Abstract

BARRETT III, GEORGE WARREN. Optimization of New Home Energy Performance. (Under the direction of Dr. Joe DeCarolis.)

The US residential sector is a key target for CO$_2$ reductions, yet there are few tools or analyses that help home buyers and homebuilders prioritize energy investments that reduce emissions. Furthermore, while models exist to simulate home energy performance, there are no publicly available tools that apply formal search techniques to help direct household investments that minimize cost or environmental impact. This thesis presents an optimization model specifically intended for home energy analysis. The model utilizes renewable energy time series data collected at the NC Solar House, a model home in North Carolina built for research and demonstration purposes. The availability of recorded sub-hourly data enables the development of an optimization model that accounts for the sub-hourly interaction between renewable electricity generation, displaced emissions from grid-purchased electricity, solar water heating, and household energy demand.

A mixed integer programming model is implemented to identify combinations of energy technologies that can minimize CO$_2$ emissions or cost. The model prescribes a set of investments that vary with the target carbon footprint and monthly household energy budget allowance. The resultant Pareto frontier traces the optimal trade-off between cost and emissions. Sensitivity analysis is performed by systematically varying household energy demand and indoor temperature. In addition, modeling to generate alternatives (MGA) is applied to test the robustness of the optimal solution by exploring the feasible, near-optimal decision space.
© Copyright 2011 by George Warren Barrett III

All Rights Reserved
Optimization of New Home Energy Performance

by
George Warren Barrett III

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Civil Engineering

Raleigh, North Carolina

2011

APPROVED BY:

__________________________  ____________________________
Dr. Ranji Ranjithan        Dr. Alex Hobbs

__________________________
Dr. Joe DeCarolis
Chair of Advisory Committee
Biography

After graduating from N.C. State University in 2007 with a B.S. degree in Mechanical Engineering, the author worked and traveled around New Zealand for a year. This experience fostered an awareness of environmental issues as well as efficiency standards within his home state of North Carolina. He worked assisting in biodiesel research with the North Carolina Solar Center before starting on the initial installation phase of the GRID$C$ equipment. Opportunities through the GRID$C$ project sent him looking for an advisor for a Master's degree, which led him to contact Dr. DeCarolis in the Department of Civil, Construction, and Environmental Engineering. With mutual interests in energy efficiency and greenhouse gas mitigation, the two began working together. The result is this thesis.
Acknowledgements

I would like to thank my advisor for his imparted knowledge, near endless patience, long hours, and encouraging words throughout the course of this project. He has allowed me to grow and helped me in times of difficulty and I am grateful. Thank you Joe.

I want to note that this project would not have been possible if it weren’t for all the members of the GRIDC project. I have enjoyed working with everyone on the project. Dr. Deluca, Pam, Nasim, and Erik, thank you for allowing me the opportunity to work on such an incredible idea.

Rachel Cook has stood beside me and held my hand throughout the entire process with an incredible amount of support and love. I hope I can do the same during the rest of your time at NC State. Thank you and I love you.

All of my friends in Raleigh and elsewhere allowed me to vent and helped me relax when necessary. I have enjoyed all our potlucks, trips climbing and elsewhere, and many nights on the porch. You are the best group of friends anyone could ever hope for. Thank you all for such a wonderful time.

My family has stood behind me during this entire project. They continued to show their support throughout my time in school and that has meant so much to me. Thank you for your love and support.
Table of Contents

List of Tables ........................................................................ vi

List of Figures ....................................................................... vii

Chapter 1 Introduction .......................................................... 1
  1.1 Motivation ........................................................................ 1
  1.2 Background ...................................................................... 4
    1.2.1 The North Carolina Solar Center and GRID C Project .... 4
    1.2.2 Data Collection ............................................................ 6

Chapter 2 Data Analysis ......................................................... 8
  2.1 Time Series Data Analysis ................................................ 8
    2.1.1 Importing Raw Data Files ............................................ 9
    2.1.2 Data Management ..................................................... 10
  2.2 Input Data ....................................................................... 13
    2.2.1 Emissions Data .......................................................... 13
    2.2.2 Hot Water Demand .................................................... 15
    2.2.3 Water Heating Technologies ....................................... 16
    2.2.4 Building Design ........................................................ 23
    2.2.5 Insulation ................................................................. 24
    2.2.6 Scalar Values ............................................................ 26

Chapter 3 Model Formulation ................................................. 27
  3.1 The Optimization Model .................................................. 27
  3.2 Objective Function .......................................................... 28
  3.3 Decision Variables ......................................................... 30
    3.3.1 Cost Calculations ...................................................... 30
    3.3.2 Insulation ................................................................. 31
    3.3.3 Renewable Energy Systems ......................................... 33
    3.3.4 Electricity Cost ........................................................ 34
    3.3.5 Natural Gas Cost ....................................................... 35
    3.3.6 Water Heating ........................................................ 36
    3.3.7 Heating and Cooling Technologies ............................... 37
    3.3.8 Renewable Energy and Energy Efficient Credits ........... 40
  3.4 Implementation of Model .................................................. 40
  3.5 Modeling to Generate Alternatives .................................... 41

Chapter 4 Analysis ................................................................ 44
  4.1 Scenarios ....................................................................... 44
    4.1.1 Base Case ................................................................. 44
    4.1.2 Sensitivity Analysis ................................................... 48
List of Tables

Table 2.1  Time Sample Discrepancies ........................................... 11
Table 2.2  Hot Water Consumption .................................................... 15
Table 2.3  Insulation Prices .............................................................. 24
Table 2.4  Window Prices ................................................................. 25
Table 3.1  Decision Variables ........................................................... 31
Table 3.2  Insulation Values .............................................................. 32
Table 3.3  Water Heating Options ....................................................... 37
Table 3.4  Space Heating and Cooling ................................................. 39
Table 4.1  72°F Results ................................................................. 45
Table 4.2  Sensitivity Results ............................................................ 49
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1.1</td>
<td>Building Energy Usage</td>
<td>2</td>
</tr>
<tr>
<td>Figure 1.2</td>
<td>Solar House</td>
<td>5</td>
</tr>
<tr>
<td>Figure 1.3</td>
<td>Research Annex</td>
<td>6</td>
</tr>
<tr>
<td>Figure 2.1</td>
<td>Solar Thermal Schematic</td>
<td>21</td>
</tr>
<tr>
<td>Figure 2.2</td>
<td>Modeled Home</td>
<td>23</td>
</tr>
<tr>
<td>Figure 3.1</td>
<td>Model Framework</td>
<td>29</td>
</tr>
<tr>
<td>Figure 4.1</td>
<td>72°F Pareto Frontier</td>
<td>47</td>
</tr>
<tr>
<td>Figure 4.2</td>
<td>82°F Pareto Frontier</td>
<td>50</td>
</tr>
<tr>
<td>Figure 4.3</td>
<td>62°F Pareto Frontier</td>
<td>52</td>
</tr>
<tr>
<td>Figure 4.4</td>
<td>72°F Low Demand MGA Results</td>
<td>55</td>
</tr>
<tr>
<td>Figure 4.5</td>
<td>72°F Medium and High Demand MGA Results</td>
<td>56</td>
</tr>
<tr>
<td>Figure 7.1</td>
<td>62°F Low Demand Raw Results</td>
<td>67</td>
</tr>
<tr>
<td>Figure 7.2</td>
<td>62°F Medium Demand Raw Results</td>
<td>68</td>
</tr>
<tr>
<td>Figure 7.3</td>
<td>62°F High Demand Raw Results</td>
<td>69</td>
</tr>
<tr>
<td>Figure 7.4</td>
<td>72°F Low Demand Raw Results</td>
<td>70</td>
</tr>
<tr>
<td>Figure 7.5</td>
<td>72°F Medium Demand Raw Results</td>
<td>71</td>
</tr>
<tr>
<td>Figure 7.6</td>
<td>72°F High Demand Raw Results</td>
<td>72</td>
</tr>
<tr>
<td>Figure 7.7</td>
<td>82°F Low Demand Raw Results</td>
<td>73</td>
</tr>
<tr>
<td>Figure 7.8</td>
<td>82°F Medium Demand Raw Results</td>
<td>74</td>
</tr>
<tr>
<td>Figure 7.9</td>
<td>82°F High Demand Raw Results</td>
<td>75</td>
</tr>
<tr>
<td>Figure 8.1</td>
<td>72°F Low MGA Results</td>
<td>77</td>
</tr>
<tr>
<td>Figure 8.2</td>
<td>Raw Results</td>
<td>78</td>
</tr>
<tr>
<td>Figure 8.3</td>
<td>Raw Results</td>
<td>79</td>
</tr>
</tbody>
</table>
1.1 Motivation

In recent years, anthropogenic climate change has emerged as a key global environmental challenge [17]. In 2007, the EPA issued a positive endangerment finding, indicating that CO₂ emissions are a form of air pollution that endangers public health [29]. Anthropogenic climate change is largely driven by CO₂ emissions from fossil fuel combustion. US energy consumption has risen steadily from 57.0 quads in 1980 to 102.3 quads in 2008 [9]. Of the 5,814 MMt of US CO₂ emissions in 2008, the residential building sector was responsible for about 1,220 MMt, or approximately 21% of total emissions [9]. One key way of reducing CO₂ emissions in the residential sector is through better design of homes to minimize energy consumption.

Energy conservation emerged as a key consideration in U.S. building design during the energy crises of the 1970s, which sent oil prices soaring by approximately a factor of nine in less than a decade [27]. As a byproduct of the energy crisis, building simulation programs were developed to assess energy performance. Simulation programs provide building engineers with a means for
calculating building energy performance. Below is a list of some well known energy simulation programs.

**Building Design Advisor** is limited in its database of building components and systems and cannot model complex 3-D geometry. These limitations are mitigated; however, by the program’s ability to simultaneously compare multiple designs based on different input parameters. A main advantage of Building Design Advisor is its user-friendliness [34].

**DOE-2** is a sophisticated simulation program that provides accurate, highly detailed, whole building energy analysis of multiple zones in a building. DOE-2 was developed in response to the energy crisis and has since undergone many refinements. Several other programs (e.g., eQuest) use DOE-2 as the simulation engine [7].

**EnergyPlus** is a more recent energy simulation engine that builds on the capabilities of DOE-2. EnergyPlus uses text-based input designed for advanced users [25].

**SOLAR-5** is a software package aimed at building design in early stages of planning. It’s designed for novice users, and is not capable of modeling complex mechanical systems [25].

**TRNSYS** is an energy simulation program that focuses on thermal and electrical systems.

![Figure 1.1](image.png)

**Figure 1.1:** (a) Increase in building energy usage and projected usage from 1980 to 2015 in Quads [9]. (b) CO₂ emissions from the three largest energy demands in a residential building [9].
within buildings. Applications include solar thermal and photovoltaic systems, low energy buildings, HVAC systems, and cogeneration [25].

A more extensive list of energy modeling software can be found on the US DOE website [25]. These programs were designed to simulate the energy usage of large buildings. Model developers recognized that building design aimed at maximizing energy performance takes into account a multitude of factors such as building shape and orientation, surface area, the ratio of window to wall area, and local climate, all of which can be simulated with the use of detailed computer models. Building energy simulation is now routinely applied during the design phase of new commercial buildings. In addition, the programs are used on existing buildings in an effort to conserve energy, decrease operational cost, and reduce emissions from electricity production [7].

Given the complexity embedded in building energy models and the requisite expertise required to operate the software and perform the simulations, their application is often limited to the design or retrofit of large commercial buildings. In addition, building simulations are performed serially, requiring the user to execute additional simulations to determine how specific design criteria impact energy performance. Given such large time commitments and expertise, only a limited number of houses are modeled for energy performance before construction.

Recently, researchers have become interested in energy modeling for residential homes. UCLA [37] and MIT [34] offer energy modeling freeware for users planning to build new homes or retrofit existing ones. The model user is provided with quantitative impacts associated with different energy conservation measures. Despite these efforts, the availability of programs targeted towards the residential sector is still limited.

More specifically, there are no programs geared towards the residential sector that use site-specific renewable energy and meteorological data at a fine temporal resolution (i.e., sub-hourly) to simulate home energy performance. The use of data at such a fine time resolution allows for an accurate representation of diurnal and seasonal variations in weather, renewable energy availability, and electricity dispatch.

In addition, building energy simulation programs require the user to manually change input
assumptions to explore the decision space, which requires significant user interaction and time. Modelers often attempt to optimize the building design by changing one or two parameters within the model, and rerunning the program to observe the effect. While existing building energy models can accurately characterize energy performance, their complexity and focus on simulation tend to obfuscate decision strategies aimed at improving home energy performance.

This thesis is motivated by the following research question: Where should a contractor or homebuyer focus major investments during new home construction to minimize the home’s carbon footprint during operation? Most home construction focuses on decreasing costs during the building phase with little or no consideration for energy savings over the life of the home. However, significant life-cycle cost and energy savings can be realized through higher upfront expenditures on energy efficient technology.

This thesis addresses the research question by describing the design and application of a mixed integer optimization model that uses recorded time series data to solve for home energy options that minimize CO$_2$ emissions subject to a monthly budget constraint. The model is based on real, time-series data is collected from the North Carolina Solar House in Raleigh, North Carolina [32]. The model takes into account key parameters related to building performance, such as renewable energy generation, meteorological data, building geometry, and technology cost and efficiency. The optimization model is used to help identify solutions and generate insight that maybe missed through simulation alone.

1.2 Background

1.2.1 The North Carolina Solar Center and GRID$C$ Project

The North Carolina Solar Center was initiated in 1981 with construction of a passive solar house built to conduct research and demonstrate the benefits of passive solar design, solar water heating, and solar electricity. The Solar House at North Carolina State University is one of the most visible and visited solar buildings in the United States. The Solar House is a resource to
help demonstrate solar energy technologies. Adjacent to the Solar House is the Research Annex, which acts as a laboratory for solar research [32].

Green Research for Incorporating Data in the Classroom (GRIDC) is a National Science Foundation (NSF)-funded project aimed at developing students’ higher order thinking skills in a data-rich learning environment. The GRIDC project is a partnership among the North Carolina Solar Center, the North Carolina State University College of Education, and the Friday Institute for Educational Innovation. As a part of the GRIDC research project, an extensive data collection system that monitors renewable energy sources at the North Carolina Solar House was installed [14]. During the first phase of the project, data collection devices were installed around the NC Solar House, Alternative Fuel Garage, and Research Annex. Data is sampled every 2-3 seconds, saved to a file, then averaged to a fifteen minute interval. The recorded data is then transmitted to a database housed on the NC State campus, which is made publicly accessible via the GRIDC website [14]. This recorded time series data served as the cornerstone of the modeling project described below.

Figure 1.2: Picture of the North Carolina Solar House with the photovoltaic and solar thermal panels. The PV array used in this analysis is located in the annex area, not the one shown here.
1.2.2 Data Collection

Data collection from the Solar House, Alternative Fuel Garage, and Research Annex began in January of 2009. Home electricity consumption was the only data field used from the Solar House. Data fields from the Research Annex used in this analysis are the PV module temperature ($^\circ$F), ambient temperature ($^\circ$F), energy produced by the 1kW solar array and 1kW wind turbine (kW), and incident solar radiation (W/m²). The complete set of data fields being collected through the GRID$_C$ project is more extensive [14]. The GRID$_C$ database stored data with a temporal resolution of fifteen minutes. The sub-hourly resolution allowed for accurate representation of
both home energy demand and renewable energy generation. Data at this temporal resolution also aided in troubleshooting the underlying data acquisition system. If time increments were larger than 15 minutes, inconsistencies in recorded data would have been harder to identify.
2.1 Time Series Data Analysis

Due to system testing and power outages, it was necessary to process the data received from the GRIDC website [14] to remove large gaps in the time series data. Scripts were written to parse the incoming data, identify the missing gaps in time series data, and eliminate the gaps across all concurrent data streams. Data from the Research Annex and Solar House were utilized for this project. Data collected directly from the house pertained to electricity consumption while the Research Annex, located adjacent to the house, collected data pertaining to renewable electricity generation and meteorology. This modeling project utilized recorded data pertaining to solar insolation (W/m$^2$), solar energy generated (kWh), wind energy generated (kWh), ambient temperature ($^\circ$F), and PV module temperature ($^\circ$F) from the Research Annex.

Each area of the Solar House Complex has a data acquisition unit, which gathers the recorded data and sends it to a computer outfitted with a Heliotronics software system [16]. The software reads the inputs taken two or three times each second, averages the values over a 15 minute
interval, then saves it in a Microsoft Access database file. A data retrieval program was written for the GRID project to retrieve the new data and update a local NC State database. This data is publicly available under the "Data Export" link in the left column of the GRID website [14]. With several different operating components and associated data acquisition systems embedded in the NC Solar House, fifty independent streams of time series data are collected and recorded. Hardware failures occurred intermittently and equipment was also taken off-line for repair and upgrade. As a result, several gaps exist in the fifty streams of recorded time series data. A key challenge of the data analysis was to identify the data gaps in each of the time series records, and then for consistency, remove the gaps across all of the time series records.

2.1.1 Importing Raw Data Files

The data imported into MATLAB were comma-separated by value (.csv) files. The first row of each .csv file imported into MATLAB contains the identifying labels for each column. The first column is the time stamp associated with the collected data, which provides the date and the time in fifteen minute intervals (e.g., 10/10/2009 2:45:00). MATLAB was unable to read .csv files containing alphanumeric symbols into a matrix, so for ease of use, the corresponding csv files were first converted to Excel spreadsheets. Use of spreadsheets simplified the import process at the expense of a slight increase in computing time. MATLAB reads the Excel file and saves the time stamps and data separately. The dates are then converted into a serial date number, which represents the fractional number of days since midnight on Year 0000. For example, January 1, 0000 is 1 and May 1, 2009 is 733894. Sub-daily times are represented as a fraction of a full day, where fifteen minutes is 0.0104166. The code for isolating full days in the recorded time series data with gaps larger than thirty minutes (reasons for gaps described in Section 2.1.2) first involves performing a rounding function to round the data to seven decimals. Seven decimal places is the required precision to specify each time interval. Data used in this analysis spans May 1, 2009 (serial date: 733894) to August 17, 2010 (serial date: 734368). The data is averaged every fifteen minutes, creating a total of 96 intervals per day.
Electricity data is obtained from a kilowatt hour (kWh) meter that records the cumulative amount of electricity consumed in the house and generated by the wind turbine and solar PV units since the unit began recording data. The difference in recorded values every 15-minute interval represents the amount of energy produced or consumed during that time increment. An error handling routine in MATLAB ensures that none of the values averaged over 15 minutes are negative. None of the systems work on a net metering basis, so the meter should never have a negative value. Any anomalous values found to be negative are changed to zero. A similar error handling procedure is performed for values that are too large. The wind turbine has a maximum rated power of one kilowatt hour (kWh) and the photovoltaic system is rated at 1.08 kW. Since neither system would be capable of generating more than 0.25 kWh in a fifteen minute period, all values greater than 0.25kWh were changed to zero.

2.1.2 Data Management

Once the data has been imported, the code compares the time stamps of both the annex and house to check for inconsistencies. If both values are present for a designated time increment, they are combined into a new matrix. If the time does not exist in both data sets, then the corresponding time stamp elements are given a zero value. The zero values were later removed by comparing the remainder of the data from the two sites.

A key challenge associated with the data analysis is handling the gaps in time series data due to power loss of the data acquisition systems. For example, the kWh meters work by logging the cumulative amount of energy that is produced or consumed. As a result, if a system goes down for four days and is then reset, the software will register all the electricity consumed during the outage in the first time interval after system restoration.

Separate MATLAB code was written to identify any section of data that was missing more than thirty minutes of data. Thirty minutes was selected instead of fifteen minutes for two main reasons. First, at the start of each day, the data logger system resets itself at 12:00:00AM. It clears data collected from 11:45:00PM from the previous day and begins logging new data. Once
fifteen minutes have passed, the computer will update the database recording information for 12:15:00AM instead of 12:00:00AM. As a result, fifteen minutes of data are lost at the beginning of each day. Second, rounding errors associated with the data logger time stamp lead to the periodic loss of a 15-minute data interval. This problem is likely caused by a software bug in the averaging script. Below is a sample of the actual sample time and the adjusted time from October 10th, 2009:

Table 2.1: Discrepancy in Sample Times

<table>
<thead>
<tr>
<th>Actual Time</th>
<th>Adjusted Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>10/10/2009 0:15:03</td>
<td>10/10/2009 0:15:00</td>
</tr>
<tr>
<td>10/10/2009 0:30:29</td>
<td>10/10/2009 0:30:00</td>
</tr>
<tr>
<td>10/10/2009 0:45:55</td>
<td>10/10/2009 0:45:00</td>
</tr>
<tr>
<td>10/10/2009 1:01:20</td>
<td>10/10/2009 1:00:00</td>
</tr>
<tr>
<td>10/10/2009 1:16:46</td>
<td>10/10/2009 1:15:00</td>
</tr>
<tr>
<td>10/10/2009 1:32:12</td>
<td>10/10/2009 1:30:00</td>
</tr>
<tr>
<td>10/10/2009 1:47:38</td>
<td>10/10/2009 1:45:00</td>
</tr>
<tr>
<td>10/10/2009 2:03:03</td>
<td>10/10/2009 2:00:00</td>
</tr>
<tr>
<td>10/10/2009 2:18:29</td>
<td>10/10/2009 2:15:00</td>
</tr>
<tr>
<td>10/10/2009 2:33:55</td>
<td>10/10/2009 2:30:00</td>
</tr>
<tr>
<td>10/10/2009 2:49:20</td>
<td>10/10/2009 2:45:00</td>
</tr>
<tr>
<td>10/10/2009 3:04:46</td>
<td>10/10/2009 3:00:00</td>
</tr>
<tr>
<td>10/10/2009 3:20:12</td>
<td>10/10/2009 3:15:00</td>
</tr>
<tr>
<td>10/10/2009 3:35:38</td>
<td>10/10/2009 3:30:00</td>
</tr>
<tr>
<td>10/10/2009 3:51:04</td>
<td>10/10/2009 3:45:00</td>
</tr>
<tr>
<td>10/10/2009 4:06:29</td>
<td>10/10/2009 4:00:00</td>
</tr>
<tr>
<td>10/10/2009 4:37:21</td>
<td>10/10/2009 4:30:00</td>
</tr>
<tr>
<td>10/10/2009 4:52:47</td>
<td>10/10/2009 4:45:00</td>
</tr>
<tr>
<td>10/10/2009 5:08:12</td>
<td>10/10/2009 5:00:00</td>
</tr>
<tr>
<td>10/10/2009 5:23:38</td>
<td>10/10/2009 5:15:00</td>
</tr>
<tr>
<td>10/10/2009 5:39:04</td>
<td>10/10/2009 5:30:00</td>
</tr>
<tr>
<td>10/10/2009 5:54:30</td>
<td>10/10/2009 5:45:00</td>
</tr>
<tr>
<td>10/10/2009 6:09:55</td>
<td>10/10/2009 6:00:00</td>
</tr>
</tbody>
</table>
Note that in just six hours, the sampling was nearly ten minutes off. Two hours later the gap is greater than fifteen minutes. When the rounding error grows to nearly 15 minutes, the recorded time correct stamp skips a 15-minute interval. Each day should contain 96 fifteen minute intervals. However, with the first interval missing from the beginning of the day, 95 daily data samples are left. Most days with collected data have 93 to 94 intervals, missing either one or two intervals due to the aforementioned rounding error. All days missing more than thirty consecutive minutes of data were completely removed from all data sets.

Data gaps were isolated by taking the difference between adjacent time stamps. A search is performed to identify differences greater than 0.02084, or slightly greater than 30 minutes. If the difference is greater than 30 minutes, the array index is tagged as the starting point of the time gap. The end gap is calculated as the gap start time plus the difference. For example, if a gap starts at 1:00:00AM on Day 1 and ends at 1:15:00AM of Day 2 then the time stamp associated with the start of the gap would be 1.0416666, while the time stamp associated with the end of the gap is 2.0520833. In this example, the code identifies the start of the gap as the beginning of Day 1 at 12:15:00AM and the end of the gap as the end of Day 2 at 11:45:00PM. Next, all times that fall between these time stamps are given a value of zero. The matrix now has only full days with gaps of thirty minutes or less. Since both .csv files use the same date and time export data, the new time and date matrix is substituted for the current matrix. A loop is implemented to eliminate all rows where the date and time value equals zero.

For the remaining days, a complete time series is created from midnight of the starting date until 11:45PM of the last day recorded. With the complete time series, MATLAB then performs a linear interpolation using all adjacent data points to populate the thirty minute gaps. The remaining time series data –consisting of whole days– includes 26,975 15-minute intervals covering 281 of the 474 days.

