ENERGY 2100
Documentation

Volume 2

Demand Module

Last Revision: June 2026

Table of Contents

Table of Tables

Table of Figures

1. Introduction

ENERGY 2100 is an integrated regional, multi-sector energy analysis system that simulates energy supply, price and demand across detailed fuel types. Its energy demand module consists of four sectors (residential, commercial, industrial, and transportation). Energy demands are calculated and sent as input to the supply module consisting of six energy producing sectors – electricity, oil and gas, refinery, hydrogen, biofuels, coal, and steam. The supply module produces the energy required to meet energy demand, calculates energy prices, and returns energy prices as feedback to the demand sector. Both energy and non-energy related emissions are tracked covering eighteen separate greenhouse gas (GHG) pollutants and criteria air contaminants (CAC) plus water usage. This volume of ENERGY 2100 documentation describes ENERGY 2100's demand module – its methodologies, theoretical derivations, and model source code.

1.1. Organization of this Document

This documentation is divided into this introduction plus additional three sections: methodology, model structures, and model code.

- Section 1. Introduction
- Section 2. Methodology
- Section 3. Model Structures
- Section 4. Model Code

The methodology section covers the theoretical principles that the demand module was built on. It also describes the methods used to initialize variables, calibrate model equations to historical data, and projecting calibration variables into the future.

The model structures section provides a look at the model's demand structures and level of granularity, describing the areas, technologies, fuels, and enduses represented along with key variables, inputs, and outputs.

The section on model code provides the routines and equations that make up the demand module in ENERGY 2100.

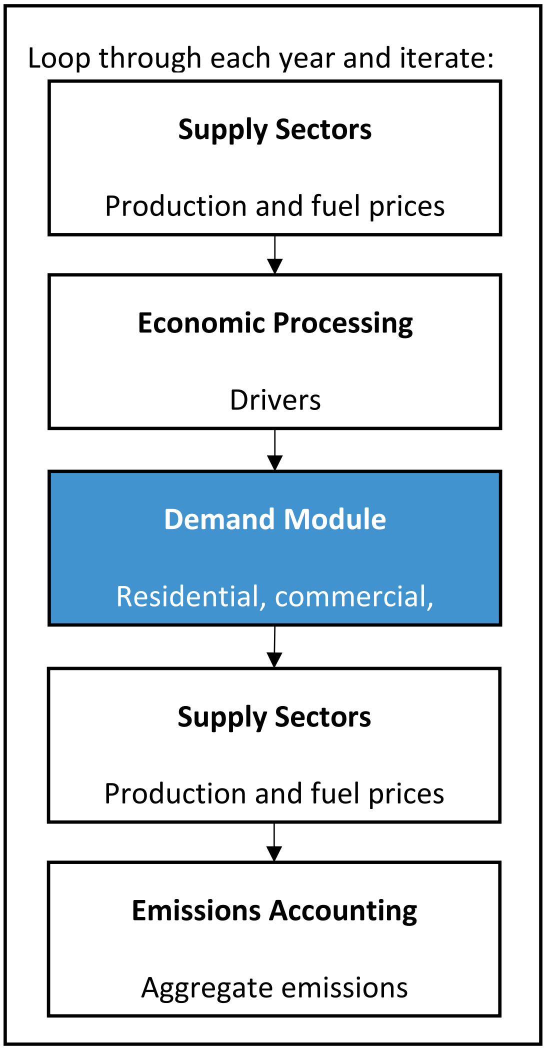
Each of the individual sectors within the demand sector (residential, commercial, industrial, and transportation) use the same basic methodology and have parallel structures, including the variable names. Model code descriptions, including routines, variables, and equations, apply to all four demand sectors.

Note that the terms “demand module” and “demand sector” are used interchangeably to refer to the set of model code in ENERGY 2100 that simulates the energy consuming portion of the energy system (distinguished from energy supply).

1.2. Demand Module Relationship to Rest of Model

For context, a flow diagram of ENERGY 2100’s model execution is illustrated in Figure 1 showing a simplified order of execution of the demand module in relationship to the rest of the model. For each forecast year, ENERGY 2100 performs the operations in the order shown in the flow diagram. Initially, several supply sectors are run to obtain energy prices and energy production levels from the prior year, which are used as inputs to the demand module. Additionally, economic processing is performed to assign the drivers of energy demand. Each demand sector is executed separately (residential, commercial, industrial, and transportation) to calculate enduse and feedstock energy demands. These energy demands are sent to the supply modules which calculate energy production required to meet demand. The resulting energy prices are also calculated within the supply module. Energy and non-energy emissions are tracked throughout and aggregated as a final step. These steps are currently executed twice each forecast year. The second iteration is run to allow feedback from the first iteration’s supply sector results to be incorporated into a second round of demand sector calculations and vice versa. For example, energy prices from the first iteration are updated and input to the demand module during the second iteration.

Figure 1. Relationship to Rest of Model

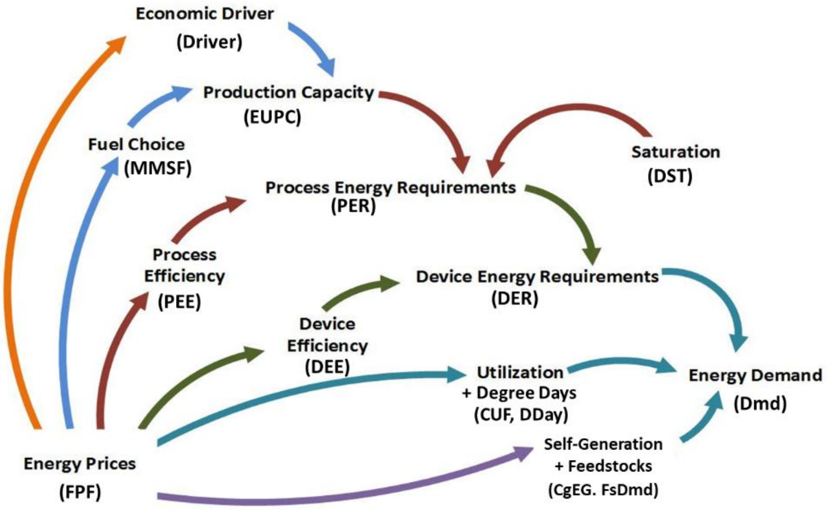


1.3. Overview of Energy Demand

The demand module simulates fuel usage from end-use energy requirements, self-generation, and feedstocks. Figure 2 illustrates the key mechanisms used to build up energy demand.

Variable names are shown in parentheses. ENERGY 2100 simulates energy demand (enduse, self-generation, and feedstock) as a function of device energy requirements combined with utilization factors and accounting for self-generation and feedstocks. Device energy requirements take into account changes in process energy requirements (e.g. building shell) as well as energy efficiency levels of the devices/equipment used to meet the process energy requirements (new and existing). Because ENERGY 2100 tracks vintages of capital stock, changes in device energy requirements and process energy requirements are also due to retirements and replacements of capital stock from wear-out and/or additions to capital stock from economic growth. Process energy requirements are determined based on changes to production capacity (from retirements, replacement, or additions), process efficiency levels of new and existing stock and device saturations. Changes to production capacity are driven by economic drivers combined with consumers' fuel choices for new capital with energy prices generally influencing all aspects of the energy system.

Figure 2. Demand Mechanisms and Key Variables in ENERGY 2100



Theoretical underpinnings of the demand module equations are based in consumer choice theory and system dynamics centering around three key concepts:

1) tracking energy capital stock - retirements, replacements, and additions;
2) projecting consumer choices of marginal energy efficiency based on trade-offs between capital costs and efficiency; and
3) projecting fuel market shares through consumer choices of new capital stock purchases.

2. Demand Module Concepts

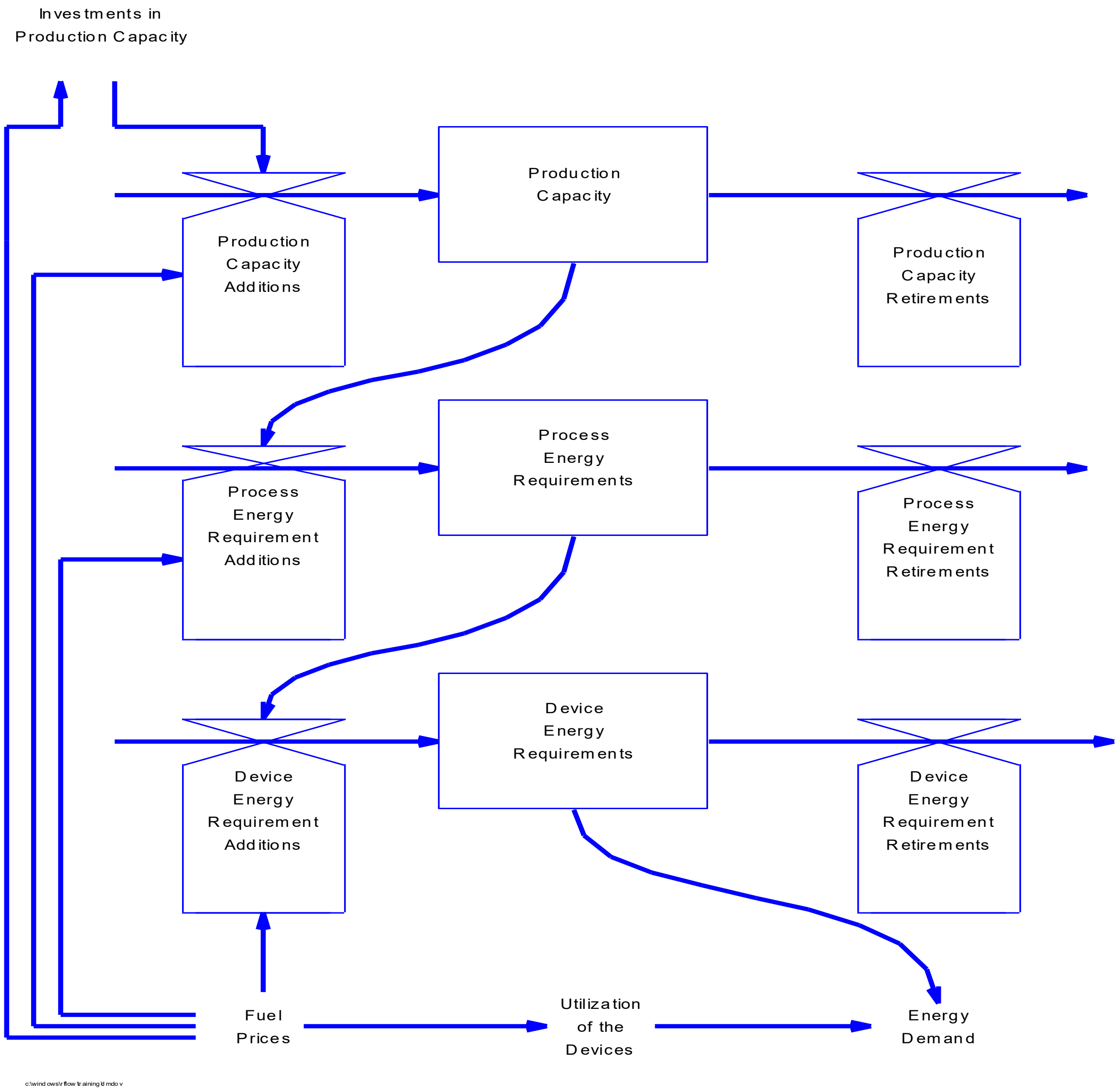
2.1. Energy Requirements

The model simulates the energy economy as a hierarchy of three levels – production capacity, process energy requirements, and device energy requirements. Production capacity is estimated from the economic driver. Process energy refers to the energy required to fulfill the needs of production capacity. Device energy is the energy required to operate machinery that produces the process energy. Annual requirements are simulated using physical units that match the level, such as input Btu of energy needed is the device energy requirement.

At the bottom of Figure 3, energy demand is represented as a function of device energy requirements and the utilization of those devices. Device energy requirements simply consist of the energy required to run existing appliances - space heaters, air conditioning, hot water heaters, lights, refrigerators, etc. Utilization of devices is related to behavior. For example, industrial customers may run their factories for one or several shifts. A plant using energy for process heat and lighting that is run for only one shift will have less energy demand than another plant with the same devices that is run for two shifts. Similarly, residential customers will scale back or increase their use of space heating or air conditioning based on temperature - a cold day will increase space heating demand and a hot one will increase air conditioning needs. For a given stock of devices, energy demand is fixed. What causes variations in demand for a fixed stock of devices are different utilization rates. Energy demand changes as utilization changes.

Distinguishing between levels and rates is important when trying to understand the dynamics in ENERGY 2100. The rectangles (boxes) in Figure 3 represent levels or stocks. Levels exist even if time stands still. If we stop all the changes occurring in the energy demand market, we still have energy requirements for existing stock. Values which have a time component are rates - rates change levels over time. For example, there is an existing level of refrigerators at a given point in time. Adding to that stock are new refrigerators being purchased, and subtracting from that level are old refrigerators being retired. The number of new refrigerators being added and the number retired are referred to in terms of rates - the number in a given period of time. In ENERGY 2100 the time frame is usually one year. Therefore, refrigerator additions would be stated in terms of the device requirement per year - such as 5 TBtu/year. Refrigerator retirements would be stated in exactly the same fashion. Note that it is the amount of energy consumed which is tracked, not the number of refrigerators. ENERGY 2100 integrates rates into levels - Device Energy Requirement additions each year (rate) go into Device Energy Requirements (level). Device Energy Requirement retirements behave in the same fashion. Devices are retired in the model usually when they exceed their normal lifetimes.

Figure 3. Demand Overview



Device energy requirements and device energy additions are a function of process energy requirements as indicated by the arrows in Figure 3. Process energy requirements are determined by the energy service that we need. For example, we need cool air inside a box to keep food fresh so we buy a refrigerator. Or we improve the thermal integrity of our home (by adding insulation or efficient windows) to keep warm air inside our home. For industrial applications, process efficiency would describe how much energy is needed to produce a certain level of output. Therefore, the amount of energy service we need influences the choice of energy devices. The amount of refrigeration we need determines the number and size of the refrigerators we buy. An industrialist's determination of his energy needs determines the size of his factory.

Process energy requirements are a function of production capacity (again refer to the arrows in Figure 3). Industrial process energy is energy required to produce a particular bundle of goods, measured in dollars of output, from existing production capacity. The industrialist needs a certain amount of mechanical energy to produce a product. A production quota for that product determines the process energy requirements which in turn determine the device energy requirements. The number of items produced represents production capacity which is measured in dollars of output. Each dollar of output has a process requirement which generates a device energy requirement. Investment in production capacity is equal to a change in the level of output of goods and services in the economy. What is being measured is not the cost of factories, stores, or homes but the value of what this capital produces. When we talk about $100M capital increase we are talking about increasing the output from capital investment by $100 M, not the cost of the capital investment itself.

The sections below describe the calculation of the stock for each level of production capacity, process energy requirements, and device energy requirements.

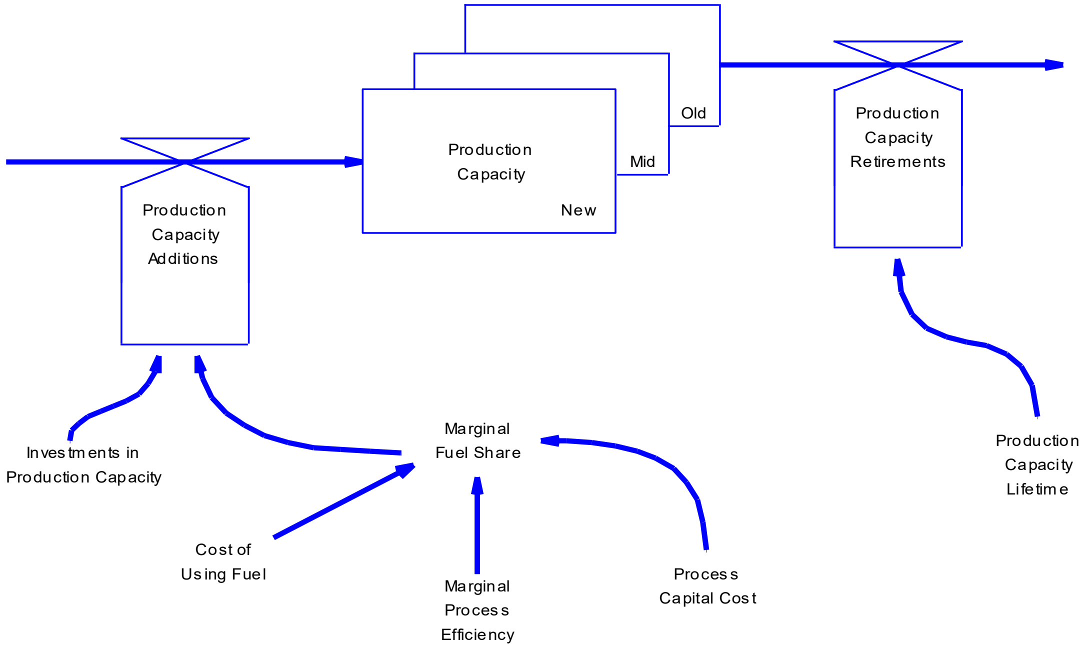
2.2. Production Capacity

Production capacity in ENERGY 2100 is simulated as a function of capital stock derived from the economic driver and a capacity utilization factor. The unit of the production capacity stock is the same as the driver where, for example, stock is expressed in terms of dollars of gross output for sectors that use gross output as the economic driver.

The three rectangles in the center of Figure 4 represent the aging of the capital stock. This aging process uses three "vintages": new, medium and old, to describe the life cycle of capital stock. Production capacity retirements depend on production capacity lifetime (PCPL) and are withdrawn from the third (old) capacity level. Production capacity has the longest lifetime, logically outlasting both process and device lifetimes.

There are several types of delays that could be employed to represent the aging of capital stock. The first is called the pipeline delay – if you add something into the capital "pipeline" today and in 40 years (or however long the lifetime is), it is retired. This is a discrete type of delay and is used for debt retirement but does not describe the aging process of capital stock very well.

Figure 4. Production Capacity



Another type of delay is the first order delay. With this type of delay, a certain percentage of the stock is retired each year – for a lifetime of 40 years, 2.5% would be retired each year. Since 2.5% of all the new stock is greater than 2.5% of say, half the stock, the biggest chunk of retirements happens the first year after the stock is added. Again, this does not seem to be a good description of the aging process for capital stock; however, it does produce the desired pattern of retirements for devices. The most likely time for devices to fail is the first year of purchase, subsequent years have fewer retirements. Therefore, the first order lag is used for devices but not capital stock.

A third type of delay, and the one selected to represent capital stock aging, is the third order delay. Here new investment goes into the first phase (box) for thirteen years (assuming a forty-year lifetime), moves to the next box for thirteen years and spends the remainder in the last box. This type of delay produces a shape that best mimics the flow of retirements - far more diffuse than the pipeline delay but without the inappropriate front loading of retirements found in a first order delay. The graphs below illustrate the shape of retirement functions from pipeline, first and third order delays resulting from an investment “spike” (if investment were constant every year, all delays would yield the same retirement patterns).

Production Capacity Additions are a function of investments in Production Capacity and marginal fuel shares. The total additions to production capacity are split into fuel shares using the Marginal Fuel Share Fraction. For example, if twenty percent of the total Production Capacity will be generated with electricity, then 20% times the total Production Capacity (in dollars of output) will be new electric demand.

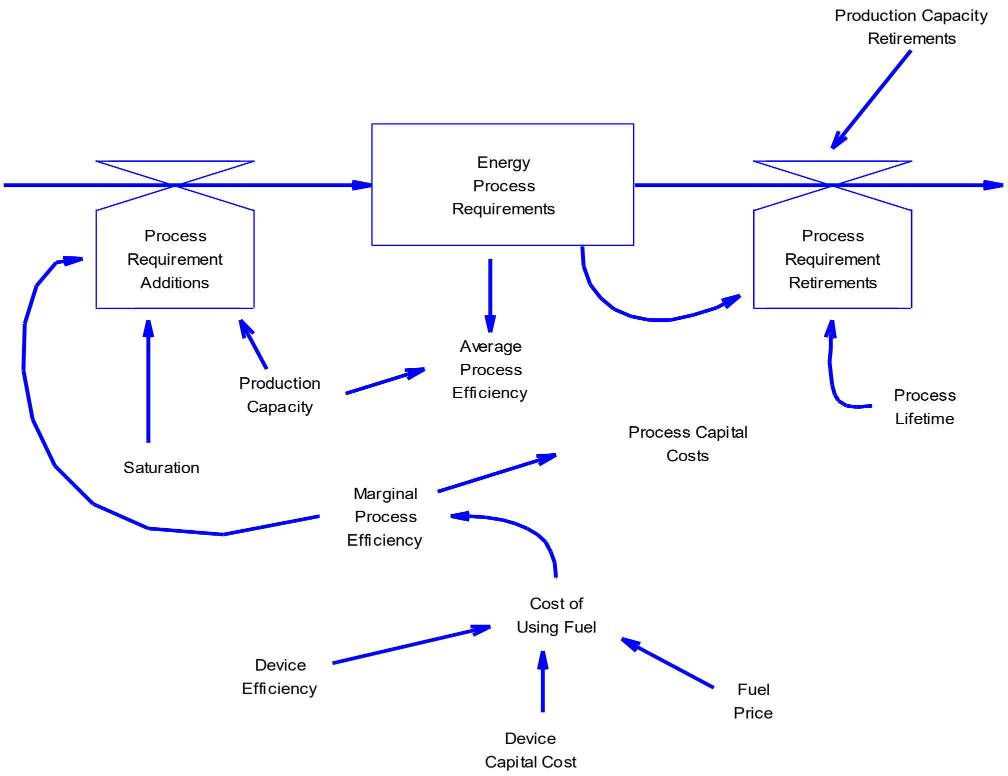
The Marginal Fuel Share fraction is a function of both price and non-price factors. The price factors are the marginal cost of using fuel, the marginal process efficiency and the process capital cost. The marginal cost of using fuel implies getting a certain BTU level of process efficiency out of the device (for example, cubic feet of heated air). The Process Efficiency is concerned with how much hot air we need to achieve those cubic feet of heated air (could be improved with insulation, for example.) The Process Capital Cost would be the cost of the insulation. The total cost of selecting each fuel would be based on interaction of these three factors: marginal cost of fuel use, process efficiency and process capital cost. Non-price factors intervene as well, preventing the purely economic selection from always being made. Non-price factors include poor consumer information plus other non-price fuel attributes. For example, because of the ease of temperature regulation, some people prefer cooking with gas even if electric stoves are less expensive. The calculation of the fuel share fraction is described in more detail later in this document.

In summary, retirements are a function of total capital stock and its lifetime. Additions to capital stock are a function of investments in production capacity and fuel market share. Fuel market shares depend on the relative costs of fuels, marginal process efficiency, and process capital cost. Also considered are non-price factors including split incentives, ease of use, and personal taste. For example, selection of a gas furnace may be influenced positively by the perception that gas heat makes a warmer home or negatively by the perception that gas is a dangerous fuel. Coal may be inexpensive to use but is not easy to acquire or use at the residential level. Electricity may not be the least expensive fuel for residential space heat but may be selected for its inexpensive installation cost if the cost of operation is not going to be borne by the builder.

2.3. Process Energy Requirements

Figure 5 is a visualization of the Process Energy Requirement structure. Starting on the left-hand side of the diagram, Process Energy Requirement Additions is shown to be a function of Marginal Process Efficiency, Production Capacity, and saturation of devices. Production capacity is determined by the output targets of the industry - as in how much mechanical energy is required to make a certain number of widgets. Device saturation is important also - if there is a change in saturation levels of a particular device, process energy will change as well. Saturation refers to the percent of production capacity actually requiring a particular end-use, not the percentage of electric devices alone. For example, commercial air conditioning saturation refers to what percentage of commercial output uses air conditioning, not how much of the air conditioning uses electricity. As more commercial establishments acquire air conditioning (natural gas or electric), the saturation rises and the Process Energy Requirement rises as well.

Figure 5. Process Energy Requirements



The marginal process efficiency is also important. A building under construction has certain heating and cooling requirements that will become part of the stock with a certain Process Energy Requirement. Rising marginal process efficiencies will gradually increase the average process efficiency; falling marginal efficiencies produce the opposite result.

Marginal process efficiency refers to how much process energy is needed per unit of output. The level of process efficiency chosen depends on the cost of using fuel which in turn, depends on Fuel Prices, Device Efficiencies, and Device Capital Cost. Once a level of device efficiency is chosen, the capital cost is known from trade-off curves contained in the model. Alternatively, the capital cost can be chosen and the efficiency determined from those curves. A certain capital cost buys a certain level of efficiency. For example, when building a home, a decision must be made as to how much insulation to include. The insulation levels in new home construction will vary with, among other things, fuel costs. Each fuel has its own cost and will determine an economic level of insulation. The cost of using a particular fuel considers fuel costs, device efficiency and device capital cost simultaneously. There is an interaction with process efficiency as well. If a home is well insulated a smaller furnace may suffice, lowering not only the cost operation and reducing fuel costs but reducing device capital costs as well.

The Marginal Process Efficiency also alters the Average Process Efficiency. If the Marginal Process Efficiency is greater than the Average Process Efficiency, then the Average Process Efficiency should be increasing; the converse is also true. In addition to the Marginal Process Efficiency, the Average Process Efficiency is affected by the additions to production capacity. If the additions each year are large, then the marginal process efficiency will have a larger effect on the average. If the economy isn't growing very much, marginal changes will not have a large impact on the average efficiencies. If fuel prices are so high that they choke off investment, then the corresponding high efficiencies that would be selected in response to high fuel prices will not have a large effect on the average process efficiency because although the new process efficiency is higher, there are few new structures being added to the capital stock.

The Process Energy Requirement Retirements are a function not only of the Production Capacity Retirements but also a separate process lifetime. When production capacity retires, the Process Energy Requirement retires as well. Process lifetime is used primarily in determining industrial energy demand and captures the effects of process obsolescence. There are times when processes change before production capacity retires creating a shorter and different lifespan for Process Energy Requirements. For example, in the paper industry there has been a gradual change from mechanical to chemical pulping. Mechanical pulping equipment is becoming obsolete and is not replaced so that mechanical pulping becomes an increasingly smaller share of pulping energy requirements. The principal reason for this shift is economic - chemical pulping is cheaper. Although we focus on price-related reasons, other reasons may play a role in process obsolescence as well. The steel and auto industries have also changed their Process Energy Requirements in existing and still used structures. Process efficiency lifetimes are usually about the same as the lifetime of the production capacity but can be shorter.

The other component of Process Energy Requirement retirements is production capacity retirements. If a plant or building is closed or destroyed, the energy requirements obviously cease.

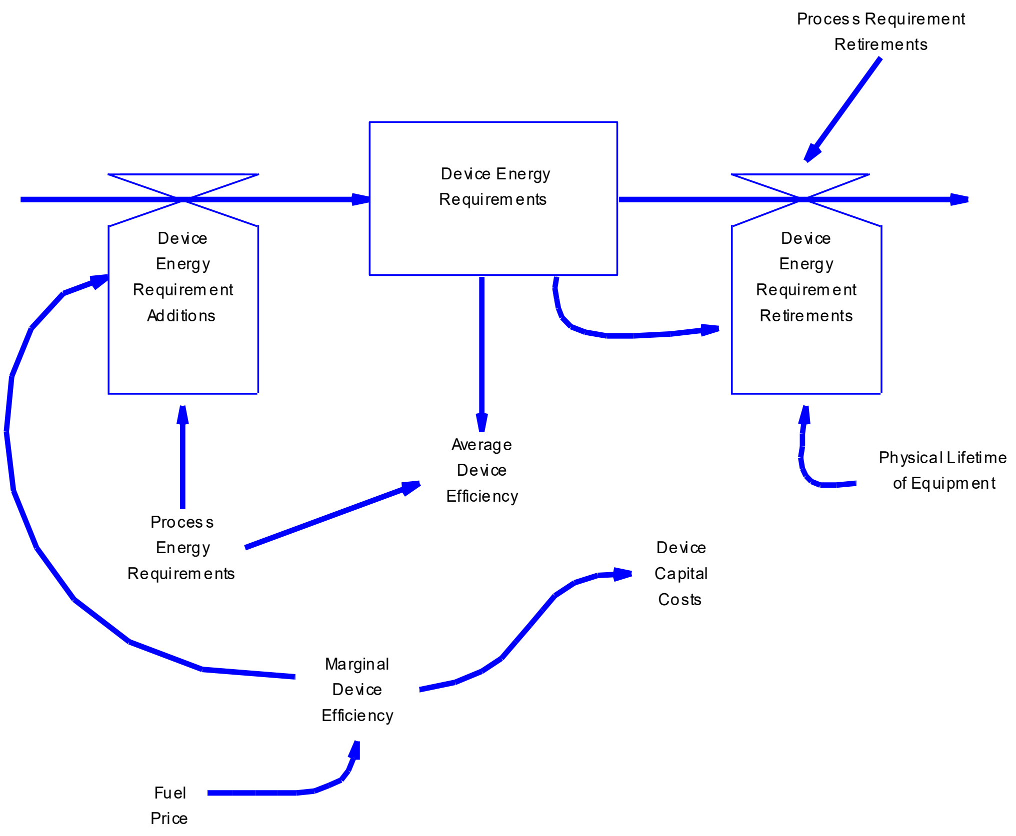
2.4. Device Energy Requirements

The general methodology simulating Devices is similar overall to Production Capacity and Processes, where a stock of energy requirements is modified each modeled year with additions and retirements. However, additional detail has been added to match the more specific input data available at the device level for retirements. The mechanics of retiring Devices is expanded to include more detailed handling of stocks by age (Vintages).

Structures Used to Model Devices

Figure 6 provides a picture of the model relationships that make up Device Energy Requirements.

Figure 6. Device Energy Requirements



Device Energy Requirements change in response to additions and retirements. Device Energy Requirement Additions are a function of Process Energy Requirements and Marginal Device Efficiency. The Process Energy Requirements determine the total energy needed, the marginal device efficiencies determine the number of devices added and are a function of fuel prices. As fuel prices rise, it becomes cost-effective to pay for greater device efficiency. To illustrate, the Device Energy Requirements of air conditioners measures how much cooling we can produce given a certain level of energy input (such as kWh). This depends in part on PER, which is how much cooling is required. The other component determining Device Energy Requirements is Marginal Device Efficiency (Marginal Device Efficiency), the ratio of energy into the device to the useable energy out of it. The Marginal Device Efficiency is determined by the fuel prices. As prices rise, marginal device efficiencies rise as well. As with marginal process efficiency, once Marginal Device Efficiency is known, Device Capital Costs are also known through the estimated trade-off curves. How much additional device efficiency will be selected as prices rise depends on consumer preferences for higher upfront capital costs over higher operating costs in the future.

Device retirements depend on the physical life of the equipment as well as the life of the building in which it is contained. In a fashion similar to Process Energy Requirement retirements, Device Energy Requirement Retirements are calculated as a function of Physical Lifetime of Equipment and Process Energy Requirement Retirements. Device Energy Requirement Retirements occur when Process Energy Requirement Retirements occur. If a building is destroyed, the devices contained therein are also destroyed. But devices have a far shorter lifespan than buildings and can be replaced many times before the building is replaced. Each time the device is replaced, the entire efficiency cycle will be repeated. If fuel prices are rising, appliances with higher device energy efficiencies will be selected. Not that this structure allows replacement of devices for only two reasons - either the process or structure housing the process is retired or the device is exhausted. No retrofits (profit maximizing replacement of equipment not yet worn out) are included in this model routine. Retrofits are included elsewhere in the model.

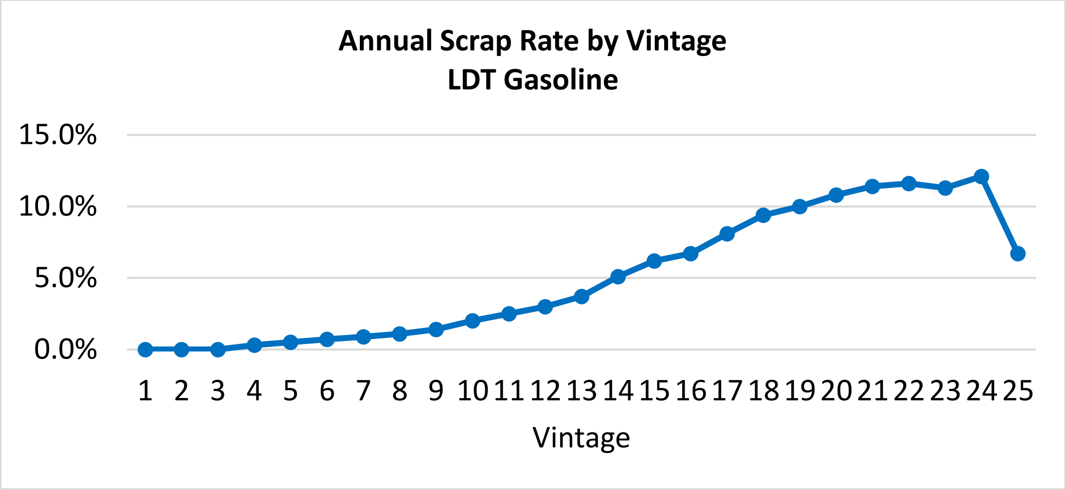
As in the process efficiency calculations, an Average Device Efficiency is also calculated from the Marginal Device Efficiency and Process Energy Requirements. If Process Energy Requirements are low due to the dampening effect on the economy of high fuel prices there will be little impact on average device efficiency, even though device efficiencies at the margin will rise.

Vintaging of Devices

‘Vintaging’ describes the separation of device stocks into more detailed lifecycle age groups using scrappage rate input data. Scrappage data allows for differing stock retirement rates as a device ages. A newer device that is a year or two old is generally much less likely to need replacing compared to an older version. Incorporating this data into the model allows for more realistic device stock representation and allows for forecasted device additions to match historical trends, such as the need to replace more stock when large ‘bubble’ years of historical additions begin to age.

Figure 7 shows an example of an increasing rate of retirement using passenger vehicle scrappage rate data.

Figure 7. Example Scrappage Rates by Vintage



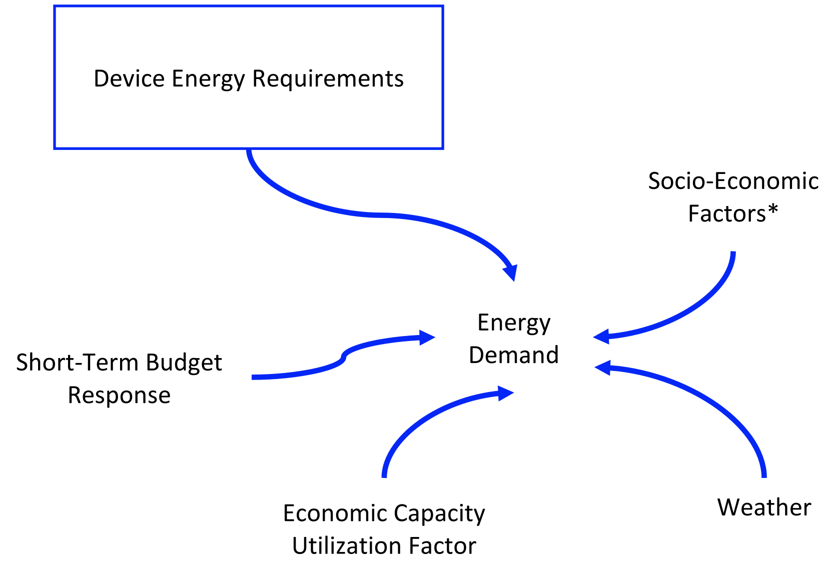
To utilize the scrappage data by age, devices are grouped into vintages containing groups of similar age (generally by age of device in years). The devices in each vintage retire at the scrappage rate after accounting for retirements due to process and capital retirements. Remaining devices are then aged into the next vintage for the next model execution year. Demand and total device energy requirement outputs are calculated using the sum of requirements across all vintages remaining after retirements.

2.5. Utilization Factors

Utilization factors, illustrated in Figure 8, modify the final Energy Demand. The calculation of Device Energy Requirements assumes the devices are run at a normal level. Often however, deviations from “normal” levels occur. The model isolates six utilization factors (eight variables) that can influence energy demand.

The first is the short-term budget response that operates in response to higher fuel prices. As energy prices rise, consumers struggle to keep within their budgets. They do this by attempting to maintain the same dollar shares of the goods and services they desire. Therefore, they must cut back on their energy consumption - by turning off lights, using less hot water, turning back the thermostat, carpooling, and the like. Since their standard of living is affected by these behaviors, they will attempt to return to their previous standard as quickly as possible. The budget response, then, is a short-term response that gives way to long-term changes in behavior and is determined by fuel prices.

Figure 8. Utilization Factors



*Calibration Variable

This response is modeled by creating two budgets: the average consumer budget and a new budget, influenced by changing fuel prices. As fuel prices change, the energy component of the budget changes as well. This new budget is compared to the average budget and an adjustment is made to the average budget. In time the new budget will equal the average budget. The length of time for this to occur is dictated by the Budget Adjustment Time. The Short-term Budget Response is a short-term effect, usually no longer than two years.

The next utilization adjustment identified is the Economic Capacity Utilization Factor. This factor measures the effects of the economy on the residential, commercial, and industrial sectors. For the residential sector, this adjustment picks up large changes in the employment rate; in the commercial sector, unusual changes in commercial output are the key and in the industrial sector, it is a measure of how hard factories are running. For example, during a recession, factories may run only one shift; during boom times factories may run twenty-four hours a day. The Economic Capacity Utilization Factor captures these dynamics. During recessionary periods, the value of Economic Capacity Utilization Factor is less than one; during very good times, greater than one.

The third factor (two variables) to be considered is weather. Two variables represent the effects of weather on energy demand - the Degree Day Multiplier and the fraction of Temperature Sensitive Load. Each end-use has a specified fraction of its load designated as temperature-sensitive. This fraction ranges from zero to one. This portion of the load is adjusted in response to a change in the degree day multiplier (usually set equal to one). If a period of time is expected to have warmer or cooler temperatures than normal, the Degree Day Multiplier is adjusted to reflect this expected temperature increase or decrease and the energy demand from temperature sensitive load is adjusted accordingly. What is considered temperature sensitive load in an end-use can vary by customer class. For example, residential air conditioning is usually entirely temperature sensitive; however, some commercial air conditioning load is constant and does not vary with temperature. The fraction of temperature sensitive load for residential air conditioning would equal one; for commercial air conditioning, it would be something less than one.

The fourth utilization variable that affects energy demand is the Socio-Economic Factor. This is a calibrated variable that reflects behavioral and structural changes that cause a change in energy demand. For example, the changing labor force participation rate for women over the past twenty years has altered energy use levels and patterns. This variable can also pick up structural changes not otherwise accounted for in the model. Some versions of ENERGY 2100 do not distinguish between single and multi-family residential dwellings. If this distinction is not made and over the historical period there has been a shift from single family to multi-family dwellings or vice versa, CERSM will capture the changing energy use patterns.

The Socio-Economic Factor lies on the “edges” of the model, capturing the effects of structure not included in ENERGY 2100. It is modeled in an aggregate fashion, and as individual effects become of interest or internalized (such as the single and multi-family residential distinction), this variable becomes less important.

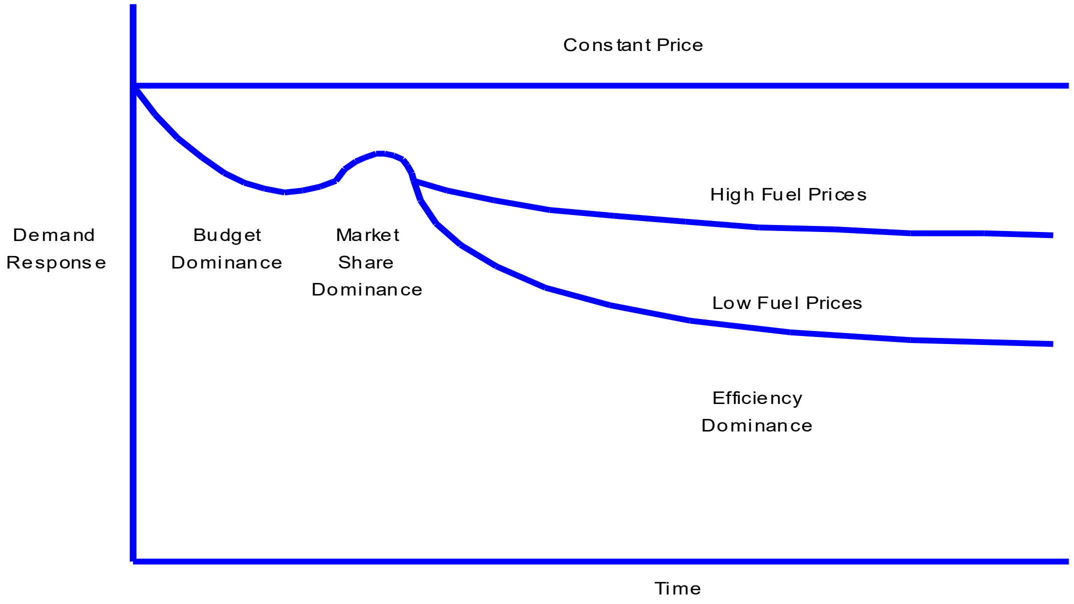
As you can see by the asterisk next to the variable, the model derives Socio-Economic Factor values through calibration. The calibration process attempts can to replicate history in all the important energy variables. While this calibration is being performed, certain parameters within the model are set at the same time. Socio-Economic Factor is one of these variables, and the values derived historically are used to forecast the relationship between changes in economic output and changes in energy demand, insofar as possible. If the value of Socio-Economic Factor is close or equal to one, then no significant changes have occurred that have not been captured by the model through the other adjustment factors. If the number is greater or less than one, then more or less energy is being used per dollar of output than before. If the Socio-Economic Factor is growing, the reason for this growth needs to be investigated or explained.

The fifth utilization factor is the Capacity Utilization Factor. This differs from the Economic Capacity Utilization Factor previously described. While Economic Capacity Utilization Factor picks up shifts in gross output, the Capacity Utilization Factor is a calibrated variable reflecting energy demand fluctuations not explained by the other utilization factors. Fluctuations in this variable should be random (no pattern of steady increase or decrease) when they occur and the variable's usual value is one.

2.6. Price Response Dynamics

Figure 3, the Demand Overview, contains all the components that have been described in the preceding section. Notice that price responses are present at every level. These price responses are summarized and illustrated in Figure 9. This diagram traces the effect of an increase in fuel price on that particular fuel's total energy demand.

Figure 9. Price Response Dynamics



For example, consider residential automobiles as the demand component and gasoline as the fuel. If gasoline prices are constant, the demand price response remains constant as well, as illustrated by the straight line at the top of the picture. However, if gasoline prices increase, an overlapping three stage process begins to take effect. First, the short-term budget constraint begins to operate and consumers begin to reduce the amount of gasoline they require. They can carpool to work or school, eliminate unnecessary pleasure trips, walk instead of drive short distances, etc. As they are given more time to adjust, the decision to purchase a new car will incorporate the higher gasoline prices. It is possible to buy a car powered by something other than gasoline (diesel, electric, or a gasoline mix) but fuel market share changes probably will not be great. However, more fuel-efficient cars will be selected. Therefore, the path that the gasoline price demand response will follow would be the "High Fuel Prices" path, since the price of other fuels matters little to a basically non-substitutable end-use.

Consider another example - residential electric hot water heating. If the price of electricity rises, again the budget constraint begins to operate and less hot water (as well as other sources of electricity use) is used. As consumers have time to adjust, and replace appliances with substitutable energy requirements, a gas water heater may be purchased instead of an electric heater. At the very least, a more efficient electric water heater will be purchased. If all energy prices rose with electric prices, then less fuel switching would occur and the demand response would follow the “High Fuel Prices” path. If only electric prices rose, more fuel switching would occur and the demand response would follow the “Low Fuel Prices” path.

These price response dynamics include both process and device efficiencies. Devices generally have shorter lifetimes and therefore their fuel requirements and energy efficiencies are the first to show consumer response to higher prices.

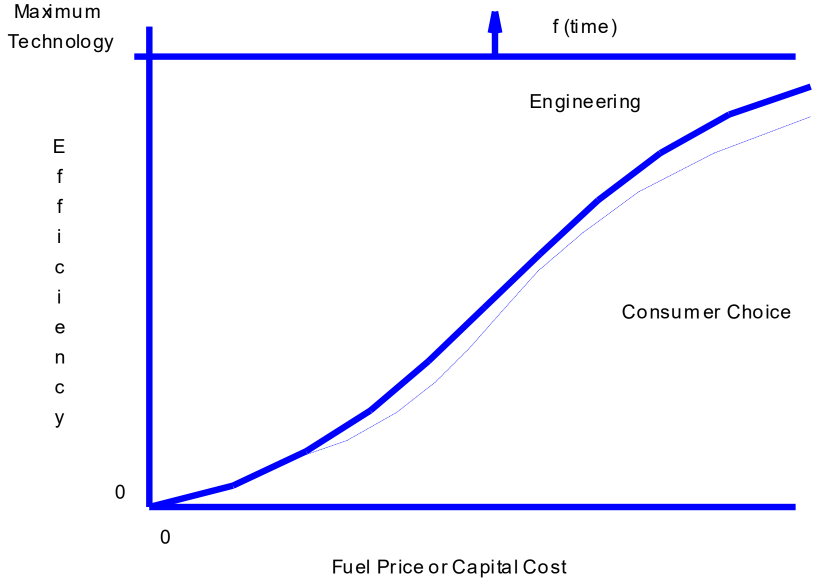
3. Consumer Choice Theory

The key theory that drives the demand sector is consumer choice theory - particularly two formulations – Efficiency Choice Curve and Market Share Mechanics - described briefly below. See Appendix 1 - Theoretical Derivation for more detail on consumer choice theory.

3.1. Efficiency Choice Curve

The first consumer choice formulation that is critical to the energy decision making process is the efficiency/capital cost trade-off. This is really a trade-off between high up-front costs and high future costs. If a very high efficiency furnace is purchased, the capital cost will be large, however, the operating costs in the future will be lower than with a lower efficiency furnace. Figure 10 illustrates this principal.

Figure 10. Efficiency Choice Curve - Efficiency/Capital Cost Trade Off



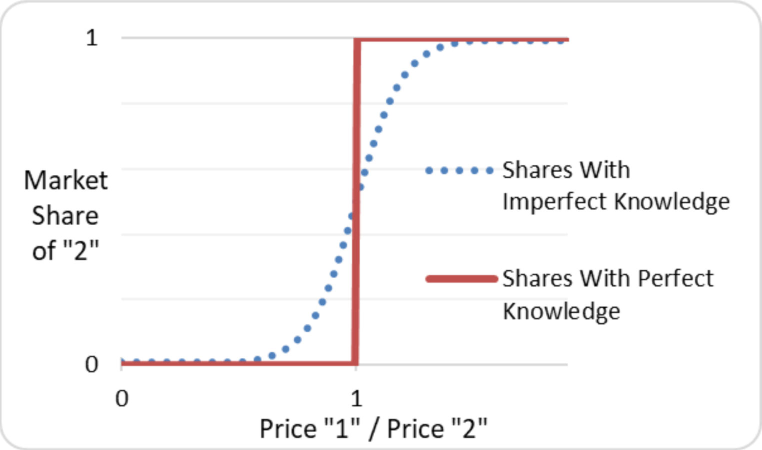
Either fuel price or capital cost can be used on the horizontal axis. Each price corresponds to two efficiency levels. The engineering curve selects the economically optimal level of efficiency for each capital cost or fuel price. The more useful and realistic curve is the consumer choice curve that shows a less than perfect relationship between efficiency and capital cost. The consumer choice curve reflects the fact that all additional capital cost dollars do not go into the purchase of higher efficiency. Top of the line appliances include many features (some energy using) that lower priced appliances do not. Self-defrosting freezers, ice makers, cold water spigots on the refrigerator door are all examples of the extra, energy using features, of high-end appliances.

The “S” curves for both the engineering and consumer choice relationships are drawn against a maximum technology curve (a ceiling on efficiency given current technology) which can change over time as technological breakthroughs occur. As the maximum technology line shifts, the engineering and consumer choice curves change as well.

3.2. Market Share Mechanics

The second consumer choice formulation that is important to the energy decision-making process is that of fuel choice. Figure 11 illustrates the process of fuel choice - trading off one fuel for another on the basis of relative prices.

Figure 11. Market Share Mechanics



If consumers behaved with perfect economic rationality and had perfect information, the market share curve would look like the share with perfect knowledge illustrated in the diagram. On the horizontal axis is the ratio of the price of fuels. As long as the price of “1” is less than the price of “2”, the fraction will be less than one, and economically driven consumers will choose all fuel “1” making the market share of “2” equal to zero. However, as soon as the price of “1” exceeds the price of “2”, then the converse occurs - “2” grabs the entire market. In reality, fuel choice is a less clear-cut process. As the price of one fuel rises relative to another, there will be a gradual shift to the cheaper fuel based on consumer perceptions of the relative prices (often made with imperfect information) as well as the influence of non-price factors. The curve formed by these decisions resembles the S-shaped curve in the diagram - even if price “1” is higher than price “2” some consumers will still choose the more expensive fuel. This can be the result of imperfect information or indifference (if fuel costs are a very small part of the budget) or because of a non-price related factor. For instance, some people choose gas stoves because they prefer to cook with them, not because of price differentials.

4. Modeling Self-Generation

Self-generation is a broad term referring to producing electricity at or near the point of use which would otherwise be purchased by a utility. Previous versions of the model focused on cogeneration (a type of self-generation also known as combined heat and power). Over time, ENERGY 2100 has expanded this sector to include all types of self-generation utilizing any number of methods, including solar panels, wind turbines, backup diesel generators, or cogeneration systems. Whereas the term “cogeneration” can be found through the documentation as well as in variables names, this is a legacy reference, and should be considered to represent the broader term of “self-generation”.

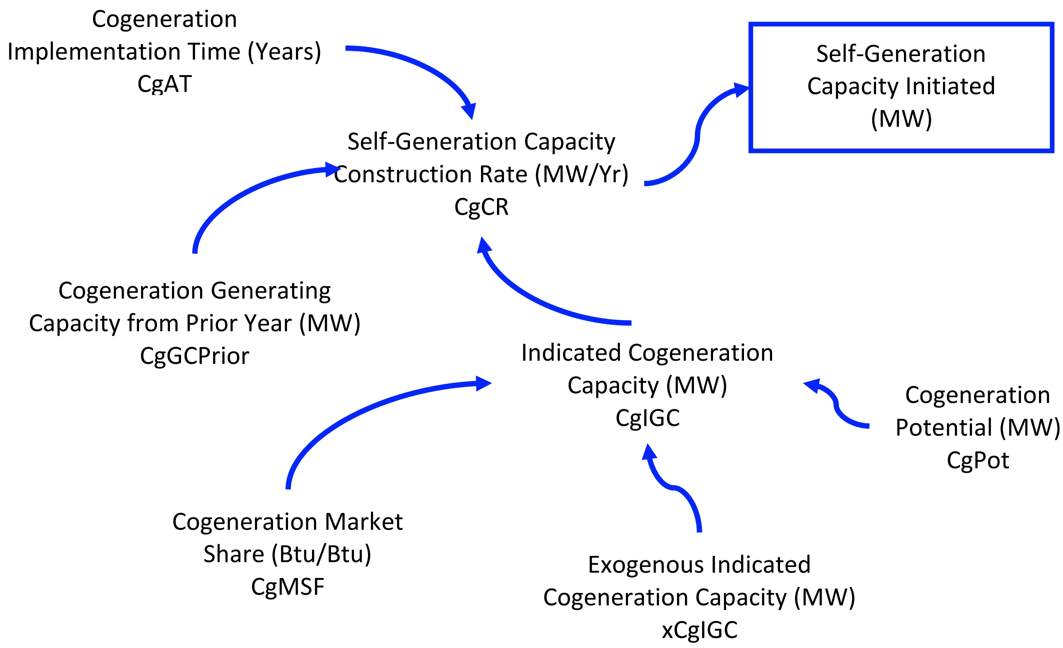
Self-generation is modeled in both in the Demand Module and in the Electric Supply Module. Individual self-generating units are defined as a type of electricity supply and are dispatched with an optimization similar to utility units. The Demand Module determines the amount of new self-generation capacity to initiate using consumer choice equations along with the marginal costs which are then sent to the Electricity Supply Module as input to the dispatch routines to determine generation.

4.1. Self-Generation Initiated

Figure 12 is the first of three diagrams illustrating the structure for modeling self-generation capacity initiated. In Figure 12, the key variable is MW of existing self-generation capacity. Self-generation capacity is modified by construction rates and retirements (additions to and subtractions from the stock of self-generation on a yearly basis). Self-generation retirements are simply a function of the physical lifetime of self-generation; however, self-generation construction rates require a more complex calculation.

Actual self-generation construction depends on planned self-generation capacity which in turn is a function of self-generation potential. The fraction of self-generation potential that is actually developed is determined by the Fraction of Potential Development. This fraction depends on the marginal cost of self-generation, the price of electricity to self-generators, and self-generation non-price factors. In a sense, this fraction is a market share choice between two options - self-generation and purchases from a utility. The choice is based on the relative prices of the two options plus non-price factors such as willingness to undertake such a project. The self-generation Non-Price Factors are calibrated in the same manner as other non-price factors - by looking at the historical market shares and prices and calculating the deviation from the economically optimal split.

Figure 12. Self-Generation Capacity Initiated



The Planned Self-Generation Capacity is derived by applying the Fraction of Potential Development to the Self-generation Potential. The Self-generation Potential is determined by the process heat requirements and the self-generation heat rate. Potentially all of these requirements could be met by self-generation; however, through the Self-generation Market Share Fraction, relative prices, and non-price factors reduce the potential to the more economically realistic planned self-generation capacity.

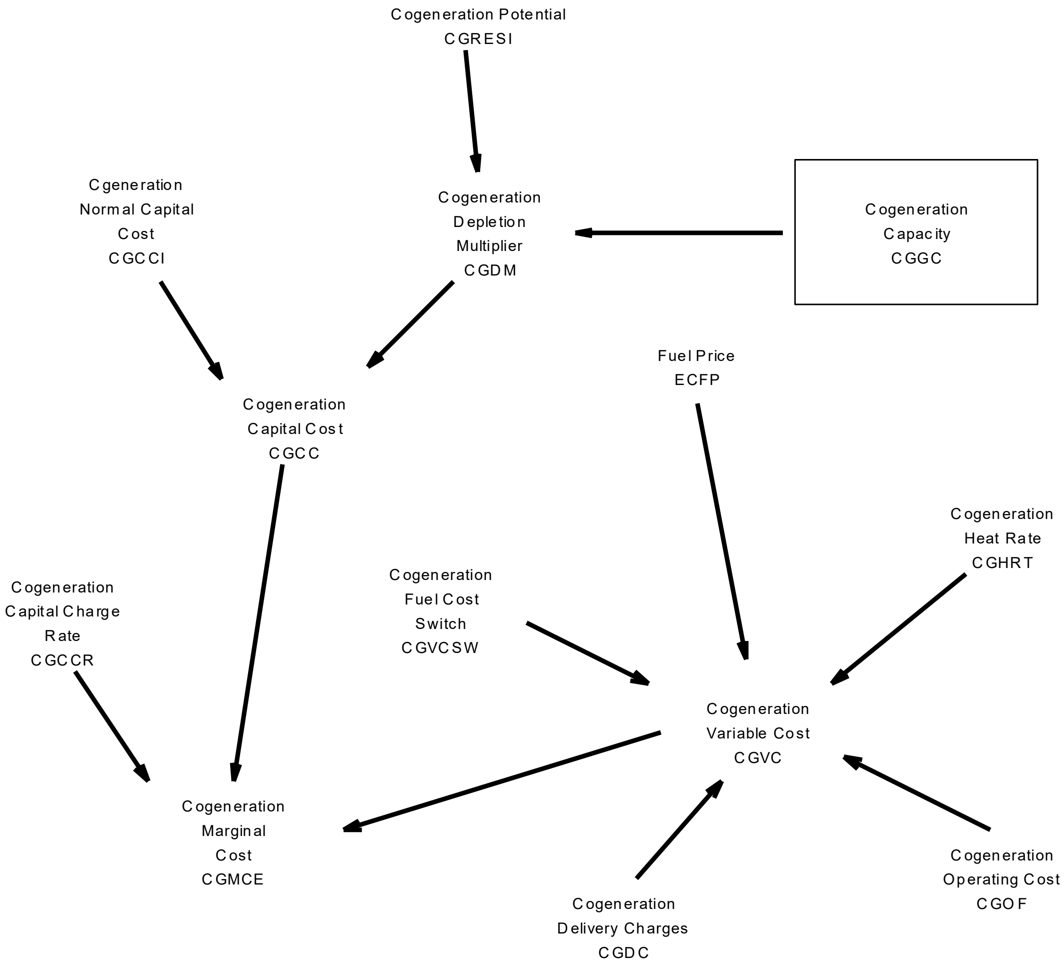
The planned self-generation capacity, additionally influenced by current self-generation capacity (existing self-generation would not be duplicated) as well as capacity retirements, is lagged by the Self-generation Implementation Time to yield the amount of self-generation construction that will be undertaken. The lag is important. Prices may dictate that self-generation is the economically viable choice today but it may take a year or more to get the system on line.

4.2. Self-Generation Marginal Cost

Figure 13 illustrates in full the calculation of the marginal cost of self-generation used in Figure 12. In the lower left-hand corner of the diagram, you can see that the Self-generation Marginal Cost is determined by the Self-generation Capital Charge Rate, Self-generation Capital Cost and Self-generation Variable Cost. Self-generation Capital Cost is multiplied by the Self-generation

Capital Charge Fate to get a levelized capital cost. The derivation of the capital charge rate is similar to the rate used in the utility sector of the model and will be discussed there. A self-generation variable is calculated and added to the levelized fixed cost yielding the cost of self-generation at the margin.

Figure 13. Cogeneration Marginal Cost



The self-generation variable cost component of marginal cost is made up of four factors: fuel prices, heat rates, delivery charges, and operating costs. Fuel prices refer to the fuel used in the boilers, the heat rate is the efficiency of the boilers, the operating costs consider operating costs associated with the self-generation alone, and the delivery prices include such miscellaneous costs as electric back service charges. There is also a self-generation fuel cost switch which may be turned off if there is no fuel cost such as is the case with hydropower self-generation.

The self-generation capital costs include the normal costs for self-generation plus a factor to account for depletion of certain self-generation resources. The Self-Generation Depletion Multiplier is calculated by assessing the Self-generation Potential and subtracting from it the existing Self-generation Capacity. For some technologies, costs increase as more self-generation is developed. For example, hydropower self-generation tends to be more expensive at later installations than the first because the cheapest (best flow, most accessible) sites are developed first. For technologies that experience this type of cost increase with subsequent installation, the Self-generation Depletion Multiplier tracks these increasing costs.

5. Demand Policies

5.1. Strategies and Policy Approaches

Table 1 provides a list of the different types of policies that can be implemented in ENERGY 2100 given its causal demand structure. It is possible through policy choices to affect consumers' decisions concerning fuel choice, device and process efficiencies, and energy use in general. Different financial treatments can be simulated such as shared savings, expensing vs. capitalizing conservation costs, rebates, loans, tax credits and such to estimate where the various costs and benefits of conservation and DSM programs will fall.

Table 1. DSM Strategies & Demand Policy Approaches

- *Trade-off Curve Choice*
- *Shared Savings*
- *Expense/Capitalize conservation*
- *Mandated Cut-Back*
- *Energy Taxes*
- *Device/Building Standards*
- *Appliance Rebate Programs*
- *Low-Interest Loans*
- *Capital Cost Subsidies*
- *Technology Improvement (Research and Development)*
- *Device Saturation*
- *Tax Credits*
- *Environmental Controls*
- *Marketing/Information Programs*

Mandated cutbacks and standards, taxes, rebates, loans, time-of-use rates and information programs are all modeled in ENERGY 2100. Given its unique model structure, these policies plug into the existing routines and their effects can be seen working their way throughout the entire energy system.

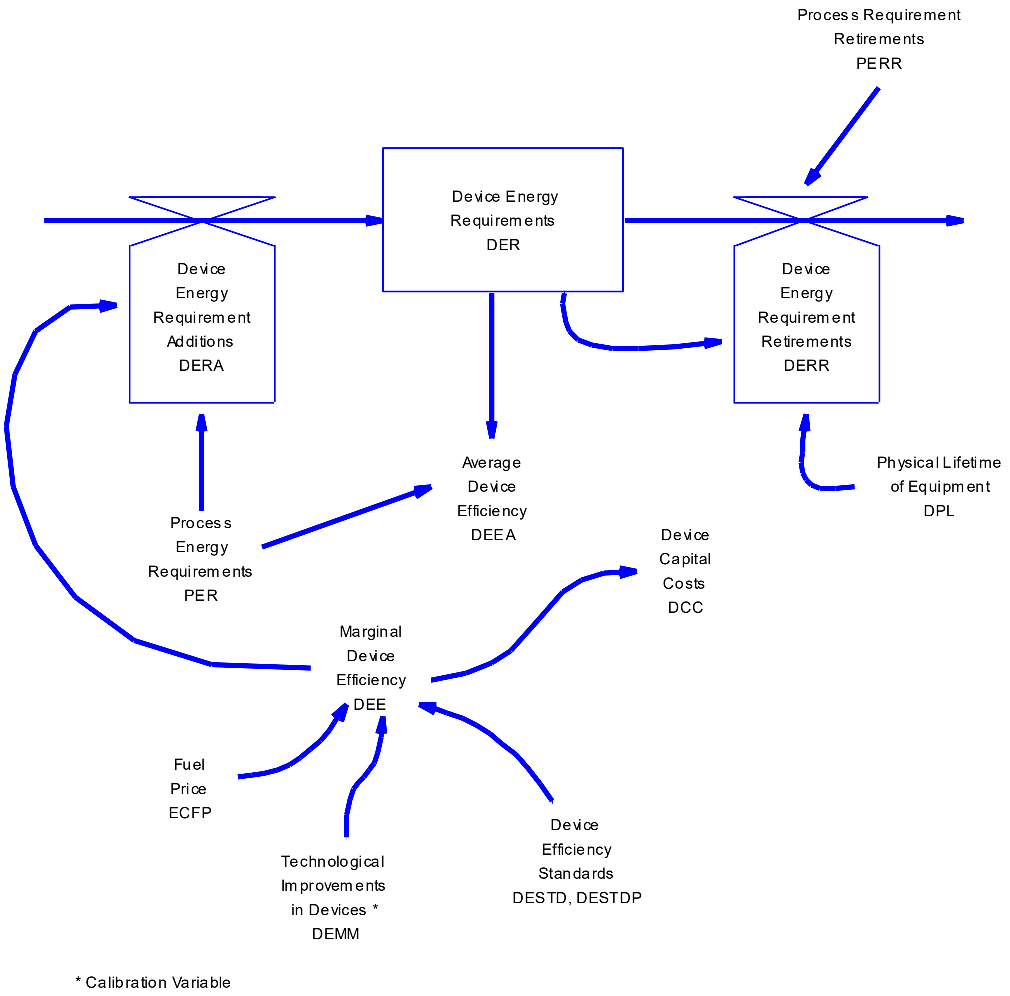
5.2. Device Technology Improvements and Efficiency Standards

Figure 14 adds structures to represent technological improvements and efficiency standards. The new variables are Technological Improvements in Devices and Device Efficiency Standards.

Technological Improvements in Devices is a non-price factor in the consideration of the level of device efficiency to be selected. It is a calibrated variable that can be altered in the future as a policy variable. During historical calibration, the model compares its estimated device efficiencies with the actual historical device efficiencies, to the extent that this information is available. If there is a difference between the two sets of values, improvements in technology are assumed.

The other non-price effect on Marginal Device Efficiency come from standards. Standards are assigned to one of two variables: one is for standards already in effect such as those included required from federal energy policy. The other is for policy development that allows the analyst to test out different standards as policies. The basecase version of the model includes the current standards for any end-use that has one.

