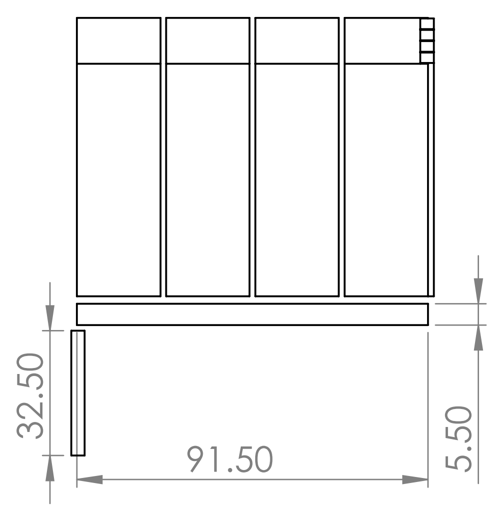


BOTTOM





BACK



LEFT

DIMENSIONS ARE IN INCHES AND DEGREES UNLESS OTHERWISE NOTED
TOLERANCES:
LINEAR ±0.005
ANGULAR: ±0.005

DO NOT SCALE DRAWING

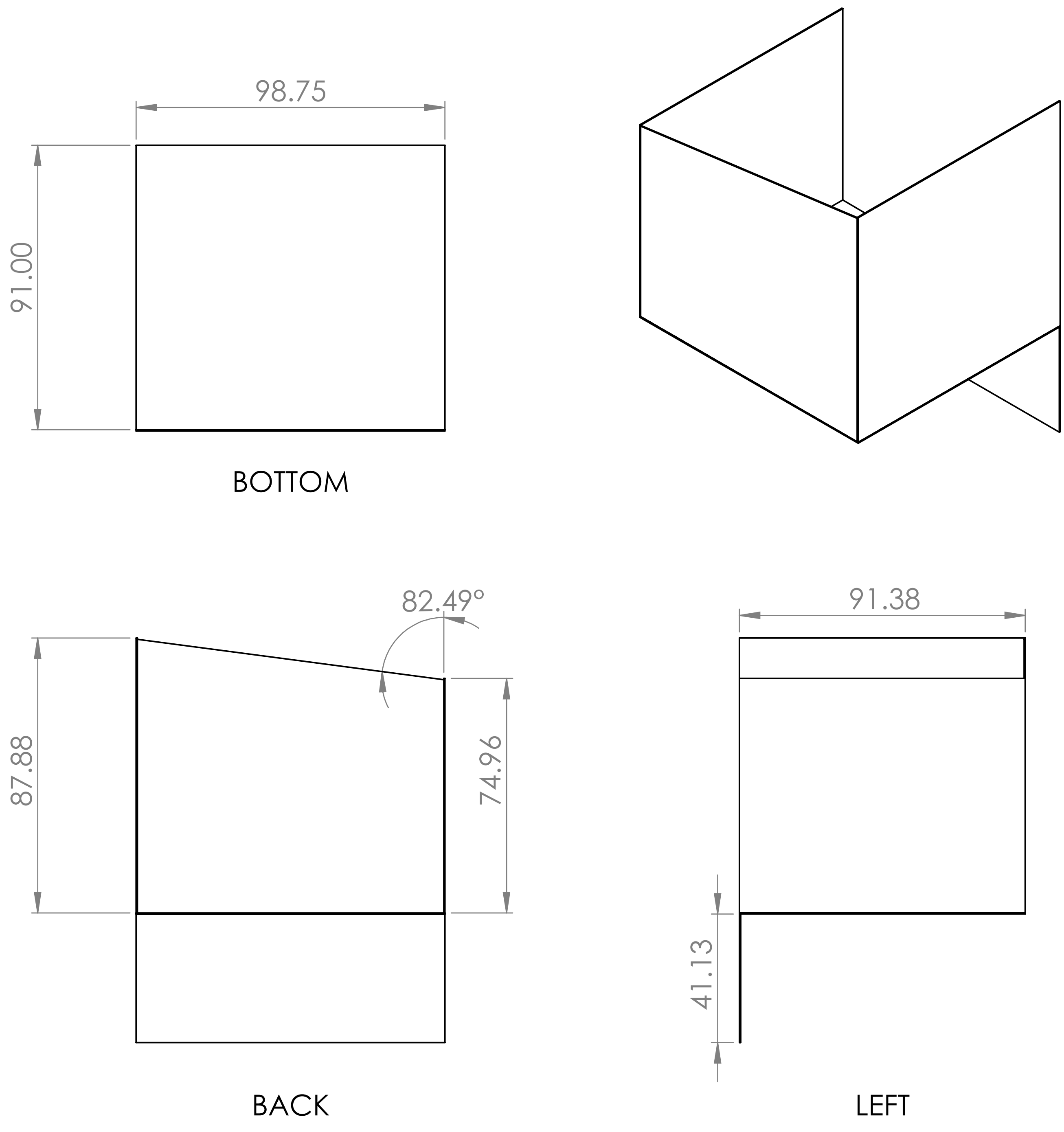
	NAME	DATE
DRAWN	M. STRATTON	07/24/20

COMMENTS: INSULATION WILL NEED TO BE CUT TO FIT, POSSIBLY IN MULTIPLE PIECES FOR ONE SECTION.

ALL INSULATION IS FOR 2X4s, EXCEPT FOR FLOOR PIECES, WHICH ARE FOR 2X6s.

SIZE A | DWG. NO. Goose_Insul | REV.

SCALE:1:50 | WEIGHT: | SHEET 1 OF 1



DIMENSIONS ARE IN INCHES
AND DEGREES UNLESS
OTHERWISE NOTED
TOLERANCES:
LINEAR ±0.005
ANGULAR: ±0.005

DO NOT SCALE DRAWING

	NAME	DATE
DRAWN	M. STRATTON	07/22/20

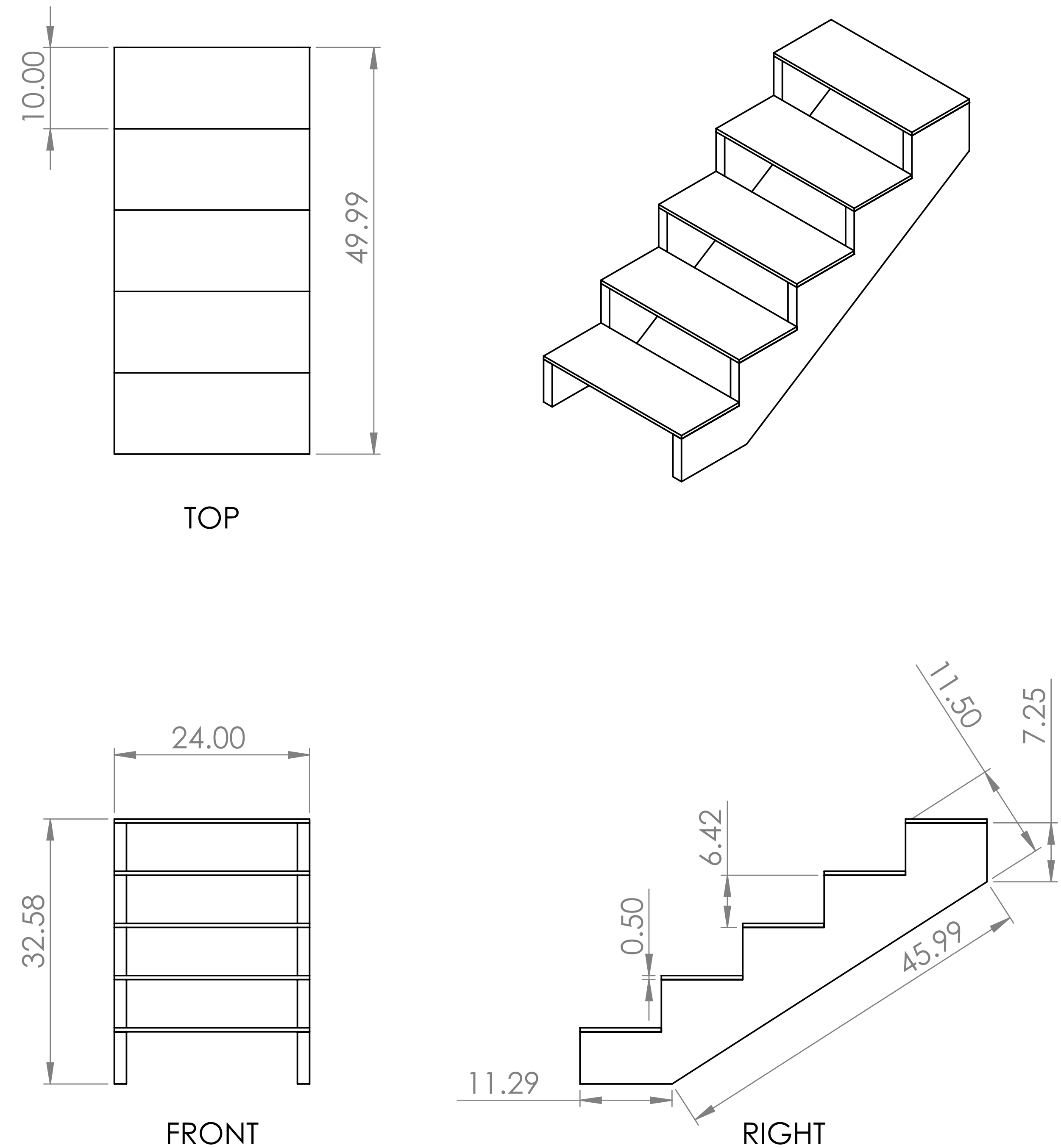
COMMENTS: PLYWOOD WILL NEED TO BE CUT TO FIT, POSSIBLY IN MULTIPLE PIECES FOR ONE SECTION.

ALL PLYWOOD IS 3/8IN THICK.

PROPRIETARY: THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF M. STRATTON IN CONJUNCTION WITH GVSU.

SIZE	DWG. NO.	REV.
A	Goose_Sheathing	

SCALE:1:50 | WEIGHT: | SHEET 1 OF 1



DIMENSIONS ARE IN INCHES AND DEGREES UNLESS OTHERWISE NOTED
TOLERANCES:
LINEAR ±0.005
ANGULAR: ±0.005

DO NOT SCALE DRAWING

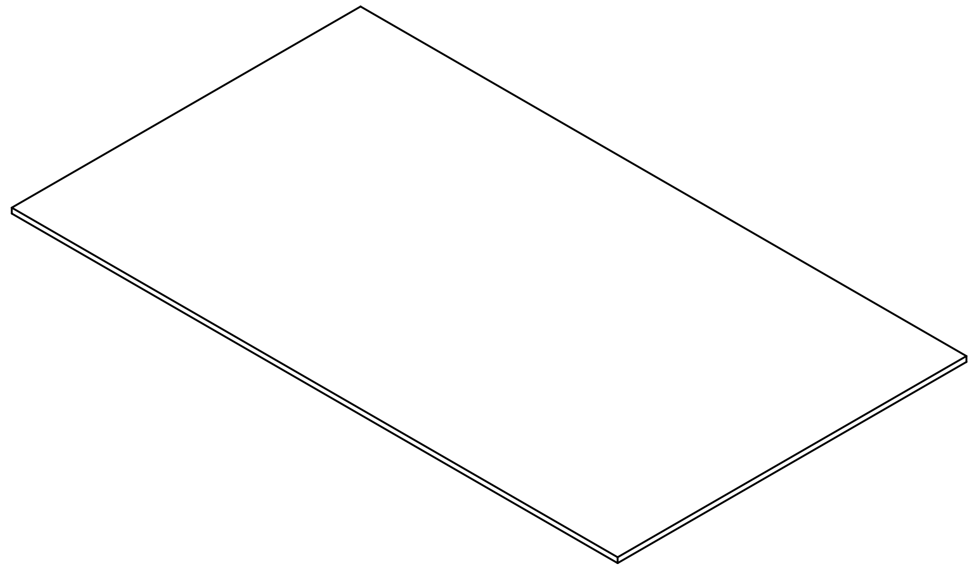
	NAME	DATE
DRAWN	M. STRATTON	07/22/20

COMMENTS: MAIN JOISTS ARE 2X12s. TRED IS 1/2IN THICK.

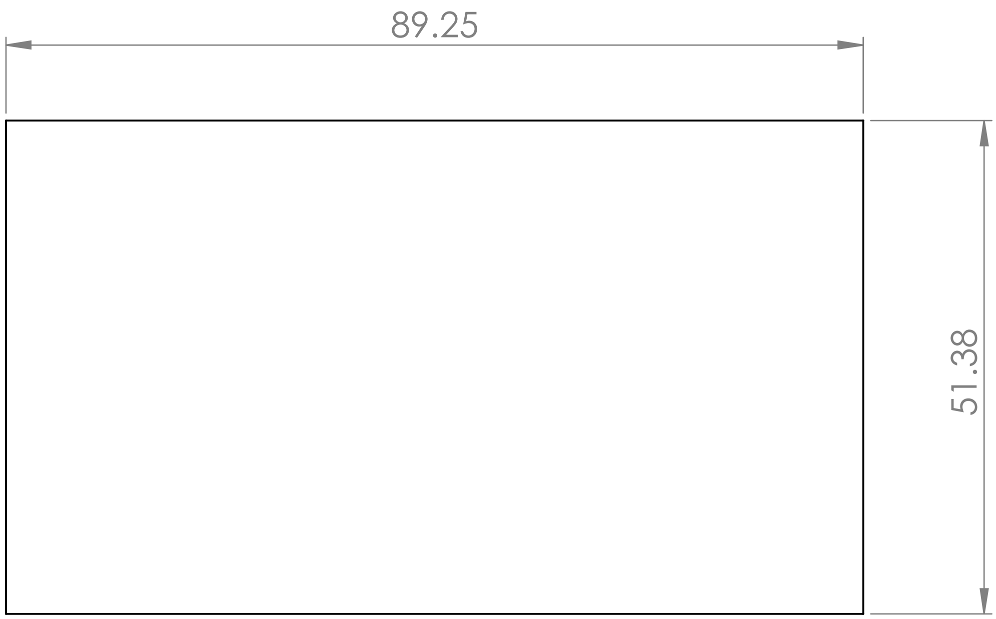
PROPRIETARY: THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF M. STRATTON IN CONJUNCTION WITH GVSU.

SIZE **A** DWG. NO. Stairs REV.

SCALE:1:20 WEIGHT: SHEET 1 OF 1

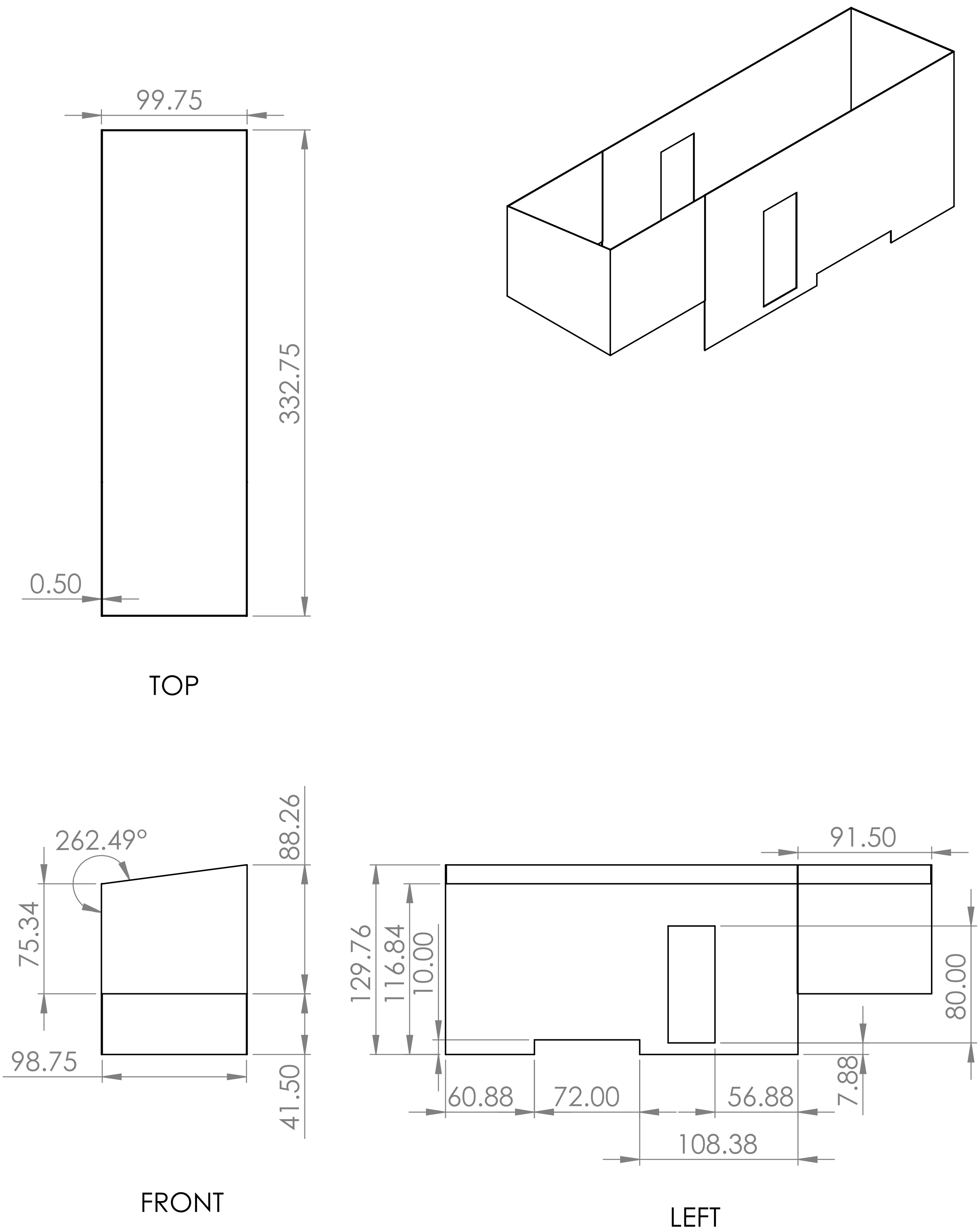


FRONT



TOP

DIMENSIONS ARE IN INCHES AND DEGREES UNLESS OTHERWISE NOTED TOLERANCES: LINEAR ±0.005 ANGULAR: ±0.005		NAME	DATE	PROPRIETARY: THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF M. STRATTON IN CONJUNCTION WITH GVSU.
	DRAWN	M. STRATTON	07/22/20	
	COMMENTS: 3/4IN THICK TILE WAS USED. MULTIPLE PIECES WILL NEED TO BE CUT TO FIT SHOWN DIMENSIONS.			SIZE A — DWG. NO. Bathroom_Tile_20ft — REV.
DO NOT SCALE DRAWING				SCALE:1:20 — WEIGHT: — SHEET 1 OF 1



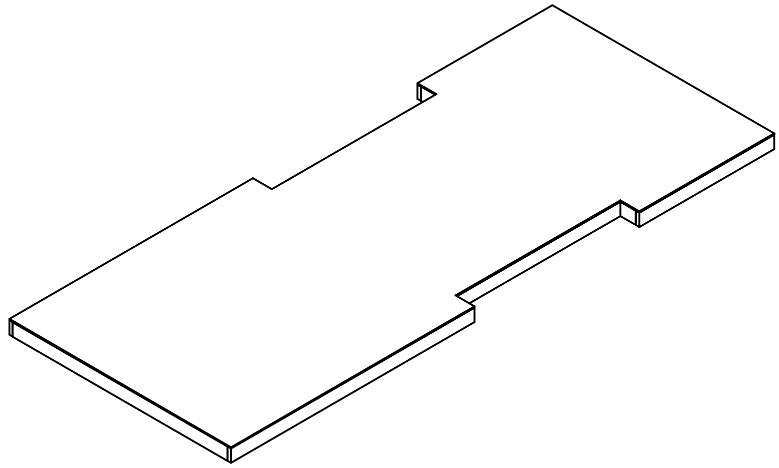
DIMENSIONS ARE IN INCHES AND DEGREES UNLESS OTHERWISE NOTED TOLERANCES: LINEAR ±0.005 ANGULAR: ±0.005		NAME / DATE
	DRAWN	M. STRATTON / 07/27/20
DO NOT SCALE DRAWING	COMMENTS: SIDING WILL NEED TO BE CUT TO FIT, POSSIBLY IN MULTIPLE PIECES FOR ONE SECTION. PLACEHOLDER THICKNESS IS 1/2IN.	

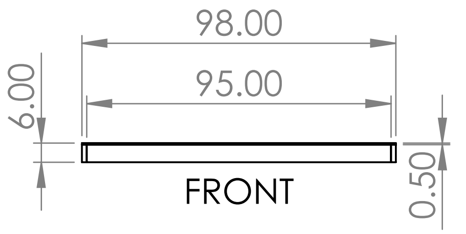
PROPRIETARY: THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF M. STRATTON IN CONJUNCTION WITH GVSU.