In January 2009, an unfortunate mishap occurred at the Research Annex during an educational training workshop. While lowering the wind turbine, loose safety wires led to a precipitous fall that destroyed the wind turbine. Only recently has a new 2.4kW SkyStream turbine been
installed. To account for the last 8 months of wind generation data, the initial eight months of wind electricity generation data are duplicated and concatenated to the last day of wind turbine operation. The extension of wind data creates a record of equal length to the other collected time series data. Due to the poor potential wind resources of the Piedmont region of North Carolina, this kludge is not expected to have a significant impact on the results. Data files to be read into the optimization model are written for the electricity generated by both solar PV and wind, solar insolation, and the household electricity consumption.

2.2 Input Data

2.2.1 Emissions Data

A key challenge of this analysis is to estimate the emissions associated with electricity consumption that takes place in the home. When electricity is generated from renewable sources within the residential sector, conventional generators on the grid – and their associated emissions – are displaced. Estimating emissions associated with electricity generation requires knowledge of how regional U.S. electricity grids are managed. The North American Electric Reliability Corporation (NERC) is responsible for ensuring the reliability of the nation’s electric grid [23]. NERC is responsible for establishing and enforcing standards to ensure the power system reliably meets regional US electricity demands. NERC oversees the interconnection of all US electric utility control areas and ensures proper measures are in place to mitigate disturbances. The US and parts of Canada are divided into eight NERC regions. In addition, the US Environmental Protection Agency’s Emissions and Generation Resource Integrated Database (eGRID) further subdivides the NERC regions into 27 separate subregions. EPA eGRID provides estimates of resource mix and emissions rates for each of the 27 subregions.

Because wind and solar photovoltaic electricity generation data from the NC Solar House are recorded every 15 minutes, this thesis sought data to characterize emission rates at equivalent temporal scales. EPA eGRID only provides average annual emissions rates; however, this
analysis uses hourly average emission rates based on Hausman et. al [15]. To estimate hourly emissions rates, Hausman et. al [15] utilize generation and emissions data for US fossil-fuel powered generating plants from the hourly emissions database of the EPA Clean Air Markets Program [15, 5]. The authors develop several methods to estimate hourly emissions rates based on the responsiveness of each plant to changes in regional electricity demand. Of the emission rate estimates developed by Hausman et. al [15], the flexibility-weighted hourly average emissions rate (FW-HAER) is used in this analysis. Because the output from wind and solar photovoltaics are both variable and intermittent, the generation units most likely to respond are capable of quickly ramping output. The FW-HAER is calculated by identifying the power plants most likely to ramp their output up or down to respond to variations in load. The FW-HAER is calculated as follows:

\[
FW-HAER_t = \frac{\sum (F_i ER_{i,t})}{\sum F_i} \]  

(2.1)

where:

- \(F_i\) is the number of hours spent ramping normalized by the total hours of operation for unit \(i\) and \(ER_{i,t}\) is the emission rate for unit \(i\) during hour \(t\) [15]. As a result, units that spend more time ramping output are given a higher flexibility weighted rating than those consistently running at or near rated capacity. Since the NC Solar House is located in North Carolina, data from the corresponding SERV Virginia/Carolina (SRVC) EPA eGRID subregion was used.

Hausman et. al [15] data is based on 2005 emissions. Little in the way of electrical power infrastructure has changed in the past five years so the effect of five year old data is expected to be negligible. Since data collection at the NC Solar House starts on May 1, the Hausman et. al [15] data is realigned to begin on the same day of the year. In addition, each hourly estimate from Hausman et. al [15] is replicated three additional times to represent each fifteen minute interval within each hour. The result is 8760 points of hourly data broken into 35,040 15-minute intervals. Because the time series data collected from the NC Solar House is longer than one year, the emissions data from [15] is replicated from the start date (May 1) until the last day of
data (August 17), which are appended to the end of the current year. The time series and the emissions rates are concatenated to form a two-column matrix with the serial date number in the first column and the CO$_2$ emissions rate corresponding to the date number in the second column. The new data set represents a complete listing of all 474 days and emissions factors. The time intervals associated with missing gaps are removed as described above, reducing it to the same length as all other data sets. The complete emissions matrix is then written to .csv format for import into the optimization model.

### 2.2.2 Hot Water Demand

Residential water heating can be a large contributor to energy bills and can account for about 11% of energy consumed in a home [19]. Residences across the US use a variety of fuel sources to heat water. Of the 111 million households in 2005, almost 59.8 million used natural gas to heat their water, nearly 43.3 million had electric water heaters, and only 10.1 million houses used fuel oil or LPG for heating their hot water [11].

The amount of hot water required each day varies by household and with point of use. To determine the amount of energy required to heat water using different water heating technologies, the usage requirements were first determined. The gallons of hot water consumed per use are shown in Table 2.2.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Gallons per Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clothes washing</td>
<td>32</td>
</tr>
<tr>
<td>Showering</td>
<td>20</td>
</tr>
<tr>
<td>Bathing</td>
<td>20</td>
</tr>
<tr>
<td>Automatic dishwashing</td>
<td>12</td>
</tr>
<tr>
<td>Preparing food</td>
<td>5</td>
</tr>
<tr>
<td>Hand dishwashing</td>
<td>4</td>
</tr>
</tbody>
</table>
The Table 2.2 values are referenced when the model calculates the amount of energy required for hot water heating; however, the usage patterns vary from household to household. A frequency for each activity is based on extrapolation from personal experience. For the home modeled in this analysis, the following is assumed: eight clothes washings per week, three showers or baths a day, four dishwashings a week, food preparation once a day, and once per day dishwashing (by hand) occurred. Under the hot water usage scenario, the total hot water required for the model home is 676 gallons a week. In addition, the hot water consumption patterns follow a diurnal pattern. Hot water consumption occurs in the early morning around 6AM and after work around 6PM. Eighty-eight gallons of hot water are consumed each weekday while 120 gallons are consumed on Saturday and 110 gallons on Sunday. Note that activities such as hand and face washing along with other small hot water demand activities are assumed to have a negligible effect on the total household energy consumption. The temporal specification of hot water demands allows the model to match electricity generation to consumption and also makes the model sensitive to diurnal and seasonal changes in electric grid emissions factors, in cases where water is heated with grid electricity.

2.2.3 Water Heating Technologies

Water is typically heated using three distinct energy sources: electricity, natural gas, and solar thermal. Natural gas water heaters are popular due mostly to the low purchase price, high efficiency, and low fuel cost. Electric hot water heaters are also common given their low purchase price, but rely on electricity, which is generally more expensive and less efficient than natural gas. Solar thermal water heaters use roof-mounted collectors to absorb solar energy and transfer the heat to a working fluid (typically water or glycol). Solar thermal units have high investment costs but near zero marginal costs to heat the water. Solar thermal often utilizes a backup heating supply for periods when the solar energy is unavailable. This backup supply is typically the same system the homeowner had in place prior to the installation of the solar thermal water heating system, such as electricity or natural gas.
Each water heating technology has a different efficiency associated with its use. Hot water heater efficiencies are rated by the Energy Factor (EF). The EF estimates the amount of hot water produced per amount of fuel consumed over a given time period (typically a day) by accounting for recovery efficiency, standby losses, and cycling losses [26]. The total cost of different technologies to heat water is determined not only by the EF, but also by the investment cost and assumed fuel price. A high EF electric water heater will usually have a higher annual and lifetime cost than a natural gas water heater with a lower EF because the cost of one kilowatt hour of electricity is higher than the equivalent amount of heat energy from natural gas.

In the model representation, the required energy is calculated as the amount of thermal energy necessary to maintain the set tank temperature. This tank temperature can change due to hot water usage or losses to the surroundings. The amount of thermal energy required to heat a given volume of water can be calculated as follows:

\[
Q = cm\Delta T
\]  
(2.2)

where:
- \(Q\) is the heat required to raise the water to desired temperature,
- \(m\) is the mass of water,
- \(c\) is the specific heat of water,
- \(\Delta T\) is the change in water temperature due to heat addition \(Q\).

Specific heat is defined as the amount of heat required to raise the temperature of a unit mass \(m\) (water in this case) by one degree [6]. For example, raising one pound of water by 1°F requires 1 Btu of heat energy\(^1\). The change in temperature is the difference between the set tank temperature and incoming water temperature (ground temperature). The desired water temperature is assumed to be 120°F since the Department of Energy and American Council for an Energy-Efficient Economy recommend this temperature as the most efficient and least expensive [26, 1]. The ground temperature varies throughout the year and across different

\(^1\)Specific heat varies as temperature changes; however, it is assumed to remain constant across all calculations
latitudes. For Raleigh, North Carolina, the average ground temperature is around 65°F [21].

**Natural Gas Water Heating**

Where available, natural gas has been the most cost-effective fuel choice for heating water, due largely to low fuel prices\(^2\). Natural gas water heaters typically burn fuel at the base of the storage tank, which heats the bottom and allows hot water to naturally circulate via convection through the storage tank. Heat transfer efficiency for the natural gas heaters generally ranges from 60% to almost 90%, depending on the model efficiency and technology. Most energy factors (EF) for conventional gas fired units are approximately 60% efficient [1]. Natural gas is generally purchased in cubic feet or therms (100,000 Btu). Approximately 1,030 Btu are in a cubic foot of natural gas at standard temperature and pressure [6]. Natural gas prices are volatile but assumed to be $1.30 per therm throughout the year [10].

**Electric Water Heating**

Electric hot water heaters are similar to natural gas heaters, but use electricity to power a resistance heater in the water storage tank. Heat transfer within electric water heaters is 100% efficient; however, upstream electricity generation is subject to the thermal efficiency of power plants. Note that the EF does not take into account the upstream thermal efficiency of power plants. The minimum allowable EF for electric models is 0.90 and some higher efficiency models can have EF ratings of approximately 0.95 [1]. The 5–10% losses stem from standby and cycling losses. When calculating the energy required for electric hot water heating, it is assumed that electricity prices in the southeast region are $0.105 per kWh [28].

**Solar Thermal Water Heating**

Solar thermal hot water production has become increasingly popular in the US as a result of rising fuel prices. The cost-effectiveness of solar hot water systems is directly proportional to

---

\(^2\)Assuming natural gas prices are $1.30 per therm (100,000 Btu) and electricity prices are $0.105 per kWh. 1 kWh = 3,413 Btu so converting both to $/Btu leaves natural gas = $0.000013/Btu and electricity = $0.0000308/Btu
the strength of the solar resource, and system performance is therefore dependent on location. In addition, most solar collectors are used in conjunction with conventional heating sources (i.e., electricity or natural gas) to provide backup heating when the solar resource is unavailable. As a result, the cost-effectiveness of solar hot water is also highly dependent on storage tank size. A larger tank allows for more water to be withdrawn and replaced with cold water without a significant drop in the total tank temperature, which reduces the need for backup heating units.

Solar water heaters utilize a solar collector to transfer photon energy to thermal energy stored in water. The collector at the NC Solar House utilizes propylene glycol as the heat transfer fluid, which flows through tubes embedded in the panel. Nearly all solar thermal hot water systems work by monitoring the temperature of the tank. When the temperature of the collector panel is approximately 10°F above the temperature of the tank, a pump will turn on and circulate the hot glycol (see Figure 2.1 for a detailed schematic). If the temperature of the panel does not reach that of the tank, then the water must be heated by the auxiliary electric or natural gas heating source. The hot glycol circulates through a closed loop in the tank, which raises the water temperature until it reaches a maximum value. When the maximum allowable temperature is reached, the system shuts down and drains the fluid to the tank to prevent further heating.

The GRIDC project has sensors on the pipes of the Schuco Slimline V solar thermal panel to monitor the amount of heat transferred into the storage tank. On sunny days, around 2PM, when the 80 gallon tank reached the maximum operating temperature and shuts off the pump, the heat flow within the system stops. At that point, no more glycol is routed to the panel and no heat transfer is registered. As a result, from around 2PM until the sun sets, no additional heat energy appears to be available to the system, even though solar insolation levels remain high. To represent hot water withdrawal under different demand scenarios, the solar thermal hot water system was modeled directly simulating the performance based on panel specifications and insolation values from the gathered data.

In addition to the initial investment cost, the model needs to calculate the demand for
auxiliary fuel associated with the solar thermal system, which is partially dependent on the location-specific quality of the resource. A model was created based on the Hottel-Whillier-Bliss equation [13], which provides an estimate of the amount of energy available to heat the hot water supply of a house. Insolation levels (W/m²) are used to calculate the thermal energy transferred to the water as follows:

\[ q_C = A_C F_R (\tau_S \alpha_S I_C - U_C (T_{C_i} - T_a)) \]  

(2.3)

where:
- \( q_C \) is the instantaneous steady-state useful energy,
- \( A_C \) is the collector surface area (2.137 m²),
- \( F_R \) is the heat removal factor of the collector,
- \( I_C \) is the insolation value on the plane to the collector,
- \( \tau_S \) is the collector transmittance (0.91),
- \( \alpha_S \) is the collector absorptivity (0.95),
- \( U_C \) is the overall heat loss coefficient for the collector (3.70),
- \( \eta_C \) is the collector efficiency (0.706),
- \( T_{C_i} \) is the inlet fluid temperature into the collector,
- \( T_a \) is the ambient temperature.

The data sheet for the Shuco Slimline V provided key information regarding unit specifications and can be found on the GRID\(_C\) website under the “System Components” link [14].

As stated above, the tank is assumed to be 50 gallons, or about 190 liters. One liter of water weighs one kilogram, so the tank contains roughly 190 kilograms of water. Also, the internal tank temperature is maintained at 120°F and the ambient outdoor temperature is obtained via the GRID\(_C\) database.

The panel temperature is determined by the tank temperature at the previous time interval and the design parameters enumerated above. Thermal energy for a given fifteen minute interval is used to update a tank temperature, and the temperature associated with the next time
Figure 2.1: Schematic of a typical solar thermal hot water system [33]. Once the temperature of the panel is warmer than the tank, a pump begins circulating the working fluid through the panel. As the water flows through the panel, it is warmed by absorbed solar energy then returned to the tank and either sent through a heat exchanger or directly mixed with the tank water.
interval is calculated based on heat loss from the tank, surrounding temperature, surface area of the tank, and R-value. To prevent the risk of excessive pressure buildup in the water tank, the maximum tank temperature is 180°F. If the temperature exceeds 180°F then it is set to 180°F. An analogous procedure is invoked for all temperatures below 120°F. The lower bound temperature is reset because the true tank temperature will never drop below the minimum set temperature as a result of the auxiliary heating unit. This low threshold also ensures the inlet temperature is correct for calculating the available energy for subsequent time intervals.

Most of the parameters associated with the performance of solar hot water units vary with the collector design. Some collectors represent a standard flat plate design while others are more advanced and involve evacuated tubes that encase a copper plate, which is used to transfer heat more efficiently. Compared to conventional water heating units, the requirements for site specific insolation values make estimation of hot water production from a solar thermal collector more challenging.

A minimum amount of thermal energy is required to maintain a temperature high enough for heat to be transferred from the collector to the water. Reviewing data in the GRIDC database shows that insolation values of 200W/m² or greater are required to enable heat transfer. The solar collector simulation assumes that the collector converts the insolation levels directly to heat without taking into account this 200W/m² threshold for heating. A short MATLAB script was written to find and remove all insolation values less than 200W/m². Without this correction, energy values would be inaccurately high.

Once the tank temperature at every time interval has been calculated, the amount of heat required from the auxiliary heating system is estimated. The required heat accounts for both the hot water demand scenario and the heat lost to the surroundings. The energy loss associated with consumption of hot water to meet specified demands and the heat lost to the surroundings are subtracted from the total amount of heat supplied by the solar thermal panel. Auxiliary heat is only necessary if the tank temperature is below 120°F. The amount of auxiliary heat required is calculated by subtracting the actual amount of thermal energy remaining in the
tank after losses and hot water usage. The required heat was matched to the appropriate time stamps and written to a .csv file.

2.2.4 Building Design

The house plan used for the model is a simple two-story structure with a square footprint. The total house is 148.6m² (1600ft²), split evenly between two floors. The footprint is 8.63m by 8.63m and each floor is 3.05m tall. This basic layout creates a structure with each side having a total surface area of 26.3m². The house is assumed to be oriented along a true North/South axis, allowing the roof maximum access to the sun for the PV panels and solar thermal hot water collector. The floor and attic have the same dimensions as the house width and depth, resulting in a surface area of 74.5m². The house has a vented crawl space, which is common in North Carolina and allows for standard insulation practices. The home has three windows per floor per face (six on each face) for a total of 24 windows. Each window measures 1.12m² with a total window area of 26.76m² leaving a total wall area of 78.4m² for wall insulation.

Figure 2.2: CAD sketch of the modeled house. Dimensions are found in Section 2.2.4.
The total roof area is the main constraint on the photovoltaic array size. The roof angle is assumed to have the same pitch as the latitude of Raleigh (36° North) and the roof is split evenly in two halves, one facing north and the other south. Eaves are 24 inches, thus the depth for each side (north and south) is 4.92m with the width remaining at 9.84m. Total south-facing roof area for the house is 59.9m$^2$. Sunpower, a solar PV manufacturer, sells panels that output 225W with dimensions of 1.59m by 0.8m. Using these panel dimensions, the roof can accommodate 35 panels. The modeled panels were assumed to produce 200W instead of 225W, to provide a more conservative estimate. The roof area limitation allows for a PV array with a maximum of 7kW installed capacity. There is sufficient area on the roof for the solar thermal collector when allowing the PV array to reach 7kW. The total roof area allows for both solar energy technologies (solar thermal and solar PV) to be implemented simultaneously, but little roof surface remains.

### 2.2.5 Insulation

Insulation is assumed to be present on all surfaces of the home: floor, ceiling, walls, and windows. Each surface has an option of three different insulation values detailed in Section 3.3.2. Insulation prices were based on retail prices found at large home improvement chains [18]:

<table>
<thead>
<tr>
<th>Insulation Level</th>
<th>Area</th>
<th>Cost ($)</th>
<th>$/ft$^2$</th>
<th>$/m^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-11</td>
<td>50.0</td>
<td>15.0</td>
<td>0.30</td>
<td>3.23</td>
</tr>
<tr>
<td>R-19</td>
<td>77.5</td>
<td>37.6</td>
<td>0.49</td>
<td>5.23</td>
</tr>
<tr>
<td>R-30</td>
<td>58.6</td>
<td>42.8</td>
<td>0.73</td>
<td>7.86</td>
</tr>
<tr>
<td>R-38</td>
<td>42.7</td>
<td>39.7</td>
<td>0.93</td>
<td>10.01</td>
</tr>
<tr>
<td>R-50</td>
<td>63.4</td>
<td>71.3</td>
<td>1.13</td>
<td>12.11</td>
</tr>
</tbody>
</table>

The window insulation values follow a similar structure to that of the other building surfaces. The windows to be installed are assumed to be new and draft-proof. Advanced window
construction such as vacuum-insulation or double and triple glazing were omitted for simplicity. Prices for the windows are from a large home improvement store [18]:

<table>
<thead>
<tr>
<th>Window Insulation Level</th>
<th>$/m^2</th>
</tr>
</thead>
<tbody>
<tr>
<td>R- 4</td>
<td>107.6</td>
</tr>
<tr>
<td>R-11</td>
<td>215.3</td>
</tr>
<tr>
<td>R-17</td>
<td>322.9</td>
</tr>
</tbody>
</table>

The total cost per surface (i.e., floor, walls, ceiling, windows) was found using the price per unit area (for insulation of a particular R-value) multiplied by the surface area.

Temperatures for Insulation

Home insulation is an important consideration in building design. The model considers four different surfaces (floor, walls, windows, and ceiling) and optimizes over three R-values (low, medium, high) of insulation per surface. An additional MATLAB script was developed to format temperature data gathered at the Research Annex of the NC Solar Center. The recorded time series temperature data allow the calculation of temperature differences between the inside and outside walls, floor, and attic.

To accurately measure heating and cooling demands, different temperature files were created based on the inside and outside temperatures. For simplicity, it is assumed that the house is kept at a constant indoor temperature, with a single-zone thermostat set to 72°F year round. This thermostatic temperature is varied parametrically in the analysis section.

Extreme summer attic temperatures are common in attics in the Southeastern US. Many homes have vented attics with dark shingles. This configuration has been shown to produce some
of the hottest temperatures of any configuration when testing attic temperatures [30]. In most southeastern states, the maximum attic temperature is around 140°F [30]. The main purpose for including the attic temperature in the insulation optimization was to include the effect of having an extremely high temperature against the ceiling insulation of the house during summer months. The temperature of the attic is normalized to a maximum of 140°F across the year as seen for the southeast region [30] by using the temperature of the PV panels as a reference for the amount of energy absorbed by a dark surface. Once normalized, the temperatures were split into two files corresponding to attic temperature above and below the internal house temperature of 72°F. A total of four different temperature .csv files were passed to the optimization model. The different files represent the temperatures for outside ambient temperatures and attic temperatures, above and below the set indoor temperature.

2.2.6 Scalar Values

Several scalar quantities were fixed in the model. The cost of retail residential electricity through Progress Energy, the local utility that supplies electricity to the NC Solar House, ranges depending on the time of year [28]. Based on average Progress Energy rates for 2010, a constant value of $0.105 was assumed for all electricity bought and sold under net metering\(^3\). The interest rate used to amortize all investments in the model was five percent. The price for natural gas was assumed to be $1.30 per therm ($0.01232/MJ). The burning of natural gas also emits carbon dioxide. The emissions rate is assumed to be 0.0503 g/kJ of fuel combusted. The ground temperature is nearly constant throughout the year in most parts of the country at depths greater than 2 meters. In North Carolina, the average ground temperature varies slightly by season, but is assumed to be 65°F [21]. This temperature is also used for the crawlspace.

---

\(^3\)Net metering is the selling of unconsumed electricity generated by an independent producer. The meter spins one way when grid electricity is being consumed and the opposite way when renewable systems produce more electricity than the user consumes.
3.1 The Optimization Model

The formatted time series data described in Chapter 2 serves as input to the optimization model. The objective of the model is to optimize a home’s energy system to minimize the CO₂ emissions produced over a range of budget constraints. The model is limited by a number of different constraints that represent technology operation and heat transfer associated with hot water supply as well as space heating and cooling.

As in a typical home, the modeled home has demands placed upon different systems within the house (e.g., water heating, electricity demand, space heating) and these demands must be met via an array of available technologies chosen by the model (e.g., wind generation, solar thermal water heater, electrical air conditioning). Figure 3.1 displays the demands on the home as well as the available technologies to meet the demands. The $\bar{X}$s represent different technology options available to the model. The thumbnail plots represent time series data pulled from the GRIDC database and analyzed according to the description in Section 2.1.
On the right hand side of Figure 3.1 are different technology sets that consume energy to meet a given end-use demand. The demands represented in the model are space heating (SH) and cooling (SC) required to maintain a static indoor temperature, hot water (HW), and general appliance electricity usage. The optimal choice of space heating and cooling technologies is related to the insulation value ($I$, for each surface $S$) which is also selected by the model as well as the outside ambient temperature ($T_o$). The left hand side of the figure shows different energy sources the model can utilize. Solar and wind profiles are provided from the GRID database while natural gas and grid-based electricity are assumed to be available on demand. The lines show how each energy source is connected to end use demands. Only energy in the form of electricity can be supplied to meet electricity demand while the hot water requirements can be met by solar heat, electricity, renewable electricity, or natural gas. Space heating and cooling can be met by natural gas or electricity using several different technologies.

### 3.2 Objective Function

The objective of the model is to minimize CO$_2$ emissions from electricity usage and the direct combustion of natural gas summed across the modeled time intervals:

$$\min_{CO_2} \sum_t \left( (E_{e,t} - \sum_r E_{r,t}) \times CO_{2e,t} \right) + \sum_t \left( NG_{HW,t} + NG_{SH,t} \right) \times CO_{2,NG}$$  (3.1)

See Appendix 6 for a description of model nomenclature. CO$_2$ emissions from electricity production are based on hourly data. Natural gas consumption is summed over time for the gas-fired hot water and space heating technologies. Technology choices are limited by a monthly budget constraint. The total monthly budget for household energy is limited as follows:

$$C_E + C_{NG} + C_r + C_{HW} + C_I + C_{SH} + C_{SC} \leq B_m$$  (3.2)
Figure 3.1: Diagram shows the model topology. The left hand side represents the fuel inputs available to the model. The $\bar{X}$’s represent capacity decision variables for the renewable energy options. The $U_t$’s represent time series electricity generation, which is derived from GRIDC time series data. The product of $U_t \times \bar{X}$ gives the total wind or solar PV production per time interval. The $T_o$ plot on the right hand side represents the ambient temperature data collect at the Solar House and used to calculate heat flows. Natural gas and grid-based electricity are available as well. The kWh plot below Home Electricity Demand represents electricity consumption from the Solar House over time. The lines indicate the connection to demand side components of the home. The model selects the optimal technologies to meet each household demand. Available technology options are listed within respective demand brackets.
3.3 Decision Variables

The model can be executed to either minimize CO$_2$ emissions or total energy-related monthly cost. Every variable has a cost associated with installation and/or operation. All discrete options are limited to only one choice per category:

\[
\sum_{I} X_{I,S} = 1
\]

\[
\sum_{HW} X_{HW} = 1
\]

\[
\sum_{SH} X_{SH} = 1
\]

\[
\sum_{SC} X_{SC} = 1
\]

3.3.1 Cost Calculations

The model tracks both investment and fuel costs associated with installation and utilization of different residential energy options. Monthly costs are used as the metric because most home utility bills and mortages are based on monthly payments. This approach is particularly relevant when considering energy options during the home construction phase, as capital investments are included in the home mortgage, which is paid on a monthly basis. In addition, fuel costs associated with electricity and natural gas consumption are also paid monthly.