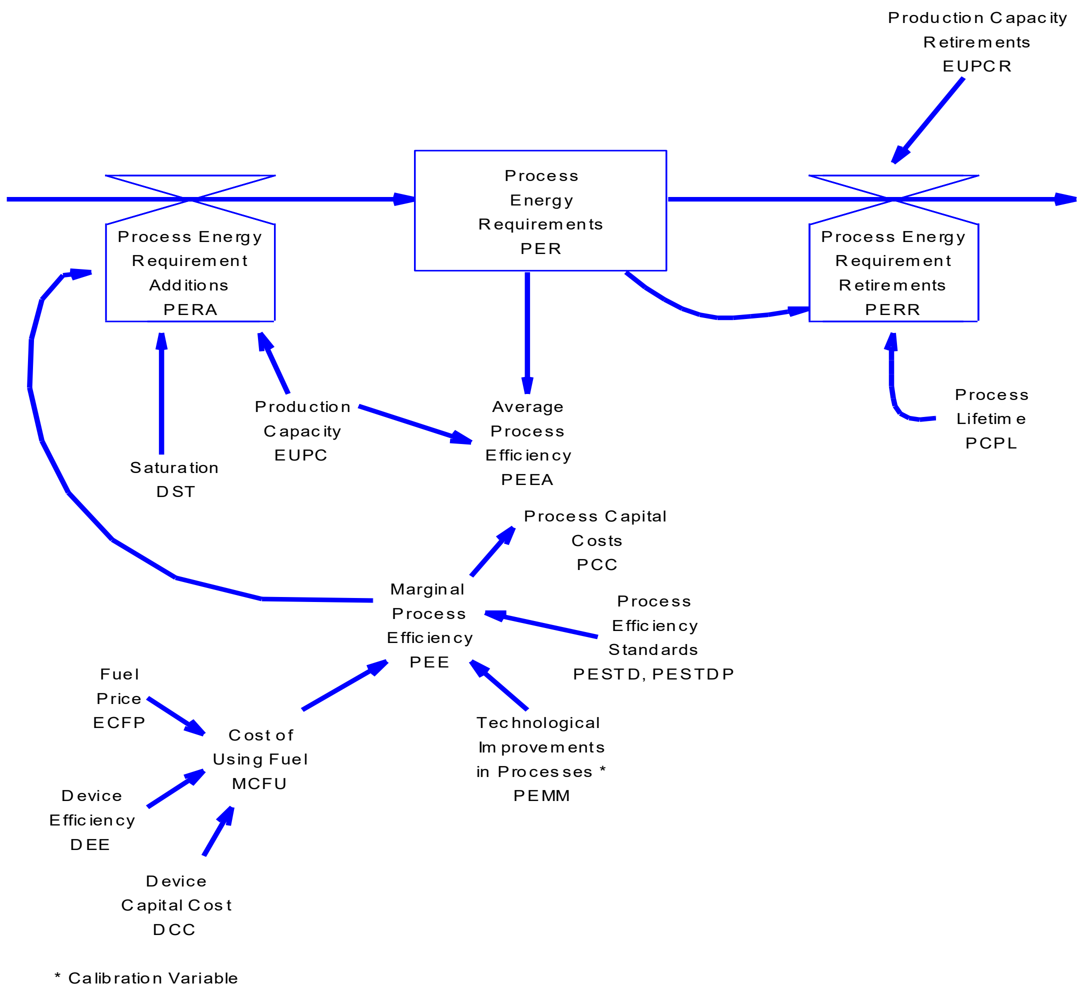
Figure 14. Devices with Technological Improvements and Efficiency Standards



5.3. Process Technology Improvements and Efficiency Standards

Figure 15 adds Technological Improvements and Efficiency Standards to Figure 5. Technological Improvements in Processes is a calibrated variable reflecting non-price induced technological improvements which may reduce or improve process efficiency. Processes that become more energy intensive, often by substituting capital for labor, cause more energy to be used per dollar of output. For example, the use of robots instead of human labor means that more, and not less, electricity is used to produce a particular product. PEMM can change over time and these changes should be evaluated and connected to “real world” events. Structural changes can show up in this variable as well. In the commercial sector, movement from small, stand-alone stores to strip malls has created new energy requirements; Technological Improvements in Processes reflects these changes.

Figure 15. Process Energy with Technological Improvements and Efficiency Standards



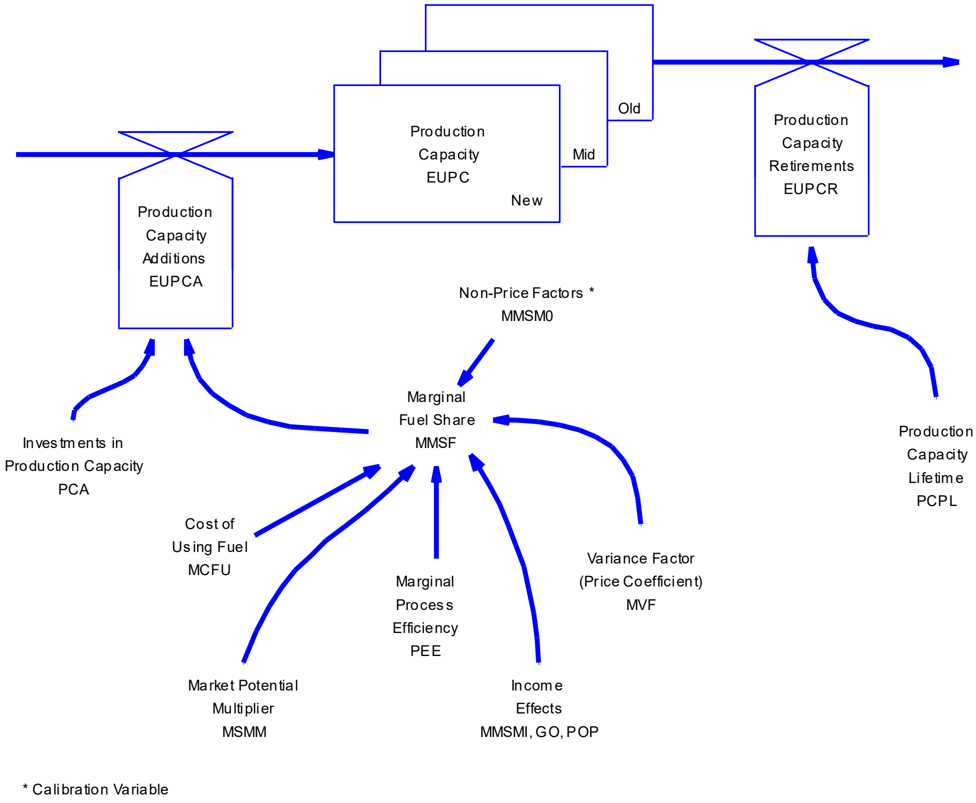
The two process standards behave in the same manner as the device standards. One reflects standards already in place and is included in the model's base case data set. Standards on process are usually building shell efficiencies such as certain insulation requirements but theoretically there exists a process efficiency for each end-use which can be manipulated with a standard. The other is the policy variable used for testing different process efficiency standards in the future.

5.4. Production Capacity with Income Effects and Non-Price Factors

In Figure 16, the Production Capacity Diagram is shown before is augmented with Income Effects and Non-Price Factors. The calculation of the Marginal Fuel Share includes the price factors as discussed before - the cost of using fuel, the Marginal Process Efficiency and the

Process Capital Cost. In addition, now the non-price factors are accounted for: income effects, a market potential multiplier and other non-price factors.

Figure 16. Production Capacity with Income Effects and Non-Price Factors



The other non-price factor is a calibrated variable derived from a comparison of the model-generated historical fuel splits with the actual splits. Using historical prices, we can estimate the non-price factors (the deviation from the economically optimal split based on historical prices). Many non-price factors are included in this variable, such as imperfect information, split incentives, or personal, non-price determined perceptions such as the perceived safety of a particular fuel.

The income effect factors are a new addition to the model and are not contained in the generic version. One is a measure of income elasticity; another represents gross annual output and the third is a population variable. Theory indicates that as income increases (gross output divided by population), fuel market shares change. In other words, as the income of residential customers begins to climb, they may show a different preference for fuel choices - a shift in the non-price factors. For example, an “all electric” house may be perceived as cleaner and a good worth purchasing as incomes rise, causing more residential customers to choose electricity over natural gas for space heating, water heating and drying needs.

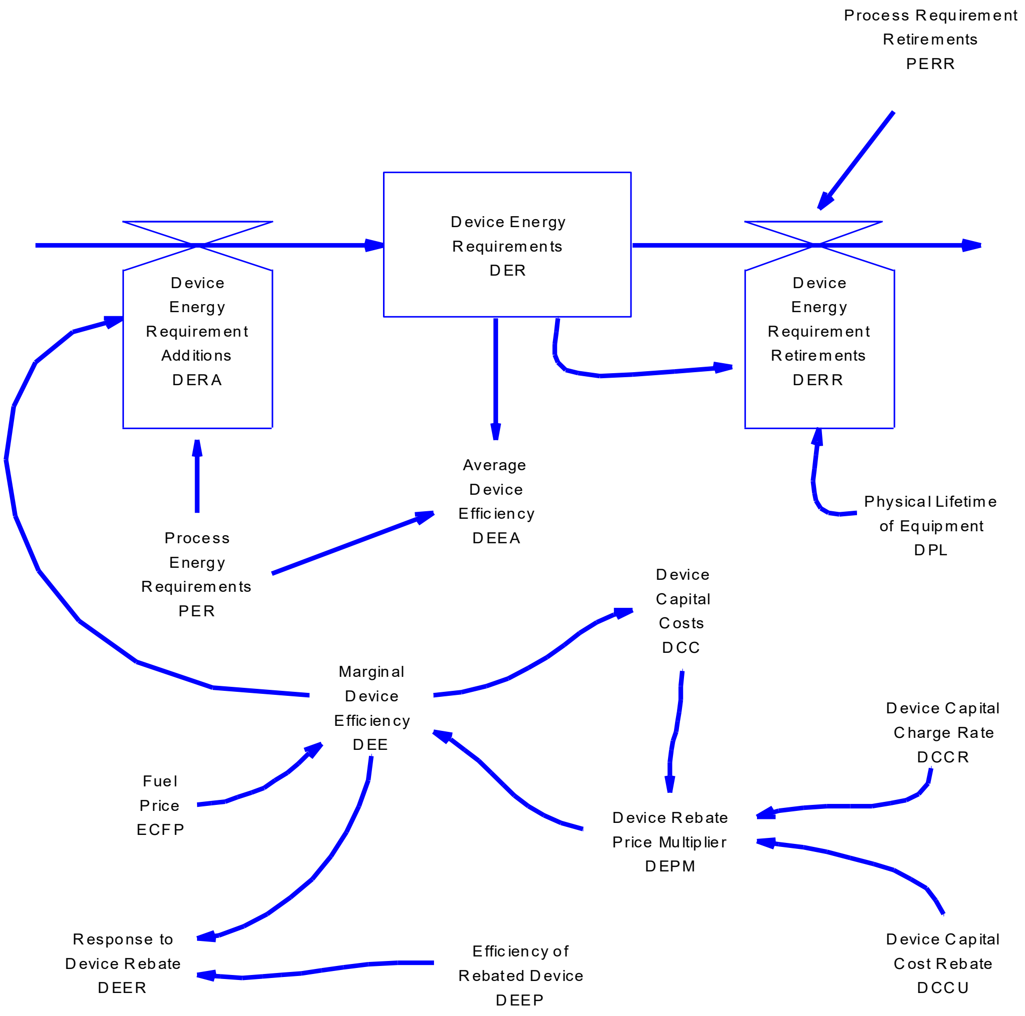
The Market Potential Multiplier is a policy variable that modifies the non-price factors influencing the marginal fuel share. For example, an information program to promote heat pumps may change consumer attitudes about unreliability or unsuitability for their climates and enhance the non-price factors for electric technology. Similarly, an information program concerning the reliability, safety and availability of natural gas space cooling could enhance the natural gas market share for that end-use.

5.5. Device Rebates with Endogenous and Exogenous Participation Rates

Figure 17 adds a Device Rebate Price Multiplier to Figure 14. This Device Rebate Price Multiplier variable represents the consumer’s response to a rebate and is based on three other variables: the Device Capital Cost that we are already familiar with; the Device Capital Charge Rate and the Device Capital Cost Rebate.

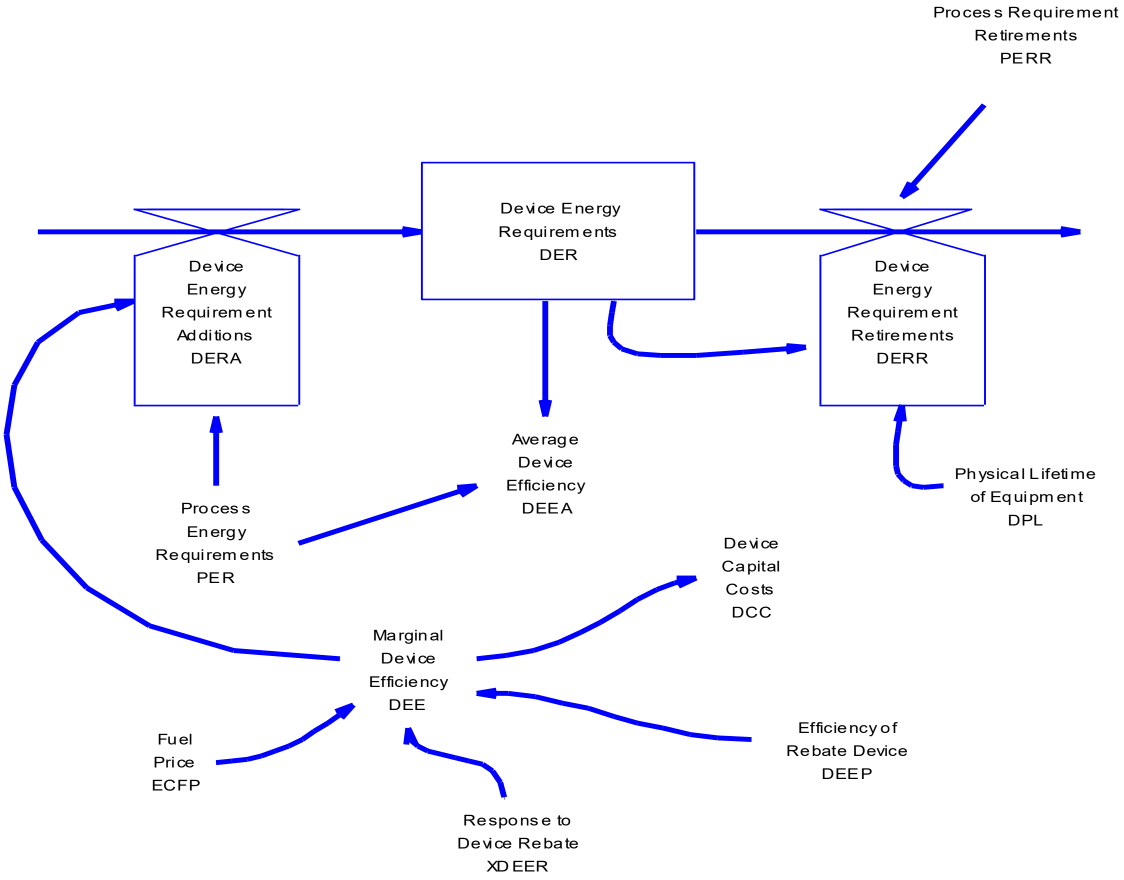
A rebate allows a consumer to pick a higher efficiency device for the same dollars. However, the actual capital cost of the device increases. Since we use efficiency curves that trade-off energy prices and efficiency levels, the model converts this higher capital cost to a higher energy price and selects the correct efficiency level. The Device Rebate Price Multiplier is the difference between the actual energy price and the “pseudo-energy” price determined by the rebate. It is calculated by deriving an annualized capital cost (Device Capital Cost times the Device Capital Charge Rate) and adding to it the value of the rebate.

Figure 17. Device Rebates with Endogenous Participation Rates



As the Device Rebate Price Multiplier changes, the Marginal Device Efficiency changes as well - a higher price buys a higher efficiency level. The device efficiency of the rebated device as well as the new Marginal Device Efficiency determine the response to the Device Rebate, the participation rate in the program. As shown in Figure 18, the response to the Device Rebate can be entered into the model exogenously as well endogenously under the variable xDEER. The Device Rebate Price Multiplier is no longer calculated when an exogenous participation rate is assumed and the xDEER now directly influences the Marginal Device Efficiency. In the exogenous participation rate, consumer behavior is "known" and outside the model.

Figure 18. Device Rebates with Exogenous Participation Rates



5.6. Process Rebates with Endogenous and Exogenous Participation Rates

A similar enhancement can be done to Process Efficiency, creating Figure 19. Here the new variables are the Process Rebate Price Multiplier, the efficiency of the Rebated Process and the Response to the Process Rebate. Again, the multiplier is derived from the process capital costs, the process capital charge rate and process capital cost rebate. A higher level of price efficiency is selected through the creation of a “pseudo-energy price” that is higher than current energy prices. The marginal process efficiency selected is increased and, coupled with the efficiency of the rebated process, determines the participation rate in the program. Alternately, Marginal Process Efficiency Retirements can be entered exogenously and a new marginal process efficiency calculated (Figure 20). It can be instructive to run both scenarios and compare results.

Figure 19. Process Rebates with Endogenous Participation Rates

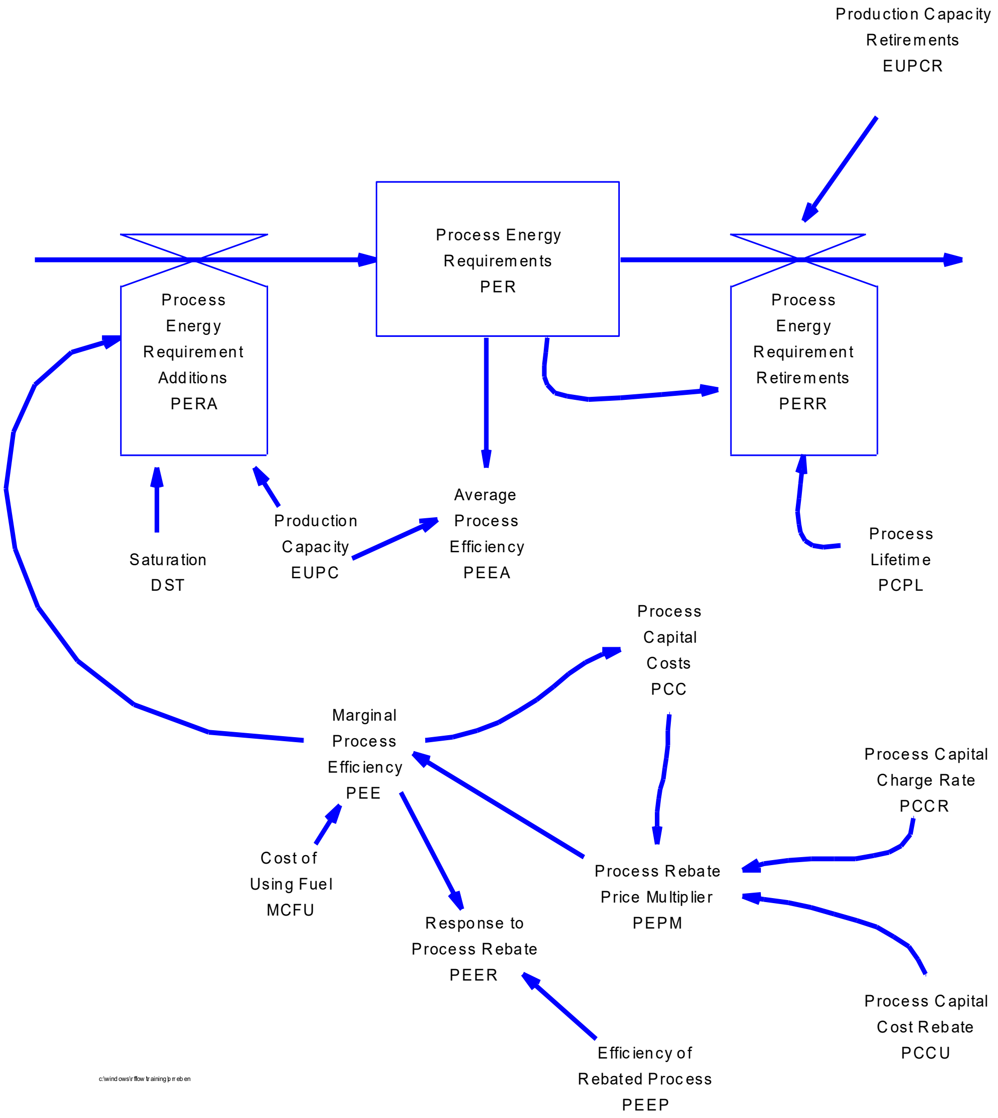
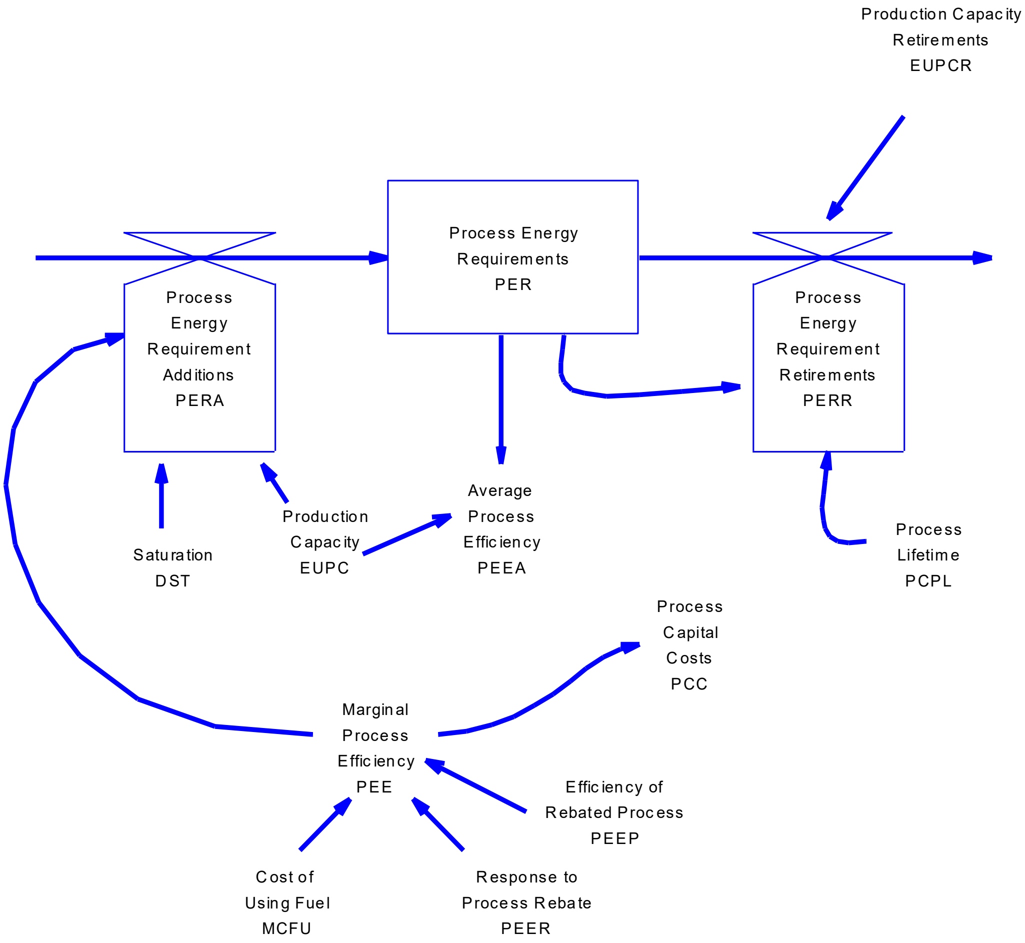


Figure 20. Process Rebates with Exogenous Participation Rates

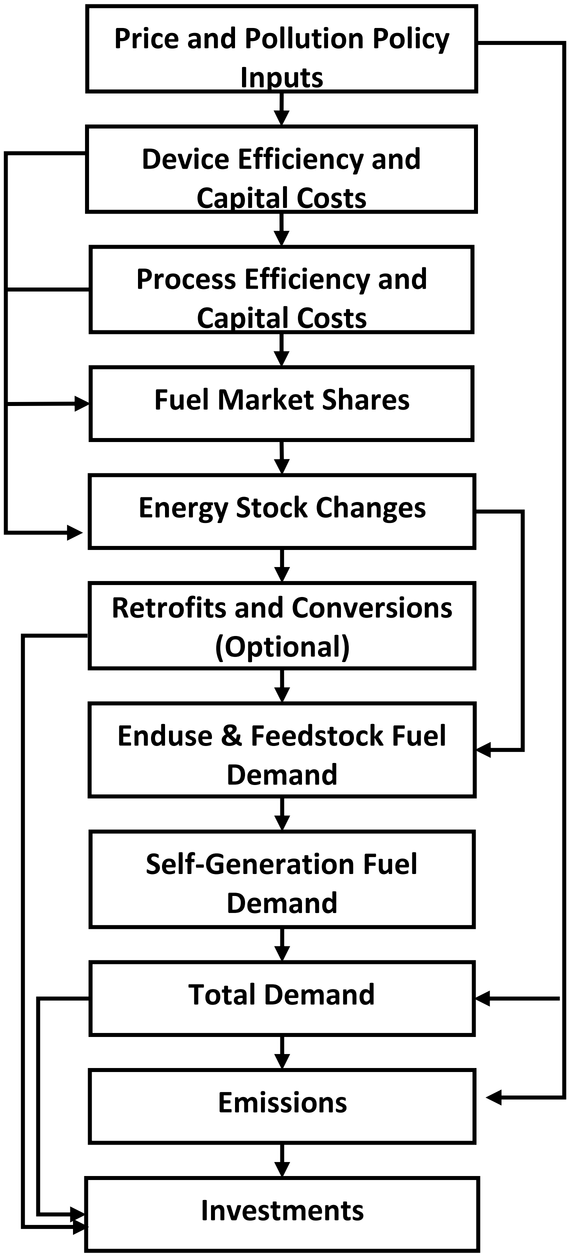


6. Demand Module Mechanics

6.1. Flow of Demand Module Routines

Figure 32 shows a flow diagram of the demand module's order of execution. The objective of each of the steps is summarized as follows:

Figure 21: Demand Module Steps and Linkages



1. The **price and pollution policy inputs** routines process price and pollution input variables for use by other routines.

2. **Device efficiency and capital costs** apply a trade-off curve to determine marginal device efficiencies and device capital costs. These routines also calculate a marginal cost of fuel use, used in later routines as input to determining fuel choices.

3. **Process efficiency and capital costs** apply a trade-off curve to determine marginal process efficiencies and capital costs.

4. The **fuel market shares** routines use consumer choice equations to calculate fuel market shares for new purchases made due to economic growth or retirement of capital stock.

5. The **energy stock changes** routines calculate capital stock, device energy requirements, and process energy requirements due to additions and retirements caused by changes in drivers, device saturations, and aging stock.

6. The **retrofits and conversions functions** optionally modify levels of capital stock due to retrofits and allow for consumers to convert stock to alternative fuel types

(using consumer choice equations) at the end of the stock's useful life (typically only occurring with new stock additions).

7. The **enduse and feedstock fuel demand** functions calculate enduse and feedstock energy demands and make optional adjustments due to impacts of energy efficiency programs.
8. The **self-generation fuel demand** functions aggregate self-generation energy demands from unit-level self-generation specified in the electric supply module.
9. The **total demand** routine creates summary demand variables representing total demand (end-use plus feedstock plus self-generation).
10. The **emissions** functions calculate energy-related emissions from the end-use, feedstock, and self-generation demands.
11. **Investments** determine marginal investments in devices and processes made by the demand sector which can be passed to a macroeconomic model to obtain economic impacts.

6.2. Source Code

ENERGY 2100 source code is written using the Julia language. Parallel model code and variable definitions exist for each of the residential, commercial, industrial, and transportation sectors. Model variables are defined in a \Database subdirectory, and code that simulates the demand module is contained within the \Engine subdirectory.

File names of source code are organized by prefix, indicating the segment or sector's code is contained in that file. For example, files beginning with 'R' represent residential demand files; whereas, files beginning with 'C' represent commercial demand files, and the same is true for Industrial ('I') and Transportation ('T').

The types of demand files and their naming conventions are shown in Table 2.

Table 2. Demand Module Source Code File Names

Demand Source Code Objective	Demand Source Code File Name		
Defines input variables, calibration variables, and output variables	RInput.jl	RCalDB.jl	ROutput.src
	CInput.jl	CCalDB.jl	COutput.src
	IInput.jl	ICalDB.jl	IOutput.src
	TInput.jl	TCalDB.jl	TOutput.src

Demand Source Code Objective	**Demand Source Code File Name**	
Assigns values to constants that are not in the input data files	RData.jl CData.jl	IData.jl TData.jl
Initializes output variables with value for first historical year (1985)	RInitial.jl CInitial.jl	IInitial.jl TInitial.jl
Contains historical calibration equations	RCalib.jl CCalib.jl	ICalib.jl TCalib.jl
Contains code to project calibration parameters	RFuture.jl CFuture.jl	IFuture.jl TFuture.jl
Contains equations for demand projections	RDemand.jl CDemand.jl	IDemand.jl TDemand.jl
Contains equations to translate electric and natural gas demands into loads to be met by the supply sector	RLoad.jl CLoad.jl	ILoad.jl TLoad.jl

6.3. Transportation Sector Differences

The Transportation segment has additional code called during execution to handle specific equations different than the rest of the Demand module. These equations are located in the *TDemand2.jl* file in the \Engine subdirectory. This file includes the following:

1) Vehicle stock calculations: Vehicle sales, retirements, and overall stocks are estimated at an individual unit level for output. This feature is available for passenger vehicles to align vehicle stocks to client-provided input data to allow for the model to provide an estimate of vehicle sales and retirements based on the equivalent energy values. Note that the energy-level outputs are still available and still follow the same methodology as the rest of the Demand module segments.

2) Process emissions: Vehicle distance travelled is calculated to be used as the driver for process emissions (MEDriver) in several sectors. Currently this driver is applied to the RoadDust and Freight sectors for process emissions. Output process emissions variables for all transportation sectors are also populated here.

3) Low Carbon Credits: Special transportation related credits are calculated here as part of the code that simulates the Clean Fuel Standard. Currently this credit is calculated at the Tech level for electric buses.

6.4. Key Input and Output Variables of the Demand Module

The primary purpose of the demand module is to calculate long term projections of energy demand (for enduse, self-generation, and feedstock fuel usage). It also forecasts emissions, energy efficiency and capital costs (of processes and devices), investments and expenditures of the demand sector. Energy demands are projected for all economic categories, end uses, fuels, and areas represented in the model. This simulation is done for each of the residential, commercial, industrial, and transportation sectors. Exogenous forecasts are able to be incorporated into the model for specific industries if desired. The key outputs of the demand module are listed in Table 3 along with their associated variable names.

Table 3. Key Output Variables from ENERGY 2100 Demand Module

Description of Model Output	Variable Name and Definition
Energy demand (TBtu/Yr) - End-use demand - Feedstock demand - Total (by ECC and fuel)	 Dmd(Enduse,Tech,EC,Area,Year) FsDmd(Tech,EC,Area,Year) TotDemand(Fuel,ECC,Area,Year)
Emissions (Tonnes/Yr) - End-use - Feedstock (by fuel) - Feedstock (total) - Total (by ECC and fuel	 Polute(Enduse,FuelEP,EC,Poll,Area,Year) FsPol(Fuel,EC,Poll,Area,Year) NcPol(ECC,Poll,Area,Year) TotPol(ECC,Poll,Area,Year)
Energy efficiency (marginal) - Device/equipment (Btu/Btu) - Process/building ($/Btu)	 DEE(Enduse,Tech,EC,Area,Year) PEE(Enduse,Tech,EC,Area,Year)
Capital costs - Device ($/mmBtu/Yr) - Process ($/($/Yr))	 DCC(Enduse,Tech,EC,Area,Year) PCC(Enduse,Tech,EC,Area,Year)
Investments (M$/Yr) - Device investments - Process investments	 DInvTech(Enduse,Tech,EC,Area,Year) PInv(ECC,Area,Year)
Expenditures (M$/Yr) - Fuel expenditures by fuel - Fuel expenditures total - Operating and maintenance	 FuelExp(Enduse,Tech,EC,Area,Year) FuelExpenditures(ECC,Area) OMExp(ECC,Area,Year)

Primary inputs to the demand module consist of economic drivers (some of which come from an exogenously input macroeconomic forecast and others calculated in the supply module), delivered and wholesale fuel prices from the supply modules, technology characteristics (for processes and devices), such as physical lifetimes and costs), and inputs from the demand calibration. Table 4 lists exogenous inputs to the demand module.

Table 4. Exogenous Inputs Required for Demand Module

Description	Variable Name and Definition
Historical energy demand (TBtu/Yr)	
- End-use	xDmd(Enduse,Tech,EC,Area,Year)
- Feedstock	xFsDmd(Tech,EC,Area,Year)
- Steam generation	xStDmd(FuelEP,Area,Year)
Macroeconomic Indicators (Historical and Future)	
- Gross Output (1985 M$/Yr)	xGO(ECC,Area,Year)
- Gross Regional Product (Real M$/Yr)	xGRP(Area,Year)
- Population (Millions)	xPopT(Area,Year)
- Personal Income (Real M$/Yr)	xRPI(Area,Year)
- Households (Number)	xHHS(ECC,Area,Year)
- Floor Space per Unit (Sq Units/Building)	FSUnit(ECC,Area,Year)
- Inflation Index ($/$)	xInflation(Area,Year)
- Exchange Rate($CN/$US)	xExchangeRateNation(Nation,Year)
Physical life of production capacity (Yrs)	PCPL(ECC,Area,Year)
Prices (Historical and Future)	
- Wholesale Fuel Price ($/mmBtu)	xENPN(Fuel,Nation,Year)
- Base Delivered Fuel Price ($/mmBtu)	xFPBaseF(Prices,Area,Year)
Emissions coefficients (Tonnes/TBtu)	
- Energy-related (GHG)	POCX(Enduse,FuelEP,EC,Poll,Area)
- Energy-related (CAC)	xEnFPol(Enduse,FuelEP,EC,Poll,Area,Year)
- Feedstock	FsPOCX(Fuel,Tech,EC,Poll,Area,Year)
- Process	MEPOCX(ECC,Poll,Area,Year
Device characteristics (in one initialization year)	
- Scrappage Rate (%)	DPLV(Enduse,Tech,EC,Area,Vintage,Year)
- Device Capital Cost ($/mmBtu/Yr)	xDCC(Enduse,Tech,EC,Area,Year)
- Historical Device Efficiency (Btu/Btu)	xDEE(Enduse,Tech,EC,Area,Year)
- Maximum Device Efficiency (Btu/Btu)	DEM(Enduse,Tech,EC,Area)
- Device Efficiency Standards (Btu/Btu)	DEStd(Enduse,Tech,EC,Area,Year)
Process characteristics (in one initialization year)	
- Process Energy Capital Cost ($/($/Yr))	xPCC(Enduse,Tech,EC,Area,Year)
- Maximum Process Efficiency ($/mmBtu)	PEM(Enduse,EC,Area)
- Process Efficiency Standards ($/Btu)	PESTD(Enduse,Tech,EC,Area,Year)

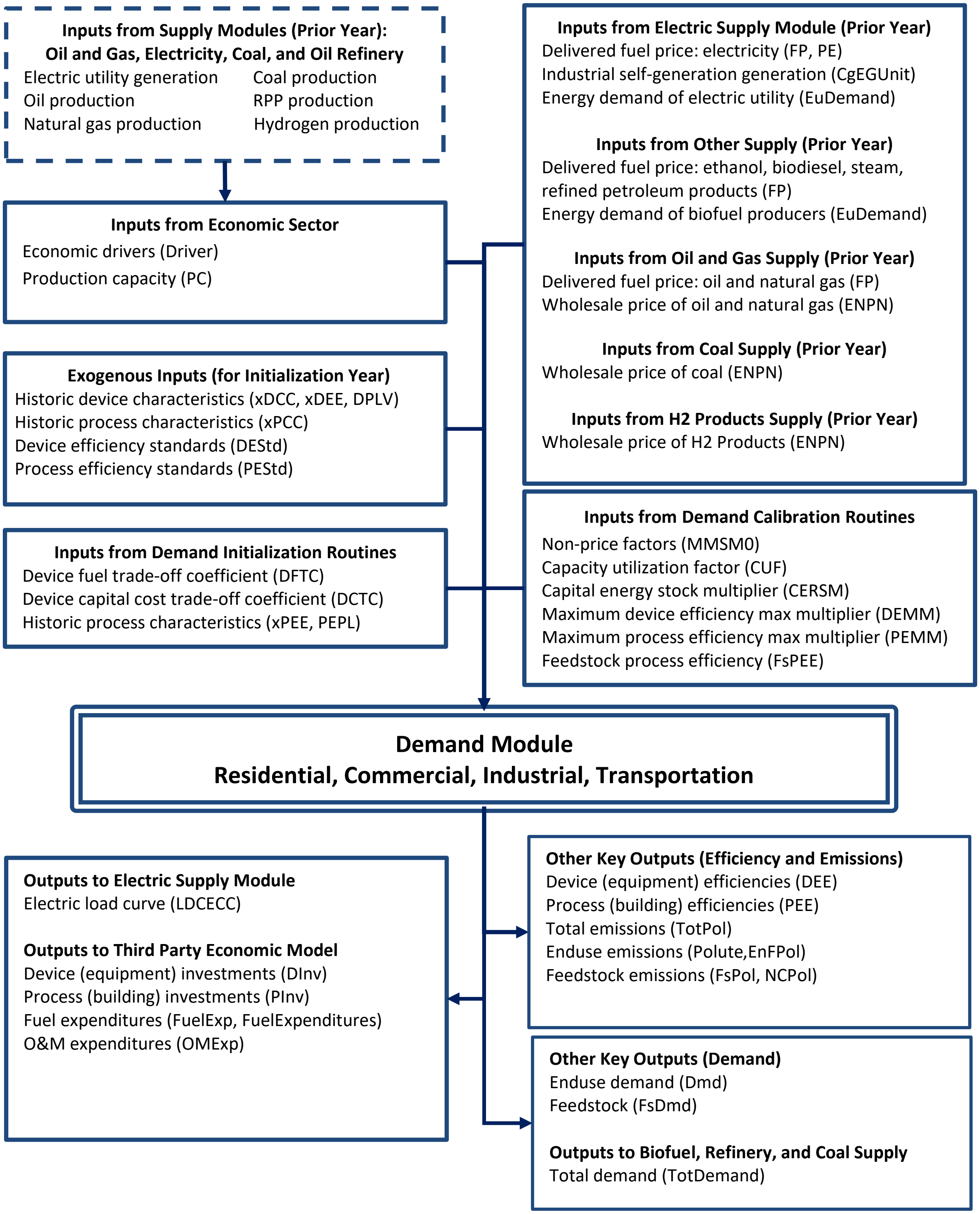
Variables that are calculated during the initialization and historical calibration and used as input to the demand equations are listed in Table 5.

Table 5. Key Inputs from ENERGY 2100

Description	Variable Name and Definition
Key Inputs from Initialization	
Process and device characteristics (derived for technologies where values unavailable) - Historical Marginal Process Efficiency ($/Btu) - Historical Marginal Device Efficiency (Btu/Btu) - Physical Life of Process Requirements (Yrs)	 xPEE(Enduse,Tech,EC,Area,Year) xDEE(Enduse,Tech,EC,Area,Year) PEPL(Enduse,Tech,EC,Area,Year)
Capital cost and fuel trade-off curve coefficients - Device Cap. Cost Trade Off Coefficient (DLess) - Process Cap. Cost Trade Off Coefficient (DLess) - Device Fuel Trade Off Coefficient (DLess) - Process Fuel Trade Off Coefficient (DLess)	 DCTC(Enduse,Tech,EC,Area,Year) PCTC(Enduse,Tech,EC,Area,Year) DFTC(Enduse,Tech,EC,Area,Year) PFTC(Enduse,Tech,EC,Area,Year)
Inputs from Calibration	
Calibration variables for energy demand - Market Share Non-Price Factors ($/$) - Capital Utilization Fraction ($/Yr/$/Yr) - Capital Energy Requirement Mult. (Btu/Btu) - Feedstock Process Efficiency ($/mmBtu)	 MMSM0(Enduse,Tech,EC,Area,Year) CUF(Enduse,Tech,EC,Area,Year) CERSM(Enduse,EC,Area,Year) FSPEE(Tech,EC,Area,Year)
Calibration variables for energy efficiency - Process Eff. Max. Multiplier ($/Btu)/($/Btu) - Device Eff. Max Multiplier (Btu/Btu)	 PEMM(Enduse,Tech,EC,Area,Year) DEMM(Enduse,Tech,EC,Area,Year)
Inputs from Economic Processing Module	
- Economic Driver (Various Millions/Yr) - Production Capacity (M$/Yr) - Production Capacity by Enduse and Tech (M$/Yr)	Driver(ECC,Area,Year) PC(ECC,Area,Year) EUPC(Enduse,Tech,Age,EC,Area,Year)
Inputs from Supply Module	
- Primary Oil Production (TBtu/Yr) - Primary Gas Production (TBtu/Yr) - Energy Demand of Electric Utilities (TBtu/Yr) - Energy Demand of Supply Sectors (TBtu/Yr)	OAProd(Process,Area,Year) GAProd(Process,Area,Year) EuDemand(Fuel,*UtiltyGen*,Area,Y) EuDemand(F,*Supply*,A,Y)

Figure 30 illustrates the inputs and outputs to the demand module in a diagram.

Figure 22. Diagram Key Inputs and Outputs of Demand Module



6.5. Initialization and Calibration

Before forecast projections can be made, the variables are initialized for one historical starting year (typically 1985) and calibrated to the historical input data. During initialization and calibration, several supply sectors along with economic processing are initialized/calibrated before the demand sector is initialized/calibrated initialized and calibrated in order to obtain the energy prices and drivers (economic and production) that are required as input to the demand module.

The purpose of the demand sector calibration is to calculate values for “calibration” variables that are used in two key model equations: 1) the consumer choice calculation of fuel market shares (probability of choosing any particular type of fuel for a given end-use); and 2) the calculation of total end-use energy demand.

This section describes the key inputs and key outputs for the model initialization and calibration code as well as identifies the names and functions of the key functions containing model code equations.

6.5.1. Demand Initialization Code

In order to simulate the mechanisms that make up consumer energy demand (such as fuel shares, process and device energy efficiency, production capacity, process and device energy requirements), historical data are used to initialize model variables and to extract relationships that help to make projections into the future. These functions are performed during the initialization and calibration stage (before model execution of the forecast years).

The two main objectives of the demand sector initialization code are: 1) to assign values to output variables in the model initialization year (typically 1985), and 2) to derive price response trade-off curves for projecting device and process efficiencies and capital costs.

Initialization Year: Whereas most variables are initialized in the year 1985 (which is the first historical year), there are some variables in which a more recent year is chosen as the model initialization year. One such example is with process and device efficiencies and capital costs in which residential and commercial efficiency and capital cost variables are initialized in the year 2000 and transportation in 2008. Alternative initialization years are selected when reliable historical data are not available back to 1985. In the case of the device and process efficiencies, the year 2000 had the most internally consistent input data available; therefore, 2000 was set to be the initialization year for those variables. Because the efficiency and capital cost data are initialized in the year 2000, the price-response trade-off curves for efficiency and capital costs are also initialized in the year 2000. The curves initialized in the year 2000 are then assigned to all years and used to determine a calculated value for historical efficiency and capital costs based on the particular year's fuel prices. With the exception of the efficiency and capital cost related variables, the initialization routines generally assign a value in a single initialization year for model calculated variables.

Initialization Functions: Eight functions perform the demand sector initialization. The names of these initialization functions and a brief description of each of their objectives are listed in Table 6. For detailed descriptions of these functions, see the *Appendix 6. Initialization Functions* and Equations section at the end of this volume.

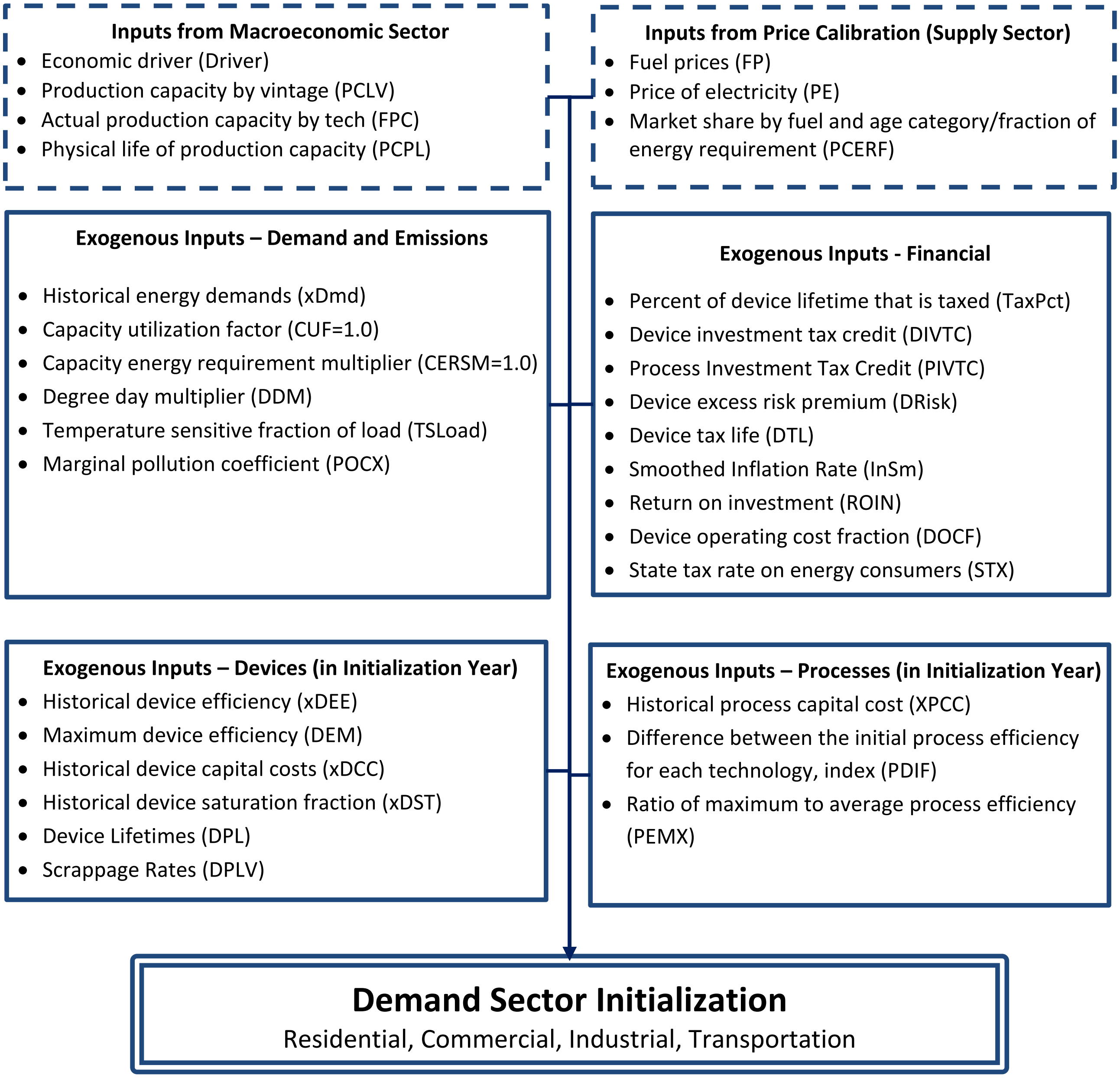
Table 6. Initialization Function Objectives

	Function Name	Function Objective
1.	Lifetimes	Initializes lifetimes and several other variables for use in the rest of the demand module.
2.	IPrice	Calculates the local fuel price by economic category and device capital charge rate. Local fuel prices by economic category are set by fuel prices except in the case of electric technologies, which instead use the electric price. Assigns fuel price for each economic category (ECFP).
3.	DEffCurve	Derives coefficients of trade-off curves for projecting device efficiency and capital costs based on input data provided for a single initialization year.
4.	Initial	Initializes variables required for developing process efficiency and cost curves.
5.	PEffCurve	Derives coefficients of trade-off curves for projecting process efficiency and process capital costs.
6.	PEffFuture	Assigns values for future years of process (and retrofit) capital cost and efficiency variables equal to the value in the initial year.
7.	PEffAdjust	Adjusts values for sectors or areas where driver is very small to allow for sectors that may not exist yet during initialization year.
8.	Pollution	Initializes the average pollution coefficients by setting equal to the marginal in the initial year.

Key Inputs and Outputs: The key inputs to the demand sector initialization routines by type of input are shown in Figure 15. The supply sector initialization and calibration routines (to obtain a value for energy prices) and economic processing (to assign values to model drivers) occurs before the demand initialization and calibration. Energy prices and drivers are key inputs to the demand sector initialization routines along with many exogenous input variables covering demand, emissions, and financial exogenous historical input data. Due to the lack of data availability related to marginal efficiencies and capital costs historically, exogenous historical inputs related to process and device technology characteristics are input for one initialization year. For efficiency data that is not available, estimates are determined based on the historical input data that are available. For example, note that exogenous historical device efficiency (xDEE) are inputs; however, historical process efficiency is calculated (PEE). By convention, variables that are assigned an "x" as a prefix, such as xDEE, represent exogenous input data.

The key outputs from the initialization routines are shown in Figure 15. Most outputs are simple assignments of exogenously input values to a model variable for the initialization year or simple transformations of input variables. For example, device energy efficiency (DEE) is an output and is simply an assignment of the value of the exogenous device energy efficiency, xDEE, to the model variable, DEE, in the initialization year. The exceptions are the coefficients of the device and process efficiency price-response trade-off curves. These coefficients are calculated in the initialization year then held constant across all other years.

Figure 24. Inputs to Demand Initialization



Inputs from Macroeconomic Sector

- Economic driver (Driver)
- Production capacity by vintage (PCLV)
- Actual production capacity by tech (FPC)
- Physical life of production capacity (PCPL)

Inputs from Price Calibration (Supply Sector)

- Fuel prices (FP)
- Price of electricity (PE)
- Market share by fuel and age category/fraction of energy requirement (PCERF)

Exogenous Inputs – Demand and Emissions

- Historical energy demands (xDmd)
- Capacity utilization factor (CUF=1.0)
- Capacity energy requirement multiplier (CERSM=1.0)
- Degree day multiplier (DDM)
- Temperature sensitive fraction of load (TSLoad)
- Marginal pollution coefficient (POCX)

Exogenous Inputs - Financial

- Percent of device lifetime that is taxed (TaxPct)
- Device investment tax credit (DIVTC)
- Process Investment Tax Credit (PIVTC)
- Device excess risk premium (DRisk)
- Device tax life (DTL)
- Smoothed Inflation Rate (InSm)
- Return on investment (ROIN)
- Device operating cost fraction (DOCF)
- State tax rate on energy consumers (STX)

Exogenous Inputs – Devices (in Initialization Year)

- Historical device efficiency (xDEE)
- Maximum device efficiency (DEM)
- Historical device capital costs (xDCC)
- Historical device saturation fraction (xDST)
- Device Lifetimes (DPL)
- Scrappage Rates (DPLV)

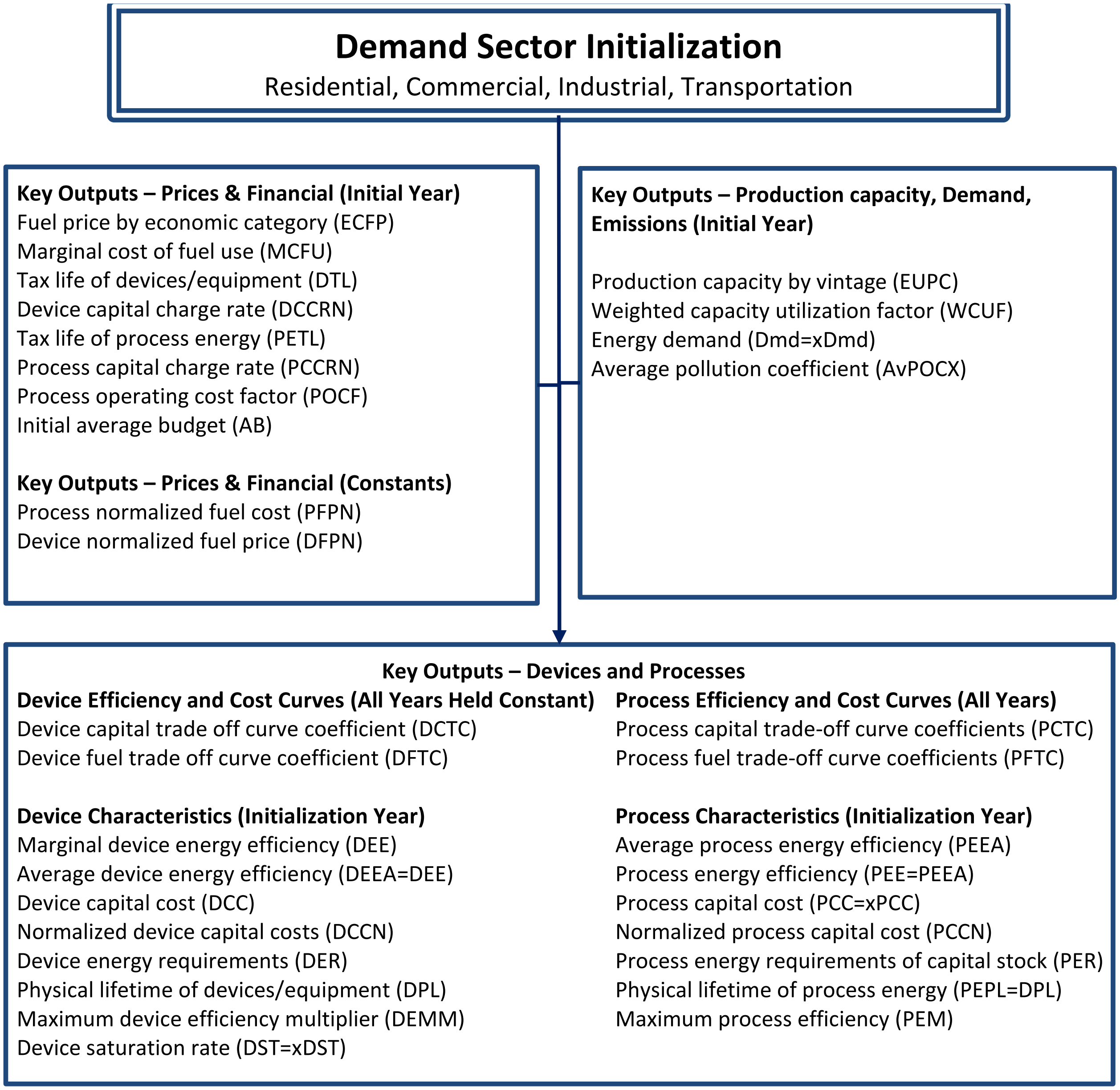
Exogenous Inputs – Processes (in Initialization Year)

- Historical process capital cost (XPCC)
- Difference between the initial process efficiency for each technology, index (PDIF)
- Ratio of maximum to average process efficiency (PEMX)

Demand Sector Initialization

Residential, Commercial, Industrial, Transportation

Figure 25. Outputs from Demand Initialization



Demand Sector Initialization
Residential, Commercial, Industrial, Transportation

Key Outputs – Prices & Financial (Initial Year)
Fuel price by economic category (ECFP)
Marginal cost of fuel use (MCFU)
Tax life of devices/equipment (DTL)
Device capital charge rate (DCCRN)
Tax life of process energy (PETL)
Process capital charge rate (PCCRN)
Process operating cost factor (POCF)
Initial average budget (AB)

Key Outputs – Prices & Financial (Constants)
Process normalized fuel cost (PFPN)
Device normalized fuel price (DFPN)

Key Outputs – Production capacity, Demand, Emissions (Initial Year)

Production capacity by vintage (EUPC)
Weighted capacity utilization factor (WCUF)
Energy demand (Dmd=xDmd)
Average pollution coefficient (AvPOCX)

Key Outputs – Devices and Processes

Device Efficiency and Cost Curves (All Years Held Constant)
Device capital trade off curve coefficient (DCTC)
Device fuel trade off curve coefficient (DFTC)

Device Characteristics (Initialization Year)
Marginal device energy efficiency (DEE)
Average device energy efficiency (DEEA=DEE)
Device capital cost (DCC)
Normalized device capital costs (DCCN)
Device energy requirements (DER)
Physical lifetime of devices/equipment (DPL)
Maximum device efficiency multiplier (DEMM)
Device saturation rate (DST=xDST)

Process Efficiency and Cost Curves (All Years)
Process capital trade-off curve coefficients (PCTC)
Process fuel trade-off curve coefficients (PFTC)

Process Characteristics (Initialization Year)
Average process energy efficiency (PEEA)
Process energy efficiency (PEE=PEEA)
Process capital cost (PCC=xPCC)
Normalized process capital cost (PCCN)
Process energy requirements of capital stock (PER)
Physical lifetime of process energy (PEPL=DPL)
Maximum process efficiency (PEM)

6.5.2. Demand Calibration Code

The main objective of the demand calibration is to estimate calibration coefficients from the historical data. Although several methods are used, one method of calculating the coefficients is to apply model equations historically then back-calculating the unknown values given known historical values.

Procedurally, several calibration routines are executed more than once since the model has specified some historical policies which require a baseline (for example a 25% improvement in efficiency). In that case, the first calibration generates the baseline while the second calibration includes the policy as part of the calibration.

The main objective of the initialization and calibration routines is to extract relationships from the historical data for use making projections of: 1) energy efficiency choices (process and device); 2) fuel market share choices; and 3) energy demand utilization factors.

Outputs from the demand calibration include variables sent to the demand execution code to project marginal fuel market share (for end uses and self-generation), energy demand (enduse, self-generation, and feedstock), and energy efficiency (devices and processes).

Key outputs include:

Used to calculate fuel shares/marginal market shares

- Marginal market share multiplier/non-price factor (MMSM0)
- Demand Fuel/Tech Fraction Non-Price Factor (DmFracMSM0)

Used to calculate end use demand, self-generation demand, and feedstock demand

- Capacity utilization factor (CUF)
- Capital energy requirement multiplier (CERSM)
- Feedstock process efficiency (FsPEE)

Used to calculate device efficiency and process efficiency

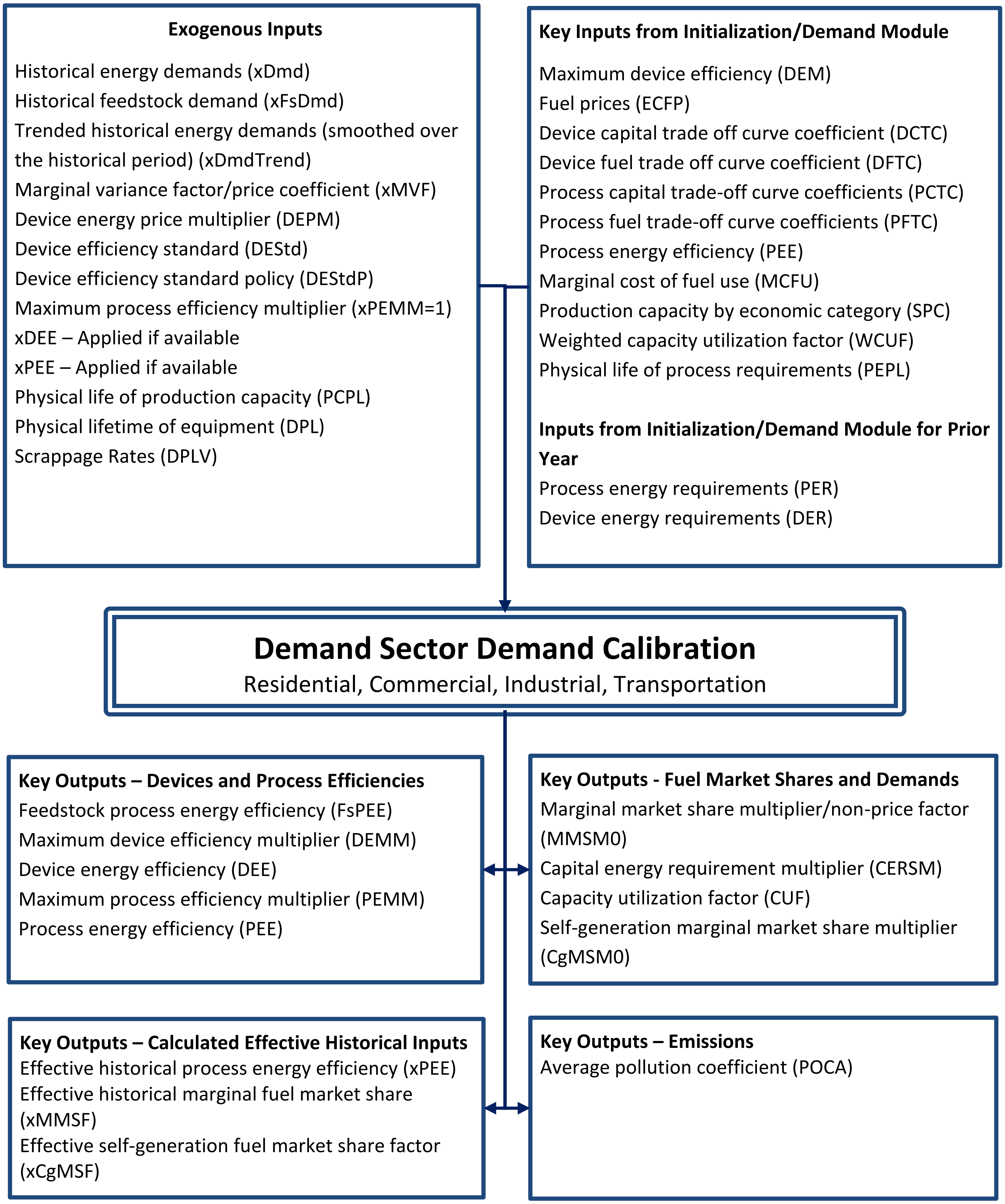
- Maximum device efficiency multiplier (DEMM)
- Maximum process efficiency multiplier (PEMM)

Other variables calculated during calibration:

- Effective historical process energy efficiency (xPEE)
- Effective historical marginal fuel market share (xMMSF)
- Device energy efficiency (DEE)
- Process energy efficiency (PEE)
- Average pollution coefficient (POCA)

Figure 17 shows a diagram of key inputs to and outputs from the demand calibration routines.

Figure 26. Inputs and Outputs of Demand Calibration



Exogenous Inputs

Historical energy demands (xDmd)
Historical feedstock demand (xFsDmd)
Trended historical energy demands (smoothed over the historical period) (xDmdTrend)
Marginal variance factor/price coefficient (xMVF)
Device energy price multiplier (DEPM)
Device efficiency standard (DEStd)
Device efficiency standard policy (DEStdP)
Maximum process efficiency multiplier (xPEMM=1)
xDEE – Applied if available
xPEE – Applied if available
Physical life of production capacity (PCPL)
Physical lifetime of equipment (DPL)
Scrappage Rates (DPLV)

Key Inputs from Initialization/Demand Module

Maximum device efficiency (DEM)
Fuel prices (ECFP)
Device capital trade off curve coefficient (DCTC)
Device fuel trade off curve coefficient (DFTC)
Process capital trade-off curve coefficients (PCTC)
Process fuel trade-off curve coefficients (PFTC)
Process energy efficiency (PEE)
Marginal cost of fuel use (MCFU)
Production capacity by economic category (SPC)
Weighted capacity utilization factor (WCUF)
Physical life of process requirements (PEPL)

Inputs from Initialization/Demand Module for Prior Year

Process energy requirements (PER)
Device energy requirements (DER)

Demand Sector Demand Calibration

Residential, Commercial, Industrial, Transportation

Key Outputs – Devices and Process Efficiencies

Feedstock process energy efficiency (FsPEE)
Maximum device efficiency multiplier (DEMM)
Device energy efficiency (DEE)
Maximum process efficiency multiplier (PEMM)
Process energy efficiency (PEE)

Key Outputs - Fuel Market Shares and Demands

Marginal market share multiplier/non-price factor (MMSM0)
Capital energy requirement multiplier (CERSM)
Capacity utilization factor (CUF)
Self-generation marginal market share multiplier (CgMSM0)

Key Outputs – Calculated Effective Historical Inputs

Effective historical process energy efficiency (xPEE)
Effective historical marginal fuel market share (xMMSF)
Effective self-generation fuel market share factor (xCgMSF)

Key Outputs – Emissions

Average pollution coefficient (POCA)

The model code that calculates the calibration variables for each of residential, commercial, industrial, and transportation is contained in RCalib.jl, CCalib.jl, ICalib.jl, and TCalib.jl files.