SIZE	DWG. NO.	REV.
A	Exterior_Siding_20ft	
SCALE:1:100	WEIGHT:	SHEET 1 OF 1

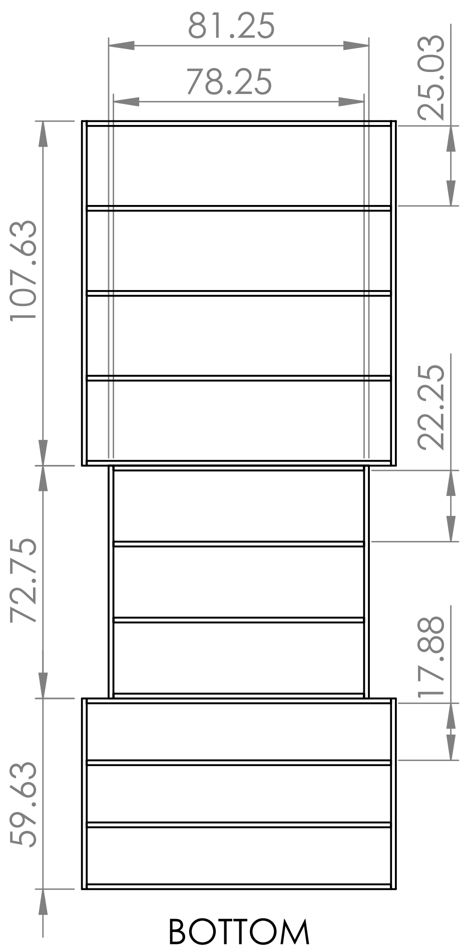
2 | 1

B





FRONT

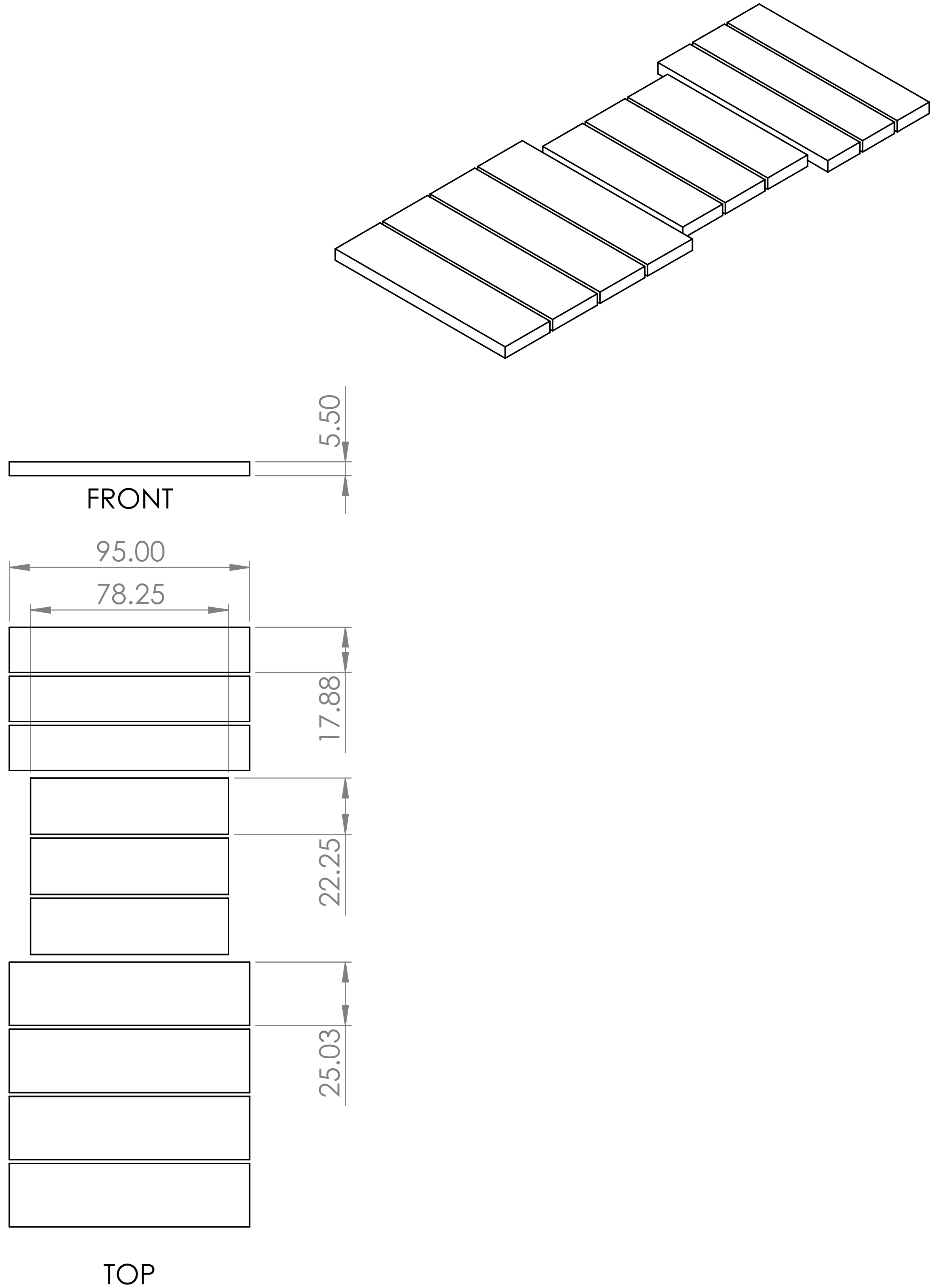


BOTTOM

A

DIMENSIONS ARE IN INCHES AND DEGREES UNLESS OTHERWISE NOTED TOLERANCES: LINEAR ±0.005 ANGULAR: ±0.005		NAME	DATE	**PROPRIETARY:** THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF M. STRATTON IN CONJUNCTION WITH GVSU.
	DRAWN	M. STRATTON	07/22/20	
	COMMENTS: ALL FLOOR JOISTS ARE 2X6s. PLYWOOD WILL NEED TO BE CUT TO FIT, POSSIBLY IN MULTIPLE PIECES FOR ONE SECTION. PLYWOOD IS 1/2IN THICK.			SIZE A / DWG. NO. Floor_20ft / REV.
DO NOT SCALE DRAWING				SCALE:1:60 / WEIGHT: / SHEET 1 OF 1

2 | 1



DIMENSIONS ARE IN INCHES AND DEGREES UNLESS OTHERWISE NOTED
TOLERANCES:
LINEAR ±0.005
ANGULAR: ±0.005

DO NOT SCALE DRAWING

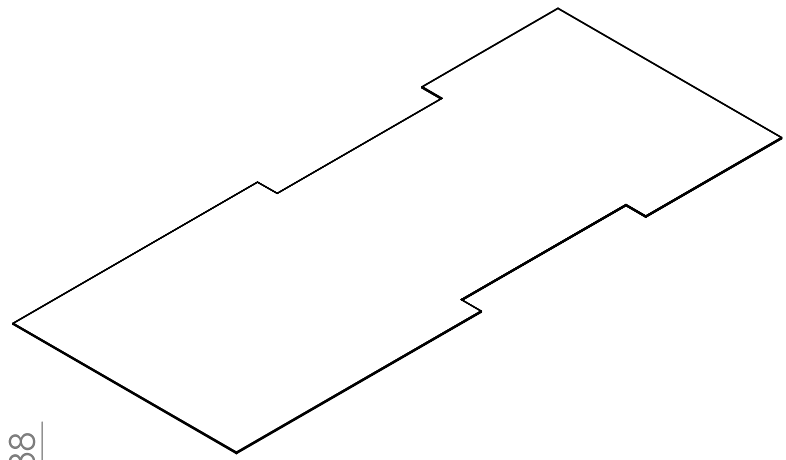
	NAME	DATE
DRAWN	M. STRATTON	07/24/20

COMMENTS: INSULATION WILL NEED TO BE CUT TO FIT, POSSIBLY IN MULTIPLE PIECES FOR ONE SECTION.

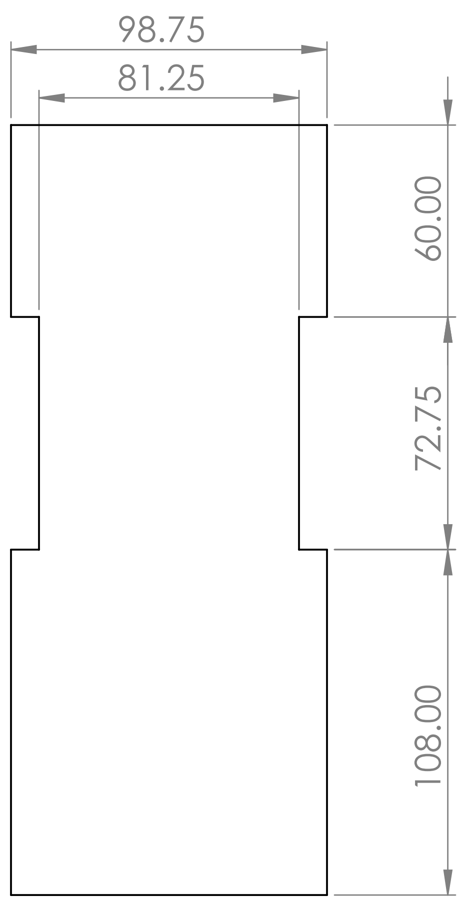
INSULATION TO FIT 2X6 FLOOR JOISTS

SIZE A | DWG. NO. Floor_Insul_20ft | REV.

SCALE:1:60 | WEIGHT: | SHEET 1 OF 1



FRONT



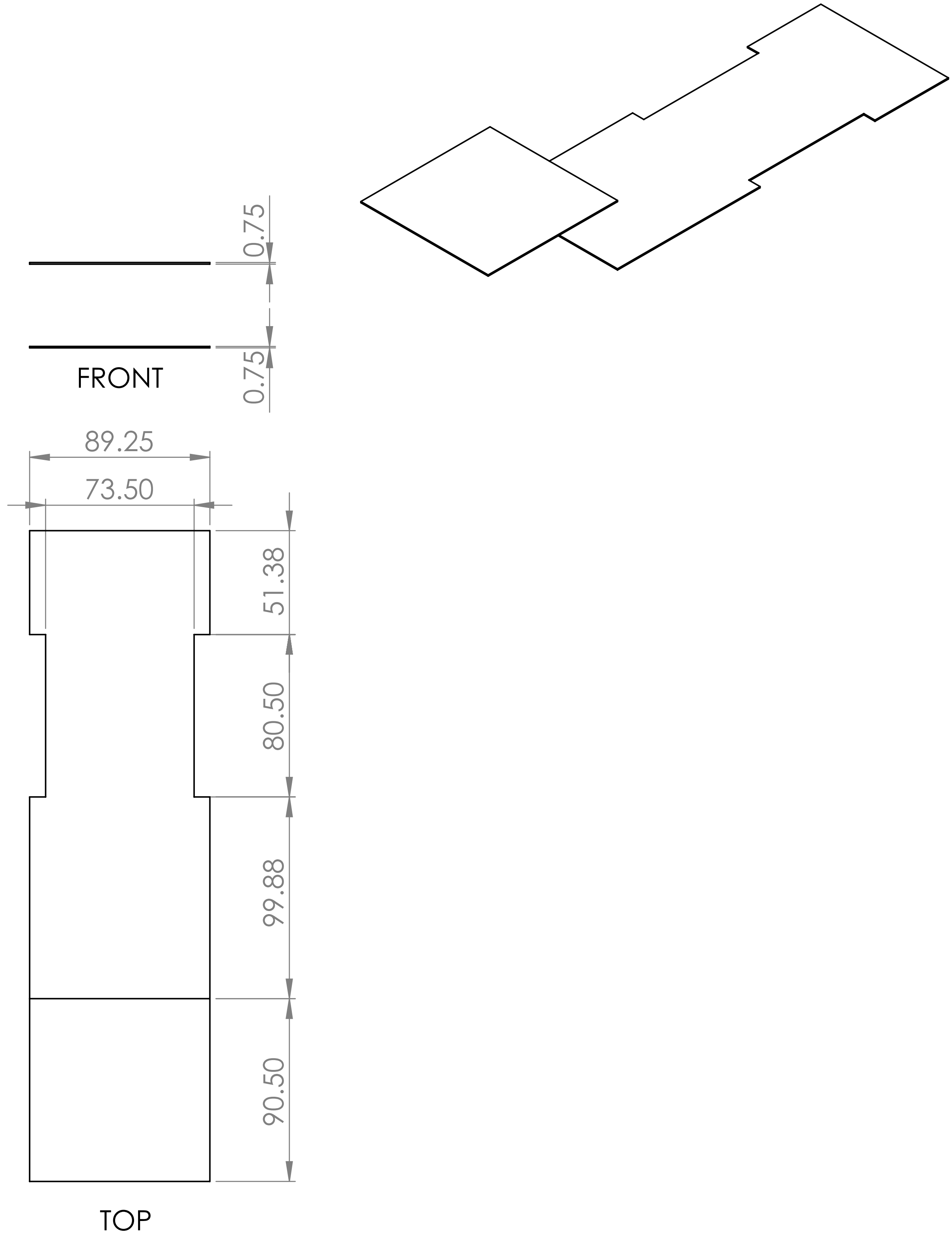
TOP

DIMENSIONS ARE IN INCHES AND DEGREES UNLESS OTHERWISE NOTED TOLERANCES: LINEAR ±0.005 ANGULAR: ±0.005		NAME:	DATE:
	DRAWN	M. STRATTON	07/24/20
DO NOT SCALE DRAWING	COMMENTS: PLYWOOD WILL NEED TO BE CUT TO FIT, POSSIBLY IN MULTIPLE PIECES FOR ONE SECTION. PLYWOOD IS 3/8IN THICK.		

PROPRIETARY: THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF M. STRATTON IN CONJUNCTION WITH GVSU.

SIZE: A | DWG. NO.: Floor_Sheathing_20ft | REV.:

SCALE:1:60 | WEIGHT: | SHEET 1 OF 1



DIMENSIONS ARE IN INCHES AND DEGREES UNLESS OTHERWISE NOTED
TOLERANCES:
LINEAR ±0.005
ANGULAR: ±0.005

DO NOT SCALE DRAWING

	NAME	DATE
DRAWN	M. STRATTON	07/24/20

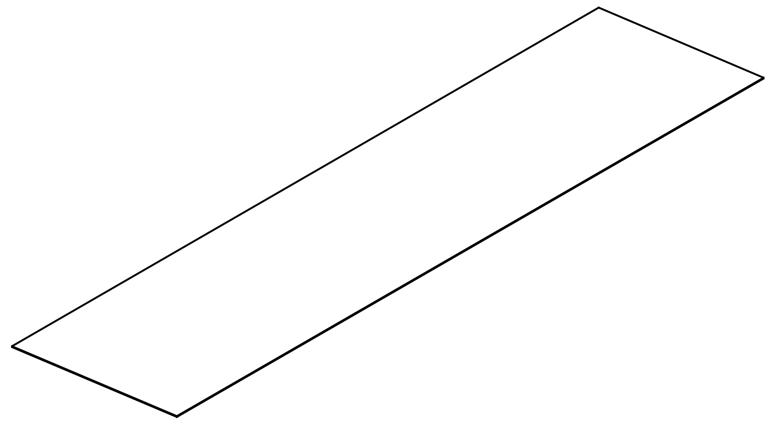
COMMENTS: FLOOR WILL NEED TO BE CUT TO FIT, POSSIBLY IN MULTIPLE PIECES FOR ONE SECTION.

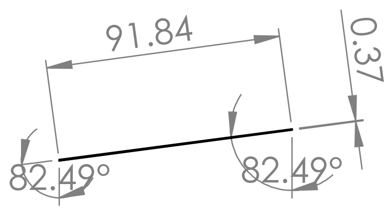
FLOOR IS 3/4IN THICK.

PROPRIETARY: THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF M. STRATTON IN CONJUNCTION WITH GVSU.

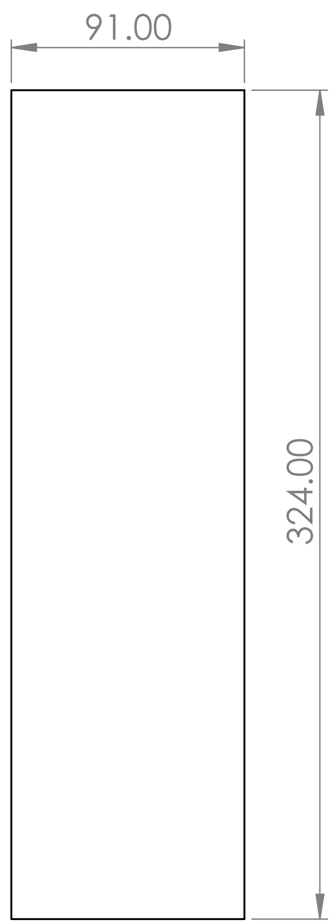
SIZE A | DWG. NO. Floor_Top_20ft | REV.

SCALE:1:75 | WEIGHT: | SHEET 1 OF 1





FRONT



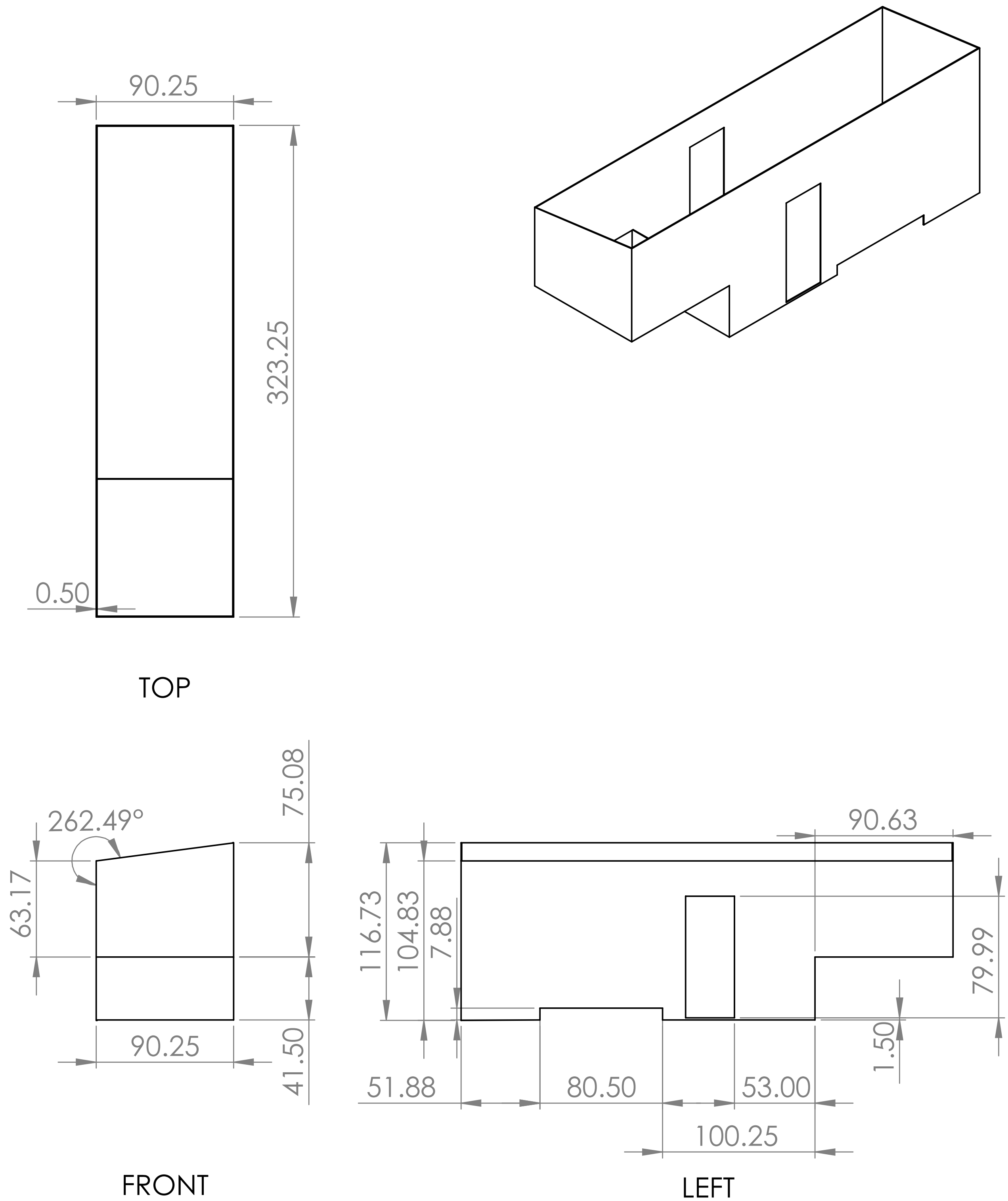
TOP

DIMENSIONS ARE IN INCHES AND DEGREES UNLESS OTHERWISE NOTED TOLERANCES: LINEAR ±0.005 ANGULAR: ±0.005		NAME	DATE
	DRAWN	M. STRATTON	07/24/20
	COMMENTS: PLYWOOD WILL NEED TO BE CUT TO FIT, POSSIBLY IN MULTIPLE PIECES FOR ONE SECTION. PLYWOOD IS 3/8IN THICK.		
DO NOT SCALE DRAWING			

PROPRIETARY: THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF M. STRATTON IN CONJUNCTION WITH GVSU.