Because only 281 days are contained in the time series simulation, marginal costs for fuel and electricity must be scaled from a daily to monthly average value. As a result, the average daily cost within each month is calculated and then multiplied by the number of days per month.
Table 3.1: The decision variables in the model. All are binary choices among a set of technology options except for wind and solar capacity, which are continuous variables with an upper bound on installed capacity.

<table>
<thead>
<tr>
<th>Decision Categories</th>
<th>Decision Variables</th>
<th>Variable Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Renewable Electricity</td>
<td>Solar PV Wind</td>
<td>Continuous Solar: [0-7] kW</td>
</tr>
<tr>
<td></td>
<td>Wind</td>
<td>Continuous Wind: [0-10] kW</td>
</tr>
<tr>
<td>Hot Water</td>
<td>Natural Gas</td>
<td>Binary</td>
</tr>
<tr>
<td></td>
<td>Electricity</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Solar Thermal (NG aux)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Solar Thermal (electric aux)</td>
<td></td>
</tr>
<tr>
<td>Space Heating</td>
<td>Geothermal Heat Pump</td>
<td>Binary</td>
</tr>
<tr>
<td></td>
<td>Air Source Heat Pump</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Natural Gas</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Radiant Heating</td>
<td></td>
</tr>
<tr>
<td>Space Cooling</td>
<td>Geothermal Heat Pump</td>
<td>Binary</td>
</tr>
<tr>
<td></td>
<td>Air Source Heat Pump</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Electric Air Conditioning</td>
<td></td>
</tr>
<tr>
<td>Insulation</td>
<td>Ceiling</td>
<td>High, Medium, Low</td>
</tr>
<tr>
<td></td>
<td>Wall</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Window</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Floor</td>
<td></td>
</tr>
</tbody>
</table>

Amortization of capital investments on a monthly basis is done as follows:

\[
C_M = P_{inv} \left( \frac{i/12}{1 - (1 + \frac{i}{12})^{L*12}} \right) \tag{3.3}
\]

where \(C_M\) is the monthly amortized cost for the technology, \(i\) is the annual interest rate (5%), \(P_{inv}\) is the investment price for a technology, and \(L\) is the lifetime of each modeled technology in years. Compared to a net present value calculation, amortization simplifies the cost calculation given the different lifetimes among technology options.

3.3.2 Insulation

The quality of insulation is measured by its R-value, which represents the material’s ability to restrict the flow of heat. The higher the R-value, the more effectively the insulation can inhibit
the flow of heat and, typically, the higher the investment cost. In practice, insulation is only available at a limited number of R-values. For simplicity, insulation is modeled at three different R-values to represent low, medium, and high R-values (See Table 3.2).

The range of R-values for insulation used in North Carolina is based on the Oak Ridge National Laboratory recommendations for new wood-framed houses [20]. Insulation investment cost is calculated as follows:

\[
C_I = \sum_{S} (A_S \bar{X}_{I,S} C_{I,S}) \left( \frac{i/12}{1 - (1 + \frac{i}{12})^{L_I \times 12}} \right)
\]

(3.4)

The first term in parenthesis yields the total cost per surface, where \(\bar{X}_{I,S}\) is a binary variable representing the choice of insulation for each surface. The second term amortizes the insulation investment over the lifetime of the technology. See Appendix 6 for a more detailed description of the nomenclature.

<table>
<thead>
<tr>
<th>Surface</th>
<th>Level</th>
<th>R-Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Floor Insulation</td>
<td>low</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>mid</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>high</td>
<td>30</td>
</tr>
<tr>
<td>Wall Insulation</td>
<td>low</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>mid</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>high</td>
<td>30</td>
</tr>
<tr>
<td>Ceiling Insulation</td>
<td>low</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>mid</td>
<td>38</td>
</tr>
<tr>
<td></td>
<td>high</td>
<td>50</td>
</tr>
<tr>
<td>Window Insulation</td>
<td>low</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>mid</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>high</td>
<td>17</td>
</tr>
</tbody>
</table>
3.3.3 Renewable Energy Systems

Pricing data for residential renewable energy systems is difficult to find in the public domain. In addition, the cost-effectiveness of renewable energy systems is affected by factors such as local labor prices, solar and wind resource availability, investment cost, PV panel design, and balance of system costs. Federal and state tax incentives also play an important role in the deployment of residential renewable energy systems by defraying the high investment cost of renewable energy systems. Some studies have been published that approximate both current and future prices for such energy systems, although not within a specific region [38]. The total monthly cost of the renewable electricity components (wind or solar PV) is calculated as follows:

\[
C_r = X_r \left[ (1 - T C_r) * P_{inv} \left( \frac{i/12}{1 - (1 + \frac{i}{12})^{L_r*12}} \right) + \left( F_r * \frac{281 \text{ days}}{365} * 30 \text{ days} \right) + \left( V_r \sum_{t} E_{r,t} * \frac{30 \text{ days}}{281 \text{ days}} \right) \right] \quad (3.5)
\]

where the first term represents the net investment cost, the second term is the yearly fixed cost per kilowatt installed ($/kW-yr), and the last term is variable cost based on utilization of each technology ($/kWh produced).

Solar Energy

In 2006, the U.S. Department of Energy estimated the cost of installing a 1kW photovoltaic system at $7,500, while the forecast price for 2010 was $4,700 [35]. Residential scale PV installations (<10kW) in 2008 ranged greatly in price, according to a NREL report [24]. The average installed cost in the US was $7.9/watt and the cost difference between building integrated systems and rack mounted systems was $0.9/W ($8.3/W vs. $7.4/W, respectively) [38]. Using this information, a cost of $7.6/W installed PV is used in the model, based on increasing competition in companies installing renewable energy systems in North Carolina. The fixed and variable costs are assumed to be $50/kW-yr and $0.01/kWh, respectively. The maximum size of
the PV system is constrained to 7 kW based on available roof space.

**Wind Energy**

The Wind Energy Operations and Maintenance (O&M) Report [2] found that the average O&M cost for wind turbines is $0.027/kWh generated for large installations. This estimate is considerably greater than the report’s original prediction of around $0.005/kWh for large installations [2]. Since this project focuses on small scale residential turbines, the O&M cost is estimated at $0.02/kWh, higher than original predictions by a factor of four. The price of wind installations has risen steadily since reaching a minimum around 2000-2001. An increase of about $170/kW occurred between 2008 and 2009 alone. Average costs for large scale systems were around $2,120/kW in 2009 [39]. The upward trend is expected to continue in 2010, resulting in an estimated investment cost of $2,230/kW [39]. For the purpose of this modeling exercise, the installation cost of residential scale turbines is estimated at $4,000/kW, a higher cost that reflects the small-scale nature of residential units. The wind turbine is assumed to have a 30 year lifespan. These costs do not include state and federal tax incentives for installation, but are taken into consideration later. A 10 kW limit is placed on the wind turbine size to prevent the installation of unrealistically large residential wind turbines\(^1\).

### 3.3.4 Electricity Cost

North Carolina has two major residential pricing schemes for renewable energy generation. Until recently, electric utilities in North Carolina would pay their avoided cost for renewable electricity generation, which amounts to approximately $0.04/kWh [8]. More recently, an independent non-profit organization named NC GreenPower pays producers of renewable energy an additional amount to offset their investment costs by distributing funds collected from utility customers who voluntarily support renewable energy development [22]. To participate in NC GreenPower, all renewable electricity generated by a customer must be sold to the local utility at a higher

---

\(^1\)The wind resource at the Solar House is extremely low (calculated to have a 2% capacity factor).
rate, then purchased back at a lower rate. Prices paid by NC GreenPower vary by the type of system installed. In addition to the $0.04/kWh of avoided cost, small PV generators receive $0.15/kWh and small wind generators receive $0.09/kWh for a total price of $0.19 and $0.13 per kWh for solar PV and wind, respectively. Net metering is an alternative to NC GreenPower for renewable electricity generators in North Carolina. Under net metering, electricity is bought and sold at the same price. For each participating household, electric utilities simply subtract the total amount of renewable electricity delivered to the grid from the amount purchased each month. If more energy is generated than consumed, the customer receives a check for the difference. Although NC GreenPower is more lucrative for residential electricity generators, the model uses net metering because it is available in many other states beyond North Carolina [8].

The equation for electricity cost is:

\[
C_e = P_e \sum_t \left[ E_{mi,t} + \left( \sum_{SH,SC,S} \frac{Q_{S,t} \bar{X}_{SH,SC,e}}{\eta_{SH,SC,e}} \right) + \left( E_{HW,t} \bar{X}_{HW,e} \right) - E_{r,t} \right] * \frac{30 \text{ days}}{281 \text{ days}} \tag{3.6}
\]

A description of each term can be found in Appendix 6. The first expression in the above equation represents the home’s miscellaneous electricity demand. The second term represents electricity demand from space heating and cooling systems. The third term is electricity demand from a hot water heating system, and the fourth term is renewable energy generated each time increment. Note that if a non-electricity-based technology is selected, the relevant term multiplied by \( \bar{X} \) will drop out. All terms are divided by the total number of days within the data set and multiplied by the number of days in a month to normalize cost to a monthly value.

### 3.3.5 Natural Gas Cost

In the model, only space heating and hot water production can contribute to natural gas consumption. The cost equation for natural gas is:

\[
C_{ng} = P_{ng} \sum_t \left[ \sum_{SH,SC} \frac{Q_{S,t} \bar{X}_{SH,SC,ng}}{\eta_{SH,SC,ng}} + \left( NG_{HW,t} \bar{X}_{HW,ng} \right) \right] * \frac{30 \text{ days}}{281 \text{ days}} \tag{3.7}
\]
A description of each term can be found in Appendix 6. The first expression in the above equation represents the natural gas demand from space heating and cooling systems. The second term is relevant natural gas demand from a hot water heating system if selected by the model. All terms are divided by the total number of days within the data set and multiplied by the number of days in a month to normalize cost to a monthly value.

3.3.6 Water Heating

The model takes into account four options for heating water: electricity, natural gas, solar thermal with auxiliary natural gas heating, and solar thermal with auxiliary electricity heating. Accounting for a family’s usage over the course of a week, hot water consumption can be well above 500 gallons. This project looked at three different scenarios of hot water use: low, medium, and high. The medium demand scenario assumes that the household consumes hot water according to an diurnal pattern, rising before home owners leave for work and peaking around 7pm once home owners return. The high demand scenario doubled the medium demand and the low demand scenario halved the medium demand. In the span of a week, the occupants are assumed to use 676 gallons in the medium demand scenario, 1352 gallons in the high demand scenario, and 338 gallons in the low demand scenario.

The tank lifespan associated with each water heating option is the same (15 years); however, the solar thermal system components (not including the storage tank) are assumed to last 30 years. Pricing of water heaters is based on the American Council for an Energy Efficient Economy [1], and local installations in Wake County, North Carolina are considered for solar thermal installations. The indicated solar thermal prices include installation, solar thermal panels, piping, pumps, and other associated equipment, after taking into account the appropriate renewable energy tax credits. Section 3.3.8 provides information regarding the tax credits applied to each system. Capital investment costs are amortized to a monthly value while the solar thermal tanks are amortized separately due to their shorter lifespan. The cost equation for
a hot water system is defined as:

\[ C_{HW} = P_{HW} \left( \frac{i/12}{1 - (1 + \frac{i}{12})L_{HW} \times 12} \right) \times \bar{X}_{HW} \]  

(3.8)

<table>
<thead>
<tr>
<th>Technology</th>
<th>Lifetime</th>
<th>Cost</th>
<th>Cost after Tax Incentive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Natural Gas</td>
<td>15</td>
<td>$850</td>
<td>—</td>
</tr>
<tr>
<td>Electric</td>
<td>15</td>
<td>$750</td>
<td>—</td>
</tr>
<tr>
<td>Solar Thermal with NG</td>
<td>30</td>
<td>$4850</td>
<td>$1995</td>
</tr>
<tr>
<td>Solar Thermal with Electric</td>
<td>30</td>
<td>$4750</td>
<td>$1875</td>
</tr>
</tbody>
</table>

Table 3.3: Water Heating Options

Efficiencies also varied by water heating technology. The conventional electric tank unit has an assumed energy efficiency of 90% and the natural gas model has an efficiency of 60% [1]. This means that for electric models, 90% of the heat produced from the elements is transferred to the water whereas only 60% of the heat from the combustion of natural gas is transferred to the tank water. The solar thermal water heating system used did not have a specified energy efficiency. The solar collector has a tested efficiency of 70.6%, which represents the percentage of the sun’s energy absorbed by the collector. Round trip thermal efficiency is generally lower due to losses associated with heat loss from the collector, heat transfer to the working fluid, and heat loss in the pipes. A description of the heat loss from the hot water tank is described in Section 2.2.3.

3.3.7 Heating and Cooling Technologies

An important part of the energy optimization model involves space heating and cooling. Heating, ventilation, and air conditioning (HVAC) systems account for approximately 56% of total energy consumed within homes. Most of the information used for pricing, lifespan, unit capacity, and efficiency is drawn from the US EPA MARKet ALlocation (MARKAL) National Model Database
MARKAL is an energy system optimization model that accounts for fuel supplies, energy technologies, demand technologies, and various end use demands. The residential portion of the US EPA MARKAL database contains cost and performance characteristics for heating and cooling technologies. It is important to note that geothermal prices are adjusted after consulting with local installers in Raleigh.

Heat gain and heat loss are calculated by comparing the difference between the inside thermostat temperature (assumed constant) and the ambient outside temperature. The equations are modifications of the basic heat transfer equation: $Q = UA\Delta T$ [6]

**Heat Loss:**

$$Q_{l,t} = \sum_S \sum_I A_S(T_{in} - T_{amb,t}) \bar{X}_{I,S}$$ (3.9)

**Heat Gain:**

$$Q_{g,t} = \sum_S \sum_I A_S(T_{amb,t} - T_{in}) \bar{X}_{I,S}$$ (3.10)

The above equations find the heat transfer across all surfaces of the home for different insulation types using the surface areas, difference in temperatures, and insulation values.

The insulation R-value selected for each surface is a key determinant of the heat gain and loss. The heat gains and losses determine the heating and cooling demand, which in turn affects the selection of heating and cooling technologies for the home. As a result, the coupling of insulation values and HVAC technologies makes the model non-linear. As discussed below, the model is made linear by fixing the HVAC technology, performing four different model runs (one for each HVAC technology), then selecting the model run with the lowest CO$_2$ emissions as the optimal result.
Table 3.4: Space Heating and Cooling Specifications [31]

<table>
<thead>
<tr>
<th>Heating Choices</th>
<th>Technology</th>
<th>Cost ($)</th>
<th>Coefficient of Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ground Source Heat Pump</td>
<td>10,000</td>
<td>3.3</td>
</tr>
<tr>
<td></td>
<td>Air Source Heat Pump</td>
<td>1,512</td>
<td>2.26</td>
</tr>
<tr>
<td></td>
<td>Natural Gas</td>
<td>1,080</td>
<td>0.78</td>
</tr>
<tr>
<td></td>
<td>Electric Radiant</td>
<td>1,368</td>
<td>1.0</td>
</tr>
<tr>
<td>Cooling Choices</td>
<td>Ground Source Heat Pump</td>
<td>10,000</td>
<td>14.1</td>
</tr>
<tr>
<td></td>
<td>Air Source Heat Pump</td>
<td>2,100</td>
<td>3.81</td>
</tr>
<tr>
<td></td>
<td>Air Conditioner</td>
<td>2,600</td>
<td>3.81</td>
</tr>
</tbody>
</table>

Four main HVAC configurations are included in the home model and are presented in Table 3.4: ground source heat pump, air source heat pump, natural gas furnace with electric air conditioning, and electric radiant heating with electric air conditioning. Each combined heating and cooling choice represents a binary decision. A capacity is assumed for each HVAC option and is compared with the calculated heat loss and heat gain to verify it would meet the maximum heating and cooling demands associated with the model home. Capacity for the natural gas furnace, air source heat pump, and electric radiant heating are scaled appropriately to fit the modeled home, by linearly scaling price with capacity. As a result, the decision variables are binary, but the binary choices implicitly have capacities large enough to meet demands. A constant internal thermostatic temperature is assumed to calculate heat transfer across each surface of the house (Section 2.2.5). The investment cost for heating and cooling system is defined below:

\[
C_{SH,SC} = P_{SH,SC} \times (1 - TC_{SH,SC}) \left( \frac{i/12}{1 - (1 + \frac{i}{12})^{LSH,SC \times 12}} \right)
\]  \hspace{1cm} (3.11)

For applicable technologies, \(TC_{SH,SC}\) is limited to a maximum of $8,400 as mentioned in Section 3.3.8.
3.3.8 Renewable Energy and Energy Efficient Credits

The federal government and many state governments have adopted incentive programs to make renewable energy and energy saving options more economical. Federal investment tax credits for renewable and energy efficient equipment represent 30% of the total installed price, and the tax credits in North Carolina are 35% [8]. Federal and state tax credits can be combined for a total savings of 65%. Unlike the federal tax incentives, the state of North Carolina has a limit on the maximum incentive. At the time of writing this, the limits are:

- A maximum of $1,400 per installation for solar water-heating systems, including solar pool-heating systems used for a non-business purpose
- A maximum of $8,400 for geothermal heat pumps and geothermal equipment that use geothermal energy for water heating or active space heating or cooling used for a non-business purpose
- A maximum of $10,500 per installation for photovoltaic systems and wind-energy systems used for a non-business purpose [8]

The current federal and North Carolina state tax incentives are incorporated to reflect current pricing of applicable systems within the model.

3.4 Implementation of Model

The original modeling environment used during the early stages of model development is MATLAB; however, as the model grew in complexity, constraint formulations based on matrix specification became tedious. As a result, the model formulation was ported to the General Algebraic Modeling System (GAMS), a high-level modeling language specifically designed for mathematical programming and optimization [12]. GAMS allows users to formulate optimization models using algebraic expressions, and unlike MATLAB, the constraint matrix is built automatically at runtime. Data inputs are simply expressed in tabular form, lists, or imported
from separate data files. Each GAMS model is based on the specification of sets, parameters, scalars, tables, variables, and equations. In addition to model development, GAMS includes a suite of integrated high-performance solvers that can handle linear, mixed integer, mixed integer non-linear, and other non-linear model formulations. Licensing restrictions only allowed for linear and mixed integer formulations, which informed model development.

When the model is executed, it first reads in all the formatted time series data along with model parameters. As described in Section 3.3.7, the original problem is nonlinear because of the coupled decisions related to insulation and heating and cooling technologies. This non-linearity is side-stepped by reorganizing the model structure to create a linear formulation. To create a linear formulation, the choice of HVAC technology is fixed for each model run, and the model is iterated four times, once for each HVAC option. The HVAC option that yields the lowest CO₂ emissions across all four runs is selected as optimal. To perform the optimization, GAMS calls the CPLEX linear solver to find the optimal solution. The solution set is written to a .csv file.

Each model scenario with a chosen HVAC technology is iterated through a set of increasing budget values. Each iteration finds the optimal solution for the given budget values, starting just above the solution for the unconstrained least cost budget and ending when the budget constraint is non-binding minimum CO₂ emissions are achieved. In an effort to expedite the run process, a short script is used to execute multiple runs across different nodes on a compute cluster for the different HVAC technologies.

Once all runs have been completed, the .csv files are analyzed to find the optimal solution at each budget value across the four HVAC options. The optimal values are compiled to create a trade-off curve that shows the balance between CO₂ emissions and budget values.

### 3.5 Modeling to Generate Alternatives

Models should be designed as tools to deliver insight during the decision making process. Optimization models provide the decision-maker with the best solution or set of non-inferior solutions to the modeled problem. In most cases; however, mathematical models incompletely
represent the system being modeled, especially when the system is complex or ill-defined. As a result, the best solution(s) may likely be found within the near-optimal, inferior region of the decision space rather than along the non-inferior frontier [3].

To systematically explore the decision space around the optimal solution, a method known as modeling to generate alternatives (MGA) is implemented. The Hop-Skip-Jump (HSJ) MGA algorithm finds a set of maximally different solutions that are near optimal using the optimal objective function value as an anchor point [4]. The HSJ MGA procedure is described below:

Minimize \( p = \sum_{k \in K} x_k \) \hspace{1cm} (3.12)

\[ \text{s.t. } f_j(\bar{x}) \leq T_j \quad \forall j \] \hspace{1cm} (3.13)

\[ \bar{x} \in X \] \hspace{1cm} (3.14)

where:

\( K \) = set of indices of the decision variables that are nonzero in the previous solutions,

\( f_j(\bar{x}) \) = \( j \)th objective function,

\( T_j \) = target specified for the \( j \)th modeled objective,

\( X \) = set of feasible solutions based on the “technical” constraints of the model.

The steps associated with the implementation of HSJ MGA are given below:

1. Obtain an initial optimization solution by any method. The original linear optimization model described in Section 3.2 is solved using the CPLEX solver via GAMS.

2. The original objective function becomes a constraint with an upper bound set by the optimal objective function value plus a specified amount of slack. Constraints from the original model formulation also apply to the MGA version of the model.

3. A new objective function formulation minimizes the sum of all non-zero decision variables
in the original, optimal solution. The new model finds a set of decision variables within the feasible, near-optimal region that are maximally different from the original solution set.

4. Subsequent MGA iterations minimize the sum of non-zero decision variables from all previous model runs.

When applied to the home energy model, a slight modification is made to the HSJ MGA procedure described above. The original objective function is encoded as a constraint in the MGA formulation with no slack: the CO$_2$ emissions value determined by the initial run is used directly as a CO$_2$ upper bound in each MGA run. However, slack is added to the budget constraint. As a result, the model can use slack in the budget constraint to find alternative solutions, but the original CO$_2$ limit must be met.

The MGA analysis is carried out with the baseline scenario (72°F temperature, medium demand) for three separate points along the non-inferior frontier, which outlines the optimal trade-off between minimizing CO$_2$ emissions with a low monthly budget. The three points represent minimum cost, near minimum CO$_2$ and an intermediate point between the two. The middle budget value is chosen by taking the difference between the minimum cost and the cost at minimum CO$_2$ emissions, then minimizing for CO$_2$ at a budget value represented by minimum cost plus half the difference. Each set of MGA runs is composed of the initial minimization of CO$_2$ emissions at the specified budget constraint followed by four separate MGA runs.

Slack is applied to the budget constraint during all MGA runs, allowing for the possibility of other decision variables to enter the solution set. The prescribed slack defines the boundaries for a local search within the inferior set close to the initial optimal solution. Invoking an objective function that minimizes the previously non-zero decision variables pushes the model to find maximally different solutions. Two slack amounts (10% and 20% of the original objective function value) are chosen. These values are not based on empirical estimates of how much homebuyers would be willing to relax their specified monthly energy budgets, but rather are selected to provide a reasonable amount of room to explore the decision space.
4.1 Scenarios

The 72°F indoor temperature, medium demand scenario is chosen as the base case scenario. Subsequent runs at indoor temperatures of 62°F and 82°F are completed with different demand profiles to test the sensitivity of the results to different input assumptions.

4.1.1 Base Case

The model runs trace a Pareto frontier by minimizing CO$_2$ emissions at varying monthly budget constraints. Selection patterns emerge by demand category (i.e., water heating, insulation, space heating/cooling, and renewable electricity generation). Pareto curves (Figure 4.1) are generated for the 72°F indoor temperature scenario under three different home electric demand and three different hot water demand scenarios.