Table 7 identifies the calibration variables used to calibrate each of the demand equations as well as identifies the location of the source code.

Table 7. Demand Module Calibration Variables

Equation to be Calibrated	Calibration Variable	Calibration Code Location
Marginal fuel market share (MMSF)	- MMSM0(Enduse,Tech,EC,Area,Year) – Marginal Fuel Market Share Multiplier/Non-Price Factor ($/$)	- Residential (RCalib.jl) - Commercial (CCalib.jl) - Industrial (ICalib.jl) - Transportation (TCalib.jl)
Enduse energy demand (Dmd)	- CERSM(Enduse,EC,Area,Year) - Capital Energy Requirement Mult. (Btu/Btu) - CUF(Enduse,Tech,EC,Area,Year) - Capacity Utilization Factor ($/Yr/$/Yr)	
Feedstock energy demand (FsDmd)	- FsPEE(Tech,EC,Area,Year) - Feedstock Process Efficiency ($/mmBtu)	
Device and process energy efficiency (DEE, PEE)	- DEMM(Enduse,Tech,EC,Area,Year) - Device Efficiency Max Multiplier (Btu/Btu) - PEMM(Enduse,Tech,EC,Area,Year) - Process Eff. Max. Multiplier ($/Btu)/($/Btu)	

The methodologies used to calculate each of the demand calibration variables listed above (MMSM0, CERSM, CUF, FsPEE, DEMM, and PEMM) in the historical period are described in the sections below.

MMSM0 - Marginal Fuel Market Share Multiplier (non-price factor)

As described in Volume 3, historical values of MMSM0 are estimated by first calculating an implied value of the marginal fuel market shares (xMMSF) from the historical data (actual marginal are not available, average market shares are used to estimate the marginal). The estimated values for xMMSF, together with the historical cost of the technologies, are then used to back-calculate an estimated historical value for MMSM0 using the market share equation:

$$xMMSF\,(i) = \frac{e^{(MMSM0+\mathrm{M}VF*Ln(\frac{MCFU0(i)}{MCFU(k)})}}{\sum_i^j e^{(MMSM0+\mathrm{M}VF*Ln\left(\frac{MCFU0(i)}{MCFU(k)}\right))}}$$

Values of MMSM0 is calculated for each area, economic category and enduse across all technology types (for example: electricity, gas, oil, biomass, LPG, geothermal, and heat pump within the residential sector). An example of the values of MMSM0 output from ENERGY 2100 for Ontario, residential single family, space heat by technology type is shown in the chart below.

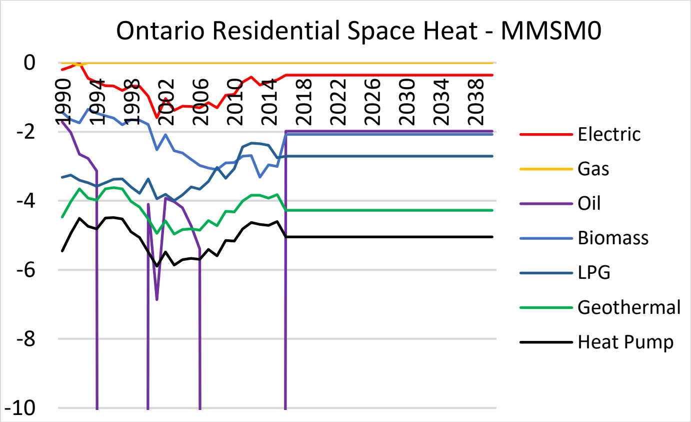


Figure 27. Marginal Fuel Market Share Multiplier - MMSM0 ($/$)

Interpretation: MMSM0 (non-price factor) values range from 0 (no resistance to technology) to < -10 (100% resistance to technology). The larger the negative value, the higher the resistance of the market to purchase the specified technology. MMSM0 values of 0 indicate there is no market resistance to the technology, and in those cases, only price is a factor in the consumer choices.

CERSM - Capital Energy Requirement Multiplier

CERSM is calculated historically using the method of least squares such that the square of the difference between the calculated historical demand (Dmd) and a trend line of historical demand is minimized. The calculated historical demand is based on the model's demand equation and with an initial assumption of CUF=1.0. The basic demand equation (a few factors are omitted for simplification) is listed below.

$$Dmd = DER*CUF*CERSM*DDM$$

Where:

DER – device energy requirements

CUF – capacity utilization factor

DDM – degree day multiplier which is applied to the temperature sensitive load

The historical and forecast values of CERSM for Ontario residential space heat are illustrated in the chart below.

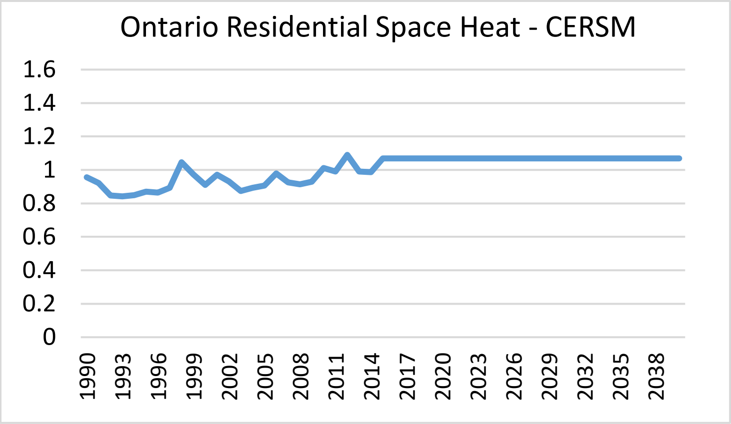


Figure 28. Capital Energy Requirement Multiplier - CERSM (Btu/Btu)

CUF – Capacity Utilization Factor

Given the value of CERSM calculated above and the known, actual historical demands (xDmd), CUF is back-calculated using the demand equation. The basic equation (a few factors have been omitted for simplification) is shown below:

$$CUF = xDmd/(DER*CERSM*DDM)$$

For illustration, the historical and forecast values of CUF are shown in the chart below for Ontario, residential space heat by type of technology (electricity, natural gas, etc.).

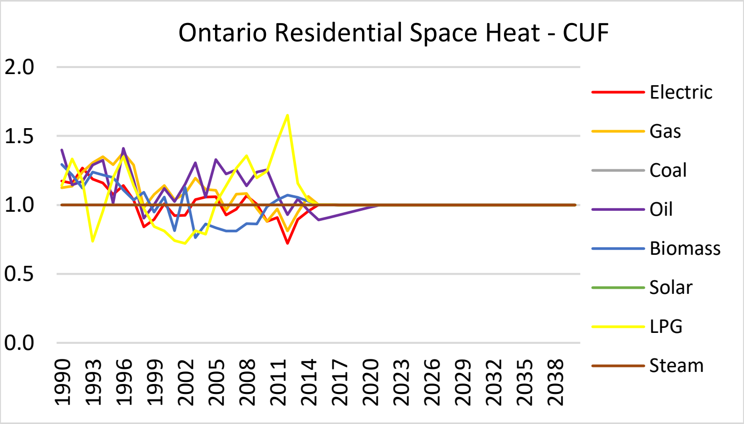


Figure 29. Capacity Utilization Factor – CUF ($/Yr)/($/Yr)

Interpretation of Capacity Utilization Factor (CUF):

The capacity utilization factor is greater than 1.0 when the capital stock cannot grow fast enough to meet the reported energy demands. This is generally due to an increase in demand when there is not a comparable increase in economic driver. This could be due to an inconsistency in the historical data, a capital stock lifetime which is too long, or fuel switching in existing capital stock.

The capacity utilization factor is less than 1.0 with normal fluctuations in energy usage. In each year there are unique situations in each household or business which will cause some change in energy demands.

The capacity utilization factor becomes very low when the capital stock is not retiring fast enough to match a drop in energy demand. This often happens with coal where the demands are dropping to nearly zero while the capital stock would be expected to have more lifetime. We now adjust for this by retiring excess capital stock in the first year of the forecast.

When the capacity utilization factor is very high, we adjust for this by adding capital stock in the first year of the forecast to bring the CUF back to 1.0

The capacity utilization factor is equal to 1.0 in the long run forecast either by the capital stock adjustment or trending back to 1.0.

FsPEE – Feedstock Process Efficiency

To calibrate the model's feedstock demand equation to historical feedstock demand, a feedstock process efficiency variable is calculated as the ratio of historical production capacity to historical feedstock demand. Then feedstock demand is projected into the future using a projection of FsPEE and a projection of production capacity. The equation for the historical calculation of the feedstock calibration variable is:

$$\text{FsPEE}=\text{Production Capacity}/\text{Historical Feedstock Demand}$$

As an illustration the figure below graphs the oil feedstock process efficiency (FsPEE) across Ontario's industrial sector. For the industrial sector, the oil feedstock process efficiency is calculated as industrial gross output divided by oil feedstock demand.

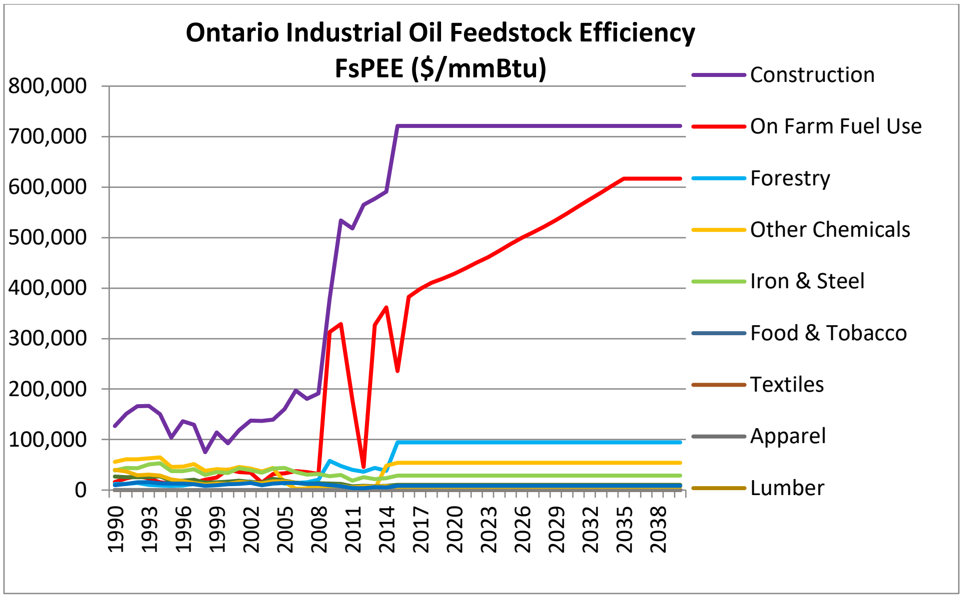


Figure 4. Feedstock Process Efficiency – FsPEE ($/mmBtu)

DEMM – Maximum Device Efficiency Multiplier

Device efficiency is calculated during the calibration process as a function of price and cost curves developed during the model initialization. Where actual input data (xDEE) or efficiency standards (DEStd, DEStdP) differ from the calculated value, DEMM is used as an adjustment to shift the device efficiency curve such that the calculated DEE matches the input xDEE. The multiplier is assigned an initial value of 1.0, and is adjusted by the maximum of xDEE or the standards (DEStd, DEStdP) divided by the endogenously calculated efficiency (DEE). This equation is expressed below where DEE is the calculated value and XDEE, DEStd, and DEStdP represent the exogenous inputs of efficiency and efficiency standards:

$$DEMM=DEMM*xmax(XDEE,DEStd,DEStdP)/DEE$$

Refrigerators are an example of an enduse in which exogenous marginal device efficiencies (xDEE) are available. The chart below illustrates the historical and forecast values of the DEMM calibration variable for Ontario, residential refrigerators. Historical values of DEMM calibrate the endogenously calculated device efficiency (DEE) to the historical input efficiency (xDEE). Forecasted values can be derived from historical data or optionally be aligned to other expectations. As an example, DEMM’s values in the forecast period in the charge reflect the device efficiency trends published in the U.S. EIA’s Annual Energy Outlook forecast.

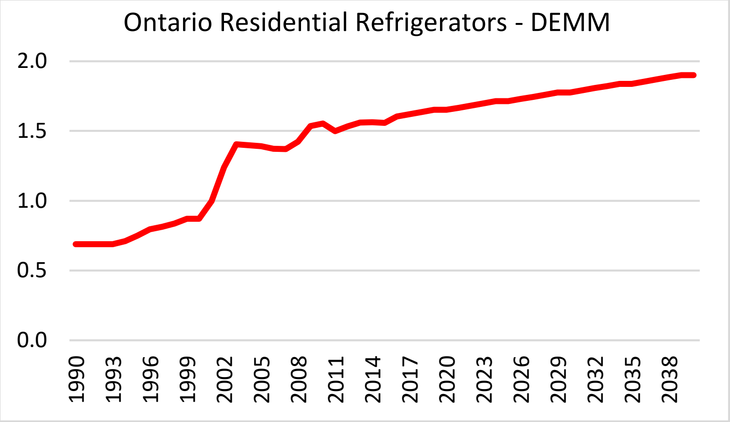


Figure 30. Device Efficiency Maximum Multiplier – DEMM (Btu/Btu)

PEMM – Maximum Process Efficiency Multiplier

Process efficiency is calculated during the calibration process as a function of price and cost curves developed during the model initialization. Where actual input data (xPEE) or efficiency standard data (PEStd, PEStdP) differs from the calculated value, PEMM is used as an adjustment to shift the process efficiency curve to shift the endogenous value to the input data. This is expressed simply below with PEEBeforeStd as the calculated value and xPEEBeforeStd, PESTD, and PESTDP being efficiency and efficiency standard inputs:

$$PEMM = PEMM*xPEEBeforeStd/PEEBeforeStd$$

$$PEE = PEEBeforeStd*PEMM$$

$$PEE = xmax(PEE,PEStd,PEStdP)$$

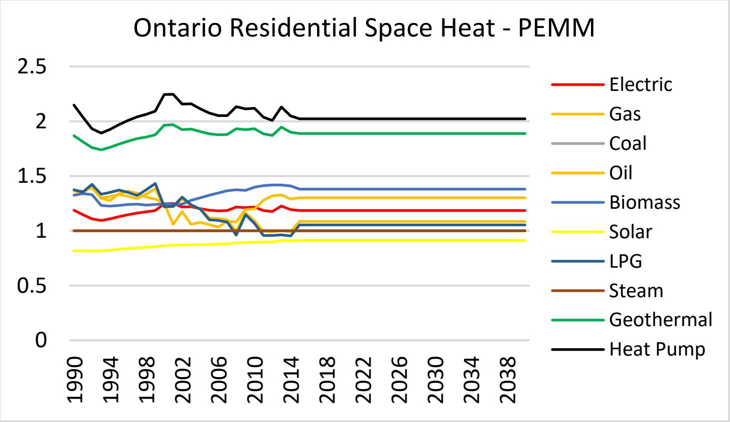


Figure 31. Process Efficiency Maximum Multiplier – PEMM (Btu/Btu)

6.5.3. Function Initial

Assigns exogenous values to key demand model variables and calculates a feedstock demand efficiency as a ratio of historical production capacity to historical feedstock demand.

Function Initial: Estimate basic demand parameters.

Key Inputs

Exogenous inputs:

- xPEMM (Enduse,Tech,EC,Area,Year) 'Process Efficiency Maximum Multiplier (($/Btu)/($/Btu))'
- xMVF (Enduse,Tech,EC,Area,Year) 'Market Share Variance Factor ($/$)
- xFsDmd (Tech,EC,Area,Year) 'Feedstock Energy (TBtu/Yr)'

Inputs from demand sector Initialization routines:

- WCUF (EC,Area) 'Capacity Utilization Factor Weighted by Output'

Inputs from Economic Processing module :

- PC (ECC,Area,Year) 'Production Capacity (M$/Yr)'

Key Outputs

- PEMM (Enduse,Tech,EC,Area,Year) 'Process Efficiency Maximum Multiplier (($/Btu)/($/Btu))
- MVF (Enduse,Tech,EC,Area,Year) 'Market Share Variance Factor ($/$)'
- FsPEE (Tech,EC,Area,Year) 'Feedstock Process Efficiency ($/mmBtu)'
- SPC (EC,Area) 'Total Production Capacity (M$/Yr)’

Key Equations

- PEMM=XPEMM
- MVF=XMVF
- SPC(EC,Area)=sum(ECC)(PC(ECC,Area)*ECCMap(EC,ECC))
- FsPEE=SPC*WCUF/XFsDmd

6.5.4. Function CalDEMM

This function calculates an implied historical device energy efficiency (DEE) before any standards were in place based on the device efficiency trade-off curve derived during initialization. It also calculates a calibration variable, the maximum device efficiency multiplier (DEMM), representing the ratio difference between the calculated DEE the effective historical efficiency (maximum of an exogenous device efficiency (XDEE) if it exists, a historical device efficiency standard (DEStd) or a device efficiency standard policy (DEStdP) if it exists).

Function CalDEMM

Key Inputs

Exogenous inputs:

- xPEMM (Enduse,Tech,EC,Area,Year) 'Process Efficiency Maximum Multiplier (($/Btu)/($/Btu))'
- xMVF (Enduse,Tech,EC,Area,Year) 'Market Share Variance Factor ($/$)
- xFsDmd (Tech,EC,Area,Year) 'Feedstock Energy (TBtu/Yr)'

Inputs from demand sector Initialization routines:

Key Outputs

- DEMM (Enduse,Tech,EC,Area,Year) ‘Maximum Device Efficiency Multiplier (btu/btu)’

Key Equations

The device efficiency maximum multiplier is set equal to the exogenously input value adjusted by the prior year’s ratio of DEMM to the exogenous value (XDEMMPrior)

- DEMM=XDEMM*DEMMPrior/XDEMMPrior

- DEEBeforeStd=DEM*DEMM*(1/(1+(ECFP/Infla*DEPM/DFPN)**DFTC))

Device energy efficiency is the maximum of the efficiency before any standards and existing efficiency standards or policy efficiency standard.

- DEE =xmax(DEEBeforeStd,DEStd,DEStdP)
- DEMM=xmax(DEMM,DEE/DEMM/(DEM*0.98))

- DEMM=DEMM*xmax(XDEE,DESTD,DESTDP)/DEE
- DEE=xmax(XDEE,DESTD,DESTDP)

Function CalPEMM: Calibrate Process Technology

This function calculates an implied historical process energy efficiency (PEE) before any standards were in place based on the process efficiency trade-off curve derived during initialization. It also calculates a calibration variable, the maximum process efficiency multiplier (PEMM), representing the ratio difference between the calculated PEE the effective historical efficiency (maximum of an exogenous device efficiency (xPEE) if it exists, a historical device efficiency standard (PEStd) or a device efficiency standard policy (PEStdP) if it exists). An adjustment is applied if the enduse is ‘AC’ to account for the linkage between air conditioning and heat producing devices.

Function CalPEMM

Key Inputs

Exogenous inputs:

- xPEMM (Enduse,Tech,EC,Area,Year) 'Process Efficiency Maximum Multiplier (($/Btu)/($/Btu))'
- xMVF (Enduse,Tech,EC,Area,Year) 'Market Share Variance Factor ($/$)

- xFsDmd (Tech,EC,Area,Year) 'Feedstock Energy (TBtu/Yr)' **Inputs from demand sector Initialization routines:**
Key Outputs
- PEMM (Enduse,Tech,EC,Area,Year) ‘Maximum Process Efficiency Multiplier (Btu/Btu)’
Key Equations
xPEE=xmax(xPEE,PEE) PEMM=PEMM*xPEE/PEE PEE=PEE*PEMM * If space heating (Heat) and air conditioning (AC) both exist, then compute process efficiency multiplier (PEMM) for air conditioning. PEE(AC,Tech,EC,Area)=sum(TE)(PEE(Heat,TE,EC,Area)* PER(Heat,TE,EC,Area))/sum(TE)(PER(Heat,TE,EC,Area))/ (CHR(EC,Area)*CHRM(EC,Area))*PEMM(AC,Electric,EC,Area) xPEE = xmax(xPEE,PEE) PEMM = PEMM*xPEE/PEE * If space heating (Heat) does not exist, but air conditioning (AC) does exist, then compute process efficiency multiplier (PEMM) for air conditioning. xPEE = xmax(xPEE,PEE) PEMM = PEMM*xPEE/PEE PEE = PEE*PEMM

Function Coefficients

Calculate Calibration Parameters**.** Multinomial logit calculation of MMSF – based on back calculating MMSM0 given all other factors known historically and marginal variance fact an exogenous input.

Inputs

- Historical, exogenous energy demands (XDmd)
- Trended historical energy demands (smoothed over the historical period) (XDmdTrend)
- Marginal variable factor (MVF)
- Marginal cost of fuel use (MCFU)

Outputs

- Historical marginal market share factor (XMMSF)
- Marginal market share multiplier/non-price multinomial logit coefficient (MMSM0)
- Capital energy requirement multiplier (CERSM)
- Capacity utilization factor (CUF)

Equations

1. Estimate historical marginal market share (XMMSF) from other variables input (or calculated) for the historical period - calculate implied historical additions and retirements of device energy and process energy based on relationships between historical production capacity, device saturation levels, historical energy demand (trended), process efficiency, device efficiency, production capacity additions, process and device energy lifetimes.

2. Marginal fuel market share for each historical year: Back Solve for MMSM0 and normalize one fuel to Zero. Using multinomial logit function equation for probability of choosing a certain fuel from a set of fuels given MCFU.

3. Capacity energy requirement multiplier: Solve for CERSM using Dmd equation and using calculated value for MMSM0 and trended historical energy demands.

4. Solve for CUF by using actual annual values for historical energy demands rather than smoothed values.

Self-generation Calibration

Inputs

- Electric price (ECFP)
- Prior year's
- Self-generation generating capacity, sector-level (CgGCSector)
- Self-generation utilization multiplier (CgUMS)
- Device energy requirements (DER)

Outputs: CgCUF, CgMSM0, XCgMSF

6.5.5. Load Shape Calibration

The load shape calibration creates load shape factors from the historical data, adjusting for differences between calculated load shapes and exogenously input load shapes. These load shape factors will be sent as input to the model execution (Loads) routines that translate energy demands from the demand code into loads for input to the electric and natural gas supply sectors.

Two functions calibrate the load shapes from the historical data: Function LoadShape and Function Normalize. Function LoadShape simply assigns exogenously specified load shape factors to model output variables, and Function Normalize modifies the output value of load shape factor by an adjustment factor (created to reconcile calculated loads with exogenously specified loads) obtained from the supply calibration then normalizes the resulting load shape factors. The key inputs and outputs are listed below.

Key Inputs (from Supply Calibration)

- Base adjustment to account for differences between calculated historical loads and exogenous loads (BaseAdj)
- Exogenous load shape factor by month for peak, average, minimum days (XLSF)
- Exogenous self-generation load shape factor (XCgLSF)
- Exogenous self-generation sold to the grid load shape factor (XCgLSFSold)

Key Outputs

- Electricity load shape factor by month for peak, average, minimum days (LSF)
- Self-generation load shape factor (CgLSF)
- Self-generation sold to the grid load shape factor CgLSFSold
- Natural gas daily use factor (DUF)

6.5.6. Calibration Projection

Overview

After the calibrated variables have been calculated for the historical years, a method needs to be determined for projecting these values into the future. Following the calibration routines, a set of routines is called from the *.Future.jl files to forecast the calibrated variables. There are a multitude of potential projection methods, and ENERGY 2100 allows the user to select the specific methodology to be used. Different methods are able to be selected for different areas, economic sectors, technologies, or end uses. The default setting for most variables is to hold the value in the last historical year constant through the projection period.

The method of projecting the calibration variables can be selected based on a set of exogenously-specified calibration control variables. The initial values of the calibration control variables are specified inside the *.Data.jl files. Depending on type of calibration variable, each variable is assigned independent variables, weights, and a method based on the "Y" name of the variable.

The outputs from the Future routines consist of future values of the demand calibration variables. Recently, the projection code has been simplified to include only the currently used statistical methods. Other methods can be identified, tested, and applied by the user. The current options of projection methods built into the model are shown in Table 8.

Table 8. Calibration Control Variable Assignments

Control Variables	User-Set Value	Methodology Options Of Projecting Calibration Variable into Future
YPEMM YCERSM YCUF YDEMM YFsPEE YMMSM	=3	Assign future values equal to last historical year's value
	=4	Assign future values equal to the mean of the historical values
	=5	Trend the future values toward an exogenous value
	=6	Develop a trend line from the historical values
	=7	Develop an exponential trend line from the historical values
	=16	Set future marginal value equal to last historical average value using last historical year input values
	=17	Set future marginal value equal to last historical average value using trended input values.

The main objective of the demand calibration is to estimate calibration coefficients from the historical data. The equations that are calibrated to historical data include marginal fuel market share, energy demand (enduse and feedstock), and energy efficiency (devices and processes).

Note that the projection code is currently being reviewed and revised. Future model versions could have differences not covered in this document.

Outliers

Methods used to project future values are based on historical datapoints. The historical inputs for these calculations optionally can be adjusted to reduce or eliminate outlier data points in the set. This is executed using the **Outlier** function, which applies weights to the equation input values to reduce or eliminate their influence if they are over two standard deviations from the average of the user selected data set.

Outlier uses the calibration switch values ('Y') for the years between the first historical calibration years ('First') and the last ('Last') to determine which years are included in the adjustment. The values for the switch determine the adjustment method used for a particular year:

- 0 – Data point not included in average
- 1 – Data point included in average, but not adjusted
- 2 – Data points adjusted using user input weighting
- 3 – Data points over two standard deviations from average are removed

Projection

As summarized above, there are currently several projection methods available to be selected for any combination of modeled enduse, area, and technology in the Demand segment. Below is a brief summary of each available option for simplified variable *y* in forecasted years [Future]:

- 3 - Future values are set to the last historical value
    - *y*[Future] = *y*[Last]
- 4 – Future values are set using a simple mean value, with optional weighting
    - *y*[Future] = *y*[Historical]*Wght[Historical]/sum*(y*[Historical]*Wght[Historical])
- 5 – Future values are trended to a historical exogenous value
    - *y*[Future+5] = 1.0
    - *y*[Future] = *y*[Last] +(*y*[Future+5]-*y*[Last])/(Future5-Last)
- 6 – Future values use outputs from linear equation
    - *y*[Future] = Intercept + Slope*[Future]
- 7 - Future values use outputs from exponential linear equation
    - Equation input = log(*y*)
    - *y*[Future] = exp(Intercept + Slope*[Future])

- 16 – Future marginal market share values are set equal to last average market share value using last historical year inputs (for MMSM0)
    - y[Future] = LN(AMSF[Last] * Outputs from market share equations[Last])
- 17 – Future marginal market share values are set equal to last average market share value using trended inputs (for MMSM0)
    - y[Future] = LN(AMSF[Last] * Outputs from market share equations[Future])

6.6. Summary of Demand Code Functions

This section provides a summary description of the eleven major steps making up the demand module, providing function names, objectives, key inputs, key outputs, and key equations. More detailed descriptions of the specific model code functions, variables, and equations are provided in the Appendix.

1. The **price and pollution policy inputs** functions process price and pollution input variables for use by other routines.
2. The **device efficiency and capital costs** functions apply a trade-off curve to determine marginal device efficiencies and device capital costs. These functions also calculate a marginal cost of fuel use which is used in later routines in the determination of fuel choices.
3. The **process efficiency and capital costs** functions use a similar methodology to the device efficiency and capital cost routines of applying a trade-off curve to determine marginal process efficiencies and capital costs.
4. The **fuel market shares** functions apply consumer choice equations to determine fuel market shares for any new purchases made due to economic growth or retirement of capital stock.
5. The **energy stock changes** functions determine levels of capital stock, device energy requirements, and process energy requirements due to additions and retirements caused by changes in drivers, device saturations, and aging stock.
6. The **retrofits and conversions functions** optionally modify levels of capital stock due to retrofits and allow for consumers to convert stock to alternative fuel types (using consumer choice equations) at the end of the stock's useful life (typically only occurring with new stock additions).
7. The **enduse and feedstock demand** functions calculate enduse and feedstock energy demands and make optional adjustments due to impacts of energy efficiency programs.
8. The **self-generation demand** functions calculate self-generation energy demands (for both sector-level self-generation and unit-level self-generation specified in the electric supply module).
9. The **total demand** functions create summary demand variables representing total demand (end-use plus feedstock plus self-generation).
10. The **emissions** functions calculate energy-related emissions from the end-use, feedstock, and self-generation demands.

11. The **investments** functions calculate marginal investments in devices and processes made by the demand sector which can be used to obtain economic impacts when sent to a macroeconomic model.

Table 9 identifies the function names, objectives, key inputs, key outputs, and key equations to the functions summarized above.

Table 9. Demand Code Functions, Key Inputs, and Key Outputs

Demand Module Code	Key Inputs, Outputs, and Equations
1. PRICE AND POLLUTION POLICY INPUTS	
Objective: Function TPrice calculates the effective price of energy (ECFP) by combining the fuel price (FP, PE) with any emission charges (PCost). Function PRReductions is used with cap and trade policies and calculates emission reductions (RP) and the energy impacts of pollution reductions (RPEI) in response to emission cap and trade policies. The energy impacts of pollution reduction (RPEI) is used to adjust the calculation of energy demand. **Related Model Code:** *Function TPrice; Function PRReductions*	
Key Inputs	• Pollution costs by fuel (PCost) • Pollution coefficient reduction multiplier (RM) • Pollution coefficient (POCX) • End use energy demand (EUDem) • Delivered fuel prices (FP) • Price of electricity (PE) • Production capacity by ECC (PC)
Key Outputs	Variable mappings include: • Permit cost by technology (PCostTech) • Effective price of fuel (ECFP) • Production capacity (SPC) • Emissions coverage (ECoverage) Pollution-related calculated outputs include: • Indicated pollution reduction (IRP) • Pollution reductions (RP) • Energy Impact of Pollution Reduction (RPEI)
Key Equations	• See functions for mappings • Pollution reductions are based on cost and stock of reduction devices (dependent on type of emissions cap and trading market)
2. DEVICE EFFICIENCY AND DEVICE CAPITAL COSTS	

Demand Module Code	Key Inputs, Outputs, and Equations
Objective: *Function DMarginal* calculates the marginal efficiency (DEE) and capital costs of devices (DCC) chosen by consumers as well as the marginal cost of fuel use (MCFU). Device efficiency and capital costs are used in the calculations of device energy requirements which are ultimately needed for end use demand calculations. The marginal cost of fuel use (MCFU) is used as input to the consumer choice market share equation in *Function MShare*. *Function DDSM* adjusts the device efficiency and capital costs by consumer responses to demand side management programs when they are activated. **Related Model Code:** *Function DMarginal; Function DDSM*	
Key Inputs	Inputs from initialization: • Device fuel trade off coefficient (DFTC) Inputs from calibration: • Maximum device efficiency multiplier (DEMM) Other inputs • Fuel prices (ECFP) • Normalized fuel prices (DFPN) • Device efficiency maximum (DEM) • Multiple financial input variables, such as o Indirect costs (IdrtCost) o return on investment (ROIN), o device investment tax credits (DIVTC)) • Methodology switch for calculating DEE and DCC (DEESw, DCCSw)
Key Outputs	Intermediate key outputs: • Device capital charge rate (DCCR) • Device operating and maintenance cost (DOMC) Final key outputs • Marginal cost of fuel use (MCFU) • Device efficiency (DEE) • Device capital costs (DCC)
Key Equations	• Device efficiency and capital cost calculations can be based on fuel prices alone (DEEPrice, DCCPrice) or based on fuel prices including pollution costs (DEEPoll, DEEPoll) based on the value of DEESw and DCCSw. See function for equation details. • MCFU=DCCR*DCC+DOMC+ECFP/DEE+IdrtCost
3. PROCESS EFFICIENCY AND PROCESS CAPITAL COSTS	
Objective: *Function CMarginal* calculates the marginal efficiency (PEE) and capital costs of processes (PCC). Process efficiency and capital costs are used in the calculations of process energy requirements which are ultimately needed for end use demand calculations. **Related Model Code**: *Function CMarginal*	

Demand Module Code	Key Inputs, Outputs, and Equations
Key Inputs	<u>Inputs from initialization:</u> - Process fuel trade off coefficient (PFTC) <u>Inputs from calibration:</u> - Maximum process efficiency multiplier (PEMM) <u>Other inputs</u> - Financial variables and costs related to processes - Normalized fuel prices (PFPN) - Process efficiency maximum (PEM) - Methodology switch for calculating PEE and PCC (PEESw, PCCSw)
Key Outputs	Process capital charge rate (PCCR) Process efficiency (PEE) Process capital costs (PCC)
Key Equations	- Process efficiency and capital cost calculations can be based on fuel prices alone (PEEPrice, PCCPrice) or based on fuel prices including pollution costs (PEEPoll, PEEPoll) based on the value of PEESw and PCCSw. See function for equation details.
4. FUEL MARKET SHARES	
Objective: Determines the fuel market share fractions of new devices (marginal market share fractions) using consumer choice equations given price factors and non-price factors calculated during calibration. **Related Model Code:** *Function MShare*	
Key Inputs	<u>Inputs from demand calibration</u>: - Marginal market share multiplier/non-price parameter in consumer choice equation (MMSM0) - Marginal variance factor/price parameter (MVF) <u>Inputs from other demand routines</u>: - Marginal cost of fuel use (MCFU)
Key Outputs	Marginal market share factor (MMSF)
Key Equations	*Marginal Market Share (i) =* $\frac{e^{(Utility\ of\ Option\ i)}}{\sum_i^j e^{(Utility\ of\ Optionij)}}$ *where* *Marginal Market Share (i) =* $\frac{e^{(NonPriceFactor+VF*Ln(\frac{Price(i)}{Price(k)})}}{\sum_i^j e^{(NonPriceFactor+VF*Ln\left(\frac{Price(i)}{Price(k)}\right))}}$ *MMSF (i) =* $\frac{e^{(MMSM0+MVF*Ln(\frac{MCFU0(i)}{MCFU(k)})}}{\sum_i^j e^{(MMSM0+MVF*Ln\left(\frac{MCFU0(i)}{MCFU(k)}\right))}}$

<table>
<tr><th>Demand Module Code</th><th>Key Inputs, Outputs, and Equations</th></tr>
<tr><td colspan="2">5. ENERGY STOCK CHANGES</td></tr>
<tr><td colspan="2">Objective: Tracks marginal additions and retirements of production capacity and the related changes to process and device energy requirements. Additions and retirements are calculated based on changes in device saturation, stock growth or decline (determined from changes to the economic drivers), and replacements due to normal failures and stock aging. This function ultimately calculates the process and device energy requirements variables that are needed to calculate overall energy demands.
Functions: MStock, TStock</td></tr>
<tr><td>Key Inputs</td><td><u>Inputs calculated in economic sector</u>
• Total production capacity (SPC)
• Production capacity by end use (EUPC)
• Production capacity additions (PCA)
<u>Inputs calculated in other demand functions</u>
• Marginal cost of fuel use (MCFU)
• Device and process efficiency (DEE, PEE)
• Market share fractions for each economic class and fuel (MMSF)
<u>Other inputs</u>
• Maximum device saturation (DSTM)
• Production capacity lifetime (PCPL)</td></tr>
<tr><td>Key Outputs</td><td>Device saturation (DST)
Production capacity
• Capital stock additions and retirements (EUPCAPC, EUPCRPC)
• Total production capacity (EUPC)
Device energy requirements
• Additions and retirements (DERAPC, DERRPC)
• Total device energy requirements (DER)
Process energy requirements
• Additions and retirements (PERAPC, PERRPC)
• Total process energy requirements (PER)</td></tr>
</table>

Demand Module Code	Key Inputs, Outputs, and Equations
Key Equations	Device saturation (DST) is a fraction of the maximum device saturation (DSTM) determined by the existing level of device saturation modified by changes in the marginal costs of using each fuel. Changes in capital stock, process energy, and device energy requirements are determined from changes in production capacity, changes in device saturation, and changes due to physical lifetimes. Capital stock is retired (EUPCRPC) depending on the lifetime (PCPL) specified. If the lifetime is 30 years, then 1/30 of the capital stock (EUPC) is retired each year. As capital stock is added or retired, those changes also impact process energy and device energy requirements.
6. Retrofits and Conversions	
Objective: The conversion and retrofit code is optional and chosen by the user using a switch. For conversions, at the end of a device's useful lifetime, by default in the model, consumers choose a new device having the same technology/fuel type as the one retired. Using the conversion code sets consumer choice equation parameters for conversions to allow the use to switch to alternative technology/fuel types. If the retrofit switch is turned on, values are assigned to retrofit efficiencies and capital costs, and devices are retired early, such as would be true with a rebate program. Devices and/or processes can be assigned retrofits endogenously or specified exogenously. **Related Model Code**: *Function Conversion; Function DeviceDynamics; Function DRetrofit; Function RCPCDynamics; Function IniRetrofits; Function Retrofit*	
Key Inputs	Fuel prices (ECFP) Marginal cost of fuel use (MCFU) Variance factor on perceived prices for conversions (CVF) Non-price factors (CMSM0) Input retrofit device efficiency standard (RDEStd) User specified retrofit device capital cost multiplier (RDCCM) Average device efficiency (DEEA)
Key Outputs	Market share fraction for conversions (CMSF) Device retrofit market share fraction (RDMSF) Process Retrofit Market Share Fraction by Device (RPMSF)
Key Equations	Conversion market share (CMSF) is determined by price and a variance factor on perceived prices (CVF). Other adjustments to market share come from non-price factors (CMSM0). Retrofit market shares (RDMSF, RPMSF) are determined based on a marginal value of retrofitting determine from the relationship between stock average efficiency and the retrofit marginal efficiency when including the retrofitting cost factors.
7. Enduse and Feedstock Demand	

<table>
<tr><th>Demand Module Code</th><th>Key Inputs, Outputs, and Equations</th></tr>
<tr><td colspan="2">Objective: Calculates enduse and feedstock energy demands using consumer choice equations and inputs of price factors and non-price factors from the historical calibration. The temperature sensitive portion of end use demand is adjusted for weather. End use demands are further adjust by the energy impacts from pollution when emissions-reductions policies are in place. Feedstock demand is a simple calculation based on production capacity and feedstock efficiency levels calculated from the historical data. After end use energy demand is calculated, it is adjusted by assumptions of reductions from energy efficiency (EE).
Related Model Code: <i>Function DmdEnduse; Function EECalculation</i></td></tr>
<tr><td>Key Inputs</td><td><u>Inputs from calibration</u>
• Capacity utilization factor (CUF)
• Capital energy requirement multiplier/lifestyle multiplier (CERSM)
• Feedstock energy efficiency (FsPEE)
<u>Exogenous inputs</u>
• Degree day multiplier (DDM)
• Temperature sensitive load (value of 1 or 0) (TsLoad)
• Fraction of load with energy efficiency (EESat)
• Reduction from energy efficiency, Btu/Btu (EEImpact)
<u>Inputs from other demand routines</u>
• Device energy requirements (DER)
• Production capacity (SPC)
• Energy impacts from pollution reductions (RPEI)</td></tr>
<tr><td>Key Outputs</td><td>• Enduse energy demand (Dmd)
• Feedstock energy demand (FsDmd)
• Endogenous energy efficiency (EE)</td></tr>
<tr><td>Key Equations</td><td>• Dmd = DER*CUF*CERSM*(DDM*TSLoad+(1-TSLoad))*RPEI
• FsDmd=SPC/FsPEE
• EE=Dmd*EEImpact*EESat
• Dmd = Dmd-EE</td></tr>
<tr><td colspan="2">8. SELF-GENERATION DEMAND</td></tr>
<tr><td colspan="2">Objective: Calculates the amount of self-generation energy demand within the residential, commercial, and industrial demand sectors. Self-generation energy demands are calculated within the demand sector module, and are able to be represented also at an individual generating unit level within the electric supply module. Currently, U.S. self-generation is represented at the aggregate level in the demand sector, and Canada self-generation is represented at the unit level within the electric supply sector.
Related Model Code: <i>Function Self-generationSector, Function Self-generationTotals</i></td></tr>
</table>

<table>
<tr><th>Demand Module Code</th><th>Key Inputs, Outputs, and Equations</th></tr>
<tr><td>Key Inputs</td><td>Inputs from calibration routines:
• Self-generation capacity utilization factor (CgCUFP)
• Self-generation generating capacity (CgGCSector)
Other inputs:
• Historical self-generation demand (XCgDmdSector)
• Self-generation fuel-tech fraction (CgFrac)</td></tr>
<tr><td>Key Outputs</td><td>Sector-level outputs:
• Self-generation generation (CgEGSector, CgGenSector)
• Self-generation capacity (CgCapSector)
• Self-generation demand (CgDemandSector, CgDmdSector)
Total self-generation (sector-level plus generating unit self-generation):
• Total self-generation demand (CgDmd, CgDemand)
• Total self-generation generation (CgEG)
• Total self-generation capacity (CgGC)
• Total self-generation by economic category (CgEC)</td></tr>
<tr><td>Key Equations</td><td>Sector level self-generation
• CgEGSector=CgGCSector*CgUMS*CgCUFP*WCUF*8760/1E3
• CgDmdSector=CgEGSector*CgHRtA/1E6
Total self-generation
• CgDemand=CgDemandSector + CgDemandUnit
• CgDmd =CgDmdSector +CgDmdUnit
• CgEG =CgEGSector + CgEGUnit
• CgGC =CgGCSector + CgGCUnit</td></tr>
<tr><td colspan="2">9. TOTAL ENERGY DEMAND</td></tr>
<tr><td colspan="2">Objective: Creates summary level outputs from the demand and price results from earlier calculations. Total energy demand is the sum of end use demand, feedstock demand and self-generation demand. The self-generation demand is the sum of the sector-level self-generation demand calculated in the demand module plus the unit-level self-generation demand calculated in the electric supply module.
Related Model Code: <i>Function EECalculation; Function TotalDemand</i></td></tr>
<tr><td>Key Inputs</td><td>Inputs from other demand module routines:
• End use energy demand (Dmd)
• Total self-generation energy demand (CgDmd)
• Feedstock energy demand (FsDmd)
• Total self-generation by economic category (CgEC)</td></tr>
</table>

Demand Module Code	Key Inputs, Outputs, and Equations
Key Outputs	• Total energy demand ($\text{TotDemand}_{\text{Fuel,ECC,Area,Year}}$) • Energy demands by sector ($\text{DmdES}_{\text{Sector,Fuel,Area,Year}}$) • Endogenous energy efficiency (EE)
Key Equations	• TotDemand=EuDemand+CgDemand+FSDemand With an adjustment for electricity demands: TotDemand(Electric)=TotDemand(Electric)-Self-generation • DmdES=sum(ECC)(TotDemand)
10. EMISSIONS	
Objective: Calculates emissions from enduse fuel demands, feedstock fuel demands and self-generation fuel demands. The emissions calculated in this module are solely those from the demand sector module. Emissions from supplying energy, such as electric generation, are calculated in the Supply Module. Emissions from non-fuel based emissions (MEPol) are calculated in the economic module. **Related Model Code:** *Function PollCoefficients; Function PollutionGenerated*	
Key Inputs	• Emissions coverages, 1=covered (ECoverage, PCCov) • Demand (EuDem, FsDmd, CgDmd), • Marginal emissions coefficient (POCX, CgPOCX), • Policy reductions (RM) • Device additions (DERA)
Key Outputs	• Average pollution coefficients (POCA, CgPOCA, FsPOCA) • Pollution totals ○ Enduse (Polute, EnFPol) ○ Self-generation (CgPolSector, CgFPol, CgPol) ○ Feedstocks/non-combustion (FSPol, NcFPol) ○ Total pollution (TotFPol) • Gross pollution before impact of pollution policies (GrossPol)
Key Equations	• Average coefficients (POCA, CgPOCA, FsPOCA) based on marginal coefficient (POCX) and energy additions (DERA) depending on model switches set • Calculate emissions inventories based on average coefficient (POCA) and total energy usage (Dmd) ○ Polute=EuDem*POCA; EnFPol=sum(Enduse,EC)(Polute) ○ FsPol=FsDem*FsPOCA; NcFPol=sum(EC,Fuel)(FsPol) ○ CgPolSectorEC=CgDemSector*CgPOCA ○ CgFPol=CgFPolSector+CgFPolUnit; CgPol=sum(FuelEP)(CgFPol) ○ TotFPol=EnFPol + CgFPol + NcFPol • GrossPol=Polute/RM*PCCov + FsPol*ECoverage(NonCombustion)

Demand Module Code	Key Inputs, Outputs, and Equations
11. INVESTMENTS	
Objective: Calculates device investments (equipment) and process (buildings) investments from annual additions to devices and production capacity made within the demand sector. These device and process investments are often sent as input to a macroeconomic model to determine the economic impacts of a particular policy. **Related Model Code:** *Function Investments*	
Key Inputs	A-1. Device and process capital costs (DCC, PCC) A-2. Device Capital Costs (DCCFC) A-3. Device energy additions (DERA) A-4. Retrofit device and process capital cost (RDCC, RPCC) A-5. Retrofit device and process market share (RDMSF, RPMSF) A-6. Production capacity additions (EUPCAPC)
Key Outputs	• Device investments (DInv, DeviceInvestments) • Process investments (PInv)
Key Equations	• Device investments are marginal costs multiplied by device energy additions, including any retrofitting. • Process investments are marginal costs multiplied by production capacity additions, including any retrofitting.

6.7. Summary of Load Curves Functions

After the demand code has been executed electricity and natural gas demands are translated into load curves for input to the electricity and natural gas supply modules. The key inputs to these functions are energy demands and load shape factors, and the key outputs are load curves. A summary of the functions making up the load curve creation is listed below. See the *Appendix 5. Load Curve Functions* and Equations section for a detailed look at each of the functions.

1. **Function ETOU:** Electric Time-Of-Use Impacts. This function calculates the impacts of different time of use pricing on peak, average and minimum loads.

2. **Function LoadMgmt:** Impacts from Load Management Programs - calculates the effects on sales and peak demand from load management programs.

3. **Function LoadCurve:** Generating Electric Load Curve - derives economic class (ECLDC) and revenue class (CLDC) load shapes from demand (Dmd). Electric sales (ESales, ECSales, and Sales) are calculated from the load shapes. Self-generation is accounted for and netted out of sales.

4. **Function NoLoadCurve:** Electric Sales when load shapes are not available Function derives electric sales (ESALES, ECSALES, and SALES) from demand (DMD). Accounts for self-generation demand.

5. **Function DailyUse:** Gas Utility Daily Use Curve - calculates EUDUC, CDUC, (daily use curves by end-use and class, respectively), GSALES and SALES (sales by end-use and total sales) for the natural gas utility.

6. **Function NoDailyUse:** Gas Sales when load shapes are not available. Calculates GSales (gas sales by end-use and class) and Sales (total sales) when no load shapes are available and CCSales(transportation gas)

6.8. Summary of Economic Processing Module

The economic processing module assigns economic inputs to model variables and converts economic drivers into production capacity. Production capacity by vintage (old, middle, and new) is needed for input to the demand module. Production capacity drives the process energy requirements which, in turn, drive the device energy requirements. Tracking production capacity by vintage (old, middle, and new) allows the model to retire old capital stock at the end of its useful lifetime and replace it with new capital stock typically at higher efficiency levels.

The key inputs to the economic processing module consist of the economic driver (Driver) and other economic indicators obtained from an exogenous economic forecast.

The key outputs consist of production capacity (PC), production capacity by vintage - old, middle, and new – (PCLV), production capacity retirements (PCR), production capacity additions (PCA), and a capacity utilization factor (ECUF). The capacity utilization factor is based on the ration of the economic driver to production capacity (ECUF=Driver/PC).

Appendix

Appendix 1 - Theoretical Derivation – Consumer Choice Theory

A. ENERGY 2100: Consumer Choice Theory

This section provides a basis for using the multinomial logit in the consumer choice components of the ENERGY 2100 model. This discussion will primarily rest on extensive quotations and the research of others. Rather than reinvent the wheel, the developers of ENERGY 2100 took advantage of the well-supported research related to consumer choice as it applies to energy simulation.

Consumer Choice Simulation

The socioeconomic environment, of which energy is a component, is the consequence of people making choices. They choose to build a house, store, or factory. They decide to emphasize capital, operating, or energy efficiency in the process of providing goods and services. They choose the fuel used to heat their homes; they choose the efficiency of the furnace and other energy using equipment; and they decide how to operate their furnaces and equipment. The basic characteristic of consumers is that they make choices: choices to acquire, specify, and use. Therefore, a proper representation of energy use must be a proper representation of how choices are made and the energy impact of those choices.

Typically, a choice can be portrayed as a selection among a spectrum of alternatives. Faced with the selection options, a particular or discrete choice is made based on the preference of the consumer. The mathematical characterization of this choice process is called discrete choice analysis. The preferences are a function of observable quantities such as price and unobservable quantities such as style or taste. Additionally, consumer uncertainty in both the observable and unobservable portions of the individual's preference function means that the mathematical formulation of the choice process must be based on an estimation process, as are those estimations performed for more common econometric representations.

Consumer Utility and the Multinomial Logit

The utility, **U**, of a preference can be defined as $\mathbf{U_{in}=V_{in}+\varepsilon_{in}}$ where **V** is a dependent term and $\boldsymbol{\varepsilon}$ is an error term. **V** may depend on any number of characteristics **x** of a choice **i** and has an arbitrary functional form. "The generality and limits of this form deserve emphasis. A variable [utility] may be a component of **x**, a function specifying a nonlinear transformation, or interacting components of **x**, or a function specifying an interaction between **x** [choice attributes] and **s** [decision-maker attributes] variables."[1]

Using these preferences, for a given situation **n**, the probability $\mathbf{P_n}$ of a consumer making a particular choice **i** can be determined with the use of a multinomial logit (MNL).

"The multinomial logit is expressed as:

$$P_n(i) = \frac{e^{V_{in}}}{\sum_{j=1}^{N} e^{V_{jn}}} \tag{1}$$

This model (or equivalent variants of it) can be derived in a great number of ways. Its original formulation is due to Luce (1959), a mathematical psychologist. He derived the form of the [above] equation by making assumptions about the choice probabilities rather than the disturbances."[2]

The EPRI REEPS model uses the MNL formulation. The EPRI report starts the discussion as follows:

"Our choice of functional forms for the choice probabilities has been guided by several considerations. First, the functions must be computationally tractable, so that calibration and simulation on relatively large populations is possible. Second, the forms must be sufficiently flexible to adapt to the patterns of substitution and complementarity found in the data, without restrictive a priori assumptions. Third, households are assumed to be motivated to minimize the lifecycle cost of achieving specified levels of service, and more generally to weigh the desirability of energy-consuming services against other commodities in allocating their incomes. The functional forms for the choice probabilities should be consistent with such

---

[1] McFadden, D., "Conditional Logit Analysis of Qualitative Choice Behavior," in *Frontiers in Econometrics,* Ed. P. Zarembka, New York, Academic Press, 1974, page 114.

[2] Ben-Akiva, M., *Discrete Choice Analysis: Theory and Applications*, MIT Press, Cambridge, MA, 1985, page 103.

behavior. A family of functional forms for choice probabilities which meet these criteria and are therefore selected for our analysis are termed *nested logit* models... A nested logit model is a generalization of [the] form called the multinomial logit."[3]

Multinomial Logit Characteristics

If the utility $\mathbf{V_i}$ of a choice i greatly exceeds that of any other option, then the probability that the choice **i**, as shown in equation (1) will be actually chosen approaches unity. If all the choices, as perceived by the consumer, have the same utility, then all the choices will have an equal probability of occurrence or **1/N** where **N** is the total number of choice options available. That is, if the utilities are the same, then consumers cannot tell the difference between the options and they are just as likely to pick any option. Relative to a large population, this implies that equal proportions of each option will be selected. If the choice were only between two options, then the probability would be 50/50 or ½ that either of the options would be chosen. This phenomenon is a natural and reasonable consequence of both equation (1) and consumer choice theory. This does not mean a choice is not being made; it simply means that a consumer has no basis for a particular choice if all the options have equal utility. Note that in reality there is only an infinitesimal chance that all the options have the same utility.

Equation (1) has the feature that it allows the full range of utilities. If the utility function is a function of price, the price can range between 0 and infinity. This use of infinitesimally small or large values is not a problem from an empirical perspective. "Since empirically, a zero probability is indistinguishable from one that is extremely small, there is little loss of generality in assuming that the selection probabilities are all possible for the positive alternative sets in the experiment."[4] Moreover, more conventional econometric methods using elasticities have the identical theoretical considerations in that infinitesimally small prices would lead to infinite

---

[3] Cambridge Systematics, Inc., *Residential End-Use Energy Planning Model System (REEPS)*, Electric Power Research Institute, Report EA-2512, Palo Alto, California, July 1982, pages 3-9.

[4] McFadden, D., "Conditional Logit Analysis of Qualitative Choice Behavior," in *Frontiers in Econometrics,* Ed. P. Zarembka, New York, Academic Press, 1974, page 109.

demands and infinitely large prices would lead to infinitesimally small demands and imply infinite energy efficiency.

Independence From Irrelevant Alternatives

"Three properties of logit probabilities [have been discussed], namely that they (1) range from zero to one, (2) sum to one over alternatives, and (3) are a sigmoid or S-shaped [cumulative distribution shaped] function of representative utility. Each of these properties is quite reasonable, and in fact, the first two are logically necessary. Logit probabilities also exhibit a property, however, that, at least in some contexts, is not desirable. This is called the independence from irrelevant alternatives property or the IIA property for short.

"The IIA property has been the focus of considerable discussion in the literature and not a small amount of confusion."[5]

"Generally, the attributes entering the [MNL] for a specific alternative **j** depend solely on features of this specific alternative, and not on features of other alternatives. In this case the multinomial model is said to have the property of independence from irrelevant alternatives (IIA).

The term 'independence from irrelevant alternatives' refers to the property that the relative odds of two alternatives are independent of the availability and attributes of other alternatives.

However, it is possible for $\mathbf{V_j}$ to depend on iterations between features of alternative **j** and other alternatives, in which case the MNL model does *not* have the IIA property."[6]

"Despite its practical advantages, the IIA property is a restriction that is not realistic in many situations. Recent work has indicated, however, that the IIA property in logit models is not as restrictive as it might at first seem...

McFadden shows that any model that specifies choice probabilities, including models that do not exhibit IIA, can be expressed in the *form* of a logit model [emphasis from original text]. That

---

[5] Train, K., *Qualitative Choice Analysis,* MIT Press, Cambridge, MA, 1986, page 18.

[6] Cambridge Systematics, Inc., *Residential End-Use Energy Planning Model System (REEPS)*, Electric Power Research Institute, Report EA-2512, Palo Alto, California, July 1982, pages 3-10.

is, it is possible to express any choice probability [in the MNL form.] [The proof follows in the K. Train text. See footnote seven].

This shows that the logit probabilities, with the appropriate specification [of parameters] equal the true probabilities. Stated another way, any choice model can, with an appropriate choice [of estimated linear parameters], be put into the logit form. This concept gives rise to the term 'mother logit'"[7]

"What this discussion implies is that the logit specification can be used in situations for which IIA does not hold. All that is required is that additional variables be added to the representative utility, in particular, variables that relate to alternatives other than the one for which the representative utility is designated."[8]

Train also says that these extra variables are constant terms simply added to the utility function for each choice alternative prior to estimation of **V**. The modeler "estimates the model with all ... alternatives in the choice set and includes a constant term in the specification of the representative utility of the ... alternatives ..."[9]

McFadden performed other tests to show that:

"In particular, if the desirability of different alternatives tends to be fairly sharply differentiated for most households, which is the case unless the weights in [the MNL equation] are small in magnitude, the market cross elasticities are primarily determined by the distribution of households and are virtually independent of whether the household choice probabilities have the IIA property or not. Furthermore, the MNL functional form is rather robust empirically in that it will often describe observed choice behavior adequately even when the forces underlying that behavior are theoretically inconsistent with the IIA property."[10]

In an earlier work, McFadden explains the concept further:

---

[7] Train, K., *Qualitative Choice Analysis,* MIT Press, Cambridge, MA, 1986, page 21.

[8] Train, K., *Qualitative Choice Analysis,* MIT Press, Cambridge, MA, 1986, page 22.

[9] Train, K., *Qualitative Choice Analysis,* MIT Press, Cambridge, MA, 1986, page 23.

[10] Cambridge Systematics, Inc., *Residential End-Use Energy Planning Model System (REEPS)*, Electric Power Research Institute, Report EA-2512, Palo Alto, California, July 1982, pages 3-11.

“Nevertheless, empirical evidence is that the MNL model is relatively robust, as measured by goodness of fit or prediction accuracy, in many cases in which the IIA [independence of irrelevant alternatives] property is theoretically implausible.

The restrictive IIA feature of the MNL model is present only when the vector $\mathbf{x}_{\mathbf{it}}$ for alternative **i** is independent of the attributes of alternatives other than **i**. When this restriction is dropped, the MNL form is sufficiently flexible to approximate any continuous positive choice probability model on a compact [limited and defined] set of explanatory variables. Specifically if **P(i|xt)** is continuous, then it can be approximated globally to any desired degree of accuracy by the [standard] MNL model ...”[11]

Distributional Basis

“If we assume that the $\mathbf{U_{in}=V_{in}+\varepsilon_{in}}$ for all **i** ... and that all the disturbances in (1) are independently distributed, (2) identically distributed, and (3) Gumbel-distributed with a location parameter **ñ** and a scale parameter $\boldsymbol{\mu} \mathbf{> 0}$, then

$$P_n(i) = \frac{e^{\mu V_{in}}}{\sum_{j=1}^{N} e^{\mu V_{jn}}} \tag{2}$$

Say $\boldsymbol{\varepsilon}$ is Gumbel-distributed. Then [the cumulative form is:]

$$F(\varepsilon) = \exp(-\exp^{-\mu*(-\tilde{n})}) \ldots \tag{3}$$

As in the case of binary logit, the assumption of a constant **ñ** for all alternatives, or **ñ=0**, is not in any sense restrictive as long as each systematic utility has a constant term. Similarly, the assumption that the disturbances are Gumbel-distributed can be defended as an approximation to the normal density. It is also used only for reasons of analytical convenience.”[12]

---

[11] McFadden, D., “Qualitative Response Models,” in *Advances in Econometrics,* Ed.. Werner Hildenbrand, Cambridge University Press, New York, 1982, p.10.

[12] Ben-Akiva, M., *Discrete Choice Analysis: Theory and Applications*, MIT Press, Cambridge, MA, 1985, p.104.

If $\tilde{\mathbf{n}}=\mathbf{0}$, then the distribution is called the Weibul distribution. The Weibul distribution is more commonly cited than the Gumbel because it is the form actually used in practice.

“Thus, the probability distribution function on the generic technology price can be derived from distributions on the specific technology costs. It can be shown that the distribution of the least-cost from a sample of independently-distributed costs approaches the Weibul distribution.”[13]

“This model, or a derivative, has been used in a variety of energy modeling applications.”[14]

McFadden chooses the Weibul distribution a priori:

“Suppose each member of a population of utility-maximizing consumers has a utility function ... [whose error terms are] distributed with the Weibul (Gnedenko, extreme value) distribution.”[15]

Despite the support for the multinomial logit, alternative distributions have been studied to achieve a more theoretical avoidance of IIA problems. First, McFadden’s experience:

“If [the error term] is assumed to be multivariate normal, the resulting discrete response model is termed the multinomial probit (MNP) model ... when correlation is permitted between alternatives, so that the [covariance of the error terms] is not diagonal, the MNP model does not have the IIA or related restrictive properties ... However, for [more than five choice options], the computational time required for [estimation] ... is excessive.”[16]

---

[13] Boyd, D.W., et.al., *Abbreviated R&D Program Portfolio Selection Workbook: Market Share Model Appendix,* Decision Focus Incorporated, Palo Alto, California, U.S. Department of Energy contract DE-AC05-7BET05474, 1979, p. 6.

[14] Boyd, D.W., et.al., *Abbreviated R&D Program Portfolio Selection Workbook: Market Share Model Appendix,* Decision Focus Incorporated, Palo Alto, California, U.S. Department of Energy contract DE-AC05-7BET05474, 1979, p. 11. See, for instance, Cazalet, E.G., *General Equilibrium Modeling: The Methodology of the SRI-Gulf Model,* Final Report prepared by Decision Focus, Inc., for the Federal Energy Administration, Stanford Research Institute, Menlo Park, California, May 1977. See also A. Masevice, *A Review and Assessment of the Fossil1 Supply Structures.* Thayer School of Engineering, Dartmouth College, Master of Science Thesis [Advisor - George Backus], September, 1978.

[15] McFadden, D., “Conditional Logit Analysis of Qualitative Choice Behavior,” in *Frontiers in Econometrics,* Ed. P. Zarembka, New York, Academic Press, 1974, page 111.

[16] McFadden, D., “Qualitative Response Models,” in *Advances in Econometrics,* Ed.. Werner Hildenbrand, Cambridge University Press, New York, 1982, p.18.

Next, Ben Akiva reviewed the topic:

"Recent works [using the Probit] have resolved some of the computational problems. However, only a few, very limited applications have appeared in [the] literature, and there is still no evidence to suggest in which situations the greater generality of multinomial probit is worth the additional computational problems resulting from its use."[17]

Ben-Akiva spent some time on the problem as noted in his earlier work:

"The basic choice model that is used in this study for all alternative models is the multinomial logit model. Other choice models that might be considered to be superior from a theoretical point of view, such as the multiple probit model, are more complicated. It is not evident, however, that the added expense for more sophisticated choice models is worthwhile."[18]

Charles River Associates (CRA) also addressed the problem:

"Three commonly used models, the probit and logit, and a third known as the Cauchy probability model, give ogives ... and are virtually indistinguishable except at probabilities close to zero or one, where the probit model approaches the limiting values most rapidly, the Cauchy model the least rapidly. Within the range of most data, these models provide essentially equivalent probability functions and except for computational reasons, there is little to choose [statistically] among them. The logit model has computational advantages since it is a closed (explicit) functional form. The probit model, on the other hand, has an argument as the limit of an integral which cannot be expressed in closed form."[19]

---

[17] Ben-Akiva, M., *Discrete Choice Analysis: Theory and Applications*, MIT Press, Cambridge, MA, 1985, p.128.

[18] Ben-Akiva, M., *Structure of Passenger Travel Demand Models,* MIT, Department of Civil Engineering, Ph.D. Thesis, June, 1973, p. 171.

[19] Charles River Associates, *A Disaggregated Behavioral Model of Urban Travel Demand,* U.S. Department of Transportation, Contract No. FH-11-756, Final Report, March, 1972, pages 5-11.

Although CRA brings up the Cauchy distribution in this entry it is never brought up anywhere else in their discussions. One possible reason for the omission is McFadden's concern for positive finite moments which the Cauchy distribution does not have.[20]

Near the end of its review, CRA is down to only two approaches, the linear probability model and the conditional logit model (a form of MNL):

"The two models for multiple choice developed above, the multiple choice linear probability model and the conditional logit model, prove to be the most useful for demand analysis ... These models provide the advantage of practical empirical implementability along with a satisfactory theoretical justification in terms of the underlying behavior of individual decision makers.[21]

By the end of their review, however, CRA's last alternative to the multinomial logit is rejected.