SIZE	DWG. NO.	REV.
A	nt_Roof_Sheathing_20ft	

SCALE:1:75	WEIGHT:	SHEET 1 OF 1



DIMENSIONS ARE IN INCHES AND DEGREES UNLESS OTHERWISE NOTED
TOLERANCES:
LINEAR ±0.005
ANGULAR: ±0.005

DO NOT SCALE DRAWING

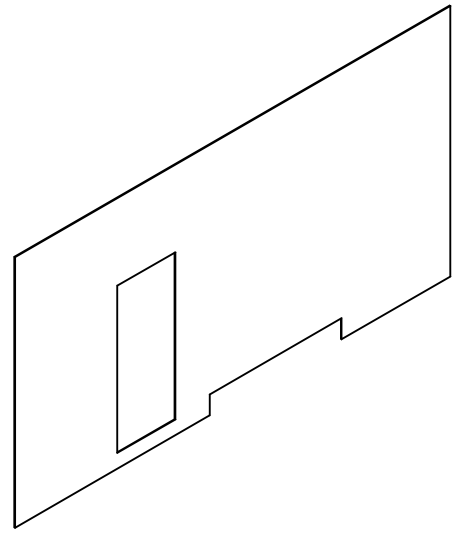
	NAME	DATE
DRAWN	M. STRATTON	09/23/20

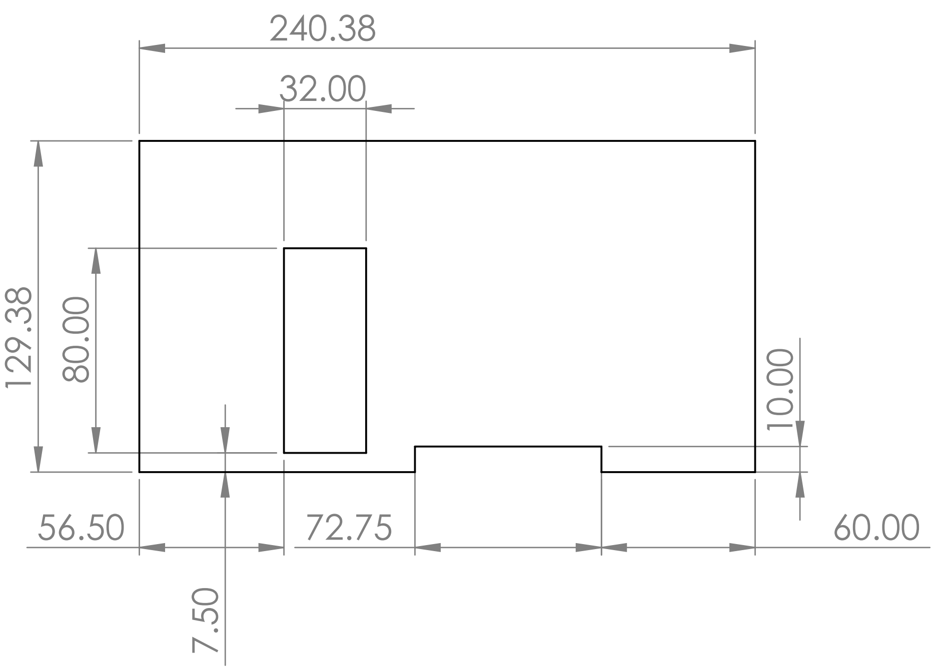
COMMENTS: SHIPLAP WILL NEED TO BE CUT TO FIT, POSSIBLY IN MULTIPLE PIECES FOR ONE SECTION.

PLACEHOLDER THICKNESS IS 1/2IN.

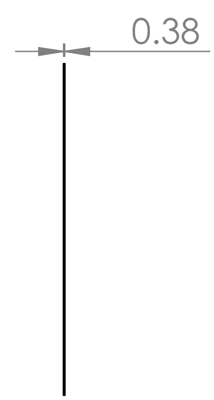
SIZE	DWG. NO.	REV.
A	Interior_Shiplap_20ft	

SCALE:1:100 WEIGHT: SHEET 1 OF 1





RIGHT



FRONT

DIMENSIONS ARE IN INCHES
AND DEGREES UNLESS
OTHERWISE NOTED
TOLERANCES:
LINEAR ±0.005
ANGULAR: ±0.005

DO NOT SCALE DRAWING

	NAME	DATE
DRAWN	M. STRATTON	07/27/20

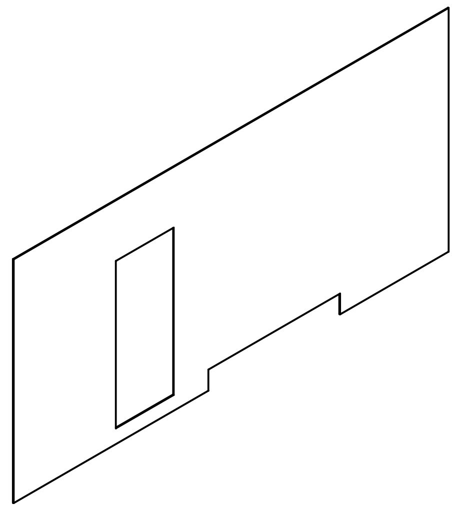
COMMENTS: PLYWOOD WILL NEED TO BE CUT TO FIT, POSSIBLY IN MULTIPLE PIECES FOR ONE SECTION.

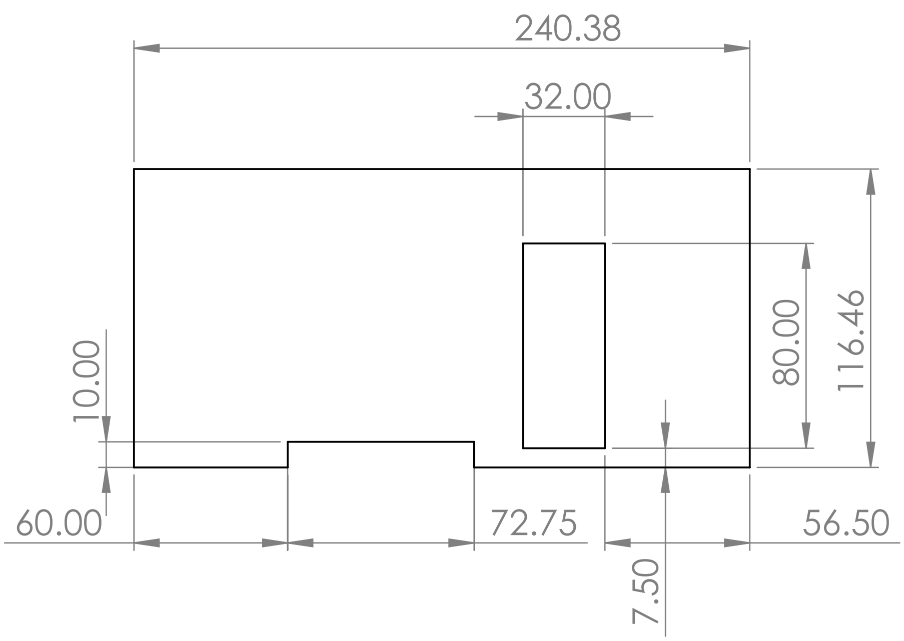
ALL PLYWOOD IS 3/8IN THICK.

PROPRIETARY: THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF M. STRATTON IN CONJUNCTION WITH GVSU.

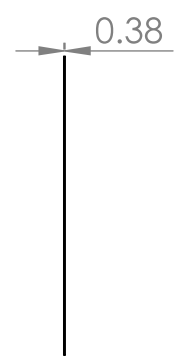
SIZE A | DWG. NO. L_Side_Wall_Sheathing_20ft | REV.

SCALE:1:75 | WEIGHT: | SHEET 1 OF 1





LEFT



FRONT

DIMENSIONS ARE IN INCHES AND DEGREES UNLESS OTHERWISE NOTED
TOLERANCES:
LINEAR ±0.005
ANGULAR: ±0.005

DO NOT SCALE DRAWING

	NAME	DATE
DRAWN	M. STRATTON	07/27/20

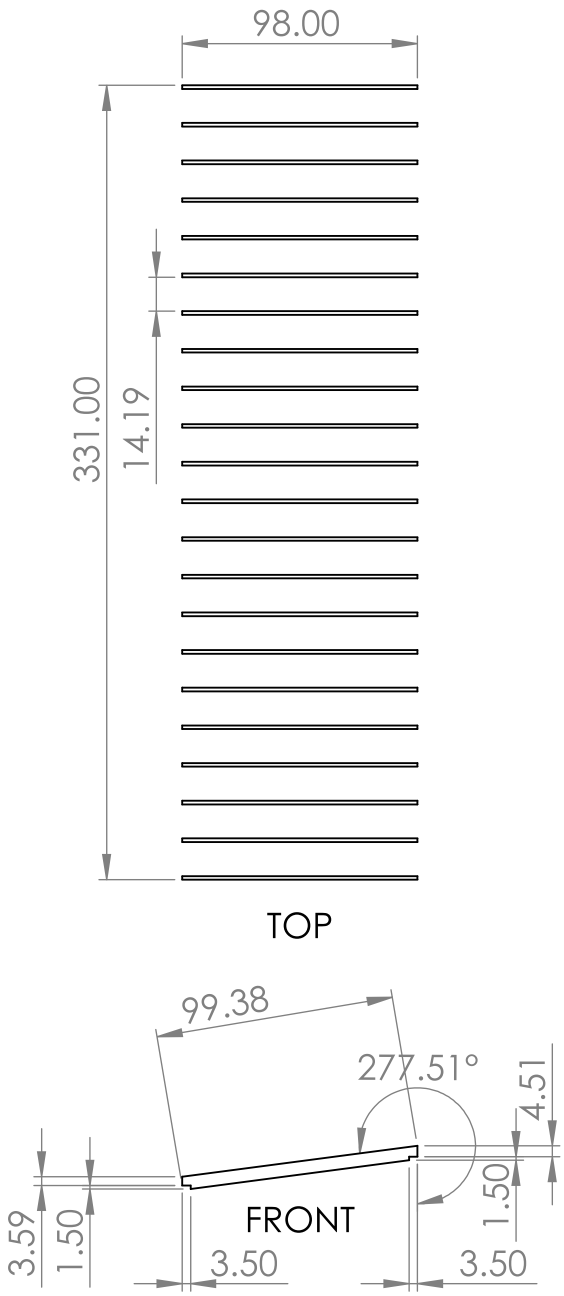
COMMENTS: PLYWOOD WILL NEED TO BE CUT TO FIT, POSSIBLY IN MULTIPLE PIECES FOR ONE SECTION.

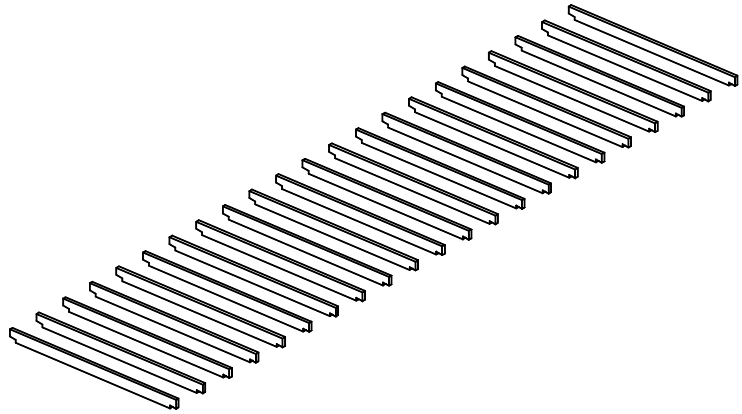
ALL PLYWOOD IS 3/8IN THICK.

PROPRIETARY: THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF M. STRATTON IN CONJUNCTION WITH GVSU.

SIZE **A** DWG. NO. R_Side_Wall_Sheathing_20ft REV.

SCALE:1:75 WEIGHT: SHEET 1 OF 1



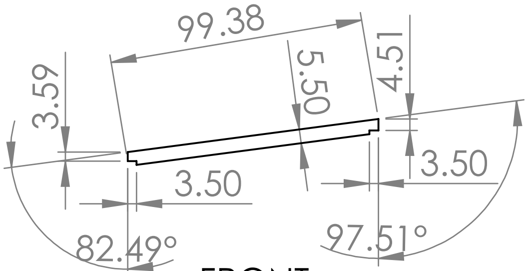


DIMENSIONS ARE IN INCHES AND DEGREES UNLESS OTHERWISE NOTED TOLERANCES: LINEAR ±0.005 ANGULAR: ±0.005		NAME	DATE
	DRAWN	M. STRATTON	07/22/20
	COMMENTS: ALL ROOF JOISTS ARE 2X6s, 22 IN TOTAL.		
DO NOT SCALE DRAWING			

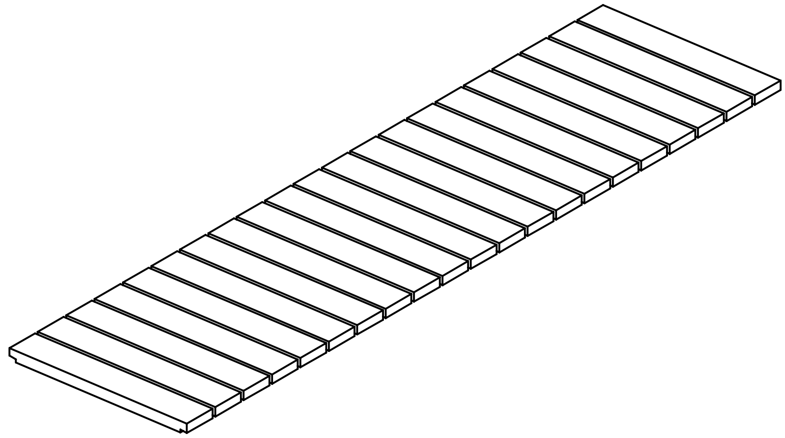
PROPRIETARY: THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF M. STRATTON IN CONJUNCTION WITH GVSU.

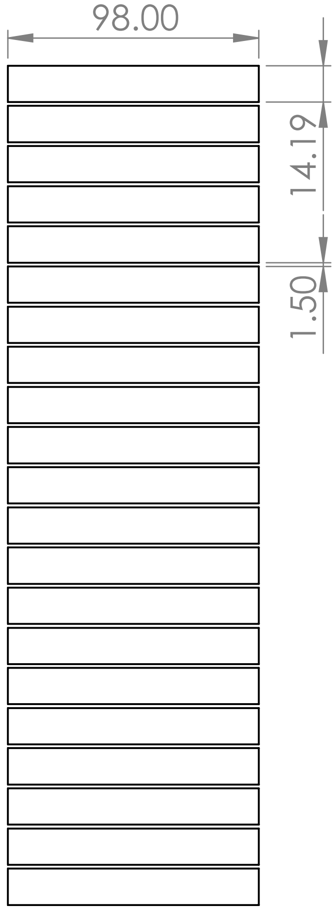
SIZE	DWG. NO.	REV.
A	Roof_20ft	

SCALE:1:80 | WEIGHT: | SHEET 1 OF 1



FRONT





TOP

DIMENSIONS ARE IN INCHES AND DEGREES UNLESS OTHERWISE NOTED
TOLERANCES:
LINEAR ±0.005
ANGULAR: ±0.005

DO NOT SCALE DRAWING

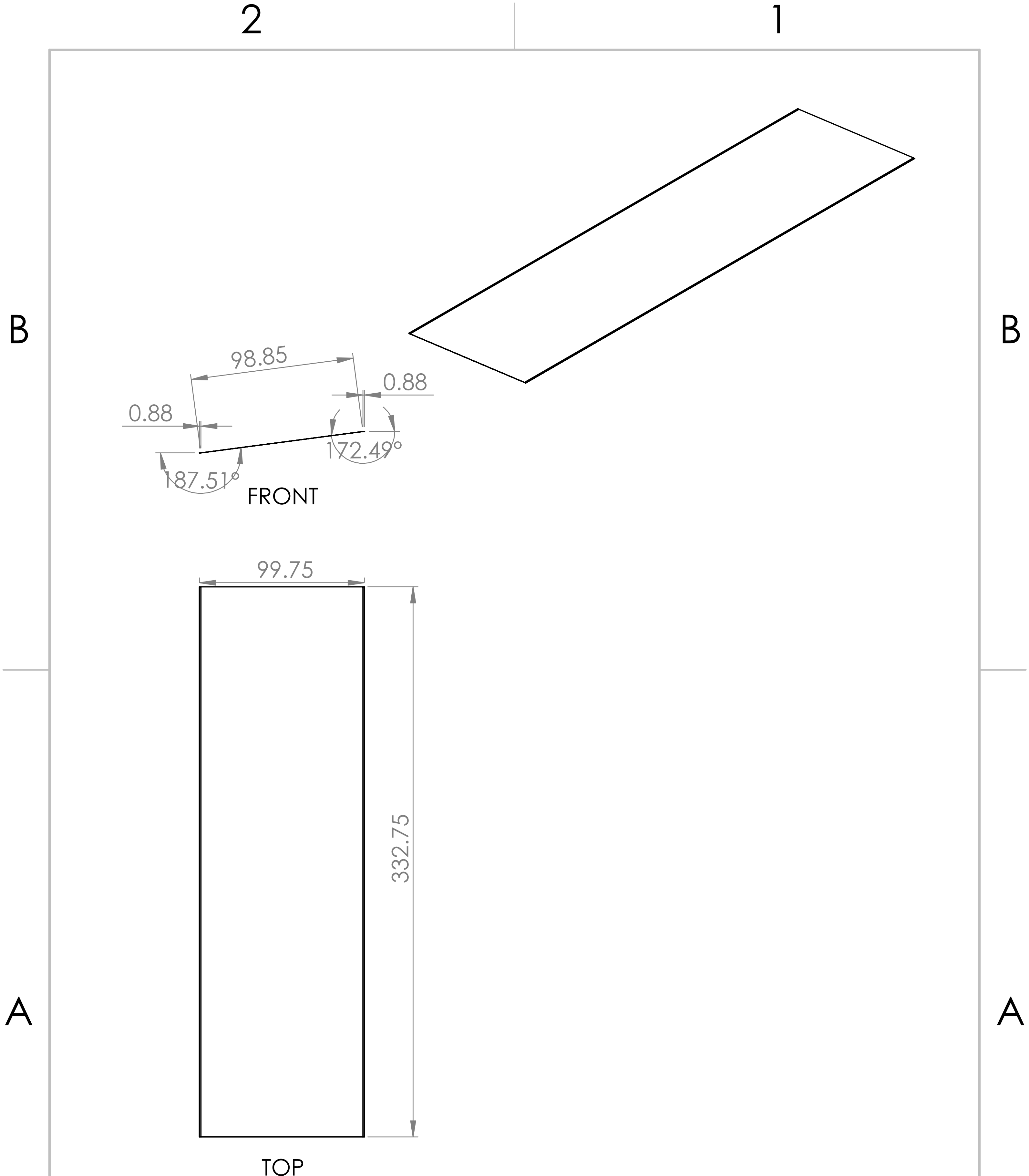
	NAME	DATE
DRAWN	M. STRATTON	07/24/20

COMMENTS: INSULATION WILL NEED TO BE CUT TO FIT, POSSIBLY IN MULTIPLE PIECES FOR ONE SECTION. 21 SECTIONS IN TOTAL.