The base case run, in which the monthly energy budget is minimized with no constraint on CO$_2$ emissions, has a monthly cost of $164 and CO$_2$ emissions of 8663kg. As the budgetary
Table 4.1: Cost and CO₂ Emissions Values for 72°F Scenarios

<table>
<thead>
<tr>
<th>Demand Scenario</th>
<th>Min Cost ($)</th>
<th>Min CO₂ (kg)</th>
<th>Zero CO₂ ($)</th>
<th>Zero CO₂ (kg)</th>
<th>Min CO₂ ($)</th>
<th>Min CO₂ (kg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>72 Low</td>
<td>119</td>
<td>5,323</td>
<td>178</td>
<td>0</td>
<td>436</td>
<td>-4,601</td>
</tr>
<tr>
<td>72 Medium</td>
<td>164</td>
<td>8,663</td>
<td>354</td>
<td>0</td>
<td>481</td>
<td>-1,264</td>
</tr>
<tr>
<td>72 High</td>
<td>253</td>
<td>15,342</td>
<td>–</td>
<td>–</td>
<td>570</td>
<td>5,415</td>
</tr>
</tbody>
</table>

Constraint is relaxed to $169 per month and CO₂ emissions are minimized, natural gas space heating became the preferred option over electrical heating (Figure 4.1). The water heating options followed a similar pattern: the preferred choice at minimum cost is natural gas, but changed to solar thermal heating with natural gas as the auxiliary heating choice when the budget reached $179 per month. The insulation R-values for walls, floor, and ceiling remained high at all monthly budget constraints. The window selections exhibited more variation due to larger differences in cost between different R-values. At low budget constraints (between $164 and $198 per month), the model chose the low R-value option, changed to the medium option above $198 per month, then switched to the high R-value option only at minimum CO₂ emissions when the budget constraint is non-binding ($481 per month). The minimum CO₂ emissions for medium demand is found to be -1,264 kg. The model is able to achieve negative CO₂ emissions by building solar PV and wind capacities that generate more electricity than is consumed by the household. This electricity is sold back to offset production of electricity on the grid. Renewable energy systems entered at $179 per month and remain a persistent feature of the solution at monthly energy budgets greater than $180. Interestingly, solar PV is selected at monthly budget constraints of $179; lower than the medium R-value windows. This selection remains true even when incentives are removed. In addition, the upper bound on solar PV capacity became binding before the wind capacity entered the solution. The base case scenario reached zero net emissions at a monthly cost of $353, which is accomplished by utilizing high R-value insulation for all surfaces but the windows (medium R-value), solar thermal water heating, the maximum 7 kW solar capacity, and nearly 4.2 kW of installed wind capacity.
Changing the demand levels had no discernible effect on the technology selections made at a given budget constraint. As discussed in Section 3.3.6, the high demand includes a doubling of hot water usage and household electricity relative to the medium demand scenario. The low demand scenario reduces both the electricity and hot water consumption by half relative to the medium demand scenario. Changes in technology selection as a function of budget constraint under both the high and low demand scenarios follow the same order as in the base case (values listed below are for the low demand scenario):

**Space Heating and Cooling:** The electric heat pump is selected as the heating option at the lowest budget constraint of $120 per month, but the natural gas furnace is preferred at a budget of $125 per month and greater.

**Water Heating:** The natural gas water heater is selected at the lowest budget, but is replaced by a solar thermal system with a natural gas auxiliary unit at a budget value of $135 per month.

**Insulation:** High R-value insulation, with the exception of windows, is selected for all surfaces at all budget constraints. The model selected low R-value windows at low budgets of $120 – $154 per month, but switches to medium R-value windows for budgets ranging from $173 – $420 per month. Not surprisingly, high R-value insulation, including windows, is selected to achieve minimum CO$_2$ emissions with no budget constraint.

**Electricity Generation:** Renewable energy systems act in the same manner as with medium demand: solar PV is installed at a budget constraint $\sim$ $135 per month, the maximum solar PV capacity is reached at a budget constraint $\sim$ $249 per month, and installation of wind capacity begins at higher budgets.
Figure 4.1: The Pareto frontier with the optimal technology selections for each of the three 72°F optimization runs at high demand (HD), medium demand (MD), and low demand (LD). The radar plots indicate which insulation options are selected within regions outlined by the dotted lines. These are ordered (from inner to outer) as the insulation R-values progress from low to high. The legend describes which heating and cooling choices are implemented for each budget constraint (Elec – Electric Heat and NG – Natural Gas Heat). The red dotted line shows the point where water heating technologies change from natural gas to solar thermal. The high insulation values for walls, floor, and ceiling –but not windows– are present from the lowest budget constraint.
As expected, the monthly energy budget varies with the demand level. The minimum monthly energy budget in the low demand scenario is $120, while the high demand scenario has a minimum monthly energy budget of $253. When the model is applied to minimize cost under the low demand scenarios, CO₂ emissions decrease by almost 40% relative to the equivalent medium demand scenario. Likewise, under the high demand scenario, CO₂ emissions increased by 77% relative to the equivalent medium demand scenario.

4.1.2 Sensitivity Analysis

Sensitivity analysis is conducted on the homes internal (thermostatic) temperature to determine the effect on technology selection. The temperature varies 10°F above and below the base case. Electricity demand and hot water consumption are increased and decreased as described in Section 4.1.1.

The low demand, low temperature (62°F) scheme achieves the lowest cost ($114/month) and lowest CO₂ emissions (4,909kg/year) of all cost minimization runs (See Table 4.1.2). The installed renewable energy capacities at both the high and low indoor temperatures respond to increases in the budget constraint in a manner very similar to the 72°F scenario. Solar PV capacity is installed at a monthly budget constraint of $127 per month until reaching the upper bound on installed capacity at $241 per month, followed by the installation of wind power. As in the base case, the window insulation R-value switches to the highest value only when CO₂ emissions are minimized with no budget constraint. Like previous runs, the natural gas water heater is the lowest cost option across each of the six runs, but is replaced with the solar thermal water heating system at a monthly budget constraint of $127 per month.

As seen in Table 4.1.2, the 82°F scenario follows the same pattern of heating and cooling technology choices. As in all the 72°F scenarios, the air source heat pump is the least expensive choice when the objective function is to minimize cost with no constraint on CO₂ emissions (Figure 4.1.2). The natural gas furnace becomes the preferred choice at a maximum budget of $162 per month in the medium demand scenario when the objective is to minimize CO₂ emissions.
Table 4.2: Cost and CO₂ Emissions Values for 62°F and 82°F Scenarios

<table>
<thead>
<tr>
<th>Demand Scenario</th>
<th>Min Cost</th>
<th>Zero CO₂</th>
<th>Min CO₂</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cost($)</td>
<td>CO₂ (kg)</td>
<td>Cost($)</td>
</tr>
<tr>
<td>62 Low</td>
<td>114</td>
<td>4,909</td>
<td>168</td>
</tr>
<tr>
<td>62 Medium</td>
<td>158</td>
<td>8,249</td>
<td>340</td>
</tr>
<tr>
<td>62 High</td>
<td>247</td>
<td>14,929</td>
<td>–</td>
</tr>
<tr>
<td>82 Low</td>
<td>131</td>
<td>6,287</td>
<td>195</td>
</tr>
<tr>
<td>82 Medium</td>
<td>175</td>
<td>9,627</td>
<td>388</td>
</tr>
<tr>
<td>82 High</td>
<td>264</td>
<td>16,307</td>
<td>–</td>
</tr>
</tbody>
</table>

emissions. Insulation R-values in the 82°F and 72°F scenarios are also congruent. The highest floor, wall, and ceiling R-values are selected across every run while the lowest level of window insulation is selected when the objective is to minimize monthly energy costs. The medium R-value window is selected when the objective is switched to CO₂ minimization with a budget constraint ($193 per month for medium demand scenario), and the high R-value window is selected only in the model run without a budget constraint ($489 per month for medium demand scenario).
Figure 4.2: The Pareto frontier with the optimal technology selections for each of the three 82°F optimization runs at high demand (HD), medium demand (MD), and low demand (LD). The radar plots indicate which insulation choices are selected with respect to the dotted lines. These are ordered (from inner to outer) as the insulation R-values vary from low to high. The legend describes which heating and cooling choices are implemented at each budget constraint (Elec – Electric Heat and NG – Natural Gas Heat). The red dotted line shows where water heating technologies change from natural gas to solar thermal with auxiliary natural gas. The switch to solar water heating corresponds closely with the change from low to medium window R-values.
The technology choices underlying the Pareto frontier associated with the 62°F scenario exhibit more variability. The demand scenarios at this low thermostatic temperature see the only emergence of the geothermal heat pump for air heating and cooling. In the medium demand scenario, the geothermal choice is made at a budget constraint of $471 per month, at the same point the upper bound constraint on installed wind capacity becomes binding. At the lowest budget constraint in the medium demand of ($158 per month), the air source heat pump is the optimal choice, but is quickly replaced by the natural gas furnace at a monthly budget of $162. As in previous model scenarios, hot water demand is met by the natural gas water heater at the lowest budget constraint but is replaced by the solar thermal unit with auxiliary natural gas heating at a budget constraint of roughly $172 per month. Ceiling insulation responded differently in the 62°F scenario compared to the 72°F and 82°F scenarios. In the 62°F scenario, the ceiling R-value is low during at low budget constraints but then jumps to high levels for subsequent runs (Figure 4.3). The selected window R-value, as in the other temperature and demand runs, initially started as low; but unlike other runs remained at low R-values until a budget constraint of approximately $286 per month, where it increased to the medium R-value. At a budget constraint of $286 per month, installed solar capacity reached the maximum capacity of 7kW. More detailed results regarding each individual run can be found in Appendix 7.
Figure 4.3: The Pareto frontier with the optimal technology selections for each of the three 62°F optimization runs at high demand (HD), medium demand (MD), and low demand (LD). The radar plots indicate which insulation options are selected with respect to the dotted lines. These are ordered (from inner to outer) as the insulation R-values progress from low to high. The legend describes which heating and cooling choices are implemented at each budget constraint (Elec – Electric Heat, NG – Natural Gas Heat, and Geo – Geothermal Heating and Cooling). The red dotted line shows the point where water heating technologies change. Note that the low R-value insulation is present at the lowest budget values. The low ceiling insulation option is replaced at the same time as the natural gas water heating system changes to solar thermal with natural gas auxiliary ($172 for the medium demand scenario).
4.2 MGA Analysis

Implementation of HSJ MGA at three different monthly budget constraints along the non-inferior frontier provides clusters of points that have similar monthly energy budgets and CO$_2$ emissions, but are maximally different in decision space. The persistence of a particular decision variable across multiple model runs suggests the choice is robust within the defined space. Some decision variables have a greater tendency to change with every run. In this case, the selection of a particular technology within a given demand category is flexible. The application of MGA can also rule out certain technologies such as radiant electric space heating, which is not a feasible choice for any combination of budget and slack.

The lower budget values (e.g., $164 per month) display a high tendency for the decision variables within the MGA solutions to change. This indicates flexibility in technology choice at low monthly budgets. MGA decisions are less likely to be bound by CO$_2$ emissions, as observed from the numerous solutions that fall within the bounds created by the slack parameter value. As evident in Figure 4.4, the monthly budget constraint with 10% added slack is not always binding. In addition, selections made with respect to the 20% slack are not limited to the space between the boundaries of 10% and 20% slack. At a 10% slack, geothermal heating is likely to be coupled with a low R-value insulation for all surfaces. Natural gas and solar thermal water heaters remain the key water heating choices, but neither option dominates.

At medium ($308) and high ($400) budgets, the solutions are limited by the CO$_2$ emissions constraint. In all the MGA solutions, the CO$_2$ emissions constraint is binding, but the budget constraint, with slack, is not (Figure 4.5). The electric heat pump technology is infeasible at 10% slack for medium and high budgets but becomes viable at 20% slack. Unlike the MGA solutions at the low monthly energy budgets, solutions for the 20% slack scenario associated with medium and high budgets exist only in the region between 10% slack and 20% slack constraint. Detailed information regarding the individual runs is found in Appendix 8.

No insulation value is dominant across the MGA solutions. Insulation R-values change frequently since insulation is relatively inexpensive when calculated as a monthly amortized cost
across the lifetime of the home. Consistencies in insulation choices are revealed by looking more closely at specific HVAC options within a specified slack value. The closer an HVAC technology pushes cost to the budget constraint, the more likely the home will have an insulation profile corresponding to the original optimal solution’s insulation values. At high budgets with CO$_2$ minimization as the objective, many low carbon emissions technologies are selected. Once the budget constraint is relaxed, technologies leading to high carbon emissions –such as low R-value insulation– are more likely to be selected and therefore the constraint on CO$_2$ emissions is binding.
Figure 4.4: MGA solutions associated with the baseline scenario (72°F indoor temperature, medium demand) at a minimum energy budget of $164/month. Slack values of 10% and 20% on the budget allow the model additional room to explore the decision space (designated by the red lines). The legend describes the original budget ($164), the heating and cooling technology (NG – Natural Gas, Geo – Geothermal Heat Pump, and E pump – Electric Heat Pump) and the amount of slack permitted (10% or 20%). Note that in most cases, neither the budget nor the CO₂ emissions constraint is binding.
Figure 4.5: MGA solutions associated with the baseline scenario (72°F indoor temperature, medium demand) at two energy budgets: medium ($308) and high ($400). Slack values of 10% and 20% on the budget allow the model to explore the near optimal decision space (designated by the red lines). The legend lists the original budgets, the heating and cooling technology (NG – Natural Gas, Geo – Geothermal Heat Pump, and E pump – Electric Heat Pump) and the amount of slack permitted (10% or 20%). Unlike Figure 4.4, the constraints on CO₂ emissions are binding for all MGA iterations.
This thesis provides useful insight for a home buyer or contractor who wishes to build a house that minimizes energy-related CO\textsubscript{2} emissions subject to a specified monthly energy budget. There is strong potential to reduce CO\textsubscript{2} emissions by implementing the technology options available to the model. Unlike previous modeling efforts, the optimization model developed in this project accounts for sub-hourly variation in electricity production from a wind turbine and solar PV system, home electricity demand, and solar thermal water heating using recorded time series data from the NC Solar House. This time series data is paired with estimated hourly CO\textsubscript{2} emissions coefficients from grid-based electricity, which quantify how residential renewable electricity generation is likely to displace CO\textsubscript{2} emissions from the electricity grid.

The model runs suggest that cooler indoor temperatures are more cost-effective than higher indoor temperatures, though this is likely due to the disproportionate number of hot summer days missing from the recorded data\textsuperscript{1}. In any case, the scale of total demand plays a much

\textsuperscript{1}While this might seem counterintuitive given the hot summers of North Carolina, the reason could be explained by the average temperature for Raleigh across the entire time series [36]. The average daily temperature for Raleigh across the days within the time series is 65\degree F while the collected data shows an average daily temperature of 57\degree F. The temperature discrepancy means warmer days are removed or missing from the data, leading to a
larger role in cost savings. Not surprisingly, reducing demand for both hot water and electricity consumption significantly reduces monthly energy-related expenditures. The highest window R-value is selected just before the minimum emissions case across every run. In addition, none of the high demand scenarios could reach a zero net emissions household even if the most aggressive CO$_2$ reduction measures are taken.

Therefore, regardless of the technology configuration, energy conservation via reduced demand is an effective way to reduce CO$_2$ emissions. The findings show that in order to reduce CO$_2$ emissions further, a homeowner should invest in energy efficiency before renewable energy options. However, optimal choices depend on the circumstances since not all efficient upgrades are equally cost-effective. The least cost option is achieved through natural gas to meet both water and space heating demands and the highest R-values for the home surfaces, excluding windows. Minimizing household CO$_2$ emissions at different budget constraints leads to the following insights:

**Insulation:** The highest R-value insulation is the best option for all surfaces but windows\(^2\).

**Space Heating and Cooling:** The natural gas furnace coupled with electric air conditioning is the preferred option across most budgets except at extreme CO$_2$ emissions reductions or budget reductions.

**Water Heating:** The solar thermal system with a natural gas auxiliary unit is the optimal choice unless trying to minimize budget expenditures.

**Electricity Generation:** Renewable electricity systems are installed only after all of the above measures are enacted.

The model selects solar PV at low monthly energy budgets and the upper bound constraint on installed solar PV capacity (dictated by the roof area facing south) becomes binding before

---

\(^2\)The model generally selected low R-values for windows due to their higher cost. Higher cost windows typically provide a better insulation value but also reduce the amount of air penetration that occurs. This model did not account for tightness of windows seals. Quantifying the amount of air penetration from specific windows and relating that to energy consumption is beyond the scope of this project.
wind energy enters the solution\textsuperscript{3}. The solar thermal water heater with natural gas auxiliary heating is selected over the natural gas system at low monthly budgets across every scenario and remains the best choice for water heating as the allowed monthly budget increases.

Sensitivity analysis of the internal home temperature and demand level for electricity and hot water confirmed that the technology selections made in the base case were robust. While the CO\textsubscript{2} emissions at a specified monthly energy budget were directly correlated with the demand level, the technology selections as a function of budget constraint remained consistent with the base case.

The MGA analysis demonstrated that several viable technology alternatives exist to keep CO\textsubscript{2} emissions constant but add additional expense to the monthly budget. In a general sense, this result indicates that if homeowners are willing to relax their monthly energy budget, additional technology alternatives are available that maintain the same CO\textsubscript{2} emissions rates. Real world considerations beyond the scope of this project may influence technology selection at the time of home construction. The MGA results indicate that home buyers have limited flexibility to select alternative home energy options with roughly the same performance if they wish to consider factors other than CO\textsubscript{2} emissions and cost. Application of HSJ MGA to three points along the non-inferior cost and emissions frontier suggests that the original solution is relatively robust, given slack parameters of 10\% and 20\% of the original budget constraint. Therefore, most technology selections made in the non-MGA run persist in the MGA runs, indicating that significant variations in technology cost make it difficult to switch options while remaining within the prescribed budget values.

It is important to note that the recommendations provided are dependent on the location. Data specific to central North Carolina is used during all runs. As a result, readers should be advised that some of the decisions may change depending on the local climate and renewable resource availability. Cities in the Midwest and parts of the Northwest will likely see wind as

\textsuperscript{3}The strong preference for solar PV over wind is due to the poor wind resource in central North Carolina. Highly concentrated wind resources in North Carolina are situated near the coast and along western mountain ridgelines.
the more viable option with solar being selected after wind turbines. The Southwest will likely be reliant solely on solar power due to consistently high solar insolation values.

Future work associated with this modeling effort could take several directions. In the short term, the results of a full year of data will likely have an effect on some of the decisions. A front end user interface could enable this model to serve as a decision support tool. In addition, the model could benefit from the implementation of more detailed homeowner demand profiles. The utilization of building energy simulation software to more accurately measure the thermal performance in different zones of the home could inform the model and increase the accuracy of the heat flow estimates. Finally, the model could be expanded to include the retrofit of preexisting homes to assist current homeowners with reducing CO$_2$ emissions from energy use.
References


Appendices
APPENDIX A

Nomenclature
### Nomenclature

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Sets</th>
<th>Subscripts</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C$</td>
<td>cost</td>
<td></td>
</tr>
<tr>
<td>$CO_2$</td>
<td>CO$_2$ emissions rate</td>
<td></td>
</tr>
<tr>
<td>$NG$</td>
<td>natural gas consumption</td>
<td></td>
</tr>
<tr>
<td>$P$</td>
<td>unit price</td>
<td></td>
</tr>
<tr>
<td>$TC$</td>
<td>tax credit fraction</td>
<td></td>
</tr>
<tr>
<td>$i$</td>
<td>annual interest</td>
<td></td>
</tr>
<tr>
<td>$F$</td>
<td>fixed O&amp;M ($/kW*yr$)</td>
<td></td>
</tr>
<tr>
<td>$V$</td>
<td>variable O&amp;M ($/kW*yr$)</td>
<td></td>
</tr>
<tr>
<td>$E$</td>
<td>electricity</td>
<td></td>
</tr>
<tr>
<td>$\eta$</td>
<td>technology efficiency</td>
<td></td>
</tr>
<tr>
<td>$B$</td>
<td>monthly budget</td>
<td></td>
</tr>
<tr>
<td>$Q$</td>
<td>heat transfer</td>
<td></td>
</tr>
<tr>
<td>$A$</td>
<td>area</td>
<td></td>
</tr>
<tr>
<td>$T$</td>
<td>temperature</td>
<td></td>
</tr>
<tr>
<td>$L$</td>
<td>lifetime of technology</td>
<td></td>
</tr>
<tr>
<td>$\bar{X}$</td>
<td>installed technology</td>
<td></td>
</tr>
</tbody>
</table>

| Decision Variables | | |
|-------------------| | |
| $\bar{X}$         | installed technology | |
APPENDIX B

Individual Run Results

B.1 Results from Optimization Runs
Figure B.1: Results from 62°F low demand optimization run. The colors are used to highlight the decisions chosen at different budget constraints. With respect to water heating technologies, natural gas (NG) and solar thermal with auxiliary natural gas (SNG) are used. Insulation R-values can take on one of three different values per surface: low (L), medium (M), and high (H). The HVAC systems include an air source heat pump ($E_{pump}$), natural gas with electric air conditioning (NG), and geothermal heating and cooling (GEO).
Figure B.2: Results from 62°F medium demand optimization run. The colors are used to highlight the decisions chosen at different budget constraints. With respect to water heating technologies, natural gas (NG) and solar thermal with auxiliary natural gas (SNG) are used. Insulation R-values can take on one of three different values per surface: low (L), medium (M), and high (H). The HVAC systems include an air source heat pump ($E_{pump}$), natural gas with electric air conditioning (NG), and geothermal heating and cooling (GEO).
Figure B.3: Results from 62°F high demand optimization run. The colors are used to highlight the decisions chosen at different budget constraints. With respect to water heating technologies, natural gas (NG) and solar thermal with auxiliary natural gas (SNG) are used. Insulation R-values can take on one of three different values per surface: low (L), medium (M), and high (H). The HVAC systems include an air source heat pump (E\textsubscript{pump}), natural gas with electric air conditioning (NG), and geothermal heating and cooling (GEO).

| Overall Monthly Cost | 114 | 251 | 261 | 280 | 299 | 318 | 337 | 356 | 375 | 394 | 413 | 432 | 451 | 470 | 489 | 508 | 527 | 546 | 560 | 577 |
|---------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| CO2 Emissions       | 14929 | 14393 | 13280 | 11428 | 10137 | 9395 | 8653 | 7911 | 7362 | 7123 | 6884 | 6645 | 6405 | 6166 | 5927 | 5750 | 5586 | 5422 | 5326 | 5259 |
| Solar Capacity      | 0 | 0 | 0.93 | 2.95 | 4.36 | 5.17 | 5.97 | 6.78 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 |
| Wind Capacity       | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.29 | 1.46 | 2.63 | 3.8 | 4.97 | 6.14 | 7.31 | 8.17 | 8.97 | 9.78 | 9.93 | 10 |
| Hot Water Choice    | NG | NG | SNG | SNG | SNG | SNG | SNG | SNG | SNG | SNG | SNG | SNG | SNG | SNG | SNG | SNG | SNG | SNG | SNG | SNG | SNG |
| Wall Insulation     | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H |
| Floor Insulation    | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H |
| Window Insulation   | L | L | L | L | L | L | L | M | M | M | M | M | M | M | M | M | M | M | M | M | M | H |
| Ceiling Insulation  | L | L | L | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H |
| HVAC                | E\textsubscript{pump} | NG | NG | NG | NG | NG | NG | NG | NG | NG | NG | NG | NG | NG | NG | NG | Geo | Geo | Geo | Geo | Geo | Geo |
Figure B.4: Results from 72°F low demand optimization run. The colors are used to highlight the decisions chosen at different budget constraints. With respect to water heating technologies, natural gas (NG) and solar thermal with auxiliary natural gas (SNG) are used. Insulation R-values can take on one of three different values per surface: low (L), medium (M), and high (H). The HVAC systems include an air source heat pump ($E_{pump}$), natural gas with electric air conditioning (NG), and geothermal heating and cooling (GEO).
Figure B.5: Results from 72°F medium demand optimization run. The colors are used to highlight the decisions chosen at different budget constraints. With respect to water heating technologies, natural gas (NG) and solar thermal with auxiliary natural gas (SNG) are used. Insulation R-values can take on one of three different values per surface: low (L), medium (M), and high (H). The HVAC systems include an air source heat pump (E\textsubscript{pump}), natural gas with electric air conditioning (NG), and geothermal heating and cooling (GEO).
Figure B.6: Results from 72°F high demand optimization run. The colors are used to highlight the decisions chosen at different budget constraints. With respect to water heating technologies, natural gas (NG) and solar thermal with auxiliary natural gas (SNG) are used. Insulation R-values can take on one of three different values per surface: low (L), medium (M), and high (H). The HVAC systems include an air source heat pump ($E_{pump}$), natural gas with electric air conditioning (NG), and geothermal heating and cooling (GEO).
Figure B.7: Results from 82°F low demand optimization run. The colors are used to highlight the decisions chosen at different budget constraints. With respect to water heating technologies, natural gas (NG) and solar thermal with auxiliary natural gas (SNG) are used. Insulation R-values can take on one of three different values per surface: low (L), medium (M), and high (H). The HVAC systems include an air source heat pump ($E_{pump}$), natural gas with electric air conditioning (NG), and geothermal heating and cooling (GEO).
Figure B.8: Results from 82°F medium demand optimization run. The colors are used to highlight the decisions chosen at different budget constraints. With respect to water heating technologies, natural gas (NG) and solar thermal with auxiliary natural gas (SNG) are used. Insulation R-values can take on one of three different values per surface: low (L), medium (M), and high (H). The HVAC systems include an air source heat pump ($E_{\text{pump}}$), natural gas with electric air conditioning (NG), and geothermal heating and cooling (GEO).
Figure B.9: Results from 82°F high demand optimization run. The colors are used to highlight the decisions chosen at different budget constraints. With respect to water heating technologies, natural gas (NG) and solar thermal with auxiliary natural gas (SNG) are used. Insulation R-values can take on one of three different values per surface: low (L), medium (M), and high (H). The HVAC systems include an air source heat pump ($E_{pump}$), natural gas with electric air conditioning (NG), and geothermal heating and cooling (GEO).
APPENDIX C