"We conclude that the linear probability model as formulated ... does not yield a practical estimation function with satisfactory statistical properties."[22]

The references above claim to include all distributions that could be justified for use in choice analysis. Other distributions have characteristics which violate the necessary requirements of consumer choice theory or provide currently untenable mathematical difficulties. All competent research to date indicates that the multinomial logit, although it has limitations just like any other approach, provides the most acceptable means to simulate consumer choice.

B. Market Share Multinomial Logit in ENERGY 2100

Consumers, as simulated in ENERGY 2100, make choices relative to fuel selection for each energy end-use. These choices are simulated in ENERGY 2100 using the multinomial logit. The use of detailed multinomial logit formulations in energy demand has already been noted in the reference to the EPRI REEPS residential energy model developed by D. McFadden. It is also used

---

[20] McFadden, D., "Conditional Logit Analysis of Qualitative Choice Behavior," in *Frontiers in Econometrics,* Ed. P. Zarembka, New York, Academic Press, 1974, page 111, footnote 4.

[21] Charles River Associates, *A Disaggregated Behavioral Model of Urban Travel Demand,* U.S. Department of Transportation, Contract No. FH-11-756, Final Report, March, 1972, pages 5-28.

[22] Charles River Associates, *A Disaggregated Behavioral Model of Urban Travel Demand,* U.S. Department of Transportation, Contract No. FH-11-756, Final Report, March, 1972, pages 5-47.

in the EPRI COMMEND commercial model[23] by incorporating the multinomial logit work of Cohen and Baughman as the market share simulation.[24] The original Oak Ridge Residential Model developed by Eric Hirst also uses the multinomial logit for the market share calculation.[25]

Utility Function Form

The utility function is often written clearly, for example, as a simple function of price **(P_i)** with the constant (non-price) term noted above by Train. [26]

$$V_i = A_i + B^*P_i \tag{4}$$

in ENERGY 2100, the log-linear form is used:

$$V_i = a_i + b^*\ln(P_i) \tag{5}$$

An implication of this form is that the consumers are more sensitive to the proportional (percent) differences in costs than in absolute ($) differences. This means a one dollar difference is less important in a thousand-dollar furnace decision than it is in a three-dollar lightbulb decision.

There is substantial support for this formulation. The derivation of the market share function based purely on technological cost distributions leads directly to the form (with **a=0** - no non-price component) as shown in the work of Decision Focus, Inc. and the Institute for Economic Analysis.[27]

---

[23] Jackson, J.R., et. al., "Conservation Policy Analysis and End-Use Models: A Commercial Sector Example" in *Proceedings: End-Use Models and Conservation Analysis*, Electric Power Research Institute, Report EPRI EA 2509, Palo Alto, CA, July 1982, page 13.

[24] Cohn, S., *Fuel Choice and Aggregate Energy Demand in the Commercial Sector,* Oak Ridge National Laboratory, ORNL/CON-27, December, 1978 and Baughman, M.L. and Joskow, P.L., "Energy Consumption and Fuel Choice by Residential and Commercial Consumers in the United States" in *Energy Systems and Policy,* Volume 1, No. 4, 1974.

[25] Hirst, E., et. al., *An Improved Engineering Model of Residential Energy Use,* Oak Ridge National Laboratory, ORNL-CON-8, April 1977, page 18.

[26] Train, K., *Qualitative Choice Analysis,* MIT Press, Cambridge, MA, 1986.

[27] Work of Decision Focus: Boyd, D.W., et.al., *Abbreviated R&D Program Portfolio Selection Workbook: Market Share Model Appendix,* Decision Focus Incorporated, Palo Alto, California, U.S. Department of Energy contract DE-AC05-7BET05474, 1979; and for the Institute for Economic Analysis: Reister, D, et. al., "The Oak Ridge Industrial Model: An

With this formulation, equation (1) becomes

$$MS(i) = m_i \frac{P_i^b}{\sum_{j=1}^{N} m_j P_j^b} \tag{6}$$

where $\mathbf{MS_i}$ replaces the probability usage in equation (1) to avoid confusion with the use of "P" for prices. Also,

$$m_i = \exp(a_i) \tag{7}$$

This m term is called the market share multiplier in ENERGY 2100 but it is just the constant required to avoid IIA concerns. When m is defined to be 1.0, as is common in technology assessment analysis, the equation becomes:

$$MS_i = \frac{P_i^{-b}}{\sum_{j=1}^{N} P_j^{-b}} \tag{8}$$

This simpler form is the most commonly used form of market share calculation as noted in the SRI/Gulf Model, the GEMS model, the DRI Energy Model and LMSTM along with the ORIM model noted above.[28] It is also the method taught by EEI and EPRI for DSM analysis.[29]

Introduction," in *Proceedings: End-Use Models and Conservation Analysis*, Electric Power Research Institute, Report EPRI EA 2509, Palo Alto, CA, July 1982, pages 6-14.

[28] **SRI/Gulf Model:** Electric Power Research Institute, *Fuel and Energy Price Forecasts,* Volume 2, Report EPRI EA-433, Palo Alto, CA, 1977, p. 6-7:
GEMS Model: Cazalet, E.G., *General Equilibrium Modeling: The Methodology of the SRI-Gulf Model,* Final Report prepared by Decision Focus, Inc., for the Federal Energy Administration, Stanford Research Institute, Menlo Park, California, May 1977, p. 4-6;
DRI Energy Model: Data Resources, Inc., *DRI Energy Modeling System Documentation,* Data Resources, Inc., Cambridge, MA, 1984, p.11;
LMSTM: Decision Focus, Incorporated, *User's Guide to the Load Management Strategy Testing Model,* Electric Power Research Institute, EPRI EA-3653-CCM, August 1984, p. C-2;
ORIM Model: Reister, D, et. al., "The Oak Ridge Industrial Model: An Introduction," in *Proceedings: End-Use Models and Conservation Analysis*, Electric Power Research Institute, Report EPRI EA 2509, Palo Alto, CA, July 1982, p. 6-14.

[29] Battelle Columbus Laboratory and Synergetic Resource Corporation, *Demand-Side Management,* Edison Electric Institute and Electric Power Research Institute, EPRI EA/EM-3597, Volume 2, December 1984, p. 32.

Nonetheless, note that the price (P) can be any complicated function (including the real price) necessary to specify the perceived value of the commodity or service.

Basic research in choice analysis also tends to favor the log-linear approach:

“The formulation employed by the McLynn and Woronk model (1969) is equivalent to (the MNL) equation if all the variables $\mathbf{X_{itk}}$ are replaced by their logs ... This formulation can be written as:” [30]

$$P(i,A)=\frac{X_{itk}^{\theta k}}{\sum_{j=1}^{N} X_{jtk}^{\theta K}} \tag{9}$$

“Specification of explicit probability functions for the ‘strict utility’ specification in the [MNL] equation can be completed by specifying parametric forms for the function **V(x,s)**. We shall consider several cases. First suppose this function is log-linear in unknown parameters ...”[31]

Parameter Specification

The parameters (b) associated with choice variables are generally the same for all choice options, consistent with the derivation of the MNL-form above. The usage stems from the concept that all choices have equal uncertainty relative to the consumer.[32] The b parameters can be allowed to vary by alternative, however, provided the data truly supports the assertion that the choices are naturally indexed [unique onto themselves]. [33] When the microcomputer-based maximum-likelihood function described below is fully functional, nonconventional analysis assuming varying b parameters can be performed.

---

[30] Ben-Akiva, M., *Structure of Passenger Travel Demand Models,* MIT, Department of Civil Engineering, Ph.D. Thesis, June, 1973, p. 177.

[31] Charles River Associates, *A Disaggregated Behavioral Model of Urban Travel Demand,* U.S. Department of Transportation, Contract No. FH-11-756, Final Report, March, 1972, pages 5-26.

[32] Train, K., *Qualitative Choice Analysis,* MIT Press, Cambridge, MA, 1986, pp. 37-40; Ben-Akiva, M., *Discrete Choice Analysis: Theory and Applications*, MIT Press, Cambridge, MA, 1985, p.111; and McFadden, D., “Qualitative Response Models,” in *Advances in Econometrics,* Ed.. Werner Hildenbrand, Cambridge University Press, New York, 1982, p.4.

[33] McFadden, D., “Qualitative Response Models,” in *Advances in Econometrics,* Ed. Werner Hildenbrand, Cambridge University Press, New York, 1982, p.5.

Thus, the multinomial logit (based on the Weibul distribution) used in ENERGY 2100 is the only form supported in the literature (other than a theoretical effort to advance the potential use of the probit model - based on the normal distribution).

C. Estimation of MNL Parameters

The estimation of MNL parameters is abundantly discussed in the literature. The functional form of the MNL equation causes the ordinary least square estimation process to be biased. Therefore, the method of maximum likelihood estimation is used. [34]

"In many multiple-choice applications using available data, regression methods are not applicable and the maximum-likelihood method is the only practical function available."[35]

ENERGY 2100 Estimation

In ENERGY 2100, a non-price and a price related parameter are estimated for each fuel by end-use and economic category. These parameters were originally estimated in the DEMAND81 model using national data and non-linear least-squares. At the time, maximum-likelihood estimation packages were not commercially available. However, as McFadden notes "an alternative to maximum-likelihood estimation is to use non-linear least squares ...."[36] Nonlinear least-square estimation is a computer intensive operation. Therefore, re-estimation of the price response portion of the function was not routinely performed. It was assumed that the price response behavior would not be locally variable. Local tastes and socioeconomic environment (the non-price) were however assumed to be local. The non-price parameter was then re-estimated by ordinary least-squares for each implementation of ENERGY 2100. Studies show that "least-squares estimation leads to substantial overestimates of the price sensitivity..."[37]

---

[34] Fomby, T., et.al, *Advanced Econometric Methods*, Springer Verlag, New York, 1984, Section 16.4.

[35] Charles River Associates, *A Disaggregated Behavioral Model of Urban Travel Demand,* U.S. Department of Transportation, Contract No. FH-11-756, Final Report, March, 1972, pages 5-49.

[36] McFadden, D., "Qualitative Response Models," in *Advances in Econometrics,* Ed. Werner Hildenbrand, Cambridge University Press, New York, 1982, p.7.

[37] Dubin, J., and McFadden, D., "An Econometric Analysis of Residential Electric Appliance Holdings and Consumption," in *Proceedings: End-Use Models and Conservation Analysis,* EPRI Report EPRI EA 2509, Palo Alto, CA, July 1982, pages 13-20.

Therefore, this process should overestimate the conservation associated with market shifts and thus be less controversial from a regulatory perspective.

Nonetheless, recent computer hardware advances now allow maximum-likelihood estimation to be performed routinely on microcomputers, the platform for ENERGY 2100. Further, as ENERGY 2100 is used for analyses where there are limited historical data, maximum-likelihood estimation becomes more important because "limited Monte Carlo studies and analytical solutions suggest the maximum-likelihood estimators are also satisfactory in small samples."[38] Limited testing of the components of a maximum-likelihood estimation routine for the ENERGY 2100 calibration has been completed. This routine will provide the statistical reporting unique to MNL estimation. The ENERGY 2100 maximum-likelihood routine is based on the work of S. Cosslett which focuses on the use of aggregate data for efficient MNL estimation.[39] This is the type of data most readily available to energy modelers.

Data Sources

Historical data, applicable to the service area, for the estimation of the MNL are obtained using published data from several sources corrected (scaled) to be self-consistent. The energy use data by economic sector (residential, commercial, industrial) at the state level are available from the U.S. Department of Energy.[40] These data are scaled to the service area based on historical utility sales by economic sector. Industrial energy use is further disaggregated into SIC (Standard Industrial Category) designations by utility billing data or the Annual Survey of Manufacturers.[41] The Survey of Manufacturers also provides the SIC-specific proportions of fuel use (coal, oil, gas, electricity, self-generation) for each historical year. End-use information

---

[38] Charles River Associates, *A Disaggregated Behavioral Model of Urban Travel Demand,* U.S. Department of Transportation, Contract No. FH-11-756, Final Report, March, 1972, pages 5-41.

[39] Cosslett, S.R., "Efficient Estimation of Discrete Choice Models", in *Structural Analysis of Discrete Data with Econometric Applications,* ed. C. Manski and D. McFadden, MIT Press, Cambridge, MA, 1986, Chapter 2.

[40] U.S. Department of Energy, *State Energy Data Report,* Energy Information Administration, DOE/EIA-0214, 1978 and later.

[41] U.S. Department of Commerce, *Annual Survey of Manufacturers,* Washington, DC, 1987 and later.

is often available from utility or other institutional surveys.[42] The most appropriate data available are used.

The utility sales are assumed to be the only values which are correct in an absolute sense. All other data are only presumed correct in a relative sense. That is, the data can be used for scaling (proportions) when the errors associated with that data can be assumed to cancel-*out in the equation*. (It is generally assumed that the information in any survey data set has the same proportional error for each fuel, industry, or end-use - all portions of the data are "equally" in error, e.g., 20% overestimated or 50% underestimated.) "Proportional data" is interpolated for missing data.

By using data for historical demands, appliance efficiency and appliance life, additions and retirements to the appliance stock by fuel and end-use can be estimated to derive historical market shares.[43] These historical market shares are then used to estimate the MNL. Price information comes from the utility, state, or the U.S. Department of Energy.[44]

The focus here is to use the best data available categorized in the same manner as the utility uses the data for required regulatory matters. This same data would be used in effectively the same way whether the formal model used were an end-use, econometric, or MNL-based model.

---

[42] See, for example, American Gas Association, *Gas Facts,* Arlington, VA., 1975 and later; U.S. Department of Commerce, *Census of Housing,* Washington, DC, 1970, 1980; U.S. Department of Energy, *End Use Energy consumption Data Base: Series I Table,* Energy Information Administration, DOE/EIA-0014, June 1978: U.S. Department of Energy, *Residential Energy consumption Survey,* Energy Information Administration, DOE/EIA-0207/5, July 1980 and later; U.S. Department of Energy, *Nonresidential Buildings Energy Consumption Survey,* Energy Information Administration, DOE/EIA-1278, June 1981 and later; and Electric Power Research Institute, EPRI EM-5126 *Energy Use Patterns and Indicators,* Palo Alto, CA, April 1987.

[43] for appliance efficiency see: Association of Home Appliance Manufacturers, *Energy Efficiency and Consumption Trends,* Chicago, Illinois, July 1, 1984, and Geller, H., *Energy and Economic Savings from National Appliance Efficiency Standards,* American Council for an Energy-Efficient Economy, Washington, D.C., August 1986. For appliance efficiencies and appliance life see U.S. Department of Energy, *Annual Report to Congress,* Energy Information Administration, DOE/EIA-0173(198X)/3, 1981 and later.

[44] See U.S. Department of Energy, *State Energy Prices by Major Economic Sector,* Energy Information Administration, DOE/EIA-0190, 1981 and U.S. Department of Energy, *State Energy Price and Expenditure Report,* Energy Information Administration, DOE/EIA-0376(8X), 1984 and later.

D. Efficiency Trade-off as Binomial Logit

The decision to invest in higher capital cost (higher energy efficiency) equipment or structures in the face of higher energy prices is a consumer choice. It is a binomial logit choice in that it is the choice between two quantities, capital cost and operating costs (fuel). The result of the choice determines the efficiency of the new equipment or structure. The multinomial logit, equation (1), reduces to a much simpler form when only two choices are involved:

$$P_n(i) = 1/(1+e^{(V1-V2)}) \tag{10.}$$

or

$$P_n(i) = 1/(1+e^{(V1)}/e^{(V2)}) \tag{11}$$

If **V** is log-linear, as used in ENERGY 2100, the "form" becomes:

$$P_n(i) = 1/(1+(V1/V2)) \tag{12}$$

Functional Form Selection

A review of capital-efficiency trade-off literature shows only algebraic variations of the two forms above for determining capital cost versus efficiency. (The function presented in the documentation can always be algebraically transformed to correspond exactly to a binomial logit.) The logit has the necessary functional S-shape. The curve must be asymptotic and reach the maximum (finite) efficiency at infinite costs. The curves estimated here are empirical continuous curves reflecting consumer choice in light of actual technology alternatives.

Least-Cost Curves

Least-cost curves which can also be used in demand analysis, including ENERGY 2100 analyses, are discrete (discontinuous) engineering curves which order a selection of energy efficient technologies based on estimated (engineering-based) energy savings. Least-cost curves are not used to determine the choice consumers make; they are used to determine the impacts of energy programs if consumers chose energy efficiency technologies based on the economic decisions used by the analyst. The ordering of least-cost options on the least-cost curve is still an open issue, hotly debated. Examples of least-cost generation are available from a variety of sources.[45] These curves have the same general shape as the binomial logit and are well

---

[45] See, for example: Meier, A. *Supply Curves of Conserved Energy,* Lawrence Berkeley Laboratory, May 1988; Krause, F., *Analysis of Michigan's Demand-Side Electricity Resources in the Residential Sector,* Lawrence Berkeley Laboratory, LBL-23025, February, 1987; Ford, A. and Naill, R., *Conservation Policy in the Pacific Northwest,* Bonneville

approximated by the logit. The primary difference is that the "least-cost logit" is shifted toward the zero axis because it would have consumers investing in higher efficiency equipment at a much lower energy price. That is, it would infer that consumers place much more utility on reducing long-term energy costs than the historical data indicate.

Binomial Logit Basis

The binomial logit curves can also be reconciled as a composite of the market share of all available technologies chosen by consumers as energy prices vary. The resulting binomial logit can then be construed as a "fit" of the average efficiency selected by those "multinomial" choices.

Those that use the log linear form of the binomial logit are ENERGY 2100 and the Oak Ridge Residential Model.[46] The linear form is used in the REEPS and COMMEND model and several independent studies.[47]

Binomial Logit Sensitivity

Empirical tests using both forms under worst case conditions (at the center point where the probability is ½ and operating cost utility and capital cost utility are equal) show that a 25% change in capital cost (the independent variable) produces a 2% difference in the model results. A 50% change leads to a 10% difference. During model usage, these curves are only affecting new

---

Power Administration, May 1985; and Synergetic Resource Corporation, *Industrial Electricity Conservation Potential in the Pacific Northwest,* Volumes I and II, report No. 7077-R2, Bala Cywyd, Pennsylvania, March 1983.

[46] For ENERGY 2100: Backus, G., and J. Amlin, *ENERGY 2100 Integrated Policy Model Documentation* (three volumes), Policy Assessment Corporation, St. Paul, Minnesota, April 1987.
For Oak Ridge: Hirst, E., et. al., "The Oak Ridge National Laboratory's Residential Energy Use Model: Version 7.1" in *Proceedings: End-Use Models and Conservation Analysis,* Electric Power Research Institute, Report EPRI EA 2509, Palo Alto, CA, July, 1982.

[47] See, for example: Corum, K., et. al., "A Simulation Analysis of Alternative Policies to Simulate Energy conservation in Commercial Buildings," in *Proceedings: End-Use Models and Conservation Analysis,* Electric Power Research Institute, Report EPRI EA 2509, Palo Alto, CA, July 1982; O'Neal, D., and Corum, K., "Investment in Energy Efficient Houses: An Estimate of Discount Rates Implicit in New Home Construction Practices," in *Energy,* Volume 7, No. 4, Pergamon Press Ltd., 1982; Ruderman, H., et.al., "The Behavior of the Market for Energy Efficiency in Residential Appliances Including Heating and Cooling Equipment," in *The Energy Journal,* Volume 8, No. 1, 1987.

For REEPS and COMMEND see Cambridge Systematics, Inc., *Residential End-Use Energy Planning Model System (REEPS)*, Electric Power Research Institute, Report EA-2512, Palo Alto, California, July 1982.

investments, so their immediate impact on model results is reduced by an additional order of magnitude. Note also, that the recently announced 25% improvement in efficiency standards for refrigeration is expected to produce only a 10% increase in capital costs. The two forms, in this situation, would agree within 0.4%! Thus the sensitivity to the form used in ENERGY 2100 and the only used alternative is indistinguishable.

E. Estimation of Trade-off Curves

The trade-off curves are only estimated once when the raw historical data on historical efficiency, capital cost, and fuel prices are entered into the ENERGY 2100 databases. The binomial logit is a two-parameter curve. Therefore, the two (binomial choices) can be thought of as two equations (both a function of energy prices) with two unknowns. These equations are solved by simple point estimates.

Algebraic Solution

Two features can be determined about the choice equation under particular conditions (the year 1972 for ENERGY 2100 calculations.) These are the actual choices of capital cost and efficiency (the first known) and the slope of the curve when the choice was made (the second known). The functional form of the curve has been derived a priori. The solution for the parameters is then simply to find two conditions for which the unknown parameters can be solved. The capital cost and efficiency can be found in readily available historical data. The slope of the curve in an infinitesimal region at the decision-point can be calculated by "perturbing" the solution of the cost function around a point. This calculation provides the ε needed to solve the parameters of the globally-applicable binomial logit. This slope calculation at one point has no other purpose and is unrelated, functionally, to the binomial logit used for all capital cost and efficiency calculations. The ε calculation is just part of a mathematical process to solve the parameters of the binomial (trade-off) logit.

The trade-off curve is only estimated at the "1972 point" because that "point" was prior to any changes in energy prices. The data for that year closely approximates an equilibrium market. This provides an easy basis for data interpretation in that marginal versus average issues need not be addressed. (Alternatively, AHAM, AGA, ASHRAE, or other survey data could be used to perform a complete maximum-likelihood estimation of the trade-off curve, but any biases or incompleteness issues must be reconciled.)

Appendix 2 - Theoretical Derivation - Demand Trade-off Curve

This section derives the cost-versus-efficiency trade-off curves used in ENERGY 2100. This derivation was originally developed for the U.S. Department of Energy's DEMAND81 model.[48] This derivation also details how the demand coefficients in the model are estimated.

The demand trade-off curve derivation begins with a generalized cost function:

$$MCO = CCR*CC+OMC+P/N \tag{1}$$

Where

MCO = Marginal cost of output ($/Unit)
CCR = Capital charge rate (($/Yr)/$)
CC = Marginal capital cost ($/(Unit/Yr))
OMC = Marginal operating and maintenance costs ($/Unit)
P = Marginal price of energy ($/BTU)
N = Marginal efficiency (Unit/BTU)

For the general economy, output is measured in dollars of goods. For an energy conversion process (here converting primary fuel BTUs to useful process BTUs), output is measured in BTUs of useful (process) energy. For a transportation sector, output would be measured in equivalent vehicle-miles.

This functional form is consistent with the classical definition:

$$MCO_j = \sum a_i * (I/O)_i \tag{2}$$

Where

a_i *= cost per unit of input factor "I"*

$(I/O)_i$ *= units of input factor "I" to produce one unit of output "j"*

For the purposes here, only capital and energy are explicitly considered. The OMC term is an aggregate variable representing all other input factors such as labor and materials. Capital costs

---

[48] Backus, George A., *DEMAND81: National Energy Policy Model,* School of Industrial Engineering, Purdue University, Reports AFC-7 through AFC-10, 1981.

(CC) are assumed to be a function of technological advance and energy costs only. Operating and maintenance costs are assumed to be proportional to capital cost (and energy costs to the extent that capital costs are a function of energy costs.) As machines become more complicated, higher cost labor and maintenance are required. Empirical studies support this assumption.[49]

$$OMC = OCF * CC \qquad (3)$$

where OCF is the unit operation cost factor (\$/Yr)/\$

Process efficiency is assumed to be a function of technological advance, capital costs, and energy costs. At the margin, perceptions of the trade-off between cost and efficiency stipulate that:

$$dMCO/dN = 0 \qquad (4)$$

Where "*d*" is the ordinary differential operator. (This analysis could proceed using partial derivatives; the results would be the same and the additional mathematical arguments would only detract from the clarity of the derivation.)

Technological advance is exogenous but assumed to be changing over time. Therefore, at any instant, capital cost can be written as a function of process efficiency for small perturbations of N as:

$$CC^*/CC_B = (N^*/N_B)^{\varepsilon} \qquad (5)$$

Where

CC^*, N^* = perturbed values
CC_B, N_B = base values before perturbation
ε = elasticity and derivative of curve at "B"

For algebraic ease:

$$CC^o = CC^*/CC_B \qquad (6)$$
$$N^o = N^*/N_B \qquad (7)$$
$$P^o = P^*/P_B \qquad (8)$$

---

[49] See Backus, G., *FOSSIL79: National Energy Policy Model*, Resource Policy Center, Thayer School of Engineering, Dartmouth College, Report No. DSD-165 through DSD-168, 1979; and U.S. Department of Energy, *FOSSIL2 Energy Policy Model Documentation.* NTIS Document DOE/70143-02, Washington, D.C., October, 1980.

Using equations 3,5,7 and 8, equation (1) can be rewritten as:

$$MCO = (CCR+OCF)*CC_B * (N^o)^{\varepsilon} + P_B/N_B * P^o/N^o \tag{9}$$

Equation 9 can be used in equation 4:

$$dMCO/dNo = (CCR+OCF) * CC_B* \varepsilon * (N^o)^{\varepsilon-1} - P_B/N_B * P^o/(N^o)^2 = 0 \tag{10}$$

or in the base year when equations 6,7 and 8 equal unity:

$$\varepsilon = [P_B/N_B]/[(CCR+OCF) * CC_B] \tag{11}$$

This equation guarantees that the value added from energy or capital is equal at the margin as required by classical economics. Note that ε is always positive.

To increase the utility of equation 5, there needs to be a function "*f*" such that:

$$CC^*/CCN = f\,(N^*/N_{max}) \tag{12}$$

Where CCN is a normalizing capital cost varying only with technological advance and N_{max} is the maximum obtainable efficiency currently available at any cost.

Now the coordinate systems can be changed by multiplying equation 5 by CCN/CCN and N_{max}/N_{max}:

$$CC^*/CCN * CCN/CC_B = (N^*/N_{max} * (N_{max}/N_B)^{\varepsilon} \tag{13}$$

or

$$CR = (\beta*NR)^{\varepsilon}/\alpha \;\; |_B \tag{14}$$

where:

$$CR = CC^*/CCN \tag{15}$$

$$NR = N^*/N_{max} \tag{16}$$

$$\alpha = CCN/CC_B \tag{17}$$

$$\beta = N_{max}/N_B \tag{18}$$

Note that the slope of equation 14 in the base year (base values) is:

$$dCR/dNR = \varepsilon*\beta^{\varepsilon}*NR^{\varepsilon-1} / \alpha \;|_B \tag{19}$$

By definition it is assumed here that as:

$$NR \rightarrow 1 \text{ then } CR \rightarrow \infty \tag{20}$$

(i.e. as $N^* \rightarrow N_{max}$)

As implied by a production function with substitution:

$$\mathrm{CR} \rightarrow 0 \text{ as } \mathrm{NR} \rightarrow 0 \tag{21}$$

This expression assumes that there can be no output without energy, which more strictly assumes that if:

$$\mathrm{CR} > 0 \text{ then } \mathrm{NR} > 0 \tag{22}$$

It also assumes that capital is required for energy to be useful, i.e., if:

$$\mathrm{CR} = 0 \text{ then } \mathrm{NR} = 0 \tag{23}$$

The market share function satisfies all these requirements:

$$\mathrm{NR} = 1/(1+\mathrm{CR}^{\mu}) \tag{24}$$

Equation 24 is the market share function with only two choices - trading energy efficiency (fuel cost) for capital costs. Here, the market share is the share of the maximum efficiency. Note that NR equals 0.5 when CR equals 1.0 (i.e., CC equals CCN) and that μ is always negative. The appearance of the market share makes sense given that it reflects how choices are made with real-world, imperfect information/perceptions.

Equation 11 can be solved using historical data. For use in equation 1, equation 24 would be rearranged to yield:

$$\mathrm{CR} = (1/\mathrm{NR}-1)^{1/\mu} = \Phi^{h} \tag{25}$$

Note that $d\mathrm{CR}/d\mathrm{NR}$ must equal the value obtained from equation 19 in the base year. From the chain rule:

$$d\mathrm{CR}/d\mathrm{NR} = d\mathrm{CR}/d\Phi * d\Phi/d\mathrm{NR} \tag{26}$$

$$= -h*\Phi^{h-1} * \mathrm{NR}^{-2} \tag{27}$$

$$= -1/\mu * (1/\mathrm{NR}-1)^{1/\mu-1} * \mathrm{NR}^{-2} \tag{28}$$

From equations 28 and 19:

$$\varepsilon*\beta^{\varepsilon} * \mathrm{NR}^{\varepsilon-1} / \alpha = -1/\mu * (1/\mathrm{NR}-1)^{1/\mu-1} * \mathrm{NR}^{-2} \tag{29}$$

In the base year (from equations 17 and 18):

$$\varepsilon * NR^{-\varepsilon} * NR^{\varepsilon-1} * CC_B/CCN = -1/\mu * (1/NR-1)^{1/\mu-1} * NR^{-2} \tag{30}$$

or by noting that (1/NR-1) equals (1-NR)/NR:

$$-\varepsilon * \mu * CC_B/CCN = (1-NR)^{1/\mu-1} * NR^{-1/\mu} \tag{31}$$

In equation 31, ε, CC_B, and NR can be obtained directly form historical data and engineering estimates (i.e., N_{max}). CCN and μ are the only unknowns in equation 31. Equation 25 also defines CCN and μ. Equation 25 can be used to generate an equation with μ as the only unknown. Equation 31 becomes:

$$-\varepsilon * \mu * (1/NR-1)^{1/\mu} = (1-NR)^{1/\mu-1} * NR^{-1/\mu} \tag{32}$$

or

$$\mu = -1/[\varepsilon * (1-NR)] \tag{33}$$

With μ known, CCN can be found by equation 25:

$$CCN = CC_B/(1/NR-1)^{1/\mu} \tag{34}$$

Now equation 1 can be rewritten by using equation 25:

$$MCO = (CCR+OCF) * CCN * (1/NR-1)^{1/\mu} + PN/N_{max} * PR/NR \tag{35}$$

where PN is a normalizing energy price and:

$$PR = P^*/PN \tag{36}$$

On the margin, equation 4 must be valid at all points; therefore:

$$d MCO/d NR = \Omega * h * (1/NR-1)^{h-1} * NR^{-2} - \theta * PR/NR^2 = 0 \tag{37}$$

where:

$$\Omega = (CCR+OCF) * CCN \tag{38}$$

$$\theta = PN/N_{max} \tag{39}$$

$$h = 1/\mu \tag{40}$$

or

$$\theta * PR = -\Omega * h * (1/NR-1)^{h-1} \tag{41}$$

or

$$NR = 1/[1+(-\theta*PR/(\Omega*h))^{1/h-1}] \qquad (42)$$

Note that from equations 38, 39 and 40, $-\theta*PR/(\Omega*h)$ equals:

$$(-\mu/Nmax) / [(CCR+OCF)*CCN] * P^{*} \qquad (43)$$

Thus PN can be redefined as:

$$PN = (CCR+OCF)*CCN / (-\mu/N_{max}) \qquad (44)$$

and

$$\sigma = \mu/(1-\mu) \qquad (45)$$

Finally, equation 42 becomes:

$$NR = 1/(1+PR^{\sigma}) \qquad (46)$$

Note that the CR and PR equations are functionally consistent as they must be.

In ENERGY 2100, σ is the fuel trade-off coefficient XXFTC (where XX is the end-use or process prefix), μ is the capital trade-off coefficient XXCTC, PN is the fuel price - normal XXFPN and CCN is the capital cost normal XXCCN. N_{max} are the XXEMs in ENERGY 2100 for each end-use or process.

XXFTC, XXCTC, XXFPN, and XXCCN are solved with 1972 historical data. In 1972 the average (recorded) data also approximate the marginal decision data because energy prices had been constant since 1940. This is long enough for the vintaging effects of capital stocks to be minimal.

Appendix 3 - Theoretical Derivation - Capital Charge Rate

The capital charge rate is the annualization of capital expenses to account for taxes, tax credits, return of principal, return on investment, and interest during construction. The "CCR" equation is:

CCR = (1+R)**(C/3)*(1-ITC/(1+NR)-TR*(TL/2)/(TL/2+NR))
*R/(1-(1+R)**(-BL))/(1-TR)

Where:

R = Real Return on Investment
NR = Nominal Return on Investment
C = Construction Time
ITC= Investment Tax Credit
TR = Tax Rate (Federal plus State income tax)
TL = Tax Life
BL = Book Life

NR=(1-TR)*(1-F)*ND+F*NE
R=(1+NR)/(1+INF)-1
ND=(1+D)*(1+INF)-1
NE=(1+E)*(1+INF)-1

Where:

F = Fraction Equity
INF= Inflation Rate
ND = Nominal Return on Debt (Interest Rate)
D = Real Interest Rate
NE = Nominal Return on Equity
E = Real Return on Equity

For small "INF" (less than 10%/yr), a simpler calculation can be used with acceptable error:
ND=D+INF
NE=E+INF
R=(1-TR)*(1-F)*D+F*E
NR=R+INF

Risk can be added to "R" to reflect uncertainty and a higher required return. ENERGY 2100 includes financial risk concerns by increasing the required rate of return. Typically, a .02 to .05 risk (RISKN) is used for new technologies.[50]

Although the standard version of ENERGY 2100 uses a constant risk adjustment, a dynamic risk adjustment can be easily calculated. As a first approximation, a technology is assumed to be mature when the demand (D) for it is 10% of the total market demand (MPD). The risk can be reduced over time to reflect this phenomenon:

RISK=RISKN*EXP (-D/MPD)
RR=R+RISK

Where "RR" is the risk-adjusted "R" that can be used instead of "R" in all appropriate equations.

The "(1+R)**(C/3)" term in the "CCR" equation represents interest during construction which must be added to the final cost of the facility. During construction, costs accumulate faster near the end of the project than at the beginning. As a good approximation, it can be assumed that all the construction costs occurred two-thirds of the way through the construction program. That means interest charges ® accumulated for a time equaling "C/3".

The "R/(1-(1+R)**(-BL)" term is the classical capital recovery term.[51] The "(1-TR)" term at the end converts the after tax calculation into before tax dollars.

Investment tax credits reduce the cost of the plant by the tax credit after the first year of operation using "original" dollars. Therefore the value of the tax credit is "ITC/(1+NR)".

Depreciation is expensed for tax purposes during each year of the tax life of the plant. With the double-declining balance method (DDB) of computing depreciation, the depreciation (DEP) of the plant for each capital dollar spent in year "t" is:

DEP(t)=2/TL*(1-2/TL)**(t-1)

Depreciation, under existing laws, is a current dollar phenomena which does not account for inflation. Therefore the net present value of the energy is calculated with the nominal rate of

---

[50] Backus, G. A., *FOSSIL79 National Energy Policy Model*, Resource Policy Center, Thayer School of Engineering, Dartmouth College, Report No. DSD-165 through DSD-168, 1979.

[51] Smith, Gerald W., *Engineering Economy: Analysis of Capital Expenditures*, Iowa University Press, Ames, Iowa 1973.

return. If the depreciation life is adequately long to neglect end year effects, then the net present value of depreciation expenses is:

(2/TL)/(NR+2/TL)

Because depreciation is a benefit (negative cost) based on the total plant before investment tax credits, it shows up as an additional negative term in the capital cost modifiers of "CCR:"

(1-ITC/(1+NR)-TR*(TL/2)/(TL/2+NR))

The CCR calculation is naturally appropriate to business decisions but its use in the residential sector may appear artificial. When the CCR calculation is used for the residential sector, TL and C are set to zero because the residential sector can neither write off depreciation expenses nor make adjustments for extended construction times. This makes the calculation exactly correct for housing and any long-term investments.

Concerns can occur when the life of the loan is much shorter than the physical life assumed in the CCR calculation. When short-term loans (2-5 years) are used, the home owner still implicitly discounts the equity portion of equipment and depreciates the equipment over its expected life time. (Consumers do not expect a car or stove to fail as soon as the loan is paid-off; they write-off its value over its actual life time.) Therefore, the CCR calculation can only be incorrect for the debt portion of the investment. When a life cycle cost analysis of the actual cash flows is performed, which levelizes the short-term interest payments with the life of the equipment, the results are essentially identical to those obtained with the CCR calculation here.

Appendix 4. Demand Module Code and Equations

A. Price and Pollution Policy Inputs Functions

Two model code functions are used to create the price and pollution policy inputs required for use in the demand module – *Function TPrice* and *Function PRReductions* – and are described in detail below.

Function TPrice: Pollution Costs, Fuel Prices and Technology Mapping

This function calculates the effective price of fuel by combining the by fuel prices with the emission fuel costs after calculating the emission fuel costs. It then populates fuel price variables at differing levels of detail for use later in the demand module code.

Add Impact of Permit Costs

Marginal emissions are calculated as the coefficient of emitting sources after reductions are applied, weighted across Enduse by fuel demands in the prior year since this code is applied before current year demands are estimated.

```
PolMarginal[fuelep,ec,poll,area] = sum((POCX[enduse,fuelep,ec,poll,area]*(1-ZeroFr[fuelep,poll,area])*
RM[fuelep,ec,poll,area]*max(EuDemPrior[enduse,fuelep,ec,area],1e-12))
for enduse in Enduses)
```

where:

PolMarginal, ‘Marginal Emissions (Tonnes/Yr)’ [FuelEP,EC,Poll,Area]

POCX, ‘Marginal Pollution Coefficients (Tonnes/TBtu) [Enduse,FuelEP,EC,Poll,Area]’

ZeroFr, ‘Fraction of Emissions from Zero Emission Sources (Tonnes/Tonnes)’ [FuelEP,Poll,Area]

RM, Reduction Multiplier (Tonnes/Tonnes) [FuelEP,EC,Poll,Area]

EuDemPrior, Enduse Demands (TBtu/Yr) [Enduse,FuelEP,EC,Area]

Emission expenditures are produced using marginal emissions multiplied by the emissions costs in nominal dollars

```
ExpCP[fuelep,ec,area] = sum((PolMarginal[fuelep,ec,poll,area]*
(PCost[fuelep,ec,poll,area]+PCostExo[fuelep,ec,poll,area])*Inflation[area])/1e6
for poll in Polls)
```

where:

ExpCP, ‘Emission Expenditures ($M/Yr)’ [FuelEP,EC,Area]

PolMarginal, ‘Marginal Emissions (Tonnes/Yr)’ [FuelEP,EC,Poll,Area]

PCost, ‘Permit Cost ($/Tonne)’ [FuelEP,EC,Poll,Area]

PCostExo, ‘Marginal Exogenous Permit Cost (Real $/Tonnes)’ [FuelEP,EC,Poll,Area]

The carbon price by FuelEP is then the emission expenditure total weighted by demands in the prior year to apply values to utilized fuels

FEPCP[fuelep,ec,area] = ExpCP[fuelep,ec,area]/(sum(DmdFEPTechPrior[fuelep,tech,ec,area] for tech in Techs))

where:

FEPCP, ‘Carbon Price by FuelEP ($/mmBtu)’ [FuelEP,EC,Area]

ExpCP, ‘Emission Expenditures ($M/Yr)’ [FuelEP,EC,Area]

DmdFEPTechPrior, ‘Energy Demands (TBtu/Yr)’ [Enduse,Fuel,Tech,EC,Area]

Calculate Pollution Costs by Technology

The cost of emissions is calculated for each technology as the sum of the price by fuel by fuel type weighted into technology using prior year fuel demands and the reduction multiplier weighted by the type of fuel consumed in the previous year as follows.

PCostTech[tech,ec,area]= sum(FEPCP[fuelep,ec,area]*DmdFEPTechPrior[fuelep,tech,ec,area] for fuelep in FuelEPs)/sum(DmdFuelTechPrior[enduse,fuel,tech,ec,area] for enduse in Enduses,fuel in Fuels)

where:

PCostTech ‘Permit Cost ($/mmBtu)’ [Tech,EC,Area]

FEPCP, ‘Carbon Price by FuelEP ($/mmBtu)’ [FuelEP,EC,Area]

DmdFEPTechPrior, ‘Energy Demands (TBtu/Yr)’ [FuelEP,Tech,EC,Area]

Apply Calculated Value to Fuel Price Variables and Calculate End-Use Price

With costs calculated, price variables are populated for use in the demand module. Many of these equations are self-explanatory in the model code (ie: setting one value equal to another). Below are short descriptions for equations that are less intiutive

- FPCP, the carbon price by Fuel is set equal to FEPCP where the Fuel set has a match in the FuelEP set. For fuels not included in FEPCP a value is set using PCostTech weighted by fuel demands in the prior year.
- FPCFS, the price of CFS, is applied to demand sectors only if it is included in the CFS policy (CoverageCFS). In addition, additional cost for obligated sectors is added on top via the following

```
FPCFS[fuel,ec,area] = FPCFS[fuel,ec,area]+
FPCFSObligated[ecc,area]*sum(POCX[1,fuelep,ec,poll,area]*(1-
ZeroFr[fuelep,poll,area])*PolConv[poll]for poll in polls)/1e6
```

where:

FPCFS 'CFS Price ($/mmBtu)' [Fuel,EC,Area]
FpCFSObligated, 'CFS Price for Obligated Sectors ($/Tonnes)' [ECC,Area]
POCX, 'Marginal Pollution Coefficients (Tonnes/TBtu) [Enduse,FuelEP,EC,Poll,Area]'
ZeroFr, 'Fraction of Emissions from Zero Emission Sources (Tonnes/Tonnes)' [FuelEP,Poll,Area]
PolConv, 'Greenhouse Gas Coversion (eCO2 Tonnes/Tonnes)' [Poll]

- FPCFSNet, the net CFS price by Fuel, has a special exception added in for British Columbia to account for historical policies in this Area. For BC, FPCFSNet is set to zero in historical years (CTime < HisTime). In forecast years, it is set to the difference of FPCFS from the equation above compared to its value in the last historical year (FPCFSLast)
- The enduse fuel prices used by consumer choice equations in the demand module (ECFP/ECFPFuel) are the sum of newly set fuel price variables

```
@. ECFPFuel = FPEC+FPCFSNet+FPCP+FPOGEC
```

where:

ECFPFuel 'Fuel Price w/CFS Price ($/mmBtu)' [Fuel,EC,Area]
FPEC 'Fuel Prices excluding Emission Costs ($/mmBtu)' [Fuel,EC,Area]
FPCFSNet, 'Net CFS Price ($/mmBtu)' [Fuel,EC,Area]
FPCP 'Carbon Price before OBA ($/mmBtu)' [Fuel,EC,Area]
FPOGEC 'OGEC Price ($/mmBtu)' [Fuel,EC,Area]

Function SmoothEmissionCosts

A smoothed value for marginal emissions costs is calculated for energy efficiency investments based on pollution costs or carbon tax price depending on switches set by the model. A marginal cost is first set as follows

```
if PrPCostSw == 1.0
  @. PrPCostMar = PCostTech+FPCFSTech
elseif PrPCostSw == 2.0
  for tech in Techs,ec in ECs,area in Areas
    PrPCostMar[tech,ec,area] = eCO2Price[area]+eCO2PriceExo[area]*Inflation[area]
  end
else
  PrPCostMar = PrPCostPrior
end
```

where:

PrPCostMar 'Marginal Pollution Cost ($/mmBtu)' [Tech,EC,Area]
PrPCostPrior 'Pollution Cost ($/mmBtu)' [Tech,EC,Area]

eCO2Price ‘Carbon Tax plus Permit Cost’ ($/eCO2 Tonnes) [Area]
eCO2PriceExo ‘Carbon Tax plus Permit Cost’ ($/eCO2 Tonnes) [Area]

The marginal cost is smoothed if previous values exist

```
if PrPCostMarPrior[tech,ec,area] == 0
  PrPCost[tech,ec,area] = PrPCostMar[tech,ec,area]
else
  PrPCost[tech,ec,area] =PrPCostMar[tech,ec,area]*(1/PrPCostAT[ec,area])+PrPCostPrior[tech,ec,area]*(1-1/PrPCostAT[ec,area])
end
```

where:

PrPCost ‘Pollution Cost ($/mmBtu)’ [Tech,EC,Area]
PrPCostAT ‘Pollution Cost Adjustment Time (Years)’ [EC,Area]

Function MarginalCostOfFuelUsage

This function contains the equations for device operation and maintenance costs (DOMC) and the marginal cost of fuel usage (MCFU).

Calculate Device Operation and Maintenance Costs

Each device has a cost to operate and maintain it over its lifetime. This is computed on an annual basis as some fraction (DOCF) of the total cost of the device (DCCFullCost). More expensive devices are assumed to have more expensive operation and maintenance costs associated with them.

The calculation for Device Operation and Maintenance Costs (DOMC) is:

@. DOMC = DOCF*DCCFullCost

where:

DOMC ‘Device Operation and Maintenance Costs ($/mmBtu/Yr)’ [Enduse,Tech,EC,Area]
DOCF ‘Device Operating Cost Fraction ($/Yr/$)’ [Enduse,Tech,EC,Area]
DCCFullCost ‘Device Capital Cost Full Cost ($/mmBtu/Yr)’ [Enduse,Tech,EC,Area]

Calculate Marginal Fuel Cost

Each specific demand for energy is associated with a stock of capital. Investment in each type of capital stock by fuel type is allocated according to the cost of using each type of fuel. This cost is the perceived cost to the user and includes a risk factor (incorporated in the calculation of DCCR), annualized capital costs (DCCR*DCC), operating and maintenance costs (DOMC), and delivered marginal fuel costs (ECFP/DEE).

With the addition of an incentive, two other terms are needed. DCCU is the amount of rebate being offered. It is subtracted from the capital cost (DCC) after it is modified by the DEER - the policy participation response rate that indicates what fraction of those purchasing a device will select the rebated device and participate in the rebate program.

For example, a 100% participation rate (DEER=1) would reduce the term in the brackets to (DCC-DCCU).

The marginal cost of using energy (MCFU) includes the cost of using energy for all end-uses. As such, a house that has a gas furnace but an electric water heater would be represented partially in the model's gas capital stock and partially in the electric capital stock. The investment includes capital using energy in addition to the energy source equipment.

MCFU is then used to determine price-based process efficiencies (PEEPrice) and the marginal market share (MMSF) of each fuel with regard to new capital additions.

MCFU[enduse,tech,ec,area] = DCCR[enduse,tech,ec,area]*DCC[enduse,tech,ec,area]+
DOMC[enduse,tech,ec,area]+ECFP[enduse,tech,ec,area]/DEE[enduse,tech,ec,area]+
IdrtCost[enduse,tech,ec,area]*Inflation[area]

where:

MCFU 'Marginal Cost of Fuel Use ($/mmBtu)' [Enduse,Tech,EC,Area]
DCCR 'Device Capital Charge Rate ($/Yr/$)' [Enduse,Tech,EC,Area]
DCC 'Device Capital Cost ($/mmBtu/Yr)' [Enduse,Tech,EC,Area]
DOMC 'Device Operation and Maintenance Costs ($/mmBtu/Yr)' [Enduse,Tech,EC,Area]
ECFP 'Fuel Price ($/mmBtu)' [Enduse,Tech,EC,Area]
DEE 'Device Efficiency (Btu/Btu)' [Enduse,Tech,EC,Area]
IdrtCost 'Indirect Costs ($/mmBtu)' [Enduse,Tech,EC,Area]

MCFUPoll is also calculated as an input for calculating process efficiencies that include pollution prices (PEEPoll).

MCFUPoll[enduse,tech,ec,area] = DCCR[enduse,tech,ec,area]* DCC[enduse,tech,ec,area]+
DOMC[enduse,tech,ec,area]+ (ECFP[enduse,tech,ec,area]-PCostTech[tech,ec,area]-
FPCFSTech[tech,ec,area])/ DEE[enduse,tech,ec,area]+
IdrtCost[enduse,tech,ec,area]*Inflation[area]

where:

MCFUPoll 'Marginal Cost of Fuel Use from Pollution Price ($/mmBtu)' [Enduse,Tech,EC,Area]
PCostTech 'Permit Cost ($/mmBtu)' [Tech,EC,Area]
FPCFSTech 'CFS Price ($/mmBtu)' [Tech,EC,Area]

Function PRReductions: Pollution Reduction Device Calculations

This function calculates cost of emission reduction devices and builds reductions in response to emission cap policies. Cost and stock of reduction devices is dependent on type of emissions cap and trading market. The type of market is specified using the user specified 'CapTrade' switch. Equations for market components, such as indicated reductions, are also selected within each market using switches.

Map Emissions Policy Variables

The following equations map global emissions policy coverage, cost, and targeted level of emissions reductions into local variables. This allows for the model to begin the process of estimating the impact of a program across possibly the entire economy for the sectors covered by the Demand file

ECoverage[ec,poll,pcov,area] = ECovECC[ecc,poll,pcov,area]

PCCovTemp[fuelep,ec,poll,pcov,area] = (ECoverage[ec,poll,pcov,area]*PCovMap[fuelep,ecc,pcov,area])
PCCov[fuelep,ec,poll,area] = maximum(PCCovTemp[fuelep,ec,poll,:,area])

PCost[fuelep,ec,poll,area] = PCostECC[ecc,poll,area]*PCCov[fuelep,ec,poll,area]
GPFrac[ec,poll,pcov,area] = GPFrECC[ecc,poll,pcov,area]
RPolicy[ec,poll,area] = RPolECC[ecc,poll,area]
ECMarket[ec,market] = ECCMarket[ecc,market]

where:

ECoverage 'Emissions Coverage (1=Covered)' [EC,Poll,PCov,Area]
ECovECC 'Emissions Coverage (1=Covered)' [ECC,Poll,PCov,Area]
PCCov 'Emissions Coverage by Tech or Fuel (Tonnes/Tonnes)'
PCovMap 'Pollution Coverage Map (1=Mapped)' [FuelEP,ECC,PCov,Area]
PCost 'Permit Cost ($/Tonne)' [FuelEP,EC,Poll,Area]
PCostECC 'Permit Cost ($/Tonne)' [ECC,Poll,Area]
GPFrac 'Emissions Gratis Permit Fraction (Tonnes/Tonnes)' [EC,Poll,PCov,Area]
GPFrECC 'Emissions Gratis Permit Fraction (Tonnes/Tonnes)' [ECC,Poll,PCov,Area]
RPolicy 'Reduction Policy (Tonnes/Tonnes)' [EC,Poll,Area]
RPolECC 'Reduction Policy (Tonnes/Tonnes)' [ECC,Poll,Area]
ECCMarket 'Economic Categories included in Market' [ECC,Market]
ECMarket 'Economic Categories (ECs) in each Market' [EC,Market]

The following equation weights the device capital charge rate (DCCRP) to the appropriate FuelEP type using the first enduse.

enduse = 1

DCCRP[fuelep,ec,area] = sum(DCCRPrior[enduse,tech,ec,area]*DmdFEPTechPrior[fuelep,tech,ec,area] for tech in Techs)/sum(DmdFEPTechPrior[fuelep,tech,ec,area] for tech in Techs)

where:

DCCRP 'Device Capital Charge Rate for Policies ($/Yr/$)' [FuelEP,EC,Area]

DCCRPrior 'Device Capital Charge Rate in Prior Year ($/Yr/$)' [Enduse,Tech,EC,Area]

DmdFEPTechPrior, 'Energy Demands in Prior Year (TBtu/Yr)' [FuelEP,Tech,EC,Area]

Voluntary reductions are based on exogenous goal and time lag.

VR[fuelep,ec,poll,area] = VRPrior[fuelep,ec,poll,area]+DT*(VRP[fuelep,ec,poll,area]-VRPrior[fuelep,ec,poll,area])/VRRT[ec]

where:

VR 'Voluntary Reduction Policy (Tonnes/Tonnes)' [FuelEP,EC,Poll,Area]
VRP 'Voluntary Reduction Policy (Tonnes/Tonnes)' [FuelEP,EC,Poll,Area]
VRRT 'Voluntary Reduction response time (Years)' [EC]
DT 'Time Period'

Calculate Emission Reductions and Costs – Cap with Trading (CapTrade = 1)

The methodology for calculating the amount of reduction and reduction costs from an emissions policy is dependent on the specified type of cap and trade market. The capital cost for reduction technologies is based on reduction cost curve input data. Available input data currently contains different sources for some sectors with a corresponding difference in methodology needed to produce a dollar per tonne reduced result. These different inputs are referred to in the model code by the year of the source.

The following equation calculates the marginal reduction capital cost for markets specified as an emissions cap with emissions trading.

RCC[fuelep,ec,poll,area] = (PCost[fuelep,ec,poll,area]*PCCov[fuelep,ec,poll,area]*Inflation[area]+PCostExo[fuelep,ec,poll,area]*Inflation[area])/ (DCCRP[fuelep,ec,area]+ROCF[fuelep,ec,poll,area])

where:

RCC 'Reduction Capital Cost ($/Tonne)' [FuelEP,EC,Poll,Area]
PCostExo, 'Marginal Exogenous Permit Cost (Real $/Tonnes)' [FuelEP,EC,Poll,Area]
ROCF 'Pollution Reduction O&M ($/Tonne)' [FuelEP,EC,Poll,Area]

The indicated reductions are a function of the marginal reduction price and reduction curve parameters. The following equation calculates indicated reductions if the sector curve data is from 2009 (RPCSw = 1).

IRP[fuelep,ec,poll,area] =ln(RCC[fuelep,ec,poll,area]/Inflation[area]/ (PCostN[fuelep,ec,poll,area]/PCostM[fuelep,ec,poll,area]/RCostM[fuelep,poll]))/PVF[fuelep,ec,poll,area]*PCCov[fuelep,ec,poll,area]

where:

IRP 'Indicated Pollutant Reduction (Tonnes/Tonnes)' [FuelEP,EC,Poll,Area]
PCostN 'Pollution Reduction Cost Normal ($/Tonne)' [FuelEP,EC,Poll,Area]
PCostM 'Permit Cost Multiplier ($/Tonne/$/Tonne)' [FuelEP,EC,Poll,Area]
RCostM 'Reduction Cost Technology Multiplier ($/$)' [FuelEP,Poll]

PVF ‘Pollution Reduction Variance Factor (($/Tonne)/($/Tonne))’ [FuelEP,EC,Poll,Area]
RPCSw ‘Pollution Reduction Curve Switch (1=2009,2=2004)’ [EC,Poll,Area]
CapTrade ‘Emission Cap and Trading Switch (1=Trade, Cap Only=2)’ [Market]

The following equation calculates indicated reductions if the sector curve data is from 2004 (RPCSw = 2).

@. CostMax = max(((PCost+PCostExo)*PCostM),0.01)

IRP[fuelep,ec,poll,area] =
1/(1+((CostMax[fuelep,ec,poll,area]*RCostM[fuelep,poll])/PCostN[fuelep,ec,poll,area])^PVF[fuelep,ec,poll,area])*PCCov[fuelep,ec,poll,area]

Changes in actual reductions (RP) is produced from changes in indicated reductions, reduction construction time and physical lifespans

RP[fuelep,ec,poll,area] =
RPPrior[fuelep,ec,poll,area]+(max(RPPrior[fuelep,ec,poll,area],IRP[fuelep,ec,poll,area])-RPPrior[fuelep,ec,poll,area])/RCD[ec,poll]-RPPrior[fuelep,ec,poll,area]/RCPL[ec,poll]

where:

RP 'Pollutant Reduction (Tonnes/Tonnes)' [FuelEP,EC,Poll,Area]
RCD 'Reduction Capital Construction Delay (Years)' [EC,Poll]
RCPL 'Reduction Capital Physical Life (Years)' [EC,Poll]

Calculate Emission Reductions and Costs – Cap with No Trading (CapTrade = 2)

In markets with no trading, the reduction capital cost is based on the amount reduced instead of the permit price. Curve parameters develop the marginal capital costs as positively correlated with need for emissions reductions. Reductions are defined by the policy variable, by a marginal emissions coefficient standard, or can be exogenously specified as follows.

RP[fuelep,ec,poll,area] = max(RPolicy[ec,poll,area], (1-minimum(POCS[:,fuelep,ec,poll,area])/max(maximum(POCX[:,fuelep,ec,poll,area]),0.000001)))*PCCov[fuelep,ec,poll,area]

@. RPFull = 1-(1-RP)*xRM

where:

POCS ‘Pollution Standards (Tonnes/TBtu)’ [Enduse,FuelEP,EC,Poll,Area]

POCX ‘Marginal Pollution Coefficients (Tonnes/TBtu)’ [Enduse,FuelEP,EC,Poll,Area]

RPFull 'Pollutant Reduction after Adjustments (Tonnes/Tonnes)' [FuelEP,EC,Poll,Area]

xRM ‘Exogenous Average Pollution Coefficient Reduction Multiplier (Tonnes/Tonnes)' [FuelEP,EC,Poll,Area]

The cost of emission reductions using 2009 input data (RPCSw = 1) is calculated as follows

RCC[fuelep,ec,poll,area] =
(PCostN[fuelep,ec,poll,area]/RCostM[fuelep,poll])*exp(PVF[fuelep,ec,poll,area]*RPFull[fuelep,ec,poll,area])*Inflation[area]

The cost of emission reductions using 2004 (RPCSw = 2) input data is calculated as follows

RCC[fuelep,ec,poll,area] =(1/RPFull[fuelep,ec,poll,area]-1)^(1/PVF[fuelep,ec,poll,area])*PCostN[fuelep,ec,poll,area]/RCostM[fuelep,poll]/(DCCRP[fuelep,ec,area]+ROCF[fuelep,ec,poll,area])*Inflation[area]

The cost of emission reductions using 2011 input data (RPCSw = 3) is calculated as follows

RCC[fuelep,ec,poll,area]=(PCostN[fuelep,ec,poll,area]/RCostM[fuelep,poll])*exp(PVF[fuelep,ec,poll,area]*RPFull[fuelep,ec,poll,area])*Inflation[area]

A smoothed permit cost is calculated using the marginal reduction cost, capital charge rate, and reduction operation and maintenance costs

IPCost=RCC*(DCCRP+ROCF)/Infla

IPCost[fuelep,ec,poll,area] = RCC[fuelep,ec,poll,area]*(DCCRP[fuelep,ec,area]+ROCF[fuelep,ec,poll,area])/Inflation[area]

PCost[fuelep,ec,poll,area] =PCostPrior[fuelep,ec,poll,area]+(IPCost[fuelep,ec,poll,area]-PCostPrior[fuelep,ec,poll,area])/RCD[ec,poll]

where:

IPCost 'Indicated Permit Cost ($/Tonne)' [FuelEP,EC,Poll,Area]

The final reduction multiplier is calculated including voluntary reductions

@. RM = (1.0-RP)*(1.0-VR)*xRM

where:

RM 'Reduction Multiplier (Tonnes/Tonnes)' [FuelEP,EC,Poll,Area]

Calculate Energy Impacts

Impact of reduction devices on energy consumption is also accounted for using an input energy multiplier

RPEI[enduse,tech,ec,area] =
prod((1+sum(RP[fuelep,ec,poll,area]*DmdFEPTechPrior[fuelep,tech,ec,area]for fuelep in FuelEPs)/sum(DmdFEPTechPrior[fuelep,tech,ec,area] for fuelep in FuelEPs)*RPEIX[enduse,tech,ec,area]) for poll in Polls)

where:

RPEI 'Energy Impact of Pollution Reduction (Btu/Btu)' [Enduse,Tech,EC,Area]

RPEIX 'Energy Impact of Pollution Reduction Coefficient (Btu/Btu/Tonne/Tonne)' [Enduse,Tech,EC,Area]

B. Device Efficiency and Device Capital Cost Functions

Function DMarginal: Device Efficiency and Capital Costs

In this function, the device efficiencies (DEE), device capital costs (DCC), the marginal cost of fuel use (MCFU) and the device capital charge rate (DCCR) are computed.

Calculate Device Capital Charge Rate

The device capital charge rate is the annualization of device capital expenses (over the life of the device - DTL), accounting for taxes (TXRT), tax credits (DIVTC, DPIVTC), and return of principal and on investment (including risk premiums and inflation: 1+ROIN+CROIN+DRISK+INSM). (1-(1/(1+ROIN+CROIN+DRISK))**DPLN)/(1-TXRT) is the classical capital recovery term. The (1-TXRT) term at the end converts the after tax calculation into before tax dollars. Investment tax credits reduce the cost of the facility by the tax credit after the first year of operation using nominal dollars. Therefore, the value of the tax credit is ((DIVTC+DPIVTC)/(1+ROIN+CROIN+DRISK+INSM). Depreciation is modeled as a current dollar phenomenon which does not account for inflation. Therefore the net present value of the energy is calculated with the nominal rate of return: (2/DTL)/(ROIN+CROIN+DRISK+INSM+2/DTL)). It shows up as an additional negative term in the capital cost modifiers of DCCR because depreciation is a benefit (negative cost).

Device capital costs (DCC) are multiplied by the DCCR to get the annualized cost of the device used in computing market share calculations.

The formula for calculating the device capital charge rate:

DCCR[enduse,tech,ec,area] = (1-(DIVTC[tech,area] +DPIVTC)/ (1+ROIN[ec,area]-CROIN[enduse,tech,ec,area]+DRisk[enduse,tech] +InSm[area])-

TxRt[ec,area] * (2/DTL[enduse,tech,ec,area]) / (ROIN[ec,area] - CROIN[enduse,tech,ec,area] + DRisk[enduse,tech] +

InSm[area] + 2 /DTL[enduse,tech,ec,area])) * (ROIN[ec,area] - CROIN[enduse,tech,ec,area] + DRisk[enduse,tech]) /

(1-(1/(1+ROIN[ec,area] - CROIN[enduse,tech,ec,area] + DRisk[enduse,tech])) ^ DPLN[enduse,tech,ec,area]) / (1-TxRt[ec,area])

where:

DCCR 'Device Capital Charge Rate ($/Yr/$)' [Enduse,Tech,EC,Area]

DIVTC 'Device Investment Tax Credit ($/$)' [Tech,Area]

DPLN 'Physical Life of Equipment (Years)' [Enduse,Tech,EC,Area]

DRisk ‘Device Excess Risk ($/$)’ [Enduse,Tech]

DTL ‘Device Tax Life (Years)’ [Enduse,Tech,EC,Area]

InSm ‘'Smoothed Inflation Rate ($/Yr/$)'

ROIN ‘Return on Investment ($/Yr/$)’ [EC,Area]

CROIN ‘Conservation Return on Investment ($/Yr/$)’ [Enduse,Tech,EC,Area]

TxRt ‘Tax Rate on Energy Consumer ($/$) [EC,Area]’

Develop Device Efficiency Curve for Pollution Costs

Device efficiencies can optionally be calculated using the costs of pollution as an additional input. This option is described in more detail in the following section covering device efficiencies. But to have this option available we must first estimate the curve parameters as a function of user input curve values and the smoothed emissions cost value

First, an initial value is estimated using the input parameters and the smoothed value:

```
DEMMMRaw[enduse,tech,ec,area] = 1+DEEC0[enduse,tech,ec,area]/
(1+DEEA0[enduse,tech,ec,area]*(PrPCost[tech,ec,area]/Inflation[area]*
Inflation2010[area])^ DEEB0[enduse,tech,ec,area])*(PrPCost[tech,ec,area]/PrPCost[tech,ec,area])
```

where:

DEMMRaw ‘Multiplier Initial Value’ [Enduse,Tech,EC,Area]
DEEC0 ‘Device C0 Coeffcient for Efficiency Program (Btu/Btu)’ [Enduse,Tech,EC,Area]
DEEA0 ‘Device A0 Coeffcient for Efficiency Program (Btu/Btu)’ [Enduse,Tech,EC,Area]
DEEB0 ‘Device B0 Coeffcient for Efficiency Program (Btu/Btu)’ [Enduse,Tech,EC,Area]
PrPCost ‘Pollution Cost ($/mmBtu)’ [Tech,EC,Area]

Endogenous and exogenous values are set dependent on model switches and emissions policy coverages. The final multiplier value is the higher of the two.

```
CO2 = Select(Poll,"CO2")
Energy = Select(PCov,"Energy")
..
DEMMMEndo[enduse,tech,ec,area] = DEMMMRaw[enduse,tech,ec,area]*
ECoverage[ec,CO2,Energy,area]+1.000*(1-ECoverage[ec,CO2,Energy,area])

DEMMMExo[enduse,tech,ec,area] = DEMMMRaw[enduse,tech,ec,area]*
ECovExo[ec,CO2,Energy,area]+1.000*(1-ECovExo[ec,CO2,Energy,area])

DEMMM[enduse,tech,ec,area] = max(DEMMMEndo[enduse,tech,ec,area],
DEMMMExo[enduse,tech,ec,area])
```

where:

DEMMM ‘Device Efficiency Multiplier for Efficiency Program (Btu/Btu)’ [Enduse,Tech,EC,Area]
DEMMEndo ‘Endogenous Value’ [Enduse,Tech,EC,Area]

DEMMExo ‘Exogenous Value’ [Enduse,Tech,EC,Area]
DEMMRaw ‘Multiplier Initial Value’ [Enduse,Tech,EC,Area]
ECoverge ‘Emissions Coverage (1=Covered)’ [EC,Poll,PCov,Area]
ECovExo ‘Emissions Coverage for Exogenous Cap-and-Trade (1=Covered)’ [EC,Poll,PCov,Area]

The efficiency capital cost multiplier is then calculated in the same manner, using the input curve values and emissions costs. A user input multiplier for cost to efficiency (DCDEM) can be used to adjust cost impacts

DCMMMRaw[enduse,tech,ec,area] = 1+DCCC0[enduse,tech,ec,area]/
(1+DCCA0[enduse,tech,ec,area]*(PrPCost[tech,ec,area]/Inflation[area]*
Inflation2010[area])^DCCB0[enduse,tech,ec,area])*
(PrPCost[tech,ec,area]/PrPCost[tech,ec,area])+
(DEMMM[enduse,tech,ec,area]-1)*DCDEM[enduse,tech,ec,area]

DCMMMEndo[enduse,tech,ec,area] = DCMMMRaw[enduse,tech,ec,area]*
ECoverage[ec,CO2,Energy,area]+1.000* (1-ECoverage[ec,CO2,Energy,area])

DCMMMExo[enduse,tech,ec,area] = DCMMMRaw[enduse,tech,ec,area]*
ECovExo[ec,CO2,Energy,area]+1.000*(1-ECovExo[ec,CO2,Energy,area])

DCMMM[enduse,tech,ec,area] =
max(DCMMMEndo[enduse,tech,ec,area],DCMMMExo[enduse,tech,ec,area])

where:

DCMMM ‘Device Cost Efficiency Multiplier for Efficiency Program (Btu/Btu)’ [Enduse,Tech,EC,Area]
DCMMEndo ‘Endogenous Value’ [Enduse,Tech,EC,Area]
DCMMExo ‘Exogenous Value’ [Enduse,Tech,EC,Area]
DCMMRaw ‘Multiplier Initial Value’ [Enduse,Tech,EC,Area]
DCDEM ‘Device Cost to Efficiency Multiplier for Efficiency Program ($/$/(Btu/Btu))’ [Enduse,Tech,EC,Area]
ECoverge ‘Emissions Coverage (1=Covered)’ [EC,Poll,PCov,Area]
ECovExo ‘Emissions Coverage for Exogenous Cap-and-Trade (1=Covered)’ [EC,Poll,PCov,Area]
DCCC0 ‘Device Capital Cost C0 Coeffcient for Efficiency Program (Btu/Btu)’ [Enduse,Tech,EC,Area]
DCCA0 ‘Device Capital Cost A0 Coeffcient for Efficiency Program (Btu/Btu)’ [Enduse,Tech,EC,Area]
DCCB0 ‘Device Capital Cost B0 Coeffcient for Efficiency Program (Btu/Btu)’ [Enduse,Tech,EC,Area]
PrPCost ‘Pollution Cost ($/mmBtu)’ [Tech,EC,Area]

Calculate Device Efficiencies

The device efficiencies selected in the model are based on local fuel prices. As fuel prices rise, consumers will select higher efficiency devices, although not at the optimum level. The level of device efficiency selected, and other factors, has an associated capital cost that becomes the device capital cost.