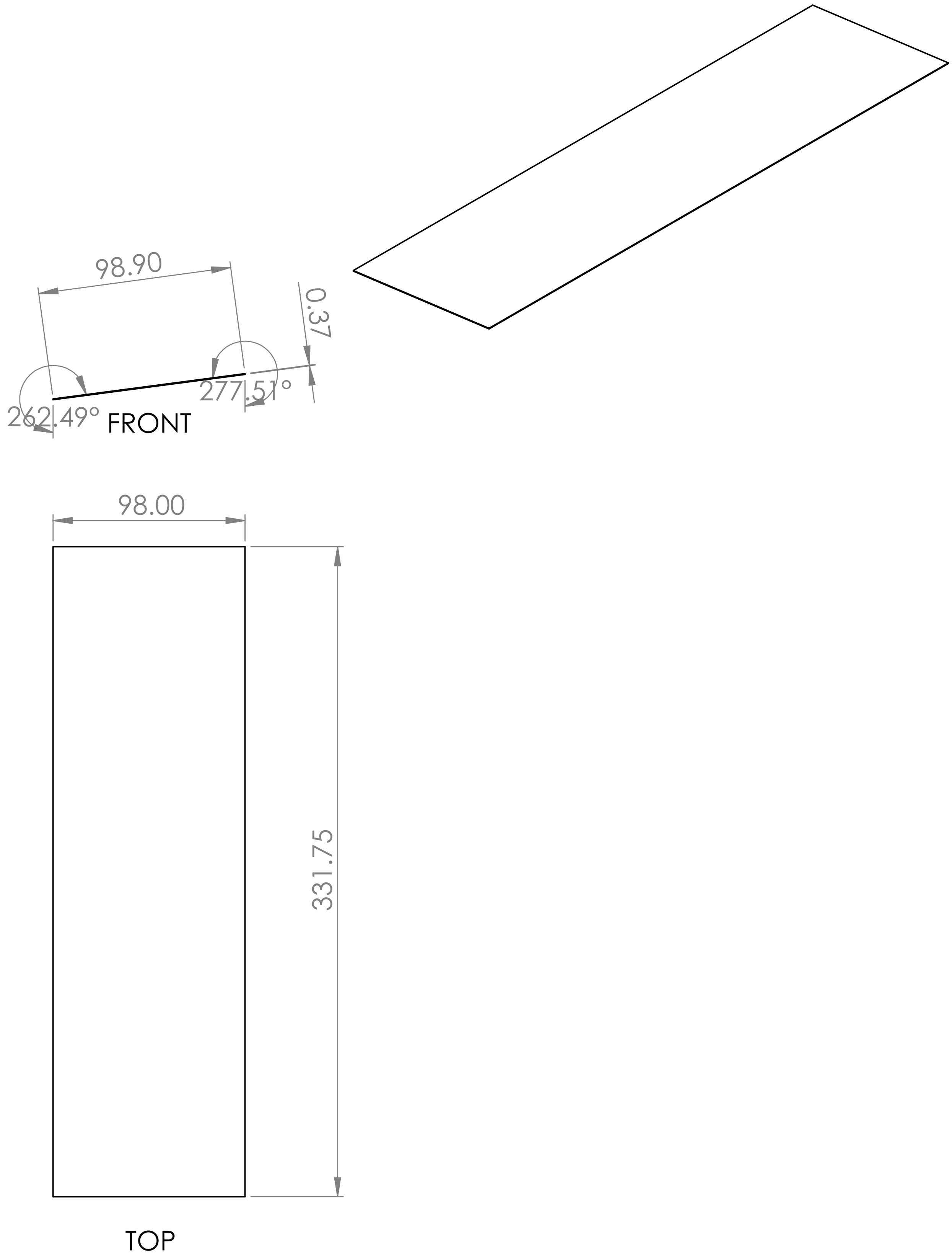
PROPRIETARY: THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF M. STRATTON IN CONJUNCTION WITH GVSU.

SIZE A DWG. NO. Roof_Insul_20ft REV.

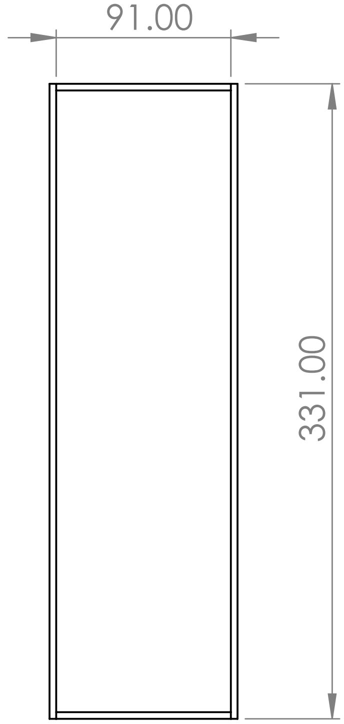
SCALE:1:75 WEIGHT: SHEET 1 OF 1



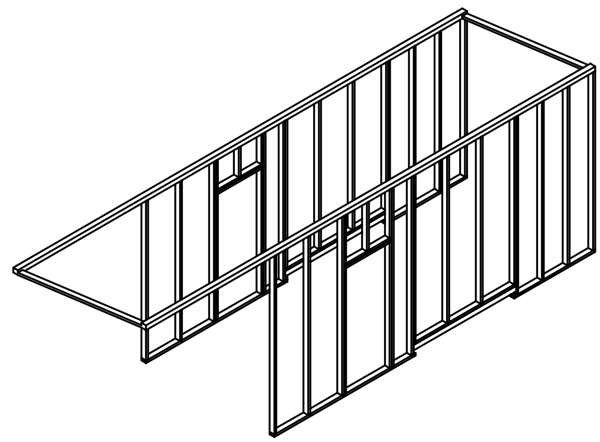
DIMENSIONS ARE IN INCHES AND DEGREES UNLESS OTHERWISE NOTED TOLERANCES: LINEAR ±0.005 ANGULAR: ±0.005		NAME	DATE	**PROPRIETARY:** THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF M. STRATTON IN CONJUNCTION WITH GVSU.
	DRAWN	M. STRATTON	07/23/20	
DO NOT SCALE DRAWING	COMMENTS: USED FLAT REPRESENTATION FOR CORRUGATED METAL ROOF PANELS.			SIZE A — DWG. NO. Roof_Metal_20ft — REV.
				SCALE:1:75 — WEIGHT: — SHEET 1 OF 1

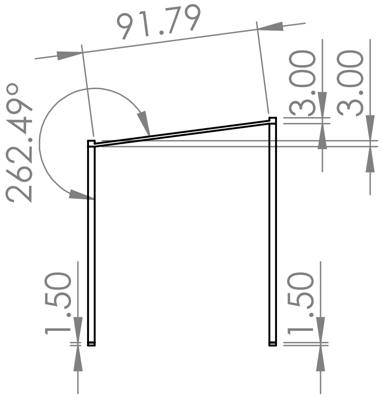


DIMENSIONS ARE IN INCHES AND DEGREES UNLESS OTHERWISE NOTED TOLERANCES: LINEAR ±0.005 ANGULAR: ±0.005		NAME	DATE	**PROPRIETARY:** THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF M. STRATTON IN CONJUNCTION WITH GVSU.
	DRAWN	M. STRATTON	07/24/20	
DO NOT SCALE DRAWING	COMMENTS: PLYWOOD WILL NEED TO BE CUT TO FIT, POSSIBLY IN MULTIPLE PIECES FOR ONE SECTION. PLYWOOD IS 3/8IN THICK.			SIZE **A** DWG. NO. Roof_Sheathing_20ft REV.
				SCALE:1:75 WEIGHT: SHEET 1 OF 1

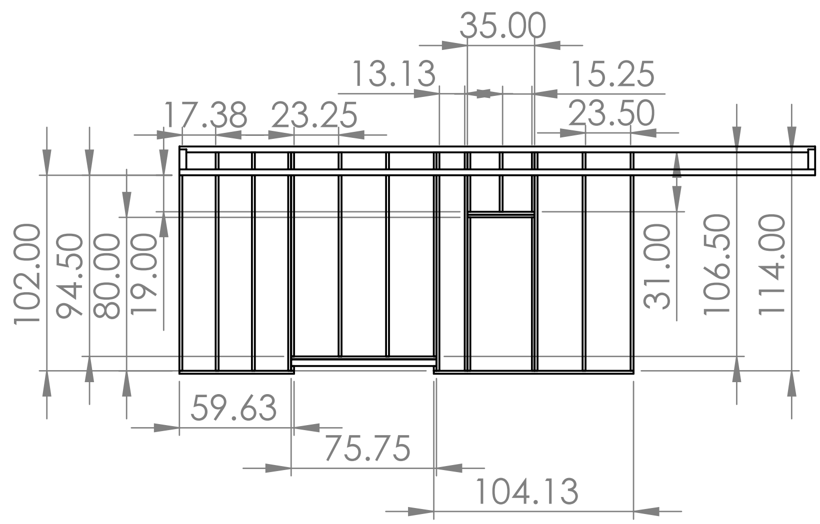


TOP



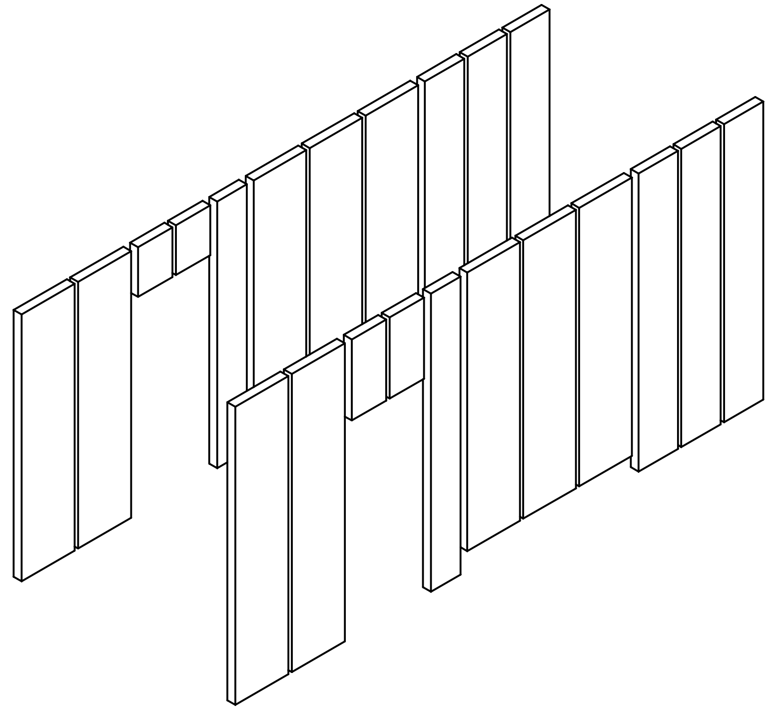


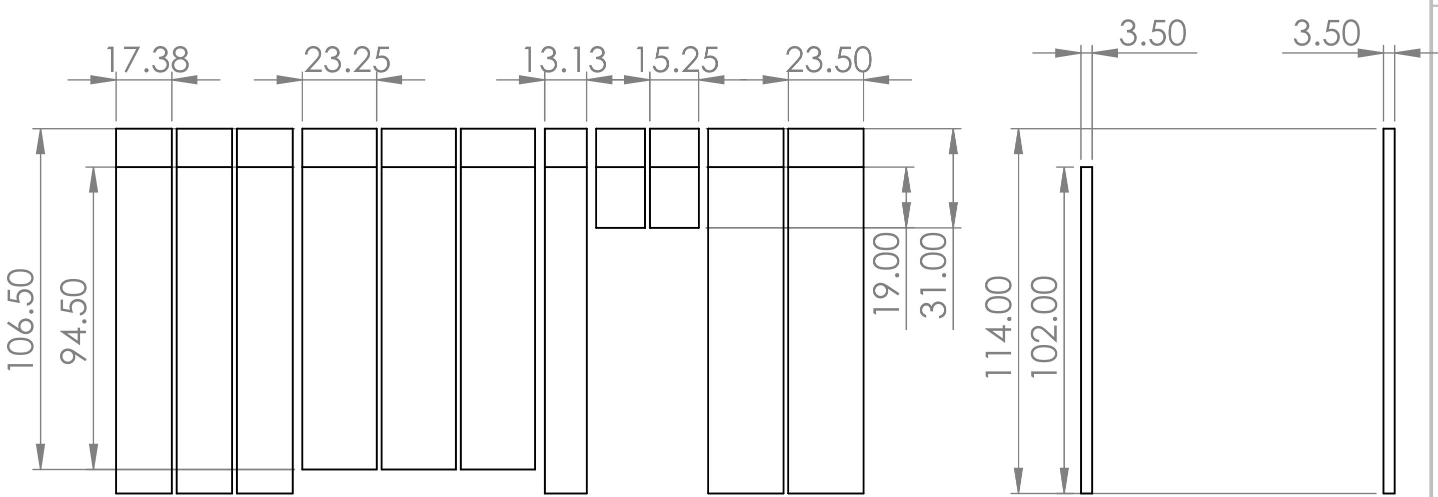
FRONT



LEFT

DIMENSIONS ARE IN INCHES AND DEGREES UNLESS OTHERWISE NOTED TOLERANCES: LINEAR ±0.005 ANGULAR: ±0.005	DRAWN — NAME: M. STRATTON — DATE: 07/27/20 COMMENTS: ALL STUDS ARE 2X4s. TOP HORIZONTAL CONNECTION IS TWO (2) 2X4s STACKED ON TOP OF EACH OTHER.	**PROPRIETARY:** THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF M. STRATTON IN CONJUNCTION WITH GVSU.
DO NOT SCALE DRAWING		SIZE A — DWG. NO. Side_Walls_20ft — REV. SCALE:1:100 — WEIGHT: — SHEET 1 OF 1





LEFT

FRONT

DIMENSIONS ARE IN INCHES AND DEGREES UNLESS OTHERWISE NOTED
TOLERANCES:
LINEAR ±0.005
ANGULAR: ±0.005

DO NOT SCALE DRAWING

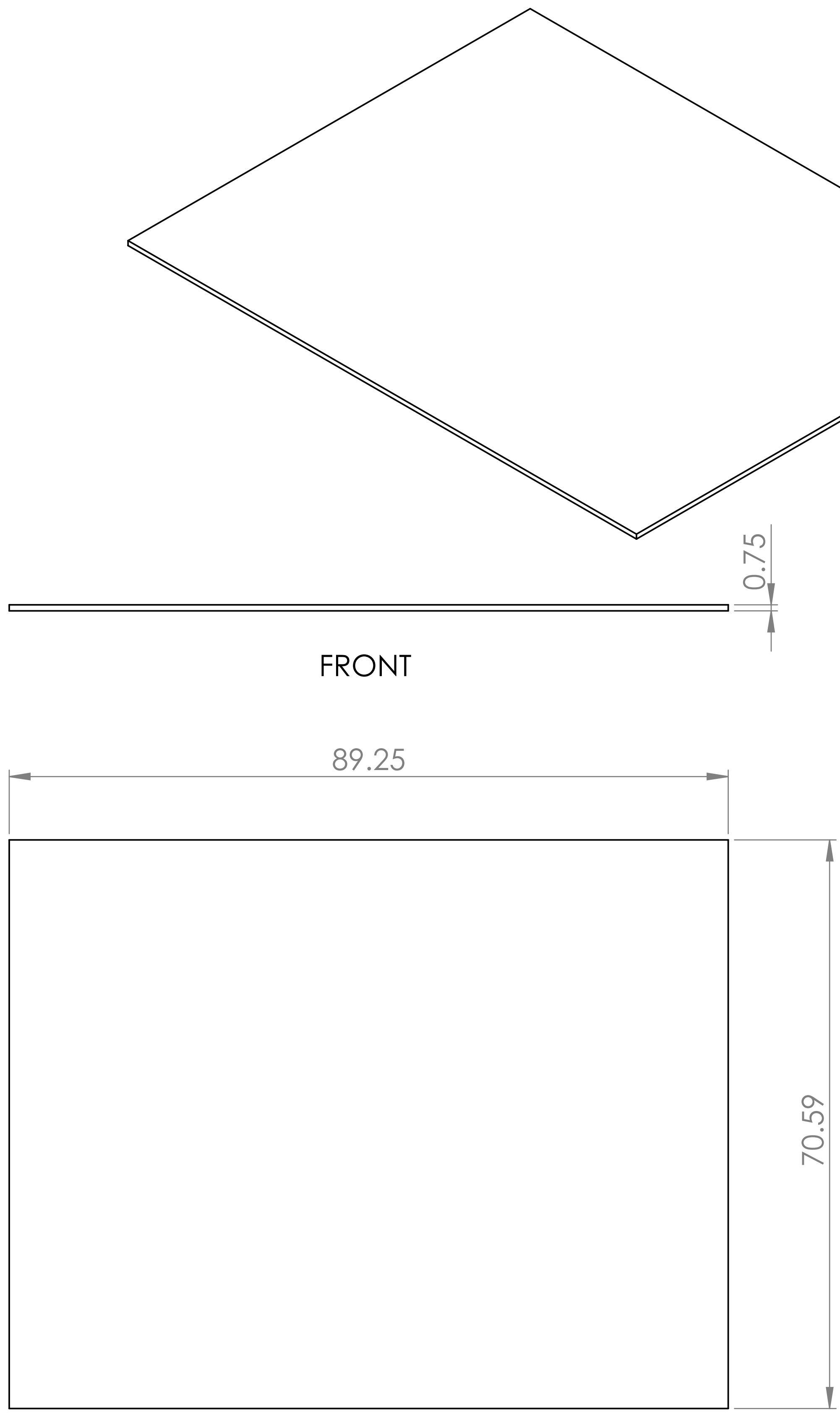
	NAME	DATE
DRAWN	M. STRATTON	10/04/20

COMMENTS: INSULATION WILL NEED TO BE CUT TO FIT, POSSIBLY IN MULTIPLE PIECES FOR ONE SECTIONS.

INSULATION TO FIT 2X4 JOISTS.

SIZE A | DWG. NO. Side_Wall_Insul_20ft | REV.

SCALE:1:60 | WEIGHT: | SHEET 1 OF 1



DIMENSIONS ARE IN INCHES
AND DEGREES UNLESS
OTHERWISE NOTED
TOLERANCES:
LINEAR ±0.005
ANGULAR: ±0.005

DO NOT SCALE DRAWING

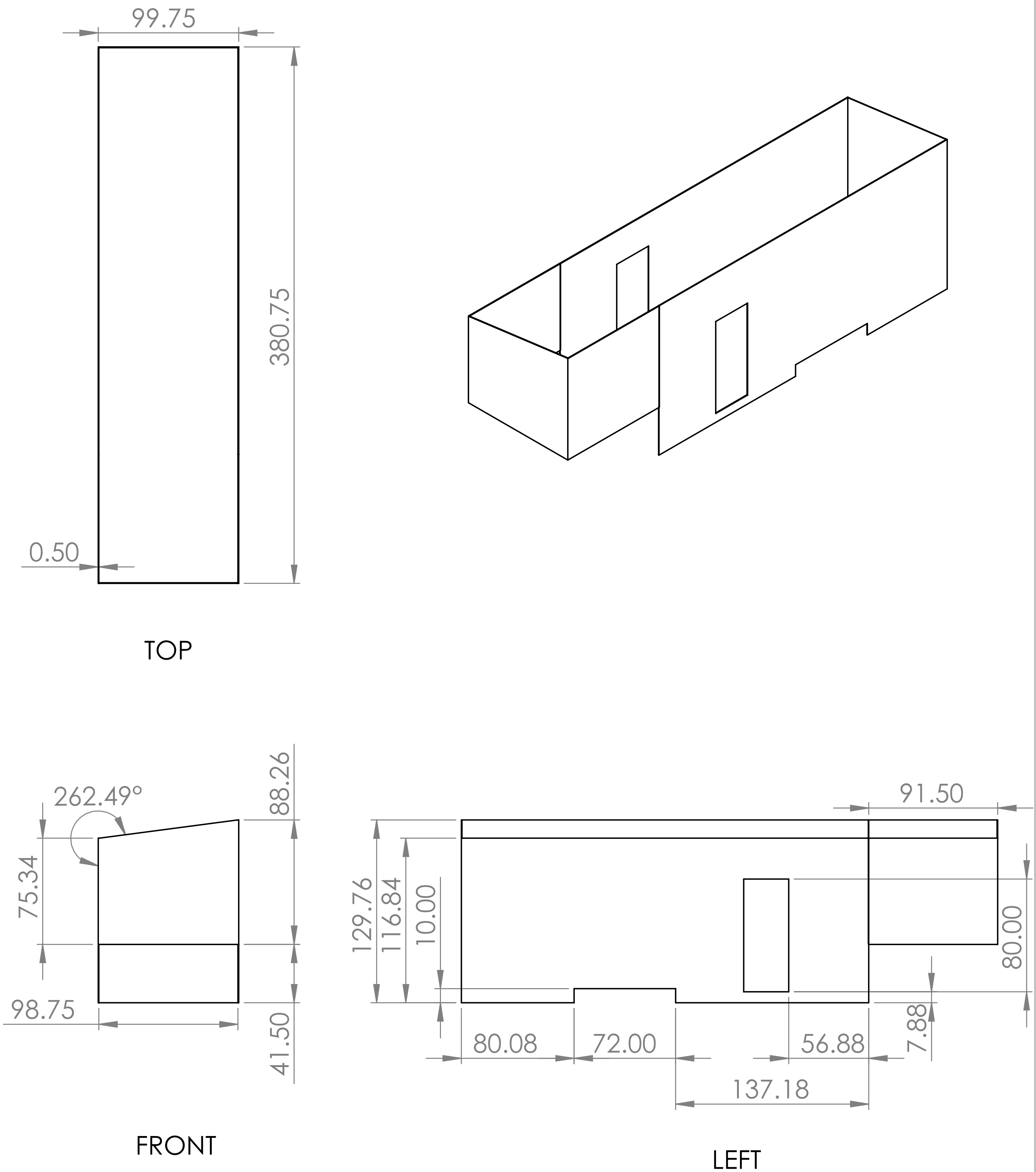
	NAME	DATE
DRAWN	M. STRATTON	07/28/20

COMMENTS: 3/4IN THICK TILE WAS USED. MULTIPLE PIECES WILL NEED TO BE CUT TO FIT SHOWN DIMENSIONS.