MGA Runs

C.1 Results from MGA Runs
Figure C.1: Optimal technology selections for the low budget scenario ($164) associated with the 72°F MGA run. Results show the monthly cost, indicates CO$_2$ emissions, installed solar PV and wind capacities, water heating selection, and levels of insulation. The heading above the results which budget constraint, HVAC technology, and slack value were used for the run. The colors for insulation represent the value of insulation selected for each run: red is high, yellow is medium, and blue is low insulation. With respect to water heating technologies, natural gas (NG) and solar thermal with auxiliary natural gas (SNG) are used. Insulation R-values can take on one of three different values per surface: low (L), medium (M), and high (H). The HVAC systems include an air source heat pump ($E_{\text{pump}}$), natural gas with electric air conditioning (NG), and geothermal heating and cooling (GEO).
Figure C.2: Optimal technology selections for the medium budget scenario ($308) associated with the 72°F MGA run. Results show the monthly cost, indicates CO₂ emissions, installed solar PV and wind capacities, water heating selection, and levels of insulation. The heading above the results which budget constraint, HVAC technology, and slack value were used for the run. The colors for insulation represent the value of insulation selected for each run: red is high, yellow is medium, and blue is low insulation. With respect to water heating technologies, natural gas (NG) and solar thermal with auxiliary natural gas (SNG) are used. Insulation R-values can take on one of three different values per surface: low (L), medium (M), and high (H). The HVAC systems include an air source heat pump (E_pump), natural gas with electric air conditioning (NG), and geothermal heating and cooling (GEO).
Figure C.3: Optimal technology selection the high budget scenario ($400) associated with the 72°F MGA run. Results show the monthly cost, indicates CO$_2$ emissions, installed solar PV and wind capacities, water heating selection, and levels of insulation. The heading above the results which budget constraint, HVAC technology, and slack value were used for the run. The colors for insulation represent the value of insulation selected for each run: red is high, yellow is medium, and blue is low insulation. With respect to water heating technologies, natural gas (NG) and solar thermal with auxiliary natural gas (SNG) are used. Insulation R-values can take on one of three different values per surface: low (L), medium (M), and high (H). The HVAC systems include an air source heat pump (E$_{pump}$), natural gas with electric air conditioning (NG), and geothermal heating and cooling (GEO).
D.1 MATLAB Code

The following code was written to take in all the GRID$_C$ data and smooth out and remove all the missing data. Solar thermal performance simulation took place within the MATLAB script.

```matlab
clear all

format long g
% Algorithm pulls info from separate xls files, finds all rows with the
% same date stamp and puts info into a new matrix.
%
% ——All dates start on may 1 2009, everything was working at this point!
```
% ——Files recent up to 8–17–10 at 11:45PM with 15 min. intervals
%
%
Problem arises when using hourly data b/c when taking the
% difference

start = 733894; %serial date associated w/ may 1, 2009
finish = 734368; %serial date associated w/ August 18, 2010

[num, time1] = xlsread('annex_wind_solar.xls'); % Imports dates
and values from "annex_wind_solar.xls"
% contains PV energy, Wind energy, solar irradiance,
module temperature and ambient temperature

time_se = time1((2:length(time1)),1); % isolate date and
time1 column into separate matrix
num_1 = [num(1,(1:2));num(:,(1:2))]; %num_1 contains
columns that must be differed (they are solar and wind energy
produced)
num = num(:,(3:end)); %the remainder of
the columns are placed into a separate matrix until num_1 is
differenced

% CAREFUL HERE — use only for incremental data
!!!!!!!!!!!!!!!!!!!!!!!!!!!!

num_1 = diff(num_1);
for i = 1:length(num_1) %convert all negative
    information to zeros
    for j = 1:2
if num_1(i,j) < 0
    num_1(i,j) = 0; num_1((i+1),j) = 0; num_1((i-1),j) = 0;
end
end
end

for i = 1:length(num_1)  %convert all very large data
to zeros
    for j = 1:2  %nothing in column 1 or 2
        if num_1(i,j) > 0.909
            num_1(i,j) = 0; num_1((i+1),j) = 0; num_1((i-1),j) = 0;
        end
    end
end

date_conv = datenum(time_se);  %change date and time to
    serial number for comparisons
date_conv = (floor(date_conv * 10000000))/10000000;
sd_annex = [date_conv num_1 num];
gone = diff(sd_annex(:,1));  %removing the rows that have
duplicate time stamps (only 2)
gone = find(gone ==0);
for i = 1:size(gone)
    sd_annex(gone(i),:) = [];
end
dlmwrite('annex_wind_solar.txt', sd_annex, 'precision', '%14.7f');
%write text file of all sd_annex contents for quicker running

[num, time2] = xlsread('house_data');    % Imports dates and values from house (only value is Home Demand)
time_se = time2((2:length(time2)),1);    % isolate date and time2 column into separate matrix
num = [num(1,:);num];
num = diff(num);
for i = 1:length(num)           %convert all negative information to zeros
    if num(i) < 0
        num(i) = 0; num(i+1) = 0; num(i-1) = 0;
    end
end
date_conv = datenum(time_se);        %change date and time to serial number for comparisons
date_conv = (floor(date_conv * 10000000))/10000000;
sd_house = [date_conv num];          %SD_HOUSE is everything in house matrix converted to number format
gone = diff(sd_house(:,1));          %removing the rows that have duplicate time stamps (only 2)
gone = find(gone ==0);
for i = 1:size(gone)
    sd_house(gone(i),:) = [];
end
dlmwrite('house_data.txt', sd_house, 'precision', '%14.7f');  
% write text file of all sd_annex contents for quicker running

clear gone;

sd_annex = dlmread('annex_windsolar.txt', ',');
sd_house = dlmread('house_data.txt', ',');

a_column = size(sd_annex, 2);  
% number of columns for annex

h_column = size(sd_house, 2);  
% number of columns for house

sd_complete = start:(1/96):finish;

sd_complete = (floor(sd_complete * 10000000))/10000000;

master_mat = zeros(length(sd_complete), (h_column+a_column-1));  
% combining columns

master_mat(:, 1) = sd_complete;

% the section of code below finds the points missing in each set
of data and removes the points from the more complete other data set

for i=1:length(sd_complete)
    a=find(sd_complete(i)==sd_annex(:, 1));
b=find(sd_complete(i)==sd_house(:,1));

if isempty(a+b)
    master_mat(i,:)=0;
else
    master_mat(i,2:a_column) = sd_annex(a,2:a_column);
    master_mat(i,(a_column+1:a_column+h_column-1)) = sd_house(b,2:h_column);
end
end

zero_mat = flipud(find(master_mat(:,1)==0)); %removes all cells where time stamps are == 0
for i = 1:size(zero_mat)
    master_mat(zero_mat(i),:) = [];
end

%---------------------------------------------------------

% ----------- BEGIN FUNCTION DATES_OFF

date_conv= [master_mat(:,1); floor(master_mat(end,1))+.9895833];
num = (master_mat(:,(2:end)));
fake_matrix =date_conv;
date_diff = diff(fake_matrix);
matrix = [];
count = 0;

86
start_date = date_conv(1); %defines a starting date

for i = 1:(length(date_conv)-1) %loop is to ID and remove any full day with data missing for a period of time greater than 30min
    if date_diff(i) >= 0.04 %for gap larger than 30min
        count = count +1;
        % matrix(count,:) = [i, date_diff(i)]; %displays where gaps exist and the length of the gaps
        gap_start = date_conv(i); %IDs the start of missing string
        gap_end = gap_start + date_diff(i); %IDs the end of the gap (serial date number of last gap)
        begin_day = floor(gap_start)+0.0104166; %first day of gap (12:00:00am with first 15min added) so 12:15:00am
        [start_row, col] = find(date_conv==begin_day); %searching to find the 12:15:00am of the gap
        while isempty(start_row) == 1
            begin_day = begin_day -1; %if not found then move to the previous day
            [start_row, col] = find(date_conv == begin_day);
            %search again until a beginning day is found

87
end

end_day = (floor(gap_end)) + .9895833;

% last day of gap at 11:45:00pm
[end_row, col] = find(date_conv == end_day);

11:45:00pm of the last day of the gap

while isempty(end_row) == 1
end_day = end_day + 1;

% if time is not found then it searches the next day
and so on until found
[end_row, col] = find(date_conv == end_day);
if end_day > floor(sd_annex(end, 1))
    continue
end
end

matrix(count, :) = [date_diff(i), date_conv(start_row, 1),
date_conv(end_row, 1)]; % displays where gaps exist and
the length of the gaps

% fake_matrix(start_row:end_row, 1)=[0];

% need to find the row number of the cell or search for
the cell!

end
end

date_conv = fake_matrix((1:(end-1)),1);
master_mat = [date_conv,num];
matrix(:,2) = matrix(:,2) - 0.01041660; %matrix(35,1) = 0.3; matrix
(66,3) = 734259.9895833;

%zero_mat = flipud(find(master_mat(:,1)==0)); %removes all cells
where time stamps are == 0
%for i = 1:size(zero_mat)
% master_mat(zero_mat(i),:) = [];
%end
%master_mat(17403,2) = 0.39;
%dlmwrite('dates_to_import.csv', (master_mat(:,1)), 'precision',
'%'14.7f');
%dlmwrite('matrix_to_thermal.csv', matrix, 'precision', '%14.7f');

% MUST FIND MISSING DATA SECTIONS AND INTERPOLATE TO FILL GAPS THEN
REMOVE
% THE GAPS IN ORDER TO HAVE ONLY REAL DATA
x_k = [];
k = 0;
new_interp = [];
for i=floor(master_mat(1:1)): (floor(master_mat(1:1))+(floor(
master_mat(end,1))-floor(master_mat(1,1)))) %create a loop
that is number of days in master_mat
for j=1:96
k=k+1;
x_k(k) = i+(j/96);
if x_k(k) > master_mat(end,1)
    continue
end
end
end
x_k = (floor(x_k * 10000000))/10000000;
% for interpolation
x = master_mat(:,1);
y = master_mat(:,(2:end));
x_i=x_k';
y_i = interp1(x,y,x_i,'linear');

interp_master_mat = [x_i((1:(end-1)),:) y_i((1:(end-1)),:)]; %the new master_mat
% clear x y x_i y_i

k=0;

for i = 1:length(matrix) %loop for removing data gaps within interpolated data
    if matrix(i,1)>= 0.04
        [r_l,c_l]=find(interp_master_mat(:,1)==matrix(i,2));
    end
end
% master_mat((20256:21408),(2:7))=master_mat((21408:22560),(2:7));

dlmwrite('master.mat.csv', master_mat, 'precision', '%14.7f');

master_mat = csvread('master.mat.csv');

% change 22.2°C for mid
% or 16.7°C for low
% or 27.8°C for high

temp = 22.2; % set inside temperature changes as seen above

mult_factor = 1.0; % either 1, 2, or 0.5

% wind energy stops on morning of Jan 24th, 2010 so code below is used to
% repeat dates when wind exists row 1–17483
wind_energy = master_mat((1:17483),3);

%extend wind_energy vector to be same length as solar_energy vector
mat_size = length(master_mat)/length(wind_energy);
rep=floor(mat_size);       %finds number of times to replicate
wind_energy
add_on = round((mat_size-rep)*length(wind_energy)); % length of
remainder to add on
% creates wind_energy to match total number of days
wind_energy= repmat(wind_energy, rep,1);
wind_energy = [wind_energy; wind_energy(1:add_on, 1)];

master_mat((17548:24570),2)= master_mat((17548:24570),2)/2;
    %division is included b/c a distinct doubling of energy is seen
    at the 17548 time step
solar_energy = [master_mat(:,2)];
    %at the same the kWh sensor starts showing numbers abnormally
    high (2x what the numbers should be... the code divides all cells
    after Jan 24th by 2
time_series = (1:1:(length(solar_energy)));
home_demand = [time_series; (master_mat(:,7)*mult_factor)'];
solar_energy = [time_series; solar_energy '];
wind_energy = [time_series; wind_energy '];

%creating a file that has the date stamps for use in converting t
    values from saved .csv files
dlmwrite('time_stamp.txt', master_mat(:,1), 'precision', '%14.7f');

% open the file with write permission
%fid = fopen('C:\Users\GW Barrett\Documents\gamsdir\projdir\home_demand.csv', 'w+');
fid = fopen('home_demand.csv', 'w+');
fprintf(fid, '/\n');
fprintf(fid, 't%lg,%6.4f\n', home_demand);
fprintf(fid, '/\n');
fclose(fid);

% open the file with write permission
%fid = fopen('C:\Users\GW Barrett\Documents\gamsdir\projdir\wind_energy.csv', 'w+');
fid = fopen('wind_energy.csv', 'w+');
fprintf(fid, '/\n');
fprintf(fid, 't%lg,%6.4f\n', wind_energy);
fprintf(fid, '/\n');
fclose(fid);

% open the file with write permission
%fid = fopen('C:\Users\GW Barrett\Documents\gamsdir\projdir\solar_energy.csv', 'w+');
fid = fopen('solar_energy.csv', 'w+');
fprintf(fid, '/\n');
fprintf(fid, 't%lg,%6.4f\n', solar_energy);
fprintf(fid, '/\n');
fclose(fid);
fclose(fid);

%-------------------------------------- SRVC EMISSIONS CODE
--------------------------------------

% emissions data below starts on May 1st 2005 and continues to the
end of the year
% then jumps to the start of the year until April 30th
srvc_data = csvread('srvc_emissions.csv');

% save srvc_data srvc_data;
srvc_emissions_renew = [];
for i=1:length(srvc_data)
    srvc_emissions_renew = [srvc_emissions_renew; repmat(srvc_data(i),4,1)];
end

day_start = start;
day_end = finish;

co2_dates=(day_start:(1/96):(day_end-(1/96)));
    % time stamps
    for CO2 emissions so cells coinciding with dates_off file can be
    removed
co2_dates = (floor(co2_dates* 10000000))/10000000;
    % rounding
    the numbers to align with dates_off
%create co2_emissions of equal length to co2_dates
mat_size = length(co2_dates)/length(srvc_emissions_renew);
rep=floor(mat_size); %finds number of times to replicate weekly_usage
add_on = (length(co2_dates)-length(srvc_emissions_renew)); % length of remainder to add on
% creates emissions to match total number of days
srvc_emissions_renew = repmat(srvc_emissions_renew, rep,1);
srvc_emissions_renew = [srvc_emissions_renew; srvc_emissions_renew (1:add_on, 1)];

srvc_emissions_renew = [co2_dates srvc_emissions_renew((1:length(co2_dates)),1)]; %full emissions matrix with time stamps

% % rm_cells = flipud(matrix(:,2:3));
% % for i=1:length(rm_cells)
% %    rm = find(srvc_emissions_renew(:,1)==rm_cells(i,1));
% %    rms = find(srvc_emissions_renew(:,1)==rm_cells(i,2));
% %    srvc_emissions_renew(((rm-1):rms,:)= [];
% % end

% % ID the full dates missing from other code and remove in
srvc_new = zeros(length(master_mat),2); %zeros(length(solar_energy),2);

for i=1:length(master_mat(:,1))
row = find (srvc_emissions_renew(:,1) == master_mat(i,1));
srvc_new(i,:) = srvc_emissions_renew(row,:);
end

marc=1:1:length(solar_energy);
srvc_emissions_renew=[marc; srvc_new(:,2) '];

% write all emissions data to text file in GAMS directory
%note: all data must be in a single row
%
%fid = fopen('C:\Users\GW Barrett\Documents\gamsdir\projdir\srvc_emissions_renewable.csv', 'w+');
fid = fopen('srvc_emissions_renewable.csv', 'w+');
fprintf(fid, '/n');
fprintf(fid, 't%lg,%6.4f/n', srvc_emissions_renew);
fprintf(fid, '/n');
close(fid);

% % % -- HOT_WATER DEMAND

% code has been commented out because already saved as .csv file
% weekly_demand = csvread('hot_water.csv');
% weekly_usage = vertcat(weekly_demand(:,7), weekly_demand(:,1),
% weekly_demand(:,2), weekly_demand(:,3), weekly_demand(:,4), ...
% weekly_demand(:,5), weekly_demand(:,6));
% %
% csvwrite('water_demand.csv', weekly_usage);
HOT_WATER_COST

**Natural Gas and Electricity Water Cost**

```
t_ground = 15.5556;  %ground water temp (15C ~ 59F)
t_surround = temp;  %temp of surrounding air (21C ~ 70F)
t_hi = 48.88889;  %temp of hot water tank (48.89C ~ 120F)
height = 1.6;  % [m]

r_value = 10;  % in American units [(hr*ft^2F)/Btu] but SI = [(K*m^2)/watts]

tank_volume = 190;  %[L]

rad = sqrt(((tank_volume*0.001)/(height*pi)));  %radius of water tank

area = (2*pi*rad*height) + (2*pi*rad^2);  %surface area of tank for h = 1.6 and tank_vol of 190 area = 23.594 ft^2
```

run hot_water_demand  %runs code for determining hot water demand

```
usage = csvread('water_demand.csv');
usage = usage * mult_factor;  %multiplication factor used for determining the demand for hot water and electricity

weekly_usage = (3.785*1*4.186*(t_hi - t_ground))*usage;  %[kJ]

now weekly_usage is in kJ per 15 min.
```
\%(\text{liters/gal}) \times (1\text{kg/L}) \times (\text{kJ to heat 1kg by 1C}) \times (\Delta T) \times \text{total water usage}

\text{master_mat}_\text{therm} = \text{master_mat}(:,1);
\text{number_days} = \text{floor}(\text{length(\text{master_mat}_\text{therm}(:,1))}/96);
\text{days} = \text{master_mat}_\text{therm}(:,1); \quad \%\text{all whole days present in data}

\text{losses} = \text{heat_loss}(t_{\text{hi}}, t_{\text{surround}}, \text{area}, r_{\text{value}}); \quad \%[\text{kJ}??] 
\text{equation looks at the amount of energy lost in fifteen min. when at typical 120deg.F}
\%((A \times \Delta T)/(R \times \text{conversion} \times 1000(\text{WtoK}) \times \text{sec in 15min} / 1000\text{J in 1kJ})

\text{loss} = \text{repmat}(\text{losses}, \text{length(\text{master_mat}_\text{therm}(:,1)),1}); \quad \%\text{repeats value for number of cells}

\% \text{replicate weekly demand matrix to match all matrix length}
\text{mat_size} = \text{length(\text{days})}/\text{length(\text{weekly_usage})};
\text{rep} = \text{floor}(\text{mat_size}); \quad \%\text{finds number of times to replicate weekly_usage}
\text{add_on} = \text{round}((\text{mat_size}-\text{rep}) \times \text{length(\text{weekly_usage}))}; \quad \% \text{length of remainder to add on}

\% \text{creates demand vector to match total number of days}
\text{water_demand} = \text{repmat(\text{weekly_usage}, \text{rep},1)};
\text{water_demand} = [\text{water_demand}; \text{weekly_usage}(1: \text{add_on}, 1)];
loss = loss(1:length(water_demand),1); %make sure loss and water_demand are the same size

time_series = (1:1:length(master_mat_therm));

% --- next code assures that the 'time_series' matches length of loss and water_demand
water_time_series = time_series(1:length(loss));

%——— Electricity Calculations ———
%e_cost= ((0.10/3600)*water_demand); %cost in $/kJ/kg_water *water usage [$0.10/kWh / 3600kJ/kWh]
%fuel cost * (kg/gal)*(kJ/kg*C)*(delta T)
e_water = ((loss + water_demand));

e_water = [water_time_series; (e_water)'];

% open the file with write permission

%fid = fopen('C:\Users\GW Barrett\Documents\gamsdir\projdir\e_water.csv', 'w+');
fid = fopen('e_water.csv', 'w+');
fprintf(fid, '/n');
fprintf(fid, 't%lg,%6.4f
', e_water);
fprintf(fid, '/n');
fclose(fid);
% NOT INCLUDING B/C GAMS HAS CODE TO CALC MULTIPLICATION!
% fid = fopen('C:\Users\GW Barrett\Documents\gamsdir\projdir\ e_water_cost.csv', 'w+');
%fprintf(fid, '/\n'); fprintf(fid, 't%lg,%6.4f\n', e_water_cost);
%fclose(fid);

% ———— Natural Gas Calculations ————
%ng_cost= ((1.30/105506)*water_demand); %[$/KJ/kg_water] 1.3/
  therm = 100,000btu = 105506KJ
  %fuel cost * (kg/gal)*(kJ/kg*C)*(delta T)
  ng_water = ((loss + water_demand));
  ng_water = [water_time_series; (ng_water)'];

%f fid = fopen('C:\Users\GW Barrett\Documents\gamsdir\projdir\ ng_water.csv', 'w+');
fid = fopen('ng_water.csv', 'w+');
fprintf(fid, '/\n');
fprintf(fid, 't%lg,%6.4f\n', ng_water);
fprintf(fid, '/\n');
fclose(fid);

% NOT INCLUDING B/C GAMS HAS CODE TO CALC MULTIPLICATION!
% fid = fopen('C:\Users\GW Barrett\Documents\gamsdir\projdir\ ng_water_cost.csv', 'w+');
%fprintf(fid, '/\n'); fprintf(fid, 't%lg,%6.4f\n', ng_water_cost);
fprintf(fid, '/\n');
% fclose(fid);

% % -- MAIN SOLAR THERMAL CONVERSION --
%

% % DATA_IMPORT_SOLAR_THERMAL

clear all
format long g

%Algorithm pulls info from separate xls files, finds all rows with the
%same date stamp and puts info into a new matrix.
%

%—All dates start on may 1 2009, everything was working at this point!
%—Files recent up to 3–10–2010 at 10PM with 15 min. intervals
%
% Problem arises when using hourly data b/c when taking the
% difference

start = 733894; % serial date associated w/ may 1, 2009
finish = 734319; % serial date associated w/ june 30, 2010

[num, time1] = xlsread('annex_data'); % Imports dates and values from "ANNEX_data.xls"
time_se = time1([2:length(time1)],1); % isolate date and time1 column into separate matrix

101
% CAREFUL HERE — use only for incremental data

% num = diff(num);
% for i = 1:length(num) %convert all negative
    % for i = 1:length(num) %convert all negative
        information to zeros
    % if num(i) < 0
    %     num(i) = 0; num(i+1) = 0; num(i-1) = 0;
    % end
% end

annex_irrad = num; % isolate annex irradiation

date_conv = datenum(time_se); % change date and time to serial
    number for comparisons

date_conv = floor(date_conv * 10000000)/10000000;

sd_annex = [date_conv num];
gone = diff(sd_annex(:,1)); % removing the rows that have
duplicate time stamps (only 2)
gone = find(gone == 0);
for i = 1:size(gone)
    sd_annex(gone(i),:) = [];
end

k=0;
good_dates = dlmread('dates_to_import.csv'); % good dates are
    the dates that have been converted
master_mat_therm = zeros(length(good_dates),3);
for i = 1:length(sd_annex)
\( r = \text{find}(sd\text{\_annex}(i,1) == \text{good}\_\text{dates}(:,1)); \)

if \( r \neq 0 \)

\( k = k+1; \)
\( \text{master}\_\text{mat}\_\text{therm}(k,:) = sd\_\text{annex}(i,:); \)

end

date\_conv = \text{master}\_\text{mat}\_\text{therm}(:,1);
\text{num} = \text{master}\_\text{mat}\_\text{therm}(:,2:end);
\text{matrix} = \text{dlmread('matrix\_to\_thermal.csv')};