ENERGY 2100 uses tables to determine device efficiencies selected at different price levels. Given fuel prices and emission costs, marginal energy efficiencies and capital costs for devices are determined using efficiency/cost trade-off curves based on consumer preferences. When energy prices increase, energy consumers could reduce their energy bills by purchasing more efficient devices or homes which use less energy (e.g. a more efficient furnace or more wall insulation). The increase in efficiency requires added equipment, design quality, or materials unless the response is to lose energy service (e.g. turn back the thermostat). DMARGINAL deals only with the increased efficiency of devices. CMARGINAL determines process efficiency in a similar manner. The consumer response of “cutting back” is addressed in UTILIZE.

Developing a trade-off curve between efficiency and capital costs begins with the identification of the upper bound on efficiency set by current technology. The technology upper bound can be raised through research and development until theoretical thermal efficiencies are reached. The rest of the curve represents the minimum cost of using energy for each efficiency level, done balancing the capital and operating costs of each device against its efficiency level and fuel costs. The values for the efficiency (DEE) and capital cost (DCC) curves are interpolated based on input values.

As fuel costs increase, total energy use costs can be minimized by increasing the energy efficiency even though higher efficiency requires equipment with a higher capital cost. The trade-off can be as a market share calculation where the “share” is the fraction of the maximum efficiency. The perceived costs of higher efficiency both by the manufacturer and the consumer determine the shape of the efficiency curve.

The marginal device efficiency (DEE) is determined by the relative energy price (ECFP,DFPN) and the device efficiency curve parameters (DEM,DFTC). Changes in the device capital charge rate (DCCRB, DCCR) use an assumed Cobb-Douglas substitution from Capital. The device efficiency multiplier (DEMM), the domestic grant fraction (DGF) and the device price multiplier (DEPM) are policy variables. The level of efficiency using only energy price is calculated below:

```
DEEPrice[enduse,tech,ec,area] =
DEM[enduse,tech,ec,area]*DEMM[enduse,tech,ec,area]*
(1/(1+(ECFP[enduse,tech,ec,area]/Inflation[area]*
 DEPM[enduse,tech,ec,area]/DFPN[enduse,tech,ec,area])^DFTC[enduse,tech,ec,area]*
 (1-DGF[enduse,tech,ec,area])*((1+STX[area])/(1+STXB[area]))*
 (DCCR[enduse,tech,ec,area]/DCCRB[enduse,tech,ec,area])))
```

where:

DEEPrice 'Device Efficiency from Energy Price (Btu/Btu)' [Enduse,Tech,EC,Area]
DEM 'Maximum Device Efficiency (Btu/Btu)' [Enduse,Tech,EC,Area]
DEMM 'Maximum Device Efficiency Multiplier (Btu/Btu)' [Enduse,Tech,EC,Area]
ECFP ‘Fuel Price ($/mmBtu)’ [Enduse,Tech,EC,Area]
DEPM 'Device Energy Price Multiplier ($/$)' [Enduse,Tech,EC,Area]
DFPN 'Normalized Fuel Price ($/mmBtu)' [Enduse,Tech,EC,Area]
DFTC 'Device Fuel Trade Off Coefficient (DLESS)' [Enduse,Tech,EC,Area]

DGF 'Domestic Grant Fraction ($/$)' [Enduse,Tech,EC,Area]
STX 'Sales Tax Rate on Energy Consumer ($/$)' [Area]
STXB 'Sales Tax Rate on Energy Consumer ($/$)' [Area]
DCCR 'Device Capital Charge Rate ($/Yr/$)' [Enduse,Tech,EC,Area]
DCCRB 'Base Case Device Capital Charge Rate ($/Yr/$)' [Enduse,Tech,EC,Area]

The equation below indicates that the final device efficiency (DEE) is the maximum of the computed efficiency and any the device efficiency standard (DEStd) in place. If the standard is effective, DEE will increase (i.e. the standard forces customers to choose higher levels of device efficiency than they would have given the current set of prices and capital costs.) DEStd is the variable the model uses to incorporate existing standards. DESTtdP is for testing policies that include new standards. The value before adjustment (DEEBeforeStd) is saved for output and use during calibration.

The efficiency output value is constrained to be only as high as 98% of the maximum value to limit unexpected exponential impacts in device cost equations. Optionally, an efficiency floor can be set (DEEFloorSw) to ensure that marginal efficiency does not drop below prior values.

DEEBeforeStd = DEEPrice

DEEPrice[enduse,tech,ec,area] = min(max(DEEPrice[enduse,tech,ec,area],

DEStd[enduse,tech,ec,area],DEStdP[enduse,tech,ec,area],

(DEEPrior[enduse,tech,ec,area]*DEEFloorSw[area])),

DEM[enduse,tech,ec,area]*DEMM[enduse,tech,ec,area]*0.98)

where:

DEEPrice 'Device Efficiency from Energy Price (Btu/Btu)' [Enduse,Tech,EC,Area]

DEEBeforeStd 'Device Efficiency Before Standard (Btu/Btu)' [Enduse,Tech,EC,Area]

DEStd ‘Device Efficiency Standards (Btu/Btu)’ [Enduse,Tech,EC,Area]

DEStdP ‘Device Efficiency Standards Policy (Btu/Btu)’ [Enduse,Tech,EC,Area]

DEEPrior ‘Device Efficiency in Prior Year (Btu/Btu)’ [Enduse,Tech,EC,Area]

DEEFloorSw ‘Switch to Activate Floor for Device Efficiency’ (1=Activate) [EC,Area]

DEM 'Maximum Device Efficiency (Btu/Btu)' [Enduse,Tech,EC,Area]

DEMM 'Maximum Device Efficiency Multiplier (Btu/Btu)' [Enduse,Tech,EC,Area]

The level of efficiency can also be estimated by adding the cost of emissions as a modifier to energy prices depending on model parameters as follows

DEEPoll[enduse,tech,ec,area] = DEM[enduse,tech,ec,area]* DEMM[enduse,tech,ec,area]*
DEMMM[enduse,tech,ec,area]* (1/(1+((ECFP[enduse,tech,ec,area]- PCostTech[tech,ec,area]-
FPCFSTech[tech,ec,area])/ Inflation[area]* DEPM[enduse,tech,ec,area]/

```
DFPN[enduse,tech,ec,area])^DFTC[enduse,tech,ec,area]* (1-
DGF[enduse,tech,ec,area])*((1+STX[area])/(1+STXB[area]))*(DCCR[enduse,tech,ec,area]/DCCRB[enduse,t
ech,ec,area])))
```

where:

DEEPoll 'Device Efficiency from Pollution Price (Btu/Btu)' [Enduse,Tech,EC,Area]
DEM 'Maximum Device Efficiency (Btu/Btu)' [Enduse,Tech,EC,Area]
DEMM 'Maximum Device Efficiency Multiplier (Btu/Btu)' [Enduse,Tech,EC,Area]
DEMMM 'Device Efficiency Multiplier for Efficiency Program (Btu/Btu)' [Enduse,Tech,EC,Area]
ECFP 'Fuel Price ($/mmBtu)' [Enduse,Tech,EC,Area]
PCostTech 'Permit Cost ($/mmBtu)' [Tech,EC,Area]
DEPM 'Device Energy Price Multiplier ($/$)' [Enduse,Tech,EC,Area]
DFPN 'Normalized Fuel Price ($/mmBtu)' [Enduse,Tech,EC,Area]
DFTC 'Device Fuel Trade Off Coefficient (DLESS)' [Enduse,Tech,EC,Area]
DGF 'Domestic Grant Fraction ($/$)' [Enduse,Tech,EC,Area]
STX 'Sales Tax Rate on Energy Consumer ($/$)' [Area]
STXB 'Sales Tax Rate on Energy Consumer ($/$)' [Area]
DCCR 'Device Capital Charge Rate ($/Yr/$)' [Enduse,Tech,EC,Area]
DCCRB 'Base Case Device Capital Charge Rate ($/Yr/$)' [Enduse,Tech,EC,Area]

Similar to DEEPrice, the pollution price efficiency value has standards applied and can be adjusted.

```
DEEPoll[enduse,tech,ec,area] = min(max(DEEPoll[enduse,tech,ec,area],
DEStd[enduse,tech,ec,area],DEStdP[enduse,tech,ec,area],
(DEEPrior[enduse,tech,ec,area]*DEEFloorSw[area])),
DEM[enduse,tech,ec,area]*DEMM[enduse,tech,ec,area]*DEMMM[enduse,tech,ec,area]*0.98)
```

Select Marginal Efficiency using DEESw

So far, we have calculated a marginal device efficiency using two different methods (DEEPrice, DEEPoll), but not directly set a value for the output efficiency used in the rest of the demand module (DEE). Here we pick a value to use using an input switch parameter (DEESw) that specifies which variable is selected for efficiency.

```
if DEESw[enduse,tech,ec,area] == 1.0
  DEE[enduse,tech,ec,area] = DEEPrice[enduse,tech,ec,area]
elseif DEESw[enduse,tech,ec,area] == 2.0
  DEE[enduse,tech,ec,area] = DEEPoll[enduse,tech,ec,area]
elseif DEESw[enduse,tech,ec,area] == 3.0
  DEE[enduse,tech,ec,area] = max(DEEPrice[enduse,tech,ec,area],DEEPoll[enduse,tech,ec,area])
elseif DEESw[enduse,tech,ec,area] == 6.0
  DEE[enduse,tech,ec,area] = DEERef[enduse,tech,ec,area]
…
elseif DEESw[enduse,tech,ec,area] == 0.0
  DEE[enduse,tech,ec,area] = xDEE[enduse,tech,ec,area]
end
```

where:

DEE 'Device Efficiency (Btu/Btu)' [Enduse,Tech,EC,Area]

DEESw ‘Switch for Device Efficiency (Switch)’ [Enduse,Tech,EC,Area]
DEEPrice 'Device Efficiency from Energy Price (Btu/Btu)' [Enduse,Tech,EC,Area]
DEEPoll 'Device Efficiency from Pollution Price (Btu/Btu)' [Enduse,Tech,EC,Area]
DEERef ‘Device Efficiency in Reference Case (Btu/Btu)’ [Enduse,Tech,EC,Area]
xDEE ‘Historical Device Efficiency (Btu/Btu)’ [Enduse,Tech,EC,Area]

Note that DEESw allows for other types of curve equations to be executed to match outside analysis or expectations. These are generally given a DEESw parameter 10 or above as a programming convention. These equations are not discussed in detail here since they can vary by model version and user.

Efficiency is constrained a final time to account for any theoretical thermal limitations added by the user. A common assumption is that certain technologies can not have an efficiency above 1.0.

DEE[enduse,tech,ec,area] = min(DEE[enduse,tech,ec,area],DEEThermalMax[enduse,tech,ec,area])

where:

DEE ‘Device Efficiency (Btu/Btu)’ [Enduse,Tech,EC,Area]
DEEThermalMax ‘Thermal Maximum Device Efficiency (Btu/Btu) [Enduse,Tech,EC,Area]

Calculate Device Capital Costs

The device capital cost (DCC) is computed based on the level of efficiency selected. The relationship between the normalized device capital cost (DCCN), the efficiency curve parameters, and device capital cost trade off coefficient (DCTC) is used to develop the capital cost. A multiplier on capital costs that calculates non-efficiency related price changes is included (such as ice makers in refrigerators -DCMM) and sales tax is also accounted for (1+STX).

DCCPrice[enduse,tech,ec,area] = DCCN[enduse,tech,ec,area]*DCMM[enduse,tech,ec,area]*
(DEM[enduse,tech,ec,area]*DEMM[enduse,tech,ec,area]/ DEEPrice[enduse,tech,ec,area]-1)^
(1/min(DCTC[enduse,tech,ec,area],-0.01))* (1+STX[area])*(1-DGF[enduse,tech,ec,area])*Inflation[area]

where:

DCCPrice 'Device Capital Cost from Energy Price ($/mmBtu/Yr)' [Enduse,Tech,EC,Area]
DCCN ‘Device Capital Cost ($/mmBtu)' [Enduse,Tech,EC,Area]
DCMM 'Capital Cost Maximum Multiplier ($/$)' [Enduse,Tech,EC,Area]
DEM 'Maximum Device Efficiency (Btu/Btu)' [Enduse,Tech,EC,Area]
DEMM 'Maximum Device Efficiency Multiplier (Btu/Btu)' [Enduse,Tech,EC,Area]
DEEPrice 'Device Efficiency from Energy Price (Btu/Btu)' [Enduse,Tech,EC,Area]
DCTC 'Device Cap. Trade Off Coefficient (DLESS)' [Enduse,Tech,EC,Area]
STX ‘Sales Tax Rate on Energy Consumer ($/$)’ [Area]
DGF 'Domestic Grant Fraction ($/$)' [Enduse,Tech,EC,Area]

The output capital cost by price can optionally be adjusted using DCCLimit as a constraint

DCCPrice[enduse,tech,ec,area] = max(0,min(DCCPrice[enduse,tech,ec,area],
xDCCYr[enduse,tech,ec,area]*DCCLimit[enduse,tech,ec,area]*
DCMM[enduse,tech,ec,area]*Inflation[area]))

where:

xDCCYr 'Historical Device Capital Cost ($/mmBtu/Yr)' [Enduse,Tech,EC,Area

DCCLimit 'Device Capital Cost Limit Multiplier ($/$)' [Enduse,Tech,EC,Area]

Similar to device efficiency, the impact of pollution costs can be used as an input for developing the marginal capital cost.

DCCPoll[enduse,tech,ec,area] = DCCN[enduse,tech,ec,area]*DCMM[enduse,tech,ec,area]*
DCMMM[enduse,tech,ec,area]*(DEM[enduse,tech,ec,area]*DEMM[enduse,tech,ec,area]*
DEMMM[enduse,tech,ec,area]/DEEPoll[enduse,tech,ec,area]-1)^(1/min(DCTC[enduse,tech,ec,area],-0.01))*(1+STX[area])*(1-DGF[enduse,tech,ec,area]) *Inflation[area]

DCCPoll[enduse,tech,ec,area] = max(0,min(DCCPoll[enduse,tech,ec,area],
xDCCYr[enduse,tech,ec,area]*DCCLimit[enduse,tech,ec,area]*
DCMM[enduse,tech,ec,area]*DCMMM[enduse,tech,ec,area]*Inflation[area]))

where:

DCCPoll 'Device Capital Cost from Pollution Price ($/mmBtu/Yr)' [Enduse,Tech,EC,Area]

DCMMM 'Device Cost Efficiency Multiplier for Efficiency Program ($/$)' [Enduse,Tech,EC,Area]

DEMMM 'Device Efficiency Multiplier for Efficiency Program (Btu/Btu)' [Enduse,Tech,EC,Area]

DEEPoll 'Device Efficiency from Pollution Price (Btu/Btu)' [Enduse,Tech,EC,Area]

The desired variable selected for use as the marginal output value (DCC) is then selected using DEESw to match the efficiency value. Finally, the full marginal capital cost value is calculated by accounting for grants and changes in tax rates:

DCCFullCost[enduse,tech,ec,area] = DCC[enduse,tech,ec,area]*
((1+STXB[area])/(1+STX[area]))/(1-DGF[enduse,tech,ec,area])

where:

DCCFullCost 'Device Capital Cost Full Cost ($/mmBtu/Yr)' [Enduse,Tech,EC,Area]

DCC 'Device Capital Cost ($/mmBtu/Yr)' [Enduse,Tech,EC,Area]

STXB 'Sales Tax Rate on Energy Consumer in Base Case ($/$)' [Area]

STX 'Sales Tax Rate on Energy Consumer ($/$)' [Area]

DGF 'Domestic Grant Fraction ($/$)' [Enduse,Tech,EC,Area]

Function DDSM: Device Demand Side Management Programs

This function simulates the consumer response (DEER) to a demand side management program (rebate, standard or interest reduction) and its effects on device efficiency (DEE) and capital costs (DCC)).

Modify Efficiency with Policy Values

The device efficiency, device capital cost, device capital charge rate, and marginal cost of fuel use before the incentive programs are saved to use for before and after comparisons.

DCCBefore = DCC

DEEBefore = DEE
DCCRBefore = DCCR
MCFUBefore = MCFU

The policy levels for cost and efficiency are set using the policy input values if the inputs are higher than the current values

DEEPolicy[enduse,tech,ec,area] = max(xDEEPolicy[enduse,tech,ec,area],DEEBefore[enduse,tech,ec,area])

DCCPolicy[enduse,tech,ec,area] = max(xDCCPolicy[enduse,tech,ec,area]*Inflation[area],
DCCBefore[enduse,tech,ec,area])

where:

DEEPolicy 'Policy Device Efficiency (Btu/Btu)' [Enduse,Tech,EC,Area]
DCCPolicy 'Capital Cost of Policy Device ($/mmBtu/Yr)' [Enduse,Tech,EC,Area]
xDEEPolicy 'Input Policy Device Efficiency (Btu/Btu)' [Enduse,Tech,EC,Area]
xDCCPolicy 'Input Capital Cost of Policy Device ($/mmBtu/Yr)' [Enduse,Tech,EC,Area]

Calculate the Device Capital Charge Rate with an Incentive

The device capital charge rate with incentives (DCCRPolicy) is very similar to the device capital charge rate without incentives (DCCR). It annualizes device capital expenses (over the life of the device - DTL), accounting for taxes (TXRT), investment tax credits plus any policy investment tax credits (DIVTC+DPIVTC), and return of principal and on investment (including interest on loans minus any subsidy (ROIN-CROIN), risk premiums and inflation: 1+(ROIN-CROIN)+DRISK+INSM). (1-1/(1+(ROIN-CROIN)+DRISK))**DPLN)/(1-TXRT) is the classical capital recovery term, with the additional term (CROIN) accounting for any interest subsidy. The (1-TXRT) term at the end converts the after-tax calculation into before tax dollars. Investment tax credits (including policy tax credits -DIVTC+DPIVTC) reduce the cost of the facility by the tax credit after the first year of operation using nominal dollars. Therefore, the value of the tax credit is ((DIVTC+DPIVTC)/(1+(ROIN-CROIN)+DRISK+INSM).

Depreciation is modeled as a current dollar phenomenon which does not account for inflation. Therefore, the net present value of the energy is calculated with the nominal rate of return: (2/DTL)/((ROIN-CROIN)+DRISK+INSM+2/DTL)). It shows up as an additional negative term in the capital cost modifiers of DCCR because depreciation is a benefit (negative cost). Device capital costs (DCC) are multiplied by DCCRPolicy to get the annualized cost of the device when incentives are present used in computing market share calculations.

The formula for calculating the device capital charge rate when subsidies are present is:

DCCRPolicy[enduse,tech,ec,area] = (1-(DIVTC[tech,area]+DPIVTC)/
(1+ROIN[ec,area]-CROIN[enduse,tech,ec,area]+DRisk[enduse,tech]+InSm[area])-
TxRt[ec,area]*(2/DTL[enduse,tech,ec,area])/
(ROIN[ec,area]-CROIN[enduse,tech,ec,area]+DRisk[enduse,tech]+InSm[area]+
2/DTL[enduse,tech,ec,area]))*(ROIN[ec,area]-CROIN[enduse,tech,ec,area]+DRisk[enduse,tech])/
(1-(1/(1+ROIN[ec,area]-CROIN[enduse,tech,ec,area]+
DRisk[enduse,tech]))^DPLN[enduse,tech,ec,area])/(1-TxRt[ec,area])

where:

DCCRPolicy ‘Device Capital Charge Rate for Policy Device ($/Yr/$)’ [Enduse,Tech,EC,Area]

DIVTC ‘Device Investment Tax Credit ($/$)’ [Tech,Area]

DIVTCP ‘Device Policy Investment Tax Credit ($/$)’

ROIN ‘Return on Investment ($/Yr/$)’ [EC,Area]

CROIN ‘Conservation Return on Investment ($/Yr/$)’ [Enduse,Tech,EC,Area]

DRisk ‘Device Excess Risk ($/$)’ [Enduse,Tech]

InSm ‘'Smoothed Inflation Rate ($/Yr/$)'

TxRt ‘Tax Rate on Energy Consumer ($/$) [EC,Area]’

DTL ‘Device Tax Life (Years)’ [Enduse,Tech,EC,Area]

DPLN ‘Physical Life of Equipment (Years)’ [Enduse,Tech,EC,Area]

C. Adjust Market Shares for High Efficiency Technologies

The policy value for the device capital charge rate and the efficiency and cost values calculated above are used to produce the policy marginal cost of fuel use. Any input subsidy values are included in the calculation.

```
MCFUPolicy[enduse,tech,ec,area] = (DCCPolicy[enduse,tech,ec,area]-
DCCSubsidy[enduse,tech,ec,area]*Inflation[area])*
(DCCRPolicy[enduse,tech,ec,area]+DOCF[enduse,tech,ec,area])+ECFP[enduse,tech,ec,area]/
DEEPolicy[enduse,tech,ec,area]+IdrtCost[enduse,tech,ec,area]*Inflation[area]
```

where:

MCFUPolicy ‘Marginal Cost of Fuel Use for Policy Device ($/mmBtu)’ [Enduse,Tech,EC,Area]

DCCPolicy ‘Capital Cost of Policy Device ($/mmBtu/Yr)’ [Enduse,Tech,EC,Area]

DCCSubsidy ‘Device Capital Cost Subsidy ($/mmBtu/Yr)’ [Enduse,Tech,EC,Area]

DCCRPolicy ‘Device Capital Charge Rate for Policy Device ($/Yr/$)’ [Enduse,Tech,EC,Area]

DOCF ‘Device Operating Cost Fraction ($/Yr/$)’ [Enduse,Tech,EC,Area]

ECFP ‘Fuel Price ($/mmBtu)’ [Enduse,Tech,EC,Area]

DEEPolicy ‘Policy Device Efficiency (Btu/Btu)’ [Enduse,Tech,EC,Area]

IdrtCost ‘Indirect Costs ($/mmBtu)’ [Enduse,Tech,EC,Area]

Adoption rates are estimated using policy input parameters (SbVF, SbMSM0) and updated using the policy marginal cost of fuel use compared it to the prior value. Optionally, a user specified exogenous value can be used for the high efficiency market share.

```
@. MAWBefore = exp(SbVF*log(MCFUBefore/MCFUBefore))
@. MAWPolicy = exp(SbMSM0+SbVF*log(MCFUPolicy/MCFUBefore))
@. DEEPolicyMSF = MAWPolicy/(MAWBefore+MAWPolicy)
..
if xDEEPolicyMSF[enduse,tech,ec,area] >= 0.0
  DEEPolicyMSF = xDEEPolicyMSF
end
```

where:

DEEPolicyMSF ‘Policy Participation Response (Btu/Btu)’ [Enduse,Tech,EC,Area]

xDEEPolicyMSF ‘Policy Participation Response Exogenous (Btu/Btu)’ [Enduse,Tech,EC,Area]
SbVF ‘Price Variance Factor for High Efficiency Device Market Share ($/$)’ [Enduse,Tech,EC,Area]
SbMSM0 ‘Non-Price Factor for High Efficiency Device Market Share ($/$)’ [Enduse,Tech,EC,Area]

Calculate Device Efficiencies and Capital Costs with Incentives

The device efficiency (DEE) is determined by the efficiency curves and the capital cost (DCC) including rebate (DCCU). A capital cost incentive increases the actual capital related expenses in that the utility picks up the ADDED cost while the customer spends the same amount of money to get a better device. For each dollar level, more efficiency can be purchased.

```
@. DEE = DEEBefore*(1-DEEPolicyMSF)+DEEPolicy*DEEPolicyMSF
```

where:

DEE ‘Device Efficiency (Btu/Btu)’ [Enduse,Tech,EC,Area]
DEEPolicyMSF ‘Policy Participation Response (Btu/Btu)’ [Enduse,Tech,EC,Area]
DCCSubsidy ‘Device Capital Cost Subsidy ($/mmBtu/Yr)’ [Enduse,Tech,EC,Area]
DEEPolicy ‘Policy Device Efficiency (Btu/Btu)’ [Enduse,Tech,EC,Area]

The device capital cost (DCC) is computed to include the impact of devices from the policy combined with other devices to produce an overall marginal capital cost value.

```
DCC[enduse,tech,ec,area] = DCCBefore[enduse,tech,ec,area]*
(1-DEEPolicyMSF[enduse,tech,ec,area])+
(DCCPolicy[enduse,tech,ec,area]-DCCSubsidy[enduse,tech,ec,area]*Inflation[area])*
 DEEPolicyMSF[enduse,tech,ec,area]
```

The full cost with is also recalculated with policy devices. The capital charge rate is also updated using the new marginal cost value.

```
DCCFullCost[enduse,tech,ec,area] = (DCC[enduse,tech,ec,area]+
DCCSubsidy[enduse,tech,ec,area]*Inflation[area]*
DEEPolicyMSF[enduse,tech,ec,area])*((1+STXB[area])/(1+STX[area]))/
(1-DGF[enduse,tech,ec,area])
…
@. DCCR = (DCCRBefore*(1-DEEPolicyMSF))+(DCCRPolicy*DEEPolicyMSF)
```

where:

DCCR 'Device Capital Charge Rate ($/Yr/$)' [Enduse,Tech,EC,Area]
DCCFullCost ‘Device Capital Cost Full Cost ($/mmBtu/Yr)’ [Enduse,Tech,EC,Area]

D. Process Efficiency and Capital Costs Functions

Function CMarginal: Process Efficiency and Capital Costs

In this function, the marginal process efficiencies (PEE), capital costs (PCC) and the process capital charge rate (PCCR) are computed.

Calculate Process Capital Charge Rate

The process capital charge rate is the annualization of process capital expenses (over the life of the process capital - PETL), accounting for taxes (TXRT), tax credits (PIVTC), and return of principal and on investment (including and inflation: 1+ROIN-CROIN+INSM). (1-(1/(1+ROIN-CROIN))**PEPLN)/(1-TXRT) is the classical capital recovery term. The (1-TXRT) term at the end converts the after tax calculation into before tax dollars. Investment tax credits reduce the cost of the facility by the tax credit after the first year of operation using nominal dollars. Therefore, the value of the tax credit is (PIVTC/(1+ROIN-CROIN+INSM). Depreciation is modeled as a current dollar phenomenon which does not account for inflation. Therefore, the net present value of the energy is calculated with the nominal rate of return: (2/PETL)/(ROIN-CROIN+INSM+2/PETL)). It shows up as an additional negative term in the capital cost modifiers of PCCR because depreciation is a benefit (negative cost).

Process capital costs (PCC) are multiplied by the PCCR to get the annualized cost of the process capital used in computing market share calculations.

PCCR[enduse,tech,ec,area] = (1-PIVTC[area]/(1+ROIN[ec,area] - CROIN[enduse,tech,ec,area] +InSm[area])-TxRt[ec,area]* (2/PETL[enduse,tech,ec,area]) /(ROIN[ec,area]-CROIN[enduse,tech,ec,area]+InSm[area]+2/PETL[enduse,tech,ec,area]))*(ROIN[ec,area] - CROIN[enduse,tech,ec,area])/(1-(1/(1+ROIN[ec,area]-CROIN[enduse,tech,ec,area]))^PEPLN[enduse,tech,ec,area])/(1-TxRt[ec,area])

Where:

PCCR 'Process Capital Charge Rate ($/Yr/$)' [Enduse,Tech,EC,Area]
PIVTC ‘Process Policy Investment Tax Credit ($/$)’ [Area]
ROIN ‘Return on Investment ($/Yr/$)’ [EC,Area]
CROIN ‘Conservation Return on Investment ($/Yr/$)’ [Enduse,Tech,EC,Area]
PEPL 'Physical Life of Process Requirements (Years)’ [Enduse,Tech,EC,Area]
PETL 'Tax Life of Process Requirements (Years)' [Enduse,Tech,EC,Area]
InSm ‘'Smoothed Inflation Rate ($/Yr/$)'
TxRt ‘Tax Rate on Energy Consumer ($/$) [EC,Area]’
PEPLN ’Physical Life of Process Requirements (Years)’ [Enduse,Tech,EC,Area]

Develop Process Efficiency Curve for Pollution Costs

Similar to devices, process efficiencies can optionally be calculated using the costs of pollution as an additional input. To have this option available we must first estimate the curve parameters as a function of user input curve values and the smoothed emissions cost value First, an initial value is estimated using the input parameters and the smoothed value:

PEMMMRaw[enduse,tech,ec,area] = 1+PEEC0[enduse,tech,ec,area]/(1+PEEA0[enduse,tech,ec,area]*(PrPCost[tech,ec,area]/Inflation[area]* Inflation2010[area])^ PEEB0[enduse,tech,ec,area])*(PrPCost[tech,ec,area]/PrPCost[tech,ec,area])

where:

PEMMRaw ‘Multiplier Initial Value’ [Enduse,Tech,EC,Area]

PEEC0 'Process C0 Coeffcient for Efficiency Program ($/Btu/($/Btu))' [Enduse,Tech,EC,Area]
PEEA0 'Process A0 Coeffcient for Efficiency Program ($/Btu/($/Btu))' [Enduse,Tech,EC,Area]
PEEB0 'Process B0 Coeffcient for Efficiency Program ($/Btu/($/Btu))' [Enduse,Tech,EC,Area]
PrPCost 'Pollution Cost ($/mmBtu)' [Tech,EC,Area]

Endogenous and exogenous values are set dependent on model switches and emissions policy coverages. The final multiplier value is the higher of the two.

```
CO2 = Select(Poll,"CO2")
Energy = Select(PCov,"Energy")
..
PEMMMEndo[enduse,tech,ec,area] = PEMMMRaw[enduse,tech,ec,area]*
ECoverage[ec,CO2,Energy,area]+1.000*(1-ECoverage[ec,CO2,Energy,area])

PEMMMExo[enduse,tech,ec,area] = PEMMMRaw[enduse,tech,ec,area]*
ECovExo[ec,CO2,Energy,area]+1.000*(1-ECovExo[ec,CO2,Energy,area])

PEMMM[enduse,tech,ec,area] = max(PEMMMEndo[enduse,tech,ec,area],
PEMMMExo[enduse,tech,ec,area])
```

where:

PEMMM 'Process Efficiency Multiplier for Efficiency Program ($/$)' [Enduse,Tech,EC,Area]
PEMMEndo 'Endogenous Value' [Enduse,Tech,EC,Area]
PEMMExo 'Exogenous Value' [Enduse,Tech,EC,Area]
PEMMRaw 'Multiplier Initial Value' [Enduse,Tech,EC,Area]
ECoverge 'Emissions Coverage (1=Covered)' [EC,Poll,PCov,Area]
ECovExo 'Emissions Coverage for Exogenous Cap-and-Trade (1=Covered)' [EC,Poll,PCov,Area]

The efficiency capital cost multiplier is then calculated in the same manner, using the input curve values and emissions costs. A user input multiplier for cost to efficiency (PCPEM) can be used to adjust cost impacts

```
PCMMMRaw[enduse,tech,ec,area] = 1+PCCC0[enduse,tech,ec,area]/
(1+PCCA0[enduse,tech,ec,area]*(PrPCost[tech,ec,area]/Inflation[area]*
Inflation2010[area])^PCCB0[enduse,tech,ec,area])*
(PrPCost[tech,ec,area]/PrPCost[tech,ec,area])+
(PEMMM[enduse,tech,ec,area]-1)*PCPEM[enduse,tech,ec,area]

PCMMMEndo[enduse,tech,ec,area] = PCMMMRaw[enduse,tech,ec,area]*
ECoverage[ec,CO2,Energy,area]+1.000* (1-ECoverage[ec,CO2,Energy,area])

PCMMMExo[enduse,tech,ec,area] = PCMMMRaw[enduse,tech,ec,area]*
ECovExo[ec,CO2,Energy,area]+1.000*(1-ECovExo[ec,CO2,Energy,area])
```

PCMMM[enduse,tech,ec,area] =
max(PCMMMEndo[enduse,tech,ec,area],PCMMMExo[enduse,tech,ec,area])

where:

PCMMM ‘Process Cost Efficiency Multiplier for Efficiency Program ($/$)’ [Enduse,Tech,EC,Area]
PCMMEndo ‘Endogenous Value’ [Enduse,Tech,EC,Area]
PCMMExo ‘Exogenous Value’ [Enduse,Tech,EC,Area]
PCMMRaw ‘Multiplier Initial Value’ [Enduse,Tech,EC,Area]
PCPEM ‘Device Cost to Efficiency Multiplier for Efficiency Program ($/$/(Btu/Btu))’ [Enduse,Tech,EC,Area]
ECoverge ‘Emissions Coverage (1=Covered)’ [EC,Poll,PCov,Area]
ECovExo ‘Emissions Coverage for Exogenous Cap-and-Trade (1=Covered)’ [EC,Poll,PCov,Area]
PCCC0 ‘Process Capital Cost C0 Coeffcient for Efficiency Program ($/Btu/($/Btu))’ [Enduse,Tech,EC,Area]
PCCA0 ‘Process Capital Cost A0 Coeffcient for Efficiency Program ($/Btu/($/Btu))’ [Enduse,Tech,EC,Area]
PCCB0 ‘Process Capital Cost B0 Coeffcient for Efficiency Program ($/Btu/($/Btu))’ [Enduse,Tech,EC,Area]
PrPCost ‘Pollution Cost ($/mmBtu)’ [Tech,EC,Area]

Calculate Process Efficiencies

The marginal process efficiency (PEE) is determined by the marginal fuel cost price (MCFU,PFPN) and the process efficiency curve parameters (PEM,PFTC). Changes in the process capital charge rate (PCCRB,PCCR) use an assumed Cobb-Douglas substitution from Capital. The process efficiency multiplier (PEMM) and the process price multiplier (PEPM) are policy variables. The level of efficiency using only marginal fuel cost is calculated below.

PEEPrice[enduse,tech,ec,area] = PEM[enduse,ec,area]*PEMM[enduse,tech,ec,area]*
(1/(1+(MCFU[enduse,tech,ec,area]/Inflation[area]*
PEPM[enduse,tech,ec,area]/PFPN[enduse,tech,ec,area])^PFTC[enduse,tech,ec,area]*
(PCCR[enduse,tech,ec,area]/PCCRB[enduse,tech,ec,area])))

where:

PEEPrice 'Process Efficiency ($/Btu)’ [Enduse,Tech,EC,Area]
PEM 'Maximum Process Efficiency ($/Btu)’ [Enduse,EC,Area]
PEMM ‘Process Efficiency Max. Mult. ($/Btu/($/Btu))’ [Enduse,Tech,EC,Area]
MCFU ‘Marginal Cost of Fuel Use ($/mmBtu)’ [Enduse,Tech,EC,Area]
PEPM 'Process Energy Price Mult. ($/$’) [Enduse,Tech,EC,Area]
PFPN ‘Process Normalized Fuel Price ($/mmBtu)’ [Enduse,Tech,EC,Area]
PFTC 'Process Fuel Trade Off Coefficient' [Enduse,Tech,EC,Area]
PCCR 'Process Capital Charge Rate ($/Yr/$)' [Enduse,Tech,EC,Area]
{CCRB 'Base Case Process Capital Charge Rate ($/Yr/$)' [Enduse,Tech,EC,Area]

The level of efficiency can also be estimated by adding the cost of emissions as a modifier to energy prices depending on model parameters as follows

PEEPoll[enduse,tech,ec,area] = PEM[enduse,ec,area]*PEMM[enduse,tech,ec,area]*
PEMMM[enduse,tech,ec,area]*(1/(1+(MCFUPoll[enduse,tech,ec,area]/Inflation[area]*
PEPM[enduse,tech,ec,area]/PFPN[enduse,tech,ec,area])^PFTC[enduse,tech,ec,area]*
(PCCR[enduse,tech,ec,area]/PCCRB[enduse,tech,ec,area])))

where:

PEEPoll 'Process Efficiency from Pollution Price ($/Btu)' [Enduse,Tech,EC,Area]
PEMMM 'Process Efficiency Multiplier for Efficiency Program ($/Btu/($/Btu))' [Enduse,Tech,EC,Area]
MCFUPoll 'Marginal Cost of Fuel Use from Pollution Price ($/mmBtu)' [Enduse,Tech,EC,Area]

Process efficiency can also be calculated as a function of the change in marginal fuel cost and a pre-defined price elasticity.

```
PEEElas[enduse,tech,ec,area] = PEM[enduse,ec,area]*PEMM[enduse,tech,ec,area] *
(1/(1+(MCFU[enduse,tech,ec,area]/Inflation[area]/MCFU0[enduse,tech,ec,area]) ^
PEElas[enduse,tech,ec,area]))
```

where:

PEEElas 'Process Efficiency from Long Term Price Elasticity ($/Btu)' [Enduse,Tech,EC,Area]
PEElas 'Long Term Price Elasticity for Process Efficiency ($/Btu)' [Enduse,Tech,EC,Area]

Select Marginal Efficiency using PEESw

Depending on model parameters, a further constraint on process efficiency (PEE) may be encountered. Final process efficiency (PEE) is the maximum of the computed efficiency and any the process efficiency standard (PEStd, PEStdP) in place. If the standard is effective, PEE will increase (i.e. the standard forces customers to choose higher levels of process efficiency than they would have given the current set of prices and capital costs.) PEStd is the variable the model uses to incorporate existing standards. PEStdP is for testing policies that include new standards.

Similar to devices, a value is selected using an input switch parameter (PEESw) that specifies which input variable is used for process efficiency in subsequent model equations. Process efficiency standards can be set using PEEPrice, PEEPoll, the higher of the two values, the elasticity efficiency, the reference case value, or the output of a separate curve equation.

The equations below executes when PEESw is 1.0. PEEBeforeStd is set to the calculated value of PEEPrice as the output value from the efficiency curve before standards are applied. PEEPrice is then adjusted if it is below a process efficiency standard (PEStd, PEStdP), below a prior value if the floor switch is activated (PEEFloorSw), or it is above a value approaching the curve maximum (PEM*PEMM*.98).

```
if PEESw[enduse,tech,ec,area] == 1
PEEBeforeStd[enduse,tech,ec,area] = PEEPrice[enduse,tech,ec,area]
PEEPrice[enduse,tech,ec,area] = min(max(PEEPrice[enduse,tech,ec,area] ,
PEStd[enduse,tech,ec,area],PEStdP[enduse,tech,ec,area],PEEPrior[enduse,tech,ec,area]*
PEEFloorSw[ec,area]) , PEM[enduse,ec,area]*PEMM[enduse,tech,ec,area]*0.98)

PEE[enduse,tech,ec,area] = PEEPrice[enduse,tech,ec,area]
```

where:

PEESw 'Switch for Process Efficiency (Switch)' [Enduse,Tech,EC,Area]
PEE 'Process Efficiency ($/Btu)' [Enduse,Tech,EC,Area]
PEEBeforeStd 'Process Efficiency Before Standards ($/Btu)' [Enduse,Tech,EC,Area]

PEEPrice 'Process Efficiency from Energy Price ($/Btu)' [Enduse,Tech,EC,Area]
PEEFloorSw 'Switch to Activate Floor for Process Efficiency (1=Activate)' [EC,Area]

PEESw will execute similar equations depending on its value:

- PEESw = 1 – PEE is set based on PEEPrice
- PEESw = 2 – PEE is set based on PEEPoll
- PEESw = 3 – PEE is set based the higher value between PEEPrice and PEEPoll
- PEESw = 5 – PEE is set based on PEEElas
- PEESw = 6 – PEE is set based on the reference case value (no standards applied)
- PEESw = 10 or above – PEE is set based on sector specific curve parameters
-

PEESw allows for other types of curve equations to be executed to match outside analysis or expectations. These are generally given a PEESw parameter 10 or above as a programming convention.

Calculate Process Capital Costs

The process capital cost (PCC) is computed based on the level of efficiency selected. The relationship between the normalized process capital cost (PCCN), the process efficiency curve parameters, and process capital cost trade off coefficient (PCTC) is used to develop the capital cost.

PCCPrice[enduse,tech,ec,area] = PCCN[enduse,tech,ec,area]*
PCCMM[enduse,tech,ec,area]*Inflation[area] *
(1+STX[area])*(PEM[enduse,ec,area]*PEMM[enduse,tech,ec,area] /
PEEPrice[enduse,tech,ec,area]-1)^(1/PCTC[enduse,tech,ec,area])

where:

PCCPrice 'Process Capital Cost from Energy Price ($/($/Yr))' [Enduse,Tech,EC,Area]
PCCN 'Normalized Process Capital Cost ($/mmBtu)' [Enduse,Tech,EC,Area]
PCTC 'Process Capital Trade Off Coefficient (DLESS)' [Enduse,Tech,EC,Area]

Similar to device efficiency, the impact of pollution costs can be used as an input for developing the marginal capital cost.

PCCPoll[enduse,tech,ec,area] = PCCN[enduse,tech,ec,area]*
PCCMM[enduse,tech,ec,area] *PCMMM[enduse,tech,ec,area]*Inflation[area]*
(1+STX[area])*(PEM[enduse,ec,area]*PEMM[enduse,tech,ec,area]*PEMMM[enduse,tech,ec,area] /
PEEPoll[enduse,tech,ec,area]-1)^(1/PCTC[enduse,tech,ec,area])

where:

PCCPoll 'Process Capital Cost from Pollution Price ($/($/Yr))' [Enduse,Tech,EC,Area]
PCMMM 'Process Cost Efficiency Multiplier for Efficiency Program ($/$)' [Enduse,Tech,EC,Area]

The desired variable selected for use as the marginal output value (PCC) is then selected using PEESw to match the efficiency value. Finally, the full marginal capital cost value is calculated by accounting for grants and changes in tax rates:

PCCFC[enduse,tech,ec,area] = PCC[enduse,tech,ec,area]*(1+STXB[area])/(1+STX[area])

where:

PCCFC 'Process Capital Cost Full Cost ($/($/Yr))' [Enduse,Tech,EC,Area]
STXB 'Sales Tax Rate on Energy Consumer in Base Case ($/$)' [Area]
STX 'Sales Tax Rate on Energy Consumer ($/$)' [Area]

Adjustment for Air Conditioning

If the segment contains an Air Conditioning (AC) enduse, then an additional adjustment is applied to simulate the link with heating systems. The air conditioning process efficiency (PEE(AC)) is based on the space heating process efficiency (PEE(Heat)), the heating-to-cooling ratio (CHR), the heating-to-cooling ratio multiplier (CHRM), and the process efficiency multipliers (PEMM, PEMMM) weighted by the process energy requirement (PEM).

For residential and commercial air conditioning:

PEE[AC,tech,ec,area] = sum(PEE[Heat,AllTechs,ec,area]*PERPrior[Heat,AllTechs,ec,area] for AllTechs in Techs)/sum(PERPrior[Heat,AllTechs,ec,area] for AllTechs in Techs) / (CHR[ec,area]*CHRM[ec,area])*PEMM[AC,tech,ec,area]*PEMMM[AC,tech,ec,area]

where:

PERPrior 'Process Energy Requirement in prior year (mmBtu/YR)' [Enduse,Tech,EC,Area]
CHR 'Cooling to Heating Ratio (Btu/Btu)' [EC,Area]
CHRM 'Cooling to Heating Ratio Multiplier' [EC,Area]

E. Energy Stock Changes Functions

Function AgeVintages

AgeVintages contains the equations that move energy requirements from the prior into the next vintage for the current. In addition, average efficiencies by vintage are calculated along with the shares of each vintage's energy requirement as part of the stock total.

Age Energy Requirements Using Prior Year's Values

First, the AgingFactor is calculated. This variable controls the rate at which energy requirements move to an older vintage. This variable is currently set as a local rule-of-thumb assumption using the older lifespan variable (DPL) as the aging rate, limited to be no lower than 100% per year (meaning stocks can not stay in the same vintage). This code is planned to be updated to add AgingFactor as an input that can be controlled by the user.

AgingFactor[enduse,tech,ec,area,vintage] = min(Int(length(Vintage))/DPL[enduse,tech,ec,area], 1.0)
DERAgedV[enduse,tech,ec,area,vintage] = DERVPrior[enduse,tech,ec,area,vintage] *AgingFactor[enduse,tech,ec,area,vintage]

where:

DERAgedV 'Energy Requirement Aged to Next Vintage (mmBtu/YR)' [Enduse,Tech,EC,Area,Vintage]
AgingFactor 'Aging Factor (1/DPL)' [Enduse,Tech,EC,Area,Vintage]

DPL ‘Physical Life of Equipment (Years)’ [Enduse,Tech,EC,Area]

DERVPrior ‘Energy Requirement in Previous Year (mmBtu/YR)’ [Enduse,Tech,EC,Area,Vintage]

The energy requirements in the first vintage from the prior year are removed from the first vintage in the current year. Using current assumptions for AgingFactor will move all the prior stock

```
FirstVintage = 1
  ..
DERV[enduse,tech,ec,area,FirstVintage] = DERVPrior[enduse,tech,ec,area,FirstVintage] -
DERAgedV[enduse,tech,ec,area,FirstVintage]
```

where:

DERV ‘Energy Requirement by Vintage (mmBtu/YR)’ [Enduse,Tech,EC,Area,Vintage]

DERVPrior ‘Energy Requirement in Previous Year (mmBtu/YR)’ [Enduse,Tech,EC,Area,Vintage]

DERAgedV ‘Energy Requirement Aged to Next Vintage (mmBtu/YR)’ [Enduse,Tech,EC,Area,Vintage]

The last vintage is defined locally as the length of the vintage set. All the vintages between the first and last are saved for use in the aging equation

```
LastVintage=Int(length(Vintage))
PenultimateVintage = LastVintage-1
vintages = collect(2:PenultimateVintage)
```

Stocks in the vintages between first and last are aged inside a for vintages loop. Stocks from the prior vintage are moved to the current vintage and stocks from the current vintage are moved to the next vintage.

```
vintageprior = vintage-1
DERV[enduse,tech,ec,area,vintage] = DERVPrior[enduse,tech,ec,area,vintage]-
        DERAgedV[enduse,tech,ec,area,vintage]+
DERAgedV[enduse,tech,ec,area,vintageprior]
```

Stocks in the final vintage in the current are adjusted by the stock in the prior vintage in the prior year. Note that stock from last vintage in the prior year is brought back. The survival rate used for the last vintage should be checked to make sure large amounts of stock are not accumulating in LastVintage.

```
DERV[enduse,tech,ec,area,LastVintage] =
DERVPrior[enduse,tech,ec,area,LastVintage]+DERAgedV[enduse,tech,ec,area,vintageprior]
```

Calculate Average Efficiency by Vintage

Average efficiency by vintage is adjusted using a similar method as the aging equations above, with the energy stocks used as a weight to estimate the efficiency of the stock per vintage. Note that the value in the first vintage is overwritten later in the demand module using the output marginal efficiency (DEE), which is not yet known for the current year.

First Vintage:

```
DEEAV[enduse,tech,ec,area,FirstVintage] =
```

```
(DERVPrior[enduse,tech,ec,area,FirstVintage] *DEEAVPrior[enduse,tech,ec,area,FirstVintage] -
DERAgedV[enduse,tech,ec,area,FirstVintage]* DEEAVPrior[enduse,tech,ec,area,FirstVintage])/
DERV[enduse,tech,ec,area,FirstVintage]
```

Second to penultimate Vintage:

```
DEEAV[enduse,tech,ec,area,vintage] =
(DERVPrior[enduse,tech,ec,area,vintage]* DEEAVPrior[enduse,tech,ec,area,vintage]-
DERAgedV[enduse,tech,ec,area,vintage]* DEEAVPrior[enduse,tech,ec,area,vintage] +
DERAgedV[enduse,tech,ec,area,vintageprior]* DEEAVPrior[enduse,tech,ec,area,vintageprior])/
DERV[enduse,tech,ec,area,vintage]
```

Last Vintage:

```
DEEAV[enduse,tech,ec,area,LastVintage] =
(DERVPrior[enduse,tech,ec,area,LastVintage]* DEEAVPrior[enduse,tech,ec,area,LastVintage]-
DERAgedV[enduse,tech,ec,area,vintageprior]* DEEAVPrior[enduse,tech,ec,area,vintageprior])/
DERV[enduse,tech,ec,area,LastVintage]
```

where:

DEEAV 'Average Device Efficiency by Vintage (Btu/Btu) [Enduse,Tech,EC,Area,Vintage]
DEEAVPrior 'Average Device Efficiency in Previous Year (Btu/Btu) [Enduse,Tech,EC,Area,Vintage]
DERAgedV 'Energy Requirement Aged to Next Vintage (mmBtu/YR)' [Enduse,Tech,EC,Area,Vintage]
DERVPrior 'Energy Requirement in Previous Year (mmBtu/YR)' [Enduse,Tech,EC,Area,Vintage]
DERV 'Energy Requirement by Vintage (mmBtu/YR)' [Enduse,Tech,EC,Area,Vintage]

Populate Device Allocations

Total energy requirement across vintage and the share of the current year total device stock per vintage is calculated for output and use in stock addition and retirement equations later in the module.

```
DERVSum[enduse,tech,ec,area] = sum(DERV[enduse,tech,ec,area,vintage] for vintage in Vintages)
for vintage in Vintages
  DERVAllocation[enduse,tech,ec,area,vintage] = DERV[enduse,tech,ec,area,vintage] /
  DERVSum[enduse,tech,ec,area]
end
```

where:

DERVAllocation 'Fraction of DER in each Vintage (mmBtu/YR)' [Enduse,Tech,EC,Area,Vintage]
DERVSum 'Sum of Energy Requirement by Vintage (mmBtu/YR)' [Enduse,Tech,EC,Area]
DERV 'Energy Requirement by Vintage (mmBtu/YR)' [Enduse,Tech,EC,Area,Vintage]

Function MStock: Marginal Stock Changes

This function calculates process and device energy additions.

Production capacity is associated with a service or process energy requirement. The process energy requirement (PER) is the delivered process energy needed to produce the output implied by the capital output. The device energy requirement (DER) represents the energy needed by furnaces, for example to provide a process energy requirement. The production capacity (PC) can last from 45 to 60 years. The devices which serve those stocks might only last 15 to 20 years. Therefore, the model separates the capital stock and their process requirements (EUPC, PER) into three age classes or vintages: new, middle, and old. As the stock passes from one age class to another, the devices which serve it are replaced. As a result, the device efficiency can change more often than the process efficiency.

Stock Life-Cycle Dynamics - Capital Stock Failure/Wear-out

Capital stock is retired (EUPCRPC) depending on the lifetime (PCPL) specified. If the lifetime is 30 years, then 1/30 of the capital stock (EUPC) is retired each year.

```
EUPCRPC[enduse,tech,age,ec,area] = EUPCPrior[enduse,tech,age,ec,area]/(PCPL[ecc,area]/3.0)
```

where:

EUPCRPC 'Production Capacity Retirements from Capacity Retirements (M$/Yr/Yr)' [Enduse,Tech,Age,EC,Area]

EUPCPrior 'Production Capacity by Enduse in Prior Year(M$/Yr)' [Enduse,Tech,Age,EC,Area]

PCPL 'Physical Life of Production Capacity (Years)' [ECC,Area]

Stock Life-Cycle Dynamics - Changes in Stock Due to New Stock Growth

Capital Additions (EUPCAPC) by technology and economic class are calculated by dividing the production capacity additions (PCA) by the market share fractions for each economic class and fuel.

```
EUPCAPC[enduse,tech,New,ec,area] = PCA[New,ecc,area]*MMSF[enduse,tech,ec,area]
```

Capital is “aged” by moving it into different vintages. The “retired” capital from one age category becomes the “new” capital in the next vintage:

```
Select Age(Mid,Old)
EUPCAPC[enduse,tech,age,ec,area] = EUPCRPC[enduse,tech,age-1,ec,area]
```

where:

EUPCAPC 'Production Capacity Additions from New Production Capacity (M$/Yr/Yr)' [Enduse,Tech,Age,EC,Area]

MMSF 'Market Share Fraction by Device ($/$)' [Enduse,Tech,EC,Area]

PCA Production Capacity Additions ((M$/YR)/YR) [Age,ECC,Area]

New Process Additions from Changes in Device Saturation

Every increase in device saturations will result in process additions equal to total production capacity (EUPC) less production capacity retirements (EUPCRPC). For example, an increase in the saturation of air conditioners will produce an increase in energy requirements for AC cooling without any increase in housing. The additional process requirements are calculated as the difference between saturation in current period (DST) and the device saturation in the previous period (DSTL) over the average process efficiency average (PEEA).

```
PERADSt[enduse,tech,ec,area] = (sum(EUPCPrior[enduse,tech,age,ec,area] for age in Ages)-
EUPCRPC[enduse,tech,Old,ec,area])*max(0,DSt[enduse,ec,area]-DStPrior[enduse,ec,area])/
PEEAPrior[enduse,tech,ec,area]
```

where:

PERADST 'Process Additions from Increases in Saturation (mmBtu/Yr/Yr)' [Enduse,Tech,EC,Area]
PEEAPrior 'Average Process Efficiency in Prior Year ($/Btu)' [Enduse,Tech,EC,Area]
DST 'Device Saturation (Btu/Btu)' [Enduse,EC,Area]
DSTPrior 'Device Saturation in Prior Year (Btu/Btu)' [Enduse,EC,Area]

Process Retirements from Changes in Device Saturation

Similar to the above equation, a decrease in device saturation compared to the previous period will retire process requirements.

PERRDSt[enduse,tech,ec,area] = (sum(EUPCPrior[enduse,tech,age,ec,area] for age in Ages)-
EUPCRPC[enduse,tech,Old,ec,area])*max(0,DStPrior[enduse,ec,area]-DSt[enduse,ec,area])/
PEEAPrior[enduse,tech,ec,area]

where:

PERRDST ‘Process Retire. from Reductions in Saturation (mmBtu/Yr/Yr)' [Enduse,Tech,EC,Area]

New Process Additions from Capital Additions

Every new capital addition (EUPCAPC) causes a fuel-specific service energy requirement (PERAPC). This requirement is modified by the device saturation and divided by the marginal process energy efficiency (dollars of output per service energy input).

PERAPC[enduse,tech,ec,area] = EUPCAPC[enduse,tech,New,ec,area]*
DSt[enduse,ec,area]/PEE[enduse,tech,ec,area]

where:

PERAPC 'Process Additions from Production Capacity Additions (mmBtu/Yr/Yr)' [Enduse,Tech,EC,Area]
PEE 'Process Efficiency ($/Btu)’ [Enduse,Tech,EC,Area]

Process Energy Retirements Due to Capital Retirements

Process retirements due to capital retirements are accounted the process energy removed due to the retired capital (EUPCRPC/PEEA - note the use of average efficiency and not marginal) modified by the device saturation as of the last period (DSTL).

PERRPC[enduse,tech,ec,area] = EUPCRPC[enduse,tech,Old,ec,area]*DStPrior[enduse,ec,area]
/PEEAPrior[enduse,tech,ec,area]

where:

PERRPC 'Process Retire. from Production Capacity Retire. (mmBtu/Yr/Yr)' [Enduse,Tech,EC,Area]

Device Retirements Due to Process Retirements and Changes in Device Saturation

When capital stock is retired, the devices are retired as well. The process energy requirements (PERRPC) eliminated due to capital stock retirements and process energy retirements from change in device saturation are divided by the average device efficiency (DEEA) to produce the device energy requirement (DERRPC) than is eliminated as well.

@. DERRPC = (PERRPC+PERRDSt)/DEEAPrior

where:

DERRPC 'Device Retire. from Production Capacity Retirements and Reductions in Device Saturation (mmBtu/Yr/Yr)' [Enduse,Tech,EC,Area]

DEEAPrior 'Average Device Efficiency in Prior Year (Btu/Btu)' [Enduse,Tech,EC,Area]

Device Replacements Due to Process Additions and Changes in Device Saturation

New devices are added to meet the needs of additions to process and increases in device saturations. Note that equation below uses marginal device efficiency (DEE) since the requirements are being met by new devices

@. DERAPC = (PERAPC+PERADSt)/DEE

where:

DERAPC 'Device Additions from PCap Additions & Increases in Device Saturation (mmBtu/Yr/Yr)' [Enduse,Tech,EC,Area]

Changes in Process Stock Due to Normal Failures and Replacement

In addition to process retirements, some stock is replaced due to process failure. Process failure or wear-out (PERRP) is calculated as the process energy requirement (PER) less capital and saturation retirements (PERRPC,PERRDST) divided by the process capital lifetime (PEPL).

@. PERRP = (PERPrior-PERRPC-PERRDSt)/PEPL

where:

PERRP 'Process Retire. from Process Retire. (mmBtu/Yr/Yr)' [Enduse,Tech,EC,Area]

Process Replacement with New Efficiency

Retired process requirements from aging are replaced by new stock using the marginal process efficiency (PEE). If the marginal efficiency is greater than the average (PEEA) then process requirements will fall (the replacement stock will be more efficient than the existing stock).

@. PERAP = PERRP*PEEAPrior/PEE

where:

PERAP 'Process Additions from Process Retire. (mmBtu/Yr/Yr)' [Enduse,Tech,EC,Area]

Device Retirements Due to Changes in Process Stock Aging

Devices are retired if the process retirements (PERRP) from aging are greater than the process additions (PERAPC) at the level of existing stock efficiency (DEEA). If additions are greater than retirements then no devices are replaced due to process aging.

DERRP[enduse,tech,ec,area] = max(0,(PERRP[enduse,tech,ec,area]-PERAP[enduse,tech,ec,area]))/
DEEAPrior[enduse,tech,ec,area]

where:

DERRP 'Device Retire. from Process Retire. (mmBtu/Yr/Yr)' [Enduse,Tech,EC,Area]

Device Additions Due to Changes in Process Stock Aging

Similar to above, devices are added from process aging if the additions (PERAPC) are greater than the retirements (PERRP) at the level of new marginal efficiency (DEE)

DERAP[enduse,tech,ec,area] = max(0,(PERAP[enduse,tech,ec,area]-PERRP[enduse,tech,ec,area]))/DEE[enduse,tech,ec,area]

where:

DERAP 'Device Additions from Process Retire. (mmBtu/Yr/Yr)' [Enduse,Tech,EC,Area]

Function TStock: Total Stock Update

The function updates the total productive capacity (EUPC), the process energy requirements (PER) and the device energy requirements (DER) with additions and retirements. All three variables are calculated in the same general fashion - the difference between additions to and subtractions from existing capacity and energy requirements are added to the current respective levels.