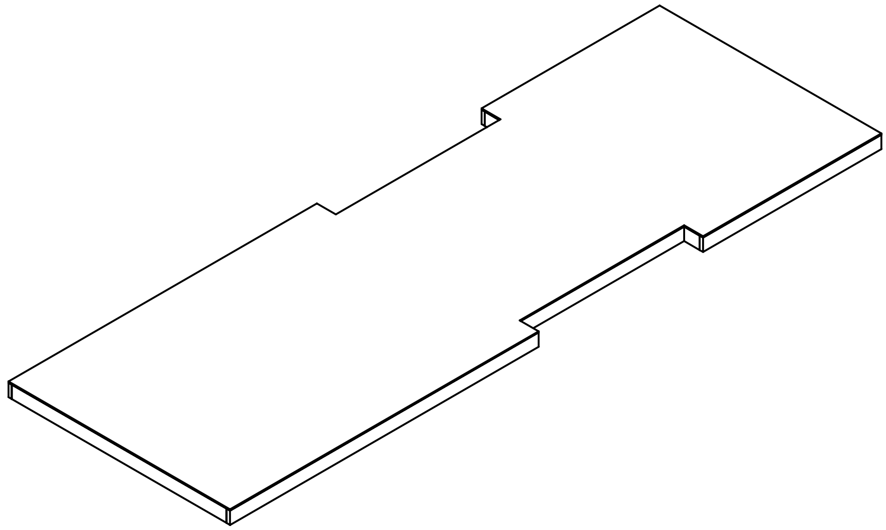
PROPRIETARY: THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF M. STRATTON IN CONJUNCTION WITH GVSU.

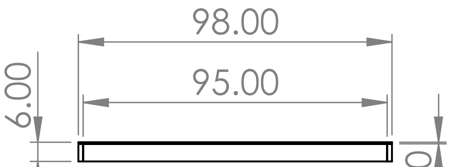
SIZE	DWG. NO.	REV.
A	Bathroom_Tile_24ft	

SCALE:1:20 WEIGHT: SHEET 1 OF 1

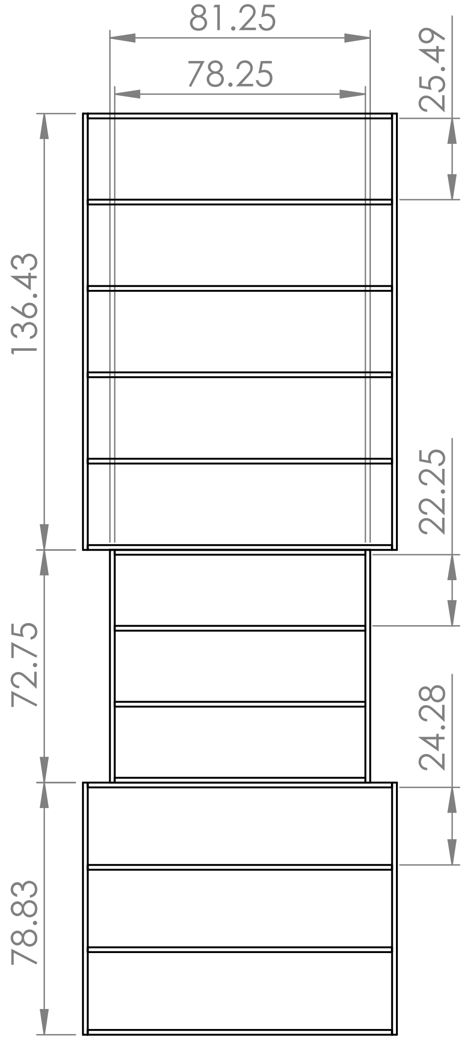


DIMENSIONS ARE IN INCHES AND DEGREES UNLESS OTHERWISE NOTED TOLERANCES: LINEAR ±0.005 ANGULAR: ±0.005		NAME	DATE	**PROPRIETARY:** THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF M. STRATTON IN CONJUNCTION WITH GVSU.
	DRAWN	M. STRATTON	07/28/20	
	COMMENTS: SIDING WILL NEED TO BE CUT TO FIT, POSSIBLY IN MULTIPLE PIECES FOR ONE SECTION. PLACEHOLDER THICKNESS IS 1/2IN.			SIZE **A** DWG. NO. Exterior_Siding_24ft REV.
DO NOT SCALE DRAWING				SCALE:1:100 WEIGHT: SHEET 1 OF 1





FRONT



BOTTOM

DIMENSIONS ARE IN INCHES AND DEGREES UNLESS OTHERWISE NOTED
TOLERANCES:
LINEAR ±0.005
ANGULAR: ±0.005

DO NOT SCALE DRAWING

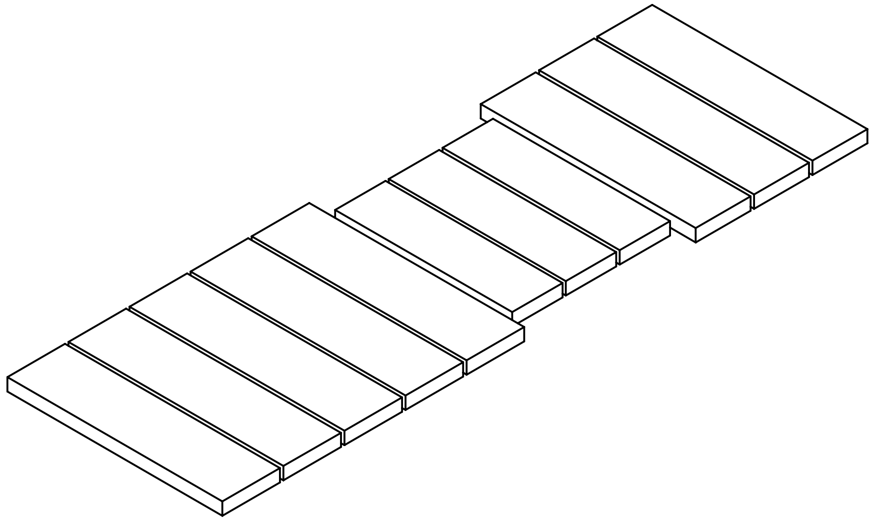
	NAME	DATE
DRAWN	M. STRATTON	07/28/20

COMMENTS: ALL FLOOR JOISTS ARE 2X6s. PLYWOOD WILL NEED TO BE CUT TO FIT, POSSIBLY IN MULTIPLE PIECES FOR ONE SECTION.

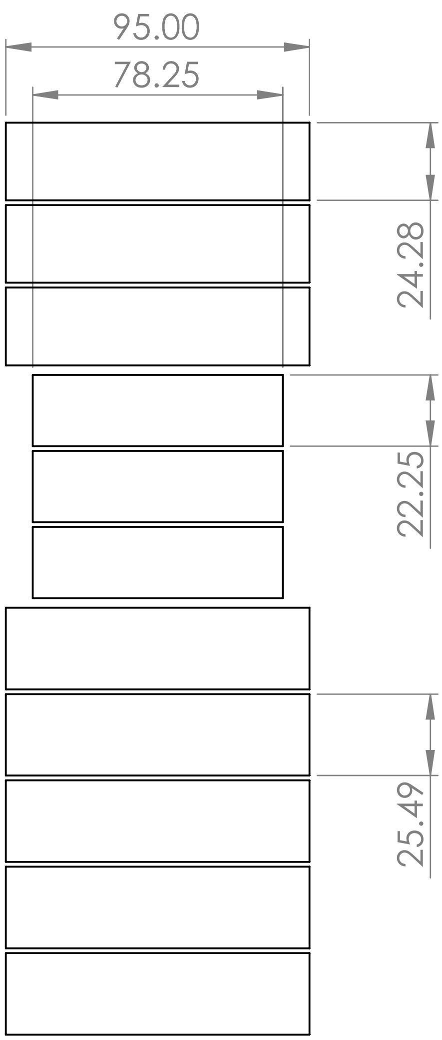
PLYWOOD IS 1/2IN THICK.

SIZE **A** DWG. NO. Floor_24ft REV.

SCALE:1:60 WEIGHT: SHEET 1 OF 1



FRONT



TOP

DIMENSIONS ARE IN INCHES AND DEGREES UNLESS OTHERWISE NOTED
TOLERANCES:
LINEAR ±0.005
ANGULAR: ±0.005

DO NOT SCALE DRAWING

	NAME	DATE
DRAWN	M. STRATTON	07/28/20

COMMENTS: INSULATION WILL NEED TO BE CUT TO FIT, POSSIBLY IN MULTIPLE PIECES FOR ONE SECTION.

INSULATION TO FIT 2X6 FLOOR JOISTS

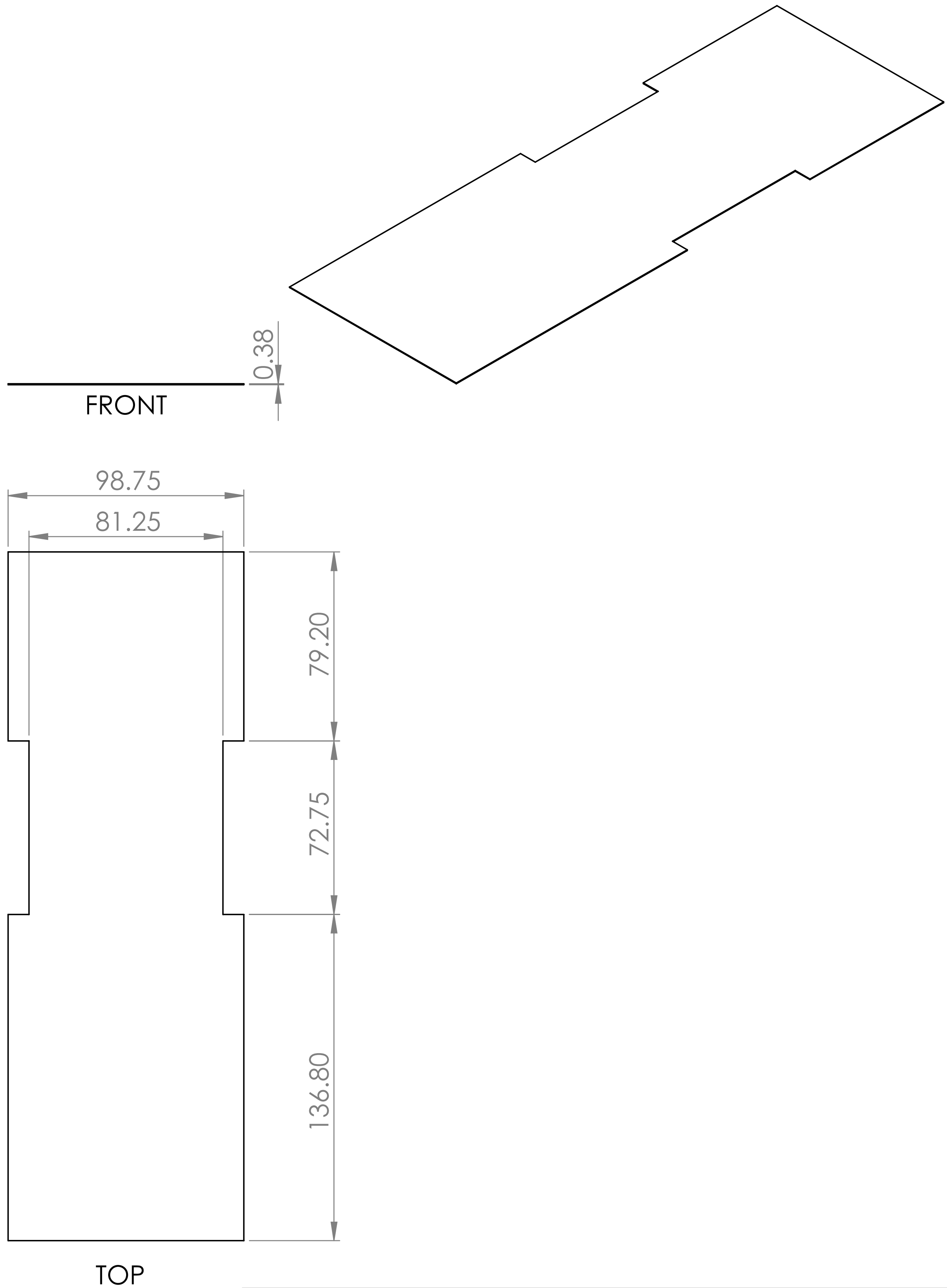
PROPRIETARY: THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF M. STRATTON IN CONJUNCTION WITH GVSU.

SIZE	DWG. NO.	REV.
A	Floor_Insul_24ft	

SCALE:1:60 | WEIGHT: | SHEET 1 OF 1

2 1

B B

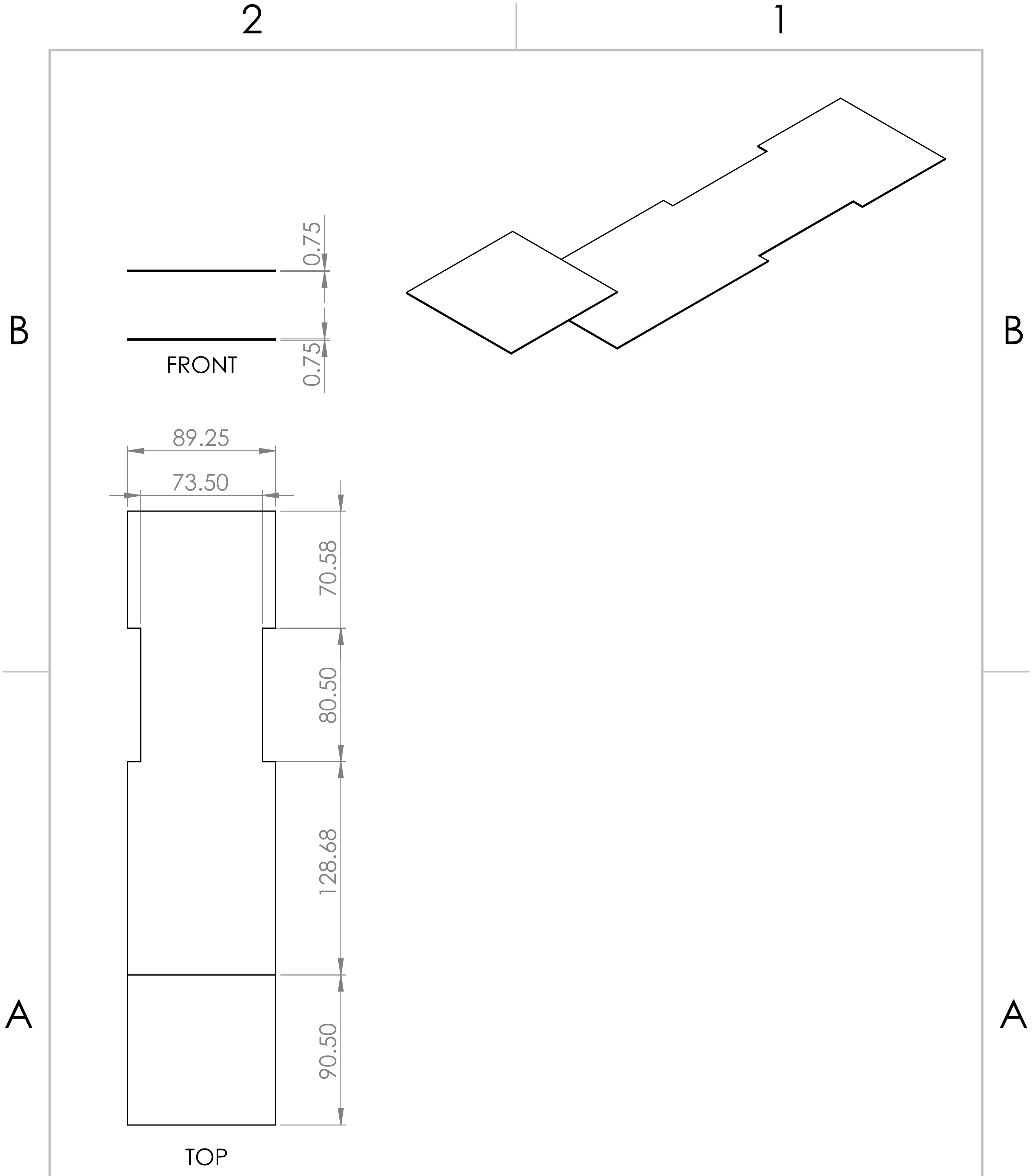


A A

DIMENSIONS ARE IN INCHES AND DEGREES UNLESS OTHERWISE NOTED TOLERANCES: LINEAR ±0.005 ANGULAR: ±0.005		
	NAME	DATE
DRAWN	M. STRATTON	07/28/20
DO NOT SCALE DRAWING	COMMENTS: PLYWOOD WILL NEED TO BE CUT TO FIT, POSSIBLY IN MULTIPLE PIECES FOR ONE SECTION. PLYWOOD IS 3/8IN THICK.	

SIZE	DWG. NO.	REV.
A	Floor_Sheathing_24ft	
SCALE:1:60	WEIGHT:	SHEET 1 OF 1

2 1

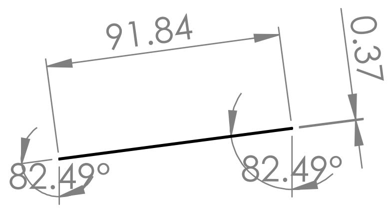


DIMENSIONS ARE IN INCHES AND DEGREES UNLESS OTHERWISE NOTED TOLERANCES: LINEAR ±0.005 ANGULAR: ±0.005		NAME	DATE
	DRAWN	M. STRATTON	07/28/20
DO NOT SCALE DRAWING	COMMENTS: FLOOR WILL NEED TO BE CUT TO FIT, POSSIBLY IN MULTIPLE PIECES FOR ONE SECTION. FLOOR IS 3/4IN THICK.		

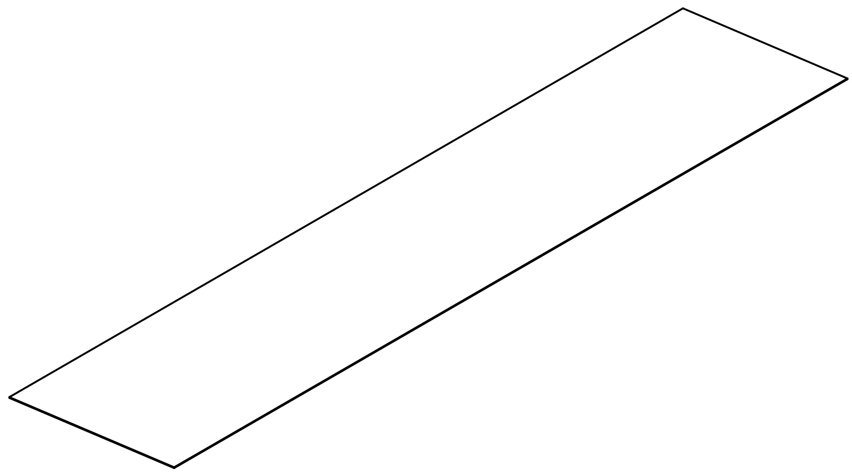
PROPRIETARY: THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF M. STRATTON IN CONJUNCTION WITH GVSU.