\% \% DATES\_OFF\_THERMAL
\% 
\% BEGIN FUNCTION DATES\_OFF

\text{date\_conv}(end,:) = \lfloor \text{date\_conv}(end,:) \rfloor + 0.9895833;
\text{fake\_matrix} = \text{date\_conv};
\text{date\_diff} = \text{diff(fake\_matrix)};
\text{matrix} = [ ];
\text{count} = 0;
\text{start\_date} = \text{date\_conv}(1); \% \text{defines a starting date}

\text{for } i = 1:(\text{length(date\_conv)-1}) \% \text{loop is to ID and remove any full day with data missing for a period of time greater than 30 min}

\text{if } \text{date\_diff}(i) >= 0.0209 \% \text{for gap larger than 30 min}
\text{count} = \text{count} +1;

matrix(\text{count}, :) = [i, \text{date_diff}(i)]; % displays where gaps exist and the length of the gaps

gap\text{\_start} = \text{date\_conv}(i); % IDs the start of missing string
gap\text{\_end} = gap\text{\_start} + \text{date\_diff}(i); % IDs the end of the gap (serial date number of last gap)

begin\_day = \text{floor}(gap\text{\_start}) + 0.0104166; % first day of gap (12:00:00am with first 15min added) so 12:15:00am
[start\_row, col] = \text{find}(\text{date\_conv} == \text{begin\_day}); % searching to find the 12:15:00am of the gap

while isempty(start\_row) == 1
    begin\_day = begin\_day - 1; % if not found then move to the previous day
    [start\_row, col] = \text{find}(\text{date\_conv} == \text{begin\_day});
    % search again until a beginning day is found
end

end\_day = (\text{floor}(gap\text{\_end})) + .9895833; % last day of gap at 11:45:00pm
[end\_row, col] = \text{find}(\text{date\_conv} == \text{end\_day}); % finds 11:45:00pm of the last day of the gap

while isempty(end\_row) == 1
end_day = end_day +1; %if time is not found
then it searches the next day and so on until found
[end_row, col] = find(date_conv == end_day);
end
matrix(count,:) = [date_diff(i), date_conv(start_row,1),
date_conv(end_row,1)]; %displays where gaps exist and
the length of the gaps
fake_matrix(start_row:end_row, 1)=[0]; %need to find the
row number of the cell or search for the cell!
end
end

zero_mat = flipud(find(master_mat_therm(:,1)==0)); %removes all
cells where time stamps are == 0
for i = 1:size(zero_mat)
    master_mat_therm(zero_mat(i),:) = [];
end

date_conv = fake_matrix;
master_mat_therm = [date_conv,num];

zero_mat = flipud(find(master_mat_therm(:,1)==0)); %removes all
cells where time stamps are == 0
for i = 1:size(zero_mat)
    master_mat_therm(zero_mat(i),:) = [];
end

105
% hi = [];  
% new_master = master_mat_therm * 1000000;  
% new_mat = matrix* 1000000;  
% csvwrite('master_mat_therm.csv', new_master);  
% csvwrite('matrix.csv', new_mat);  

% --------------------------------- SRVC_EMISSIONS_THERMAL ---------------------------------

Don't need b/c have info from electricity section  

%emissions data below starts on May 1st 2005 and continues to the end of the year  
%then jumps to the start of the year until April 30th  
srvc_data = csvread('srvc_emissions.csv');  

% save srvc_data srvc_data;  
srvc_emissions_therm=[];  
for i=1:length(srvc_data)  
    srvc_emissions_therm = [srvc_emissions_therm; repmat(srvc_data(i,:),4,1)];  
end  

day_start=start;  
day_end = finish;
co2_dates = \{day_start:(1/96):(day_end-(1/96))\}';  % time stamps for CO2 emissions so cells coinciding with dates_off file can be removed
co2_dates = (floor(co2_dates* 10000000))/10000000;  % rounding the numbers to align with dates_off

% create co2_emissions of equal length to co2_dates
mat_size = length(co2_dates)/length(srvc_emissions_therm);
rep=floor(mat_size);  % finds number of times to replicate weekly_usage
add_on = floor((mat_size-rep)*length(srvc_emissions_therm));  % length of remainder to add on
% creates emissions to match total number of days
srvc_emissions_therm = repmat(srvc_emissions_therm, rep,1);
srvc_emissions_therm = [srvc_emissions_therm;
     srvc_emissions_therm(1:add_on, 1)];

srvc_emissions_therm = [co2_dates srvc_emissions_therm];  % full emissions matrix with time stamps

% ID the full dates missing from other code and remove in
srvc_emissions_therm
rm_cells = flipud(matrix(:,2:3));
for i=1:length(rm_cells)
    rm = find(srvc_emissions_therm(:,1)==rm_cells(i,1));
    rms = find(srvc_emissions_therm(:,1)==rm_cells(i,2));
srvc_emissions_therm((rm:rms,:),:) = [];
marc = [1:1:(size(srvc_emissions_therm))];
srvc_emissions_therm = [marc; srvc_emissions_therm(:,2) ];

% write all emissions data to text file in GAMS directory
% note: all data must be in a single row

fid = fopen('C:\Users\GW Barrett\Documents\gamsdir\projdir\srvc_emissions_thermal.csv', 'w+');
fprintf(fid, '/n');
fprintf(fid, 't%lg,%6.4f\n', srvc_emissions_therm);
fprintf(fid, '/\n');
fclose(fid);

% INTERPOLATION_DATA_THERMAL
ADDED TO DATA_IMPORT.M FILE!!!
clear all

master_mat_therm = csvread('master_mat_therm.csv');
master_mat_therm(:,1) = (master_mat_therm(:,1)/10000000);
master_mat_therm = master_mat_therm/10000000;
matrix = csvread('matrix.csv');
% MUST FIND MISSING DATA SECTIONS AND INTERPOLATE TO FILL GAPS THEN REMOVE

% THE GAPS IN ORDER TO HAVE ONLY REAL DATA

x_k = [];
k = 0;
new_interp = [];
for i = floor(master_mat_therm(1:1)) : (floor(master_mat_therm(1:1)) + (floor(master_mat_therm(end, 1)) - floor(master_mat_therm(1, 1))))
    for j = 1:96
        k = k + 1;
        x_k(k) = i + (j / 96);
        if x_k(k) > master_mat_therm(end, 1)
            continue
        end
    end
end
x_k = (floor(x_k * 10000000)) / 10000000;
% FOR INTERPOLATION
x = master_mat_therm(:, 1);
y = master_mat_therm(:, (2:end));
x_i = x_k';
y_i = interp1(x, y, x_i, 'spline');

interp_master_mat = [x_i y_i]; % THE NEW MASTER_MAT_ThERM
clear x y x_i y_i
k = 0;
for i = 1:length(matrix)
if matrix(i,1)>= 0.1
    [r_l,c_l]=find(interp_master_mat(:,1)==matrix(i,2));
    [r_h,c_h]=find(interp_master_mat(:,1)==matrix(i,3));
    interp_master_mat((r_l:r_h,:), :) = [];
end
end

master_mat_therm=interp_master_mat;
%csvwrite('master_mat_therm.csv', master_mat_therm);
%csvwrite('matrix.csv', (matrix*1000000));
clear new_interp interp_master_mat

dlmtree( 'data_for_gams_thermal.csv', master_mat_therm,'precision',
    '%14.7f');
%csvwrite('master_mat_thermal.csv', master_mat_therm);

%display ('Included in data_for_gams_thermal.csv file in order of
    columns:')
%display (time2(1,:))
%display (time1(1,(2:end)))

clear fake_matrix date_diff sd_annex sd_house

%[time_series] = textread('time_series.txt', '%q');       SHOULD NO
LONGER NEED THIS 2–6–2010

time_series = [1:1:length(master_mat_therm)];
HOT_WATER_DEMAND

code has been commented out because already saved as .csv file
weekly_demand = csvread('hot_water.csv');
weekly_usage = vertcat(weekly_demand(:,7), weekly_demand(:,1),
weekly_demand(:,2), weekly_demand(:,3), weekly_demand(:,4), ...
weekly_demand(:,5), weekly_demand(:,6));
csvwrite('water_demand.csv', weekly_usage);
%weekly_usage = [week weekly_usage];

%%
HOT_WATER_COST

% Natural Gas and Electricity Water Cost
t_ground = 15.5556; %ground water temp (15C ~ 59F)
t_surround = 21; %temp of surrounding air (21C ~ 70F)
t_hi = 48.88889; %temp of hot water tank (48.89C ~ 120F)
height = 1.6; % [m]
r_value = 10; % in American units [(hr ft^2 F)/Btu] but
SI = [(K*m^2) / watts]
tank_volume = 190; %[L]
rad = sqrt((tank_volume*0.001)/(height*pi)); %radius of water
tank
area = (2*pi*rad*height) + (2*pi*rad^2); %surface area of tank for h
tank = 1.6 and tank_vol of 190 area = 23.594 ft^2

111
usage = csvread('water_demand.csv');
weekly_usage = (3.785*1*4.186*(t_hi - t_ground))*usage;  %[kJ]
now weekly_usage is in kJ per 15 min.
%(liters/gal) * (1kg/L) * (kJ to heat 1kg by 1C)* (delta T) *
total water usage

master_mat_therm = csvread('data_for_gams_thermal.csv');
number_days = floor(length(master_mat_therm(:,1))/96);
days = master_mat_therm(:,1);  %all whole days present in data

losses = heat_loss(t_hi,t_surround,area,r_value);  %[kJ??]
equation looks at the amount of energy lost in fifteen min.
%((A*deltaT)/(R*conversion*1000(WtokW)))* (sec in 15min / 1000J
in 1kJ)

loss = repmat(losses,length(master_mat_therm(:,1)),1);  %repeats
value for number of cells

%%%NEED TO CHANGE FOR 180 WATER TANK %%%

% replicate weekly demand matrix to match all matrix length
mat_size = length(days)/length(weekly_usage);
rep=floor(mat_size);  %finds number of times to replicate
weekly_usage
add_on = floor((mat_size-rep)*length(weekly_usage)); %
length of remainder to add on
% creates demand vector to match total number of days
water_demand = repmat(weekly_usage, rep, 1);
    water_demand = [water_demand; weekly_usage(1:add_on, 1)];

loss = loss(1:length(water_demand),1); %make sure loss and
water_demand are the same size

time_series = [1:1:length(master_mat_therm)];

% —— next code assures that the 'time_series' matches length of
loss and water_demand
water_time_series = time_series(1:length(loss));

% ——— Electricity Calculations ———
%e_cost= ((0.10/3600)*water_demand); %cost in $/kJ/kg_water *water
usage [$0.10/kWh / 3600kJ/kWh]
    %fuel cost * (kg/gal)*(kJ/kg*C)*(delta T)
e_water = ((loss + water_demand));
e_water = [water_time_series; (e_water) '];

% open the file with write permission

fid = fopen('C:\Users\GW Barrett\Documents\gamsdir\projdir\e_water.csv', 'w+');
    fprintf(fid, '/n');

113
% % NOT INCLUDING B/C GAMS HAS CODE TO CALC MULTIPLICATION!

\%
\% e_water_cost = fopen('C:\Users\GW Barrett\Documents\gamsdir\projdir\e_water_cost.csv', 'w+');
\% fprintf(fid, '/\n'); fprintf(fid, 't%lg,%.4f\n', e_water);
\% fclose(fid);

\%
\% --- Natural Gas Calculations ---
\%
\% ng_cost = ((1.30/105506)*water_demand);
\% $/KJ/kg\text{water}$ 1.3/
\%\text{therm} = 100,000\text{btu} = 105506\text{KJ}
\% fuel cost \ast (kg/gal\text{)}\ast(kJ/kg\text{C})\ast(delta T)
\% ng_water = ((\text{loss} + \text{water\_demand})�
\% ng\_water = [\text{water\_time\_series; (ng\_water)}�]

\% fid = fopen('C:\Users\GW Barrett\Documents\gamsdir\projdir\ng\_water.csv', 'w+');
\% fprintf(fid, '/\n'); fprintf(fid, 't%lg,%.4f\n', ng\_water);
\% fclose(fid);

% % NOT INCLUDING B/C GAMS HAS CODE TO CALC MULTIPLICATION!
% fid = fopen('C:\Users\GW Barrett\Documents\gamsdir\projdir\ng_water_cost.csv', 'w+');
%fprintf(fid, '/n'); fprintf(fid, 't%lg,%6.4fn', ng_water_cost);
fprintf(fid, '/n');
% fclose(fid);

% % SOLAR_THERMAL_CONSTANT

%values of T_f, T_a, and I_c will vary causing F_r to vary as well

% Method for heat gain into hot water storage tank from solar collector
%
% To pull information from Irradiance and Ambient Temperature from GRIDc
% data
%
% Most values are dependent upon the collector specified and will vary
%
% Start with an assumed minimum set temperature of the Hot Water Tank
% Temp of water at start = 120 F or 49 C
%Constants for Solar Thermal Collector
m=190; c=4.186;

%collector transmittance
tau = 0.91;

%collector absorptivity
alpha = 0.95;

%heat loss coefficient to surroundings
U_c = 3.7;

%efficiency of collector
eta = 0.706;

%A_c is area of collector aperture (m^2)
A_c = 2.137;

%T_f is initial temperature into collector and will change with time and heat input
%(deg C)
T_f = 48.88;

%Volume of tank is 50 gal or ~190L (L)
tank_vol = 190; %units are liters
tank_mass = 190;  \%total mass of water in the tank, 1 kg/L is water density

\% Import information from GRIDc data (amb temp., Insolation)
I_c = csvread('data_for_gams_thermal.csv', 0,2);
    I_c = I_c(:,1);
T_a = csvread('data_for_gams_thermal.csv', 0,1);
    T_a = T_a(:,1);
z = length(T_a);

\%from looking at previous data, it appears that any time the insolation is \%below 200W/m2 the thermal unit does not transfer heat to the fluid;
    thus,
    \%this next line of code:
    for i=1:length(I_c)
        if I_c(i) <= 200
            I_c(i) = 0;
        end
    end

    \%remove later!!!
T_a_new = [T_a I_c];
\%T_fvec(1) = T_f;

for i = 1:z
    T_fvec(i) = T_f;
%Calculate the F_r (heat removal factor) values for above values
F_r(i) = (eta) / ((tau*alpha) - (U_c*(T_f - T_a(i))) / I_c(i));%rearranged from thermal efficiency

% q_u[W] is the heat into the collector fluid (J) CALLED ”HOTTEL–WHILLIER–BLISS EQUATION
q_u(i) = A_c * F_r(i) *((alpha * tau* I_c(i)) - (U_c * (T_f - T_a(i)))^2) *(0.9); % (eqn in [kJ]) need to multiply by time to get heat flow in energy units (15min*60s/min)

% q_u causes an increase in temperature of the hot water tank - 1 L h2o
% weighs 1kg
T_f = (q_u(i) / (tank_mass*4.186)) + T_f; % Q = m*c_v*(T_f–Tin), solve for T_f
T_f = temp_loss(T_f,t_surround,area,r_value);

T_f(T_f<48.889)=48.889; % 48.9 is 120F
T_f(T_f>82.2222)=82.2222; % 82.2 is 180F

end
% T_fvec = T_fvec ’; q_u = q_u ’; F_r = F_r ’;

% q_u into Btu if desired 1 kJ = 0.947817 Btu and 1J = 0.0000947817 Btu
q_u_btu = q_u * 0.947817;
%Determining the Efficiency of the Collector from SRCC Data in SI units

intercept = 0.715;
slope = 3.994;  %units are W / m^2 * deg.C
del_T = T_fvec' - T_a;

%Determine efficiency based on MAE 421 efficiency calculation from SRCC
%data: Eff = y-intercept - slope*(delta T / insolation)
for i = 1:z
    efficiency(i) = intercept - (slope * (del_T(i) / I_c(i)));
    if efficiency(i) < 0
        efficiency(i) = 0;
    end
end

end
efficiency = efficiency';

%q_u_data is in watt per m^2, need to convert to joules for the 15 minute period so multiply the number by 15 min/period * 60 s/min = 900 s/period
%then convert to kJ from joules so divide by 1e+3
for i = 1:z  %now q_u_data is in kJ

119
q_u_data(i) = efficiency(i) * I_c(i) * A_c * 60*15 / 1e+3;
end

thermal_data=[water_time_series(1,1:length(q_u)); q_u];

%% open the file with write permission
% fid = fopen('C:\Users\GW Barrett\Documents\gamsdir\projdir\solar_water_heating.csv', 'w+');
% fprintf(fid, '/n');
% fprintf(fid, 't%lg,%6.4f
', thermal_data);
% fprintf(fid, '/n');
% fclose(fid)

%% ———————————————————— CALCULATION OF SOLAR THERMAL TEMPS AND HEAT REQUIRED

total_heat=zeros(length(thermal_data),1); heat_required = zeros(
    length(thermal_data),1); T_new= zeros((length(thermal_data)+1),1)
    ; % preallocating the vectors
T_new(1) = 48.89;
for i = 2:length(thermal_data+1)    % converts insolation to
    temperature rise
    total_heat(i) = thermal_data(2,i) - water_demand(i) - heat_loss(
        T_new(i-1), t_surround, area, r_value);
totalheat = heat gain from insolation - heat loss from demand - heat loss to surroundings

T_new(i) = heat_to_temp(m,c,water_heat(m,c,T_new(i-1)) + total_heat(i), 15.56);

% subtract heat from previous temp then change to temp
if T_new(i) > 82.22
    T_new(i) = 82.22;
end

if T_new(i) < 48.888889
    heat_required(i) = (26511.33 - water_heat(m,c,T_new(i)));
    %26511.3 = heat in water from 60–120F
    T_new(i) = 48.89;
end
end

heat_required = [water_time_series(1,1:length(heat_required));
                 heat_required '];

% open the file with write permission
fid = fopen('C:\Users\GW Barrett\Documents\gamsdir\projdir\solar_water_heating.csv', 'w+');
fprintf(fid, '/n');
fprintf(fid, 't%lg,%6.4f\n', heat_required);
fprintf(fid, '/\n');
fclose(fid);
D.2 Functions for MATLAB

The following are shorter bits of code for making the MATLAB section run. All the function names are embedded in the code.

function [Temp_out] = heat_gain_solar (T_in, heat) %finds new temperature of tank with heat addition
% *** heat may be positive or negative
    m = 190; %mass of water = volume of water [L]
    c = 4.186; %specific heat of water [kJ/kg*C]
    Temp_out = (heat/(m *c)) + T_in;
end

function [heat_lost] = heat_loss (t_in, t_surround, area, r_value)
%calculates new temperature based on the heat loss from tank
    r_conv = 0.17611; %converts r_value to SI units [k*m^2/W]
    heat_lost = (area*(t_in−t_surround)/(r_value*r_conv))*0.9;
%0.9 converts watts to kJ/15min interval
end

function [temp] = heat_to_temp(volume, c, heat, t_low)
    temp = (heat/(volume*c)) + t_low;
end

function [temp_c] = temp_cel(temp_f)
    temp_c = (temp_f −32) *(5/9);
end
function [temp_f] = temp_far(temp_c)
    temp_f = (temp_c*(9/5))+32;
end

function [temp_lost] = temp_loss (t_in, t_surround, area, r_value)
    %calculates new temperature based on the heat loss from tank
    r_conv = 0.17611; %converts r_value to SI units [k*m^2/W]
    heat_lost = (area*(t_in−t_surround)/(r_value*0.17611))*0.9;
    %0.9 converts watts to kJ/15min interval
    temp_lost = t_in − (heat_lost/(190*4.186));
end

%function calculates the heat in water at a certain temperature
function [heated_water] = water_heat(m, c, t_hi) %temps need to
    be in Celcius!
    heated_water = (m*c*(t_hi−temp_cel(60)));
end
D.3 GAMS Inner Shell Code

The following bit of code is for the inner optimization run conducted by GAMS. It runs to optimize CO\textsubscript{2} emissions for one HVAC system. This is only for the 72\textdegree F medium demand scenario. The code must be changed in order to incorporate the different temperatures and demands.

```
****************************************************************************** // COMBINATION OF BOTH CODES!
*$eolcom //
$Offsymlist
$Offsymxref
Option Limrow=0;
Option Limcol=0;
Option Solprint=off;
*
Sets

t time / t1*t26783 /
tech technologies / solar, wind /

*runs different runs / run1*run4 /
*hvac_run tech for heating and cooling / elec_pump, geo_pump, elec_rad, ng_furn, AC_central/

heat_tech(hvac_run) technology for heating / elec_pump, geo_pump, elec_rad, ng_furn /
```
cool_tech(hvac_run) technology for cooling /elec_pump,
    geo_pump, AC_central /
heat_elec(heat_tech) techs using elec for heating /elec_pump,
    geo_pump, elec_rad /
cool_elec(cool_tech) techs using elec for cooling /elec_pump,
    geo_pump, AC_central/

h2o_tech water heating technologies / ng, elec,
    solar_ng, solar_elec/

insul_tech technology for insulation /wall,
    floor, window, ceiling/
r_value insulation values /low, mid,
    high / //

*******************
Parameters

hvac_vec(hvac_run)
$offlisting

solar_energy(t) "energy provided from solar – normalized"
$ondelim
$include solar_energy.csv
wind_energy(t)  "energy provided from wind – normalized"

srvc_emissions_renew(t)  "carbon emissions from grid – units in kg_CO2/kWh"

home_demand(t)  "home energy demand from Solar House"

****  HOT WATER

e_water(t)  "energy required to heat water via elect [units of kJ]"

*e_water is in units of kJ, all values must be divided by 3600 to be in kWh...
ng_water(t) "energy required for nat. gas water heating"

*solar_water_heating(t) "energy required if solar thermal hot water system is installed"

*solar_water_heating is the additional amount of heat required to warm water to correct temps... treat as NG and electricity [kJ]

*elec_emissions_therm(t) "emissions from grid electricity"

** units are in kg/kWh or metric tons/MWh

*ambient_temp_cold(t) "ambient temperature from cold days (days below 72F or 22.2C)"
ambient_temp_hot(t)  
"ambient temperature from hot days (days above 72F or 22.2C)"

$ondelim
$include ambient_hot_temperature.csv
$offdelim

attic_temp_cold(t)  
"cold temp of attic – needs heat (days below 72F or 22.2C)"

$ondelim
$include attic_cold_temperature.csv
$offdelim

attic_temp_hot(t)  
"hot temp of attic – needs cooling (days above 72F or 22.2C)"

$ondelim
$include attic_hot_temperature.csv
$offdelim

$oncontext

INSULATION PRICES FROM LOWES website

<table>
<thead>
<tr>
<th>sq ft</th>
<th>price $</th>
<th>$/ft2</th>
<th>$/m2</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-11</td>
<td>50</td>
<td>15</td>
<td>0.3</td>
</tr>
<tr>
<td>R-13</td>
<td>106.5</td>
<td>39.1</td>
<td>0.367</td>
</tr>
<tr>
<td>R-19</td>
<td>77.5</td>
<td>37.6</td>
<td>0.485</td>
</tr>
<tr>
<td>R-30</td>
<td>58.6</td>
<td>42.8</td>
<td>0.73</td>
</tr>
<tr>
<td>R-38</td>
<td>42.7</td>
<td>39.7</td>
<td>0.93</td>
</tr>
<tr>
<td>R-50</td>
<td>1.125</td>
<td>12.11</td>
<td></td>
</tr>
</tbody>
</table>
***** FOR ELECTRICITY AND HOT WATER

*fixed operation and maintenance cost for wind and solar ($/kW)
fixom(tech) fixed operation and maint. cost for technology
  /solar  50
  wind  100/ //number found to be $11.5/kW

*variable operation and maintenance cost for wind and solar ($/kW)
varom(tech) variable operation and maint. cost for technology
  /solar  0.01
  wind  0.027/

*capital costs for wind and solar ($/kW)
cap_cost(tech) capital cost for technology
  /solar  7600
  wind  4000/

renew_life(tech) life span of renewable energy technologies
  /solar  30
  wind  30/

capcost_h2o (h2o_tech) "capital cost of each technology including installation"
/ng 850
elec 750
solar_ng 1145  // discounts are added to this
cost due to nonlinear nature of incentives
solar_elec 1125/  // originally $4000 for both
then add in cost of extra tank for different heating
style

water_life(h2o_tech)  "life span of water technologies"
/ng 15
elec 15
solar_ng 30
solar_elec 30 /

***** HOME INSULATION
*values for cost — — — — ASSUMING INSULATION HAS 30YR
LIFESPAN !!!
*** Should cost function take into consideration the necessity
of having insulation (base insulation is 0 with incremental being
only additional?)

floor_cost(r_value) "cost of floor insulation $/m2"
/ low  3.23
mid 5.23
high 7.86/

wall_cost(r_value) "cost of wall insulation $/m2"
/ low  3.23
mid 5.23
high 7.86/
ceil_cost(r_value) "cost of ceiling insulation $/m2"
/ low 7.86
mid 10.01
high 12.11/
window_cost(r_value) "cost of window insulation $/m2 - NO NUMBERS YET!!"
/ low 107.6
mid 215.3
high 322.9/

*values for insulation
floor_insul(r_value) floor insulation
/ low 11
mid 19
high 30/
wall_insul(r_value) wall insulation
/ low 11
mid 19
high 30/
ceil_insul(r_value) ceiling insulation
/ low 30
mid 38
high 50/
window_insul(r_value) window insulation
/ low 4
DATA FOR HEATING AND COOLING TECHNOLOGIES

coefficient of performance, for now, should be a ratio of heat removed to electricity consumed
by using the temperature outside and the internal "ideal" temp should be able to find how much needs to be moved out of the home

heat_cost(heat_tech) cost for heating tech (cost of NGFURN is uncertain!) COST IN $$$
/ ng_furn 1080 //original price was 1500 but
normalized for capacity
elec_rad 1368 //original price was 1900 but
normalized for capacity
elec_pump 1512 //original price was 2100 but
normalized for capacity
go_pump 10000 / // original price was $2677.5:
30% and 35% incentives applied for federal and state respectively state maxed at 8400 total (no where close)

heat_eff(heat_tech) "efficiency of heating tech – Original units are in kJOut/kJin"
/ ng_furn 0.78
elec_rad 1.0
elec_pump 2.26
<table>
<thead>
<tr>
<th></th>
<th>Value</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>geo_pump</td>
<td>3.3</td>
<td>/</td>
</tr>
<tr>
<td>heat_cap(heat_tech)</td>
<td>&quot;capacity for heating tech units kJ/15 min&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>/ ng_furn 9495.5</td>
<td>// 13188.2 before normalized</td>
</tr>
<tr>
<td></td>
<td>elec_rad 9495.5</td>
<td>// 13188.2 before normalized</td>
</tr>
<tr>
<td></td>
<td>elec_pump 9495.5</td>
<td>// 13188.2 before normalized</td>
</tr>
<tr>
<td>geo_pump</td>
<td>9495.5</td>
<td>/</td>
</tr>
<tr>
<td>heat_life(heat_tech)</td>
<td>&quot;efficiency of heating tech – Original units are in kJout/kJin&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>/ ng_furn 30</td>
<td></td>
</tr>
<tr>
<td></td>
<td>elec_rad 30</td>
<td></td>
</tr>
</tbody>
</table>
|                     | elec_pump 30    | //15 why is the number half the lifespan as the others???
| geo_pump            | 30             | /                                        |
|                     | //15 why is the number half the lifespan as the others???
| cool_cost(cool_tech)| cost for cooling tech COST IN $$$           |
|                     | / AC_central 2600 |                                           |
|                     | elec_pump 2100  |                                           |
| geo_pump            | 10000          | // 892.5 original price was $: 30% and 35% incentives applied for federal and state respectively state maxed at 8400 total (no where close)

133
cool_eff(cool_tech) "efficiency of cooling tech – Original units are in kJout/kJin"
    / AC_central 3.81 //efficiency conversion done at electricity totaling
elec_pump 3.81
go_pump 14.10 /

cool_cap(cool_tech) "capacity for cooling tech units kJ/15 min"
    / AC_central 9495.5
elec_pump 9495.5
go_pump 9495.5 /

cool_life(cool_tech) cost for cooling tech COST IN $$$
    / AC_central 30
elec_pump 30 //15 why is the number half the lifespan as the others???
go_pump 30/ //15 why is the number half the lifespan as the others???