Productive capacity is updated by summing the calculated additions and retirements and using the summed values to update the total stock. Total capacity is then updated with any input stock adjustments.

```
@. EUPCA=EUPCAPC+EUPCAC
@. EUPCR=EUPCRPC+EUPCRC
@. EUPC=EUPC+DT*(EUPCA-EUPCR)

 EUPCAdj[enduse,tech,age,ec,area] = EUPC[enduse,tech,age,ec,area]*
        StockAdjustment[enduse,tech,ec,area]

 EUPC[enduse,tech,age,ec,area] = EUPC[enduse,tech,age,ec,area]+EUPCAdj[enduse,tech,age,ec,area]
```

where:

EUPC 'Production Capacity by Enduse (M$/Yr)' [Enduse,Tech,Age,EC,Area]
EUPCA 'Production Capacity Additions ((M$/YR)/YR)' [Enduse,Tech,Age,EC,Area]
EUPCR 'Production Capacity Retirement ((M$/YR)/YR)' [Enduse,Tech,Age,EC,Area]
EUPCAdj 'Production Capacity Adjustment (M$/Yr/Yr)' [Enduse,Tech,Age,EC,Area]
StockAdjustment 'Exogenous Capital Stock Adjustment ($/$)' [Enduse,Tech,EC,Area]

Process energy requirements are updated by summing the calculated additions and retirements and using the summed values to update the total stock. Note the process requirements additionally includes changes from retrofits and saturations.

```
@. PERA=PERAPC+PERADST+PERAP+PERARC
@. PERR=PERRPC+PERRDST+PERRP+PERRR+PERRRC

PER[enduse,tech,ec,area] = PERPrior[enduse,tech,ec,area]+
DT* (PERA[enduse,tech,ec,area]-PERR[enduse,tech,ec,area])

@. PERAdj = PER*StockAdjustment
@. PER = PER+PERAdj
```

Exogenous reductions from policies are also calculated and removed from the stock.

```
@. PERRExo = PER*PERReduction

PER[enduse,tech,ec,area] = max(0,(PER[enduse,tech,ec,area]-PERRRExo[enduse,tech,ec,area]))
```

where:

PER 'Process Energy Requirement (mmBtu/Yr/Yr)' [Enduse,Tech,EC,Area]
PERPrior 'Process Energy Requirement in Prior Year (mmBtu/Yr/Yr)' [Enduse,Tech,EC,Area]
PERA 'Process Energy Requirement Additions (mmBtu/Yr/Yr)' [Enduse,Tech,EC,Area]
PERR 'Process Energy Requirement Retirements (mmBtu/Yr/Yr)' [Enduse,Tech,EC,Area]
PERAdj 'Process Energy Requirement Adjustments (mmBtu/Yr/Yr)' [Enduse,Tech,EC,Area]
PERReduction 'Process Energy Exogenous Retrofits Percentage ((mmBtu/Yr)/(mmBtu/Yr))' [Enduse,Tech,EC,Area]

Total device energy additions are updated in a similar manner as above

```
@. DERA = DERAPC+DERAP+DERAD+DERARC
```

where:

DERA 'Energy Requirement Addition (mmBtu/YR)' [Enduse,Tech,EC,Area]
DERAPC 'Device Additions from PCap Additions & Increases in Device Saturation (mmBtu/Yr/Yr)' [Enduse,Tech,EC,Area]
DERAP 'Device Additions from Process Retire. (mmBtu/Yr/Yr)' [Enduse,Tech,EC,Area]
DERAD 'Device Additions from Device Retirements (mmBtu/yr)' [Enduse,Tech,EC,Area]
DERARC 'Device Additions from Conversions (mmBtu/Yr/Yr)' [Enduse,Tech,EC,Area]

Stock at the vintage level is updated by applying device energy additions to the first vintage. The average efficiency of the first vintage is also updated using marginal device efficiency.

```
firstvintage = 1
DERAV[enduse,tech,ec,area,firstvintage] = DERA[enduse,tech,ec,area]
DEEAV[enduse,tech,ec,area,firstvintage] =  (DERV[enduse,tech,ec,area,firstvintage]*
        DEEAV[enduse,tech,ec,area,firstvintage]+
        DERA[enduse,tech,ec,area]*DEE[enduse,tech,ec,area])/
        (DERV[enduse,tech,ec,area,firstvintage]+DERA[enduse,tech,ec,area])
```

where:

DERAV 'Energy Requirement Addition (mmBtu/YR)'[Enduse,Tech,EC,Area,Vintage]
DERV 'Energy Requirement by Vintage (mmBtu/YR)' [Enduse,Tech,EC,Area,Vintage]
DEEAV 'Average Device Efficiency by Vintage (Btu/Btu)' [Enduse,Tech,EC,Area,Vintage]
DEE 'Device Efficiency (Btu/Btu)' [Enduse,Tech,EC,Area]

Device retirements are calculated at the vintage level, using the device retirements and conversions calculated early in the module. Retirements not at the vintage level, such as retirements caused from capacity or retrofits, are allocated at the vintage level using DERVAllocation. Process retirements from retrofits are calculated directly (PERRR/DEEAPrior) instead of using an intermediary variable.

```
DERRV[enduse,tech,ec,area,vintage] =
DERRDV[enduse,tech,ec,area,vintage]+DERRRCV[enduse,tech,ec,area,vintage]+
(DERRPC[enduse,tech,ec,area]+DERRP[enduse,tech,ec,area]+DERRR[enduse,tech,ec,area]+
PERRR[enduse,tech,ec,area]/DEEAPrior[enduse,tech,ec,area])*
DERVAllocation[enduse,tech,ec,area,vintage]
```

where:

DERRV ‘Device Energy Requirement Retirements by Vintage (mmBtu/YR)’ [Enduse,Tech,EC,Area,Vintage]

DERRDV ‘Device Retire from Device Retire. by Vintage (mmBtu/YR)’ [Enduse,Tech,EC,Area,Vintage]

DERRRCV ‘Device Retirements from Conversions by Vintage (mmBtu/Yr/Yr)’ [Enduse,Tech,EC,Area,Vintage]

DERRPC 'Device Retire. from Production Capacity Retirements and Reductions in Device Saturation (mmBtu/Yr/Yr)' [Enduse,Tech,EC,Area]

DERRP 'Device Retire. from Process Retire. (mmBtu/Yr/Yr)' [Enduse,Tech,EC,Area]

DERRR 'Device Energy Retire. Retrofit ((mmBtu/Yr)/Yr)' [Enduse,Tech,EC,Area]

PERRR 'Process Energy Retire. Process Retrofit ((mmBtu/Yr)/Yr)' [Enduse,Tech,EC,Area]

DEEAPrior ‘Average Device Efficiency in Prior Year (Btu/Btu)’ [Enduse,Tech,EC,Area]

DERVAllocation ‘Fraction of DER in each Vintage (mmBtu/YR)’ [Enduse,Tech,EC,Area,Vintage]

The current device stock levels are updated using additions and retirements at the vintage level. This level is then adjusted using stock adjustments

```
DERV[enduse,tech,ec,area,vintage] = DERV[enduse,tech,ec,area,vintage]+DT*
(DERAV[enduse,tech,ec,area,vintage]-DERRV[enduse,tech,ec,area,vintage])
…
DERV[enduse,tech,ec,area,vintage] =
DERV[enduse,tech,ec,area,vintage]*(1+StockAdjustment[enduse,tech,ec,area])
```

where:

DT ‘Time Period'

StockAdjustment ‘Exogenous Capital Stock Adjustment ($/$)’ [Enduse,Tech,EC,Area]

The current device stock levels are updated using additions and retirements at the vintage level. This level is then adjusted using stock adjustments. This variable can be utilized by the user to adjust stock levels for events not covered by changes in economy activity, aging, retrofitting, or conversions. As an example, the stock can be exogenously adjusted to simulate the impacts of a natural disaster.

```
DERV[enduse,tech,ec,area,vintage] = DERV[enduse,tech,ec,area,vintage]+DT*
(DERAV[enduse,tech,ec,area,vintage]-DERRV[enduse,tech,ec,area,vintage])
…
DERV[enduse,tech,ec,area,vintage] =
DERV[enduse,tech,ec,area,vintage]*(1+StockAdjustment[enduse,tech,ec,area])
```

where:

DERV ‘Energy Requirement by Vintage (mmBtu/YR)’ [Enduse,Tech,EC,Area,Vintage]

DT ‘Time Period'

StockAdjustment ‘Exogenous Capital Stock Adjustment ($/$)’ [Enduse,Tech,EC,Area]

Other exogenous stock reductions are incorporated to adjust the device requirement stock. These reductions are typically calculated as an impact from a policy. These variables are currently not defined at the vintage level, so they are applied using the summed stock across vintage (DER) then re-allocated to vintages.

First, we sum across stock, then save the value before adjusting it.

```
DER[enduse,tech,ec,area] = sum(DERV[enduse,tech,ec,area,vintage] for vintage in Vintages)
@. DERBefore = DER
```

Then the level of reduction is calculated and applied. Reductions from process stock retirements are removed directly via using average device efficiency.

```
@. DERRRExo = DER*DERReduction
DER[enduse,tech,ec,area] =
max(0,(DER[enduse,tech,ec,area]-DERRRExo[enduse,tech,ec,area]-
PERRRExo[enduse,tech,ec,area]/DEEAPrior[enduse,tech,ec,area]))
```

Finally, the reduction is allocated to the stock as a percentage adjustment using the before and after total stock value.

```
DERV[enduse,tech,ec,area,vintage] = DERV[enduse,tech,ec,area,vintage]*
DER[enduse,tech,ec,area]/DERBefore[enduse,tech,ec,area]
```

where:

DER ‘Device Energy Requirement (mmBtu/YR)’ [Enduse,Tech,EC,Area]
DERRRExo ‘Device Energy Exogenous Retrofits ((mmBtu/Yr)/Yr)’ [Enduse,Tech,EC,Area]
DERReduction ‘Device Energy Reduction Fraction ((mmBtu/Yr)/(mmBtu/Yr))’ [Enduse,Tech,EC,Area]
PERRRExo ‘Process Energy Exogenous Retrofits ((mmBtu/Yr)/Yr)’ [Enduse,Tech,EC,Area]

Once stocks are finalized, the average stock market share (AMSF) for each technology in a given sector, area, and enduse is calculated as a function of the productive capacity stock (EUPC).

```
AMSF[enduse,tech,ec,area] = sum(EUPC[enduse,tech,age,ec,area] for age in Ages)/
sum(EUPC[enduse,alltechs,age,ec,area] for alltechs in Techs,age in Ages)
```

where:

AMSF 'Average Market Share ($/$)' [Enduse,Tech,EC,Area]

Finally, variables that accumulate stock across vintage or age are updated

```
DERVSum[enduse,tech,ec,area] = sum(DERV[enduse,tech,ec,area,vintage] for vintage in Vintages)
for vintage in Vintages
  DERVAllocation[enduse,tech,ec,area,vintage] = DERV[enduse,tech,ec,area,vintage] /
  DER[enduse,tech,ec,area]
end
…
PCEU[enduse,tech,ec,area] = sum(EUPC[enduse,tech,age,ec,area] for age in Ages)
```

where:

DERVAllocation ‘Fraction of DER in each Vintage (mmBtu/YR)’ [Enduse,Tech,EC,Area,Vintage]
DERVSum ‘Sum of Energy Requirement by Vintage (mmBtu/YR)’ [Enduse,Tech,EC,Area]
PCEU ‘Production Capacity (Driver/Yr)’ [Enduse,Tech,EC,Area]

F. Fuel Market Shares Functions

Function MShare: Marginal Fuel Market Shares

This function sets the marginal market share (MMSF) that allocates new productive capacity and conversions into technologies by using price and non-price factors from the model calibration.

Endogenous Market Share Determination

ENERGY 2100 uses standard market share concepts to allocate fuels to specific end uses. Not all investment funds are allocated to the least expensive energy form. Uncertainty, regional variations and limited knowledge make the perceived price on which choices are based, a distribution. The investments allocated to any fuel type are then proportional to the fraction of times one fuel is perceived as less expensive than all others.

A market allocation weight (MAW) is computed for each fuel (Tech) in each end-use (Enduse) in each economic class (EC). Summing over the fuels, a total allocation weight by (EC) and (Enduse) is calculated. The ratio of the MAW/TAW yields the market share fraction by device for each economic class (MMSF).

In the most basic terms, market share (MMSF) is determined by price [more specifically, the increase in the efficiency-adjusted real prices: (MCFU/INFLA/PEE)/ (MCFU0/INFLA0/PEE0)] and a variance factor on perceived prices (MVF). Other adjustments to market share come from non-price factors (MMSM0), income induced buying pattern changes (MMSMI, modified by population and capital output variables), and a market potential multiplier (MSMM) that captures non-price determinants such as color or style. The natural log form of MSMM implies that it is a direct multiplier on MAW (for example, if MSMM=0.5, then the value of MAW is MAW*0.5).

First a constraint specific for Storage technologies is applied before we begin the market share equations.

```
 if Tech[tech] == "Storage"
  MSLimit[enduse,tech,ec,area] = ThermalLimit[area]
else
  MSLimit[enduse,tech,ec,area] = 1.00
```

where:

MSLimit ‘Supply Limit on Market Share (Btu/Btu)’ [Enduse,Tech,EC,Area]

ThermalLimit ‘Thermal Limit (Btu/Btu)’ [Area]

Then we determine each fuel market allocation weight (MAW) for each EC:

```
MAW[enduse,tech,ec,area] = exp(MMSM0[enduse,tech,ec,area]+log(MSMM[enduse,tech,ec,area]) +
MMSMI[enduse,tech,ec,area]*(SPC[ec,area]/SPop[ec,area]) /  (SPC0[ec,area]/SPop0[ec,area]) +
MVF[enduse,tech,ec,area]*log((MCFU[enduse,tech,ec,area]/Inflation[area]/PEE[enduse,tech,ec,area])/
(MCFU0[enduse,tech,ec,area]/Inflation0[area]/PEE0[enduse,tech,ec,area])))*
MSLimit[enduse,tech,ec,area]
```

where:

MAW ‘Marginal Allocation Weight ($/$) [Enduse,Tech,EC,Area]

MMSM0 ‘Non-Price Factors. ($/$)’ [Enduse,Tech,EC,Area]

MSMM ‘Non-Price Market Share Factor Mult.’ [Enduse,Tech,EC,Area]

MMSMI ‘Market Share Mult. from Income ($/$)’ [Enduse,Tech,EC,Area]

SPC ‘Total Production Capacity (M$/Yr)’ [EC,Area]

SPC0 ‘Total Production Capacity in First Year(M$/Yr)’ [EC,Area]

SPop ‘Population (Millions)’ [EC,Area]

SPop0 ‘Population in First Year (Millions)’ [EC,Area]

MVF ‘Market Share Variance Factor ($/$)’ [Enduse,Tech,EC,Area]
MCFU ‘Marginal Cost of Fuel Use ($/mmBtu)’ [Enduse,Tech,EC,Area]
MCFU ‘Marginal Cost of Fuel Use in First Year ($/mmBtu)’ [Enduse,Tech,EC,Area]
PEE 'Process Efficiency ($/Btu)’ [Enduse,Tech,EC,Area]
PEE0 'Process Efficiency in First Year ($/Btu)’ [Enduse,Tech,EC,Area]

The total market allocation weight for each EC is the individual MAWs summed over fuel:

```
TMAW[enduse,ec,area] = sum(MAW[enduse,tech,ec,area] for tech in Techs)
```

The marginal market share fraction for each fuel for each EC is the ratio of the MAW of the particular fuel to the total (TMAW):

```
MMSF[enduse,tech,ec,area] = MAW[enduse,tech,ec,area]/TMAW[enduse,ec,area]
```

where:

MMSF ‘Market Share Fraction by Device ($/$)’ [Enduse,Tech,EC,Area]
TMAW ‘Total Allocation. Weight ($/$)’ [Enduse,EC,Area]

An additional adjustment can be added to account for emerging technologies values in a policy case compared to a base case. The equation below only has impact if ETSwitch is activated:

```
MMSF[enduse,tech,ec,area] = MMSF[enduse,tech,ec,area]+
max(MMSF[enduse,tech,ec,area]-MMSFB[enduse,tech,ec,area],0)*ETSwitch[tech,area]
```

where:

MMSFB ‘Market Share Fraction from Base Case ($/$)’ [Enduse,Tech,EC,Area]
ETSwitch ‘Emerging Technology Switch (1=Emerging Technology)’ [Tech,Area]

Exogenous Market Share Determination

The output market share value can also be set directly at the technology by setting the MShare ProcSw to ‘Exogenous’ and activating the MMSFSwitch. First, the exogenous market shares are collected and an exogenous total is calculated:

```
techs = findall(MMSFSwitch[enduse,:,ec,area] .== 0)
if !isempty(techs)
  for tech in techs
    MMSF[enduse,tech,ec,area] = MMSFExogenous[enduse,tech,ec,area]
    MMSFExoTotal = sum(MMSF[enduse,tech,ec,area] for tech in techs)
```

where:

MMSFSwitch ‘Market Share Switch (1=Endogenous, 0=Exogenous)’ [Enduse,Tech,EC,Area]
MMSFExogenous ‘Exogenous Market Share Fraction by Device ($/$)’ [Enduse,Tech,EC,Area]

Then the exogenous values are scaled if their sum is greater than 1.0

```
if MMSFExoTotal > 1.0
  MMSF[enduse,tech,ec,area] = MMSF[enduse,tech,ec,area]/MMSFExoTotal
  MMSFExoTotal = 1.0
end
```

Finally, the technologies that aren't exogenous are scaled relative to the exogenous values

```
techs = findall(MMSFSwitch[enduse,:,ec,area] .== 1)
if !isempty(techs)
  MMSFEndoTotal = sum(MMSF[enduse,tech,ec,area] for tech in techs)
  for tech in techs
    MMSF[enduse,tech,ec,area] = MMSF[enduse,tech,ec,area]/MMSFEndoTotal*(1-MMSFExoTotal)
  end
end
```

Function MarketShareAC: Air Conditioning Market Shares

Air conditioning has specific code designed to account for the dual heating and cooling outputs for heat pumps, geothermal, and similar devices, which often isn't easily captured in historical input data. This function sets the marginal market share (MMSF) for these technologies for the 'AC' enduse. Note that the code below is not executed if the demand segment does not contain an enduse for air conditioning.

First the technologies that have both heating and cooling output are selected

```
GeoHeatPump = Select(Tech,["Geothermal","HeatPump","DualHPump","FuelCell"])
```

For these technologies, the market share in the AC enduse are a function of production capacity in the Heat enduse in the prior year and current year capacity additions.

```
MMSF[AC,tech,ec,area] =sum((EUPCPrior[Heat,tech,age,ec,area]-EUPCPrior[AC,tech,age,ec,area])
for age in Ages)/PCA[New,ecc,area]*0.6
MMSF[AC,tech,ec,area] = max(MMSF[AC,tech,ec,area],0.0)
```

where:

EUPCPrior 'Production Capacity by Enduse (Driver/Yr)' [Enduse,Tech,EC,Area]
PCA 'Production Capacity Additions (M$/Yr/Yr)' [Age,ECC,Area]

The sum of market shares by these technologies are accumulated for use in scaling other market shares. The total value is constrained to be at or below 1.0. Market shares of other devices are accumulated as 'MMSFOther'. These are individually scaled using their existing value combined with the values from the heat pump technologies

```
MMSFHPs[ec,area] = min(sum(MMSF[AC,tech,ec,area] for tech in GeoHeatPump),1.00)
…
Other1 = Select(Tech,!=("Geothermal"))
Other2 = Select(Tech,!=("HeatPump"))
Other3 = Select(Tech,!=("DualHPump"))
Other4 = Select(Tech,!=("FuelCell"))
Other = intersect(Other1,Other2,Other3,Other4)
for ec in ECs,area in Areas
  MMSFOther[ec,area] = sum(MMSF[AC,tech,ec,area] for tech in Other)
```

```
end
for tech in Other,ec in ECs,area in Areas
  MMSF[AC,tech,ec,area] = MMSF[AC,tech,ec,area]/MMSFOther[ec,area]*(1-MMSFHPs[ec,area])
end
```

Finally, the new market shares for all technologies are scaled equally so that the sum (MMSFAll) is equal to 1.0

```
MMSFAll[ec,area] = sum(MMSF[AC,tech,ec,area] for tech in Techs)
..
MMSF[AC,tech,ec,area] = MMSF[AC,tech,ec,area]/MMSFAll[ec,area]
```

Function Fungible: Fuel/Tech Fractions

This function sets the fuel/tech market share (DmFrac) that determines the fuel used by technologies by using fuel price and non-price factors from the model calibration.

The methodology and code for determining the fuel fraction shares many characteristics with marginal fuel market share (MMSF) calculations from the MShare function. A fuel fraction non-price factor (DmFracMSM0) is estimated from historical data during the model calibration, which is combined with a price factors and a variance factor to estimate the fuel share. However, the fungible equations differ in two ways:

1. The output fraction in constrained by a floor (DmFracMin) and a ceiling (DmFracMax), generally set during the model calibration using an assumption of realistic fuels that each technology could use.
2. The final fuel fraction is iterated with the floor and ceiling to account for differing constraints across fuel.

Endogenous Fuel Fraction Determination

First, we determine each fuel's fraction market allocation weight (DmFracMAW) for each technology:

```
DmFracMAW[enduse,fuel,tech,ec,area] = exp(DmFracMSM0[enduse,fuel,tech,ec,area]+

DmFracVF[enduse,fuel,tech,ec,area]*log((ECFPFuel[fuel,ec,area]/Inflation[area])/

(ECFP0[enduse,tech,ec,area]/Inflation0[area])))
```

where:

DmFracMAW 'Allocation Weights for Demand Fuel/Tech Fraction (DLess)' [Enduse,Fuel,Tech,EC,Area]
DmFracMSM0 'Demand Fuel/Tech Fraction Non-Price Factor (Btu/Btu)' [Enduse,Fuel,Tech,EC,Area]
DmFracVF 'Demand Fuel/Tech Fraction Variance Factor (Btu/Btu)' [Enduse,Fuel,Tech,EC,Area]
ECFPFuel 'Fuel Price ($/mmBtu)' [Enduse,Tech,EC,Area]
ECFPFuel0 'Fuel Price in First Year ($/mmBtu)' [Enduse,Tech,EC,Area]

The total market allocation weight for each technology is the individual DmFracMAWs summed over fuel:

```
DmFracTMAW[enduse,tech,ec,area] = sum(DmFracMAW[enduse,fuel,tech,ec,area] for fuel in Fuels)
```

The marginal market share fraction for each fuel for each EC is the ratio of the MAW of the particular fuel to the total (TMAW):

```
DmFracMSF[enduse,fuel,tech,ec,area]=
        DmFracMAW[enduse,fuel,tech,ec,area]/DmFracTMAW[enduse,tech,ec,area]
```

where:

DmFracMSF 'Demand Fuel/Tech Fraction Market Share (Btu/Btu)' [Enduse,Fuel,Tech,EC,Area]
DmFracTMAW 'Total Allocation. Weight ($/$)' [Enduse,Tech,EC,Area]

Applying Minimums and Maximums

The fuel share is constrained inside a while loop for 10 iterations. Limits are applied at the fuel level, then the sum is used to allocated the market share across fuels in each iteration.

```
@. DmFracMarginal = DmFracMSF
DmFracCount::Int = 1

while DmFracCount < 10
 @. DmFracMarginal = min(max(DmFracMarginal,DmFracMin),DmFracMax)
 for area in Areas,ec in ECs,tech in Techs,fuel in Fuels,enduse in Enduses
  DmFracMarginal[enduse,fuel,tech,ec,area]
        =DmFracMarginal[enduse,fuel,tech,ec,area]*FuelLimit[fuel,area]
  end
  for area in Areas,ec in ECs,tech in Techs,enduse in Enduses
   DmFracTotal[enduse,tech,ec,area] = sum(DmFracMarginal[enduse,fuel,tech,ec,area] for fuel in Fuels)
  end
  for area in Areas,ec in ECs,tech in Techs,fuel in Fuels,enduse in Enduses
   DmFracMarginal[enduse,fuel,tech,ec,area] =
   DmFracMarginal[enduse,fuel,tech,ec,area]/DmFracTotal[enduse,tech,ec,area]
  end
  DmFracCount += 1
 end
```

where:

DmFracMarginal 'Demand Fuel/Tech Fraction Marginal Market Share (Btu/Btu)'
[Enduse,Fuel,Tech,EC,Area]
DmFracMin 'Demand Fuel/Tech Fraction Minimum (Btu/Btu) [Enduse,Fuel,Tech,EC,Area]'
DmFracMax 'Demand Fuel/Tech Fraction Maximum (Btu/Btu)' [Enduse,Fuel,Tech,EC,Area]
FuelLimit 'Fuel Limit Multiplier (Btu/Btu)' [Fuel,Area]

Then, the fraction is calculated for each fuel using the output of the loop combined with the value from the prior year. The rate of impact of the curve output can be adjusted by modifying the input adjustment time variable (DmFracTime)

```
@. DmFrac = DmFracPrior+(DmFracMarginal-DmFracPrior)/DmFracTime
```

where:

DmFrac 'Demand Fuel/Tech Fraction Split (Btu/Btu) [Enduse,Fuel,Tech,EC,Area]
DmFracPrior 'Demand Fuel/Tech Fraction Split in Prior Year (Btu/Btu) [Enduse,Fuel,Tech,EC,Area]
DmFracTime 'Demand Fuel/Tech Adjustment Time (Years)' [Enduse,Fuel,Tech,EC,Area]

Finally, the DmFrac value is adjusted again using the total values to weight the overall fraction by Technology to 1.0.

```
DmFracTotal = 0.0

for area in Areas,ec in ECs,tech in Techs,fuel in Fuels,enduse in Enduses
  DmFracTotal[enduse,tech,ec,area] = DmFrac[enduse,fuel,tech,ec,area]+
  DmFracTotal[enduse,tech,ec,area]
end

for area in Areas,ec in ECs,tech in Techs,fuel in Fuels,enduse in Enduses
  DmFrac[enduse,fuel,tech,ec,area] =
  DmFrac[enduse,fuel,tech,ec,area]/(DmFracTotal[enduse,tech,ec,area])
End
```

Function FeedstockFungible: Feedstock Fuel/Tech Fractions

This function sets the fuel/tech market share (FsFrac) that determines the fuel type used for feedstock demands. This function shares the methodology as the demand fraction in Fungible. The only exception is the usage of the ceiling variable (FsFracMax) in the final weighting equation to allow for a fraction below 1.0 as a policy lever.

Endogenous Fuel Fraction Determination

First, we determine the market allocation weight (FsFracMAW) for each feedstock:

```
FsFracMAW[fuel,tech,ec,area] = exp(FsFracMSM0[fuel,tech,ec,area]+

FsFracVF[fuel,tech,ec,area]*log(FsFP[fuel,es,area]/FsFP0[fuel,es,area]))
```

where:

FsFracMAW 'Allocation Weights for Feedstock Fuel/Tech Fraction (DLess)' [Fuel,Tech,EC,Area]

FsFracMSM0 'Feedstock Fuel/Tech Fraction Non-Price Factor (Btu/Btu)' [Fuel,Tech,EC,Area]

FsFracVF 'Feedstock Fuel/Tech Fraction Variance Factor (Btu/Btu)' [Fuel,Tech,EC,Area]

FsFP 'Feedstock Fuel Price ($/mmBtu)' [Fuel,ES,Area]

FsFP0 'Feedstock Fuel Price in First Year ($/mmBtu)' [Fuel,ES,Area]

The total market allocation weight for each technology is the individual FsFracMAWs summed over fuel:

```
FsFracTMAW[tech,ec,area] = sum(FsFracMAW[fuel,tech,ec,area] for fuel in Fuels)
```

The marginal market share fraction for each fuel is the ratio of the MAW of the particular fuel to the total (TMAW):

```
FsFracMSF[fuel,tech,ec,area]= FsFracMAW[fuel,tech,ec,area]/FsFracTMAW[tech,ec,area]
```

where:

FsFracMSF 'Feedstock Fuel/Tech Fraction Market Share (Btu/Btu)' [Fuel,Tech,EC,Area]

FsFracTMAW 'Total Allocation. Weight ($/$)' [Tech,EC,Area]

Applying Minimums and Maximums

The fuel share is constrained inside a while loop for 10 iterations. Limits are applied at the fuel level, then the sum is used to allocated the market share across fuels in each iteration.

```
@. FsFracMarginal = FsFracMSF
FsFracCount::Int = 1

while FsFracCount < 10
  @. FsFracMarginal = min(max(FsFracMarginal,FsFracMin),FsFracMax)
  for area in Areas,ec in ECs,tech in Techs,fuel in Fuels
    FsFracMarginal[fuel,tech,ec,area] = FsFracMarginal[fuel,tech,ec,area]*FuelLimit[fuel,area]
  end
  for area in Areas,ec in ECs,tech in Techs,fuel in Fuels
    FsFracTotal[tech,ec,area] = sum(FsFracMarginal[fuel,tech,ec,area] for fuel in Fuels)
  end
  for area in Areas,ec in ECs,tech in Techs,fuel in Fuels
    FsFracMarginal[fuel,tech,ec,area] =   FsFracMarginal[fuel,tech,ec,area]/FsFracTotal[tech,ec,area]
  end
  FsFracCount += 1
end
```

where:

FsFracMarginal 'Feedstock Fuel/Tech Fraction Marginal Market Share (Btu/Btu)' [Fuel,Tech,EC,Area]
FsFracMin 'Feedstock Fuel/Tech Fraction Minimum (Btu/Btu) [Fuel,Tech,EC,Area]'
FsFracMax 'Feedstock Fuel/Tech Fraction Maximum (Btu/Btu)' [Fuel,Tech,EC,Area]
FuelLimit 'Fuel Limit Multiplier (Btu/Btu)' [Fuel,Area]

Then, the fraction is calculated for each fuel using the output of the loop combined with the value from the prior year. The rate of impact of the curve output can be adjusted by modifying the input adjustment time variable (FsFracTime)

```
 @. FsFrac = FsFracPrior+(FsFracMarginal-FsFracPrior)/FsFracTime
```

where:

FsFrac 'Feedstock Fuel/Tech Fraction Split (Btu/Btu) [Fuel,Tech,EC,Area]
FsFracPrior 'Feedstock Fuel/Tech Fraction Split in Prior Year (Btu/Btu) [Fuel,Tech,EC,Area]
FsFracTime 'Feedstock Fuel/Tech Adjustment Time (Years)' [Fuel,Tech,EC,Area]

Finally, the FsFrac value is adjusted again using the total values to weight the overall fraction by Technology to 1.0. This is constrained by the user input FsFracMax value, which can limit the total to be below 1.0 to simulate reductions in policies.

```
for area in Areas,ec in ECs,tech in Techs
  FsFracTotal[tech,ec,area] = sum(FsFrac[fuel,tech,ec,area] for fuel in Fuels)
end

for area in Areas,ec in ECs,tech in Techs,fuel in Fuels
  FsFrac[fuel,tech,ec,area] =
  min(FsFrac[fuel,tech,ec,area]/FsFracTotal[tech,ec,area],FsFracMax[fuel,tech,ec,area])
end
```

G. Retrofits and Conversions Functions

Function InitRetrofits: Initialize Retrofit Variables

ENERGY 2100 allows users the option to implement retrofitting policies, where existing energy requirements operating at the stock level of energy efficiency and cost can be replaced by new requirements at a higher level of efficiency. Retrofitting can be used to improve efficiency for existing stocks during the middle of their lifespan instead of waiting for stocks to retire and be replaced by higher efficiency devices.

This function calculates the retrofit device and capital charge rates for use in the main Retrofit function. These equations are similar to the device and process capital charge rate methodology. The equation below is used for the retrofit device capital charge rate.

```
RDCCR[enduse,tech,ec,area] = (1-(DIVTC[tech,area]+DRIVTC[tech,area]) /
(1+ROIN[ec,area]-RCROIN[enduse,tech,ec,area]+RRisk[enduse,tech,area]+InSm[area]) - TxRt[ec,area]
        *(2/DTL[enduse,tech,ec,area])/(ROIN[ec,area]-RCROIN[enduse,tech,ec,area]+
        RRisk[enduse,tech,area]+
InSm[area]+2/DTL[enduse,tech,ec,area]))*(ROIN[ec,area] -RCROIN[enduse,tech,ec,area]
+RRisk[enduse,tech,area]) /
(1-(1/(1+ROIN[ec,area]-RCROIN[enduse,tech,ec,area]+
RRisk[enduse,tech,area]))^DPL[enduse,tech,ec,area])/ (1-TxRt[ec,area])
```

where:

RDCCR 'Device Retrofit Capital Charge Rate ($/Yr/$)' [Enduse,Tech,EC,Area]
DIVTC 'Device Investment Tax Credit ($/$)' [Tech,Area]
DRIVTC 'Device Retrofit Investment Tax Credit ($/$)' [Tech,Area]
InSm 'Smoothed Inflation Rate ($/Yr/$)' [Area]
TxRt 'Tax Rate on Energy Consumer ($/$)' [EC,Area]
DTL 'Device Tax Life (Years)' [Enduse,Tech,EC,Area]
ROIN 'Return on Investment ($/Yr/$)' [EC,Area]
RCROIN 'Retrofit Conservation Subsidy on Return on Investment ($/Yr/$)' [Enduse,Tech,EC,Area]
RRISK 'Retrofit Excess Risk ($/$)' [Enduse,Tech,EC,Area]

DPL 'Physical Life of Equipment (Years)' [Enduse,Tech,EC,Area]

The retrofit process capital charge rate is calculated in a similar manner as the process capital charge rate. Note that value is calculated using 50% of the lifespan value (PEPL/2).

```
RPCCR[enduse,tech,ec,area] = (1-(PIVTC[area]+RPIVTC[area]) / (1+ROIN[ec,area]
        RCROIN[enduse,tech,ec,area]+InSm[area])-TxRt[ec,area]*(4/PETL[enduse,tech,ec,area]) /
(ROIN[ec,area]+0+InSm[area]+4/PETL[enduse,tech,ec,area]))*(ROIN[ec,area] -
 RCROIN[enduse,tech,ec,area]) /(1-(1/(1+ROIN[ec,area])) ^(PEPL[enduse,tech,ec,area]/2))/(1-
        TxRt[ec,area])
```

where:

RPCCR 'Process Retrofit Capital Charge Rate ($/Yr/$)' [Enduse,Tech,EC,Area]
PIVTC 'Process Policy Investment Tax Credit ($/$)' [Area]
RPIVTC 'Process Retrofit Policy Investment Tax Credit ($/$)' [Area]
PETL 'Tax Life of Process Requirements (Years)' [Enduse,Tech,EC,Area]
PEPL 'Physical Life of Process Requirements (Years)' [Enduse,Tech,EC,Area]

The model code allows for setting the retrofit capital charge rate equations equal to a base case value if requested by the user.

Function Retrofit: Retrofit Efficiency and Capital Costs

Retrofits are applied to device and process efficiencies dependent on policy input variables and an execution switch (RetroSw) specified by the user. Retrofits can be applied to devices (RetroSw = 1), process (RetroSw = 2), both (RetroSw = 3), or be exogenously specified (RetroSw = 0). The default switch value in the current model is -99, which instructs the model to bypass the code in this function.

Device Retrofits

The following code is executed if RetroSw is equal to 1 or 3 for any given enduse, sector, or area. The average unit cost of the device stock (ADCC) is estimated using the capital cost curve and the average level of efficiency in the prior year (DEEAPrior)

```
ADCC[enduse,tech,ec,area] = DCCN[enduse,tech,ec,area]*DCMM[enduse,tech,ec,area]*
DCMMM[enduse,tech,ec,area]*Inflation[area]*(DEM[enduse,tech,ec,area] *
DEMM[enduse,tech,ec,area]*DEMMM[enduse,tech,ec,area] /
DEEAPrior[enduse,tech,ec,area]-1)^(1/min(DCTC[enduse,tech,ec,area],-0.01))
```

where:

ADCC 'Average Device Capital Cost ($/mmBtu/Yr)' [Enduse,Tech,EC,Area]
DCCN 'Device Capital Cost ($/mmBtu)' [Enduse,Tech,EC,Area]
DCMM 'Device Cost Efficiency Multiplier ($/$)' [Enduse,Tech,EC,Area]
DCMMM 'Device Cost Efficiency Multiplier for Efficiency Program ($/$)' [Enduse,Tech,EC,Area]

DEM 'Maximum Device Efficiency (Btu/Btu)' [Enduse,Tech,EC,Area]
DEMM 'Maximum Device Efficiency Multiplier (Btu/Btu)' [Enduse,Tech,EC,Area]
DEMMM 'Device Efficiency Multiplier for Efficiency Program (Btu/Btu)' [Enduse,Tech,EC,Area]
DEEAPrior 'Average Device Efficiency in Prior Year (Btu/Btu)' [Enduse,Tech,EC,Area]
DCTC 'Device Cap. Trade Off Coefficient (DLESS) [Enduse,Tech,EC,Area]'

Marginal efficiency for retrofit devices (RDEE) is set equal to the input retrofit device efficiency standard (RDEStd) or the current marginal efficiency (DEE) times the user specified retrofit multiplier (RDEMM), whichever is higher

RDEE[enduse,tech,ec,area] = max(DEE[enduse,tech,ec,area]*RDEMM[enduse,tech,ec,area],
RDEStd[enduse,tech,ec,area])

where:

RDEE 'Retrofit Device Efficiency (Btu/Btu)' [Enduse,Tech,EC,Area]
DEE 'Device Efficiency (Btu/Btu)' [Enduse,Tech,EC,Area]
RDEMM 'Retrofit Maximum Device Efficiency Multiplier (Btu/Btu)' [Enduse,Tech,EC,Area]
RDEStd 'Retrofit Device Efficiency Standards (Btu/Btu)' [Enduse,Tech,EC,Area]

Retrofit device capital costs (RDCC) are a function of device capital costs and the user specified retrofit device capital cost multiplier (RDCCM).

RDCC[enduse,tech,ec,area] = DCCPrior[enduse,tech,ec,area]*RDCCM[enduse,tech,ec,area]

where:

RDCC 'Retrofit Device Capital Cost ($/($/Yr))' [Enduse,Tech,EC,Area]
DCC 'Device Capital Cost ($/mmBtu/Yr)' [Enduse,Tech,EC,Area]
RDCCM 'Retrofit Device Capital Cost Multiplier ($/$)' [Enduse,Tech,EC,Area]

A marginal value of device retrofits (MVDR) is calculated based on the relationship between stock device efficiency (DEEA) and the retrofit efficiency (RDEE) when including the retrofitting cost factors.

MVDR[enduse,tech,ec,area] = (ECFP[enduse,tech,ec,area] *
(1/DEEAPrior[enduse,tech,ec,area]-1/RDEE[enduse,tech,ec,area]))/
RDCC[enduse,tech,ec,area]-RRisk[enduse,tech,area]

where:

MVDR 'Marginal Value of Device Retrofits ($/$)' [Enduse,Tech,EC,Area]
ECFP 'Fuel Price ($/mmBtu)' [Enduse,Tech,EC,Area]
RRisk 'Retrofit Excess Risk ($/$)' [Enduse,Tech,Area]

The retrofit market share fraction per device (RDMSF) is based on the marginal value of retrofits, return on investment, and user input retrofit parameters.

RDMSF[enduse,tech,ec,area] = max(0,

MVDR[enduse,tech,ec,area])^(-RDVF[enduse,tech,ec,area])*
RDMSM[enduse,tech,ec,area]/(max(0,MVDR[enduse,tech,ec,area])^
(-RDVF[enduse,tech,ec,area])*RDMSM[enduse,tech,ec,area]+(ROIN[ec,area]-
RCROIN[enduse,tech,ec,area])^(-RDVF[enduse,tech,ec,area]))

where:

RDMSF 'Device Retrofit Market Share Fraction by Device (1/Yr)' [Enduse,Tech,EC,Area]
RDVF 'Device Retrofit Market Share Variance Factor (DLESS) [Enduse,Tech,EC,Area]
RDMSM 'Device Retrofit Market Share Multiplier (1/Yr)' [Enduse,Tech,EC,Area]
ROIN 'Return on Investment ($/Yr/$)' [EC,Area]
RCROIN 'Retrofit Conservation Subsidy on Return on Investment ($/Yr/$)' [Enduse,Tech,EC,Area]

Process Retrofits

The following code is executed if RetroSw is equal to 2 or 3 for any given enduse, sector, or area. The average unit cost of the process stock (APCC) is estimated using the capital cost curve and the average level of efficiency in the prior year (PEEAPrior). The value is set for just the first enduse in the segment to be consistent with the process capital cost methodology

APCC[enduse,tech,ec,area] = PCCN[enduse,tech,ec,area]*PCMM[enduse,tech,ec,area]*
PCMMM[enduse,tech,ec,area]*Inflation[area]*(PEM[enduse,tech,ec,area] *
PEMM[enduse,tech,ec,area]*PEMMM[enduse,tech,ec,area] /
PEEAPrior[enduse,tech,ec,area]-1)^(1/min(PCTC[enduse,tech,ec,area],-0.01))

where:

APCC 'Average Process Capital Cost ($/($/Yr))' [Enduse,Tech,EC,Area]
PCCN 'Normalized Process Capital Cost ($/($/Yr))' [Enduse,Tech,EC,Area]
PCMM 'Process Cost Efficiency Multiplier ($/$)' [Enduse,Tech,EC,Area]
PCMMM 'Process Cost Efficiency Multiplier for Efficiency Program ($/$)' [Enduse,Tech,EC,Area]
PEM 'Maximum Process Efficiency ($/Btu)' [Enduse,Tech,EC,Area]
PEMM 'Maximum Process Efficiency Multiplier (($/Btu/($/Btu))' [Enduse,Tech,EC,Area]
PEMMM 'Process Efficiency Multiplier for Efficiency Program (($/Btu/($/Btu))' [Enduse,Tech,EC,Area]
PEEAPrior 'Average Process Efficiency in Prior Year ($/Btu)' [Enduse,Tech,EC,Area]
PCTC 'Process Capital Trade Off Coefficient (DLESS)' [Enduse,Tech,EC,Area]

Marginal efficiency for retrofit processes (RPEE) is set equal to the current marginal efficiency, average efficiency in the prior year (PEEAPrior) times the user specified retrofit multiplier (RPEMM), or the process retrofit standard, whichever is higher. The RPEEFr user input variable serves to weight the stock and marginal efficiency values.

RPEE[enduse,tech,ec,area] = max((PEE[enduse,tech,ec,area]*
RPEEFr[enduse,tech,ec,area]+PEEAPrior[enduse,tech,ec,area]*
(1-RPEEFr[enduse,tech,ec,area]))*RPEMM[enduse,tech,ec,area],
RPEStd[enduse,tech,ec,area])

where:

RPEE 'Retrofit Device Efficiency ($/Btu/($/Btu)' [Enduse,Tech,EC,Area]

RPEEFr 'Process Efficiency Fraction for Retrofits ($/Btu/($/Btu))' [Enduse,Tech,EC,Area]
RPEMM 'Process Efficiency Max. Mult. ($/Btu/($/Btu))' [Enduse,Tech,EC,Area]
RPEStd 'Retrofit Process Efficiency Standards ($/Btu/($/Btu))' [Enduse,Tech,EC,Area]

Capital costs for retrofitted processes (RPCC) are a function of process capital costs and the user specified retrofit process capital cost multiplier

@. RPCC=PCC*RPCCM

where:

RPCC 'Process Retrofit Capital Cost ($/($/Yr))' [Enduse,Tech,EC,Area]
PCC 'Process Capital Cost ($/($/Yr))' [Enduse,Tech,EC,Area]
RPCCM 'Retrofit Process Capital Cost Multiplier ($/$)' [Enduse,Tech,EC,Area]

A marginal value of process retrofitting (MVPR) is calculated based on the relationship between stock process efficiency (PEEA) and the retrofit efficiency (RPEE) when including the retrofitting cost factors.

MVPR[enduse,tech,ec,area] = (ECFP[enduse,tech,ec,area]/
DEEAPrior[enduse,tech,ec,area]*(1/PEEAPrior[enduse,tech,ec,area]-1/
RPEE[enduse,tech,ec,area])-(RPCCR[enduse,tech,ec,area]+
POCF[enduse,tech,ec,area]+RHCM[ec,area])*RPCC[enduse,tech,ec,area])/
RPCC[enduse,tech,ec,area]-RRisk[enduse,tech,area]

where:

MVPR 'Marginal Value of Process Retrofits ($/$)' [Enduse,Tech,EC,Area]
ECFP 'Fuel Price ($/mmBtu)' [Enduse,Tech,EC,Area]
RPCCR 'Process Retrofit Capital Charge Rate ($/Yr/$)' [Enduse,Tech,EC,Area]
POCF 'Process Operating Cost Fraction ($/Yr/$)' [Enduse,Tech,EC,Area]
RRisk 'Retrofit Excess Risk ($/$)' [Enduse,Tech,Area]

POCF(Enduse,Tech,EC,Area) 'Process Operating Cost Fraction ($/Yr/$)'

The retrofit market share fraction per process (RPMSF) is based on the relationship between stock and retrofit efficiency adjusted by the user specified device retrofit market share multiplier (RDMSM) and a maximum bound (RPMSLimit).

RPMSF[enduse,tech,ec,area] = min(max(0,(1/(1+(PEEAPrior[enduse,tech,ec,area]/
RPEE[enduse,tech,ec,area]))-0.5))/2*RPMSM[enduse,tech,ec,area],RPMSLimit[ec,area])

where:

RPMSF 'Process Retrofit Market Share Fraction by Device (1/Yr) '[Enduse,Tech,EC,Area]
RPMSM 'Process Retrofit Market Share Multiplier (1/Yr)' [Enduse,Tech,EC,Area]
RPMSLimit 'Process Retrofit Market Share Limit (1/Yr)' [EC,Area]

Similar to the process efficiency calculation code, due to their linked nature an exception is made for air conditioning retrofitting efficiency to be set based on the retrofitting efficiencies for space heat.

```
RPEE[AC,tech,ec,area] =
sum(RPEE[Heat,tech,ec,area]*PERPrior[Heat,tech,ec,area] for tech in Techs)/
sum(PERPrior[Heat,tech,ec,area] for tech in Techs)/
CHR[ec,area]*CHRM[ec,area]*RPEMM[AC,tech,ec,area]

RPCC[AC,tech,ec,area] = RPCC[Heat,tech,ec,area]

PFS[AC,tech,ec,area] = ECFP[AC,tech,ec,area]/DEEAPrior[AC,tech,ec,area]*
(1/PEEAPrior[AC,tech,ec,area]-1/RPEE[AC,tech,ec,area])
```

where:

PER 'Process Energy Requirement (mmBtu/YR)' [Enduse,Tech,EC,Area]
RPEMM 'Process Efficiency Max. Mult. ($/Btu/($/Btu))' [Enduse,Tech,EC,Area]
CHR(EC,Area) 'Cooling to Heating Ratio (Btu/Btu)' [EC,Area]
CHRM(EC,Area) 'Cooling to Heating Ratio Multiplier' [EC,Area]
PFS 'Process Fuel Savings ($/Yr)' [Enduse,Tech,EC,Area]

Exogenous Retrofits

The following code is executed if RetroSw is equal to 0 for any given enduse, sector, or area. Retrofitting parameters are directly set equal to input values specified by the user as follows:

```
@. RPEE=xRPEE
@. RPCC=xRPCC
@. RPMSF=xRPMSF
@. RDEE=RDEStd
@. RDCC=DCC
@. RDMSF=xRDMSF
```

where:

xRPEE 'Exogenous Retrofit Process Efficiency ($/BTU)' [Enduse,Tech,EC,Area]
xRPCC 'Exogenous Process Retrofit Capital Cost ($/($/Yr))' [Enduse,Tech,EC,Area]
xRPMSF 'Exogenous Process Retrofit Market Share Fraction by Device (1/Yr)' [Enduse,Tech,EC,Area]
xRDMSF 'Exogenous Device Retrofit Market Share Fraction by Device (1/Yr)' [Enduse,Tech,EC,Area]

Retrofitting Stock Adjustments

Any enduses that contain retrofitting use the parameters set above to adjust energy requirement stocks and costs where relevant. Process requirements are reduced based on the retrofit process market share and level of efficiency based on the current capital stock level

```
FXCO[enduse,tech,ec,area] = sum(EUPCPrior[enduse,tech,age,ec,area] for age in Ages)-
EUPCRPC[enduse,tech,Old,ec,area]
```

PERRR[enduse,tech,ec,area] = RPMSF[enduse,tech,ec,area]* FXCO[enduse,tech,ec,area]*(1/PEEAPrior[enduse,tech,ec,area]-1/RPEE[enduse,tech,ec,area])

where:

FXCO 'Capital Output Capacity by Enduse (M$/Yr)' [Enduse,Tech,EC,Area]
EUPCPrior 'Production Capacity by Enduse in Prior Year(M$/Yr)' [Enduse,Tech,Age,EC,Area]
EUPCRPC 'Production Capacity Retirements from Capacity Retirements (M$/Yr/Yr)' [Enduse,Tech, Age,EC,Area]
PERRR 'Process Energy Retire. Process Retrofit ((mmBtu/Yr)/Yr)' [Enduse,Tech,EC,Area]

Device retrofitting retirements are based on the retrofit device market share and level of efficiency based on the current process requirements level.

DERRR[enduse,tech,ec,area] = RDMSF[enduse,tech,ec,area]* (PERPrior[enduse,tech,ec,area]-PERRPC[enduse,tech,ec,area])* (1/DEEAPrior[enduse,tech,ec,area]-1/RDEE[enduse,tech,ec,area])

where:

DERRR 'Device Energy Retire. Retrofit ((mmBtu/Yr)/Yr)' [Enduse,Tech,EC,Area]
PERRPC 'Process Retire. from Production Capacity Retire. (mmBtu/Yr/Yr)' [Enduse,Tech,EC,Area]

Function Conversion: Device Retirements and Fuel Conversions

A device conversion (for example changing from an electric to a natural gas water heater) is a fuel choice decision. The decision to replace with a device using a different fuel is based on the relative total cost of energy service (MCFU). Policies used in this routine represent rebates. Low interest loans would require additional calculations.

Market Share

If the choice is made endogenously, then it is a function of the cost of the new technology (MCFU) less the "hurdle" cost (FDCC). The hurdle cost for space heating represents the cost of ducts. These costs are "hurdle" costs in the sense that if change from electric baseboard heat to a gas furnace you must add duct work to the house.

In the most basic terms, conversion market share (CMSF) is determined by price [more specifically, the increase in the efficiency-adjusted real prices: (MCFU/INFLA/ (MCFU0/INFLA0] and a variance factor on perceived prices (CVF). Other adjustments to market share come from non-price factors (CMSM0) and income induced buying pattern changes, modified by population and capital output variables.

First determine each fuel market allocation weight (CMAW) for each EC:

CMAW[tech,ctech,ec,area] = exp(CMSM0[enduse,tech,ctech,ec,area] + CVF[enduse,tech,ctech,ec,area] *log(((MCFU[enduse,tech,ec,area]-DCCR[enduse,tech,ec,area]* (FDCC[enduse,tech,ctech,area]+FDCCU[enduse,tech,ctech,area])*

```
Inflation[area])/Inflation[area])/(MCFU0[enduse,tech,ec,area]/Inflation0[area]))+
CMSMI[enduse,tech,ctech,ec,area]* (SPC[ec,area]/SPop[ec,area])/(SPC0[ec,area]/SPop0[ec,area]))
```

The total market allocation weight for each EC is the individual CTMAWs summed over fuel:

```
CTMAW[ctech,ec,area] = sum(CMAW[tech,ctech,ec,area] for tech in Techs)
```

The marginal market share fraction for each fuel for each EC is the ratio of the CMAW of the particular fuel to the total (CTMAW):

```
CMSF[enduse,tech,ctech,ec,area] = CMAW[tech,ctech,ec,area]/CTMAW[ctech,ec,area]
```

where:

CMAW 'Conversion Alloc. Weight ($/$)' [Tech,CTech,EC,Area]
CTMAW 'Conversion Alloc. Weight ($/$)' [Tech,CTech,EC,Area]
CMSF 'Conversion Market Share Fraction by Device ($/$)' [Enduse,Tech,CTech,EC,Area]
CMSM0 'Conversion Market Share Multiplier ($/$)' [Enduse,Tech,CTech,EC,Area]
CMSMI 'Conversion Market Share Multiplier ($/$)' [Enduse,Tech,CTech,EC,Area]
MCFU 'Marginal Cost of Fuel Use ($/mmBtu)' [Enduse,Tech,EC,Area]
MCFU0 'Marginal Cost of Fuel Use ($/mmBtu) in First Year' [Enduse,Tech,EC,Area]
CVF 'Conversion Market Share Variance Factor (DLESS)' [Enduse,Tech,CTech,EC,Area]
Infla 'Inflation Index ($/$)'
Infla0 Inflation Index in First Year ($/$)
SPOP 'Population (Millions)' [EC,Area]
SPOP0 'Population in First Year (Millions)' [EC,Area]
SPC 'Total Production Capacity (M$/Yr)' [EC,Area]
SPC0 Production Capacity in First Year (M$/Yr)' [EC,Area]
FDCC 'Fixed Device Capital Cost ($/(MBTU/YR))' [Enduse,Tech,CTech,EC,Area]
FDCCU 'Conversion Rebate' [Enduse,Tech,CTech,EC,Area]
DCCR 'Device Capital Charge Rate ($/Yr/$)' Enduse,Tech,EC,Area]

The marginal market share fraction can also be specified exogenously by setting a value for xCMSF and activating CMSFSwitch. When exogenous market shares are active the remaining endogenous shares are scaled using the exogenous total:

```
CMSF[enduse,tech,ctech,ec,area] = xCMSF[enduse,tech,ctech,ec,area]
…
CMSFExoTotal = sum(CMSF[enduse,tech,ctech,ec,area] for tech in techs)
techs = findall(CMSFSwitch[enduse,:,ctech,ec,area] .== 1)
if !isempty(techs)
  CMSFEndoTotal = sum(CMSF[enduse,tech,ctech,ec,area] for tech in techs)
  for tech in techs
    CMSF[enduse,tech,ctech,ec,area] = CMSF[enduse,tech,ctech,ec,area]/CMSFEndoTotal*(1-
            CMSFExoTotal)
  end
end
```

where:

xCMSF 'Conversion Market Share by Device ($/$)' [Enduse,Tech,CTech,EC,Area]

CMSFSwitch ‘Conversion Market Share Switch (1=Endogenous, 0=Exogenous) [Enduse,Tech,CTech,EC,Area]

Function DeviceDynamics: Device Dynamics with Conversions

This function simulates the dynamics of device wear-out (DERRD) and device replacements (DERAD). When a device wears out, the device is replaced since the process energy requirements still exist. The new device has the current marginal device efficiency (DEE), while the old device is assumed to have a device efficiency equal to the average device efficiency (DEEA). The new device is normally the same technology or fuel type as the old device; however, if the "conversion" switch has been turned on, then the new device may select a different technology.

The device retirements or failures at the vintaging level (DERRDV) due to conversions are equal to the sum of device energy requirements (DERVSum) less production capacity retirements (DERRPC) and process retirements (DERRP) allocated into vintages (DERVAllocation) multiplied by the scrappage rate by vintage (DPLV). The total amount of retirements due to conversions is then summed across vintage (DERRC).

```
DERRRCV[enduse,tech,ec,area,vintage] = (DERVSum[enduse,tech,ec,area]-
DERRPC[enduse,tech,ec,area] -
          DERRP[enduse,tech,ec,area])*DERVAllocation[enduse,tech,ec,area,vintage]*
DPLV[enduse,tech,ec,area,vintage]*CFraction[enduse,tech,ec,area]
..
DERRRC[enduse,tech,ec,area] = sum(DERRRCV[enduse,tech,ec,area,vintage] for vintage in Vintages)
```

where:

DERRRCV ‘Device Retirements from Conversions by Vintage (mmBtu/Yr/Yr)’ [Enduse,Tech,EC,Area,Vintage]
DERVSum ‘Sum of Energy Requirement by Vintage (mmBtu/YR)’ [Enduse,Tech,EC,Area]
DERRPC 'Device Retire. from Production Capacity Retirements and Reductions in Device Saturation (mmBtu/Yr/Yr)' [Enduse,Tech,EC,Area]
DERRP 'Device Retire. from Process Retire. (mmBtu/Yr/Yr)' [Enduse,Tech,EC,Area]
DERVAllocation ‘Fraction of DER in each Vintage (mmBtu/YR)’ [Enduse,Tech,EC,Area,Vintage]
DPLV ‘Scrappage Rate of Equipment by Vintage (1/1)’ [Enduse,Tech,EC,Area,Vintage]
CFraction ‘Fraction of Production Capacity open to Conversion ($/$)’ [Enduse,Tech,EC,Area]
DERRRC ‘Device Retirements from Conversions (mmBtu/Yr/Yr)’ [Enduse,Tech,EC,Area]

If there are no conversions, then the device additions from device wear-outs (DERAD) are equal to the device retirements (DERRD) times the old (average) device efficiency (DEEA) divided by the new (marginal) device efficiency (DEE).

```
@. DERAD = DERRD*DEEAPrior/DEE
```

where:

DERAD 'Device Additions from Device Retirements (mmBtu/yr)' [Enduse,Tech,EC,Area]
DERRD ‘Device Retire. from Device Retire. (mmBtu/Yr/Yr)’ [Enduse,Tech,EC,Area]
DEE 'Device Efficiency (Btu/Btu)' [Enduse,Tech,EC,Area]
DEEAPrior 'Average Device Efficiency in Prior Year (Btu/Btu)' [Enduse,Tech,EC,Area]

If there are conversions, then the device additions from device wear-out and conversions (DERARC) are equal to the device retirements times the old (average) device efficiency (DEEAPrior) times the fraction of the devices converted (CMSF) from one Technology (CTech) to another Technology (Tech) divided by the new (marginal) device efficiency (DEE).

DERARC[enduse,tech,ec,area] = sum(DERRRC[enduse,ctech,ec,area]*DEEAPrior[enduse,ctech,ec,area]*
CMSF[enduse,tech,ctech,ec,area] for ctech in CTechs)/DEE[enduse,tech,ec,area]

where:

DERARC ‘Device Additions from Conversions (mmBtu/Yr/Yr)’ [Enduse,Tech,EC,Area]
CMSF 'Conversion Market Share Fraction by Device ($/$)' [Enduse,Tech,CTech,EC,Area]

Function RCPCDynamics: Secondary Retrofit and Conversion Dynamics

This function calculates the impact on process energy requirements and production capacity from conversions. ‘FromTech’ in the code signifies which technology the conversions are moving to. ‘ToTech’ is the new technology.

The process removals from conversions (PERRRC) are equal to the device replacements from conversions (DERRRC) times the conversion market share (CMSF). The terms are multiplied times the average device efficiency (DEEA) to convert from device to process energy requirements.

PERRRC[enduse,FromTech,ec,area] = PERRRC[enduse,FromTech,ec,area]+
DERRRC[enduse,FromTech,ec,area]*DEEAPrior[enduse,FromTech,ec,area]*
CMSF[enduse,ToTech,FromTech,ec,area]

where:

PERRRC 'Process Retire. from Device Conversions ((MBTU/YR)/YR)'[Enduse,Tech,EC,Area]
DERRRC ‘Device Retirements from Conversions (mmBtu/Yr/Yr)’ [Enduse,Tech,EC,Area]
DEEAPrior 'Average Device Efficiency in Prior Year (Btu/Btu)' [Enduse,Tech,EC,Area]
CMSF 'Conversion Market Share Fraction by Device ($/$)' [Enduse,Tech,CTech,EC,Area]

The process additions from conversions (PERARC) are calculated the same as the process removals except for the index in the conversion market share variable (CMSF). The process energy requirements are simply being moved from one technology (CTech) to another (Tech)

PERARC[enduse,ToTech,ec,area] = PERARC[enduse,ToTech,ec,area]+
DERRRC[enduse,FromTech,ec,area]*DEEAPrior[enduse,FromTech,ec,area]*
CMSF[enduse,ToTech,FromTech,ec,area]

where:

PERARC 'Process Additions from Device Conversions ((MBTU/YR)/YR)' [Enduse,Tech,EC,Area]

The production capacity "retirements" due to conversions (EUPCRC) are the process energy "retirements" due to conversions (PERRRC) times the average process efficiency (PEEA). These "retirements" are split between the vintages based on the age distribution of the production capacity (EUPC).

```
EUPCRC[enduse,tech,age,ec,area] = PERRRC[enduse,tech,ec,area]*
    PEEAPrior[enduse,tech,ec,area]*EUPCPrior[enduse,tech,age,ec,area]/
    (sum(EUPCPrior[enduse,tech,V,ec,area] for V in Ages))
```

where:

EUPCRC 'Production Capacity Retirements from Device Conversions ((M$/YR)/YR)'
[Enduse,Tech,Age,EC,Area]
PEEAPrior 'Average Process Efficiency in Prior Year ($/Btu)' [Enduse,Tech,EC,Area]
EUPCPrior 'Production Capacity by Enduse in Prior Year (M$/Yr)' [Enduse,Tech,Age,EC,Area]

Similarly, the production capacity "additions" due to conversions (EUPCAC) are the process energy retirements due to conversions (PERRRC) times the average process efficiency (PEEAPrior) times the conversion market share (CMSF). These values are split between the vintages based on the age distribution of the production capacity (EUPC).

```
EUPCAC[enduse,ToTech,age,ec,area] = EUPCAC[enduse,ToTech,age,ec,area]+
PERRRC[enduse,FromTech,ec,area]*PEEAPrior[enduse,FromTech,ec,area]*
CMSF[enduse,ToTech,FromTech,ec,area]*
EUPCPrior[enduse,FromTech,age,ec,area]/(sum(EUPCPrior[enduse,FromTech,V,ec,area] for V in Ages))
```

where:

EUPCAC 'Production Capacity Additions from Device Conversions ((M$/YR)/YR)' [Enduse,Tech,Age,EC,Area]

H. End Use and Feedstock Demand Functions

Function CapacityUtilization

In the following equations, production capacity, population, and a capacity utilization factor are mapped from ECC to EC. The capacity utilization factor (WCUF) is weighted by economic output (PC) to represent changes in economic conditions affecting capacity utilization in the determination of final demand (DMD).

```
for ec in ECs, area in Areas
 ecc = Select(ECC,EC[ec])
 SPC[ec,area] = PC[ecc,area]
 SPC0[ec,area] = PC0[ecc,area]
 SPop[ec,area] = max(Pop[ecc,area],0.000001)
 SPop0[ec,area] = max(Pop0[ecc,area],0.000001)
 WCUF[ec,area] = ECUF[ecc,area]
end
```

where:

SPC ‘Total Production Capacity (M$/Yr)’ [EC,Area]
SPC0 ‘Total Production Capacity in Initial Year (M$/Yr)’ [EC,Area]
SPOP ‘Population (Millions)’ [EC,Area]
SPOP0 ‘Population in Initial Year (Millions)’ [EC,Area]
WCUF ‘Capacity Utilization Factor Weighted by Output ($/$)’ [EC,Area]
PC ‘Production Capacity (M$/Yr)’ [ECC,Area]

PC0 ‘Production Capacity in Initial Year (M$/Yr)’ [ECC,Area]

POP ‘Population (Millions)’ [ECC,Area]

POP0 ‘Population in Initial Year (Millions)’ [ECC,Area]

ECUF ‘Capital Utilization Fraction (Btu/Btu)’ [ECC,Area]

Function EnduseSaturation: Air Conditioning Saturation

If a demand segment contains an Air Conditioning enduse (‘AC’) then special equations are applied to project AC saturation rates based on degree days and curve parameters. Saturation is set based on the input value across all enduses:

```
@. DSt = xDSt
```

where:

DST 'Device Saturation (Btu/Btu)’ [Enduse,EC,Area]

xDST 'Input Device Saturation (Btu/Btu)’ [Enduse,EC,Area]

If AC does not exist in this segment, then the function is finished. If it does (if "AC" in Enduse) then the code below is executed.

First the degree day values relevant for AC saturation (DDSatFlag) are collected.

```
DDSat[AC,area] = sum(DDayMonthly[AC,month,area]*DDSatFlag[AC,month] for month in Months)
```

where:

DDSat ' Degree Days for Saturation Equation (Degree Days)’ [Enduse,Area]

DDayMonthly 'Monthly Degree Days (Degree Days)’ [Enduse,Month,Area]

DDSatFlag ‘Flag for Degree Days in Saturation Equation (1=include)’ [Enduse,Month]

This value is then smoothed for use in the saturation equations.

```
@. DDSmooth = DDSmoothPrior+(DDSat-DDSmoothPrior)/DDSmoothingTime
```

where:

DDSmooth 'Smoothed Degree Days (Degree Days) [Enduse,Area]

DDSmoothPrior 'Smoothed Saturation Equation Degree Days in Previous Year (Degree Days) [Enduse,Area]

DDSmoothingTime ‘Smoothing Time for Saturation Equation Degree Days (Years)’ [Enduse,Area]

An assumption is applied to the heating saturations of heat pump and related devices that set them equal to the average market shares in the prior year. This value is an input to AC saturations later in this function.

```
Heat = Select(Enduse,"Heat")
techs = union(findall(Tech[:] .== "Geothermal"),
              findall(Tech[:] .== "HeatPump"),
              findall(Tech[:] .== "DualHPump"),
              findall(Tech[:] .== "FuelCell"))
if !isempty(techs)
  for ec in ECs,area in Areas
```

```
    DStHPsPrior[ec,area] = sum(AMSFPrior[Heat,tech,ec,area] for tech in techs)
  end
end
```

where:

DStHPsPrior ‘AC Saturation for Various Heat Pump Systems in Prior Year (Btu/Btu)’ [EC,Area]
AMSFPrior ‘Average Market Share in Prior Year ($/$) [Enduse,Tech,EC,Area]

Saturation for AC is then calculated using the Sailor equation if the modeled year is past 2012.

```
if CTime > 2012
…
DSt[AC,ec,area] = DStC0[AC,ec,area]+DStB0[AC,ec,area]*exp(DStA0[AC,ec,area]*DDSmooth[AC,area])
```

where:

DStC0 ‘Device Saturation Constant Term (Btu/Btu)’ [Enduse,EC,Area]
DStB0 ‘Device Saturation Area Adjustment (Btu/Btu)’ [Enduse,EC,Area]
DStA0 ‘Device Saturation Degree Day Coefficient (Btu/Btu/DD)’ [Enduse,EC,Area]

Saturation for AC is assumed to not drop. The variable is constrained to be at least the prior year’s value or to be no higher/lower than a user input max/min value.

```
DSt[AC,ec,area] = min(max(DSt[AC,ec,area],DStPrior[AC,ec,area],DStMin[AC,ec,area]),
DStMax[AC,ec,area])
```

where:

DDSatPrior ' Degree Days for Saturation Equation in Prior Year(Degree Days)’ [Enduse,Area]
DSMin ‘Minimum Device Saturation (Btu/Btu)’ [Enduse,EC,Area]
DSMax ‘Maximum Device Saturation (Btu/Btu)’ [Enduse,EC,Area]

It is assumed that heat pumps and similar heating devices also contain an air conditioning component, so an additional constraint is applied that assumes that AC saturations will be no lower than the calculated saturation value for these devices in the prior year.

```
DSt[AC,ec,area] = min(max(DSt[AC,ec,area],DStHPsPrior[ec,area]),DStMax[AC,ec,area])
```

Finally, a sector level AC saturation variable is populated for output using the saturation value weighted by production capacity.

```
SaturationWeighted = sum(DSt[AC,ec,area]*(sum(PCEUPrior[AC,tech,ec,area] for tech in Techs)) for ec
      in ECs)
CapacityWeights = sum(PCEUPrior[AC,tech,ec,area] for tech in Techs,ec in ECs)
DStAC[sector,area] = SaturationWeighted/CapacityWeights
```

where:

DStAC 'Device Saturation for AC (Btu/Btu)’ [Sector,Area]
PCEUPrior ‘Production Capacity in Prior Year (Driver/Yr)’ [Enduse,Tech,EC,Area]

Function Utilize: Budget Multiplier

This function currently simply sets the budget multiplier variable equal to the exogenous input value:

```
@. BM = BMM
```

where:

BM 'Budget Multiplier ($/$)' [Enduse,Tech,EC,Area]
BMM 'Budget Exogenous Multiplier (Btu/Btu)' [Enduse,Tech,EC,Area]

Function SequesteringEnergyPenalty: Sequestration Demands and Emissions

This function calculates the sequestration energy demands as a function of net emissions and the energy 'penalty' of running sequestration devices (SqPenaltyTech). The penalty in terms of emissions (SqPolCCPenalty) for use in gross emissions calculations later in the model execution. Sequestration fuel costs are also estimated as a function of energy demands, net sequestration, and fuel costs.