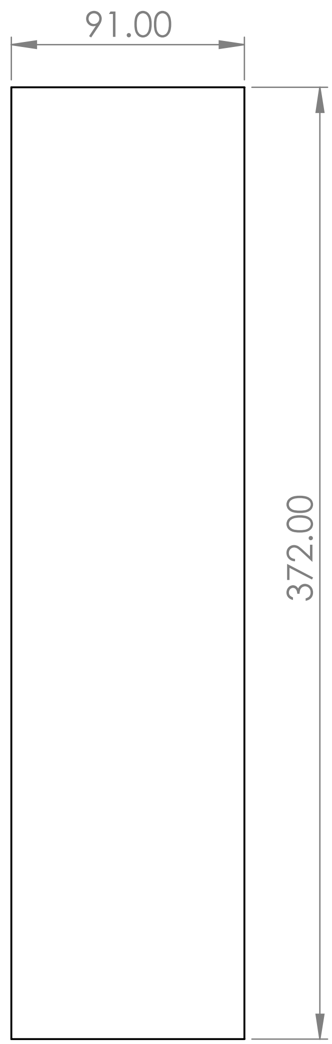
SIZE	DWG. NO.	REV.
A	Floor_Top_24ft	

SCALE:1:75	WEIGHT:	SHEET 1 OF 1



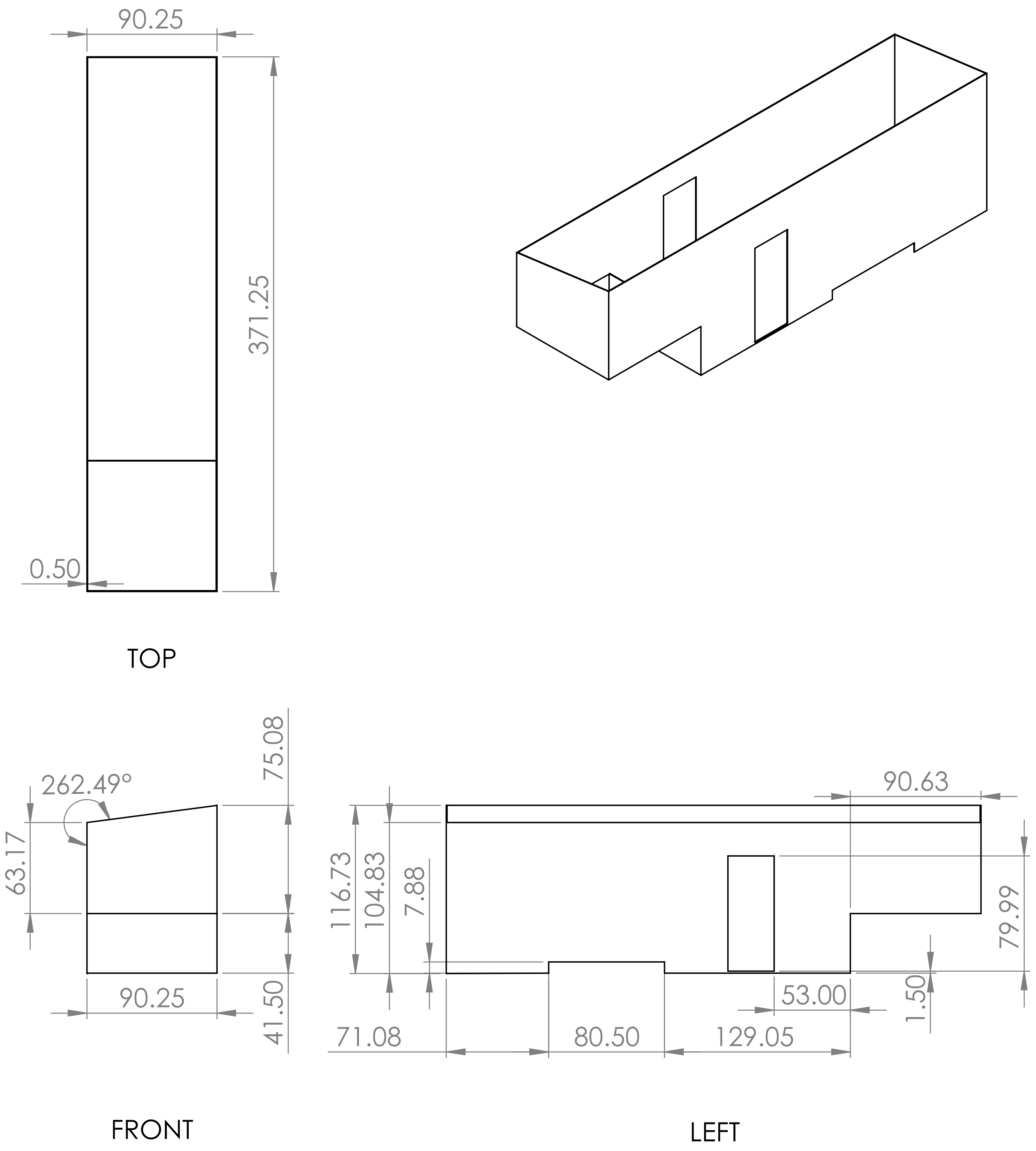
FRONT





TOP

DIMENSIONS ARE IN INCHES AND DEGREES UNLESS OTHERWISE NOTED TOLERANCES: LINEAR ±0.005 ANGULAR: ±0.005		NAME	DATE	**PROPRIETARY:**	THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF M. STRATTON IN CONJUNCTION WITH GVSU.
	DRAWN	M. STRATTON	07/28/20		
	COMMENTS: PLYWOOD WILL NEED TO BE CUT TO FIT, POSSIBLY IN MULTIPLE PIECES FOR ONE SECTION. PLYWOOD IS 3/8IN THICK.			SIZE **A**	DWG. NO. nt_Roof_Sheathing_24ft — REV.
DO NOT SCALE DRAWING				SCALE:1:75	WEIGHT: — SHEET 1 OF 1



DIMENSIONS ARE IN INCHES AND DEGREES UNLESS OTHERWISE NOTED
TOLERANCES:
LINEAR ±0.005
ANGULAR: ±0.005

DO NOT SCALE DRAWING

	NAME	DATE
DRAWN	M. STRATTON	09/23/20

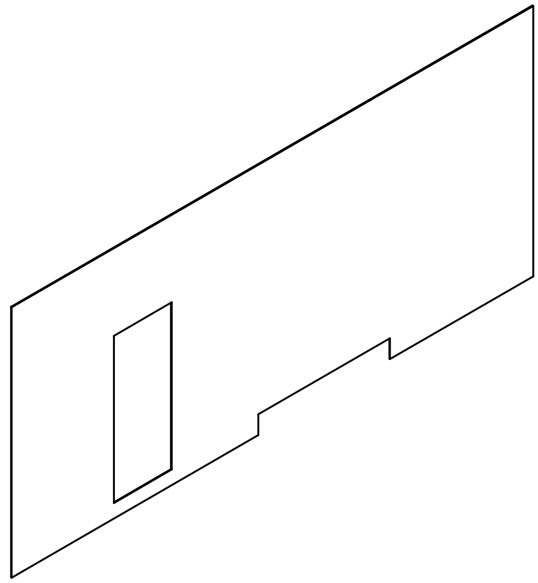
COMMENTS: SHIPLAP WILL NEED TO BE CUT TO FIT, POSSIBLY IN MULTIPLE PIECES FOR ONE SECTION.

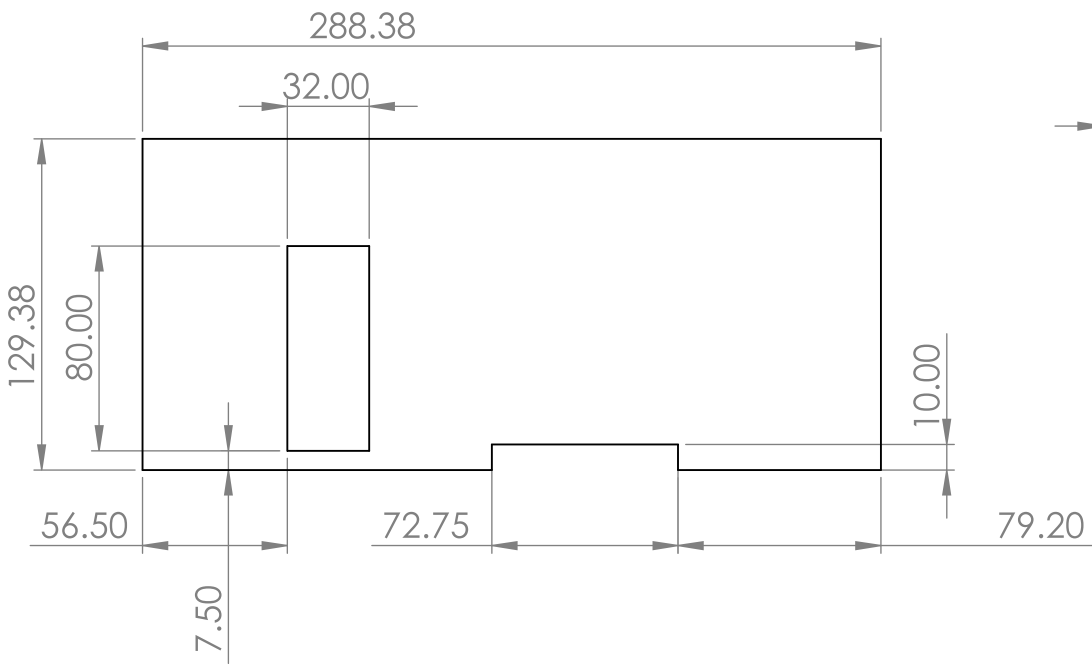
PLACEHOLDER THICKNESS IS 1/2IN.

PROPRIETARY: THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF M. STRATTON IN CONJUNCTION WITH GVSU.

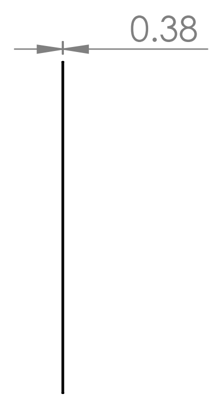
SIZE **A** DWG. NO. Interior_Shiplap_24ft REV.

SCALE:1:100 WEIGHT: SHEET 1 OF 1



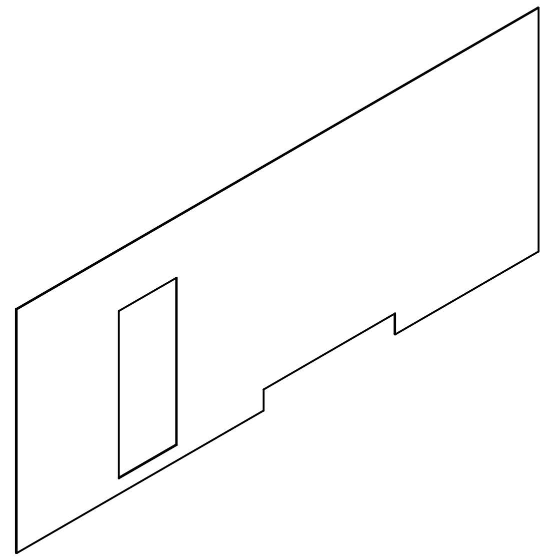


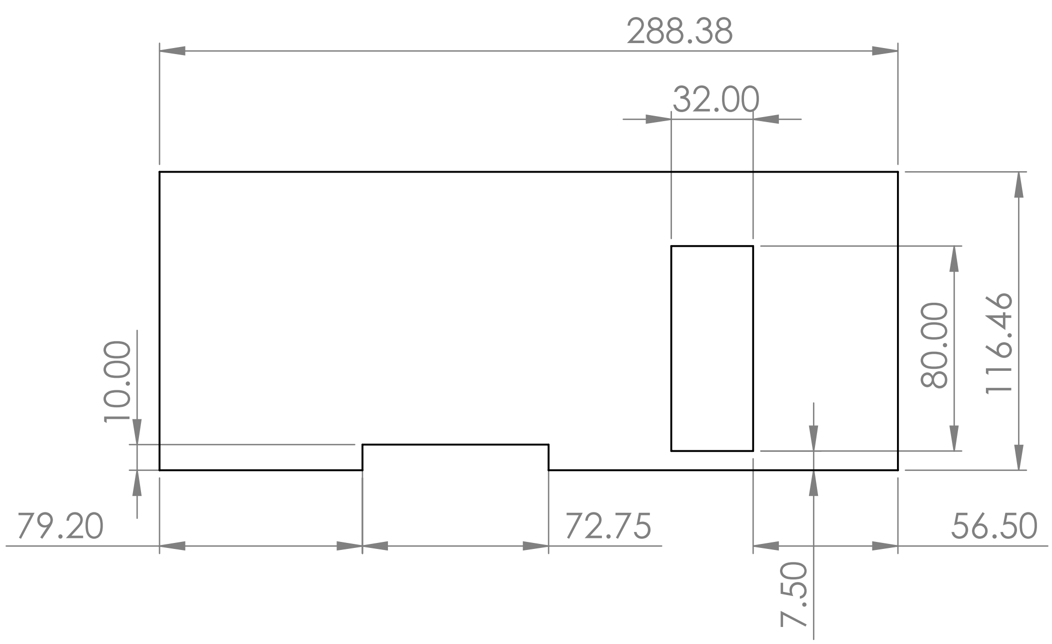
RIGHT



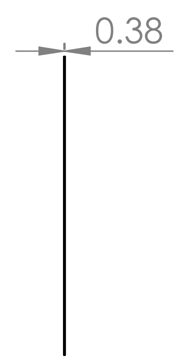
FRONT

DIMENSIONS ARE IN INCHES AND DEGREES UNLESS OTHERWISE NOTED TOLERANCES: LINEAR ±0.005 ANGULAR: ±0.005	NAME	DATE	**PROPRIETARY:** THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF M. STRATTON IN CONJUNCTION WITH GVSU.
	DRAWN M. STRATTON	07/28/20	
	COMMENTS: PLYWOOD WILL NEED TO BE CUT TO FIT, POSSIBLY IN MULTIPLE PIECES FOR ONE SECTION. ALL PLYWOOD IS 3/8IN THICK.		SIZE **A** DWG. NO. L_Side_Wall_Sheathing_24ft REV.
DO NOT SCALE DRAWING			SCALE:1:75 WEIGHT: SHEET 1 OF 1





LEFT



FRONT

DIMENSIONS ARE IN INCHES
AND DEGREES UNLESS
OTHERWISE NOTED
TOLERANCES:
LINEAR ±0.005
ANGULAR: ±0.005

DO NOT SCALE DRAWING

	NAME	DATE
DRAWN	M. STRATTON	07/28/20

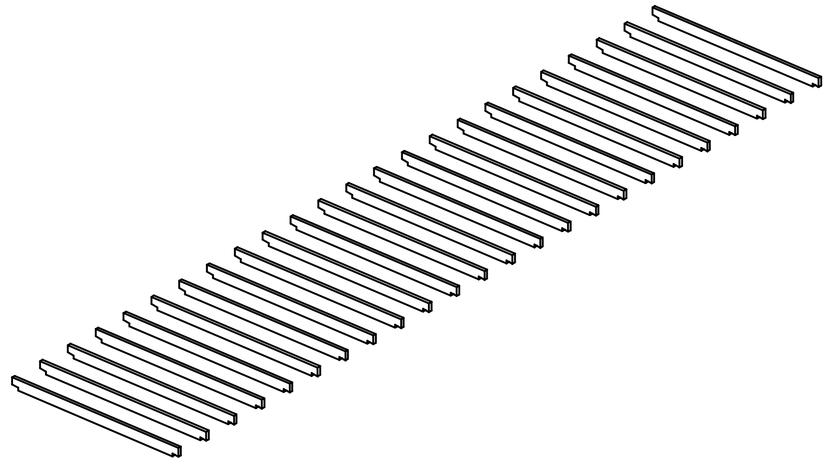
COMMENTS: PLYWOOD WILL NEED TO BE CUT TO FIT, POSSIBLY IN MULTIPLE PIECES FOR ONE SECTION.

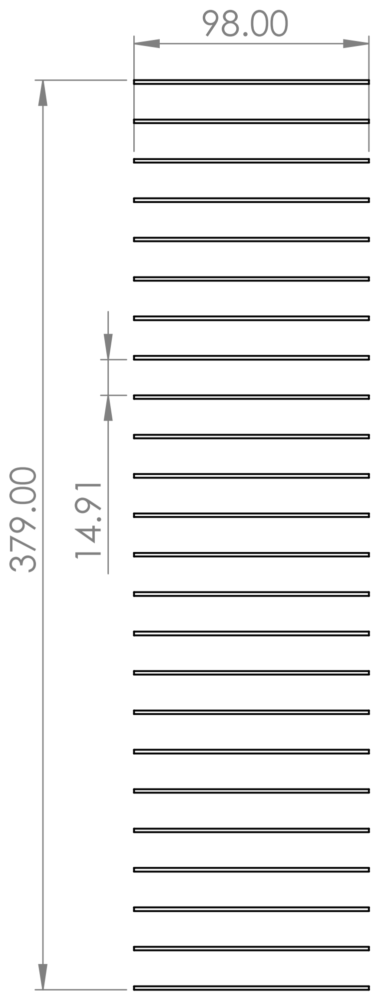
ALL PLYWOOD IS 3/8IN THICK.

PROPRIETARY: THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF M. STRATTON IN CONJUNCTION WITH GVSU.

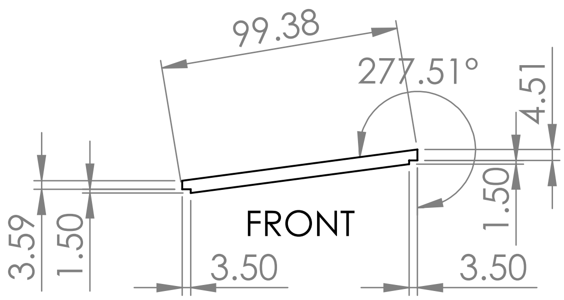
SIZE **A** DWG. NO. R_Side_Wall_Sheathing_24ft REV.

SCALE:1:75 WEIGHT: SHEET 1 OF 1





TOP



FRONT

DIMENSIONS ARE IN INCHES AND DEGREES UNLESS OTHERWISE NOTED
TOLERANCES:
LINEAR ±0.005
ANGULAR: ±0.005

DO NOT SCALE DRAWING

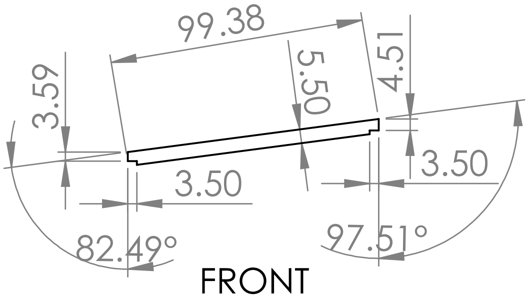
	NAME	DATE
DRAWN	M. STRATTON	07/28/20

COMMENTS: ALL ROOF JOISTS ARE 2X6s, 24 IN TOTAL.

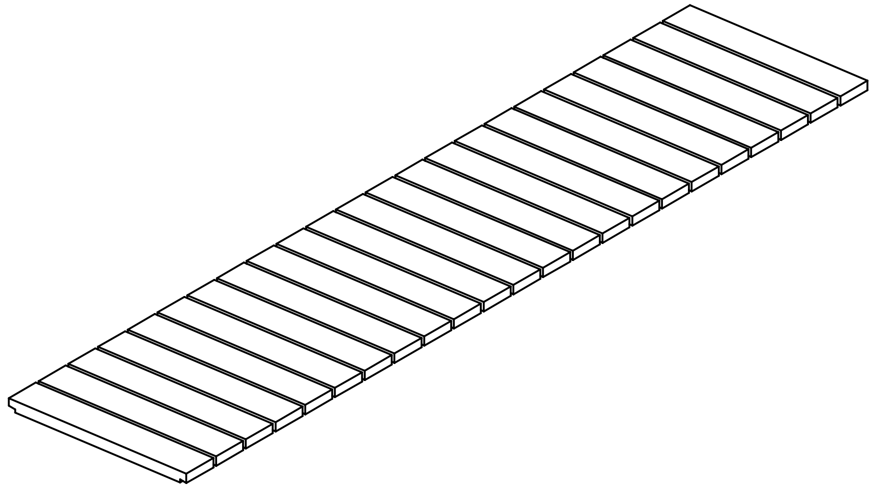
PROPRIETARY: THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF M. STRATTON IN CONJUNCTION WITH GVSU.

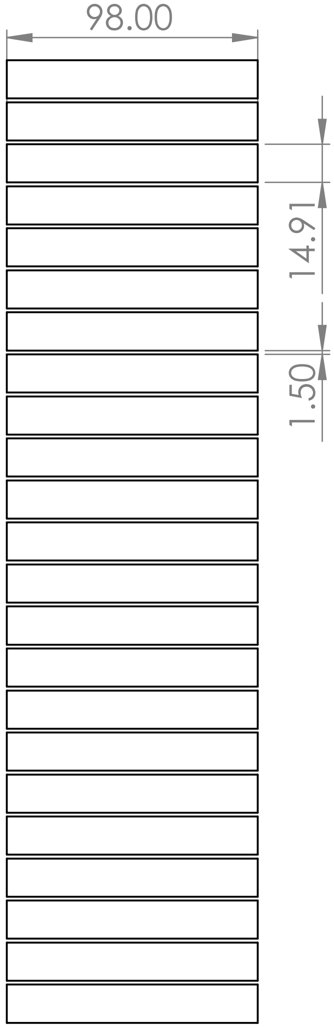
SIZE A | DWG. NO. Roof_24ft | REV.