;

*********************************************************************
Scalars

*price of electricity either to buy from utility or to sell to utility
cost_purchased "price of elec [$/kwh]" /0.105 /
*for NPV calculations

    int_rate        "daily rate for 5% APR int"
    / 0.00013699 /

        heat_to_elec        "convert heat to electricity 1/3600" /
    0.0002778 /

        ng_cost        "cost of nat. gas $/kJ"
    / 0.00001232 /

        ng_emissions        "emissions factor associated with burning
                     NG to heat water kg/kJ"
    / 0.0000503 /

*area of surfaces ALL IN m²

    floor_area    / 74.32 /
    ceiling_area  / 74.32 /
    wall_area     / 78.41 /
    window_area   / 26.76 /

    ground_temp      65F    / 18.3 /
    inside_temp_sum  72F    / 22.2 /
    inside_temp_win  72F    / 22.2 /

    r_conversion   r−value to metric    / 0.17611 /

    life30        days in 30 years    / 10950 /

    days         "total days included in finances (26783/96)"
    / 281 /
\(\text{state\_tax}\) NC tax credit for systems /0.35/
\(\text{federal\_tax}\) US tax credit for systems /0.30/

\$offlisting

******************************

Variables

\(x_1(\text{tech})\) capacity of technology 'tech'
\(x_2(\text{tech})\) capacity of extra technology
\(x(\text{tech})\) combined technologies
\(e_p(t)\) hourly electricity purchase from grid
\(e_p1(t)\) hourly electricity purchase from grid
\(e_p2(t)\) hourly electricity purchase from grid
\(e_p3(t)\) hourly electricity purchase from grid
\(\text{*overall\_cost}\) variable for objective function
\(\text{total\_cost}\) total cost

\(\text{e\_elec}\) kwh – amount of electricity required
\(\text{electricity\_cost}\) total cost of electricity
\(\text{solar\_en}\) "sum of solar\_energy(t)"
\(\text{wind\_en}\) "sum of wind\_energy(t)"

\(\text{ng\_fuel}\) amount of NG needed
\(\text{co2\_emissions}\) total CO2 from heating and cooling
\(\text{co2\_ng\_heat}\) emissions from natural gas heating
from hot water:

- **tech_h2o(h2o_tech)**: "all technologies are variables"
- **cost_h2o**: "system wide hot water cost"
- **cost_h2o_ng**: cost for NG hot water system
- **cost_h2o_elec**: cost for elec hot water system
- **cost_h2o_solar_elec**: cost for solar with elec backup
- **cost_h2o_solar_ng**: cost for solar with NG backup

- **ng_fuel_water**: Nat gas cost for water heating
- **ng_fuel_solar**: Nat gas cost for solar hotwater
- **ng_fuel_furn**: Nat gas cost for space heating
- **ng_fuel_cost**: total cost for natural gas

- **budget**: amount of money available to spend on upgrades
- **budget_wind**: cost of wind
- **budget_solar**: cost of solar
- **carbon_total**: "amount of carbon emitted"

- **ceiling(r_value)**: ceiling insulation technology
- **window(r_value)**: window insulation technology
- **floor(r_value)**: floor insulation technology
- **wall(r_value)**: wall insulation technology

- **heat_loss(t)**: total amount of heat lost to outside
- **heat_gain(t)**: total amount of heat gained in the house
- **tot_heat_loss**: kJ – total heat lost to outside
tot_heat_gain  kj – total heat gain from outside

floor_loss  heat lost through floor
wall_loss  ” ”
ceiling_loss  ” ”
window_loss  ” ”

floor_gain  heat gained through floor
wall_gain  ” ”
ceiling_gain  ” ”
window_gain  ” ”

insul_cost  cost of insulation

h_cost  cost of heating equipment
c_cost  cost of cooling equipment
h_cost1  cost of heating equipment
c_cost1  cost of cooling equipment
h_cost2  cost of heating equipment
c_cost2  cost of cooling equipment
h_cost3  cost of heating equipment
c_cost3  cost of cooling equipment

solar_incen  Tax incentive for solar installation
wind_incen  Tax incentive for wind installation
geo_incen1  Tax incentive for geothermal heating installation
geo_incen2  Tax incentive for geothermal cooling installation

state_solar_incen
state_wind_incen
state_geo_incen1
state_geo_incen2

demand_home  Summed amount of electricity used by home
e_water_demand
solar_e_demand

heating_demand
cooling_demand
heating_demand1
cooling_demand1
heating_demand2
cooling_demand2
heating_demand3
cooling_demand3

***;

******************************************************************************
Positive variable x(tech), x1(tech), x2(tech), state_solar_incen,
state_wind_incen, state_geo_incen1, state_geo_incen2,
state_therm_incen;
*budget;

Binary variable tech_h2o, window(r_value), wall(r_value), floor(r_value), ceiling(r_value);

Equations

x_equ combination of x variables
x1_solar_equ max amount for state solar
x1_wind_equ wind
x2_solar_equ limiting rest of solar
x2_wind_equ wind

home_demand_equ
e_water_demand_equ
solar_e_demand_equ

heating_demand_equ
cooling_demand_equ
heating_demand_equ1
cooling_demand_equ1
heating_demand_equ2
cooling_demand_equ2
heating_demand_equ3
cooling_demand_equ3
<table>
<thead>
<tr>
<th>Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>electricity_cost_equ</td>
<td>total amount spent on electricity</td>
</tr>
<tr>
<td>overall_cost_equ</td>
<td>define objective function</td>
</tr>
<tr>
<td>demand_elec</td>
<td>constraint that supply equals demand</td>
</tr>
<tr>
<td>demand_elec1</td>
<td>constraint that supply equals demand</td>
</tr>
<tr>
<td>demand_elec2</td>
<td>constraint that supply equals demand</td>
</tr>
<tr>
<td>demand_elec3</td>
<td>constraint that supply equals demand</td>
</tr>
<tr>
<td>npv_elec</td>
<td>constraint for cost of system</td>
</tr>
<tr>
<td>npv_elec_solar</td>
<td>solar cost</td>
</tr>
<tr>
<td>npv_elec_wind</td>
<td>wind cost</td>
</tr>
<tr>
<td>e_bought</td>
<td>checking amount of electricity bought</td>
</tr>
<tr>
<td>solar_prod_equ</td>
<td>amount of solar energy produced</td>
</tr>
<tr>
<td>wind_prod_equ</td>
<td>amount of wind produced</td>
</tr>
<tr>
<td>cost_h2o_equ</td>
<td>total hot water cost equation</td>
</tr>
<tr>
<td>cost_h2o_ng_equ</td>
<td>cost for NG hot water system</td>
</tr>
<tr>
<td>cost_h2o_elec_equ</td>
<td>cost for elec hot water system</td>
</tr>
<tr>
<td>cost_h2o_solar_elec_equ</td>
<td>cost for solar with elec backup</td>
</tr>
<tr>
<td>cost_h2o_solar_ng_equ</td>
<td>cost for solar with NG backup</td>
</tr>
<tr>
<td>ng_water_equ</td>
<td>Nat gas cost for water heating</td>
</tr>
<tr>
<td>ng_solar_equ</td>
<td>Nat gas cost for solar hotwater</td>
</tr>
<tr>
<td>ng_furn_equ</td>
<td>Nat gas cost for space heating</td>
</tr>
<tr>
<td>ng_fuel_cost_equ</td>
<td>Total cost of NG fuel</td>
</tr>
<tr>
<td>one_h2o_tech</td>
<td>&quot;can only choose one technology to implement&quot;</td>
</tr>
</tbody>
</table>
*budget_equ limitation on budget

max_solar_equ maximum amount of solar installed
max_wind_equ maximum amount of wind installed

*Start Heating and Cooling Section

one_window_tech only one technology allowed for each section
one_wall_tech
one_ceiling_tech
one_floor_tech

*one_hvac_run only one cooling and heating technology allowed

*one_hvac_run

*geo_const_equ constraint on geothermal: choose for both heating and cooling

*elec_pump_const_equ constraint on electric h.pump: choose for both heating and cooling

floor_loss_equ(t,r_value) r-values to select
window_loss_equ(t,r_value) " "
wall_loss_equ(t,r_value) " "
ceiling_loss_equ(t,r_value) " "

floor_gain_equ(t,r_value) r-values to select
window_gain_equ(t,r_value) " "

142
wall_gain_equ(t, r_value)  " " "
ceiling_gain_equ(t, r_value)  " " "
heat_loss_equ(t)  amount of heat loss
heat_gain_equ(t)  amount of heat gain
tot_heat_loss_equ  total heat loss
tot_heat_gain_equ  total heat gain
c02_emiss_equ  amount of CO2 emitted to heat and cool house
c02_ng_heat_equ  CO2 emissions from natural gas heating
insul_cost_equ  equation for finding cost of insulation
h_cost_equ  equ for heating equipment
c_cost_equ  equ for cooling equipment
h_cost_equ1  equ for heating equipment
c_cost_equ1  equ for cooling equipment
h_cost_equ2  equ for heating equipment
c_cost_equ2  equ for cooling equipment
h_cost_equ3  equ for heating equipment
c_cost_equ3  equ for cooling equipment
solar_incen_equ 
    Tax incentive for solar installation
wind_incen_equ 
    Tax incentive for wind installation
geo_incen_equ1 
    Tax incentive for geothermal heating installation
geo_incen_equ2 
    Tax incentive for geothermal cooling installation

maxsolar_incen_equ 
    Max tax incentive for solar installation
maxwind_incen_equ 
    Max tax incentive for wind installation
maxgeo_incen_equ 
    Max tax incentive for geothermal installation

state_solar_incen_equ 
    state solar incentive
state_wind_incen_equ 
    state wind incentive
state_geo_incen1_equ 
    state geo heating incentive
state_geo_incen2_equ 
    state geo cooling incentive

;

*****
BEGINNING OF MODEL EQUATION FORMULATIONS
*
****

obj func for net metering
overall_cost_equ .. overall_cost =e= budget;

144
total electricity purchased

e_bought .. e_elec = sum(t, e_p(t));

e_bought .. e_elec = sum(t, e_p(t) + e_p1(t) + e_p2(t) + e_p3(t));

for net metering .. electricity produced and bought must == house demand

demand_elec(t, hvac_run('geo_pump')) .. e_p(t) = e(((heat_loss(t)/(3600*heat_eff('geo_pump'))) + (heat_gain(t)/(3600*cool_eff('geo_pump'))) + home_demand(t) + ((e_water(t)/3600)*tech_h2o('elec'))) + ((solar_water_heating(t)/3600)*tech_h2o('solar_elec'))) - (x('solar')*solar_energy(t) + x('wind')*wind_energy(t))) * hvac_vec(hvac_run); // Use only if choose something when electricity is required for heating and/or cooling

demand_elec1(t, hvac_run('elec_pump')) .. e_p1(t) = e(((heat_loss(t)/(3600*heat_eff('elec_pump'))) + (heat_gain(t)/(3600*cool_eff('elec_pump'))) + home_demand(t) + ((e_water(t)/3600)*tech_h2o('elec'))) + ((solar_water_heating(t)/3600)*tech_h2o('solar_elec'))) - (x('solar')*solar_energy(t) + x('wind')*wind_energy(t))) * hvac_vec(hvac_run);

demand_elec2(t, hvac_run('ng_furn')) .. e_p2(t) = e((heat_gain(t)/(3600*cool_eff('AC_central'))) + home_demand(t) + ((e_water(t)/3600)*tech_h2o('elec'))) + ((solar_water_heating(t)/3600)*tech_h2o('solar_elec'))) - (x('solar')*solar_energy(t) + x('wind')*wind_energy(t))) * hvac_vec(hvac_run); // gap is due to
missing heating electricity costs but factors in NG fuel costs

demand_elec3(t, hvac_run('elec_rad')) ..
e_p3(t) = e= (((heat_loss(t)/(3600*heat_eff('elec_rad'))) + (heat_gain(t)/(3600*cool_eff('AC_central'))) + home_demand(t) + ((e_water(t)/3600)*tech_h2o('elec')) + ((solar_water_heating(t)/3600)*tech_h2o('solar_elec'))) - (x('solar')*solar_energy(t) + x('wind')*wind_energy(t))) * hvac_vec(hvac_run);

* Cost for heating and cooling equipment

h_cost_equ(hvac_run('geo_pump')) ..
h_cost = e= (((heat_cost('geo_pump') - geo_incen1) * ((int_rate)/(1 - (1 + int_rate)**(-heat_life('geo_pump') *365))))*30)* hvac_vec(hvac_run);

c_cost_equ(hvac_run('geo_pump')) ..
c_cost = e= (((cool_cost('geo_pump') - geo_incen2) * ((int_rate)/(1 - (1 + int_rate)**(-cool_life('geo_pump') *365))))*30)* hvac_vec(hvac_run);

h_cost_equ1(hvac_run('elec_pump')) ..
h_cost1 = e= (((heat_cost('elec_pump')) * ((int_rate)/(1 - (1 + int_rate)**(-heat_life('elec_pump') *365))))*30)* hvac_vec(hvac_run);

c_cost_equ1(hvac_run('elec_pump')) ..
c_cost1 = e= (((cool_cost('elec_pump')) * ((int_rate)/(1 - (1 + int_rate)**(-cool_life('elec_pump') *365))))*30)* hvac_vec(hvac_run);

h_cost_equ2(hvac_run('ng_furn')) ..
h_cost2 = e= (((heat_cost('ng_furn')) * ((int_rate)/(1 - (1 +
int_rate)**(-heat_life('ng_furn')*365))))*30)*hvac_vec(hvac_run);

c_cost_equ2(hvac_run('AC_central')); c_cost2 =e= (((
cool_cost('AC_central')) * ((int_rate) /(1-((1+
int_rate)**(-cool_life('AC_central')*365)))))))*30)*hvac_vec(hvac_run);

h_cost_equ3(hvac_run('elec_rad')); h_cost3 =e= (((
heat_cost('elec_rad')) * ((int_rate) /(1-((1+
int_rate)**(-heat_life('elec_rad')*365)))))))*30)*hvac_vec(hvac_run);

c_cost_equ3(hvac_run('AC_central')); c_cost3 =e= (((
cool_cost('AC_central')) * ((int_rate) /(1-((1+
int_rate)**(-cool_life('AC_central')*365)))))))*30)*hvac_vec(hvac_run);

* .. fuel

********** NOTE: must be changed if using 'NG_furn' otherwise numbers will be off!!!

ng_water_equ .. ng_fuel_water=e= (ng_cost* sum(t,ng_water(t)))*tech_h2o('ng'));

ng_solar_equ .. ng_fuel_solar=e= (ng_cost* sum(t,
solar_water_heating(t))*tech_h2o('solar_ng'));

ng_furn_equ(hvac_run('ng_furn')) .. ng_fuel_furn =e= (ng_cost* sum(t,heat_loss(t))/heat_eff('ng_furn'))*hvac_vec(hvac_run); // 0;// 'used if NG_furn is selected for heating system'
ng_fuel_cost_equ .. ng_fuel_cost =e= ((ng_fuel_solar + 
ng_fuel_water + ng_fuel_furn )*30) /days;

* Breakdown of electricity usage

home_demand_equ .. demand_home =e= sum(t, home_demand(t));
e_water_demand_equ .. e_water_demand =e= sum(t, ((e_water(t) 
/3600))\*tech_h2o('elec '));
solar_e_demand_equ .. solar_e_demand =e= sum(t, ((
solar_water_heating(t)/3600))\*tech_h2o('solar_elec '));

* just for curiosity – figuring how much is spent on

electricity for heating and cooling

heating_demand_equ(hvac_run('geo_pump')) .. heating_demand =e= 
(sum(t, (heat_loss(t)/(3600*heat_eff('geo_pump'))))) \* 
hvac_vec(hvac_run) ; //the 3600 allows the numbers to be
in electricity

cooling_demand_equ(hvac_run('geo_pump')) .. cooling_demand =e= 
(sum(t, (heat_gain(t)/(3600*cool_eff('geo_pump'))))) \* 
hvac_vec(hvac_run) ; //the 3600 allows the numbers to
be in electricity

heating_demand_equ1(hvac_run('elec_pump')) .. heating_demand1 =e= 
(sum(t, (heat_loss(t)/(3600*heat_eff('elec_pump'))))) \* 
hvac_vec(hvac_run) ;
cooling_demand_equ1(hvac_run('elec_pump')) .. cooling_demand1 =e= 
(sum(t, (heat_gain(t)/(3600*cool_eff('elec_pump'))))) \* 
hvac_vec(hvac_run) ;
heating_demand_equ2  ..  heating_demand2 \equiv 0；

//No electricity when using NG [old equ: sum(t, (heat_loss(t) / (3600*heat_eff('ng_furn'))))];
cooling_demand_equ2(hvac_run('AC_central'))  ..  cooling_demand2 \equiv (sum(t, (heat_gain(t)/(3600*cool_eff('AC_central'))))) * 
hvac_vec(hvac_run) ;
heating_demand_equ3(hvac_run('elec_rad'))  ..  heating_demand3 \equiv (sum(t, (heat_loss(t)/(3600*heat_eff('elec_rad'))))) * 
hvac_vec(hvac_run) ;
cooling_demand_equ3(hvac_run('AC_central'))  ..  cooling_demand3 \equiv (sum(t, (heat_gain(t)/(3600*cool_eff('AC_central'))))) * 
hvac_vec(hvac_run) ;

* info on renewable energy produced
solar_prod_equ ..  solar_en \equiv sum(t,x('solar')*solar_energy(t));
wind_prod_equ ..  wind_en \equiv sum(t,x('wind')*wind_energy(t));

*** ALL COST EQUATIONS!
* electricity cost ....

for 30 days
\[ \text{electricity\_cost\_equ} \quad \text{electricity\_cost} = \frac{(e\_elec \ast \text{cost\_purchased}) \ast 30}{\text{days}}; \]

* 

for budget constraint

\[ \text{npv\_elec\_solar} \quad \text{budget\_solar} = \frac{((x('\text{solar'}) \ast \text{cap\_cost('solar')} - \text{solar\_incen}) \ast (\text{int\_rate}) / (1 - ((1 + \text{int\_rate})^{(-\text{renew\_life('solar' \ast 365))}})) \ast 30) + (x('\text{solar'})) \ast \text{fixom('solar')} \ast (30/\text{days}) + (x('\text{solar'})) \ast \text{varom('solar')} \ast (30/\text{days}) \ast \text{sum(t, solar\_energy(t)))}}{\text{}}; \]

\[ \text{npv\_elec\_wind} \quad \text{budget\_wind} = \frac{((x('\text{wind'})) \ast \text{cap\_cost('wind')} - \text{wind\_incen}) \ast (\text{int\_rate}) / (1 - ((1 + \text{int\_rate})^{(-\text{renew\_life('wind' \ast 365))}})) \ast 30) + (x('\text{wind'})) \ast \text{fixom('wind')} \ast (30/\text{days}) + (x('\text{wind'})) \ast \text{varom('wind')} \ast (30/\text{days}) \ast \text{sum(t, wind\_energy(t)))}}{\text{}}; \]

\[ \text{npv\_elec} \quad \text{budget} = g = \text{electricity\_cost} + \text{ng\_fuel\_cost} + \text{budget\_solar} + \text{budget\_wind} + \text{cost\_h2o} + \text{insul\_cost} + \text{c\_cost} + \text{h\_cost} + \text{c\_cost1} + \text{h\_cost1} + \text{c\_cost2} + \text{h\_cost2} + \text{h\_cost3}; \]

* 

constraint on total monthly cost – can only spend

...(now a monthly amount)

\[ \text{budget\_equ} \quad \text{overall\_cost} = l = 400; \quad // \]

\[ \left(100000 \ast (1 + \text{int\_rate})^{\ast \text{(life30)}} \ast (\text{life30}) \right) * 30; \]

* 

TOTAL COST FOR HOT H2O

150
\[ \text{cost_h2o_equ} \quad \text{cost_h2o} = \text{cost_h2o_ng} + \text{cost_h2o_elec} + \text{cost_h2o_solar_elec} + \text{cost_h2o_solar_ng}; \]

\* hot water equations.. cost for equipment

\[ \text{cost_h2o_ng_equ} \quad \text{cost_h2o_ng} = \text{capcost_h2o} ('ng') \text{tech_h2o ('ng') } \times ( (\text{int_rate})/(1-((1 + \text{int_rate})**((-\text{water_life ('ng')})*365)))) \times 30; \]

\[ \text{cost_h2o_elec_equ} \quad \text{cost_h2o_elec} = \text{capcost_h2o ('elec') \text{tech_h2o ('elec') } \times ( (\text{int_rate})/(1-((1 + \text{int_rate})**((-\text{water_life ('elec')})*365)))) \times 30; \]

\[ \text{cost_h2o_solar_elec_equ} \quad \text{cost_h2o_solar_elec} = ((\text{capcost_h2o ('solar_elec') } \times ( (\text{int_rate})/(1-((1 + \text{int_rate})**((-\text{water_life ('solar_elec')})*365)))) \times 30) + (\text{capcost_h2o ('elec') } \times ( (\text{int_rate})/(1-((1 + \text{int_rate})**((-\text{water_life ('elec')})*365)))) \times 30) \times \text{tech_h2o ('solar_elec');} \]

\[ \text{cost_h2o_solar_ng_equ} \quad \text{cost_h2o_solar_ng} = ((\text{capcost_h2o ('solar_ng') } \times ( (\text{int_rate})/(1-((1 + \text{int_rate})**((-\text{water_life ('solar_ng')})*365)))) \times 30) + (\text{capcost_h2o ('ng') } \times ( (\text{int_rate})/(1-((1 + \text{int_rate})**((-\text{water_life ('ng')})*365)))) \times 30) \times \text{tech_h2o ('solar_ng');} \]