Net carbon sequestration emissions from the supply module are constrained to be a negative value, essentially removing any unexpected positive net emissions from being used in the result of the function.

```
SqPolCCNet[ecc,CO2,area] = min(SqPolCCNet[ecc,CO2,area],-0.00001)
```

where:

SqPolCCNet 'Sequestering Non-Self-generation Emissions (Tonnes/Yr)' [ECC,Poll,Area]

Energy usage for sequestration are calculated using net emissions and the energy penalty user input. Note that SqPolCCNet is negative, so SqDmd will be positive using the equation below. SqDmd is used as an input for overall demand (Dmd) later in the demand module.

```
for tech in Techs, ec in ECs, area in Areas
  ecc = Select(ECC,EC[ec])
  SqDmd[tech,ec,area] = sum(0-SqPolCCNet[ecc,poll,area]*SqPenaltyTech[tech,ec,poll,area] for poll in
        Polls)
end
```

where:

SqDmd 'Sequestering Energy Demand (TBtu/Yr)' [Tech,EC,Area]
SqPenaltyTech 'Sequestering Energy Penalty (TBtu/Tonne)' [Tech,EC,Poll,Area]

Demands are mapped to the Fuel and FuelEP level using the demand fuel fraction (DmFrac). The first enduse ("Heat") is used as an assumption for the fraction as the best fit for sequestration devices.

```
Heat = Select(Enduse,"Heat")
for area in Areas, ec in ECs, fuel in Fuels
  SqDmdFuel[fuel,ec,area] = sum(SqDmd[tech,ec,area]*DmFrac[Heat,fuel,tech,ec,area] for tech in Techs)
end
…
```

```
for fuelep in FuelEPs, ec in ECs, area in Areas
  SqDmdFuelEP[fuelep,ec,area] = sum(SqDmdFuel[fuel,ec,area]*FFPMap[fuelep,fuel] for fuel in Fuels)
end
```

where:

SqDmdFuel 'Sequestering Fuel Demands (TBtu/Yr)' [Fuel,EC,Area]

DmFrac 'Demand Fuel/Tech Fraction Split (Btu/Btu) [Enduse,Fuel,Tech,EC,Area]

SqDmdFuelEP 'Sequestering Fuel Demands (TBtu/Yr)' [FuelEP,EC,Area]

FFPMap 'Map between FuelEP and Fuel' [FuelEP,Fuel]

The penalty in emissions terms is calculated using sequestration demands and the emissions coefficient for all emitting fuel types. Again, the first enduse is used as the assumption for sequestration devices.

```
for ec in ECs, area in Areas, poll in Polls
  SqPolCCPenaltyEC[ec,poll,area] = 0-sum(SqDmdFuelEP[fuelep,ec,area]*
  POCX[1,fuelep,ec,poll,area]*(1-ZeroFr[fuelep,poll,area]) for fuelep in FuelEPs)
end

for ec in ECs, poll in Polls, area in Areas
  ecc = Select(ECC,EC[ec])
  SqPolCCPenalty[ecc,poll,area] = SqPolCCPenaltyEC[ec,poll,area]
end
```

where:

SqPolCCPenalty 'Sequestering Emissions Penalty (Tonnes/Yr)' [ECC,Poll,Area]

POCX 'Marginal Pollution Coefficients (Tonnes/TBtu)' [Enduse,FuelEP,EC,Poll,Area]

ZeroFr 'Fraction of Emissions from Zero Emission Sources (Tonnes/Tonnes)' [FuelEP,Poll,Area]

Finally, the marginal per tonne costs for sequestration are produced as a function of demand, fuel prices, and net emissions

```
for ec in ECs, area in Areas
  ecc = Select(ECC,EC[ec])
  SqFuelCost[ecc,area] = sum(SqDmd[tech,ec,area]*(ECFP[1,tech,ec,area]-
  PCostTech[tech,ec,area]) for tech in Techs)/
  (0.0-sum(SqPolCCNet[ecc,poll,area] for poll in Polls))*1000000.0
end
```

where:

SqFuelCost 'Sequestering Fuel Costs ($/Tonnes) [ECC,Area]

ECFP 'Fuel Price ($/mmBtu)' [Enduse,Tech,EC,Area]

PCostTech 'Permit Cost ($/mmBtu)' [Tech,EC,Area]

ZeroFr 'Fraction of Emissions from Zero Emission Sources (Tonnes/Tonnes)' [FuelEP,Poll,Area]

Function DmdEnduse: End-Use Demand Dynamics

This function calculates the new and average consumer energy budgets, average and marginal process efficiencies and end use demand. The new and average consumer budgets are used to formulate a budget induced usage multiplier for the end-use demand equation. The average and marginal process efficiencies are calculated for use in the new budget equation as well as for many other equations in the model. End use demand is used in the total demand function.

Budget Response

The budget constraint or response is the fuel-specific capacity utilization representing the short-term response of an energy user to rising energy prices. This response takes the form of a budget constraint which limits how much a user can afford to pay for energy in the short-term and what temporary energy saving actions the user can take (for example, turning down the thermostat, closing off rooms, biking to work).
At any point in time a change in the cost of using energy must at first be kept within the budget. Thus, if energy costs rise 10%, the first consumer response is to cut back 10%. Efficiency changes, however, alter the amount of energy required and thus reduce the energy bill. Therefore, the budget response is always less than the indicated response. After the economy and the household adjusts to the new prices, or at least the perception of them, energy use increases until the current budget matches the remembered history of the budget (called the average budget (AB)).

The current energy bill (NB - new budget) is a function of energy prices and average efficiencies. In the short run, efficiencies are constant, so a rise in energy prices causes an increase in the current energy bill, given no behavioral or capital changes.

```
NB[enduse,tech,ec,area] = ECFP[enduse,tech,ec,area]/Inflation[area]*
    DSt[enduse,ec,area]/(PEEAPrior[enduse,tech,ec,area]*DEEAPrior[enduse,tech,ec,area])
```

The “average budget” is the remembered energy bill, or in system dynamics terms, the “smoothed” energy bill. It changes from year by an increment determined by the difference between the new budget, the old average budget divided by the budget averaging time, that incrementally moves the average budget toward the new budget. For example, if the new budget is $100 more than the average budget, and the budget averaging time is equal to five years, then the average budget will increase by twenty dollars.

```
@. AB = ABPrior+DT*(NB-ABPrior)/BAT
```

where:

AB 'Average Budget ($/$)' [Enduse,Tech,EC,Area]
BAT ‘Short Term Utilization Adjustment Time (YR)’
NB 'New Budget ($/$)' [Enduse,Tech,EC,Area]

A usage multiplier is derived using input device activity rates weighted by allocation by vintage. When populated with activity rate date, this reflects the aging of stock and the lower rates of utilization in older equipment.

UMS[enduse,tech,ec,area] =
sum(DActV[enduse,tech,ec,area,vintage]*DERVAllocation[enduse,tech,ec,area,vintage] for vintage in Vintages)

where:

UMS 'Short Term Price Response (Btu/Btu)' [Enduse,Tech,EC,Area]
DActV 'Activity Rate of Equipment by Vintage (1/1)' [Enduse,Tech,EC,Area,Vintage]
DERVAllocation 'Fraction of DER in each Vintage (mmBtu/YR)' [Enduse,Tech,EC,Area,Vintage]

Average Process and Device Efficiency

Average process efficiency is calculated by multiplying the end-use production capacity for each economic class by the class device saturation divided by the process energy requirement.

PEEA[enduse,tech,ec,area] =sum(EUPC[enduse,tech,age,ec,area] for age in Ages)*
DSt[enduse,ec,area]/(PER[enduse,tech,ec,area])

The average device efficiency (DEEA) is simply the process energy requirements (PER) divided by the device energy requirements (DER):

@. DEEA=PER/DER

where:

PEEA 'Average Process Efficiency ($/Btu)' [Enduse,Tech,EC,Area]
DEEA 'Average Device Efficiency (Btu/Btu)' [Enduse,Tech,EC,Area]

Endogenous End-use Demand

The final demand for energy can now be calculated. The device energy requirements (DER) are first multiplied by a short-term budget constraint and by the utilization of capital stock (UMS and CUF). The budget constraint reflects the assumption that the short-term response to price fluctuations is primarily a cutback on consumption. The capital utilization considers that a factory may be equipped and ready but unless there is a demand for the product made in that factory, no energy will be needed. Another utilization factor (WCUF) represents utilization changes resulting from general economic disturbances.

Non-price impacts (CERSM) are included and capture such socioeconomic effects as reduced housing sizes, an increase in multifamily dwellings, and all-parent labor participation (leaving an empty house during the day).

Temperature sensitive load considerations are also included (DDM*TSLOAD) along with any changes in energy demand due to emissions reductions (RPEI).

Dmd[enduse,tech,ec,area] = DER[enduse,tech,ec,area]*
UMS[enduse,tech,ec,area]*CERSM[enduse,ec,area]*
CUF[enduse,tech,ec,area]*WCUF[ec,area]*RPEI[enduse,tech,ec,area]/1.0e6 *
(TSLoad[enduse,ec,area]*(DDay[enduse,area]/DDayNorm[enduse,area])^DDCoefficient[enduse,ec,area]+
(1.0-TSLoad[enduse,ec,area]))

where:

Dmd 'Total Energy Demand (TBtu/Yr)' [Enduse,Tech,EC,Area]

DER ‘Device Energy Requirement (mmBtu/YR)’ [Enduse,Tech,EC,Area]
CERSM 'Capital Energy Requirement (Btu/Btu)' [Enduse,EC,Area]
CUF 'Capacity Utilization Factor ($/Yr/$/Yr)' [Enduse,Tech,EC,Area]
WCUF 'Capacity Utilization Factor Weighted by Output' [EC,Area]
RPEI 'Energy Impact of Pollution Reduction (Btu/Btu)' [Enduse,Tech,EC,Area]
TSLoad 'Temperature Sensitive Fraction of Load (Btu/Btu)' [Enduse,EC,Area]
DDay ‘Annual Degree Days (Degree Days) [Enduse,Area]’
DDayNorm ‘Normal Annual Degree Days (Degree Days)’ [Enduse,Area]
DDM ‘Annual Energy Degree Day Coefficient (DD/DD)’ [Enduse,EC,Area]

Marginal Energy Requirements

A marginal energy requirement is calculated as a function of saturation, efficiency, utilitization, and market share. This variable is used as needed in the supply segment.

```
if PEE[enduse,tech,ec,area] == 0 || DEE[enduse,tech,ec,area] == 0
  DmdRq[enduse,tech,ec,area] = 0
else
  DmdRq[enduse,tech,ec,area] = DSt[enduse,ec,area]/PEE[enduse,tech,ec,area]/
  DEE[enduse,tech,ec,area]*AMSF[enduse,tech,ec,area]*CERSM[enduse,ec,area]*ECUF[ecc,area]*
  RPEI[enduse,tech,ec,area]/1.0e6
```

where:

DmdRq Tech ‘Marginal Energy Demand (TBtu/Driver)’ [Enduse,Tech,EC,Area]

End-use Demand Adjustments

The energy penalty for sequestering is added to the overall demand value.

```
Dmd[enduse,tech,ec,area] = Dmd[enduse,tech,ec,area]+
SqDmd[tech,ec,area]*SqEUTechMap[enduse,tech]
```

Specific to Industrial Light Oil Mining, demands for additional motors for enhanced oil recovery are also added to Dmd.

```
if SectorName == "Industrial"
  ec = Select(EC,"LightOilMining")
  tech = Select(Tech,"Electric")
  enduse = Select(Enduse,"Motors")
  process = Select(Process,"LightOilMining")
  for area in Areas
    Dmd[enduse,tech,ec,area] = Dmd[enduse,tech,ec,area]+OAPrEOR[process,area]*EORDmd[area]
  end
end
```

Natural gas pipelines also have demands modified when Hydrogen is in the pipeline

```
if SectorName == "Commercial"
  NGPipeline = Select(EC,"NGPipeline")
```

```
  Gas = Select(Tech,"Gas")
  OthSub = Select(Enduse,"OthSub")
  for area in Areas, ec in ECs, tech in Techs, enduse in Enduses
    if (ec == NGPipeline) && (tech == Gas) && (enduse == OthSub)
      Dmd[enduse,tech,ec,area] = Dmd[enduse,tech,ec,area]*H2PipelineMultiplier[area]
    end
  end
end
```

Electric demand is further modified by in impact of demand side management programs

@. Dmd = Dmd-DSMEU

where:

OAPrEOR 'Oil Production from EOR (TBtu/Yr)' [Process,Area]
EORDmd 'Demand for Motors for EOR (TBtu/TBtu)' [Area]
H2PipelineMultiplier 'Pipeline Efficiency Multiplier from H2 in Pipeline (Btu/Btu)' [Area]
DSMEU 'Exogenous Enduse DSM Adjustment (GWh/Yr)' [Enduse,EC,Area]

Feedstock Demands

Feedstock demands are calculated as a function of total capacity output, divided by the feedstock process efficiency (FSPEE) and modified by a utilization factor.

FsDmd[tech,ec,area] = SPC[ec,area]/FsPEE[tech,ec,area]*WCUF[ec,area]

where:

FsDmd 'Feedstock Energy Demand (TBtu/Yr)' [Tech,EC,Area]
FsPEE 'Feedstock Process Efficiency ($/mmBtu)' [Tech,EC,Area]
SPC 'Total Production Capacity (M$/Yr)' [EC,Area]

Exogenous Demands

Some demand for certain technologies or economic categories may be exogenously determined in a primarily endogenous routine. For selected technologies and/or EC's:

```
for area in Areas, ec in ECs, tech in Techs
   if DmdSw[tech,ec,area] == Exogenous
     for enduse in Enduses
       Dmd[enduse,tech,ec,area] = xDmd[enduse,tech,ec,area]
     end
     FsDmd[tech,ec,area] = xFsDmd[tech,ec,area]
  end
end
```

where:

DmdSw 'Demand Switch (0 = Exogenous)' [Tech,EC,Area]
xDmd 'Historical Energy Demand (TBtu/Yr)' [Enduse,Tech,EC,Area]
xFsDmd 'Historical Feedstock Energy (TBtu/Yr)' [Tech,EC,Area]

Function EECalculation: Impact of Energy Efficiency Programs on Demand

This function applies energy savings from energy efficiency programs to the model demand output and accumulates program costs. The equation used for the output is dependent on the value of the energy efficiency switch (EESw).

If EESw is 1.0, then Energy Efficiency (EE) is enduse demand (Dmd) times reduction from EE (EEImpact) times the fraction of load with EE (EESat).

@. EE=Dmd*EEImpact*EESat

where:

Dmd 'Total Energy Demand (TBtu/Yr)' [Enduse,Tech,EC,Area]
EE 'Energy Efficiency (TBtu/Yr)' [Enduse,Tech,EC,Area]
EEImpact 'Energy Efficiency Impact (Btu/Btu)' [Enduse,Tech,EC,Area]
EESat 'Energy Efficiency Saturation (Btu/Btu)' [Enduse,Tech,EC,Area]

When EESw is 0.0 then Energy Efficiency (EE) is set equal to the exogenous input variable

@. EE=XEE

where:

XEE 'Exogenous Energy Efficiency (TBtu)' [Enduse,Tech,EC,Area]

Total Energy Efficiency costs (EECosts) are calculated using the input cost per unit (EEUCosts)

@. EECosts=EE*EEUCosts

where:

EECosts 'Energy Efficiency Costs ($M)' [Enduse,Tech,EC,Area]
EEUCosts 'Energy Efficiency Unit Costs ($/mmBtu)' [Enduse,Tech,EC,Area]

Demand is adjusted downwards with Energy Efficiency savings

@. Dmd=Dmd-EE

Energy Efficiency savings and costs by global sector is accumulated

```
for fuel in Fuels,ec in ECs,area in Areas,
  ecc = Select(ECC,EC[ec])
  EEECC[fuel,ecc,area] = sum(EE[enduse,tech,ec,area]*DmFrac[enduse,fuel,tech,ec,area] for enduse in
          Enduses,tech in Techs)

  EECoECC[fuel,ecc,area] = sum(EECosts[enduse,tech,ec,area]*DmFrac[enduse,fuel,tech,ec,area] for
          enduse in Enduses,tech in Techs)
end
```

where:

EEECC 'Energy Efficiency (TBtu/Yr)' [Fuel,ECC,Area]
DmFrac 'Energy Demands Fuel/Tech Split (Btu/Btu)' [Enduse,Fuel,Tech,EC,Area]
EECoECC 'Energy Efficiency Costs ($M)' [Fuel,ECC,Area]

I. Self-generation Demand Functions

Function Self-generationSector: Self-generation Potential and Demand by Sector

Traditionally, the industrial sector met its electricity requirements entirely with purchases from a utility. Now the industrial sector, as well as other sectors, can meet some or all of their electricity needs by converting some of its waste heat into usable electric energy when economics warrant such an action.

This function simulates self-generation initialization at the ‘sector’ level of the model, where self-generation capacity is developed by economic category. After initialization, Self-generation is simulated at the unit level. Unit self-generation code is contained in the electric generation portion of the model.

Self-generation Emission Charges

Previous year permit costs for self-generation at the global level are inflated and mapped to the appropriate technology.

```
CgPCost1[fuelep,ecc,area] = CgPCostPrior[fuelep,ecc,area]+
        CgPCostExoPrior[fuelep,ecc,area]*Inflation[area]
CgPCost2[fuel,ecc,area] = CgPCost1[fuelep,ecc,area]
CgPCost3[fuel,ec,area] = CgPCost2[fuel,ecc,area]
…
CgPCostTech[tech,ec,area] = sum(CgPCost3[fuel,ec,area]*CgDemandPrior[fuel,ecc,area] for fuel in fuels)/
sum(CgDemandPrior[fuel,ecc,area] for fuel in fuels)
```

where:

CgPCost 'Cogen Permit Cost in Previous Year ($/mmBtu)' [FueEP,ECC,Area]
CgPCost1 'Cogen Permit Cost ($/mmBtu)' [FuelEP,ECC,Area]
CgPCost2 'Cogen Permit Cost ($/mmBtu)' [Fuel,ECC,Area]
CgPCost3 'Cogen Permit Cost ($/mmBtu)' [Fuel,EC,Area]
CgPCostExo 'Exogenous Cogen Permit Cost (Real $/mmBtu)' [FuelEP,ECC,Area]
CgPCostTech 'Self-generation Permit Cost ($/mmBtu)' [Tech,EC,Area]

Apply emissions charges to self-generation fuel price

```
@. CgECFP = FPTech+CgPCostTech
```

where:

CgECFP 'Self-generation Fuel Price ($/mmBtu)' [Tech,EC,Area]
FPTech 'Fuel Prices ($/mmBtu)' [Tech,EC,Area]

Renewable Resources and Cost Curves

```
CgDM[tech,area] = 1+CgDmSw[tech]*((CgResI[tech,area]-CgGCPrior[tech,ec,area])/CgResI[tech,area])
CgDM[tech,area] = max(CgDM[tech,area],Epsilon)
CgCC[tech,ec,area] = CgCC[tech,ec,area]*CgDM[tech,area]
```

where:

CgDM 'Depletion Multiplier ($/$)' [Tech,Area]

CgDmSw 'Depletion Multiplier Switch for Selecting Technology [Tech]
CgCC 'Self-generation Capital Cost ($/mmBtu/Yr)' [Tech,EC,Area]
CgResI 'Resource Base (mmBtu)' [Tech,Area]

Self-generation Capital Charge Rate

The self-generation capital charge rate is the annualization of self-generation capital expenses (over the life of the self-generation facility - CGTL), accounting for taxes (TXRT), tax credits (CGIVTC), and return of principal and on investment (including a self-generation risk premium and inflation: 1+ROIN+CGRISK+INSM). (1-(1/(1+ROIN+CGRISK))**CGBL)/(1-TXRT) is the classical capital recovery term. The (1-TXRT) term at the end converts the after tax calculation into before tax dollars. Investment tax credits reduce the cost of the facility by the tax credit after the first year of operation using nominal dollars. Therefore, the value of the tax credit is (CGIVTC/(1+ROIN+CGRISK+INSM). Depreciation is modeled as a current dollar phenomena which does not account for inflation. Therefore the net present value of the energy is calculated with the nominal rate of return: (2/CGTL)/(ROIN+CGRISK+INSM+2/CGTL)). It shows up as an additional negative term in the capital cost modifiers of CGCCR because depreciation is a benefit (negative cost).

Self-generation capital costs (CGCC) are multiplied by the CGCCR to get the annualized cost of the self-generation used in computing market share calculations.

The formula for calculating the self-generation capital charge rate:

```
CgCCR[ec,area] = (1-CgIVTC/(1+ROIN[ec,area]+CgRisk[tech]+
    InSm[area])-TxRt[ec,area]*(2/CgTL[tech,ec,area]))*
    (ROIN[ec,area]+CgRisk[tech])/(1-(1/(1+ROIN[ec,area]+CgRisk[tech]))^
    CgBL[tech,ec,area])/(1-TxRt[ec,area])
```

where:

CgCCR 'Self-generation Capital Charge Rate ($/Yr/$)' [EC,Area]
CgIVTC 'Self-generation Inv. Tax Credit ($/$)'
ROIN 'Return on Investment ($/Yr/$)' [EC,Area]
CgRisk 'Self-generation Excess Risk (DLESS)' [Tech]
InSm 'Smoothed Inflation Rate ($/Yr/$)'
TxRt 'Tax Rate on Energy Consumer ($/$) [EC,Area]
CgTL 'Self-generation Tax Life (YR)' [Tech,EC,Area]
CgBL 'Self-generation Equipment Book Value Lifetime (Years)' [Tech,EC,Area]

Marginal Cost of Self-generation

The amount of investment in self-generation and the amount of self-generation produced depend on the competitiveness of cogenerated electricity with utility supplied electricity. Questions of competitiveness are resolved by the calculation of market shares.

The marginal cost of self-generation is the fixed cost (CGCCR*CGCC/CGCUFP) plus the variable costs (CGVC). The fixed cost is the capital cost of self-generation equipment (CGCC), multiplied by the self-generation capital charge rate (CGCCR) to get the annualized capital cost. Variable costs (CGVC) are composed of self-generation operating costs (CGOMC - calculated as a fraction (CGOCF) of capital costs (CGCC)) plus the cost of additional fuel, if any (CGVCSW - if this switch is on then an additional fuel cost is calculated), for the self-generation equipment (fuel cost (ECFP) multiplied by the self-generation heat rate (CGHRT)) and a delivery charge (CGDC) which electric utilities may still require because they must have the capacity to supply all industrial electricity needs even when no self-generation occurs. A normal capacity utilization term (CGCUFP) is incorporated to reflect the wide variety in self-generation capacity utilization across economic sectors.

CgVC[tech,ec,area] = (CgCC[tech,ec,area]*CgOF[tech,ec,area]+
  CgDC[tech,area])*Inflation[area]+CgVCSw[tech]*(CgECFP[tech,ec,area]*
  CgHRtM[tech,ec,area]/EEConv)
CgMCE[tech,ec,area] = CgCCR[ec,area]*CgCC[tech,ec,area]/
  CgCUFP[tech,ec,area]*Inflation[area]+CgVC[tech,ec,area]

where:

CgVC 'Self-generation Variable Costs ($/mmBtu)' [Tech,EC,Area]
CgCC 'Self-generation Capital Cost ($/mmBtu/Yr)' [Tech,EC,Area]
CgOF 'Self-generation Operation Cost Fraction ($/Yr/$)' [Tech,EC,Area]
CgDC 'Self-generation Delivery Charge ($/mmBtu)' [Tech,Area]
CgVCSw 'Self-generation Variable Costs Switch for Selecting Tech' [Tech]
CgHRtM 'Marginal Self-generation Heat Rate (Btu/KWh)' [Tech,EC,Area]
EEConv 'Electric Energy Conversion (Btu/KWh)'
CgMCE 'Cogen. Marginal Cost of Energy ($/mmBtu)' [Tech,EC,Area]
CgCUFP 'Normal Cogen. Cap. Utilization Factor (Btu/Btu)' [Tech,EC,Area]

Allocation Weight and Market Share

The marginal cost of self-generation is then compared with the cost of purchased electricity (ECFP- electric) to produce the self-generation market share. A marginal allocation weight (CGMAW) is calculated for all the self-generation fuel possibilities. Non-price influences (CGMSM0) such as the additional work involved by the cogenerator, and an economic factor (CGMSMI). The CGMAW is compared to the allocation weight associated with purchased electricity (CGEAW - CGVF is the variance factor for energy decisions) to obtain the market share (CGMSF).

CgMAW[tech,ec,area] = exp(CgMSM0[tech,ec,area]+log(CgMSMM[tech,ec,area])+
CgMSMI[tech,ec,area]*(SPC[ec,area]/SPC0[ec,area])+CgVF[tech,ec,area]*log(CgMCE[tech,ec,area]/
CgMCE0[tech,ec,area]))

CgEAW[tech,ec,area] = exp(CgVF[tech,ec,area]*log(CgFP[ec,area]/CgFP0[ec,area]))

@. CgMSF = CgMAW/(CgMAW+CgEAW)

where:

CgMAW 'Self-generation Market Allocation Weight ($/$) [Tech,EC,Area]
CgMSM0 'Self-generation Market Share Non-Price Factor (Btu/Btu)' [Tech,EC,Area]

CgMSMM 'Self-generation Market Share Non-Price Factor Multiplier (Btu/Btu)' [Tech,EC,Area]
CgMSMI 'Self-generation Market Share Income Factor (Btu/\$)' [Tech,EC,Area]
SPC 'Total Production Capacity (M\$/Yr)' [EC,Area]
SPC0 'Total Production Capacity in First Year(M\$/Yr)' [EC,Area]
CgVF 'Self-generation Variance Factor (\$/\$)' [Tech,EC,Area]
CgEAW 'Self-generation Electricity Allocation Weight (\$/mmBtu)' [Tech,EC,Area]
CgFP 'Electric Price (\$/mmBtu)' [EC,Area]
CgFP0 'Electric Price in First Year (\$/mmBtu)' [EC,Area]
CgMSF 'Self-generation Market Share (Btu/Btu)' [Tech,EC,Area]

Self-generation Capacity Additions

All energy used for heating is a candidate for self-generation. This Self-generation Potential (CGPot) is the maximum self-generation power that could be used given maximum market share conditions. The basis for the potential can either be process heat demand (CgPot0) or electricity demand (CgPot1) depending on the switch (CgPotSw) set by the model user.

```
CgPot0[tech,ec,area] =sum(DER[enduse,tech,ec,area] for enduse in
        Enduses)/CgHRtM[tech,ec,area]/8760*1000
CgPotElec[ec,area] = sum(DER[enduse,tech,ec,area]/EEConv/8760*1000
     for enduse in Enduses,tech in techs)
CgPot1[tech,ec,area] = CgPotElec[ec,area]/CgCUFP[tech,ec,area]

@. CgPot = (CgPot0*(1-CgPotSw)+CgPot1*CgPotSw)*CgPotMult
```

where:

CgPot0 'Self-generation Potential (MW)' [Tech,EC,Area]
DER 'Energy Requirement (mmBtu/YR)' [Enduse,Tech,EC,Area]
CgPotElec 'Self-generation Potential Electricity Demands (MW)' [EC,Area]
WCUF 'Capacity Utilization Factor Weighted by Output' [EC,Area]
CgPot1 'Self-generation Potential (MW)' [Tech,EC,Area]
CgCUFP 'Normal Cogen. Cap. Utilization Factor (Btu/Btu)' [Tech,EC,Area]
CgPot 'Self-generation Potential (MW)' [Tech,EC,Area]
CgPotSw 'Self-generation Potential Switch (0=Steam, 1=Electric)' [Tech,EC,Area]
CgPotMult 'Self-generation Potential Multiplier (Btu/Btu)' [Tech,EC,Area]

Self-generation Construction

For simplicity, self-generation additions (CgCR) are captured as a delayed response of consumers' perceptions of self-generation's competitive advantages; actual investments are not calculated. The delay (CgAT) represents the time to approve and build a self-generation facility. The market share declines with retirements (CgR) of self-generation facilities (using a 20 year lifetime - CgPL) if the indicated market share is less than the actual share. Since it is not possible to have a negative construction rate, a XMAX function is added to keep CgCR zero or positive. The equations for self-generation capacity construction (CgCR) are:

```
@. CgIGC = max(CgPot*CgMSF,xCgIGC)
@. CgCR = (CgIGC-CgGCPrior)/CgAT+CgR
```

```
@. CgCR = max(CgCR,0)
```

where:

CgIGC 'Indicated Self-generation Capacity (MW)' [Tech,EC,Area]
XCgIGC 'Exogenous Indicated Self-generation Capacity (MW)' [Tech,EC,Area]
CgCR 'Self-generation Capacity Construction Rate (MW/Yr)' [Tech,EC,Area]
CgAT 'Self-generation Implementation Time (Years)'

Self-generation Initiation

Self-generation capacity is then initiated for construction (CgGCCIPlant) each year as a function of the construction rate (CgCR) and the user specified electric, plant, and node fractions

```
if (ElecMap[tech] == 1) && (CgElectricFraction[plant,ec,area] > 0.0)
  CgGCCIPlant[plant,ec,area] = CgCR[tech,ec,area]*CgElectricFraction[plant,ec,area]
elseif (CgPlantFraction[tech,plant] > 0.0)
  CgGCCIPlant[plant,ec,area] = CgGCCIPlant[plant,ec,area]+CgCR[tech,ec,area]*CgPlantFraction[tech,plant]
end

CgGCCI[plant,ecc,node,area] = CgGCCIPlant[plant,ec,area]*CgNodeFraction[ec,node,area]
```

where:

CgGCCI ‘Self-generation Capacity Initiated (MW)’ [Plant,ECC,Node,Area]
CgElectricFraction ‘Self-generation Electric Tech to Plant Fraction (MW/MW)’ [Plant,EC,Area]
CgPlantFraction ‘Self-generation Tech to Plant Fraction (MW/MW)’ [Tech,Plant]
CgNodeFraction ‘Self-generation EC and Area to Node Fraction (MW/MW)’ [EC,Node,Area]

Self-generation Load and Energy

The utilization of self-generation depends on the variable cost (CgVC) of operating the cogenerator versus the cost of electricity. Only the decision to build includes all the costs of self-generation; once these costs are incurred they are “sunk” and do not figure in the decision to run the self-generation unit.

```
CgRatio[tech,ec,area] = CgFP[ec,area]/
      (max(CgVC[tech,ec,area],0.000001)*CgSCM[tech])
CgRatio[tech,ec,area] = min(1.0,CgRatio[tech,ec,area])
```

where:

CgRatio 'Self-generation Cost Ratio $/$' [Tech,EC,Area]
CgSCM 'Self-generation Shared Cost Multiplier ($/$)' [Tech]

The utilization multiplier also includes a delay (CgAT) reflecting the time an owner waits to determine whether a change in economic conditions is temporary or semi-permanent. The delay also captures the distribution of many cogenerators whose costs vary from the mean. Therefore, if the current utilization factor is 60% but the ratio of electric to cogenerator energy prices (CgRatio) indicates the price differential makes it economic to cogenerate (e.g. equals 1.0), the utilization multiplier will increase not by 40% but by some fraction of that determined by the CGAT.

```
CgUMS[tech,ec,area] = CgUMSPrior[tech,ec,area]+DT*(CgRatio[tech,ec,area]-
        CgUMSPrior[tech,ec,area])/CgAT[tech,ec,area]
```

where:

CgUMS 'Self-generation Utilization Multiplier (Btu/Btu)' [Tech,EC,Area]
CgAT 'Self-generation Implementation Time (Years)' [Tech,EC,Area]

Function Self-generationTotals: Self-generation Mapping and Summation

This function maps and sums self-generation demands, generation, and capacities to combine the outputs from the sector and unit level calculations

Sector self-generation mapped to Tech

```
CgDmd[tech,ec,area] = sum(CgDemand[fuel,ecc,area]*FTMap[fuel,tech] for fuel in Fuels)
CgEG[tech,ec,area] = sum(CgGen[fuel,ecc,area]*FTMap[fuel,tech] for fuel in Fuels)
CgGC[tech,ec,area] = sum(CgCap[fuel,ecc,area]*FTMap[fuel,tech] for fuel in Fuels)

if Tech[tech] == "Electric"
  fuels = Select(Fuel,["Hydro","Wind"])
  CgDmd[tech,ec,area] = sum(CgDemand[fuel,ecc,area] for fuel in fuels)
  CgEG[tech,ec,area] = sum(CgGen[fuel,ecc,area] for fuel in fuels)
  CgGC[tech,ec,area] = sum(CgCap[fuel,ecc,area] for fuel in fuels)
end
```

where:

CgDmd 'Energy Demand from Industrial Units (TBtu/Yr)' [Tech,EC,Area]
CgEG 'Generation from Industrial Units (GWh/Yr)' [Tech,EC,Area]
CgGC 'Generating Capacity from Industrial Units (MW)' [Tech,EC,Area]
CgDemand 'Self-generation Demands (TBtu/Yr)' [Fuel,ECC,Area]
CgGen 'Self-generation Generation (GWh/Yr)' [Fuel,ECC,Area]
CgCap 'Self-generation Capacity (MW)' [Fuel,ECC,Area]

Sum both variable types to produce totals

```
CgDem[fuelep,ec,area] = sum(CgDemand[fuel,ecc,area]*FFPMap[fuelep,fuel] for fuel in Fuels)
CgEC[ecc,area] = sum(CgGen[fuel,ecc,area] for fuel in Fuels)
```

where:

CgDem 'Self-generation Demands (TBtu/Yr)' [FuelEP,EC,Area]
CgEC 'Self-generation by Economic Category (GWh/YR)' [ECC,Area]

J. Accumulate Total Energy Demand Functions

Function EnduseDemand: Total Demands

This function develops summary level outputs from the demand and price results from earlier calculations. In this function total demand by sector are calculated as well as total energy requirements.

Demand Totals by Type

The equations below sum energy demands into variables of differing set definitions.

```
EuDemF[enduse,fuel,ecc,area] = sum(Dmd[enduse,tech,ec,area]*DmFrac[enduse,fuel,tech,ec,area] for
        tech in Techs)
EuDem[enduse,fuelep,ec,area] = sum(EuDemF[enduse,fuel,ecc,area]*FFPMap[fuelep,fuel] for fuel in
        Fuels)
EuDem(EU,FuelEP,EC,A)=sum(F,T)(Dmd(EU,T,EC,A)*DmFrac(EU,F,T,EC,A)*FFPMap(FuelEP,F))
EuDemand[fuel,ecc,area] = sum(EuDemF[enduse,fuel,ecc,area] for enduse in Enduses)
```

where:

EuDemF 'Energy Demands (tBtu/Yr)' [Enduse,Fuel,ECC,Area]
DmFrac 'Energy Demands Fuel/Tech Split (Btu/Btu)' [Enduse,Fuel,Tech, EC,Area]
EuDem 'Enduse Demands (TBtu/Yr)' [Enduse,FuelEP,EC,Area]
EuDemand 'Enduse Energy Demands (TBtu/Yr)' [Fuel,ECC,Area]

Solar and Geothermal technology use a slightly difference calculation based on device efficiency (DEEA):

```
EuDemF[enduse,fuel,ecc,area] = sum(Dmd[enduse,tech,ec,area]*
    DEEA[enduse,tech,ec,area]*FTMap[fuel,tech] for tech in Techs)
```

Demands at both the Fuel/FuelEP and technology level are also estimated and saved here. These variables are used for output and to estimate fuel prices at the technology level.

```
if DmFrac[enduse,fuel,tech,ec,area] > 0
  DmdFuelTech[enduse,fuel,tech,ec,area] =
        max(Dmd[enduse,tech,ec,area],0.00001)*DmFrac[enduse,fuel,tech,ec,area]
else
  DmdFuelTech[enduse,fuel,tech,ec,area] = 0
end

DmdFEPTech[fuelep,tech,ec,area] = sum(DmdFuelTech[enduse,fuel,tech,ec,area]*FFPMap[fuelep,fuel] for
        fuel in Fuels,enduse in Enduses)
```

where:

DmdFuelTech 'Energy Demands (TBtu/Yr)' [Enduse,Fuel,Tech,EC,Area]
DmdFEPTech 'Energy Demands (TBtu/Yr)' [Enduse,FuelEP,Tech,EC,Area]

Marginal Energy Requirements

Marginal energy requirements are also accumulated into variables for use in the Supply segment.

DemRq[fuel,ecc,area] = sum(DmdRq[enduse,tech,ec,area]*DmFrac[enduse,fuel,tech,ec,area] for enduse in Enduses,tech in Techs)

where:

DemRq Tech 'Marginal Energy Demand (TBtu/Driver)' [Fuel,ECC,Area]
DmdRq Tech 'Marginal Energy Demand (TBtu/Driver)' [Enduse,Tech,EC,Area]

Special exemptions are made for US and Mexico Areas in the current model version to substitute requirements from AB due to missing data.

Function FeedstockDemand: Total Feedstock Demands

This function simply accumulates the feedstock demand values at the fuel and global sector (ECC) levels.

FsDem[fuel,ec,area] = sum(FsDmd[tech,ec,area]*FsFrac[fuel,tech,ec,area] for tech in Techs)
FsDemand[fuel,ecc,area] = FsDem[fuel,ec,area]

where:

FsDem 'Feedstock Demands (TBtu/Yr)' [Fuel,EC,Area]
FsFrac 'Feedstock Demands Fuel/Tech Split (Fraction)' [Fuel,Tech,EC,Area]
FsDemand 'Feedstock Demands (tBtu)' [Fuel,ECC,Area]

Function TotalDemand: Total Fuel Demands

This function simply accumulates components of energy to populate totals.

The sum of all demand types (TotDemand) is calculated with an adjustment to remove self-generation from Electric.

TotDemand[fuel,ecc,area] =
EuDemand[fuel,ecc,area]+FsDemand[fuel,ecc,area]+CgDemand[fuel,ecc,area]

fuel = Select(Fuel,"Electric")
…
TotDemand[fuel,ecc,area] = TotDemand[fuel,ecc,area]-CgEC[ecc,area]*EEConv/1E6

where:

TotDemand 'Energy Demands (TBtu/Yr)' [Fuel,ECC,Area]

The sum of all demand types across all sectors (DmdES) is the sum of total demand (TotDemand)

DmdES[es,fuel,area] = sum(TotDemand[fuel,ecc,area] for ecc in ECCs)

where:

DmdES 'Energy Demand (TBtu/Yr)' [ES,Fuel,Area]

Function SteamSalesPurchases: Steam Sales and Purchases

Steam sales to the market are calculated as a function of demand in any existing Steam enduse. Purchases are made to meet the needs of the Steam technology across all enduses.

First the heat rate at the technology and sector are populated from area-level input data

ESHRt[tech,ec,area] = StHR[area]

where:

ESHRt 'Excess Steam Heat Rate (Btu/KWh)' [Tech,EC,Area]
StHR 'Steam Generation Heat Rate (Btu/Btu)' [Area]

Steam sales (StSold) are a function of demand (Dmd) in the Excess Steam enduse and the steam heat rate (ESHrt)

StSold[ecc,area] = sum(Dmd[steam_enduse,tech,ec,area]/
ESHRt[tech,ec,area] for tech in Techs)

where:

StSold 'Excess Steam Generated (tBtu/Yr)' [ECC,Area]
Dmd 'Total Energy Demand (TBtu/Yr)' [Enduse,Tech,EC,Area]

Steam purchases (StPur) are a function of demand (Dmd) in the Steam technology for all enduses

StPur[ecc,area] = sum(Dmd[enduse,steam_tech,ec,area] for enduse in Enduses)

where:

StPur 'Net Steam Purchases (tBtu/Yr)' [ECC,Area]

K. Emissions Functions

Function PollCoefficients: Pollution Coefficient Calculations

In this accounting function, emissions coefficients burning of fuels at the consumer end-use and energy-supply technology level are calculated. The methodology for calculating coefficients varies depending on the switch (PolSw) set by the user.

Calculation using embedded values

The equations below are used when the pollution switch (PolSw) is set to 1 or 3.

Pollution from New Devices

Pollution from new devices (POEMA) is the sum across end-uses of the product of the device energy requirement additions (DERA) and the marginal pollution coefficient associated with operating each device (POCX).

POEMA[enduse,fuelep,ec,poll,area] = DERA[enduse,tech,ec,area]*
POCX[enduse,fuelep,ec,poll,area]/1e6

where:

POEMA 'Pollution Additions (Tonnes/Yr)' [Enduse,FuelEP,EC,Poll,Area]
DERA 'Energy Requirement Addition (mmBtu/YR)' [Enduse,Tech,EC,Area]
POCX 'Marginal Pollution Coefficients (Tonnes/TBtu)' [Enduse,FuelEP,EC,Poll,Area]

Pollution from Retired Devices

Pollution from retired devices (POEMR) is calculated in a similar fashion - as the sum across end-uses of the product of the energy associated with retired devices (DERR) and the average pollution level (POCA) when accounting for policy emission reductions (RMPrior).

POEMR[enduse,fuelep,ec,poll,area] = DERR[enduse,tech,ec,area]*
POCAPrior[enduse,fuelep,ec,poll,area]/RMPrior[fuelep,ec,poll,area]/1e6

where:

POEMR 'Pollution Retirements (Tonnes/Yr)' [Enduse,FuelEP,EC,Poll,Area]
DERR 'Device Energy Rqmt. Retire. (mmBtu/Yr/Yr)' [Enduse,Tech,EC,Area]
POCAPrior 'Average Pollution Coefficients in Prior Year (Tonnes/TBtu)' [Enduse,FuelEP,EC,Poll,Area]
RMPrior 'Reduction Multiplier in Prior Year (Tonnes/Tonnes)' [FuelEP,EC,Poll,Area]

Embedded Pollution

Embedded pollution is calculated as the existing pollution level plus any addition from new devices and reduced by pollution eliminated by retired devices. Any stock adjustments are also removed.

POEM[enduse,fuelep,ec,poll,area] = POEMPrior[enduse,fuelep,ec,poll,area]+
DT*(POEMA[enduse,fuelep,ec,poll,area]-POEMR[enduse,fuelep,ec,poll,area])

POEM[enduse,fuelep,ec,poll,area] = POEM[enduse,fuelep,ec,poll,area]*
(1+StockAdjustment[enduse,tech,ec,area])

where:

POEM 'Embodied Pollution (Tonnes/Yr)' [Enduse,FuelEP,EC,Poll,Area]
StockAdjustment ‘Exogenous Capital Stock Adjustment ($/$)’ [Enduse,Tech,EC,Area]

Average Pollution

Average pollution (POCA) is calculated as the embodied pollution divided by the device energy requirement (DER) accounting for reductions (RM).

POCA[enduse,fuelep,ec,poll,area] = POEM[enduse,fuelep,ec,poll,area]/
DER[enduse,tech,ec,area]*1E6*RM[fuelep,ec,poll,area]

where:

POCA 'Average Pollution Coefficients (Tonnes/TBtu)' [Enduse,FuelEP,EC,Poll,Area]
DER ‘Device Energy Requirement (mmBtu/YR)’ [Enduse,Tech,EC,Area]
RM(FuelEP,EC,Poll,Area) 'Reduction Multiplier(Tonnes/Tonnes)'

Calculation assuming Average equals Marginal

The equations below are used when the pollution switch (PolSw) is set to 2. This instructs the model to set average coefficients equal to the marginal without accounting for additions and retirements.

Average Pollution

Average pollution (POCA) is set equal to the marginal after accounting for reductions (RM).

```
POCA[enduse,fuelep,ec,poll,area] = POCX[enduse,fuelep,ec,poll,area]*
    RM[fuelep,ec,poll,area]
```

Embedded Pollution

Embedded pollution is produced using the average coefficient (POCA) calculated above combined with the device energy requirements (DER). Any stock adjustments are also removed.

```
POEM[enduse,fuelep,ec,poll,area] = POCA[enduse,fuelep,ec,poll,area]*
    DER[enduse,tech,ec,area]/1e6/RM[fuelep,ec,poll,area]

POEM[enduse,fuelep,ec,poll,area] = POEM[enduse,fuelep,ec,poll,area]*
    (1+StockAdjustment[enduse,tech,ec,area])
```

Non-Energy Pollution

The feedstock average emission coefficient (FsPOCA) is set equal to the marginal value (FsPOCX) or an emissions standard (FsPOCS), whichever is higher.

```
FsPOCA[fuel,ec,poll,area] = min(FsPOCX[fuel,ec,poll,area],FsPOCS[fuel,ec,poll,area])
```

where:

FsPOCA 'Feedstock Pollution Coefficients (Tonnes/TBtu)' [Fuel,EC,Poll,Area]
FsPOCX 'Feedstock Marginal Pollution Coefficients (Tonnes/TBtu)' [Fuel,EC,Poll,Area]
FsPOCS 'Feedstock Pollution Standards (Tonnes/TBtu)' [Fuel,EC,Poll,Area]

Black Carbon

An exception for Black Carbon emissions is available to instead calculate its coefficients based on a ratio of the coefficient of Particulate Matter if a model switch (BCarbonSw) is set.

```
POCX[eu,fuelep,ec,bc,area] = POCX[eu,fuelep,ec,pm25,area]*BCMult[fuel,ecc,area]
POCA[eu,fuelep,ec,bc,area] = POCA[eu,fuelep,ec,pm25,area]*BCMult[fuel,ecc,area]
FsPOCX[fuel,ec,bc,area] = FsPOCX[fuel,ec,pm25,area]   *BCMult[fuel,ecc,area]
FsPOCA[fuel,ec,bc,area] = FsPOCA[fuel,ec,pm25,area]   *BCMult[fuel,ecc,area]
RM[fuelep,ec,bc,area] = RM[fuelep,ec,pm25,area]
```

where:

BCarbonSw   'Black Carbon coefficient switch (1=POCX set relative to PM25)'
BCMult 'Fuel Emission Multipler between Black Carbon and PM 2.5 (Tonnes/Tonnes)' [Fuel,ECC,Area]

Function SequesteringPotential: Potential for Emissions Sequestrations

Emissions sequestration potential is calculated as a function of the calculated emissions inventories and emissions penalties for operating sequestration devices. Energy based inventories available for sequestration can be added or removed to the calculation using the user specified SqEnMap variable.

Inventory Accumulation

Inventories at the energy, feedstock, and self-generation levels are accumulated. A map is applied to the energy emissions to limit the enduses available.

EnSqPot[ecc,poll,area] = sum(Polute[enduse,fuelep,ec,poll,area]*SqEnMap[enduse] for enduse in Enduses,fuelep in FuelEPs)
FsSqPot[ecc,poll,area] = sum(FsPol[fuel,ec,poll,area] for fuel in Fuels)
CgSqPot[ecc,poll,area] = sum(CgFPolGross[fuelep,ecc,poll,area] for fuelep in FuelEPs)

where:

EnSqPot 'Enduse Sequestering Potential (Tonnes/Yr)' [ECC,Poll,Area]
FsSqPot 'Feedstock Sequestering Potential (Tonnes/Yr)' [ECC,Poll,Area]
CgSqPot 'Self-generation Sequestering Potential (Tonnes/Yr)' [ECC,Poll,Area]
Polute 'Pollution (Tonnes/Yr)' [Enduse,FuelEP,EC,Poll,Area]
FSPol 'Feedstock Pollution (Tonnes/Yr)' [Fuel,EC,Poll,Area]
CgFPolGross 'Gross Self-generation Emissions (Tonnes/Yr)' [FuelEP,ECC,Poll,Area]
SqEnMap 'Sequestering Enduse Map (1=include)' [Enduse]

Process inventories only use CO2 emissions.

MESqPot[ecc,CO2,area] = MEPol[ecc,CO2,area]

where:

MESqPot 'Process Sequestering Potential (Tonnes/Yr)' [ECC,Poll,Area]
MEPol 'Process Pollution (Tonnes/Yr)' [ECC,Poll,Area]

Potential Calculation

The total potential for sequestration is simply the sum of potential inventories calculated above after accounting for the sequestration penalties. The penalties can be expressed in terms of tonnes emitted or as a fraction.

@. SqPotential = (EnSqPot+FsSqPot+CgSqPot+MESqPot+SqPolCCPenalty) * (1+SqPenaltyFrac)

where:

SqPotential 'Potential Sequestering Emissions (Tonnes/Yr)' [ECC,Poll,Area]
SqPolCCPenalty 'Sequestering Emissions Penalty (Tonnes/Yr)' [ECC,Poll,Area]
SqPenaltyFrac 'Sequestering Emission Penalty (Tonne/Tonne)[ECC,Poll,Area]' [ECC,Poll,Area]

Function PRAccounting: Pollution Reduction Accounting

The following function simulates the construction, stock, and retirement of emission reduction devices to meet reduction goals from an emissions reductions program. Reduction devices, such as smokestack scrubbers, have an associated cost, construction time, and useful lifespan in the model. Retired reductions need to be replaced with new purchases in order for the emission reduction to stay at a constant level. A gradually harsher emissions cap will require more reductions constructed annually to continue to meet the target.

Reduction Accounting

The pollution reduction value is first calculated in the emission reduction section of the model. Here, the value used in this function is adjusted with any existing exogenous policy reductions (xRM).

```
@. RPFull = 1-(1-RP)*xRM
```

where:

RPFull 'Pollutant Reduction after Adjustments (Tonnes/Tonnes)' [FuelEP,EC,Poll,Area]
RP 'Pollutant Reduction (Tonnes/Tonnes)' [FuelEP,EC,Poll,Area]
xRM ‘Exogenous Average Pollution Coefficient Reduction Multiplier (Tonnes/Tonnes)' [FuelEP,EC,Poll,Area]

Indicated Reduction Capacity (RICap) is emissions divided by the reduction multiplier (to give back total embodied emissions) times the pollution reduction (RPFull) which indicates how many tonnes of pollution can be reduced.

```
RICap[fuelep,ec,poll,area] = sum(Polute[enduse,fuelep,ec,poll,area] for enduse in Enduses) +
    CgPolEC[fuelep,ec,poll,area]/RM[fuelep,ec,poll,area]*RPFull[fuelep,ec,poll,area]
```

where:

RICap 'Indicated Reduction Capital (Tonnes/Yr)' [FuelEP,EC,Poll,Area]
Polute ‘Pollution (Tonnes/Yr)’ [Enduse,FuelEP,EC,Poll,Area]
CgPolEC 'Self-generation Pollution (Tonnes/Yr)' [FuelEP,EC,Poll,Area]
RM 'Reduction Multiplier (Tonnes/Tonnes)' [FuelEP,EC,Poll,Area]

The rate of reduction capacity initiation is based previous year’s values for existing capacity (RCapPrior) and completion rate (RCR) coupled with the current indicated reduction capacity (RICap) and the construction delay (RCD)

```
RCI[fuelep,ec,poll,area] = max(RICap[fuelep,ec,poll,area]-
    RCapPrior[fuelep,ec,poll,area]-RCRPrior[fuelep,ec,poll,area]*RCD[ec,poll]+
    RCapPrior[fuelep,ec,poll,area]/RCPL[ec,poll],0.0)/RCD[ec,poll]
```

where:

RCI 'Reduction Capital Initiation (Tonnes/Yr/Yr)' [FuelEP,EC,Poll,Area]
RCapPrior 'Reduction Capital in Prior Year (Tonnes/Yr)' [FuelEP,EC,Poll,Area]
RCRPrior 'Reduction Capital Completion Rate in Prior Year (Tonnes/Yr/Yr)' [FuelEP,EC,Poll,Area]
RCD 'Reduction Capital Construction Delay (Years)' [EC,Poll]
RCPL 'Reduction Capital Physical Life (Years)' [EC,Poll]

Embedded reduction capital costs are a function of the completion rate (RCR) and marginal reduction capital cost (RCC). Costs are accumulated from the previous year’s value.

```
@. RCCEm = RCCEmPrior+DT*RCC*RCRPrior
```

where:

RCCEm 'Embedied Reduction Capital Cost ($)' [FuelEP,EC,Poll,Area]
RCC 'Reduction Capital Cost ($/Tonne)' [FuelEP,EC,Poll,Area]

Reduction capacity for the current year is calculated based on the previous year's value, the previous amount of capacity completed (RCR) and existing capacity retirements.

```
RCap[fuelep,ec,poll,area] = RCapPrior[fuelep,ec,poll,area]+DT*
    (RCRPrior[fuelep,ec,poll,area]-RCapPrior[fuelep,ec,poll,area]/RCPL[ec,poll])
```

where:

RCap 'Reduction Capital (Tonnes/Yr)' [FuelEP,EC,Poll,Area]

Finally, the reduction completion rate for the current year is calculated based on the amount initiated (RCI) and the amount of construction delay (RCD)

```
RCR[fuelep,ec,poll,area] = RCRPrior[fuelep,ec,poll,area]+DT*
    (RCI[fuelep,ec,poll,area]-RCRPrior[fuelep,ec,poll,area])/RCD[ec,poll]
```

Cost Accounting

The total amount spent on reductions is calculated as a model output. Expenses are generated using the capital cost (RCC) of completed devices (RCR) and the operation costs (ROCF) of existing devices (RCCEm). A ratio variable (AGFr) is used to allow for splitting of costs into the private or government sectors as specified by the user.

```
PRExp(ECC,Poll,Area)=sum(F)(RCR(F,EC,Poll,Area)*RCC(F,EC,Poll,Area)*
      (1-AGFr(ECC,Poll,Area))+RCCEm(F,EC,Poll,Area)*ROCF(F,EC,Poll,Area))/1e6

GRExp(ECC,Poll,Area)=sum(F)(RCR(F,EC,Poll,Area)*RCC(F,EC,Poll,Area)*
      AGFr(ECC,Poll,Area))/1e6+PAdCost(ECC,Poll,Area)
```

where:

PRExp 'Reduction Private Expenses (M$/Yr)' [ECC,Poll,Area]
AGFr 'Government Subsidy ($/$)' [ECC,Poll,Area]
GRExp 'Reduction Government Expenses (M$/Yr)' [ECC,Poll,Area]
ROCF 'Pollution Reduction O&M ($/Tonne)' [FuelEP,EC,Poll,Area]
PAdCost 'Policy Administrative Cost (Exogenous)' [ECC,Poll,Area]

Total reduction expenditures are the sum of the two expenses across pollutant types.

```
PRExpenditures[ecc,area] = sum(GRExp[ecc,poll,area]+PRExp[ecc,poll,area] for poll in Polls)
```

where:

PRExpenditures 'Pollution Reduction Expenditures (M$/Yr)' [ECC,Area]

Function PollutionGenerated: Emission Inventories

The following function produces emissions inventories from emissions coefficients and energy consumption by type

Pollution by Enduse and Technology

Energy emissions are calculated using enduse demands, the average emissions coefficients, and an option exogenous adjustment.

```
Polute[enduse,fuelep,ec,poll,area] = EuDem[enduse,fuelep,ec,area]*
    POCA[enduse,fuelep,ec,poll,area]+xPolute[enduse,fuelep,ec,poll,area]

EuFPol[fuelep,ecc,poll,area] = sum(Polute[enduse,fuelep,ec,poll,area]*
    (1-ZeroFr[fuelep,poll,area]) for enduse in Enduses)

EuPol[ecc,poll,area] = sum(EuFPol[fuelep,ecc,poll,area] for fuelep in FuelEPs)
```

where:

Polute 'Pollution (Tonnes/Yr)' [Enduse,FuelEP,EC,Area]
EuFPol 'Energy Related Pollution (Tonnes/Yr) [FuelEP,ECC,Poll,Area]
EuPol 'Enduse Energy Related Pollution' (Tonnes/Yr) [ECC,Poll,Area]
EuDem 'Enduse Demands (TBtu/Yr)' [Enduse,FuelEP,EC,Area]
POCA 'Average Pollution Coefficients (Tonnes/TBtu)' [Enduse,FuelEP,EC,Poll,Area]
xPolute 'Exogenous Pollution Adjustment (Tonnes/Yr)' [Enduse,FuelEP,EC,Poll,Area]
ZeroFr, 'Fraction of Emissions from Zero Emission Sources (Tonnes/Tonnes)' [FuelEP,Poll,Area]

OffRoad Emissions

Emissions from the Industrial OffRoad enduse are also saved to a special variable for output

```
if SectorName == "Industrial"
  enduse=Select(Enduse,"OffRoad")
  …
     OREnFPol[fuelep,ecc,poll,area]=Polute[enduse,fuelep,ec,poll,area]
```

where:

OREnFPol 'Off Road Actual Energy Related Pollution (Tonnes/Yr)' [FuelEP,ECC,Poll,Area]

Non-Combustion Pollution

```
FsPol[fuel,ec,poll,area] = FsDem[fuel,ec,area]*FsPOCA[fuel,ec,poll,area]

NcFPol[fuel,ecc,poll,area] = FsPol[fuel,ec,poll,area]
NcPol[ecc,poll,area] = sum(NcFPol[fuel,ecc,poll,area] for fuel in Fuels)
```

where:

FsPol 'Feedstock Pollution (Tonnes/Yr)' [Fuel,EC,Poll,Area]
NcFPol 'Non Combustion Related Pollution (Tonnes/Yr)' [FuelEP,ECC,Poll,Area]
NcPol 'Non Combustion Related Pollution (Tonnes/Yr)' [ECC,Poll,Area]
FsDem 'Feedstock Demands (TBtu/Yr)' [Fuel,EC,Area]
FsPOCA 'Feedstock Pollution Coefficients (Tonnes/TBtu)' [Fuel,EC,Poll,Area]

Self-Generation Pollution

CgPol[ecc,poll,area] = sum(CgFPol[fuelep,ecc,poll,area] for fuelep in FuelEPs)
CgPolEC[fuelep,ec,poll,area] = CgFPol[fuelep,ecc,poll,area]

@. SqPolCg = 0 – CgPolSq
@. SqPolCgPenalty = 0 - CgPolSqPenalty

where:

CgPol 'Self-generation Related Pollution (Tonnes/Yr)' [ECC,Poll,Area]
CgPolEC 'Self-generation Pollution (Tonnes/Yr)' [FuelEP,EC,Poll,Area]
CgFPol 'Self-generation Pollution (Tonnes/Yr) [FuelEP,ECC,Poll,Area]
SqPolCg 'Sequestering Self-generation Emissions (Tonnes/Yr)' [ECC,Poll,Area]
SqPolCgPenalty 'Sequestering Self-generation Emissions Penalty (Tonnes/Yr)' [ECC,Poll,Area]
CgPolSq 'Self-generation Gross Emissions Sequestered (Tonnes/Yr)' [ECC,Poll,Area]
CgPolSqPenalty 'Self-generation Sequestering Emissions Penalty (Tonnes/Yr) [ECC,Poll,Area]

Pollution Totals

TotFPol[fuelep,ecc,poll,area] = EuFPol[fuelep,ecc,poll,area]+
  sum(NcFPol[fuel,ecc,poll,area]*FFPMap[fuelep,fuel] for fuel in Fuels)+
  CgFPol[fuelep,ecc,poll,area]

TFPol[sector,fuelep,poll,area] = sum(TotFPol[fuelep,ecc,poll,area] for ecc in ECCs)
TSPol[sector,poll,area] = sum(TFPol[sector,fuelep,poll,area] for fuelep in FuelEPs)

where:

TotFPol 'Pollution (Tonnes/Yr)' [FuelEP,ECC,Poll,Area]
TFPol 'Energy Sector Pollution (Tonnes/Yr)' [FuelEP,Poll,Area]
TSPol 'Energy Sector Pollution (Tonnes/Yr)' [Poll,Area]

Gross Pollution Total

Gross Pollution (GrossPol) is the covered pollution before the impact of a pollution policy.

GrossEnFPol[fuelep,ec,poll,area] =
    (EuFPol[fuelep,ecc,poll,area]+CgFPol[fuelep,ecc,poll,area])/
    RM[fuelep,ec,poll,area]*PCCov[fuelep,ec,poll,area]

where:

GrossEnFPol 'Energy Emissions before Reductions (Tonnes/Yr)' [FuelEP,EC,Poll,Area]
RM 'Reduction Multiplier (Tonnes/Tonnes)' [FuelEP,EC,Poll,Area]
PCCov 'Emissions Coverage by Tech or Fuel (Tonnes/Tonnes)' [FuelEP,EC,Poll,Area]

The final sector level value only include feedstock emissions if they are covered in the policy

GrossPol[ecc,poll,area] =
   sum(GrossEnFPol[fuelep,ec,poll,area] for fuelep in FuelEPs)+
   sum(FsPol[fuel,ec,poll,area]*ECoverage[ec,poll,noncombustion,area] for fuel in Fuels)

where:

GrossPol ‘Gross Pollution - before any policies (Tonnes/Yr)' [ECC,Poll,Area]
ECoverage ‘Emissions Coverage (1=Covered)’ [EC,Poll,PCov,Area]

L. Investments Functions

Function Investments: Total Cost Accounting

The following function calculates total investments and operation and maintenance costs for devices, process, and self-generation. Total fuel expenditures are also generated as a function of demand and fuel price.