SCALE:1:80 | WEIGHT: | SHEET 1 OF 1



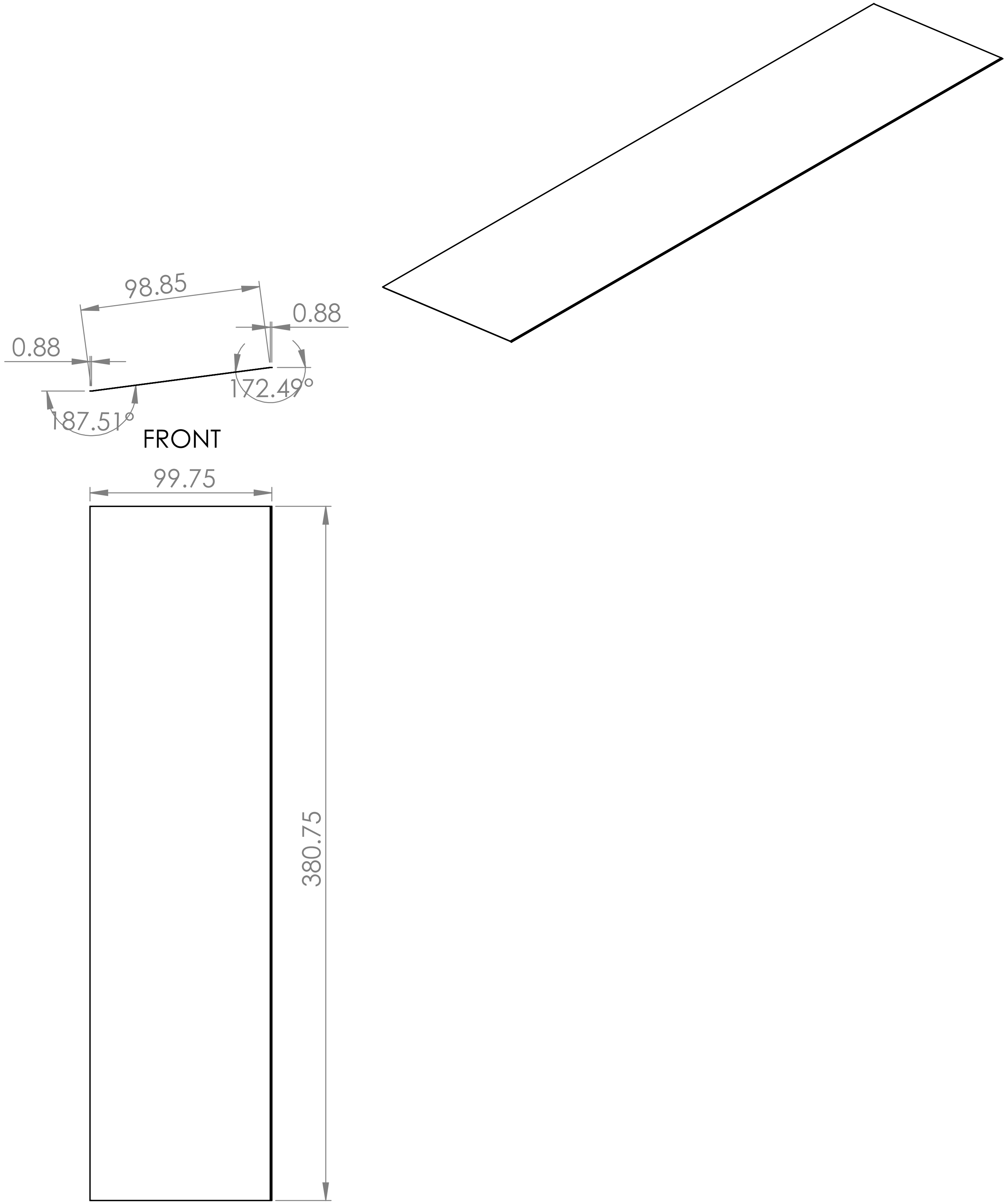
FRONT





TOP

DIMENSIONS ARE IN INCHES AND DEGREES UNLESS OTHERWISE NOTED TOLERANCES: LINEAR ±0.005 ANGULAR: ±0.005		NAME	DATE	**PROPRIETARY:** THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF M. STRATTON IN CONJUNCTION WITH GVSU.
	DRAWN	M. STRATTON	07/28/20	
	COMMENTS: INSULATION WILL NEED TO BE CUT TO FIT, POSSIBLY IN MULTIPLE PIECES FOR ONE SECTION. 23 SECTIONS IN TOTAL.			SIZE **A** DWG. NO. Roof_Insul_24ft REV.
DO NOT SCALE DRAWING				SCALE:1:75 WEIGHT: SHEET 1 OF 1



DIMENSIONS ARE IN INCHES
AND DEGREES UNLESS
OTHERWISE NOTED
TOLERANCES:
LINEAR ±0.005
ANGULAR: ±0.005

DO NOT SCALE DRAWING

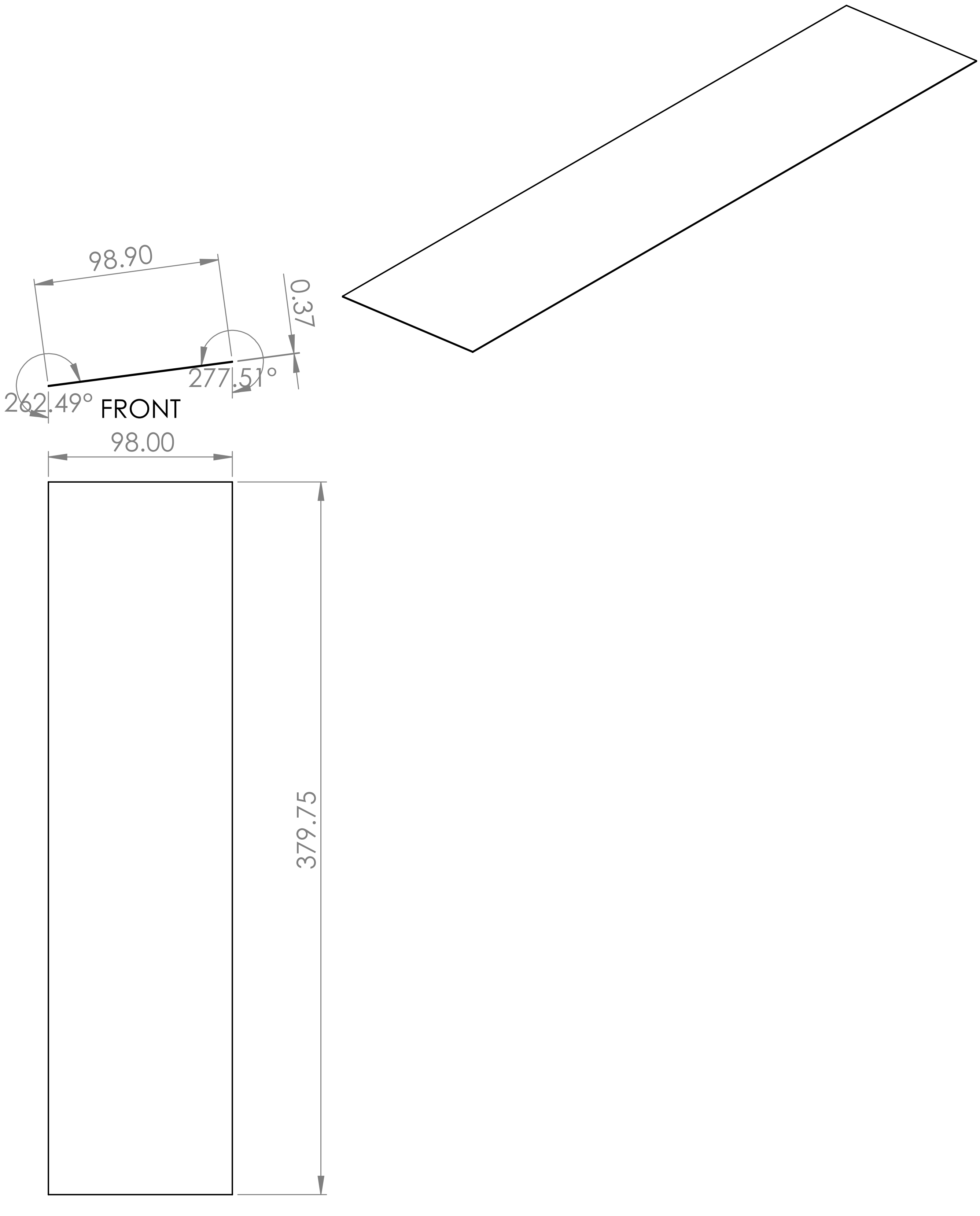
	NAME	DATE
DRAWN	M. STRATTON	07/28/20

COMMENTS: USED FLAT REPRESENTATION FOR CORRUGATED METAL ROOF PANELS.

PROPRIETARY: THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF M. STRATTON IN CONJUNCTION WITH GVSU.

SIZE A | DWG. NO. Roof_Metal_24ft | REV.

SCALE:1:75 | WEIGHT: | SHEET 1 OF 1

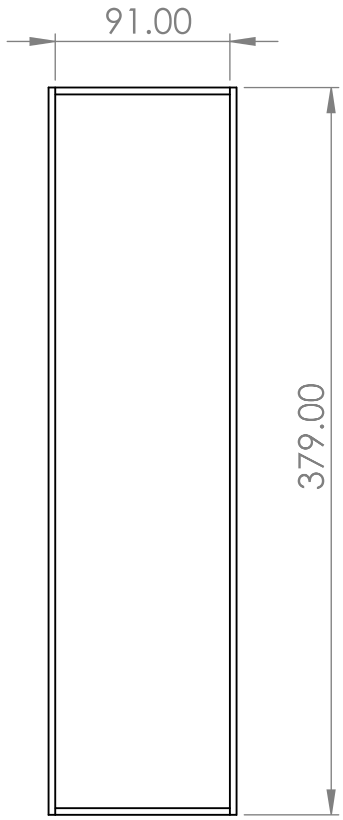


TOP

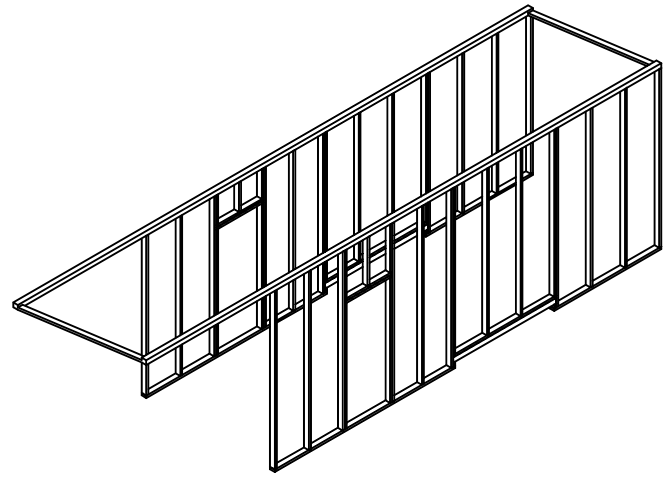
DIMENSIONS ARE IN INCHES AND DEGREES UNLESS OTHERWISE NOTED TOLERANCES: LINEAR ±0.005 ANGULAR: ±0.005		NAME	DATE
	DRAWN	M. STRATTON	07/28/20
	COMMENTS: PLYWOOD WILL NEED TO BE CUT TO FIT, POSSIBLY IN MULTIPLE PIECES FOR ONE SECTION. PLYWOOD IS 3/8IN THICK.		
DO NOT SCALE DRAWING			

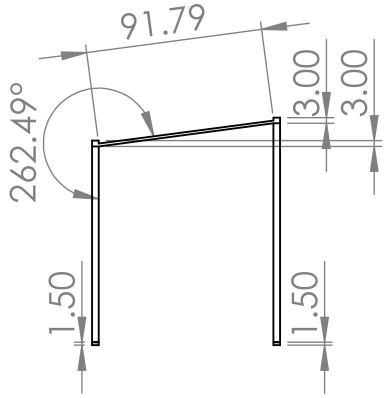
PROPRIETARY: THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF M. STRATTON IN CONJUNCTION WITH GVSU.

SIZE	DWG. NO.	REV.
A	Roof_Sheathing_24ft	
SCALE:1:75	WEIGHT:	SHEET 1 OF 1

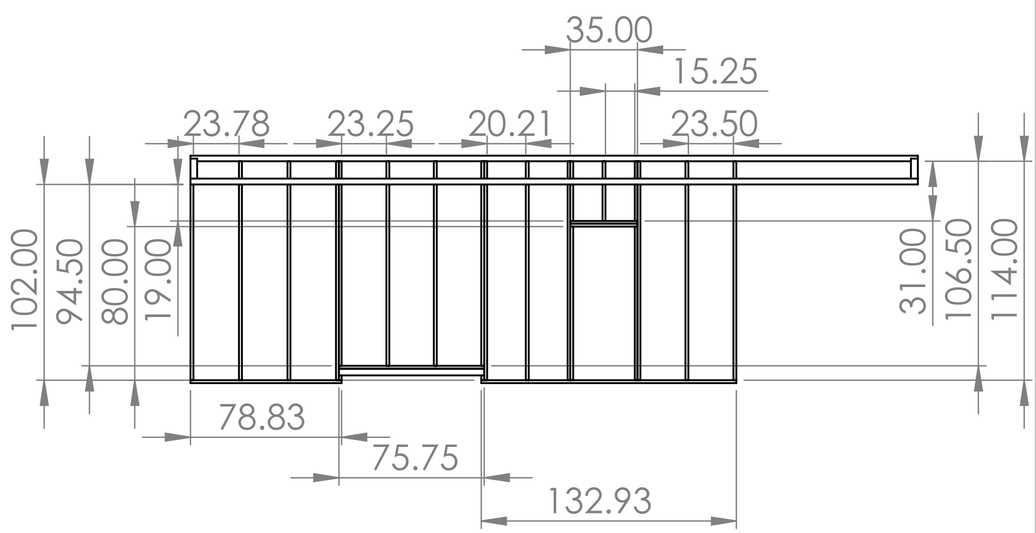


TOP





FRONT



LEFT

DIMENSIONS ARE IN INCHES AND DEGREES UNLESS OTHERWISE NOTED
TOLERANCES:
LINEAR ±0.005
ANGULAR: ±0.005

DO NOT SCALE DRAWING

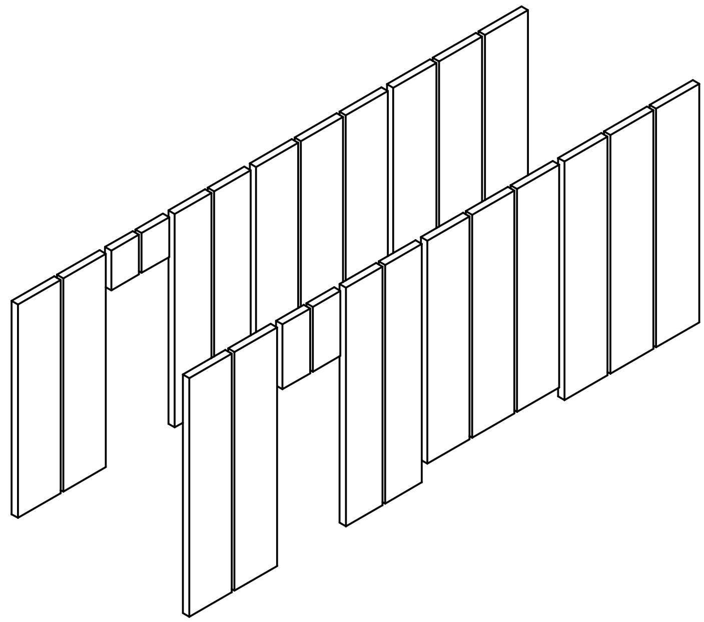
	NAME	DATE
DRAWN	M. STRATTON	07/28/20

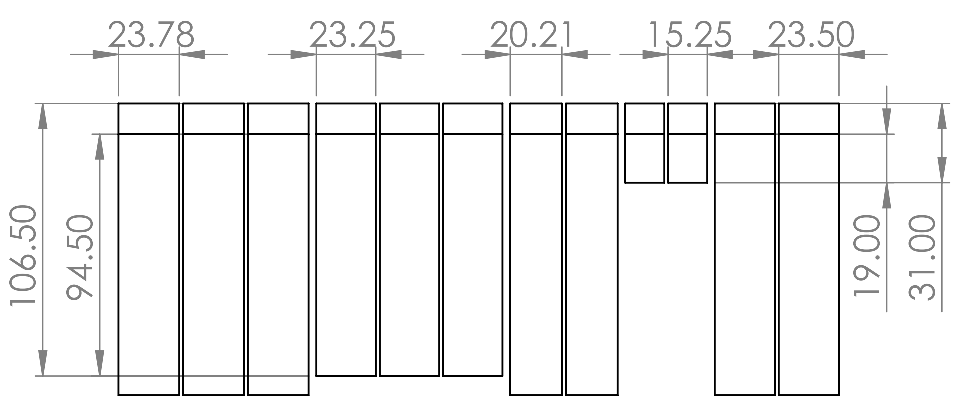
COMMENTS: ALL STUDS ARE 2X4s. TOP HORIZONTAL CONNECTION IS TWO (2) 2X4s STACKED ON TOP OF EACH OTHER.

PROPRIETARY: THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF M. STRATTON IN CONJUNCTION WITH GVSU.

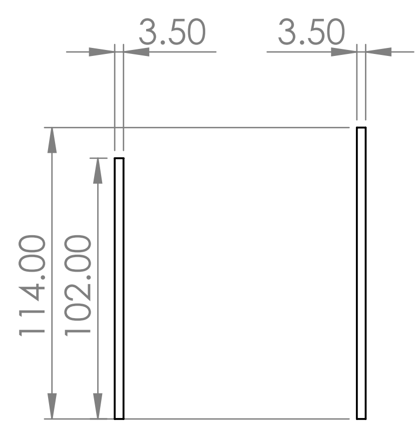
SIZE **A** DWG. NO. Side_Walls_24ft REV.

SCALE:1:100 WEIGHT: SHEET 1 OF 1





LEFT



FRONT

DIMENSIONS ARE IN INCHES AND DEGREES UNLESS OTHERWISE NOTED
TOLERANCES:
LINEAR ±0.005
ANGULAR: ±0.005

DO NOT SCALE DRAWING

	NAME	DATE
DRAWN	M. STRATTON	07/28/20

COMMENTS: INSULATION WILL NEED TO BE CUT TO FIT, POSSIBLY IN MULTIPLE PIECES FOR ONE SECTIONS.

INSULATION TO FIT 2X4 JOISTS.

SIZE A | DWG. NO. Side_Wall_Insul_24ft | REV.

SCALE:1:75 | WEIGHT: | SHEET 1 OF 1

References

[1] "Cottage Shed: Prefabricated Cottage: Tiny Cabin Shed," *Jamaica Cottage Shop*, 01-Aug-2020. [Online]. Available: https://jamaicacottageshop.com/shop/8x16-cross-gable-tiny-house/. [Accessed: 22-Sep-2020].

[2] "Willamette Farmhouse," *Tiny Smart House*. [Online]. Available: https://www.tinysmarthouse.com/models/willamette-farmhouse-tiny-house/. [Accessed: 07-Oct-2020].

[3] "Tiny House Big Living," *Backcountry Containers*. [Online]. Available: https://backcountrycontainers.com/tiny-house-big-living/. [Accessed: 07-Oct-2020].

[4] "Signature Models on Foundation," *B&B Micro Manufacturing, Inc.*, 10-Sep-2020. [Online]. Available: https://bbtinyhouses.com/stick-built-homes/. [Accessed: 07-Oct-2020].

[5] K. A. Calluari and F. Alonso-Marroquín, "Structural analysis of an off-grid tiny house," Jun. 2017.

[6] M. Kostoff, "Think Tiny; An Alternative House Design for an Aging Population," *University of Colorado Boulder*, 2016.

[7] A. Eberle, "Life Cycle Assessment of a Tiny House," *University of Washington*, 2015.

[8] N. Gromicko and B. Gromicko, "Structural Design Basics of Residential Construction for the Home Inspector," *InterNACHI*®. [Online]. Available: https://www.nachi.org/structural-design-basics-residential-construction.htm. [Accessed: 10-Oct-2020].

[9] S. Bady-Holmes, "A Step-by-Step Guide to the Home Building Process," *NewHomeSource*. [Online]. Available: https://www.newhomesource.com/guide/articles/a-step-by-step-guide-to-the-home-building-process. [Accessed: 10-Oct-2020].

[10] C. Lunsford and G. Lunsford, "Engineering a TOURING Tiny House #1," *YouTube*, 08-Aug-2015. [Online]. Available: https://www.youtube.com/watch?v=XeYfIGaQjWU. [Accessed: 10-Oct-2020].

[11] C. Lunsford and G. Lunsford, "Engineering a TOURING Tiny House #2," *YouTube*, 13-Aug-2015. [Online]. Available: https://www.youtube.com/watch?v=i69zfczp_5Q. [Accessed: 10-Oct-2020].

[12] C. Lunsford and G. Lunsford, "Engineering a TOURING Tiny House #3," *YouTube*, 20-Aug-2015. [Online]. Available: https://www.youtube.com/watch?v=XcqtaubE4ZY. [Accessed: 10-Oct-2020].

[13] C. Lunsford and G. Lunsford, “Engineering a TOURING Tiny House #4,” *YouTube*, 27-Aug-2015. [Online]. Available: https://www.youtube.com/watch?v=6q2wln-nTew. [Accessed: 10-Oct-2020].

[14] J. Bartlett, “Big City, Tiny House,” *Carleton University*, 2016.

[15] A. Mutter, “Growing Tiny Houses; Motivations and Opportunities for Expansion Through Niche Markets,” *MESPOM Programme*, 2013.

[16] E. Gentili, “Exploring Wellbeing in Small and Unconventional Dwellings; Understanding living in small and unconventional dwellings through a multidimensional perspective of space,” *Linnæus University*, 2017.

[17] Y. Gal, “A Very Small House: Designing for Good Living,” *Virginia Polytechnic Institute and State University*, 2008.

[18] F. Schneider, “Tiny Houses; What potential implications for the future does the Tiny House Movement show as an emerging urban lifestyle in the Netherlands?,” *Radboud University Nijmegen*, 2017.

[19] M. Specht, “The Current War: Why Did Westinghouse (AC) Beat Edison (DC)?,” *Union of Concerned Scientists*, 31-Oct-2019. [Online]. Available: https://blog.ucsusa.org/mark-specht/the-current-war-why-did-westinghouse-ac-beat-edison-dc. [Accessed: 17-Oct-2020].