\*

\* Cost of insulation for house

\[ \text{insul_cost_equ} \quad \text{insul_cost} = \text{sum(r_value, (floor_area } \times \text{floor(r_value)} \times \text{floor_cost(r_value)}) + (\text{wall(r_value)} \times \text{wall_area} \times \text{wall_cost(r_value)}) + (\text{ceiling_area} \times \text{ceiling(r_value)} \times \text{ceiling_cost( r_value)}) + (\text{window_area} \times \text{window(r_value)} \times \text{window_cost(r_value))}); \]

151
* \((\text{int\_rate})/(1-(\text{int\_rate})\times(\text{life}30))) \times 30\; ;

* Constraints on INCENTIVES for Renewable Energy Systems

\begin{align*}
solar\_incen\_equ & \quad \text{solar\_incen} = l = ((x1(\text{'solar'}) + x2(\text{'solar'})) \times \text{cap\_cost(\text{'solar'}) \times federal\_tax} + \text{state\_solar\_incen}; \\
wind\_incen\_equ & \quad \text{wind\_incen} = l = ((x1(\text{'wind'}) + x2(\text{'wind'})) \times \text{cap\_cost(\text{'wind'}) \times federal\_tax} + \text{state\_wind\_incen}; \\
geo\_incen\_equ1(\text{hvac\_run(\text{'geo\_pump'})}) & \quad \text{geo\_incen1} = e = ((\text{heat\_cost(\text{'geo\_pump'}) \times federal\_tax} + \text{state\_geo\_incen1}) \times \text{hvac\_vec(\text{hvac\_run})}; \\
geo\_incen\_equ2(\text{hvac\_run(\text{'geo\_pump'})}) & \quad \text{geo\_incen2} = e = ((\text{cool\_cost(\text{'geo\_pump'}) \times federal\_tax} + \text{state\_geo\_incen2}) \times \text{hvac\_vec(\text{hvac\_run})}; \\
*\text{thermel\_incen\_equ} & \quad \text{thermel\_incen} = e = ((\text{capcost\_h2o(\text{'elec'}) + capcost\_h2o(\text{'solar\_elec'}) \times federal\_tax}) + \text{state\_thermel\_incen}; \\
*\text{thermng\_incen\_equ} & \quad \text{thermng\_incen} = e = ((\text{capcost\_h2o(\text{'ng'}) + capcost\_h2o(\text{'solar\_ng'}) \times federal\_tax}) + \text{state\_thermng\_incen}; \\
\*\text{maximum amount of incentives at state level} \quad \text{***no cap for federal!} \\
max\text{solar\_incen\_equ} & \quad \text{state\_solar\_incen} = l = 10500; \\
max\text{wind\_incen\_equ} & \quad \text{state\_wind\_incen} = l = 10500; \\
max\text{geo\_incen\_equ} & \quad \text{state\_geo\_incen1 + state\_geo\_incen2} = l = 8400;
\end{align*}
*maxthermel_incen_equ.. state_thermal_incen =l= 1400;
*maxthermng_incen_equ.. state_thermng_incen =l= 1400;

* formulation of state incentives for cap
state_solar_incen_equ .. state_solar_incen =l= x1('solar')*
cap_cost('solar')*state_tax;
state_wind_incen_equ .. state_wind_incen =l= x1('wind') *
cap_cost('wind')*state_tax;
state_geo_incen1_equ(heat_tech('geo.pump')) .. state_geo_incen1 =e=
heat_tech(heat_tech)*state_tax;
state_geo_incen2_equ(cool_tech('geo.pump')) .. state_geo_incen2 =e=
cool_tech(cool_tech)*state_tax;
*state_thermal_incen_equ .. state_thermal_incen =e= (capcost_h2o('elec')+capcost_h2o('solar_elec'))*state_tax;
*state_thermng_incen_equ .. state_thermng_incen =e= (capcost_h2o('ng') +capcost_h2o('solar_ng')) *state_tax;

* CO2 – carbon emissions constraint
co2_emiss_equ .. co2_emissions =e= sum(t, (e_p(t)+e_p1(t)+
e_p2(t)+e_p3(t))*srvc_emissions_renew(t)) + sum(t,ng_water(t) *
ng_emissions *tech_h2o('ng')) + sum(t,solar_water_heating(t) *
ng_emissions * tech_h2o('solar_ng')) + co2_ng_heat;

co2_ng_heat_equ(hvac_run('ng.furn')) .. co2_ng_heat =e= ((
ng_fuel_furn/ng_cost)*ng_emissions)*hvac_vec(hvac_run);
*carbon_total_equ .. co2_emissions =l= 4000;
* max renewable capacity based on roof area and wind size

max_solar_equ .. x('solar') =l= 7;
max_wind_equ .. x('wind') =l= 10;

x_equ(tech) .. x(tech) =e= x1(tech) + x2(tech);
x1_solar_equ .. x1('solar') =l= 3.947;
x1_wind_equ .. x1('wind') =l= 7.5;
x2_solar_equ .. x2('solar') =l= 3.053;
x2_wind_equ .. x2('wind') =l= 2.5;

*** EQUATIONS FOR ONE TECHNOLOGY CONSTRAINT

* one technology
one_h2o_tech .. 1 =e= sum(h2o_tech, tech_h2o(h2o_tech));

* Only one choice per part of house
one_window_tech .. 1 =e= sum(r.value, window(r.value));
one_wall_tech .. 1 =e= sum(r.value, wall(r.value));
one_ceiling_tech .. 1 =e= sum(r.value, ceiling(r.value));
one_floor_tech .. 1 =e= sum(r.value, floor(r.value));

* Only one choice per heating and cooling technology
*one_hvac_run .. 1 =e= sum(hvac_run, cooling(hvac_run));
*one_hvac_run .. 1 =e= sum(hvac_run, heating(hvac_run));

* Constraint tying both techs to heating and cooling
*geo_const_equ .. 0 =e= heating('geo_pump') - cooling('geo_pump');
*elec_pump_const_equ.. 0 =e= heating('elec_pump')- cooling('elec_pump');

*** EQUATIONS FOR CALCULATING HEAT LOSS AND HEAT GAIN [kJ]

* heat loss due to cold weather [kJ]

floor_loss_equ(t,r_value) .. floor_loss(t,r_value) =e= (((
  floor_area*abs(ambient_temp_cold(t)-inside_temp_win))/(floor_insul(r_value)*r_conversion))*0.9)*floor(r_value);
//0.9 to convert watts to kJ over the 15 minute interval

window_loss_equ(t,r_value) .. window_loss(t,r_value) =e= (((
  window_area*abs(ambient_temp_cold(t)-inside_temp_win))/(window_insul(r_value)*r_conversion))*0.9)*window(r_value);

wall_loss_equ(t,r_value) .. wall_loss(t,r_value) =e= (((
  wall_area*abs(ambient_temp_cold(t)-inside_temp_win))/(wall_insul(r_value)*r_conversion))*0.9)*wall(r_value);

ceiling_loss_equ(t,r_value) .. ceiling_loss(t,r_value) =e= (((
  ceiling_area*abs(attic_temp_cold(t)-inside_temp_win))/(ceil_insul(r_value)*r_conversion))*0.9)*ceiling(r_value);

* heat gain from hot weather [kJ]

floor_gain_equ(t,r_value) .. floor_gain(t,r_value) =e= (((
  floor_area*abs(ambient_temp_hot(t)-inside_temp_sum))/(floor_insul(r_value)*r_conversion))*0.9)*floor(r_value);
//0.9 to convert watts to kJ over the 15 minute interval
window_gain_equ(t, r_value) .. window_gain(t, r_value) =e= (((window_area*abs(ambient_temp_hot(t)-inside_temp_sum))/window_insul(r_value)*r_conversion)) * 0.9) * window(r_value);

wall_gain_equ(t, r_value) .. wall_gain(t, r_value) =e= (((wall_area*(ambient_temp_hot(t)-inside_temp_sum))/(wall_insul(r_value)*r_conversion)) * 0.9) * wall(r_value);

celing_gain_equ(t, r_value) .. ceiling_gain(t, r_value) =e= (((ceiling_area*abs(attic_temp_hot(t)-inside_temp_sum))/(ceiling_insul(r_value)*r_conversion)) * 0.9) * ceiling(r_value);

* Heat loss and heat gain per time interval

heat_loss_equ(t) .. heat_loss(t) =e= sum(r_value, floor_loss(t, r_value) + window_loss(t, r_value) + wall_loss(t, r_value) + ceiling_loss(t, r_value));

heat_gain_equ(t) .. heat_gain(t) =e= sum(r_value, floor_gain(t, r_value) + window_gain(t, r_value) + wall_gain(t, r_value) + ceiling_gain(t, r_value));

* Total heat loss and heat gain

tot_heat_loss_equ .. tot_heat_loss =e= sum(t, heat_loss(t));
tot_heat_gain_equ .. tot_heat_gain =e= sum(t, heat_gain(t));

$onlisting

model total_home_energy / all /;
*solve total_home_energy using mip min overall_cost;
*solve total_home_energy using mip min co2_emissions;
D.4 GAMS Optimization Outer Shell Code

The following set of code runs multiple sets of the above “inner” shell depending on what HVAC option is selected and what is being optimized.

******************************
$eolcom //
 Scalars

col 'column number' /1/
row 'row number' /1/

Sets
runs different runs / run1* run4 / //
hvac_run tech for heating and cooling /geo_pump,
        elec_pump, ng_furn, elec_rad, AC_central/

** Escape from running heating and cooling as variables
table hvac_mat(runs, hvac_run) running with geothermal as choice
        geo_pump elec_pump ng_furn
        elec_rad AC_central
run1    1     0     0
        0     0     0
run2 0 1 0
0 0
run3 0 0 1
0 1
run4 0 0 0
1 1

parameter hvac_vec(hvac_run);

*now want to run initial optimization with original objective function
** NOTE: this include file only has information to run obj. not the solve statement, needs to be run separately below 'hvac_mat' to get proper technologies

$include total_home_energy_min.inc

file opt_values_geo /opt_values_min_co2.10-12-10.csv/;

opt_values_geo.ap = 0;
put opt_values_geo;
col = 1; row=1;
put 'Overall Monthly Cost', ', ', / ;
put 'CO2 Emissions', ', ', / /;
put 'Installed Solar Capacity', ', ', / ;
put 'Installed Wind Capacity', ', ', / /;
loop (h2o_tech, put 'Hot Water: ' h2o_tech.tl, ', ', 
   / ) put /;
loop (insul_tech, put 'Insulation: ' insul_tech.tl;
   put ', ',
   
   loop (r_value, put @22 r_value.tl, put @26 ', ', /
       ) ; put /
   
   );

loop (hvac_run, put hvac_run.tl, ', ', / ) put / ;

opt_values_geo.ap = 1;

loop (runs,
    hvac_vec(hvac_run) = hvac_mat(runs, hvac_run);
    col = col +26;

    solve total_home_energy using mip min co2_emissions
    ;

    put @col#row overall_cost.l, ', ', / ; row = row +
    1;

160
put @col#row co2_emissions.l, ',' /; row = row + 2;
put @col#row x.l('solar'), ',' /; row = row + 1;
put @col#row x.l('wind'), ',' /; row = row + 2;

loop(h2o_tech, put @col#row
tech_h2o.l(h2o_tech), ',' /; row = row + 1;)
put /; row = row+1;

loop(r_value, put @col#row
wall.l(r_value), ',' /; row = row + 1;)
; row = row +1;
loop(r_value, put @col#row
floor.l(r_value), ',' /; row = row + 1;)
; row = row +1;
loop(r_value, put @col#row
window.l(r_value), ',' /; row = row + 1;)
; row = row +1;
loop(r_value, put @col#row
ceiling.l(r_value), ',' /; row = row + 1;)
put /; row = row +1;
loop(hvac_run, put @col#row
hvac_mat(runs,hvac_run), ',' /; row = row + 1;)
put /;
row = 1;

display solar_incen.1, wind_incen.1,
    state_solar_incen.L, state_wind_incen.L;

display "*** Chosen variables by program ";
display x.1, x1.1, x2.1, tech_h2o.1, window.1, wall
    .1, ceiling.1, floor.1;

display "*** CO2 emissions ";
display co2_emissions.1, co2_emissions.m,
    co2_ng_heat.1; //

display "*** Costs added to budget ";
display overall_cost.1, electricity_cost.1,
    cost_h2o.1, insul_cost.1, budget_wind.1,
    budget_solar.1, ng_fuel_cost.1;

display "*** NG Fuel costs ";
display ng_fuel_water.1, ng_fuel_solar.1,
    ng_fuel_furn.1;

display "*** Total Heat and Cooling Demand ";
display tot_heat_loss.1, tot_heat_gain.1;

display "*** Fuel and Electricity Demands ";

162
display demand_home.1, e_water_demand.1,
    solar_e_demand.1, heating_demand.1,
    cooling_demand.1, heating_demand1.1,
    cooling_demand1.1, heating_demand2.1,
    cooling_demand2.1, heating_demand3.1,
    cooling_demand3.1, e_elec.1;

display "*** Electricity Created ";
display solar_en.1, wind_en.1;

};
putclose opt_values_geo;
D.5 MGA Outer Shell

This section of code runs the outer MGA shell. It uses the core CO₂ optimization code but then loops multiple times over different budget constraints. The code must be modified according to what is being solved (CO₂ emissions or cost and slack parameters). The provided example is based on a Natural Gas heating system, medium cost ($308), and 10% slack to the cost. (Note, the spacing has been modified in order to better fit on the printed page. There is a chance some of the code may have been deleted. Please refer to the original code for a complete version.

Note: The spacing has been modified in order to better fit on the printed page. There is a chance some of the code may have been deleted. Please refer to the original code for a complete version.
table hvac_mat1(rund, hvac_run) \ running with geothermal as choice

<table>
<thead>
<tr>
<th>geo_pump</th>
<th>elec_pump</th>
<th>ng_furn</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>elec_rad</td>
<td>AC_central</td>
<td></td>
</tr>
<tr>
<td>run1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>run2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>run3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>run4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

** Escape from running heating and cooling as variables

table hvac_mat(runs, hvac_run) running with geothermal as choice

<table>
<thead>
<tr>
<th>geo_pump</th>
<th>elec_pump</th>
<th>ng_furn</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>elec_rad</td>
<td>AC_central</td>
<td></td>
</tr>
<tr>
<td>run3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

parameter
hvac_vec(hvac_run)
window_coeff(r_value)
wall_coeff(r_value)
floor_coeff(r_value)
ceil_coeff(r_value)
water_coeff(h2o_tech)

*for total MGA - 2nd runs and beyond
hvac_vec1(hvac_run)
window_coeff_a(r_value)
wall_coeff_a(r_value)
floor_coeff_a(r_value)
ceil_coeff_a(r_value)
water_coeff_a(h2o_tech)

;               //
*now want to run initial optimization with original objective function
**        NOTE: this include file only has information to run obj.
        not the solve statement, needs to be run separately below 'hvac_mat' to get proper technologies

$include total_home_energy.inc

file   opt_values_elec   /MGA_ng_308cost_10slack.txt/;
        opt_values_elec.ap = 0;
        put opt_values_elec;
        col = 1; row=1;
* put @col#row 'Heating and Cooling Techs'; put ', '; col = col+35;
* loop (runs, put @col runs.tl, ', '; col=col+22) put ;
* col = 1;

put 'Overall Monthly Cost', ', '/ ;
put 'Budget', ', '/ ;
put 'CO2 Emissions', ', '/ ;
* put 'CO2 Target', ', '/ ;
put 'MGA Dec. Var', ', '/ ;
put 'Installed Solar Capacity', ', '/ ;
put 'Installed Wind Capacity', ', '/ ;
loop (h2o_tech, put 'Hot Water: ' h2o_tech.tl, ', '/ ) put / ;
loop (insul_tech, put 'Insulation: ' insul_tech.tl;
put ', ',

loop (r_value, put @22 r_value.tl, put @26 ', ',/ ) ; put /
);

loop (hvac_run, put hvac_run.tl, ', '/ ) put / ;
opt_values_elec_ap = 1;

* initializing coefficients for MGA
window_coeff(r_value) = 0;
wall_coeff(r_value) = 0;
floor_coeff(r_value) = 0;
ceil_coeff(r_value) = 0;

* solar_coeff = 0;
* wind_coeff = 0;
water_coeff(h2o_tech) = 0;

*for further MGA runs

window_coeff_a(r_value) = 0;
wall_coeff_a(r_value) = 0;
floor_coeff_a(r_value) = 0;
ceil_coeff_a(r_value) = 0;
water_coeff_a(h2o_tech) = 0;

* wind_coeff_a = 0;
* solar_coeff_a = 0;

loop(runs,
    hvac_vec(hvac_run) = hvac_mat(runs, hvac_run);
col = col +24;

    solve total_home_energy using mip min co2_emissions ;

    * Set Constraint for CO2 Emissions based on min value from above, then relax cost constraint and resolve
    * co2_target = co2_emissions.l;
put @col#row overall_cost.l, ',' / ; row = row + 1;
put @col#row budget.l, ',' / ; row = row + 1;
put @col#row co2_emissions.l, ',' / ; row = row + 1;
    put @col#row de_var.l, ',' / / ; row = row + 2;
put @col#row x.l('solar'), ',' / ; row = row + 1;
put @col#row x.l('wind'), ',' / / ; row = row + 2;

loop (h2o_tech, put @col#row
tech_h2o.l(h2o_tech), ',' / ; row = row + 1;)
    put / ; row = row + 1;

loop (r_value, put @col#row
wall.l(r_value), ',' / ; row = row + 1;)
    ; row = row + 1;
loop (r_value, put @col#row
floor.l(r_value), ',' / ; row = row + 1;)
    ; row = row + 1;
loop (r_value, put @col#row
window.l(r_value), ',' / ; row = row + 1;)
    ; row = row + 1;
loop ( r_value , put @col#row

    ceiling . l ( r_value ) , ' , ' / ; row = row + 1 ; )

    put / ; row = row +1;

loop ( hvac_run , put @col#row

    hvac_mat ( runs , hvac_run ) , ' , ' / ; row = row + 1 ; )

    put / ;

    row = 1;

display solar_incen . l , wind_incen . l ,

    state_solar_incen . L , state.wind_incen . L ;

display " *** Chosen variables by program " ;
display x . l , x1 . l , x2 . l , tech_h2o . l , window . l , wall

    . l , ceiling . l , floor . l ;

display " *** CO2 emissions " ;
display co2_emissions . l , co2_emissions . m ,

    co2_ng_heat . l ; // . l

display " *** Costs added to budget " ;
display overall_cost . l , electricity_cost . l ,

    cost_h2o . l , insul_cost . l , budget_wind . l ,

    budget_solar . l , ng_fuel_cost . l ;

display " *** NG Fuel costs " ;
display ng_fuel_water.l, ng_fuel_solar.l, ng_fuel_furn.l;

display "*** Total Heat and Cooling Demand ";
display tot_heat_loss.l, tot_heat_gain.l;

display "*** Fuel and Electricity Demands ";

display "*** Electricity Created ";
display solar_en.l, wind_en.l;

*for MGA runs—this will be where the coefficients are updated *
update of coefficients within MGA loop

    window_coef(r_value) = window_coef(r_value) + window.l(r_value);
    wall_coef(r_value) = wall_coef(r_value) + wall.l(r_value);
    floor_coef(r_value) = floor_coef(r_value) + floor.l(r_value);
    ceil_coef(r_value) = ceil_coef(r_value) + ceiling.l(r_value);

171
solar_coeff = solar_coeff + (x.l('solar')/7);

wind_coeff = wind_coeff + (x.l('wind')/10);

water_coeff(h2o_tech) = water_coeff(h2o_tech) + tech_h2o.l(h2o_tech);

window_coeff_a(r_value) = window_coeff(r_value);
wall_coeff_a(r_value) = wall_coeff(r_value);
floor_coeff_a(r_value) = floor_coeff(r_value);
ceil_coeff_a(r_value) = ceil_coeff(r_value);
solar_coeff_a = solar_coeff;
wind_coeff_a = wind_coeff;
water_coeff_a(h2o_tech) = water_coeff(h2o_tech);

hvac_vec(hvac_run) = hvac_mat1('run2', hvac_run);
hvac_vec(hvac_run) = hvac_vec1(hvac_run);

***New section to run minimizing de_var for MGA and summing the
total coefficients used in each run without effecting the latter runs

cost_target = overall_cost.l * 1.1;
budget.l = cost_target;
co2_target = co2_emissions.l;
loop (iter,
    loop (rund,
        hvac_vec(hvac_run) = hvac_mat1(rund, hvac_run);
        col = col + 13;
    )
*)

Reseting the coefficients incase of an infeasible solution

overall_cost.l = 0;
c02_emissions.l = 0;
window.l(r_value)=0;
wall.l(r_value)=0;
floor.l(r_value)=0;
ceiling.l(r_value)=0;
tech_h2o.l(h2o_tech)=0;
x.l(tech)=0;

solve total_home_energy using mip min de_var;

put @col#row overall_cost.l, ' , ' / ; row = row + 1;
put @col#row budget.l, ' , ' / ; row = row + 1;
put @col#row co2_emissions.l, ' , ' / ; row = row + 1;
    put @col#row de_var.l, ' , ' / / ; row = row + 2;
put @col#row x.l('solar'),     ',' / ; row = row + 1;
put @col#row x.l('wind'),     ',' / ; row = row + 2;

loop(h2o_tech,         put   @col#row
tech_h2o.l(h2o_tech),   ',' / ; row = row + 1;
put / ; row = row+1;
loop(r_value,         put   @col#row
    wall .l(r_value),   ',' / ; row = row + 1;
    ; row = row +1;
loop(r_value,         put   @col#row
    floor .l(r_value),   ',' / ; row = row + 1;
    ; row = row +1;
loop(r_value,         put   @col#row
    window .l(r_value),   ',' / ; row = row + 1;
    ; row = row +1;
loop(r_value,         put   @col#row
    ceiling .l(r_value),   ',' / ; row = row + 1;
    put / ; row = row +1;
loop(hvac_run,         put   @col#row
    hvac_mat1(rund ,hvac_run),   ',' / ; row = row + 1;
    put / ;

display solar_incen.l, wind_incen.l,
    state_solar_incen.l, state_wind_incen.l ;
display "***    Chosen variables by program  ";
Display x.1, x1.1, x2.1, tech_h2o.1, window.1, wall.1, ceiling.1, floor.1;
display "*** CO2 emissions ";
display co2_emissions.1, co2_emissions.m, co2_ng_heat.1;
display "*** Costs added to budget ";
display overall_cost.1, electricity_cost.1, cost_h2o.1, insul_cost.1, budget_wind.1, budget_solar.1, ng_fuel_cost.1;
display "*** NG Fuel costs ";
display ng_fuel_water.1, ng_fuel_solar.1, ng_fuel_furn.1;
display "*** Total Heat and Cooling Demand ";
display tot_heat_loss.1, tot_heat_gain.1;
display "*** Fuel and Electricity Demands ";
display demand_home.1, e_water_demand.1, solar_e_demand.1, heating_demand.1, cooling_demand.1, heating_demand1.1, cooling_demand1.1, heating_demand2.1, cooling_demand2.1, heating_demand3.1, cooling_demand3.1, e_elec.1;
display "*** Electricity Created ";
display solar_en.1, wind_en.1;
display "*** MGA Coefficients ";
display window_coeff, wall_coeff, floor_coeff, ceil_coeff, solar_coeff, wind_coeff, water_coeff;
window_coeff_a(r_value) = window_coeff_a(r_value) + window
    .l(r_value);
wall_coeff_a(r_value) = wall_coeff_a(r_value) + wall.l(
    r_value);
floor_coeff_a(r_value) = floor_coeff_a(r_value) + floor.l
    (r_value);
ceil_coeff_a(r_value) = ceil_coeff_a(r_value) + ceiling.
    l(r_value);
* solar_coeff_a = solar_coeff_a + (x.l('solar')/7)
    ;
* wind_coeff_a = wind_coeff_a + (x.l('wind')/10);
water_coeff_a(h2o_tech)= water_coeff_a(h2o_tech) +
    tech_h2o.l(h2o_tech);
    row = 1;
    );
window_coeff(r_value)= window_coeff_a(r_value);
wall_coeff(r_value) = wall_coeff_a(r_value);
floor_coeff(r_value) = floor_coeff_a(r_value);
ceil_coeff(r_value) = ceil_coeff_a(r_value);
* solar_coeff = solar_coeff_a;
* wind_coeff = wind_coeff_a;
water_coeff(h2o_tech)= water_coeff_a(h2o_tech);
    );