Device Investments

Spending on new devices (DInvTech) is calculated based on the marginal device full capital cost (DCCFullCost), additions to the process energy requirements (PERA), the cost and amount of retrofit devices, and any exogenous policy investments. A special case is also applied for LightOilMining investments to include investments in enhanced oil recovery devices.

```
DInvTech[enduse,tech,ec,area] = (DCCFullCost[enduse,tech,ec,area]*PERA[enduse,tech,ec,area]+
    RDCC[enduse,tech,ec,area]*PER[enduse,tech,ec,area]*RDMSF[enduse,tech,ec,area])/1e6+
    DInvTechExo[enduse,tech,ec,area]*Inflation[area]
…
if SectorName == "Industrial"
  LightOilMining = Select(EC,"LightOilMining")
…
  DInvTech[enduse,tech,ec,area] = DInvTech[enduse,tech,ec,area]+(OAPrEOR[process,area]-
          OAPrEORPrior[process,area])*EORDInv[area]
```

where:

DInvTech 'Device Investments (M$/Yr)' [Enduse,Tech,EC,Area]
DCCFullCost 'Device Capital Cost Full Cost ($/mmBtu/Yr)' [Enduse,Tech,EC,Area]
PERA 'Process Energy Rqmt. Addition (mmBtu/Yr/Yr)' [Enduse,Tech,EC,Area]
RDCC 'Retrofit Device Capital Cost ($/($/Yr))' [Enduse,Tech,EC,Area]
PER 'Process Energy Requirement (mmBtu/YR)' [Enduse,Tech,EC,Area]
RDMSF 'Device Retrofit Market Share Fraction by Device (1/Yr)' [Enduse,Tech,EC,Area]
DInvTechExo 'Device Investments (M$/Yr)' [Enduse,Tech,EC,Area]
OAPrEOR ‘Oil Production from EOR (TBtu/Yr)’ [Process,Area]
EORDInv ‘Device Investments for EOR (M$/TBtu) [Area]

Process investments from devices (PInvDevice) is calculated as a function of the change in device investments levels compared to growth in the driver. This variable is optionally passed to linked macroeconomic models to provide better investment detail. This code is limited to execute only in the forecast years to avoid conflicts with historical data.

```
PInvDevice[enduse,tech,ec,area] = DInvTech[enduse,tech,ec,area]-
  DInvTechLast[enduse,tech,ec,area]*Driver[ecc,area]/DriverLast[ecc,area]
```

where:

PInvDevice ‘Process Investments by Technology (M$/Yr)’ [Enduse,Tech,EC,Area]
Driver ‘Economic Driver (Various Millions/Yr)’ [ECC,Area]

Total device investments are summed across enduse and technology

```
DInv[ecc,area] = sum(DInvTech[enduse,tech,ec,area] for tech in Techs,enduse in Enduses)
```

where:

DInv ‘Device Investments (M$/Yr)’ [ECC,Area]

Operation and Maintenance Costs

The equation for calculating O&M costs are similar across each cost category, where O&M spending is based on an input ratio of capital cost of the total stock for the year. Two versions of device O&M costs are estimated, one with self-generation (OMExp) and one without (DOMExp).

```
OMExp[ecc,area] = sum(DCCFullCost[enduse,tech,ec,area]*DOCF[enduse,tech,ec,area]*
        PER[enduse,tech,ec,area]/1000000 for tech in Techs,enduse in Enduses)+
        sum((CgCC[tech,ec,area]*CgOF[tech,ec,area]+CgDC[tech,area])*
        CgDmd[tech,ec,area] for tech in Techs)*Inflation[area]

DOMExp[ecc,area] = sum(DCCFullCost[enduse,tech,ec,area]*DOCF[enduse,tech,ec,area]*
    PER[enduse,tech,ec,area]*Inflation[area] for tech in Techs,enduse in Enduses)/1000000

POMExp[ecc,area] = sum(PCCFC[enduse,tech,ec,area]*POCF[enduse,tech,ec,area]*
    PCEU[enduse,tech,ec,area]*Inflation[area] for tech in Techs,enduse in Enduses)

CgOMExp[ecc,area] = sum((CgCC[tech,ec,area]*CgOF[tech,ec,area]+CgDC[tech,area])*
   CgDmd[tech,ec,area] for tech in Techs)*Inflation[area]
```

where:

OMExp 'O&M Expenditures (M$) [ECC,Area]
DOCF 'Device Operating Cost Fraction ($/Yr/$)' [Enduse,Tech,EC,Area]
CgOF 'Self-generation Operation Cost Fraction ($/Yr/$)' [Tech,EC,Area]
CgDC 'Self-generation Delivery Charge ($/mmBtu)' [Tech,Area]
CgDmd 'Self-generation Energy Demand (TBtu/Yr)' [Tech,EC,Area]
DOMExp 'Device O&M Expenditures (M$)' [ECC,Area]
POMExp 'Process O&M Expenditures (M$)' [ECC,Area]
POCF 'Process Operating Cost Fraction ($/Yr/$)' [Enduse,Tech,EC,Area]
PCEU ‘Production Capacity (Driver/Yr)’ [Enduse,Tech,EC,Area]

Fuel Expenditures

Total fuel expenditures (FuelExpenditures) is a function of energy consumption and the fuel prices. Self-generation fuel expenditures are also saved to a separate variable.

```
FuelExpenditures[ecc,area] =
        sum(EuDemand[fuel,ecc,area]*FPEC[fuel,ec,area]+CgDemand[fuel,ecc,area]*FPEC[fuel,ec,area]+
        FsDemand[fuel,ecc,area]*FsFP[fuel,es,area]for fuel in Fuels)
```

CgFuelExpenditures[ecc,area] = sum(CgDemand[fuel,ecc,area]*FPEC[fuel,ec,area]
for fuel in Fuels)

where:

FuelExpenditures 'Fuel Expenditures (M$)' [ECC,Area]
CgFuelExpenditures 'Self-generation Fuel Expenditures' (M$) [ECC,Area]
EuDemand 'Enduse Energy Demands (TBtu/Yr)' [Fuel,ECC,Area]
FPEC 'Fuel Prices excluding Emission Costs ($/mmBtu)' [Fuel,EC,Area]
CgDemand 'Self-generation Demands (TBtu/Yr)' [Fuel,ECC,Area]
FsDemand(Fuel,ECC,Area) 'Feedstock Demands (tBtu)' [Fuel,ECC,Area]
FsFP 'Feedstock Fuel Price ($/mmBtu)' [Fuel,ES,Area]

Process Investments

Process investments are set using the output of the technology level equations that use capital cost combined with capital additions (PInvTech) or a floor value based on existing stock (PInvMininum) to provide a baseline level of investment in non-growing industries that have device investments. Note that the model selects the values from the first enduse (Heat) to avoid any issues with double counting spending. Process investments are also not calculated for the transportation segment.

Process investments at the technology level (PInvTech) are a function of the process capital cost (PCC), capacity additions (EUPCAPC), any process retrofits, and any exogenous reductions.

PInvTech[enduse,tech,ec,area] =
(PCC[enduse,tech,ec,area]*EUPCAPC[enduse,tech,New,ec,area]+RPCC[enduse,tech,ec,area]*sum(EUPC[enduse,tech,age,ec,area] for age in Ages)*RPMSF[enduse,tech,ec,area])+PInvExo[enduse,tech,ec,area]*Inflation[area]

where:

PInvTech 'Process Investments by Technology (M$/Yr)' [Enduse,Tech,EC,Area]
PCC 'Process Capital Cost ($/($/Yr))' [Enduse,Tech,EC,Area]
EUPCAPC 'Production Capacity Additions from New Production Capacity (M$)' [Enduse,Tech,Age,EC,Area]
RPCC 'Process Retrofit Capital Cost ($/($/Yr))' [Enduse,Tech,EC,Area]
EUPC 'Production Capacity by Enduse (M$/Yr)' [Enduse,Tech,Age,EC,Area]
RPMSF 'Process Retrofit Market Share Fraction by Device (1/Yr)' [Enduse,Tech,EC,Area]
PInvExo 'Process Exogenous Investments (M$/Yr)' [Enduse,Tech,EC,Area]

Process investments minimums (PInvMinimum) are a function of the process capital cost (PCC), capital stock (EUPC), any process retrofits, device investments, and any exogenous reductions. A user input fraction (PInvMinFrac) can be used to modify the level of the minimum or to remove it.

PInvMinimum[ecc,area] = sum(PCC[enduse,tech,ec,area]*EUPC[enduse,tech,age,ec,area] for age in Ages, tech in Techs)*PInvMinFrac[ec,area]+sum(PInvExo[enduse,tech,ec,area] for tech in Techs)*Inflation[area]+sum(DInvTech[enduse,tech,ec,area] for tech in Techs)

where:

PInvMinimum 'Process Investments Minimum Level (M$/Yr)' [ECC,Area]

EUPC ‘Production Capacity by Enduse (Driver/Yr) [Enduse,Tech,Age,EC,Area]
PInvMinFrac ‘Minimum Fraction for Process Investments ($/$) [EC,Area]

The technology level value is summed across enduse and technology. The output investment level is the higher of the two investment variables. PInvTech is adjusted to match the minimum value if it selected.

PInvDriver[ecc,area] = sum(PInvTech[enduse,tech,ec,area] for tech in Techs)
PInv[ecc,area] = max(PInvDriver[ecc,area],PInvMinimum[ecc,area])
PInvTech[enduse,tech,ec,area] = PInvTech[enduse,tech,ec,area]/PInvDriver[ecc,area]*PInv[ecc,area]

where:

PInv ‘Process Investments (M$/Yr)’ [ECC,Area]

Reduction Investments

Emissions reduction investments are using the reduction completion rate (RCR) with the reduction capital cost (RCC) after accounting for government subsidies.

RInv[ecc,area] = sum(RCR[fuelep,ec,poll,area] *RCC[fuelep,ec,poll,area]*
(1-AGFr[ec,poll,area]) for poll in Polls,fuelep in FuelEPs)/ 1000000

where:

RInv ‘Emission Reduction Investments (M$/Yr)’ [ECC,Area]
RCR ‘Reduction Capital Completion Rate (Tonnes/Yr/Yr)’ [FuelEP,EC,Poll,Area]
RCC ‘Reduction Capital Cost ($/Tonne)’ [FuelEP,EC,Poll,Area]
AGFr ‘Government Subsidy ($/$)’ [ECC,Poll,Area]

Operation and maintenance expenditures use embedded costs and the operation and maintenance fraction.

ROMExp[ecc,area] = sum(RCCEm[fuelep,ec,poll,area]*ROCF[fuelep,ec,poll,area] for poll in Polls, fuelep in FuelEPs)/1e6

where:

ROMExp ‘Emission Reduction O&M Expenditures (M$)’ [ECC,Area]
RCCEm ‘Embedded Reduction Capital Cost ($)’ [FuelEP,EC,Poll,Area]
ROCF ‘Pollution Reduction O&M ($/Tonne)’ [FuelEP,EC,Poll,Area]

Appendix 5. Load Curve Functions and Equations Detail

This section describes code from the Load.jl files in the model Engine subfolder. The functions below serve as a link between the demand and electric generation and gas supply sections of the model. Demands (Dmd) from the demand segment are accumulated and sorted into variables dimensioned by time periods using input load shape values for use as an input for supply production and dispatch.

Function CogenGeneration

Self-generation is sorted into the appropriate technology and sector using variable maps. These values are using in the next function as a modifier to sales and loads sent to the electric generation module.

Total sector self-generation is collected across fuel type.

```
CgEC[ecc,area] = sum(CgGen[fuel,ecc,area] for fuel in Fuels)
```

where:

CgEC 'Self-generation by Economic Category (GWh/YR)' [ECC,Area]

CgGen 'Self-generation Demands (TBtu/Yr)' [Fuel,ECC,Area]

Self-generation is collected at the technology level using a map between technology and fuel. Demands from Hydro and Wind fuels are explicitly placed into the Electric technology.

```
CgEG[tech,ec,area] = sum(CgGen[fuel,ecc,area]*FTMap[fuel,tech] for fuel in Fuels)
if Tech[tech] == "Electric"
  fuels = Select(Fuel,["Hydro","Wind"])
  CgEG[tech,ec,area] = sum(CgGen[fuel,ecc,area] for fuel in fuels)
end
```

where:

CgEG 'Electricity from Self-generation (GWh/YR)' [Tech,EC,Area]

FTMap 'Map between Fuel and Tech (Map)' [Fuel,Tech]

Function ElectricitySales

Total electric sales at the sector level are calculated as a function of electric fuel demands from the demand module after accounting for self-generation and any sales to the grid.

Electric sales across technology are accumulated using the demand fuel/technology fraction and converted to electric units (GWh).

```
fuel = Select(Fuel,"Electric")
..
ESales[enduse,ec,area] = sum(Dmd[enduse,tech,ec,area]*
      DmFrac[enduse,fuel,tech,ec,area] for tech in Techs)/EEConv*1e6
```

where:

ESales 'Electricity Gross Demands (GWh/Yr)' [Enduse,EC,Area]

Dmd 'Energy Demand (TBtu/Yr)' [Enduse,Tech,EC,Area]

DmFrac ‘Demand Fuel/Tech Fraction Split (Btu/Btu)’ [Enduse,Fuel,Tech,EC,Area]
EEConv ‘Electric Energy Conversion (Btu/KWh)’

Demands are also accumulated across enduse. The final electric sales is then calculated after subtracting self-generation and adding grid sales.

```
ElecDmd[ecc,area] = sum(ESales[enduse,ec,area] for enduse in Enduses)
SaEC[ecc,area] = ElecDmd[ecc,area]-CgEC[ecc,area]+PSoECC[ecc,area]
```

where:

ElecDmd ‘Electricity Gross Demands (GWh/Yr)’ [ECC,Area]
SaEC ‘Electricity Sales (GWh/Yr)’ [ECC,Area]
PSoECC ‘Power Sold to Grid (GWh/Yr)’ [ECC,Area]

Function LoadCurve

Each electric end-use has a set of load shape factors (LSF) attributed to it. The LSF are estimated, end-use specific ratios that compare the kW contribution to the system seasonal peak and minimum load to the average load. (The load shape classifications are Peak, Average, Minimum load for each season - Winter, Spring, Summer, Fall, Late Fall). For example, the LSF (Winter Peak) for residential air conditioning is 0.0 while the LSF (Summer Peak) may be 6.0. The LSF (Average) over the year is always 1.0 unless a load management policy is specified. The seasonal LSF (Average) may, however, be much different than one.

Sales at the enduse, sector, and area level (ESales) are allocated into variables that include hours, months, and days using the load shape factor. ‘Day’ in the current model version consists of three representative time periods for loads of ‘Average’, ‘Minimum’, and ‘Peak’, each set using a slightly different equation.

The ‘Average’ portion is set using the load shape after converting the units from GWh to MW (/8750*1e3).

```
day = Select(Day,"Average")
LDCEU[enduse,ec,hour,day,month,area] =
        ESales[enduse,ec,area]*LSF[enduse,ec,hour,day,month,area]/8760*1e3
```

where:

LDCEU ‘Electric Load Curve (MW)’ [Enduse,EC,Hour,Day,Month,Area]
LSF ‘Load Shape Factor (MW/MW)’ [Enduse,EC,Hour,Day,Month,Area]

Load in the minimum portion is additionally adjusted using a output from the supply calibration (BaseAdj). This variable is populated by adjusting estimated minimum loads with historical minimum load data.

```
day = Select(Day,"Minimum")
LDCEU[enduse,ec,hour,day,month,area] = ESales[enduse,ec,area]*
        LSF[enduse,ec,hour,day,month,area]*BaseAdj[day,month,area]/8760*1e3
```

where:

BaseAdj ‘Adjustment Based on All Years (MW/MW)’ [Day,Month,Area]

Load in the peak period contains an additional adjustment for temperature sensitive loads (TSLoad) using a peak multiplier factor (HPKM). Additionally, any savings from peak savings programs (xPkSav) are subtracted from the final load.

```
day = Select(Day,"Peak")
LDCEU[enduse,ec,hour,day,month,area] = ESales[enduse,ec,area]*LSF[enduse,ec,hour,day,month,area]*
        BaseAdj[day,month,area]/8760*1e3*(HPKM[month,area]*TSLoad[enduse,ec,area]+(1-
        TSLoad[enduse,ec,area]))-xPkSav[enduse,ec,area]
```

where:

HPKM 'Peak Day Multiplier (MW/MW)' [Month,Area]
TSLoad 'Temperature Sensitive Fraction of Load (Btu/Btu)' [Enduse,EC,Area]
xPkSav 'Exogenous Peak Savings (MW)' [Enduse,EC,Area]

The output values from these equations are then accumulated across enduse.

```
LDCEUECC[ecc,hour,day,month,area] = sum(LDCEU[enduse,ec,hour,day,month,area]
   for enduse in Enduses)
```

where:

LDCEUECC 'Electric Load Curve (MW)' [ECC,Hour,Day,Month,Area]

Self-generation loads are then calculated similarly to enduse demands by using the input self-generation load shape factors (CgLSF). Self-generation uses the same equation across different Day types. These are then accumulated across technology.

```
CgLDC[tech,ec,hour,day,month,area] = CgEG[tech,ec,area]*
    CgLSF[tech,ec,hour,day,month,area]/8760*1E3
CgLDCECC[ecc,hour,day,month,area] = sum(CgLDC[tech,ec,hour,day,month,area]
```

where:

CgLDC 'Self-generation Load Curve (MW)' [Tech,EC,Hour,Day,Month,Area]
CgLSF 'Self-generation Load Shape (MW/MW)' [Tech,EC,Hour,Day,Month,Area]
CgLDCECC 'Self-generation Load Curve (MW)' [ECC,Hour,Day,Month,Area]

Similarly, any self-generation sales to grid are calculated using values from the demand module and a grid sales load shape (CgLSFSold)

```
CgLDCSold[ec,hour,day,month,area] = PSoECC[ecc,area]*CgLSFSold[ec,hour,day,month,area]/8760*1E3
CgLDCSoldECC[ecc,hour,day,month,area] = CgLDCSold[ec,hour,day,month,area]
```

where:

CgLDCSold 'Self-generation Sold to Grid Load Curve (MW)' [EC,Hour,Day,Month,Area]
CgLSFSold 'Self-generation Sold to Grid Load Shape (MW/MW)' [EC,Hour,Day,Month,Area]
CgLDCSoldECC 'Self-generation Sold to Grid Load Curve (MW)' [ECC,Hour,Day,Month,Area]

Finally, totals for dispatched electric and electricity from the grid are calculated after accounting for self-generation

```
LDCECCGrid[ecc,hour,day,month,area] = LDCEUECC[ecc,hour,day,month,area]-
        CgLDCECC[ecc,hour,day,month,area]+CgLDCSoldECC[ecc,hour,day,month,area]
..
```

```
LDCECC[ecc,hour,day,month,area] = LDCEUECC[ecc,hour,day,month,area]-
            CgLDCECC[ecc,hour,day,month,area]
```

where:

LDCECCGrid ‘Electric Loads from Grid (MW)’ [ECC,Hour,Day,Month,Area]
LDCECC ‘Electric Loads Dispatched (MW)’ [ECC,Hour,Day,Month,Area]

A temperature sensitive peak load is also calculated for output

```
day = Select(Day,"Peak")
LDCTS[ecc,hour,month,area] = sum(LDCEU[enduse,ec,hour,day,month,area]*TSLoad[enduse,ec,area] for
            enduse in Enduses)
```

where:

LDCTS ‘Temperature Sensitive Electric Peak Load (MW)’ [ECC,Hour,Month,Area]

Function GasSales

Similar to Electric Sales, sales of Natural Gas are accumulated from the demand module to serve as inputs for the gas load shape equations. These equations can also include any gas values from feedstock demands (FsDmd).

Gas sales across technology are accumulated using the demand fuel/technology fraction for enduse demands. Feedstock demands are included depending on the value of the input feedstock switch (DUCFSw) at the enduse level. Demands are then converted to units fitting the gas dispatch module using GECONV.

```
fuel = Select(Fuel,"NaturalGas")
…
GSales[enduse,ec,area] =
   (sum(Dmd[enduse,tech,ec,area]*DmFrac[enduse,fuel,tech,ec,area] for tech in Techs)+
    sum(FsDmd[tech,ec,area]*FsFrac[fuel,tech,ec,area]*DUCFSw[enduse] for tech in Techs))*
    GECONV
```

where:

GSales ‘Gas Sales (MTherm/Yr)’ [Enduse,EC,Area]
FsDmd ‘Feedstock Energy Demand (TBtu/Yr) [Tech,EC,Area]
FsFrac ‘Feedstock Demands Fuel/Tech Split (Fraction)’ [Fuel,Tech,EC,Area]
DUCFSw ‘Switch for Self-generation and Feedstock Demand’ [Enduse]
GEConv ‘Gas Energy Conversion (Therm/mmBtu)’

Gas sales are accumulated across enduse and sector.

```
Sales[class,fuel,area] = sum(GSales[enduse,ec,area] for enduse in Enduses, ec in ECs)
```

where:

Sales ‘Gas Sales (MTherm/Yr)’ [Class,Fuel,Area]

Function DailyUse: Gas utility Daily use curve

Each end-use has a set of load shape factors (DUF) associated with it. The DUFs are estimated end-use specific ratios that compare the MTherm/Day contribution to the system seasonal peak and minimum load to the average load. (The load shape classifications are Peak, Average, and Minimum Load for each season). As an example, the DUF(winter peak) for residential space heat may be 3.0 whereas the DUF(summer peak) is 0.0. The DUF(average) over the year is always 1.0 unless a load management policy is specified. The seasonal DUF average may, however, be much different than 1.0.

The end-use daily use curve (ECDUC) by class, season, and load shape classification are the gas sales (GSales) multiplied by the respective DUF, converted into a daily value

```
ECDUC[enduse,day,month,ec,area] = GSales[enduse,ec,area]*DUF[enduse,ec,day,month,area]/365
```

where:

ECDUC ‘Gas End-use Load Curve (MTherm/Day) [Enduse,Day,Month,EC,Area]
DUF ‘'Daily Use Factor (Therm/Therm) [Enduse,Day,Month,EC,Area]

The winter daily use curves (EUDUC for Peak loads) are comprised largely of temperature sensitive primary heating load (TSLoad). This load is corrected for daily peak weather effects (DPKM).

```
day = Select(Day,"Peak")
ECDUC[enduse,day,month,ec,area] =
        GSales[enduse,ec,area]*(DPKM[month,area]*TSLoad[enduse,ec,area]+(1-
        TSLoad[enduse,ec,area]))
```

where:

DPKM ‘Gas Peak Day Multiplier (Therm/Day/Therm/Day)’ [Month,Area]

Finally, gas load shapes are accumulated across enduse and sector

```
CDUC[class,day,month,area] = sum(ECDUC[enduse,day,month,ec,area] for enduse in Enduses,ec in ECs)
TSDUC[day,month,class,area] = sum(ECDUC[enduse,day,month,ec,area]*TSLoad[enduse,ec,area] for
        enduse in Enduses,ec in ECs)
```

where:

CDUC ‘Gas Gross Load Curve (MTherm/Day)’ [Class,Day,Month,Area]
TSDUC ‘Temperature Sensitive Load Curve (MTherm/Day)’ [Day,Month,Class,Area]

Appendix 6. Initialization Functions and Equations Detail

'Initialization' code describes the series of equations that are used to prepare the model for demand calibration found in the Initial.jl files in the model Engine. These equations primarily consist of estimating demand requirements and parameters for the first modeled year from historical data and populating the device and process cost curves.

A summary of the functions that make up the demand initialization code is listed below. Details on the key inputs, outputs, and equations of each of these functions follows.

Table 10. Initialization Function Objectives

	Function Name	Function Objective
1.	Lifetimes	Initializes lifetimes and several other variables for use in the rest of the demand module.
2.	IPrice	Calculates the local fuel price by economic category and device capital charge rate. Local fuel prices by economic category are set by fuel prices except in the case of electric technologies, which instead use the electric price. Assigns fuel price for each economic category (ECFP).
3.	DEffCurve	Derives coefficients of trade-off curves for projecting device efficiency and capital costs based on input data provided for a single initialization year.
4.	Initial	Initializes variables required for developing process efficiency and cost curves.
5.	PEffCurve	Derives coefficients of trade-off curves for projecting process efficiency and process capital costs based on input data provided for a single initialization year.
6.	PEffFuture	Assigns values for future years of process (and retrofit) capital cost and efficiency variables equal to the value in the initial year.
7.	PEffAdjust	Adjusts values for sectors or areas where driver is very small to allow for sectors that may not exist yet during initialization year.

8.	Pollution	Initializes the average pollution coefficients by setting equal to the marginal in the initial year.

A. Function Lifetimes

This function initializes device and process lifetimes and several other variables for use in the rest of the demand module.

Function Lifetimes

Key Inputs

- xDPL 'Exogenous Physical Life of Equipment (Years)' [Enduse,Tech,EC,Area,Year]
- PCPL ‘Physical Life of Production Capacity (Years)' [ECC,Area,Year]
- TaxPct 'Standard accounting percent of device life that is taxed' [Area]
- PEPL 'Physical Life of Process Energy Requirements (Years)' [Enduse,Tech,EC,Area,Year]
- xDst ‘Device Saturation (Btu/Btu)’ [Enduse,EC,Area,Year]
- xDEMM ‘Maximum Device Efficiency Multiplier (Btu/Btu)’ [Enduse,Tech,EC,Area,Year]
- xDCMM ‘Capital Cost Maximum Multiplier ($/$)’ [Enduse,Tech,EC,Area,Year]

Key Outputs

- DPL 'Physical Life of Equipment (Years)' [Enduse,Tech,EC,Area,Year]
- DTL 'Device Tax Life (Years)' [Enduse,Tech,EC,Area,Year]
- PETL 'Tax Life of Process Energy Requirements (Years)' [Enduse,Tech,EC,Area,Year]
- YDst ‘Device Saturation (Btu/Btu)’ [Enduse,EC,Area,Year]
- DEMM ‘Maximum Device Efficiency Multiplier (Btu/Btu)’ [Enduse,Tech,EC,Area,Year]
- DCMM ‘Capital Cost Maximum Multiplier ($/$)’ [Enduse,Tech,EC,Area,Year]

Key Equations

Device lifespans (DPL) are assumed to be equal to the input value (XDPL) or the productive capacity lifespan (PCPL), whichever is shorter

```
DPL[enduse,tech,ec,area,year] = min(xDPL[enduse,tech,ec,area,year],
                                    PCPL[ecc,area,year])
```

The device tax lifetime is based on the device lifespan (DPL) and the percentage of taxable lifespan variable (TAXPCT)

```
DTL[enduse,tech,ec,area,year] = DPL[enduse,tech,ec,area,Zero]*
                                TaxPct[area,year]
```

Process requirement lifespans (PEPL) are equal to device lifespans (DPL), except in the case of the Heat and AC enduses which instead use the productivity capacity lifespan (PCPL). The tax life is set similarly as devices.

```
@. PEPL=DPL
if (Enduse[enduse] == "Heat") || (Enduse[enduse] == "AC") || (Enduse[enduse] == "Carriage")
  PEPL[enduse,tech,ec,area,year] = sum(PCPL[ecc,area,year])
end
```

The process energy requirement is set similarly as devices.

PETL[enduse,tech,ec,area,year] = PEPL[enduse,tech,ec,area,Zero]*
TaxPct[area,year]

Finally, device saturation and curve multipliers are set equal to the user input value.

YDSt[enduse,ec,area,year] = xDSt[enduse,ec,area,year]
DEMM[enduse,tech,ec,area,year] = xDEMM[enduse,tech,ec,area,year]
DCMM[enduse,tech,ec,area,year] = xDCMM[enduse,tech,ec,area,year]

B. Function PRReductions

This function initializes the demands variable that contain any historical emission market parameters, including coverages, costs, and exogenous reductions.

Function PRReductions
Key Inputs
• ECovECC 'Emissions Coverage (1=Covered)' [ECC,Poll,PCov,Area,Year] • PCovMap 'Pollution Coverage Map (1=Mapped)' [FuelEP,ECC,PCov,Area,Year] • PCostECC 'Permit Cost ($/Tonne)' [ECC,Poll,Area,Year] • xRM 'Exogenous Average Pollution Coefficient Reduction Multiplier (Tonnes/Tonnes) [FuelEP,EC,Poll,Area]
Key Outputs
• ECoverage 'Emissions Coverage (1=Covered)' [EC,Poll,PCov,Area,Year] • PCCov 'Emissions Coverage by Tech or Fuel (Tonnes/Tonnes)' [FuelEP,EC,Poll,Area,Year] • PCost 'Permit Cost ($/Tonne)' [Tech,EC,Area,Year] • RM 'Reduction Multiplier (Tonnes/Tonnes)' [FuelEP,EC,Poll,Area,Year]
Key Equations
This function primarily maps variables from the supply segment to its equivalent on the demand side for usage in demand equations. Coverages are mapped from the ECC to EC and into fuels using PCovMap. ECoverage[ec,poll,pcov,area,year] = ECovECC[ecc,poll,pcov,area,year] … PCCov[fuelep,ec,poll,area,year] = maximum(ECoverage[ec,poll,pcov,area,year]* PCovMap[fuelep,ecc,pcov,area,year] for pcov in PCovs) Costs are similarly mapped using coverages. PCost[fuelep,ec,poll,area,year] = PCostECC[ecc,poll,area,year]*PCCov[fuelep,ec,poll,area,year] The reduction multiplier is set equal to any existing exogenous value to be included in the model calibration. This typically is set by historical emissions policies. @. RM = xRM

C. Function TPrice

This function maps uses input demands to weight the impacts of emissions costs at the technology level. Local fuel prices by economic category are set by fuel prices except in the case of electric technologies, which instead use the electric price. These costs are incorporated into fuel price variables that will be used to pass along impacts to the consumer via the efficiency and cost curves. Equations in this function are called from ‘IPrice’ and modify variables just for the year ‘curtime’, which is the initialization year for the curve equations.

This section will highlight only the key inputs and outputs from TPrice since the code mirrors the function of the same name in the Demand and Calib module code. Please see the documentation for this Demand code for TPrice more detail about the methodology.

Function TPrice
Key Inputs
• xDmdPrior ‘Energy Demand (TBtu/Yr)’ [Enduse,Tech,EC,Area,Year]
• POCX, ‘Marginal Pollution Coefficients (Tonnes/TBtu) [Enduse,FuelEP,EC,Poll,Area,Year]’
• ZeroFr, ‘Fraction of Emissions from Zero Emission Sources (Tonnes/Tonnes)’ [FuelEP,Poll,Area,Year]
• RM, Reduction Multiplier (Tonnes/Tonnes) [FuelEP,EC,Poll,Area,Year]
• PCost, ‘Permit Cost ($/Tonne)’ [FuelEP,EC,Poll,Area,Year]
• PCostExo, ‘Marginal Exogenous Permit Cost (Real $/Tonnes)’ [FuelEP,EC,Poll,Area,Year]
Key Outputs
Calculated Outputs (For Initialization Year)
• PCostTech ‘Permit Cost ($/mmBtu)’ [Tech,EC,Area,Year]
• FEPCP, ‘Carbon Price by FuelEP ($/mmBtu)’ [FuelEP,EC,Area,Year]
• FPCFS ‘CFS Price ($/mmBtu)’ [Fuel,EC,Area,Year]
• ECFP ‘Fuel Price ($/mmBtu)’ [Enduse,Tech,EC,Area,Year]
• ECFPFuel ‘Fuel Price ($/mmBtu)’ [Fuel,EC,Area,Year]
• FPEC ‘Fuel Prices excluding Emission Costs ($/mmBtu)’ [Fuel,EC,Area,Year]
• FPCFSNet, ‘Net CFS Price ($/mmBtu)’ [Fuel,EC,Area,Year]
• FPCP ‘Carbon Price before OBA ($/mmBtu)’ [Fuel,EC,Area,Year]

D. Function IPrice

This function calculates the local fuel price by economic category (via TPrice) and sets the device capital charge rate.

Function IPrice
Key Inputs
Inputs from Function Lifetimes
• DPL 'Physical Life of Equipment (Years)' [Enduse,Tech,EC,Area,Year]

- DTL 'Device Tax Life (Years)' [Enduse,Tech,EC,Area,Year]

Exogenous Inputs

- DIVTC ‘Device Investment Tax Credit ($/$)’ [Tech,Area]
- DRisk ‘Device Excess Risk ($/$)’ [Enduse,Tech]
- DTL ‘Device Tax Life (Years)’ [Enduse,Tech,EC,Area,Year]
- InSm ‘'Smoothed Inflation Rate ($/Yr/$)' [Area,Year]
- ROIN ‘Return on Investment ($/Yr/$)’ [EC,Area]
- TxRt ‘Tax Rate on Energy Consumer ($/$) [EC,Area,Year]’

Key Outputs

Calculated Outputs (For Initialization Year)

- DCCRN 'Device Capital Charge Rate ($/Yr/$)' [Enduse,Tech,EC,Area,Year]

Key Equations

IPrice first selects the initialization year (CurTime) and calls the prior functions to populate the variables

```
curtime = Int(CurTime)

PRReductions(data)
TPrice(data,curtime,ECs,ECCs,Areas)
```

Investment Levelization Rate - the Capital Charge Rate

The device capital charge rate is the annualization of device capital expenses (over the life of the device - DTL), accounting for taxes (TXRT), tax credits (DIVTC), and return of principal and on investment (including risk premiums and inflation: 1+ROIN+DRISK+INSM). (1-(1/(1+ROIN+DRISK))**DPL)/(1-TXRT) is the classical capital recovery term. The (1-TXRT) term at the end converts the after tax calculation into before tax dollars. Investment tax credits reduce the cost of the facility by the tax credit after the first year of operation using nominal dollars. Therefore the value of the tax credit is (DIVTC/(1+ROIN+DRISK+INSM). Depreciation is modeled as a current dollar phenomena which does not account for inflation. Therefore the net present value of the energy is calculated with the nominal rate of return: (2/DTL)/(ROIN+DRISK+INSM+2/DTL)). It shows up as an additional negative term in the capital cost modifiers of DCCR because depreciation is a benefit (negative cost).

Device capital costs (DCC) are multiplied by the DCCRN to get the annualized cost of the device used in computing market share calculations.

The formula for calculating the device capital charge rate:

```
DCCRN[enduse,tech,ec,area,curtime] =
   (1-DIVTC[tech,area,curtime]/(1+ROIN[ec,area]+DRisk[enduse,tech]+
   InSm[area,curtime])-TxRt[ec,area,curtime]*
   (2/DTL[enduse,tech,ec,area,curtime])/
   (ROIN[ec,area]+DRisk[enduse,tech]+InSm[area,curtime]+
   2/DTL[enduse,tech,ec,area,curtime]))*(ROIN[ec,area]+DRisk[enduse,tech])/
   (1-(1/(1+ROIN[ec,area]+DRisk[enduse,tech]))^
   DPL[enduse,tech,ec,area,curtime])/(1-TxRt[ec,area,curtime])
```

E. Function DEffCurve

This function derives the device Consumer Preference Efficiency and Capital Cost Curves based on the input data provided for the single initialization year (CurTime). Once calculated, the capital cost and fuel cost coefficients (DCTC, DFTC) are held constant for all years in the model.

Function DEffCurve

Key Inputs

Inputs from Function IPrice

- ECFP ‘Fuel Price ($/mmBtu)’ [Enduse,Tech,EC,Area,Year]
- DCCRN 'Device Capital Charge Rate ($/Yr/$)' [Enduse,Tech,EC,Area,Year]

Exogenous Inputs

- xDEE 'Historical Device Efficiency (Btu/Btu)' [Enduse,Tech,EC,Area,Year]
- DESTD 'Device Efficiency Standards (Btu/Btu)' [Enduse,Tech,EC,Area,Year]
- DESTDP 'Device Efficiency Standards Policy (Btu/Btu)' [Enduse,Tech,EC,Area,Year]
- DOCF 'Device Operating Cost Fraction' [Enduse,Tech,EC,Area,Year]
- DEM 'Maximum Device Efficiency (Btu/Btu)' [Enduse,Tech,EC,Area,Year]
- xDCC 'Device Capital Cost ($/mmBtu/Yr)' [Enduse,Tech,EC,Area,Year]

Key Outputs

Calculated Outputs (for Initialization Year)

- DEEStd 'Device Efficiency with Standard (Btu/Btu)' [Enduse,Tech,EC,Area]
- DCCN 'Normalized Device Capital Cost ($/mmBtu)' [Enduse,Tech,EC,Area]
- DFPN 'Normalized Fuel Price ($/mmBtu)' [Enduse,Tech,EC,Area]

Outputs (for All Years)

- DCTC 'Device Cap. Trade Off Coefficient (DLESS)' [Enduse,Tech,EC,Area,Year]
- DFTC 'Device Fuel Trade Off Coefficient (DLESS)' [Enduse,Tech,EC,Area,Year]

Key Equations

Apply Device Efficiency Standard

Assume that the input capital cost data (XDCC) is based on the level of efficiency of a historical standard for the initialization year if one exists.

```
DEEStd[enduse,tech,ec,area] = max(xDEE[enduse,tech,ec,area,curtime],
    DEStd[enduse,tech,ec,area,curtime],DEStdP[enduse,tech,ec,area,curtime])
```

Capital Cost Coefficient

```
DCTC[enduse,tech,ec,area,curtime] =
    -1/((ECFP[enduse,tech,ec,area,curtime]/Inflation[area,curtime]/
    xDEE[enduse,tech,ec,area,curtime])/
    ((DCCRN[enduse,tech,ec,area,curtime]+DOCF[enduse,tech,ec,area,curtime])*
    xDCC[enduse,tech,ec,area,curtime])*
    (1-xDEE[enduse,tech,ec,area,curtime]/DEM[enduse,tech,ec,area]))
```

Fuel Cost Coefficient

```
@. DFTC = DCTC/(1-DCTC)
```

Normal Capital Cost

```
DCCN[enduse,tech,ec,area] = xDCC[enduse,tech,ec,area,curtime]/
    (DEM[enduse,tech,ec,area]/DEEStd[enduse,tech,ec,area]-1)^
    (1/DCTC[enduse,tech,ec,area,curtime])
```

Normal Fuel Cost

```
DFPN[enduse,tech,ec,area] =
   -(DOCF[enduse,tech,ec,area,curtime]+DCCRN[enduse,tech,ec,area,curtime])*
   DCCN[enduse,tech,ec,area]*DEM[enduse,tech,ec,area]/
   DCTC[enduse,tech,ec,area,curtime]
```

F. Function Initial

This function performs variable initializations for various device, process, and budget parameters. This section is used as an input for developing the process efficiency curves.

The calibrated capacity utilization factor (by ECC) is weighted by its share of economic driver and summed over ECC to yield a weighted capacity utilization factor by EC. The ECCMap maps the ECCs (subclasses such as offices or food processing) into the correct EC (economic class such as commercial or industrial).

Function Initial

Key Inputs

Exogenous Inputs

- xDmd 'Energy Demand (TBtu/Yr)' [Enduse,Tech,EC,Area,Year]
- xCgVF 'Cogen. Variance Factor ($/$)' *[Tech,EC]*
- xDST 'Device Saturation (Btu/Btu)' [Enduse,EC,Area,Year]
- DEM 'Maximum Device Efficiency (Btu/Btu)' [Enduse,Tech,EC,Area]
- DESTD 'Device Efficiency Standards (Btu/Btu)' [Enduse,Tech,EC,Area,Year]
- DESTDP 'Device Efficiency Standards Policy (Btu/Btu)' [Enduse,Tech,EC,Area,Year]
- STX 'Sales Tax Rate on Energy Consumer ($/$)' [Area,Year]
- DDAnnual 'Annual Degree Days (Degree Days)' [Enduse,Area,Year]
- DDAnnualNorm 'Normal Annual Degree Days (Degree Days)' [Enduse,Area]
- DDCoefficient 'Annual Energy Degree Day Coefficient (DD/DD)'[Enduse,EC,Area,Year]
- TSLoad 'Temp. Sensitive Fraction of Load' [Enduse,EC,Area]
- PDIF 'Difference between the Initial Process Efficiency for each Fuel' [Enduse,Tech,EC,Area]
- DCCRN 'Device Capital Charge Rate ($/Yr/$)' [Enduse,Tech,EC,Area]
- DPLV ‘Scrappage Rate of Equipment by Vintage (1/1)’ [Enduse,Tech,EC,Area,Vintage]

Inputs from Economic Sector

- FPC 'Actual Production Capacity by Tech ($/YR)' [Enduse,Tech,EC,Area]
- PCLV 'Production Capacity (M$/Yr)' [Age,ECC,Area,Year]

Inputs from Supply Sector

- PCERFF 'Fraction of Energy Requirement by Age and Fuel' [Fuel,Age,ECC,Area]
- PCgRFF 'Fraction of Energy Requirement by Age' [Age,ECC,Area]

Inputs from DEffCurve Function

- DCCN 'Normalized Device Capital Cost ($/mmBtu)' [Enduse,Tech,EC,Area]
- DCTC 'Device Cap. Trade Off Coefficient (DLESS)' [Enduse,Tech,EC,Area,Year]
- DFTC 'Device Fuel Trade Off Coefficient (DLESS)' [Enduse,Tech,EC,Area,Year]

Key Outputs

Output from Function Initial – Calculated Values

- DCC 'Device Capital Cost ($/mmBtu/Yr)' [Enduse,Tech,EC,Area,Year]
- DEE 'Device Efficiency (Btu/Btu)' [Enduse,Tech,EC,Area,Year]
- DEEA 'Average Device Efficiency (Btu/Btu)' [Enduse,Tech,EC,Area,Year]
- DER 'Energy Requirement (mmBtu/YR) [Enduse,Tech,EC,Area,Year]
- DERV ‘‘Energy Requirement by Vintage (mmBtu/YR) [Enduse,Tech,EC,Area,Vintage,Year]
- PEE 'Process Efficiency ($/Btu)' [Enduse,Tech,EC,Area,Year]
- PER 'Process Energy Requirement (mmBtu/YR)' [Enduse,Tech,EC,Area,Year]
- DEMM 'Maximum Device Efficiency Multiplier (Btu/Btu)' [Enduse,Tech,EC,Area,Year]
- PEEA 'Average Process Efficiency ($/Btu)' [Enduse,Tech,EC,Area,Year]
- AB 'Initial Average Budget ($/$)' [Enduse,Tech,EC,Area,Year]
- EUPC 'Production Capacity by Vintage (M$/Yr)' [Enduse,Tech,Age,EC,Area,Year]

Key Equations

Device Efficiency

Device marginal efficiency (DEE) is a function of maximum efficiency, fuel price (ECFP), and the capital and fuel cost coefficients (DCTC, DFTC).

```
DEE[enduse,tech,ec,area,InitialYear] =
    DEM[enduse,tech,ec,area]*DEMM[enduse,tech,ec,area,InitialYear]*
    (1/(1+(ECFP[enduse,tech,ec,area,InitialYear]/
    Inflation[area,InitialYear]*DEPM[enduse,tech,ec,area,InitialYear]/
    DFPN[enduse,tech,ec,area])^DFTC[enduse,tech,ec,area,InitialYear]))

DEE[enduse,tech,ec,area,InitialYear] = max(DEE[enduse,tech,ec,area,InitialYear],
     DEStd[enduse,tech,ec,area,InitialYear],DEStdP[enduse,tech,ec,area,InitialYear])
```

Special equations can be used as well to match specific curve input data provided for combinations of individual economic sectors and enduses. These equations are applied using the appropriate value of DEESw as set when their input parameters are read in. Please refer to the individual calibration files that set DEESw for more information on the sources for these equations.

Device efficiency is also constrained t to be at least the value of xDEE in the initialization year.

```
DEE[enduse,tech,ec,area,InitialYear] = max(xDEE[enduse,tech,ec,area,InitialYear],
      DEStd[enduse,tech,ec,area,InitialYear],DEStdP[enduse,tech,ec,area,InitialYear])
```

The device efficiency multiplier (DEMM) is used as an adjustment if the efficiency calculation above differs produces a value approaching the maximum (DEM)

```
DEMM[enduse,tech,ec,area,InitialYear] =
    max(DEMM[enduse,tech,ec,area,InitialYear],
    DEE[enduse,tech,ec,area,InitialYear]/(DEM[enduse,tech,ec,area]*0.98))
```

Device Capital Cost

Device marginal capital cost (DCC) is a function of the normal capital cost (DCCN), the level of efficiency (DEE), and the capital cost coefficient (DCTC)

```
DCC[enduse,tech,ec,area,InitialYear] = DCCN[enduse,tech,ec,area]*Inflation[area,InitialYear]*
   (1+STX[area,InitialYear])*(DEM[enduse,tech,ec,area]/DEE[enduse,tech,ec,area,InitialYear]-1)^
   (1/DCTC[enduse,tech,ec,area,InitialYear])
```

Similar to above, special equations can be used to match specific input data sets for cost curves. These are set using the DEESw value.

Weighting the Capacity Utilization Factor by Output

The calibrated capacity utilization factor is mapped from the input data set.

```
WCUF[ec,area] = ECUF[ecc,area,InitialYear]
```

Energy Requirements by End-use

Demand is set equal to exogenous historical demands:

```
Dmd[enduse,tech,ec,area,InitialYear] = xDmd[enduse,tech,ec,area,InitialYear]
```

Device Energy Requirements by End-use are calculated as from the exogenous demands by dividing by the weighted capacity utilization factor and normalizing for weather effects:

```
DER[enduse,tech,ec,area,InitialYear] =
   Dmd[enduse,tech,ec,area,InitialYear]/WCUF[ec,area]*1e6/
   (TSLoad[enduse,ec,area]*(DDay[enduse,area,InitialYear]/
   DDayNorm[enduse,area])^DDCoefficient[enduse,ec,area,InitialYear]+(1-TSLoad[enduse,ec,area]))
```

If there is no initial demand, then set the initial demand to 1/1e12 times the maximum demand (XDmd).

```
DER[enduse,tech,ec,area,InitialYear] = max(DER[enduse,tech,ec,area,InitialYear],
   maximum(xDmd[enduse,tech,ec,area,year] for year in Years)/1e12)
```

Energy requirements by vintage (DERV) are initialized using the initial demand and aged over time using the scrappage rate (DPLV).

```
DERV[enduse,tech,ec,area,1,InitialYear] = DER[enduse,tech,ec,area,InitialYear]
DERV[enduse,tech,ec,area,vintage,InitialYear] = DERV[enduse,tech,ec,area,vintageprior,InitialYear]*
                              (1-DPLV[enduse,tech,ec,area,vintage,InitialYear])
```

The accumulated value is collected and used to weight energy requirements by vintage so that the sum across vintage equals the DER total.

```
DERVSum[enduse,tech,ec,area,InitialYear] = sum(DERV[enduse,tech,ec,area,vintage,InitialYear] for vintage in
      Vintages)
DERVAllocation[enduse,tech,ec,area,vintage,InitialYear] = DERV[enduse,tech,ec,area,vintage,InitialYear]/
                                        DERVSum[enduse,tech,ec,area,InitialYear]
DERV[enduse,tech,ec,area,vintage,InitialYear] = DER[enduse,tech,ec,area,InitialYear]*
                              DERVAllocation[enduse,tech,ec,area,vintage,InitialYear]
DERVSum[enduse,tech,ec,area,InitialYear] = sum(DERV[enduse,tech,ec,area,vintage,InitialYear] for vintage in
      Vintages)
```

The initial average efficiency (DEEA) is set equal to the initial marginal efficiency (DEE):

```
DEEA[enduse,tech,ec,area,InitialYear] = DEE[enduse,tech,ec,area,InitialYear]
```

Process Energy Requirements of Capital Stock

The process energy requirements of capital stock are equal to the device requirements times the average device efficiency:

```
PER[enduse,tech,ec,area,InitialYear] = DER[enduse,tech,ec,area,InitialYear]*DEEA[enduse,tech,ec,area,InitialYear]
```

Saturation

The initial device saturation is set equal to its exogenous value:

```
DSt[enduse,ec,area,InitialYear] = xDSt[enduse,ec,area,InitialYear]
```

Calculate indicated EUPC as FPC

Actual Production Capacity by Technology ($) is equal to the process energy requirements by technology (MBTU) divided by the device saturation (BTU/BTU) times the difference in process efficiency between technologies (PDIF).

```
FPC[enduse,tech,ec,area] =
    PER[enduse,tech,ec,area,InitialYear]/DSt[enduse,ec,area,InitialYear]*PDif[enduse,tech,ec,area]
```

FPC is allocated fractionally to sum to the initial production capacity (PCLVI (EUPC)):

First total actual production capacity is calculated by summing the actual production capacity across technologies:

```
TFPC[enduse,ec,area] = sum(FPC[enduse,tech,ec,area] for tech in Techs)
```

The desired production capacity (FPCI) is derived from the initial production capacity (PCLV) by splitting PCLV by the ratio of a given technology's production capacity to the total capacity (FPC/TFPC).

```
FPCI[enduse,tech,ec,area] = sum(PCLV[age,ecc,area,InitialYear] for age in Ages)*
    FPC[enduse,tech,ec,area]/TFPC[enduse,ec,area]
```

Calculate Process Efficiency

Average process efficiency is calculated by multiplying the production capacity by the device saturation and dividing by the energy requirements:

```
PEEA[enduse,tech,ec,area,InitialYear] = FPCI[enduse,tech,ec,area]*DSt[enduse,ec,area,InitialYear]/
    PER[enduse,tech,ec,area,InitialYear]
```

For the initial year the average efficiency is also the marginal efficiency:

```
PEE[enduse,tech,ec,area,InitialYear] = PEEA[enduse,tech,ec,area,InitialYear]
```

Split Capital Stock by Age

This routine uses the market share for each age category (PCERF) to split the capital stock (FPCI) into capital stock by age and fuel (EUPCI) for each end-use.

The first step is to use the fuel tech fraction to select the most representative fuel for each technology.

```
for fuel in Fuels
  FuelSort[fuel] = xDmFrac[enduse,fuel,tech,ec,area,InitialYear]
end
fuels_sorted = fuels[sortperm(FuelSort[fuels],rev = true)]
HighestFuel = fuels_sorted[1]
PCERF[tech,age,ec,area] = PCERFF[HighestFuel,age,ecc,area]
```

Re-normalize PCERF to be consistent with PCGRF.

```
PCERFI[tech,age,ec,area] = PCERF[tech,age,ec,area]
```

Develop a weighted sum of PCERF (LOC3):

```
Loc3[age,ec,area] = sum(PCERF[tech,age,ec,area]*FPCI[enduse,tech,ec,area] for tech in Techs)/
      sum(FPCI[enduse,tech,ec,area] for tech in Techs)
```

Scale to match PCERG (growth rate):

```
PCERF[tech,age,ec,area] = PCERF[tech,age,ec,area]*
      PCgRF[age,ec,area]/Loc3[age,ec,area]
```

Move residual to next Age group (AgeNext). Since AgeNext is defined as Age-1, this moves the residual into the Mid category from the Old category so that it works with large negative growth:

```
PCERF[tech,AgeNext,ec,area] = PCERF[tech,AgeNext,ec,area]+
      PCERFI[tech,age,ec,area]-PCERF[tech,age,ec,area]
```

PCERF[tech,AgeNext,ec,area] = max(PCERF[tech,AgeNext,ec,area],0.001)

Normalize PCERF to sum to 1.0:

PCERF[tech,age,ec,area] = PCERF[tech,age,ec,area]/sum(PCERF[tech,a,ec,area] for a in Ages)

Production capacity by end-use (EUPC) is then the sum over fuel and economic subclass of the product of the fraction of energy requirement by Age and Fuel and the desired production capacity. The desired production capacity is allocated by technology, age and economic class in EUPC.

EUPC[enduse,tech,age,ec,area,InitialYear] =PCERF[tech,age,ec,area]*FPCI[enduse,tech,ec,area]

Cooling to Heating Efficiency Ratio

If there is no Heat, skip the following equation for CHR. The cooling to heating efficiency ratio is the ratio of the heating compared to AC.

CHR[ec,area] = (sum(PEE[Heat,tech,ec,area,InitialYear]*PER[Heat,tech,ec,area,InitialYear] for tech in Techs)/
sum(PER[Heat,tech,ec,area,InitialYear] for tech in Techs))/(sum(PEE[AC,tech,ec,area,InitialYear]*
PER[AC,tech,ec,area,InitialYear] for tech in Techs)/sum(PER[AC,tech,ec,area,InitialYear] for tech in Techs))

Initialize Average Budget

AB[enduse,tech,ec,area,InitialYear] = ECFP[enduse,tech,ec,area,InitialYear]/Inflation[area,InitialYear]*
DSt[enduse,ec,area,InitialYear]/ (PEEA[enduse,tech,ec,area,InitialYear]*DEEA[enduse,tech,ec,area,InitialYear])

Initialize Self-generation

CgVF[tech,ec,area] = xCgVF[tech,ec]
CgUMS[tech,ec,area,InitialYear] = 1.0

G. Function PEffCurve

This function derives the process Consumer Preference Efficiency and Capital Cost Curves based on the input data provided for the single initialization year.

Function PEffCurve

Key Inputs

Exogenous Inputs

- xPCC 'Process Capital Cost ($/($/Yr))' [Enduse,Tech,EC,Area,Year]
- PEMX 'Ratio of Maximum to Average Process Efficiency ($/Btu/($/Btu))' [Enduse,Tech,EC,Area]
- PIVTC 'Process Investment Tax Credit (DLESS)' [Year]

Inputs from Function Lifetimes

- PEPL 'Physical Life of Process Energy Requirements (Years)' [Enduse,Tech,EC,Area,Year]
- PETL 'Tax Life of Process Energy Requirements (Years)'[Enduse,Tech,EC,Area,Year]

Inputs from Function IPrice

- MCFU 'Marginal Cost of Fuel Use ($/mmBtu)' [Enduse,Tech,EC,Area,Year]

Inputs from Function Initial

- PEE 'Process Efficiency ($/Btu)' [Enduse,Tech,EC,Area,Year]
- PEEA 'Average Process Efficiency ($/Btu)' [Enduse,Tech,EC,Area,Year]

Key Outputs

Output from Function Initial – Calculated Values

- PCCRN 'Process Capital Charge Rate' [Enduse,Tech,EC,Area]
- POCF 'Process Operating Cost Fraction' [Enduse,Tech,EC,Area]

- PEM 'Maximum Process Efficiency ($/mmBtu)' [Enduse,EC,Area]
- PFPN 'Process Normalized Fuel Price ($/mmBtu)' [Enduse,Tech,EC,Area]
- PCCN 'Normalized Process Capital Cost ($/mmBtu)' [Enduse,Tech,EC,Area]
- PCTC 'Process Capital Cap. Trade Off Coef. (DLESS)' [Enduse,Tech,EC,Area,Year]
- PFTC 'Process Fuel Trade Off Coefficient' [Enduse,Tech,EC,Area,Year]

Key Equations

Cost of Using a Device

Each specific demand for energy is associated with a stock of capital. Investment in each type of capital stock by fuel type is allocated according to the cost of using each type of fuel. This cost is the perceived cost to the user and includes a risk factor (incorporated in the calculation of DCCRN), annualized capital costs (DCCRN*XDCC), and delivered marginal fuel costs (ECFP/XDEE).

The marginal cost of using energy (MCFU) includes the cost of using energy for all end-uses. As such, a house that has a gas furnace but an electric water heater would be represented partially in the model’s gas capital stock and partially in the electric capital stock. The investment includes capital using energy in addition to the energy source equipment.

```
MCFU[enduse,tech,ec,area,InitialYear] = (DCCRN[enduse,tech,ec,area,InitialYear]+
    DOCF[enduse,tech,ec,area,InitialYear])*DCC[enduse,tech,ec,area,InitialYear]+
    ECFP[enduse,tech,ec,area,InitialYear]/DEE[enduse,tech,ec,area,InitialYear]
```

Capital Charge Rate

```
PCCRN[enduse,tech,ec,area,Zero] =
   (1-PIVTC[InitialYear]/(1+ROIN[ec,area]+0+InSm[area,InitialYear])-
   TxRt[ec,area,InitialYear]*(2/PETL[enduse,tech,ec,area,InitialYear])/
   (ROIN[ec,area]+0+InSm[area,InitialYear]+
   2/PETL[enduse,tech,ec,area,InitialYear]))*(ROIN[ec,area]+0)/
   (1-(1/(1+ROIN[ec,area]+0))^PEPL[enduse,tech,ec,area,InitialYear])/
   (1-TxRt[ec,area,InitialYear])
```

Operating Cost Factor

```
POCF[enduse,tech,ec,area] = (Inflation[area,InitialYear]-MCFU[enduse,tech,ec,area,InitialYear]/1000000/
   PEEA[enduse,tech,ec,area,InitialYear])/xPCC[enduse,tech,ec,area,InitialYear]-PCCRN[enduse,tech,ec,area,Zero]
```

Special Process Efficiency Curves

Equations for process efficiency for specific sectors and enduses are used depending on the value of PEESw to match external data sources. Please refer to calibration files for sources for curve equations.

Maximum Process Efficiency

Maximum Process Efficiency is based on the highest fuel.

```
PEM[enduse,ec,area] = maximum(PEEA[enduse,tech,ec,area,InitialYear]*
   PEMX[enduse,tech,ec,area] for tech in Techs)
```

Process Fuel and Capital Cost Coefficients

A constraint for capital cost coefficients is currently used for Commercial and Transportation to correct for irregular historical data

```
PCTC[enduse,tech,ec,area,InitialYear] = -1/((MCFU[enduse,tech,ec,area,InitialYear]/1000000/
    PEEA[enduse,tech,ec,area,InitialYear])/((PCCRN[enduse,tech,ec,area,Zero]+POCF[enduse,tech,ec,area])*
    xPCC[enduse,tech,ec,area,InitialYear])*(1-PEEA[enduse,tech,ec,area,InitialYear]/PEM[enduse,ec,area]))

if (ESKey == "Commercial") || (ESKey == "Transportation")
    PCTC[enduse,tech,ec,area,InitialYear] = min(max(PCTC[enduse,tech,ec,area,InitialYear],-200),-5)
```

```
PFTC[enduse,tech,ec,area,InitialYear] = PCTC[enduse,tech,ec,area,InitialYear]/
    (1-PCTC[enduse,tech,ec,area,InitialYear])
```

Process Normal Fuel Cost

```
PFPN[enduse,tech,ec,area] = -(PCCRN[enduse,tech,ec,area,Zero]+POCF[enduse,tech,ec,area])*
   xPCC[enduse,tech,ec,area,InitialYear]*PEM[enduse,ec,area]*1000000/
   (PCTC[enduse,tech,ec,area,InitialYear]*(PEM[enduse,ec,area]/
   PEE[enduse,tech,ec,area,InitialYear]-1)^ (1/PCTC[enduse,tech,ec,area,InitialYear]))
```

Normal Process Capital Cost

For the first enduse only:

```
PCCN[enduse,tech,ec,area] = abs(xPCC[enduse,tech,ec,area,InitialYear]/(PEM[enduse,ec,area] /
      PEE[enduse,tech,ec,area,InitialYear]-1)^ (1/PCTC[enduse,tech,ec,area,InitialYear]))
```

otherwise:

```
PCCN[enduse,tech,ec,area] = 0
PCTC[enduse,tech,ec,area,InitialYear] = 0
PFTC[enduse,tech,ec,area,InitialYear] = 0
```

Initialize Process Capital Cost

```
PCC[enduse,tech,ec,area,InitialYear] = xPCC[enduse,tech,ec,area,InitialYear]
```

H. Function PEffAdjust

This function was developed to adjust the process initialization variables after calculation if the value for Driver in a given sector or area is very small or zero. This allows for the process cost and efficiency curves to operate normally for sectors that might not exist until after the initial year. The average coefficients area applied in the event that the driver is very small or zero.

Function PEffAdjust

Key Inputs

Inputs from Macroeconomic Sector

- Driver 'Economic Driver (Various Millions/Yr)' [ECC,Area]

Inputs from Function PEffCurve

- PCCN 'Normalized Process Capital Cost ($/mmBtu)' [Enduse,Tech,EC,Area]
- PCTC 'Process Capital Cap. Trade Off Coef. (DLESS)' [Enduse,Tech,EC,Area,Year]
- PCCRN 'Process Capital Charge Rate' [Enduse,Tech,EC,Area]
- POCF 'Process Operating Cost Fraction' [Enduse,Tech,EC,Area]
- PEM 'Maximum Process Efficiency ($/mmBtu)' [Enduse,EC,Area]
- PFPN 'Process Normalized Fuel Price ($/mmBtu)' [Enduse,Tech,EC,Area]

Key Outputs

Calculated Outputs

- PCCN 'Normalized Process Capital Cost ($/mmBtu)' [Enduse,Tech,EC,Area]
- PCTC 'Process Capital Cap. Trade Off Coef. (DLESS)' [Enduse,Tech,EC,Area,Year]
- PCCRN 'Process Capital Charge Rate' [Enduse,Tech,EC,Area]
- POCF 'Process Operating Cost Fraction' [Enduse,Tech,EC,Area]

- PEM 'Maximum Process Efficiency ($/mmBtu)' [Enduse,EC,Area]
- PFPN 'Process Normalized Fuel Price ($/mmBtu)' [Enduse,Tech,EC,Area]

Key Equations

The average coefficient of all areas is weighted based on the driver for each area over the driver of all areas.

```
AvPEM[enduse,ec] = sum(PEM[enduse,ec,area]*Driver[ecc,area,InitialYear] for area in Areas)/
    sum(Driver[ecc,area,InitialYear] for area in Areas)
AvPOCF[enduse,tech,ec] = sum(POCF[enduse,tech,ec,area]*Driver[ecc,area,InitialYear] for area in Areas)/
    sum(Driver[ecc,area,InitialYear] for area in Areas)
AvPCTC[enduse,tech,ec] = sum(PCTC[enduse,tech,ec,area,InitialYear]*
    Driver[ecc,area,InitialYear] for area in Areas)/sum(Driver[ecc,area,InitialYear] for area in Areas)
AvPFPN[enduse,tech,ec] = sum(PFPN[enduse,tech,ec,area]*Driver[ecc,area,InitialYear] for area in Areas)/
    sum(Driver[ecc,area,InitialYear] for area in Areas)
AvPCCN[enduse,tech,ec] = sum(PCCN[enduse,tech,ec,area]*Driver[ecc,area,InitialYear] for area in Areas)/
    sum(Driver[ecc,area,InitialYear] for area in Areas)
```

The average coefficients area applied in the event that the driver is very small or zero.

```
Select Area if Driver lt 0.0001
Do If Driver lt 0.0001
 PEM[enduse,ec,area] = AvPEM[enduse,ec]
 POCF[enduse,tech,ec,area] = AvPOCF[enduse,tech,ec]
 PCTC[enduse,tech,ec,area,InitialYear] = AvPCTC[enduse,tech,ec]

 PFTC[enduse,tech,ec,area,InitialYear] = PCTC[enduse,tech,ec,area,InitialYear]/
     (1-PCTC[enduse,tech,ec,area,InitialYear])

 PFPN[enduse,tech,ec,area] = AvPFPN[enduse,tech,ec]
 PCCN[enduse,tech,ec,area] = AvPCCN[enduse,tech,ec]
```

I. Function PEffFuture

This function simply sets the process and retrofit fuel and capital cost variables for all years equal to the value in the initial year.

Function PEffFuture

Key Outputs

Calculated Outputs

- PEECurveM ‘Process Efficiency from Cost Curve Multiplier(1/1)’ [Enduse,Tech,EC,Area,Year]
- PCTC 'Process Capital Cap. Trade Off Coef. (DLESS)' [Enduse,Tech,EC,Area,Year]
- PFTC 'Process Fuel Trade Off Coefficient' [Enduse,Tech,EC,Area,Year]
- RPCTC 'Retrofit Process Capital Trade Off Coefficient (DLESS)' [Enduse,Tech,EC,Area,Year]
- RPFTC 'Retrofit Process Fuel Trade Off Coefficient' [Enduse,Tech,EC,Area,Year]

Key Equations

This function simply sets the process and retrofit fuel and capital cost variables for all years equal to the value in the initial year.

PEECurveM[enduse,tech,ec,area,year] = PEECurveM[enduse,tech,ec,area,InitialYear] PCTC[enduse,tech,ec,area,year] = PCTC[enduse,tech,ec,area,InitialYear] PFTC[enduse,tech,ec,area,year] = PFTC[enduse,tech,ec,area,InitialYear] RPCTC[enduse,tech,ec,area,year] = PCTC[enduse,tech,ec,area,InitialYear] RPFTC[enduse,tech,ec,area,year] = PFTC[enduse,tech,ec,area,InitialYear]

J. Function Pollution

This function initializes the pollution variables by setting average coefficients equal to the marginal exogenous pollution coefficients.

Function Pollution
Key Inputs
Exogenous Inputs • POCX 'Marginal Pollution Coefficients (Tonnes/TBtu)' [Enduse,FuelEP,EC,Poll,Area,Year]
Key Outputs
Assigned Outputs (for initial year) • POCA ‘Pollution Coefficients (Tonnes/TBtu)' [Enduse,FuelEP,EC,Poll,Area,Year]
Key Equations
Embodied Pollution The average pollution coefficient is set equal to the exogenous marginal value: POCA[enduse,fuelep,ec,poll,area,Zero] = POCX[enduse,fuelep,ec,poll,area,Zero] POCA=POCX