[20] “How Much Is The Average Electric Bill: APG&E,” *How Much Is The Average Electric Bill | APG&E*. [Online]. Available: https://www.apge.com/average-electric-bill. [Accessed: 25-Sep-2020].

[21] “Household Water Usage,” *Utility Service Center*. [Online]. Available: https://rwd3rogers.com/documents/331/Household_Water_Usage_Estimates.pdf. [Accessed: 25-Sep-2020].

[22] “Stress and Strain,” *NDT Resource Center*. [Online]. Available: https://www.nde-ed.org/EducationResources/CommunityCollege/Materials/Mechanical/StressStrain.htm#:~:text=Engineering%20strain%20is%20defined%20as,initial%20length%20of%20the%20material.&text=Elastic%20deformation%20only%20occurs%20in,stress%20called%20the%20yield%20strength. [Accessed: 17-Oct-2020].

[23] “Stress–strain curve,” *Wikipedia*, 12-Oct-2020. [Online]. Available: https://en.wikipedia.org/wiki/Stress%E2%80%93strain_curve. [Accessed: 17-Oct-2020].

[24] “Off–axis loading of a lamina,” *University of Cambridge*. [Online]. Available: https://www.doitpoms.ac.uk/tlplib/fibre_composites/lamina_loading.php?printable=1. [Accessed: 17-Oct-2020].

[25] "Predicts Vehicle and Axle Weights When Towing Using an Extended Hitch," *Visualsc*. [Online]. Available: http://www.visualsc.com/hitch_calc.htm. [Accessed: 13-Oct-2020].

[26] H. F. Mollashahi, K. Khajavi, and A. K. Ghaeini, "Safety Evaluation and Adjustment of Superelevation Design Guides for Horizontal Curves Based on Reliability Analysis," *Journal of Transportation Engineering, Part A: Systems*, vol. 143, no. 6, p. 04017013, 2017.

[27] "Basic Refrigeraion Cycle," *RCA Technical Institute*. [Online]. Available: https://static1.squarespace.com/static/53f50cd6e4b0593b948bd441/t/5422ea77e4b0d80fe8b761ec/1411574391517/Course+1+-+Basic+Refrigeration+Cycle.pdf. [Accessed: 18-Oct-2020].

[28] S. S, "Difference Between Conduction, Convection and Radiation (with Comparison Chart)," *Key Differences*, 10-Feb-2018. [Online]. Available: https://keydifferences.com/difference-between-conduction-convection-and-radiation.html. [Accessed: 18-Oct-2020].

[29] "Latent and Sensible Heat," *North Carolina Climate Office*. [Online]. Available: https://climate.ncsu.edu/edu/Heat. [Accessed: 18-Oct-2020].

[30] "Scilab Help," *Scilab*. [Online]. Available: https://help.scilab.org/docs/5.5.2/en_US/index.html. [Accessed: 15-Oct-2020].

[31] "Introduction to Finite Element Analysis (FEA) or Finite Element Method (FEM)," *University of Victoria Engineering*. [Online]. Available: https://www.engr.uvic.ca/~mech410/lectures/FEA_Theory.pdf. [Accessed: 18-Oct-2020].

[32] "MapCustomizer.com: Plot multiple locations on Google Maps," *MapCustomizer*. [Online]. Available: https://www.mapcustomizer.com/map/Host%20Cities%202. [Accessed: 22-Sep-2020].

[33] "Gooseneck Tiny House Trailers," *Tiny House Basics*. [Online]. Available: https://www.tinyhousebasics.com/tinyhousetrailers/gooseneck-trailers/. [Accessed: 22-Sep-2020].

[34] "Body Measurements," *Centers for Disease Control and Prevention*, 20-May-2020. [Online]. Available: https://www.cdc.gov/nchs/fastats/body-measurements.htm. [Accessed: 04-Oct-2020].

[35] "2 Rules for Building Comfortable Stairs," *Fine Homebuilding*, 17-Jul-2020. [Online]. Available: https://www.finehomebuilding.com/2016/08/04/2-rules-comfortable-stairs. [Accessed: 04-Oct-2020].

[36] "Tiny House Trailer Fenders," *Tiny House Basics*. [Online]. Available: https://www.tinyhousebasics.com/fenders/. [Accessed: 22-Sep-2020].

[37] "Tiny House Blog Trailer Quote," *Tiny House Basics*. [Online]. Available: https://www.tinyhousebasics.com/tinyhouseblogquote/. [Accessed 22 September 2020].

[38] "Federal Size Regulations for Commercial Motor Vehicles," *U.S. Department of Transportation*. [Online]. Available: https://ops.fhwa.dot.gov/FREIGHT/publications/size_regs_final_rpt/index.htm. [Accessed: 20-Oct-2020].

[39] "Insulation," *Energy.gov*. [Online]. Available: https://www.energy.gov/energysaver/weatherize/insulation. [Accessed: 24-Sep-2020].

[40] "UltraTouch R-13 Denim Insulation Batts 23 in. x 93 in. (8-Bags)-10003-01323," *The Home Depot*. [Online]. Available: https://www.homedepot.com/p/UltraTouch-R-13-Denim-Insulation-Batts-23-in-x-93-in-8-Bags-10003-01323/204082521. [Accessed: 25-Sep-2020].

[41] "Handy Home Products Small Square Window," *The Home Depot*. [Online]. Available: https://www.homedepot.com/p/Handy-Home-Products-Small-Square-Window-18810-7/100592157. [Accessed: 04-Oct-2020].

[42] *Wood Handbook: Wood as an Engineering Material*. Madison, WI: USDA Forest Products Laboratory, 1999.

[43] "Green Lumber," *InterNACHI®*. [Online]. Available: https://www.nachi.org/green-lumber.htm. [Accessed: 23-Sep-2020].

[44] "Green Lumber vs. Dry Lumber," *Hanson Pole Buildings, LLC*. [Online]. Available: https://www.hansenpolebuildings.com/2011/09/499green-lumber-vs-dry-lumber/. [Accessed: 13-Oct-2020].

[45] *Residential structural design guide: a state-of-the-art engineering resource for light-frame homes, apartments, and townhouses*. Washington, District of Columbia: U.S. Department of Housing and Urban Development, Office of Policy Development and Research, 2017.

[46] *A policy on geometric design of highways and streets*, 2001. Washington, D.C.: American Association of State Highway and Transportation Officials, 2001.

[47] "Properly loading a trailer to maintain a proper tongue weight is paramount, especially when it comes to handling.," *Tongue Weight Is Key To Safe Towing — GMC Life*. [Online]. Available: https://www.gmc.com/gmc-life/trucks/why-tongue-weight-is-important-for-safe-towing. [Accessed: 22-Sep-2020].

[48] "2 in. x 4 in. x 96 in. Prime Whitewood Stud," *The Home Depot*. [Online]. Available: https://www.homedepot.com/p/2-in-x-4-in-x-96-in-Prime-Whitewood-Stud-058449/312528776. [Accessed: 07-Oct-2020].

[49] “2 in. x 6 in. x 8 ft. #2 BTR KD-HT SPF Dimensional Lumber,” *The Home Depot*. [Online]. Available: https://www.homedepot.com/p/2-in-x-6-in-x-8-ft-2-BTR-KD-HT-SPF-Dimensional-Lumber-058443/312528832. [Accessed: 07-Oct-2020].

[50] “2 in. x 8 in. x 8 ft. #2 and Better Prime Doug Fir Lumber,” *The Home Depot*. [Online]. Available: https://www.homedepot.com/p/2-in-x-8-in-x-8-ft-2-and-Better-Prime-Doug-Fir-Lumber-2x8-8-2-btr-prime-doug-fir/300177910. [Accessed: 07-Oct-2020].

[51] “2 in. x 12 in. x 8 ft. Premium #2 and Better Douglas Fir Lumber,” *The Home Depot*. [Online]. Available: https://www.homedepot.com/p/2-in-x-12-in-x-8-ft-Premium-2-and-Better-Douglas-Fir-Lumber-707195/202094201. [Accessed: 07-Oct-2020].

[52] “KOHLER Memoirs Ceramic Pedestal Combo Bathroom Sink with Classic Design in White with Overflow Drain,” *The Home Depot*. [Online]. Available: https://www.homedepot.com/p/KOHLER-Memoirs-Ceramic-Pedestal-Combo-Bathroom-Sink-with-Classic-Design-in-White-with-Overflow-Drain-K-2238-8-0/202515622. [Accessed: 07-Oct-2020].

[53] “Daltile QuicTile 12 in. x 24 in. Calacatta Marble Polished Porcelain Locking Floor Tile (9.6 sq. ft. / case),” *The Home Depot*. [Online]. Available: https://www.homedepot.com/p/Daltile-QuicTile-12-in-x-24-in-Calacatta-Marble-Polished-Porcelain-Locking-Floor-Tile-9-6-sq-ft-case-QC011224CLKHD1L/312597904. [Accessed: 07-Oct-2020].

[54] “R-13 Denim Insulation Batts 15 in. x 93 in. (12-Bags),” *The Home Depot*. [Online]. Available: https://www.homedepot.com/p/R-13-Denim-Insulation-Batts-15-in-x-93-in-12-Bags-10003-01315/204622768. [Accessed: 07-Oct-2020].

[55] “Glacier Bay All-in-One Dual Mount Stainless Steel 25 in. 2-Hole Single Bowl Tight Radius Kitchen Sink in Brushed Finish with Faucet,” *The Home Depot*. [Online]. Available: https://www.homedepot.com/p/Glacier-Bay-All-in-One-Dual-Mount-Stainless-Steel-25-in-2-Hole-Single-Bowl-Tight-Radius-Kitchen-Sink-in-Brushed-Finish-with-Faucet-VDR2522A1/302928165. [Accessed: 07-Oct-2020].

[56] “Gibraltar Building Products 8 ft. Corrugated Galvanized Steel Utility-Gauge Roof Panel,” *The Home Depot*. [Online]. Available: https://www.homedepot.com/p/Gibraltar-Building-Products-8-ft-Corrugated-Galvanized-Steel-Utility-Gauge-Roof-Panel-13513/202092961. [Accessed: 07-Oct-2020].

[57] "Low-Ambient 36,000 BTU 3 Ton 17 SEER Ductless Mini Split Wall Mounted Inverter Air Conditioner with Heat Pump 208/230V," *The Home Depot*. [Online]. Available: https://www.homedepot.com/p/Pioneer-Low-Ambient-36-000-BTU-3-Ton-17-SEER-Ductless-Mini-Split-Wall-Mounted-Inverter-Air-Conditioner-with-Heat-Pump-208-230V-WYT036GLFI19RL/314096014. [Accessed: 22-Nov-2020].

[58] "1/2 4 ft. x 8 ft. Oriented Strand Board," *The Home Depot*. [Online]. Available: https://www.homedepot.com/p/1-2-4-ft-x-8-ft-Oriented-Strand-Board-660663/100072564. [Accessed: 07-Oct-2020].

[59] "Best Price Mattress 10-Inch Memory Foam Mattress, Queen," *Amazon*. [Online]. Available: https://www.amazon.com/Best-Price-Mattress-10-Inch-Memory/dp/B00GTCL3SQ. [Accessed: 06-Oct-2020].

[60] "Magic Chef 10.1 cu. ft. Top Freezer Refrigerator in White," *The Home Depot*. [Online]. Available: https://www.homedepot.com/p/Magic-Chef-10-1-cu-ft-Top-Freezer-Refrigerator-in-White-HMDR1000WE/302245066. [Accessed: 07-Oct-2020].

[61] "1 in. x 6 in. x 8 ft. Premium Spruce Whitewood Shiplap Board," *The Home Depot*. [Online]. Available: https://www.homedepot.com/p/1-in-x-6-in-x-8-ft-Premium-Spruce-Whitewood-Shiplap-Board-270959/300452729. [Accessed: 07-Oct-2020].

[62] "24 in. 2.9 cu. ft. Gas Range in White," *The Home Depot*. [Online]. Available: https://www.homedepot.com/p/Summit-Appliance-24-in-2-9-cu-ft-Gas-Range-in-White-RG244WS/308495918. [Accessed: 07-Oct-2020].

[63] "2-piece 1.1 GPF/1.6 GPF High Efficiency Dual Flush Complete Elongated Toilet in White, Seat Included," *The Home Depot*. [Online]. Available: https://www.homedepot.com/p/Glacier-Bay-2-piece-1-1-GPF-1-6-GPF-High-Efficiency-Dual-Flush-Complete-Elongated-Toilet-in-White-Seat-Included-N2316/100676582. [Accessed: 07-Oct-2020].

[64] "Submerse 60 in. Right-Hand Drain Alcove Bathtub with Integral Flange in White," *The Home Depot*. [Online]. Available: https://www.homedepot.com/p/KOHLER-Submerse-60-in-Right-Hand-Drain-Alcove-Bathtub-with-Integral-Flange-in-White-K-R825-RA-0/205813027. [Accessed: 07-Oct-2020].

[65] "0.7 cu. ft. Countertop Microwave in White," *The Home Depot*. [Online]. Available: https://www.homedepot.com/p/RCA-0-7-cu-ft-Countertop-Microwave-in-White-RMW733-WHITE/205047629. [Accessed: 07-Oct-2020].

[66] "2.7 cu. ft. White 115-Volt All-in-One Washer Dryer Combo," *The Home Depot*. [Online]. Available: https://www.homedepot.com/p/Danby-2-7-cu-ft-White-115-Volt-All-in-One-Washer-Dryer-Combo-DWM120WDB-3/311701196. [Accessed: 07-Oct-2020].

[67] "Small Square Window," *The Home Depot*. [Online]. Available: https://www.homedepot.com/p/Handy-Home-Products-Small-Square-Window-18810-7/100592157. [Accessed: 07-Oct-2020].

[68] *ASHRAE Climatic Design Conditions 2009/2013/2017*. [Online]. Available: http://ashrae-meteo.info/v2.0/. [Accessed: 25-Sep-2020].

[69] "Gallons Used Per Person Per Day." [Online]. Available: https://www.phila.gov/water/educationoutreach/Documents/Homewateruse_IG5.pdf. [Accessed: 25-Sep-2020].

[70] "Heat Gain from Lights," *Engineering ToolBox*. [Online]. Available: https://www.engineeringtoolbox.com/heat-gain-lights-d_709.html. [Accessed: 25-Sep-2020].

[71] "How to Determine How Many LED Lumens You'll Need to Properly Light Your Space," *Language of Light — The Intersection of Lighting, Efficiency and Architecture*, 07-Jul-2020. [Online]. Available: https://www.alconlighting.com/blog/residential-led-lighting/how-do-i-determine-how-many-led-lumens-i-need-for-a-space/. [Accessed: 25-Sep-2020].

[72] C. E. Molina, "HEAT GAINS, HEATING AND COOLING IN NORDIC HOUSING," 25-Feb-2015. [Online]. Available: https://pdfs.semanticscholar.org/3b0b/407d4cbb6a116eb0f85aa2f404121716a5b6.pdf. [Accessed: 25-Sep-2020].

[73] *Refrigerator Normal Run Time*. [Online]. Available: https://products.geappliances.com/appliance/gea-support-search-content?contentId=16932. [Accessed: 25-Sep-2020].

[74] M. Gadala, "How Many Watts Does a Mini Split Use?," *Power Save AC*, 16-Dec-2019. [Online]. Available: https://powersaveac.com/blog/post/how-many-watts-does-a-mini-split-use. [Accessed: 25-Sep-2020].

[75] "Power Consumption of Typical Household Appliances," *Daft Logic*. [Online]. Available: https://www.daftlogic.com/information-appliance-power-consumption.htm. [Accessed: 25-Sep-2020].

[76] "DEA3500: Ambient Environment: Thermal Conditions," *Thermal Conditions Lecture*. [Online]. Available: http://ergo.human.cornell.edu/studentdownloads/DEA3500notes/Thermal/thcondnotes.html. [Accessed: 25-Sep-2020].

[77] C. Wilkins and M. H. Hosni, "Heat Gain From Office Equipment," *ASHRAE Journal*, Jun. 2000.

[78] *ASHRAE Fundamentals Handbook 2001*. Atlanta, GA: ASHRAE, 2001.

[79] "Douglas-Fir," *The Wood Database*. [Online]. Available: https://www.wood-database.com/douglas-fir/. [Accessed: 25-Sep-2020].

[80] T. Akgul, A. Apay, E. Aydin, and Y. Sumer, "Study of Bending Strength and Numerical Modeling of Wooden and Plywood Frame Elements," *Acta Physica Polonica A*, vol. 127, no. 4, 2015.

[81] *ASTM D3501-94, Standard Test Methods for Wood-Based Structural Panels in Compression*. West Conshohocken, PA: ASTM International, 2000.

[82] "Building materials weights guide," *Soundproofing Company*, 17-Jun-2020. [Online]. Available: https://www.soundproofingcompany.com/soundproofing_101/building-materials-weights-guide. [Accessed: 25-Sep-2020].

[83] "Insulation Values For Selected Materials," *R-Value Table*. [Online]. Available: http://www.coloradoenergy.org/procorner/stuff/r-values.htm. [Accessed: 25-Sep-2020].

[84] "Quick Reference: Snow Weight by Snow Type," *Weight of Snow by Snow Type*. [Online]. Available: https://roofonline.com/weight-of-snow. [Accessed: 30-Sep-2020].

[85] "Unbranded Pattern Stock Shiplap Board," *The Home Depot*. [Online]. Available: https://www.homedepot.com/p/Pattern-Stock-Shiplap-Board-Nominal-1-in-x-6-in-x-12-ft-Actual-0-625-in-x-5-37-in-x-144-in-905178/202180060. [Accessed: 01-Oct-2020].

[86] B. Yuen, "Bathtub," *The GrabCAD Community Library*. [Online]. Available: https://grabcad.com/library/bathtub—5. [Accessed: 04-Oct-2020].

[87] R. Mendoza, "Sink (Solidworks)," *The GrabCAD Community Library*. [Online]. Available: https://grabcad.com/library/sink-solidworks-1. [Accessed: 04-Oct-2020].

[88] "Tile Buying Guide," *Lowe's*. [Online]. Available: https://www.lowes.com/n/buying-guide/tile-flooring-buying-guide. [Accessed: 04-Oct-2020].

[89] "Kitchen Sink With Wooden Cabinet 3D Model," *The GrabCAD Community Library*. [Online]. Available: https://grabcad.com/library/kitchen-sink-with-wooden-cabinet-3d-model-1. [Accessed: 04-Oct-2020].

[90] "Solid Hardwood Flooring," *Hosking Hardwood Flooring*. [Online]. Available: https://www.hoskinghardwood.com/Department/Hardwood-Floors/All-About-Solid-Hardwood-Flooring.aspx?dId=7. [Accessed: 04-Oct-2020].

[91] "Best Price Mattress 10-Inch Memory Foam Mattress, Queen," *Amazon*. [Online]. Available: https://www.amazon.com/Best-Price-Mattress-10-Inch-Memory/dp/B00GTCL3SQ. [Accessed: 04-Oct-2020].

[92] Z. Fabijanic, “Fridge,” *The GrabCAD Community Library*. [Online]. Available: https://grabcad.com/library/fridge. [Accessed: 04-Oct-2020].

[93] Z. Fabijanic, “Oven 2,” *The GrabCAD Community Library*. [Online]. Available: https://grabcad.com/library/oven-2. [Accessed: 04-Oct-2020].

[94] Z. Fabijanic, “Kitchen table,” *The GrabCAD Community Library*. [Online]. Available: https://grabcad.com/library/kitchen-table—1. [Accessed: 04-Oct-2020].

[95] R. Mendoza, “Toilet (Solidworks),” *The GrabCAD Community Library*. [Online]. Available: https://grabcad.com/library/toilet-solidworks-1. [Accessed: 04-Oct-2020].

[96] R. Ramalho, “Washer and dryer,” *The GrabCAD Community Library*. [Online]. Available: https://grabcad.com/library/washer-and-dryer-1. [Accessed: 04-Oct-2020].

[97] B. Cushman-Roisin, “Sustainable Design,” *In Search of Indoor Comfort*, 04-Apr-2019. [Online]. Available: http://www.dartmouth.edu/~cushman/courses/engs44/comfort.pdf. [Accessed: 03-Oct-2020].

[98] “What is the Thickness of Glass Used in Most Residential Replacement Windows?,” *Energy Swing Windows*. [Online]. Available: https://www.energyswingwindows.com/about-us/news-and-events/27217-what-is-the-thickness-of-glass-used-in-most-residential-replacement-windows.html. [Accessed: 03-Oct-2020].

[99] “Small Square Window.” [Online]. Available: https://images.homedepot-static.com/catalog/pdfImages/47/4745d0ec-1158-46c8-8f12-07b694325bb1.pdf. [Accessed: 04-Oct-2020].

[100] A. Shitzer, “Wind-chill-equivalent temperatures: regarding the impact due to the variability of the environmental convective heat transfer coefficient,” *International Journal of Biometeorology*, vol. 50, no. 4, pp. 224–232, 2006.